



**AHŞAP KUSUR TESPİTİ İÇİN BİLGİSAYAR GÖRÜŞÜ TABANLI  
MAKİNE ÖĞRENİMİ YAKLAŞIMI**

**Kenan KILIÇ**

**DOKTORA TEZİ  
AĞAÇIŞLERİ ENDÜSTRİ MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**MART 2025**

## ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
  - Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
  - Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
  - Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
  - Bu tezde sunduğum çalışmanın özgün olduğunu,
- bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Kenan KILIÇ

03/03/2025

# AHŞAP KUSUR TESPİTİ İÇİN BİLGİSAYAR GÖRÜŞÜ TABANLI MAKİNE ÖĞRENİMİ YAKLAŞIMI

(Doktora Tezi)

Kenan KILIÇ

GAZİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

Mart 2025

## ÖZET

Dünya çapındaki ağaçişleri endüstrilerinde kalite ve üretim süreçlerinde önemli bir konu olan ahşap kusur tespiti üzerine odaklanmaktadır. Farklı bilgisayarla görü tabanlı yaklaşımlar kullanılarak ahşap kusurlarının tespiti gerçekleştirilmiştir. İlk olarak, optimize edilmiş ESA (Evrışimsel Sinir Ağı) mimarileri ve transfer öğrenme yaklaşımı ile ahşap kusur tespiti yapılmıştır. İkinci olarak, öznitelik çıkarma yöntemleri ve klasik makine öğrenme sınıflandırma algoritmaları ile kusur tespiti gerçekleştirilmiştir. Üçüncü olarak ise, optimize edilmiş ESA mimarisi ile öznitelik çıkarılmış ve klasik makine öğrenme sınıflandırma algoritmaları ile sınıflandırma yapılmış hibrit bir yaklaşım kullanılmıştır. Birinci aşamada, optimize edilmiş ESA mimarileri olan VGG-16, MobileNet, DenseNet-121, ResNet-50, Xception ve InceptionV3 kullanılmıştır. Ayrıca, bu mimarilere en uygun optimizasyon algoritmasını belirlemek için farklı optimizasyon algoritmaları test edilmiş ve farklı nöron sayılarının (256, 512, 1024, 2048) gösterdiği performanslar belirlenmiştir. İkinci aşamada, öznitelik çıkarma yöntemleri olarak LBP (Local Binary Patterns), HOG (Histogram of Oriented Gradients), GLCM (Gray-Level Co-occurrence Matrix), GABOR, SOBEL ve MB-LBP kullanılmış ve klasik makine öğrenme sınıflandırma algoritmaları olarak Random Forest, K-Nearest Neighbors, Support Vector Machine, Decision Tree, Naive Bayes, Logistic Regression, Gradient Boosting, XGBoost, LightGBM, CatBoost, AdaBoost, ve MLP (Multilayer Perceptron) performansları değerlendirilmiştir. Üçüncü aşamada ise, ESA ile öznitelik çıkarımı yapıp, bu öznitelikler klasik makine öğrenme sınıflandırma algoritmaları ile sınıflandırılmıştır. Hibrit yaklaşım ile %99,32 en yüksek doğruluk oranı, optimize edilmiş ESA mimarileri ile %97,57 doğruluk ve öznitelik çıkarımı ve klasik makine öğrenme algoritmaları ile kusur sınıflamada %96,75 doğruluk oranı elde edilmiştir. Sonuç olarak, bu çalışmada bilgisayarlı görü tabanlı ahşap kusur tespiti yöntemleri değerlendirilmiş ve ahşap malzemelerde yüksek doğruluk oranlarıyla kusur tespiti yapılabilirliği gösterilmiştir. Bu yaklaşımların ağaçişleri endüstrisinde kullanılması, üretim süreçlerini hızlandırabilir, malzeme israfını önleyebilir ve genel olarak endüstriyel uygulamalarda daha etkili bir üretim sağlanabilir.

Bilim Kodu : 121601

Anahtar Kelimeler : Bilgisayar görüşü, yapay zeka, görüntü işleme, görüntü analizi, sınıflandırma, ahşap kusurları, ağaçişleri endüstri mühendisliği

Sayfa Adedi : 189

Danışman : Prof. Dr. Uğur ÖZCAN

# A COMPUTER VISION-BASED MACHINE LEARNING APPROACH FOR WOOD DEFECT DETECTION

(Ph. D. Thesis)

Kenan KILIÇ

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

March 2025

## ABSTRACT

It focuses on wood defect detection, an important issue in quality and manufacturing processes in woodworking industries worldwide. Wood defects are detected using different computer vision-based approaches. First, wood defect detection is performed using optimized CNN (Convolutional Neural Network) architectures and transfer learning approach. Second, defect detection is performed using feature extraction methods and classical machine learning classification algorithms. Third, a hybrid approach is used in which features are extracted with optimized CNN architecture and classification is performed using classical machine learning classification algorithms. In the first stage, optimized CNN architectures VGG-16, MobileNet, DenseNet-121, ResNet-50, Xception and InceptionV3 are used. Furthermore, different optimization algorithms are tested to determine the most suitable optimization algorithm for these architectures and the performances of different neuron numbers (256, 512, 1024, 2048) are determined. In the second stage, LBP (Local Binary Patterns), HOG (Histogram of Oriented Gradients), GLCM (Gray-Level Co-occurrence Matrix), GABOR, SOBEL and MB-LBP were used as feature extraction methods and the performances of Random Forest, K-Nearest Neighbors, Support Vector Machine, Decision Tree, Naive Bayes, Logistic Regression, Gradient Boosting, XGBoost, LightGBM, CatBoost, AdaBoost, and MLP (Multilayer Perceptron) as classical machine learning classification algorithms were evaluated. In the third stage, feature extraction was performed with CNN and these features were classified with classical machine learning classification algorithms. The highest accuracy rate of 99.32% was obtained with the hybrid approach, 97.57% accuracy with optimized CNN architectures and 96.75% accuracy rate in feature extraction and defect classification with classical machine learning algorithms. As a result, in this study, computer vision-based wood defect detection methods were evaluated and it was shown that defect detection in wood materials with high accuracy rates can be done. The use of these approaches in the woodworking industry can speed up production processes, prevent material waste and provide more effective production in industrial applications in general.

Science Code : 121601

Key Words : Computer vision, artificial intelligence, image processing, image analysis, classification, wood defects, wood products engineering

Page Number : 189

Supervisor : Prof. Dr. Uğur ÖZCAN

## TEŞEKKÜR

Doktora eğitimim süresi boyunca değerli katkılarıyla destek olan, tecrübelerini aktaran ve tüm süreçte motivasyonumu yüksek tutan danışmanın Prof. Dr. Uğur ÖZCAN'a, tüm doktora süresince desteklerini ve bilgilerini paylaşan değerli öneri ve yönlendirmeleri ile bu tezin kalitesini artırmada ve içeriğini zenginleştirmede yardımcı olan değerli arkadaşım Öğr. Gör. Kazım KILIÇ'a, tez izleme komitesinde yer alan ve destekleriyle katkı sunan Prof. Dr. Musa ATAR'a, Prof. Dr. İbrahim ALPER Doğru'ya, savunma jüri üyelerim Prof. Dr. Diyar AKAY ve Doç. Dr. Mehmet ŞİMŞEK' e çalışma ve eğitim hayatım boyunca beni destekleyen eşim Tuğba KILIÇ'a ve aileme teşekkür ederim.

## İÇİNDEKİLER

	<b>Sayfa</b>
ÖZET .....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER.....	vii
ÇİZELGELERİN LİSTESİ.....	xii
ŞEKİLLERİN LİSTESİ.....	xiii
RESİMLERİN LİSTESİ.....	xviii
SİMGELER VE KISALTMALAR.....	xix
1. GİRİŞ.....	1
2. AHŞAP KUSURLARI.....	15
2.1. Ağaç Kusurları .....	15
2.2. Ağaç Hastalıkları.....	17
3. LİTERATÜR ÖZETİ .....	21
3.1. Klasik Makine Öğrenimi ile Ahşap Kusur Tespiti.....	22
3.2. Derin Öğrenme ile Ahşap Kusur Tespiti.....	25
4. GÖRÜNTÜDEN ÖZNİTELİK ÇIKARIMI.....	29
4.1. Yerel İkili Örüntüler Operatörü (Local Binary Pattern) .....	29
4.2. Yönlü Gradyanlar Histogramı (Histogram of Oriented Gradients) .....	31
4.3. Gri Seviye Eş-Oluşum Matrisi (Gray-Level Co-occurrence Matrix) .....	32
4.4. Sobel.....	34
4.5. Gabor.....	35
4.6. Çok Bloklü Yerel İkili Örüntü (Multi Block LBP / MB-LBP).....	36
5. MAKİNE ÖĞRENMESİ.....	39

	<b>Sayfa</b>
5.1. Denetimli Öğrenme .....	42
5.1.1. Sınıflandırma.....	43
5.1.2. Regresyon.....	53
5.2. Denetimsiz Öğrenme.....	57
5.2.1. Kümeleme .....	58
5.2.2. İlişkilendirme .....	60
5.2.3. Boyut Azaltma .....	60
5.3. Takviyeli Öğrenme.....	61
6. DERİN ÖĞRENME .....	63
6.1. Yapay Sinir Ağları .....	64
6.1.1 Yapay sinir ağı yapısı.....	65
6.1.2. İleri beslemeli ağ modeli.....	66
6.1.3. Geri beslemeli ağ modeli .....	67
6.2. Derin Öğrenme Modelleri .....	67
6.2.1. Otomatik kodlayıcılar.....	67
6.2.2. Kısıtlı boltzman makineleri (Restricted boltzman machines - RBM).....	68
6.2.3. Derin inanç ağları (Deep belief network – DBN) .....	69
6.2.4. Tekrarlayan sinir ağları (Recurrent neural networks - RNN) .....	70
6.2.5. Uzun kısa hafıza ağları (Long-short term memory - LSTM).....	71
6.2.6. Üretken çekişmeli ağlar (Generative adversarial networks - GAN) .....	71
6.3. Evrişimsel Sinir Ağları (Convolutional neural network - ESA) .....	73
6.3.1. Evrişim katmanı .....	74
6.3.2. Aktivasyon katmanı .....	75
6.3.3. Havuzlama katmanı.....	75

	<b>Sayfa</b>
6.3.4. Düzleştirme katmanı .....	76
6.3.5. Tam bağlı katman.....	77
6.3.6 Dropout (Bırakma) katmanı .....	78
6.3.7. Sınıflandırma katmanı .....	78
6.3.8. Yumuşatma (Softmax) katmanı .....	79
6.3.9. Normalizasyon katmanı .....	79
6.4. Optimizasyon Algoritmaları.....	79
6.4.1. Adam.....	80
6.4.2. AdaGrad .....	81
6.4.3. Adadelta .....	81
6.4.4. SGD.....	82
6.4.5. RMSprop.....	83
6.5. Aktivasyon Fonksiyonları .....	84
6.5.1. Basamak (step) fonksiyonu .....	84
6.5.2. Doğrusal (linear) fonksiyon .....	85
6.5.3. Sigmoid .....	86
6.5.4. Tanh .....	87
6.5.5. ReLU .....	87
6.5.6. Sızdıran ReLU.....	88
6.5.7. Üstel doğrusal birim (Exponential linear unit - ELU) .....	90
6.5.8. Swish (A self-gated/kendinden geçitli) fonksiyonu.....	90
6.5.9. Softmax .....	91
6.6. Evrimsel Sinir Ağı Mimarileri.....	92
6.5.1 VGGNet .....	92

	<b>Sayfa</b>
6.5.2. MobileNet .....	93
6.5.3. ResNet .....	94
6.5.4. DenseNet .....	96
6.5.5. Inception .....	97
6.5.6. Xception .....	98
6.6. Transfer Öğrenme .....	99
<b>7. MATERYAL VE YÖNTEM .....</b>	<b>103</b>
7.1. Veri Seti .....	103
7.2. Uygulama Ortamı ve Kütüphaneler .....	105
7.3. Yöntem .....	109
7.3.1. Veri artırma ve ön işlemler .....	109
7.3.2. Öznitelik çıkarımı ve seçimi .....	110
7.3.3. Derin öğrenme mimarileri ile sınıflandırmada kullanılan parametreler ...	114
7.3.4. Klasik makine öğrenmesi sınıflandırma algoritmaları parametreleri.....	119
7.4. Deneysel Kurulum .....	127
7.5. Sınıflandırma .....	128
7.5.1. Derin öğrenme mimarileri ile ahşap yüzey kusur tespiti .....	128
7.5.2. Klasik makine öğrenme yöntemleriyle ahşap yüzey kusur tespiti .....	130
7.5.3. Hibrit yaklaşım ile ahşap yüzey kusur tespiti .....	131
7.6. Hiper parametre .....	131
7.7. Çapraz Doğrulama (Cross-Validation) .....	132
7.8. Değerlendirme Metrikleri .....	133
<b>8. BULGULAR VE TARTIŞMA .....</b>	<b>135</b>
8.1. Derin Öğrenme Mimarileri ile Ahşap Yüzey Kusur Tespiti .....	135

	<b>Sayfa</b>
8.2. Klasik Makine Öğrenme Yöntemleri ile Ahşap Yüzey Kusur Tespiti .....	141
8.3. Hibrit Yaklaşım ile Ahşap Yüzey Kusur Tespiti .....	153
9. SONUÇ VE ÖNERİLER .....	161
KAYNAKLAR .....	171
ÖZGEÇMİŞ .....	187

## ÇİZELGELERİN LİSTESİ

<b>Çizelge</b>	<b>Sayfa</b>
Çizelge 7.1. Klasik makine öğrenmede denemelerde elde edilen en iyi hiper parametreler .....	127
Çizelge 8.1. ESA mimarilerinin ve optimizasyon algoritmalarının en başarılı sonuçları.....	136
Çizelge 8.2. ESA mimarileri, RMSprop optimizasyon algoritması ve nöron sayısı performansları.....	139
Çizelge 8.3. LBP öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları .	142
Çizelge 8.4. HOG öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları.....	143
Çizelge 8.5. GLCM öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları.....	144
Çizelge 8.6. SOBEL öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları.....	145
Çizelge 8.7. GABOR öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları.....	146
Çizelge 8.8. MB-LBP öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları.....	147
Çizelge 8.9. Hibrit öznitelikler ile sınıflandırma algoritmalarının performansları .....	149
Çizelge 8.10. Hibrit yaklaşım ile sınıflandırma algoritmalarının performansları.....	154

## ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1. Klasik makine öğrenme yöntemleri ile ahşap kusur tespiti işlem basamakları.....	4
Şekil 1.2. Derin öğrenme yöntemler ile ahşap kusur tespiti işlem basamakları .....	5
Şekil 1.3. Hibrit öğrenme yaklaşımı ile ahşap kusur tespiti işlem basamakları .....	5
Şekil 4.1. Çeşitli dairesel LBP operatörler.....	29
Şekil 4.2. LBP operatörü şeması örneği.....	30
Şekil 4.3. Ahşap tanımada HOG ile özellik çıkarımı örneği .....	32
Şekil 4.4. GLCM hesaplama örneği .....	34
Şekil 4.5. Sobel filtre uygulama örneği .....	35
Şekil 4.6. Çeşitli giriş görüntüleri için Gabor filtreleme ve kenar algılama çıktıları ....	36
Şekil 4.7. MB-LBP ile görüntünün değerini elde etme örneği .....	37
Şekil 5.1. Makine öğrenmesi algoritmaları hiyerarşik yapısı .....	40
Şekil 5.2. Klasik makine öğrenimi yöntemleriyle görüntü sınıflandırma genel yöntem .....	40
Şekil 5.3. Denetimli makine öğrenmesi şematik gösterimi. ....	42
Şekil 5.4. Destek vektör makinesi şematik gösterimi .....	44
Şekil 5.5. Naive Bayes şematik gösterimi örneği .....	45
Şekil 5.6. Karar ağaçları şematik gösterimi .....	46
Şekil 5.7. Rastgele orman algoritması örnek gösterimi .....	47
Şekil 5.8. Lojistik regresyon dağılım gösterim örneği.....	47
Şekil 5.9. K-En yakın komşu şematik gösterimi.....	48
Şekil 5.10. Yapay sinir ağları şematik gösterimi .....	49
Şekil 5.11. Gradient boosting çalışma prensibi şematik gösterimi.....	50
Şekil 5.12. XGB genel yapısı.....	51

<b>Şekil</b>	<b>Sayfa</b>
Şekil 5.13. LightGBM şematik gösterim örneği .....	51
Şekil 5.14. CatBoost algoritması çalışma prensibi örneği .....	52
Şekil 5.15. AdaBoost algoritması şematik gösterimi.....	53
Şekil 5.16. Doğrusal regresyon grafik örneği .....	54
Şekil 5.17. Doğrusal olmayan regresyon modeli gösterimi.....	55
Şekil 5.18. Regresyon ağaçları tahmin mantığı şeması gösterimi .....	56
Şekil 5.19. Polinom regresyon dağılım örnek gösterimi.....	56
Şekil 5.20. Bayesçi doğrusal regresyon şematik gösterimi.....	57
Şekil 5.21. Denetimsiz makine öğrenmesi örnek şematik gösterimi .....	58
Şekil 5.22. Kümeleme algoritması örneği.....	59
Şekil 5.23. Birliktelik kurallı öğrenme modeli örneği .....	60
Şekil 6.1. Derin ile görüntü sınıflandırma şematik gösterimi.....	63
Şekil 6.2. Gerçek sinir ağı modeli: (a) insan sinir hücresi, (b) yapay sinir hücresi, (c) biyolojik sinaps, (d) YSA sinapsları.....	64
Şekil 6.3. YSA sinir hücresinin şematik gösterimi .....	66
Şekil 6.4. Yapay sinir ağının bir modeli .....	66
Şekil 6.5. Otomatik kodlayıcı ağ yapısı .....	68
Şekil 6.6. Kısıtlı Boltzman makine modeli.....	69
Şekil 6. 7. Üç katmanlı DBN modeli örneği .....	69
Şekil 6.8. Tekrarlayan sinir ağı modeli .....	70
Şekil 6.9. LSTM ağı genel katman yapısı.....	71
Şekil 6.10. GAN modeli.....	72
Şekil 6.11. ESA modeli genel yapısı şematik gösterimi.....	74
Şekil 6.12. Evrişim süreci işlemi .....	75
Şekil 6.13. Şekil havuzlama katmanları şematik gösterimi .....	76

<b>Şekil</b>	<b>Sayfa</b>
Şekil 6.14. Düzleştirme işlemine ait süreç .....	77
Şekil 6.15. Tam bağlı katman yapısı örneği .....	77
Şekil 6.16. Yapay sinir ağı (a), Dropout uygulanmış sinir ağı (b).....	78
Şekil 6.17. Basamak fonksiyonu ve türevi .....	85
Şekil 6.18. Doğrusal fonksiyon ve türevi.....	85
Şekil 6.19. Sigmoid fonksiyonu ve türevi.....	86
Şekil 6.20. Tanh fonksiyonu ve türevi .....	87
Şekil 6.21. ReLU fonksiyonu ve türevi .....	88
Şekil 6.22. Sızdıran ReLU fonksiyonu ve türevi .....	89
Şekil 6.23. Parametrik ReLU .....	89
Şekil 6.24. ELU aktivasyon fonksiyonu grafiği .....	90
Şekil 6.25. Swish fonksiyonu ve türevi .....	91
Şekil 6.26. VGG-16 mimarisi şematik gösterimi.....	93
Şekil 6.27. MobileNet mimarisi şematik gösterimi .....	94
Şekil 6.28. ResNet mimarisi şematik genel şematik gösterimi.....	95
Şekil 6.29. Residual blok yapısı.....	95
Şekil 6.30. DenseNet mimarisi şematik gösterimi.....	96
Şekil 6.31. Inception mimarisi şematik gösterimi .....	98
Şekil 6.32. Xception mimarisi şematik gösterimi.....	99
Şekil 6.33. Transfer öğrenme süreci .....	100
Şekil 8.1. ESA modellerine ve optimizasyon algoritmalarına göre doğruluk değerleri .	137
Şekil 8.2. Xception ESA mimarisi RMSprop optimizasyon algoritması karmaşıklık matrisi .....	138
Şekil 8.3. Xception ESA mimarisi RMSprop optimizasyon algoritması ROC eğrisi....	138

<b>Şekil</b>	<b>Sayfa</b>
Şekil 8.4. ESA modellerine, optimizasyon algoritmalarına ve nöron sayılarına bağlı doğruluk oranları .....	140
Şekil 8.5. Xception ESA mimarisi RMSprop 512 nöron karmaşıklık matrisi .....	141
Şekil 8.6. Xception ESA mimarisi RMSprop 512 nöron ROC eğrisi .....	141
Şekil 8.7. LBP öznitelik çıkarımı ve algoritmaların doğruluk performansları .....	143
Şekil 8.8. HOG öznitelik çıkarımı ve algoritmaların doğruluk performansları .....	144
Şekil 8.9. GLCM öznitelik çıkarımı ve algoritmaların doğruluk performansları .....	145
Şekil 8.10. SOBEL öznitelik çıkarımı ve algoritmaların doğruluk performansları .....	146
Şekil 8.11. GABOR öznitelik çıkarımı ve algoritmaların doğruluk performansları .....	147
Şekil 8.12. MB-LBP öznitelik çıkarımı ve algoritmaların doğruluk performansları .....	148
Şekil 8.13. Hibrit öznitelik çıkarımı ve algoritmaların doğruluk performansları .....	149
Şekil 8.14. LBP öznitelik çıkarımı Random Forest Classifier karmaşıklık matrisi .....	150
Şekil 8.15. LBP öznitelik çıkarımı Random Forest Classifier ROC eğrisi .....	151
Şekil 8.16. LBP öznitelik çıkarımı Gradient Boosting Classifier karmaşıklık matrisi ...	151
Şekil 8.17. LBP öznitelik çıkarımı Gradient Boosting Classifier ROC eğrisi .....	152
Şekil 8.18. LBP öznitelik çıkarımı XGB Classifier karmaşıklık matrisi .....	152
Şekil 8.19. LBP öznitelik çıkarımı XGB Classifier ROC eğrisi .....	153
Şekil 8.20. Hibrit yaklaşım sınıflandırma doğruluk performansları .....	155
Şekil 8.21. Hibrit yaklaşım Random Forest Classifier karmaşıklık matrisi .....	156
Şekil 8.22. Hibrit yaklaşım Random Forest Classifier ROC eğrisi .....	156

<b>Şekil</b>	<b>Sayfa</b>
Şekil 8.23. Hibrit yaklaşım Gradient Boosting Classifier karmaşıklık matrisi .....	157
Şekil 8.24. Hibrit yaklaşım Gradient Boosting Classifier ROC eğrisi .....	157
Şekil 8.25. Hibrit yaklaşım CatBoost Classifier karmaşıklık matrisi .....	158
Şekil 8.26. Hibrit yaklaşım CatBoost Classifier ROC eğrisi .....	158

**RESİMLERİN LİSTESİ**

<b>Resim</b>	<b>Sayfa</b>
Resim 7.1. Veri kümesindeki ahşap kusurlarının tipik örnekleri: (A) Kaynamış Budak, (B) Düşen Budak, (C) Ur, (D) Çatlak Budak, (E) Eksik Budak, (F) Çatlak, (G) Aşırı Büyüme, (H) Reçine, (I) Öz (J) Mavi Leke .....	104
Resim 7.2. Kusursuz ahşap yüzeyleri (a), kusurlu ahşap yüzeyleri (b) .....	105

## SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

<b>Simgeler</b>	<b>Açıklamalar</b>
<b>GHz</b>	Gigahertz (bir frekans birimidir ve bir saniyede milyar döngüyü temsil eder)
<b>KHz</b>	Kilohertz (bir frekans birimidir ve bir saniyede bin döngüyü temsil eder)
<b>Kısaltmalar</b>	<b>Açıklamalar</b>
<b>Adadelta</b>	Adaptive Delta
<b>AdaGrad</b>	Adaptive Gradient Algorithm (Uyarlanabilir Gradyan Algoritması)
<b>Adam</b>	Adaptive Momentum (Uyarlanabilir Moment)
<b>ANN</b>	Artificial Neural Network (Yapay Sinir Ağı)
<b>AUC</b>	Area Under The Curve (Eğri Altındaki Alan)
<b>ESA</b>	Convolutional Neural Network (Evrişimli Sinir Ağı)
<b>CPU</b>	Central Processing Unit (Merkezi İşlem Birimi)
<b>DBN</b>	Deep Belief Networks (Derin İnanç Ağları)
<b>ELU</b>	Exponential Linear Unit
<b>ESA</b>	Evrişimli Sinir ağı
<b>GB</b>	Gigabayt
<b>GAN</b>	Generative Adversarial Network (Üretken Çekişmeli ağlar)
<b>GLCM</b>	Gray-Level Co-occurrence Matrix (Gri Seviye Ortak Oluşum Matrisi)
<b>GPU</b>	Graphics Processing Unit (Grafik İşleme Birimi)
<b>HOG</b>	Histogram of Oriented Gradients (Yönlü Gradyanların Histogramı)

**Kısaltmalar****Açıklamalar**

<b>BP</b>	Back Propagation (Geri yayılım)
<b>KNN</b>	K-Nearest Neighbors (K en yakın komşu)
<b>LBP</b>	Local Binary Pattern (Yerel İkili Desen)
<b>LSTM</b>	Long Short Term Memory (Uzun-Kısa Hafıza Ağları)
<b>MB-LBP</b>	Multi-block Local Binary Pattern (Çoklu Blok Yerel İkili Desen)
<b>ML</b>	Machine Learning (Makine Öğrenmesi)
<b>MLP</b>	Multi Layer Perceptron (Çok Katmanlı Algılayıcı)
<b>RAM</b>	Random Access Memory (Rastgele Erişimli Bellek)
<b>RBM</b>	Restricted Boltzmann Machine (Kısıtlı Boltzmann Makinesi)
<b>ReLU</b>	Rectified Linear Unit (Düzleştirilmiş Doğrusal Birim)
<b>RMSProp</b>	Root Mean Square Error Propability (Ortalama Kareysel Hata Olasılığının Karekökü)
<b>RNN</b>	Recurrent Neural Network (Tekrarlayan Sinir Ağı)
<b>RNN</b>	Recurrent Neural Network (Tekrarlayan Sinir Ağı)
<b>ROC</b>	Receive-Operation Characteristic (Alıcı İşlem Karakteristiği)
<b>SGD</b>	Stochastic Gradient Descent (Stokastik Gradyan İniş Metodu)
<b>SVC</b>	Support Vector Classification (Destek Vektör Sınıflandırması)
<b>SVM</b>	Support Vector Machine (Destek Vektör Makinesi)
<b>Tanh</b>	Hyperbolic Tangent (Hiperbolik Tanjant)
<b>XGB</b>	Extreme Gradient Boosting
<b>YSA</b>	Yapay Sinir Ağı

## 1. GİRİŞ

Ağaçları endüstri mühendisliği kalite kontrol süreçlerinde, ahşap yüzeylerindeki kusurların incelenmesi ve tespiti son derece önemlidir. Bu durum, nihai ürünün kalitesinin tutarlılığını sağlar ve üretim sürecinin verimliliğini artırmaktadır. Kusurlu ürünlerin mümkün olduğunca erken tespit edilip üretim hattından çıkarılması gereklidir. Ahşap ürünlerde kusurlar özellikle dikkat gerektirir, çünkü kusurlar ticari değerlerini önemli ölçüde etkilemektedir. Bu kusurlar, düşük kaliteli hammadde ve etkisiz üretim süreçleri gibi nedenlerden kaynaklanabilir. Bazı ülkelerde, bu kusurlar nedeniyle ham ahşap malzemelerin kullanımı azalmaktadır. Günümüzde, ahşap ürünleri yüzey işleme şartnamelerine uygun olarak üretilmektedir ve modern ahşap panel işleme endüstrileri düzgün ve geçerli bir ahşap kusur tespit ve tanımlama sistemine ihtiyaç duymaktadır. Günümüzde ahşap yüzey kusur tespiti kalite denetimleri, eğitimli personeller tarafından gerçekleştirilmektedir.

Ahşabın kalitesi ve ticari değerinin değerlendirilmesi açısından ahşaptaki kusurlar son derece önemli bir standarttır. Ahşabın kullanımında yararlanma oranını artırmak için, ahşabın belirli özelliklerine yönelik farklı endüstrilerin ihtiyaçlarını karşılamak için ahşabın kusurlarını tespit etmek ve sınıflandırmak gerekir. Geleneksel ahşap işlemede, ahşabın yüksek kaliteli kusur tespiti ve sınıflandırılması genellikle deneyimli denetçilerin çıplak gözle gözlemlerine dayanmaktadır; bu nedenle, bu yöntemin etkinliği ve doğruluğu garanti edilemez (Zhang ve diğerleri, 2016).

Ahşap endüstrisinde, üretim sürecinin her adımı malzeme kullanımını ve maliyet etkinliğini etkiler (Broman ve Fredrikson, 2012). Ağaçları endüstrideki üretim işlemlerinin karmaşıklığı ile birlikte ahşap malzemenin heterojenliği, yalnızca ahşabın mekanik özelliklerini bozmakla kalmayan çeşitli kusurlara neden olabilir. Bunlar; mukavemeti ve sertliği azaltır, ancak aynı zamanda estetik değerini de azaltır (Ding ve diğerleri, 2020). Bu mekanik ve estetik kusurlar ayrıca ahşabın ticari değeri üzerinde büyük bir etkiye sahiptir ve bu tür malzemelerin daha sonraki işlemler için kullanımını azaltabilir. Birçok farklı nedenden kaynaklanan birçok farklı kusur türü vardır. Başlıca ahşap kusurları arasında budaklar, mantar hasarı, çatlaklar, eğrilme, eğiklik, böcek tahribatları ve reçine kusurları bulunur. Bir kusurun ciddiyeti ve dolayısıyla malzemenin

derecesi ve maliyeti, öncelikle kusurun boyutu, yeri, tipi ve ahşap ürünün kullanılacağı amaç dahil olmak üzere dört kriter tarafından belirlenmektedir (Çetiner ve diğerleri, 2014).

İmalat endüstrilerindeki kusur tespit prosedürlerini otomatikleştirme talebi, yaklaşan endüstriyel 4.0 çağıyla birlikte hızla artmaktadır (Haoran ve diğerleri, 2021). Verim oranını iyileştirme ve üretim maliyetini düşürme bakış açılarından, kusur tespit prosedürlerinin otomatikleştirilmesi yalnızca daha fazla analiz için çevrimiçi üretim verileri oluşturmakla kalmaz, aynı zamanda insan emeğinin maliyetlerinden de tasarruf sağlar. Ayrıca artan işçilik maliyetleri de bu konuyu daha da önemli hale getirmektedir (Lin ve Sanjaya, 2021).

Ağaçşeri endüstrisi ürünleri, çok sayıda gelişmekte olan ülkenin ekonomisine önemli katkılar sağlamaktadır. Yüksek kaliteli ahşap, genellikle görsel güzelliği ve yapısal güçlülüğü nedeniyle aranılmaktadır. Ahşap, biyolojik bir malzeme olduğu, doğal ortam ve yetiştirme koşullarına göre kalitesinin sıklıkla etkilendiği için, kusurlar içeren ahşaplar oldukça yaygın bir şekilde bulunmaktadır. Ayrıca, kerestenin ahşaba dönüşüm aşamasında da kusurlar oluşabilmektedir. Ahşap kusurlarının varlığı, ahşabın estetiğini ve yapısal güçlülüğünü büyük ölçüde etkilemektedir (Qayyum ve diğerleri, 2016). Bu nedenle, ahşap üreticileri, ürünlerin kalitesinin bozulmasını önlemek için üretim süreçleri sırasında kusurlu ahşapları tespit etmek ve üretim hatlarından hariç tutmak için önemli bir insan gücü yetiştirmiştir (Lim ve diğerleri, 2022).

Bu endüstriyel sektördeki otomasyonun büyümesine rağmen, birçok pazar lideri şirket, istenmeyen özellikleri tespit etmek ve kalite derecelendirmesi yapmak için hala eğitimli alan uzmanlarından yararlanmaktadır (Gu ve diğerleri, 2019). Manuel incelemenin sıkıcı, yanılıcı ve taraflı olmasının yanı sıra, alan uzmanlarının büyük üretim hacimlerini kontrol edemediği görülmüştür (Urbonas ve diğerleri, 2019). Muayenenin tekrarlanabilirliği ve kalitesine göre yapılan çalışmada göz yorgunluğu veya dikkat dağınıklığı gibi faktörler nedeniyle manuel incelemenin nadiren %70 güvenilirliğe ulaştığını belirtilmiştir. Manuel incelemenin dezavantajlarının üstesinden gelmek için araştırmacılar, doğru olan ve üretim sürecini yavaşlatmayan otomatik sistemler geliştirmeye çalışırlar (Lycken, 2006). Otomatik sistemlerin insan sınıflandırıcılardan biraz daha iyi performans gösterdiğini zaten kanıtladı. Bu sistemlerin çoğu, denetimli öğrenme algoritmalarıyla birlikte geleneksel görüntü işleme tekniklerine dayanıyordu, ancak son on yılda derin öğrenme, ormancılık ve

ağaçşileri endüstrisinde dikkate değer bir başarı elde ettikleri görülmüştür (Liu ve diğerleri, 2018).

### Problem durumu / Konunun tanımı

Ahşap, tarih boyunca birçok endüstri için temel bir yapı malzemesi olarak kullanılmıştır ve günümüzde de çok çeşitli uygulamalarda yer almaktadır. Ahşap malzemenin benzersiz ve doğal yapısı, estetik cazibesi, taşıma kapasitesi ve çevre dostu özellikleri, inşaat sektöründen mobilya üretimine kadar bir dizi endüstriye büyük bir değer sunmaktadır. Ancak, ahşap malzemelerin üretimi ve kullanımı sırasında oluşabilecek kusurlar, ciddi ekonomik kayıplara ve güvenlik risklerine yol açabilir. Bu nedenle, ahşap endüstrisinde, bu tür kusurların erken tespiti ve etkili bir şekilde sınıflandırılması büyük bir öneme sahiptir. Ahşap malzemeler, günümüzde inşaat, mobilya üretimi, dekorasyon ve pek çok endüstriyel uygulamada yaygın bir şekilde kullanılmaktadır. Ancak, bu değerli malzemenin üretim aşamasında veya sonraki kullanımlarında oluşabilecek kusurlar hem ekonomik hem de güvenlik açısından ciddi sorunlara yol açabilir. Bu nedenle, ahşap malzeme üreticileri ve kullanıcıları, kusurların erken tespiti ve tanımlanmasını sağlayacak etkili yöntemlere olan ihtiyaçlarını anlaşılmaktadır.

### Tezin amacı

Bu araştırmada, ahşap malzemelerin sınıflandırılması ve kusurlarının tespiti için kullanılan çeşitli görüntü işleme ve yapay zeka yöntemlerinin performansını araştırmaktır. Ahşap yüzey kusur tespiti için klasik makine öğrenme ve derin öğrenme yöntemlerinin ahşap kusurları sınıflamada göstereceği performansları belirlemek amacıyla hazırlanmıştır. Bu yöntemler, endüstriyel süreçlerde ahşap malzemelerin kullanım verimliliğini artırmak ve kusurların erken tespitini sağlamak için kritik bir rol oynamaktadır.

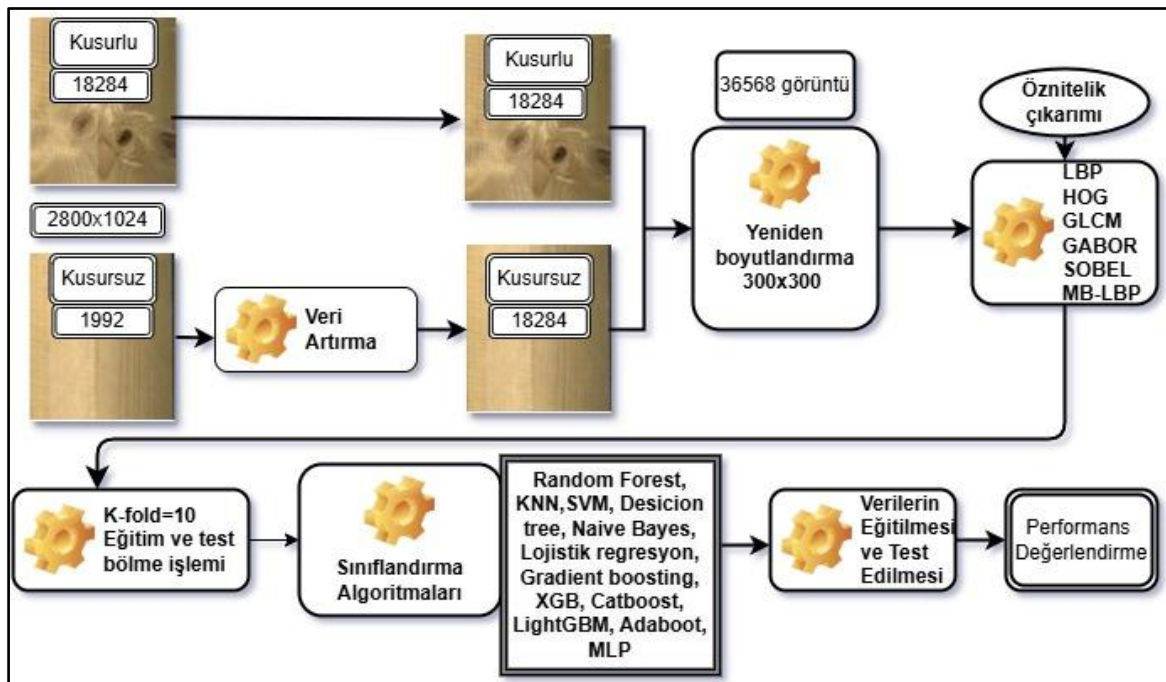
Birinci amaç; günümüzde ahşap kusur tespiti için farklı geleneksel yöntemler uygulanır. Bunlardan başlıcaları şunlardır; lazer testi, ultrasonik test ve akustik test yöntemleridir. Bunlara ek olarak gözleme dayalı veya tecrübeye dayalı tespit yöntemleri de kullanılır. Fakat, bu yöntemler hem özel laboratuvar ekipmanı gerektirir hem de zaman açısından maliyetli bir yöntemdir. Geleneksel yöntemlerin açıklarını kapatmak için yapay zeka görüntü işleme yöntemi kullanılması amaçlanmaktadır.

İkinci amaç; bu tez çalışmanın amacı ve bilime olan katkısı ise hem ekipman bazlı hataları hem de insan ön yargısına dayalı olabilecek hataları yapay zeka desteği ile ortadan kaldırmak hem zamandan tasarruf etmek hem de olabilecek deneysel bazlı ve kontrol maliyetin önüne geçmektir.

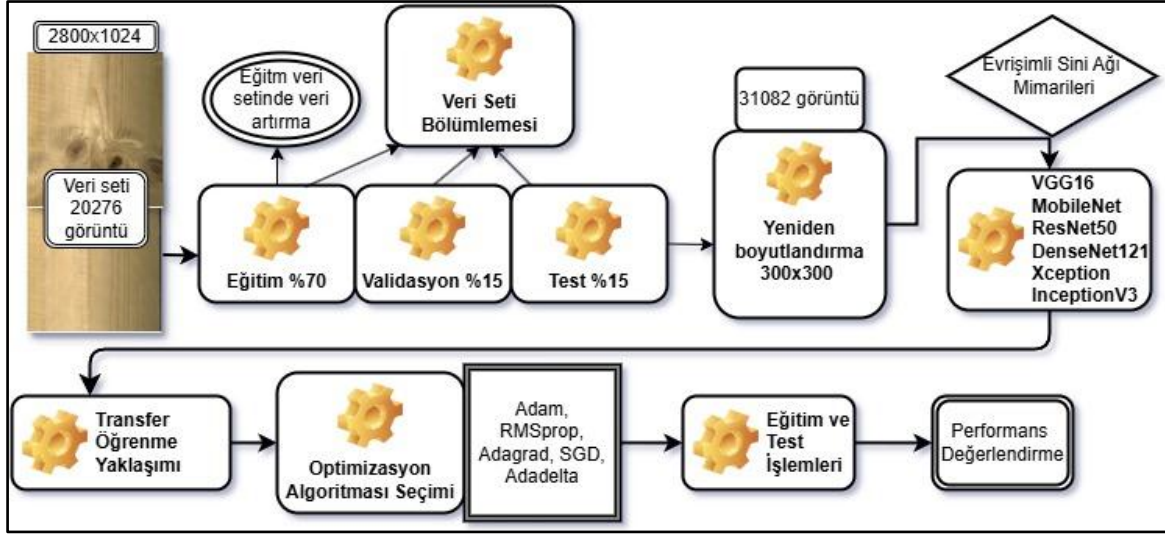
Üçüncü amaç; ahşap kusur tespitinde oluşan hata ver zaman kaybını en az indirerek kalite kontrol süreçlerini bilgisayarlı görü vasıtasıyla montaj hatalarına parça geçmeden üretim hattı üzerinde kusur tespiti yapılarak ek maliyet ve zaman kaybının önüne geçilmesi amaçlanmaktadır.

Dördüncü amaç; bir kalite kontrol programında kullanılabilir yöntemleri araştırmak ve geliştirmektir. Ağaçşileri endüstri mühendisliği alanında yapay zekanın uygulanabilirliği ve kullanılabilirliğin yaygınlaştırılması bu tez kapsamında amaçlanmaktadır.

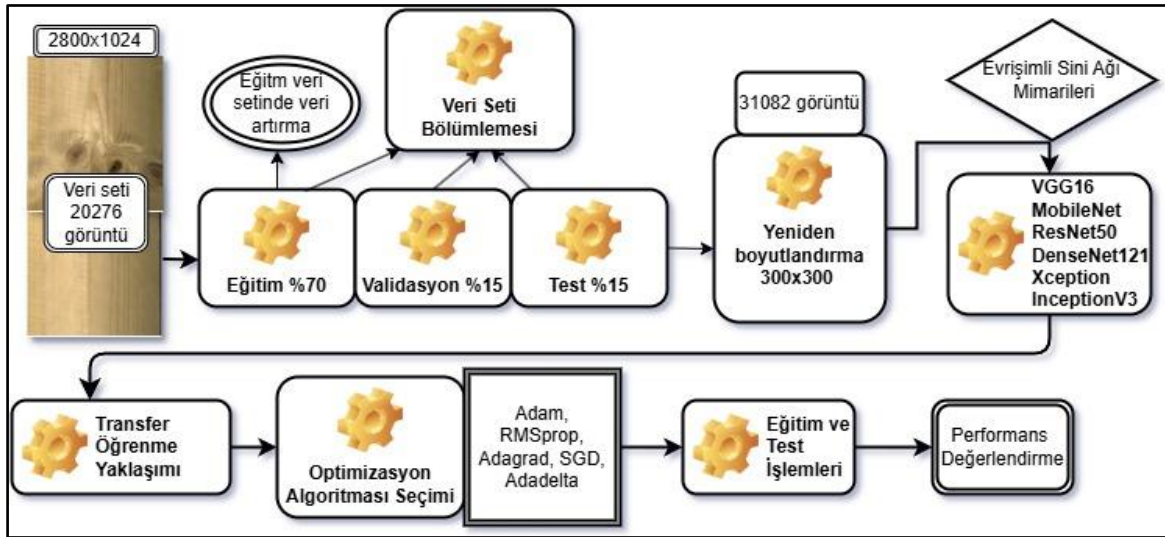
Tezin amaçlarına ulaşmak için kullanılan yöntem ve işlemler Şekil 1.1’de klasik makine öğrenme yöntemleri ile ahşap kusur tespiti işlem basamakları, Şekil 1.2’de derin öğrenme yöntemleri ile ahşap kusur tespiti işlem basamakları, Şekil 1.3’te hibrit öğrenme yaklaşımı ile ahşap kusur tespiti işlem basamakları verilmiştir.



Şekil 1.1. Klasik makine öğrenme yöntemleri ile ahşap kusur tespiti işlem basamakları



Şekil 1.2. Derin öğrenme yöntemler ile ahşap kusur tespiti işlem basamakları



Şekil 1.3. Hibrit öğrenme yaklaşımı ile ahşap kusur tespiti işlem basamakları

Tez kapsamında belirlenen amaçlara ulaşmak için farklı araştırma soruları belirlenmiş ve aşağıda sunulmuştur:

- 1- Ahşap kusur tespitinde yapay zeka destekli görüntü işleme yöntemleri, geleneksel yöntemlere kıyasla ne ölçüde daha doğru, hızlı ve maliyet etkin bir çözüm sunmaktadır?
- 2- Yapay zeka ve bilgisayarlı görü teknikleri, insan önyargısından kaynaklanabilecek hataları ortadan kaldırmada ve üretim süreçlerinde zaman tasarrufu sağlamada nasıl bir avantaj sağlar?
- 3- Üretim hattında bilgisayarlı görü kullanılarak kusur tespitinin, montaj hatalarına yol açabilecek kalite kontrol problemlerini önlemedeki etkinliği nedir?

- 4- Ahşap endüstrisinde yapay zeka tabanlı bir kalite kontrol programının uygulanabilirliği ve yaygınlaştırılabilirliği nasıl sağlanabilir ve bu programın geliştirilmesinde hangi yöntemler öncelikli olarak değerlendirilmelidir?

### Tezin önemi

Ahşap malzemelerin üretim ve kullanım süreçlerindeki kusurların tespiti için potansiyel olarak kullanışlı tekniklerin ve algoritmaların keşfedilmesine katkıda bulunmayı hedeflemektedir. Ahşap endüstrisinde kalite kontrol ve güvenlik konularında gelişmeler sağlamak amacıyla yapılan bu araştırmanın sonuçları, endüstri profesyonelleri, araştırmacılar ve üretim yapan endüstriyel tesisler için büyük öneme sahiptir. Geleneksel olarak, ahşap kusurlarının tespiti ve sınıflandırılması manuel işçilikle gerçekleştirilirdi. Ancak bu yaklaşım, hızlı üretim ihtiyaçlarını karşılamak ve daha fazla hassasiyet sağlamak amacıyla otomatik sınıflandırma makinelerinin geliştirilmesiyle yerini otomasyona bıraktı. Bu otomatik sınıflandırma makineleri, görüntü tanıma yöntemlerini kullanarak ahşap yüzeylerdeki kusurları tespit etme yeteneğine sahiptir. Bu gelişmeler, ahşap endüstrisinde kalite kontrolünü iyileştirmeye ve malzeme israfını azaltmaya yardımcı olması bakımından bu tez hazırlanmıştır.

### Varsayımlar / Sayıtlar / Sınırlılıklar

Veri Seti Özellikleri: Bu çalışmanın temelini oluşturan veri setinin nitelikleri, kullanılan özellik çıkarım yöntemleri ve sınıflandırma algoritmalarının performansını etkileyebilir. Veri setindeki örneklerin çeşitliliği, temsil edilen kusur türleri ve görüntü kalitesi gibi faktörler, sonuçların genellenebilir olması durumu etkileyebilir.

Özellik Çıkarım Yöntemleri: Çalışmada kullanılan altı farklı özellik çıkarım yöntemi, ahşap malzemelerdeki kusurların tespiti üzerinde farklı etkilere sahip olabilir. Her bir yöntemin avantajları ve dezavantajları dikkate alınmalıdır.

Sınıflandırma Algoritmaları: Farklı sınıflandırma algoritması kullanılmıştır. Ancak, bu algoritmaların performansı, veri seti özellikleri ve kullanılan özellik çıkarım yöntemlerine bağlı olarak değişebilir. Ayrıca, algoritmaların hiper parametrelerinin ayarlanması da sonuçları etkileyebilir.

Hassasiyet ve Spesifiklik: Makine öğrenimi modelinin başarı ölçütleri olan hassasiyet (precision) ve spesifiklik (specificity) gibi faktörler, kusurların tespiti açısından önemlidir. Ancak, bu ölçütler arasında bir denge bulunması gerekebilir.

Donanım ve Yazılım Sınırlamaları: Kullanılan bilgisayar donanımı ve yazılımı, görüntü işleme ve makine öğrenimi algoritmalarının hız ve performansını etkileyebilir. Bu sınırlamalar, uygulamaların gerçek dünya koşullarında kullanılabilirliğini sınırlandırabilir.

Zaman ve Maliyet: Çeşitli özellik çıkarım yöntemleri ve sınıflandırma algoritmalarının uygulanması, zaman ve maliyet faktörleriyle ilişkilidir. Bu faktörler, endüstriyel uygulamalarda kullanılabilirliği etkileyebilir.

Değişen Çevresel Koşullar: Ahşap malzemelerin üretildiği ve kullanıldığı ortamlardaki değişen çevresel koşullar, görüntü kalitesi üzerinde etkili olabilir. Bu durum, algoritmaların gerçek dünya koşullarında ne kadar etkili olduğunu etkileyebilir.

### Tanımlar

Tezde geçen kavramlar için okuyuculara ön bilgi vermek ve tezle ilgili kavramları tanıtmak amacıyla yapılmıştır.

Yapay zeka: Bilgisayar sistemlerine problem çözme insan benzeri bir şekilde düşünme yetenekleri kazandırmayı amaçlayan bir bilim ve mühendislik alanıdır. Bu sistemler, öğrenme, problem çözme, dil anlama, görüntü işleme gibi alanlarda insan benzeri yeteneklere sahip olabilirler. Yapay zeka, genelde sembolik ve bağlamsal olmak üzere iki ana yaklaşımı içermektedir (Russell ve Norving, 2010; Nilsson, 1998).

Makine öğrenimi: Bilgisayar sistemlerinin verilerden öğrenme kapasitesini geliştiren bir yapay zeka disiplini. Algoritmalar, veri setlerindeki desenleri inceleyerek genelleme yeteneği kazanır ve yeni veriler üzerinde tahminlerde bulunabilir. Makine öğrenimi, denetimli, denetimsiz ve güçlendirme öğrenme gibi çeşitli öğrenme paradigmalarına sahiptir (Murphy, 2012; Bishop ve Nasrabadi, 2006).

Derin öğrenme: Karmaşık veri setlerinde otomatik öğrenme kapasitesini artırmayı hedefleyen bir yapay zeka dalıdır. Derin öğrenme genellikle çok katmanlı yapay sinir ağları kullanılarak gerçekleştirilir. Bu metod, özellikle geniş veri setleri ve karmaşık görevlerle başa çıkmak için etkili bir yaklaşımdır.

Bilgisayarlı görü: Bilgisayar sistemlerinin görsel bilgiyi analiz etme ve anlama yeteneğini temsil eder. Görüntü işleme, nesne tanıma, yüz tanıma gibi konuları içerir. Bilgisayarlı görü, yapay zeka uygulamalarında sıkça kullanılan bir alan olup, derin öğrenme teknikleri ile birlikte daha da güçlendirilmiştir (Szeliski, 2022; Forsyth ve Ponce, 2012).

Kusur tespiti, özellikle endüstriyel üretim süreçlerinde kullanılan bilgisayarlı görü ve makine öğrenimi teknikleriyle, ürünlerdeki kusurları otomatik olarak tespit etmeyi amaçlar. Bu, ürün kalitesini artırmak ve hatalı ürünleri erken aşamada tanımlamak için önemli bir uygulamadır. Kusur tespiti, belirli bir veri kümesindeki anormal veya anormal verileri tespit eden önemli bir veri analizi görevidir. Verilerdeki büyüleyici ve nadir kalıpları keşfetmeyi içerdiğinden veri madenciliği araştırmasının ilginç bir alanıdır. (Ahmed ve diğerleri, 2014).

### Tezin yöntemi

Görüntü işleme ve yapay zeka modellerini oluşturmak için Python programlama dili kullanılacak. Python diline entegre olan Keras, Tensorflow, Scikit-learn ve Scikit-image kütüphaneleri kullanılarak veri için tasarlanmış algoritma modeli uygulamaya geçirilecek. Bunlara ek olarak, farklı makine ve derin öğrenme modelleri (sinir ağları, karar ağaçları, evrimsel sinir ağları) kullanılacaktır. Makine öğrenme ve derin öğrenme algoritmalarında ahşap kusuru için iki sınıflama (Binary Classification) yöntemi kullanılacaktır. Doğruluklar makine öğrenmesi değerlendirme metriklerine göre değerlendirilecektir.

### Araştırmanın katkısı

Endüstrilerde robotik otomasyona olan ilgi artmakta ve sürekli olarak yeni yöntemler ve robotlar geliştirilmektedir. Bununla birlikte, bazı alanlarda, özellikle ürünü görsel olarak inceleme ve doğru çalışma hızıyla derecelendirme yeteneğine sahip bir robot kullanarak otomasyona sahip olmak zordur. Ahşap endüstrisinde, ahşabın kalite güvencesi hala bir

insan tarafından kontrol edilmektedir çünkü kusurlar alan, renk, şekil ve miktar gibi birçok yönden farklılık gösterebilir. Manuel kontrol süreci çok sıkıcı ve hatalı olabilir ve insan yargısına göre hatalara eğilimlidir. Küçük bir insan hatası, kereste fabrikaları için gelir kaybına neden olabilmektedir (Yap, 2019; Jabo, 2011).

Yapay zekanın amacı, makinenin sorunu anlaması ve çözmesi için insan zekası seviyesine ulaşabilen akıllı bir makine yapmaktır. Makine öğrenme yöntemiyle, otomatik ahşap kusur tespiti ve sınıflandırma probleminin çözülmesine yardımcı olabilir. Alanında uzman bir kişi, gördüğü ahşabı ve üzerindeki kusurları kolaylıkla tanımlayabilirken, yapay zeka, uzmanların yaptığı bu işlemi daha hızlı, hatasız ve güvenli bir şekilde gerçekleştirebilecek şekilde eğitilebilir.

Bu araştırma, ahşap yüzey kusurlarının tespiti için görüntü işleme ve yapay zeka yöntemlerinin performanslarını incelemiştir. Elde edilen sonuçlar, kullanılan özellik çıkarım yöntemleri ve sınıflandırma algoritmalarının etkisiyle birlikte değerlendirilmiştir. Ahşap endüstrisi için kritik öneme sahip olan bu konuda elde edilen bulgular, ağaçişleri endüstri profesyonelleri, araştırmacılar ve karar vericiler için değerli olabilir.

Bu tez, ahşap yüzey kusurlarının tespiti için görüntü işleme ve yapay zeka yöntemlerinin etkinliğini değerlendirmeyi amaçlamıştır. Elde edilen sonuçlar, endüstriyel süreçlerde ahşap malzemelerin kullanım verimliliğini artırmak ve kusurların erken tespitini sağlamak adına kritik bir rol oynayabilecek teknik ve algoritmaların keşfedilmesine katkıda bulunmayı hedeflemektedir. Gelecekteki çalışmalar, bu alandaki teknolojik gelişmeleri takip ederek, endüstriyel uygulamalarda daha etkili ve güvenilir çözümler sunma potansiyelini daha da geliştirebilir.

### Kısıtlar

Klasik makine öğrenmesi çalışmalarında, öznitelik çıkarımı zorlu ve uzman bilgisi gerektiren bir süreçtir. Doğru öznitelikleri seçmek için genellikle domain uzmanlığına ihtiyaç duyulur. Bunun yanı sıra, öznitelik mühendisliği manuel olarak yapıldığı için elde edilen doğruluk, modelin başarısına ve kullanılan özelliklerin kalitesine bağlıdır. Ağaçişleri endüstrisinde, klasik makine öğrenmesi yöntemleri uzman ihtiyacı doğurduğundan, uygulamalarda bazı zorluklarla karşılaşabilmektedir.

ESA gibi derin öğrenme yöntemlerinde, öznitelikler otomatik olarak öğrenilir, bu da manuel öznitelik mühendisliğine olan ihtiyacı azaltır. Ancak, bu durum uzman bilgisine ihtiyaç duyulmadığı anlamına gelmez. Modelin hiper parametre ayarları, veri artırma teknikleri, transfer öğrenme uygulamaları gibi süreçler derin öğrenme uzmanlığı gerektirmektedir.

Bunun yanı sıra, derin öğrenme modelleri genellikle yüksek hesaplama gücü gerektirdiğinden güçlü GPU veya bulut tabanlı sistemler kullanılmalıdır. Bu, klasik makine öğrenmesine kıyasla bir dezavantaj oluşturabilir. Sonuç olarak hem klasik makine öğrenmesi hem de derin öğrenme belirli bir düzeyde uzmanlık gerektirir, ancak uzmanlığın türü farklıdır.

Ahşap kusurlarının birden fazla olduğu durumlarda, kusurların ayrılmaması veya belirsizlik yaşanması mümkün olmaktadır. Özellikle bazı ahşap kusurlar birbirine yakın veya örtüşen özellikler taşıyorsa, yapay zeka algoritmalarının bu kusurları doğru bir şekilde ayırt etmesi zorlaşabilmektedir. Bu tür durumlarda, kusurların net bir şekilde sınıflandırılabilmesi için daha fazla veri, daha gelişmiş algoritmalar ve daha hassas etiketleme gerekebilir. Ayrıca, kusurlar arasındaki benzerlikleri ve farklılıkları daha iyi anlayabilmek için farklı özelliklerin (örneğin, renk, doku, boyut) dikkate alınması faydalı olabilir.

#### Araştırmanın literatüre sağladığı katkılar

Ahşap kusur tespiti için bilgisayar görüşü tabanlı makine öğrenimi yaklaşımı adlı tez çalışması literatüre aşağıdaki katkıları sağlamıştır:

- 1- Ahşap yüzey kusurlarının büyük ölçekli görüntü veri kümesi adlı veri kümesi kullanılarak en çok sayıda 20 276 gerçek ahşap işleme tesisinden elde edilmiş görüntüler bu çalışmada kullanılmıştır.
- 2- Yüksek sayıda görüntü içermesi her türlü kusur türünün mevcut olmasıyla elde edilen sonuçlar gerçek dünya problemlerine yüksek seviye uyum göstermektedir.
- 3- Kullanılan veri setinde üretim hatlarında kusur türlerinden ziyade kusurlu olup olması önem gösterdiğinden dolayı hızlı karar vermek adına bu veri setine ilk kez ikili sınıflandırma (binary classificaiton) yaklaşımı uygulanmıştır.

- 4- Tez araştırması kapsamında, klasik makine öğrenme yaklaşımı kullanılmayıp, buna ek olarak derin öğrenme yöntemleri ve hibrit yaklaşımda uygulanarak daha yüksek doğrulukla ahşap kusur tespiti yapılabilmesi derinlemesine ve her yönden analiz edilmiştir.
- 5- Araştırma öznitelik çıkarımında oldukça farklı öznitelik çıkarımları yapılmış ve etkileri incelenmiştir.
- 6- Farklı öznitelik çıkarım yöntemleri kullanılmış, 6 farklı öznitelik çıkarımı yapılmıştır.
- 7- Klasik sınıflandırma algoritmaların ahşap kusur tespitinde başarıları görmek için oldukça farklı 12 klasik sınıflandırma algoritması kullanılmıştır.
- 8- Bu çalışmada 6 farklı evrişimli sinir ağı mimarisi ahşap kusur tespiti için optimize edilmiştir.
- 9- Evrişimli sinir ağları ile yüksek doğrulukla %99,32 oranında ahşap kusur tespiti gerçekleştirilerek yüksek düzeyde kusur tespiti gerçekleştirilmiştir.
- 10- Araştırma klasik makine öğrenme, derin öğrenme ve hibrit öğrenme yöntemleriyle geniş analizleriyle mevcut literatüre katkı sağlamıştır.
- 11- Ülkemizde ağaçları endüstri mühendisliği alanında kalite kontrol süreçlerinin iyileştirilmesi, hızlı ve doğru kararlar vermesi ve insan (uzman) kaynaklı hataların azaltılması ve maliyetlerin düşürülmesi konusunda ülke ekonomisine katkı sağlamaktadır.
- 12- Ahşap kusurların tespiti fire oranını düşürmekte ve üretilen ürünlerin kalitesini artırıcı etki yapması bakımından da bu tez çalışması literatüre olumlu katkılar sağlamıştır.

Mevcut bilgimize göre, bu tez çalışması birçok gerçek gerçek ahşap görüntü kullanarak yapılan üç farklı türde analiz sunan en yüksek doğruluk elde edilen ülkemizde ağaçları endüstri mühendisliği alanında hazırlanmış olan oldukça geniş kapsamlı, özgün ve yüksek doğrulukla ahşap kusur tespit yöntemlerini önermekte olan ilk çalışmadır.

#### Tez içerik planı

Bu araştırma kapsamında yapılan çalışmalar sonraki başlıklarda verilmektedir. Her bir bölüm içerisinde yapılanları açıklaya tez içerik planlaması şu şekildedir:

İkinci bölümde, ahşap kusur tespiti ile ilgili literatür araştırmaları verilmiştir. Literatürde derin öğrenme ve klasik makine öğrenme çalışmalarıyla ilgili 2 başlıkta literatür özeti sunulmaktadır.

Üçüncü bölümde, ağaç kusurları ve ağaç hastalıkları ilgili genel bilgiler ve tanımlamalar yapılmaktadır. Ağaç kusurları ve ağaç hastalıkları ne türde oldukları sınıflandırmaları, ağaç malzeme üzerindeki etikleri ve görünüşleri ilgili genel bilgiler verilmektedir.

Dördüncü bölümde, görüntüler üzerinden öznitelik çıkarımı ve bu çıkarılmaların nasıl olduğuna dair teorik bilgiler sunulmuştur. Araştırmada kapsamında kullanılan; Yerel İkili Örüntüler Operatörü (Local Binary Pattern, Yönlü Gradyanlar Histogramı (Histogram of Oriented Gradients), Gri Seviye Eş-Oluşum Matrisi (Gray-Level Co-occurrence Matrix), Sobel, Gabor ve Çok Bloklü Yerel İkili Örüntü (Multi Block LBP / MB-LBP) hesaplama yöntem ve teknikleri verilmiştir.

Beşinci bölümde, makine öğrenme hakkında bilgiler ve örnekler verilmiştir. Makine öğrenme türleri; denetimli öğrenme, denetimsiz ve takviyeli öğrenme türleri hakkında teorik bilgilere yer verilmektedir. Denetimli öğrenmede; sınıflandırma, regresyon konuları hakkında bilgiler verilmiştir. Denetimsiz öğrenmede; kümeleme, ilişkilendirme, boyut azaltma konuları verilmiştir. Takviyeli öğrenmeni genel yapısı hakkında bilgilere de yer verilmiştir.

Altıncı bölümde, derin öğrenme konusu tanıtılmaktadır. Yapay sinir ağları, derin öğrenme modelleri, evrişimsel sinir ağları, optimizasyon algoritmaları, aktivasyon fonksiyonları, evrişimsel sını ağı mimarileri ve transfer öğrenme yaklaşımları hakkında detaylı bilgiler ve özellikleri hakkında bilgiler verilmektedir.

Yedinci bölümde, tez çalışmasında kullanılan materyal ve yöntem bilgileri paylaşılmıştır. Bu başlık altında veri seti, uygulama ortamı ve kütüphaneler, yöntem, deneysel kurulum, sınıflandırma, hiper parametre ayarları, çapraz doğrulama ve değerlendirme metrikleri açıklanmıştır.

Sekizinci bölümde, ahşap kusur tespitinde üç farklı yöntemle yapılmış olan tezin bulgular ve tartışmaları paylaşılmıştır. İlk olarak derin öğrenme mimarileri optimize edilerek ahşap yüzey kusur tespiti gerçekleştirilmiştir. Daha sonra öznitelik çıkarımı yapılmış ardından

klasik makine öğrenme algoritmaları ile sınıflandırma işlemleri gerçekleştirilmiştir. Son olarak hibrit yaklaşıma evrişimli sinir ağları optimize edilmiş öznetelik çıkarımı yapılaşmış ve ardından kusurlu kusursuz ahşap sınıflama işlemi gerçekleştirilmiştir.

Dokuzuncu bölümde, teze ait elde edilen bulguların sonuçları, elde edilen verilerin katkıları ve bunların bilim dünyasına ve katkıları paylaşılmaktadır.



## 2. AHŞAP KUSURLARI

Ahşap özellikleri ve kullanma olanaklarını değiştiren büyüme bozuklukları ile ahşapta kullanım yerinde bulunması istenmeyen özellikler kusur sayılır. Ahşap kusurları ağaç dikili halde iken odun oluşumu sırasında gerçekleşmeye başlar. Ahşap kusurları anatomik yapı ile ilgili, hava şartları nedeniyle, bitkisel ve hayvansal zararlıların oluşturduğu kusurlar olarak sınıflandırılmaktadır (Örs ve Keskin, 2001).

Ağaçta meydana gelen kusur ve hastalıkların genel nedeni ağacı çevre ve çevre şartlarından etkilenmesidir. Bu etkilenme dış hava koşullarından, ormandaki bulunan diğer canlılardan ve insanlardan dolayı kaynaklanmaktadır. Kusur ve hastalıkların bazıları kalıtsaldır. Ağaçta görülen kusur ve hastalıklar genelde dikili durumda iken meydana gelmektedir. Ancak, ağaçlar kesildikten sonra da kusur ve hastalıklar meydana gelebilmektedir. Ağaç işleri endüstri mühendisliği alanında kusurlu ve hastalıklı ağaçlardan elde edilen kerestelerinden elde edilecek ahşapların kullanılması estetik ve teknik açılarından uygun değildir.

Ağaç hastalıkları ve kusurları ağacı işlemeye başlamadan önce bilmeniz gerekenlerdir. Birtakım hastalıklar oymayı engellediği gibi birtakım hastalıklarda tüm işlem bittikten sonra ürün haline getirdiğiniz parçanın çatlamasına neden olabilir ağaç hastalıkları aşağıda verilmiştir.

### 2.1. Ağaç Kusurları

Ağaç kusurlarının meydana geliş biçimini ikiye ayırmak mümkündür. Doğal kusurların ağacın kalıtsal yapısından; yapay kusurlar ise insan ve diğer canlıların yaptığı tahribattan kaynaklanmaktadır. Ağaç kusurları şu şekildedir:

- 1- Budaklar: Ağaç dallarının gövdeler üzerinde bıraktığı iz şeklinde tanımlanmaktadır. Düşer ve kaynar budak olmak üzere ikiye ayrılmaktadır. Düşer budaklar kurumuş olan dalların, kaynar budaklar ise yaşamakta olan dalların bıraktığı izlerdir. Budak ahşabın dayanıklılığını oldukça azaltmaktadır. Budaklar sert olduğu için budaklı ağaçların işlenmesi oldukça zordur.

