

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**GERÇEK ZAMAN-İMGE İŞLEME TEMELLİ  
KUMAŞ KALİTE KONTROL SİSTEMİ**

**DOKTORA TEZİ**

**Mehmet ERDOĞAN**

**Mekatronik Mühendisliği Anabilim Dalı**

**Mekatronik Mühendisliği Programı**

**MART 2025**



**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**GERÇEK ZAMAN-İMGE İŞLEME TEMELLİ  
KUMAŞ KALİTE KONTROL SİSTEMİ**

**DOKTORA TEZİ**

**Mehmet ERDOĞAN  
(518172020)**

**Mekatronik Mühendisliği Anabilim Dalı**

**Mekatronik Mühendisliği Programı**

**Tez Danışmanı: Prof. Dr. Mustafa DOĞAN**

**MART 2025**



**ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL**

**REAL TIME-IMAGE PROCESSING BASED  
FABRIC QUALITY CONTROL SYSTEM**

**Ph.D. THESIS**

**Mehmet ERDOĞAN  
(518172020)**

**Department of Mechatronics Engineering**

**Mechatronics Engineering Programme**

**Thesis Advisor: Prof. Dr. Mustafa DOĞAN**

**MARCH 2025**



İTÜ, Lisansüstü Eğitim Enstitüsü'nün 518172020 numaralı Doktora Öğrencisi Mehmet ERDOĞAN, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “GERÇEK ZAMAN-İMGE İŞLEME TEMELLİ KUMAŞ KALİTE KONTROL SİSTEMİ” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

**Tez Danışmanı :** **Prof. Dr. Mustafa DOĞAN** .....  
İstanbul Teknik Üniversitesi

**Jüri Üyeleri :** **Prof. Dr. Mehmet Turan SÖYLEMEZ** .....  
İstanbul Teknik Üniversitesi

**Doç. Dr. Janset DAŞDEMİR** .....  
Yıldız Teknik Üniversitesi

**Prof. Dr. Ufuk SAKARYA** .....  
Yıldız Teknik Üniversitesi

**Prof. Dr. Müjde GÜZELKAYA** .....  
İstanbul Teknik Üniversitesi

**Teslim Tarihi** : 10 Ocak 2025  
**Savunma Tarihi** : 07 Mart 2025







*Canım Babama,*



## ÖNSÖZ

Tezimin bu aşamaya gelmesinde bana destek olan birçok kişiye teşekkür etmek isterim. Öncelikle, araştırmam süresince bana sağladıkları destek ve olanaklar için Eliar Elektronik firmasına sonsuz teşekkürlerimi sunarım. Firmanın katkıları, çalışmamın başarısında büyük bir rol oynamıştır.

Ailem ve eşim her zaman yanımda olarak bana güç verdi. Zorlu süreçlerde gösterdikleri sabır, anlayış ve sevgi için onlara minnettarım. Her adımda beni destekledikleri için, bu tezin bir parçası olarak onların emeklerini de görmekteyim.

Ayrıca, tez savunmam sırasında değerli katkıları, yapıcı eleştirileri ve yol gösterici değerlendirmeleriyle çalışmamın gelişmesine yardımcı olan izleme komitesi ve jüri üyeleri hocalarıma da içten teşekkürlerimi sunarım.

Son olarak, danışmanım Sayın Mustafa DOĞAN'a teşekkürü bir borç bilirim. Kendisi, bilgi birikimi ve sürekli desteğiyle bu tezin oluşumunda bana rehberlik etti. Verdiği değerli öneriler ve gösterdiği yol, çalışmamın kalitesini artırdı ve beni her zaman motive etti.

Bu tez, bahsi geçen kişilerin ve kurumların katkılarıyla hayat bulmuştur. Her birine ayrı ayrı teşekkür eder, şükranlarımı sunarım.

Ocak 2025

Mehmet ERDOĞAN  
(Elektrik Mühendisi)



## İÇİNDEKİLER

### Sayfa

|                                                                       |           |
|-----------------------------------------------------------------------|-----------|
| ÖNSÖZ .....                                                           | ix        |
| İÇİNDEKİLER .....                                                     | xi        |
| KISALTMALAR .....                                                     | xiii      |
| SEMBOLLER .....                                                       | xv        |
| ÇİZELGE LİSTESİ .....                                                 | xvii      |
| ŞEKİL LİSTESİ .....                                                   | xix       |
| ÖZET .....                                                            | xxi       |
| SUMMARY .....                                                         | xxiii     |
| <b>1. GİRİŞ .....</b>                                                 | <b>1</b>  |
| 1.1 Tezin Amacı .....                                                 | 1         |
| 1.2 Literatür Araştırması .....                                       | 1         |
| 1.3 Hipotez .....                                                     | 13        |
| <b>2. MATERYAL VE YÖNTEMLER .....</b>                                 | <b>15</b> |
| 2.1 Kamera Düzenegi .....                                             | 15        |
| 2.1.1 Alan tarama kameraları .....                                    | 15        |
| 2.1.1.1 UV (Ultraviyole) kameralar .....                              | 15        |
| 2.1.1.2 Termal ve IR (Kızılötesi) kameralar .....                     | 15        |
| 2.1.1.3 Renk kameraları .....                                         | 16        |
| 2.1.2 Çizgi tarama kameraları .....                                   | 16        |
| 2.1.3 Hiperspektral kameralar .....                                   | 16        |
| 2.1.4 Multispektral kameralar .....                                   | 17        |
| 2.1.5 3D görüntüleme kameraları .....                                 | 17        |
| 2.1.6 Endüstriyel akıllı kameralar .....                              | 17        |
| 2.1.7 Lensler .....                                                   | 17        |
| 2.2 Görüntü İşleme Donanımları .....                                  | 18        |
| 2.2.1 PC tabanlı görüntü işleme sistemleri .....                      | 19        |
| 2.2.2 Gömülü görüntü işleme sistemleri .....                          | 19        |
| 2.2.3 Bağımsız görüntü işleme sistemleri .....                        | 19        |
| 2.3 Haberleşme Arayüzleri .....                                       | 20        |
| 2.3.1 CSI haberleşme .....                                            | 20        |
| 2.3.2 USB haberleşme .....                                            | 21        |
| 2.3.3 GigE haberleşme .....                                           | 21        |
| 2.4 Aydınlatma Çeşitleri .....                                        | 22        |
| 2.5 Test Düzenegi .....                                               | 23        |
| 2.6 Görüntü Veri Setinin Hazırlanması .....                           | 27        |
| <b>3. İSTATİKSEL VE SPEKTRAL HATA TANIMA ALGORİTMALARI .....</b>      | <b>29</b> |
| 3.1 Sobel, Prewitt, Roberts, Canny Kenar Algılama Algoritmaları ..... | 29        |
| 3.2 Kontur Algoritması .....                                          | 33        |
| 3.2.1 Kare izleme algoritması .....                                   | 33        |
| 3.2.2 Moore-Neighbor takip algoritması .....                          | 34        |

|                                                                       |            |
|-----------------------------------------------------------------------|------------|
| 3.3 Eğrilik Algoritması .....                                         | 35         |
| 3.3.1 Ayrik eğrilik algoritması.....                                  | 36         |
| 3.3.2 Modifiye edilmiş ayrik eğrilik algoritması .....                | 41         |
| 3.4 Spektral Yöntemler .....                                          | 43         |
| 3.4.1 Gabor dönüşümü .....                                            | 43         |
| 3.4.2 Curvelet dönüşümü.....                                          | 48         |
| 3.4.2.1 Matematiksel temeller .....                                   | 48         |
| 3.4.2.2 Anizotropik ölçeklendirme .....                               | 50         |
| 3.4.2.3 Curvelet dönüşümünün uygulamaları.....                        | 50         |
| 3.4.2.4 Curvelet dönüşümünün uygulamalarında iyileştirmeler .....     | 53         |
| 3.4.3 Hough dönüşümü .....                                            | 55         |
| 3.4.3.1 Matematiksel temeller .....                                   | 56         |
| 3.4.3.2 Hough dönüşümünün kumaş kusur tespitindeki uygulamaları ..... | 57         |
| <b>4. ÖĞRENME TABANLI HATA TANIMA ALGORİTMALARI .....</b>             | <b>61</b>  |
| 4.1 Evrişim Katmanı .....                                             | 61         |
| 4.2 ReLU Aktivasyon Fonksiyonu .....                                  | 63         |
| 4.3 Ortaklama Katmanı .....                                           | 63         |
| 4.4 Yığın Normalleştirme.....                                         | 64         |
| 4.5 Unutma .....                                                      | 64         |
| 4.6 Düzleştirme Katmanı .....                                         | 65         |
| 4.7 Tam Bağlı Katman .....                                            | 66         |
| 4.8 Transfer Öğrenme .....                                            | 66         |
| 4.9 Derin Öğrenme Uygulamaları .....                                  | 68         |
| 4.9.1 Basit ESA model uygulaması .....                                | 68         |
| 4.9.1.1 Model mimarisi .....                                          | 68         |
| 4.9.1.2 Modelin değerlendirilmesi ve tahmin yapılması.....            | 72         |
| 4.9.2 RestNet50 model uygulaması .....                                | 75         |
| 4.9.2.1 Veri hazırlığı .....                                          | 75         |
| 4.9.2.2 Model mimarisi .....                                          | 76         |
| 4.9.2.3 Modelin eğitimi.....                                          | 78         |
| 4.9.2.4 Modelin değerlendirilmesi ve tahmin yapılması.....            | 79         |
| 4.9.2.5 Sonuç ve ileri adımlar.....                                   | 80         |
| 4.9.3 YoloV5 uygulaması.....                                          | 81         |
| 4.9.3.1 Veri setini hazırlama.....                                    | 82         |
| 4.9.3.2 Ortam kurulumu .....                                          | 82         |
| 4.9.3.3 Modelin eğitimi.....                                          | 83         |
| 4.9.3.4 Eğitilmiş model ile kusur tanıma .....                        | 83         |
| 4.9.4 Otoenkoder anomali tespiti uygulaması .....                     | 84         |
| 4.9.4.1 Veri kümesi ve eğitim .....                                   | 85         |
| 4.9.4.2 Modelin yapılandırılması.....                                 | 85         |
| 4.9.4.3 Anomali tespiti için latent alanın kullanımı.....             | 87         |
| <b>5. DENEYSEL DÜZENEK .....</b>                                      | <b>91</b>  |
| <b>6. SONUÇLAR VE GELECEK ÇALIŞMALAR.....</b>                         | <b>95</b>  |
| 6.1 Gelecek Uygulama Alanı .....                                      | 99         |
| <b>KAYNAKLAR.....</b>                                                 | <b>101</b> |
| <b>EKLER .....</b>                                                    | <b>109</b> |
| <b>ÖZGEÇMİŞ.....</b>                                                  | <b>111</b> |

## KISALTMALAR

|                 |                                                                          |
|-----------------|--------------------------------------------------------------------------|
| <b>AR</b>       | : Artırılmış gerçeklik                                                   |
| <b>BA</b>       | : Bayes Ağları                                                           |
| <b>CD</b>       | : Curvelet Dönüşümü                                                      |
| <b>CSI</b>      | : Kamera Seri Arayüzü (Camera Serial Interface)                          |
| <b>DSP</b>      | : Dijital Sinyal İşleme                                                  |
| <b>DFT</b>      | : Ayırık Fourier dönüşümü                                                |
| <b>EA</b>       | : Eğrilik Algoritması                                                    |
| <b>ESA(CNN)</b> | : Evrişimli Sinir Ağı                                                    |
| <b>FCM</b>      | : Fuzzy C-Means                                                          |
| <b>FFL</b>      | : Sabit Odak Uzunluğu (Fixed Focus Length)                               |
| <b>FPR</b>      | : Yanlış Pozitif Oranı                                                   |
| <b>GD</b>       | : Gabor Dönüşümü                                                         |
| <b>GLCM</b>     | : Gri Seviye Eşleşim Matrisi                                             |
| <b>HD</b>       | : Hough Dönüşümü                                                         |
| <b>IoU</b>      | : Kesişim Birleşim Oranı (Intersection over Union)                       |
| <b>KA</b>       | : Karar Ağaçları                                                         |
| <b>KDE</b>      | : Çekirdek Yoğunluk Tahmini (Kernel Density Estimation)                  |
| <b>KEYK</b>     | : K En Yakın Komşu                                                       |
| <b>MDEA</b>     | : Modifiye Edilmiş Eğrilik Algoritması                                   |
| <b>MIPI</b>     | : Mobile Industry Processor Interface                                    |
| <b>RMSE'ler</b> | : Ortalama kare hataları                                                 |
| <b>TPR</b>      | : Doğru Pozitif Oranı                                                    |
| <b>UDCT</b>     | : Uniform Ayırık Curvelet Dönüşümü (Uniform Discrete Curvelet Transform) |





## SEMBOLLER

|                         |                                                                               |
|-------------------------|-------------------------------------------------------------------------------|
| $C$                     | : Üçgenin çevreleyen çemberinin merkezi                                       |
| $\bar{x}_a, \bar{x}_b$  | : $x_i$ 'nin a-nıncı ve b-ninci komşusunu                                     |
| $c_{j,l,k}$             | : Curvelet Dönüşümü                                                           |
| $f_N$                   | : Curvelet serisinin ilk $N$ terimi kullanılarak elde edilen $f$ fonksiyonunu |
| $G(f)$                  | : Gabor dönüşümü                                                              |
| $j$                     | : Curvelet ölçeği                                                             |
| $k$                     | : Konum                                                                       |
| $\ell$                  | : Curvelet yönelim                                                            |
| $P_{j-1}, P_j, P_{j+1}$ | : Eğrilik çizgisi noktaları                                                   |
| $r_+, r_-$              | : Sonraki noktaya uzaklık, Önceki noktaya uzaklık                             |
| $r_j$                   | : Üçgenin yarıçapı                                                            |
| $s_A$                   | : Anomali puanı                                                               |
| $t$                     | : Zaman                                                                       |
| $\tau$                  | : Ölçeklendirme parametresi                                                   |
| $x, y$                  | : Piksel konumu                                                               |
| $\xi$                   | : Frekans değişkeni                                                           |
| $\lambda$               | : Lambda dalga boyu                                                           |
| $\theta$                | : Teta yönelim açısı                                                          |
| $\psi$                  | : Fi faz açısı                                                                |
| $\sigma$                | : Sigma                                                                       |
| $\gamma$                | : Gamma                                                                       |
| $\psi_{j,l,k}(x)$       | : Curvelet temel fonksiyonu                                                   |



## ÇİZELGE LİSTESİ

|                                                                                                                                                                                                 | <u>Sayfa</u> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| Çizelge 2.1 : Kamera haberleşmesi karşılaştırması.....                                                                                                                                          | 22           |
| Çizelge 3.1 : Eğrilik karmaşıklık matrisi. ....                                                                                                                                                 | 40           |
| Çizelge 3.2 : Modifiye edilmiş eğrilik karmaşıklık matrisi. ....                                                                                                                                | 42           |
| Çizelge 3.3 : Gabor dönüşümü karmaşıklık matrisi. ....                                                                                                                                          | 47           |
| Çizelge 3.4 : Curvelet dönüşümü karmaşıklık matrisi. ....                                                                                                                                       | 55           |
| Çizelge 3.5 : Hough dönüşümü karmaşıklık matrisi. ....                                                                                                                                          | 60           |
| Çizelge 4.1 : Kullanılan ESA modelleri, boyutları ve parametre sayıları. ....                                                                                                                   | 68           |
| Çizelge 4.2 : ESA karmaşıklık matrisi. ....                                                                                                                                                     | 72           |
| Çizelge 4.3 : Restnet50 karmaşıklık matrisi. ....                                                                                                                                               | 81           |
| Çizelge 4.4 : YOLOv5 karmaşıklık matrisi. ....                                                                                                                                                  | 83           |
| Çizelge 6.1 : Hata tiplerinde başarı oranı. (Eğrilik Algoritması, Modifiye Eğrilik Algoritması, Gabor Dönüşümü, Curvelet Dönüşümü, Hough Dönüşümü, Evrişimli Sinir Ağı, RestNet50, YoloV5)..... | 95           |
| Çizelge 6.2 : Hata başarı oranı. (Eğrilik Algoritması, Modifiye Eğrilik Algoritması, Gabor Dönüşümü, Curvelet Dönüşümü, Hough Dönüşümü, Evrişimli Sinir Ağı, RestNet50, YoloV5).....            | 96           |



## ŞEKİL LİSTESİ

### Sayfa

|                                                                                                                                                                                                                                                                           |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Şekil 2.1 : Lens çalışma biçimi. ....                                                                                                                                                                                                                                     | 18 |
| Şekil 2.2 : Prototip şasi montaj görüntüsü. ....                                                                                                                                                                                                                          | 23 |
| Şekil 2.3 : Taban tambur motor görüntüsü. ....                                                                                                                                                                                                                            | 24 |
| Şekil 2.4 : Prototip birleştirilmiş görüntü. ....                                                                                                                                                                                                                         | 24 |
| Şekil 2.5 : Aydınlatma sistemi. ....                                                                                                                                                                                                                                      | 25 |
| Şekil 2.6 : Denetleyici yapısı. ....                                                                                                                                                                                                                                      | 26 |
| Şekil 2.7 : Sistem genel görünüm.....                                                                                                                                                                                                                                     | 26 |
| Şekil 2.8 : Hatalı kumaş veri seti (delik, hatasız, iğne kırılması, likra kaçığı, may izi).<br>.....                                                                                                                                                                      | 28 |
| Şekil 3.1 : Sobel konvolüsyon matrisi. ....                                                                                                                                                                                                                               | 29 |
| Şekil 3.2 : Sobel konvolüsyon matrisi hesaplanması. ....                                                                                                                                                                                                                  | 30 |
| Şekil 3.3 : Prewitt çekirdek matrisleri. ....                                                                                                                                                                                                                             | 31 |
| Şekil 3.4 : Robert Cross konvolüsyon matrisleri. ....                                                                                                                                                                                                                     | 31 |
| Şekil 3.5 : Moore piksel kümesi.....                                                                                                                                                                                                                                      | 34 |
| Şekil 3.6 : Hatalı resim görüntüsü.....                                                                                                                                                                                                                                   | 35 |
| Şekil 3.7 : Hatalı resim Canny kenar metodu uygulanmış resmi.....                                                                                                                                                                                                         | 36 |
| Şekil 3.8 : Görüntünün eğimi sol tarafında olacak şekilde, saat yönünde yönlendirilmiş<br>(kırmızı ok) düz bir çizgi parçası, sınırlanmış yarıçapın tersi olarak,<br>seviye çizgisinin ayrıklaştırılması ve her bir tepe noktasında ayrık<br>eğriliğin hesaplanması. .... | 37 |
| Şekil 3.9 : Kontur uygulanmış hatalı kumaş resim parçaları. ....                                                                                                                                                                                                          | 39 |
| Şekil 3.10 : $\theta_j$ salınımlı daire (osculating circle) yarıçapı grafiği.....                                                                                                                                                                                         | 39 |
| Şekil 3.11 : Kontur üzerindeki noktaların yarıçap değerleri grafiği.....                                                                                                                                                                                                  | 40 |
| Şekil 3.12 : Kontur üzerindeki noktaların açı değerleri grafiği.....                                                                                                                                                                                                      | 40 |
| Şekil 3.13 : (A) Örnek eğri, (B) standart eğrilik, (C) optimize edilmiş eğrilik, (D)<br>verimli eğrilik.....                                                                                                                                                              | 42 |
| Şekil 3.14 : Hatalı kumaş ve gabor filtresi uygulanmış görüntüsü (1).....                                                                                                                                                                                                 | 45 |
| Şekil 3.15 : Hatalı kumaş ve gabor filtresi uygulanmış görüntüsü (2).....                                                                                                                                                                                                 | 45 |
| Şekil 3.16 : Hatalı kumaş ve gabor filtresi uygulanmış görüntüsü (3).....                                                                                                                                                                                                 | 46 |
| Şekil 3.17 : Hatalı kumaş ve gabor filtresi uygulanmış görüntüsü (4).....                                                                                                                                                                                                 | 46 |
| Şekil 3.18 : Kumaş hata test arayüzü (1). ....                                                                                                                                                                                                                            | 47 |
| Şekil 3.19 : Kumaş hata test arayüzü (2). ....                                                                                                                                                                                                                            | 47 |
| Şekil 3.20 : . a) Dalgacık Dönüşümü b) Curvelet Dönüşümünde kenarların temsili. ....                                                                                                                                                                                      | 49 |
| Şekil 3.21 : Orijinal görüntü, fark haritası ve işlenmiş görüntüler. ....                                                                                                                                                                                                 | 52 |
| Şekil 3.22 : Orijinal görüntü, filtresiz fark haritası ve fark işaretlenmiş görüntü. ....                                                                                                                                                                                 | 53 |
| Şekil 3.23 : Orijinal görüntü, filtresiz fark haritası ve fark işaretlenmiş görüntü. ....                                                                                                                                                                                 | 53 |
| Şekil 3.24 : Orijinal görüntü, filtresiz fark haritası ve fark işaretlenmiş görüntü. ....                                                                                                                                                                                 | 53 |
| Şekil 3.25 : Orijinal görüntü, filtresiz fark haritası ve fark işaretlenmiş görüntü. ....                                                                                                                                                                                 | 54 |
| Şekil 3.26 : Orijinal görüntü, filtreli fark haritası ve fark işaretlenmiş görüntü. ....                                                                                                                                                                                  | 54 |
| Şekil 3.27 : Orijinal görüntü, filtreli fark haritası ve fark işaretlenmiş görüntü. ....                                                                                                                                                                                  | 54 |
| Şekil 3.28 : Orijinal görüntü, filtreli fark haritası ve fark işaretlenmiş görüntü. ....                                                                                                                                                                                  | 54 |

|                                                                                                       |     |
|-------------------------------------------------------------------------------------------------------|-----|
| Şekil 3.29 : Orijinal görüntü, filtreli fark haritası ve fark işaretlenmiş görüntü. ....              | 54  |
| Şekil 3.30 : Orijinal görüntü, kenar filtresi uygulanmış görüntü, çıkış görüntüsü. ...                | 59  |
| Şekil 4.1 : Basit bir ESA mimarisi. ....                                                              | 61  |
| Şekil 4.2 : Evrişim işlemi, filtre ve aktivasyon haritası. ....                                       | 62  |
| Şekil 4.3 : ReLU fonksiyonu. ....                                                                     | 63  |
| Şekil 4.4 : Maksimum ortaklama ve ortalama ortaklama örneği. ....                                     | 64  |
| Şekil 4.5 : Standard sinir ağı ve unutma işlemi uygulanmış sinir ağı. ....                            | 65  |
| Şekil 4.6 : Düzleştirme işlemi örneği. ....                                                           | 65  |
| Şekil 4.7 : İkili sınıflandırma örneğinde tam bağlı katman (Sreenivas vd. 2020). ....                 | 66  |
| Şekil 4.8 : Farklı ince ayar stratejileri. ....                                                       | 67  |
| Şekil 4.9 : Model: "model_basic" ....                                                                 | 71  |
| Şekil 4.10 : ESA ile hata tespiti (1). ....                                                           | 72  |
| Şekil 4.11 : ESA ile hata tespiti (2). ....                                                           | 73  |
| Şekil 4.12 : ESA ile hata tespiti (3). ....                                                           | 73  |
| Şekil 4.13 : ESA ile hata tespiti (4). ....                                                           | 74  |
| Şekil 4.14 : ESA ile hata tespiti (5). ....                                                           | 74  |
| Şekil 4.15 : ESA ile hata tespiti (7). ....                                                           | 75  |
| Şekil 4.16 : RestNet50 modeli. ....                                                                   | 77  |
| Şekil 4.17 : RestNet50 mimarisi. ....                                                                 | 78  |
| Şekil 4.18 : ResNet50 model kayıp grafiği. ....                                                       | 78  |
| Şekil 4.19 : RestNet50 model doğruluğu. ....                                                          | 79  |
| Şekil 4.20 : RestNet50 ile hata sınıflandırma. ....                                                   | 80  |
| Şekil 4.21 : RestNet50 ile hata sınıflandırma. ....                                                   | 80  |
| Şekil 4.22 : RestNet50 ile hata sınıflandırma. ....                                                   | 81  |
| Şekil 4.23 : YOLOv5 ile hata etiketleme. ....                                                         | 82  |
| Şekil 4.24 : YOLOv5 ile hata tespiti. ....                                                            | 84  |
| Şekil 4.25 : YOLOv5 ile hata sonuçları ve geçmiş hata raporu. ....                                    | 84  |
| Şekil 4.26 : Anomali tanıma modeli. ....                                                              | 86  |
| Şekil 4.27 : Eğitim model doğruluğu ve kaybı. ....                                                    | 87  |
| Şekil 4.28 : Anomali hata tespiti. ....                                                               | 88  |
| Şekil 4.29 : Anomali olmayan hata tespiti. ....                                                       | 89  |
| Şekil 5.1 : Sistem modeli. ....                                                                       | 92  |
| Şekil 5.2 : Sistem Çalışma Diyagramı. ....                                                            | 93  |
| Şekil A.1 : Saha Test Görüntüleri: (a)Tek Plaka. (b)Çift Plaka. (c)Üstten Bağlantı (d) tek plaka .... | 110 |

# GERÇEK ZAMAN-İMGE İŞLEME TEMELLİ KUMAŞ KALİTE KONTROL SİSTEMİ

## ÖZET

Tekstil sektöründe üretimin her aşamasında kalite kontrol, küresel pazarda rekabet edebilmek için hayati bir unsurdur. Manuel kumaş kusur incelemesinin sorunları, hassasiyet eksikliği ve zaman alıcı olmasıdır. Bu nedenle erken ve doğru kumaş hatası belirlenmesi, kalite kontrolün kritik bir aşamasıdır. Başarılı bir otomatik kumaş hata inceleme sistemi oluşturmak için iki ana zorluk vardır: kusur tespiti ve kusur sınıflandırılması. Geleneksel olarak, kumaş kusurlarının sınıflandırılması verimsiz ve emek yoğun bir süreç olan insan görsel muayenesiyle yapılmaktadır. Artan kumaş hata çeşitliliği nedeniyle, kumaş ürünlerinin kalitesini garanti altına almak için daha yüksek doğrulukla hataları sınıflandırabilen etkili yöntemlerin geliştirilmesi gerekmektedir. Tekstil kumaş malzemeleri ve ürünleri için otomatik kalite güvencesi, gerçek dünya uygulamalarındaki en karmaşık algoritmaların kullanıldığı yapay görme problemlerinden biridir [1-3].

Örme işlemi sırasında, kumaş ipliğinin kalitesindeki ve üretim ile çalışma koşullarındaki rastgele değişiklikler genellikle boyut, şekil, görünüm ve renk bakımından değişen dinamik hatalara yol açar. Tekstil ürün kalitesinde görsel denetimden kaynaklanan ekonomik faydalar çok büyüktür ve ürün kalite güvencesi için otomatik bilgisayarlı görüntü işleme çözümlerine yapılan yatırımları haklı çıkarmaktadır. En yetenekli denetçilerin bile kumaş hatalarının yalnızca yaklaşık %70'ini tespit edebildiği ve kumaş hatalarının üretilen kumaşların değerini yaklaşık %45-65 oranında azalttığı tahmin edilmektedir. Mevcut tespit sistemlerinin maliyetleri önemli ölçüde yüksektir ve tespit edebildikleri kusur türleri oldukça sınırlıdır. Düşük maliyetli yüksek hızlı bilgisayarlar, yüksek çözünürlüklü dijital kameralar ve düşük maliyetli depolamanın artan kullanılabilirliği, güçlü otomatik tekstil denetim çözümlerinin gelişeceğini ve yaygınlaşacağını göstermektedir [4-6].

Tez çalışmasında, histogram tabanlı yöntemler, renk tabanlı yaklaşımlar, görüntü segmentasyon tekniği, frekans dönüşümü ile algılama, doku tabanlı kusur algılama, görüntü morfolojisi işlemleri ve derin öğrenme yöntemlerine ilişkin kapsamlı bir genel çözüm üzerinde durulmaktadır. Araştırma, farklı kumaş kusurlarını tespit etmek için bilgisayarla görme ve dijital görüntü işleme uygulamaları kullanarak akan kumaşlarda hata tespiti için gerçek zamanlı ve yüksek performanslı çalışan algoritmaları gerektirmektedir. Bu nedenle, grafik kartı tabanlı geliştirmeler ile tespit ve sınıflandırma yapacak, ayrıca farklı yöntemler kullanarak kumaşın hız ve genişlik bilgilerini de hesaplayarak hata geometrisini kaydetmeyi hedeflemektedir.





# **REAL TIME-IMAGE PROCESSING BASED FABRIC QUALITY CONTROL SYSTEM**

## **SUMMARY**

Quality control at every stage of production in the textile industry is a vital factor for competing in the global market. The textile industry's reliance on high-quality production demands a robust inspection system to ensure minimal defects in fabric products. Traditionally, manual inspection methods have been the primary approach to fabric defect detection. However, these methods suffer from critical limitations, such as a lack of precision and high time consumption. These shortcomings often lead to missed defects or inconsistencies in inspection outcomes, highlighting the need for more reliable and efficient solutions. Consequently, early and accurate fabric defect identification has become a critical phase of quality control in the textile production process, ensuring consistent product quality and meeting customer expectations [1-3].

The primary challenges in developing a successful automatic fabric defect inspection system are defect detection and defect classification. Defect detection involves identifying anomalies on the fabric surface, while defect classification requires categorizing these anomalies into predefined types based on their characteristics. Manual methods, which involve human visual inspection, are inherently inefficient and labor-intensive. These methods also pose challenges related to inspector fatigue and subjective judgment, which can result in variability and errors in defect detection. With the increasing diversity and complexity of fabric defects, such traditional approaches are no longer sufficient. Thus, the development of automatic methods capable of achieving higher accuracy in defect detection and classification has become imperative for maintaining product quality and meeting global market demands.

Automatic quality assurance for textile materials and products represents one of the most complex artificial vision problems. This complexity arises from the need to analyze dynamic and variable defect patterns in real time. The implementation of advanced algorithms in real-world applications has shown promising results in addressing these challenges. Several state-of-the-art approaches leverage developments in artificial intelligence, digital image processing, and computer vision technologies to create innovative solutions for automatic defect inspection systems. These technologies provide the capability to process large volumes of data, identify intricate defect patterns, and adapt to various production conditions. By integrating these advanced methodologies, the industry can achieve significant improvements in precision, reliability, and speed compared to manual inspection.

During the knitting process, random variations in yarn quality and operational conditions often result in dynamic defects. These defects differ significantly in their size, shape, appearance, and color, making their detection a complex task. For instance, defects such as holes, slubs, stains, or misalignments may vary across different fabric types and production conditions. Such variability necessitates the use of adaptive algorithms capable of handling a wide range of defect types and severities. Moreover, the economic implications of these defects are substantial, as fabric defects can reduce the value of the produced materials by approximately 45-65%. This loss impacts not only the profitability of manufacturers but also the perception of quality by customers. Furthermore, it is estimated that even the most skilled inspectors can detect only about 70% of fabric defects during manual inspections. These limitations underscore the necessity of investing in automated solutions for fabric quality assurance, which can address both economic and operational challenges in the industry [4-6].