- 2- Eğri gövdeler: Genellikle orman kenarlarında yetişmekte olan ağaçların tek yönlü rüzgarların etkisinde kalması durumunda ya da bozuk bir ırktan kalıtım yoluyla meydana gelmekte olan bir kusur türüdür. Eğri gövdeli ağaçlardan elde edilen ahşabın işlenmesi zordur. İşlenme tamamlansa dahi sonrada lif yönünde eğilmeler devam edebilir.
- 3- Kaçık öz: Farklı nedenlerden sebebiyle ağacın öz kısmının gövde merkezinde değil, bir kenara daha yakın olması durumudur. Tek taraflı aşırı rüzgar ve tek taraflı beslenme kaçık özün oluşumunda etkili sebeplerdir. Kaçık özlü ağaçtan elde edilen ahşap homojen yapıda olmadığından, zor işlenmektedir.
- 4- Lif kıvrıklığı (spiral liflilik): Elyaf dönüklüğü ya da burukluk olarak tanınan bu kusur türü ağacın yukarıya doğru dönerek büyümesi şeklinde oluşmaktadır. Bu kusurda lifler, ağaç gövdesi eksenine paralel olmak yerine büyük veya büyük bir açı teşkil edecek şekilde spiral şeklinde gövde etrafında dolanarak büyümesi durumudur. Lif kıvrıklığının önemli sebebi kalıtsal olabileceği gibi orman kenarında bulunan ağaçların tek taraflı kuvvetli aşırı rüzgarların etkisinde kalmasında da meydana gelebilmektedir. Kıvrık lifli tomrukları işlerken, düz damarlı tomruklara göre daha fazla lif çapraz olarak kesilir ve bölünür, bu da kesilen ağaç malzemenin mekanik özelliklerini azaltmaktadır. Kıvrık lifli ahşabın dayanıklılığı çok düşürmekte ve işlenmesini zorlaştırmaktadır.
- 5- Çift özlü ağaçlar: Genel olarak ormanda yan yana büyümüş aynı türden olan ağacın büyüme esnasında birleşerek kaynaşmasından meydana gelmekte bir kusur türüdür. Çift özlü ahşap malzemedeki çalışma miktarı fazladır ve dengesizdir.
- 6- Uurlu ağaçlar: Ur, ağaçlarda yıllık halkaların birbiri üzerine yığılması sonucu gövdenin dışına taşma durumudur. Uurların meydana gelmesinde, ağacın düzensiz büyümesi ve aldığı yarayı onarması için çalışması gibi nedenler bulunmaktadır. Uurlu ahşap malzeme masif olarak kullanılmaz mümkün değildir. Ancak, kaplama olarak değerlendirilebilmesi mümkündür. Uurlu ağaçlar masif halde işlenmesi oldukça zordur. Ancak kaplama üretiminde tercih edilerek estetik görüntülü ürünler üretilebilir.
- 7- Yıllık halka yapraklanması: Farklı sebeplerle yıllık halkaların birbirine kaynaşmaması durumu oluşan bir ahşap kusur türüdür. Ormanda bulunan başka bir ağacın baskısı ve gölgesi altında büyümüş ağacın bu baskıdan birden kurtularak hızlı ve ani büyümesi bu kusurun oluşmasında bir nedendir.
- 8- Çatlaklar: Çok hızlı ve ani kuruma, kış mevsiminde ağaç bünyesindeki bulunan suyun donması ve öz çevresinde bulunan rutubetin hızlı bir şekilde kaybolması durumunda

kerestenin muhtelif yerlerinde meydana gelen yarılmalardır. Öz çatlakları, çevre çatlakları, don çatlaklar şeklinde meydana gelmektedir.

- 9- Çevre çatlakları: Genelde ormandan kesilmiş olan tomrukların yaz aylarında güneş altında kalması ile ani bir şekilde kuruma sonucunda oluşmaktadır. Bu kusur türünde meydana gelen çatlaklar mantarların oluşması ve yaşaması için uygun ortam sağlamaktadır.
- 10- Öz çatlakları: Öz ve çevresindeki rutubetin birden ortadan kalkmasıyla meydana gelen kusur türüdür.
- 11- Don çatlakları: Kış aylarında dikili haldeki ağaçta bulunan fazla su donmaktadır. Donmuş halde bulunan su ağacın çevresine basınç yaparak öz ışınlar yönünde çatlaklar oluşturmaya başlar. Bu durumda don çatlakları meydana gelmektedir.
- 12- Yıldırım çarpması: Meşe, ceviz, dişbudak gibi bazı ağaç türü tomruklarında, yıldırım düşmesi sonucunda gövdelerde ortaya çıkan yanma durumudur. Bu ağaç türleri genelde kaplama üretiminde kullanılmaktadır.
- 13- Reçine keseleri: İğne yapraklı ağaç türlerinde meydana gelen kusurdur. Budak olan bölgelerde ve ağacın bazı bölgelerinde reçine birikintileri şeklinde meydana gelir. Reçine keseleri kusurun oluşmasına bazı manatların ve böceklerin neden olduğu bilinmektedir (Gürtekin ve Oğuz, 2002).

## 2.2. Ağaç Hastalıkları

Ağaçlarda hastalıklar, ya ağacın iç yapısından dolayı kaynaklanan etkenlerle ya da dışarıdan gelen etkilerle meydana gelmektedir. Ayrıca, her iki faktörün bir arada etkisiyle de gelişebilir. Mantarlar, hücrelerin suyunu emerek beslenir ve salgıladıkları toksinlerle hücreleri zehirlerler; bu da hücrelerin ölmesine yol açmaktadır. Bu süreç diğer hücrelere de yayılır ve ağaç hastalanır, lekeler meydana gelir ve çürümeye başlar.

Mantarlar, +3 °C ile +38 °C arasındaki sıcaklık koşullarında ve en az %20 nem oranında faaliyet gösterirler. Mantarları tamamen yok etmek için sıcaklığın 66 °C'nin üzerine çıkması gereklidir. Hastalığa yakalanmış ağaçlarda renk değişikliği ve direnç kaybı gözlemlenir, bu da yüzey kaplama ve empenye işlemlerinde etkili sonuçlar alınmasını engeller. Ağaçlarda en çok görülen hastalıklar şunlardır:

Ağaçta hastalığı meydana getiren mantarlardır.

- 1- Ardaklanma: Beyazımsı renkte, oval veya dairesel biçimde ortaya çıkan beneklerdir. Ardaklanma, ağacın tüm yüzeylerinde veya sadece diri odun kısmında gözlemlenebilir. Ağaç, beyaz ve yumuşak bir hal alır. Depolarda üst üste yığılmış kerestelerde de ardaklanma görülebilir.
- 2- Öz çürüğü: Genellikle dikili kayın ağacının özünde ortaya çıkar. Ağaç özünün çürümesiyle beyaz renkte bir görünüm alır.
- 3- Yalancı göbek: Bu hastalık, kayın ağacının özünde ve öz çevresinde iç oduna benzer bir renk değişikliği olarak kendini gösterir. Zamanla renk kızılılaşır. Yalancı göbek, kayın ağacında genellikle 80-100 yaşlarındaki dikili ağaçlarda oluşur.
- 4- Delikli çürük: Bu hastalık türü, geniş yapraklı ve iğne yapraklı ağaçların hepsinde görülebilir. Ağaçların dal ve budaklarından giren mantarlar, göbek odununa kadar ilerleyerek gövdeyi tahrip ederler. Mantarlar, gövdede hücre zarının odunlaşmış kısmını tüketir ve küçük delikler açarlar. Ağaç kesildikten sonra, ortam mantarların yaşaması için uygun olduğunda tahribat devam edebilir.
- 5- Morarma: Bu hastalık türü hem iğne yapraklı hem de geniş yapraklı ağaçlarda görülebilir. Ancak, hastalık genellikle kesilmiş ve nemli ortamlarda bırakılmış tomruklarda daha sık ortaya çıkar. Dikili ağaçlarda ise bu hastalığa daha nadir rastlanır.
- 6- Mavi çürük: Kesildikten sonra toprağa temas eden iğne yapraklı ağaçlarda ortaya çıkar ve en çok kızılçamda görülür. Hastalığa yakalanan ağacın diri odunu mavi renkte bir görünüm kazanır.
- 7- Siyah çürük: İğne yapraklı ağaçlardan ladin ve göknar, kesildikten sonra uzun süre açıkta bırakıldıklarında bu hastalığa yakalanabilirler. Hastalık, ağacın diri odununda siyah çizgiler şeklinde kendini gösterir. Siyah çürük, ahşabın direncini azaltmaz; sadece renk bozukluğu meydana getirir.
- 8- Kırmızı çürük: Zararlı mantarların etkisiyle göbek odununun kırmızı-kahverengi bir renge dönüşmesiyle ortaya çıkar.
- 9- Beyaz çürük: Bazı mantar türlerinin ağaca girerek besin suyunu mayalaması, pıhtılaştırması ve küflenmesine bağlı olarak meydana gelir.
- 10- Yıllık halka çürümesi: Beslenme yetersizliği veya bozukluğu nedeniyle bir veya daha fazla yıl halkasının hasar görmesidir (Gürtekin ve Oğuz, 2002).

Bu hastalık ve kusurların yanı sıra, ormanda dikili durumda olan veya kesildikten sonra ağaçlara zarar veren çeşitli canlılar bulunmaktadır. Bu canlılar, açtıkları yaralardan

mantarların girmesine neden olabilirler. Ađaç, bu yarayı kendiliđinden iyileřtirmeye alıřır, ancak bu srete byme yavařlayabilir hatta durabilir. Onarma sreci uzun zaman alır ve gvdenin dzgnlđ bozulur. Ađalara zarar veren canlılar arasında geyik, karaca, sincap, fare, kunduz, ađakakan, bcekler ve kurtuklar yer alır (Grtekin ve Ođuz, 2002).



### 3. LİTERATÜR ÖZETİ

Ağaçşileri endüstrisinde, ahşap sınıflandırma ve sınıflandırma işlemleri büyük bir öneme sahiptir. Geleneksel yaklaşımla, ahşap sınıflandırma işlemi manuel olarak gerçekleştirilmektedir. Ancak günümüzde bu süreci hızlandırmak amacıyla otomatik sıralama makineleri yaygın bir şekilde kullanılmaktadır. Aynı mekanik cihazlar grubunu kullanarak ve sıralama görevini değiştirerek, farklı sıralama görevleri kolayca tamamlanabilir. Bu iki görev arasında sınıflandırma, belki de en büyük öneme sahip olanıdır (Hu ve diğerleri, 2019).

Ahşap yüzey kusurlarını tespit etmek için kullanılan yöntemler, bilgisayar analiziyle ahşap yüzey görüntülerini incelemeyi içermektedir. Bu işlem için genellikle bir renkli kamera sistemleri (Charge Coupled Device - CCD) kamera kullanılır ve ek ekipman gerektirmez, bu nedenle maliyeti düşüktür. Görüntü tanıma yöntemi, operatör müdahalesine uygun olduğu için ahşap üretiminde sıkça tercih edilir. Ahşap kusurlarını tanıma işlemi, görüntü analiz algoritmasının tasarımına dayanır ve genellikle dijital görüntü işleme kullanılmıştır (Xie, 2013).

Bilgisayarlı görüde genellikle ilk aşama olarak, görüntü gri tonlama dönüşümü, histogram eşitleme, uzamsal veya frekans alanı filtrelemesi gibi ön işlemlerden geçirilir. Daha sonra, ahşap kusur görüntüleri kusur özelliklerini çıkarmak için işleme tabi tutulur. Son olarak, görüntüleri sınıflandırmak için bir makine öğrenme algoritması kullanılır (Chen ve diğerleri, 2023).

Kusur tespiti ve segmentasyon, ahşap, tekstil, ilaç gibi farklı endüstrilerdeki otomatik üretim hatlarında kalite kontrolünün kritik adımlarından birisi olduğunu ifade etmişlerdir (Gao ve diğerleri, 2021; Shahrabadi ve diğerleri, 2022; Liu ve diğerleri, 2021). Son yıllarda, ahşap kaynaklarının azalması, orman kaynaklarının daha sürdürülebilir bir şekilde kullanılması araştırmacıların önemli bir odak noktası haline gelmiştir. Ahşap ürünlerdeki yüzey kusurlarını hızlı ve etkili bir şekilde tespit ederek, ahşabın kullanım verimliliği artırılabilir ve fazla ahşap tüketimi azaltılabileceğini bildirmişlerdir (Pölzleitner ve Schwingshagl, 1992; Schmoldt ve diğerleri, 1997; Norlander ve diğerleri 2015; Qui, 1996).

Dijital görüntü işleme ve yapay zeka algoritmalarının bir araya getirilmesi, ahşap budak kusurlarının tespiti ve sınıflandırılmasında sıkça tercih edilen bir yöntemdir. Bu belirtilen süreçte özellikle sabit özellik çıkarımı ve sınıflandırma tanıma teknolojisi oldukça yaygın bir biçimde kullanılmakta olduğunu bildirmişlerdir (Qi ve diğerleri, 2010; Jie ve diğerleri, 2019; Mu ve Qi, 2009).

### **3.1. Klasik Makine Öğrenimi ile Ahşap Kusur Tespiti**

Ahşap spektral verilerin boyutunu azaltmak için temel bileşen analizi kullanılan bir çalışma yapılmıştır. Bu çalışmaya ek olarak, bu verileri işlemek ve ahşap türlerini sınıflandırmak ve tanımlamak amacıyla bir geriye yayılım (BP, backpropagation) sinir ağı modeli oluşturmuşlardır. Çalışmanın önemli bir bulgusu, ağaç türü çeşitliliği arttıkça tanı doğruluğunun da arttığıydı. Yani, daha fazla ağaç türü olduğunda, tanıma doğruluğu artma eğiliminde olduğunu bildirmişlerdir (Li ve diğerleri, 2019).

Ahşap kusur görüntüleme segmentasyonu için Gabor filtrelerini kullanarak özellik parametreleri çıkartılmıştır. Çok yönlü filtreleme ile %98,29'luk bir segmentasyon doğruluk oranına ulaşıldı ve bu gri seviye eş-zamanlı matris tabanlı segmentasyon yöntemlerinin performansını aştığını bildirmişlerdir (Raheja, 2013).

Ahşap kusurlarını gri tonlamalı görüntülerde tanımak için Gabor filtre yöntemini dalgacık dönüşümü ile birleştirerek karmaşık arka planın etkisi altında ahşap kusurlarını tanıyabilen çok kanallı Gabor filtresini kullanıldığı bir çalışmada, düşük ışıktaki iyileştirme (Low-Light Enhancement), çok ölçekli özellik çıkartma (Multi-Scale Feature Extraction), ve hata tespiti önerilmiştir. Önerilen model, özellikle yüzey kusurlarını tespit etmek için geliştirilmiş nesne tespit ağı ile karşılaştırılmış ve "Fine Cans Defect" veri kümesinde %94,3'lük ortalama doğruluk elde edilmiştir (Hu ve diğerleri, 2021).

Ahşap yüzey kusurlarını tanımak için bir özellik birleştirme tabanlı bir tespit yöntemi önerilen bir çalışmada, ilk olarak ahşap yüzey kusuru görüntülerinden HOG çıkartılıyor ve bu HOG özelliklerinin boyutunu azaltmak için çekirdekli temel bileşen analizi (KPCA) kullanılıyor. Daha sonra, boyutu azaltılmış HOG-KPCA özellikleri ile ahşap yüzey kusuru görüntülerinden çıkartılan matris GLCM birleştirilerek HOG-KPCA-GLCM özellikleri

oluşturuluyor. Sınıflandırma işlemi için destek SVM kullanılıyor. Bu yöntemle %91,26'lık bir doğruluk elde edilmiştir (Wu ve diğerleri, 2022).

Ahşap plaka görüntülerinden ahşap kusurlarını tespit etmek için yapılan bir çalışmada PCA ve sıkıştırılmış algılama kullanarak yeni bir yöntem geliştirdi. 25 farklı özellik, özellikle geometri, bölgesel, gri tonlama doku ve değişmez anlık özellikler, ahşap plaka görüntülerinden çıkarıldı ve PCA ile birleştirildi. Test edilen 50 Xylosma örneğinde %92'lik bir doğruluk elde edildiğini göstermiştir (Zhang ve diğerleri, 2015).

Bilgisayar teknolojisinin ilerlemesiyle birlikte, araştırmacılar, ahşap kusurlarının tespiti ve sınıflandırması için farklı özellik tabanlı bilgisayarlı görme algoritmalarını kullanmaya başlamışlardır. Bu algoritmalar arasında histogram analizi, destek vektör makineleri gri seviye uyum matrisi gibi yöntemler bulunmaktadır. Veri artırma işlemi, orijinal görüntülerin gri tonlamalı dönüşümü, uzamsal alanı ve histogram işlemesi gibi teknikler kullanılarak gerçekleştirilir. Ardından, görüntüler segmentasyon edilir ve özellikler çıkarılır. Son aşamada ise ahşap kusurları, BP sinir ağı ve SVM gibi yapay zeka yöntemleri kullanılarak tanımlanmıştır (Song ve diğerleri, 2015; Gu ve diğerleri, 2010; Ye ve diğerleri, 2017; Yan ve diğerleri, 2018; YongHua ve Cong, 2015).

GLCM tabanlı özellikler ve parçacık sürü optimizasyonu PSO eğitilmiş sinir ağı kullanılarak ahşap kusur sınıflandırması üzerine yapılan çalışmada, önerilen yöntem, farklı ahşap kusurlarının 90 örneğinden gri seviye birlikte oluşum matrisine dayalı özellikleri çıkarılmasıdır. GLCM tabanlı özellikleri kullanarak kusurları sınıflandırmak için bir Parçacık Sürü Optimizasyonu eğitilmiş ileri beslemeli sinir ağını çalışmalarında kullanılmışlardır. Eğitim veri seti için ağın ortalama kare hatası 0,3483 olarak bulunurken, test veri seti için doğruluk oranı %78,26 olarak bulunmuştur (Qayyum ve diğerleri, 2016).

Canlı budaklar, ölü budaklar ve çatlaklı budaklar gibi üç farklı türdeki kusurları tanımlamak ve tespit etmek için PSO algoritmasını kullanmışlar ve %78,26 oranında doğruluk oranı bulmuşlardır (Yang ve Yu, 2017).

Ahşap kusur tespitinde, gri tonlamalı birlikte oluşum matrisi ve doku enerjisi ölçüsünü kullanarak doku özelliklerini çıkarmışlar ve sonuçlar, ahşap budak kusurlarının genel

sınıflandırma doğruluğunun sırasıyla %84,3 ve %90,5 olduğunu göstermişlerdir (Kamal ve diğerleri, 2017). Yapılan bir diğer çalışmada, meşe ahşabındaki canlı budak, ölü budak ve çatlaklar gibi üç farklı kusuru tespit etmek için temel bileşen analizi ve basınç algılama yöntemlerini kullanmışlardır. Deney sonuçları, PCA özellik füzyonunun algılama hızını artırabileceğini göstermiştir. SOM sinir ağı kullanıldığında, algılama doğruluğu %87'den %92'ye yükseldiği görülmüştür (Zhang ve diğerleri, 2015).

SVM kullanarak kereste kusurlarının tespiti ve sınıflandırması için çoklu öznitelik çıkarımı üzerine yapılan bir çalışmada, özellik hızlandırıcı Speed Up Robust Features (SURF) ve yerel ikili desen tanımlayıcı LBP gibi yöntemler kullanılmıştır. Budakları, çatlakları ve kusurları sınıflandırmak için SVM kullanılmıştır, nihai sonuçlar ortalama %93 doğruluk ile tespit ettiği bildirilmiştir (Hittawe ve diğerleri, 2015).

Ahşap kusurlarının sınıflandırılması ve bölümlenmesi için dışbükey optimizasyonu ve Otsu bölütleme yöntemini kullanıldığı bir araştırmada yapılmıştır. Bu araştırmada delikler, çatlaklar, canlı budaklar ve ölü budaklar gibi dört farklı türdeki kusurların tanımlanmasını sağlamıştır. Önceki tespit yöntemleriyle karşılaştırıldığında, özellik tabanlı bilgisayarlı görüntü tespiti in daha az maliyetli, daha verimli ve daha doğru olduğu gözlemlenmiştir (Zhang ve diğerleri, 2015).

Ahşap kusur tespiti için hazırlanmış bir yüksek lisans tezinde, ahşap kusurlar tespit edilmiş ve sınıflandırılmıştır. Denetimli bir sınıflandırıcı önce Adaboost ile eğitilmiş ve daha sonra leke türü kusurların renklerini çıkarmak için kullanılır ve görüntü nesnelere öznitelikleri çıkarmak için integral görüntü, Dinamik programlama (DP) ve Hough dönüşümü gibi yöntemler birleştirilir. Önerilen yöntemler, sınıflandırma oranını (düşük yanlış oranı) en üst düzeye çıkarmak ve bu arada endüstri uygulamalarında önemli bir faktör olan hesaplama süresini en aza indirmek için seçilmiştir. Ahşap görüntüleri üzerinde deneyler yapılmış ve sonuçlar önerilen yaklaşımın etkinliğini gösterdiği bildirilmiştir (Jabo, 2011).

Budaklılık oranına göre parke kalite sınıflarını görüntü işleme teknikleri kullanılarak belirlenmesi üzerine yapılan bir doktora çalışmada, kusurlar dalgacık moment (DM), moment değişmezleri yöntemi (IM), GLCM, IM+GLCM ve DM+GLCM momentleriyle öznitelikleri çıkarılmıştır. Sınıflandırılmada YSA, doğrusal diskriminant analiz LDA,

KNN, ve Naive Bayes (BAYES) sınıflandırıcıları kullanılarak budak kusur tespiti gerçekleştirilmiştir (Çetiner, 2014).

SVM adı verilen ortak bir sınıflandırıcı ile LBP adı verilen etkili bir özellik çıkarma tekniği önerilmiştir. Ahşap yüzeyindeki doğal kusurları sınıflandırmaktır. İlk olarak, RGB görüntüleri gri tonlamalı görüntülere dönüştürmek için ön işleme uygulanmıştır. Sonra SVM sınıflandırıcısını uygulanmıştır, sınıflandırma için ve önerilen tekniğin performansını ölçmek için kullanılmıştır. Deneysel sonuç,  $P=8$  ve  $R=1$  ile dengeli veri setinde elde edilen ortalama doğruluğun %65 olduğunu göstermiştir (İbrahim ve diğerleri, 2021).

### 3.2. Derin Öğrenme ile Ahşap Kusur Tespiti

Ahşap yüzey görüntülerinde karar vermede ESA gibi çeşitli sinir ağları da mevcuttur (Packianather ve Drake, 2005). Ahşap kusur sınıflandırması için BP sinir ağı modeli, SVM sınıflandırma modeli ve ESA evrimsel sinir ağı modelinin etkililiğini karşılaştırmıştır. Sonuç olarak, ESA evrimsel sinir ağı modeli ve SVM modeli, ahşap kusur tespiti ve sınıflandırması için BP sinir ağı modeline göre daha uygun bulunmuş ve daha yüksek sınıflandırma doğruluğuna sahip olduğu sonucuna ulaşılmıştır (Luo, 2019).

Derin evrimsel sinir ağı (DESA) aracılığıyla bir lazer tarayıcı kullanılarak toplanan ahşap görüntülerinden ahşap özelliklerini algılamak ve kusurları otomatik olarak sınıflandırmak için bir öğrenme yöntemi bir arada kullanılmıştır. DESA modelinin eğitimi ve değerlendirilmesi için 600 adet kızılçam ve kafur çamı ağacından bir ahşap kusuru veri seti toplanmış ve ardından eğitim (42 750 budak, 40 050 çatlak ve 41 200 küf lekesi dahil), doğrulama ve veri artırmanın ardından test veri setleri işlenmiştir. Bu geliştirilmiş DESA modelini kullanarak genel olarak %99,13'lük bir doğruluk elde ettik; yani algılama için yalnızca 1,12 sn gerçekleşmiştir (He ve diğerleri, 2020).

Derin öğrenme ile ahşap kusur tespiti ile ilgili hazırlanmış olduğu yüksek lisans tezinde, ESA, iki boyutlu verilerde kullanılan derin sinir ağlarından birisini kullanmıştır. ESA modelinin farklı parametreler kullanılarak eğitilmesi ve test edilmesi için budak, çatlak ve normal gibi üç set girdi verisi hazırlanmıştır. Farklı konfigürasyonların sonuçları, karşılaştırıldı ve analiz edilmiştir (Ren, 2019).

Bir ahşap firmasının deneyimli çalışanları tarafından laboratuvarında geliştirilen ve etiketlenen veri toplama ekipmanı kullanılarak 2838 adet ahşap kaplama görüntüsü toplanmıştır. Ardından, bir göz atma ağı ve birçok kanallı maske R-ESA içeren ahşap kaplama kusurlarını tespit etmek için entegre bir model olan bakış çok kanallı maske bölgesi evrişimli sınır ağı (R-ESA) inşa edilmiştir. Çok Kanallı Maske R-ESA daha sonra kusurları sınıflandırmak ve bulmak için kullanılmıştır. Deneysel sonuçlar, önerilen yöntemin %98,70'lik bir genel sınıflandırma doğruluğuna ve %95,31'lik bir ortalama kesinliğe ulaştığını ve 50 standart görüntü ve 50 kusurlu görüntüden oluşan bir partiyi algılamak için yalnızca 2,5 saniyeye ihtiyaç duyulduğunu gösterdiği bildirilmiştir (Shi ve diğerleri, 2020).

İyileştirilmiş single-shot multibox detector (SSD) algoritmasına dayalı masif ahşap panellerdeki kusurların tespiti araştırmıştır. Akagi ve Pinus sylvestris ağaçlarından iki tür ahşabın yüzey görüntülerini toplamak için renkli şarj bağlantılı bir cihaz kamerası kullanılmıştır. 200×200 piksel boyutunda ahşap budaklar, ölü budaklar ve kontrol kusurları içeren toplam 500 adet görüntü elde edilmiştir. Transfer öğrenme yöntemi, bir hedef tespit algoritması olan tek atış çoklu kutu dedektörünü (SSD) uygulamak için kullanıldı ve algoritmayı iyileştirmek için DenseNet ağı tanıtılmıştır. Üç tür kusuru, kaynar budaklar, düşer budaklar ve kontrolü tespit etmek için ortalama kesinlik %96,1 olduğu saptandığı bildirilmiştir (Ding ve diğerleri, 2020).

Üretimden önce ahşap kusurları tanımak için Adaboost kademeli sınıflandırıcı algoritması, geleneksel bölütleme yönteminin çoklu nesne işleme için etkili olmadığı problemini çözmek için görüntüdeki ahşap yüzey kusur bölgesinin aday çerçevelerini çıkarmak için kullanılmıştır. Geleneksel sınıflandırma yöntemlerindeki öznelik seçme zorluğunun üstesinden gelmek için girdi aday kutularını sınıflandırmak için kendi kendine öğrenme yeteneğine sahip bir ESA modeli kullanılır. Test için 200 çok hedefli örnek seçilmiştir ve sonuçlar hatırlama oranının %94, tespit doğruluğunun %99 ve sınıflandırma doğruluğunun %97,9 olduğunu göstermiş olduğu bildirilmiştir (Wang ve diğerleri, 2019).

Derin öğrenme özelliği çıkarma yöntemi ve aşırı öğrenme makinesi (ELM) sınıflandırma yöntemi, ahşap görüntü kusur tespiti için derinlemesine bir aşırı öğrenme makinesi modeli oluşturmak üzere birleştirilmiştir. Tek başına ESA algoritması, hatalı kusur konumları, eksik kusur konturu ve sınır bilgisi ve kusur tiplerinin yanlış tanınmasını sağlama

eğilimindedir. Alt örneklemelessiz shearlet dönüşümü (NSST) burada, görüntü işleminin karmaşıklığını ve hesaplamasını azaltan ahşap görüntüleri ön işlemek için kullanılır. Ahşap görüntülerin derin algoritma tasarımını yönetmek için ESA uygulanır. Ahşap kusuru tanıma doğruluğu yalnızca 187 ms'lik bir test süresinde %96,72'ye ulaştığı bildirilmiştir (Yang ve diğerleri, 2020).

Ahşap budakların tespitinde Transfer Learning (TL)-ResNet34'ün eğitim veri setinde %99,22 gibi yüksek bir tanıma oranı ve doğrulama veri setinde %2,83 gibi düşük bir eğitim kaybı elde ettiğini göstermektedir. Test veri setine TL-ResNet34 uygulandığında genel doğruluk %98,69'a ulaşmış ve kayıp eğrisi ile doğruluk eğrisinin dalgalanma aralıkları küçüldüğü bildirilmiştir (Gao ve diğerleri, 2021).

Ahşap budak tespiti için You Only Look Once (YOLO)-v5 detektörü kullanıldığı bir araştırmada, yüzeydeki budak türleri özellikleri öğrenilerek adaptif olarak çıkarılmış ve daha sonra budaklar renk ve doku bakımından farklılık gösterse de budak kusurları doğru bir şekilde tanımlanmıştır. Önerilen yöntem YOLO-v3 SPP ile karşılaştırılmıştır ve iki veri setinde daha hızlı R-ESA sonuçları incelenmiştir. Deneysel sonuçlar, YOLO-v5 modelinin yüzey budak kusurlarını tespit etmede en iyi başarılı olduğunu göstermiştir. Veri kümesi 1'deki F-Skoru %91,7 ve veri kümesi 2'nin F-Skoru %97,7 olduğu sonucu verdiği bildirilmiştir (Fang ve diğerleri, 2021).

Gerçek zamanlı defekt tespiti ve sınıflandırması için derin öğrenme tabanlı bir yaklaşım kullanılmıştır. Bu yaklaşımda, özellik çıkarımı için derin ESA kullanılmış ve daha hızlı tespit için bölge tabanlı ESA (Faster R-ESA) ile birlikte omurga ağı olarak MobileNet-V3 tercih edilmiştir. Özellik çıkarımı için MobileNet V3'ü büyük görüntü veri kümesi üzerinde önceden eğitilmiştir. Hataların tespiti ve sınıflandırılması için Faster R-ESA kullanılmış, düşük maliyetli ve düşük bellekli GPU (Grafik İşleme Birimi) kullanarak giriş video karelerinden saniyede ortalama 37 kare gerçek zamanlı algılama ve sınıflandırma yapabilmektedir. Yöntemin, kusurları tespit etme ve sınıflandırmada genel olarak %99'luk bir doğruluk elde ettiği bildirilmiştir (Mohsin, 2022).

YOLOv4-Tiny mimarisine dayanan dört tip ahşap kusurunun tespiti için hafif bir nesne algılama modeli önerilmiştir. Modelin doğruluğu, YOLOv4-Tiny'nin kayıp işlevinin, Kavşak üzerinden birliği nesnellik kaybına dahil etmesi için değiştirilerek iyileştirildi.

Sonuçlar, yapılan iyileştirmenin modelin doğruluğunu başarılı bir şekilde artırdığını ve en iyi modelin saniyede 225,22 kare hızla çalışırken ortalama %88,32'lik bir kesinlik elde edebildiğini göstermiştir (Lim ve diğerleri, 2022).

Bu bilgiler ve literatür özeti ışığında, bu tez çalışmasında ahşap malzemelerin yüzeylerindeki kusurların tespiti için kullanılan farklı görüntü işleme yöntemleri ve sınıflandırma algoritmalarının performanslarını bulmak, incelemek ve karşılaştırmak amaçlamaktadır. Bu çalışmada, ahşap malzemelerdeki kusurların erken tespitini kolaylaştırmayı hedeflemekte ve farklı özellik çıkarım yöntemlerinin ve sınıflandırma algoritmalarının bu amaç doğrultusunda nasıl performans gösterdiğini değerlendirilmesi ve literatüre katkı olan katılarının belirlenmesi adına mevcut literatür özeti çalışmanın bu bölümünde verilmiştir.

## 4. GÖRÜNTÜDEN ÖZİNİTELİK ÇIKARIMI

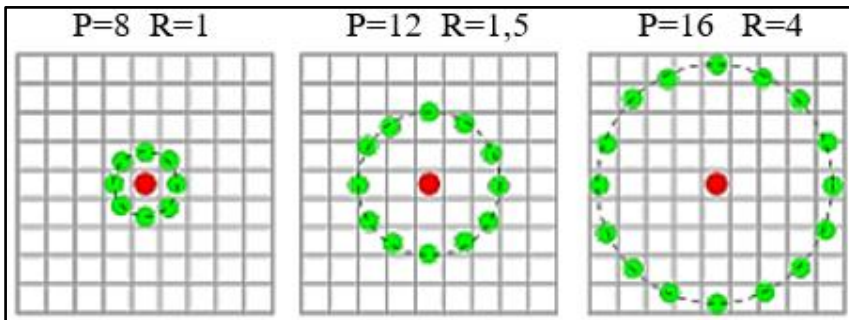
Bilgisayarlı görü ve makine öğrenimi alanlarında sıkça kullanılan bir konsepttir. Bu süreç, bir görüntüden anlamlı özelliklerin çıkarılması ve bu özelliklerin daha yüksek düzeyde anlamların elde edilmesini sağlamak amacıyla kullanılmasıdır. Öz nitelikler, bir nesnenin veya desenin tanınması için önemli olan görsel özelliklerdir.

### 4.1. Yerel İkili Örüntüler Operatörü (Local Binary Pattern)

LBP operatörü, gri tonlamadan bağımsız bir doku ölçüm yöntemidir. Her piksel için bir etiket oluşturularak, bu etiketler birler ve sıfırlardan oluşur. Etiketler, merkez pikselin  $N \times N$  komşuluğundaki piksellerin karşılaştırılmasıyla türetilir. Genel olarak,  $LBP_{P,R}$  üç farklı dairesel komşulukla tanımlanabilir.  $P$ , komşu sayısını temsil ederken,  $R$  örnekleme yarıçapını ifade eder. Şekil 4.1'de çeşitli LBP operatörleri gösterilmektedir (Yang ve Chen, 2013; Tuncer ve Engin, 2017). LBP, piksellerin komşu piksellerle karşılaştırıldığı bir yöntemdir. Her piksel, etrafındaki piksellerle karşılaştırılır ve bu karşılaştırmalardan bir desen oluşturulur. LBP'nin matematiksel formülü şöyle ifade edilebilir:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (4.1)$$

Burada,  $LBP_{P,R}(x_c, y_c)$  merkezi pikselin LBP değerini temsil eder,  $P$  komşu piksellerin sayısını,  $R$  yarıçapı,  $(x_c, y_c)$  merkezi pikselin koordinatlarını,  $g_p$  komşu piksellerin yoğunluğunu,  $g_c$  ise merkezi pikselin yoğunluğunu,  $s$  eşik değerini ifade eder (Ojala ve diğerleri, 2002). Dairesel LBP operatörleri Şekil 4.1'de verilmiştir.



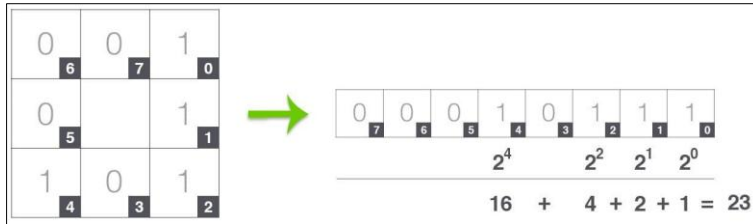
Şekil 4.1. Çeşitli dairesel LBP operatörler (Ojala ve diğerleri, 2002)

LBP, özellikle görüntü işleme ve desen tanıma alanlarında kullanılan bir özellik çıkarma yöntemidir. LBP, bir pikselin komşu pikselleriyle karşılaştırılarak bir desen oluşturur ve bu desenin ardından bir özellik vektörü elde edilir. LBP'nin temel matematiksel işlemleri: Piksel Komşulukları: LBP, bir pikselin etrafındaki komşu pikselleri kullanarak desenleri tanımlar. Bir pikselin komşu pikselleri genellikle 8 yönde (örneğin, doğu, batı, kuzey, güney ve köşe yönleri) belirlenir.

Eşikleme (Thresholding): Her bir pikselin değeri, ortamdaki piksellerle karşılaştırılarak bir eşik değeriyle karşılaştırılır. Eğer bir pikselin değeri, eşik değerinden büyükse, 1 olarak etiketlenir; aksi takdirde 0 olarak etiketlenir.

Desen Oluşturma: Piksellerin eşikleme sonucunda elde edilen etiketler, bir deseni oluşturur. Bu desen genellikle bir ikili sayıdır ve örneğin, 8 bitlik bir sayı olabilir.

Histogram Oluşturma: Tüm pikseller üzerinde LBP desenlerinin oluşturulması sonucunda, bir histogram oluşturulur. Bu histogram, görüntünün LBP özelliklerini temsil eder (Ojala ve diğerleri, 2002). LBP operatörünün en önemli özelliği ise parlaklık değişimlerine karşı yüksek derecede dayanıklı olmasıdır (Tuncer ve Engin, 2017). Şekil 4.2'de LBP operatörü şeması örneği verilmiştir.



Şekil 4.2. LBP operatörü şeması örneği (Ojala ve diğerleri, 2002)

Matematiksel olarak, LBP algoritması aşağıdaki adımları içerir:

1. Piksel Komşulukları Tanımlama:  $P=(P_7,P_6,P_5,P_4,P_3,P_2,P_1,P_0)$ , burada  $P_0$  hedef piksel, ve  $P_i$  komşu piksellerdir.
2. Thresholding:  $T=(t_7,t_6,t_5,t_4,t_3,t_2,t_1,t_0)$ , burada  $t_i$  eşik değerleridir.
3. Desen Oluşturma:  $S = \sum_{i=0}^7 2^i \cdot (P_i \geq t_i)$  (4.2)

4. Histogram Oluşturma: Görüntü üzerindeki her piksel için  $S$  değerleri kullanılarak bir histogram oluşturulur (Ojala ve diğerleri, 2002).

Bu adımlar, bir görüntü üzerinde LBP özelliklerini çıkarmak için temel matematiksel işlemleri temsil eder. Bu özellikler genellikle görüntü tanıma ve yüz tanıma gibi uygulamalarda kullanılır LBP'nin bu özellikleri genellikle görüntü tanıma uygulamalarında kullanılır. Özellikle yüz tanıma, tekstür analizi ve nesne tanıma gibi alanlarda LBP'nin başarıyla kullanıldığı gözlemlenmiştir. LBP, görüntüdeki lokal desenleri belirlemek için güçlü bir özellik çıkarma tekniği olmasına rağmen, daha karmaşık desenleri tanıma yeteneği sınırlı olabilir. Bu nedenle, çeşitli görüntü işleme uygulamalarında kullanılmadan önce dikkatlice değerlendirilmelidir (Ojala ve diğerleri, 2002).

#### **4.2. Yönlü Gradyanlar Histogramı (Histogram of Oriented Gradients)**

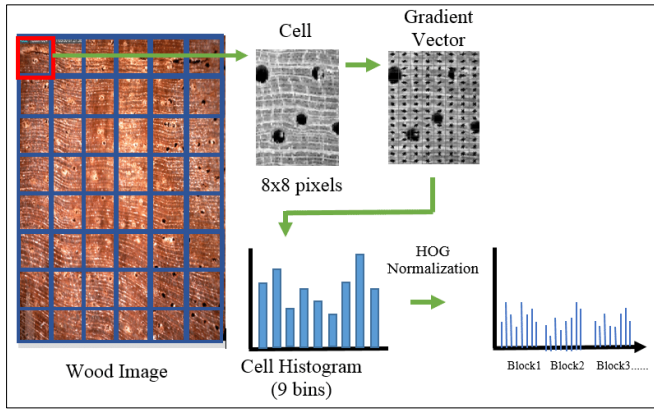
HOG tanımlayıcıları, nesne algılama amacıyla bilgisayarlı görü alanında yaygın olarak kullanılmaktadır. HOG temelinde, kenar konumu hakkında önceden bilgi sahibi olmadan yerel yoğunluk gradyanlarının veya kenar yönlerinin dağılımı ile yerel nesne görünümünü ve şeklini karakterize etmektir. Gradyan Hesaplama, görüntüdeki her pikselin gradyanını hesaplamak, yani pikseldeki renk değişimini belirlemek. Hücrelerin oluşturulması, görüntü, belirli boyutlardaki hücelere bölünür. Her hücre, bir bölgeyi temsil eden gradyan bilgilerini içerir. Hücrelerden blokların oluşturulması, hücreler birleştirilerek bloklar oluşturulur. Bu bloklar, daha büyük özellikleri temsil eder. HOG Vektörünün Oluşturulması, Bloklar birleştirilerek nihai HOG vektörü oluşturulur. Bu vektör, görüntüdeki desenleri ve kenarları temsil eden sayısal bir özellik vektörüdür (Dalal ve Triggs, 2005; Chandrasekhar ve diğerleri, 2010).

HOG tanımlayıcısının belirli bir görüntü konumu, görüntüyü hücre olarak bilinen küçük alt görüntülere bölerek elde edilir. Kenar histogramını biriktiren gradyanı niceleyerek her bir hücre içindeki yönelimleri yönleri ve bu tür her yönelim kutusu için gradyan ekleyen o hücre içindeki her piksel için büyüklüklerdir (Chandrasekhar ve diğerleri, 2010). Histogramlar tek bir vektör halinde yığılır ve ortaya çıkan vektör tanımlayıcıyı temsil eder. Gradyan, standart gri tonlamalı görüntü  $I$ 'nin sonraki filtre çekirdekleriyle filtrelenmesiyle hesaplanmaktadır (Muralidharan ve Chandrasekar, 2012).

HOG, nesne tanıma için kullanılan bir özellik çıkarım yöntemidir. Görüntüyü küçük hücrelere böler, her hücrede gradyanları hesaplar ve ardından bu gradyanların histogramlarını oluşturur. Matematiksel formülasyonu şu şekildedir:

$$HOG(x, y) = \theta(x, y) * G(x, y) \quad (4.3)$$

Burada,  $HOG(x,y)$  HOG vektörünü temsil eder,  $\theta(x,y)$  gradyan yönlerini,  $G(x,y)$  gradyanların yoğunluğunu ifade eder (Dalal ve Triggs, 2005). Ahşap tanımlamada HOG ile özellik çıkarımı örneği Şekil 4.3'te verilmiştir.



Şekil 4.3. Ahşap tanımda HOG ile özellik çıkarımı örneği (Sugiarto ve diğerleri, 2017)

### 4.3. Gri Seviye Eş-Oluşum Matrisi (Gray-Level Co-occurrence Matrix)

GLCM, 1973 yılında geliştirilen doku sınıflandırması için eski bir özellik çıkarma yöntemidir (Haralick ve diğerleri, 1973). Birçok doku sınıflandırma uygulamasında yaygın olarak kullanılmaktadır ve doku sınıflandırma alanında önemli bir özellik çıkarma yöntemi olmaya devam etmektedir. Bu, bir görüntüdeki piksel çiftleri arasındaki ilişkileri hesaplayan istatistiksel bir yöntemdir. Geleneksel yöntemlerde kontrast, korelasyon, enerji, entropi ve tekdüzelik gibi doku özellikleri oluşturulan GLCM'den hesaplanır (Petrou ve Sevilla, 2006). Ancak son yıllarda GLCM sıklıkla diğer tekniklerle birlikte kullanılmakta ve nadiren tek başına kullanılmaktadır (Recio ve diğerleri, 2005; Arivazhagan ve diğerleri, 2006; Kim ve diğerleri, 2006; Othmen ve diğerleri, 2008). Geleneksel uygulamaya ek olarak, GLCM'nin aşağıdakiler de dahil olmak üzere başka birkaç uygulaması da vardır. Tek boyutlu bir GLCM'nin (Tou ve diğerleri, 2008) doku özelliklerine ek olarak ikinci

dereceden istatistiksel yöntemleri ekleyen ve birinci dereceden istatistikler (Tou ve diğerleri, 2009) yerine ham GLCM'nin kendisini kullanan orijinal uygulama (Othmen ve diğerleri, 2008). GLCM, renk örtüşme matrislerinde farklı renk uzaylarına da uygulanabilmektedir (Porebski ve diğerleri, 2007).

GLCM, bir görüntüdeki gri seviye pikseller arasındaki ilişkileri yakalar. GLCM'nin matematiksel formülü şu şekildedir:

$$P(i, j, \theta) = \sum_{x=1}^N \sum_{y=1}^M I(x, y) \cdot I(x+i, y+j) \quad (4.4)$$

Burada,  $P(i, j, \theta)$  GLCM matrisini temsil eder,  $(i, j)$  değişiklik vektörünü,  $\theta$  açığı,  $I(x, y)$  ise görüntü piksel değerini ifade eder (Haralick ve diğerleri, 1973).

GLCM, bir gri tonlu görüntü üzerindeki pikseller arasındaki ilişkileri ifade eden bir özellik matrisidir. Genellikle görüntü işleme ve özellik çıkarma alanlarında kullanılır. GLCM'nin matematiksel modelini şu şekildedir:

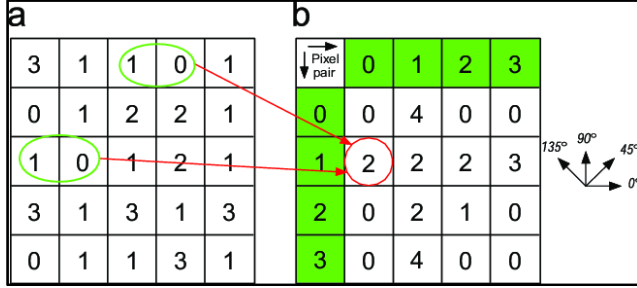
Verilen bir gri tonlu görüntüyü düşünelim ve  $I$  ile temsil edelim. Bu görüntü  $M \times N$  boyutlarında olsun, burada  $M$  satır sayısını ve  $N$  sütun sayısını temsil eder. Her bir pikselin gri seviyesini  $I_{(i,j)}$  olarak ifade edelim, burada  $i$  satır numarasını ve  $j$  sütun numarasını temsil eder. Gri seviye değerleri genellikle tamsayılardır.

GLCM, her bir pikselin belirli bir yönde (örneğin, yatay, dikey, çapraz) diğer piksellerle olan gri seviye ilişkilerini hesaplar. Bu ilişkiler, bir çift gri seviye için kaç kez bir arada bulduklarını içeren bir matris oluşturur. Matematiksel olarak:

$$P(i, j, d, \theta) = \frac{\text{sayı}(I(x, y) = i \text{ ve } I(x + d\cos(\theta), y + d\sin(\theta)) = j)}{\text{Toplam piksel sayısı}} \quad (4.5)$$

- $P(i, j, d, \theta)$ , GLCM matrisindeki belirli bir elemanı temsil eder.  $i$  ve  $j$  gri seviyelerini,  $d$  uzaklığı ve  $\theta$  yönelimi belirtir.
- $\text{sayı}(I(x, y) = i \text{ ve } I(x + d\cos(\theta), y + d\sin(\theta)) = j)$ , görüntüdeki belirli bir yönde ve uzaklıkta  $i$  ve  $j$  gri seviyelerine sahip piksel çiftlerinin sayısını ifade eder.
- Toplam piksel sayısı, görüntünün toplam piksel sayısını temsil eder.

Bu formül, GLCM matrisini oluşturmak için belirli bir yönde ve uzaklıkta piksel çiftlerinin sayısını normalize ederek kullanılır. Elde edilen matris, görüntünün belirli özelliklerini ifade eder ve genellikle özellik çıkarma ve desen tanıma uygulamalarında kullanılır (Haralick ve diğerleri, 1973). GLCM hesaplama örneği Şekil 4.4'te verilmiştir.



Şekil 4.4. GLCM hesaplama örneği (Verma ve diğerleri, 2015)

#### 4.4. Sobel

Sobel, kenar tespiti için kullanılan bir filtreleme yöntemidir. Sobel operatörleri görüntüdeki piksel yoğunluğunun gradyanını hesaplar. Sobel operatörlerinin matematiksel ifadesini açıklamak gerekirse, görüntü yoğunluğunun  $x$ -yönündeki gradyanını ( $G_x$ ) ve  $y$ -yönündeki gradyanını ( $G_y$ ) hesaplamak için evrişim kullanılır:

$$G_x = \sum_{i=0 \text{ to } 2} \sum_{j=0 \text{ to } 2} (\text{image}(x+i, y+j) * G_x\_kernel(i, j)) \quad (4.6)$$

Burada,

"görüntü ( $x+i, y+j$ )", görüntüdeki koordinatları ( $x+i, y+j$ ) temsil eder.

" $G_x\_kernel(i, j)$ ",  $x$ -yönündeki Sobel operatörü için çekirdek değerlerini ifade eder.

$$G_y = \sum_{i=0 \text{ to } 2} \sum_{j=0 \text{ to } 2} (\text{görüntü}(x+i, y+j) * G_y\_kernel(i, j)) \quad (4.7)$$

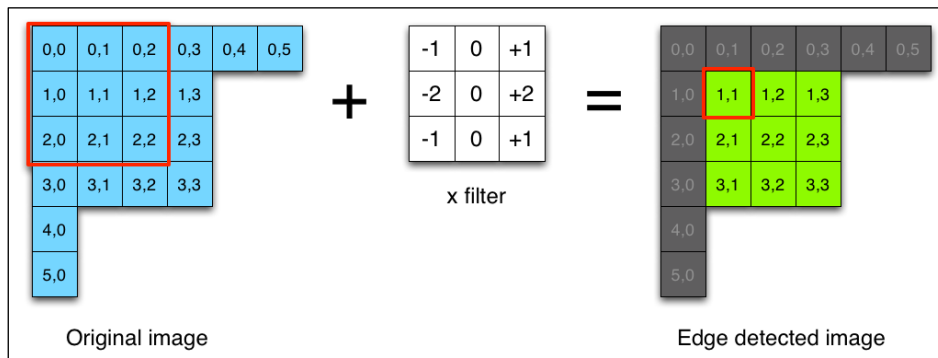
Burada,

"görüntü ( $x+i, y+j$ )" koordinatlarındaki piksel yoğunluğunu temsil eder. ( $x+i, y+j$ ) görüntüdeki piksel yoğunluğunu temsil eder.

" $G_y\_kernel(i, j)$ " önceki yanıtta verildiği gibi  $y$  yönündeki Sobel operatörü için kernel değerleridir (Duda ve Hart., 1973).

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (4.8)$$

Görüntü üzerinde Sobel filtresi uygulanırken, bu matrisler görüntü üzerinde belirli bir konumda gezdirilir ve her bir evrişim işlemi sonucunda elde edilen değerlerle yeni bir görüntü oluşturulur. Bu yeni görüntü, orijinal görüntüdeki kenarların yönünü ve şiddetini temsil eder (Szeliski, 2022). Sobel filtre uygulama örneği Şekil 4.5'te verilmiştir.



Şekil 4.5. Sobel filtre uygulama örneği (Szelizki, 2022).

#### 4.5. Gabor

Gabor filtreleri, görüntü işleme ve özellik çıkarımında kullanılan özel filtrelerdir. Bu filtreler, farklı frekansta ve farklı yönde titreşen sinusoidal dalgaları kullanarak görüntülerden özellikler çıkarır. Gabor filtrelerinin matematiksel modeli şu şekildedir:

$$G(x, y; f, \theta, \sigma, \lambda, \gamma) = e^{-\frac{(x'^2 + \gamma^2 \cdot y'^2)}{(2 \cdot \sigma^2)}} \cdot \cos(2\pi \cdot f \cdot x' + \theta) \quad (4.9)$$

Burada,

$G(x, y; f, \theta, \sigma, \lambda, \gamma)$ : Gabor filtresinin  $(x, y)$  konumunda değerini temsil eder.

$f$ : Gabor filtresinin frekansını veya dalga boyunu belirtir.



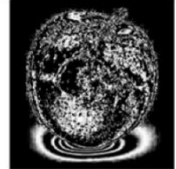



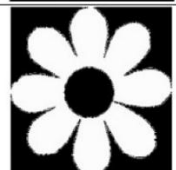
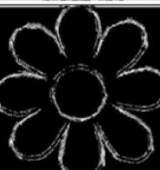
$\theta$ : Gabor filtresinin yönü veya açısını belirtir.

$\sigma$ : Gabor filtresinin standart sapmasını belirtir.

$\lambda$ : Gabor filtresinin dalga boyunu belirtir.

$\gamma$ : Gabor filtresinin enine keskinliğini belirtir.

' $x$ ' ve ' $y$ ':  $x$  ve  $y$  koordinatlarını, Gabor filtresinin dönüşüne uygun şekilde ölçeklendirilmiş koordinatlar olarak hesaplar (Daugman, 1985). Çeşitli giriş görüntüleri için Gabor filtreleme ve kenar algılama çıktıları Şekil 4.6'da verilmiştir.

Sr.No	Image name	Input image	Gray image	Gabor filtered Image	Gabor Filter Edge Detected image
1	Apple (256x256)				
2	Flower (256x256)				

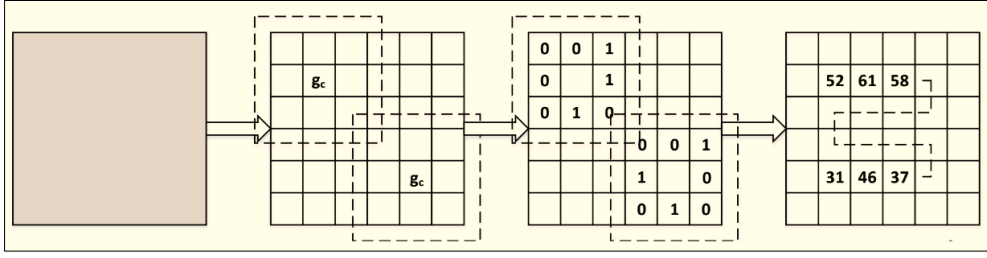
Şekil 4.6. Çeşitli giriş görüntüleri için Gabor filtreleme ve kenar algılama çıktıları (Dakshayani ve diğerleri, 2022)

#### 4.6. Çok Bloklü Yerel İkili Örüntü (Multi Block LBP / MB-LBP)

Çok Bloklü Yerel İkili Örüntü MB-LBP, geleneksel Local Binary Pattern (LBP) yönteminin bir genişlemesidir. MB-LBP, bir görüntüyü belirli boyutlardaki bloklara böler. Her blok içinde, LBP algoritması kullanılarak piksellerin etrafındaki komşu piksellerle karşılaştırılması yapılır. MB-LBP'nin matematiksel modeli şu şekildedir:

$$MB-LBP_{N,M}^P(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) * 2^p \quad (4.10)$$

Bu ifade MB-LBP'nin bir piksel için değerini hesaplamak için kullanılan formülü temsil eder. Burada,  $N$  ve  $M$ , bloğun boyutlarını,  $P$  komşu piksellerin sayısını,  $(x_c, y_c)$  merkezi pikselin koordinatlarını,  $g_p$  komşu piksellerin yoğunluğunu,  $g_c$  ise merkezi pikselin yoğunluğunu ifade eder.  $s(g_p - g_c)$ , karşılaştırma sonucunda elde edilen deseni ve farklı ağırlıkları içerir (Liu ve diğerleri, 2019). MB-LBP ile görüntünün değerini elde etme örneği Şekil 4.7'de verilmiştir.



Şekil 4.7. MB-LBP ile görüntünün değerini elde etme örneği (Liu ve diğerleri, 2019)

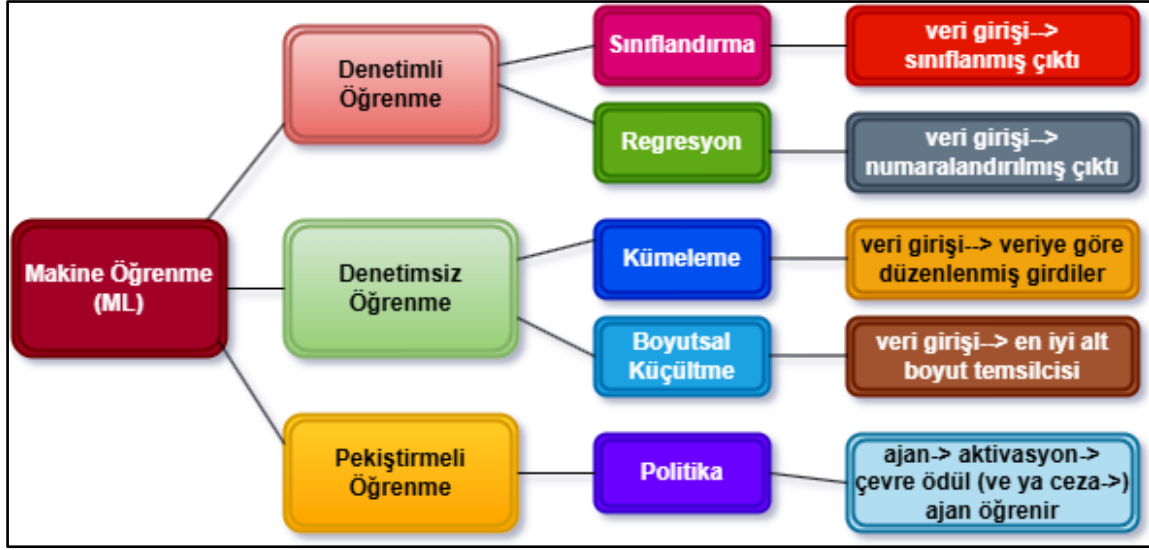


## 5. MAKİNE ÖĞRENMESİ

Makine öğrenimi, yeni bilgi ve beceriler edinmek, mevcut bilgiyi belirlemek, performans ve başarıyı sürekli iyileştirmek amacıyla insan öğrenme faaliyetlerini ve hesaplamalı kişisel gelişim yöntemlerini simüle etmek için bilgisayarların kullanılmasıdır ve bunun nasıl yapılacağına araştırılmasıyla ilgilidir (Wang ve diğerleri, 2009). Varlığından bu yana insanlar görevleri daha kolay gerçekleştirmek için farklı türde araçlar kullanıyorlar. İnsan beyninin yaratıcı gücü çeşitli icatlara yol açmıştır. İnsan hayatını kolaylaştıran bu makineler, insanların seyahat, sanayi, bilgisayar gibi çeşitli yaşam ihtiyaçlarını karşılamasını sağlıyor. Bunlardan biri makine öğrenimi yaklaşımıdır (Mahesh, 2020).

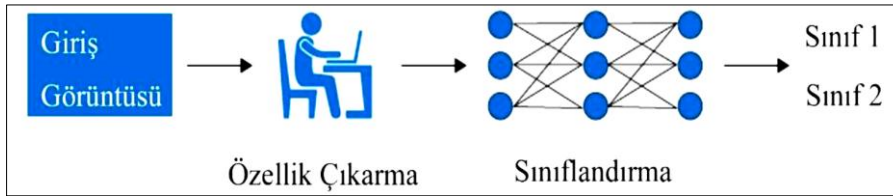
Makine öğrenimi (ML), verilere veya veri kümelerine dayalı olarak öğrenmek ve tahminler yapmak için matematiksel modeller ve teknikler kullanan bir yapay zeka türüdür. Makine öğreniminin temel amacı, bilgisayarların verilere dayalı otomatik öğrenme yoluyla akıllı, insan benzeri kararlar vermesini sağlamaktır. Bir ML algoritmasının karar vermeden önce verilerden öğrenmesi gerekir. Öğrenme, uygun veriler kullanılarak veya uygun veri yapıları oluşturularak yapılır. Verilerin özellikleri, ML algoritmasına girilmeden önce hazırlanmalıdır. ML algoritmaları ham verileri doğrudan kullanamaz. Özelliklerin seçilmesi verileri anlamlı hale getirir ve ML algoritmalarında kullanılır (Rende ve diğerleri, 2016).

ML, makinenin insan gibi düşünmesini ve insan müdahalesi olmadan kendi başına karar vermesini sağlayan bir yapay zeka dalıdır (Shaheamlung ve diğerleri, 2020; Zeebaree ve diğerleri, 2021). Makineleri doğrudan programlanmadan otomatik olarak öğrenecek hale getirme işlemidir. Makine öğreniminin temel amacı, verilere erişebilen ve onu öğrenme süreçleri için kullanabilen bir bilgisayar programı oluşturmaktır. Makine öğrenmesi algoritmaları hiyerarşik yapısı Şekil 5.1’de verilmiştir.



Şekil 5.1. Makine öğrenmesi algoritmaları hiyerarşik yapısı

Endüstriyel teknolojiler özelinde değerlendirildiğinde, makine öğrenimi yöntemleri tahmin yöntemlerine dayalı analitik ya da tahmine dayalı modelleme olarak kullanıldığı belirtilmektedir. Son dönemlerde makine öğrenimi yaklaşımları görüntü sınıflandırma problemlerinde yaygın olarak kullanılmaya başlanmıştır. Klasik makine öğrenimi yöntemleriyle vasıtasıyla görüntüleri sınıflandırmada kullanılan genel yöntem Şekil 5.2’de verilmiştir (Kılıç, 2021).



Şekil 5.2. Klasik makine öğrenimi yöntemleriyle görüntü sınıflandırma genel yöntem

Giriş veri setinde mevcut olan görüntüleri sınıflandırabilme yapabilmek için görüntüden çıkarılan ayırıcı ve tanımlayıcı özelliklerin belirlenmesi gereklidir. Tespit edilen özellikler seçilecek olan sınıflandırma algoritmasına giriş verisi olarak verilir ve model eğitimi yapılmaya başlanılır. Sonraki süreçte test verileri üzerinde tahminde algoritma tahminlerde bulunulur. Klasik makine öğrenimi teknikleri ile sınıflandırma görevini yapabilmek için kullanılan bazı kavramlar şu şekildedir:

**Model:** Makine öğrenme algoritmaları uygulanarak verilerden öğrenilen özel bir sunumdur. Bir modele hipotez de diyebiliriz. Günlük hayattaki bir durumun temsili şeklinde de tanımlanabilir.

**Özellik:** Veri setine ait ölçülebilir, tanımlanabilir bir özelliği veya parametresidir. Herhangi bir gözlemi temsil eden (genelde sayısal) verilerdir. Örneğin; bir arabanın fiyatını tahmin etmek için (cinsi, modeli, kilometresi vb.)

**Özellik Vektörü:** Birden çok karmaşık özelliğe sahip olan veri kümedir. Makine öğrenimi modelinde makinelere tanıtmak için girdi olarak kullanılır. Gerçek dünyada ele alınamayacak kadar kompleks halde olabilen verilerin makinelerin anlayabileceği bir şekilde basitleştirerek bir hale getirilmesi durumudur.

**Eğitim:** Bir algoritma, girdi olarak "eğitim verileri" olarak bilinen bir dizi veriyi tanımlar ve anlar. Öğrenme yapan algoritma, giriş verilerindeki modelleri bulur ve modeli beklenen sonuçlar (hedef) için eğitmeye çalışır. Eğitim süreci de oluşan çıktı makine öğrenimi modeli diye tanımlanmaktadır.

**Tahmin:** Eğitim verileriyle tamamen öğrenme işlemini bitiren modelin yeni verilerin özelliklerine göre sınıflarını tayin etme, saptama, belirleme veya çıkarımda bulunması durumudur.

**Hedef (Etiket):** Makine öğrenimi modellerinin tahmin etmesi beklenen değere hedef veya etiket denir. Her verinin bir kategoriye ait olduğunu temsil eder.

**Aşırı uyum:** Çok fazla miktarda makine öğrenimi modelini eğittiğinde, gürültüden ve hatalı veri girişlerinden öğrenme eğilimine geçebilmektedir. Burada model, verileri doğru bir şekilde karakterize edemez ve aşırı uyum oluşur.

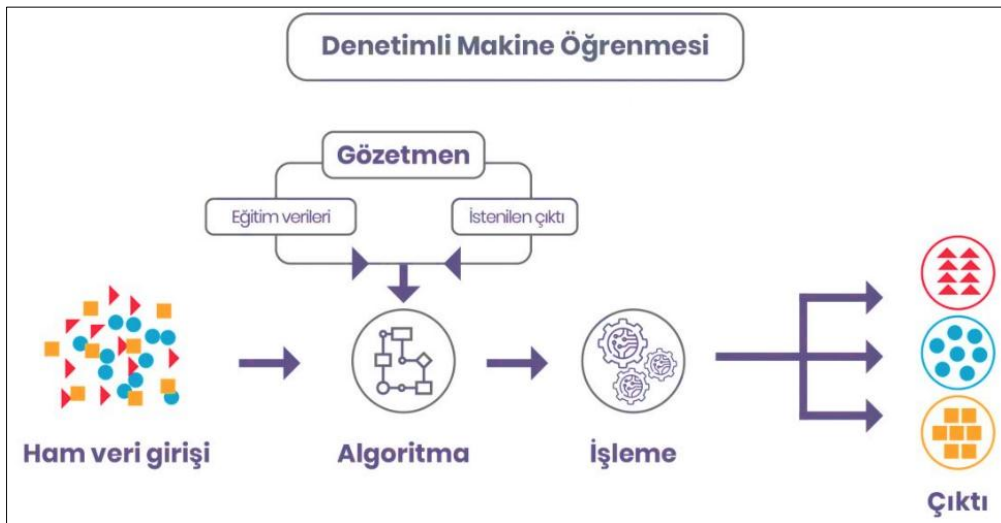
**Yetersiz uyum:** Bir makine öğrenme algoritmasının veya bir modelin verilerin temelindeki eğilimi yakalayamaması durumunda ortaya çıkan durumdur Makine öğrenimi modelinin doğruluğunu yok durumu ifade eder. Başka bir ifadeyle, model veya algoritma verilere yeterince uyum sağlayamamaktadır (Kılıç, 2021).

Makine öğreniminin iki temel esası bulunmaktadır; birinci esas tasarlanan modellere göre verileri ya da girdileri sınıflandırmak, ikinci esas ise bu modellere dayalı olarak gelecekteki gelen verilere göre sonuçlar için tahminler yapma şeklindedir. Makine öğrenimi yaklaşımları, öğrenme sisteminde mevcut olan "sinyalin" veya "geri bildirim" doğasına göre geleneksel olarak üç farklı ve geniş kategoriye ayrılmaktadır (Sahla, 2018).

### 5.1. Denetimli Öğrenme

Denetimli öğrenme, belirli örnek girdi-çıkı çiftlerine dayanarak girdileri çıktılarla eşleştiren bir işlevi öğrenen bir makine öğrenimi yöntemidir. Bu süreç, etiketlenmiş eğitim verilerinden özelliklerin çıkarılmasını içerir. Denetimli makine öğrenimi algoritmaları, dışarıdan yardım gerektiren algoritmalardır. Giriş veri seti genellikle eğitim veri seti ve test veri seti olarak ikiye ayrılır. Eğitim veri seti, tahmin edilmesi veya sınıflandırılması gereken çıktı değişkenlerini içerir. Tüm algoritmalar, bir eğitim veri setinden belirli kalıpları öğrenir ve bu bilgiyi tahmin veya sınıflandırma yapmak için bir test veri setine uygular (Mahesh, 2020).

Denetimli öğrenme modelinde en kritik nokta etiketli bir veri kümesi (labeled dataset) kullanılmasıdır. Denetimli makine öğreniminde temel amaç, bağımsız değişken üzerinden tanımlanan bir fonksiyonu kullanarak hedef değişkeni tahmin etme sürecidir. Denetimli makine öğrenmesi şematik gösterimi Şekil 5.3'te verilmiştir.



Şekil 5.3. Denetimli makine öğrenmesi şematik gösterimi (Mahesh, 2020).

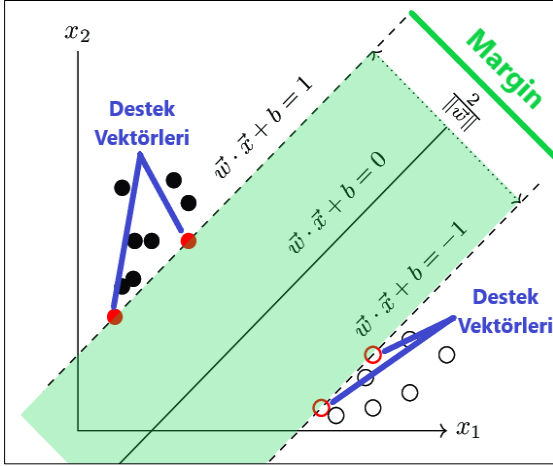
Denetimli öğrenme algoritması çeşitlerinden aktif öğrenme, regresyon ve sınıflandırma bulunmaktadır. Sınıflandırma ve Regresyon algoritmalarının her ikisi de etiketli veri kümeleriyle çalışmaktadır. Ancak sınıflama ve regresyon arasındaki değişkenlik, farklı makine öğrenimi problemleri için ne amaçla kullanımlarına yöneliktir.