The adoption of automated computer vision-based systems for textile quality control has several economic and operational advantages. These systems enable the detection and classification of a wider range of defects while significantly reducing the time and labor involved. Additionally, advancements in low-cost, high-speed computing, high-resolution digital cameras, and efficient data storage solutions have made automated inspection systems increasingly accessible and practical. With the reduction in costs associated with these technologies, small- and medium-sized enterprises are now able to implement advanced inspection systems that were previously cost-prohibitive. As these technologies continue to evolve, the adoption of powerful automated textile inspection solutions is expected to become more widespread, leading to higher standards of quality control across the industry. Furthermore, automated systems ensure consistent inspection quality, reducing the likelihood of customer complaints and returns, thereby strengthening brand reputation and customer loyalty.

The methodologies explored in this thesis are designed to address these multifaceted challenges. By presenting a comprehensive general solution for automatic fabric defect detection and classification, the research aims to improve existing systems significantly. The proposed methodologies incorporate a variety of techniques, including histogram-based methods, color-based approaches, image segmentation techniques, frequency transformation-based detection, texture-based defect detection, image morphology operations, and deep learning algorithms. Each of these approaches contributes to enhancing the system's ability to identify and classify defects with greater accuracy and efficiency. For instance, histogram-based methods provide a statistical foundation for identifying anomalies, while frequency transformation techniques, such as Fourier and wavelet analysis, allow for the detection of periodic defects. Deep learning methods, on the other hand, introduce adaptability and precision by leveraging large datasets and training neural networks to recognize intricate patterns. Texture-based detection, combined with image morphology operations, further refines the accuracy by capturing microstructural differences in fabric surfaces.

To ensure real-time performance, the research focuses on developing high-performance algorithms capable of processing flowing fabric data. These algorithms leverage computer vision and digital image processing technologies to analyze the fabric surface dynamically. Additionally, the thesis explores the use of graphics card-based enhancements to accelerate computation and improve overall system performance. By integrating these techniques, the system not only detects defects but also records their geometry by calculating the speed and width information of the fabric. This detailed analysis enables manufacturers to trace defects back to specific stages in the production process, facilitating targeted interventions and process improvements. Moreover, real-time monitoring systems allow for immediate feedback and corrective actions, minimizing waste and downtime in production lines.

In conclusion, the economic and operational advantages of automated fabric defect inspection systems make them an indispensable tool for modern textile production. By addressing the limitations of manual inspection and leveraging advanced technologies, this thesis aims to contribute to the development of efficient, reliable, and cost-effective solutions for ensuring high-quality textile products. The integration of innovative methodologies and cutting-edge technologies underscores the potential for transformative advancements in fabric quality assurance, paving the way for a more competitive and sustainable textile industry. The outcomes of this research are expected to have a far-reaching impact, not only in improving defect detection rates but also in enhancing the overall efficiency and sustainability of textile manufacturing processes. Furthermore, the successful implementation of such systems will encourage ongoing innovation, setting new benchmarks for quality and reliability in the textile industry.



## 1. GİRİŞ

Tekstil endüstrisinde üretimin her aşamasında kalite kontrol, küresel pazarda varlığını sürdürebilmek için kilit bir unsurdur. Manuel kumaş kusuru incelemesinin sorunları, hassasiyet eksikliği ve yüksek zaman tüketimidir; bu yüzden erken ve doğru kumaş kusuru tespiti, kalite kontrolün önemli bir aşamasıdır. Başarılı bir otomatik kumaş kusuru inceleme sistemi elde etmek için çözülmesi gereken iki ana zorluk vardır: Kumaş hata tespiti ve kusur sınıflandırılması. Kumaş kusur sınıflandırması, geleneksel olarak verimsiz ve yoğun emek gerektiren insan görsel muayenesiyle yapılmaktadır. Artan kumaş kusurları çeşitliliği ile birlikte, kumaş ürünlerinin kalitesinin sağlanmasına katkıda bulunabilecek daha yüksek bir doğrulukla kusurları sınıflandırmak için etkili yöntemler geliştirmek gerekmektedir. Tekstil kumaş üretimi gerçekleştiren firmalar açısından yüksek kalite güvencesinin sağlanması, gerçek endüstriyel uygulamalarda karşılaşılan en zorlu bilgisayarla görme problemlerinden biri olarak öne çıkmaktadır.

### 1.1 Tezin Amacı

Kumaş üzerinde kamera ile hata kontrolü yapan sistemin amacı, üretim sürecinde ortaya çıkabilecek hataların tespit edilmesi, kaydedilmesi, sınıflandırılması ve raporlanmasıdır. Bu sistem, yüksek çözünürlüklü kameralar kullanarak kumaştaki kusurları otomatik olarak algılamakta ve bu hataların detaylarını kayıt altına almaktadır. Ardından, tespit edilen hatalar belirli kategorilere ayrılarak sınıflandırılır ve sistematik bir şekilde depolanır. Toplanan veriler, operatörlere anlık olarak bildirilir ve düzenli aralıklarla firmalara rapor olarak sunulmaktadır. Bu sayede, kalite kontrol süreçleri iyileştirilir ve üretim verimliliği artırılması amaçlanmaktadır.

### 1.2 Literatür Araştırması

“Kumaş Hatalarının Isıl Görüntüleme ve Görüntü İşleme Teknikleri ile Tespit Edilmesi” adlı Kazım Yıldız tarafından 2014 yılında yayımlanan doktora tezinde kumaş hata denetimi ile ilgili iki farklı çalışma gerçekleştirilmiştir. İlk çalışma hatalı

kumaş görüntülerine, görüntü işleme algoritmaları uygulanmıştır. Hatalı alanın tespitinden sonra şekilsel ve histogram özellikleri çıkarılmıştır. Bu özellikler kullanılarak Fuzzy C-Means (FCM) algoritması ile kümeleme işlemi gerçekleştirilmiştir. Ayrıca tespit edilen hatanın kumaş üzerindeki dikey ve yatay koordinatlarının belirlenmesi de gerçekleştirilmiştir. İkinci çalışma ise kaydedilen video üzerinde hatalı görüntünün tespiti yapılmıştır. Hata, K En Yakın Komşu (KEYK), Bayes Ağları (BA) ve Karar Ağaçları (KA) algoritmaları kullanılarak anlık olarak sınıflandırılmıştır. Ayrıca hatanın yeri, cinsi ve kumaşın hangi metresinde yer aldığı belirlenmiştir [93].

“Yuvarlak Örgü Makineleri için Görüntü İşleme Tabanlı Kumaş Hatası Tespit Sistemi” adlı Kazım Hanbay tarafından 2016 yılında yayımlanan doktora tezinde yuvarlak örgü makinesi üzerinde gerçek zamanlı çalışabilen bir kumaş hatası tespit sistemi geliştirilmiştir. Bu sistem görüntü elde etme düzeneğinin kurulması; kumaş veri tabanının inşası, hata tespit yöntemlerinin geliştirilmesi; üretim esnasında hata tespit süreçlerini içermektedir. Tez sürecinde yapılan çalışmalar neticesinde 6 farklı kumaş tipi ve 10 farklı kumaş hatasını içeren yeni bir kumaş veri tabanı inşa edilmiştir. Bu görüntülerin uzaysal ve frekans alanında öznelilikleri elde edilmiş ve sınıflandırılma işlemleri gerçekleştirilmiştir. Tezin en önemli katkısı 6 yeni öznelilik çıkarma yöntemini literatüre kazandırmış olmasıdır: GDF-HOG, Eig(Hess)-HOG, ikinci mertebe HOG, Eig(Hess)-CoHOG, GM-CoHOG ve yüzey etiketleme temelli CoHOG. Her bir yöntemin klasik yöntemler ile karşılaştırmalı olarak sağladığı üstünlükler ve eksiklikler tez içerisinde detaylı bir şekilde irdelenmiştir. Uzaysal alandaki bu yöntemlere ek olarak, Fourier, dalgacık ve shearlet dönüşüm yöntemleri kullanılarak spektral kumaş görüntülerinin belirli istatistiksel öznelilikleri çıkarılmış ve sınıflandırılmaları sağlanmıştır. Sınıflandırıcı olarak Yapay Sinir Ağları kullanılmıştır [94].

“Development of a New Software for Fabric Defect Detection and Classification Using Image Processing and Machine Learning Methods” adlı Ahmad Mones NAWAF tarafından 2019 yılında yayımlanan yüksek lisans tezinde, kumaş yüzeyindeki hataları otomatik olarak tespit etmek ve sınıflandırmak için görüntü işleme ve derin öğrenme yöntemlerine dayalı bir uygulama oluşturmaktır. Kusur tespitinde Fourier dönüşümü (DFT), Gabor filtreleri kullanılmış, Fast-R CNN sınıflandırmada kullanılmıştır. Ortalama kare hataları (RMSE'ler) tespit ve

sınıflandırma hatalarını hesaplamak için kullanılmıştır. Elde edilen sonuçlara dayanarak, görüntü işleme yöntemlerinin, özellikle DFT'nin hata tespitinde kabul edilebilir bir doğrulukla kullanılabileceği kanıtlanmıştır [95].

“Real time fabric defect detection system on an embedded DSP platform” adlı Jagdish Lal Raheja, Bandla Ajay, Ankit Chaudhary tarafından 2013 yılında yayımlanan makalede, endüstriyel kumaş üretimlerinde, küçük kusurları bulmak için otomatikleştirilmiş gerçek zamanlı bir sistem sunulmuştur. Hatalı ürünleri nakil etmeyerek maliyet tasarrufu sağlayacak ve aynı zamanda sadece kusurlu ürünleri göndererek kaliteli kumaşların şirket imajını oluşturmaya yardımcı olacaktır. Gömülü bir DSP platformunda uygulanan gerçek zamanlı bir kumaş hatası tespit sistemi (FDDS) burada sunulmaktadır. Kumaş görüntüsünün dokusal özellikleri, gri seviye eş oluşum matrisine (GLCM) dayalı olarak çıkarılır. Kusur tespiti için pencerenin tüm görüntü üzerinde hareket ettiği ve kumaş görüntüsünün GLCM'sinden bir dokusal enerji hesapladığı bir kayan pencere tekniği kullanılır. Enerji değerleri bir referansla karşılaştırılır ve bir eşiğin ötesindeki sapmalar kusur olarak rapor edilir ve ayrıca bir pencere ile görsel olarak temsil edilir. Uygulama, bir TI TMS320DM642 platformunda gerçekleştirilir ve kod geliştirici stüdyo yazılımı kullanılarak programlanır. Bu uygulamanın gerçek zamanlı çıktısı bir monitörde izlenme yapılmasına olanak vermektedir [96].

“Automatic Fabric Defect Detection Using Cascaded Mixed Feature Pyramid with Guided Localization” adlı Henry Y.T. Ngan, Grantham K.H. Pang, Nelson H.C. Yung tarafından 2011 yılında yayımlanan makalede, optik görüntü veri setlerinde kumaş hatası tespiti sistematik olarak incelenmiştir. Genel veri kümelerinin aksine, kusurlu görüntüler çok ölçekli, gürültülü ve bulanıktır. Arka ışık yoğunluğu da görsel algı için hassas olacaktır. Büyük ölçekli kumaş hatası veri kümeleri, bu dengesiz sorunları ele almak için endüstriyel uygulamada tespit gereksinimlerini karşılamak için toplanır, seçilir ve kullanılır. Daha iyi genelleme yapmak için geliştirilmiş iki aşamalı bir hata dedektörü oluşturulmuştur. Yığınlanmış özellik piramidi ağları, birinci aşamada derinlemesine karma bloğun enterpolasyonunda çapraz ölçek hata modellerini bir araya getirmek için ayarlanır. Özellik haritalarını paylaşarak, merkezlilik ve şekil dalları, önerilen kılavuzlu sabitleyicileri filtrelemek ve iyileştirmek için kademeli modülleri deforme olabilen evrişimle birleştirir. Dengeli örneklemeden sonra, ikinci aşamadaki kumaş kusur görüntüleri arasındaki etkileşimleri karakterize etmek için,

ilgililenen bölge için konuma duyarlı havuzlama yoluyla öneriler aşağı örneklenir. Deneyler, uçtan uca mimarının, mevcut dedektörlere kıyasla bölge tabanlı nesne dedektörlerinin tıkanmış kusur performansını iyileştirdiğini göstermektedir [97].

“Motif-based defect detection for patterned fabric” adlı Henry Y.T. Ngan, Grantham K.H. Pang, Nelson H.C. Yung tarafından 2018 yılında yayımlanan makalede, 2B desenli dokudaki 17 duvar kâğıdı grubundan 16'sında kusurları tespit etmek için genelleştirilmiş bir motif tabanlı yöntem önerilmektedir. Makalede çoğu desenli dokunun kafeslere ve bileşenlerine- motiflere ayrıştırılabileceğini varsayar. Ardından, hareketli çıkarma enerjisini ve farklı motifler arasındaki varyansını hesaplamak için motiflerin simetri özelliğini kullanır. Bu değerlerin bir dizi hatasız model üzerinden dağılımını öğrenerek, kusurlu ve hatasız modelleri ayırt etmek için sınır koşulları belirlenebilir. Bu makale, yöntemin teorik temellerini sunmakta ve motifler ile kafes arasındaki ilişkileri tanımlamaktadır; buradan hareketle çıkarma enerjisi adı verilen yeni bir kavram, motifin dairesel kaydırma matrisler ile kendisi arasında norm ölçüsü kullanılarak türetilmiştir. Bu yazıda, hareketli çıkarma enerjisinin kusurlu motifin kusur bilgisini arttırdığı gösterilmiştir. Varyansı ile birlikte, kusurlu ve hatasız motifleri sınıflandırmak için karar sınırlarının çizildiği bir enerji varyans alanı daha da tanımlanır. Desenli kumaştan oluşan 16 duvar kâğıdı grubu üç ana gruba dönüştürülebildiğinden, önerilen yöntem bu üç ana grup üzerinden değerlendirilir ve bu üç ana gruptan 160 hatasız kafes numunesi karar sınırlarını tanımlamak için kullanılır, 140 hatasız ve 113 test için kullanılan hatalı numunelerdir. Önerilen yöntem için %93,32'lik genel bir tespit başarı oranı elde edilmiştir. Daha önce sunulan başka hiçbir genelleştirilmiş yaklaşım bu başarı oranına ulaşamaz ve bu nedenle bu sonuç daha önce yayınlanan diğer tüm yaklaşımlardan daha iyi performans göstermektedir [98].

“Fabric defect detection via low-rank decomposition with gradient information and structured graph algorithm” adlı Boshan Shi, Jiuzhen Liang, Lan Di, Chen Chen, Zhenjie Hou tarafından 2021 yılında yayımlanan makalede, düşük aşamalı ayrıştırma modeli, bir matrisin hatasız bölgeyi (arka plan) temsil eden ve kusur alanını (ön plan) tanımlayan seyrek bir matrise ayrıştırıldığı kumaş hatası algılama potansiyelini gösterir. Ancak yine de iki eksiklik vardır. İlk olarak, kusurlu görüntünün dokusu yüksek gradyan özelliğine sahip olduğunda, mevcut model ile elde edilen seyrek matris hala çok sayıda tespit edilmemiş bölgelerin kenarlarını muhafaza etmektedir.



İkinci olarak, önceki bilgilerin kesin olmaması nedeniyle, çoğu model, küçük kusur alanı veya birden fazla döngü içeren kusur alanı ile uğraşırken, kusur bloğu etrafındaki kusursuz noktaları yanlış değerlendirecektir. Bu problemleri çözmek için, bu makale, gradyan bilgisinin düşük kademeli ayrıştırılmasına ve yapılandırılmış grafik algoritmasına dayanan bir kumaş hatası algılama yöntemi ortaya koymaktadır: 1) Yapılandırılmış grafik algoritması, kumaş kusur görüntüsünün özelliklerine göre, kumaş kusur görüntüsü, yerel özellik ve kusurlu hasar süresi ile hatasız bloğa ayrılır. 2) Birleştirme işleminde, kafes içi birleştirmeyi teşvik etmek ve kusurlu blokların ve çevredeki kusurlu olmayan blokların birleşmesini önlemek için mevcut blokta bulunan döngü sayısına göre uyarlanabilir bir eşik ayarlanır. 3) Segmentasyon sonuçlarından hesaplanan hata ön bilgisi, hatasız bölgeyi zayıflatmak ve seyrek terim altındaki kusur alanını vurgulamak için matris ayrışmasına rehberlik etmek için kullanılır. Model standart bir veritabanı üzerinde değerlendirilmiştir ve en son dört yöntemle karşılaştırılmıştır. Bu yöntemin toplam TPR ve fpr değerleri, en iyi performans olan kutu, yıldız ve nokta veri tabanlarında sırasıyla %87,3 ve %1,21'dir [99].

“Fabric defect detection systems and methods—A systematic literature review” adlı Kazım Hanbay, Muhammed Fatih Talu, Ömer Faruk Özgüven tarafından 2016 yılında yayımlanan makalede, kumaş hatası tespit yöntemlerinin kapsamlı bir literatür incelemesini sunmaktadır. İlk olarak, kamera ve lens gibi temel görüntü elde etme sistemi bileşenlerini kısaca açıklamaktadır. Hata tespit yöntemleri yapısal, istatistiksel, spektral, model tabanlı, öğrenme, hibrit ve karşılaştırma çalışmaları olarak yedi sınıfa ayrılır. Bu yöntemler doğruluk, hesaplama maliyeti, güvenilirlik, dönme / ölçekleme değişmezliği, çevrimiçi / çevrimdışı çalışma yeteneği ve gürültü hassasiyeti gibi kriterlere göre değerlendirilir. Her yaklaşımın güçlü ve zayıf yönleri karşılaştırmalı olarak vurgulanmaktadır. Ayrıca makinelerde dokuma ve örme yöntemlerinin kullanılabilirliği araştırılmıştır. Mevcut inceleme çalışmaları, tekstil ve bilgisayar görüşü alanında araştırma yapan okuyucular için kumaş hatası tespit sistemleri hakkında yeterli bilgi sağlamamaktadır. Görüntü elde etme sisteminin verimli bir şekilde kurulması için bir dizi bilgi eklenmiştir. Özellikle lens ve ışık kaynağı seçimi matematiksel olarak ifade edilmektedir [1].

“A visual long-short-term memory based integrated CNN model for fabric defect image classification” adlı Yudi Zhao, Kuangrong Hao, Haibo He, Xuesong Tang, Bing Wei tarafından 2020 yılında yayımlanan makalede anlatılan, kumaş kusur

sınıflandırması, geleneksel olarak verimsiz ve yoğun emek gerektiren insan görsel muayenesiyle elde edilmesine bir çözüm getirmeye çalışmaktadır. Bu nedenle, bu sorunu çözmek için akıllı ve otomatik yöntemler kullanmak, sıcak bir araştırma konusu haline geldiği belirtilmektedir. Artan kumaş kusurları çeşitliliği ile birlikte, kumaş ürünlerinin kalitesinin sağlanmasına katkıda bulunabilecek daha yüksek bir doğrulukla kusurları sınıflandırmak için etkili yöntemler tasarlamak önem kazanmıştır. Kumaş kusurunun doku arka planında açık olmadığı ve birçok türünün ayırt edilemeyecek kadar kafa karıştırıcı olduğu göz önüne alındığında, bu makalede görsel bir uzun-kısa süreli bellek (visual long-short-term memory) tabanlı entegre CNN modeli önerilmiştir. İnsan görsel algısı ve görsel bellek mekanizmasından esinlenerek, üç özellik kategorisi çıkarılır; bunlar, yığılmış evrişimli otomatik kodlayıcılar (stacked convolutional auto-encoders) tarafından çıkarılan görsel algı (visual perception) bilgileri, görsel kısa süreli bellek (visual short-term memory) ile karakterize edilir. Sığ evrişimli sinir ağı (convolutional neural network) ve yerel olmayan sinir ağları ile karakterize edilen görsel uzun süreli bellek (visual long-term memory) bilgileri ile üç kumaş hatası veri setine ilişkin deneysel sonuçlar, önerilen modelin kumaş hatası sınıflandırması ile ilgili mevcut son teknoloji yöntemlere rekabetçi sonuçlar sağladığı gösterilmiştir [99].

“Fabric defect detection using morphological filters” adlı K.L. Mak, P. Peng, K.F.C. Yiu tarafından 2009 yılında yayımlanan makalede, dokuma kumaşlar için otomatik hata algılama sorununu çözmek için morfolojik filtrelerle dayalı yeni bir hata tespit şeması önerilmektedir. Önerilen şemada, tekstil kumaşının önemli doku özellikleri önceden eğitilmiş bir Gabor dalgacık ağı kullanılarak çıkarılmıştır. Bu doku özellikleri daha sonra kumaş arka planını çıkarmak ve kusurları izole etmek için sonraki morfolojik işlemlerde yapılandırma elemanlarının yapımını kolaylaştırmak için kullanılır. Önerilen hata tespit şeması yalnızca birkaç morfolojik filtre gerektirdiğinden, ilgili hesaplama yükü miktarı önemli değildir. Önerilen şemanın performansı, farklı tipte yaygın kumaş kusurlarına sahip çok çeşitli homojen tekstil görüntüleri kullanılarak değerlendirilir. Elde edilen test sonuçları, düşük yanlış alarmlarla doğru hata tespiti sergiler, böylece önerilen tespit şemasının etkinliğini ve sağlamlığını gösterir. Ek olarak, önerilen tespit şeması, prototipli otomatik bir denetim sistemi kullanılarak gerçek zamanlı olarak daha da değerlendirilir [100].

“Automatic fabric defect detection with a wide-and-compact network” adlı Yuyuan Li, Dong Zhang, Dah-Jye Lee tarafından 2019 yılında yayımlanan makalede, kumaş hatalarının otomatik tespiti, tekstil endüstrisi için önemli bir süreç olduğu için hata algılama modeli sunulmuştur. Tespit doğruluğunun yanı sıra, kaynakları sınırlı bir sistem için otomatik bir kumaş hatası algılama çözümü, işlem süresi ve basitlik açısından üstün performans gerektirir. Bu makale, birkaç yaygın doku kusurunun tespiti için kompakt bir evrişimli sinir ağı mimarisi önermektedir. Önerilen mimari, ağı optimize etmek için çok katmanlı algılayıcıya sahip birkaç mikro mimari kullanır. Bir mikro mimarinin ana bileşeni, kompakt bir modelde algılama doğruluğunu iyileştirmek için çok ölçekli analiz, filtre ayrıştırma, çoklu konum havuzlama ve parametre azaltma teknikleri kullanılarak oluşturulur. Deneysel sonuçlar, ana akım evrişimli sinir ağı mimarileri ile karşılaştırıldığında, önerilen ağı çok daha küçük bir model boyutuyla algılama doğruluğu açısından üstün performans elde ettiğini göstermektedir. Yalnızca kumaş kusurlarının tespiti için değil, aynı zamanda birkaç genel veri kümesinde nesne tanıma için de iyi çalıştığı gösterilmektedir [101].

“Automated defect detection in uniform and structured fabrics using Gabor filters and PCA” adlı Lucia Bissi, Giuseppe Baruffa, Pisana Placidi, Elisa Ricci, Andrea Scorzoni, Paolo Valigi tarafından 2013 yılında yayımlanan makalede, TILDA görüntü veri tabanında test edilmiş, tek tip ve yapılandırılmış kumaşlarda doku kusur tespiti için geliştirilen algoritmayı açıklamaktadır. Önerilen yaklaşım, karmaşık bir simetrik Gabor filtre bankasına ve Ana Bileşen Analizine (Principal Component Analysis) dayanan bir özellik çıkarma aşamasında ve öklid özellik normuna ve bununla karşılaştırmalı olarak bir kusur tanımlama aşamasına göre kumaş tipine özel parametrelerle yapılandırılmıştır. Analizler, tek pikselleri dikkate almak yerine yama bazında gerçekleştirilmektedir. Performans, insan gözlemciler tarafından tespit edilen referans kusur konumları olarak kullanılarak, görünür doku ve ızgara benzeri yapılarla sahip tek tip dokulu kumaşlar ve kumaşlarla değerlendirilmiştir. Sonuçlar, algoritmanın çoğu durumda önceki yaklaşımlardan daha iyi performans gösterdiğini, %98,8’lik bir algılama oranına ve %0,20–0,37 gibi düşük bir yanlış alarm oranına ulaştığını gösterirken, ağır yapılandırılmış iplikler için yanlış algılama oranı %5 kadar düşük olabilmektedir [102].

“Defect Detection of Industrial Products based on Improved Hough Transform” başlıklı 2018 yılında Qingcai Ge, M.F., Xu, J tarafından hazırlanan çalışma,

endüstriyel üretim süreçlerinde merkezi simetri dağılım kurallarına sahip ürünlerin kusurlarını tespit etmek için geliştirilmiş bir Hough dönüşümü yöntemi sunmaktadır. Çalışmada, geleneksel yöntemlerin sınırlamaları analiz edilmiş ve endüstriyel ürünlerin merkezi bölgesinden başlayarak algılama hedefinin bulunduğu yöne doğru yönlü kümeleme yaklaşımı önerilmiştir. İlk olarak, ilgi bölgesinin (ROI) konturu ve merkezi belirlenmiş, ardından kontur üzerindeki noktalara Hough dönüşümü uygulanmıştır. Merkezi alan üzerinden geçen doğruların kısıt kurallarına uygun olarak oylama yapılmış ve oylama alanında elde edilen tepe noktaları kullanılarak endüstriyel ürün bileşenlerindeki kusurlar tespit edilmiştir. Deneysel sonuçlar, önerilen algoritmanın yüksek anti-parazit yeteneğine sahip olduğunu ve algılama hedefi ile arka plan arasındaki benzerlikten kaynaklanan algılanamazlık sorunlarını çözebildiğini göstermektedir. Ayrıca, bu yöntemin geleneksel Hough yöntemine kıyasla belirgin bir şekilde daha dayanıklı olduğu belirtilmiştir. Bu çalışma, merkezi simetri dağılım kurallarına sahip endüstriyel parçaların algılanmasında gerçek zamanlı işlem gerçekleştirebilen ve geleneksel yöntemlerin boşluklarını dolduran bir yaklaşım sunmaktadır. Deneysel sonuçlar, yöntemin merkezi simetri kurallarına uygun endüstriyel ürünleri kararlılıkla algılayabildiğini ve endüstriyel üretim süreçlerinde uygulanabilir olduğunu kanıtlamaktadır [54].

“Fabric defect detection using Discrete Curvelet Transform” adlı Anandan P, R S Sabeenian tarafından 2018 yılında yayımlanan makalede, moda pazarında kumaş seçimine yönelik artan müşteri talebiyle birlikte, kumaş dokusu çok çeşitli hale gelmekte ve bu çalışmada da hatasız kumaşın kontrolü için Curvelet Dönüşümü kullanarak bir çözüm sunulmaktadır. GLCM tabanlı, dalgacık temelli, ayrıca eğri çizgi temelli tekniklerin karşılaştırmalı bir çalışması da eklenmiştir. Planlanan teknikle elde edilen yüksek doğruluk, kumaş hatası için ekonomik bir çözüm önermektedir. Bu çalışma, ilk belgelenen, malzeme kusuru meselesini ele almak için dijital eğri dönüşümü olarak adlandırılan yepyeni bir çoklu-çözünürlüklü analiz aracının olasılıklarını araştırmak için düzenlenmesidir. Tanıyıcı, görüntü toplama cihazı ile dijital kumaş resimlerini alır ve bu görüntüyü Ayrık Eğrilik (Curvelet) Dönüşümü kullanarak ikili görüntüye dönüştürür. Önerilen algoritmik kural MATLAB'da simüle edilmiştir. Önerilen hata tespit modelinin performansı, çeşitli gerçek kumaş örnekleri ile derinlemesine deneyler aracılığıyla değerlendirilmiştir. Planlanan algılama modeli

etkili olmaya ve algılama doğruluğu ve yanlış alarm açısından bazı temsili algılama modellerinden üstün olmaya çalışılmıştır [45].

“Computer vision-based fabric defect analysis and measurement” adlı A. Kumar tarafından yılında 2011 yayınlanan bölümde, tekstil kumaş malzemeleri ve ürünleri için otomatik kalite güvencesi, gerçek dünya uygulamalarındaki en zorlu bilgisayarla görme sorunlarından biri olduğu anlatılmaktadır. Örme işleminde, kumaş ipliğinin kalitesindeki, üretim ve çalışma koşullarındaki rastgele değişiklikler, genellikle boyut, şekil, görünüm ve renk bakımından değişen dinamik olarak doldurulmuş kusurlara neden olur. Tekstil ürün kalitesi için görsel incelemeden kaynaklanan ekonomik faydalar çok büyüktür ve ürün kalite güvencesi için otomatikleştirilmiş görüntü işleme tabanlı çözümlere yapılan yatırımı gerekli hale getirmektedir. Son günlerde piyasada bulunan bazı kumaş kontrol makineleri pazara girmiştir. Bununla birlikte, maliyetleri önemli ölçüde daha yüksektir ve tespit edilebilen kusur aralıkları oldukça sınırlıdır. Düşük maliyetli yüksek hızlı bilgisayarların, yüksek çözünürlüklü dijital kameraların ve düşük maliyetli depolamanın artan kullanılabilirliği, sağlam otomatik tekstil denetim çözümlerinin yakın gelecekte popüler hale gelmesi için büyük umutlar yarattığı anlatılmaktadır [2].

“Automated Fabric Defect Inspection: A Survey of Classifiers” adlı Md. Tarek Habib, Rahat Hossain Faisal, M. Rokonzaman, Farruk Ahmed tarafından 2014 yılında yayımlanan makalede, tekstil endüstrisinde üretimin her aşamasında kalite kontrol, son derece rekabetçi küresel pazarda varlığını sürdürmek için kilit bir faktör haline geldiği belirtilmektedir. Manuel kumaş hatası incelemesinin sorunları, hassasiyet eksikliği ve yüksek zaman tüketimidir; burada erken ve doğru kumaş hatası tespiti, kalite kontrolün önemli bir aşamasıdır. Bilgisayarlı görüntü işlemeye dayalı, yani otomatik kumaş hata denetim sistemleri, farklı ülkelerdeki birçok araştırmacı tarafından bu sorunları çözmek için çok faydalı olduğu düşünülmektedir. Başarılı bir otomatik kumaş hatası inceleme sistemi elde etmek için çözülmesi gereken iki büyük zorluk, kusur tespiti ve sınıflandırılmasıdır. Bu çalışmada, otomatik kumaş kusur sınıflandırması için kullanılan farklı teknikleri tartışılmış, ardından otomatik kumaş kusur kontrol sistemlerinde kullanılan sınıflandırıcıların bir araştırması gösterilmiş ve son olarak bu sınıflandırıcıları performans ölçütlerini kullanarak karşılaştırılmıştır. Bu çalışmanın, bu alandaki birçok potansiyel seçeneği anlamak ve değerlendirmek için otomatik

kumaş hata denetimi alanındaki araştırmacılar için çok yararlı olması beklenmektedir [103].