### 5.1.1. Sınıflandırma

Veri kümesini değişik parametrelere göre sınıflama yapabilmesine yardımcı olan bir işlev bulma süreci şeklinde tanımlanabilir. Hedef ayrık ise (örneğin nominal veya sıralı), verilen göreve sınıflandırma denir. Sınıflandırmada, bilgisayar sistemi eğitim veri setine göre eğitilir ve bu eğitime dayanarak verileri farklı sınıflara ayırması sağlanır (Ławrynowicz ve Tresp, 2014). Sınıflandırma algoritmasının asıl amacı, girişi ( $x$ ) ayrık çıkışa ( $y$ ) eşlemek için eşleme fonksiyonunu bulmaya çalışmaktır. Örneğin, kusurlu ahşap görüntüleri ve kusursuz ahşap görüntüleri girdileri ile eğitilen model yeni girdilerle ahşabın kusurlu ya da kusursuz olması durumunu tahmin etmektedir. Yaygın kullanılan bazı sınıflandırma algoritmaları bu tez kapsamında 12 farklı sınıflandırması olarak verilmiştir.

#### Destek vektör makinesi (Support vector machine)

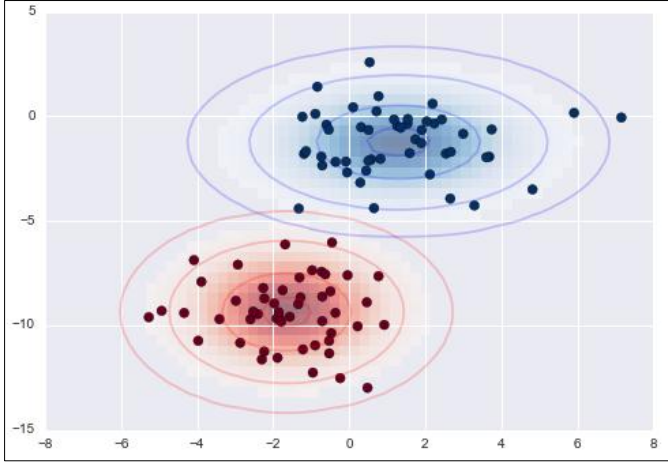
SVM, ilk eğitim verilerini daha yüksek bir boyuta dönüştürür, doğrusal olmayan bir haritalama kullanır. Bu yeni boyutta, doğrusal optimal ayırıcı hiper düzlemi (yani, bir sınıfın demetlerini diğerinden ayıran bir karar sınırı) arar. Bu yüksek boyutlu uzayda, farklı sınıflara ait değişkenleri doğrusal olarak ayıran algoritma arama karar sınırı. Bu optimum ayırma sınırına hiper düzlem denir. SVM, eğitim noktaları olan destek vektörlerini kullanarak hiper düzlem konumuna ve kenar boşluklarına karar verir. Basit olması için, sınıfların doğrusal olarak ayrılabilirdiği iki sınıflı bir örnek düşünülebilir (Koç, 2019; Han ve diğerleri, 2012). Destek vektör makinesi şematik gösterimi Şekil 5.4'te verilmiştir.



Şekil 5.4. Destek vektör makinesi şematik gösterimi (Koç, 2019)

### Naive bayes

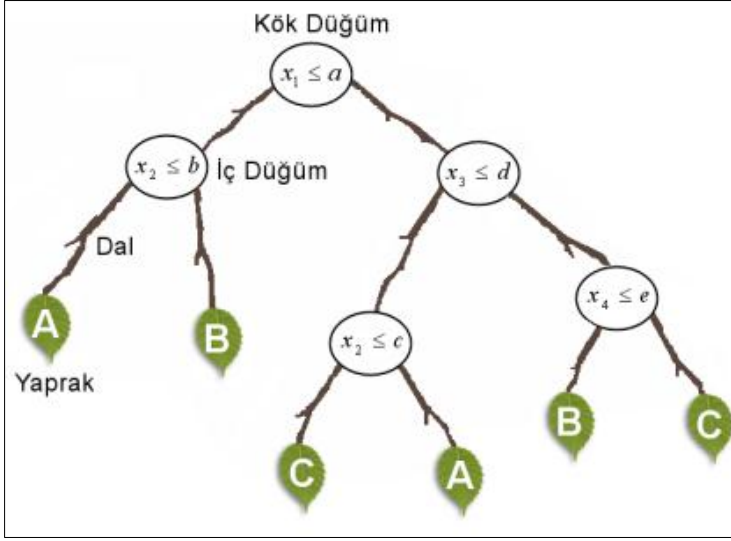
Naif Bayes sınıflandırıcısı, beyaz kutu olarak adlandırılan istatistiksel bir sınıflandırıcıdır; bu, makine öğrenimi sürecinin nasıl çalıştığının açık ve anlaşılır bir şekilde görülebmesini sağlar. Bayes sınıflandırma yöntemi, Thomas Bayes'in adını taşıyan ve Bayes Teorisi'ne dayanan bir yaklaşımdır. Bu yöntem, her örneğin posterior olasılığını, kendi sınıfları göz önüne alarak hesaplar. Naif Bayes sınıflandırıcısı, öznelik vektörü tarafından tanımlanan belirli bir örneğe en olası sınıfı atar. Bu algoritmanın, özelliklerin ilgili sınıf için bağımsız olduğunu varsayarak öğrenme sürecini basitleştirmesi, algoritmanın "naif" olarak adlandırılmasına neden olur. Bazı makine öğrenimi karşılaştırma çalışmaları, Naif Bayes sınıflandırıcısının, sinir ağı sınıflandırma algoritmalarıyla rekabetçi bir performans sergilediğini göstermiştir. Ayrıca, büyük veri setlerine uygulandığında Bayes sınıflandırıcıları, yüksek doğruluk ve iyi yürütme süresi verimliliği sunmuştur (Koç, 2019; Friedman ve diğerleri, 1997; Li ve Zhong, 2012; Schraudolph ve diğerleri, 2007). Naive Bayes şematik gösterimi örneği Şekil 5.5'te verilmiştir.



Şekil 5.5. Naive Bayes şematik gösterimi örneği (VanderPlus, 2016)

### Karar ağaçları

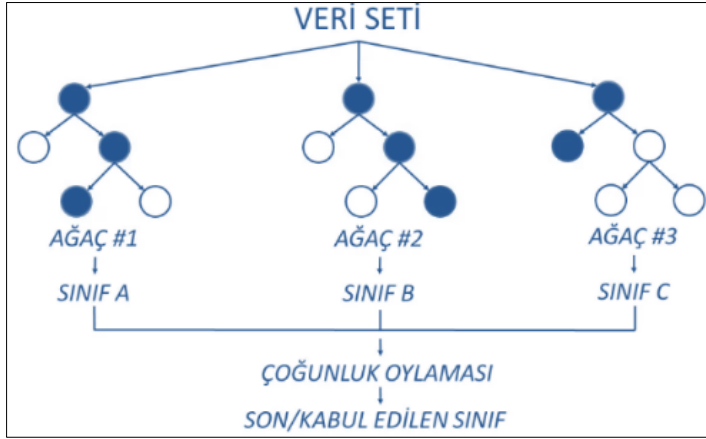
Karar Ağaçları (Decision Tree-DT), örnekleri özellik değerlerine göre sıralayarak sınıflandıran yapılar olarak tanımlanır. Karar ağacında her düğüm, sınıflandırılacak örnekteki bir özelliği temsil ederken, her dal, bu düğümün alabileceği bir değeri temsil eder. Sınıflandırma işlemi kök düğümden başlar ve özellik değerlerine göre sıralama yapılır (Kotsiantis ve diğerleri, 2007). Veri madenciliği ve makine öğreniminde, karar ağaçları bir öge hakkındaki gözlemleri, ögenin hedef değeri ile eşleştiren tahmine dayalı bir model olarak kullanılır. Bu tür ağaç modelleri, sınıflandırma ağaçları veya regresyon ağaçları olarak da adlandırılabilir (Hastie ve diğerleri, 2009). Karar ağaçları genellikle bir doğrulama kümesi kullanılarak budanır, bu nedenle karar ağaçlarının performansını değerlendiren budama sonrası teknikler uygulanır. Herhangi bir düğüm kaldırılabilir ve ilgili eğitim örneklerinin en yaygın sınıfına atanabilir (Kotsiantis ve diğerleri, 2007). Karar ağaçları şematik gösterimi Şekil 5.6'da verilmiştir.



Şekil 5.6. Karar ağaçları şematik gösterimi (Kavzoğlu ve Çölkesen, 2010).

### Rastgele orman

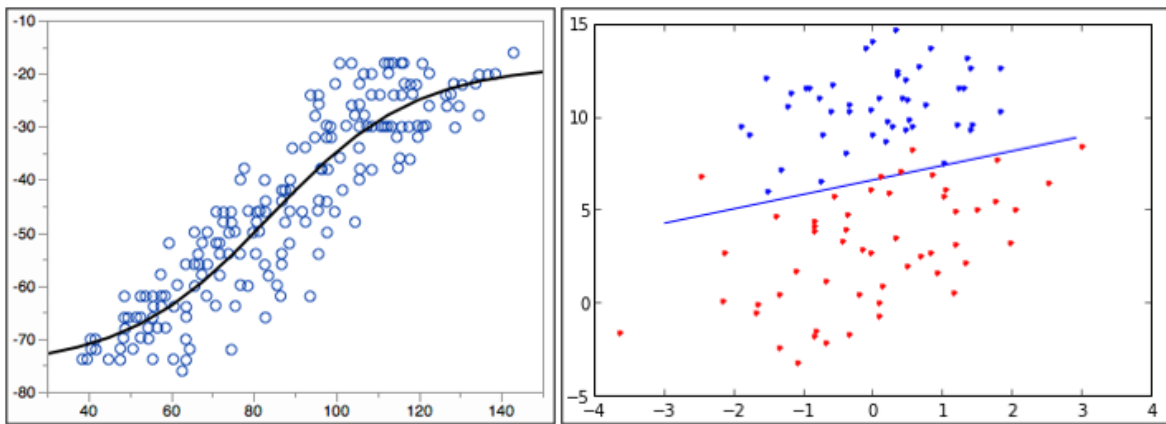
Rastgele Orman, bir dizi karar ağacını eğiterek ve sınıfa topluluktaki tüm ağaçların üzerinde çoğunluk sağlayarak işleyen bir topluluk yöntemidir (Lorena ve diğerleri, 2011). RF'ler, genellikle SVM'lerin biraz önünde, sınıflandırmadaki birçok sorunun galibidir. Hızlı, ölçeklenebilir, gürültüye karşı dayanıklıdır, aşırı uymazlar, yorumlanması ve görselleştirilmesi kolaydır, yönetilecek parametreler yoktur. Bununla birlikte, ağaç sayısı arttıkça, algoritma gerçek zamanlı tahmin için yavaşlar. Bölünmüş seçimde çeşitli öznelik değerlendirme önlemleri kullanarak, ağaçlar arasındaki korelasyonu azaltmak gibi RF'leri geliştirmek için çeşitli girişimlerde bulunulmuştur. Başka bir mekanizma, önce yeni örneğe en çok benzeyen örneklerde ağaçların ortalama marjını tahmin etmeyi ve daha sonra, ağaçları negatif marjla attıktan sonra, ağaçların oylarını marjla tartmayı önermektedir (Robnik-Šikonja, 2004). Rasgele orman algoritması örnek gösterimi Şekil 5.7'de verilmiştir.



Şekil 5.7. Rastgele orman algoritması örnek gösterimi

### Lojistik regresyon

Lojistik Regresyon (LR) sınıflandırıcıları, bir sınıfın meydana gelme olasılığını modelleyen, veri kümesine bir lojistik eğrinin yerleştirildiği istatistiksel modellerdir (Kleinbaum ve diğerleri, 2002). LR sınıflandırıcıları aynı zamanda lojistik model, logit model ve maksimum entropi sınıflandırıcıları olarak da bilinir. LR'deki ilk adım, sınıfın meydana gelme veya gelmeme olasılıklarının doğal günlüğünü içeren bir logit değişkeni oluşturmaktan oluşur. Daha sonra olasılıkları tahmin etmek için bir maksimum olabilirlik tahmin algoritması uygulanır. LR modelleri, İstatistik alanında büyük ölçüde kullanılır ve birçok gerçek dünya problemde başarı göstermiştir (Lorena ve diğerleri, 2011). Lojistik regresyon dağılım gösterimi örneği Şekil 5.8'de verilmiştir.

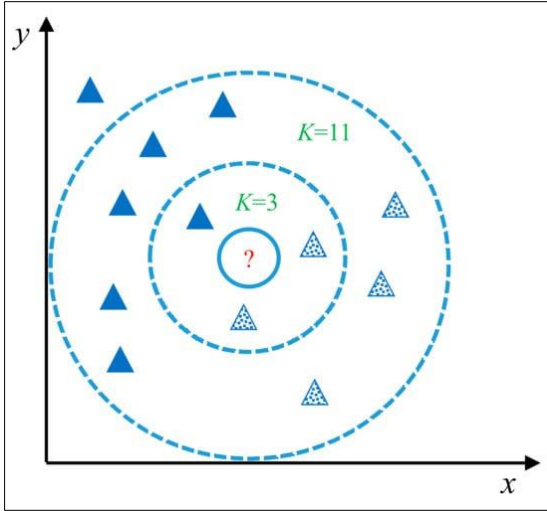


Şekil 5.8. Lojistik regresyon dağılım gösterim örneği

### K-En yakın komşu

KNN algoritması, örnek tabanlı ML tekniklerinin en basit temsilcisidir. Tüm eğitim verilerini depolar ve verilen veri kümesindeki  $k$  en yakın komşularının çoğunluğunun sınıfına göre yeni bir veri noktasını sınıflandırır. Her veri için en yakın komşuları elde etmek üzere KNN, veri ögesi çiftleri arasındaki mesafeyi hesaplamak üzere bir ölçü kullanır. Genel olarak, kullanılan ölçü Öklid mesafesidir (Lorena ve diğerleri, 2011).

KNN, nesnel işlevin yerel yaklaşımlarını, sınıflandırılan her yeni veri noktası için farklı şekilde oluşturabilir. Bu özellik, nesnel fonksiyon karmaşık olduğunda avantajlı olabilir, ancak düşük karmaşıklığa sahip birkaç yerel yaklaşımla tanımlanabilir. KNN'nin bir diğer avantajı da basitliğidir. Bununla birlikte, tüm eğitim verilerinin tekrar gözden geçirilmesi gerektiğinden, tahmin süreleri genellikle maliyetlidir (Lorena ve diğerleri, 2011). K-En yakın komşu şematik gösterimi Şekil 5.9'da verilmiştir.



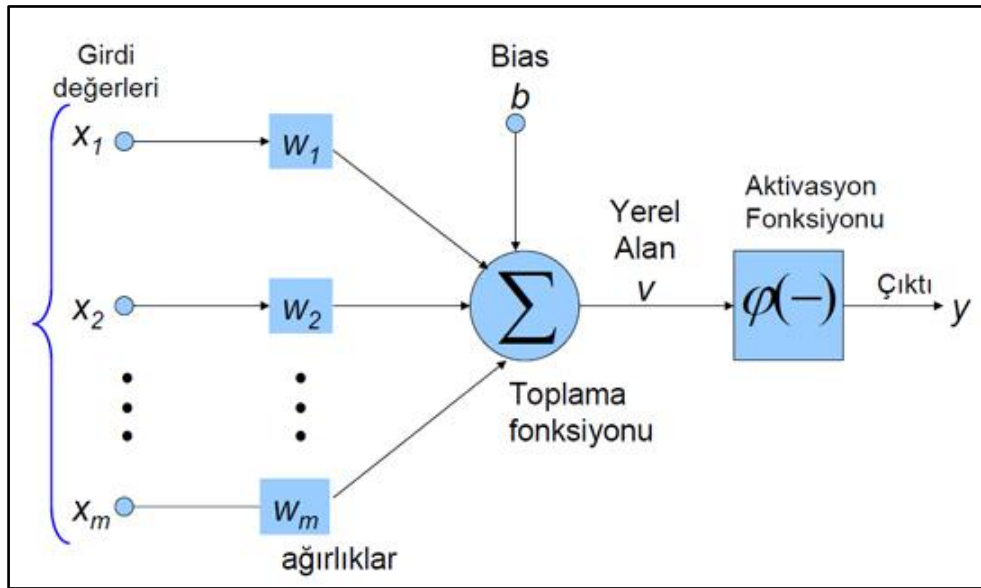
Şekil 5.9. K-En yakın komşu şematik gösterimi (Wang ve diğerleri, 2017)

### Yapay sinir ağları

YSA beynin yapısına, işleme yöntemine ve öğrenme yeteneğine dayanan hesaplamalı sistemlerdir (Haykin, 1994). Biyolojik nöronları simüle eden basit işlem birimlerinden oluşurlar. Düğümler olarak da adlandırılan bu yapay nöronlar, bir veya daha fazla katmana atılır. Her düğüm, biyolojik sinapsları simüle eden ağırlıklı bağlantılar aracılığıyla bir veya

daha fazla düğüme bağlanır. Çok Katmanlı Algılayıcı (MLP) YSA'ları şeklindedir (Lorena ve diğerleri, 2011).

Veriler hakkındaki temsil ve bilgi, bağlantıların ağırlıklarını ayarlayarak bir ANN'de edinilir ve saklanır. ANN eğitimi için çeşitli algoritmalar vardır. Genellikle ANN ağırlıklarını, ANN'nin çıkışlarını eğitim verileri için bilinen istenen çıkışlara yaklaştıracak şekilde ayarlamaya çalışırlar. Geri yayılma adı verilen algoritma, bu hata düzeltme kavramına dayanmaktadır. Genel olarak, ANN'lerin avantajları arasında gürültülü verilere karşı sağlamlıkları ve çeşitli form ve karmaşıklıkların doğrusal ve doğrusal olmayan işlevlerini temsil etme yetenekleri bulunmaktadır. Dezavantajları, parametre ayarlama ihtiyacını ve ağırlıklarda kodlanan ANN tarafından öğrenilen kavramları yorumlamadaki zorluğu içerir (Lorena ve diğerleri, 2011). Yapay sinir ağı şematik gösterimi Şekil 5.10'da verilmiştir.

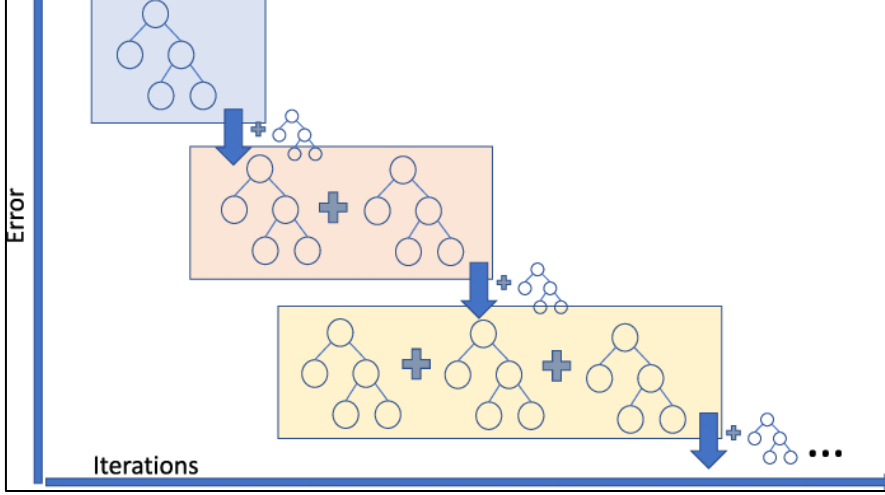


Şekil 5.10. Yapay sinir ağı şematik gösterimi

### Gradient boosting

Gradient Boosting, bir makine öğrenimi tekniği olarak başlayan bir dizi zayıf öğreniciyi bir araya getirerek güçlü bir tahmin modeli oluşturan bir topluluk öğrenme yöntemidir. Gradient Boosting algoritmaları genellikle sınıflandırma ve regresyon problemleri için kullanılır. Gradient Boosting algoritmalarının temel prensibi, bir hata fonksiyonunu minimize etmeye çalışırken her bir zayıf öğreniciyi birbirini takip eden şekilde eğitmektir.

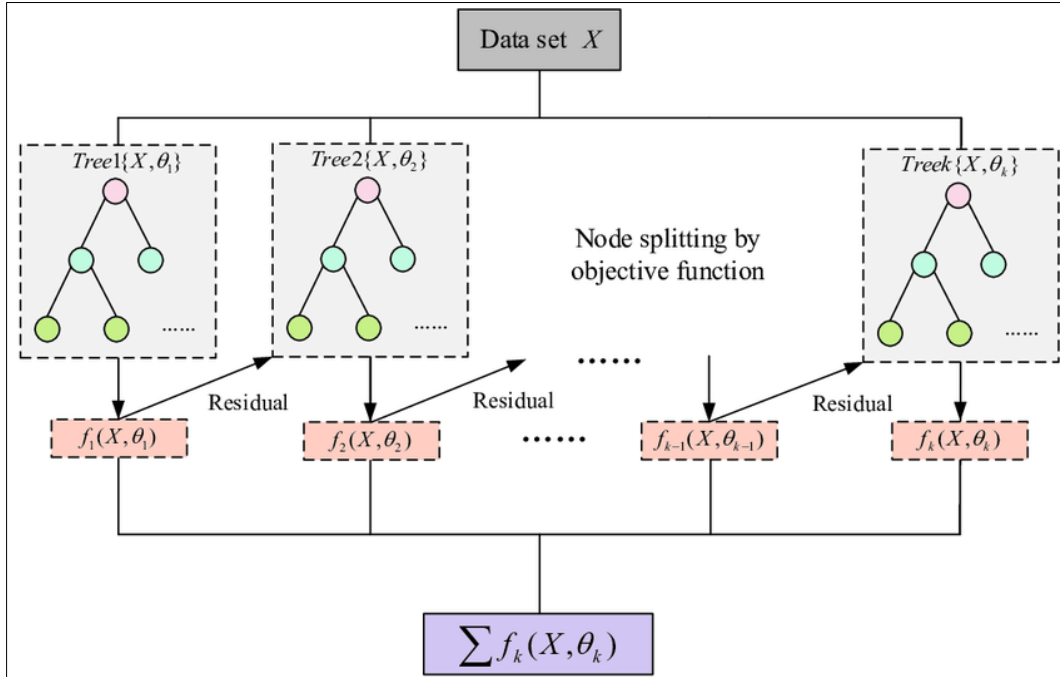
Bir önceki öğrenici tarafından yapılan hatalar, bir sonraki öğrenici tarafından düzeltilmeye çalışılır. Bu, modelin daha iyi tahminler yapabilmesine ve genel performansının artmasına yol açar (Freidman, 2001). Gradient boosting çalışma prensibi şematik gösterimi Şekil 5.11'de verilmiştir.



Şekil 5.11. Gradient boosting çalışma prensibi şematik gösterimi (Freidman, 2001)

### Extreme gradient boosting (XGB)

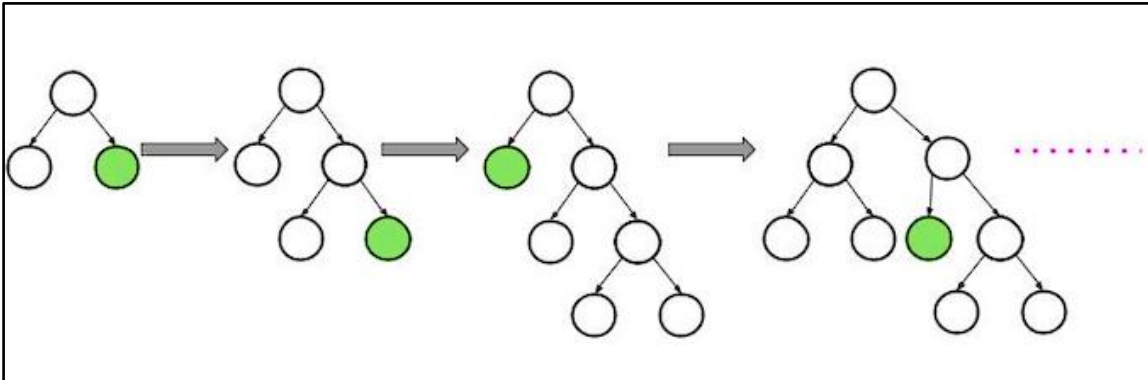
XGBoost, özellikle sınıflandırma ve regresyon problemleri için kullanılan bir makine öğrenimi algoritmasıdır. "Extreme Gradient Boosting" ifadesinin kısaltması olan XGBoost, yüksek performansı, hızı ve genel başarısıyla bilinir. Algoritma, Tianqi Chen tarafından başlatılmış ve daha sonra çeşitli geliştirmelere tabi tutularak popüler hale gelmiştir. XGBoost, gradient boosting framework'ünü temel alır ve karar ağaçları kullanarak güçlü tahmin modelleri oluşturur. Ancak, XGBoost, geleneksel gradient boosting yöntemlerinden daha birçok yenilik ve optimizasyon içerir. Özellikle şu özelliklere sahiptir: Regularization (Düzenleme): XGBoost, aşırı uydurmayı (overfitting) önlemek için L1 ve L2 düzenleme (regularization) tekniklerini içerir. Parallelization (Parallelleştirme): Algoritma, eğitim işlemlerini paralel olarak yürüterek hızlı eğitim süreleri sağlar. Cross-validation Entegrasyonu: XGBoost, model performansını değerlendirmek için cross-validation'ı doğrudan içerir. XGBoost, geniş bir kullanım alanına sahiptir ve sınıflandırma, regresyon, sıralama, tıbbi teşhis, finansal analiz ve birçok endüstri uygulamasında başarıyla kullanılmaktadır (Chen ve Guestrin, 2016). XGBoost'un genel yapısı Şekil 5.12'de verilmiştir.



Şekil 5.12. XGB genel yapısı (Guo ve diğerleri, 2020)

### LightGBM

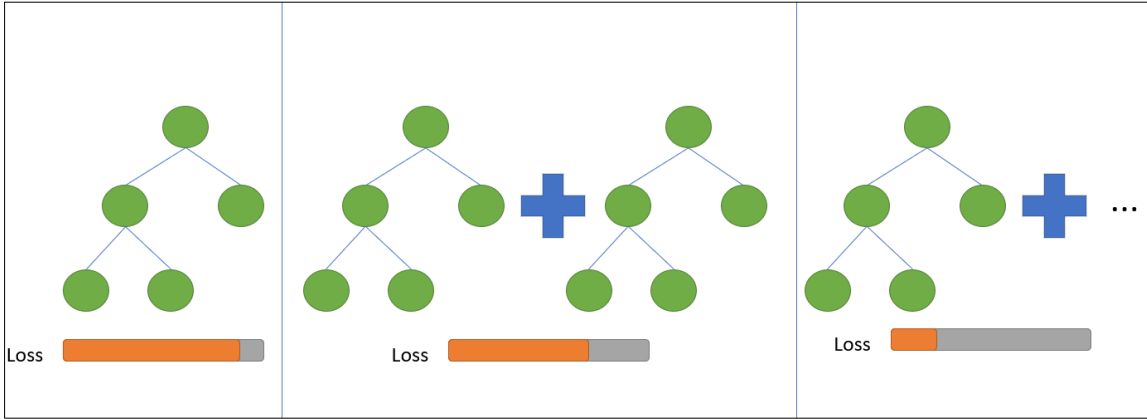
LightGBM, temel olarak bir karar ağacı tabanlı gradient boosting algoritmasıdır. Toplam tahmin, birden fazla karar ağacının toplamıdır. LightGBM, Microsoft tarafından geliştirilen ve açık kaynaklı bir makine öğrenimi kütüphanesidir. Bu kütüphane, özellikle büyük veri setleri üzerinde hızlı ve etkili bir şekilde çalışabilen bir gradyan artırma çerçevesidir (Ke ve diğerleri, 2017). LightGBM şematik gösterim örneği Şekil 5.13'te verilmiştir.



Şekil 5.13. LightGBM şematik gösterim örneği (Ke ve diğerleri, 2017)

### CatBoost

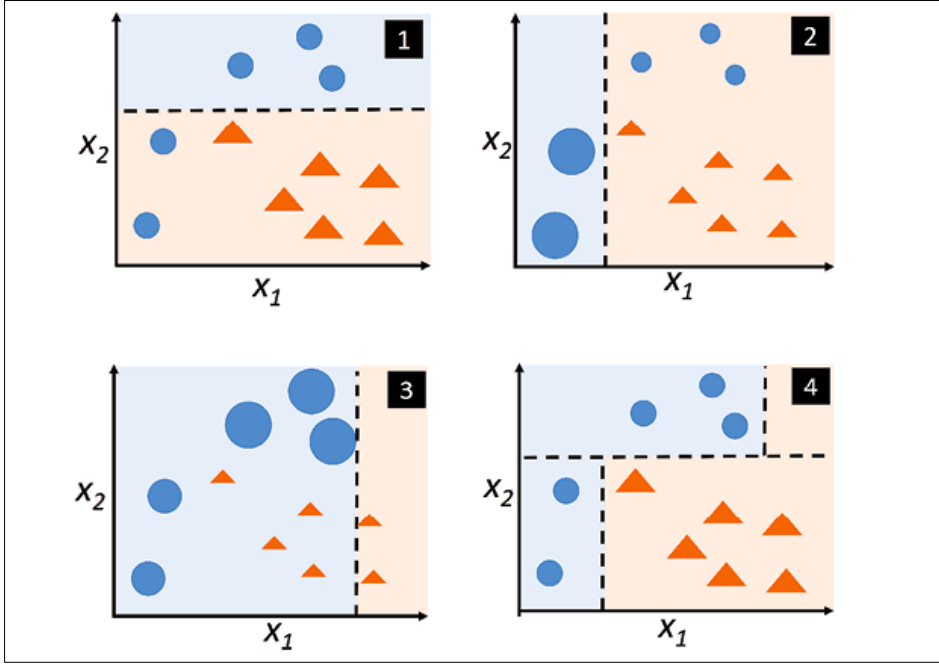
CatBoost, kategorik deęişkenleri etkili bir şekilde ele alabilen açık kaynaklı bir gradyan arttırma (gradient boosting) kütüphanesidir. Yandex tarafından geliştirilen bu kütüphane, özellikle kategorik deęişkenlerin işlenmesindeki kolaylık ve etkililięi ile öne çıkar. CatBoost, Python, R, Java ve dięer birçok programlama dilinde kullanılabilir (Prokhorenkova, 2018). CatBoost algoritması çalışma prensibi örneęi Şekil 5.14'te verilmiştir.



Şekil 5.14. CatBoost algoritması çalışma prensibi örneęi (Prokhorenkova, 2018)

### AdaBoost

Zayıf öęrenicileri birleştirek güçlü bir sınıflandırıcı oluşturan bir topluluk öęrenme teknięidir. AdaBoost, özellikle sınıflandırma problemlerinde kullanılır ve başlangıçta hatalı sınıflandırılan örnekler üzerine odaklanır (Fruend ve Schapire., 1997). AdaBoost algoritması şematik gösterimi Şekil 5.15'te verilmiştir.



Şekil 5.15. AdaBoost algoritması şematik gösterimi (Tang ve diğerleri, 2021)

### 5.1.2. Regresyon

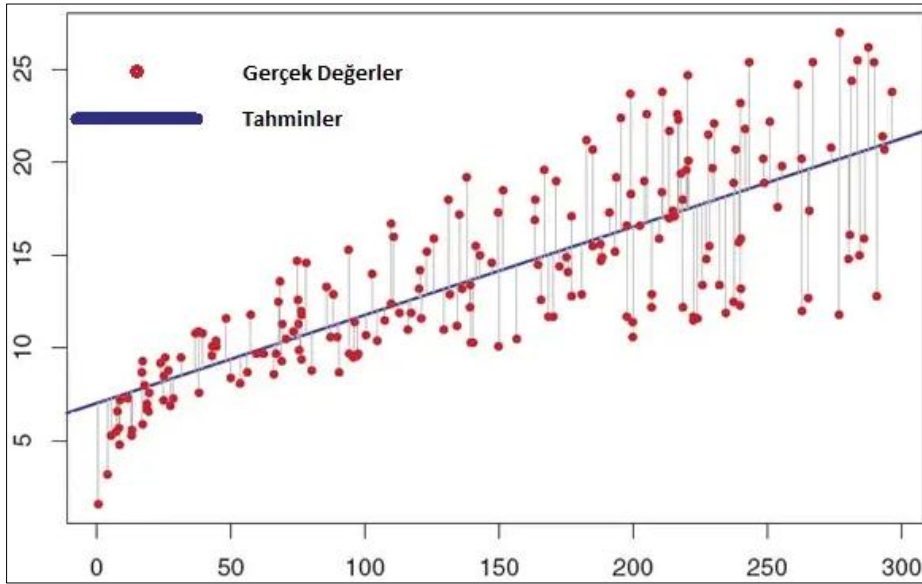
Regresyon, iki farklı amaç için kullanılan bir tekniktir. Birincisi, regresyon analizleri genellikle tahmin ve öngörü amaçları için kullanılır, bu uygulamaların makine öğrenimi ile büyük örtüşmeleri vardır. İkincisi, regresyon analizi bazı durumlarda bağımsız ve bağımlı değişkenler arasındaki nedensel ilişkileri belirlemek için de kullanılabilir. Ancak, regresyonlar tek başına yalnızca bağımlı bir değişken ile çeşitli değişkenlerden oluşan sabit bir veri kümesi arasındaki ilişkileri ortaya koyar (Wu ve diğerleri, 2019). Regresyon algoritmasının amacı, verilen bir girdi değişkenini ( $x$ ) sürekli bir çıktı değişkenine ( $y$ ) eşlemek için uygun bir eşleme fonksiyonu bulmaktır. Yaygın olarak kullanılan regresyon modelleri şunlardır:

- Doğrusal Regresyon
- Doğrusal olmayan regresyon
- Regresyon Ağaçları
- Polinom Regresyon
- Bayesçi doğrusal regresyon

### Doğrusal regresyon

Basit Doğrusal Regresyon, tek bir bağımsız değişkene sahip bir vaka modelidir (Abdulazeez ve diğerleri, 2020). Basit Doğrusal regresyon, değişkenin bağımlılığını tanımlar. Doğrusal regresyon, bağımsız değişkenlerin etkisini bağımlı değişkenlerin etkileşiminden ayırır (Acharya ve diğerleri, 2019).

Doğrusal Regresyon, denetimli öğrenmeye dayanan bir makine öğrenimi algoritmasıdır ve regresyon görevini üstlenir. Regresyon, bağımsız değişkenler temelinde bir hedef tahmin değerini modellemeyi amaçlar. Genellikle, değişkenler ile tahmin arasındaki ilişkiyi belirlemek için kullanılır. Doğrusal regresyon grafiği örneği Şekil 5.16'da verilmiştir.

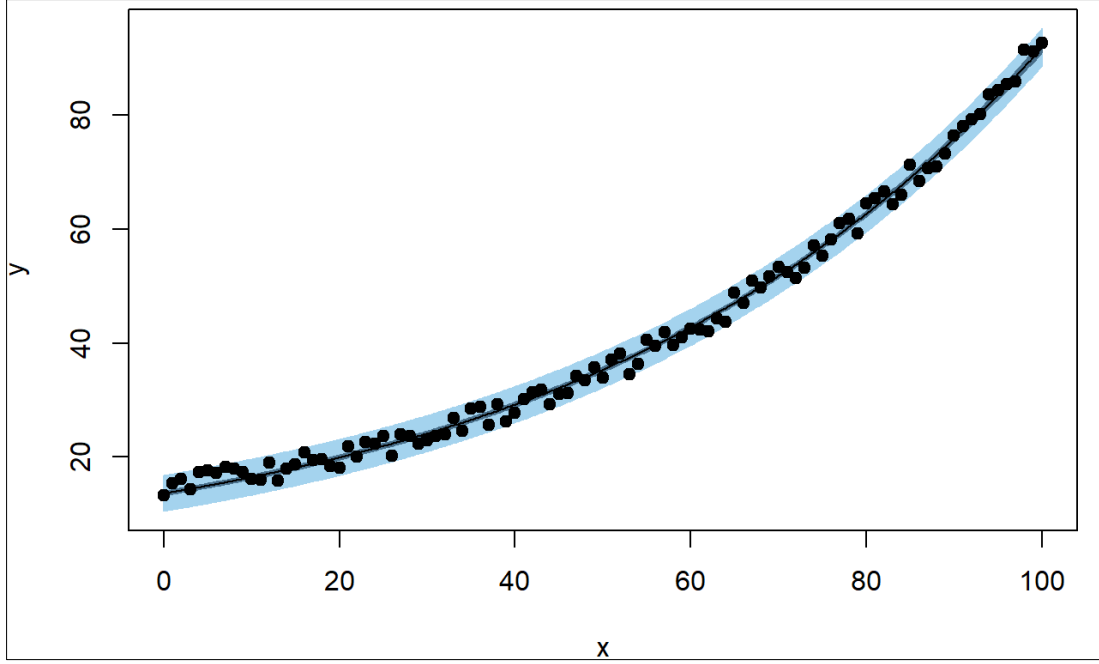


Şekil 5.16. Doğrusal regresyon grafik örneği (Acharya ve diğerleri, 2019)

### Doğrusal olmayan regresyon

Doğrusal olmayan regresyon, bağımlı değişken ile bir dizi bağımsız değişken arasındaki ilişkinin doğrusal olmayan bir modelini bulmak için kullanılan bir yöntemdir. Geleneksel doğrusal regresyon, tahminlerin doğrusal modellerle sınırlı olduğu durumların aksine, doğrusal olmayan regresyon, bağımsız ve bağımlı değişkenler arasında daha esnek ve karmaşık ilişkilerle tahmin yapılmasını sağlar. Doğrusal olmayan regresyon, tahmin denkleminin doğrusal olmayan bir şekilde bir veya daha fazla bilinmeyen parametreye bağlı olması ile karakterize edilir. Doğrusal regresyon genellikle tamamen ampirik bir

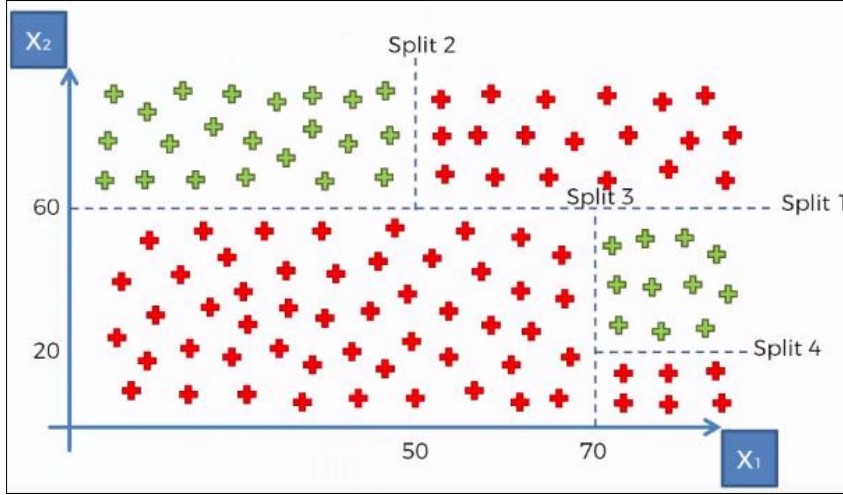
model oluşturmak için kullanılırken, doğrusal olmayan regresyon genellikle yanıt ve öngörücüler arasındaki ilişkinin belirli bir işlevi takip ettiğine inanmak için fiziksel nedenler olduğunda ortaya çıkarmaktadır (Smyth, 2002). Doğrusal olmayan regresyon modeli gösterimi Şekil 5.17’de verilmiştir.



Şekil 5.17. Doğrusal olmayan regresyon modeli gösterimi (Smyth, 2002).

### Regresyon ağaçları

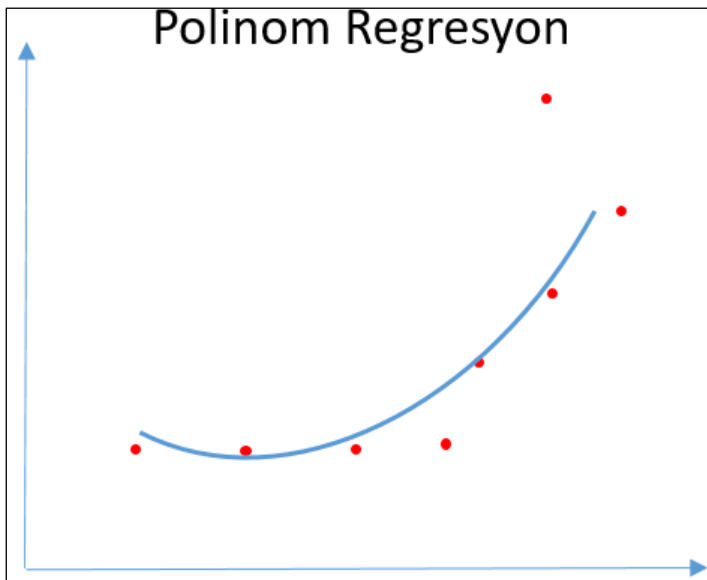
Köküni makine öğrenimi teorisine dayanan bir karar ağacı, sınıflandırma ve regresyon problemlerinin çözümü için etkili bir araçtır. Tek bir karar adımında sınıflandırma yapmak için bir dizi özelliği (veya bandı) birlikte kullanan diğer sınıflandırma yaklaşımlarından farklı olarak, karar ağacı çok aşamalı veya hiyerarşik bir karar şemasına veya ağaç benzeri bir yapıya dayanır. Ağaç bir kök düğümden (tüm verileri içerir), bir dizi dahili düğümden (bölünmeler) ve bir dizi terminal düğümden (yapraklar) oluşur. Karar ağacı yapısının her bir düğümü, bir sınıfı veya bazı sınıfları kalan sınıflardan ayıran ikili bir karar verir. İşleme genellikle ağaçta yaprak düğüme ulaşılan kadar aşağı doğru hareket ettirilerek gerçekleştirilir. Bu, yukarıdan aşağıya bir yaklaşım olarak bilinir (Xu ve diğerleri, 2005). Regresyon ağaçları tahmini mantığı şematik gösterimi Şekil 5.18’de verilmiştir.



Şekil 5.18. Regresyon ağaçları tahmin mantığı şeması gösterimi (Xu ve diğerleri, 2005)

### Polinom regresyon

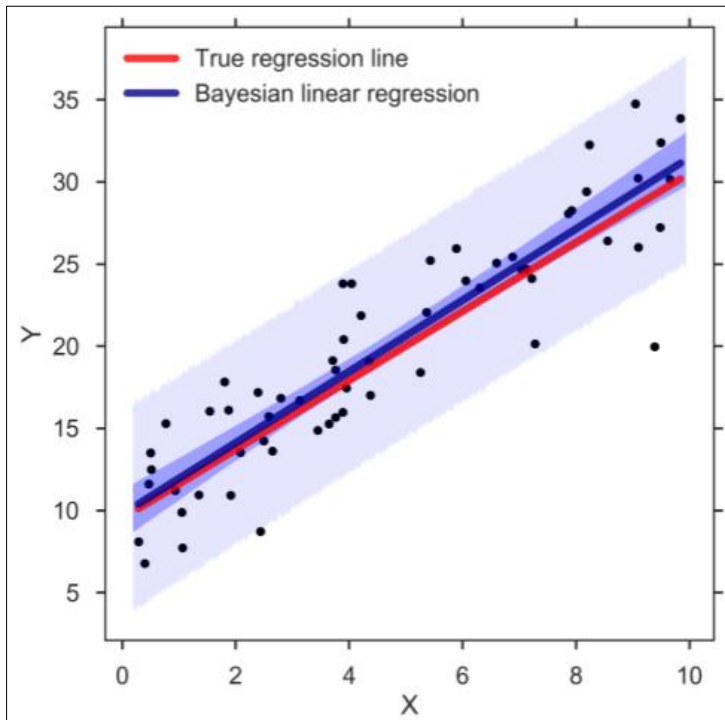
Polinom regresyonu, (x) girdi değişkenleri ile (y) çıkış değişkeni arasındaki ilişkinin bir polinom olarak modellendiği doğrusal bir regresyon şeklidir. Polinom regresyonu, verilere doğrusal olmayan bir modele uymasına rağmen, istatistiksel bir tahmin problemi olarak, verilerden tahmin edilen bilinmeyen parametrelerde regresyon fonksiyonunun doğrusal olması anlamında doğrusaldır. Bu nedenle, polinom regresyonu, doğrusal regresyonun özel bir durumu olarak kabul edilir (Peckov, 2012). Polinom regresyon dağılım örnek gösterimi Şekil 5.19'da verilmiştir.



Şekil 5.19. Polinom regresyon dağılım örnek gösterimi (Peckov, 2012)

## Bayesçi doğrusal regresyon

Bayes çıkarımı oldukça basittir. Modeldeki parametreler hakkında bir tahminle başlamaktadır. Örneğin, iki grup arasındaki fark veya  $x$  ile  $y$  arasındaki korelasyondur. Spesifik olarak, belirli parametre değerlerinin olasılığı hakkında tahminlerde bulunuruz. Örneğin, pozitif korelasyonlar negatif korelasyonlardan daha makul mü yoksa tüm korelasyonlar eşit derecede makul mü diye sorgulanır. Daha sonra parametrelerin olasılığı hakkındaki tahminleri güncellemek için veriler kullanılır. Daha basitçe söylemek gerekirse, Bayes analizi, parametreler hakkındaki tahminlerin ve verilerden parametreler hakkında öğrenilenlerin birleşimi olan modeldeki parametrelerin olasılığı hakkında bilgi üretmektedir (Kruschke, 2014; McElreath, 2020). Bayesçi doğrusal regresyon şematik gösterimi Şekil 5.20'de verilmiştir.

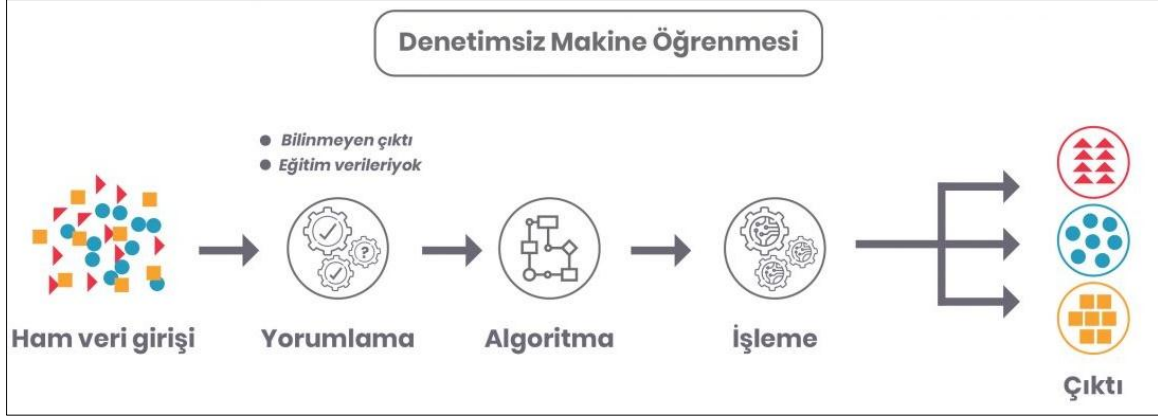


Şekil 5.20. Bayesçi doğrusal regresyon şematik gösterimi (Kruschke, 2014)

## **5.2. Denetimsiz Öğrenme**

Denetimsiz öğrenmeye bu adın verilmesinin nedeni, denetimli öğrenmenin aksine doğru cevapların ve bir öğreticinin bulunmamasıdır. Algoritmalar, verilerdeki ilginç yapıları keşfetmek ve sunmak için kendi başlarına çalışır. Denetimsiz öğrenme algoritmaları,

verilerden genellikle çok az özellik öğrenir ve yeni veriler sunulduğunda, verilerin sınıfını belirlemek için önceden öğrenilen özelliklere başvurur. Bu tür algoritmalar genellikle kümeleme ve özellik azaltma işlemleri için kullanılır (Mahesh, 2020). Denetimsiz makine öğrenmesi örnek şematik gösterimi Şekil 5.21’de verilmiştir.

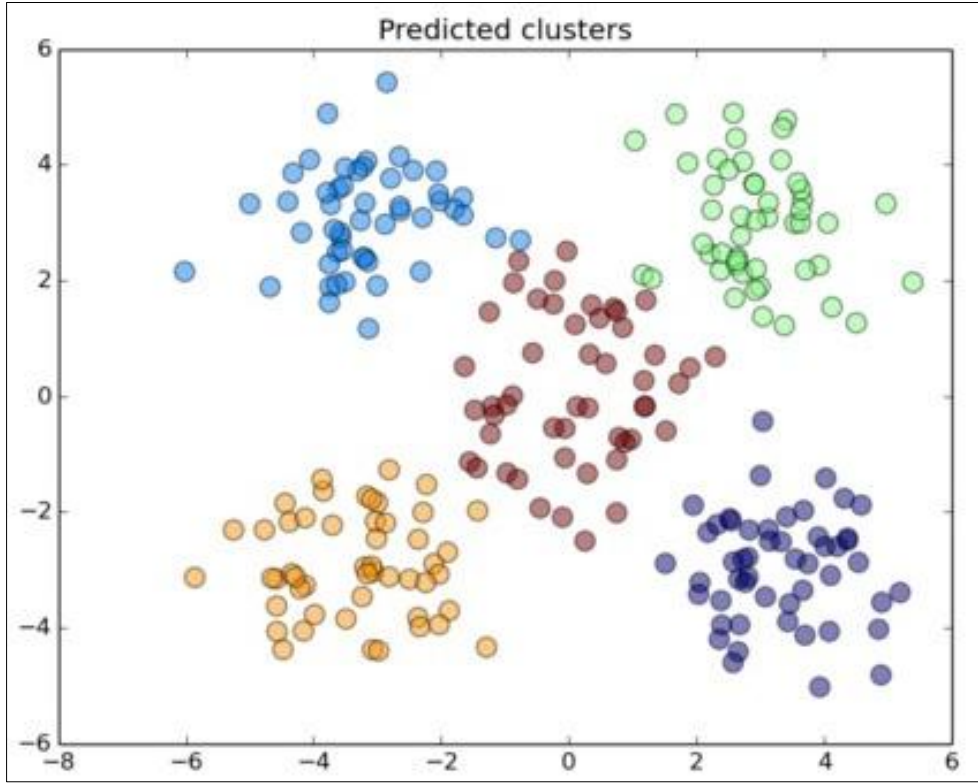


Şekil 5.21. Denetimsiz makine öğrenmesi örnek şematik gösterimi (Mahesh, 2020)

Denetimsiz öğrenme algoritmaları şu şekildedir; kümeleme, ilişkilendirme ve boyut azaltma algoritmaları bulunmaktadır (Ebert ve Louridas, 2016).

### 5.2.1. Kümeleme

Kümeleme, en fazla benzerliğe ve belirliğe sahip nesnelerin aynı grupta kalması ve başka bir grubun nesneleriyle nispeten daha az benzerlik göstermesi sonucu veya hiç olmaması için nesnelere farklı kümeler şeklinde gruplamadır (Kılıç, 2021). Kümeleme algoritmaları, etiketlerin kolayca kullanılmadığı sorun ayarlarında (kaynak ayırma sorunları gibi) yaygın olarak kullanılan denetimsiz ML algoritmalarıdır. Dahası, son zamanlarda, kümeleme için adalet konusu, Chierichetti ve arkadaşlarının 2017 yılındaki adil kümeleme üzerine ilk çalışmasının öncülük ettiği ML topluluğunda büyük ilgi görmüştür (Chierichetti ve diğerleri, 2017). Kümeleme algoritması örneği Şekil 5.22’de verilmiştir.



Şekil 5.22. Kümeleme algoritması örneği (Chierichetti ve diğerleri, 2017)

Çalışma yöntemlerine bağlı olarak, kümeleme de kendi içerisinde dört grupta incelenebilir:

**Özel Kümeleme:** Bir veri noktasının veya nesnenin yalnızca bir kümede bulunabileceğini ifade eder.

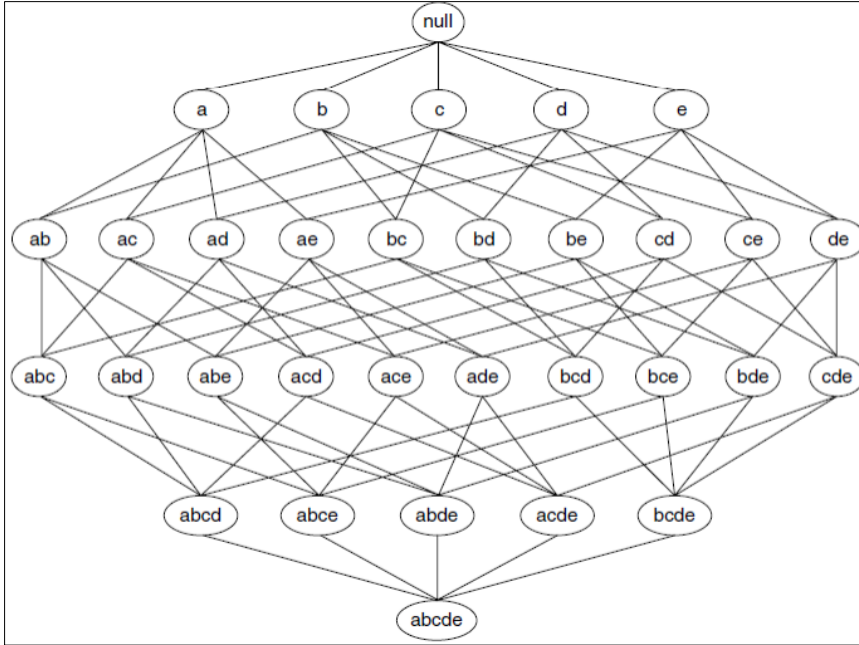
**Hiyerarşik Kümeleme:** İki farklı hiyerarşik yapı oluşturmaya yönelik bir yöntemdir. İki türü vardır: Aglomeratif ve bölücü. Aglomeratif yöntem, aşağıdan yukarıya bir yaklaşımı benimser. Başlangıçta her veri noktasını ayrı bir küme olarak değerlendirir ve daha sonra bu kümeler hiyerarşide yukarı doğru hareket ederken birleştirilir. Bölücü yöntem ise aglomeratif yöntemin tersidir. Her veri noktası tek bir kümede başlar ve hiyerarşide aşağıya indikçe bu küme bölünür (Chierichetti ve diğerleri, 2017).

**Örtüşen Kümeleme:** Bir veri noktasının iki veya daha fazla kümede yer almasına izin verir.

**Olasılıksal Kümeleme:** Olasılık kullanarak kümeler oluşturur ve olasılık dağılımlarını dikkate alır. Örneğin, yeşil çoraplar, mavi çoraplar, yeşil tişörtler ve mavi tişörtler "yeşil ve mavi" ya da "çorap ve tişört" gibi iki farklı kategoriye ayrılabilir (Chierichetti ve diğerleri, 2017).

### 5.2.2. İlişkilendirme

Birliktelik kuralı öğrenme, belirli öğelerin belirli bir veri kümesinde ne sıklıkta bir arada görüldüğünü analiz ederek bir öğenin diğerini nasıl etkilediğini bulmayı amaçlayan bir makine öğrenme tekniğidir. Bu, destek ve güven olmak üzere iki kriter kullanılarak yapılır. Destek, bu tür öğelerin sıklığını gösterirken, güven, tüm veri kümesindeki bu kuralların kaç kez doğru olduğunu gösterir. Birliktelik kuralına bir örnek şu şekildedir: “Ekmek ve tereyağı satın alan işlemlerin yüzde doksanı aynı zamanda süt de alır” (Agrawal ve diğerleri, 1993). Birliktelik kurallı öğrenme modeli örneği Şekil 5.23’te verilmiştir.

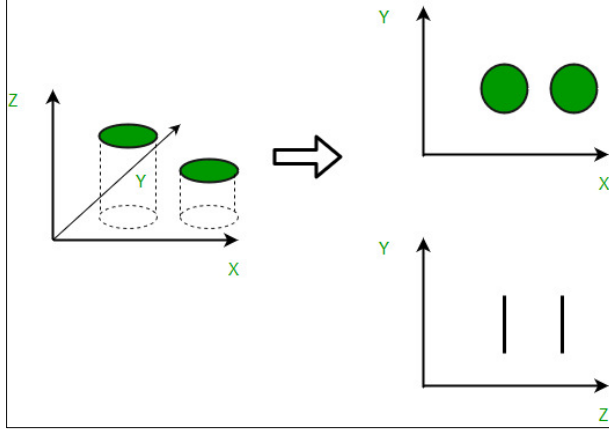


Şekil 5.23. Birliktelik kurallı öğrenme modeli örneği (Altameem ve Ykhlef, 2018)

### 5.2.3. Boyut Azaltma

Veri kümelerinin giderek büyümesiyle birlikte, tekil verilerin boyutu da artıyor; bu sorun bazen çok yüksek boyutluluk olarak anılır (Tan ve diğerleri, 2014). Boyut indirgeme teknikleri, yüksek boyutlu verileri daha düşük boyutlu bir uzaya eşlemektedir (Weinberger ve Saul, 2006). Bu tür teknikler, gerekli bir veri ön işleme yöntemi olarak makine öğreniminde, özellikle derin öğrenmede yaygın olarak kullanılmaktadır. Boyut indirgemenin temel amacı, yalnızca hesaplama karmaşıklığını azaltmakla kalmayan, aynı zamanda en önemlisi modeli büyük ölçüde uyarlayan en kullanışlı ve bilgilendirici alt

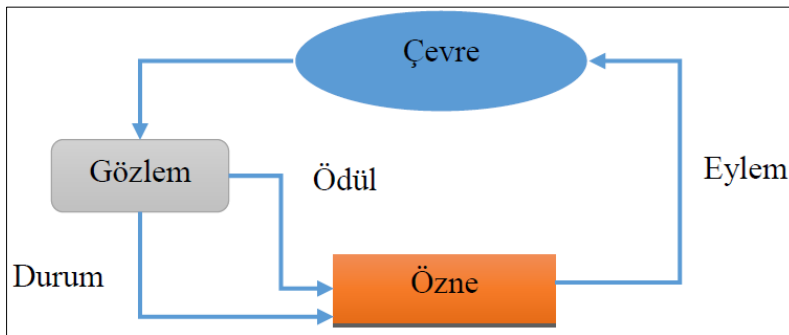
uzayı bulmaktır (Kohavi ve John, 1997). Boyutsal azaltma şematik gösterimi Şekil 5.24'te verilmiştir.



Şekil 5.24. Boyutsal azaltma şematik gösterimi (Kohavi ve John, 1997)

### 5.3. Takviyeli Öğrenme

Pekiştirmeli öğrenme, yazılım ajanlarının bir ortamda kümülatif ödülü en üst düzeye çıkarmak amacıyla nasıl hareket etmeleri gerektiğini öğrenmelerine odaklanan bir makine öğrenimi alanıdır. Denetimli öğrenme ve denetimsiz öğrenmenin yanı sıra, pekiştirmeli öğrenme, makine öğreniminin üç temel paradigmasından biridir (Mahesh, 2020). Takviyeli öğrenme sistemi şematik gösterimi Şekil 5.25'de verilmiştir.



Şekil 5.25. Takviyeli öğrenme sistemi şematik gösterimi (Kılıç, 2021)

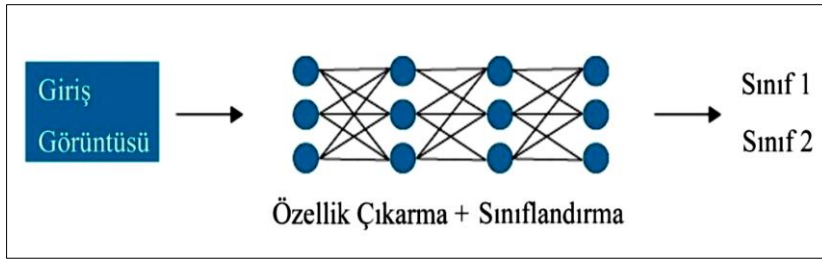
Bir pekiştirmeli öğrenme sisteminde, etmen ve çevre dışında opsiyonel bir unsur olmak üzere dört temel bileşen bulunmaktadır: politika, ödül, değer ve çevre modeli.



## 6. DERİN ÖĞRENME

Derin öğrenme, büyük veri kümeleri üzerinde karmaşık hesaplamalar yapabilen ve esas olarak yapay sinir ağlarını kullanan bir makine öğrenimi alt alanıdır (Schmidhuber ve diğerleri, 2015). Derin öğrenme modelleri, insan beyninin yapısını ve işleyişini taklit ederek geliştirilmiş ve ilerlemiştir. Temelde, yapay sinir ağlarının derinleştirilmesiyle ortaya çıkmış ve bu yüzden literatürde derin sinir ağları olarak da adlandırılmıştır. Temel yapay sinir ağı modellerinde genellikle 2 veya 3 gizli katman bulunurken, derin sinir ağları 100'den fazla gizli katmana sahip olabilir (Kılıç, 2021).

Derin öğrenme modelleri, klasik makine öğrenimi tekniklerinden farklı olarak verinin temsilinden öğrenme yeteneğine sahiptir. Geleneksel makine öğrenimi yöntemlerinde ses, metin veya görüntü gibi verilerden sınıflama yapabilmek için özellik çıkarma işlemi gereklidir. Bu durumda, eğitilen modelin performansı, çıkarılan özellik vektörlerinin kalitesine doğrudan bağlıdır ve özellik çıkarma süreci ayrı bir uzmanlık gerektirir. Ancak derin öğrenme modelleri, bu özellikleri doğrudan verinin temsilinden öğrenerek sınıflama işlemini gerçekleştirebilir (Lecun ve diğerleri, 2015). Derin ile görüntü sınıflandırma şematik gösterimi Şekil 6.1'de verilmiştir.



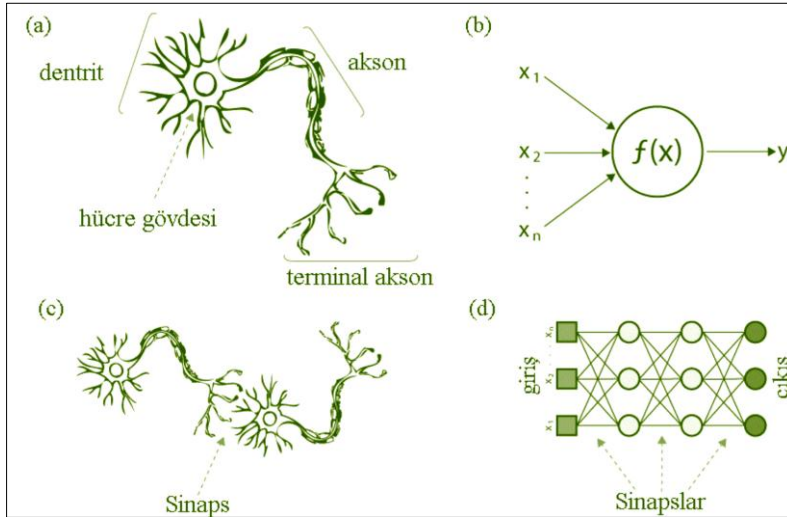
Şekil 6.1. Derin ile görüntü sınıflandırma şematik gösterimi (Kılıç, 2021)

Derin öğrenme algoritmalarının performansı, kullanılan veri miktarına büyük ölçüde bağlıdır. Daha geniş veri setleri, bu algoritmaların daha etkili bir şekilde eğitilmesine ve daha doğru tahminler yapabilmesine olanak tanır. Grafik işlemcilerinin gelişmesiyle birlikte, bu algoritmalar çok sayıda parametreyi hesaplayabilme kapasitesine sahip hale gelmiştir. Bu gelişme, derin öğrenme algoritmalarının çeşitli alanlarda daha yaygın bir şekilde kullanılmasına imkan tanımıştır (Pacal ve diğerleri, 2020).

## 6.1. Yapay Sinir Ağları

YSA, insan beyninin işleyişinden ilham alarak geliştirilen bir makine öğrenme algoritmasıdır ve derin öğrenmenin temelini oluşturur. İnsan beyni, aksonlar, sinapslar ve dendritlerden oluşan bir nöron ağı aracılığıyla hesaplama işlemlerini gerçekleştirir. Elektriksel sinyaller aracılığıyla iletişim kuran bu sinirsel kablolar, YSA'daki hesaplamaların benzerini oluşturur (Haykin, 1994).

1943 yılında McCulloch ve bir meslektaşı, bir nöronun başka bir nörondan aldığı girdiye göre toplam ağırlıklı girdiye dayanarak aktif veya pasif olmasını sağlayan bir anahtar olarak modellendi (McCulloch ve Pitts, 1943). Bu nöron ağının 1960'lı yıllarda insan beynine benzer şekilde çalıştığı gösterilmiştir (Minsky, 1961). Yapay Sinir Ağı (YSA) modeli, insan beyninin hesaplama yapısını temel alır; bir nörondan gelen girişler, ilgili sinapsların elektriksel etkileşimini simüle eden ağırlıklarla çarpılır. Ağdaki nöronlar, işlemci elemanlarına, dendritler toplama fonksiyonuna, hücre gövdesi transfer fonksiyonuna ve aksonlar ise yapay nöron çıkışına karşılık gelir (Dongare ve diğerleri 2012). Gerçek sinir ağı modeli Şekil 6.2'de gösterilmektedir.



Şekil 6.2. Gerçek sinir ağı modeli: (a) insan sinir hücresi, (b) yapay sinir hücresi, (c) biyolojik sinaps, (d) YSA sinapsları (Maltarollo ve diğerleri, 2013)

### 6.1.1. Yapay sinir ağı yapısı

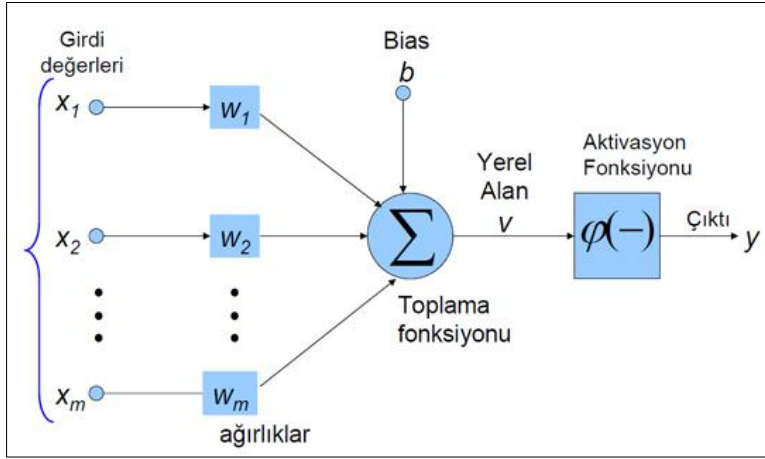
YSA'nın yapısı genellikle beş bölümden oluşur: girişler, ağırlıklar, birikim fonksiyonu, aktivasyon fonksiyonu ve çıkıştır. Girdiler, ağı öğrenmesini sağlayan veri kümesindeki bilgilerdir. Giriş adı verilen bu bilgi, her ne kadar ilk katmanın giriş katmanına doğrudan verilen ham bilgiyi ifade etse de bu bilgi daha sonraki katmanlardaki nöronlardan gelen veriler olarak tanımlanabilir. Bu girdiler toplayıcı nöronun çekirdeğine gönderilir (Kılıç, 2021).

Ağırlıklar, nörona gönderilen bilginin gönderilen bağlantı yolunun değeriyle çarpımını temsil eder. Ağırlıklar sayesinde giriş verilerinden elde edilen çıktının etki değeri ayarlanabilmektedir. Ağırlıklar ilk antrenman döneminde sabit olup, antrenmanın ilerleyen aşamalarında başlangıç durumuna bağlı olarak değişmektedir. Ağırlıkların nihai değeri öğrenilen ağı yapısını temsil eder ve bu açıdan öğrenme sürecinde oldukça önemlidir (Kılıç, 2021).

Toplama fonksiyonu, bir nörona gelen tüm değerleri önceki nöronların ağırlığıyla çarparak birleştirme işlemini gerçekleştirir. İşlem sonunda o hücrenin net gelirini hesaplar (Kılıç, 2021).

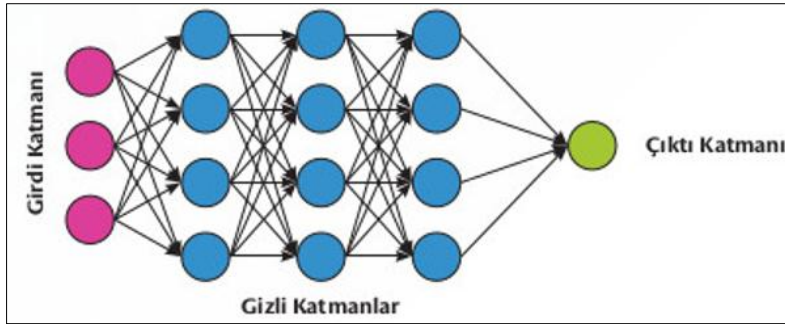
Aktivasyon fonksiyonu, nöronda hesaplanan net giriş değerini aktivasyon sürecinden geçirerek çıkış değerini üretir. Bu adımda aktivasyon fonksiyonu doğrusal olmayan bir fonksiyon olarak seçilir. Etkinleştirme işlevinin seçimi sorun giderme ve ağ performansı açısından önemlidir. YSA'lar geriye yayılım yoluyla öğrendiğinden, aktivasyon fonksiyonunun kolayca ayırt edilebilmesi gerekir. Sigmoid fonksiyonu nöronal aktivasyon için yaygın olarak kullanılır (Kılıç, 2021).

Çıktı, toplam net gelir miktarına göre aktivasyon fonksiyonu tarafından üretilen çıktı değerini temsil eder. Sinir ağlarındaki hücrelerin birçok girdisi olmasına rağmen ortaya çıkan çıktı yalnızca bir tanedir. Şekil 6.3'te YSA sinir hücresinin şematik gösterimi verilmiştir.



Şekil 6.3. YSA sinir hücresinin şematik gösterimi

Yapay sinir ağlarının katman yapısı 3 bölümden oluşur: giriş katmanı, gizli katman ve çıkış katmanı. Giriş katmanı, veri setlerinin ara katmanlara gönderilmesi işlemini gerçekleştirir. Gizli katmanlarda giriş katmanından alınan veriler üzerinde işlemler yapılır ve işlenen veriler çıkış katmanına gönderilir (Maind ve Wankar, 2014). Yapay sinir ağının bir modeli Şekil 6.4'te gösterilmektedir.



Şekil 6.4. Yapay sinir ağının bir modeli (Maind ve Wankar, 2014)

### 6.1.2. İleri beslemeli ağ modeli

İleri beslemeli sinir ağı modeli, giriş katmanından alınan bilgiyi herhangi bir değişiklik yapmadan doğrudan gizli katmana iletir. Gizli katmanlarda işlenen veriler, çıkış katmanına aktarılır ve bu sayede giriş değerlerine karşılık gelen çıkış değerleri üretilir. Bu ağ yapısında, işlem yalnızca girişten çıkışa doğru tek yönlü olarak gerçekleşir ve girilen değerlerle elde edilecek çıkış değeri belirlenir (Abiodun ve diğerleri, 2018).

### 6.1.3. Geri beslemeli ağ modeli

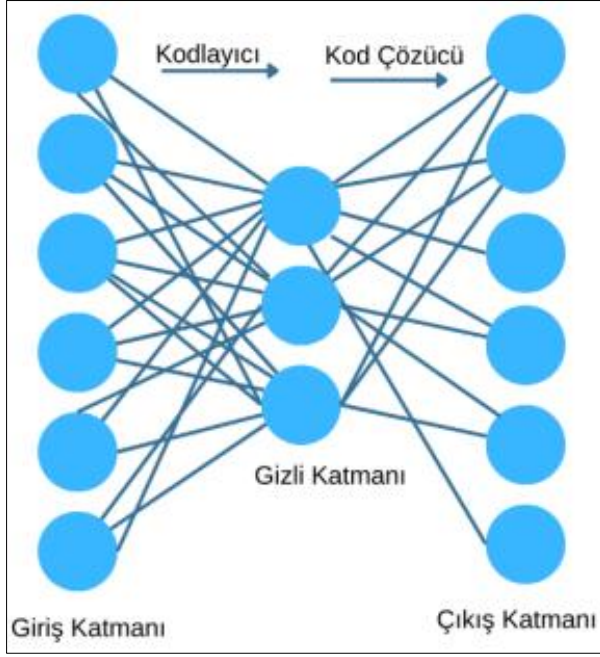
Geri bildirim süreci, Yapay Sinir Ağı (YSA) modelinin eğitim sırasında örnek verilerden öğrenmesi için kritik öneme sahiptir. Bu süreç, bir geri yayılım algoritması kullanılarak gerçekleştirilir. Geri yayılım algoritması, YSA'nın denetimli öğrenme yapabilmesini sağlar ve genellikle gradyan iniş olarak da adlandırılır. Bu süreçte, ağın işleme sonucunda elde edilen çıktı ile gerçek çıktı arasındaki fark hesaplanır; bu fark hata değeri olarak bilinir. Tüm gizli katmanlar ve çıktı ağırlıkları hata değerine göre değerlendirilir ve mevcut hata geriye yayılır. Geri yayılım sırasında modelin ağırlık katsayıları güncellenerek optimizasyon gerçekleştirilir. En iyi ağırlıkları bulmak ve ağırlık katsayılarını güncelleyerek minimum hata değerine ulaşmak ağdaki öğrenmeyi tanımlar (Dongare ve diğerleri, 2012).

## 6.2. Derin Öğrenme Modelleri

Derin öğrenme, görüntü analizi, metin madenciliği, doğal dil işleme ve boyut azaltma gibi çeşitli alanlarda geniş bir uygulama yelpazesine sahiptir. Büyük veri setleri üzerinde karmaşık hesaplamalar gerçekleştirmek ve verilerin temsili hakkında derinlemesine bilgi edinmek amacıyla bu alanlarda çeşitli modeller geliştirilmiştir. Bu bölümde, farklı veri türleri için kullanılan derin öğrenme modelleri ele alınacaktır.

### 6.2.1. Otomatik kodlayıcılar

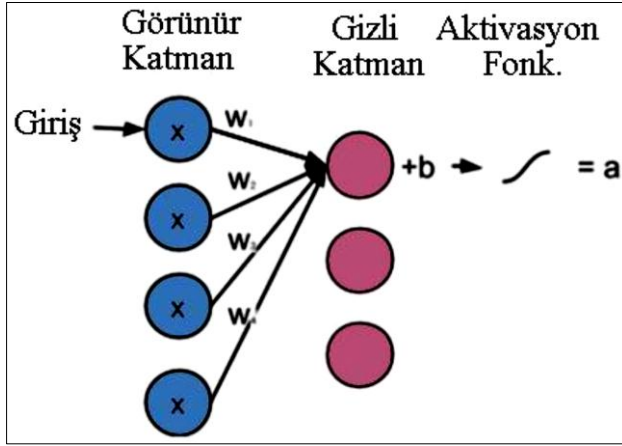
1980'li yıllarda, Geoffrey Hinton çalışmalarıyla geliştirilen denetimsiz öğrenme sorunlarına yönelik otomatik kodlayıcılar, aynı giriş ve çıkış verilerine sahip sinir öncesi ağ modelleridir. Genellikle görüntü analizi ve boyut azaltma problemlerinde kullanılırlar. Otomatik kodlayıcılar, bir veri kümesinden girdi almak, bunları farklı gösterimlere dönüştürmek ve ardından bunları doğru bir şekilde yeniden oluşturmak üzere tasarlanmıştır. Bu ağ yapısı üç bölümden oluşur: kodlayıcı, kod ve kod çözücü. Kodlayıcı, girişi daha küçük bir boyuta sıkıştırır ve bir kod üretir. Ancak kod çözücü, giriş değerini yalnızca kodu kullanarak yeniden üretir (Doğan ve Türkoğlu, 2019). Otomatik kodlayıcıların ağ yapısı Şekil 6.5'te gösterilmektedir.



Şekil 6.5. Otomatik kodlayıcı ağ yapısı (Doğan ve Türkoğlu, 2019)

### 6.2.2. Kısıtlı boltzman makineleri (Restricted boltzman machines - RBM)

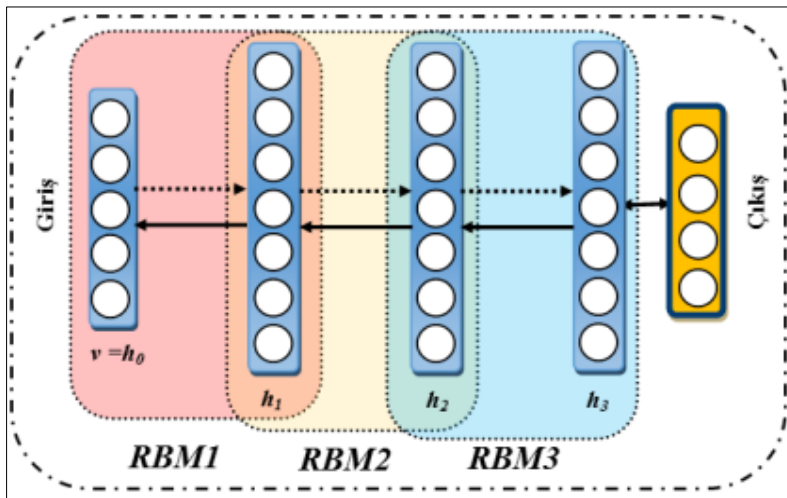
Hinton ve arkadaşları tarafından geliştirilen RBM (Restrictive Boltzmann Machine) modeli, girdilerin olasılık dağılımından öğrenebilen ve derin bir inanç ağının temelini oluşturan sinir ağlarıdır (Ackley ve diğerleri, 1985). En yaygın kullanımları boyut azaltma olsa da sınıflandırma ve özellik öğrenme için de kullanılabilirler. RBM ağları iki katmandan oluşur: görünür birimler ve gizli birimler. Görünür varlıklar tüm gizli katmanlara bağlıdır ancak aynı katmandaki düğümler bağlı değildir ve iletişim kuramaz. Bu iletişimin oluşturulamayacağını, "sınırlı" olduğunu gösterir; bu terim kullanılmaktadır. Bu ağ yapısının gizli katmanlara bağlı olan ve çıkış katmanı olmayan bir ofset değeri vardır. RBM modelinde iki tür işleme vardır: ileri ve geri. Daha sonraki işlemlerde giriş değerleri ağırlıklandırılıp öngerilimlendirilerek gizli katmana aktarılır. Yeniden yapılandırma sürecinde bu değerler yeniden yapılandırma aktivasyon fonksiyonundan geçirilir. Ortaya çıkan değerler öngerilim ve uygun ağırlıklarla birleştirilir ve yeniden yapılandırma için görünür katmana aktarılır. Görünür katmanda elde edilen sonucun kalitesi, çıktı değerinin orijinal girdiyle karşılaştırılması yoluyla ölçülür (Sutskever ve diğerleri, 2009). Kısıtlı Boltzman makine modeli Şekil 6.6'da verilmiştir.



Şekil 6.6. Kısıtlı Boltzman makine modeli (Abiodun ve diğerleri, 2018)

### 6.2.3. Derin inanç ağları (Deep belief network – DBN)

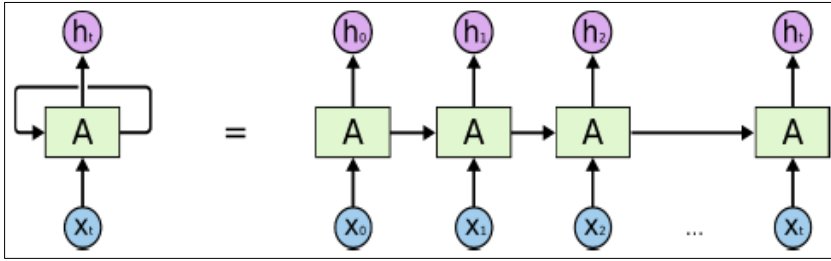
DBN (Derin İnanç Ağı), her katmanın birbiriyle iletişim kurduğu bir RBM (Kısıtlayıcı Boltzmann Makinesi) yığımindan oluşan derin bir sinir ağı modelidir. DBN, görüntü tanıma, video tanıma ve hareket yakalama gibi problemler için kullanılır. Bu modelde herhangi bir katmandaki düğümler birbirleriyle iletişim kurmamaktadır. Her gizli katman, bir sonraki katman için görünür (girdi) katman görevi görür. Öğrenme sürecinde açgözlü algoritmalar kullanılır. Algoritma, girişten başlayıp ileriye doğru ilerleyerek üretken ve verimli ağırlıkları katman belirler. İkinci değerler, en verimli ve etkili ağırlıklara sahip seviyeler arasında baştan itibaren aktarılır (Hinton ve diğerleri, 2012a). 3 katmanlı DBN modeli örneği Şekil 6.7’de gösterilmektedir.



Şekil 6.7. Üç katmanlı DBN modeli örneği (Altan, 2019)

#### 6.2.4. Tekrarlayan sinir ağıları (Recurrent neural networks - RNN)

Tekrarlayan Sinir Ağları (RNN), sıralı veriler ve zaman serileri ile çalışan bir sinir ağı modelidir. Konuşma tanıma, dil çevirisi, doğal dil işleme ve görüntü açıklaması oluşturma gibi problemlerde sıklıkla kullanılmaktadır. Önceki sinir ağıları gibi, öğrenmek için eğitim verilerini kullanırlar. Geleneksel sinir ağlarının bağımsız giriş-çıkış varsayımının aksine, bir RNN'nin çıkışı, dizideki önceki girişlere bağlıdır. Bu model, etiket giriş ve çıkış değerlerine kadar önceki katmanlardan gelen hafızadaki bilgileri kullanır. Bir sonraki adımı tahmin etmek için girdiler arasında bağlantılar oluşturur ve eğitim aşamasında tüm bu bağlantıları hatırlar. Şekil 6.8'de tekrarlayan sinir ağı modeli, aynı zamanda gelirler arasındaki ilişkiyi de göstermektedir.  $h_0$ 'ı elde etmek için  $X_0$  ve  $X_1$  birlikte değerlendirilir. Bir sonraki adımda  $h_0$  ve  $X_1$  bir sonraki adım için yeni bir giriş oluşturur ve bu şekilde devam eder (Kılıç, 2021).



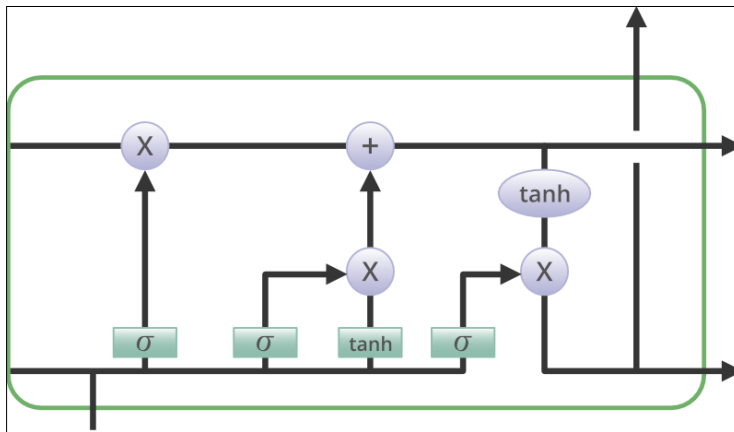
Şekil 6.8. Tekrarlayan sinir ağı modeli (Kılıç, 2021)

Tekrarlayan sinir ağlarının dört farklı türü vardır: bire bir, bire çok, çoktan bire ve çoktan çoğa. Bire bir: Bu, vanilya sinir ağı olarak bilinir ve makine öğrenimi problemlerini tek giriş ve tek çıkışla çözmek için kullanılır. Bire-çok: Bir girdi ve birden fazla çıktının olduğu problemlerde kullanılır. Çoktan Bire: Bir dizi girdiyi alır ve tek bir çıktı verir; Özellikle duygu analizi tespiti gibi durumlarda sıklıkla kullanılır. Çoktan çoğa: bir dizi girdi alır ve sonuç olarak bir dizi çıktı üretir; Genellikle makine çevirisi gibi durumlarda kullanılır. RNN modelleri, kaybolan gradyan ve patlayan gradyan problemlerinden etkilenebilir. Bu nedenle RNN modelinden çeşitli modeller türetilmiştir. BRNN (Çift Yönlü Tekrarlayan Sinir Ağları): RNN mimarisi, gelecekteki verileri işleyebilen ve geçmiş verileri işleyerek çıktıyı tahmin edebilen bir modeldir. GRU (Gated Recurrent Unit): Kısa süreli hafıza sorunlarına çözüm sunar ve gizli katmanında iki adet port bulunur: yenileme

ve sıfırlama portları. Bu kapılar hangi bilgilerin saklandığını ve ne kadarının güncellendiğini kontrol eder (Kılıç, 2021).

### 6.2.5. Uzun kısa hafıza ağları (Long-short term memory - LSTM)

Uzun Kısa Süreli Bellek (LSTM), Tekrarlayan Sinir Ağı (RNN) modelinden türetilmiş ve 1997 yılında Sepp Hochreiter ve Juergen Schmidhuber tarafından RNN ve kaybolan gradyan problemine bir çözüm olarak geliştirilmiştir (Hochreiter ve Schmidhuber, 1997). . RNN modelleriyle karşılaştırıldığında bu ağ modeli, tahmini etkileyen ve yakın geçmişte uzun süre erişilemeyen bilgileri ezberleyerek bağımlılıkları öğrenebilir. LSTM ağları ağırlıklı olarak doğal dil işlemede yaygın olarak kullanılsa da görüntü analizi gibi diğer alanlarda da kullanılmaktadır LSTM ağında bir gizli katmanda üç kapı bulunur ve bu kapılar, çıktıyı tahmin etmek için gereken bilgi akışını yönetir. Şekil 6.9'da LSTM ağının katmanı gösterilmektedir. Giriş Kapısı, mevcut hafızayı güncellemek için hangi giriş değerlerinin kullanılacağını belirler. Unutma Kapısı, hafızadan hangi bilgilerin atılacağını karar verir. Çıkış Kapısı ise, giriş kapısı ve hafıza bilgilerini kullanarak çıkış değerini hesaplar. Gizli katmanlarda, bu işlemleri yönetmek için sigmoid ve tanh aktivasyon fonksiyonları kullanılır (Kılıç, 2021).



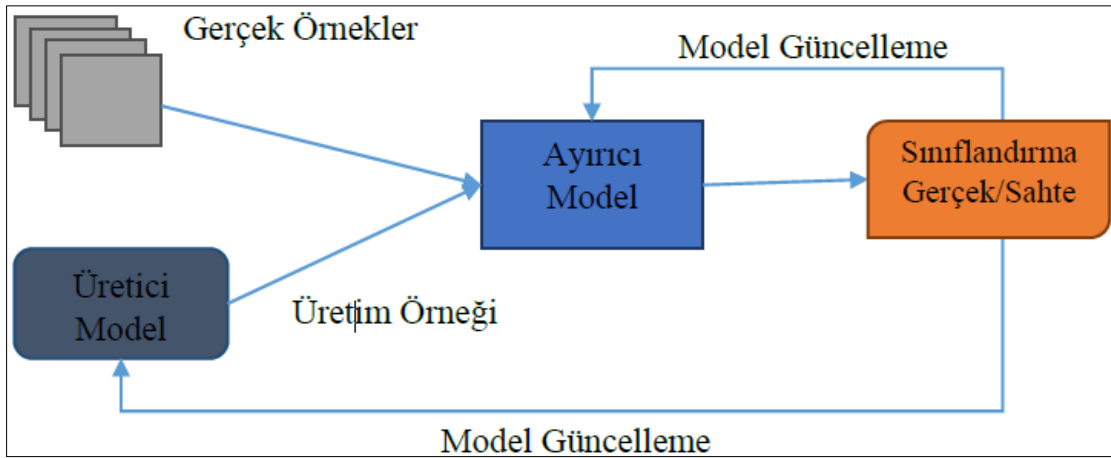
Şekil 6.9. LSTM ağı genel katman yapısı

### 6.2.6. Üretken çekişmeli ağlar (Generative adversarial networks - GAN)

GAN (Generative Adversarial Network), diğer modeller gibi girdi verisinin üretimini tahmin etmek yerine gerçek veri üzerinde öğrenme işlemi gerçekleştirerek gerçek veriye benzeyen sahte ve yapay veri üretir. Bu mimari, eğitim sırasında iki sinir ağı modelini

birbiriyle karşı karşıya getirerek düşmanca bir ilişki yaratır. GAN'lar özellikle sentetik görüntü, ses ve video oluşturmak için yaygın olarak kullanılır. Mevcut veri setlerinin görüntü sınıflandırma problemlerini çözmekte yetersiz kaldığı durumlarda veri büyütme uygulamalarında GAN'lar tercih edilmektedir (Kılıç, 2021).

Bir GAN modeli iki parçadan oluşur: bir üretici ve bir ayırıcı. Jeneratör gerçek verilerden öğrenerek sahte veriler üretir. Ayırıcı, üretici bölümünden alınan bilgilerin gerçek mi yoksa yanlış mı olduğunu sınıflandırır ve işaretler. Her eğitim oturumunda ayırıcı hızlı bir şekilde verinin yanlış olduğunu tespit eder ve üreticiye geri bildirim sağlar. Bu sayede eğitim ilerledikçe jeneratör, ayırmacılığı aldatabilecek yanlış veriler yaratmaya yaklaşır. Başarılı bir eğitimin ardından jeneratör, gerçek verileri sahte verilerden ayırmak için Ayırmacılığın doğruluğunu en aza indirir. Bu ağ yapısında jeneratörün çıkışı doğrudan dağıtıcıya bağlanır. Geri yayılım yoluyla ayırıcı sınıflandırma süreci, üreticilerin ağırlıklarını güncellemek için kullanılan bir sinyal sağlar (Goodfellow ve diğerleri, 2014). GAN modeli Şekil 6.10'da gösterilmektedir.



Şekil 6.10. GAN modeli (Kılıç, 2021)

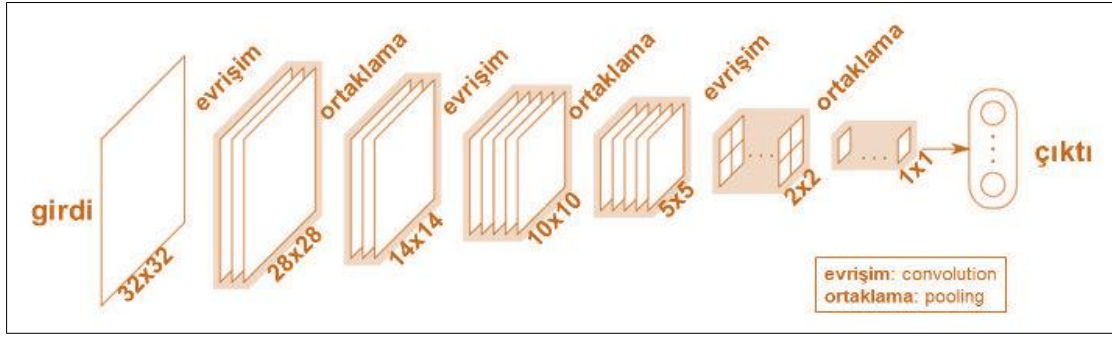
GAN modelinin birçok farklı versiyonu bulunmaktadır. Bazıları; “VanilyaGAN, DCGAN, InfoGAN, DiscoGAN, WassersteinGAN” olarak bilinir.

### 6.3. Evrişimsel Sinir Ağları (Convolutional Neural Network - ESA)

ESA, derin öğrenmenin en çok kullanılan ve popüler modellerinden biridir. Bu model, canlıların görsel korteksinden ilham alan, biyoloji ve matematik alanlarını birleştiren ve bilgisayarlı görme alanında bir devrim niteliğinde olan derin bir yapay sinir ağıdır (Gu ve diğerleri, 2018). Evrişimli, adını filtreleme yoluyla çıkarılan özellik işlemlerinden alır. Görüntü piksel matrisleri üzerinde filtreleme işlemlerini gerçekleştirir ve karakteristik özellikleri otomatik olarak çıkarır.

ESA modeli 1998 yılında Yann LeCun tarafından tanıtıldı ve LeNet olarak adlandırıldı (Le Cun ve diğerleri, 1998). LeNet mimarisi, 32x32 görsellerdeki el yazısıyla yazılan sayıları tanıyabiliyordu. Ancak o yıllarda yeterli bilgi ve donanımın olmaması nedeniyle ESA modelinin uygulaması sınırlıydı. Alex Krizhevsky ve meslektaşları tarafından 2012 yılında geliştirilen AlexNet modeli, ImageNet yarışmasını kazanarak ESA modelini popüler hale getirdi (Krizhevsky ve diğerleri, 2012). Grafik işlemcilerin gelişmesi ve teknolojinin yaygınlaşması, ESA'nın görüntü boyutlarının işlenmesinde daha da yaygınlaşmasına katkıda bulundu. Günümüzde ESA'ları özellikle görüntü sınıflandırma ve nesne tanıma gibi görüntü analizi gerektiren durumlar için önerilmektedir (Kılıç, 2021).

ESA modeli, klasik makine öğrenimi tekniklerinden farklı olarak ham verilerden sınıflandırma sürecinde özellik çıkarma adımı gerçekleştirebilmesi nedeniyle görüntü sınıflandırma problemlerinin çözümünde önemli bir avantaj sunmaktadır. Ancak ESA modellerinin eğitimi ve performansı, kullanılan veri miktarıyla doğrudan ilişkilidir. Modeli eğitmek ve tahmin doğruluğunu artırmak için büyük miktarda veri gereklidir. Ancak büyük miktarda verinin kullanılması bilgisayar maliyetini arttırmakta ve yüksek işlem gücü gerektirmektedir. Bu sorunlara çözüm olarak günümüzde birçok şirket, derin öğrenme uygulamaları geliştirmek için ücretsiz bulut tabanlı hizmetler sunmaktadır. Yeterli veri yoksa ESA modellerini sıfırdan eğitmek yerine daha önce yüksek boyutlu veri kümeleri üzerinde eğitilmiş modellerin ağırlıkları kullanılabilir. Bu sürece transfer öğrenme denir. Bir ESA modeli genellikle iki bölümden oluşur: özellik öğrenme ve sınıflandırma. Özellik öğrenme açısından evrişim, aktivasyon ve havuzlama katmanları vardır. Sınıflandırma bölümü yumuşatıcı, tamamen bağlantılı katmanlara ve sınıflandırıcılara sahiptir (Kılıç, 2021). Bir ESA modelinin genel yapısı şematik gösterimi Şekil 6.11'de gösterilmektedir.

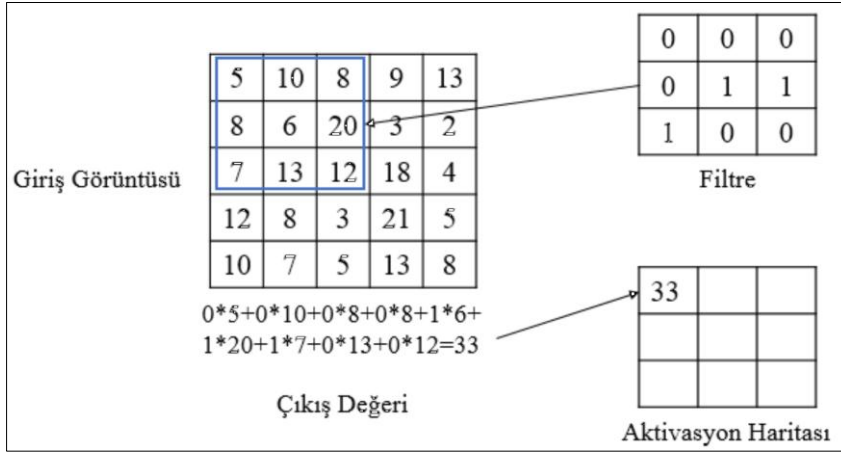


Şekil 6.11. ESA modeli genel yapısı şematik gösterimi (Kızrak ve Bolat, 2018)

### 6.3.1. Evrişim katmanı

Bir ESA modelinin eğitim ve test aşamalarında, görüntüler ham haliyle veya sayısallaştırılmış bir matris formatında giriş olarak verilebilir. Bu matrislerde, tanımlanan filtre değerleri ile filtrenin bulunduğu matrisin değerlerinin çarpılıp toplanmasına evrişim denir. ESA modelinin temel işlemlerinden biri olan bu evrişim işleminin amacı, görüntüdeki yüksek seviyeli özellikleri yakalamaktır. Evrişim işleminde kullanılan filtreye "çekirdek" denir. Çekirdek üzerindeki sayısal değerler, ağırlık matrisini temsil eder. Ağırlık matrisi, görüntü matrisinde adım adım hareket eder ve her adımda elde edilen toplam değer, çıktı matrisine kaydedilir (Kılıç, 2021).

Adım sayısı, yazdırma matrisinin görüntü boyunca bir veya daha fazla adımda hareket edip etmeyeceğini belirler. Çıkış dizisinin boyutunu belirlerken bu önemlidir. Tek kanallı görüntülerde evrişim işlemi anında yapılırken, 3 kanallı renkli görüntülerde bu işlemler her kanal için ayrı ayrı gerçekleştirilir. Sonuç matrisinin boyutu ile görüntü matrisinin evrişim işleminden önceki boyutu arasında boyut farkı vardır. Çıkış matrisinin boyutu sabit kalırsa matrisin etrafına pikseller eklenir (İnik ve Ülker, 2017). Evrişim süreci işlemi Şekil 6.12'de gösterilmektedir.



Şekil 6.12. Evrişim süreci işlemi (Kılıç, 2021)

### 6.3.2. Aktivasyon katmanı

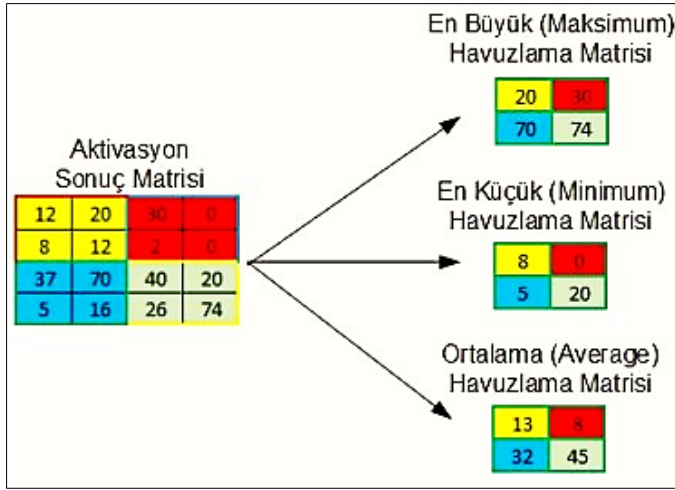
Aktivasyon katmanı genellikle evrişim katmanından sonra gelir. Aktivasyon fonksiyonları olarak sigmoid, hiperbolik tanjant, sinüs, adım ve eşik fonksiyonları kullanılmaktadır. Çok katmanlı sinir ağlarında doğrusal olmayan dönüşümler kullanılır. Birçok aktivasyon fonksiyonu olmasına rağmen derin öğrenme mimarilerinde Rectified Linear Unit (Relu) fonksiyonu kullanılmaktadır (Nair ve Hinton, 2010; Krizhevsky ve diğerleri, 2012). Aktivasyon işlemi sonucunda elde edilen değer negatif ise 0, pozitif ise 1 olmaktadır (Jarrett ve diğerleri, 2009).

### 6.3.3. Havuzlama katmanı

Artıklık katmanı ve ardından ReLU etkinleştirme işlevi, veri hacmini azaltır ve ağı bir sonraki katmanına yapılan girişin boyutunu azaltır. Bağlantı işlemi sırasında veri kaybı yaşanabilse de ağı işleyebileceği veri miktarını azaltır. Bu azalma ağı daha hızlı olmasını sağlar. Ağıdaki veri miktarının azaltılması, hesaplamaların ve kullanılan belleğin azaltılması anlamına gelir (Hinton ve diğerleri, 2012a).

Toplama katmanında işlem belirli bir filtre kullanılarak gerçekleştirilir. Bu filtre veri tabanını inceler ve matriste doğru değeri bulmaya çalışır. NxN boyutsal filtre ile yapılan işlemlerde en sık kullanılan havuzlama yöntemleri maksimum değer (maksimum havuzlama) ve ortalama değer (ortalama havuzlama) bulunmasıdır. Maksimum birleştirme işleminde, bir NxN boyutlu matrisin en büyük değeri, 1x1 boyutunda yeni bir

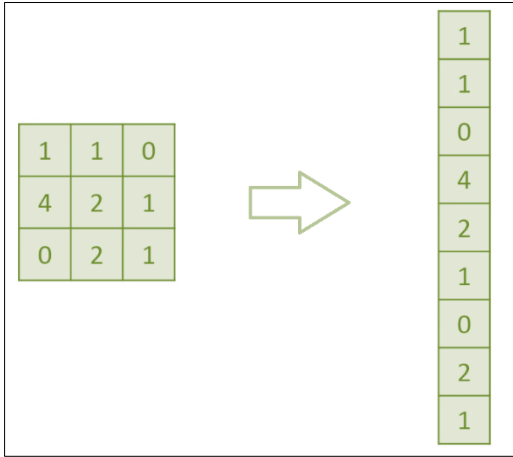
veri matrisi değeri oluşturur. Bu, Zeiler ve Fergusand'ın 2013 yılında yaptığı bir çalışmada görülen bir süreçtir (Zeiler ve Fergus, 2013). Ortalama toplama işleminde ise  $N \times N$  matrisinin tüm değerlerinin ortalaması alınarak yeni  $1 \times 1$  veri matrisi değeri elde edilir. Bu toplama yöntemleri, bilgileri bir özellik haritasında toplayarak, önemli özellikleri vurgulayarak veya daha genel bir temsil oluşturarak boyutluluğun azaltılmasına yardımcı olur. Bu süreçler genellikle derin öğrenme modellerinde kullanılarak ağız öğrenme kapasitesinin artırılmasına ve bilgisayar maliyetlerinin azaltılmasına yardımcı olur (Doğan ve Türkoğlu, 2019). Şekil havuzlama katmanları şematik gösterimi Şekil 6.13'te verilmiştir.



Şekil 6.13. Şekil havuzlama katmanları şematik gösterimi (Metlek ve Kayaalp, 2020)

#### 6.3.4. Düzleştirme katmanı

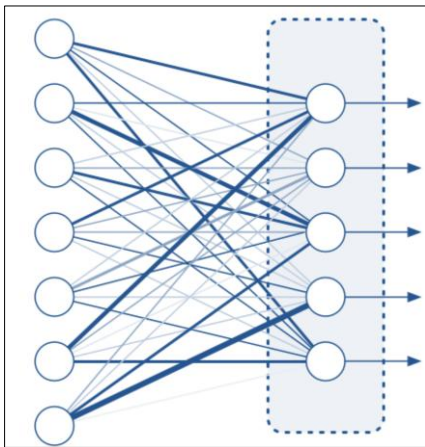
Evrişim, aktivasyon ve toplama işlemlerinin ardından görüntülerden öğrenilen özellikler ağıdaki çok boyutlu bir matriste depolanır. Düzleştirme katmanında çok boyutlu matrisin özellikleri tek boyutlu özellik vektörüne dönüştürülür. Şekil 6.14 düzleştirme işlemine ait sürecini göstermektedir (Kılıç, 2021).



Şekil 6.14. Düzleştirme işlemine ait süreç

### 6.3.5. Tam bağlı katman

Bu katmandaki tüm nöronlar bir matris olarak görünür. Bu katmandaki tüm nöronlar bu katmana bağlıdır. Tüm aktivasyonlarla tamamen entegredir. Önceki katmana bağlı olarak tamamen bağlı katman özelliği görüntülenir. Nesnenin tanımlayıcı özelliklerinin hangi sınıfa ait olduğunu belirler. Model ele alınacak olursa; Bir kişinin bir örüntü içerisinde tanımlanmasını sağlayan özellikler, aktivasyon haritalarında en yüksek değere sahip olan nöronda yer almaktadır. Tamamen bağlı bir katman, bir sınıfla yüksek ilişkisi olan üst düzey işlevlere bakar. Nöronların bu özellikleri gösteren ağırlıklarla incelenmesi onların hangi sınıfa ait olduklarını ortaya çıkarır (Lin ve diğerleri, 2013; LeCun ve Bengio, 1995). Şekil 6.15'te tamamen bağlı bir katman yapısının bir örneğini göstermektedir.

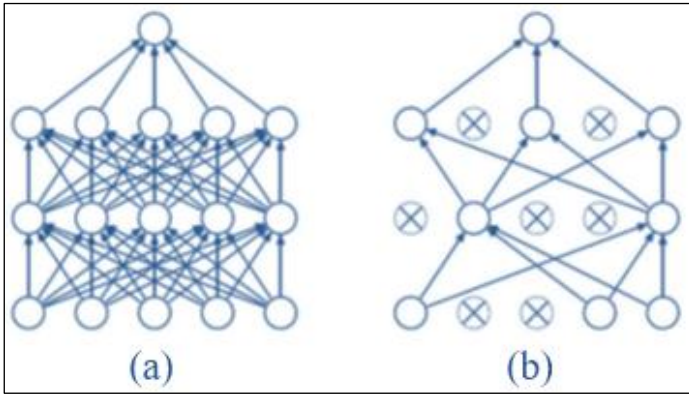


Şekil 6.15. Tam bağlı katman yapısı örneği

### 6.3.6. Dropout (Bırakma) katmanı

Çok katmanlı yapay sinir ağlarında, aşırı öğrenme adı verilen olgu, ağın veri kümesini ezberlemesi durumunu ifade eder ki bu istenmeyen bir durumdur. Ağın ezber yapmasını engellemek için, ağdaki bazı düğümlerin ortadan kaldırılması yöntemi kullanılır. Bu, ağın ezberlenmesini önlemeye çalışır (Srivastava ve diğerleri, 2014).

Dropout katmanı, tam bağlı katmanlar için bir düzenleme mekanizması olarak Hinton ve ekibi tarafından önerilmiştir. Dropout işlemi, sinir ağının genelleme yeteneğini artırdığı ve yapılan test performanslarıyla kanıtlandığı bir düzenleme yöntemidir (Hinton ve diğerleri, 2012b). Şekil 6.16'da çok katmanlı sinir ağı ile dropout işlemi uygulanmış sinir ağı yapısı gösterilmektedir. Dropout işlemi uygulanmış sağdaki şekilde, bazı düğümlerin rastgele olarak kaldırıldığı görülmektedir. Bu yöntem, ağın farklı düğümlerini eğitirken birbirinden bağımsız olarak düşünmesini sağlayarak aşırı öğrenmeyi azaltmaya yardımcı olur (Doğan ve Türkoğlu, 2019).



Şekil 6.16. Yapay sinir ağı (a), Dropout uygulanmış sinir ağı (b)

### 6.3.7. Sınıflandırma katmanı

Tam bağlı katmandan sonra gelir ve sınıflandırma işleminde kullanılacak öge sayısı kadar sonuç üretir. Her bir sonuç, bir sınıfı temsil eder. Sınıflandırıcı katman olarak da bilinen bu katmanda, farklı türde sınıflandırıcılar kullanılabilir, ancak genellikle softmax sınıflandırıcısı tercih edilir (Ciresan, 2001).

### 6.3.8. Yumuşatma (Softmax) katmanı

Sınıflandırma katmanının ardından gelir ve kendisinden önceki tam bağlı katmandan girdi olarak sınıflandırma yapar. Bu katman, olasılıksal bir giriş verisini belirli bir sınıfa ait olma durumunu gösteren bir çıkış üretir. Her bir sınıf için olasılık değerini hesaplamak için softmax işlemi gerçekleştirir. Bu işlem genellikle çapraz entropi kullanılarak gerçekleştirilir (Tang, 2013).

### 6.3.9. Normalizasyon katmanı

Derin evrimsel sinir ağlarının eğitimi sırasında hesaplama açısından önemli bir süreç olan aktivasyonları normalize etmek için kullanılır. Bu katman, genellikle ReLU katmanı sonrasında gelir ve gizli katmanlardaki durumları stabilize etmede etkilidir (Li ve diğerleri, 2015).

Normalizasyon ağın performansını etkileyebilir. Bu nedenle, katmanlardan gelen veriler belirli bir düzen içinde olmalıdır. Girdi verileri çok büyük veya çok küçük olabilir, bu nedenle bu değerlerin belirli bir aralıkta normalize edilmesi, eğitim ve süreç açısından önemlidir. Girdi verilerinin normalize edilerek belirli bir aralıkta temsil edilmesi gereklidir (Ioffe ve Szegedy, 2015).

## 6.4. Optimizasyon Algoritmaları

Optimizasyon teknikleri, programın yürütülmesi sırasında ortaya çıkabilecek hataları en aza indirmek için kullanılır. Bu teknikler genellikle gradyan inişi olarak adlandırılır. Optimizasyon yöntemleri birden fazla adımdan oluşan sistematik bir yaklaşım içerir. Bu işlem sırasında gerçekleştirilen adım sayısı öğrenme oranı olarak bilinir. Öğrenme oranı için uygun bir değer seçmek çok önemlidir. Küçük bir öğrenme oranının seçilmesi çözüm sürecini uzatırken, büyük bir öğrenme oranı minimum noktanın aşılmasına neden olabilir. En yaygın kullanılan optimizasyon yöntemleri arasında SGD, Adagrad, RMSProp, Adadelata ve Adam algoritmaları bulunmaktadır.

Bu tez çalışmada Adam, AdaGrad, Adadelata, SGD ve RMSprop algoritmaları kullanılmıştır.

### 6.4.1. Adam

Adam'ın optimizasyon algoritması derin öğrenme uygulamalarında yaygın olarak kullanılmaktadır. Bu algoritma, diğerleri gibi, her parametrenin öğrenme oranını değiştirir. Adam algoritması, hızlanma hızını bir miktar azaltır ve doğru parametreleri bulmak için hassas bir arama yapılmasına olanak tanır. Adam'ın optimizasyon algoritması çok hızlıdır ve alt küme toplama işlemini hızlı bir şekilde gerçekleştirir. Bu, öğrenme kaybı ve yüksek varyans sorununu çözer. Adam algoritması, AdaDelta ve RMSprop algoritmaları gibi önceki karelerin gradyanlarının üstel olarak azalan ortalamasını saklamanın yanı sıra, önceki gradyanların üstel olarak azalan ortalamasını da saklar RMSProp ve Momentum yöntemlerinin avantajlı yönlerinin birleştirilmesiyle önerilen bu gradient descent algoritması, Momentum yöntemindeki  $V$  ve RMSProp yöntemindeki  $S$  bileşenlerini bir araya getirir (Kingma ve diğerleri, 2015; Ruder, 2016). Aşağıda adam optimizasyon algoritması ile ilgili matematiksel ifadeler verilmiştir:

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{S}_t + \epsilon}} \cdot \hat{V}_t \quad (6.1)$$

$$\hat{V}_t = \frac{V_t}{1 - \beta_1^t} \quad (6.2)$$

$$\hat{S}_t = \frac{S_t}{1 - \beta_2^t} \quad (6.3)$$

$$V_t = \beta_1 V_{t-1} + (1 - \beta_1) \frac{\partial L}{\partial w_t} \quad (6.4)$$

$$S_t = \beta_2 S_{t-1} + (1 - \beta_2) \left( \frac{\partial L}{\partial w_t} \right)^2 \quad (6.5)$$

Bu algoritma, RMSProp yöntemindeki adaptif öğrenme oranını ( $S$ ) ve Momentum yöntemindeki hızlandırma etkisini ( $V$ ) birleştirerek daha hızlı ve kararlı bir optimizasyon süreci sağlamaktadır. Burada  $S$  ve  $V$  başlangıç olarak 0,  $\alpha=0.001$ ,  $\beta_1=0.9$ ,  $\beta_2=0.999$  ve  $\epsilon$  ise  $10^{-8}$  olarak tercih edilmektedir (Ruder, 2016; Kingma ve diğerleri, 2015).

### 6.4.2. AdaGrad

AdaGrad (Adaptive gradient algorithm) birçok optimizasyon algoritmasında öğrenme oranı tüm parametreler ve her döngü için sabittir. AdaGrad'ın optimizasyon algoritması öğrenme deneyimini değiştirir. Bu algoritma her t adımında her parametrenin öğrenme 'n'sini değiştirir. AdaGrad ikinci dereceden bir optimizasyon algoritmasıdır ve hata fonksiyonunun türeviyle çalışır (Duchi ve diğerleri, 2011). Bu algoritmanın avantajları öğrenme oranının manuel olarak ayarlanmaması ve seyrek verilerle çalışabilmesi olsa da ikinci dereceden türev hesaplamaları pahalı ve zaman alıcı olabilir. SGD ve Momentum yöntemlerindeki sabit öğrenme katsayısı problemini ortadan kaldırmak için önerilmiş bir optimizasyon algoritmasıdır. Her parametre için öğrenme oranını geçmiş gradyanlara göre uyarlayan bir optimizasyon algoritmasıdır. Bu yöntem, özellikle seyrek veri veya özelliklerin ölçeklerinin önemli ölçüde farklılık gösterdiği durumlarda etkilidir (Doshi, 2019; Ruder, 2016). Aşağıda Adagrad optimizasyon algoritması ile ilgili matematiksel ifadeler verilmiştir:

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (6.6)$$

$$S_t = S_{t-1} + \left( \frac{\partial L}{\partial w_t} \right)^2 \quad (6.7)$$

$S_0=0$ ,  $\epsilon$  genellikle sıfıra bölme hatalarını engellemek için  $10^{-7}$  olarak alınmaktadır (Ruder, 2016).

### 6.4.3. Adadelta

Adadelta optimizasyon algoritması, önceki döngülerde kareleri saklanan tüm gradyanları biriktirmek yerine, geçmiş gradyan hesaplamalarının boyutunu sabit bir "w" boyutuyla sınırlayarak öğrenme hızı düşürme sorununa çözüm getirmiştir. Bu sınırlama işlemi, üstel olarak hareketli ortalamadaki tüm gradyanların toplamını değil, belirli bir pencere boyutundaki gradyanları içerir. Bu yaklaşım, öğrenme hızındaki azalmanın neden olduğu eğitim kesintisini engelleyerek daha etkili bir optimizasyon sağlar Adadelta yönteminde, adagrad ve RMSProp yöntemlerinden farklı olarak öğrenme katsayısı seçme zorunluluğu yoktur. Öğrenme katsayısı yerine, geçerli ağırlıklar ile güncellenen ağırlıklar arasındaki

farkı ifade eden delta değerlerinin karelerinin momentumlu toplamları kullanılmaktadır (Zeiler, 2012; Ruder, 2016). Adadelta optimizasyon algoritması matematiksel ifadesi şu şekildedir:

$$w_{t+1} = w_t - \frac{\sqrt{D_{t-1}}}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (6.8)$$

$$D_t = \beta D_{t-1} + (1 - \beta) [\Delta w_t]^2 \quad (6.9)$$

$$S_t = \beta S_{t-1} + (1 - \beta) \left[ \frac{\partial L}{\partial w_t} \right]^2 \quad (6.10)$$

$$\Delta w_t = w_t - w_{t-1} \quad (6.11)$$

Burada  $S$  ve  $D$  başlangıçta 0,  $\beta=0.95$  ve  $\epsilon$  ise  $10^{-6}$  olarak tercih edilmektedir (Zeiler, 2012; Ruder, 2016)

#### 6.4.4. SGD

Stokastik Dereceli Alçalma (SGD) bu algoritma, eğitilen sinir ağı modelinin parametrelerini her seferinde güncellemeye çalışır. Algoritma, her eğitim döngüsünde parametre güncellemelerini gerçekleştirecek şekilde bir kayıp fonksiyonu hesaplar. İniş algoritmasına göre daha az hafıza yoğunluğuna sahiptir ve küçük bir alanda kullanılabilir. Ancak sık parametre güncellemeleri büyük parametre varyasyonlarına neden olur, bu nedenle öğrenme oranının azaltılması gerekir (Ruder, 2016).

İtme (Momentum), SGD algoritmasıyla birlikte kullanılan bir tekniktir. SGD'nin yüksek varyans sorununu azaltmak için geliştirilmiştir, çünkü SGD'de sık parametre güncellemeleri genellikle dalgalanmaya neden olur. Bu teknik, parametre güncellemelerine bir tür "momentum" ekleyerek yayılım yönünde daha hızlı bir şekilde yakınsamayı sağlar ve bu sayede dalgalanma problemini aşar. İtme algoritması, geçmiş gradyanlara dayalı bir momentum terimi kullanır. Bu momentum terimi, önceki güncellemelerin birleştirilmiş etkisini temsil eder. Bu nedenle, güncelleme yapılırken geçmiş gradyanların etkisiyle birlikte güncel gradyanın etkisi de dikkate alınır. Bu algoritmanın kullanımında,

momentum parametresinin belirlenmesi önemlidir. Momentumun değeri, algoritmanın ne kadar geçmiş bilgiyi dikkate alacağını belirler. Bu parametre, öğrenme sürecinin performansını etkileyen kritik bir faktördür (Qian, 1999; Doshi, 2019). SGD optimizasyon algoritması matematiksel olarak şu şekilde ifade edilebilir:

$$w_{t+1} = w_t - \alpha \frac{\partial L}{\partial w_t} \quad (6.12)$$

Burada,  $w_t$  modelin mevcut ağırlıklarıdır ve  $\alpha$  öğrenme oranını belirler.  $\frac{\partial L}{\partial w_t}$ , kayıp fonksiyonunun gradyanı olup, ağırlıkların nasıl güncellenmesi gerektiğini gösterir. SGD, ağırlıkları bu gradyan yönünde güncelleyerek kayıp fonksiyonunu minimize etmeye çalışır. Denklem, modelin parametrelerini kayıp fonksiyonunun eğimine göre günceller ve modeli en iyi sonucu verecek şekilde ayarlamaya çalışır (Qian, 1999; Doshi, 2019).

#### 6.4.5. RMSprop

RMSProp, öğrenme hızının azalması sorununu çözmek için Hinton tarafından tanıtılan bir optimizasyon algoritmasıdır ve AdaDelta ile AdaGrad algoritmalarından bağımsız olarak geliştirilmiştir. Bu algoritma, önceki gradyanların karelerini hesaplar, ancak önceki gradyanların karelerinin sürekli olarak artmasını engeller ve böylece eğitim sürecinde bu değerlerin azalmasına neden olur. Kareli gradyanların toplamı, bir bozulma oranı "a" ile çarpılır ve (1-a) ağırlıklı mevcut gradyan değerine eklenir. Güncelleme adımında, geçerli gradyan değeri, gradyan karelerinin toplamına bölünerek bir boyut boyunca hareketi hızlandırırken, diğer boyut boyunca hareketi yavaşlatacak şekilde ayarlanır (Ruder, 2016; Yazan ve Talu, 2017).

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (6.13)$$

$$S_t = \beta S_{t-1} + (1 - \beta) \left( \frac{\partial L}{\partial w_t} \right)^2 \quad (6.14)$$

Burada  $S$  başlangıçta 0,  $\alpha=0.001$ ,  $\beta=0.9$  ve  $\epsilon$  ise  $10^{-6}$  olarak tercih edilmektedir (Ruder, 2016).

## 6.5. Aktivasyon Fonksiyonları

Sinir Ağları, verileri sınıflandırmak ve tahmin etmek için kullanılan çok sayıda nöron katmanından oluşan bir ağ yapısını ifade eder. Bu ağ, giriş katmanı, bir veya daha fazla gizli katman ve bir çıkış katmanından oluşur. Her katman, düğümler içerir ve her düğüm, bir katmandan diğerine bilgi iletilirken dikkate alınan belirli bir ağırlığa sahiptir. Bir sinir ağında aktivasyon fonksiyonunun olmaması durumunda, çıkış sinyali basit bir doğrusal fonksiyon, esasen birinci dereceden bir polinom olacaktır. Doğrusal denklemlerin çözülmesi kolay olsa da, yetenekleri kısıtlıdır ve veri içindeki karmaşık kalıpları anlama ve tanımlama kapasitesinden yoksundur. Aktivasyon fonksiyonları olmayan bir sinir ağı esasen Doğrusal Regresyon Modeli olarak çalışır ve genellikle sınırlı performans ve kabiliyet sergiler. İdeal olarak, bir sinir ağı yalnızca doğrusal işlevleri öğrenmenin ve hesaplamının ötesine geçmeli; daha karmaşık görevleri yerine getirmede mükemmel olmalıdır. Bu, görüntüler, videolar, ses, konuşma ve metin gibi karmaşık veri türlerinin modellenmesini içerir. Aktivasyon fonksiyonları, ağa doğrusal olmama özelliğinin kazandırılmasında çok önemli bir rol oynayarak çeşitli ve karmaşık veri kümelerinde bulunan karmaşık ilişkileri yakalamasını ve temsil etmesini sağlar. Öğrenme gibi yapay sinir ağı teknikleri Modelin birden fazla gizli katmana sahip olduğu karmaşık, yüksek boyutlu ve doğrusal olmayan veri kümelerini anlamlandırır (Sharma ve diğerleri, 2017). Bazı aktivasyon fonksiyonları ve açıklamaları örnek olarak verilmiştir.

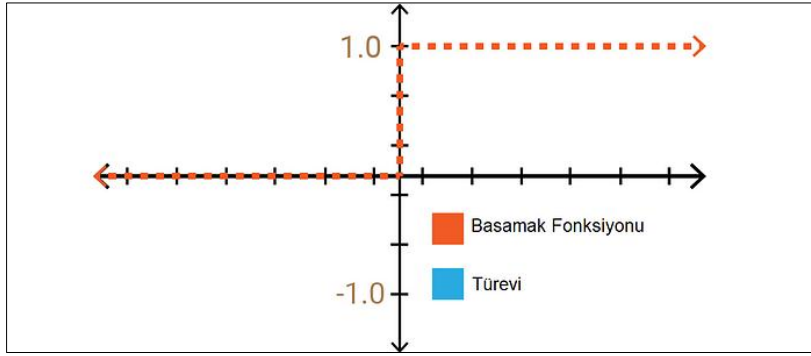
### 6.5.1. Basamak (step) fonksiyonu

İkili Adım Fonksiyonu var olan en basit aktivasyon fonksiyonudur ve Python'da basit if-else ifadeleri ile uygulanabilir. İkili bir sınıflandırıcı oluştururken genellikle ikili aktivasyon fonksiyonu kullanılır. Ancak, ikili adım fonksiyonu hedefte çok sınıflı sınıflandırma durumunda kullanılır taşınabilir. Ayrıca, ikili adım fonksiyonunun gradyanı sıfırdır, bu da geri dönüşte bir engelle neden olabilir yayılma adımı, yani eğer türevini hesaplırsak  $f(x)$   $x$ 'e göre sıfıra eşittir (Sharma ve diğerleri, 2017). Matematiksel olarak ikili adım fonksiyonu şu şekilde tanımlanabilir:

$$f(x) = 1, x \geq 0 \quad (6.15)$$

$$f(x) = 0, x < 0 \quad (6.16)$$

Basamak fonksiyonu ve türevi Şekil 6.17’ de verilmiştir.



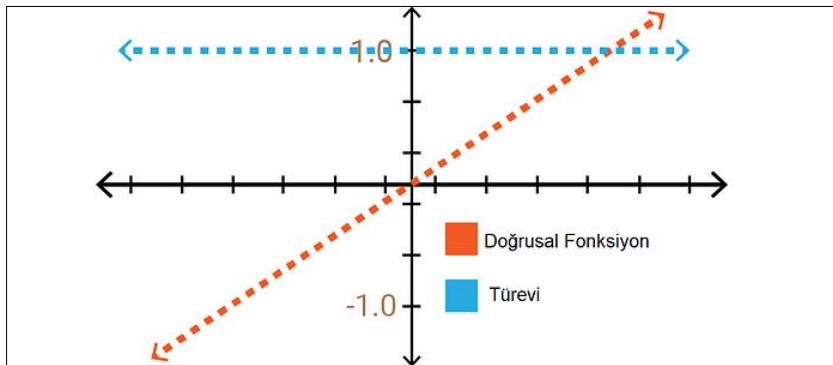
Şekil 6.17. Basamak fonksiyonu ve türevi (Kızrak, 2019)

### 6.5.2. Doğrusal (linear) fonksiyon

Doğrusal aktivasyon fonksiyonu giriş ile doğru orantılıdır. İkili adım fonksiyonunun ana dezavantajı sıfır gradyana sahip olmasıdır çünkü ikili adım fonksiyonunda x bileşeni yoktur. Bunu ortadan kaldırmak için doğrusal fonksiyon kullanılabilir. Bu fonksiyon şu şekilde tanımlanabilir:

$$F(x) = ax \quad (6.17)$$

a değişkeninin değeri herhangi bir sabit değer olabilir kullanıcı tarafından seçilir (Sharma ve diğeleri, 2017). Doğrusal fonksiyon ve türevi Şekil 6.18’de verilmiştir.



Şekil 6.18. Doğrusal fonksiyon ve türevi (Kızrak, 2019)

### 6.5.3. Sigmoid

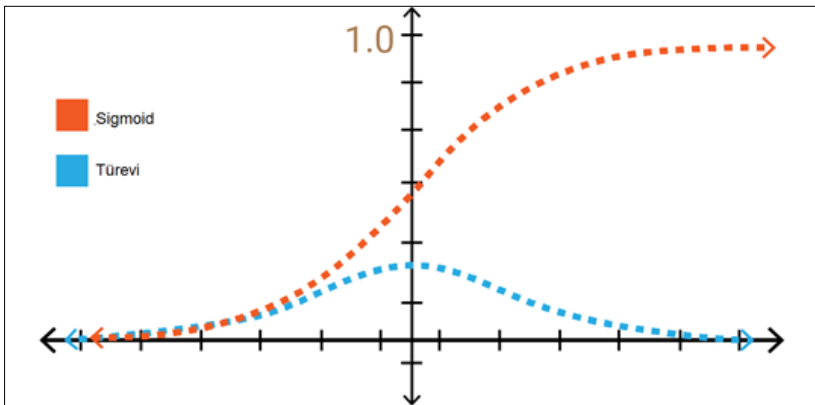
Bu, 0'dan 1'e kadar bir aktivasyon fonksiyonudur (Wang ve diğeri, 2019). İkili sınıflandırma problemlerinde sonuç olasılığını hesaplamak faydalıdır. Çıkış negatif değerler alıp 0'a gidiyorsa, daha yüksek değerler 1'i tahmin eder. Sigmoid fonksiyonu, derin ağlarda gradyan kaybı probleminden etkilenebileceği için az sayıda gizli katmana sahip ağ yapılarında kullanıma uygundur (Kılıç, 2021). Doğrusal olmayan bir fonksiyon olduğu için en yaygın kullanılan aktivasyon fonksiyonudur. Sigmoid fonksiyonu 0 ile 1 aralığındaki değerleri dönüştürür:

$$f(x) = 1/e^{-x} \quad (6.18)$$

Sigmoid fonksiyon sürekli türevlenebilir ve düzgün bir S-şekilli fonksiyondur. Fonksiyonun türevi şöyledir:

$$f'(x) = 1 - \text{sigmoid}(x) \quad (6.19)$$

Ayrıca, sigmoid fonksiyonu sıfır etrafında simetrik değildir, bu da nöronların tüm çıkış değerlerinin işaretlerinin aynı olacağı anlamına gelir. Bu sorun sigmoid fonksiyonu ölçeklendirilerek iyileştirilebilir (Sharma ve diğeri, 2017). Sigmoid fonksiyonu ve türevi Şekil 6.19'da verilmiştir.



Şekil 6.19. Sigmoid fonksiyonu ve türevi (Kızrak, 2019)

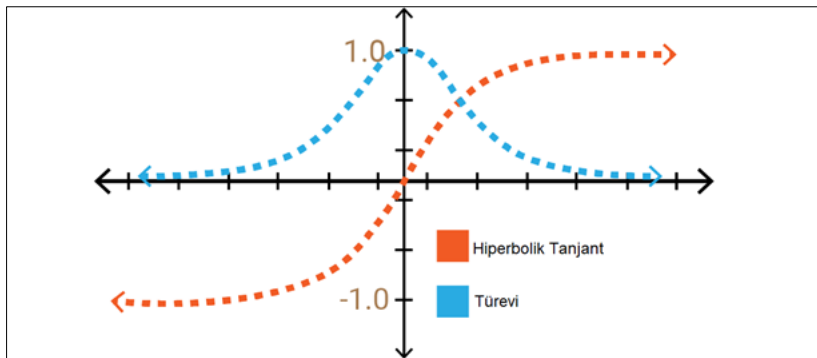
#### 6.5.4. Tanh

Hiperbolik tanjant (tanh) aktivasyon fonksiyonu, sigmoid fonksiyonuna oldukça benzer ve genellikle ikili sınıflandırma problemlerinde kullanılır. Ancak, bu aktivasyon fonksiyonunun bir özelliği, çıkışın -1 ile 1 arasında bulunmasıdır. Giriş değeri ne kadar büyükse, çıkış 1'e çok yakın olurken, giriş değeri ne kadar küçük ve negatifse, çıkış -1'e çok yakın olacaktır (Wang ve diğerleri, 2019).

Tanh fonksiyonunun sigmoid'e benzer olmasına rağmen, tanh fonksiyonu giriş değerlerini -1 ile 1 arasında sıkıştırırken, sigmoid fonksiyonu 0 ile 1 arasında sıkıştırır. Bununla birlikte, tanh fonksiyonunun derin sinir ağlarında kaybolan gradyan problemine neden olabileceği bir sorundur. Bu, özellikle ağın daha derin katmanlarında gradyanların çok küçük olması veya çok büyük olması durumunda ortaya çıkabilir. Bu durum, ağın öğrenme sürecini zorlaştırabilir (Kılıç, 2021). Tanh fonksiyonu matematiksel olarak şu şekilde tanımlanabilir:

$$f(x) = 2\text{sigmoid}(2x) - 1 \quad (6.20)$$

Tanh fonksiyonu ve türevi Şekil 6.20' de verilmiştir.



Şekil 6.20. Tanh fonksiyonu ve türevi (Kızrak, 2019)

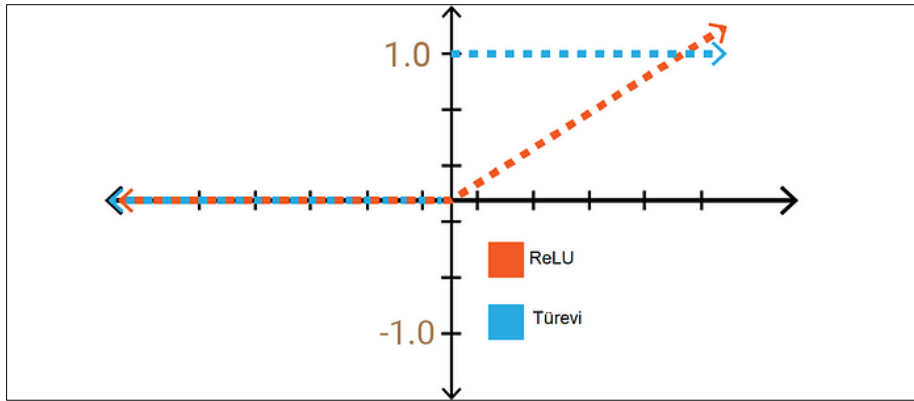
#### 6.5.5. ReLU

Düzgün Doğrusal Birim (ReLU), derin öğrenme uygulamalarında en yaygın kullanılan aktivasyon fonksiyonudur. Çoğu evrişimli sinir ağı mimarisi, ReLU aktivasyon fonksiyonlarını kullanır. Basit ve etkili bir aktivasyon işlemidir. ReLU negatif değerler için

0 değeri üretir ve hiçbir şey yapmaz. Giriş iyiye çıkışla aynı değeri değiştirecektir. Bu aktivasyonun temel avantajı, tüm nöronların giriş değerinin tamamını aynı anda etkinleştirebilmesidir. Diğer çalışmaların aksine, ReLU aktivasyon fonksiyonu derin sinir ağlarındaki gradyan kaybı sorununa karşı daha az duyarlıdır (Nair ve Hinton, 2010; Gu ve diğerleri, 2018). Matematiksel olarak şu şekilde ifade edilebilir:

$$f(x) = \max(0, x) \quad (6.21)$$

ReLU fonksiyonu ve türevi Şekil 6.21’de verilmiştir.