“ImageNet Classification with Deep Convolutional Neural Networks” başlıklı çalışma, 1,2 milyon yüksek çözünürlüklü görüntüyü 1000 farklı sınıfa sınıflandırmak için büyük ve derin bir evrişimli sinir ağı (CNN) eğitmiştir. Araştırma, ImageNet LSVRC-2010 yarışmasında test verileri üzerinde %37,5 top-1 ve %17,0 top-5 hata oranlarına ulaşarak, önceki durumu önemli ölçüde aşmıştır. 60 milyon parametre ve 650.000 nörona sahip olan ağ, beş evrişim katmanı, maksimum havuzlama katmanları, üç tam bağlantılı katman ve son olarak 1000 yönlü bir softmax içerir. Eğitim sürecini hızlandırmak için doymuş olmayan nöronlar ve GPU tabanlı hızlı evrişim işlemleri kullanılmıştır. Aşırı öğrenmeyi azaltmak için tam bağlantılı katmanlarda “dropout” düzenleme yöntemi uygulanmıştır. Çalışma, ILSVRC-2012 yarışmasında modelin bir varyantıyla %15,3 top-5 hata oranıyla birinci olmuş, ikinci en iyi girişin %26,2 hata oranını geride bırakmıştır. Bu çalışma, derin evrişimli sinir ağlarının bilgisayarlı görü topluluğu üzerindeki etkisini kanıtlamış ve Google, Facebook, Microsoft gibi birçok şirketin derin öğrenme teknolojilerini benimsemesini sağlamıştır. Daha iyi donanım, daha fazla gizli katman ve teknik yenilikler ile 2015 yılına kadar derin sinir ağlarının hata oranı üç kat daha azaltılmış ve insan performansına oldukça yaklaşmıştır. Bu başarıda FeiFei ve ekibinin geniş bir etiketli veri seti oluşturma çabalarının etkili olduğu vurgulanmaktadır [62].

“Fabric Defect Detection Using Computer Vision Techniques: A Comprehensive Review” adlı Aqsa Rasheed, Bushra Zafar, Amina Rasheed, Nouman Ali, Muhammad Sajid, Saadat Hanif Dar, Usman Habib, Tehmina Shehryar ve Muhammad Tariq Mahmood tarafından 2020 yılında yayımlanan makalede, çeşitli uygulamalı alanlarda ve otomatikleştirilmiş üretim sürecinde bilgisayarla görü ve dijital görüntü işlemenin farklı uygulamaları bulunmaktadır. Tekstil endüstrisinde, herhangi bir tekstil ürününün kalitesi ve fiyatı otomatik kusur tespitinin verimliliğine ve etkinliğine bağlı olduğundan, kumaş kusur tespiti zorlu bir görev olarak kabul edilir. Önceden, tekstil endüstrisinde kumaş üretim sürecindeki kusurları tespit etmek için manuel insan çabası uygulanmaktadır. Konsantrasyon eksikliği, insan yorgunluğu ve zaman tüketimi, manuel kumaş hatası tespit işlemi ile ilişkili ana dezavantajlardır. Bilgisayarla görme ve dijital görüntü işlemeye dayalı uygulamalar, yukarıda bahsedilen sınırlamaları ve dezavantajları ele alabilir. Son yirmi yıldan bu yana, bu

sınırlamaları gidermek için çeşitli araştırma makalelerinde çeşitli bilgisayar görüşü temelli uygulamalar önerilmiştir. Bu derleme yazısında, kumaş hatalarını tespit etmek için tekstil endüstrisindeki uygulamalarla birlikte çeşitli bilgisayarla görmeye dayalı yaklaşımlar hakkında ayrıntılı bir çalışma sunulması hedeflenmektedir. Önerilen çalışma, histogram tabanlı yaklaşımlar, renk tabanlı yaklaşımlar, görüntü bölümlenme tabanlı yaklaşımlar, frekans etki alanı işlemleri, doku tabanlı hata algılama, seyrek özellik tabanlı işlem, görüntü morfolojisi işlemleri ve son derin öğrenme eğilimlerine ilişkin ayrıntılı bir genel bakış sunmaktadır. Otomatik kumaş hatası tespiti için performans değerlendirme kriterleri de sunulmakta ve tartışılmaktadır. Mevcut yayınlanmış araştırmalarla ilişkili dezavantajlar ve sınırlamalar ayrıntılı olarak tartışılır ve gelecekteki olası araştırma yönlerinden de bahsedilmektedir. Araştırma çalışması, farklı kumaş kusurlarını tespit etmek için bilgisayarla görme ve dijital görüntü işleme uygulamaları hakkında kapsamlı ayrıntılar sağlamaktadır [104].

“Real-time Texture Error Detection on Textured Surfaces with Compressed Sensing” adlı Tobias Böttger tarafından 2014 yılında yayımlanan çalışmada sıkıştırılmış Algılama (Compressed Sensing) çerçevesinde gri tonlamaya çevirip dokulardaki kusurları tespit etmek ve yerini belirlemek için gerçek zamanlı bir yaklaşım sunulmaktadır. Doku sınıflandırmasındaki son sonuçlardan esinlenerek, doku tanımlaması için sıkıştırılmış yerel gri tonlamalı yamalar kullanılmaktadır. İlk adımda, bir hatasız doku örneğinden çıkarılan öznetelikler ile bir Gauss Karışım modeli eğitilir. İkinci adımda, doku örneklerinin yenilik tespiti, her pikselin eğitim sürecinde elde edilen olabilirlik ile karşılaştırılmasıyla gerçekleştirilir. Denetim aşaması, gerçek zamanlı kusur tespiti ve lokalizasyonu sağlamak için çok ölçekli bir çerçeveye yerleştirilmiştir. Doku hatası tespiti için sıkıştırılmış gri ölçekli yamaların performansı, iki bağımsız veri kümesi üzerinde değerlendirilmektedir. Önerilen yöntem, doğruluk ve hız açısından sıkıştırılmamış gri ölçekli yamaların performansını geride bırakabilmektedir [105].

“Fabric Defect Detection Using Activation Layer Embedded Convolutional Neural Network” adlı Wenbin Ouyang, Jue Hou ve Bugao Xu tarafından 2019 yılında yayımlanan çalışmada, görüntü ön işleme, kumaş motifi belirleme, hata haritası oluşturma ve evrişimli sinir ağları (CNN'ler) tekniklerini birleştirerek bir tezgâh üstü kumaş kusur inceleme sistemi için bir derin öğrenme algoritması geliştirilmiştir. Hatalı kumaş üretiminin ana nedeni dokuma tezgâhı arızalarından kaynaklanmaktadır. Bir

kumaş kontrol sistemi, kalite güvencesi için kumaş kusurlarını tespit etmek için kullanılan özel bir bilgisayarlı görme sistemidir. Bir CNN'ye yeni bir çift yönlü potansiyel aktivasyon katmanı tanıtılmış ve bu da karmaşık özelliklere ve dengesiz veri kümesine sahip kumaşlar üzerinde yüksek kusur segmentasyonu doğruluğuna yol açmıştır. Mevcut görüntülerdeki kusurları saptamanın ortalama kesinliği ve geri çağırılması, piksel düzeyinde sırasıyla %90'ın ve %80'in üzerine ulaşarak ve halka açık bir veri kümesinden kusurların sayısını sayma doğruluğu %98'i aşmaktadır [106].

“Fine-tuning Convolutional Neural Networks for Biomedical Image Analysis: Actively and Incrementally” başlıklı C. Tajbakhsh, J.Y.S., Gurudu, B tarafından hazırlanan çalışma, biyomedikal görüntülemeye anotasyon maliyetlerini azaltmak için aktif öğrenme ve transfer öğrenimini birleştiren yenilikçi bir yöntem olan AIFT'yi (Active, Incremental Fine-Tuning) sunmaktadır. Çalışmada, önceden eğitilmiş bir CNN ile başlanarak anotasyon için "değerli" örneklerin seçilmesi ve ardından yeni anotasyonların iteratif olarak dahil edilerek CNN performansının kademeli olarak iyileştirilmesi amaçlanmıştır. Üç farklı biyomedikal görüntüleme uygulamasında yapılan değerlendirmeler, anotasyon maliyetinin en az yarı yarıya azaltılabileceğini göstermektedir. Çalışmada ayrıca, farklı aktif seçim yöntemlerinin yedi temel desenle sınıflandırılabilmesi ve bu sınıfların gerçek klinik uygulamalarda performans karşılaştırmalarının yapılabilmesi vurgulanmıştır. Bu makale, anotasyon maliyetini düşürmek için geliştirilen AIFT yönteminin biyomedikal görüntüleme alanındaki etkinliğini kanıtlamakta ve özellikle aktif öğrenme ile transfer öğreniminin entegrasyonunun klinik uygulamalara katkı sağlayabileceğini göstermektedir [87].

“Learning Transferable Architectures for Scalable Image Recognition” başlıklı B. Zoph, V.V., Shlens, J tarafından yapılan çalışma, sinir ağı mimarilerinin veri setine özel olarak öğrenilmesini sağlayan bir yöntem sunmaktadır. Çalışmada, büyük veri setleri üzerindeki mimari aramanın maliyetli olması sebebiyle, küçük bir veri setinde bir mimari yapı taşı arama ve bu yapıyı büyük veri setlerine aktarma yöntemi önerilmiştir. “NASNet search space” adı verilen yeni bir arama alanı tasarımı ile bu aktarılabilirlik sağlanmıştır. CIFAR-10 veri setinde en iyi evrişim katmanı bulunmuş ve bu katman, ImageNet veri setine aktarılmıştır. Ayrıca, NASNet modellerinde genelleştirmeyi önemli ölçüde iyileştiren ScheduledDropPath adlı bir düzenleme tekniği tanıtılmıştır. CIFAR-10 üzerinde %2,4 hata oranıyla, ImageNet üzerinde ise %82,7 top-1 doğruluk oranıyla insan tasarımı modellerin üzerinde performans



sergilenmiştir. NASNet, %28 daha düşük hesaplama maliyetiyle önceki en iyi modellerden daha yüksek doğruluk sağlamıştır. Ayrıca, öğrenilen görüntü özelliklerinin diğer bilgisayarlı görü problemlerine aktarılabilir olduğu gösterilmiştir. COCO veri setinde Faster-RCNN çerçevesiyle %43,1 mAP ile nesne algılamada mevcut durumu aşmaktadır. Bu çalışma, veri setlerinden ölçeklenebilir evrimsel hücrelerin öğrenilebileceğini ve bu hücrelerin çoklu görüntü sınıflandırma görevlerine aktarılabilirliğini göstermektedir. NASNet mimarileri, mobil uygulamalardan yüksek doğruluk gerektiren görevler için ölçeklenebilir bir yapıya sahiptir ve hem CIFAR-10 hem de ImageNet veri setlerinde, hesaplama maliyeti daha düşük olmasına rağmen, insan tasarımı modellerin performansını aşmaktadır. Öğrenilen mimariler, görüntü sınıflandırma ve nesne algılama gibi çeşitli bilgisayarlı görü problemlerinde önemli avantajlar sağlamaktadır [88].

### 1.3 Hipotez

Örme işlemi sırasında kumaş ipliğinin kalitesi ile üretim ve çalışma koşullarındaki rastgele değişiklikler, genellikle boyut, şekil, görünüm ve renk bakımından değişen dinamik kusurlara neden olmaktadır. Mevcut görsel kalite kontrol yöntemleri, kumaş kusurlarının yalnızca bir kısmını tespit edebilmekte ve bu kusurlar, üretilen kumaşların değerini önemli ölçüde azaltmaktadır. Ayrıca, mevcut otomatik tespit sistemlerinin yüksek maliyetleri ve sınırlı kusur aralıkları, tekstil endüstrisinde etkin ve uygun maliyetli çözümler geliştirilmesini zorunlu kılmaktadır. Günümüzde düşük maliyetli, yüksek işlem gücüne sahip işlemciler, yüksek çözünürlüklü hızlı kameralar ve bulut tabanlı depolama çözümleri gibi teknolojilerin artan erişilebilirliği, otomatik tekstil kalite kontrol sistemlerinin yaygınlaşmasına olanak sağlamaktadır.

Bu tez çalışmasında, kumaş kusurlarını tespit etmek için otomatik görüntü işleme tabanlı bir sistemin geliştirilmesi ve etkinliğinin araştırılması hedeflenmektedir. Önerilen sistem, ön işleme, filtreleme, frekans alanı işlemleri, derin öğrenme yöntemleri gibi çeşitli tekniklerle desteklenecektir. Sistem, hızlı kameralarla elde edilen verilerin grafik işlem birimleri (GPU) tabanlı hesaplama gücüyle işlenmesi yoluyla, akan kumaşlarda gerçek zamanlı kusur tespiti ve sınıflandırma gerçekleştirecek; aynı zamanda kumaşın hız ve yön bilgilerini analiz ederek hata geometrisini kaydetme işlevini yerine getirecektir.

Bu dođrultuda, otomatik g r nt  i leme tabanlı sistemlerin, manuel denetim y ntemlerine kıyasla daha y ksek dođruluk, verimlilik ve izlenebilirlik sađlayarak tekstil end strisinde kalite kontrol s re lerinde yenilik yaratacađı  ng r lmektedir.



## **2. MATERYAL VE YÖNTEMLER**

### **2.1 Kamera Düzenegi**

#### **2.1.1 Alan tarama kameraları**

Alan tarama kameraları, statik veya düşük hızda hareket eden kumaş yüzeylerinin incelenmesinde sıklıkla tercih edilen optik sistemlerdir. Bu kameralar, kumaş yüzeyini tek bir karede yüksek çözünürlükle görüntüleyerek ayrıntılı analiz yapılmasına olanak sağlamaktadır. Ancak, pozlama süresi hareketli nesnelerde bulanıklığa yol açabileceği için, genellikle statik yüzeylerin analizi için daha uygun kabul edilmektedir. Çeşitli alt türleri bulunan alan tarama kameraları, farklı analiz gereksinimlerine yanıt verebilecek şekilde yapılandırılabilir.

##### **2.1.1.1 UV (Ultraviyole) kameralar**

UV kameralar, kumaş yüzeylerini ultraviyole ışık altında görüntüleyerek çıplak gözle algılanamayan detayların tespit edilmesine olanak tanımaktadır. Kimyasal işlemlerden kaynaklanan yüzey kusurları, ince iplik problemleri veya düşük kontrastlı bölgelerin analizi gibi durumlar için ideal bir çözüm sunmaktadır. Alan tarama teknolojisi temelinde çalışan bu kameralar, özellikle ayrıntılı yüzey analizlerinde yaygın olarak tercih edilmektedir.

##### **2.1.1.2 Termal ve IR (Kızılötesi) kameralar**

Termal kameralar, kumaş yüzeyindeki sıcaklık farklılıklarını algılayarak yüzey ve alt katman kusurlarının detaylı bir şekilde analiz edilmesini sağlamaktadır. Kızılötesi ışık spektrumunda çalışan bu kameralar, özellikle kumaşın üretim aşamasında maruz kaldığı ısı süreçlerinden kaynaklanan kusurların tespitinde yüksek etkinlik göstermektedir. Isı geçişleri, malzeme yoğunluğu farklılıkları ve mikro düzeydeki hataların analizi için kullanılabilir. Ayrıca, yüzey altı katmanlarında meydana gelen iplik kopmaları, yabancı maddeler veya dokusal bozukluklar gibi problemlerin

tespitinde etkili bir araçtır. Görünür olmayan kusurları ve materyal yoğunluğu farklılıklarını algılayarak daha kapsamlı bir değerlendirme imkânı sunmaktadır.

#### **2.1.1.3 Renk kameraları**

Renk kameraları, kumaş üzerindeki renk varyasyonlarını ve baskı hatalarını tespit etmek amacıyla kullanılan önemli bir görüntüleme teknolojisidir. Bu kameralar, özellikle desenli veya renkli kumaşlarda, renk solmaları, homojen olmayan renk dağılımları ve istenmeyen renk değişimlerini algılamada yüksek doğruluk sağlamaktadır. Renk analizine dayalı yöntemlerle, kumaşın estetik ve kalite standartlarını değerlendirmek için etkili bir çözüm sunmaktadır.

#### **2.1.2 Çizgi tarama kameraları**

Çizgi tarama kameraları, hareketli kumaşların analizi için optimize edilmiş, yüksek hızlı görüntüleme sistemleridir. Bu kameralar, kumaş yüzeyinden akış yönünde çizgi formunda veri toplayarak detaylı ve yüksek çözünürlüklü görüntüler elde edilmesini sağlamaktadır. Enkoder sistemi ile senkronize bir şekilde çalışarak kumaşın hareket hızına uyum sağlar ve doğrusal taramalar sırasında bulanıklığı en aza indirir. Bu özellikleri sayesinde, hızlı üretim hatlarında ve bulanıklığın kabul edilemez olduğu uygulamalarda yaygın olarak kullanılmaktadır. Günümüzde, 140 kHz gibi yüksek frekanslarla çalışan çizgi tarama kameraları, tekstil sektöründe ileri seviye analizler için etkin bir araç olarak öne çıkmaktadır.

#### **2.1.3 Hiperspektral kameralar**

Hiperspektral kameralar, kumaş yüzeyindeki spektral yansımaları detaylı bir şekilde analiz ederek geniş bir spektral bant aralığında yüksek çözünürlüklü veri sağlamaktadır. Bu kameralar, renk varyasyonlarını, malzeme anormalliklerini ve homojen olmayan bölgeleri tespit etme konusunda son derece etkili bir performans sergilemektedir. Özellikle doku analizi ve mikro düzeydeki kusurların algılanması gereken durumlarda tercih edilen hiperspektral kameralar, kumaş yüzeyinin optik özelliklerinin ayrıntılı incelenmesi için ileri düzey bir teknolojik araç sunmaktadır.

#### **2.1.4 Multispektral kameralar**

Multispektral kameralar, hiperspektral kameralara benzer şekilde spektral bantlarda çalışmakla birlikte, daha sınırlı sayıda spektral bilgi sağlamaktadır. Ekonomik bir alternatif olarak öne çıkan bu kameralar, kumaş üzerindeki renk ve doku kusurlarının tespiti için etkili bir çözüm sunmaktadır. Renk hatalarının, baskı kusurlarının ve yüzey anormalliklerinin belirlenmesi gibi uygulamalarda yaygın olarak kullanılmaktadır.

#### **2.1.5 3D görüntüleme kameraları**

3D görüntüleme kameraları, kumaş yüzeyinin derinlik bilgilerini analiz ederek doku ve profil hatalarını tespit etmek için kullanılan ileri düzey görüntüleme sistemleridir. Lazer tabanlı sistemler veya stereo kamera teknolojileri aracılığıyla, kumaş yüzeyindeki çıkıntılar, dalgalanmalar ve çukurlar üç boyutlu olarak detaylı bir şekilde incelenebilmektedir. Bu kameralar, özellikle karmaşık yüzey yapılarına sahip kumaşların detaylı analizinde yüksek doğruluk ve etkinlik sağlamaktadır.

#### **2.1.6 Endüstriyel akıllı kameralar**

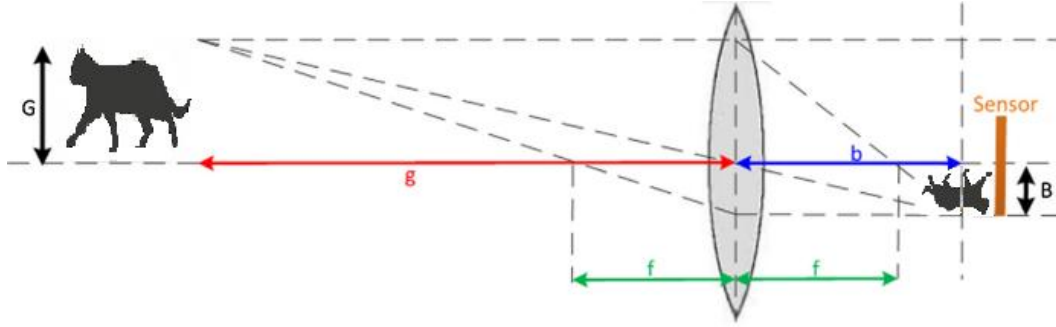
Endüstriyel akıllı kameralar, gömülü görüntü işleme sistemlerine sahip, gerçek zamanlı analiz yapabilen gelişmiş sistemlerdir. Bu kameralar, yerleşik derin öğrenme algoritmaları sayesinde kumaş kusurlarını otomatik olarak tespit ederek sınıflandırabilmektedir. Alan tarama veya çizgi tarama özellikleriyle entegre edilerek hızlı üretim hatlarında verimliliği arttırmaktadır. Kumaş kusuru tespiti, sınıflandırma ve hata türlerine yönelik çözümleme için sıklıkla tercih edilmektedir.

Sonuç olarak, alan tarama kameralarının statik kumaş analizi için daha uygun olduğu, hareketli kumaş analizi için ise çizgi tarama kameralarının tercih edilmesi gerektiği görülmektedir. Bununla birlikte, günümüz teknolojisinde düşük pozlama sürelerinin yüksek aydınlatma koşulları ile birleştirilmesi durumunda, alan tarama kameralarının hareketli kumaş analizinde de etkin bir şekilde kullanılabileceği öngörülmektedir [7].

#### **2.1.7 Lensler**

Uygun kamera seçiminin ardından, analizin gereksinimlerini karşılayacak uygun bir lens seçimi yapılması gerekmektedir. Bir kameranın görüş alanı ve görüntüleme

kapasitesi, kullanılan lensin özelliklerine göre şekillenmektedir. Bundan dolayı, çalışma mesafesi, görüş alanı ve sensör boyutu gibi parametreler dikkate alınarak, en uygun lens seçimi gerçekleştirilmelidir. Elde edilecek görüntünün boyutları, şekli ve netliği, seçilen lensin optik özellikleriyle doğrudan ilişkilidir ve bu seçim, analiz sonuçlarının doğruluğunu doğrudan etkileyebilmektedir.



**Şekil 2.1 : Lens çalışma biçimi.**

Bu nedenle odak uzaklığı  $f$  değeri hesaplanmalıdır.  $f$  değeri aşağıdaki formülle hesaplanabilir:

$$f = \frac{B}{G} \times g \quad (2.1)$$

Burada  $B$  ve  $G$ , sırasıyla görüntü boyutunu ve nesne boyutunu temsil etmektedir. Şekil 2.1, görüntü ve nesne mesafesi ile lens odak uzunluğu ( $f$ ) arasındaki geometrik ilişkileri göstermektedir. Her bir lensin sabit odak uzunluğu (FFL) değeri bulunmaktadır ve  $f$  ile hesaplanan değeri için  $f - 2 \leq \text{FFL} \leq f + 2$  eşitliğini sağlayan lenslerin FFL değerleri arasından herhangi biri tercih edilmektedir.

Bu formülasyon, nesne ve kamera arasındaki belirsiz mesafeyi içeren bir odak uzaklığı ölçüsünü resmi olarak tanımlamaktadır. Ayrıca, optimal görüntü kalitesini sağlamak için büyütme faktörü de dikkate alınmaktadır. Endüstriyel görüntü işleme uygulamalarında, büyütme faktörü genellikle  $>1:10$  (sensör boyutu: nesne boyutu) olmaktadır. Bu oran, endüstriyel uygulamalarda istenen detay seviyesini elde etmek için kritik bir parametre olarak öne çıkmaktadır.

## 2.2 Görüntü İşleme Donanımları

Yapay görme uygulamaları, çeşitli fiziksel platformlarda uygulanabilir ve geniş bir cihaz yelpazesi içermektedir. Bu platformlar arasında, PC tabanlı sistemler, 3D ve çoklu kamera 2D uygulamaları için tasarlanmış özel görüntü denetleyicileri, bağımsız

alıřan grntleme sistemleri ve temel grnt sensrleri bulunmaktadır. Uygun yapay grme platformunun seimi, genellikle uygulamanın gereksinimleri, geliřtirme ortamı, teknik yetenekler, sistem mimarisi ve maliyet gibi faktrlere baėlı olarak belirlenmektedir.

### **2.2.1 PC tabanlı grnt iřleme sistemleri**

PC tabanlı sistemler, doėrudan baėlantılı kameralar veya grnt toplama kartları ile kolayca arayz oluřturabilen ve yapılandırılabilir yapay grme uygulama yazılımlarıyla desteklenen esnek bir platform sunmaktadır. Bu sistemler, Python, Visual C/C++, Visual Basic ve Java gibi yaygın olarak kullanılan programlama dillerinin yanı sıra, grafik tabanlı programlama ortamlarını kullanarak zel kod geliřtirme imkânı saėlamaktadır. Ancak, bu tr sistemlerin geliřtirme sreci genellikle karmařık olmakta ve uzun zaman gerektirmektedir. Bu nedenle, PC tabanlı yapay grme platformları genellikle byk lekli kurulumlarda tercih edilmekte ve ileri dzey grnt iřleme uzmanları ile deneyimli programcılara hitap etmektedir.

### **2.2.2 Gml grnt iřleme sistemleri**

Grnt iřleme kontrolrleri, PC tabanlı sistemlerin tm gcn ve esnekliėini sunarken, zorlu fabrika ortamlarının gereksinimlerine daha iyi uyum saėlayabilecek bir yapısal dayanıklılıėa sahiptir. Bu kontrolrler, zellikle 3D ve oklu kamera 2D uygulamaları iin daha kolay bir yapılandırma imkânı sunmakta ve tek seferlik grevler iin sınırlı zaman ile bte kořullarında etkili bir zm oluřturmaktadır. Ayrıca, daha karmařık uygulamaların uygun maliyetli bir řekilde geliřtirilmesine olanak tanıyarak hem esneklik hem de performans aısından avantaj saėlamaktadır.

### **2.2.3 Baėımsız grnt iřleme sistemleri**

Baėımsız grnt iřleme sistemleri, uygun maliyetli olmalarının yanı sıra hızlı ve kolay bir řekilde geliřtirme yapılmasına olanak tanıyan kompakt zmler sunmaktadır. Bu sistemler genellikle kamera sensr, iřlemci ve iletiřim altyapısı ile entegre řekilde tasarlanmıřtır. Bazı modeller, harici aydınlatma desteėi ve otomatik odaklama iin gerekli optik bileřenleri de iermektedir. Kompakt yapıları sayesinde,

fabrika ortamlarında farklı noktalarda kolayca kurulabilmekte ve uygun maliyet avantajı sağlamaktadır.

Bağımsız görüntü işleme sistemlerinin en önemli avantajlarından biri, üretim sürecindeki kusurların erken tespit edilmesini ve ekipman sorunlarının hızla belirlenmesini sağlamalarıdır. Bu sistemler, yerleşik Ethernet iletişim altyapısı sayesinde birden fazla sistemin bir ağ üzerinden birbirine bağlanmasına ve ana bir bilgisayar tarafından yönetilmesine olanak tanımaktadır. Böylece, kullanıcılar yalnızca üretim süreçlerinde görme sistemlerini etkin bir şekilde kullanmakla kalmaz, aynı zamanda sistemler arasında veri alışverişi yaparak tüm fabrika genelinde ölçeklenebilir bir görüş sistemi ağı oluşturabilmektedir. Bu ağ, fabrika ve kurumsal ağlara kolayca entegre edilerek TCP/IP protokolüne sahip iş istasyonları üzerinden görüş sonuçları, görüntüler, istatistiksel veriler ve diğer bilgilerin uzaktan izlenmesini mümkün kılmaktadır.

Bağımsız sistemler hem basit yönlendirmeli kurulumlar hem de gelişmiş programlama ve komut dosyası oluşturma imkânı sunan yapılandırılabilir ortamlarla donatılmıştır. Bazı sistemler, daha fazla kontrol ve görüntü uygulama verilerinin işlenmesi için esnek geliştirme ortamları sağlayarak hem ek güç gereksinimlerini karşılar hem de karmaşık uygulamaların kolaylıkla uygulanmasına olanak tanımaktadır.

## **2.3 Haberleşme Arayüzleri**

Görüntüleme sistemlerinde kamera sensörlerinin sürekli artan veri işleme kapasiteleri, haberleşme arayüzlerinin seçimini kritik bir unsur hâline getirmiştir. Uygun bir arayüz seçimi, sistemin performansını, kurulum kolaylığını ve maliyet etkinliğini doğrudan etkilemektedir. Günümüzde, USB, Gigabit Ethernet (GigE) ve Camera Serial Interface (CSI) gibi arayüzler, farklı uygulama alanlarına yönelik olarak tercih edilmektedir. Aşağıda, bu arayüzlerin teknik özellikleri, avantajları ve dezavantajları ayrıntılı bir şekilde incelenmiştir.

### **2.3.1 CSI haberleşme**

Camera Serial Interface (CSI), gömülü sistemlerde yüksek performans ve düşük güç tüketimi gereksinimlerini karşılamak için geliştirilmiş bir haberleşme arayüzüdür. MIPI (Mobile Industry Processor Interface) standartları çerçevesinde geliştirilen CSI,



mobil cihazlar, drone'lar, robotik sistemler ve diğer entegre çözümlerde yaygın olarak kullanılmaktadır.

CSI-2 standardı, paralel veri aktarımı sağlayan 1 ila 4 veri hattı üzerinden çalışır ve her bir veri hattı başına yaklaşık 2,5 Gbit/s aktarım hızına ulaşabilmektedir. Bu özellikler, gerçek zamanlı görüntü işleme ve düşük güç tüketimi gerektiren uygulamalar için oldukça ideal bir yapı sunmaktadır. Ayrıca, CSI arayüzü minimal donanım gereksinimi ve kompakt tasarımı ile gömülü sistemlerde kolay entegrasyon imkânı sağlamaktadır.

### **2.3.2 USB haberleşme**

USB (Universal Serial Bus) standardı, yüksek hızda veri aktarımı ve kolay kullanım sunan bir haberleşme arayüzüdür. İlk olarak 1996 yılında USB 1.0 standardı ile tanıtılmış ve günümüze kadar USB 3.2 Gen 2x2'ye kadar bir dizi iyileştirme geçirmiştir. USB 3.0 ve sonraki sürümleri, 5 Gbit/s, 10 Gbit/s ve 20 Gbit/s bant genişliği sunmaktadır. Bu hızlar, modern görüntüleme sistemleri için yeterli veri aktarım hızını sağlamaktadır. Ancak, farklı USB standartları arasında yapılan isim değişiklikleri, özellikle kullanıcılar için kafa karıştırıcı olabilmektedir.

USB'nin en önemli avantajı, "tak ve çalıştır" özelliği sayesinde kolay kurulum imkânı sunmasıdır. Çoğu bilgisayar birden fazla USB bağlantı noktasına sahip olup, ek bir yapılandırma gerektirmeden cihazlar çalıştırılabilmektedir. Bununla birlikte, USB bağlantılarının maksimum kablo uzunluğu 10 metre ile sınırlıdır, bu da uzun mesafeli uygulamalarda sınırlayıcı bir unsur oluşturabilmektedir.

### **2.3.3 GigE haberleşme**

Gigabit Ethernet (GigE), özellikle endüstriyel uygulamalarda yaygın olarak kullanılan bir haberleşme arayüzüdür. USB'nin kablo uzunluğu kısıtlamalarının aksine, GigE bağlantıları 100 metreye kadar uzanabilir, bu da fabrika otomasyonu ve uzun menzilli izleme uygulamaları için önemli bir avantaj sunmaktadır. Daha yüksek hız gereksinimleri için 10/25 GigE standartları geliştirilmiştir. Ancak bu standartlar, CAT6 kablolar ve özel ağ ekipmanları gibi ilave donanım gereksinimlerini beraberinde getirmektedir. GigE'nin bir diğer avantajı, birden fazla kameranın aynı ağ üzerinde bir araya getirebilmesidir. Ağ anahtarları aracılığıyla kameralar kolayca

yönetilebilmekte ve geniş alanlara esnek bir şekilde konumlandırılabilir. Bununla birlikte, GigE arayüzünün kullanımı IP adresi yapılandırması ve ağ bağlantı noktası tahsis gibi ek işlemleri gerektirmektedir [8].