Şekil 6.21. ReLU fonksiyonu ve türevi (Kızrak, 2019)

### 6.5.6. Sızdıran ReLU

ReLU fonksiyonunun negatif değerli nöronları aktif hale getirmemesi ve o nöronlara gelen girdilerin her zaman negatif değerlere sahip olmaması ve o nöronlarda herhangi bir öğrenme sürecinin gerçekleşmemesi. Bu soruna "ölmekte olan ReLU" sorunu denir. Bu sorunun üstesinden gelmek için ReLU fonksiyonunun çeşitli versiyonları geliştirilmiştir. Leaky ReLU işlevi, 0 yerine 0,01 olan negatif değerleri döndürür ve x ekseninde küçük bir doğrusal bileşen olarak çizilir. Bu süreç sayesinde ilgili nöronda öğrenme gerçekleşebilir. Parametrik ReLU (PReLU) işlevi, 0,01 gibi sabit bir değer yerine değiştirilebilen ve öğrenilebilen bir "a" parametresi kullanarak bu sorunu çözmektedir (Sharma ve diğerleri, 2017). Sızdıran ReLU matematiksel olarak şu şekilde ifade edilebilir:

$$f(x) = 0.01x, x < 0 \quad (6.22)$$

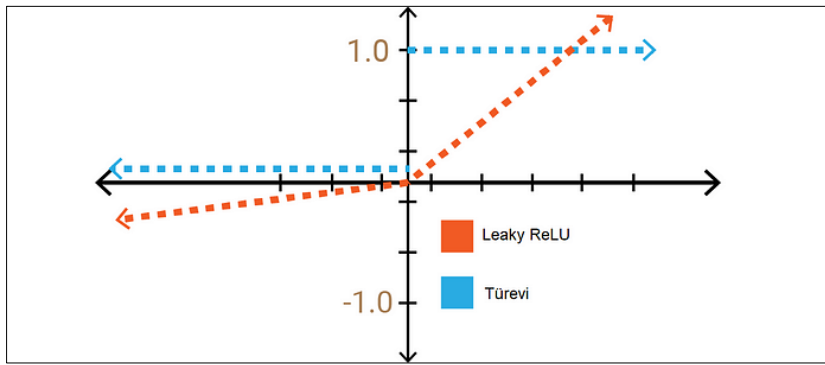
$$f(x) = x, x \geq 0 \quad (6.23)$$

Parametrik ReLU şu şekilde ifade edilir:

$$f(x) = x, x \geq 0 \quad (6.24)$$

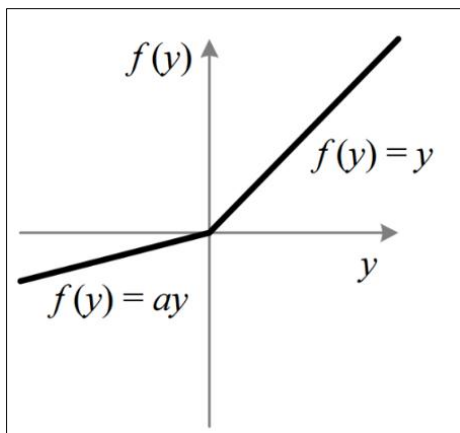
$$f(x) = ax, x < 0 \quad (6.25)$$

Sızdıran ReLU fonksiyonu ve türevi Şekil 6.22’de verilmiştir.



Şekil 6.22. Sızdıran ReLU fonksiyonu ve türevi (Kızırak, 2019)

Parametrik ReLU fonksiyonu ve türevi Şekil 6.23’te verilmiştir.



Şekil 6. 23. Parametrik ReLU (Sharma ve diğerleri, 2017)

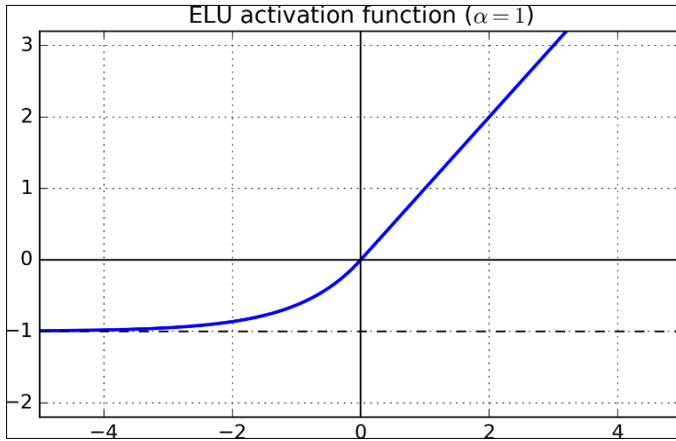
### 6.5.7. Üstel doğrusal birim (Exponential linear unit - ELU)

Üstel Doğrusal Birim veya ELU aynı zamanda Doğrultulmuş Doğrusal Birimin bir çeşididir. ELU,  $x$ 'in negatif değerleri için bir eğim parametresi sunar. Negatif değerleri tanımlamak için bir log eğrisi kullanır (Sharma ve diğerleri, 2017). Şu şekilde ifade edilirler:

$$f(x) = x, x \geq 0 \quad (6.26)$$

$$f(x) = a(ex - 1), x < 0 \quad (6.27)$$

ELU aktivasyon fonksiyonu grafiği şekil 6.24'te verilmiştir.



Şekil 6.24. ELU aktivasyon fonksiyonu grafiği (Sharma ve diğerleri, 2017)

### 6.5.8. Swish (A self-gated/kendinden geçitli) fonksiyonu

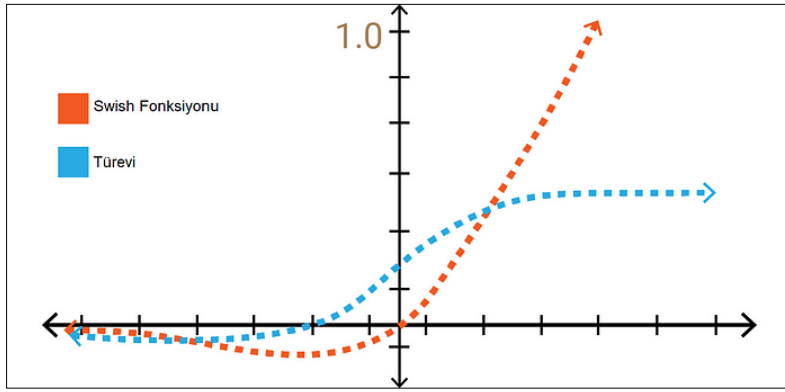
Swish fonksiyonu Google'daki araştırmacılar tarafından keşfedilen nispeten yeni bir aktivasyon fonksiyonudur. Swish fonksiyonunun ayırt edici özelliği nit Monotonik olmasıdır, yani girdilerin değerleri artıyor olsa bile fonksiyonun değeri azalabilir. Bazı durumlarda Swish, ReLU fonksiyonundan bile daha iyi performans gösterir (Sharma ve diğerleri, 2017).

Matematiksel olarak şu şekilde ifade edilir:

$$f(x) = x * \text{sigmoid}(x) \quad (6.28)$$

$$f(x) = x/(1 - e^{-x}) \quad (6.29)$$

Swish fonksiyonu ve türevi Şekil 6.25'te verilmiştir.



Şekil 6.25. Swish fonksiyonu ve türevi (Kızrak, 2019)

### 6.5.9. Softmax

Softmax aktivasyon fonksiyonu sigmoid fonksiyonunun bir uzantısıdır. Bu 0'dan 1'e aktivasyon fonksiyonu, çok sınıflı sınıflandırma problemlerinde yaygın olarak kullanılmaktadır (Nwankpa ve diğerleri, 2018). Yaygın olarak kullanılan evrişimli sinir ağı modellerinin son sınıflandırma katmanı, softmax fonksiyonudur (Kılıç, 2021). Softmax matematiksel ifadesi şu şekildedir:

$$\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ for } j=1, \dots, K \quad (6.30)$$

$\text{softmax}(\mathbf{z})_i$ ,  $i$ -inci sınıfın softmax çıkışını temsil eder.

$e$  Euler sayısı (yaklaşık olarak 2.71828).

$K$  toplam sınıf sayısını temsil etmektedir (Sharma ve diğerleri, 2017).

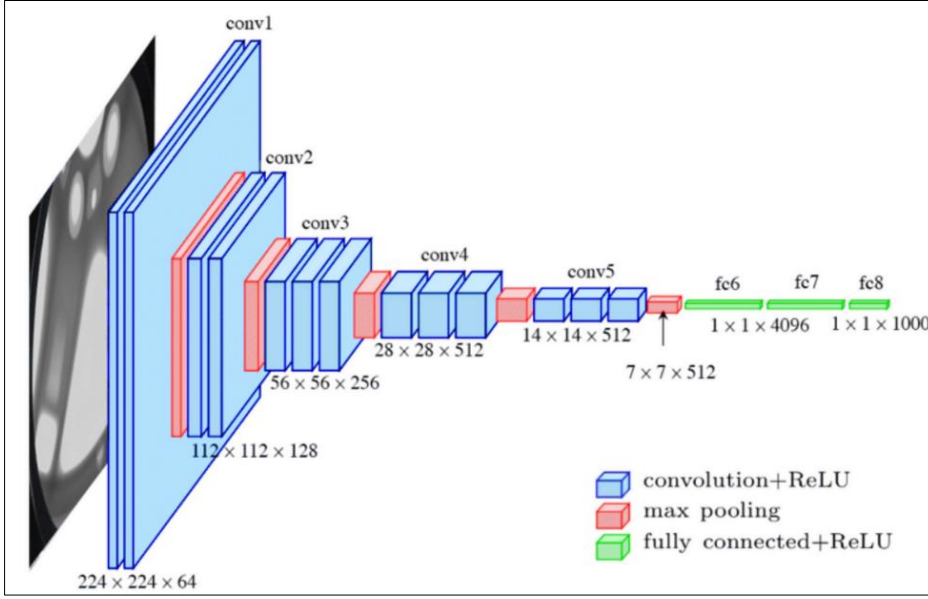
## 6.6. Evrişimsel Sinir Ağı Mimarileri

### 6.5.1. VGGNet

VGGNet mimarisi, 2014 yılında Oxford Üniversitesi'nden K. Simonyan ve A. Zisserman tarafından yayınlanan "Büyük Ölçekli Görüntü Tanıma için Çok Derin Evrişimli Ağlar" adlı makalede tanıtılmıştır ve adını Visual Geometry Group (Görsel Geometri Grubu) isimli araştırma grubundan almıştır. VGGNet, ImageNet yarışmasında kullanılan 11, 16 ve 19 katmanlı olmak üzere üç farklı versiyona sahiptir ve bu mimariler 2014 yılı ImageNet yarışmasında %7,3'lük bir hata oranı elde etmeyi başarmıştır. VGGNet'in diğer benzer ağlardan farkı, ağın katmanlarını çoğaltmak yerine derinleştirmeye odaklanmasıdır. Bu özellik, görüntü sınıflandırma problemlerinde etkin bir şekilde kullanılmasını sağlar. Özellikle, 16 katmanlı versiyon 13 evrişim katmanı ve 3 tam bağlı katman içermektedir. Bu yapı, özellik çıkarımı için geniş bir perspektif sağlamak amacıyla tasarlanmıştır (Kılıç, 2021).

VGGNet mimarisi, 138 milyon parametre içerir ve AlexNet'e benzemesine rağmen, evrişim katmanlarında filtre boyutları küçültülerek sabit 3x3 filtreler kullanılmıştır. Ağın giriş görüntü boyutu 224x224x3'tür. Ardışık evrişim katmanlarının ardından maksimum havuzlama katmanları eklenmiştir. Aktivasyon fonksiyonu olarak ReLU tercih edilmiştir. Sınıflandırma aşamasında, tam bağlı her katmanda 4096 nöron bulunmaktadır. Son katmanda ise sınıfları ayırmak için Softmax fonksiyonu kullanılmıştır. Bu özelliklerle VGGNet, geniş bir özellik çıkarım kapasitesine sahip ve ImageNet yarışmasında düşük hata oranları elde etmiştir (Simonyan ve Zisserman, 2014).

Bu tez çalışmasında, VGG-16 mimarisi kullanılmıştır toplam 21 katmandan oluşur. Bu katmanların 13'ü evrişimli katmanlar, beşi maksimum havuzlama katmanları ve üçü tamamen bağlı katmanlardır. Bu ağın dikkate değer bir özelliği, tüm katmanların 3x3 piksel uzamsal boyuta sahip olmasıdır. VGG-16'nın mimarisi Şekil 6'da görselleştirilebilir. Giriş için VGG-16 mimarisi, genişlik, yükseklik ve üç renk kanalına sahip bir görüntüyü temsil eden 224x224x3 boyutlarında bir vektör bekler. Ağın çıktısı, görüntünün ait olduğu tahmin edilen sınıfı gösteren 1000 değerli bir vektördür (Simonyan ve Zisserman, 2014). VGG-16 mimarisi şematik gösterimi Şekil 6.26'da verilmiştir.

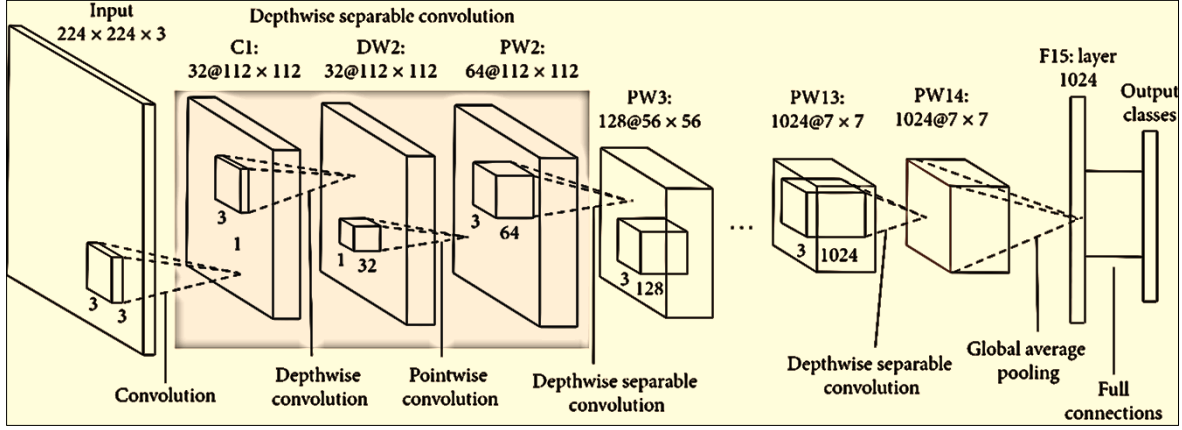


Şekil 6.26. VGG-16 mimarisi şematik gösterimi (Simonyan ve Zisserman, 2014)

### 6.5.2. MobileNet

MobileNet, Google tarafından geliştirilen hafif ve hızlı bir evrişimli ESA mimarisidir. Mobil cihazlar ve düşük güçlü uygulamalar için özel olarak optimize edilmiştir. MobileNet, mobil uygulamalarda gerçek zamanlı nesne tespiti ve sınıflandırma gibi görevlerde kullanılmak üzere özel olarak tasarlanmıştır. MobileNetand'ın bazı önemli özellikleri kolay ve hızlı olması MobileNet, daha küçük model boyutları ve daha az parametre kullanarak hafif bir mimari sağlar. Bu özellik, düşük güçlü cihazlarda ve kaynakların kısıtlı olduğu ortamlarda verimli çalışmasını sağlar. Mobil optimize edilmiş yapısı sayesinde gerçek zamanlı uygulamalarda yüksek performans elde etmek mümkündür. Profil seçenekleri, MobileNet, farklı gereksinimlere uyarlanabilecek çeşitli profiller sunar. Örneğin, MobileNetV1 hafif bir temel modeli temsil ederken, MobileNetV2 ve V3 daha iyi performans ve doğruluk sunar. Derin ayrılabilir evrişim, MobileNet, geleneksel evrişim yerine "Derinlemesine Ayrılabilir Evrişim" teknolojisini kullanır. Bu da yapının daha verimli kullanılmasını sağlar. Bu, daha düşük hesaplama maliyetleri ve daha az parametre anlamına gelir. Aktivasyon fonksiyonları, MobileNetand'ın öğrenimini hızlandırmak için ReLU (Rectified Linear Unit) gibi güçlü aktivasyon fonksiyonları kullanılır. Öğrenme yeteneğinin transferi, MobileNet mimarisi, transfer öğrenme yöntemleri kullanılarak eğitilebilir, böylece önceden eğitilmiş modellerin kullanımı ve belirli uygulamalara uyarlanması daha kolay hale gelir. MobileNet, mobil ve

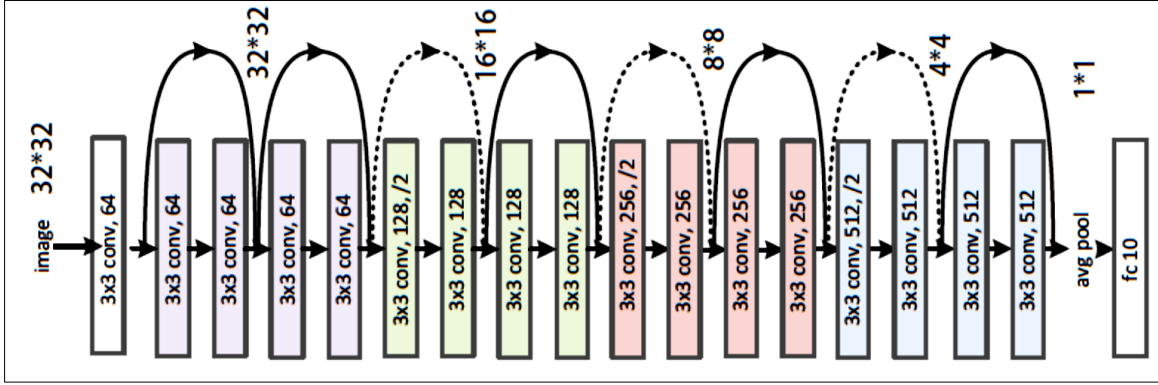
uç cihazlar için popüler, hafif, hızlı ve etkili bir derin öğrenme mimarisi olmaya devam etmektedir (Howard ve diğerleri, 2017). MobileNet mimarisi şematik gösterimi Şekil 6.27’de verilmiştir.



Şekil 6.27. MobileNet mimarisi şematik gösterimi (Wang ve diğerleri, 2020)

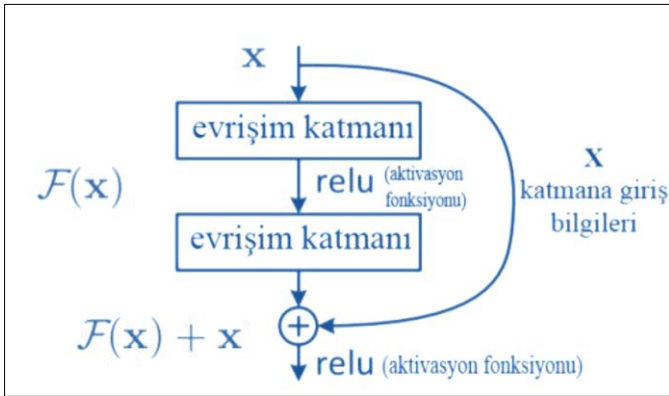
### 6.5.3. ResNet

2015 yılında Kaiming He ve ekibi tarafından geliştirilen ResNet mimarisi, artık (residual) blokları kullanarak ImageNet yarışmasında üstün bir performans sergileyerek yarışmanın birincisi olmuştur. İnsan görüşünün ortalama hata oranı genellikle %5 olarak kabul edilirken, ResNet mimarisi nesnelere ayırt etme görevinde hata oranını %3.57’ye düşürerek dikkat çekici bir başarı elde etmiştir. Bu başarı, özellikle derin sinir ağları alanında önemli bir ilerleme olarak kabul edilmiştir. ResNet’in ana özelliklerinden biri, artık blokları kullanarak ağır ve derin olmasını sağlamak ve eğitim sürecindeki gradyan kaybını önlemektir (He ve diğerleri, 2016). ResNet mimarisi genel şematik gösterimi Şekil 6.28’de verilmiştir.



Şekil 6.28. ResNet mimarisi şematik genel şematik gösterimi (He ve diğerleri, 2016)

ESA mimarilerinde katman sayısını artırmak, bir noktadan sonra hata oranını artırma ve ağırların uyum sorunlarına yol açma eğilimindeydi. Ancak, Kaiming He ve ekibinin 2015 yılında geliştirdiği ResNet mimarisi, bu sorunları artık (residual) bloklarla başarılı bir şekilde çözmüştür. ResNet, kendisinden önceki mimarilere kıyasla çok daha fazla katmana sahiptir, ancak bu artışın aksine performansı artırmıştır. Artık bloklar, ağırların daha derin olmasını mümkün kılar ve aynı zamanda gradyan kaybı sorununu ele alarak eğitim sürecini stabilize eder. Bu bloklar, her katmandan geçen bir tür geçiş yolunu içerir, böylece bilgi ağı boyunca daha iyi akar. Bu, ağırların daha derin ve karmaşık özellikleri öğrenmesine olanak tanıırken ağırların uyumu önler. Sonuç olarak, ResNet mimarisi, evrimsel sinir ağı alanında derin ağların kullanımını daha etkili hale getirerek, daha yüksek doğruluk oranlarına ve daha düşük hata oranlarına ulaşmayı mümkün kılmıştır. Artık bloklar, derin öğrenme modellerinin inşasında ve eğitiminde önemli bir paradigma değişikliği sağlamıştır (He ve diğerleri, 2016). Residual blok yapısı Şekil 6.29’da verilmiştir.

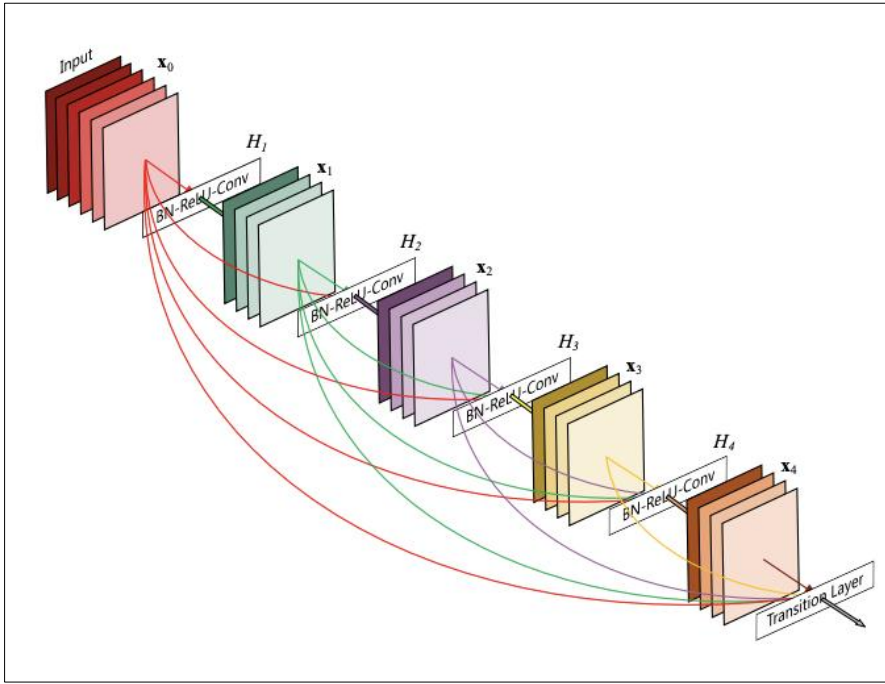


Şekil 6.29. Residual blok yapısı (He ve diğerleri, 2016)

ResNet mimarisinin 18-34-50-101-152 katmanlı versiyonları görüntü sınıflandırma, nesne tanıma problemlerinde oldukça sık kullanılmaktadır. Bu çalışmada 50 katmanlı Resnet-50 mimarisi kullanılmıştır.

#### 6.5.4. DenseNet

Son yıllarda sıkça kullanılan DenseNet mimarisi, ESA mimarilerinin derinleştirilmesi çabaları sonucunda geliştirilmiştir. Bu mimari, 2016 yılında Gao Huang ve ekibi tarafından "Densely Connected Convolutional Networks" adlı makalede tanıtılmıştır. DenseNet, önceki mimarilerden farklı bir yaklaşım benimseyerek yoğun bağlantılar kullanmış ve böylece ağın daha etkili öğrenme yetenekleri elde etmesini sağlamıştır (Huang ve diğerleri, 2017). DenseNet mimarisi şematik gösterimi Şekil 6.30'da verilmiştir.



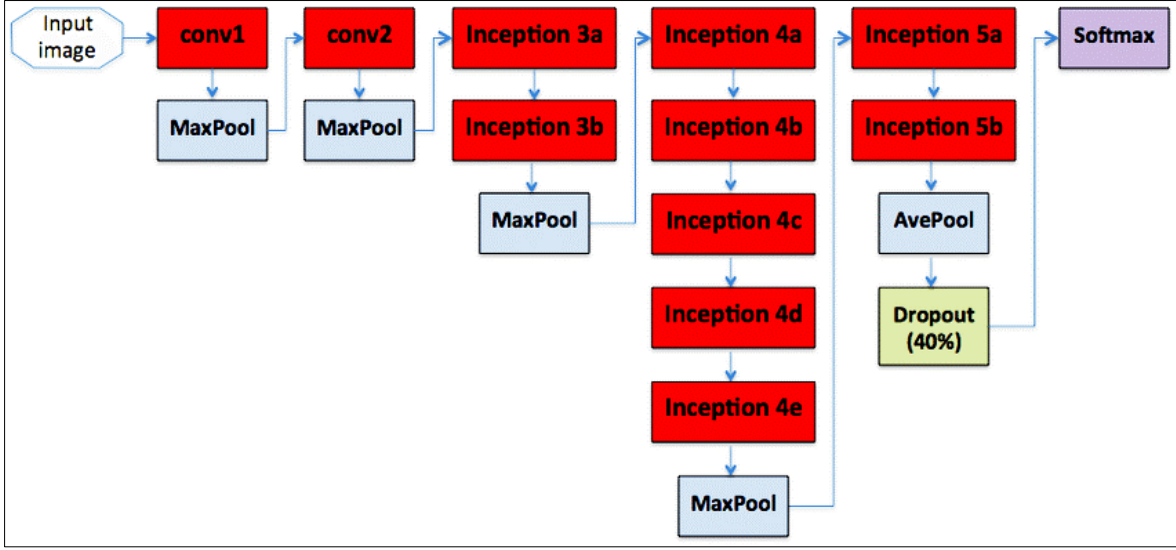
Şekil 6.30. DenseNet mimarisi şematik gösterimi (Huang ve diğerleri, 2017)

Geleneksel ESA mimarileri her evrişim katmanından çıkan bilgileri bir sonraki evrişim katmanına bağlayarak ilerlerken, DenseNet mimarisinde her katmanın çıktıları doğrudan kendisinden sonraki katmanlara bağlanır. Bu işlem, her katmanın çıktılarını birleştirmeyi mümkün kılar. Farklı katmanlardan gelen özellik haritalarının birleştirilmesi, sonraki katmanların girdilerinde çeşitliliği artırır ve ağın daha verimli öğrenmesine olanak tanır. DenseNet, bu özellikleriyle önceki ResNet ve Inception ağlarına göre daha basit ve üstün

bir mimari sunar. DenseNet'in önemli avantajları şunlardır: gradyan kaybını azaltır, özellik yayılımını güçlendirir, özellikleri daha etkili bir şekilde yeniden kullanabilir ve parametre sayısını önemli ölçüde azaltır. DenseNet mimarisinin yaygın olarak kullanılan 121, 169 ve 201 katmanlı versiyonları bulunmaktadır. Ağın giriş görüntü boyutu 224x224'dir. Yoğun bloklarda 1x1 ve 3x3 evrişim filtreleri kullanılır. İlk katmanda maksimum havuzlama için 3x3 boyutunda bir filtre kullanılırken, geçiş katmanlarında ortalama havuzlama için 2x2 boyutunda bir filtre kullanılmıştır. Tüm katmanlarda adım sayısı sabit olarak 2 olarak belirlenmiştir (Huang ve diğerleri, 2017).

### 6.5.5. Inception

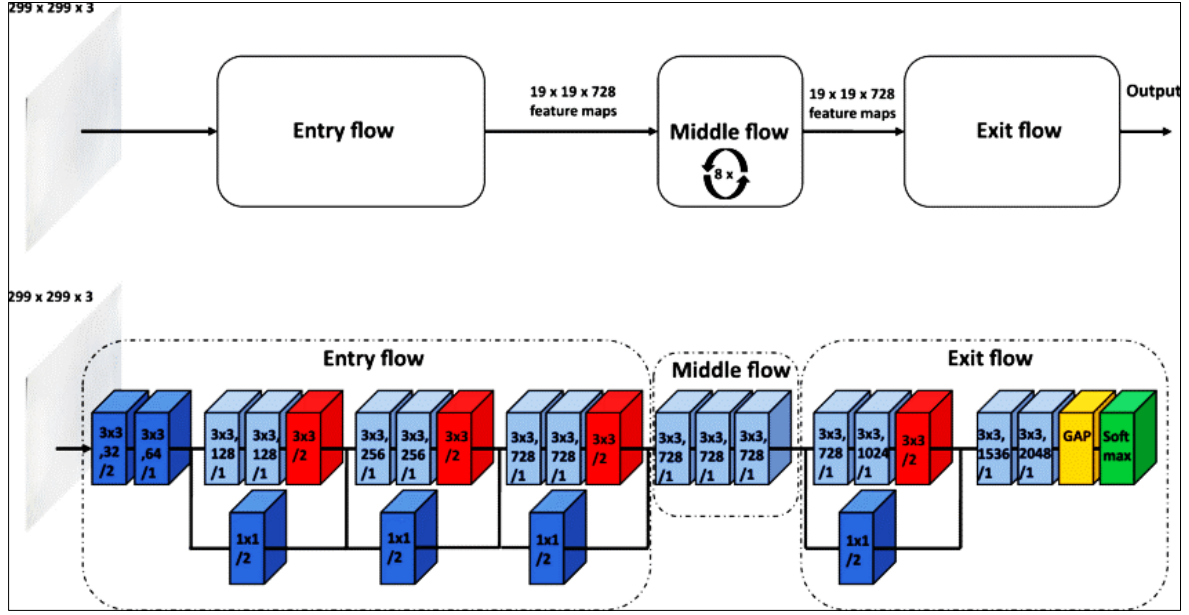
2014 yılında Google'da çalışan araştırmacılar tarafından geliştirilen bu mimari, ImageNet yarışmasında %6,7'lik bir hata oranı ile büyük bir başarı elde etmiş ve yarışmayı birinci olarak tamamlamıştır. Bu mimari, GoogleNet olarak da bilinmektedir ve AlexNet'e kıyasla daha fazla katman içermesine rağmen, parametre sayısını 12 kat daha az tutmayı başarmıştır. GoogleNet mimarisi, toplamda 27 katmandan oluşur ve bu katmanlar, bazıları Inception (başlangıç) modülü içeren bir yapıdadır. Ağın giriş görüntü boyutu 224x224 piksel olup, bu mimaride 1x1 boyutunda filtreler kullanılmıştır. Bu, işlem yükünü azaltmayı ve daha derin özelliklere odaklanmayı hedefler. Inception modülleri, 1x1 filtrelerle elde edilen görüntüleri, modül içinde daha büyük filtrelerden geçirerek çeşitli anlamları çıkarmayı amaçlar. Bu modüller, farklı filtreler aracılığıyla çeşitli ölçeklerde özellikleri algılar ve boyutsal küçültme kullanarak geniş bir ağı daha verimli bir şekilde eğitir. Evrişim katmanlarında maksimum havuzlama uygulanırken, tam bağlı katmandan önce ortalama havuzlama katmanı bulunur. ReLU aktivasyon fonksiyonu kullanılırken, sınıflandırma için softmax fonksiyonu tercih edilmektedir. Bu mimari, yüksek doğruluk oranlarına ulaşırken düşük parametre sayısı ile etkili bir şekilde çalışmasıyla dikkat çekmektedir (Szegedy ve diğerleri, 2015). Inception mimarisi şematik gösterimi Şekil 6.31'de verilmiştir. Tüm evrişimli katmanlar ve başlangıç modülleri iki derinliğe sahiptir.



Şekil 6.31. Inception mimarisi şematik gösterimi (Szegedy ve diğerleri, 2015)

### 6.5.6. Xception

Xception, Inception-V3 mimarisine dayanan bir ESA modelidir. Zaman ve mekan karmaşıklığını azaltmak için çeşitli iyileştirmeler sunar. Model, derinlik açısından ayrılabilir evrişim katmanlarından oluşan doğrusal bir yığın kullanır ve artık bağlantıları içerir. Xception'ın temel özelliği, kanal tabanlı ve uzamsal tabanlı özelliklerin öğrenilmesini ayıran derinlemesine ayrılabilir evrişimdir. Bu, hesaplama karmaşıklığını azaltırken karmaşık kalıpları yakalamaya yardımcı olur. Ek olarak, Xception, ağ içinde kısayollar oluşturarak kaybolan gradyanlar ve temsili darboğazlar sorunlarını ele almak için artık bağlantı kullanır. Genel olarak, Xception, Inception-V3'e kıyasla gelişmiş verimlilik ve performans sunarak onu çeşitli bilgisayarla görme görevleri için popüler bir seçim haline getirir (He ve diğerleri, 2016) Xception mimarisi şematik gösterimi Şekil 6.32'de verilmiştir.



Şekil 6.32. Xception mimarisi şematik gösterimi (Westphal ve Seitz, 2021)

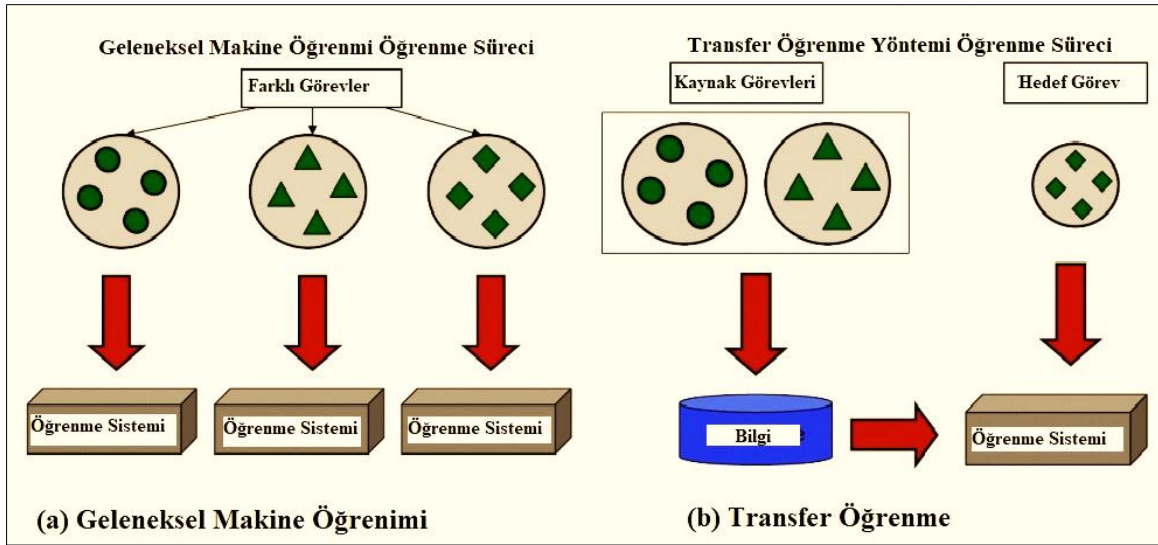
## 6.6. Transfer Öğrenme

ESA modelleri görüntü sınıflandırma problemlerinde gösterdiği yüksek başarı sayesinde giderek daha fazla önem kazanmıştır. Görüntü sınıflandırmada genellikle iki temel yöntem kullanılmaktadır. İlk yöntemde, belirli bir problem için sıfırdan yeni bir model oluşturulurken, ikinci yöntemde ise daha önce geliştirilmiş, başarısı kanıtlanmış ve yaygın olarak kullanılan bir model ilgili probleme uyarlanır. Sıfırdan bir model geliştirme sürecinde, girdi ve çıktı verilerinin özellikleri analiz edilir ve bu veriler arasında dönüşüm sağlayacak eğitim aşamaları tasarlanır. Daha sonra farklı yaklaşımlar deneyerek modelin performansı artırılır ve en iyi sonuçlar elde edilmeye çalışılır. Öte yandan, ikinci yöntemde, önceden eğitilmiş bir modelin giriş ve çıkış katmanları ilgili probleme göre düzenlenerek model uygulanmaktadır. Ardından, verilen probleme uygun şekilde modelin optimize edilmesi ve hassas ayarlamalar yapılması sağlanır. Bu ikinci yönteme transfer öğrenme adı verilmektedir (Pan, 2016; Fırıldak ve Talu, 2019).

Derin öğrenme modellerini sıfırdan eğitmek ve geliştirmek, bazı durumlarda olumlu sonuçlar ortaya koyabilir. Ancak, bu süreç için geniş miktarda veri gerekmektedir. Yetersiz veri olması durumunda, model yeterli öğrenme sağlayamayabilir ve bu da modelin tahmin doğruluğunu olumsuz yönde etkileyebilmektedir. Ayrıca, sıfırdan yeni model eğitimi, transfer öğrenme yöntemine kıyasla çok daha uzun bir süreç gerektirebilir (Kılıç, 2021).

ImageNet'in 1000 sınıflı görüntü sınıflandırma yarışmasında kullanılmış olan ve bir milyondan daha fazla görüntüyle eğitilmiş birçok ESA modeli bulunmaktadır. Bu durum, özellikle görüntü sınıflandırma ve nesne tanıma sorunlarında transfer öğrenme yöntemlerini son derece popüler hale gelmesine olanak sağlamıştır. Transfer öğrenme süreci, önceden eğitilmiş büyük bir veri setindeki bilgilerin, daha küçük veya özelleşmiş bir veri seti için modelin başlangıç noktası olarak kullanılması anlamına gelir. Bu sayede, modelin yeni görevde daha iyi performans sergileme olasılığı artar (Kılıç, 2021).

Bu çalışmada, seçilen ESA'larının altı farklı mimarisi, beş farklı optimizasyon algoritması ve dört farklı nöron sayısı (256, 512, 1024, 2048) test edildi. ESA mimarileriyle uyumlu en iyi optimizasyon algoritmasını belirlemek için 256 nöronla 30 (6×5) deney gerçekleştirildi. Daha sonra en iyi sonucu veren optimizasyon algoritması belirlendikten sonra 256, 512, 1024 ve 2048 nöron kullanılarak 24 (6×4) deney gerçekleştirildi. Toplamda 54 deney gerçekleştirilmiştir. Transfer öğrenme süreci Şekil 6.33'de verilmiştir.



Şekil 6.33. Transfer öğrenme süreci

Ahşap kusur tespiti için VGGNet, ResNet50, MobileNet, DenseNet121, Xception ve InceptionV3 olmak üzere altı farklı, iyi bilinen ESA mimarisi görüntüler kullanılarak eğitilmiştir. Tüm mimariler aynı hiper parametrelerle eğitilmiştir. Eğitimi sıfırdan başlatmak yerine transfer öğrenme yöntemi tercih edilmiştir. Bu yöntem ile aşırı öğrenmenin önlenmesi, zamandan tasarruf edilmesi ve doğruluğun artırılması amaçlanmıştır. Ağın ağırlıkları için ImageNet ağırlıkları kullanılmıştır.

Bilinen transfer öğrenme mimarileri, özellik çıkarıcı katmanlara ve ek bir sınıflandırıcı softmax katmanına sahiptir. Ağ içinde aktivasyon için Adam aktivasyon fonksiyonu kullanılmıştır. Deneylerde, bu mimarilerin softmax katmanına öncelikle toplu normalizasyon katmanı, tam bağlantılı katman ve %0,25 oranında dropout katmanları eklenmiştir. Bunun amacı, aşırı öğrenmeyi önlemektir.

Karşılaştırmalı ve kapsamlı bir analiz sağlamak amacıyla deneylerde Adam, RMSprop, Adadelat, Adagrad ve SGD gibi farklı optimizasyon algoritmaları kullanılmıştır. Aynı zamanda, en başarılı optimizasyon algoritması belirlendikten sonra, her bir mimariye 256, 512, 1024 ve 2048 nöron eklenerek performansları ayrı ayrı ölçülmüştür.

Ağların başlangıçtaki öğrenme oranı standart olarak 0.0001 olarak belirlenmiştir. Her eğitim döngüsünde doğrulama kaybı kontrol edilmiş ve eğer kayıp 5 epoch boyunca azalmadıysa, öğrenme oranı %50 oranında düşürülmüştür. Böylece dinamik öğrenme oranı kullanılması hedeflenmiştir. Çalışmada çok sayıda deney ve karşılaştırmalı analiz bulunduğundan, eğitim süresi dikkate alınarak epoch sayısı 20 olarak belirlenmiştir. RAM, grafik kartı kapasitesi ve veri seti boyutu göz önünde bulundurularak batch size 64 olarak seçilmiştir.



## 7. MATERYAL VE YÖNTEM

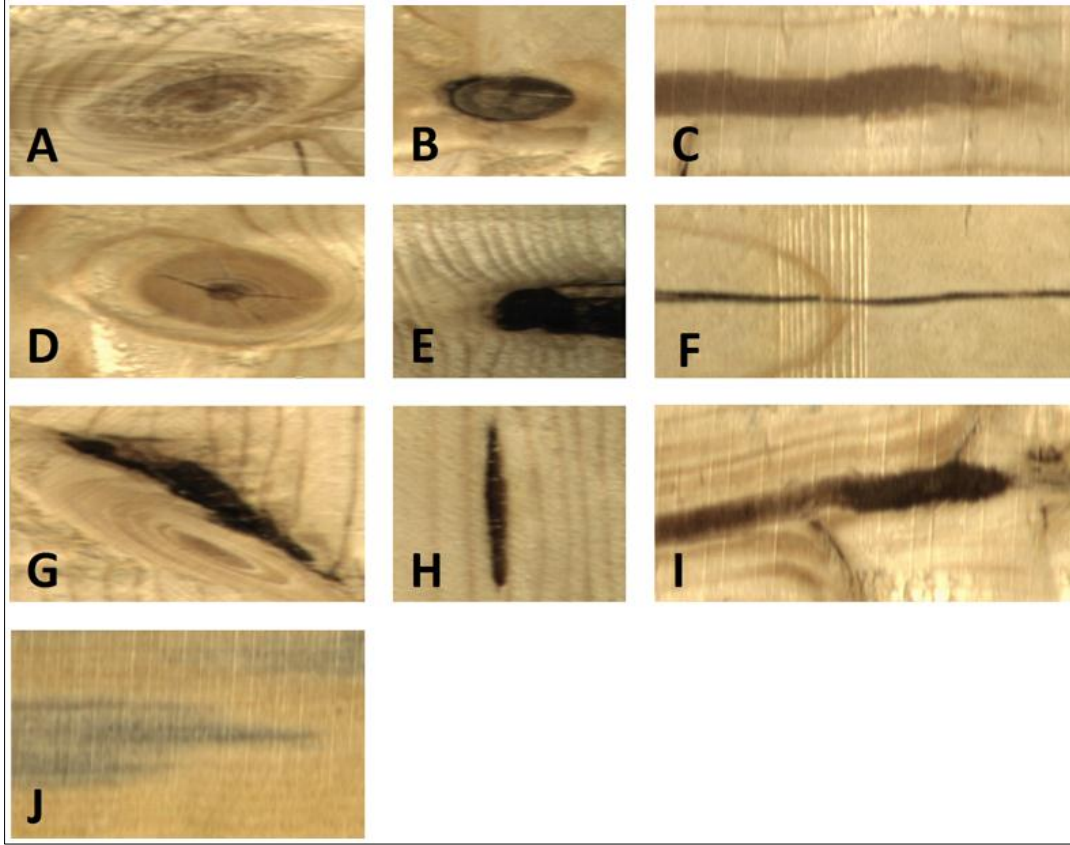
Bu başlı altında tezde kullanılacak tüm yöntem teknik ve materyaller verilecektir. Ana başlıkları şu şekildedir. Tez kapsamında klasik makine öğrenme, evrişimli sinir ağları ile transfer öğrenme ve hibrit öğrenme; evrişimli sinir ağları ile öznelik çıkarımı yapıp sonrasında klasik makine öğrenme ile sınıflandırma yapılarak ahşap kusurları tespit edilmeye çalışılmıştır.

### 7.1. Veri Seti

Ağaçşleri endüstrisi mühendisliği alanındaki en kapsamlı veri tabanlarının eksikliğini gidermek amacıyla hazırlanan bir veri seti bu tezde kullanılmıştır. Kodytek ve diğerleri (2021) tarafından ahşap yüzey kusurlarının büyük ölçekli bir veri kümesini elde etmek amacıyla oluşturulmuştur. Bu amaçla, yapılan diğer çalışmalardan farklı olarak, gerçek üretim sırasında endüstriyel bir ortama yerleştirildi ve bu, üretim hattından büyük miktarda gerçek veri elde edilmiştir. Konveyör bandının yüksek hızı ve ağır titreşimler gibi üretim sürecinden kaynaklanan zorluklarla yüzleşmek için, 66 KHz'lik alım hızında yüksek çözünürlüklü görüntülerin alınmasını sağlayan bir yazılım çözümünün yanı sıra bir donanım tasarlanmıştır. Bu deneyde, biçilmiş kereste yüzeyinin 20 276 orijinal veri örneğini elde edilmiştir. Bu örnek görüntülerin 1992 görüntüde herhangi bir yüzey kusuru yoktur ve 18 284 görüntüde toplam 10 yaygın ahşap yüzey kusurunu kapsayan bir veya daha fazla kusur tespit edilmiştir. En sık görülen kusurlar, veri setinde sırasıyla %58,8 ve %41,2'lik bir genel oluşumla kaynamış budak ve düşen budakları içermektedir (Kodytek ve diğerleri, 2021). Ağaçşleri endüstrisinde rastlanabilecek ahşap kusurlarının hepsi bu veri seti içerisinde endüstriyel hayatta olduğundan bu veri seti le çalışmaya yapılamaya karar verilmiştir. Veri kümesindeki ahşap kusurların genel örnekleri Resim 7.1'de verilmiştir.

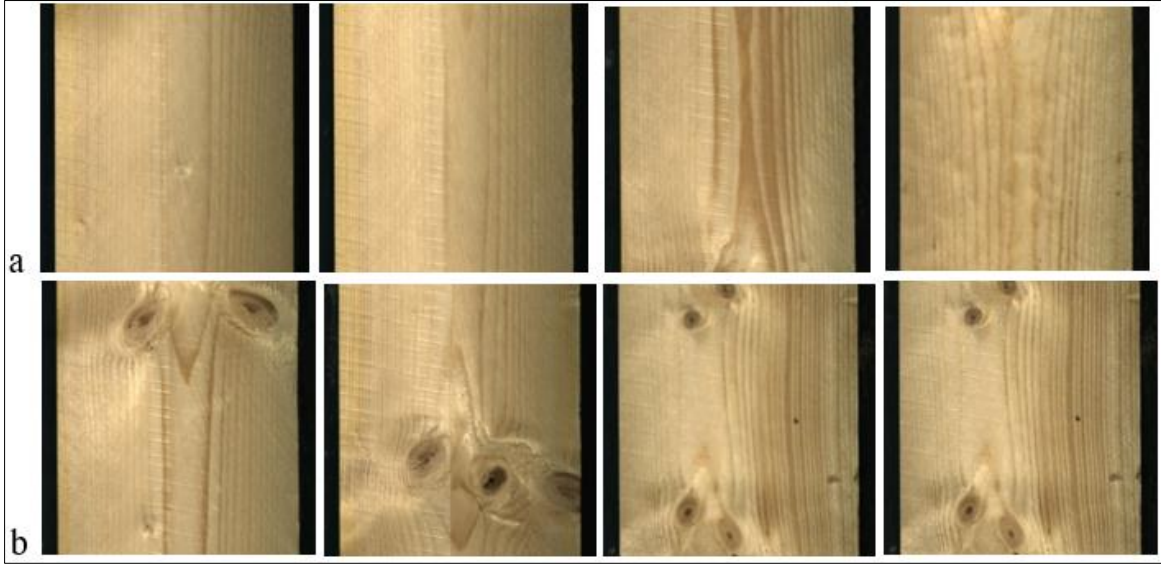
Ağaçşleri endüstride otomasyon artsa da, birçok lider şirket hâlâ kalite derecelendirme ve kusur tespiti için uzmanlara güvenmektedir. Ancak manuel inceleme sıkıcı, önyargılı ve büyük üretim hacimlerini kontrol etmekte yetersizdir Göz yorgunluğu ve dikkat dağınıklığı nedeniyle manuel incelemenin güvenilirliği %70'i nadiren aşmaktadır. Bu sınırlamaları aşmak için araştırmacılar, üretimi yavaşlatmadan daha doğru sonuçlar sağlayan otomatik

sistemler geliřtirmektedir (Urbonas ve diđerleri, 2019). Otomatik sistemlerin insanlardan daha tutarlı olduđunu göstermiřtir (Lycken, 2006). Geleneksel goruntu iřleme teknikleri yaygın olsa da, son yıllarda derin ğrenme ormancılık ve ahřap rnleri endstrisinde byk bařarı elde etmiřtir (Liu ve diđerleri, 2018).



Resim 7.1. Veri kmesindeki ahřap kusurlarının tipik rnekleri: (A) Kaynamıř Budak, (B) Dřen Budak, (C) Ur, (D) atlak Budak, (E) Eksik Budak, (F) atlak, (G) Ařırı Byme, (H) Reine, (I) z (J) Mavi Leke (Kodytek ve diđerleri, 2021)

Veri seti ierisinde yer alan kusurlu ahřap ve kusursuz ahřap yzey rnekleri Resim 7.2’de verilmiřtir.



Resim 7.2. Kusursuz ahşap yüzeyleri (a), kusurlu ahşap yüzeyleri (b)

Ahşap, doğal olarak heterojen ve anizotrop bir malzeme olup, bu özellikler kusur tespiti ve sınıflandırma süreçlerinde önemli zorluklar yaratmaktadır. Heterojenlik, farklı ahşap türleri ve aynı tür içindeki değişkenlikler nedeniyle malzeme bileşiminin düzensiz olmasını ifade eder. Yoğunluk, gözeneklilik ve lif yapısı gibi faktörler, ahşabın fiziksel ve optik özelliklerini doğrudan etkileyerek tutarlı özellikler belirlemeyi zorlaştırır. Anizotropi ise ahşabın farklı doğrultularda değişen mekanik ve optik özellikler sergilemesi anlamına gelir. Lif yönüne bağlı olarak doku, renk ve sertlik değişiklik gösterdiğinden, kusur tespitinde standart bir model geliştirmek güçleşmektedir. Bu nedenle, makine ve derin öğrenme tabanlı yaklaşımlar, geniş ve çeşitlendirilmiş veri kümeleri kullanarak bu değişkenlikleri yönetmek zorundadır.

## 7.2. Uygulama Ortamı ve Kütüphaneler

Son 20 yılda derin öğrenme uygulamalarının popülerliğindeki bir neden, gelişmiş donanımın sağladığı olanaklardır. Derin öğrenme modellerinin performansı genellikle büyük miktarda veri ile ilişkilidir. Büyük veri setlerini işlemek için yüksek kapasiteli işlemci ve grafik kartlarına ihtiyaç vardır. Özellikle derin öğrenme tabanlı görüntü sınıflandırma gibi görevlerde, gelişmiş grafik kartlarına olan ihtiyaç önemlidir. Ancak, bu yüksek kapasiteli donanımın maliyeti sınırlayıcı olabilir. Bu maliyet sınırlamalarını aşmak için, yapay zeka tabanlı uygulamalar geliştirmek isteyenler için çeşitli bulut tabanlı hizmetler sunulmaktadır. Bu hizmetler, kullanıcılara güçlü ve ölçeklenebilir bilgi işlem

kaynaklarına erişim sağlar. Bu sayede, kullanıcılar kendi yerel cihazlarında yeterli donanımına sahip olmasalar bile bulut üzerinden gelişmiş bilgi işlem gücünü kullanabilirler. Bu, derin öğrenme modellerini eğitmek ve uygulamak için gereken yüksek hesaplama gücüne erişim sağlayarak, maliyeti düşürüp daha geniş bir kullanıcı kitlesine hitap etmeyi mümkün kılar. Bu tez çalışmasında ahşap görüntüleri üzerinde yapılan ön işlemler İntel 3.nesil (İ5-3470) 3.20 GHz Intel işlemci, 4gb AMD Radeon R7 240 ekran kartı ve 16 GB DDR3 RAM boyutuna sahip masaüstü bilgisayar kullanılarak gerçekleştirilmiştir.

Kaggle, veri bilimi ve makine öğrenimi alanında projeler geliştirmek için kullanılan bir platformdur. Kaggle, kullanıcılara çeşitli veri setlerine erişim, bilgi ve deneyim paylaşımı, yarışmalara katılım ve farklı programlama dilleri üzerinde çalışma imkanı sunmaktadır. Kaggle'in uygulama ortamı, kullanıcıların projelerini geliştirmeleri ve yarışmalara katılmaları için çeşitli araçlar ve kaynaklar sunan bir platformdur. Ön işlemde geçirilen ahşap yüzey görüntüleri Kaggle, genellikle NVIDIA'nın yüksek performanslı GPU'larından Tesla P100, modeli kullanılmıştır. Tesla P100, NVIDIA'nın Pascal mimarisini temel alır. Pascal mimarisi, derin öğrenme ve genel amaçlı GPU hesaplamaları için optimize edilmiş bir mimaridir. GPU, 3584 CUDA çekirdeği ile donatılmıştır. Ön işlemlerin ardından hazırlanan veri seti Kaggle bulut ortamına yüklenmiştir. Ardından klasik makine öğrenme, derin öğrenme ve hibrit yöntemler kullanılarak modellerin performans değerleri incelenmiştir.

### Os (işletim sistemi) kütüphanesi

Python programları içinde işletim sistemi işlemlerini gerçekleştirmek için kullanılır. Bu kütüphane, işletim sistemi bağımsızlığı sağlar ve Python programlarının farklı platformlarda çalışabilmesini mümkün kılar. Os kütüphanesinin temel fonksiyonları arasında dosya ve izin işlemleri, ortam değişkenlerine erişim, sistem çağruları yapma, ve işletim sistemi ile etkileşim bulunur.

### NumPy

NumPy, büyük ve çok boyutlu diziler ile matrisler üzerinde yüksek performanslı matematiksel işlemleri kolaylaştıran bir kütüphane olma özelliğindedir. Bu kütüphane, temelde numpy dizilerinden oluşur ve Python'daki liste yapısına benzerdir. Ancak,

Python listelerinden farklı olarak numpy dizilerindeki verilerin hepsinin aynı türde olması gerekmektedir. NumPy dizileri, daha hızlı ve işlevsel olmalarıyla dikkat çekmektedir.

### Pandas

Pandas, Python programlama dilinde geliştirilmiş ve veri analizi ile manipülasyonu sağlayan açık kaynaklı bir kütüphanedir. Bu çalışma, Pandas'ın veri bilimi alanındaki önemini ve temel özelliklerini ele almayı amaçlamaktadır. Pandas, tablo benzeri veri yapıları üzerinde etkili bir şekilde çalışmak için geliştirilmiştir ve bu özellikleri sayesinde geniş bir kullanıcı kitlesi tarafından tercih edilmektedir.

### Matplotlib

Matplotlib, Python programlama dilinde yaygın olarak kullanılan bir veri görselleştirme kütüphanesidir. Bilimsel grafikler, çizgi grafikleri, histogramlar, dağılım grafikleri ve daha pek çok görselleştirme türünü oluşturmak için kullanılır. Matplotlib, veriyi anlamak ve sonuçları etkili bir şekilde iletmek için geniş bir grafiksel araçlar yelpazesi sunar.

### Seaborn

Veri görselleştirmesi için geliştirilmiş bir Python kütüphanesidir ve Matplotlib temel alınarak oluşturulmuştur. Seaborn, estetik ve bilgilendirici grafikler oluşturmak amacıyla önceden belirlenmiş renk paletleri ve temalar gibi özelliklere sahiptir. Bu kütüphane, özellikle istatistiksel analiz ve veri keşfi süreçleri için optimize edilmiştir.

### Augmentor

Veri artırma işlemleri için oldukça sık kullanılan bir Python kütüphanesidir. Veri artırma, bir makine öğrenimi modelini eğitirken kullanılan veri kümesini çeşitlendirmek ve zenginlik sağlamak için kullanılan bir yöntemdir. Bu çeşitlendirme, modelin genelleme yeteneğini artırabilir ve overfitting'i azaltabilir. Augmentor, özellikle bilgisayarlı görü işleme ve derin öğrenme projelerinde kullanılmak üzere tasarlanmıştır. Bu kütüphane, özellikle görüntü verileri üzerinde çeşitli teknik ve yöntemlerle veri artırma tekniklerini uygulamak için geliştirilmiştir (Bloice ve diğerleri, 2017).

## OpenCV

OpenCV (Open Source Computer Vision), bilgisayar görüntüsü ve makine görüşü uygulamaları geliştirmek için kullanılan açık kaynaklı bir kütüphanedir. Temel olarak C++ dilinde geliştirilmiş olup, Python, Java, C#, ve MATLAB gibi birçok programlama dilini destekler. OpenCV'nin temel özellikleri: Görüntü İşleme, nesne algılama ve tanıma, kamera kalibrasyonu, stereo görüntü işleme, görüntü yakalama ve video işleme, derin öğrenme entegrasyonu, görüntü segmentasyonu, geniş platform desteği bulunmaktadır. OpenCV, geniş bir topluluk tarafından desteklenir ve sürekli olarak geliştirilmektedir.

## Scikit-learn

Python programlama dilinde kullanılan bir makine öğrenimi kütüphanesidir. Geniş bir makine öğrenimi algoritmaları yelpazesi, veri madenciliği araçları ve önceden işlenmiş veri setleri sunarak kullanıcılarına kolay bir şekilde makine öğrenimi modelleri oluşturma ve değerlendirme imkanı sağlar. Temel özellikleri arasında sınıflandırma, regresyon, kümeleme, boyut azaltma, model seçimi ve doğrulama algoritmaları bulunur. Ayrıca, veri ön işleme araçları, model seçimi ve doğrulama için araçlar, önceden işlenmiş veri setleri ve paralel işleme desteği gibi avantajlar sunar. Scikit-learn, Python diline tam entegre olmuş ve diğer veri bilimi kütüphaneleriyle uyumlu çalışabilen bir yapıya sahiptir, bu da kullanıcılarına geniş bir kullanım esnekliği sağlamaktadır.

## TensorFlow

Google tarafından geliştirilen ve özellikle makine öğrenimi ve derin öğrenme alanında kullanılan bir açık kaynaklı kütüphanedir. Büyük ölçekli sayısal hesaplamaları ve karmaşık sinir ağı modellerini oluşturmayı amaçlar. TensorFlow, graf tabanlı bir hesaplama modelini benimser, bu da matematiksel işlemleri bir grafik üzerinde temsil ederek optimize etmeyi mümkün kılar. Bilgisayar görüntüsü, doğal dil işleme, ses tanıma gibi geniş bir uygulama yelpazesine hitap eden TensorFlow, özellikle derin öğrenme alanında güçlü bir araç olarak öne çıkmaktadır.

## Keras

Yüksek seviyeli bir derin öğrenme API'sidir ve başlangıçta Theano ve TensorFlow gibi backend motorlarını desteklemekteydi. Ancak, TensorFlow 2.0 sürümünden itibaren TensorFlow'un resmi bir yüksek seviyeli API'si haline gelmiştir. Keras, kullanıcıların hızlı ve kolay bir şekilde derin öğrenme modelleri oluşturmasını sağlar. Modüler yapısı sayesinde farklı modelleri basit veya karmaşık bir şekilde inşa etmek mümkündür ve aynı kodun farklı backend motorları üzerinde çalışabilmesine olanak tanır. Keras, özellikle başlangıç düzey ve orta düzey kullanıcılar için etkili bir araç olarak öne çıkmaktadır.

### **7.3. Yöntem**

Tez çalışmasında ahşap yüzeylerinde kusur tespit etmeye yönelik ikili sınıflandırma (binary classification) ile kusurlu ahşap ve kusursuz ahşap şeklinde sınıflandırma gerçekleştirilmiştir. Bu sınıflandırma işlemleri öncelikle transfer öğrenme yöntemi ile evrişimli sinir ağları vasıtasıyla derin öğrenme yöntemleriyle ikili sınıflandırma ya tabi tutulmuştur. İkinci olarak görüntülerden farklı öznelik çıkarımı yapılarak, klasik makine öğrenme algoritmaları ile sınıflandırma yapılmıştır. Üçüncü olarak ise ESA ile öznelik çıkarımı yapılarak makine öğrenme algoritmaları ile sınıflandırma yapılarak ahşap yüzey kusur tespiti yapılmıştır. İlk işlem olarak kusurlu ve kusursuz ahşap yüzey görüntüleri iki klasöre ayrılmıştır. 1 992 kusursuz ahşap yüzey görüntüsü, 18 284 kusurlu ahşap yüzey görüntüsü bulunmaktadır. Kusurlu ahşap yüzeylerinde bir ya da birden çok ahşap yüzey kusur bulunmaktadır. Ancak üretim bantlarında kusursuz ahşap üretime alınır bir ya da daha fazla sayıda kusura sahip olan ahşap üretime kalite standartları gereği alınmaz. Bu belirtilen gerekçe ile çalışma hızlı karar ve amaca ulaşma adına ikili sınıflandırma ile ahşap yüzey kusurları tespit edilmesi amaçlanmıştır.

#### **7.3.1. Veri artırma ve ön işlemler**

Ahşap görüntüler 2800 x 1024 piksel bmp formatından 300 x 300 bmp yeniden boyutlandırılmıştır. Boyutlandırma işlemi arkasından veri artırma işlemleri yapılarak veri dengeli hale getirilmiştir.

Veri artırma, işlenecek veri sayısını artırmak ve stabilize etmek için veri ekleme işlemidir (Purnama ve diğerleri, 2019). Yüksek oranlı örnekleme ve yetersiz örnekleme yöntemleri, dengesiz veri kümelerinde yaygın olarak kullanılmaktadır. Araştırmalar, veri artırmanın derin öğrenme ağ mimarileri ile gerçekleştirilen sınıflandırma problemlerinde başarıyı artırdığını göstermektedir (Lopez vd., 2017; Ayan ve Ünver, 2018). Veri setinde kusurlu ve kusursuz sınıf dengesizliğinin önlenmesi ve derin öğrenme mimarisinin performansının artırılması için Augmentor kütüphanesi ile veriler artırılmıştır. Bu kütüphanede kullanılan özellikler; yatay ve dikey döndürme, maksimum 355 dereceye kadar açı değişimi, minimum 1,1 ve maksimum 1,5 değerinde yakınlaştırma gibi teknikler uygulanarak veri setindeki görüntü sayısının artırılması, maksimum 0,2 değeriyle rastgele kontrast ve parlaklık geliştirme olarak yapılmıştır.

### 7.3.2. Öznitelik çıkarımı ve seçimi

Özellik çıkarımı (feature extraction), bir veri kümesindeki orijinal ve genellikle yüksek boyutlu verilerden daha anlamlı, düşük boyutlu ve anlam açısından zengin özellikler çıkarma işlemidir. Bu süreç, veri kümesinin karmaşıklığını azaltmayı ve özellikle makine öğrenimi modellerinin performansını iyileştirmeyi amaçlamaktadır. Özellik çıkarma, bilgi kaybını en aza indirerek veri kümesinin temsilini iyileştirmeyi amaçlamaktadır. Görüntü işlemede, kenar algılama, köşe algılama veya renk histogramları gibi nitelikler çıkarılabilir. Metin madenciliğinde, kelime frekansları, belge vektörleştirme teknikleri veya duygu analizi özellik olarak kullanılabilir (Zheng ve Casari, 2018).

Öznitelik seçimi (feature selection), belirli bir kriter veya algoritma kullanarak bir veri kümesindeki en önemli öznitelikleri seçme veya sıralama işlemidir. Bu, öznitelik sayısını azaltarak modelin karmaşıklığını azaltır ve ayrıca gereksiz veya bağıntılı özniteliklerin etkisini azaltır. Öznitelik seçimi, modelin daha iyi genelleştirilmesine ve gereksiz hesaplama maliyetinin azaltılmasına yardımcı olabilir (Zheng ve Casari, 2018).

Bu iki süreç, makine öğrenimi projelerinde veri ön işleme aşamasının kritik bir parçasını oluşturur. Veri kümesinin özellikleri doğru bir şekilde ayıklanıp seçildiğinde, daha etkili ve daha hızlı öğrenme modelleri oluşturmak mümkün olabilmektedir (Zheng ve Casari, 2018).

Çalışmada öznitelik çıkarımında kullanılan parametreler verilmiştir. Bu parametreler ön çalışmalar ve denemeler yapılarak bulunmuştur. Elde edilen başarılı sonuçlar tez kapsamında sunulmuştur.

### LBP öznitelik çıkarımı

LBP özellik çıkarma, belirli bir yarıçap ve nokta sayısı kullanılarak bir görüntü üzerinde gerçekleştirilen bir işlemdir. Görüntü renkli halden gri tonlamalı hale dönüştürülür. İlk olarak, her pikselin etrafında dolaşmak için kullanılan yarıçap belirlenir. Bu yarıçap, her pikseli komşu piksellerle karşılaştırmak için kullanılan mesafedir. Nokta sayısı, her pikselin LBP hesaplamasında kullanılan komşu piksel sayısını belirler ve genellikle nokta sayısının yarıçapın 8 katı ile çarpımı olarak belirlenir.

Spesifik olarak, LBP, belirtilen yarıçap ve her pikselin etrafındaki nokta sayısı kullanılarak hesaplanır ve bir özellik haritası oluşturmak için pikselin etrafındaki komşu piksellerle karşılaştırılır. Bu harita, her pikselin belirli bir desenle kodlandığı bir yapı sunar. Bu LBP özellik haritası daha sonra görüntüden çıkarılan LBP özellik vektörünü oluşturmak için bir histograma dönüştürülür.

LBP özellik ekstraksiyonu genellikle LBP'nin düzenli bir versiyonunu ifade eden "uniform" adı verilen bir yöntem kullanılarak gerçekleştirilir. Bu yöntem, oluşturulan öznitelik vektörlerinin daha düzenli ve anlamlı olmasını sağlar. Tüm bu adımlar, LBP özelliklerini içeren görüntülerin bir temsilini oluşturur. Bu gösterim daha sonra bu özellikleri öğrenmek için makine öğrenimi modellerine girdi olarak verilebilir. LBP ile öznitelik çıkarmada, yarıçap = 1, öznitelik çıkarımı için  $n\_points=8*radius$  değerleri kullanılmış ve 10 adet özellik elde edilmiştir. Özellik sayısı az olduğu için herhangi bir seçim işlemi yapılmamıştır. Tüm özellikler sınıflandırmada kullanılmıştır. 12 farklı klasik makine öğrenmesi algoritmaları ile sınıflandırma yapılmıştır.

### HOG öznitelik çıkarımı

HOG özellik çıkarma parametreleri şu şekildedir: Hücre başına piksel sayısı (pixels\_per\_cell) = (16, 16) ve blok başına hücre sayısı (cells\_per\_block) = (2, 2) olarak belirlenmiştir. Bu parametreler, HOG özellik çıkarma işlemi sırasında görüntünün nasıl

bölüneceğini ve bu bölmelerden elde edilen özellik vektörlerinin nasıl birleştirileceğini tanımlar. Toplamda 10 404 özellik oluşmuştur. Daha sonra ANOVA testini kullanarak (f\_classif) en iyi 10 özelliği seçmek için SelectKBest sınıfı kullanılarak en iyi 10 seçilmiştir. Bu durumun sebebi diğer özellik çıkarımı yapan metotlar ile kıyaslama yapmak ve işlem süresini kısa tutmak adına bu şekilde ayarlanmıştır.