Haberleşme arayüzlerinin özelliklerinin karşılaştırılması Çizelge 2.1’de verilmektedir.

**Çizelge 2.1 : Kamera haberleşmesi karşılaştırması.**

| Özellik           | USB            | GigE                    | CSI                 |
|-------------------|----------------|-------------------------|---------------------|
| Bant Genişliği    | 5–20 Gbit/s    | 1–25 Gbit/s             | 2,5 Gbit/s per lane |
| Kablo Uzunluğu    | 10 metre       | 100 metre               | <1 metre            |
| Kurulum Kolaylığı | Yüksek         | Orta                    | Yüksek              |
| Güç Tüketimi      | Orta           | Orta-Yüksek             | Düşük               |
| Uygulama Alanı    | Genel kullanım | Endüstriyel uygulamalar | Gömülü sistemler    |

## 2.4 Aydınlatma Çeşitleri

Aydınlatma, birçok yapay görme ve görüntü elde etme sistemi için kritik bir bileşen olarak değerlendirilmektedir. Özellikle otomatik kumaş kontrol sistemlerinde; ön aydınlatma, arka aydınlatma, fiber optik aydınlatma ve yapısal aydınlatma teknikleri gibi farklı aydınlatma şemaları kullanılmaktadır. Ön aydınlatma tekniği, genellikle kalın kumaşlardaki hataların tespitinde tercih edilmekte olup, aydınlatma kaynağı kamera ile aynı tarafta, kumaşın önüne yerleştirilmektedir. Arka aydınlatma tekniği ise yarı saydam kumaş türlerinde, örneğin süprem kumaşlarda, gölgelenme etkilerini ortadan kaldırarak daha belirgin bir görüntü elde etmek için kullanılmaktadır. Kumaşın arkasına yerleştirilen çizgi ışık kaynağı, ön aydınlatmaya kıyasla daha yüksek kontrastlı görüntüler sunabilmektedir.

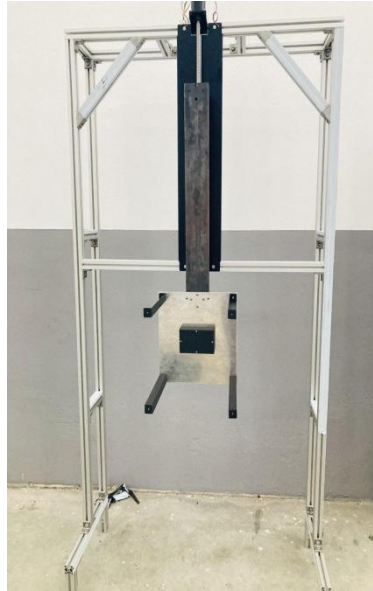
Fiber optik aydınlatma, belirli bir ölçüden büyük kumaş görüntüleri için ekonomik olmamaktadır. Yapısal aydınlatma sistemlerinde ise kızılötesi ışık ve yüksek kare hızlı kameralar kullanılarak kusurlu ve hatasız kumaş görüntülerinin birbirinden ayrıştırılması mümkün hâle gelmektedir. Literatürde, floresan, halojen ve ışık yayan diyot (LED) ışık kaynaklarının yaygın olarak kullanıldığı görülmektedir. Kaliteli bir görüntü elde edebilmek için ışık kaynağının genişliği önemli bir parametre olarak öne çıkmaktadır. Işığın optimal genişliği, aşağıdaki formül yardımıyla belirlenebilir:

$$\text{Işık genişliği (mm)} = \text{ışık alanı} + (2 * \text{kamera çalışma mesafesi}) \quad (2.2)$$

Elektromanyetik spektrumda, insan gözü yalnızca 400-700 nm dalga boyu aralığındaki radyasyonları algılayabilmektedir. LED aydınlatma, bu dalga boyu aralığına yakın bir ışık spektrumu sağlayarak, insan gözüne benzer algılama özellikleri sunmaktadır. Bu özelliklerinden dolayı, son yıllarda geliştirilen kumaş kontrol sistemlerinde, etkin ve başarılı yapay görme uygulamaları için hat LED aydınlatma yaygın olarak tercih edilmektedir. LED aydınlatmanın verimli bir şekilde kullanımı, ışık genişliğinin yukarıda verilen formüle göre doğru şekilde belirlenmesiyle mümkün hâle gelmektedir.

## 2.5 Test Düzeneği

Yuvarlak örme makinası hareketini yapacak bir test düzeneği kurularak, görüntü algılama işlemleri yapılması planlanmıştır. Örme makinesinin örme aşamasındaki kumaş hareketlerini taklit edebilmek amacıyla, Şekil 2.2'deki görselde gösterilen prototip tasarlanmış ve üretilmiştir. Prototip kamera ve aydınlatma sisteminin monte edildiği bir adet doğrusal hareketli eksen bulunmaktadır.



**Şekil 2.2 :** Prototip şasi montaj görüntüsü.

Örme makinelerinde kumaş, açık en veya tüp formunda üretilmektedir. Açık en çalışan makinelerde, tüp formunda örülen kumaş kesilerek, alt kısımda bulunan topa sarılmaktadır. Kumaşın kesilmeden önce maruz kaldığı gerilimi simüle edebilmek

amacıyla, bir döner eksen ve üzerinde bir tambur kullanılmaktadır. Örmek makinesinde üretilen kumaşın çapı yaklaşık 80 cm olup, kullanılan tamburun çapı da bu ölçüye eşdeğerdir. Bu yapı, kumaşın örülme anındaki gerçek çalışma koşullarını modellemeyi sağlamaktadır.

Örmek hareketini gerçeğe uygun şekilde simule edebilmek için doğrusal eksen üzerinde bulunan kamera 3 m/dk ile aşağıya doğru hareket ederken, tambur da eşzamanlı olarak 34 devir/dk hızla döndürülmüş ve böylece örmek aşamasındaki kumaş görüntüsü oluşturulmuştur. Yukarı-aşağı hareket step motor kontrolüyle sağlanmıştır. Tamburun dönmesi için kullanılan motor Şekil 2.3’de, yukarı aşağı hareketi gerçekleştiren step motor ise Şekil 2.4’de gösterilmektedir.



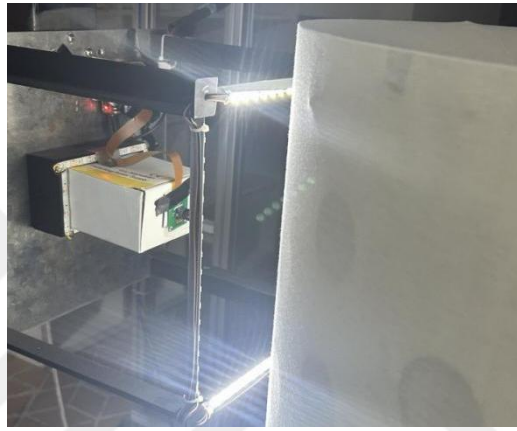
**Şekil 2.3 :** Taban tambur motor görüntüsü.



**Şekil 2.4 :** Prototip birleştirilmiş görüntü.

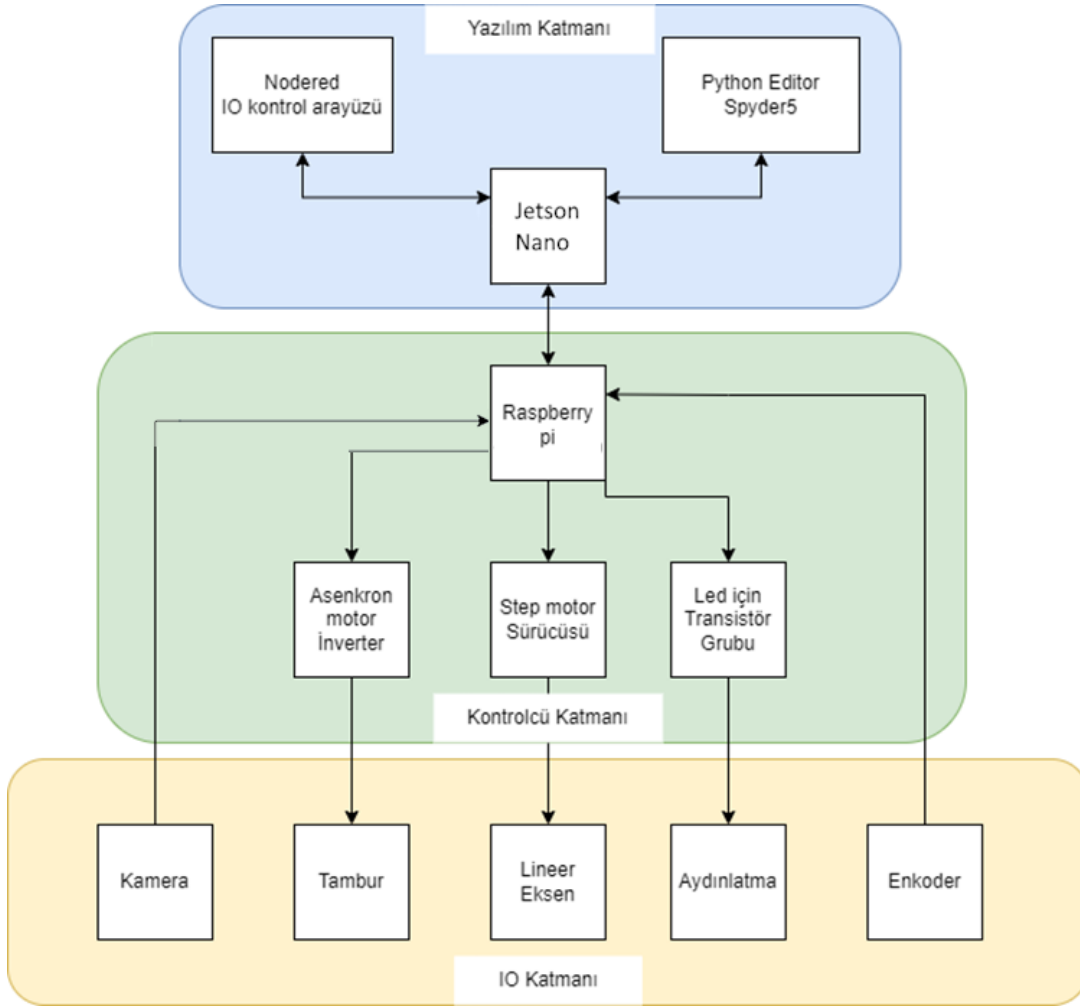
Maksimum dönme hızlarında görüntü alımının gerçekleştirilebilmesi amacıyla monochrome global shutter özellikli olan kameralar kullanmaya karar verilmiştir. Bu kamera ile hareket halinde iken 100  $\mu$ s pozlama süresi (shutter speed) ile görüntüde bozulma olmadan fotoğraflar alınabilmiş ve elde edilen görüntüler birleştirilerek 0.5 mm çözünürlüğünde görüntüler oluşturulmuştur.

Şekil 2.5’de görüldüğü üzere, aydınlatma için LED’lerin kumaşa yaklaştırılabildiği ve açı verilebildiği bir tasarım uygulanmış; bu sayede alt-üst, yan ve UV aydınlatmaları bağımsız olarak kontrol edilebilmesiyle, farklı hata türlerinin daha belirgin hâle getirilerek görüntülenmesi sağlanmıştır.



**Şekil 2.5 : Aydınlatma sistemi.**

Şekil 2.6’de görüldüğü üzere, sistem; IO katmanı, kontrolcü katmanı ve yazılım katmanı olmak üzere üç temel katman ile özetlenmiştir. IO katmanında, saha ekipmanlarının bağlandığı pano içerisindeki kontrol elemanları ile bu katmanın kontrolcü ve yazılım katmanlarına olan bağlantıları gösterilmektedir. Yazılım katmanında, IO’ların kontrolü için web tabanlı Node-RED programlama arayüzü kullanılmıştır. Node-RED’in IoT uyumluluğu sayesinde farklı donanımlar ile etkili şekilde haberleşme sağlanabilmektedir. Python programlama dili için Spyder 5 editörü tercih edilmiştir. Bu editörün tercih edilme nedeni, akademik çalışmalarda yaygın olarak kullanılan editörlere benzemesi ve değişken analizlerinin kolaylıkla yapılabilmesine olanak sağlamasıdır. Python ortamında hem kamera hem de hız enkoderine ait veriler işlenerek görüntü birleştirme algoritması çalıştırılmaktadır. Ayrıca, görüntü işleme ve makine öğrenmesine dayalı hata tespit algoritmalarının geliştirilmesinde de Python aktif olarak kullanılmaktadır.



**Şekil 2.6 :** Denetleyici yapısı.

Prototip tüm sistem görüntüsü Şekil 2.7’de gösterilmektedir.



**Şekil 2.7 :** Sistem genel görünüm.

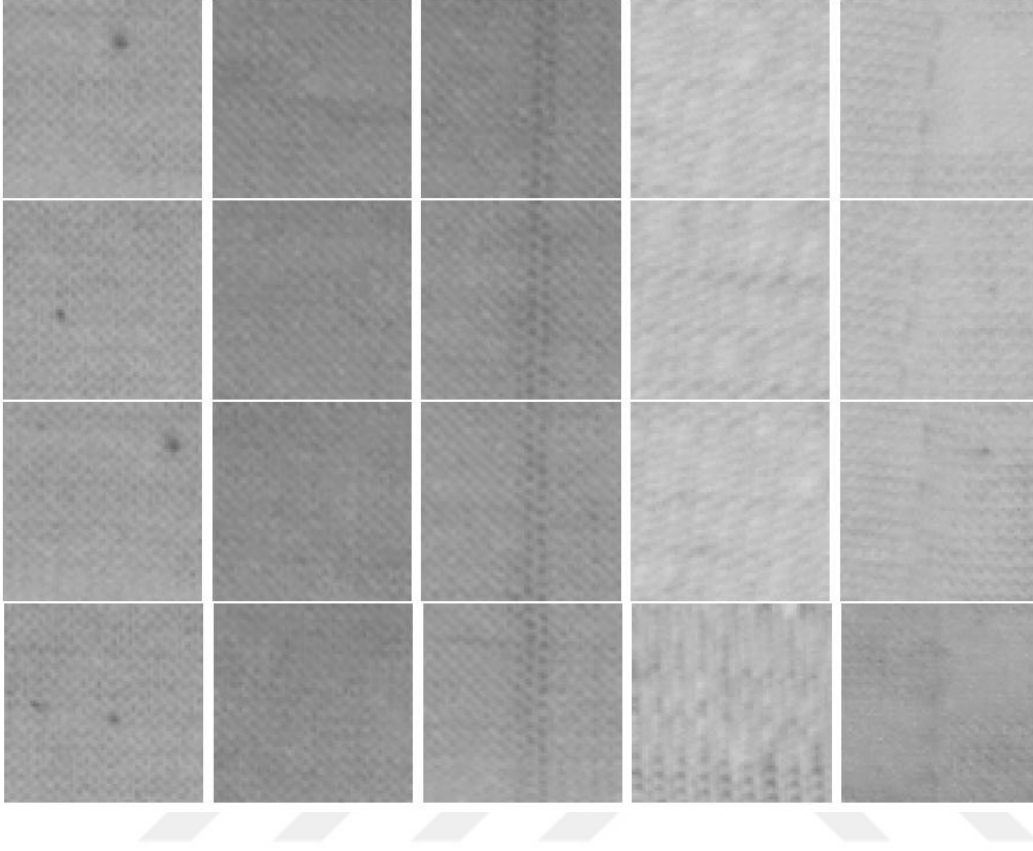
## 2.6 Görüntü Veri Setinin Hazırlanması

Bir görüntü işleme uygulamasında, veri algılama; çevreden görsel bilgi toplayıp sistemin kullanımına sunan temel bir adımdır. Bu süreç, sistemin çeşitli uygulamalarda görsel verileri analiz etmesine ve yorumlamasına olanak tanımaktadır. Veri algılama, sistem için ham veri girişini sağlar; kameralar veya diğer sensörler aracılığıyla görüntü ya da video karelerini yakalayarak, sonraki işleme ve analiz aşamaları için bir temel oluşturmaktadır.

Elde edilen verilerin kalitesi ve doğruluğu, sistemin genel performansını doğrudan etkilemektedir. Yüksek kaliteli, net ve uygun şekilde kalibre edilmiş görüntüler, analiz ve karar verme sürecinde daha doğru ve güvenilir sonuçlar elde edilmesine katkıda bulunmaktadır. Döner sistemlerde, makine hızı nedeniyle bulanıklık ve yetersiz aydınlatma gibi sorunlar ortaya çıkabilmektedir. Test düzeneğinde dönme hızı kontrol edilebilir olsa da gerçek bir örgü makinesinde kullanılan kamera sisteminin yakalama frekansına uyum sağlaması ve mevcut aydınlatma koşullarına göre optimize edilmesi gerekmektedir. Etkili bir kusur tespit sisteminin geliştirilebilmesi için, görüş sistemi; aydınlatma değişimleri, hareket kaynaklı bulanıklık, lens yüzeyindeki toz ve yağ gibi çeşitli çevresel faktörlere karşı dinamik bir şekilde uyum sağlayabilecek şekilde yapılandırılmalıdır. Sürekli ve çeşitli veri toplama işlemleri, sistemi bu tür zorluklara karşı eğitmek ve hassas ayarlar yapmak için kritik öneme sahiptir. Gerçek zamanlı analiz gerektiren uygulamalarda, veri toplamanın verimli bir şekilde gerçekleştirilmesi esastır. Bu sayede sistem, bilgiyi zamanında yakalayarak işleyebilmekte; hızlı yanıtlar ve kararlar üretebilmektedir. Özetle, veri toplama, bir görüntü işleme sisteminin işleyişinde kritik bir başlangıç adımıdır. Elde edilen verilerin kalitesi, miktarı ve çeşitliliği; sistemin görevlerini doğru şekilde yerine getirme, farklı çevresel koşullara uyum sağlama ve çeşitli uygulamalar için anlamlı çıktılar üretme yetkinliğini doğrudan etkilemektedir.

Görüntüler farklı boyutlarda kırılmış ve önceden eğitilmiş bir model üzerinden geçirilerek, hatalı ve hatasız olmak üzere iki ayrı klasör altında kayıt altına alınmıştır. Her bir kırılan parça, resim üzerindeki konum bilgisine göre adlandırılarak kaydedilmiştir. Hazırlanan bu veri seti, farklı yöntemlerde kullanılmak üzere depolanmakta ve saklanmaktadır. Klasör altında yer alan görüntüler, gözle yapılan

sınıflandırmalara göre yeniden düzenlenmiştir. Sınıflandırılmış hata türlerine ilişkin örnekler Şekil 2.8’de sunulmaktadır.



**Şekil 2.8 :** Hatalı kumaş veri seti (delik, hatasız, iğne kırılması, likra kaçığı, may izi).



### 3. İSTATİKSEL VE SPEKTRAL HATA TANIMA ALGORİTMALARI

#### 3.1 Sobel, Prewitt, Roberts, Canny Kenar Algılama Algoritmaları

Görüntüyü tek renkliye (grayscale) dönüştürdükten sonra, kenar bulma algoritmaları arasında en yaygın kullanılan yöntemlerden biri Sobel kenar bulma filtresidir. Şekil 3.1’de gösterilen çekirdek matrisler (konvolüsyon matrisleri), görüntüdeki dikey, yatay ve köşegen kenarların tespit edilmesi için kullanılmaktadır. Sobel operatörü, görüntü üzerindeki kenarlara karşılık gelen yüksek frekanslı bölgeleri (keskin kenarları) belirlemek amacıyla etkili bir şekilde uygulanmaktadır. Bu operatör, 3×3 boyutunda bir konvolüsyon matrisinden oluşmaktadır ve kenar belirleme işlemlerinde yaygın olarak tercih edilmektedir.

|    |   |    |
|----|---|----|
| -1 | 0 | +1 |
| -2 | 0 | +2 |
| -1 | 0 | +1 |

G<sub>x</sub>

|    |    |    |
|----|----|----|
| +1 | +2 | +1 |
| 0  | 0  | 0  |
| -1 | -2 | -1 |

G<sub>y</sub>

Şekil 3.1 : Sobel konvolüsyon matrisi.

Bu matrisler, yatay ve dikey olarak görünen kenarları ayrı ayrı ortaya çıkaracak şekilde düzenlenmiştir. Matrisler görüntüler üzerine tekrar tekrar bağımsız olarak uygulanabilmektedir. Böylece kullanılan farklı operatörler için her bir pikselin değeri ayrı ayrı elde edilmektedir [9].

Bu durumda 0 derece yatay duran çizgileri, 90 derece ise dikey duran çizgileri gösterecektir. Eğik duran çizgilerde diğer açıları oluşturacaktır. Burada sıfır derece çizgide alt kısım siyahtan, üst kısım beyaza doğru geçişi göstermektedir. 0 derecenin bir benzeri olan yine yatay duran çizgi 180 derecede ise üst kısım siyah bölgeden alt kısım beyaz bölgeye doğru bir geçişi gösterecektir. Açılar saatin tersi yönüne göre

ölçülmektedir. Burada, Gx konvolüsyon matrisi tek başına kullanıldığında, yalnızca yatay düzlemdeki renk değişimlerini algılayacağından, ortaya çıkan çizgiler dikey yönde görülmektedir. Benzer şekilde, Gy konvolüsyon matrisi tek başına kullanıldığında yukarıdan aşağıya (örneğin siyah-beyaz geçişlerinde) renk geçişlerini algılar ve bu durumda ortaya çıkan çizgiler yatay olarak görünür. Görüntü üzerindeki çizgilerin yalnızca yatay ya da dikey değil, doğal yönelimleriyle algılanabilmesi için Gx ve Gy konvolüsyon matrislerinin yukarıda verilen formüle göre birlikte kullanılması gerekmektedir. Aynı şekilde, daha önce belirtilen açığa bağlı kenar belirleme formülü de benzer amaçlarla kullanılabilir. Bu yöntem ile, resim üzerinde belirli açılarda bulunan kenar ya da çizgilerin belirgin hâle getirilmesi mümkün olmaktadır.

İki matrisi aynı anda kullanmak için ve matris üzerindeki noktaları aşağıdaki şekilde temsil etmek için hesaplama yapılırsa Şekil 3.2'deki matrisler elde edilmektedir.

|                |                |                |    |   |    |    |    |    |
|----------------|----------------|----------------|----|---|----|----|----|----|
| P <sub>1</sub> | P <sub>2</sub> | P <sub>3</sub> | -1 | 0 | +1 | +1 | +2 | +1 |
| P <sub>4</sub> | P <sub>5</sub> | P <sub>6</sub> | -2 | 0 | +2 | 0  | 0  | 0  |
| P <sub>7</sub> | P <sub>8</sub> | P <sub>9</sub> | -1 | 0 | +1 | -1 | -2 | -1 |
|                |                |                | Gx |   |    | Gy |    |    |

**Şekil 3.2 :** Sobel konvolüsyon matrisi hesaplanması.

Buradaki her bir çekirdek matrisi ve daha sonra bu çekirdek matris kullanılarak pikselin sobel değeri yaklaşık formülü benzer şekilde kullanılarak hesaplanmaktadır:

$$\begin{aligned}
 |Gx| &= |-P_1 + P_3 - 2P_4 + 2P_6 - P_7 + P_9| \\
 |Gy| &= |P_1 + 2P_2 + P_3 - P_7 - 2P_8 - P_9|
 \end{aligned}
 \tag{3.1}$$

Prewitt algoritması, Sobel algoritmasına benzer şekilde çalışmakta olup, yalnızca çekirdek matrislerinde küçük farklılıklar içermektedir. Bu yöntemde kullanılan çekirdek matrisler aşağıda gösterildiği gibidir ve kenar tespiti sırasında elde edilen sonuçlar, görüntünün her bölgesinde aynı düzeyde olmayabilir. Şekil 3.3'te Prewitt çekirdek matrisleri sunulmaktadır [10].

|    |   |    |
|----|---|----|
| -1 | 0 | +1 |
| -1 | 0 | +1 |
| -1 | 0 | +1 |

**Gx**

|    |    |    |
|----|----|----|
| +1 | +1 | +1 |
| 0  | 0  | 0  |
| -1 | -1 | -1 |

**Gy**

**Şekil 3.3 :** Prewitt çekirdek matrisleri.

Robert Cross algoritması, basit ve hızlı bir kenar algılama yöntemi olarak öne çıkmaktadır. Şekil 3.4'te gösterilen iki boyutlu geçişlerin hesaplanmasını sağlayarak, gri tonlamalı bir görüntü üzerinde işlem gerçekleştirmekte ve keskin kenarları belirgin bir şekilde ortaya çıkarmaktadır. Bu algoritma, tespit ettiği kenarları oldukça ince bir şekilde gösterebilme özelliğine sahiptir. İşlem, 90 derece açıyla birbirine dik konumlandırılmış  $2 \times 2$  boyutundaki iki çekirdek matris yardımıyla uygulanmaktadır. Robert Cross algoritması, kullanım açısından Sobel operatörüne benzemekle birlikte daha küçük bir çekirdek kullanarak daha basit bir hesaplama sunmaktadır.

|    |    |
|----|----|
| +1 | 0  |
| 0  | -1 |

**Gx**

|    |    |
|----|----|
| 0  | +1 |
| -1 | 0  |

**Gy**

|       |       |
|-------|-------|
| $P_1$ | $P_2$ |
| $P_3$ | $P_4$ |

**Şekil 3.4 :** Robert Cross konvolüsyon matrisleri.

Sobel algoritmasında kullanılan konvolüsyon matrisleri, genellikle x ve y eksenlerine göre hizalanmıştır. Ancak, Robert Cross algoritmasında matrislerin yönleri, görüntünün ızgarasına 45 derece ve 135 derece açıyla yerleştirilmiştir. Bu nedenle, kullanılan formüller yine Gx ve Gy şeklinde ifade edilmekle birlikte, bu yönlerin ızgaraya göre eğimli olduğu, yani 45 derece ve 135 derece açıyla durduğu kabul edilmektedir. Bu farklı hizalama, Robert Cross algoritmasının, görüntü üzerinde eğik kenarların algılanmasında daha etkin olmasını sağlamaktadır. Bu operatör, yalnızca 4 piksel üzerinde işlem yaptığı için, çıkış değerinin tam olarak hangi piksele karşılık geldiği net olarak belirlenememektedir. Bu nedenle yalnızca yarım piksellik bir kayma

mümkün olabilmektedir. Uygulama sırasında, referans piksel olarak sol üst köşedeki ilk piksel esas alınmıştır.

Canny operatörü, kenar algılama sürecinde birden fazla aşamadan oluşan sofistike ve etkili bir yöntemdir. Bu yöntemde ilk olarak, elde edilen görüntü Gauss filtresi ile evrişim uygulanarak düzleştirilir ve böylece görüntüdeki gürültü azaltılır. Daha sonra, görüntü üzerinde yüksek birinci uzamsal türevleri vurgulamak amacıyla, Roberts Cross operatörü gibi basit bir iki boyutlu birinci türev operatörü uygulanmaktadır. Bu işlem sonucunda, kenarlar gradyan büyüklük görüntüsünde belirgin çıkıntılar şeklinde ortaya çıkmaktadır.

Algoritma, bu çıkıntıların tepe noktalarını izleyerek kenarları daha ince bir çizgi halinde göstermek için maksimal olmayan bastırma adı verilen bir işlem uygulamaktadır. Bu aşamada, belirlenen eşik değerinin üzerinde olmayan tüm pikseller sıfıra ayarlanmaktadır. İzleme süreci, histerezis olarak bilinen bir yaklaşım ile iki eşik değeri ( $T_1$  ve  $T_2$ , burada  $T_1 > T_2$ ) tarafından kontrol edilmektedir. İzleme, yalnızca  $T_1$ 'den yüksek bir sırtta başlar ve bu noktadan itibaren, sırtın yüksekliği  $T_2$ 'nin altına düşene kadar her iki yönde de devam ettirilir. Bu histerezis yaklaşımı, gürültü kaynaklı kenarların bölünerek birden fazla kenar parçasına ayrılmasını önlemeye yardımcı olmaktadır.

Canny operatörünün etkisi, üç temel parametreye bağlıdır: Gauss çekirdeğinin genişliği ve üst-alt eşik değerleri. Yumuşatma aşamasında kullanılan Gauss çekirdeğinin genişliği artırıldığında, algılayıcının gürültüye duyarlılığı azalır, ancak bu durum ince ayrıntıların bir kısmının kaybolmasına yol açabilmektedir. Ayrıca, Gauss genişliği arttıkça algılanan kenarlardaki lokalizasyon hatalarında da hafif bir artış gözlemlenmektedir.

Genellikle, iyi sonuçlar için üst izleme eşiği oldukça yüksek, alt eşik ise oldukça düşük olarak ayarlanmalıdır. Alt eşiğin yüksek değerlere ayarlanması sonucunda gürültülü kenarların kırılmasına neden olmaktadır. Üst eşiğin çok düşük verilmesi sonucu, çıktıda görünen yanlış kenar parçalarının sayısını arttırmaktadır.

Temel Canny operatörüyle ilgili bir sorun, gradyan büyüklük görüntüsünde üç sırtın bulunduğu yerler olan Y şeklindeki kavşaklardır. Bu tür bağlantılar, bir kenarın tam olarak görülemediği ve kısmen kapatıldığı durumlarda oluşabilmektedir. İzleme algoritması, bu çıkıntılardan ikisini tek bir çizgi parçası olarak değerlendirebilir;

üçüncü kenar ise bu doğru parçasına yaklaşan ancak tam olarak bağlanmayan bir kenar parçası şeklinde algılanabilir [11].

### 3.2 Kontur Algoritması

Literatürde, kontur izleme amacıyla kullanılan birçok farklı algoritma bulunmaktadır. Bunlar arasında en yaygın olanları Kare İzleme algoritması ve Moore-Komşu izleme algoritmasıdır. Bu yöntemler, uygulanabilirlik açısından oldukça kolaydır ve bu nedenle herhangi bir nesnenin dış hatlarını izlemek amacıyla yaygın olarak tercih edilmektedir. Ancak, bu algoritmaların sahip olduğu bazı sınırlılıklar, özellikle belirli bağlantı türlerine karşı duyarsız olmaları nedeniyle, geniş bir model sınıfının çevresini doğru şekilde izleyememelerine neden olmaktadır.

Kare İzleme ve Moore-Komşu algoritmaları, özellikle kumaş yüzeylerinde yer alan delikleri yok sayma eğilimindedir. Eğer delik içeren bir desen görüntüsü analiz ediliyorsa, bu algoritmalar yalnızca dış konturu takip eder ve iç boşlukları görmezden gelmektedir. Bu durum bazı uygulamalarda tolere edilebilir olsa da, karakter tanıma gibi daha hassas uygulamalarda, karakteri tanımlayan iç konturların da algılanması gerekmektedir. Bu algoritma ile her bir delik belirlenmeli ve ardından her deliğin konturu üzerinde ayrı ayrı kontur izleme algoritmaları uygulanarak genel kontur yapısı tam olarak çıkarılmalıdır [12].

#### 3.2.1 Kare izleme algoritması

Kare izleme algoritmasının temelinde oldukça basit bir fikir yatmaktadır. Bu algoritma, ikili bir desenin dış hatlarını çıkarmaya yönelik geliştirilen ilk yöntemlerden biri olarak kabul edilmektedir. Algoritma, dijital bir model üzerinde çalışır; bu model, siyah piksellerden oluşan bir şeklin, beyaz piksellerden oluşan bir arka plan üzerinde yer aldığı bir ızgarayı temsil etmektedir. Süreç, bir siyah pikselin bulunup "başlangıç" pikseli olarak ilan edilmesiyle başlamaktadır. "Başlangıç" pikselinin tespiti, birkaç farklı yöntemle gerçekleştirilebilir. Örneğin, ızgaranın sol alt köşesinden başlanarak her piksel sütunu, alttan yukarı doğru taranmaktadır. En soldaki sütundan başlayarak sağa doğru ilerlemekte ve siyah bir pikselle karşılaşıldığında, bu piksel "başlangıç" pikseli olarak ilan etmektedir. Bu başlangıç noktasından itibaren, algoritma deseni izleyerek dış hatları belirlemektedir.

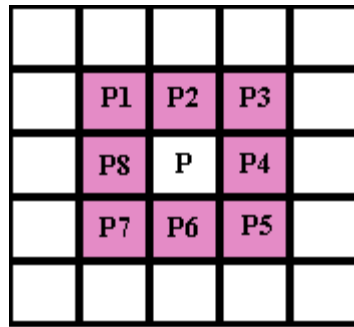
Başlangıç pikseli belirlendikten sonra, kontur izleme işlemi siyah pikseller üzerinden yürütülmektedir. Algoritma, siyah bir piksele ulaşıldığında sola; beyaz bir piksele ulaşıldığında ise her seferinde sağa dönüş gerçekleştirerek işlemi sürdürmektedir. İzleme süreci, başlangıç pikseline yeniden ulaşıldığında tamamlanmaktadır. Bu süreçte üzerinden geçilen siyah pikseller, desenin dış konturunu oluşturmaktadır.

Kare izleme algoritmasında, izleme yönünün doğru şekilde sürdürülmesi algoritmanın başarımı açısından kritik bir unsurdur. “Yön duygusu” olarak tanımlanan bu kavram, her bir piksele hangi yönden girildiğinin dikkate alınması ile yön değişimlerinin doğru şekilde hesaplanmasını sağlamaktadır. Yapılan her dönüş, mevcut konuma ve giriş yönüne göre belirlenmektedir. Bu nedenle, algoritmanın doğru çalışabilmesi için mevcut yön bilgisinin sürekli olarak güncellenmesi ve sonraki adımların bu bilgiye dayalı olarak kesin biçimde hesaplanması gerekmektedir.

### 3.2.2 Moore-Neighbor takip algoritması

Bir pikselin Moore komşuluğu, P noktasındaki, o piksele bir köşe veya kenarı bulunan 8 pikseli temsil etmektedir. Bu pikseller, aşağıdaki Resim 11'de gösterildiği üzere P1, P2, P3, P4, P5, P6, P7 ve P8 olarak adlandırılmaktadır.

Şekil 3.5'de verilen Moore algoritması (8-komşu veya dolaylı komşu olarak da bilinmektedir) literatürde sıkça kullanılan önemli bir kavramdır.



Şekil 3.5 : Moore piksel kümesi.

Bir dijital model, yani siyah piksellerden oluşan bir nesne, beyaz piksellerden oluşan bir arka plan üzerinde ızgara yapısında tanımlandığında, kontur izleme algoritmasının başlatılabilmesi için öncelikle başlangıç pikselinin tespit edilmesi gerekmektedir. Başlangıç pikseli, çeşitli yöntemlerle belirlenebilir. Bu çalışmada kullanılan yöntem, ızgaranın sol alt köşesinden başlanarak, her sütunun alttan yukarı doğru taranması ve en soldaki sütundan sağa doğru ilerlenerek ilk karşılaşılan siyah pikselin başlangıç

pikseli olarak atanmasıdır. Başlangıç pikseli belirlendikten sonra, modelin dış konturunu izlemek amacıyla saat yönünde izleme işlemi gerçekleştirilir. İzleme yönü bir kez belirlendikten sonra sabit kalmalı, algoritmanın genelliği korunmalıdır.

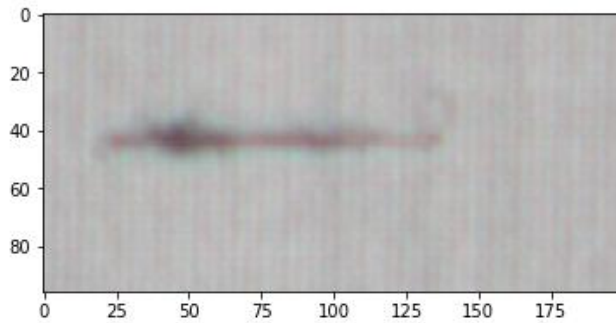
İzleme işlemi sırasında, her siyah piksele (P) ulaşıldığında, algoritma bir önceki konumda bulunan beyaz piksele geri döner ve ardından P pikselinin Moore komşuluğu saat yönünde taranarak kontrol edilir. Komşuluk içerisinde tespit edilen ilk siyah piksel, kontur izleme yolunun bir sonraki adımı olarak seçilir. Bu işlem, başlangıç pikseli ikinci kez ziyaret edilene kadar devam eder. İzleme süreci sonunda üzerinden geçilen tüm siyah pikseller, ilgili nesnenin dış konturunu tanımlamaktadır [13].

### 3.3 Eğrilik Algoritması

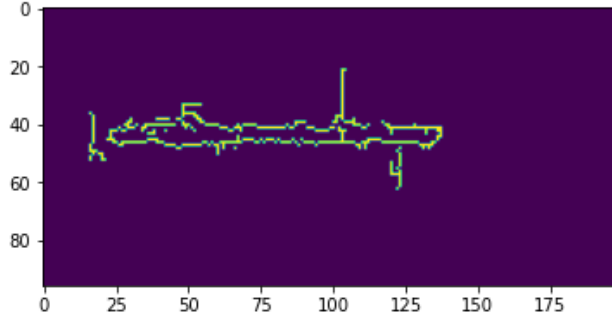
Eğrilik algoritması, piksel altı çözünürlükte doğru görüntü eğriliklerini hesaplayan ve görsel algımıza uygun eğrilik haritasının bir görselleştirmesini sağlamaktadır. Eğrilikler, bağımsız yumuşatıldıktan sonra doğrudan bir görüntünün düz çizgileri üzerinde değerlendirilmektedir [14-16].

Eğriliklerin görsel algıdaki rolü uzun süredir tartışılmakta ve eğrilikleri doğru bir şekilde tahmin etmek için çok sayıda girişimde bulunmaktadır. Şekil 3.6'da gösterilen hatalı kumaş resmi üzerinde, taslağı çizilen algoritmanın kapsamlı tanımı ve uygulanması hedeflenmektedir [17].

Eğriliğin görsel algıdaki rolü ilk olarak nörolojik temellere dayanarak görsel uyarının fazlasıyla gereksiz olduğunu ve “bilginin konturlar boyunca ve daha sonra yönünün değiştiği bir kontur üzerindeki noktalarda yoğunlaştığını savunan Attneave tarafından gündeme getirilmiştir [18]. Hatalı kumaş resmi ve konturlara ayrılmak üzere Canny filtresi uygulanmış hali Şekil 3.7’de görülmektedir.



Şekil 3.6 : Hatalı resim görüntüsü.



**Şekil 3.7 :** Hatalı resim Canny kenar metodu uygulanmış resmi.

Asada ve Brady, Attneave'in fikrini uygulayarak eğrilik ilkel çizim kavramını ortaya koymuşlardır. Bu yaklaşım, şekillerin çok ölçekli bir yorumuna dayanmaktadır: Nesnelerin dış hatları, eğrilikte önemli değişiklik noktaları olan düğümler aracılığıyla eğri çizgilerle temsil edilmektedir. Bu düğümler, Gauss'un birinci ve ikinci derece türevleri kullanılarak belirlenmektedir. Söz konusu temsil, desteklenen ölçek uzayı fikrini takip etmektedir ve eğriliğin doğru bir şekilde hesaplanabilmesi için giderek daha karmaşık işlemler gerektirmektedir.

Bu süreçte karşılaşılan iki temel zorluk bulunmaktadır:

- (i) Eğriliğin hesaplanması gereken konturların doğru bir şekilde çıkarılması,
- (ii) Konturların topolojik ve topografik özellikleri korunarak yumuşatılması.

Ham görüntüler üzerinde doğrudan eğrilik hesaplamaları, sonlu ızgaralarda kullanılan fark yaklaşımlarına dayandığı için hâlâ yeterince güvenilir değildir. Bu yöntemler, ortalama eğrilik ile görüntü hareketini simüle etmek açısından belirli bir verimliliğe sahip olsa da gerçek eğrilik haritasının görselleştirilmesi söz konusu olduğunda yetersiz kalmaktadır. Ayrıca, çapraz çizgiler boyunca sahte salınımlar meydana gelmektedir. Görüntü eğrilikleri, düzleştirme sonrası çift doğrusal enterpolasyonla hesaplanarak her çözünürlükte hassas görselleştirme sağlanabilmektedir. Ancak, orijinal görüntüye veya görüntü yumuşatıldıktan sonra uygulanan sonlu fark şemalarının tatmin edici sonuçlar vermediği gözlemlenmiştir [19,20].

### 3.3.1 Ayrık eğrilik algoritması

Ayrık eğrilikleri hesaplamak için daha önceki çoğu yaklaşımlar sonlu fark şemaları yaklaşımlarına dayanmaktadır. Ancak bu yaklaşımlar, grid değerlerine dayalı oldukları için yatay ve dikey çizgiler boyunca sahte salınımlara yol açmaktadır.



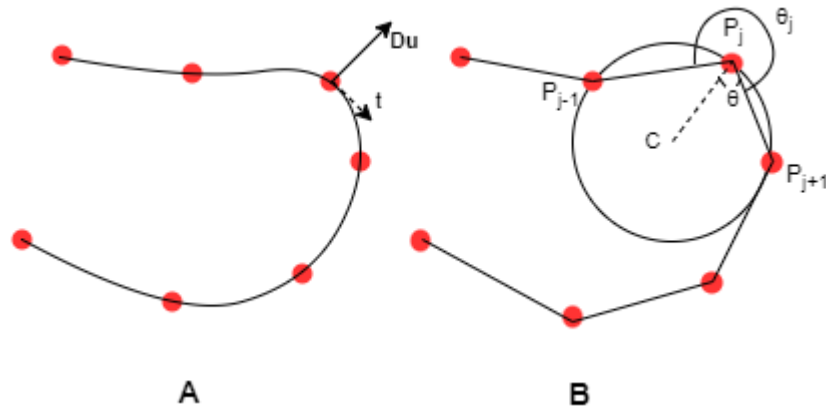
Aşağıda, doğrudan seviye çizgilerinde ayrık eğrilikleri hesaplamak için bir formül sunulmaktadır. Bu formül kullanılarak geliştirilen algoritma öncesinde kenar algılama ve kontur oluşturma işlemleri gerçekleştirilir. Ardından elde edilen konturlar ile ayrık eğrilik fonksiyonu çalıştırılarak tespit yapılmaktadır.

Her bir seviye çizgisinin, sıralı köşeleri ile tanımlanan bir çokgen olarak saklandığını ve sürekli bir seviye çizgisinin ayrıklaştırılmasından oluşmaktadır. Kolaylık olması için,  $\lambda$ 'ya olan bağımlılığı düşürülerek, genel olarak bir seviye çizgisini  $\Sigma = \{P_j(x_j, y_j)\}$  ile  $P_0 = P_N$  olarak gösterilmektedir. Herhangi bir  $P_j$  köşesinde, sınırlanmış yarıçapın tersi olarak ayrı bir eğrilik  $\kappa(P_j)$  hesaplanabilir. Daha kesin olarak, her  $j = 0, 1, \dots, N$  için ardışık üç  $P_{j-1}, P_j, P_{j+1}$  noktası tarafından belirlenen üçgeni göz önünde bulundurur,  $r_j$  salınımlı daire (osculating circle) yarıçapını hesaplar ve (işaretli) ayrık eğriliği tanımlayabilir. Bir eğri üzerinde alınan noktalar ile oluşturulan bir örnek Şekil 3.8'de verilmektedir [14].

$$\kappa(P_j) = \pm 1/r_j \quad (3.2)$$

Düzey çizgisinin, görüntünün gradyanı sol tarafta olacak şekilde yönlendirilmesi gerekmektedir.

$$n = D_u / \|D_u\| \quad (3.3)$$



**Şekil 3.8 :** Görüntünün eğimi sol tarafında olacak şekilde, saat yönünde yönlendirilmiş (kırmızı ok) düz bir çizgi parçası, sınırlanmış yarıçapın tersi olarak, seviye çizgisinin ayrıklaştırılması ve her bir tepe noktasında ayrık eğriliğin hesaplanması.

$P_j$  noktasındaki ayrık eğrilik şu şekilde verilmektedir:

$$K(P_j) := \frac{-2 \sin \theta_j}{\|P_{j-1}P_{j+1}\|} = \frac{-2 \det(P_j P_{j-1} P_j P_{j+1})}{\|P_{j-1}P_j\| \|P_j P_{j+1}\| \|P_{j-1}P_{j+1}\|} \quad (3.4)$$

burada  $\theta_j = \angle(P_j P_{j-1}, P_j P_{j+1})$  açısıdır ve sağ tarafın payı terimi içerir.

$$\det(P_j P_{j-1}, P_j P_{j+1}) := \det \begin{pmatrix} x_{j-1} - x_j & x_{j+1} - x_j \\ y_{j-1} - y_j & y_{j+1} - y_j \end{pmatrix} \quad (3.5)$$

Üçgenin çevreleyen çemberinin merkezi C ve  $\theta = \angle(P_j P_{j+1}, P_j C)$  olsun. C noktası iki dik açıortay üzerinde olduğundan,

$$r_j \cos \theta := \frac{\|P_j P_{j+1}\|}{2} \quad (3.6)$$

$r_j \cos(\theta_j + \theta) := \frac{\|P_{j-1}P_j\|}{2} r_+ = \|P_j P_{j+1}\|$  ve  $r_- = \|P_{j-1}P_j\|$  alınırsa, her iki denklem arasında  $r_j$ 'yi ortadan kaldırarak ve bir toplamın kosinüsünü genişleterek, şu sonuca varılmaktadır:

$$(r_+ \cos \theta_j - r_-) \cos \theta = r_+ \sin \theta_j \sin \theta \quad (3.7)$$

Böylece;

$$\tan \theta = \frac{r_+ \cos \theta_j - r_-}{r_+ \sin \theta_j} \quad (3.8)$$

Elde edilir;

$$\frac{1}{|\cos \theta|} = \sqrt{1 + \tan^2 \theta} = \frac{\sqrt{r_+^2 + r_-^2 - 2r_+ r_- \cos \theta_j}}{r_+ |\sin \theta_j|} = \frac{\|P_j P_{j+1}\|}{r_+ |\sin \theta_j|} \quad (3.9)$$

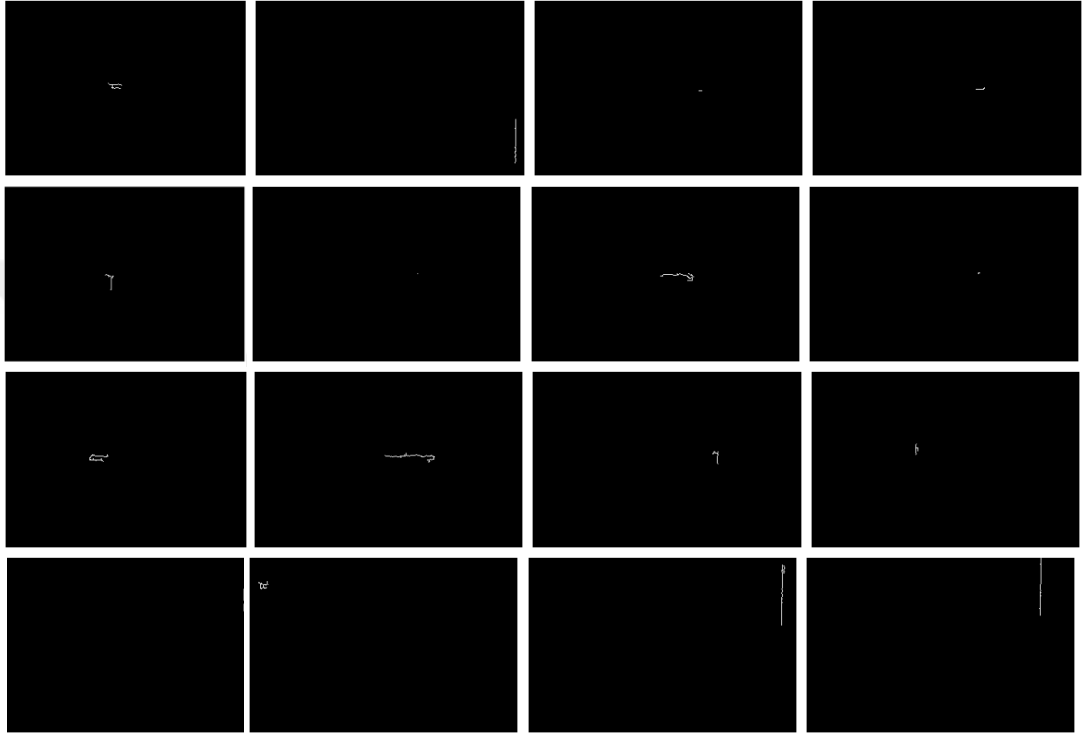
Kosinüs yasasını son eşitlikle birleştirdiğimiz takdirde,

$$\frac{1}{|K(P_j)|} = r_j = \frac{\|P_{j-1}P_{j+1}\|}{2 |\sin \theta_j|} \quad (3.10)$$

Önermenin son eşitliği vektör çarpım özelliğinden gelmektedir.

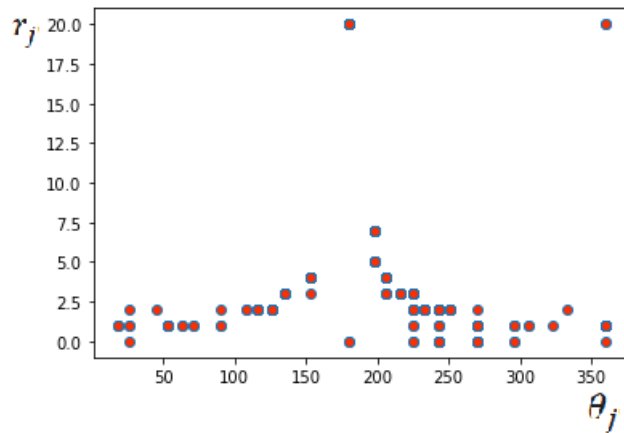
$$\det(P_j P_{j-1} P_j P_{j+1}) = r_+ r_- \sin(\theta_j) \quad (3.11)$$

Bu önermeler kullanılarak geliştirilen Ayrık eğrilik fonksiyonu Şekil 3.6'daki hatalı kumaş görüntüsü ve Canny kenar tespiti ile Şekil 3.7 elde edildikten sonra, kontur algoritması kullanılarak her bir kenarların parça olarak elde edilmesi gerekmektedir. Daha sonra elde edilen 5 nokta üzerindeki her bir kontur ayrık eğrilik fonksiyonu için kullanılarak eğrilik değerleri hesaplanmaktadır. Şekil 3.9'de kontur algoritmasının sonucunda elde edilen her bir parçanın resimleri verilmiştir.



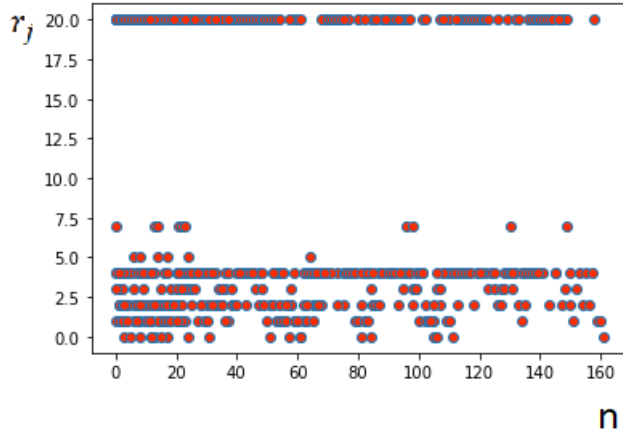
**Şekil 3.9 :** Kontur uygulanmış hatalı kumaş resim parçaları.

Elde edilen konturlar sonucunda DiscreateCurvature fonksiyonunun hesapladığı değerlerden elde edilen dış açı, yarıçap grafiği Şekil 3.10'da gösterilmektedir.



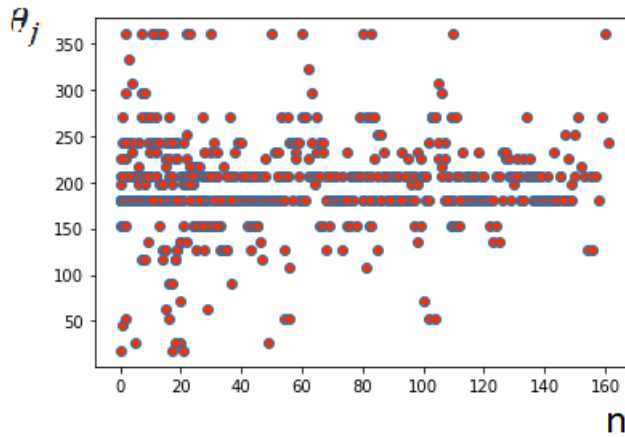
**Şekil 3.10 :**  $\theta_j$ , salınlımlı daire (osculating circle) yarıçapı grafiği.

Elde edilen konturlar sonucunda DiscreateCurvature fonksiyonunun hesapladığı değerlerden elde edilen yarıçap grafiği Şekil 3.11’de gösterilmektedir.



**Şekil 3.11 :** Kontur üzerindeki noktaların yarıçap değerleri grafiği.

Elde edilen konturlar sonucunda DiscreateCurvature fonksiyonunun hesapladığı değerlerden elde edilen dış aç (θ<sub>j</sub>) grafiği Şekil 3.12’de gösterilmektedir.



**Şekil 3.12 :** Kontur üzerindeki noktaların aç değeri grafiği.

DiscreateCurvature fonksiyonu kontur dizileri girdisi ile çalışarak hesaplandığı her bir P<sub>j</sub> noktası için yarıçap(r<sub>j</sub>), aç değeri(θ) ve dış aç (θ<sub>j</sub>) değerleri hesaplamaktadır.

**Çizelge 3.1 :** Eğrilik karmaşıklık matrisi.

|        |          | Tahmin    |            |        |
|--------|----------|-----------|------------|--------|
|        |          | Kusursuz  | Kusurlu    | Toplam |
| Gerçek | Kusursuz | 50<br>%22 | 102<br>%45 | 152    |
|        | Kusurlu  | 52<br>%23 | 22<br>%10  | 74     |
|        | Toplam   | 102       | 124        | 226    |

Eğrilik algoritması kullanılarak elde edilen karmaşıklık matrisi Çizelge 3.1’de verilmektedir. Doğruluk oranı %32 olarak elde edilmiştir.

### 3.3.2 Modifiye edilmiş ayrık eğrilik algoritması

Standart eğrilik algoritmasında, eğri boyunca ardışık 3 nokta seçilmektedir. Ancak, bu seçim aşırı yarıçap değerlerine neden olmaktadır. Bu nedenle, makul yarıçap değerleri elde etmek için algoritmayı eğrinin geometrisini kontrol ederek değiştirilmiştir. Bu güncellemeyi uygulayabilmek için, konkav geometri (örneğin,  $f''(x) < 0$ ) veya konveks geometri (örneğin,  $f''(x) > 0$ ) için eğri fonksiyonunun ikinci türevini incelenmesi planlanmaktadır.

Bu algoritma pseudo kod olarak algoritma 1’de anlatılmıştır.

---

**Algoritma 1: Modifiye Edilmiş Ayrık Eğrilik Algoritması**

---

**Require:** image ( $n \times n$ )  $\neq$  nonempty  
    if Filter selection type then  
        filteredimage = filter(image)  
    end if  
    contours  $\leftarrow$  findContours(filteredimage) ▷ Countourların çıkarılması  
    while  $N \leq \text{len}(\text{counters})$  do  
        CIList = DiscreateCurvature(counters)  
        return(rj,theta,thetaj,kpj) ▷ Yarıçap ve aç ı değ erleri  
    end while  
    while  $m \leq \text{CIList}$  do  
        if thetajis  $< 180$  then ▷ Konveks konkav kontrolü  
            dircurv = 1  
        else if thetajis  $> 180$  then  
            dircurv = 2  
        else if thetajis = 180 then  
            dircurv = 3  
        end if  
        dircurv is changed append values to the CIListedge  
    end while  
    while  $m \leq \text{CIListedge}$  do  
        if CIListedge is between  $x1 < rj < y1, a1 < \text{thetaj} < b1$ type1 then

---

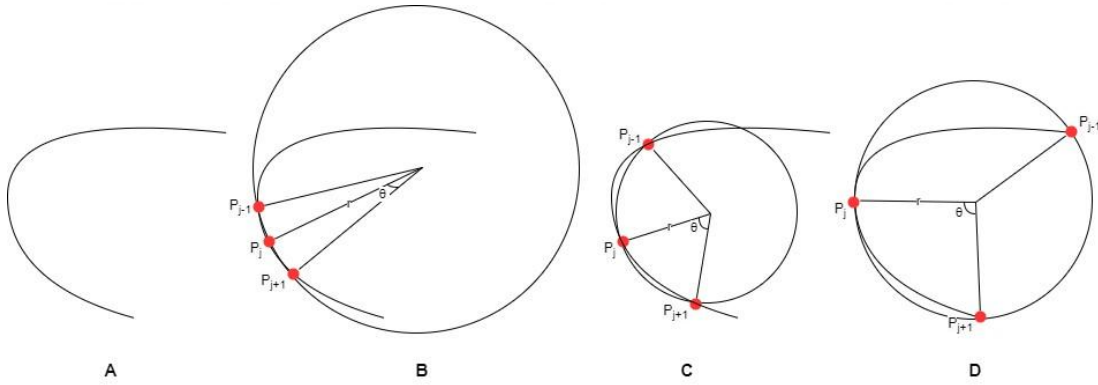
**Algoritma 1: Modifiye Edilmiş Ayırık Eğrilik Algoritması**

---

```
defect type is hole or oil                                ▷ Dairesel hata tespiti
end if
if CIListedge is between  $x2 < r_j < y2$ ,  $a2 < \theta_{taj} < b2$  then
    defect type is vertical or horizontal                ▷ Yatay dikey hata tespiti
end if
end while
```

---

Yeni 3 nokta seçimi, Şekil 3.13'de gösterildiği gibi kapsama alanını maksimize ederek gerçekleştirilir. Böylece, eğrilikler daha doğru bir şekilde hesaplanır. Ayrıca, daha hızlı işlem için toplam nokta sayısı önemli ölçüde azaltılmıştır.



**Şekil 3.13 :** (A) Örnek eğri, (B) standart eğrilik, (C) optimize edilmiş eğrilik, (D) verimli eğrilik.

3 nokta yapısı kullanılarak ayırık eğrilik algoritması ile açı ve yarıçap değerlerinin hesaplanmasını göstermektedir. Şekil 3.13A, hesaplamaların değerlendirileceği bir örnek eğriyi göstermektedir. Şekil 3.13B, noktaların ardışık seçildiği yaygın eğriliği göstermektedir. Noktalar çok yakın olduğunda, yüksek yarıçap değerine ve yanlış açı hesaplamalarına neden olur. Şekil 3.13C ve 3.13D'de, daha doğru ve optimal sonuçlar elde etmek için farklı yöntemler sunulmaktadır.

**Çizelge 3.2 :** Modifiye edilmiş eğrilik karmaşıklık matrisi.

|        |          | Tahmin     |           |        |
|--------|----------|------------|-----------|--------|
|        |          | Kusursuz   | Kusurlu   | Toplam |
| Gerçek | Kusursuz | 108<br>%48 | 44<br>%19 | 152    |
|        | Kusurlu  | 6<br>%3    | 68<br>%30 | 74     |
|        | Toplam   | 114        | 112       | 226    |

Sonuçlar, önerilen yeni yöntemin eğrilik algoritması ile daha doğru sonuçlar elde edilebileceğini göstermektedir ve bu sonuçlar Çizelge 3.2'deki karışıklık matrisinde gösterilmiştir. Doğruluk oranı %78 olarak elde edilmiştir. Bu şekilde, kusur tespiti için kullanılabileceği belirtilmektedir. Bu nedenle, bu yaklaşımı geliştirmek için çalışmalara devam edilecektir.

### 3.4 Spektral Yöntemler

#### 3.4.1 Gabor dönüşümü

Gabor dönüşümü, kısa süreli Fourier dönüşümünün özel bir halidir. Zamanla değişen bir sinyalin yerel bölümlerinin sinüzoidal frekansını ve faz içeriğini belirlemek için kullanılmaktadır. Dönüştürülecek fonksiyon önce bir pencere fonksiyonu olarak kabul edilebilecek bir Gauss fonksiyonu ile çarpılmakta ve elde edilen fonksiyon daha sonra zaman-frekans analizini türetmek için bir Fourier dönüşümü ile dönüştürülmektedir [21-22].

Pencere işlevi, analiz edilen zamana yakın sinyalin daha yüksek ağırlığa sahip olacağı anlamına gelir. Bir  $x(t)$  sinyalinin Gabor dönüşümü şu formülle tanımlanmaktadır:

$$G(f) = \hat{f}_g(t, w) \quad (3.15)$$

$$= \int_{-\infty}^{+\infty} f(\tau) e^{-jw\tau} g(t - \tau) d\tau \quad (3.16)$$

Gabor filtreleri, bilgisayarla görme ve dijital görüntü işleme yöntemleri karşılaştırılması sonucu farklı doğruluk oranları arasında değişen bu yöntemlerden en iyi sonuç veren yöntemlerden olduğu belirtilmiştir. Ancak birçok hata içeren kumaş görüntülerinde Gabor filtreleri yaklaşımı da yüksek doğruluk oranına sahip yöntem olduğu ifade edilmektedir [23,24,25].

Aşağıdaki  $g()$  fonksiyonu, filtre içerisinde kullanılacak Gabor çekirdeği oluşturmasını sağlamaktadır. Bu denklemden,  $\lambda$  sinüzoidal faktörün dalga boyunu temsil eder,  $\theta$  normalin bir Gabor işlevinin paralel şeritlerine yönelimini temsil eder,  $\psi$  faz ofseti,  $\sigma$  Gauss zarfının standart sapmasıdır ve  $\gamma$  uzamsal en-boy oranıdır ve Gabor fonksiyonunun desteğinin eliptikliği belirtilmektedir [26,27].

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right) \quad (3.17)$$

Bu algoritma pseudo kod olarak algoritma 2’de anlatılmıştır.