### GLCM öznitelik çıkarımı

Her bir görüntü üzerinde, GLCM ile özellik çıkarımı için skimage.feature modülündeki graycomatrix fonksiyonu kullanılmaktadır. Bu tez çalışmasında; graycomatrix (image\_uint, [5], [0], symmetric=True, normed=True) ifadesi, belirli bir uzaklık vektörü [5] ve açı vektörü [0] (yatay yönde) kullanılarak giriş görüntüsü için GLCM hesaplanmaktadır. Daha sonra, graycoprops fonksiyonu aracılığıyla GLCM'den çeşitli doku özellikleri çıkarılmaktadır: Kontrast, piksel ile komşuları arasındaki yoğunluk farkını ölçer; Benzerlik, piksel yoğunluklarındaki yerel farklılıkları ölçer; Homojenlik, GLCM içindeki elemanların çizgiye ne kadar yakın dağıldığını ölçer; Enerji, piksel yoğunluklarının GLCM içindeki karelerinin toplamını ölçer; Korelasyon, görüntüdeki piksel yoğunluklarının lineer bağımlılığını ölçer. Çıkarılan GLCM özellikleri (kontrast, benzerlik, homojenlik, enerji, korelasyon), features (özellikler) dizisinde depolanmaktadır. İlgili etiketler ("kusurlu" için 1 ve "kusursuz" için 0), target dizisinde saklanmaktadır. Son olarak, özellik ve etiket listeleri NumPy dizilerine dönüştürülerek, ileri işlemler için kullanılmak üzere hazırlanmaktadır. GLCM ile öznitelik çıkarımında 5 özellik çıkmıştır ve öznitelik seçimi öznitelik sayısı az olduğu için yapılmamıştır. Çıkarılan 5 öznitelik ile sınıflandırma algoritmaları ile sınıflandırma yapılmıştır.

### SOBEL öznitelik çıkarımı

Görüntülerden özellik çıkarmak amacıyla kenar tespiti ve gradient özelliklerini kullanılmaktadır. İlk adımda, belirtilen klasördeki görüntü dosyaları sınıflara ayrılarak yüklenir ve renkli görüntüler gri tonlamaya dönüştürülür. Gri tonlamaya dönüştürülen görüntüler üzerinde Sobel filtresi uygulanarak kenar bilgileri elde edilir. X ve Y eksenlerinde tespit edilen kenar bilgileri kullanılarak gradient özellikleri oluşturulur. Gradient büyüklüğü ve yönü hesaplanır. Özellik seçimi adımında ise gradientin yönü üzerinde histogram tabanlı bir özellik kullanılarak, belirli bir aralıkta ve belli bir sayıda bin

içeren bir histogram hesaplanır. Elde edilen özellikler, belirlenen sınıflarla ilişkilendirilerek ilgili dizilere eklenir. Son olarak, elde edilen özellikler ve etiketler NumPy dizilerine dönüştürülerek, ileri işlemler için kullanılmak üzere hazır hale getirilir. Sobel ile öznitelik çıkarımında toplam 9 öznitelik çıkarılmıştır, tüm özellikler sınıflandırma algoritmalarında kullanılmıştır.

### GABOR öznitelik çıkarımı

Gabor filtrelerini kullanarak görüntülerden özellik çıkarma sürecini gerçekleştirir. Gabor filtrelerinin çeşitli parametreleri, özellikle theta (yön), sigma (standart sapma), lambda (dalga uzunluğu) ve gamma (en-boy oranı) belirlenmiştir. Theta değerleri dört farklı açıda ( $0$ ,  $\pi/4$ ,  $2\pi/4$ ,  $3\pi/4$ ) seçilmiş, sigma değerleri ise 1 ve 3 olarak belirlenmiştir. Ayrıca, lambda parametresi  $\pi/4$  aralığında değişen değerler alırken, gamma parametresi iki farklı değeri (0.05 ve 0.5) içermektedir.

Bu parametre kombinasyonlarıyla oluşturulan Gabor filtreleri, farklı yönlere, dalga uzunluklarına ve en-boy oranlarına duyarlılık gösterir. Her bir filtre, görüntü üzerinde belirli desenlerin, kenarların veya tekstürlerin vurgulanmasını sağlar. Her bir görüntü gri tonlamaya dönüştürüldükten sonra, bu Gabor filtreleri uygulanarak filtrelenmiş görüntüler elde edilir. Ardından, her bir filtre uygulandıktan sonra elde edilen görüntülerin ortalama ve standart sapma değerleri hesaplanarak, her Gabor filtresi için iki özellik değeri elde edilir.

Çıkarılan bu özellikler, görüntülerin içerdikleri desen, kenar ve tekstür bilgilerini temsil eder. Bu özellikler, her bir görüntü için oluşturulan özellik vektörlerinde birleştirilir ve bu vektörler, makine öğrenimi modellerinin eğitilmesi veya sınıflandırma görevlerinde kullanılması için uygun formata getirilir. Gabor filtreleri tarafından vurgulanan özellikler, görüntülerin içerdiği bilgileri daha belirgin ve anlamlı hale getirmek adına kullanılır. Gabor ile öznitelik çıkarımında 128 çıkarımı yapılmıştır. Threshold değişkeni, sabit özellikleri çıkartma işleminde kullanılacak eşik değerini belirler. Bu değer, değişkenliği ifade eden bir ölçüdür. Eğer bir özellik, belirlenen eşik değerinden daha düşük bir değişkenlik gösteriyorsa, bu özellik çıkartılır. Bu yöntemle özellikler seçilmiş ve 10 özellik ortaya çıkmıştır. Sınıflandırma işleminde bu 10 öznitelik kullanılmıştır.

### MB-LBP öznitelik çıkarımı

MB-LBP (Multi-Block Local Binary Pattern) yöntemi kullanılarak görüntülerden özellik çıkarma işlemi gerçekleştirilmiştir. MB-LBP, özellikle doku ve desen bilgisini çıkarmak için tasarlanmış bir özellik çıkarma yöntemidir. Bu çalışmada, kullanılan kodun parametreleri şu şekildedir: radius (çekirdek yarıçapı) = 2, n\_points (örnek sayısı) = 4 ve block\_size (blok boyutu) = 6 olarak belirlenmiştir. Bu parametreler, özellikle görüntülerin desenlerini ve yapılarını tanımlamak için ayarlanabilir. Özellik çıkarma süreci, her bir görüntüyü gri tonlamalı olarak alıp MB-LBP özellik vektörlerini hesaplar. Bu vektörler, görüntülerin içerdikleri desenlere dair bilgiyi temsil eder. Elde edilen özellik vektörleri ve sınıf etiketleri daha sonra makine öğrenimi modellerine giriş olarak kullanılabilir. NumPy dizilerine dönüştürülür. MB-LBP ile öznitelik çıkarımında toplam 6 öznitelik çıkarılmıştır. Tüm özellikler makine öğrenme algoritmalarında sınıflandırma yapmada kullanılmıştır.

### Hibrit öznitelikler ile sınıflandırma

Bu çalışma kapsamında kullanılan 6 farklı öznitelik çıkarımı, LBP, HOG, GLCM, SOBEL, GABOR, MB-LBP kullanılarak farklı öznitelikler elde edilmiş ve klasik makine öğrenme sınıflandırma algoritmaları ile sınıflandırılmıştır. Çıkarılan bu farklı 6 öznitelik çıkarım yöntemlerinin hepsi yan yana getirilmek suretiyle hibrit bir yaklaşımda gerçekleştirilmiştir. Toplamda 50 özellik yan yana getirilmiş ve hibrit olarak sınıflandırma görevi gerçekleştirilmiştir.

### **7.3.3. Derin öğrenme mimarileri ile sınıflandırmada kullanılan parametreler**

ESA ile öğrenme iki farklı şekilde sırayla oluşmaktadır. İlk olarak görüntülerden evrişim katmaları kullanarak öznitelik çıkarımı yapılır. İkinci aşama olarak ise çıkarılan özniteliklerden tam bağlı katman vasıtasıyla öğrenme işlemi gerçekleştirilerek görüntünün hangi sınıfa ait olduğu eğitilerek gerçekleştirilir.

Araştırma deneylerinde kullanılan ESA mimarileri VGG-16, MobileNet, ResNet-50, DenseNet-121, Xception ve Inception-V3'tür. Mimarilerin her biri aynı hiper parametreler kullanılarak eğitilmiştir. Eğitime sıfırdan başlamak yerine transfer öğrenme yöntemi tercih

edilmiştir. Bu yöntem ile aşırı uyumun önüne geçilmesi, zamandan tasarruf edilmesi ve doğruluğun artırılması amaçlanmaktadır. Ağın ağırlıkları için ImageNet ağırlıkları kullanılmıştır. İyi bilinen transfer öğrenme mimarileri, özellik çıkarıcı katmanlarına ve ek bir sınıflandırıcı softmax katmanına sahiptir. Ayrıca ağ içinde aktivasyon için "adam" aktivasyon işlevini kullanır. Deneyler için öncelikle bu mimarilerin softmax katmanına 0,25 oranlı toplu normalizasyon katmanı, tam bağlı katman ve bırakma katmanları eklenmiştir. Bunun yapılan işlemlerin amacı, ağın öğrenmesini engellemektedir. Bu optimize işlemi tüm ESA mimarilerinde kullanılmıştır. Tüm ESA mimarilerine uygulanan optimizasyon parametreleri sırayla verilmiştir.

Bir Sequential modeli oluşturarak katmanları sıralı bir şekilde birleştirir. İlgili katmanlar şu şekildedir: Sequential Model, Bu model, Keras'ın Sequential model tipini kullanarak katmanları sıralı bir şekilde birleştirir. Bu yapı, özellikle evrişim tabanlı bir sinir ağı üzerine ek katmanlar eklemek için kullanılır. Base\_model, katmanı önceden eğitilmiş bir VGG16 modelini içerir. VGG16 modeli, görüntülerden özellik çıkarmak için evrişim tabanı olarak hizmet eder. Transfer learning amacıyla önceki bir evrişimli sinir ağını kullanarak, modelin görsel özellikleri öğrenmesi sağlanır.

Batch Normalization, bu katman, önceki evrişim tabanındaki çıkışları normalize eder. Normalizasyon, eğitim sürecini stabilize etmeye ve genelleştirmeyi artırmaya yardımcı olabilir. axis=-1 parametresi, normalizasyonun her bir özellik kanalına uygulanmasını belirtir. Ayrıca, momentum=0.99 ve epsilon=0.001 parametreleri, normalizasyonun hiper parametreleridir.

Dense, bu katman, 256, 512, 1024 ve 2048 nöron içeren bir tam bağlı katmandır. Ağırlıklar, aktivasyon fonksiyonu ve bias için düzenleme (regularization) uygulamak amacıyla kernel\_regularizer, activity\_regularizer ve bias\_regularizer parametreleri kullanılır. Aktivasyon fonksiyonu olarak ReLU (activation='relu') kullanılmıştır.

Dropout, ağdaki belirli bir yüzdeyi (yüzde 45) rastgele atarak ağın aşırı uymaya karşı daha dirençli hale gelmesine yardımcı olur. seed=123 parametresi, rastgele sayı üretiminde kullanılan tohum değeridir.

Dense(class\_count), çıkış sınıflarını içerir ve sınıf sayısı kadar nörona sahiptir. Softmax aktivasyon fonksiyonu kullanılarak çoklu sınıf sınıflandırma problemleri için uygun bir çıkış sağlanır.

Bu yapı, özellikle transfer öğrenme kullanarak önceki bir modelin özelliklerini aktararak ve ardından özelleştirilmiş bir sınıflandırma görevini gerçekleştirerek modelin performansını artırmayı amaçlar. Düzenleme, normalizasyon ve dropout gibi teknikler, ağırlık genellebilirliğini artırmak için entegre edilmiştir.

### VGG-16

Bu çalışmada, 16 katmanlı VGG-16 ESA mimarisi optimize edilerek kullanılmıştır. TensorFlow ve Keras kullanılarak eğitilen bir VGG-16 modelinin oluşturulmuştur. Ahşap kusur tespiti için kullanılan görüntülerinin giriş boyutunu belirten img\_size değişkeni, (yükseklik, genişlik) olarak tanımlanmış ve 224x224 piksel olarak belirlenmiştir. Channels değişkeni, giriş görüntülerinin renk kanallarının sayısını (RGB renk modeli için 3) belirtir. img\_shape ise giriş görüntülerinin şeklini ifade eder ve img\_size ile channels'ın birleşiminden oluşmaktadır. "class\_count" değişkeni, eğitim veri kümesindeki sınıf sayısını belirtir ve çıkış katmanındaki nöron sayısını tanımlar. "base\_model", önceden eğitilmiş bir VGG-16 modelini içerir. "include\_top=False" parametresi, VGG-16 modelinin tam bağlı katmanlarını kullanmamızı engeller. weights="imagenet" parametresi, VGG-16 modelinin ImageNet veri kümesi üzerinde eğitilmiş ağırlıklarını kullanacağımızı belirtir. Ayrıca, giriş görüntülerinin şeklini belirten input\_shape ve evrişim tabanlı özellik haritasını global maksimum havuzlama ile özetlemeyi ifade eden pooling='max' parametreleri tanımlanır.

Modelin katmanları, "Sequential" modeli kullanılarak sıralı bir şekilde tanımlanır. "base\_model" evrişim tabanını içerir ve özellik çıkarımı için kullanılır. Batch Normalization katmanı, ağırlıkların ve bias'ın öğrenilmesini düzenler. İlk Dense katmanında ReLU aktivasyon fonksiyonu ve L1, L2 düzenlemeleri uygulanmıştır. Dropout katmanı, ağırlık genelleştirilmesini artırmak için kullanılır. Son Dense katmanı, çıkış sınıflarını üretmek için softmax aktivasyon fonksiyonunu kullanır. Modelin derlendiği bölümde, "model.compile" fonksiyonu RMSprop optimizer'i, "categorical\_crossentropy" kaybını ve doğruluk metriğini kullanır. Son olarak, "model.summary()" fonksiyonu

modelin özetini gösterir. Bu özet, içerdiği katman türlerini, çıkış şekillerini ve toplam parametre sayılarını içerir. Önceden eğitilmiş bir VGG-16 modelini kullanarak transfer learning yaklaşımını benimseyerek özellik çıkarımı ve sınıflandırma görevi için özel bir ağ oluşturmuş ve sınıflandırma için kullanılmıştır.

### MobileNet

MobileNet görüntü sınıflandırma modeli oluştururken kullanılan temel parametreleri belirleyerek modelin giriş özelliklerini şunlardır: “img\_size” = (224, 224): Giriş görüntülerinin boyutunu belirler. Bu değer, genellikle yaygın olarak kullanılan bir standart olan 224 piksel x 224 piksel olarak belirlenmiştir. Bu boyut, birçok ESA modelinde yaygın olarak kullanılır. “channels” = 3: Giriş görüntülerinin renk kanallarının sayısını belirtir. RGB (Red, Green, Blue) renk modeli kullanıldığı için 3 olarak belirlenmiştir. Her bir piksel, üç renk kanalıyla temsil edilir. “img\_shape” = (img\_size[0], img\_size[1], channels): Bu değişken, giriş görüntülerinin tam şeklini belirler. Yani, (yükseklik, genişlik, kanallar) olarak bir demet içinde ifade edilir. Bu, modelin giriş veri şeklini belirlemek için kullanılır. “class\_count” = len(list(train\_gen.class\_indices.keys)): Bu değişken, eğitim veri setinde bulunan sınıf sayısını belirler. Eğitim veri setindeki sınıfları ve bunlara karşılık gelen sayısal indisleri içeren bir sözlük kullanılır. class\_indices sözlüğünün keys() metodunu kullanarak alınan sınıf etiketlerinin listesi, len() fonksiyonu ile de sınıf sayısını belirler. Bu ESA mimarisi ardına optimize edilen parametreler eklenerek sınıflandırma görevi gerçekleştirilmiştir.

### ResNet-50

ResNet-50 katmanlı ESA mimarisi deneylerde kullanılmıştır. ResNet mimarisinin 18, 101 ve 152 katmanlı modelleri de bulunmaktadır. Kullanılan filtre boyutları 7x7 adım sayısı ise 2’dir. Giriş boyutu (224, 224, 3) şeklindedir. Transfer öğrenme uygulanabilir bir ESA mimarisi olduğu için çalışmada tercih edilmiştir. İlk olarak 7x7’ lik evrişim uygulanır. Arkasından 3 x 3’lük maksimum havuzlama işlemi gerçekleştirilir. Bu işlemler sonunda önemli özellikler vurgulanır. Bu aşamalardan sonra 3x3 evrişim ve 2x2 boyutunda ortak havuzlama gerçekleştirilir. Sonra Residual bloklar kullanılarak işlemler devam edilir, optimize edilen parametrelerle sınıflandırma işlemi gerçekleştirilmiştir.

### DenseNet-121

DenseNet-121, yoğun bağlantıları (dense connections) kullanarak öne çıkan bir derin öğrenme modelidir. Mimari, 121 katmandan oluşur ve girişten başlayarak evrişim, yoğun ve geçiş bloklarını içerir. İlk evrişim katmanında 7x7 filtreler ve ardından gelen maksimum havuzlama katmanında 3x3 filtreler kullanılır, adım sayısı ise 2'dir. Yoğun bloklarda 1x1 ve 3x3 filtrelerin yanı sıra adım sayısı 1'dir. Geçiş katmanları, özellik haritasının boyutunu azaltmak ve bilgi akışını yönetmek için kullanılır, ortalama havuzlama ve 1x1 filtreler içerir. Global ortalama havuzlama katmanı, özellik haritasının boyutunu küçülterek modelin öğrendiği özellikleri özetler. Son olarak, tam bağlı bir katman ve softmax aktivasyonu içeren çıkış katmanı, sınıflandırma sonuçlarını üretir. DenseNet-121, yoğun bağlantıların sağladığı bilgi transferi sayesinde daha etkili öğrenme ve derin ağların eğitilmesinde başarı sağlamış bir modeldir, çalışma kapsamında DenseNet-121 optimize edilmiş hali kullanılmıştır.

### Xception

Xception ESA mimarisindeki evrişim katmanlarının sayısı toplamda 36'dır. Bu katmanlar, derinlik temelli evrişim ve melez evrişim katmanlarını içermektedir. Her biri özellik çıkarma ve öğrenme yeteneklerini temsil eder. Xception'ın derin öğrenme modelinin bu katman yapısı, karmaşık özellikleri etkili bir şekilde çıkarmasına ve geniş bir öğrenme kapasitesine sahip olmasına olanak tanır. Giriş boyutu 299 x 299'dur. Giriş verisi, evrişim tabanına iletilir ve bu taban, özellik çıkarma işlemi için kullanılır. Elde edilen özellik haritaları orta aşamada daha da işlenir, yüksek düzeyde özelliklerin öğrenilmesine katkıda bulunur. Çıkış katmanlarında, özellik haritaları sınıflandırma için düzenlenir ve modelin tahminlerini içeren bir vektör elde edilir. Bu tahminler, softmax aktivasyon fonksiyonu kullanılarak olasılık dağılımına dönüştürülür. Model eğitilirken, tahmin hataları geri yayılım algoritması ile ağı tüm katmanlarına iletilir, bu da ağı güncellenmesini sağlar ve öğrenme sürecini devam ettirir. Bu çalışmada, Xception ESA mimarisi farklı nöron sayıları kullanılarak eğitilmiş ve sınıflandırma görevlerinde kullanılmıştır.

### Inception-V3

Inception-V3, temel bileşenleri arasında evrişim katmanları, Inception modülleri ve global ortalama havuzlama bulunur. Evrişim katmanları, düşük seviyeli özellikleri çıkarmak için kullanılır ve ReLU aktivasyon fonksiyonları içermektedir. Inception modülleri, farklı boyutlarda filtreleri ve evrişim kernellerini paralel olarak kullanarak modelin çeşitli özellikleri etkili bir şekilde öğrenmesine olanak tanır. Önce ve sonra bağlantılar, gradient kaybını önler. Global ortalama havuzlama, özellik haritalarını daha da küçülmektedir. Inception-V3, genellikle transfer learning için kullanılan bir modeldir ve özellikle görüntü sınıflandırma görevlerinde başarılı sonuçlar elde edebilir. Bu çalışmada ahşap kusur tespiti ikili sınıflama için kullanılmıştır.

#### **7.3.4. Klasik makine öğrenmesi sınıflandırma algoritmaları parametreleri**

Çalışmada kullanılan 12 farklı sınıflandırma algoritması ve kullanılan hiper parametre ayarları bu bölümde verilmiştir.

#### Random forest classifier

Ağaç tabanlı bir topluluk algoritmasıdır. Karar ağaçlarını rastgele örnekleme ve oylamayı kullanarak sınıflandırma yapar. Her bir karar ağacı, aşağıdaki gibi temel bir matematiksel ifade ile temsil edilir (Breiman, 2001).

Çalışmada, Random Forest Classifier adlı bir makine öğrenimi sınıflandırma modeli oluşturmak için kullanılmış ve modelin temel parametreleri şu şekildedir: n\_estimators=500 (500 ağaç), criterion='gini' (gini bölme ölçütü), max\_depth=None (maksimum ağaç derinliği belirlenmemiş), min\_samples\_split=2 (minimum bölme için örnek sayısı 2), min\_samples\_leaf=1 (minimum yaprak için örnek sayısı 1), max\_features='sqrt' (her bölme için maksimum özellik sayısı karekökü), bootstrap=True (Bootstrap örnekleme kullanılsın mı?), ve random\_state=42 (rastgele başlangıç değeri 42). Bu parametrelerde deneyler gerçekleştirilmiştir.

Bu parametreler, modelin genelleme yeteneği, karmaşıklığı ve bölme kriterlerini belirler. Özellikle, ağaç sayısı, bölme ölçütü ve örnekleme kriterleri gibi faktörler, modelin eğitim

sürecindeki performansını etkiler. Rastgele orman, birden çok karar ağacının bir araya getirilmesiyle oluşturulduğu bir topluluk yöntemidir ve genellikle sınıflandırma problemlerinde kullanılmaktadır.

### KNN

KNN, veri noktalarını çevresindeki k en yakın komşunun etiketlerine dayalı olarak sınıflandırır (Cover ve Hart, 1967).

KNN sınıflandırıcı modeline kullanılan temel parametreleri şu şekildedir: n\_neighbors=1 (bir örneğin tahmininde kullanılacak komşu sayısı), weights='distance' (komşuların uzaklıklarına ters orantılı olarak etkileme ağırlığı uygulama seçeneği), algorithm='kd\_tree' (veri setini hızlı bir şekilde aramak için kullanılan KD ağacı algoritması), leaf\_size=10 (KD ağacı yapısındaki yaprak düğüm boyutu), p=2 (Minkowski mesafe ölçüsü için parametre, genellikle öklidyen mesafe kullanıldığında p=2 olarak belirlenir) ve metric='minkowski' (komşuluk ölçüm metriği olarak Minkowski mesafe ölçüsünün kullanılacağını belirten seçenektir).

Bu araştırmada kullanılan parametre seçimleri, her bir örneğin en yakın komşusuna göre sınıflandırılmasını sağlar ve mesafeye dayalı ağırlıklandırma sayesinde daha yakın komşuların daha fazla etkiye sahip olmasına imkan tanımaktadır. Farklı n değerleri ve ağırlıklandırma yöntemleri çalışma kapsamında denedikten sonra en iyi sonuçların bu kombinasyonla elde edilmiştir. Bu, veri kümesinin yapısına bağlı olarak daha düşük n değerlerinin daha iyi genelleştirme sağlayabildiğini ve veri noktalarının birbirine olan mesafelerinin sınıflandırma üzerinde önemli bir rol oynadığını göstermektedir. Özellikle n=1 seçeneği, veri setinin sınıflar arasında belirgin ayrımlara sahip olduğunu ve daha büyük k değerlerinin bu ayrımları bulanıklaştırarak doğruluğu düşürdüğü görülmüştür.

Bu parametreler, KNN modelinin temel özelliklerini belirler. Çalışmada bu parametreler kullanılmıştır. KNN, bir örneğin sınıflandırılması için çevresindeki komşularının etiketlerine dayanır. Model, özellikle küçük veri setlerinde ve basit sınıflandırma görevlerinde etkili olabilmektedir.

## SVM

SVM, iki sınıf arasındaki optimal hiper düzlemi bulur ve veriyi bu hiper düzlemlerle sınıflandırır (Cortes ve Vapnik, 1995).

SVM sınıflandırıcı modeli olan SVC'yi oluşturulmuştur. Modelin temel parametreleri şunlardır: C=1.0 (SVM'in düzenleme parametresi, yüksek C değerleri karmaşıklığı artırır ancak aşırı uyum riskini de artırabilir), kernel='linear' (çekirdek fonksiyonu olarak doğrusal kullanılması, doğrudan özellikleri kullanarak sınıflandırma gerçekleştirir), degree=4 (polinom çekirdek kullanıldığında polinomun derecesi, burada doğrusal çekirdek kullanıldığı için etkisiz), gamma='scale' (RBF çekirdek kullanıldığında çekirdeğin genişliğini kontrol eden parametre, 'scale' seçeneği ölçeklendirmeyi belirtir).

Bu parametreler, SVM modelinin karmaşıklığını, düzenlemesini ve kullanılan çekirdek tipini belirler. SVM, sınıflar arasındaki karar sınırlarını belirgin bir şekilde çizerek sınıflandırma yapar. Parametrelerin seçimi, modelin performansını ve genelleme yeteneğini etkileyebilmektedir.

## Decision tree classifier

Karar ağaçları, veri kümesini temsil eden bir ağaç yapısı oluşturur ve bu yapıyı kullanarak veri noktalarını sınıflandırır. Her iç düğüm (node) bir özellik testini temsil eder ve her yaprak düğüm (leaf node) bir sınıf tahmini yapar (Quinlan, 1986).

Bir Karar Ağacı Sınıflandırıcısı olan DecisionTreeClassifier'ı belirli parametrelerle yapılandırarak oluşturulmuştur, çalışmada kullanılan temel parametreler şunlardır: criterion='gini' (Bölme kriteri olarak 'gini' veya 'entropy' seçenekleri kullanılabilir), splitter='random' (Düğüm bölme stratejisi olarak 'random' veya 'best' seçenekleri), max\_depth=None (Ağacın maksimum derinliği, None olarak ayarlandığında genişletilebileceği anlamına gelir), min\_samples\_split=2 (İç düğümü bölmek için gerekli minimum örnek sayısı), min\_samples\_leaf=1 (Bir yaprak düğümde olması gereken minimum örnek sayısı), min\_weight\_fraction\_leaf=0.0 (Bir yaprak düğümde olması gereken minimum örnek ağırlığı, örnek ağırlıkları kullanıldığında dikkate alınır), max\_features=None (Her bölme için kullanılacak maksimum özellik sayısı, None tüm

özellikleri kullanır), `random_state=None` (Rastgelelik için kullanılacak tohum değeri), `max_leaf_nodes=None` (Maksimum yaprak düğüm sayısı, `None` olarak ayarlandığında sınırlama getirilmez), `min_impurity_decrease=0.0` (Bölme yapmadan önce gereken minimum impurity azalma değeri), `class_weight=None` (Sınıf ağırlıkları dengesizliklerini ele alır, `None` tüm sınıfların eşit ağırlığa sahip olduğu anlamına gelir; "balanced" seçeneği sınıf frekanslarına göre ağırlıkları otomatik olarak ayarlar). Bu parametreler, Karar Ağacı modelinin yapısal özelliklerini ve davranışını belirleyerek, modelin belirli bir veri kümesine uyum sağlamasını kontrol eder. Bu parametrelerle sınıflandırma görevi gerçekleştirilmiştir.

### Naive bayes

Temelinde Bayes Teoremi'ne dayanır. Bu algoritma "saflik varsayımı" olarak bilinen bir özelliği kullanır. Yani, veri noktalarının bir sınıfa ait olma olasılığı, bu veri noktasının özelliklerine bağımsız olarak hesaplanır (Lewis, 1998).

Gaussian Naive Bayes sınıflandırıcısını (GaussianNB) kullanarak bir model oluşturulmuş, Gaussian Naive Bayes, özelliklerin veri setindeki dağılımlarını normal (Gaussian) olarak varsayar. Parametre olarak yalnızca priors kullanılmıştır. priors, eğitim örneklerinin sınıf etiketlerinin sıklığını ifade eden önceki olasılıkları belirler; bu değerler varsayılan olarak sınıf etiketlerinin veri setindeki frekanslarına dayanarak tahmin edilir. Gaussian Naive Bayes, özellikle özellikler arasındaki bağımlılıkların basit olduğu durumlarda etkilidir. Çalışmada bu parametre seçilerek sınıflama yapılmıştır.

Naive Bayes sınıflandırıcıları, genellikle özelliklerin birbirinden bağımsız olduğunu ve belirli bir dağılıma sahip olduğunu varsayan olasılıksal modellerdir. Özellikle Gaussian Naive Bayes yöntemi, özelliklerin normal (Gaussian) dağılım gösterdiğini kabul eder. Ancak, gerçek dünyadaki veriler çoğu zaman bu varsayımı karşılamaz. Bu varsayım sağlanmadığında, olasılıklar hatalı tahmin edilir ve karar sınırları yanlış konumlanabilir (Murphy, 2012). Çıkarılan öznitelikler normal dağılıma uygun olmadığından dolayı sağlıklı ve yüksek başarılı sonuçlar alınmayacaktır.

### Logistic regression

Sınıflandırma için kullanılan bir istatistiksel bir modeldir. Lojistik regresyon, doğrusal bir regresyon modelini kullanarak sınıflandırma yapar. Yani, bir veri noktasının belirli bir sınıfa ait olma olasılığını tahmin eder. Lojistik regresyon, bir logit fonksiyonunu kullanarak sınıf olasılıklarını tahmin eder (Hosmer ve diğerleri, 2013).

Lojistik Regresyon Sınıflandırıcısı, modelini belirli parametrelerle oluşturulmuştur. Temel parametreler şu şekildedir: `penalty='l2'`, bu seçenek L2 düzenlemesini belirtir, diğer seçenekler arasında 'l1' ve 'elasticnet' bulunur. `C=0.01` parametresi, düzenleme miktarını kontrol eder; düşük C değerleri daha fazla düzenleme uygular. `solver='liblinear'`, optimizasyon algoritmasını belirler; 'liblinear' küçük veri setleri için uygundur. `max_iter=100`, maksimum iterasyon sayısını belirler; optimizasyon algoritması belirli bir iterasyon sayısına ulaştığında durur. `random_state=None`, rastgele başlatma için kullanılacak tohum değeridir; bu değer belirlendiğinde, her çalıştırmada aynı sonuçlar elde edilir.

### Gradient boosting classifier

Gradient Boosting, zayıf öğrenicileri (genellikle karar ağaçları) birleştirerek güçlü bir sınıflandırıcı oluşturur. Her ağaç, önceki ağaçların hatalarını düzeltmeye çalışır (Friedman, 2001).

Sınıflandırmada kullanılan temel parametreler şu şekildedir: `n_estimators=500` (Boosting turu sayısı), `learning_rate=0.1` (Öğrenme oranı), `max_depth=10` (Ağaçların maksimum derinliği), `min_samples_split=2` (Bir iç düğümü bölme için gereken minimum örnek sayısı), `min_samples_leaf=1` (Bir yaprak düğümünde olması gereken minimum örnek sayısı), `subsample=1.0` (Eğitim verilerinin alt kümesi oranı), `max_features=None` (En iyi bölünmeyi bulmak için düşünülecek maksimum özellik sayısı), `random_state=42` (Rastgele başlatma için kullanılacak tohum değeri). Bu parametreler, Gradient Boosting modelinin karmaşıklığını, öğrenme hızını ve ağaçların yapısını belirleyerek, modelin belirli bir veri kümesine uyum sağlamasını kontrol eder. Gradient Boosting, zayıf öğrenicileri bir araya getirerek yüksek performanslı bir model elde etme yeteneğiyle öne çıkmaktadır.

## XGBoost

Sınıflandırma ve regresyon problemleri için güçlü bir makine öğrenme algoritmasıdır. XGBoost, Gradyan Artırma (Gradient Boosting) tekniğini optimize edilmiş bir sürümüdür ve birçok yarışma ve uygulama alanında başarıyla kullanılmıştır (Chen ve Guestrin, 2016).

XGBoost sınıflandırıcısını (XGBClassifier) belirli parametrelerle oluşturarak bir model oluşturulmuştur. Temel parametreler şu şekildedir: `n_estimators=500` (Boosting turu sayısı), `learning_rate=0.1` (Öğrenme oranı), `max_depth=10` (Ağaçların maksimum derinliği), `min_child_weight=1` (Bir çocuk düğüm oluşturmak için gerekli minimum örnek ağırlığı), `subsample=1.0` (Eğitim verilerinin alt kümesi oranı), `colsample_bytree=1.0` (Her bir ağaç oluşturulurken düşünülecek özelliklerin oranı), `objective='binary:logistic'` (İkili sınıflandırma için amaç fonksiyonu), `random_state=42` (Rastgele başlatma için kullanılacak tohum değeri). Bu parametreler, XGBoost modelinin karmaşıklığını, öğrenme hızını ve ağaçların yapısını belirleyerek, modelin belirli bir veri kümesine uyum sağlamasını kontrol eder.

## LightGBM

LightGBM, temel olarak bir karar ağacı (decision tree) tabanlı gradient boosting algoritmasıdır. Toplam tahmin, birden fazla karar ağacının toplamıdır (Ke ve diğerleri, 2017).

LightGBM sınıflandırıcısını (LGBMClassifier) oluşturarak bir model inşa edilmiştir. Temel parametreler ayarları şunlardır: `n_estimators=500` (Boosting turu sayısı), `learning_rate=0.1` (Öğrenme oranı), `max_depth=10` (Ağaçların maksimum derinliği), `min_child_weight=1` (Bir çocuk düğüm oluşturmak için gerekli minimum örnek ağırlığı), `subsample=1.0` (Eğitim verilerinin alt kümesi oranı), `colsample_bytree=1.0` (Her bir ağaç oluşturulurken düşünülecek özelliklerin oranı), `objective='binary'` (İkili sınıflandırma için amaç fonksiyonu), `random_state = 42` (Rastgele başlatma için kullanılacak tohum değeri). Bu parametreler, LightGBM modelinin karmaşıklığını, öğrenme hızını ve ağaçların yapısını belirleyerek, modelin belirli bir veri kümesine uyum sağlamasını kontrol eder. LightGBM, yüksek performans ve hızlı eğitim süreleri sağlayan bir sınıflandırıcı olarak bilinir, özellikle büyük veri setleriyle etkili bir şekilde çalışmaktadır.

## CatBoost

CatBoost, temel olarak bir karar ağacı (decision tree) tabanlı bir gradient boosting algoritmasıdır. Toplam tahmin, birden fazla karar ağacının toplamıdır (Prokhorenkova, 2018).

CatBoost sınıflandırıcısını (CatBoost Classifier) eğitmek için kullanılan temel parametreleri şunlardır: Kayıp fonksiyonu (loss\_function) olarak 'CrossEntropy' kullanılmıştır. Modelin eğitimini gerçekleştirmek için 500 iterasyon (iterations), 0.1 öğrenme oranı (learning\_rate), ve maksimum ağaç derinliği (depth) olarak 10 kullanılmıştır. Eğitim sürecinde ayrıntılı çıktıların görüntülenmemesi için verbose parametresi False olarak ayarlanmıştır.

Bu parametreler, CatBoost modelinin karmaşıklığını, öğrenme hızını ve ağaçların yapısını belirleyerek, modelin belirli bir veri kümesine uyum sağlamasını kontrol eder. Cat Boost Classifier, kategorik değişkenlerle otomatik olarak başa çıkma yeteneği ile bilinir ve genellikle kategorik veri setleri üzerinde etkili bir şekilde çalışmaktadır.

## AdaBoost

Zayıf öğrencileri birleştirerek güçlü bir sınıflandırıcı oluşturan bir topluluk öğrenme tekniğidir. AdaBoost, özellikle sınıflandırma problemlerinde kullanılır ve başlangıçta hatalı sınıflandırılan örnekler üzerine odaklanır (Fruend ve Schapire, 1997).

AdaBoost sınıflandırıcısını belirli parametrelerle oluşturmuştur, model inşasında temel parametreler arasında n\_estimators=500 (Boosting turu sayısı), learning\_rate=0.1 (Öğrenme oranı) ve random\_state = 42 (Rastgele başlatma için kullanılan tohum değeri) bulunmaktadır. Bu parametreler, AdaBoost modelinin karmaşıklığını, öğrenme hızını ve birleştirilen zayıf öğrencilerin katkılarının büyüklüğünü kontrol ederek, modelin belirli bir veri kümesine uyum sağlamasını sağlar. AdaBoost, zayıf öğrencileri birleştirerek güçlü bir model oluşturur ve özellikle ensemble (topluluk) öğrenme yöntemleri arasında popülerdir.

## MLP

Derin öğrenme modelleri arasında yer alan ve yapay sinir ağı sınıfına dahil olan bir türdür. MLP, özellikle sınıflandırma ve regresyon problemleri için kullanılır ve birden çok katmandan oluşmaktadır (LeCun ve diğerleri, 1998).

MLP sınıflandırıcısını oluşturulmuş ve ardından MLPClassifier sınıfının parametreleri belirlenerek model mimarisi tanımlanmıştır. `hidden_layer_sizes = (100, 50)` ifadesi ile birinci gizli katmanda 100 nöron, ikinci gizli katmanda ise 50 nöron bulunmaktadır. `max_iter=500` parametresi, eğitim için belirlenen maksimum iterasyon sayısını belirtir. Model, bu iterasyon sayısına ulaştığında öğrenmeyi durdurur. `random_state = 42` ise rastgele başlatma için kullanılan tohum değeridir. MLP ile bu parametreler kullanılarak sınıflandırma görevi gerçekleştirilmiştir.

Klasik makine öğrenmede yapılan deneylerde denemeler sonucunda elde edilen en iyi hiper parametre ayarları Çizelge 7.1' de verilmiştir.

Çizelge 7.1. Klasik makine öğrenmede denemelerde elde edilen en iyi hiper parametreler

Model	Parametre	Değer	Açıklama
Random Forest	n_estimators	500	500 karar ağacı kullanıldı.
	criterion	'gini'	Gini katsayısı kullanıldı.
	max_depth	None	Derinlik sınırlanmadı.
	min_samples_split	2	Bölünme için minimum örnek sayısı 2.
	min_samples_leaf	1	Yaprak düğümde minimum örnek sayısı 1.
	max_features	'sqrt'	Kareköklü özellik seçimi.
	bootstrap	True	Bootstrap örnekleme etkin.
KNN	n_neighbors	1	Bir komşu kullanıldı.
	weights	'distance'	Mesafeye dayalı ağırlık verildi.
	algorithm	'kd_tree'	KD Ağacı algoritması tercih edildi.
	leaf_size	10	Yaprak boyutu 10 olarak belirlendi.
SVM	C	1.0	Regularizasyon parametresi.
	kernel	'linear'	Lineer çekirdek fonksiyonu seçildi.
	gamma	'scale'	Ölçeklendirme sağlandı.
Decision Tree	criterion	'gini'	Gini katsayısı kullanıldı.
	splitter	'random'	Rastgele bölünme stratejisi.
	max_depth	None	Derinlik sınırlanmadı.
Logistic Regression	penalty	'l2'	L2 regularizasyonu.
	solver	'liblinear'	Optimizasyon algoritması.
Naive Bayes	-	-	Gaussian dağılım varsayımıyla kullanıldı.
Gradient Boosting	n_estimators	500	500 zayıf öğrenici kullanıldı.
	learning_rate	0.1	Öğrenme oranı 0.1.
XGBoost	n_estimators	500	500 zayıf öğrenici kullanıldı.
	learning_rate	0.1	Öğrenme oranı 0.1.
LightGBM	n_estimators	500	500 zayıf öğrenici kullanıldı.
	learning_rate	0.1	Öğrenme oranı 0.1.
AdaBoost	n_estimators	500	500 zayıf öğrenici kullanıldı.
	learning_rate	0.1	Öğrenme oranı 0.1.
MLPClassifier	hidden_layer_sizes	(100, 50)	Gizli katman yapısı 100 ve 50.
	max_iter	500	Maksimum iterasyon sayısı 500.

#### 7.4. Deneysel Kurulum

Bu araştırma için kullanılan veri seti iki kategoriye ayrılmıştır; kusurlu ve kusursuz ahşap görüntüleri şeklinde ayrılmıştır. Kullanılan veri setindeki sınırlı sayıda kusursuz ahşap görüntü nedeniyle, veri dağılımını dengelemek ve ağın performansını artırmak için eğitim seti klasörü için deki görüntü miktarı artırılmıştır. Bu artırma işlemi, kusurlu ahşap görüntülerin sayısına eşitlenerek tamamlanmıştır. Veri artırma işlemi sadece eğitim seti üzerinde gerçekleştirilmiştir. Doğrulama ve test klasörleri içerisinde bulunan görüntülerde

artırma yapılmamıştır. Veri seti, 1 992 mükemmel ahşap görüntüsünü ve 18 284 kusurlu ahşap görüntüsünü kapsayan toplam 20 276 görüntüden oluşmaktadır. Veri kümesi rastgele üç alt kümeye eğitim için %70, doğrulama için %15 ve test için %15 olmak üzere 3'e ayrılmıştır. Veri artırmanın kusursuz ahşap kümesine dahil edilmesiyle, 12798 kusurlu ahşap görüntüsü ve 12 789 kusursuz ahşap görüntüsü olmak üzere 25 596 görüntü veri setinde hazırlanmıştır.

Klasik makine öğrenimi için 18 284 kusurlu ahşap yüzey mevcut veri setinin orijinal haline vardır. 1 992 olan kusursuz ahşap sayısı veri artırımı yapılarak 18 284 adet görüntü oluşturulmuştur. Toplamda 36 568 görüntü sınıflandırmada kullanılmıştır. Sınıflandırma için önce öznitelik çıkarımı yapılmış ve elde edilen özellikler kaydedilmiştir. Öznitelik çıkarımları ekran kartında değil CPU da yapılmıştır. Bu durumda hızlı ve düşük donanımlı bilgisayarlar ile işlem yapmada klasik makine öğrenim yöntemlerini kullanma avantajın sağlamıştır. Sonra çıkarılan özellikle 12 farklı sınıflandırma algoritmasında sınıflandırılmıştır.

Hibrit yaklaşımda sınıflandırmada ise ESA mimarilerin çıkardığı özellikler dosyalara yazdırılmıştır. Daha sonra klasik makine öğrenme algoritmalarından sınıflandırma görevi gerçekleştirilmiştir.

Deneyle, bir Nvidia Tesla P100 grafik kartında, Google altyapısı kullanılarak Kaggle çekirdekleri bulut ortamında gerçekleştirilir. Uygulamanın yazılımı için Python programlama dili kullanılmıştır. Sınıflandırma sonucunda sınıflandırma raporları oluşturulmuş, AUC eğrileri ve karmaşıklık matrisleri de yazdırılmıştır. Daha sonra bu sonuçlar karşılaştırmalı olarak değerlendirilmiştir.

## **7.5. Sınıflandırma**

### **7.5.1. Derin öğrenme mimarileri ile ahşap yüzey kusur tespiti**

Bu sınıflandırma görevi için 6 farklı tanınmış ESA mimarisi (VGGNet-16, ResNet50, MobileNet, DenseNet121, Xception ve InceptionV3) görüntüler kullanılarak ahşap yüzey kusur tespiti için eğitildi. Mimarilerin her biri aynı hiper parametreler kullanılarak eğitilmiştir. Eğitime sıfırdan başlamak yerine transfer öğrenme yöntemi tercih edilmiştir.

Bu yöntem ile aşırı uyumun önüne geçilmesi, zamandan tasarruf edilmesi ve doğruluğun artırılması amaçlanmaktadır. Ağın ağırlıkları için ImageNet ağırlıkları kullanılmıştır. İyi bilinen transfer öğrenme mimarileri, özellik çıkarıcı katmanlarına ve ek bir sınıflandırıcı softmax katmanına sahiptir. Ayrıca ağ içinde aktivasyon için "adam" aktivasyon işlevini kullanır. Deneyler için öncelikle bu mimarilerin softmax katmanına 0.25 oranlı toplu normalizasyon katmanı, tam bağlı katman ve bırakma katmanları eklenmiştir. Bunun amacı, ESA mimarisinin aşırı öğrenme (overfitting) önlemektir.

Karşılaştırmalı ve kapsamlı analiz sağlamak amacıyla deneylerde Adam, RMSprop, Adadelta, Adagrad ve SGD gibi farklı optimize ediciler kullanılmıştır. Aynı zamanda en başarılı optimize edici için mimarilerin her birine 256, 512, 1024 ve 2048 nöron eklenir ve performansları ayrı ayrı ölçülür.

ESA mimarileri, eğitim başlangıcındaki öğrenme oranları standart 0,0001 olarak belirlenmiş ve her eğitim döngüsünde doğrulama kaybı kontrol edilmiş, 5 epoch boyunca kayıp azalmadıysa öğrenme oranı %50 oranında azaltılmıştır. Böylece dinamik öğrenme oranının kullanılması hedeflenmektedir. Çalışma çok sayıda deney ve karşılaştırmalı analiz içerdiğinden, eğitim süreleri dikkate alınarak dönem sayısı 20 olarak belirlenmiştir. Toplu iş boyutu, ram ve grafik kartı kapasitesi ve veri boyutu dikkate alınarak 64 olarak ayarlanmıştır. Bu özelliklerde deneyler gerçekleştirilmiştir.

Derin öğrenme mimarileri, verileri temsil ederek özellikler öğrenir ve sınıflandırma işlemini bu öğrenilen özellikler üzerinden gerçekleştirir. Veri setinin boyutu ve çeşitliliği, sınıflandırma performansını artıran önemli faktörlerdendir. Bu nedenle, veri seti %70 eğitim, %15 doğrulama ve %15 test olarak üçe ayrılmıştır.

Eğitim için 12 798 kusurlu ve 12 798 kusursuz ahşap görüntüsü kullanılmış olup, toplamda 25 596 görüntü eğitim setine ayrılmıştır.

Doğrulama aşamasında, 2 473 kusurlu ahşap görüntüsü ve 299 kusursuz ahşap görüntüsü kullanılarak toplam 2 772 görüntü doğrulama setine dahil edilmiştir.

Test aşaması için ise, 2 473 kusurlu ahşap görüntüsü ve 299 kusursuz ahşap görüntüsü seçilmiş olup, test seti toplam 2 772 görüntüden oluşmaktadır.

Çalışmada, ESA ile derin öğrenme modelini eğitirken eğitim, doğrulama ve test veri setlerini sırasıyla %70, %15 ve %15 oranlarında ayrılmıştır. Bu veri bölme yaklaşımı, genellikle derin öğrenme modellerinde yaygın olarak kullanılan bir yöntemdir (Joulin ve diğerleri, 2016). Eğitim seti modelin öğrenmesi için kullanılırken, doğrulama seti modelin hiper parametre ayarlarını optimize etmek ve modelin aşırı uyum yapmasını engellemek için kullanılmıştır. Test seti ise modelin son genel performansını değerlendirmek amacıyla ayrılmıştır.

### **7.5.2. Klasik makine öğrenme yöntemleriyle ahşap yüzey kusur tespiti**

Önce gerçek boyut 2800 x 1024 bmp formatında olan görüntüler 300 x 300 boyutunda bmp formatında yeniden boyutlandırılmıştır. Daha sonra 1 992 kusursuz ahşap yüzey data Augmentor kütüphanesi yardımıyla artırılarak 18 284 kusursuz ahşap yüzey görüntüsü elde edilmiştir. Bu şekilde, 18 284 kusurlu ahşap yüzey sayısı ile dengeli hale getirilmiş, toplamda 36 568 ahşap yüzey görüntüsü sınıflandırma görevi için hazır hale getirilmiştir. Bu işlemlerden sonra veri seti LBP, HOG, GLCM, Sobel, Gabor, MB-LBP özellik çıkarım yöntemleri ile özellik çıkarımı yapılmıştır. Çıkarılan özellikler makine öğrenme algoritmalarında kullanılmak için standardizasyon yapılmıştır.

Klasik makine öğrenme de kullanılan 12 farklı sınıflandırma algoritması: Random Forest, KNN, SVM, Decision Tree, Naive Bayes, Logistic Regression, Gradient Boosting, XGB, LightGBM, CatBoost, AdaBoost, MLP sınıflandırma algoritmaları seçilmiştir.

Makine öğrenme algoritmalarının gerçek hayattaki uyumluluğunu sağlamak adına, veri seti rastgele %10'luk bölümlere ayrılarak 10 katlı çapraz doğrulama (k=10) yöntemi uygulanmıştır. Bu doğrulama yöntemiyle algoritmalar çalıştırılmış ve sınıflandırma işlemi gerçekleştirilmiştir.

Makine öğrenmesi yöntemlerinde ise çapraz doğrulama kullanılmıştır. Bu yöntem, veri setini k=10 eşit parçaya böler ve her bir parça sırayla doğrulama seti olarak kullanılırken geri kalan parçalar eğitim için kullanılır. Bu sayede modelin genel performansı, tüm veri seti üzerinde değerlendirilmiş olur ve modelin farklı veri parçalarındaki doğruluğu daha güvenilir şekilde ölçülür. Bu yöntem, veri setinin daha verimli kullanılmasını sağlar ve modelin genelleme yeteneğini artırır (Kohavi, 1995; Stone, 1974). Makine öğrenme k=10 seçildiği için verinin her bir döngüsünde %90 eğitim, %10 test olarak çalışmaktadır. Bu

yöntem, veri setini 10 eşit parçaya böler ve her bir parça sırayla doğrulama seti olarak kullanılırken geri kalan 9 parça eğitim için kullanılır. Böylece modelin performansı, tüm veri seti üzerinde daha güvenilir bir şekilde değerlendirilir.

ESA modellerinde sabit bir eğitim, doğrulama ve test bölmesi kullanmak daha hızlı ve pratik bir değerlendirme sağlarken, k-fold çapraz doğrulama yöntemi, veri setinin tamamını eğitim ve test için kullanarak modelin daha sağlam ve güvenilir bir şekilde değerlendirilmesine olanak tanır. Bu nedenle, derin öğrenme ve makine öğrenmesi modellerinde kullanılan veri bölme ve doğrulama yöntemleri, modelin performansını en iyi şekilde ölçmek için farklı stratejiler sunmaktadır.

### **7.5.3. Hibrit yaklaşım ile ahşap yüzey kusur tespiti**

Derin öğrenme ile sınıflandırma yapılan 6 farklı tanınmış ESA mimarisi (VGGNet-16, ResNet50, MobileNet, DenseNet121, Xception ve InceptionV3) görüntüler kullanılarak ahşap yüzey kusur tespiti için özellikler çıkarılmıştır. Çıkarılan bu özellikler klasik makine öğrenme algoritmasında sınıflandırma görevinde kullanılan algoritmaların aynı parametre değerleri kullanılarak sınıflandırma yapılmıştır. Klasik makine öğrenmede sınıflandırma görevinde kullanılan 12 farklı sınıflandırıcının deney sonuçları verilmiştir. ESA ile öznelik çıkarımlarında deneyler arası karşılaştırma yapabilmek adına derin öğrenme ile özellik çıkarımında kullanılan aynı parametreler kullanarak öznelik çıkarımları yapılmış ve sınıflandırma işlemi gerçekleştirilmiştir.

### **7.6. Hiper parametre**

Bir makine öğrenimi modelinin performansını belirleyen, ancak modelin eğitimi sırasında otomatik olarak öğrenilmeyen parametrelerdir. Öğrenme oranı, epok sayısı, mini-batch boyutu, aktivasyon fonksiyonları, ağırlık inisyalizasyonu (başlangıç değeri atama), dropout oranı ve doğrulama bölümü gibi önemli hiper parametreler, modelin başarısını etkiler. Örneğin, yüksek öğrenme oranları modelin hızlı öğrenmesine yol açabilir, ancak aşırı uygulamaya neden olabilir. Epoch sayısı, modelin veri setini kaç kez geçeceğini belirler ve aşırı uyuma karşı dikkatli bir denge gerektirir. Mini-batch boyutu, bellek kullanımını ve eğitim süresini etkiler. Aktivasyon fonksiyonları, ağırlık inisyalizasyonu ve dropout oranı gibi hiper parametreler, modelin mimarisini ve genelleme yeteneklerini

etkiler. Doğrulama bölümü, modelin genelleme performansını değerlendirmek için ayrılan veri setini belirler. Bu hiper parametrelerin optimal ayarı, deneme-yanılma yöntemleri veya otomatik hiper parametre ayarlama araçları kullanılarak gerçekleştirilebilir (Géron, 2022). Çalışmada kullanılan hiper parametre ayarları verilmiştir.

Makine öğrenmesi ve derin öğrenme modellerinde hiper parametre ayarı önemli bir rol oynar. Her iki alanda da hiper parametreler, modelin eğitim sürecini etkileyen ve modelin performansını iyileştirmek için optimize edilmesi gereken ayarlardır. Derin öğrenme, genellikle daha büyük veri setleri ve karmaşık modeller kullanması nedeniyle hiper parametre ayarlarını daha geniş bir yelpazede yapmayı gerektirirken, makine öğrenmesi modelleri daha sınırlı parametre alanlarıyla çalışabilir. Ancak her iki alanda da doğru hiper parametre ayarları, modelin doğruluğunu ve genelleme kabiliyetini artırmak için kritik öneme sahiptir.

### **7.7. Çapraz Doğrulama (Cross-Validation)**

Bir makine öğrenimi modelinin performansını güvenilir bir şekilde değerlendirmek amacıyla kullanılan önemli bir tekniktir. Bu yöntem, veri setini eğitim ve test alt kümelerine bölerek modelin genelleme yeteneğini değerlendirmeyi sağlar. İlk olarak, veri seti genellikle belirli bir sayıda (K) alt küme veya katmana bölünür. Her bir katman sırayla test seti olarak kullanılır, böylece model farklı veri alt kümeleri üzerinde değerlendirilebilir. K-Fold sürecinin her bir aşamasında, model eğitim verilerini kullanarak K-1 katman üzerinde eğitilir ve kalan 1 katman üzerinde test edilir. Bu adım, her katmanın hem eğitim hem de test verileri olarak kullanılmasını sağlar. Her K-Fold aşamasının tamamlanmasının ardından, elde edilen performans ölçümleri toplanır. Bu ölçümler genellikle doğruluk, hassasiyet, geri çağırma gibi metrikleri içerir. Son olarak, elde edilen performans ölçümlerinin ortalamaları alınarak modelin genel performansı hesaplanır. Bu, modelin farklı veri alt kümelerindeki performansını daha güvenilir bir şekilde değerlendirmeye yardımcı olur. Çapraz doğrulama, özellikle veri setinin küçük olduğu durumlarda veya modelin performansının belirli bir veri bölümüne bağlı olmadığı durumlarda kullanışlıdır. K-Fold, Leave-One-Out (LOO) ve Stratified K-Fold gibi farklı çapraz doğrulama stratejileri, çeşitli durumlara uygun olarak kullanılabilir (Géron, 2022).

## 7.8. Değerlendirme Metrikleri

### Doğruluk (accuracy)

Chicco ve Jurman (2020), doğruluk metriğinin doğru tahmin edilen veri noktaları ile veri kümesindeki tüm veri noktaları arasındaki bir oran olduğunu sunmuştur. Denklem (7.15), doğruluk ölçümlerinin formülünü ifade etmektedir.

$$doğruluk = \frac{DP + DN}{DP + YN + YP + YN} \quad (7.15)$$

DP, DN, YP ve YN, sırasıyla gerçek pozitif, gerçek negatif, yanlış pozitif ve yanlış negatif terimlerini temsil eden kısaltmalardır. Modellerin genel başarısı değerlendirmek amacıyla kullanılmaktadır.

### Kesinlik (precision)

Bu metrik, karmaşıklık matrisinde doğru tahmin edilen pozitif örneklerin, modelin pozitif olarak tahmin ettiği tüm pozitif örnekler içindeki oranını gösterir. Bu oran, her iki sınıf için de hesaplanabilir. Bu metrik, bir sınıf için yapılan tahminlerin doğruluk oranını belirler. (Power, 2020). Denklem (7.16) kesinlik puanını göstermektedir.

$$kesinlik = \frac{DP}{DP + YP} \quad (7.16)$$

### Hassasiyet (recall)

Bu metrik, modelin doğru bir şekilde tahmin ettiği pozitif veri sayısının, gerçekte var olan toplam pozitif veri sayısına bölünmesi ile elde edilir. Bu metrik aynı zamanda modelin doğru pozitif oranı olarak da bilinir. Bu değer, her iki sınıf için de hesaplanabilir (Power, 2020). Denklem (7.17) geri çağırma puanını göstermektedir.

$$hassasiyet = \frac{DP}{DP + YN} \quad (7.17)$$

F1 skoru (F1-score)

Chicco ve Jurman'a (2020) göre, F1 puanı, Denklem (7.18)'te açıklandığı gibi harmonik kesinlik ve geri çağırma ortalamasını hesaplayan bir metriktir.

$$F1-Skoru = \frac{2 \times kesinlik \times hassasiyet}{kesinlik + hassasiyet} \quad (7.18)$$

Tüm değerlendirme metrikleriyle ilgili olarak, en kötü değer 0, en iyi değer ise 1'dir.

ROC - AUC

Receiver Operating Characteristic (ROC) eğrisi, bir sınıflandırma modelinin performansını değerlendirmek ve göstermek için kullanılan bir grafiktir (Adegun ve Viriri, 2020).

Eğrinin Altındaki Alan (AUC), ROC Eğrisinin altındaki alanı ölçerek bir sınıflandırma mimarisinin ayırt etme yeteneğinin kapsamını ölçer. Mimarinin olumlu ve olumsuz örnekler arasında ne kadar iyi ayırım yapabileceğinin bir ölçüsünü sağlar. Eğrinin altında daha geniş bir alanı gösteren AUC değeri arttıkça, mimarinin ayırt etme yeteneği gelişmektedir (Adegun ve Viriri, 2020).

Hata oranı (error rate)

Hata Oranı, gerçek değer ile sınıflandırıcı tarafından elde edilen değer arasındaki mutlak farkın oranını gösterir (Adegun ve Viriri, 2020).

## 8. BULGULAR VE TARTIŞMA

Yapılan çalışmalarda 3 farklı sınıflandırma görevi ile ahşap yüzey kusur tespiti gerçekleştirilmiştir. Bu görevler; derin öğrenme, klasik makine öğrenme ve hibrit yaklaşımla ahşap yüzey kusuru tespit etme şeklindedir. Görüntü boyutları 300 x 300 bmp formatında yeniden boyutlandırılmış ve veri setleri dengeli hale getirildikten sonra kusurlu ve kusursuz ahşap yüzeyleri tespit edilmiştir. Kullanılan ESA mimarileri sıfırdan eğitilmemiş ve transfer öğrenme tekniği kullanılarak deneyler gerçekleştirilmiştir. ESA mimarileri ImageNet eğitim ağırlıkları kullanılarak veri saatinde bulunan ahşap görüntüleri ile yeniden eğitilmiştir. ESA mimarileri 20 epoch eğitilmiş, Batch size 64 olarak kullanılmıştır. Bu bahsedilen özellikler ESA ile özellik çıkarımı aşamasında da aynı olarak kullanılmıştır.

Klasik makine öğrenme ile sınıflandırma görevinde veri seti kfold=10 olarak çapraz doğrulama yapılarak kullanılmıştır. Veri seti dengeli hale getirilmiştir. Klasik makine öğrenme de 6 farklı öznetelik çıkarımı ve 12 farklı sınıflandırma algoritması kullanarak sınıf görevi gerçekleştirilmiştir.

### 8.1. Derin Öğrenme Mimarileri ile Ahşap Yüzey Kusur Tespiti

Deneylerde kullanılan ESA mimarileri VGG-16, MobileNet, ResNet-50, DenseNet-121, Xception ve Inception-V3'tür. Deneyler için 256 nöron ile beş farklı optimizasyon algoritması kullanılarak deneyler gerçekleştirilmeye başlanmıştır. Daha sonra en iyi optimizasyon algoritmasını bulmak için, 5 farklı optimizasyon algoritması kullanılmıştır. Toplamda bu sınıflandırma görevi için 6 ESA mimarisi ve 5 farklı optimizasyon algoritması kullanılarak 30 (6x5) deney gerçekleştirilmiştir. Her bir ESA mimarisi farklı optimizasyon algoritması ile 256 nöron sayısı ile deney gerçekleştirilmiştir.

Sınıflandırma görevinde Adam, RMSprop, Adagrad, SGD ve Adadelta optimizasyon algoritması kullanılmıştır. Kusurlu ve kusursuz ahşap görüntüleri sınıflandırmak için kullanılan mimarilerin performansı Çizelge 8.1'de gösterilmektedir. Deneyler, artırılmış ve dengeli veri seti ile gerçekleştirilir. Eğitim veri kümesinde veri dengelemesi yapılmıştır.

Doğrulama ve test kümelerinde herhangi bir artırma ve dengeleme işlem gerçekleştirilmemiştir.

Çizelge 8.1. ESA mimarilerinin ve optimizasyon algoritmalarının en başarılı sonuçları

ESA Model	Optimizasyon algoritması	Kesinlik	Hassasiyet	F1-skoru	AUC	Süre	Doğruluk
VGG16	Adam	0,9706	0,9698	0,9701	0,9884	309,44 s	0,9698
VGG16	RMSprop*	0,9754	0,9747	0,9750	0,9914	309,19 s	0,9747*
VGG16	Adagrad	0,9705	0,9711	0,9707	0,9918	309,92 s	0,9711
VGG16	SGD	0,9549	0,9566	0,9530	0,9885	308,92 s	0,9566
VGG16	Adadelata	0,9616	0,9586	0,9597	0,9833	312,30 s	0,9586
MobileNet	Adam	0,9720	0,9701	0,9708	0,9866	148,97 s	0,9701
MobileNet	RMSprop*	0,9749	0,9737	0,9742	0,9880	156,01 s	0,9737*
MobileNet	Adagrad	0,9633	0,9612	0,9620	0,9840	146,99 s	0,9612
MobileNet	SGD	0,9542	0,9533	0,9537	0,9781	145,91 s	0,9533
MobileNet	Adadelata	0,9316	0,9316	0,9316	0,9565	145,02 s	0,9316
ResNet50	Adam	0,9709	0,9684	0,9693	0,9899	260,55 s	0,9684
ResNet50	RMSprop*	0,9740	0,9734	0,9736	0,9920	258,28 s	0,9734*
ResNet50	Adagrad	0,9715	0,9701	0,9706	0,9851	259,85 s	0,9701
ResNet50	SGD	0,9695	0,9635	0,9653	0,9880	264,58 s	0,9635
ResNet50	Adadelata	0,9567	0,9556	0,9561	0,9817	260,07 s	0,9556
DenseNet121	Adam*	0,9715	0,9714	0,9715	0,9911	268,23 s	0,9714*
DenseNet121	RMSprop	0,9653	0,9648	0,9651	0,9878	267,61 s	0,9648
DenseNet121	Adagrad	0,9689	0,9668	0,9676	0,9868	266,89 s	0,9668
DenseNet121	SGD	0,9692	0,9665	0,9665	0,9882	266,62 s	0,9665
DenseNet121	Adadelata	0,9580	0,9536	0,9553	0,9797	267,39 s	0,9536
Xception	Adam	0,9711	0,9678	0,9689	0,9892	389,93 s	0,9678
Xception	RMSprop*	0,9756	0,9756	0,9734	0,9911	383,37 s	0,9724*
Xception	Adagrad	0,9662	0,9635	0,9645	0,9823	384,23 s	0,9635
Xception	SGD	0,9626	0,9579	0,9596	0,9828	383,87 s	0,9579
Xception	Adadelata	0,9585	0,9464	0,9501	0,9774	385,66 s	0,9464
InceptionV3	Adam	0,9699	0,9691	0,9695	0,9887	200,64 s	0,9691
InceptionV3	RMSprop*	0,9732	0,9717	0,9723	0,9915	198,71 s	0,9717*
InceptionV3	Adagrad	0,9649	0,9619	0,9630	0,9873	199,61 s	0,9619
InceptionV3	SGD	0,9643	0,9615	0,9626	0,9852	199,06 s	0,9615
InceptionV3	Adadelata	0,9444	0,9352	0,9386	0,9676	199,59 s	0,9352

Not: "\*" sembolü en başarılı sonucu ifade etmek için kullanılmıştır.

Doğruluk oranı bakımından incelenecek olursa; en yüksek doğruluk değerleri, VGG16 RMSprop %94,47 doğruluk, MobileNet RMSprop %97,37 doğruluk, ResNet RMSprop %97,34 doğruluk, DenseNet121 Adam %97,14 doğruluk, Xception RMSprop %97,24 doğruluk, InceptionV3 RMSprop %97,17'tür.

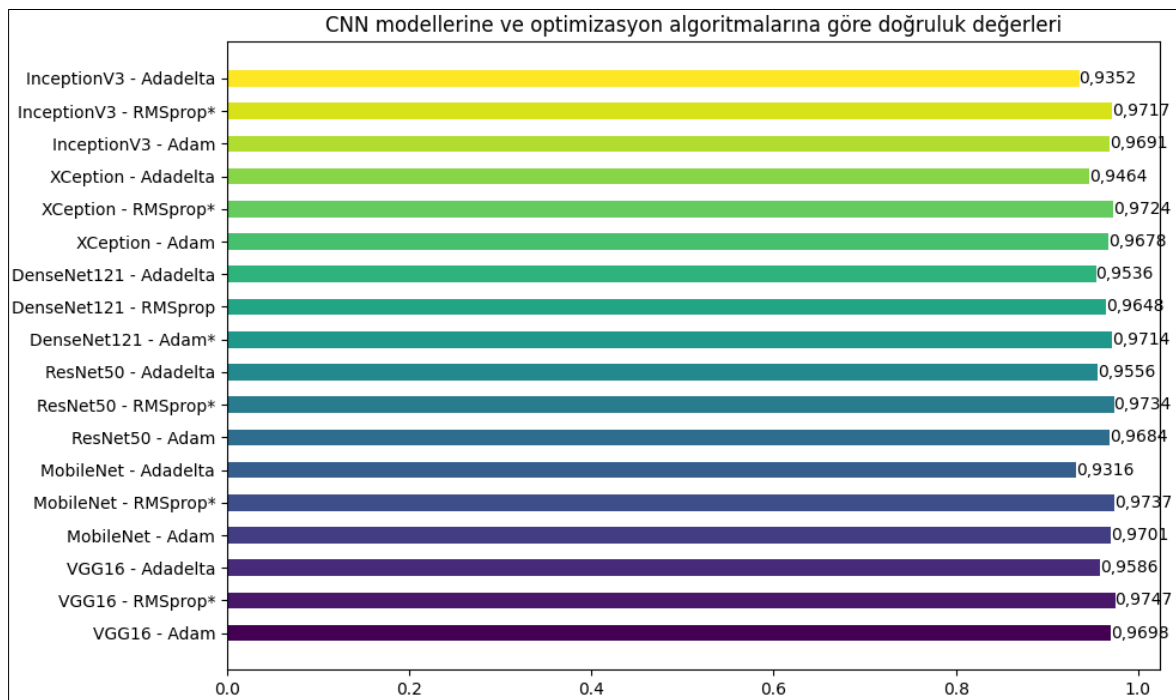
EN yüksek üç kesinlik değerine sahip ESA mimarileri ve optimizasyon algoritmaları; DenseNet121 Adam %97,15 kesinlik, Xception RMSprop %97,56 kesinlik, MobileNet RMSprop: %97,49 olarak elde edilmiştir.

Hassasiyet bakımından tablo incelendiğinde en yüksek hassasiyet oranına sahip 3 ESA modeli ve optimizasyon algoritması şunlardır; Xception RMSprop %97,56 hassasiyet, VGG16 RMSprop %97,47 hassasiyet, Xception – Adam %97,11'tür.