---

### Algoritma 2: Gabor Dönüşümü Algoritması

---

**Require:** image ( $n \times n$ )  $\neq$  nonempty

kernel  $\leftarrow$  getGaborKernel ((ksize, ksize),  $\sigma$ ,  $\theta$ ,  $\lambda$ ,  $\gamma$ ,  $\phi$ )

if Filter selection type then

    filteredimage = f ilter(image)

end if

Gaborimage  $\leftarrow$  f ilter2D (image, kernel) ▷ Gabor Dönüşümü

while ( $n \times n$ )  $\leq$  GaborTransformImage () do

    Sum the energy density of all pixels. ▷ Enerji yoğunluğu hesabı  
density

end while

if Energy density is  $\geq$  Threshold Energy then

    Image is defected

end if

---

$\lambda$ : Kosinüs çarpanının dalga boyunu belirleyen katsayıdır. Katsayının 1 olması durumunda kosinüs ifadesi sürekli 1 olacağından ( $\cos(2.\pi.x')=1$ ) katsayı 2 veya daha büyük bir tamsayı seçilmelidir.

$\theta$ : Teta doğrudan formül içerisinde görünmese de Gabor filtresinin aslında en önemli değişkenlerinden biridir. Bu değişken  $x'$  ve  $y'$  değerlerinin hesaplanmasında kullanılır ve oluşturulmak istenen Gabor çekirdeğinin yönelim açısıdır.  $x'$  ve  $y'$  değişkenleri verilen bir teta değeri için aşağıdaki formül ile hesaplanmaktadır.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.18)$$

$\phi$ : Fi açısı oluşturulacak çekirdek matrisinin faz açısıdır. Bu değer değiştirilerek filtre  $x$  ekseninde ötelenmektedir.

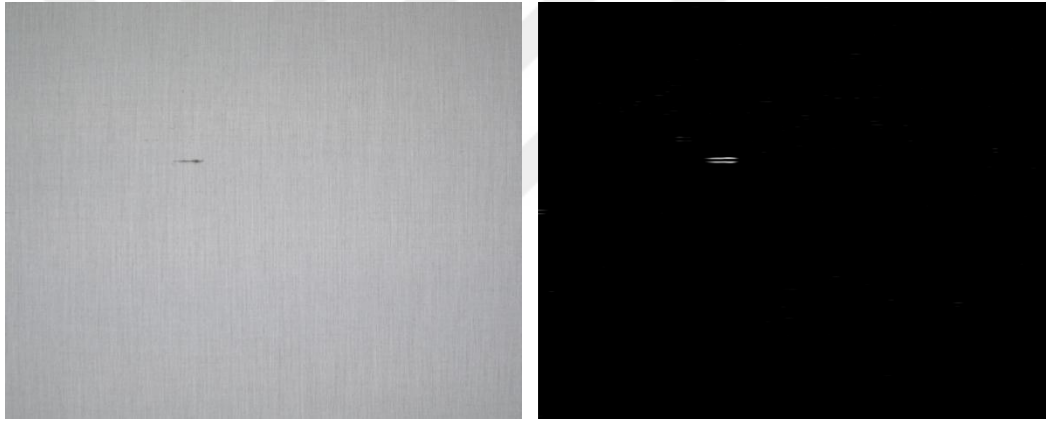


$\sigma$ : Sigma değeri Gaussian fonksiyonun standart sapmasını belirleyen katsayıdır. Bu parametre Gaussian fonksiyonun açıklığını belirlediğinden bu değerin küçük seçilmesi ile Gabor dalgacıkları bir birlerine yaklaşıacaktır.

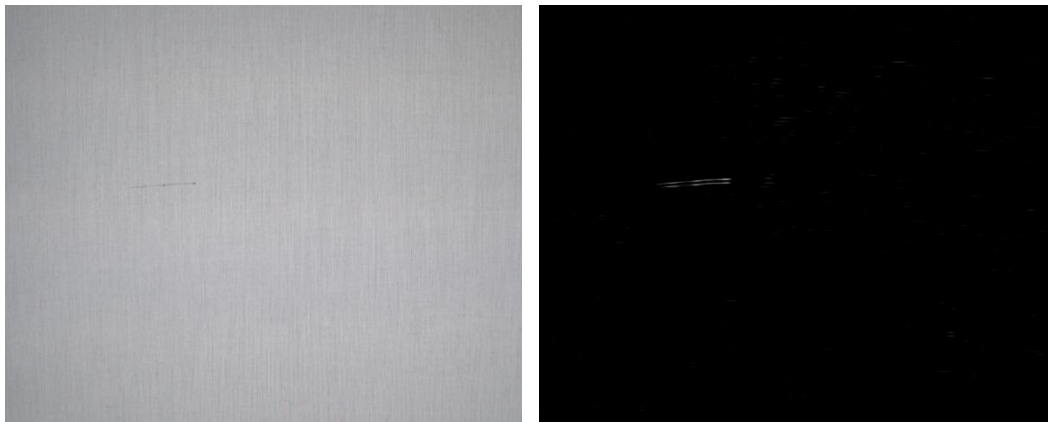
$\gamma$ : Bu değer de verilen standart sapma değerinin  $y'$  için belirlenmesinin sağlamaktadır. Bu değerin 1 olması durumunda oluşacak çekirdek matris  $x$  ve  $y$  için eşit standart sapmaya sahip olduklarından eşit uzunlukta olacakken, farklı bir oran seçildiğinde çekirdek matris dikdörtgene benzer bir şekilde oluşacaktır.

$x, y$ : Bu iki değer oluşturulacak 2 boyutlu çekirdek matrisin koordinatlarını temsil etmektedir.  $N \times N$  büyüklükte bir çekirdek matris için  $x$  ve  $y$  değerleri  $[-(N-1)/2, (N-1)/2]$  aralığında gezilerek çekirdek matris hesaplanmaktadır [28,29].

Şekil 3.14, Şekil 3.15, Şekil 3.16 ve Şekil 3.17’de ham kumaş görüntüsü ve gabor filtresi sonucu verilmektedir.



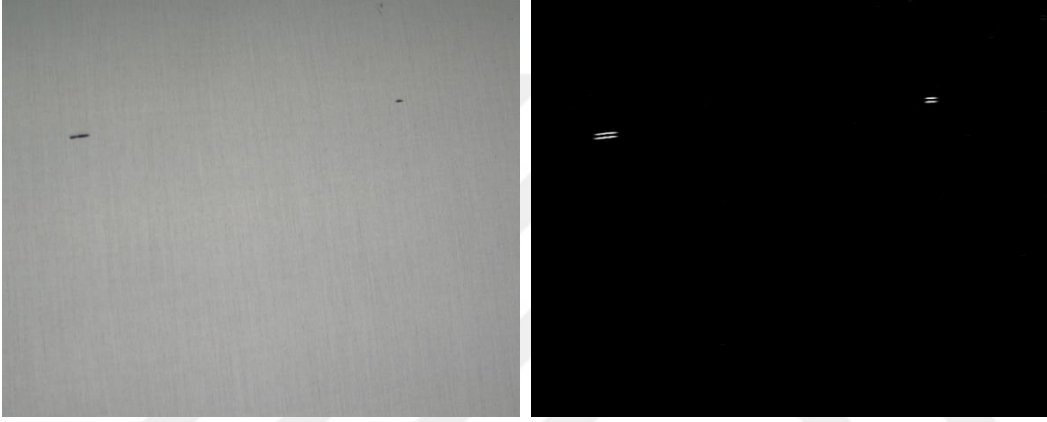
**Şekil 3.14** : Hatalı kumaş ve gabor filtresi uygulanmış görüntüsü (1).



**Şekil 3.15** : Hatalı kumaş ve gabor filtresi uygulanmış görüntüsü (2).

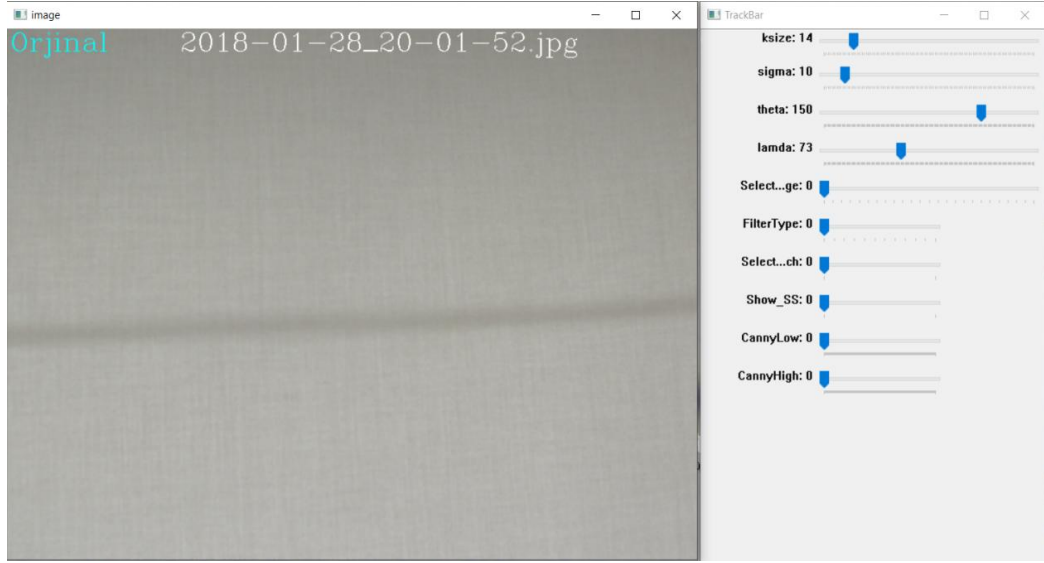


**Şekil 3.16 :** Hatalı kumaş ve gabor filtresi uygulanmış görüntüsü (3).

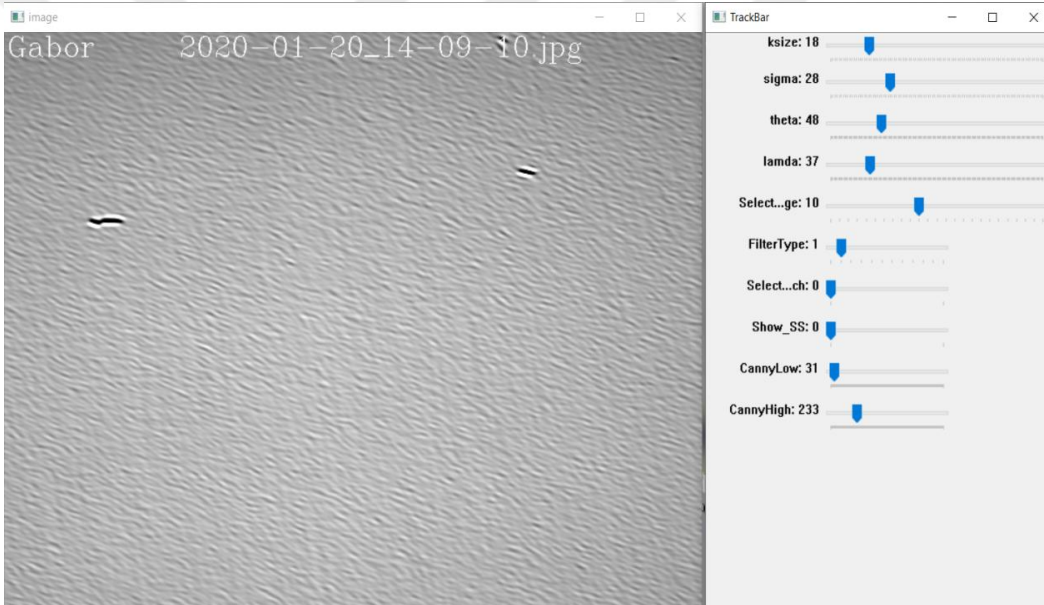


**Şekil 3.17 :** Hatalı kumaş ve gabor filtresi uygulanmış görüntüsü (4).

Python üzerinde istatistiksel ve spektral yöntemlerin test edilebilmesi amacıyla bir test arayüzü geliştirilmiştir. Bu arayüz, farklı resimler üzerinde fonksiyon parametrelerinin kolayca değiştirilmesine olanak tanıyarak, kullanıcıların çeşitli analiz senaryolarını hızlı bir şekilde test etmesini sağlamaktadır. Aynı zamanda, farklı kumaş örneklerinin incelenmesi ve çeşitli dokular üzerindeki etkilerin karşılaştırmalı olarak değerlendirilmesine imkân tanımaktadır. Böylece, yöntemlerin farklı kumaş türlerine adaptasyonu ve performanslarının karşılaştırılması kolaylaşmaktadır. Şekil 3.18 ve 3.19'da bu arayüzün kullanıcıya sunduğu test ortamı ve örnek kullanım senaryoları gösterilmektedir.



**Şekil 3.18 : Kumaş hata test arayüzü (1).**



**Şekil 3.19 : Kumaş hata test arayüzü (2).**

Gabor yöntemi ile elde edilen sonuçlar ve karmaşıklık matrisi Çizelge 3.3'de verilmiştir. Doğruluk oranı %90 olarak elde edilmiştir.

**Çizelge 3.3 : Gabor dönüşümü karmaşıklık matrisi.**

|        |          | Tahmin     |           |        |
|--------|----------|------------|-----------|--------|
|        |          | Kusursuz   | Kusurlu   | Toplam |
| Gerçek | Kusursuz | 132<br>%58 | 29<br>%9  | 152    |
|        | Kusurlu  | 1<br>%1    | 73<br>%32 | 74     |
|        | Toplam   | 133        | 93        | 226    |

### 3.4.2 Curvelet dönüşümü

Candes ve Donoho tarafından tanıtılan Curvelet Dönüşümü, özellikle görüntü verilerindeki anizotropik özellikler ve eğrisel tekillikler gibi zorlukların üstesinden gelmede Wavelet Dönüşümüne kıyasla önemli başarı sağlamaktadır. Wavelet Dönüşümü, noktasal tekillikleri yakalamada oldukça etkiliyken, eğrisel özellikleri ve geometrileri temsil etmede zorlanmaktadır [30, 31]. Bu sınırlama, wavelet'lerin izotropik ölçeklendirme doğasından kaynaklanmaktadır ve doğal görüntülerin geometrisi ile uyumlu olmamaktadır.

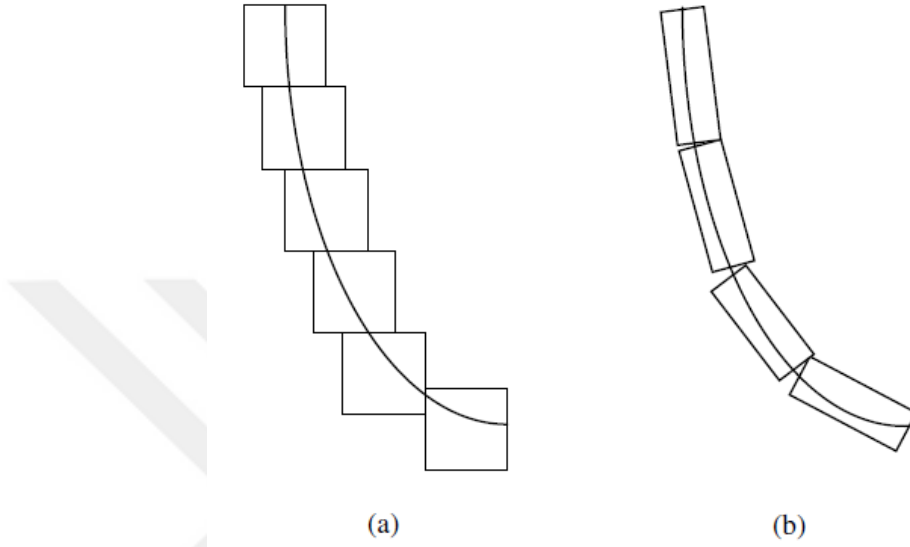
Curvelet Dönüşümü ise, anizotropik ölçeklendirme ve yön hassasiyeti kullanarak kenarlar, konturlar ve diğer geometrik olarak karmaşık yapılar için ideal bir analiz sağlamaktadır [32]. Dönüşüm, temel fonksiyonlarını eğrisel tekilliklere hizalayarak bu tür özelliklerin seyrek ve verimli bir temsilini sunmaktadır. Bu özellik, görüntü gürültüsünün giderilmesi, kenar algılama, doku bölütleme ve görüntü sıkıştırma gibi sağlam görüntü analizi gerektiren uygulamalarda Curvelet Dönüşümünü önemli bir araç haline getirmektedir [33].

Curvelet Dönüşümünün temel ilkesi, bir görüntüyü belirli geometrik özellikleri birden fazla ölçek ve yönde temsil eden bir dizi katsayıya ayırabilmesinden gelmektedir. Bu ayrıştırma süreci, Curvelet temel fonksiyonlarının görüntünün geometrik yapısına göre uzatılmış ve yönlendirilmiş olmasını sağlayan anizotropik ölçeklendirme yöntemiyle gerçekleştirilir. Anizotropik ölçekleme yasası "genişlik  $\sim$  uzunluk<sup>2</sup>" olarak tanımlanmakta olup, Curvelet Dönüşümünün uzatılmış yapılara doğal olarak uyum sağlamasını mümkün kılmaktadır. Bu özellik, özellikle yumuşak eğriler veya uzatılmış kenarları içeren senaryolarda, geleneksel dönüşüm tekniklerine kıyasla üstün performans sergilemesini sağlamaktadır [34, 35].

#### 3.4.2.1 Matematiksel temeller

Curvelet, dönüşümü, yüksek anizotropik özelliğe sahip özel bir iki boyutlu dalgacık dönüşümüdür. Diğer yerleştirilmiş dalgacık dönüşümleri gibi, çok ölçekli analiz, zaman-frekans sınırlaması ve yönsel sınırlama sağlamaktadır. Ancak, klasik dalgacık dönüşümünden farklı olarak, Curvelet dönüşümü daha hassas bir açısız ayrıştırma sunarak özellikle eğri kenarları ve doğrusal olmayan yapıları daha etkili bir şekilde temsil edebilmektedir.

Şekil 3.20, Curvelet dönüşümünün bu önemli özelliklerinden birini görsel olarak ortaya koymaktadır. Bu dönüşüm, geleneksel dalgacık dönüşümlerine kıyasla kenarları daha seyrek bir şekilde temsil etmekte ve böylece yüksek doğrulukla kenar tespiti ve yapı analizi gerçekleştirebilmektedir.



**Şekil 3.20 :** . a) Dalgacık Dönüşümü b) Curvelet Dönüşümünde kenarların temsili.

Uzayda bir Curvelet, ölçek ( $j$ ), konum ( $b$ ) ve yön ( $\theta$ ) bileşenlerine göre şu şekilde ifade edilir:

$$\Psi_{j,b,\theta}(x) = 2^{\frac{-3j}{4}} \Psi \left( D_j R_{\theta}^{-1} (x - b) \right) \quad (3.19)$$

$$D_j = \begin{pmatrix} 2^{-j} & 0 \\ 0 & 2^{-j/2} \end{pmatrix}, R_{\theta}^{-1} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \quad (3.20)$$

Matematiksel olarak Ayrık Curvelet Dönüşümü,  $f(x)$  fonksiyonunun Curvelet katsayılarına ayrıştırılması olarak tanımlanır. Curvelet dönüşümü aşağıdaki şekilde hesaplanmaktadır:

$$c(j, l, k) = \langle f(x), \Phi\{j, l, \theta\} \rangle \quad (3.21)$$

Burada:

$j$  : Curvelet'in ölçeğini,

$l$  : yönelim (veya açığı),

$k$  : uzamsal konumu,

$\phi_{j,l,k}$ : Curvelet baz fonksiyonu.

Bu ayrıştırma, giriş fonksiyonunu çoklu ölçekli ve çoklu yönlü bileşenlere ayırmaktadır. Her bileşen, belirli bir ölçek ve yönde fonksiyonun geometrik özelliklerine karşılık gelmektedir [33,36].

Curvelet dönüşümü, enerji korunumu ilkesine uyar:

$$|f|^2 = \sum_{j,l,k} |c_{j,l,k}|^2 \quad (3.22)$$

Bu, dönüşümün kayıpsız olduğunu göstermektedir [37,38].

### 3.4.2.2 Anizotropik ölçeklendirme

Curvelet Dönüşümünün ayırt edici özelliklerinden biri, anizotropik ölçeklendirme kullanmasıdır. Wavelet'lerde olduğu gibi tüm yönleri eşit şekilde ele alan izotropik ölçeklendirme yerine, Curvelet Dönüşümü temel fonksiyonları anizotropik bir şekilde ölçeklendirmektedir:

$$Width \sim Length^2 \quad (3.23)$$

Bu ölçeklendirme, Curvelet elemanlarının görüntüdeki eğrisel veya uzatılmış özelliklerle hizalanmasını sağlamaktadır. Bu sayede dönüşüm, bu tür özellikleri izotropik dönüşümlere kıyasla daha seyrek ve doğru bir şekilde temsil edebilmektedir [39,40].

### 3.4.2.3 Curvelet dönüşümünün uygulamaları

Wavelet'lerin izotropik ve nokta benzeri tekillikler için ideal olmasına karşın, Curvelet'ler anizotropiktir ve çizgi ile eğri tekilliklerini analiz etmek için daha uygun olmaktadır. Bu özellik, Curvelet Dönüşümünü, hata tespiti ve özellik çıkarımı gibi uygulamalar için özellikle değerli kılmaktadır [41, 42]. Bu çalışmada Curvelet Dönüşümü, Uniform Discrete Curvelet Transform (UDCT) yöntemi kullanılarak uygulanmıştır. UDCT, Curvelet katsayılarını verimli bir şekilde hesaplamakta ve görüntünün çok ölçekli, çok yönlü bir analizini mümkün kılmaktadır. Her ölçek, ince

ayrıntılardan kaba yapılara kadar belirli bir boyuttaki özellikleri yakalarken, yön bantları belirli yönelimlere karşı hassasiyeti artırmaktadır. Bu ayrıştırma, görüntünün geometrik yapısının kapsamlı bir şekilde analiz edilmesini sağlamaktadır [43]. Curvelet dönüşümü pseudo kod olarak algoritma 3’de anlatılmıştır.

---

**Algoritma 3: Curvelet Dönüşümü Algoritması**

---

```
Require: image ( $n \times n$ )  $\neq$  nonempty
# Curvelet dönüşümünü uygula
apply_curvelet_transform(image, nscales, nbands_per_direction)

return transform, coefficients                                ▷ Curvelet Katsayıları

# Görüntüyü yeniden oluştur
reconstructed_image(transform, coefficients)                  ▷ Yapılandırılmış Görüntü

# Yeniden oluşturma farkını analiz et
diff=analyze_reconstruction(image, reconstructed_image)      ▷ Fark analizi

# Kusurları belirginleştir ve işaretle
defects = highlight_defects(image, diff, threshold)
```

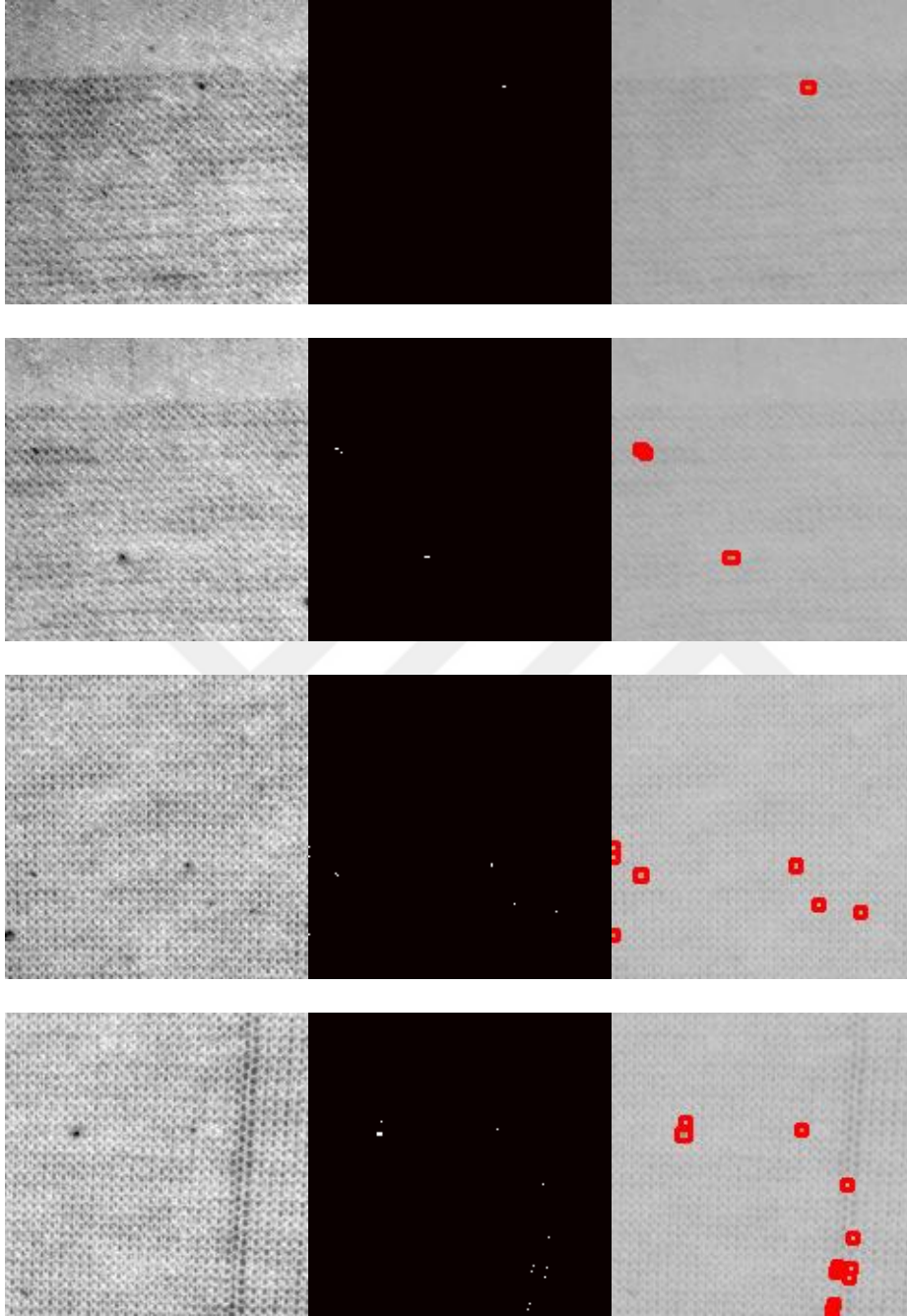
---

Curvelet dönüşümünde kritik bir parametre, görüntünün analiz edildiği ölçek veya çözünürlük seviyelerinin sayısını belirleyen nscales parametresidir. Bu çalışmada nscales = 5 olarak seçilmiş olup, bu değer, görüntünün beş farklı çözünürlük seviyesine bölündüğünü ifade etmektedir. Bu ölçeklendirme, dönüşümün hem küçük kusurlar gibi ince detayları hem de daha büyük düzensizlikler veya desenler gibi geniş ölçekli yapıları yakalayabilmesini sağlamaktadır.

Diğer önemli bir parametre ise, her bir ölçek içerisinde yer alan yön bantlarının sayısını belirleyen yön başına bant sayısıdır. Curvelet dönüşümü doğası gereği yönelimlidir ve bu parametre, yön analizi hassasiyetini kontrol etmektedir. Bu çalışmada nbands per direction = 32 olarak belirlenmiş olup, bu seçim yüksek yön hassasiyeti sağlamaktadır. Bu ayar, farklı yönlerdeki özelliklerin hassas tespitini sağlayarak Curvelet dönüşümünü karmaşık yapıları analiz etmede etkili kılmaktadır.

Aşağıda, Curvelet dönüşümünün uygulandığı görüntülerin analiz sonuçları sunulmaktadır. Şekil 3.21’de sırasıyla orijinal görüntü, fark haritası ve işlenmiş

görüntüler gösterilmektedir. Bu görseller, dönüşümün ölçek ve yön bileşenleriyle kusur tespitine katkısını göstermektedir.



**Şekil 3.21 :** Orijinal görüntü, fark haritası ve işlenmiş görüntüler.

Curvelet dönüşümü, `transform.forward(image)` yöntemi ile gerçekleştirilmektedir. Bu yöntem, görüntüyü Curvelet temsiline dönüştürmektedir. Daha sonra, `transform.backward(coefficients)` yöntemi kullanılarak, Curvelet katsayılarıyla görüntü tekrar oluşturmaktadır. Bu yeniden oluşturulan görüntü, yalnızca Curvelet katsayıları

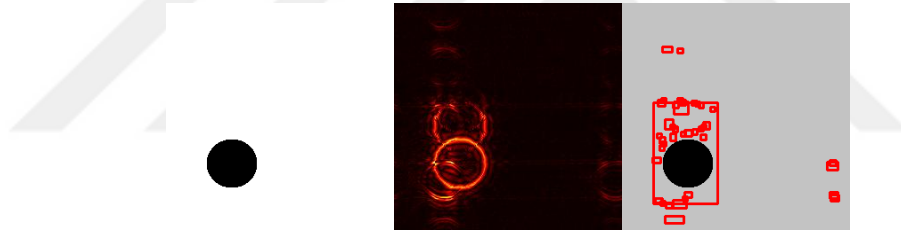


tarafından yakalanan bilgileri içermekte ve böylece gürültü ve ilgisiz ayrıntılar etkili bir şekilde filtrelenmektedir [44-46].

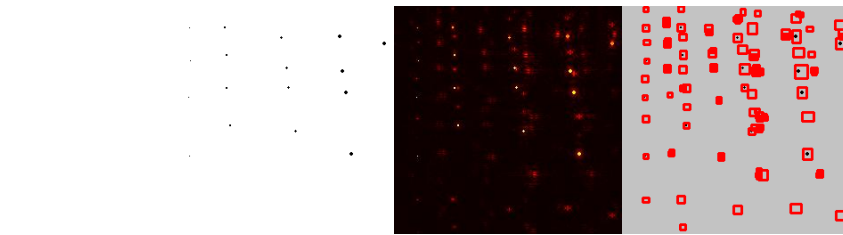
#### 3.4.2.4 Curvelet dönüşümünün uygulamalarında iyileştirmeler

Curvelet dönüşümü, özellikle çoklu ölçek ve yön duyarlılığı sayesinde parazitleri azaltmada etkili bir yöntemdir. Görüntü üzerindeki parazitler genellikle yüksek frekanslı bileşenler olarak yer alırken, Curvelet dönüşümü bu bileşenleri farklı ölçeklere ve yönlere ayırmaktadır. Parazitleri filtrelemek için yüksek frekanslı katsayılar eşik değerine göre sıfırlanarak veya baskılanarak işlenmektedir. Böylece, düşük frekanslı bileşenlerde yer alan asıl yapısal özellikler korunurken, parazit etkisi önemli ölçüde azaltılmaktadır. Sonuç olarak, yeniden inşa edilen görüntü hem daha net hem de parazitlerden arındırılmış bir yapıya sahip olur ve bu da özellikle dokusal analiz veya hata tespiti gibi uygulamalar için büyük avantaj sağlamaktadır.