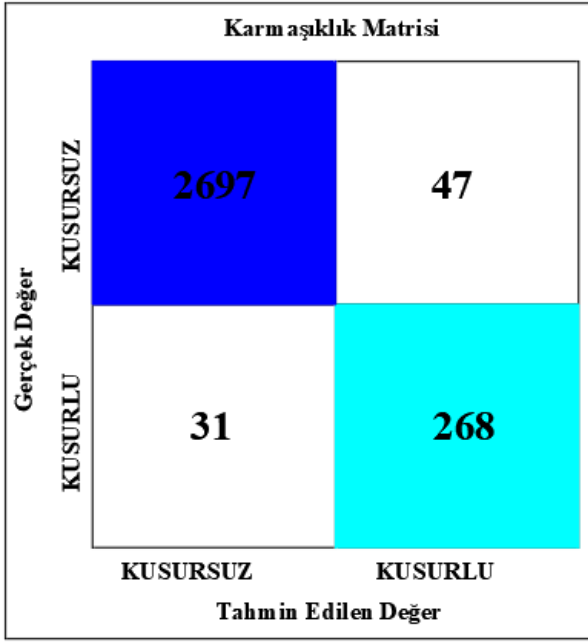
F1-skoru bakımından en iyi sonuç veren mimari ve optimizasyon algoritmaları; Xception RMSprop %97,34 F1-skoru, VGG16 %97,50 ve Xception Adam %96,89 olarak bulunmuştur.

En yüksek AUC değerine sahip olan ESA modeli ve optimizasyon algoritmaları; ResNet50 RMSprop %99,20 AUC, Xception RMSprop %99,11 AUC, InceptionV3 RMSprop %99,15 olarak bulunmuştur.

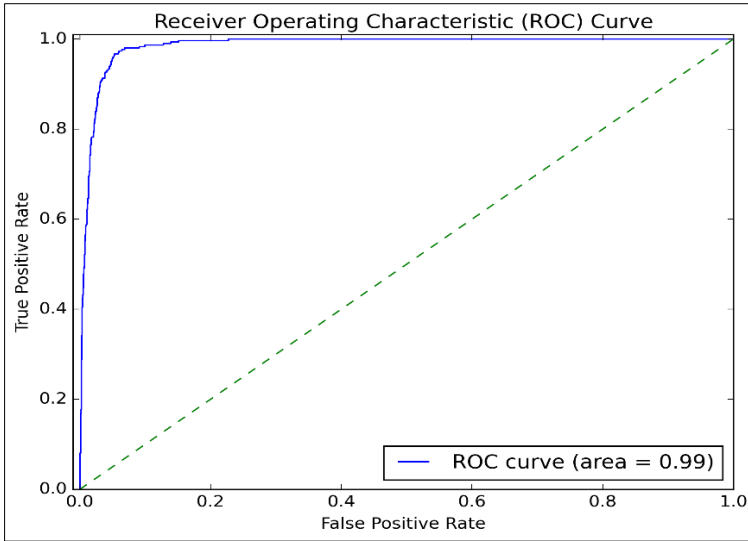
Deneyler esnasında 256 nöron kullanılmıştır. 6 ESA mimarisi arasında RMSprop optimizasyon algoritmasının en iyi sonucu veren optimizasyon algoritması olmuştur. En yüksek doğruluk, VGG-16 ESA mimarisi RMSprop 256 nöron deneyinde ahşap kusur tespitinde %97,47 doğrulukla başarı elde edilmiştir. ESA modellerine ve optimizasyon algoritmalarına göre doğruluk değerleri Şekil 8.1'de verilmiştir. Xception ESA mimarisi RMSprop algoritması karmaşıklık matrisi Şekil 8.2'de, ROC eğrisi Şekil 8.3'de verilmiştir.



Şekil 8.1. ESA modellerine ve optimizasyon algoritmalarına göre doğruluk değerleri



Şekil 8.2. XCeption ESA mimarisi RMSprop optimizasyonu karmaşıklık matrisi



Şekil 8.3. XCeption ESA mimarisi RMSprop optimizasyonu ROC eğrisi

Çizelge 8.1’de elde edilen sonuçlara göre belirlenen en başarılı performansı gösteren o optimizasyon algoritması kullanılarak, deneyler 256, 512, 1024 ve 2048 nöron olmak üzere dört farklı değişken nöron sayısı ile yeni deneyler yapılmıştır. Çizelge 8.2. ESA mimarileri, RMSprop optimizasyon algoritması ve nöron sayısı performansları verilmiştir. Yapılan deneylerin sonuçlarına göre elde edilen başarılı sonuçlar verilmiş ve derin öğrenme ile ahşap yüzey kusur tespitine yönelik önerilerde bulunulmuş ardından en başarılı sonuçların karmaşıklık matrisleri ve ROC eğrileri grafikleri verilmiştir. Nöron

sayıların ESA mimarileri ve optimizasyon algoritmalarına katkıları incelendiğinde elde edilen bulgular tartışılmıştır.

Çizelge 8.2. ESA mimarileri, RMSprop optimizasyon algoritması ve nöron sayısı performansları

ESA Model	Optimizasyon Algoritması / nöron sayısı	Kesinlik	Hassasiyet	F1-skoru	AUC	Süre	Doğruluk
VGG16	RMSprop / 256	0,9754	0,9747	0,9750	0,9914	309,19 s	0,9747
VGG16	RMSprop / 512	0,9730	0,9730	0,9730	0,9905	308,68 s	0,9730
VGG16	RMSprop / 1024*	0,9754	0,9750	0,9752	0,9920	309,74 s	0,9750*
VGG16	RMSprop / 2048	0,9729	0,9724	0,9726	0,9906	308,78 s	0,9724
MobileNet	RMSprop / 256	0,9749	0,9737	0,9742	0,9880	156,01 s	0,9737*
MobileNet	RMSprop / 512	0,9704	0,9698	0,9700	0,9813	145,78 s	0,9698
MobileNet	RMSprop / 1024	0,9722	0,9704	0,9711	0,9883	145,86 s	0,9704
MobileNet	RMSprop / 2048	0,9751	0,9734	0,9740	0,9910	147,48 s	0,9734
ResNet50	RMSprop / 256*	0,9740	0,9734	0,9736	0,9920	258,28 s	0,9734*
ResNet50	RMSprop / 512	0,9628	0,9632	0,9630	0,9885	258,92 s	0,9632
ResNet50	RMSprop / 1024	0,9741	0,9724	0,9730	0,9910	259,27 s	0,9724
ResNet50	RMSprop / 2048	0,9726	0,9717	0,9721	0,9906	259,00 s	0,9717
DenseNet121	RMSprop / 256	0,9653	0,9648	0,9651	0,9878	267,61 s	0,9648
DenseNet121	RMSprop / 512	0,9697	0,9684	0,9689	0,9893	266,79 s	0,9684
DenseNet121	RMSprop / 1024	0,9734	0,9724	0,9728	0,9909	267,34 s	0,9724
DenseNet121	RMSprop / 2048*	0,9748	0,9734	0,9739	0,9915	267,08 s	0,9734*
Xception	RMSprop / 256	0,9756	0,9756	0,9734	0,9911	383,37 s	0,9724
Xception	RMSprop / 512*	0,9769	0,9757	0,9761	0,9870	383,81 s	0,9757*
Xception	RMSprop / 1024	0,9755	0,9753	0,9754	0,9923	383,54 s	0,9753
Xception	RMSprop / 2048	0,9636	0,9642	0,9639	0,9737	383,84 s	0,9642
InceptionV3	RMSprop / 256	0,9732	0,9717	0,9723	0,9915	198,71 s	0,9717
InceptionV3	RMSprop / 512	0,9737	0,9714	0,9722	0,9928	198,73 s	0,9714
InceptionV3	RMSprop / 1024	0,9706	0,9691	0,9697	0,9871	198,59 s	0,9691
InceptionV3	RMSprop / 2048*	0,9720	0,9724	0,9721	0,9901	199,45 s	0,9724*

Not: "\*" sembolü en başarılı sonucu ifade etmek için kullanılmıştır.

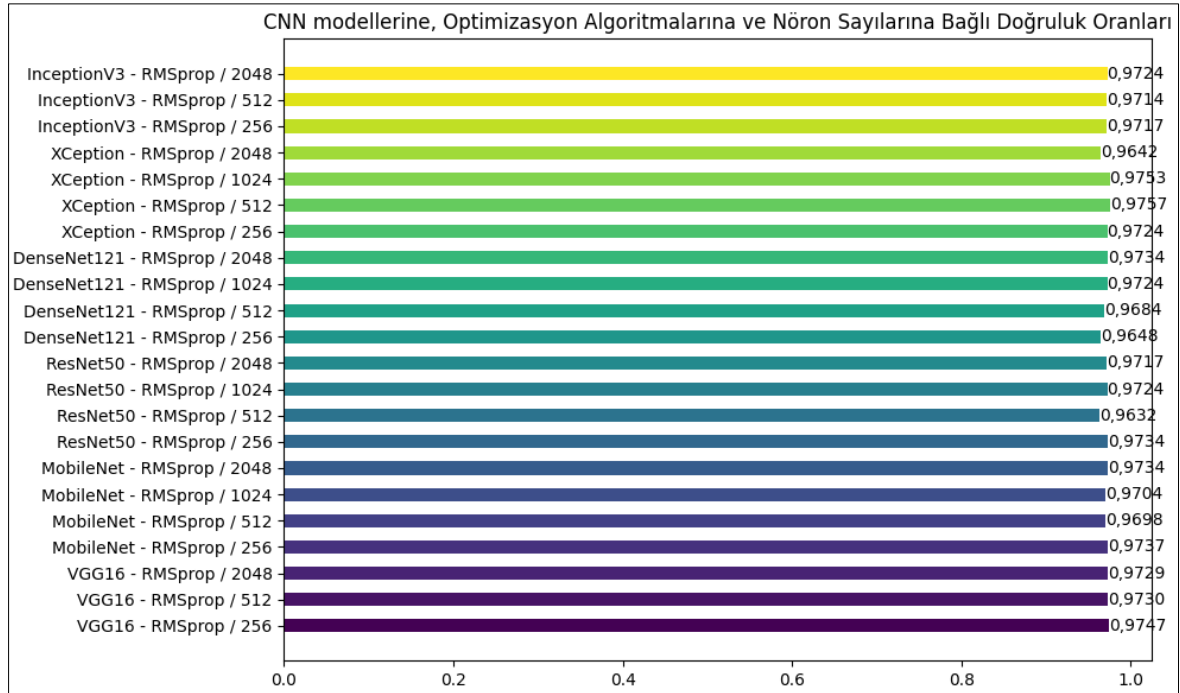
RMSprop algoritması kullanılarak farklı nöron sayıları ile yapılan deneylerde VGG16 1024 nöron ile %97,50 doğruluk, MobileNet 256 nöron ile %97,37 doğruluk, Resnet50 ile 256 nöron ile %97,34 doğruluk, DenseNet121 2048 nöron ile %97,34 Inceptionv3 2048 nöron ile %97,24 doğruluk bulunmuştur. ResNet50 mimarisini 256 nöronla artırmak doğruluğu artırmamıştır. Xception mimarisi, 512 nöronda en yüksek doğruluğa ulaştı. Densenet121 ve InceptionV3 mimarilerinde en yüksek doğruluk 2048 nöronda elde edilir. Tüm deneyler arasında, Xception mimarisi RMSprop optimizasyon algoritması 512 nöron ile %97,57 ile en yüksek doğruluk sonucunu vermiştir.

Süre bakımından incelendiğinde MobileNet mimarileri diğer ESA mimarilerine göre oldukça hızlı sürede hemen hemen aynı doğruluk derecesinde ahşap yüzey kusuru yapabildiği görülmektedir.

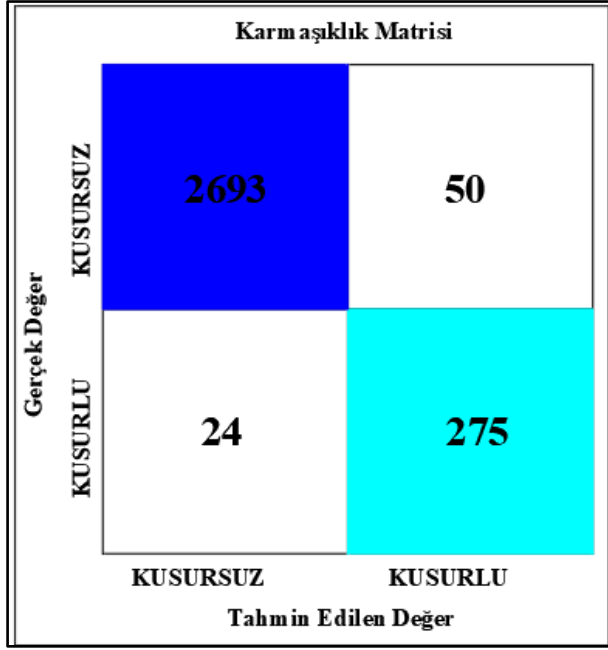
Xception mimarisi %97,57 doğruluğa ulaşsa da eğitim ve test süresi MobileNet mimarisinin 2 katından fazladır. MobileNet mimarisi %97,37 doğruluğa ulaşmaktadır ve

bu da en kısa sürede en yüksek sonuca yakındır. Aynı zamanda zaman açısından düşük olan InceptionV3 mimarisi 2048 nöron ile %97,24 doğruluğa ulaşmaktadır. Kesinlik ve doğruluk öncelikli tercih olacak ise Xception mimarisi RMSprop optimizasyon algoritması 512 ile model önerilebilir. Ancak süre önemli ise MobileNet mimarisi RMSprop optimizasyon algoritması 512 nöronla olan ESA modeli önerilmektedir.

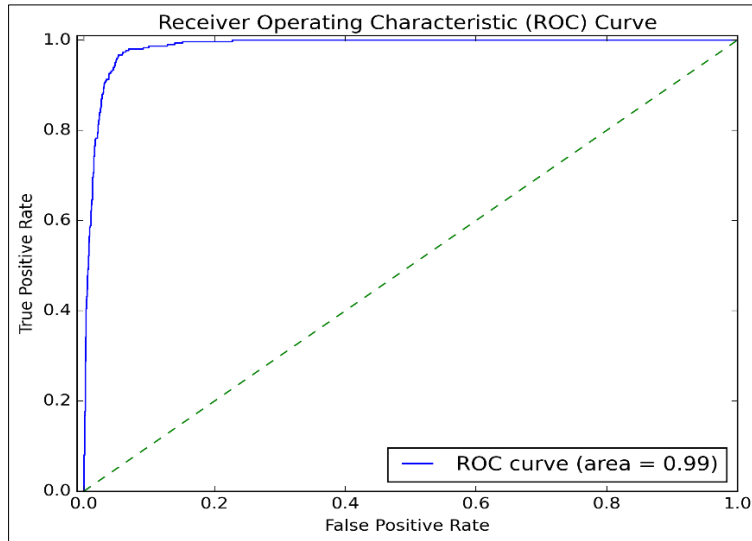
Çizelge. 8.2'ye göre tüm değerlendirme metriklerine göre en başarılı sonuç veren 3 ESA mimarileri şunlardır: Xception mimarisi RMSprop optimizasyon algoritması 512 nöron, VGG16 mimarisi RMSprop 1024 nöron, VGG16 mimarisi RMSprop 256 nöron ile bulunmuştur. Burada doğruluk metriği başta olmak üzere diğer metriklerle göre değerlendirme yapılmıştır. ESA modellerine, optimizasyon algoritmalarına ve nöron sayılarına bağlı doğruluk oranları Şekil 8.4'te verilmiştir. En başarılı Xception mimarisi RMSprop optimizasyon algoritması 512 nöron olan ESA mimarilerine ait karmaşıklık matrisi Şekil 8.5'de verilmiştir. En başarılı olan ESA modelin ROC eğrileri Şekil 8.6' de verilmiştir.



Şekil 8.4. ESA modellerine, optimizasyon algoritmalarına ve nöron sayılarına bağlı doğruluk oranları



Şekil 8.5. Xception ESA mimarisi RMSprop 512 nöron karmaşıklık matrisi



Şekil 8.6. Xception ESA mimarisi RMSprop 512 nöron ROC eğrisi

## 8.2. Klasik Makine Öğrenme Yöntemleri ile Ahşap Yüzey Kusur Tespiti

Bu bölümde kusur tespiti ile ilgili yapılan özellik çıkarımı ve sınıflandırma algoritmalarının göstermiş oldukları performanslar verilmiş ve yorumlanmıştır.

Ahşap kusur tespitinde LBP öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları Çizelge 8.3'te verilmiştir.

Çizelge 8.3. LBP öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları

Algoritma	AUC	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk
LBP RandomForestClassifier	0,9851	0,9680	0,9675	0,9675	0,9675*
LBP KNeighborsClassifier	0,9198	0,9217	0,9199	0,9198	0,9199
LBP Support Vector Machine	0,8737	0,8701	0,8388	0,8353	0,8388
LBP DecisionTreeClassifier	0,9048	0,9081	0,9050	0,9047	0,9050
LBP Naive Bayes	0,8484	0,8319	0,8063	0,8025	0,8063
LBP LogisticRegression	0,8595	0,8504	0,8226	0,8190	0,8226
LBP GradientBoostingClassifier	0,9839	0,9661	0,9656	0,9656	0,9656
LBP XGBClassifier	0,9829	0,9631	0,9628	0,9628	0,9628
LBP LightGBM	0,9750	0,9289	0,9262	0,9261	0,9262
LBP CatBoost	0,9816	0,9546	0,9539	0,9539	0,9539
LBP AdaBoost	0,8966	0,8642	0,8398	0,8370	0,8398
LBP MLPClassifier	0,9048	0,8729	0,8458	0,8429	0,8458

Not: "\*" sembolü en başarılı sonucu ifade etmek için kullanılmıştır.

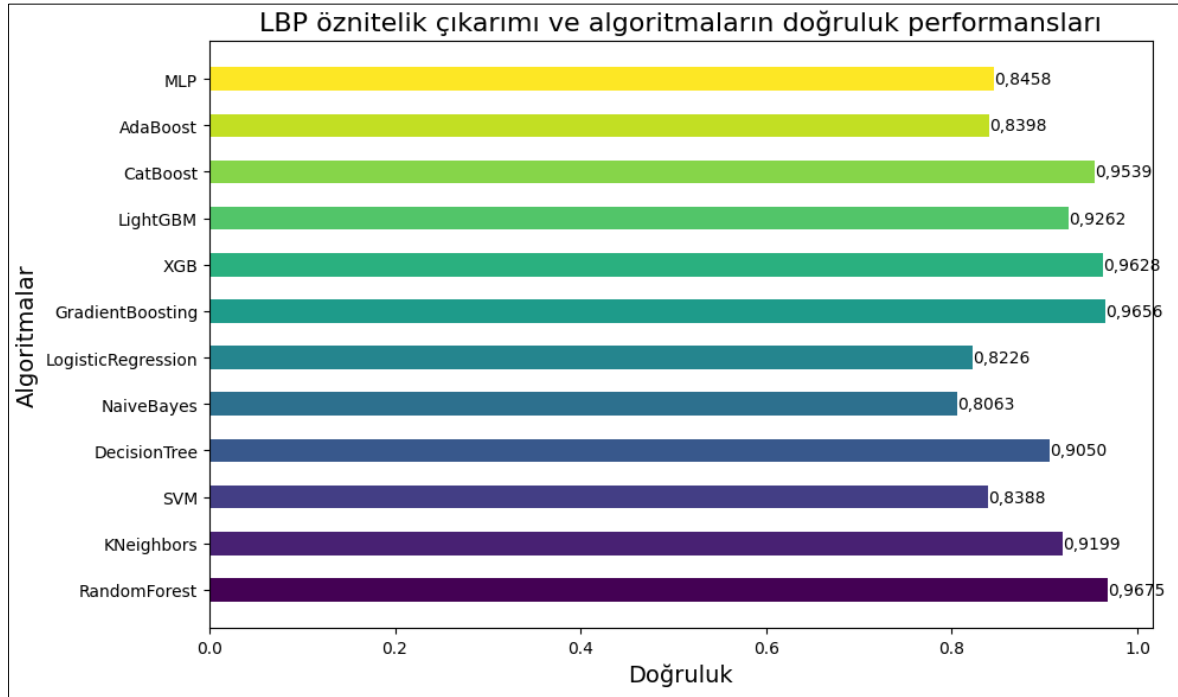
Performans metrikleri açısından en iyi sonuçları veren algoritmanın Random Forest Classifier olduğunu gözlemliyoruz. Bu algoritma, %96,75 doğruluk oranına sahiptir. Diğer tüm metriklerde yüksek performans göstermektedir. Özellikle %98,51 AUC değeri oldukça yüksektir, bu da modelin sınıflandırma yeteneğinin genel olarak çok iyi olduğunu göstermektedir.

Gradient Boosting Classifier da %96,56 doğruluk oranı ahşap kusur tespitinde sınıflandırma başarısı göstermektedir. Random Forest Classifier ile hemen hemen aynı oranda ahşap kusur tespitinde başarı göstermektedir.

XGB Classifier %96,28 oranında kusur tespitinde başarı göstermiştir. Random Forest Classifier ve Gradient Boosting Classifier ile aynı düzeye yakın yüzdeler dilim olarak bakılacak olursa %96 doğruluk oranı ile görüntülerden öznitelik çıkarımı ve klasik makine öğrenme algoritmaları ile sınıflandırmada başarı elde edilmiştir. En başarılı sınıflandırma değeri olan %96,75 doğruluk oranı ile LBP ile öznitelik çıkarımı Random Forest Classifier ile sınıflandırmada yaklaşık %97 oranında ahşap kusur tespiti yapılabilir olduğu deneyler sonucunda elde edilmiştir.

Kesinlik, hassasiyet ve F1-Skoru metriklerine baktığımızda, Random Forest Classifier ve Gradient Boosting Classifier modellerinin biraz daha avantajlı olduğu görülmektedir. Ancak bu üç makine öğrenme algoritması LBP ile öznitelik çıkarımı yapıldığında yüksek doğruluk oranı ile kusur tespiti sınıflandırması yapabilmektedir.

LBP öznitelik çıkarımı ve algoritmaların doğruluk performansları Şekil 8.7’de verilmiştir.



Şekil 8.7. LBP öznitelik çıkarımı ve algoritmaların doğruluk performansları

HOG öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları Çizelge 8.4’te verilmiştir.

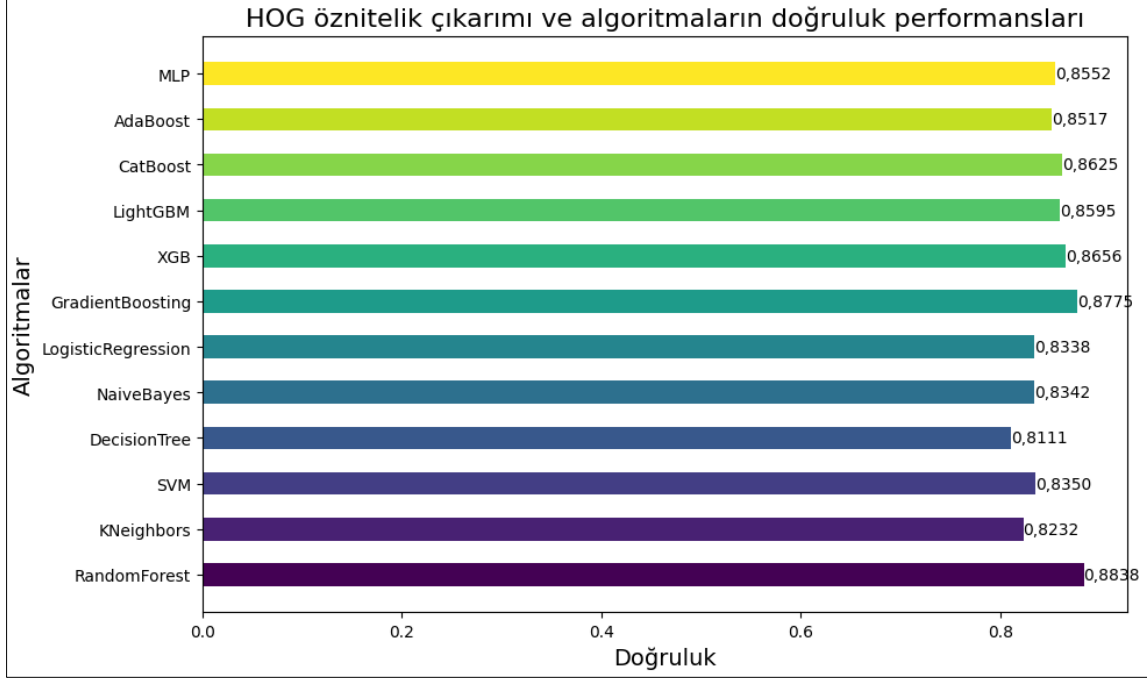
Çizelge 8.4. HOG öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları

Algoritma	AUC	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk
HOG RandomForestClassifier	0,9198	0,8903	0,8838	0,8834	0,8838*
HOG KNeighborsClassifier	0,8232	0,8234	0,8232	0,8232	0,8232
HOG Support Vector Machine	0,8496	0,8451	0,8350	0,8338	0,8350
HOG DecisionTreeClassifier	0,8110	0,8118	0,8111	0,8110	0,8111
HOG Naive Bayes	0,8735	0,8433	0,8342	0,8331	0,8342
HOG LogisticRegression	0,8488	0,8432	0,8338	0,8326	0,8338
HOG GradientBoostingClassifier	0,9138	0,8839	0,8775	0,8770	0,8775
HOG XGBClassifier	0,9124	0,8760	0,8656	0,8646	0,8656
HOG LightGBM	0,9084	0,8726	0,8595	0,8582	0,8595
HOG CatBoost	0,9111	0,8750	0,8625	0,8614	0,8625
HOG AdaBoost	0,8918	0,8696	0,8517	0,8499	0,8517
HOG MLPClassifier	0,8940	0,8764	0,8552	0,8531	0,8552

Not: “\*” sembolü en başarılı sonucu ifade etmek için kullanılmıştır.

HOG ile öznitelik çıkarımı Random Forest Classifier algoritması ile sınıflandırmada doğruluk değeri bakımından %88,38 doğrulukla kusur tespiti yapabilmıştır.

HOG öznitelik çıkarımı ve algoritmaların doğruluk performansları Çizelge 8.8’de verilmiştir.



Şekil 8.8. HOG öznitelik çıkarımı ve algoritmaların doğruluk performansları

Çizelge 8.5’te GLCM öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları verilmiştir.

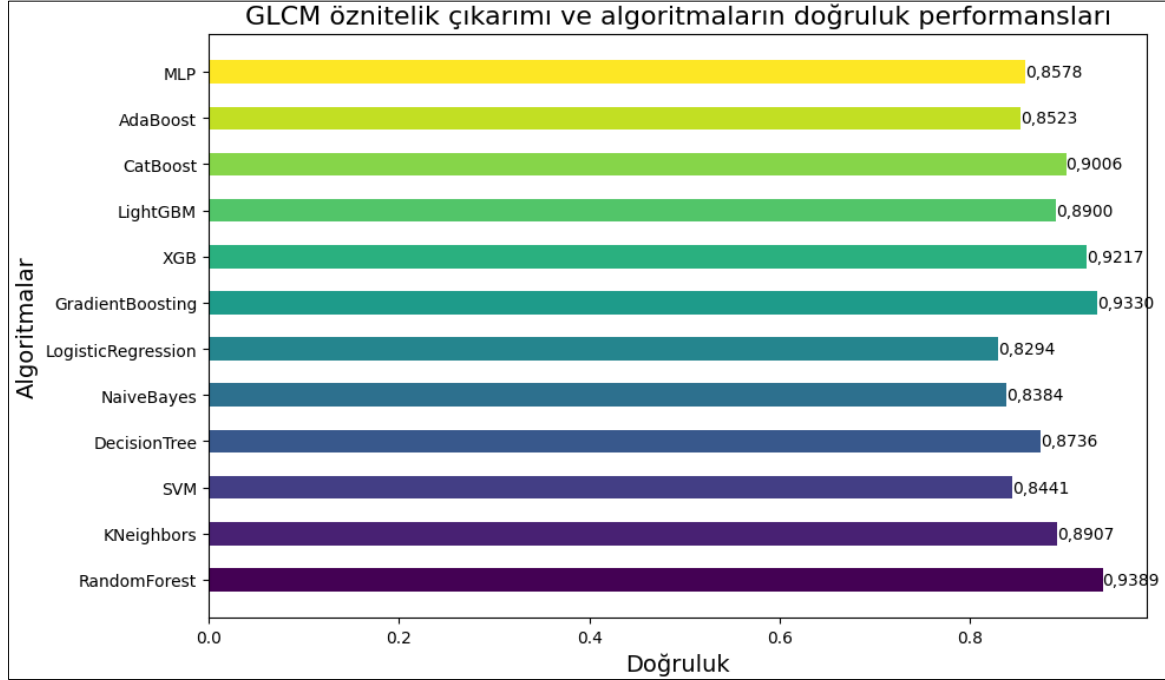
Çizelge 8.5. GLCM öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları

Algoritma	AUC	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk
GLCM RandomForestClassifier	0,9653	0,9415	0,9389	0,9388	0,9389*
GLCM KNeighborsClassifier	0,8906	0,8918	0,8907	0,8906	0,8907
GLCM Support Vector Machine	0,8866	0,8762	0,8441	0,8407	0,8441
GLCM DecisionTreeClassifier	0,8735	0,8759	0,8736	0,8733	0,8736
GLCM Naive Bayes	0,8831	0,8557	0,8334	0,8362	0,8384
GLCM LogisticRegression	0,8729	0,8510	0,8294	0,8267	0,8294
GLCM GradientBoostingClassifier	0,9609	0,9352	0,9330	0,9330	0,9330
GLCM XGBClassifier	0,9592	0,9250	0,9217	0,9215	0,9217
GLCM LightGBM	0,9499	0,8993	0,8900	0,8894	0,8900
GLCM CatBoost	0,9546	0,9075	0,9006	0,9002	0,9006
GLCM AdaBoost	0,9034	0,8775	0,8523	0,8498	0,8523
GLCM MLPClassifier	0,9044	0,8808	0,8578	0,8557	0,8578

Not: “\*” sembolü en başarılı sonucu ifade etmek için kullanılmıştır.

GLCM Random Forest Classifier %93,89 ve Gradient Boosting Classifier %93,30 algoritmaları doğruluk seviyelerine sahiptir.

GLCM öznitelik çıkarımı ve algoritmaların doğruluk performansları Çizelge 8.9’da verilmiştir.



Şekil 8.9. GLCM öznitelik çıkarımı ve algoritmaların doğruluk performansları

SOBEL öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları Çizelge 8.6’da verilmiştir.

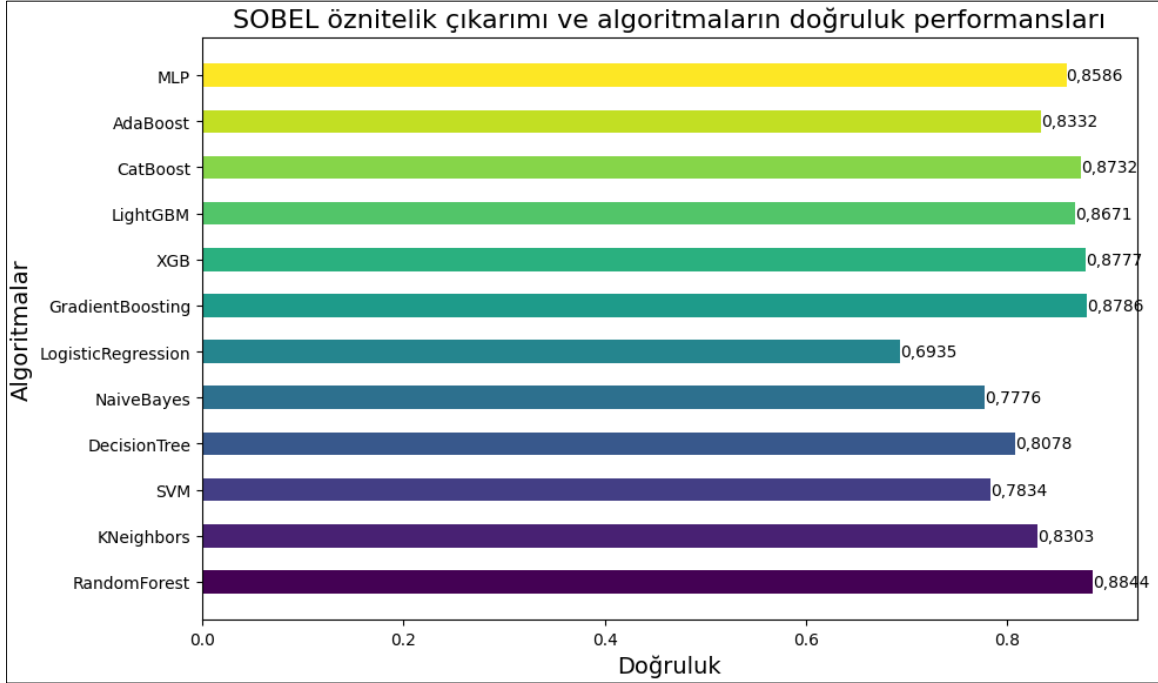
Çizelge 8. 6. SOBEL öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları

Algoritma	AUC	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk
SOBEL RandomForestClassifier	0,9326	0,8899	0,8844	0,8840	0,8844*
SOBEL KNeighborsClassifier	0,8303	0,8304	0,8303	0,8303	0,8303
SOBEL Support Vector Machine	0,8209	0,8140	0,7834	0,7730	0,7834
SOBEL DecisionTreeClassifier	0,8078	0,8081	0,8078	0,8078	0,8078
SOBEL Naive Bayes	0,8424	0,8156	0,7776	0,7706	0,7776
SOBEL LogisticRegression	0,7248	0,7172	0,6935	0,6868	0,6935
SOBEL GradientBoostingClassifier	0,9256	0,8838	0,8786	0,8782	0,8786
SOBEL XGBClassifier	0,9264	0,8834	0,8777	0,8773	0,8777
SOBEL LightGBM	0,9226	0,8750	0,8671	0,8664	0,8671
SOBEL CatBoost	0,9271	0,8803	0,8732	0,8726	0,8732
SOBEL AdaBoost	0,8880	0,8502	0,8332	0,8311	0,8332
SOBEL MLPClassifier	0,9104	0,8754	0,8586	0,8571	0,8586

Not: “\*” sembolü en başarılı sonucu ifade etmek için kullanılmıştır.

SOBEL Random Forest Classifier %88,44 en yüksek doğruluk oranına sahiptir. Gradient Boosting Classifier %87,86 ve XGB Classifier %87,77 doğruluk oranına sahiptirler.

SOBEL öznitelik çıkarımı ve algoritmaların doğruluk performansları Şekil 8.10'da verilmiştir.



Şekil 8.10. SOBEL öznitelik çıkarımı ve algoritmaların doğruluk performansları

GABOR özellik çıkarımı ve sınıflandırma algoritmalarının performansları Çizelge 8.7'de verilmiştir.

Çizelge 8. 7. GABOR öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları

Algoritma	AUC	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk
GABOR RandomForestClassifier	0,9753	0,9521	0,9496	0,9495	0,9496
GABOR KNeighborsClassifier	0,9011	0,9022	0,9012	0,9011	0,9012
GABOR Support Vector Machine	0,9048	0,8861	0,8622	0,8600	0,8622
GABOR DecisionTreeClassifier	0,8866	0,8880	0,8866	0,8865	0,8866
GABOR Naive Bayes	0,8933	0,8766	0,8580	0,8562	0,8580
GABOR LogisticRegression	0,8918	0,8789	0,8586	0,8566	0,8586
GABOR GradientBoostingClassifier	0,9765	0,9543	0,9523	0,9522	0,9523*
GABOR XGBClassifier	0,9756	0,9486	0,9460	0,9459	0,9460
GABOR LightGBM	0,9658	0,9216	0,9146	0,9142	0,9146
GABOR CatBoost	0,9704	0,9318	0,9265	0,9262	0,9265
GABOR AdaBoost	0,9157	0,8954	0,8770	0,8756	0,8770
GABOR MLPClassifier	0,9178	0,8989	0,8802	0,8788	0,8802

Not: "\*" sembolü en başarılı sonucu ifade etmek için kullanılmıştır.

GABOR Gradient Boosting Classifier %95,23 doğruluk, Random Forest Classifier %94,96, XGB Classifier %94,60 doğruluk oranı vermektedir.

GABOR öznitelik çıkarımı ve algoritmaların doğruluk performansları Şekil 8.11’de verilmiştir.



Şekil 8.11. GABOR öznitelik çıkarımı ve algoritmaların doğruluk performansları

MB-LBP öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları Çizelge 8.8’de verilmiştir.

Çizelge 8.8. MB-LBP öznitelik çıkarımı ve sınıflandırma algoritmalarının performansları

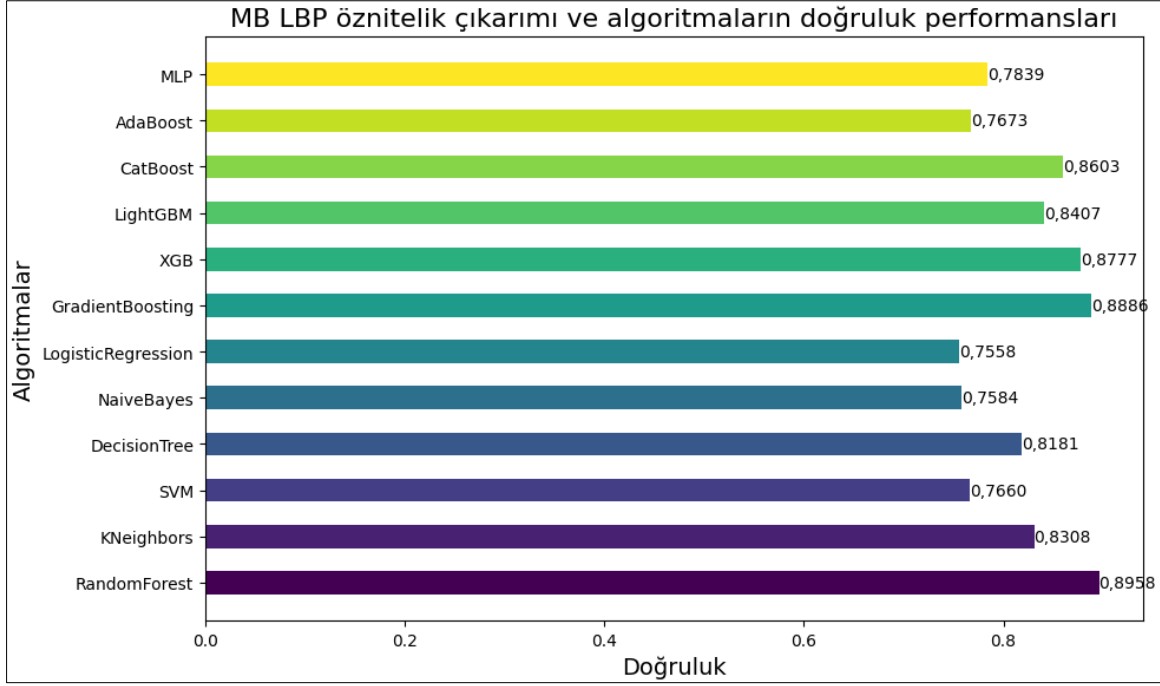
Algoritma	AUC	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk
MB LBP RandomForestClassifier	0,9397	0,8984	0,8958	0,8956	0,8958*
MB LBP KNeighborsClassifier	0,8307	0,8321	0,8308	0,8306	0,8308
MB LBP Support Vector Machine	0,7983	0,8025	0,7660	0,7586	0,7660
MB LBP DecisionTreeClassifier	0,8181	0,8201	0,8101	0,8178	0,8181
MB LBP Naive Bayes	0,7988	0,7961	0,7584	0,7504	0,7584
MB LBP LogisticRegression	0,7995	0,7742	0,7558	0,7516	0,7558
MB LBP GradientBoostingClassifier	0,9335	0,8907	0,8886	0,8844	0,8886
MB LBP XGBClassifier	0,9294	0,8802	0,8777	0,8775	0,8777
MB LBP LightGBM	0,9121	0,8486	0,8407	0,8398	0,8407
MB LBP CatBoost	0,9240	0,8638	0,8603	0,8600	0,8603
MB LBP AdaBoost	0,8365	0,7990	0,7673	0,7609	0,7673
MB LBP MLPClassifier	0,8575	0,8098	0,7893	0,7794	0,7839

Not: “\*” sembolü en başarılı sonucu ifade etmek için kullanılmıştır.

MB-LBP Random Forest Classifier, %89,58 doğruluk oranı ile en yüksek doğruluk oranına sahiptir. MB-LBP öznitelikleri üzerinde yapılan sınıflandırmada, doğruluk metriği de dikkate alındığında, Gradient Boosting Classifier %88,86 doğruluk oranına sahiptir. XGB

Classifier %87,77 doğruluk ve CatBoost %86,03 doğruluk ile yüksek doğruluk oranlarına sahiptirler. LightGBM, %84,07 doğruluk ile tatmin edici bir performans sunmaktadır.

MB-LBP öznitelik çıkarımı ve algoritmaların doğruluk performansları Şekil 8.12’de verilmiştir.



Şekil 8.12. MB-LBP öznitelik çıkarımı ve algoritmaların doğruluk performansları

Çalışma kapsamında klasik makine öğrenme yöntemleri ile sınıflandırmada elde edilen tüm öznitelik çıkarımları birleştirilmek sureti ile hibrit bir yaklaşımın gösterdiği performansta tez kapsamında değerlendirilmiştir. 6 farklı öznitelik çıkarım yönteminde elde edilen ve seçilmiş olan sınıflandırmalarda kullanılan tüm özellikler yan yana getirilerek hibrit özellikler oluşturulmuş ve sınıflandırma algoritmaları ile sınıflandırılmış, gösterdiği performanslar incelenmiştir.

Hibrit öznitelikler ile sınıflandırma algoritmalarının performansları Çizelge 8.9’da verilmiştir.

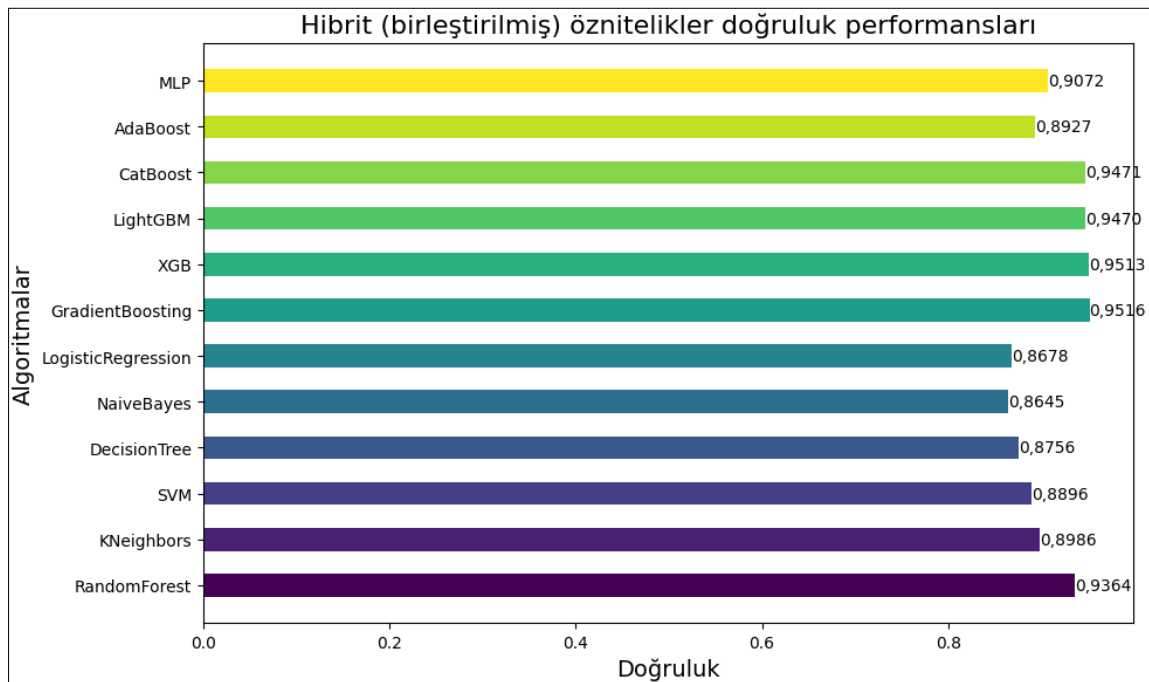
Çizelge 8.9. Hibrit öznelikler ile sınıflandırma algoritmalarının performansları

Algoritma	AUC	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk
RandomForestClassifier	0.9778	0.9420	0.9364	0.9362	0.9364
KNeighborsClassifier	0.8985	0.8986	0.8986	0.8985	0.8986
Support Vector Machine	0.9262	0.9087	0.8896	0.8883	0.8896
DecisionTreeClassifier	0.8755	0.8762	0.8756	0.8755	0.8756
Naive Bayes	0.8822	0.8843	0.8645	0.8627	0.8645
LogisticRegression	0.9036	0.8886	0.8678	0.8660	0.8678
GradientBoostingClassifier	0.9820	0.9545	0.9516	0.9515	0.9516*
XGBClassifier	0.9830	0.9538	0.9513	0.9513	0.9513
LightGBM	0.9811	0.9492	0.9470	0.9469	0.9470
CatBoost	0.9800	0.9504	0.9471	0.9470	0.9471
AdaBoost	0.9489	0.9058	0.8927	0.8918	0.8927
MLPClassifier	0.9581	0.9168	0.9072	0.9068	0.9072

Not: "\*" sembolü en başarılı sonucu ifade etmek için kullanılmıştır.

Birleştirilmiş (hibrit) özellikler de en yüksek doğruluğu %95,16 doğruluk ile Gradient Boosting Classifier vermektedir. XGBClassifier %95,13 ile oldukça başarılı sonuç vermektedir. LightGBM %94,70 doğruluk oranı, CatBoost %94,71 doğruluk vermiştir. Hibrit öğrenme bazı özellik çıkarımlarına göre doğruluklarda artırma sağladığı bazı algoritmalarda da doğruluk performanslarında azalma yaptığı görülmüştür.

Hibrit (birleştirilmiş) öznelik çıkarımı ve algoritmaların doğruluk performansları Şekil 8.13'te verilmiştir.

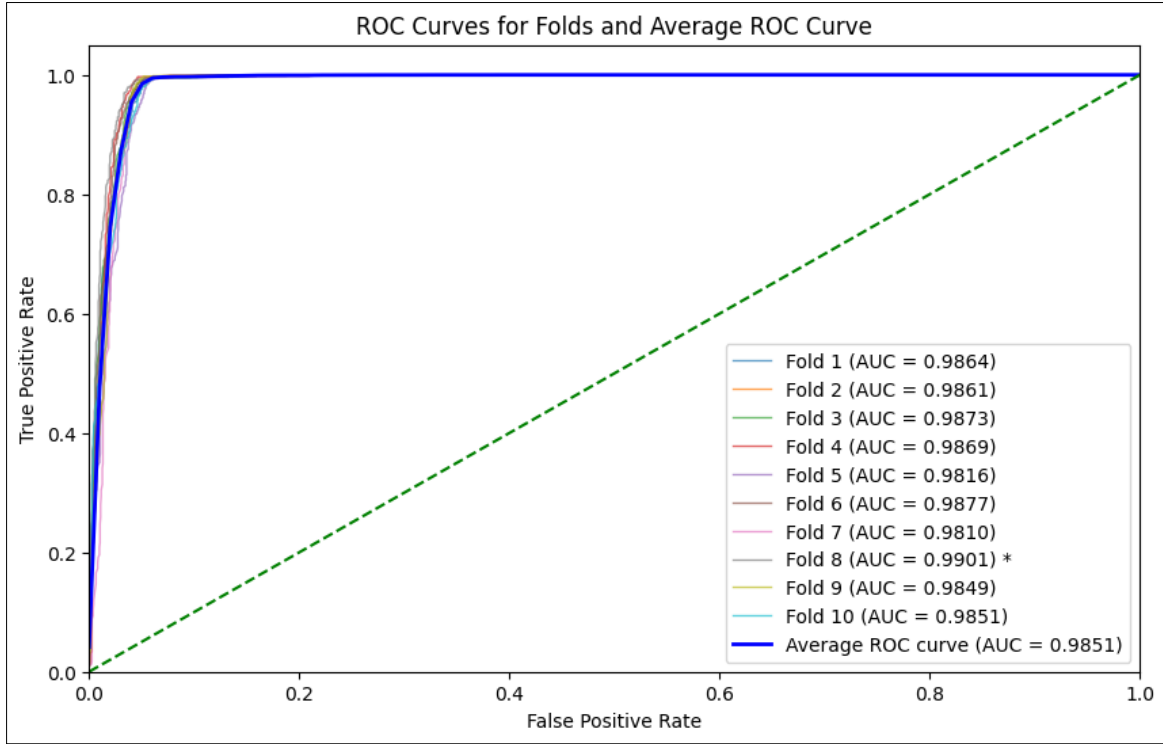


Şekil 8.13. Hibrit öznelik çıkarımı ve algoritmaların doğruluk performansları

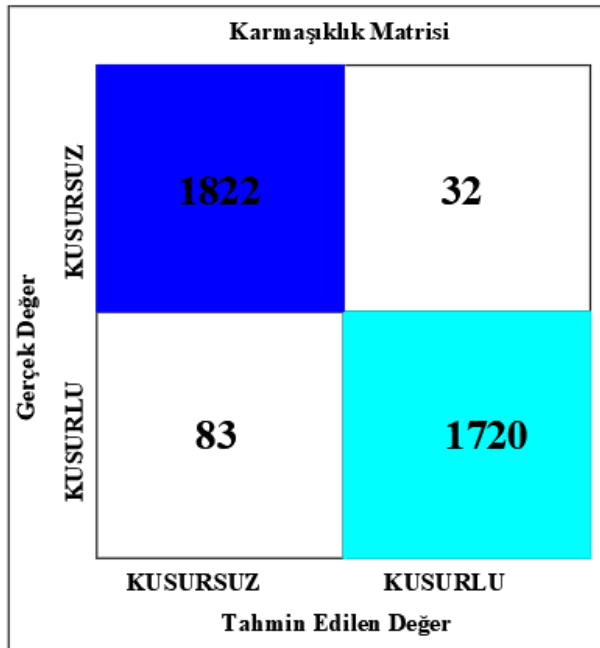
Ahşap yüzey görüntülerinden öznelik çıkarımı ve klasik makine öğrenme algoritmalarıyla sınıflandırma da en iyi sonuçları veren LBP ile öznelik çıkarım yöntemidir. LBP ile öznelik çıkarımında en başarılı algoritmalar; Random Forest Classifier, Gradient Boosting Classifier ve XGB Classifier'dir. LBP öznelik çıkarımı Random Forest Classifier'a ait karmaşıklık matrisi Şekil 8.14'te, ROC eğrisi Şekil 8.15'te verilmiştir. LBP öznelik çıkarımı Gradient Boosting Classifier'a ait karmaşıklık matrisi Şekil 8.16'da, ROC eğrisi Şekil 8.17'de verilmiştir. LBP ile öznelik çıkarımı XGB Classifier'a ait karmaşıklık matrisi Şekil 8.18'de, ROC eğrisi Şekil 8.19'da verilmiştir. Çalışmada %10 çapraz doğrulama yapıldığı için en iyi Fold ait karmaşıklık matrisleri ve ROC eğrileri verilmiştir.

		Karmaşıklık Matrisi	
		KUSURSUZ	KUSURLU
Gerçek Değer	KUSURSUZ	1828	26
	KUSURLU	77	1726
		Tahmin Edilen Değer	
		KUSURSUZ	KUSURLU

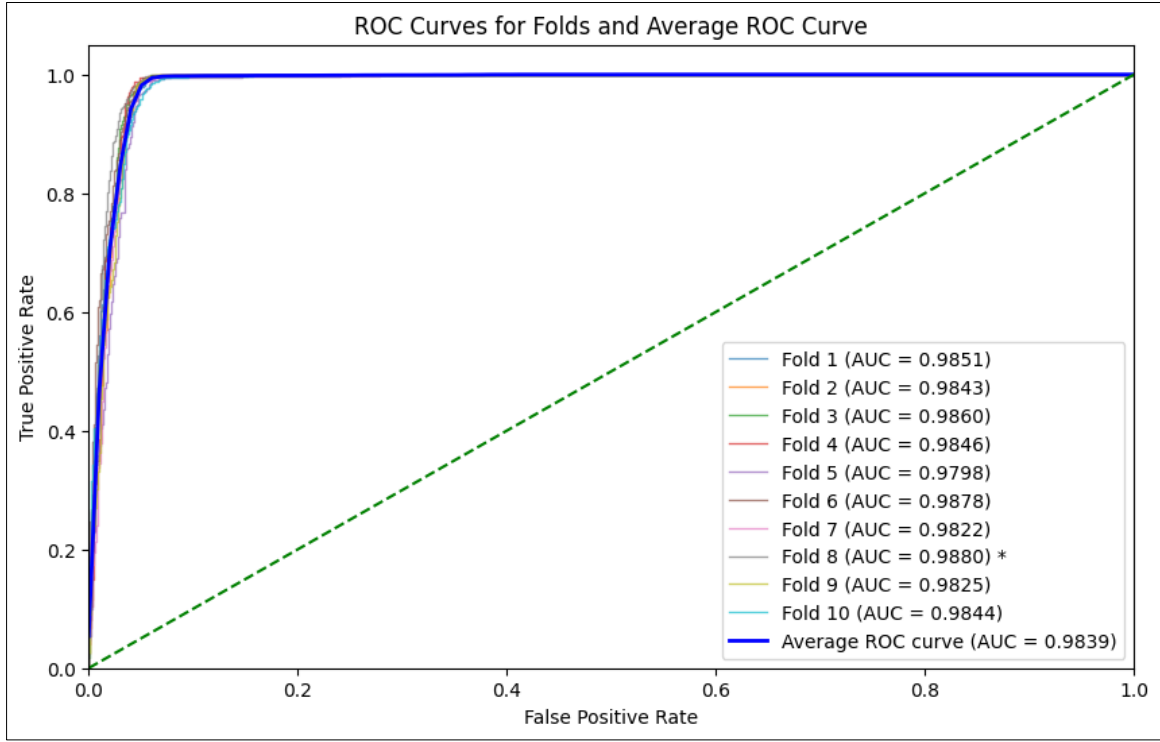
Şekil 8.14. LBP öznelik çıkarımı Random Forest Classifier karmaşıklık matrisi



Şekil 8.15. LBP öznitelik çıkarımı Random Forest Classifier ROC eğrisi



Şekil 8.16. LBP öznitelik çıkarımı Gradient Boosting Classifier karmaşıklık matrisi

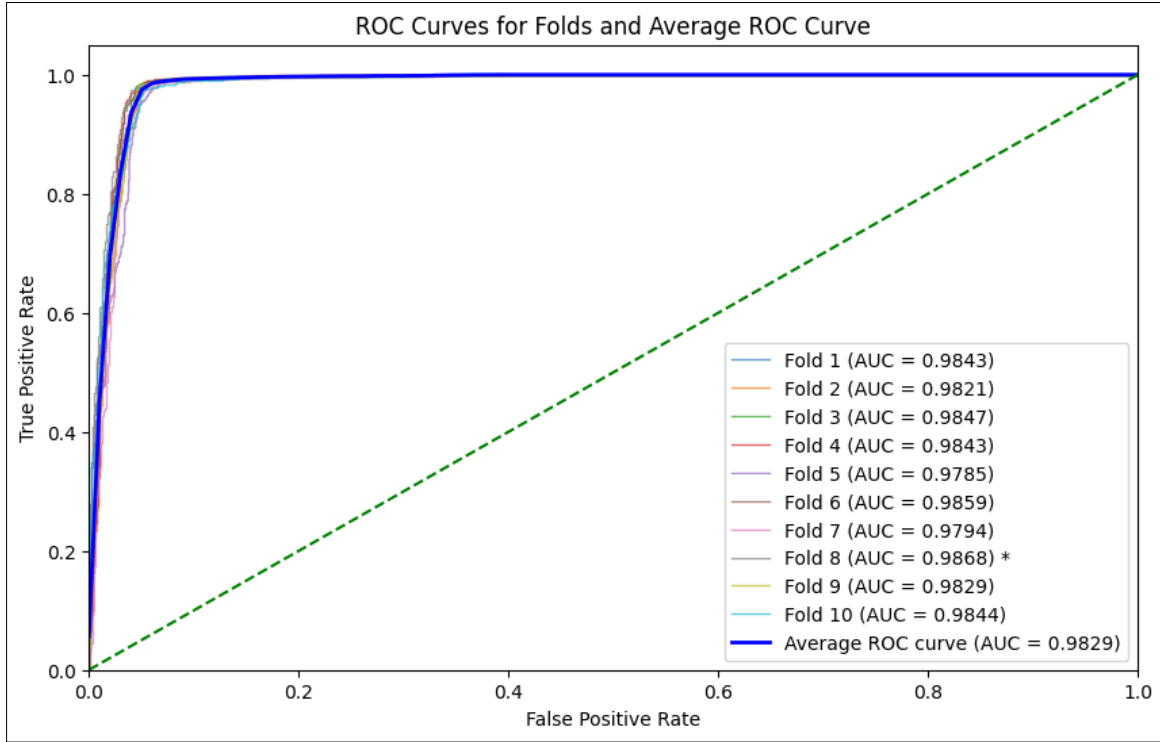


Şekil 8.17. LBP öznelik çıkarımı Gradient Boosting Classifier ROC eğrisi

**Karmaşıklık Matrisi**

Gerçek Değer	KUSURSUZ	1809	45
	KUSURLU	79	1724
		KUSURSUZ	KUSURLU
		Tahmin Edilen Değer	

Şekil 8.18. LBP öznelik çıkarımı XGB Classifier karmaşıklık matrisi



Şekil 8.19. LBP öznelik çıkarımı XGB Classifier ROC eğrisi

### 8.3. Hibrit Yaklaşım ile Ahşap Yüzey Kusur Tespiti

Hibrit yaklaşımla, derin öğrenmede kullanılan optimize edilmiş Xception ESA mimarisi RMSprop optimizasyon algoritması 512 nöron ile en yüksek değerde doğruluk edilmiştir. Burada elde edilen %97,57 doğruluk oranını daha yüksek seviyelere getirebilirliğinin araştırılması sonucu ESA mimarisinin çıkarmış olduğu öznelikler klasik makine öğrenme sınıflandırma algoritmaları ile sınıflandırma yapılarak hibrit yaklaşımla ahşap yüzey kusur tespiti deneyleri gerçekleştirilmiştir. ESA ile çıkarılmış özneliklerin normal ESA mimarisi yerine optimize edilmiş olması ve transfer öğrenme yaklaşımı ile model eğitilerek çıkarılmış öznelikler olduğu göz önüne alınmalıdır. Optimize edilmiş Xception mimarisinin sonuna eklenmiş batch normalizasyon katmanından oluşan 2048 öznelik özellik seçimi yapılmadan sınıflandırma algoritmaları ile sınıflandırma yapılmıştır. Hibrit yaklaşım ile sınıflandırma algoritmalarının performansları Çizelge 8.10'da verilmiştir. Sınıflandırma algoritmalarının performansları değerlendirmelerde kıyaslama sağlamak adına aynı hiper parametreler kullanılarak gerçekleştirilmiştir.

ESA ile çıkarılan özniteliklerin değerlendirilmesi yapılırken, kullanılan mimarinin normal bir ESA yerine optimize edilmiş bir model olduğu göz önünde bulundurulmalıdır. Optimize edilmiş ESA mimarisi, geleneksel mimarilere kıyasla daha verimli çalışarak daha güçlü öznitelikler çıkarma yeteneğine sahiptir. Bu optimizasyon süreci, modelin daha iyi genelleştirilmesini ve daha yüksek doğruluk oranlarına ulaşmasını sağlamaktadır.

Modelin eğitimi sırasında transfer öğrenme yaklaşımı kullanılmıştır. Bu yöntem sayesinde, model sıfırdan eğitilmek yerine önceden edinilmiş bilgi ile donatılarak daha hızlı ve etkili bir öğrenme süreci gerçekleştirilmiştir.

Sonuç olarak, çıkarılan özniteliklerin kalitesi, kullanılan ESA mimarisinin optimize edilmiş olması ve transfer öğrenme ile eğitilmesi sayesinde önemli ölçüde artırılmıştır. Bu sayede, elde edilen öznitelikler klasik makine öğrenme algoritmalarıyla daha başarılı bir şekilde sınıflandırılabilmiştir.

Çizelge 8.10. Hibrit yaklaşım ile sınıflandırma algoritmalarının performansları

Algoritma	AUC	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk
Hibrit RandomForestClassifier	0,9974	0,9931	0,9931	0,9931	0,9931
Hibrit KNeighborsClassifier	0,9885	0,9893	0,9894	0,9894	0,9894
Hibrit Support Vector Machine	0,9987	0,9926	0,9927	0,9926	0,9927
Hibrit DecisionTreeClassifier	0,9881	0,9886	0,9886	0,9886	0,9886
Hibrit Naive Bayes	0,9941	0,9923	0,9923	0,9923	0,9923
Hibrit LogisticRegression	0,9991	0,9925	0,9925	0,9925	0,9925
Hibrit GradientBoostingClassifier	0,9991	0,9931	0,9931	0,9931	0,9931
Hibrit XGBClassifier	0,9990	0,9928	0,9928	0,9928	0,9928
Hibrit LightGBM	0,9990	0,9928	0,9928	0,9928	0,9928
Hibrit CatBoost	0,9991	0,9932	0,9932	0,9932	0,9932
Hibrit AdaBoost	0,9946	0,9921	0,9922	0,9922	0,9922
Hibrit MLPClassifier	0,9981	0,9919	0,9919	0,9919	0,9919

Hibrit yaklaşımla sınıflandırma algoritmaları %98,81 ile %99,91 arasında yüksek AUC değerlerine sahiptir. Bu durum modellerin sınıflandırma yeteneklerinin genel olarak yüksek olduğunu göstermektedir.

Kesinlik değerleri genellikle yüksektir ve çoğu algoritma için %99 civarındadır, bu da algoritmaların pozitif tahminlerinin genellikle doğru olduğunu göstermektedir.

Hassasiyet değerleri de genellikle yüksektir ve %99 civarındadır, bu da algoritmaların gerçek pozitif tahminlerinin yüksek oranda doğru olduğunu göstermektedir.

F1-Skoru, hassasiyet ve kesinlik arasındaki dengeyi göstermektedir Tüm hibrit yaklaşımda kullanılan sınıflandırma algoritmaları yüksek %99,19 ile %99,32 F1-Skorlarına sahiptir.

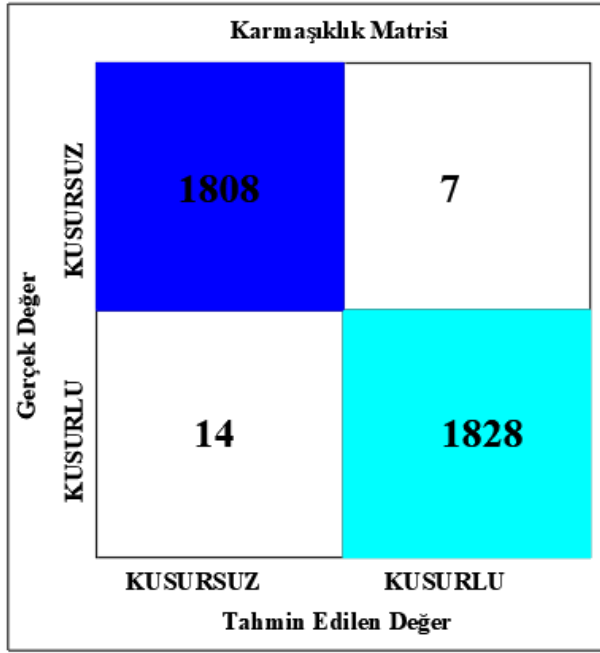
Doğruluk değerleri genellikle yüksektir %99,19 ile %99,32 arasında, bu da algoritmaların genel doğru sınıflandırma yeteneklerini göstermektedir.

Hibrit yaklaşım sınıflandırma doğruluk performansları Şekil 8.20' de verilmiştir.

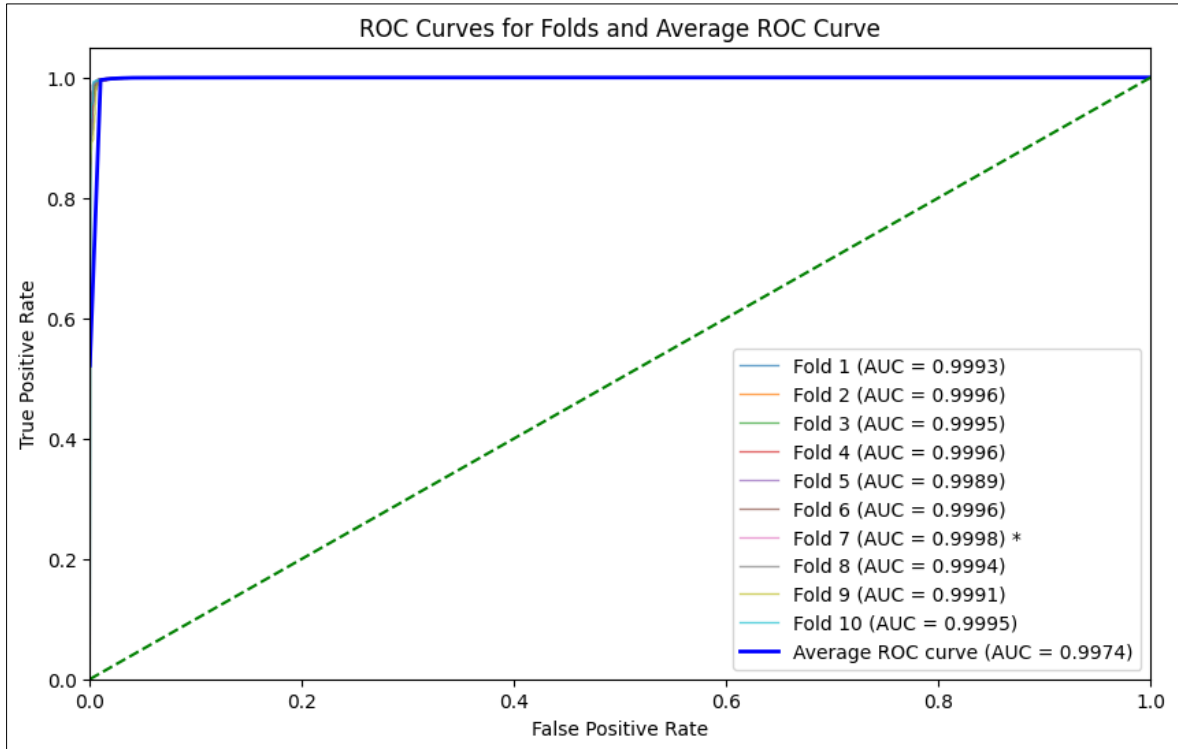


Şekil 8.20. Hibrit yaklaşım sınıflandırma doğruluk performansları

ESA ile öznelik çıkarımında tüm sınıflandırma algoritmaları %99 oranında sınıflandırma yaptığı ve birbirlerine yakın olduğu için sadece 3 en başarılı sınıflandırma algoritmasının karmaşıklık matrisi ve ROC eğrileri verilecektir. Hibrit yaklaşım Random Forest Classifier karmaşıklık matrisi Şekil 8. 21'de, ROC eğrisi Şekil 8.22'de verilmiştir.