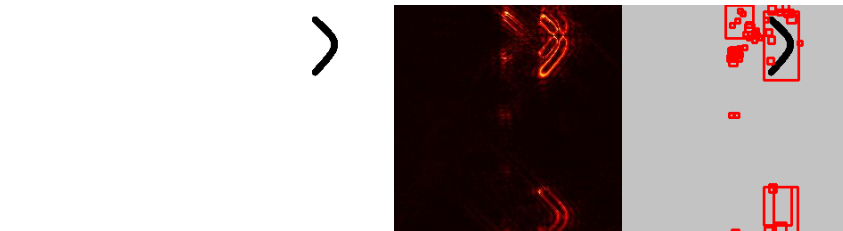
Aşağıda, Şekil 3.22, Şekil 3.23, Şekil 3.24 ve Şekil 3.25’de Curvelet filtreleme işlemi uygulanmamış ve parazitleri hala içeren farklı tiplerde görüntüler bulunmaktadır.



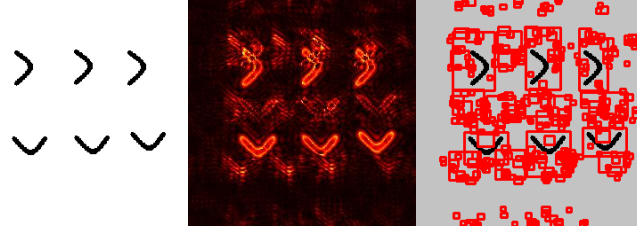
Şekil 3.22 : Orijinal görüntü, filtersiz fark haritası ve fark işaretlenmiş görüntü.



Şekil 3.23 : Orijinal görüntü, filtersiz fark haritası ve fark işaretlenmiş görüntü.

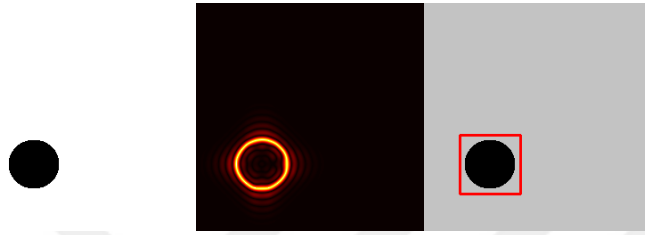


Şekil 3.24 : Orijinal görüntü, filtersiz fark haritası ve fark işaretlenmiş görüntü.

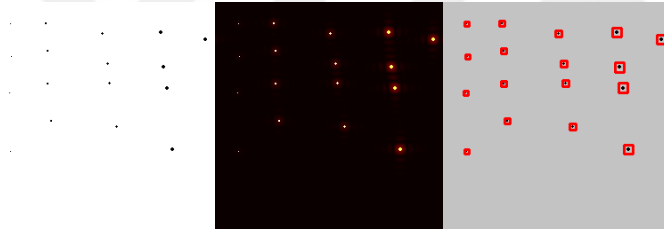


**Şekil 3.25 :** Orijinal görüntü, filtresiz fark haritası ve fark işaretlenmiş görüntü.

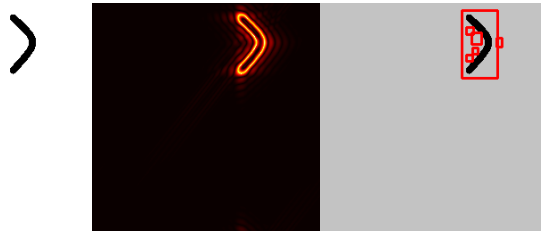
Aşağıda, Şekil 3.26, Şekil 3.27, Şekil 3.28 ve Şekil 3.29’de Curvelet filtreleme işlemi uygulanmış ve parazitlerden büyük ölçüde arındırılmış görüntüler bulunmaktadır.



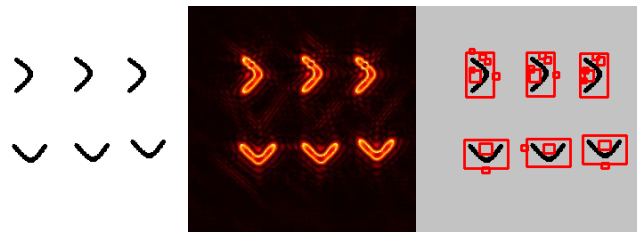
**Şekil 3.26 :** Orijinal görüntü, filtreli fark haritası ve fark işaretlenmiş görüntü.



**Şekil 3.27 :** Orijinal görüntü, filtreli fark haritası ve fark işaretlenmiş görüntü.



**Şekil 3.28 :** Orijinal görüntü, filtreli fark haritası ve fark işaretlenmiş görüntü.



**Şekil 3.29 :** Orijinal görüntü, filtreli fark haritası ve fark işaretlenmiş görüntü.

Curvelet dönüşümünün uygulanması sonrası elde edilen karmaşıklık matrisi ise Çizelge 3.4’de verilmiştir. Doğruluk oranı %80 olarak elde edilmiştir.

**Çizelge 3.4 : Curvelet dönüşümü karmaşıklık matrisi.**

|        |          | Tahmin     |           |        |
|--------|----------|------------|-----------|--------|
|        |          | Kusursuz   | Kusurlu   | Toplam |
|        |          |            |           |        |
| Gerçek | Kusursuz | 122<br>%54 | 30<br>%14 | 152    |
|        | Kusurlu  | 13<br>%6   | 61<br>%26 | 74     |
|        | Toplam   | 135        | 91        | 226    |

Bu uygulamada kullanılan parametreler (nscales=5 ve nbands per direction=32), hesaplama verimliliği ve özellik hassasiyeti arasında bir denge sağlamaktadır. Bu kombinasyon hem ince hataları hem de daha büyük, daha yapılandırılmış düzensizlikleri tespit ederek görüntünün kapsamlı bir analizini mümkün kılmaktadır.

### 3.4.3 Hough dönüşümü

Hough Dönüşümü, Paul Hough tarafından 1962 yılında tanıtılan, parametrik şekilleri (çizgiler, daireler ve elipsler gibi) görüntülerde algılamak için tasarlanmış temel bir özellik çıkarım tekniğidir. Geleneksel kenar algılama yöntemleri yalnızca piksel düzeyindeki yoğunluk gradyanlarına dayanırken, Hough Dönüşümü kenar noktalarını görüntü uzayından daha yüksek boyutlu bir parametre uzayına eşleyerek çalışmaktadır. Bu eşleme, yöntemin gürültülü veri, eksik şekiller veya kısmi örtüşme gibi zorlu koşullarda bile şekilleri tanımlamasına olanak tanımaktadır. Bu dayanıklılık, Hough Dönüşümünü endüstriyel denetim, medikal görüntüleme, bilgisayarla görü ve otonom navigasyon gibi birçok alanda önemli bir araç haline getirmiştir [47-49].

Hough Dönüşümünün gücü, parametrik bir denklemin tanımladığı tüm olası şekilleri sistematik olarak keşfetme yeteneğinde yatmaktadır. Örneğin, bir çizgi algılama durumunda, Kartezyen koordinatlardaki  $y=mx+by$  çizgi denklemi, şu polar forma dönüştürülmektedir:

$$\rho = x \cos \theta + y \sin \theta \quad (3.24)$$

Bu dönüşüm, sonsuz eğimler ile ilgili sorunları çözmekte ve çizgi algılama için daha kararlı bir temsil sağlamaktadır. Görüntü uzayındaki her bir kenar noktası, parametre uzayında bir sinüzoidal eğriye atanmakta ve bu eğrilerin kesişimleri tespit edilen çizginin parametrelerine karşılık gelmektedir. Bu prensip, daireler  $((x - a)^2 + (y - b)^2 = r^2)$  ve elipsler gibi diğer geometrilere, parametre uzayının boyutunu artırarak genişletilebilmektedir [50].

Parametre uzayının ayrık bir temsili olan biriktirici dizi (accumulator array), belirli parametrelere sahip şekillerin varlığını gösteren tepe noktalarını içermektedir. Bu biriktirme mekanizması, Hough Dönüşümünün gürültü veya şekil parçalanması gibi durumlarda bile dayanıklı bir şekilde çalışmasını sağlamaktadır. Bu özellik, özellikle tekstil kusur tespiti ve kalite kontrolünde uyarlanabilirliğini ortaya koymaktadır [51].

### 3.4.3.1 Matematiksel temeller

#### Çizgi algılama

Kartezyen koordinatlarda bir çizgi şu denklemle ifade edilmektedir:

$$y = mx + b \quad (3.25)$$

Ancak, sonsuz eğim değerleri gibi sorunlarla başa çıkmak için bu denklem polar koordinatlara şu şekilde dönüştürülmektedir:

$$\rho = x \cos \theta + y \sin \theta \quad (3.26)$$

Burada:

$\rho$ , orijinden çizgiye olan dik mesafeyi,

$\theta$ , x-ekseni ile çizgiye dik olan doğrunun arasındaki açığı ifade etmektedir.

Görüntüdeki her bir kenar pikseli  $(x, y)$ , parametre uzayında  $(\rho, \theta)$  bir sinüzoidal eğriye katkıda bulunmaktadır. Bu eğrilerin kesişimleri, birden çok kenar pikselinin kolektif kanıtını temsil ettiği için tespit edilen çizginin parametrelerine karşılık gelmektedir [52].

### Biriktirici dizi

Biriktirici dizi, parametre uzayının ayrık bir temsildir. Kenar algılanmış görüntüdeki her piksel, üzerinden geçen tüm olası çizgiler için biriktirici dizinin ilgili hücrelerini arttırmaktadır. Biriktirici dizideki tepe noktaları, görüntüdeki en olası çizgilerin parametrelerini  $(\rho, \theta)$  temsil etmektedir. Bu oylama mekanizması, birden fazla pikselden elde edilen birleşik kanıtlara dayandığı için, gürültü ve eksik veriler karşısında yüksek düzeyde dayanıklılık sağlamaktadır [53, 54].

### Daire algılama

Bilinen bir yarıçapa  $r$  ve bilinmeyen bir merkez  $(a, b)$  noktasına sahip bir daire şu denklemle ifade edilmektedir:

$$(x - a)^2 + (y - b)^2 = r^2 \quad (3.27)$$

Merkezi  $(a, b)$  parametreleştirerek, her kenar pikseli belirli bir  $r$  için  $(a, b)$  uzayındaki biriktirici diziye oy katkısında bulunmaktadır. Bu üç boyutlu parametre uzayındaki  $(a, b, r)$  tepe noktaları, tespit edilen dairelerin parametrelerini göstermektedir. Bu yaklaşım, parametre uzayının boyutunu genişleterek değişen yarıçaplı daireleri işlemek için de genelleştirilebilmektedir [55].

#### 3.4.3.2 Hough dönüşümünün kumaş kusur tespitindeki uygulamaları

Hough Dönüşümü, gürültülü ve parçalanmış kenar verilerini bir araya getirme yeteneği sayesinde, tekstil kusur tespitinde kullanılabilecek bir araç haline gelmiştir ve tekstil üretimindeki kalite kontrol süreçlerini iyileştirme potansiyeli bulunmaktadır.

Bu çalışmada, sonsuz çizgiler yerine çizgi segmentlerini algılamak için standart Hough Dönüşümünün bir uzantısı olan HoughLinesP yöntemi kullanılmıştır. Önemli parametreler şunlardır:

$\rho$ : Piksel cinsinden parametre uzayının çözünürlüğü,

$\theta$ : Açı çözünürlüğü (radyan cinsinden),

threshold: Bir çizginin algılanması için gereken minimum oy sayısı,

minLineLength: Çizgi segmentinin geçerli sayılması için gereken minimum uzunluk (piksel cinsinden),

maxLineGap: Çizgi segmentlerini tek bir çizgi olarak kabul etmek için izin verilen maksimum boşluk.

Bu çalışmada kullanılan parametreler  $\rho=1$ ,  $\theta=\pi/180$  (1 derece çözünürlük),  $\text{threshold}=5$ ,  $\text{minLineLength}=2$  piksel ve  $\text{maxLineGap}=10$  olarak ayarlanmıştır. Hough dönüşümü algoritması pseudo kod olarak algoritma 4’de anlatılmıştır.

Kenar algılama işlemi Canny algoritması kullanılarak gerçekleştirilmiş ve ardından Hough Dönüşümü uygulanarak doğrusal özellikler belirlenmiştir. Bu özellikler, orijinal görüntü üzerine mavi çizgilerle işaretlenmiş olarak görselleştirilmiştir. Bu yaklaşım, doğrusal kusurları etkili bir şekilde vurgulayarak, yapılandırılmış düzensizlikleri algılamada yöntemin dayanıklılığını ortaya koymaktadır [56,57].

Şekil 3.30, Hough Dönüşümü kullanılarak kusur tespit sürecinin adım adım görselleştirmesini sunmaktadır. İlk görüntü, analiz edilen belirli bir alanı vurgulayan orijinal görüntüden kırılmış ilgi bölgesini vermektedir. İkinci grafik, Canny algoritması kullanılarak gerçekleştirilen kenar tespiti sonucunu göstermektedir.

---

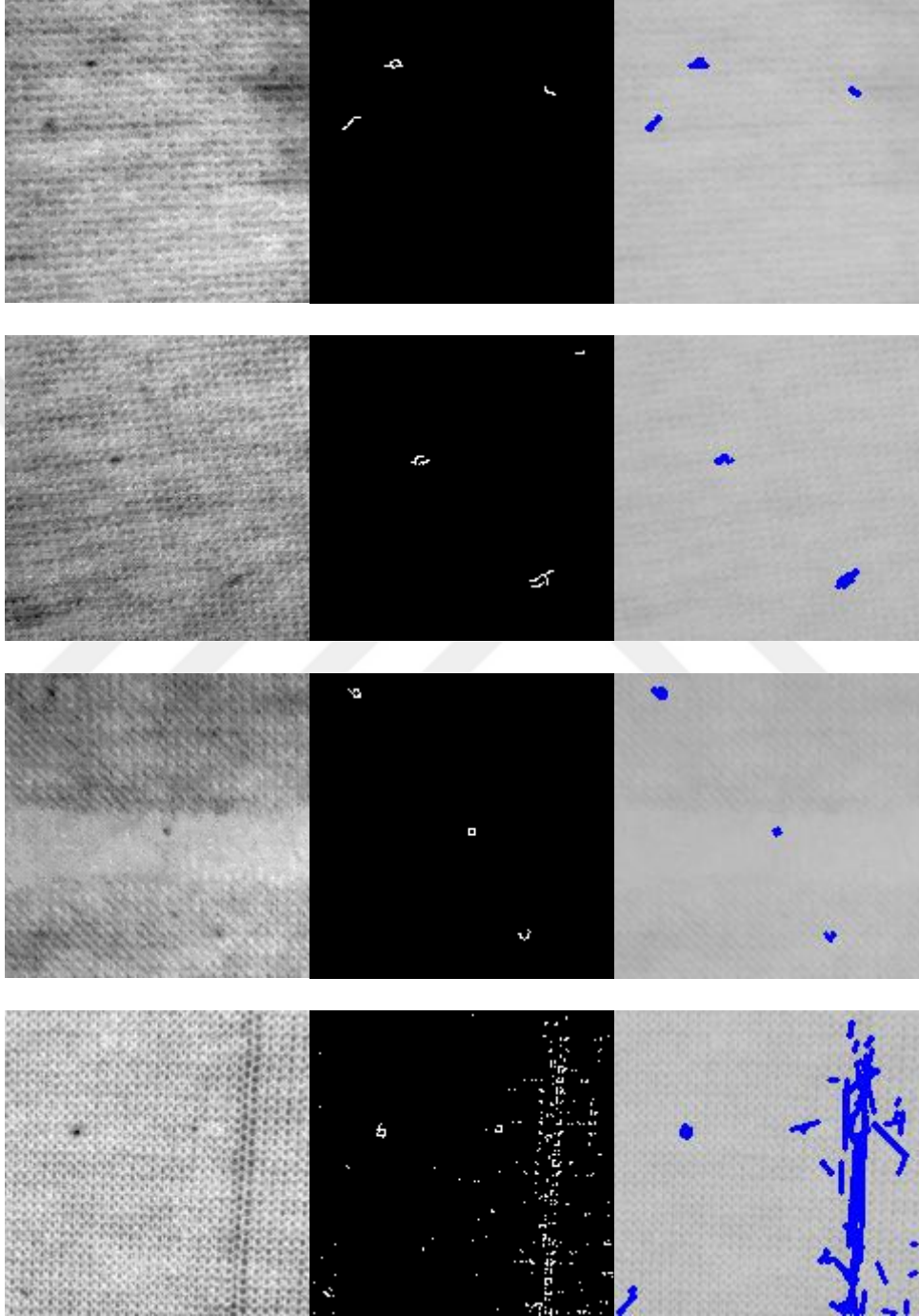
#### Algoritma 4: Hough Dönüşümü Algoritması

---

```
Require: image ( $n \times n$ )  $\neq$  nonempty
    if Filter selection type then
        filteredimage = filter(image)
    end if
    edges  $\leftarrow$  Canny(image, treashold1, treashold2)            $\triangleright$  Canny kenar alg.
    lines,holes = HoughLinesP(edges, rho, theta, threshold, minLineLength,
maxLineGap)            $\triangleright$  Hough dönüşümü
    # Kusurları belirginleştir ve işaretle
    defects = highlight_defects(image, threshold)
    if Energy density is  $\geq$  Threshold Energy then
        Image is defected
    end if
```

---

Bu görüntüde kenarlar, siyah bir arka plan üzerinde parlak beyaz çizgiler olarak vurgulanmış olup, önemli yoğunluk değişikliklerini temsil etmektedir. Bu adım, görüntü verilerini sadeleştirerek temel yapısal özellikleri izole etmekte ve Hough Dönüşümü için uygun giriş verisini hazırlamaktadır.



**Şekil 3.30 :** Orijinal görüntü, kenar filtresi uygulanmış görüntü, çıkış görüntüsü.

Üçüncü görüntü, Hough Dönüşümü ile tanımlanan doğrusal özelliklerin orijinal görüntünün üzerine mavi çizgilerle işaretlendiği tespit edilen kusurları göstermektedir.

Hough Dönüşümünün uygulanmasının ardından elde edilen karışıklık matrisi Çizelge 3.5'de sunulmaktadır. Doğruluk oranı %87 olarak elde edilmiştir.

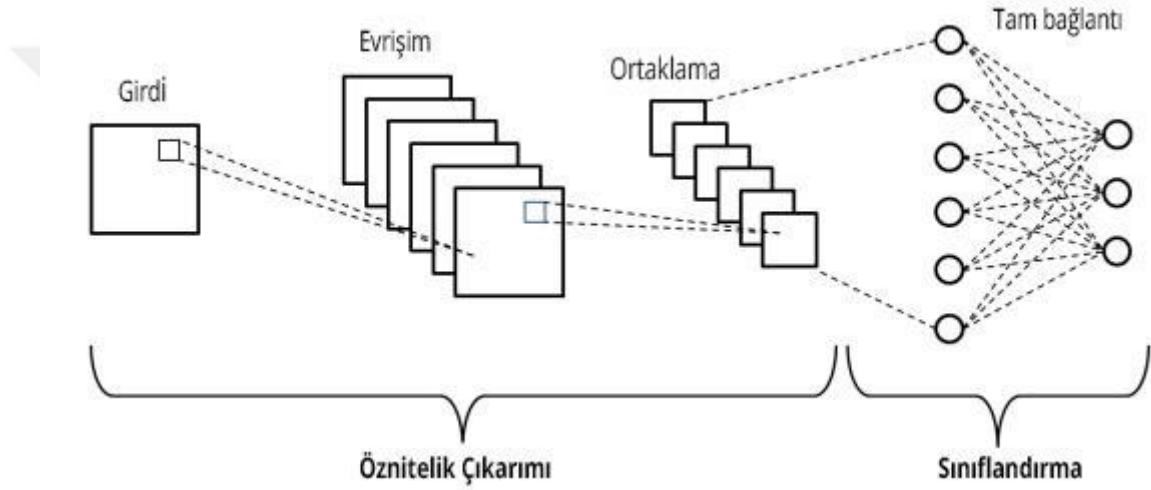
**Çizelge 3.5 :** Hough dönüşümü karmaşıklık matrisi.

|        |          | Tahmin     |           |        |
|--------|----------|------------|-----------|--------|
|        |          | Kusursuz   | Kusurlu   | Toplam |
|        |          |            |           |        |
| Gerçek | Kusursuz | 133<br>%59 | 19<br>%8  | 152    |
|        | Kusurlu  | 12<br>%5   | 62<br>%28 | 74     |
|        | Toplam   | 145        | 81        | 226    |



#### 4. ÖĞRENME TABANLI HATA TANIMA ALGORİTMALARI

Evrişimli sinir ağları (ESA), yapay sinir ağlarının özellikle görüntü sınıflandırma ve bilgisayarlı görü gibi problemlerde kullanılan özel bir araştırma alanıdır. Bir evrişimli Şekil 4.1’de de görüleceği üzere sinir ağı evrişim, ortaklama, tam bağlantı gibi katmanlardan oluşmaktadır [58].



Şekil 4.1 : Basit bir ESA mimarisi.

##### 4.1 Evrişim Katmanı

Evrişimli sinir ağlarında bir görüntünün piksel matrisinden çıkarılan özellikler bu katmandaki evrişim işlemi sayesinde meydana gelmektedir. Evrişim işleminde filtre (kernel) adı verilen 3x3, 5x5 vb. boyutlardaki ağırlık matrisleri görüntü üzerinde gezdirilmektedir. Ağı her eğitim turu sonunda bu filtrelerin üzerindeki ağırlık değerleri güncellenmekte ve filtrelerde öğrenme işlemi gerçekleştirilmektedir [59].

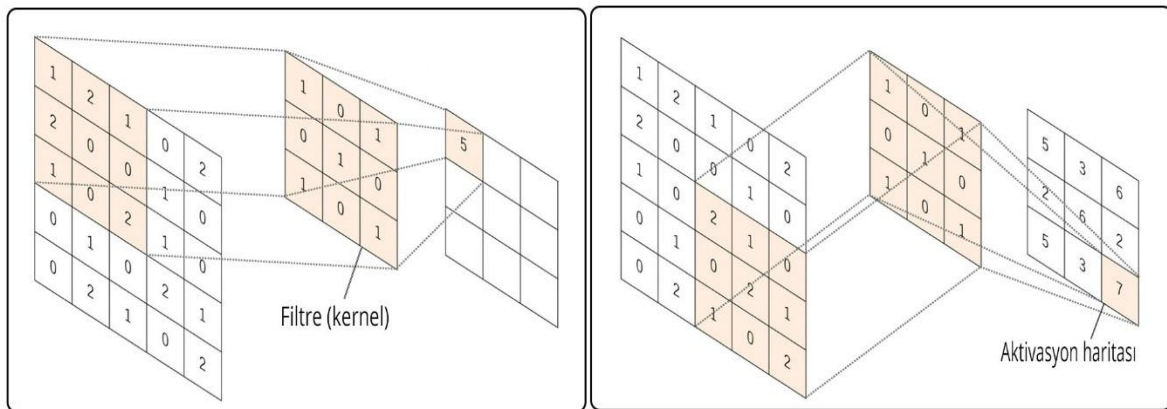
Filtreleme işlemi bir görüntünün sol üst kısmından başlayarak filtrenin sağa kaydırılması sureti ile gerçekleştirilir. Filtre uygulamada ağırlık matrisi ile girdi matrisinin filtre ile örtüşen kısmındaki değerler birbirleri ile çarpılmaktadır (Hadamard çarpımı). Görüntünün en sağına geldiğinde ise filtre tekrar görüntünün soluna ancak alttaki hücrelerine gelmekte ve sağa kaydırma işlemi devam etmektedir. Kaydırma

işlemi (stride) genelde 1 piksel olarak uygulansa da bu değer ESA mimarisi ve girdi matrisinin özelliklerine göre model tasarımcısı tarafından değiştirilebilmektedir. Bunun yanında bazen görüntü matrisi, filtre boyutu ve kaydırma değerlerindeki uyumsuzluklar olabilmektedir. Bu durumda genellikle görüntü matrisinin dış tarafı doldurulmaktadır. Padding adı verilen bu işlemde görüntü çevresine yeni piksel değerleri eklenir. Örneğin padding değeri p, doldurma işleminin 0 değeri ile olacağı düşünülürse, bu çerçevenin p piksel kadar büyütülerek yeni piksellerin 0 ile doldurulacağı anlamına gelmektedir. Doldurma işlemi komşu piksellerin klonlanması (same-padding) ile de gerçekleştirilebilmektedir [60,61].

Ağın ilk evrişim katmanlarında kenar, köşe, çizgi gibi basit özellikler öğrenilirken daha derin evrişim katmanlarında karmaşık özellikler öğrenilmektedir. Evrişim katmanının çıktıların bulunduğu matris aktivasyon haritası olarak adlandırılır. Görüntü boyutu olan bir evrişim işlemi sonrası elde edilen aktivasyon haritasının genişliği ( $W_{out}$ ), yüksekliği ( $H_{out}$ ) ve derinliği ( $D_{out}$ );

$$\begin{aligned} W_{out} &= (W_{in} - F + 2P)/S + 1 \\ H_{out} &= (H_{in} - F + 2P)/S + 1 \\ D_{out} &= K \end{aligned} \quad (4.1)$$

Formülleri ile hesaplanmaktadır. Burada değeri boyutundaki filtrenin boyut uzunluk değerini, P padding (doldurma) değerini, S kaydırma (stride) değerini ve K filtre sayısını ifade etmektedir. Şekil 4.2’de evrişim işlemi, filtre ve aktivasyon haritası gösterilmektedir [62].



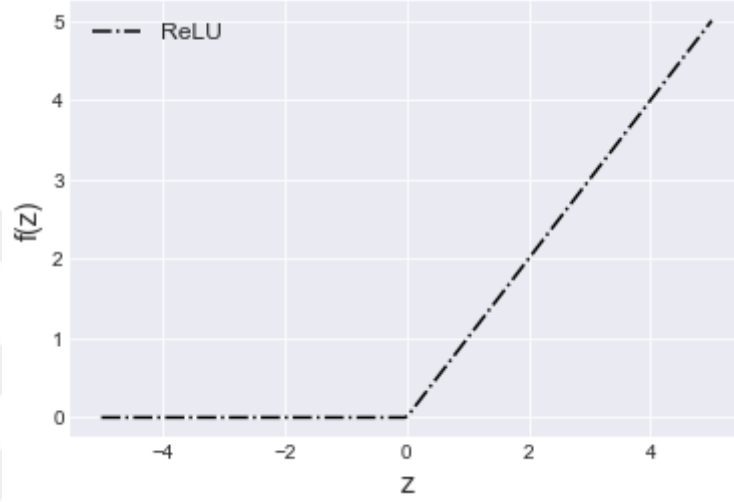
**Şekil 4.2 :** Evrişim işlemi, filtre ve aktivasyon haritası.

## 4.2 ReLU Aktivasyon Fonksiyonu

ReLU (rectified linear unit) aktivasyon fonksiyonu özellikle ESA'larda yaygın olarak kullanılmaktadır. Formülü;

$$F(z)=\max(0,z) \quad (4.2)$$

olarak tanımlanmaktadır ve grafiği Şekil 4.3'de gösterilmektedir.

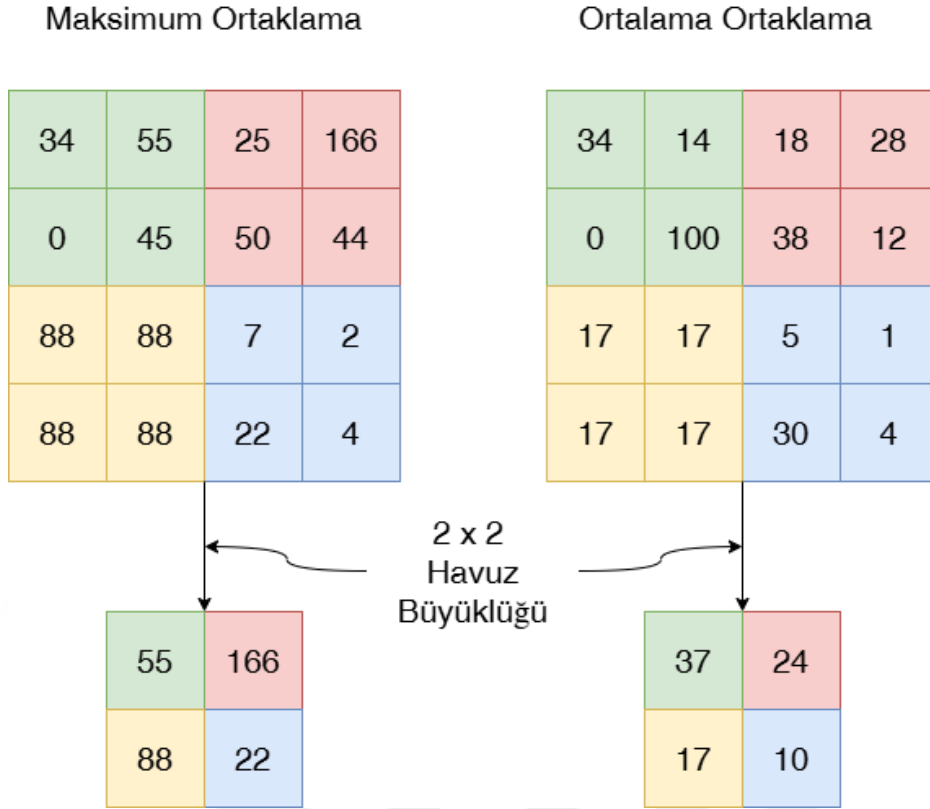


Şekil 4.3 : ReLU fonksiyonu.

Relu aktivasyon fonksiyonu ile aktivasyon haritasındaki özellikler negatif olmayan değerlerden oluşan yeni bir özellik haritasına aktarılmaktadır. Relu aktivasyon fonksiyonunun avantajı kolay türev alınabilir bir fonksiyon olmasından kaynaklanmaktadır.

## 4.3 Ortaklama Katmanı

Ortaklama katmanı genellikle aktivasyon fonksiyonu ile sonraki evrişim katmanı arasında yer almaktadır. Bu katmanda aktivasyon fonksiyonu ile elde edilen yeni özellik haritası daha küçük boyuta indirgenmektedir. Resim 4.4'de 2 x 2 havuz büyüklüğü ve 2 kaydırma (stride) değerli maksimum ortaklama ve ortalama ortaklama örneği verilmektedir [63].