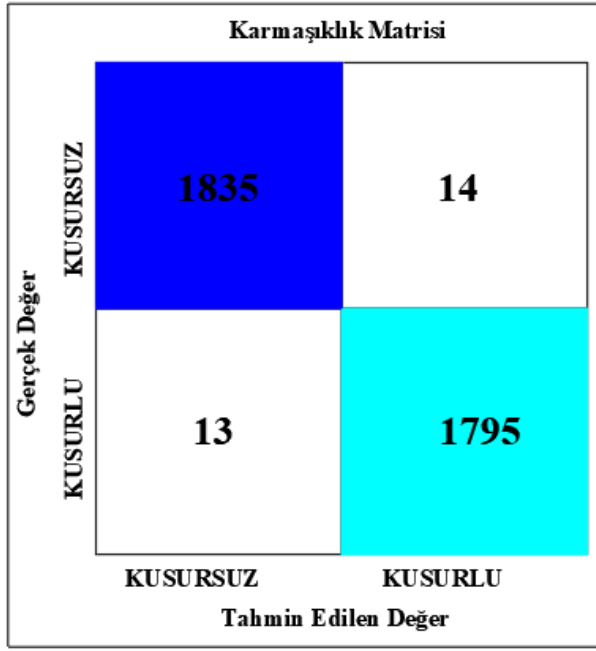


Şekil 8.21. Hibrit yaklaşım Random Forest Classifier karmaşıklık matrisi

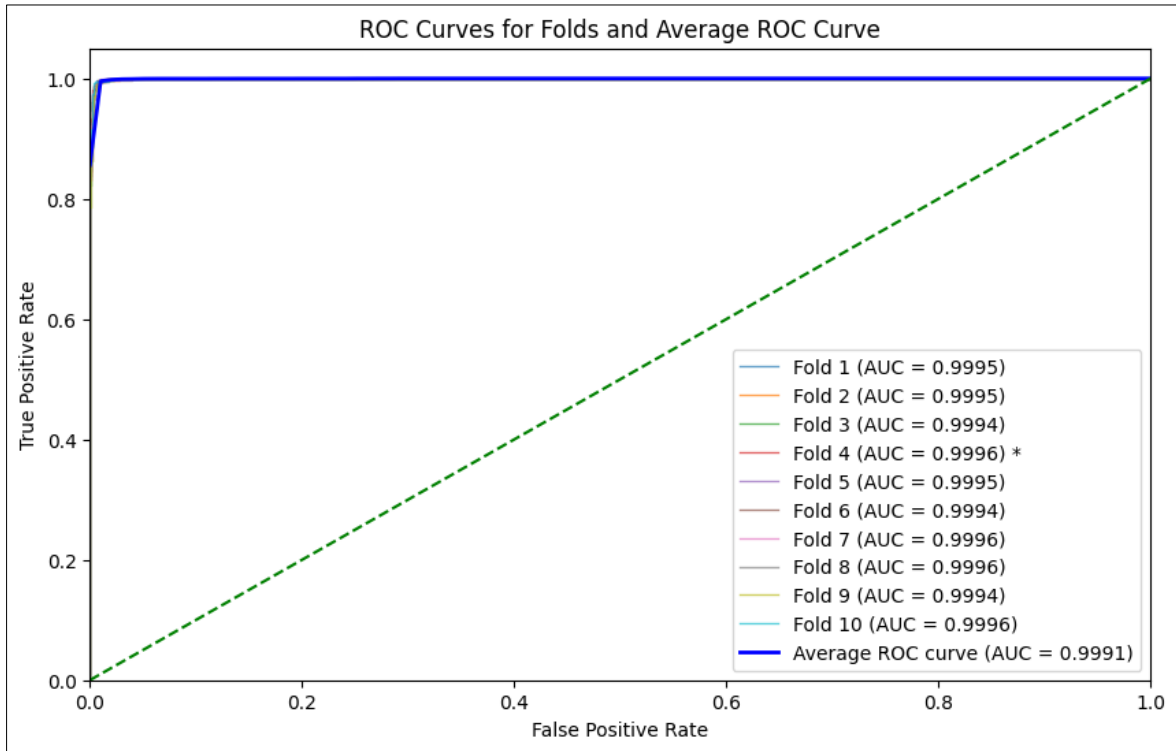


Şekil 8.22. Hibrit yaklaşım Random Forest Classifier ROC eğrisi

Hibrit yaklaşım Gradient Boosting Classifier karmaşıklık matrisi Şekil 8.23'te, ROC eğrisi Şekil 8.24'te verilmiştir.

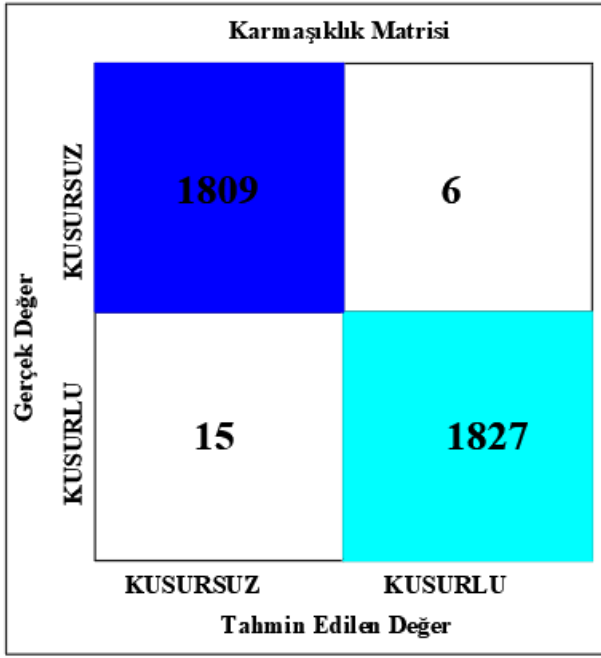


Şekil 8.23. Hibrit yaklaşım Gradient Boosting Classifier karmaşıklık matrisi

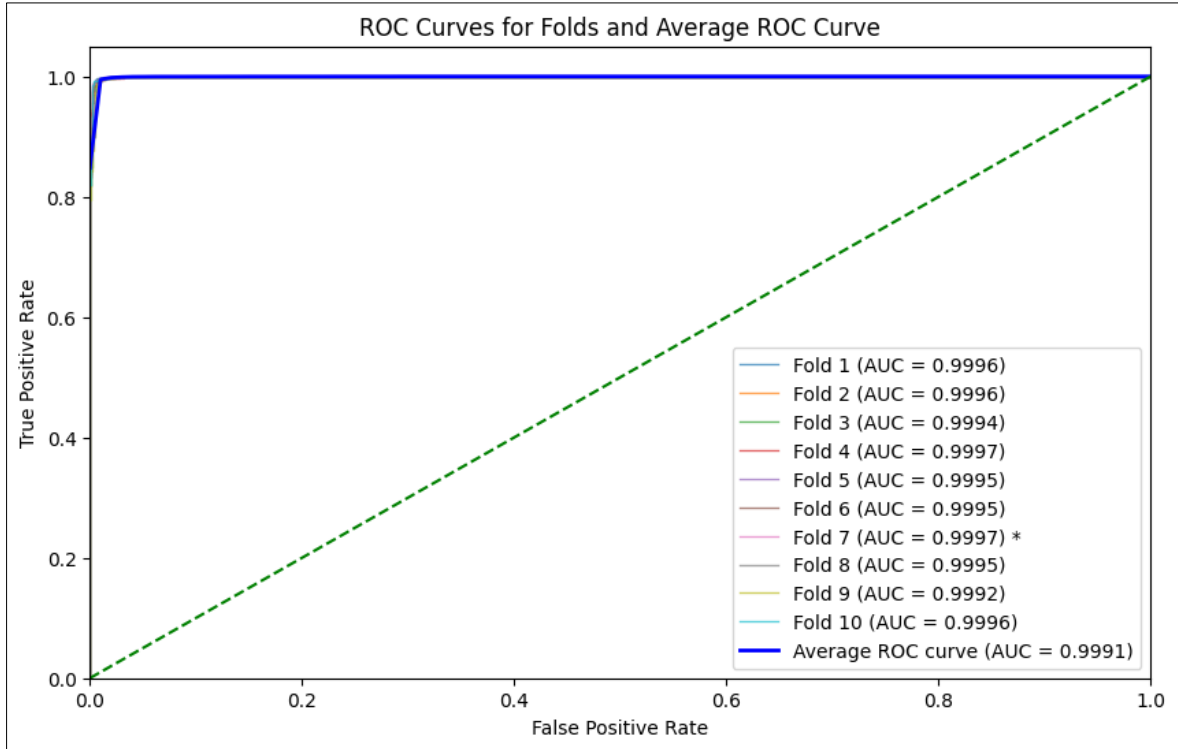


Şekil 8.24. Hibrit yaklaşım Gradient Boosting Classifier ROC eğrisi

Hibrit yaklaşım CatBoost Classifier karmaşıklık matrisi Şekil 8.25'te, ROC eğrisi Şekil 8.26'te verilmiştir.



Şekil 8.25. Hibrit yaklaşım CatBoost Classifier karmaşıklık matrisi



Şekil 8.26. Hibrit yaklaşım CatBoost Classifier ROC eğrisi

Bu çalışma kapsamında üç farklı yaklaşımla ahşap yüzey kusuru tespit etme görevi gerçekleştirilmiştir. İlk olarak optimize edilmiş ESA mimarileri ve transfer öğrenme yaklaşımı kullanılmıştır. Bu aşamada hangi optimizasyon algoritmasının seçileceği

konusunda denemeler yapılmış ve RMPS prop algoritmasının kullanılmasına karar verilmiştir. 256, 512, 1024 ve 2048 nöron kullanılarak yapılan deneylerde en iyi sonucun Xception ESA mimarisi ile RMSprop kullanılarak 512 nöronda %95,75 en yüksek doğruluk elde edilmiştir.

Çalışmada ikinci yaklaşım olarak görüntülerden öznelik çıkarımı yapılmış ve 12 farklı klasik makine öğrenme sınıflandırma algoritması kullanılmıştır. En yüksek düzeyde LBP ile öznelik çıkarımı başarılı olmuştur. LBP ile öznelik çıkarımında Random Forest Classifier %96,75 doğruluk ile sınıflandırma görevini yerine getirmiştir. LBP ile öznelik çıkarımında Gradient Boosting Classifier ve XGB Classifier %96 dolayında doğrulukla kusur tespiti gerçekleştirmişlerdir.

Eğitilmiş bir derin öğrenme modeli, yeni bir görüntü geldiğinde önceden öğrendiği özellikleri kullanarak analiz yapar. Öncelikle, yeni görüntü modelin eğitiminde kullanılan format ve boyutlara uyumlu hale getirilerek ön işleme aşamasından geçmektedir. Ardından, modelin filtreleri ve katmanları aracılığıyla görüntüden anlamlı özellikler çıkarılmakta ve ahşap yüzeyindeki lif yapısı, renk farklılıkları ve olası kusurlar analiz edilmektedir. Bu özellikler kullanılarak, model görüntünün hangi sınıfa ait olduğunu veya kusur içerip içermediğini tahmin eder. Ancak modelin başarılı sonuçlar vermesi, eğitim sırasında karşılaştığı verilerle yeni görüntülerin benzerliğine bağlıdır. Farklı ışık koşulları, açılar veya ahşap türleri gibi değişkenler performansı etkileyebilir. Bu nedenle, modelin genelleme yeteneğini artırmak için çeşitli ahşap türleri ve kusur örnekleriyle eğitilmesi, transfer öğrenme ile yeni verilerle güncellenmesi veya online öğrenme yöntemiyle kendini sürekli geliştirmesi gerekmektedir. Böylece model, yeni görüntülerde daha güvenilir ve doğru tespitler yapabilir.



## 9. SONUÇ VE ÖNERİLER

Tez çalışması kapsamında ahşap kusur tespiti yöntemleri ve bu yöntemlerin arasında farklılıklar ve sonuçlar için aşağıdaki sorulara yanıtlar aranmıştır:

- 1- Ahşap kusur tespitinde yapay zeka destekli görüntü işleme yöntemleri, geleneksel yöntemlere kıyasla ne ölçüde daha doğru, hızlı ve maliyet etkin bir çözüm sunmaktadır?
- 2- Yapay zeka ve bilgisayarlı görü teknikleri, insan önyargısından kaynaklanabilecek hataları ortadan kaldırmada ve üretim süreçlerinde zaman tasarrufu sağlamada nasıl bir avantaj sağlar?
- 3- Üretim hattında bilgisayarlı görü kullanılarak kusur tespitinin, montaj ve ürün hatalarına yol açabilecek kalite kontrol problemlerini önlemedeki etkinliği nedir?
- 4- Ahşap endüstrisinde yapay zeka tabanlı bir kalite kontrol programının uygulanabilirliği ve yaygınlaştırılabilirliği nasıl sağlanabilir ve bu programın geliştirilmesinde hangi yöntemler öncelikli olarak değerlendirilmelidir?

Birinci araştırma sorusuna cevap ararken, bilgisayarlı görü yöntemleri, ahşap yüzeyindeki kusurları tespit etmek için lazer testi, ultrasonik test ve akustik test gibi geleneksel yöntemlere güçlü bir alternatif sunar. Bu yöntemler, ahşap yüzeylerdeki çatlaklar, budaklar, çürükler, eğrilikler ve diğer kusurların hızlı ve doğru bir şekilde tespit edilmesini sağlar. Geleneksel yöntemlerin aksine, bilgisayarlı görü tabanlı yaklaşımlar özel laboratuvar ekipmanlarına ihtiyaç duymaz ve analiz süreçlerini otomatikleştirerek zamandan ve maliyetten tasarruf sağlamaktadır.

Optimize edilmiş ESA mimarileri (Xception, VGG-16, ResNet-50 gibi) ve hibrit yaklaşımlar, %97'nin üzerinde doğruluk oranları sunarak kusurların hassas bir şekilde sınıflandırılmasını mümkün kılar. Örneğin, Xception mimarisi RMSprop optimizasyon algoritması ile öznetelik çıkarımı ve klasik makine öğrenme ile sınıflamada %99,32 doğruluk oranı elde etmiştir. Bu, bilgisayarlı görü yöntemlerinin, insan faktörünün neden olduğu hata risklerini azaltarak ve üretim süreçlerini hızlandırarak endüstride devrim yaratma potansiyeline sahip olduğunu göstermektedir.

İkinci araştırma sorusuna cevap ararken, yapay zeka destekli sistemler, karar verme süreçlerinde insan faktöründen kaynaklanan önyargıları ve hataları azaltma açısından büyük avantajlar sunmaktadır. Bilgisayarlı görü tabanlı yapay zeka modelleri, sabit algoritmalar üzerinde çalıştıkları için tutarlı ve objektif sonuçlar üretirler. Bu sistemler, yüzey kusurlarını belirlerken insan algısının sınırlarını aşar ve hata oranını minimuma indirmektedir.

Bu çalışmada kullanılan ahşap kusur tespiti için özel olarak optimize edilmiş ESA mimarileri (Xception, DenseNet-121, MobileNet vb.) ve öznetelik çıkarma yöntemleri (LBP, GABOR, GLCM) ile insan müdahalesine gerek kalmadan ahşap yüzeyler hızlı ve doğru bir şekilde analiz edilmiştir. Özellikle LBP tabanlı özellik çıkarımı ile %96,75 doğruluk oranı elde edilmiştir. Ayrıca hibrit yaklaşımlarla bu oran %99'a kadar yükselmiştir. Bu tür sistemler, üretim hattındaki kalite kontrol süreçlerini otomatikleştirerek insan önyargılarından kaynaklanabilecek hataları tamamen ortadan kaldırma kapasitesine sahiptir.

Üçüncü araştırma sorusuna yanıt ararken, bilgisayarlı görü ile üretim hattında kusur tespiti, kalite kontrol süreçlerinde devrim yaratma potansiyeline sahiptir. Bu sistemler, kusurlu parçaların üretim hattında erken aşamada tespit edilmesine olanak tanımaktadır. Böylece kusurlu parçaların montaj aşamasına geçmesi engellenerek ek maliyetlerin ve zaman kayıplarının önüne geçilmektedir. Örneğin, hibrit modeller kullanılarak yapılan sınıflandırma sonuçlarında %99 doğruluk oranına ulaşılmıştır. Bu, ahşap yüzey kusurlarının erken tespitiyle üretim hattında kalite standartlarının sürekli olarak korunabileceğini göstermektedir.

Manuel kontrollerde oluşabilecek insan kaynaklı hatalar ve zaman kayıpları, bilgisayarlı görü sistemleri sayesinde tamamen ortadan kaldırılabilir. Ayrıca, üretim hattındaki bu tür otomasyon çözümleri, genel üretim hızını artırır ve süreç verimliliğini yükseltir. Bu avantajlar, üreticilerin maliyetlerini düşürmelerini ve müşterilere daha kaliteli ürünler sunmalarını sağlamaktadır.

Dördüncü araştırma sorusuna yanıt ararken, ahşap endüstrisinde yapay zeka tabanlı bir kalite kontrol programının uygulanabilirliği, optimize edilmiş ESA mimarilerinin ve klasik makine öğrenmesi yöntemlerinin başarısı ile açıkça ortaya konmuştur. Bu çalışmada

kullanılan ESA modelleri, transfer öğrenme yaklaşımını da kullanarak optimize edilmiş yüksek doğruluk oranlarına ulaşmıştır. Örneğin, Xception mimarisi %99,32 doğruluk, %97,61 F1 skoru ve %99,70 AUC ile en iyi sonuçları vermektedir.

Buna ek olarak, öznelik çıkarma yöntemleriyle klasik makine öğrenme algoritmalarının kombinasyonu, sektörde daha yaygın ve maliyeti düşük bir alternatif sunabilir. Örneğin, LBP ile özellik çıkarımı ve Random Forest sınıflandırıcı kullanılarak %96,75 doğruluk oranına ulaşılmıştır. Hibrit yaklaşım ise, ESA ile çıkarılan özelliklerin klasik makine öğrenmesi algoritmalarıyla sınıflandırılması yoluyla, %99 doğruluk oranı ile tüm yöntemlerden daha başarılı sonuçlar elde etmiştir.

Bu bulgular, yapay zeka tabanlı sistemlerin ahşap endüstrisinde uygulanabilirliğini ve yaygınlaşma potansiyelini göstermektedir. Üretim süreçlerinde maliyet etkinliği, hata oranının azaltılması ve kalite kontrol süreçlerinin iyileştirilmesi, bu tür sistemlerin kullanımını destekleyen önemli faktörlerdir. Ayrıca, geliştirilecek yeni algoritmalar ve veri işleme teknikleri, bu tür uygulamaların gelecekte daha da yaygın hale gelmesini sağlayabilir.

Ahşap kusur tespiti, dünya çapında ağaççileri endüstrisinde kalite ve üretim süreçlerinde çok önemli bir konudur. Bu çalışmada kusurlu ahşap görüntülerin tespiti için farklı bilgisayarlı görü yaklaşımları ile ahşap kusurları tespit edilmiştir. İlk olarak optimize edilmiş ESA mimarileri ve transfer öğrenme yaklaşımı ile ahşap kusur tespiti gerçekleştirilmiştir. İkinci olarak öznelik çıkarma ve klasik makine öğrenme sınıflandırma algoritmaları ile ahşap kusur tespiti yapılmıştır. Üçüncü olarak optimize edilmiş ESA mimarisi ile öznelik çıkarılıp klasik makine öğrenme sınıflandırma algoritmalarında sınıflandırma yapılan hibrit yaklaşımdır. Kusurun bir ya da daha fazla olması üretim ve kalite süreçleri açısından istenmeyen bir durumdur. Bu sebeple hızlı karar verme adına çalışmada ikili sınıflandırma; kusurlu ahşap ve kusursuz ahşap şeklinde sınıflandırma yapılmasına karar verilmiştir.

Tezde optimize edilerek kullanılan ESA mimarileri; VGG-16, MobileNet, DenseNet-121, ResNet-50, Xception, InceptionV3 kullanılmıştır. Mimarilere en uygun optimizasyon algoritmasını belirlemek için Adam, RMSprop, Adagrad, SGD, Adadelta kullanılmıştır. Bu veri seti ile en iyi sonucu veren optimizasyon algoritmasının RMSprop olduğu deney

sonucunda belirlenmiştir. ESA mimarilerinin ve optimizasyon algoritmalarının performansını belirlemek için 256, 512, 1024, 2048 nöronla değişken nöron sayılarıyla deneyler yapılmıştır. VGG-16 RMSprop 1024 nöron ile %97,50 doğruluk oranı vermiştir. ResNet-50 RMSprop 256 nöron ile %97,37 doğruluk oranı vermiştir. DenseNet-121 2048 nöronla %97,34 doğruluk oranı vermiştir. Inception-V3 RMSprop 2048 nöronla %97,24 doğruluk oranı vermiştir. En başarılı sonuç Xception RMSprop algoritması 512 nöron ile %97,69 kesinlik, %97,57 hassasiyet, %97,61 F1 skoru, %99,70 AUC ve %97,57 doğruluk oranı vermiştir. Genel olarak optimize edilmiş ESA mimarileri transfer öğrenme kullanılarak %97 doğruluk ile ahşap kusur tespiti yapılabilmektedir.

Bu araştırmada ikinci olarak, ahşap kusur tespiti için farklı özellik çıkarma yöntemlerinin ve sınıflandırma algoritmalarının performansını değerlendirilmiştir. Farklı sınıflandırma algoritmaları (RandomForestClassifier, KNeighborsClassifier, Support Vector Machine, Decision Tree Classifier, Naive Bayes, Logistic Regression, Gradient Boosting Classifier, XGB Classifier, LightGBM, CatBoost, AdaBoost, MLP) ile farklı özellik çıkarma yöntemlerini (LBP, HOG, GLCM, SOBEL, GABOR ve MB-LBP) kullanılarak ahşap kusur tespit yetenekleri açısından test edilmiştir.

LBP yönteminin kusurlu ahşabın tespitinde en etkili özellikleri çıkardığı görülmüştür. Ayrıca sınıflandırma sonuçlarına göre GABOR ve GLCM yöntemleri oldukça başarılı özellik çıkarımı gerçekleştirmiştir.

SOBEL, MB-LBP ve HOG özellikleri diğer yöntemlere göre daha az etkili olmasına rağmen kusurlu ahşabın tespitinde başarılı sonuçlar vermiştir. Tüm özellik çıkarma yöntemleri ve sınıflandırıcı kombinasyonları kullanıldığında doğruluk performansı %75 ile %97 arasında değişmektedir.

En başarılı sınıflandırma sonucu LBP özelliklerinde Rastgele Orman algoritması ile elde edilmiştir. Gradient Boosting ve XGB sınıflandırıcılarının performansları, Random Forest sınıflandırıcısını takip eden başarılı doğruluk oranlarına sahiptir. Ağaç tabanlı artırılmış topluluk yöntemleri tüm özellik kümelerinde diğer klasik makine öğrenmesi algoritmalarına göre daha başarılıdır. Gradient Boosting %96,56 doğruluk, XGB Classifier %96,28 doğruluk oranı ile ahşap kusur tespiti yapabilmektedir. LBP ile öznelik çıkarımı Random Forest Classifier ile %98,51 AUC, %96,80 kesinlik, %96,75 hassasiyet, %96,75

F1-skoru ve %96,75 doğruluk değeri ile klasik makine öğrenme sınıflandırma en yüksek başarıya sahip olduğu görülmektedir.

Tüm özelliklerin birleştirilmesiyle yapılan deneylerde doğruluk oranı LBP-Random Forest Classifier kombinasyon sonucuna göre artmamış, diğer kombinasyonlara göre ise artış göstermiştir. Gradient Boosting Classifier ile %95,16 doğruluk oranı en yüksek doğruluk olarak bulunmuştur.

Üçüncü yaklaşım olarak ESA ile öznelik çıkarımı ve klasik makine öğrenme algoritmaları ile sınıflandırma yapılan hibrit yaklaşım kullanılmıştır. Bu hibrit yaklaşıma WD (Wood Defect) Detector adı verilmiştir. Optimize Edilmiş ESA mimarisi Xception mimarisi RMSprop optimizasyon algoritması 512 nöron ile eğitim yapılarak öznelikler çıkarılmış ve 12 farklı klasik sınıflandırma algoritması ile sınıflandırılmıştır ve tüm algoritmalar %99 oranında ahşap kusur tespiti yapabilmektedir. Bu kapsamda bakıldığında hibrit yaklaşım ile ahşap kusur tespiti yapılışı yüksek doğruluk ve diğer değerlendirme metrikleri sebebiyle başarılı bulunmuştur. Hibrit yaklaşımla Random Forest %99,31 doğruluk, KNN %98,94 doğruluk, SVM %99,27 doğruluk, Decision tree %98,86 doğruluk, Naive Bayes %99,23 doğruluk, Logistic Regression %99,25 doğruluk, Gradient boosting %99,31 doğruluk, XGB %99,28 doğruluk, LightGBM %99,28 doğruluk, CatBoost %99,32 doğruluk, Adaboost %99,22 ve MLP %99,19 doğruluk ile ahşap yüzey kusur tespiti gerçekleştirmişlerdir.

İlk olarak Hibrit yaklaşım en yüksek doğruluk oranı %99,32 vermiştir. İkinci olarak optimize edilmiş ESA mimariler %97,57 en yüksek doğruluk oranı vermiştir. Üçüncü olarak klasik makine öğrenme algoritmaları ile sınıflandırmada %96,75 en yüksek doğruluk oranı elde edilmiştir. Sonuçlar ışığında heterojen olan ve farklı kusurla sahip ahşaplarda kusur tespiti konusunda oldukça başarılı bir performans elde edildiği söylenebilir.

Tez kapsamında karşılaşılan zorluklar ve kısıtlamalar şu şekilde listelenebilir:

- 1- Veri seti kısıtlamaları: Ahşap kusur tespiti için yeterli miktarda ve çeşitli veri setlerinin bulunmaması çalışmayı sınırlayan önemli bir faktör olmuştur. Mevcut veri setlerinin çoğunun sınırlı sayıda örnek içermesi, özellikle derin öğrenme modellerinin eğitimi için

bir zorluk oluşturmuştur. Kusurların gerçek dünyada çok çeşitli görünümlere sahip olması, sınıflandırma performansını etkileyebilecek bir faktördür.

- 2- Model optimizasyonu ve parametre ayarları: ESA mimarilerinin optimize edilmesi, doğru hiper parametrelerin seçilmesini gerektirmiş ve bu süreç oldukça zaman alıcı olmuştur. Her bir mimari için en iyi optimizasyon algoritmasını ve nöron sayısını belirlemek amacıyla yapılan deneyler, hesaplama gücü ve zaman açısından yüksek bir maliyet oluşturmuştur.
- 3- Klasik makine öğrenme modelleriyle öznitelik çıkarma: Klasik makine öğrenme yöntemleri için uygun öznitelik çıkarma yöntemlerinin seçimi, zaman alan bir süreç olmuştur. Bazı öznitelik çıkarma yöntemlerinin (örneğin HOG ve SOBEL) daha düşük performans göstermesi, sonuçların genel doğruluğunu düşürme potansiyeline sahiptir. Özniteliklerin seçimi ve boyut azaltma aşamaları, işlem maliyetini artırmıştır.
- 4- Hibrit yaklaşımın hesaplama gereksinimleri: Hibrit yaklaşımda, ESA ile öznitelik çıkarımı sonrası klasik makine öğrenme modellerinin kullanımı, hem yüksek hesaplama gücü hem de veri işleme süresi gerektirmiştir. Büyük boyutlu öznitelik vektörlerinin işlenmesi, bellek ve işlemci üzerinde yüksek bir yük oluşturmuştur.
- 5- Değerlendirme ve karşılaştırma süreçleri: Farklı yöntemlerin performansını karşılaştırmak için geniş bir metrik seti kullanılmıştır. Bu durum, analiz süreçlerini daha karmaşık hale getirmiştir. Farklı modellerin ve yöntemlerin elde ettiği başarı oranlarını yorumlarken, yöntemlerin avantaj ve dezavantajlarının doğru bir şekilde tartışılması gerekmiştir.
- 6- Kusur çeşitliliği ve sınıflandırma karmaşıklığı: Ahşap yüzey kusurlarının (örneğin çatlak, çürük, eğrilik) farklı boyut, şekil ve doku özelliklerine sahip olması, kusurların tespiti ve sınıflandırılmasını daha zor hale getirmiştir. Bazı kusurların benzer görünümlere sahip olması, sınıflar arasında ayırım yapmayı güçleştirmiştir.
- 7- Donanım ve yazılım kısıtlamaları: Derin öğrenme modellerinin eğitimi ve hibrit yaklaşımların uygulanması için yüksek performanslı donanıma ihtiyaç duyulmuştur.

Bu durum, çalışmanın yürütülmesini etkileyen bir kısıt olmuştur. Büyük veri işleme ve model eğitimi sırasında işlem süreleri, donanım yetersizliği nedeniyle uzamıştır.

- 8- Transfer öğrenme sınırları: Kullanılan ESA mimarilerinin transfer öğrenme ile optimize edilmesi, genelleştirme kabiliyeti üzerinde sınırlı bir etkiye sahip olmuştur. Ahşap kusur tespiti için kullanılan transfer öğrenme yöntemlerinin, farklı veri kümelerine uygulanabilirliği kısıtlıdır.
- 9- Gerçek zamanlı uygulama zorlukları: Modelin gerçek zamanlı çalışması için optimize edilmesi ve kusur tespiti sırasında düşük gecikme süresi sağlanması, ek bir zorluk olarak ortaya çıkmıştır. Özellikle hibrit yöntemin gerçek zamanlı uygulamalar için daha yüksek işlem süreleri gerektirmesi, pratik uygulamalarda sınırlamalara neden olmuştur. Bu zorluklar, çalışmanın hem teknik hem de operasyonel boyutlarını etkileyerek daha fazla optimizasyon ve yenilikçi yaklaşımlar gerektirmiştir.

Tez sürecinde elde edilen bilgiler ve tecrübeler şu şekilde sunulmuştur:

- 1- Optimize edilmiş ESA mimarileri ile ahşap kusur tespiti: Kullanılan ESA mimarileri: VGG-16, MobileNet, DenseNet-121, ResNet-50, Xception, InceptionV3. En iyi optimizasyon algoritması RMSprop olarak belirlenmiştir. Xception mimarisi, 512 nöronla %97,69 kesinlik, %97,57 hassasiyet, %97,61 F1 skoru, %99,70 AUC ve %97,57 doğruluk oranı ile en başarılı sonucu vermiştir. Tüm ESA mimarileri genel olarak transfer öğrenme kullanılarak %97 doğruluk oranına ulaşmıştır.
- 2- Öznitelik çıkarma ve klasik makine öğrenme yöntemleri ile ahşap kusur tespiti: Kullanılan öznitelik çıkarma yöntemleri: LBP, HOG, GLCM, SOBEL, GABOR, MB-LBP. En başarılı özellik çıkarım yöntemi LBP olarak belirlenmiştir. LBP yöntemi ile Random Forest kullanılarak %96,80 kesinlik, %96,75 hassasiyet, %96,75 F1 skoru, %98,51 AUC ve %96,75 doğruluk oranı elde edilmiştir. Gradient Boosting (%96,56) ve XGB Classifier (%96,28) de yüksek doğruluk oranlarına ulaşmıştır. Tüm özellik çıkarma yöntemleri ve sınıflandırıcı kombinasyonlarıyla doğruluk oranı %75 ila %97 arasında değişmiştir.

- 3- Hibrit yaklaşım (ESA ile öznitelik çıkarma + klasik makine öğrenme): Xception mimarisi kullanılarak öznitelik çıkarılmış ve 12 farklı klasik makine öğrenme algoritmasıyla sınıflandırma yapılmıştır. Tüm algoritmalar genel olarak %99 doğruluk oranı ile başarılı sonuçlar vermiştir. En yüksek doğruluk oranı %99,32 ile CatBoost algoritmasıyla elde edilmiştir. Hibrit yaklaşımın en başarılı kombinasyonları: Random Forest: %99,31 doğruluk, SVM: %99,27 doğruluk, Gradient Boosting: %99,31 doğruluk, Logistic Regression: %99,25 doğruluk sonuçlarını vermişlerdir.
- 4- Genel karşılaştırma: Hibrit yaklaşım, %99,32 doğruluk oranı ile en yüksek başarıya ulaşmıştır. Optimize edilmiş ESA mimarileri, %97,57 doğruluk oranı ile ikinci sırada yer almıştır. Klasik makine öğrenme algoritmaları, %96,75 doğruluk oranı ile üçüncü sırada yer almıştır.
- 5- Ahşap kusur tespitinin endüstriye katkıları: Kusurların erken tespitiyle üretim süreçleri hızlanmış ve malzeme israfı önlenmiştir. Kalite kontrol süreçleri otomatikleşerek insan hataları minimize edilmiştir. Üretim maliyetleri düşürülmüş ve endüstriyel verimlilik artırılmıştır. Sürekli veri toplama ve analiz ile süreç iyileştirmelerine katkı sağlanmıştır. Ahşap endüstrisinde daha güvenilir ve etkili bir kalite kontrol sistemi geliştirilmiştir.

Bu sonuçlar, bilgisayarlı görü ve yapay zeka tekniklerinin ahşap kusur tespiti alanında yüksek doğruluk oranları ve endüstriyel faydalar sağladığını göstermektedir. Ahşap kusur tespiti için bilgisayar görüşü tabanlı yaklaşım, ağaçşileri endüstri mühendisliğinde önemli bir uygulama alanıdır ve çeşitli avantajlar sunabilir.

Bilgisayarlı görü ile hata tespit etme ve kontrolü, ahşap ürünlerin kesitlerindeki kusurları hızlı ve hassas bir şekilde tespit edebilir. Çatlaklar, çürükler, eğrilikler ve diğer kusurların erken tespiti, ürün kalitesini artırabilir.

Bilgisayarlı görü ile üretim verimliliği artırılabilir, manuel denetimlere göre daha hızlıdır. Bu, üretim süreçlerini hızlandırabilir ve üretim verimliliğini artırabilir. Malzeme optimizasyonu, ahşap yüzeylerdeki kusurların erken tespiti, malzeme israfını önleyebilir. Hatalı veya kusurlu kesitleri belirleyerek, sadece sağlam parçaların kullanılmasını sağlayabilir ve böylece malzeme maliyetlerini azaltabilir.

Güvenilirlik ve tutarsızlık kontrolü insan hatasını minimize eder ve tutarsızlık riskini azaltır. Her bir ahşap yüzey kusuru, aynı standartlara göre değerlendirilir, bu da ürünler arasında daha yüksek bir kalite tutarlılığına yol açar.

Veri toplama ve izleme, üretim sürecindeki verileri sürekli olarak izleyebilir ve kaydedebilir. Bu, üretim süreçlerini optimize etmek ve gelecekteki iyileştirmeler için veri sağlamak için kullanılabilir.

İnsan sağlığı ve güvenliği, tehlikeli durumları tespit edebilir ve insanların manuel olarak müdahale etmesini gerektiren riskleri azaltabilir. Bu durum işyerindeki güvenliği artırabilir.

Sonuç olarak, bilgisayarlı görü ile ahşap yüzeylerdeki kusurların tespiti, üretim süreçlerini iyileştirebilir, kaliteyi artırabilir, ahşap malzeme israfını azaltabilir ve genel olarak endüstriyel uygulamalarda daha etkili ve güvenilir bir üretim ve kalite kontrol sistemi sağlayabilir.

Gelecekte ahşap kusur tespiti kapsamında yapılabilecek araştırmalar için öneriler:

- 1- Ahşap kusur tespiti alanında makine öğrenimi ve derin öğrenme yöntemlerini daha da ileriye taşıyabilir. Öncelikle, Vision Transformer (ViT) ve Swin Transformer gibi yeni nesil derin öğrenme mimarileri kullanılarak model performansı artırılabilir.
- 2- Üretim hatlarında gerçek zamanlı kusur tespiti sağlayabilecek hızlı ve düşük gecikmeli sistemlerin geliştirilmesi de önemli bir hedef olabilir.
- 3- Ayrıca, görsel verilerin yanı sıra ses veya termal görüntüleme gibi farklı veri türlerinin entegre edildiği multimodal yaklaşımlar, tespit süreçlerinin doğruluğunu ve kapsamını genişletebilir.
- 4- Modellerin karar süreçlerini daha anlaşılır hale getirmek için açıklanabilir yapay zeka yöntemlerinin uygulanması, bu alanda güvenilirliği artırabilir.
- 5- Son olarak, taşınabilir cihazlarda çalışabilecek hafif ve enerji verimli modeller tasarlanarak sistemlerin daha geniş bir uygulama alanına yayılması sağlanabilir.



## KAYNAKLAR

- Abdulazeez, A. M., Salim, B. W., Zeebaree, D. Q., and Doghramachi, D. (2020). Comparison of VPN Protocols at Network Layer Focusing on Wire Guard Protocol. *International Journal of Interactive Mobile Technologies (iJIM)*, 14(18), 157–177.
- Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., and Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11), 1-41.
- Acharya, M. S., Armaan, A., and Antony, A. S. (2019, February). *A comparison of regression models for prediction of graduate admissions*. IEEE In 2019 International Conference on computational intelligence in Data Science (ICCIDS), Chennai, India, 1-5.
- Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1), 147-169.
- Adegun, A. A., and Viriri, S. (2020). FCN-based DenseNet framework for automated detection and classification of skin lesions in dermoscopy images. *IEEE Access*, 8, 150377-150396.
- Agrawal, R., Imieliński, T., and Swami, A. (1993, June). *Mining association rules between sets of items in large databases*. In Proceedings of the 1993 ACM SIGMOD international conference on Management of data, Washington, DC, USA, 207-216.
- Ahmed, M., Mahmood, A. N., and Maher, M. J. (2015). An efficient technique for network traffic summarization using multiview clustering and statistical sampling. *EAI Endorsed Transactions on Scalable Information Systems*, 2(5), 1-9.
- Altameem, A., and Ykhlef, M. (2018). Hybrid approach for improving efficiency of apriori algorithm on frequent itemset. *International Journal of Computer Science and Network Security*, 18(5), 151-155.
- Altan, G. (2019). DeepGraphNet: grafiklerin sınıflandırılmasında derin öğrenme modelleri. *Avrupa Bilim ve Teknoloji Dergisi*, (Özel Sayı), 319-327.
- Arivazhagan, S., Ganesan, L., and Kumar, T. S. (2006, August). *Texture classification using curvelet statistical and co-occurrence features*. IEEE Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), 2, 938–941.
- Ayan, E., and Ünver, H. M. (2018, April). *Data augmentation importance for classification of skin lesions via deep learning*. IEEE 2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT), 1–4.
- Baturynska, I., and Martinsen, K. (2021). Prediction of geometry deviations in additive manufactured parts: comparison of linear regression with machine learning algorithms. *Journal of Intelligent Manufacturing*, 32, 179-200.

- Bishop, C. M., and Nasrabadi, N. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer, 4(4), 738.
- Bloice, M. D., Stocker, C., and Holzinger, A. (2017). Augmentor: an image augmentation library for machine learning. *Journal of Open Source Software*, 2(19), 432.
- Breiman, L. (2001). Random forests. *Machine learning*, 45, 5-32.
- Broman, O., and Fredriksson, M. (2012). Wood material features and technical defects that affect yield in a finger joint production process. *Wood Material Science and Engineering*, 7(4), 167-175.
- Chandrasekhar, V., Reznik, Y., Takacs, G., Chen, D., Tsai, S., Grzeszczuk, R., and Girod, B. (2010, June). *Quantization schemes for low bitrate compressed histogram of gradients descriptors*. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, 33-40.
- Chen, T., and Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, 785-794.
- Chen, Y., Sun, C., Ren, Z., and Na, B. (2023). Review of the Current State of Application of Wood Defect Recognition Technology. *BioResources*, 18(1), 2288-2302.
- Chicco, D., and Jurman, G. (2020). Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. *BMC Medical Informatics and Decision Making*, 20(1), 1-16.
- Chierichetti, F., Kumar, R., Lattanzi, S., and Vassilvitskii, S. (2017). Fair clustering through fairlets. *Advances in Neural Information Processing Systems*, 30, 1-12.
- Ciresan, D. C., Meier, U., Masci, J., Maria Gambardella, L., and Schmidhuber, J. (2011, July). *Flexible, high performance convolutional neural networks for image classification*. In IJCAI proceedings-international joint conference on artificial intelligence, Manno-Lugano, Switzerland, 22(1), 1237-1243.
- Cortes, C., and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20, 273-297.
- Cover, T., and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21-27.
- Çetiner, İ. (2014). *Budak faktörüne göre kızılçam (pinus brutia ten.) ahşap döşeme tahtalarının kalite sınıfının görüntü işleme teknikleri kullanılarak belirlenmesi*, Doktora Tezi, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, Isparta.
- Çetiner, İ., Var, A. A., and Çetiner, H. (2014). *Wood surface analysis with image processing techniques*. IEEE In 2014 22nd Signal Processing and Communications Applications Conference (SIU), 393-396.

- Dakshayani, V., Locharla, G. R., Pławiak, P., Datti, V., and Karri, C. (2022). Design of a Gabor filter-based image denoising hardware model. *Electronics*, 11(7), 1063-7074.
- Dalal, N., and Triggs, B. (2005, June). *Histograms of oriented gradients for human detection*. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), (1), 886-893.
- Ding, F., Zhuang, Z., Liu, Y., Jiang, D., Yan, X., and Wang, Z. (2020). Detecting defects on solid wood panels based on an improved SSD algorithm. *Sensors*, 20(18), 5315-1532.
- Doğan, F. ve Türkoğlu, İ. (2019). Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10(2), 409-445.
- Dongare, A. D., Kharde, R. R., and Kachare, A. D. (2012). Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(1), 189-194.
- İnternet: Doshi, S. (2019). *Various Optimization Algorithms For Training Neural Network*  
URL1: <https://medium.com/data-science/optimizers-for-training-neural-network5945d71caf6>, Son Erişim Tarihi: 03.01.2025.
- Duchi, J., Hazan, E., Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2121-2159.
- Ebert, C., and Louridas, P. (2016). Machine learning. *IEEE Software*, 33(5), 110-115.
- Ergün, E. ve Kılıç, K. (2021). Derin öğrenme ile artırılmış görüntü seti üzerinden cilt kanseri tespiti. *Black Sea Journal of Engineering and Science*, 4(4), 192-200.
- Fang, Y., Guo, X., Chen, K., Zhou, Z., and Ye, Q. (2021). Accurate and automated detection of surface knots on sawn timbers using YOLO-V5 model. *BioResources*, 16(3), 5390-5406.
- Fırıldak, K., Talu, M. F. (2019). Evrimsel Sinir Ağlarında Kullanılan Transfer Öğrenme Yaklaşımlarının İncelenmesi. *Bilgisayar Bilimleri*, 4(2), 88-95.
- Forsyth, D.A.; Ponce, J. (2002). *Computer Vision: A Modern Approach*; Prentice Hall Professional Technical Reference: Hoboken, NJ, USA, 237-251.
- Freund, Y., and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119-139.
- Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 23(5), 1189-1232.

- Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29, 131-163.
- Gao, M., Qi, D., Mu, H., and Chen, J. (2021). A transfer residual neural network based on ResNet-34 for detection of wood knot defects. *Forests*, 12(2), 212- 227.
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 27-51.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., and Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139-144.
- Gu, I. Y. H., Andersson, H., and Vicen, R. (2010). Wood defect classification based on image analysis and support vector machines. *Wood Science and Technology*, 44, 693-704.
- Gu, I. Y., Andersson, H., and Vicen, R. (2009, November). *Automatic classification of wood defects using support vector machines*. In *Computer Vision and Graphics: International Conference, ICCVG 2008*, 10-12, 356-367, Warsaw, Poland.
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354-377.
- Guo, R., Zhao, Z., Wang, T., Liu, G., Zhao, J., and Gao, D. (2020). Degradation state recognition of piston pump based on ICEEMDAN and XGBoost. *Applied Sciences*, 10(18), 6593.
- Gürtekin, A. ve Oğuz, M. (2002). *Mobilya ve dekorasyon gereç bilgisi*. Ankara: Devlet Kitapları, Millî Eğitim Basımevi, 402.
- Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques* (3rd ed.). Waltham, Massachusetts, ABD: Morgan Kaufmann, 323-359.
- Haoran, Z., Yuzeng, W., and Enkai, B. (2021, December). *Smart Classification System of Solid Wood Board Based on Convolutional Neural Networks*. IEEE In 2021 International Conference on Intelligent Computing, Automation and Systems (ICICAS), 302-307.
- Haralick, R. M., Shanmugam, K., and Dinstein, I. H. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, (6), 610-621.
- Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, New York: Springer, 2, 1-758.
- Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1-768.

- He, K., Zhang, X., Ren, S., and Sun, J. (2016). *Deep residual learning for image recognition*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778.
- He, T., Liu, Y., Yu, Y., Zhao, Q., and Hu, Z. (2020). Application of deep convolutional neural network on feature extraction and detection of wood defects. *Measurement*, 152, 107357- 107365.
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012a). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv*, 1-18.
- Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., and Kingsbury, B. (2012b). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6), 82-97.
- Hittawe, M. M., Muddamsetty, S. M., Sidibé, D., and Mériaudeau, F. (2015, September). *Multiple features extraction for timber defects detection and classification using SVM*. In 2015 IEEE International Conference on Image Processing (ICIP), 427-431.
- Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.
- Hosmer Jr, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied logistic regression*. Hoboken, New Jersey, ABD: John Wiley and Sons, 91-135.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications, *arXiv*, 1-9.
- Hu, J., Song, W., Zhang, W., Zhao, Y., and Yilmaz, A. (2019). Deep learning for use in lumber classification tasks. *Wood Science and Technology*, 53, 505-517.
- Hu, W., Wang, T., Wang, Y., Chen, Z., and Huang, G. (2022). LE-MSFE-DDNet: a defect detection network based on low-light enhancement and multi-scale feature extraction. *The Visual Computer*, 38(11), 3731-3745.
- Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). *Densely connected convolutional networks*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4700-4708.
- Ibrahim, E. A. B., Hashim, U. R. A., Salahuddin, L., Ismail, N. H., Choon, N. H., Kanchymalay, K., and Zabri, S. N. (2021). Evaluation of texture feature based on basic local binary pattern for wood defect classification. *International Journal of Advances in Intelligent Informatics*, 7(1), 26-36.
- Ioffe, S., and Szegedy, C. (2015, June). *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. In International Conference on Machine Learning, 448-456.

- İnik, Ö. ve Ülker, E. (2017). Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri. *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6(3), 85-104.
- Jabo, S. (2011). Machine Vision for Wood Defect Detection and Classification. Master's Thesis, Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden.
- Jarrett, K., Kavukcuoglu, K., Ranzato, M. A., and LeCun, Y. (2009, September). *What is the best multi-stage architecture for object recognition?*. In 2009 IEEE 12th International Conference on Computer Vision, 2146-2153.
- Ji, X., Guo, H., and Hu, M. (2019, September). *Features extraction and classification of wood defect based on HU invariant moment and wavelet moment and BP neural network*. In Proceedings of the 12th International Symposium on Visual Information Communication and Interaction, 1-5.
- Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification, *arXiv*, 1-5.
- Kamal, K., Qayyum, R., Mathavan, S., and Zafar, T. (2017). Wood defects classification using laws texture energy measures and supervised learning approach. *Advanced Engineering Informatics*, 34, 125-135.
- Kavzoğlu, T. ve Çölkesen, İ. (2010). Karar ağaçları ile uydu görüntülerinin sınıflandırılması. *Harita Teknolojileri Elektronik Dergisi*, 2(1), 36-45.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., and Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30(1), 1-9.
- Kılıç, K. (2021). *Cilt Kanseri Tespitinde Evrimsel Sinir Ağları Mimarilerinin Karşılaştırılması*, Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi Lisansüstü Eğitim Enstitüsü, Samsun.
- İnternet: Kızrak, M., A. (2019). *Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması*. URL2: <https://ayyucekizrak.medium.com/derin-ogrenme-icin-aktivasyon-fonksiyonlarinin-karsilastirilmesi-cee17fd1d9cd>, Son Erişim Tarihi: 20.02.2025.
- Kim, J. H., Kim, S. C., and Kang, T. J. (2006, November). *Fractal dimension co-occurrence matrix method for texture classification*. In TENCON 2006-2006 IEEE Region 10 Conference, 1-4.
- Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv*, 1-15.
- Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., and Klein, M. (2002). *Logistic Regression*, New York: Springer-Verlag, 536.

- Koç, O. (2019). Comparison of Machine Learning Algorithms on Consumer Credit Classification. Master's Thesis, Middle East Technical University Institute of Applied Mathematics, Ankara.
- Kodytek, P., Bodzas, A., and Bilik, P. (2022). A large-scale image dataset of wood surface defects for automated vision-based quality control processes. *F1000Research*, 10, 581-597.
- Kohavi, R. (1995). *A study of cross-validation and bootstrap for accuracy estimation and model selection*. International Joint Conference on Artificial Intelligence, 2, 1137-1143.
- Kohavi, R., and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2), 273-324.
- Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, 160(1), 3-24.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1-9.
- Kruschke, J. K. (2014). *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan*. San Diego, California, ABD: Academic Press, 15-31.
- Ławrynowicz, A., and Tresp, V. (2014). Introducing machine learning. *Perspectives on Ontology Learning*, 18, 35-50.
- LeCun, Y., and Bengio, Y. (1995). *Convolutional networks for images, speech, and time series*. In: The handbook of brain theory and neural networks, Cilt (3361) Citeseer, 255-258.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Lewis, D. D. (1998, April). *Naive (Bayes) at forty: The independence assumption in information retrieval*. In European conference on machine learning, Springer Berlin Heidelberg, 4-15.
- Li, H., Lin, Z., Shen, X., Brandt, J., and Hua, G. (2015). *A convolutional neural network cascade for face detection*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5325-5334.
- Li, S., Li, D., and Yuan, W. (2019). Wood defect classification based on two-dimensional histogram constituted by LBP and local binary differential excitation pattern. *IEEE Access*, 7, 145829-145842.

- Li, X.-L., and Zhong, Y. (2012). An overview of personal credit scoring: Techniques and future work. *International Journal of Intelligence Science*, 2(4), 181-189.
- Lim, W. H., Bonab, M. B., and Chua, K. H. (2022, June). *An optimized lightweight model for real-time wood defects detection based on yolov4-tiny*. In 2022 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), 186-191.
- Lim, W. H., Bonab, M. B., and Chua, K. H. (2022, June). *An optimized lightweight model for real-time wood defects detection based on yolov4-tiny*. In 2022 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), pp. 186-191.
- Lin, H. I., and Sanjaya, S. D. (2021, October). *Wood polish classification for automated quality inspection based on AI vision*. In 2021 21st International Conference on Control, Automation and Systems (ICCAS), 1974-1977.
- Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *arXiv*, 1-10.
- Liu, X., Song, L., Liu, S., and Zhang, Y. (2021). A review of deep-learning-based medical image segmentation methods. *Sustainability*, 13(3), 1224-1253.
- Liu, Y., Xu, K., and Xu, J. (2019). An improved MB-LBP defect recognition approach for the surface of steel plates. *Applied Sciences*, 9(20), 4222-4236.
- Liu, Z., Peng, C., Work, T., Candau, J. N., DesRochers, A., and Kneeshaw, D. (2018). Application of machine-learning methods in forest ecology: recent progress and future challenges. *Environmental Reviews*, 26(4), 339-350.
- Lopez, A. R., Giro-i-Nieto, X., Burdick, J., and Marques, O. (2017, February). *Skin lesion classification from dermoscopic images using deep learning techniques*. In 2017 13th IASTED International Conference on Biomedical Engineering (BioMed), 49-54.
- Lorena, A. C., Jacintho, L. F., Siqueira, M. F., De Giovanni, R., Lohmann, L. G., De Carvalho, A. C., and Yamamoto, M. (2011). Comparing machine learning classifiers in potential distribution modelling. *Expert Systems with Applications*, 38(5), 5268-5275.
- Luo, W. (2019). Research on Wood Classification and Sorting Algorithms Based on Image Multi-Feature Pattern Recognition. PhD Dissertation, Northeast Forestry University, Harbin, China.
- Lycken, A. (2006). Comparison between automatic and manual quality grading of sawn softwood. *Forest Products Journal*, 56(4), 13-18.
- Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR)*, 9(1), 381-386.

- Maind, S. B., Wankar, P. (2014). Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(1), 96-100.
- Maltarollo, V. G., Honório, K. M., and da Silva, A. B. F. (2013). Applications of artificial neural networks in chemical problems. *Artificial Neural Networks-Architectures and Applications*, 1, 203-223.
- McCulloch, W. S., Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115-133.
- McElreath, R. (2018). *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman and Hall/CRC, 1-505.
- Metlek, S. ve Kayaalp, K. (2020). Derin öğrenme ve destek vektör makineleri ile görüntüden cinsiyet tahmini. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 8(3), 2208-2228.
- Minsky, M. (1961). Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1), 8-30.
- Mohsin, M., Balogun, O. S., Haataja, K., and Toivanen, P. (2022). Real-time defect detection and classification on wood surfaces using deep learning. *Electronic Imaging*, 34(10), 382-1.
- Mu, H., and Qi, D. (2009, October). *Pattern recognition of wood defects types based on Hu invariant moments*. IEEE In 2009 2nd International Congress on Image and Signal Processing, 1-5.
- Muralidharan, R., and Chandrasekar, C. (2012, March). *Combining local and global feature for object recognition using SVM-KNN*. IEEE In International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012), 1-7.
- Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press, 247-274.
- Nair, A., Srinivasan, P., Blackwell, S., Alcicek, C., Fearon, R., De Maria, A., and Silver, D. (2015). Massively parallel methods for deep reinforcement learning. *arXiv*, 1-14.
- Nilsson, N. J. (1998). *Artificial intelligence: A new synthesis*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 37-43.
- Norlander, R., Grahn, J., and Maki, A. (2015). *Wooden knot detection using convnet transfer learning*. In Image Analysis: 19th Scandinavian Conference, SCIA 2015, Copenhagen, Denmark, June 15-17, Proceedings, 19, 263-274.
- Nwankpa, C., Ijomah, W., Gachagan, A., Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv*, 1-20.

- Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 971-987.
- Othmen, M. B., Sayadi, M., and Fnaiech, F. (2008, May). *Interest of the multi-resolution analysis based on the co-occurrence matrix for texture classification*. In MELECON 2008-The 14th IEEE Mediterranean Electrotechnical Conference, 852-856.
- Örs, Y. ve Keskin, H. (2001). *Ağaç malzeme bilgisi*. Türkiye: Atlas Yayın Dağıtım, 26-38.
- Pacal, I., Karaboga, D., Basturk, A., Akay, B., and Nalbantoglu, U. (2020). A comprehensive review of deep learning in colon cancer. *Computers in Biology and Medicine*, 126, 104003-104036.
- Packianather, M. S., and Drake, P. R. (2005). Comparison of neural and minimum distance classifiers in wood veneer defect identification. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 219(11), 831-841.
- Pan, W. (2016). A survey of transfer learning for collaborative recommendation with auxiliary data. *Neurocomputing*, 177, 447-453.
- Internet: Peckov, A. (2012). *A machine learning approach to polynomial regression*. URL3: [https://kt.ijs.si/wp-content/uploads/2021/11/phd\\_aleksandar\\_peckov.pdf](https://kt.ijs.si/wp-content/uploads/2021/11/phd_aleksandar_peckov.pdf), Son Erişim Tarihi: 10.02.2025.
- Petrou, M., & Sevilla, P. G. (2006). *Image Processing: Dealing with Texture*. West Sussex, England: John Wiley & Sons, 1-51.
- Porebski, A., Vandenbroucke, N., and Macaire, L. (2007, September). *Iterative feature selection for color texture classification*. In 2007 IEEE International Conference on Image Processing, 3, 509-512.
- Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv*, 1-27.
- Pölzleitner, W., and Schwingshakl, G. (1992). Real-time surface grading of profiled wooden boards. *Industrial Metrology*, 2(3-4), 283-298.
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 1-23.
- Purnama, I. K. E., Hernanda, A. K., Ratna, A. A. P., Nurtanio, I., Hidayati, A. N., Purnomo, M. H., and Rachmadi, R. F. (2019, November). *Disease classification based on dermoscopic skin images using convolutional neural network in teledermatology system*. In 2019 international conference on computer engineering, network, and intelligent multimedia (CENIM), 1-5.

- Qayyum, R., Kamal, K., Zafar, T., and Mathavan, S. (2016, September). *Wood defects classification using GLCM based features and PSO trained neural network*. IEEE In 2016 22nd International Conference on Automation and Computing (ICAC),, 273-277.
- Qi, D., Zhang, P., Jin, X., and Zhang, X. (2010, June). *Study on wood image edge detection based on Hopfield neural network*. IEEE In The 2010 IEEE International Conference on Information and Automation, 1942-1946.
- Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1), 145-151.
- Qiu, Z.F. (1996). A Simple Machine Vision System for Improving the Edging and Trimming Operations Performed in Hardwood Sawmills. Master's Thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1, 81-106.
- Raheja, J. L., Kumar, S., and Chaudhary, A. (2013). Fabric defect detection based on GLCM and Gabor filter: A comparison. *Optik*, 124(23), 6469-6474.
- Rashidi, H. H., Tran, N. K., Betts, E. V., Howell, L. P., and Green, R. (2019). Artificial intelligence and machine learning in pathology: the present landscape of supervised methods. *Academic Pathology*, 6, 1-17.
- Recio, J. A. R., Fernandez, L. A. R., and Fernández-Sarriá, A. (2005). Use of Gabor filters for texture classification of digital images. *Física de la Tierra*, 17, 47-59.
- Rende, F. Ş., Bütün, G., & Karahan, Ş. (2016). *Derin öğrenme algoritmalarında model testleri: Derin testler*. 10. Ulusal Yazılım Mühendisliği Sempozyumu (UYMS 2016), 24-26 Ekim, 54-59, Çanakkale, Türkiye.
- Robnik-Šikonja, M. (2004). Improving random forests. In Boulicaut, J.-F., Esposito, F., Giannotti, F., and Pedreschi, D. (Eds.), *Machine Learning: ECML*, 359–370.
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv*, 1-14.
- Russell, S. J., and Norvig, P. (2016). *Artificial intelligence: A modern approach*. Pearson Education, Inc., 1-136.
- Sahla, N. (2018). *A Deep Learning Prediction Model for Object Classification*. Master's Thesis, Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, Delft, Netherlands.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85-117.
- Schmoldt, D. L., Li, P., and Abbott, A. L. (1997). Machine vision using artificial neural networks with local 3D neighborhoods. *Computers and Electronics in Agriculture*, 16(3), 255-271.

- Schraudolph, N. N., Yu, J., and Günter, S. (2007, March). *A stochastic quasi-Newton method for online convex optimization*. In Artificial intelligence and statistics, 436-443.
- Shaheamlung, G., Kaur, H., and Kaur, M. (2020, June). *A Survey on machine learning techniques for the diagnosis of liver disease*. IEEE In 2020 International Conference on Intelligent Engineering and Management (ICIEM), 337-341.
- Shahrabadi, S., Castilla, Y., Guevara Lopez, M. A., Magalhães, L., Garcia, D., and Adão, T. (2022, April). Defect detection in the textile industry using image-based machine learning methods: A brief review. *Journal of Physics: Conference Series*, 2224(1), 1-13.
- Sharma, S., Sharma, S., and Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, 6(12), 310-316.
- Shi, J., Li, Z., Zhu, T., Wang, D., and Ni, C. (2020). Defect detection of industry wood veneer based on NAS and multi-channel mask R-ESA. *Sensors*, 20(16), 4398-4415.
- Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv*, 1-14.
- Smyth, G. K. (2002). Nonlinear regression. *Encyclopedia of Environmetrics*, 3, 1405-1411.
- Song, W., Chen, T., Gu, Z., Gai, W., Huang, W., and Wang, B. (2015, July). *Wood materials defects detection using image block percentile color histogram and eigenvector texture feature*. In First International Conference on Information Sciences, Machinery, Materials and Energy, 779-783.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.
- Stone, M. (1974). Cross validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2), 111-133.
- Sugiarto, B., Prakasa, E., Wardoyo, R., Damayanti, R., Dewi, L. M., Pardede, H. F., and Rianto, Y. (2017, November). *Wood identification based on histogram of oriented gradient (HOG) feature and support vector machine (SVM) classifier*. IEEE In 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 337-341.
- Sutskever, I., Hinton, G. E., Taylor, G. W. (2009). The recurrent temporal restricted boltzmann machine. In *Advances in Neural Information Processing Systems*, 1601-1608.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A. (2015). *Going deeper with convolutions*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1-9.

- Szeliski, R. (2022). *Computer vision: algorithms and applications*. Heidelberg, Germany: Springer Nature, 85-144.
- Tan, M., Tsang, I. W., and Wang, L. (2014). Towards ultrahigh dimensional feature selection for big data. *Journal of Machine Learning Research*, 1371-1429.
- Tang, J., Henderson, A., and Gardner, P. (2021). Exploring AdaBoost and Random Forests machine learning approaches for infrared pathology on unbalanced data sets. *Analyst*, 146(19), 5880-5891.
- Tang, Y. (2013). Deep learning using linear support vector machines. *arXiv*, 1-6.
- Tou, J. Y., Khoo, K. K. Y., Tay, Y. H., and Lau, P. Y. (2009). *Evaluation of Speed and accuracy for comparison of texture classification implementation on embedded platform*. In Proceedings of the Korean Society of Broadcast Engineers Conference, 89-93.
- Tou, J. Y., Tay, Y. H., and Lau, P. Y. (2008, August). *One-dimensional grey-level co-occurrence matrices for texture classification*. IEEE In 2008 International Symposium on Information Technology, 3, 1-6.
- Tuncer, T. ve Engin, A. V. C. I. (2017). Yerel İkili Örüntü Tabanlı Veri Gizleme Algoritması: LBP-LSB. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 10(1), 48-53.
- Urbonas, A., Raudonis, V., Maskeliūnas, R., and Damaševičius, R. (2019). Automated identification of wood veneer surface defects using faster region-based convolutional neural network with data augmentation and transfer learning. *Applied Sciences*, 9(22), 4898-4918.
- VanderPlas, J. (2016). *Python data science handbook: Essential tools for working with data*. Sebastopol, Kaliforniya, ABD: O'Reilly Media, Inc, 1-529.
- Verma, M., Raman, B., and Murala, S. (2015). Local extrema co-occurrence pattern for color and texture image retrieval. *Neurocomputing*, 165, 255-269.
- Wang, F., Zhen, Z., Wang, B., and Mi, Z. (2017). Comparative study on KNN and SVM based weather classification models for day ahead short term solar PV power forecasting. *Applied Sciences*, 8(1), 28-51.
- Wang, H., Li, Z., and Zou, X. (2019). Wood surface defect detection based on adaboost and ESA. *Journal of System Simulation*, 31(8), 1636-1645.
- Wang, H., Ma, C., and Zhou, L. (2009, December). *A brief review of machine learning and its application*. IEEE In 2009 International Conference on Information Engineering and Computer Science, 1-4.
- Wang, W., Li, Y., Zou, T., Wang, X., You, J., and Luo, Y. (2020). A novel image classification approach via dense-MobileNet models. *Mobile Information Systems*, 1, 1-8.

- Weinberger, K. Q., and Saul, L. K. (2006, July). *An introduction to nonlinear dimensionality reduction by maximum variance unfolding*. AAAI'06: proceedings of the 21st national conference on Artificial intelligence - Cilt 2, 1683-1686.
- Westphal, E., and Seitz, H. (2021). A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks. *Additive Manufacturing*, 41, 101965-101981.
- Wu, C., Zou, X., and Yu, Z. (2022, December). *A Detection Method for Wood Surface Defect Based on Feature Fusion*. IEEE In 2022 4th International Conference on Frontiers Technology of Information and Computer (ICFTIC), 876-880.
- Wu, J., Liu, C., Cui, W., and Zhang, Y. (2019, November). *Personalized Collaborative Filtering Recommendation Algorithm based on Linear Regression*. In 2019 IEEE International Conference on Power Data Science (ICPDS), 139-142.
- Xie, D. Y. (2013). Analysis to situation and countermeasure of wood manufacture industry of our country. *Forest Investigation Design*, 3, 85-92.
- Xu, M., Watanachaturaporn, P., Varshney, P. K., and Arora, M. K. (2005). Decision tree regression for soft classification of remote sensing data. *Remote Sensing of Environment*, 97(3), 322-336.
- Yan, H., Ye, Q., Zhang, T. A., Yu, D. J., Yuan, X., Xu, Y., and Fu, L. (2018). Least squares twin bounded support vector machines based on L1-norm distance metric for classification. *Pattern Recognition*, 74, 434-447.
- Yang, B., and Chen, S. (2013). A comparative study on local binary pattern (LBP) based face recognition: LBP histogram versus LBP image. *Neurocomputing*, 120, 365-379.
- Yang, H., and Yu, L. (2017). Feature extraction of wood-hole defects using wavelet-based ultrasonic testing. *Journal of Forestry Research*, 28(2), 395-402.
- Yang, Y., Zhou, X., Liu, Y., Hu, Z., and Ding, F. (2020). Wood defect detection based on depth extreme learning machine. *Applied Sciences*, 10(21), 7488.
- Yap, Y. R. (2019). *Wood defect detection and classification using deep learning*. Master's thesis, University of Malaya, Malaysia.
- Yazan, E., Talu, M. F. (2017). *Comparison of the stochastic gradient descent based optimization techniques*. IEEE In 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), 1-5.
- Ye, Q., Zhao, H., Li, Z., Yang, X., Gao, S., Yin, T., and Ye, N. (2017). L1-Norm distance minimization-based fast robust twin support vector  $k$ -plane clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9), 4494-4503.
- YongHua, X., and Jin-Cong, W. (2015). Study on the identification of the wood surface defects based on texture features. *Optik-International Journal for Light and Electron Optics*, 126(19), 2231-2235.

- Zeebaree, D. Q., Abdulazeez, A., Zebari, D. A., Haron, H., and Hamed, H. N. A. (2021). Multi-Level Fusion in Ultrasound for Cancer Detection Based on Uniform LBP Features. *Computers, Materials and Continua*, 66(3), 3363-3382.
- Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv*, 1-6.
- Zeiler, M. D., and Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks. *arXiv*, 1-9.
- Zhang, H., Guan, C., and Wen, J. (2016). Applications and research development of nondestructive testing of wood based materials. *Journal of Forestry Engineering*, 1(6), 1-9.
- Zhang, Y., Xu, C., Li, C., Yu, H., and Cao, J. (2015). Wood defect detection method with PCA feature fusion and compressed sensing. *Journal of Forestry Research*, 26, 745-751.
- Zheng, A., and Casari, A. (2018). *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. Sebastopol, California, ABD: O'Reilly Media, Inc, 1-215.





*Gazili olmak ayrıcalıktır*