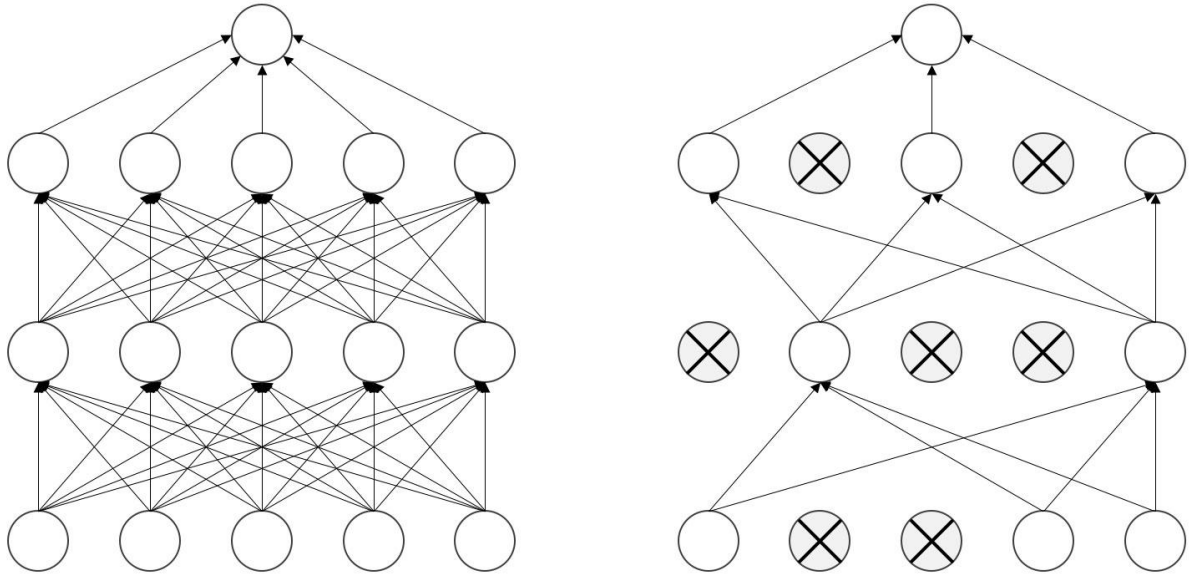
**Şekil 4.4 :** Maksimum ortaklama ve ortalama ortaklama örneği.

#### 4.4 Yığın Normalleştirme

Yığın normalleştirme (batch normalization) Szegedy ve Ioffe (2015) tarafından önerilen ve dahili ortak değişken kaymasını (internal covariate shift) azaltıp derin ağ eğitiminin hızlandırılmasını sağlayan bir tekniktir. Genellikle evrişim katmanı ile aktivasyon katmanı arasında kullanılmaktadır. Burada çıktılar küçük yığınlar (mini batch) halinde normalize edilerek hesaplama maliyeti düşürülmektedir. Böylelikle eğitim süresi kısalmaktadır. Ayrıca bu teknikle ağ daha stabil, düzenli ve kararlı hâle gelmektedir.

#### 4.5 Unutma

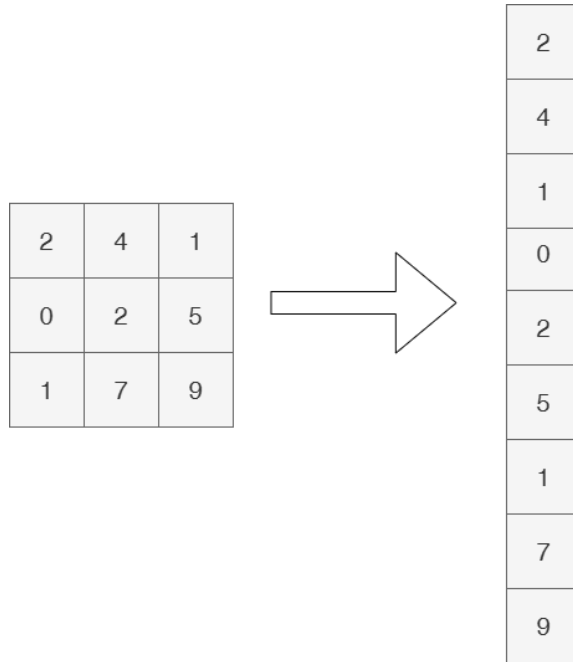
Unutma (drop-out) da yığın normalleştirme gibi sinir ağını düzenleme (regularization) metodlarından biridir. Unutma işleminde, Şekil 4.5’de görüldüğü üzere, her eğitim iterasyonunda katmanlardan belli bir oranda rastgele düğüm silinmektedir. Ağın çok büyük ve karmaşık, veri sayısının az olduğu durumlarda ezberleme riski ile karşılaşmaktadır. Unutma tekniği ile ezberleme problemine karşı model daha dirençli hâle getirilmektedir [64,65].



**Şekil 4.5 :** Standard sinir ağı ve unutmama işlemi uygulanmış sinir ağı.

#### 4.6 Düzleştirme Katmanı

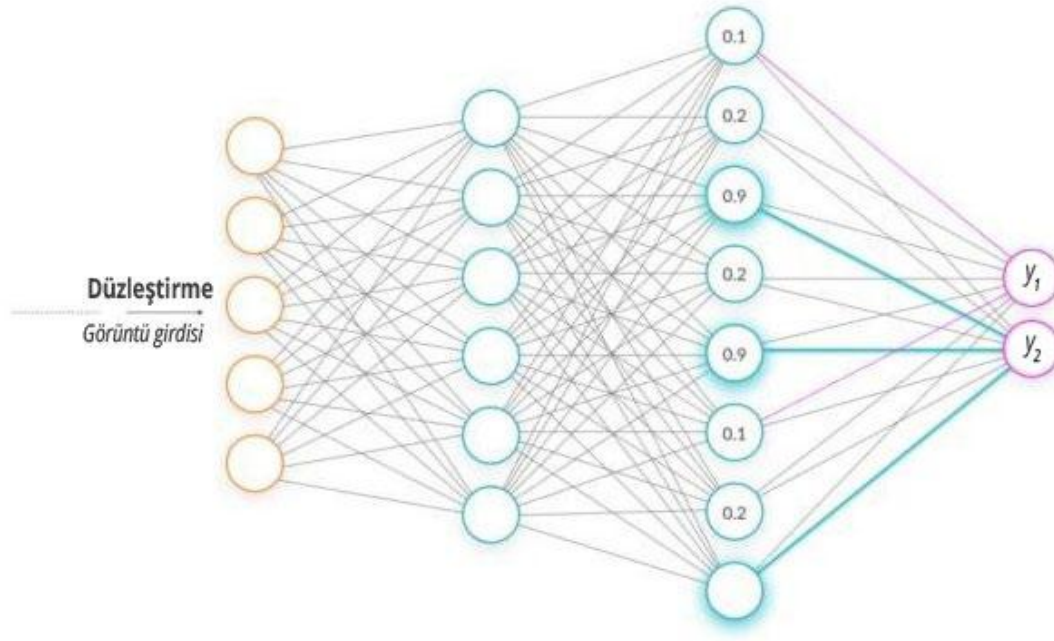
Bir ESA'da evrişim ve ortaklama işlemlerinden sonra elde edilen aktivasyon haritalarının tek boyutlu bir vektör hâline dönüştürme işleminin gerçekleştiği katmana düzleştirme katmanı adı verilmektedir. Düzleştirme işleminden sonra tek boyutlu hâle getirilen veri tam bağlı katmana gönderilir. Şekil 4.6'da düzleştirme yapılan bir matris örneği verilmiştir.



**Şekil 4.6 :** Düzleştirme işlemi örneği.

#### 4.7 Tam Bağlı Katman

Düzleştirme işleminden sonra, önceki katmanlardan tek boyutlu bir vektör hâlinde gelen bilgi artık sınıflandırma için hazır olmaktadır. Sınıflandırma işleminin yapılacağı bu katmana tam bağlı katman adı verilmektedir. Resim 4.7’de tam bağlı katman ve ikili sınıflandırma örneği verilmiştir [66].



Şekil 4.7 : İkili sınıflandırma örneğinde tam bağlı katman (Sreenivas vd. 2020).

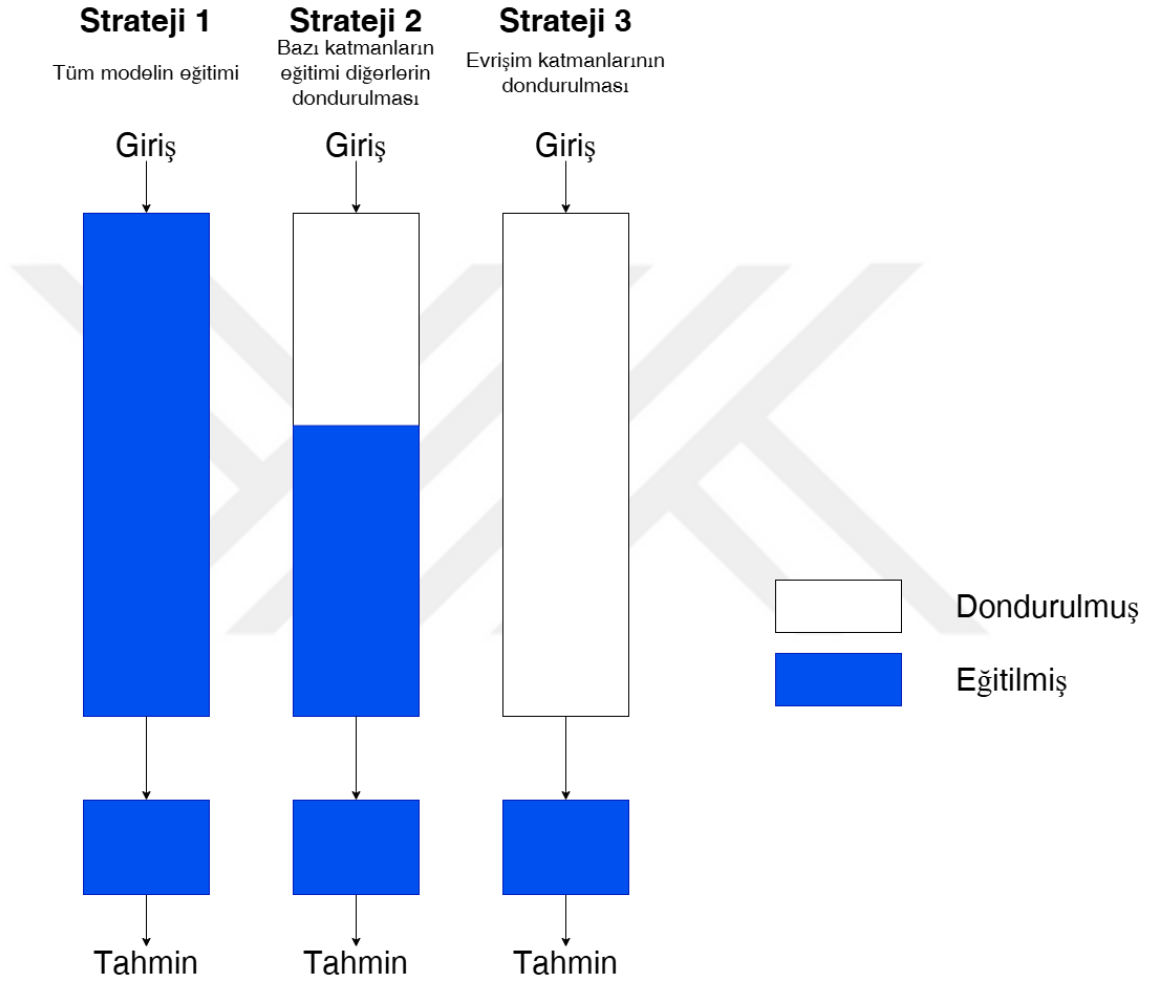
#### 4.8 Transfer Öğrenme

Transfer öğrenme ya da öğrenme aktarımı, daha önce eğitilmiş bir modelden özellik ve ağırlık gibi değerlerin başka bir modele aktarılmasıdır. Bu aktarım imkânı ile yüksek başarılı modelleri çok kısa zamanda elde etmek mümkün olmaktadır. Ayrıca büyük ölçekli verilerde daha önce eğitilmiş modeller daha sonra farklı veri setlerinde eğitildiğinde de yüksek başarı gösterebilmektedir (Rawat ve Wang 2017).

İnce ayar (fine tuning): Derin transfer öğrenmede ince ayar terimi bir evrimsel sinir ağındaki ağırlıkların yeni bir problemde de güncellenmesine izin verilmesi anlamına gelmektedir. Bir başka deyişle ince ayar, daha önce ImageNet (Deng vd. 2009) gibi bir veri setinde eğitilmiş evrimsel sinir ağının transfer edilerek parametrelerinin yeniden eğitilmesi işlemi olarak belirtilmektedir [67,68].

Görüntü sınıflandırma ve nesne tespiti gibi problemlerde 1000 sınıflı ImageNet veri setinde eğitilen ESA’lar pratikte sıklıkla kullanılmaktadır. Bu ESA’lar ImageNet’te

eğitildikten sonra ağırlıkları daha sonra kullanılmak üzere kaydedilmekte ve modellerde tekrar tekrar kullanılmaktadır. Transfer öğrenmede kullanılan modellerin eğitimi için farklı ince ayar (fine-tuning) stratejileri bulunmaktadır. Bu stratejiler, Şekil 4.8’de de görüleceği üzere evrişim bloklarındaki ağırlıkların sabitlenmesi (freezing), evrişim katmanlarından bazılarının sabit bazılarının eğitime açılması ve evrişim katmanlarının tümünün eğitilebilir seçilmesinden oluşmaktadır.



**Şekil 4.8 :** Farklı ince ayar stratejileri.

Bu tez çalışmasında en son geliştirilmiş ESA mimarilerinden Xception, InceptionV3, ResNet50V2, Xception ve DenseNet121 incelenmiştir. Zaman ve hesaplama maliyeti kısıtlarından dolayı Xception, InceptionV3 ve ResNet50V2 modelleri ImageNet’te eğitilen diğer ESA mimarilerine göre daha uygun boyutlu ve düşük parametre sayılarına sahip olmaları nedeniyle seçilmiştir. ResNet50V2 mimarisi ise kumaş hatalarının sınıflandırılmasında alınan veriler ile eğitilmesi ile tercih edilmiştir. Çizelge 4.1’de incelenen modeller, boyutları ve parametre sayıları sunulmaktadır [69].

**Çizelge 4.1 :** Kullanılan ESA modelleri, boyutları ve parametre sayıları.

| <b>Model</b>  | <b>Boyut<br/>(MB)</b> | <b>Parametre sayısı<br/>(milyon)</b> |
|---------------|-----------------------|--------------------------------------|
| Xception      | 88                    | 22,9                                 |
| ResNet50V2 B  | 98                    | 25,6                                 |
| InceptionV3 C | 92                    | 23,8                                 |
| Vgg19         | 549                   | 143,7                                |
| DenseNet121   | 33                    | 8,1                                  |

ESA ile eğitilen model üzerine kumaşlar tespit algoritmasından geçirilerek hatalı skoru yüksek olan alanlar işaretlenmiş ve skoru bu alan üzerinde işaretlenmiştir. Bu tespiti yapılan basit model ve kumaş örnekleri aşağıda gösterilmektedir. Daha sonra model Resnet50V2 ile testleri yapılacaktır.

## **4.9 Derin Öğrenme Uygulamaları**

### **4.9.1 Basit ESA model uygulaması**

Bu çalışmada, basit bir derin öğrenme modeli kullanılarak görüntü sınıflandırma görevi ele alınacaktır. Bu model, Evrişimli Sinir Ağı (ESA) mimarisine dayalı olup, daha karmaşık yapılar yerine daha temel ve geleneksel bir yapı ile eğitilecektir.

#### **4.9.1.1 Model mimarisi**

Model, daha basit bir ESA yapısı kullanılarak inşa edilmiştir. Modelin katmanları aşağıdaki gibi açıklanmaktadır:

**Giriş Katmanı (Input Layer):** Modelin ilk katmanı, 32x32x1 boyutlarında gri tonlamalı görüntüleri kabul etmektedir. Bu katman, her bir görüntüyü giriş olarak alır ve modelin ilk katmanına iletilmesini sağlamaktadır.

**İlk Konvolüsyonel Katman (Conv2D):** İlk konvolüsyonel katman, 32x32x32 boyutlarında çıkışlar üretmektedir. Bu katman, giriş görüntüsüne filtreler (kernels) uygulanarak görüntüdeki özellikleri çıkarmaktadır. Burada kullanılan 320 parametre filtrelerin sayısını vermektedir.

**İlk Batch Normalization Katmanı (BatchNormalization):** Batch normalization, modelin eğitim sürecini hızlandırmakta ve daha stabil hale getirmektedir. Bu katman, her konvolüsyon katmanının çıkışını normalize ederek, öğrenme hızını arttırmakta ve



aşırı öğrenmeyi (overfitting) engellemektedir. Bu katmanda 128 parametre bulunmaktadır.

İkinci Konvolüsyonel Katman (Conv2D\_1): İkinci konvolüsyonel katman, yine 32x32x32 boyutlarında çıkışlar üretmektedir. Bu katman da aynı şekilde giriş görüntüsündeki özellikleri çıkarmaya devam etmektedir. Bu katmanda 9,248 parametre bulunmaktadır.

İkinci Batch Normalization Katmanı (BatchNormalization\_1): İkinci batch normalization katmanı da aynı işlevi görerek modelin eğitim sürecini optimize etmektedir. Bu katmanda 128 parametre bulunmaktadır.

Max Pooling Katmanı (MaxPooling2D): Bu katman, görüntüdeki önemli özellikleri sıkıştırarak modelin daha hızlı öğrenmesini sağlamaktadır. Max pooling katmanı, 32x32 boyutundaki görüntüyü 16x16 boyutuna indirgemektedir. Bu katman, 2x2 pencerelerle en yüksek değerleri seçerek boyut küçültme işlemi yapmaktadır.

Üçüncü Konvolüsyonel Katman (Conv2D\_2): Üçüncü konvolüsyonel katman, daha fazla özellik çıkarmayı yapmak için 16x16x64 boyutlarında bir çıkış üretmektedir. Bu katmanda 18,496 parametre bulunmaktadır.

Üçüncü Batch Normalization Katmanı (BatchNormalization\_2): Üçüncü batch normalization katmanı, daha stabil bir eğitim süreci için çıktıyı normalize etmektedir. Bu katmanda 256 parametre bulunmaktadır.

Dördüncü Konvolüsyonel Katman (Conv2D\_3): Dördüncü konvolüsyonel katman da 16x16x64 boyutlarında çıkışlar üretmektedir. Bu katman, görüntüdeki daha derin özellikleri çıkartmaktadır. Bu katmanda 36,928 parametre bulunmaktadır.

Dördüncü Batch Normalization Katmanı (BatchNormalization\_3): Bu katman, üçüncü batch normalization katmanı gibi eğitim sürecini hızlandırmakta ve daha stabil hale getirmektedir. Bu katmanda 256 parametre bulunmaktadır.

İkinci Max Pooling Katmanı (MaxPooling2D\_1): Bu katman, ikinci bir max pooling işlemi uygulayarak 16x16 boyutundaki görüntüyü 8x8 boyutuna indirgemektedir. Bu işlem, özelliklerin daha kompakt bir hale gelmesini sağlamaktadır.

Flatten Katmanı (Flatten): Flatten katmanı, 3D boyutlarında olan veriyi tek bir vektöre dönüştürmektedir. Bu vektör, 4096 elemandan oluşur ve modelin sonraki katmanlarına aktarılmaktadır.

Dense Katmanı (Dense): Son olarak, modelin çıkış katmanı olan Dense katmanı kullanılmaktadır. Bu katman, 2 sınıf (binary classification) için olasılık tahminleri üretmektedir. Son katmanda 8,194 parametre bulunmaktadır. Aktivasyon fonksiyonu olarak softmax kullanılmakta ve her sınıf için bir olasılık değeri sağlamaktadır [70,71].

Katmanları açıklanan bu basit model katmanları Şekil 4.9'da verilmiştir. Bu modelin toplam parametresi 73.954, eğitilebilir parametresi 73.570 ve eğitilemez parametresi 384'dir. ESA modeli pseudo kod olarak algoritma 5'de anlatılmıştır.

---

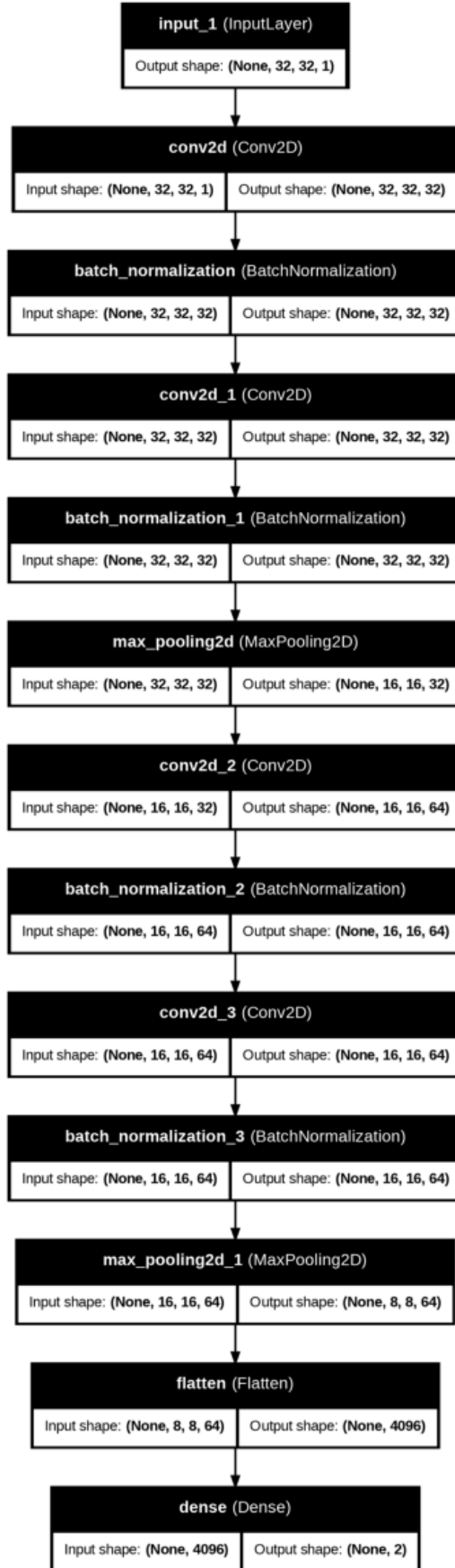
#### Algoritma 5: ESA Algoritması

---

**Require:** Her türde ve sınıfla farklı dizine bölünmüş görseller bulunmalıdır.

```
names and image (n × n) ≠ nonempty.  
#Verinin Hazırlanması  
val_ds = tf.keras.preprocessing.image_dataset_from_directory(parameters)  
class_names = train_ds.class_names  
#Model Eğitimi  
model = Sequential()  
model.add(Conv2D), model.add(MaxPooling2D)  
model.add(Conv2D), model.add(MaxPooling2D)  
model.add(Conv2D), model.add(MaxPooling2D)  
model.add(Dropout(Parameter))  
model.add(Flatten())  
model.add(Dense(512, activation = 'relu'))  
model.add(Dropout(Parameter))  
model.add(Dense(5, activation = 'softmax'))  
model.compile()  
# Model Hata Tanıması  
pred = model.predict(image)  
print(pred)
```

---



Şekil 4.9 : Model: "model\_basic"

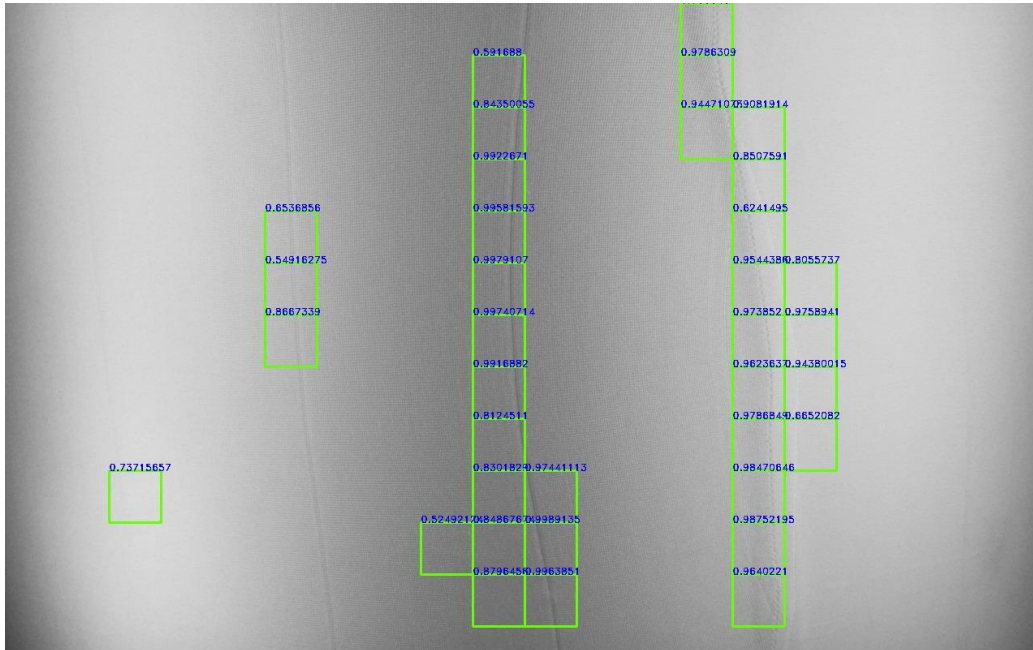
#### 4.9.1.2 Modelin değerlendirilmesi ve tahmin yapılması

Model eğitildikten sonra, test verisi üzerinden tahminler yapılmıştır. Bu aşamada, belirli bir sınıf (örneğin, KIRIKIGNE) içinden rastgele seçilen bir görüntü modele verilmiş ve model bu görüntü için bir tahmin yapmıştır. Modelin tahmin ettiği sınıf, en yüksek olasılığa sahip olan sınıf olarak belirlenmiştir. Çizelge 4.2’de karmaşıklık matrisi verilmiştir ve doğruluk oranı %93 olarak elde edilmiştir.

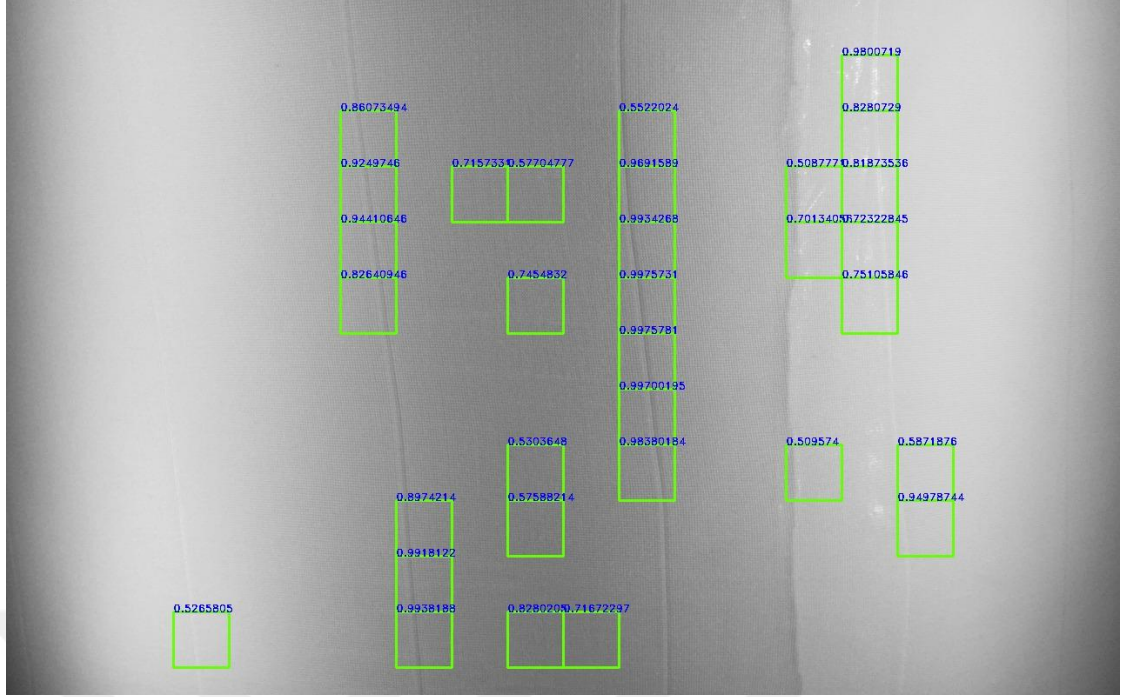
**Çizelge 4.2 : ESA karmaşıklık matrisi.**

|  |          | Tahmin     |           |        |
|--|----------|------------|-----------|--------|
|  |          | Kusursuz   | Kusurlu   | Toplam |
|  | Gerçek   |            |           |        |
|  | Kusursuz | 145<br>%65 | 7<br>%3   | 152    |
|  | Kusurlu  | 9<br>%4    | 65<br>%28 | 74     |
|  | Toplam   | 154        | 72        | 226    |

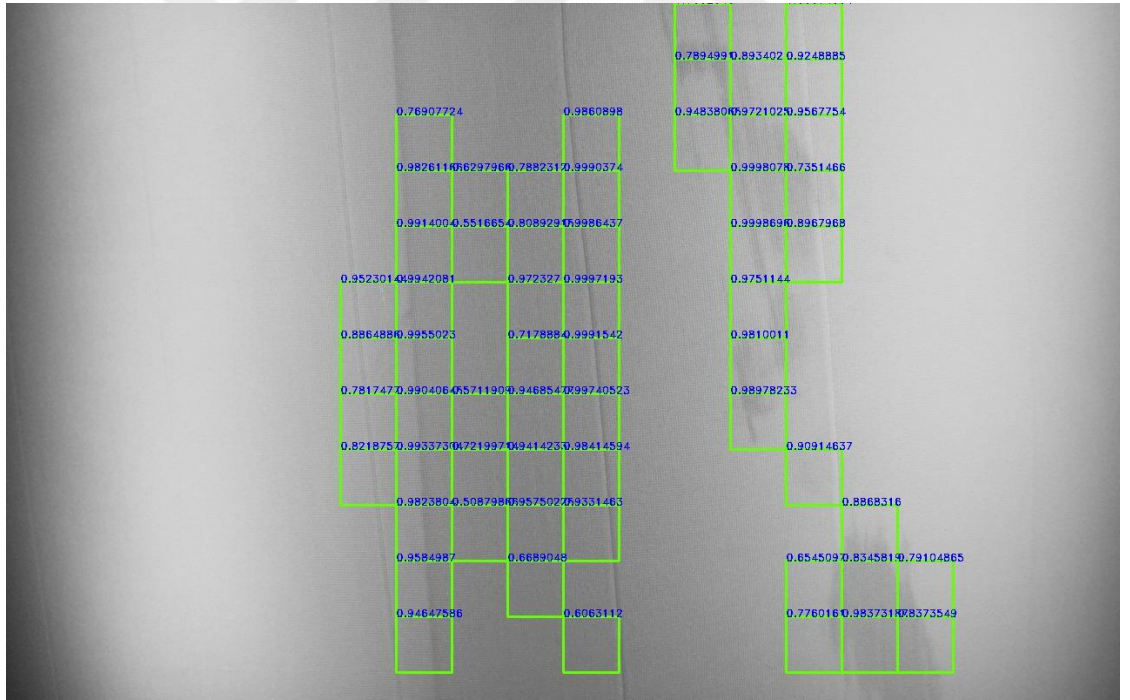
Tahmin sonuçları, modelin doğruluğunu test etmek için gerçek sınıflar ile karşılaştırılmıştır. Ayrıca, eğitim sürecindeki doğruluk ve kayıp grafikleri, modelin performansını görselleştirerek hangi epoch’ta daha iyi sonuçlar elde edildiğini göstermiştir. Bu model çalışması sonucu hatalı tespit edilmiş bölgeler işaretli resimler Şekil 4.10, Şekil 4.11, Şekil 4.12, Şekil 4.13, Şekil 4.14 ve Şekil 4.15’de verilmektedir.



**Şekil 4.10 : ESA ile hata tespiti (1).**

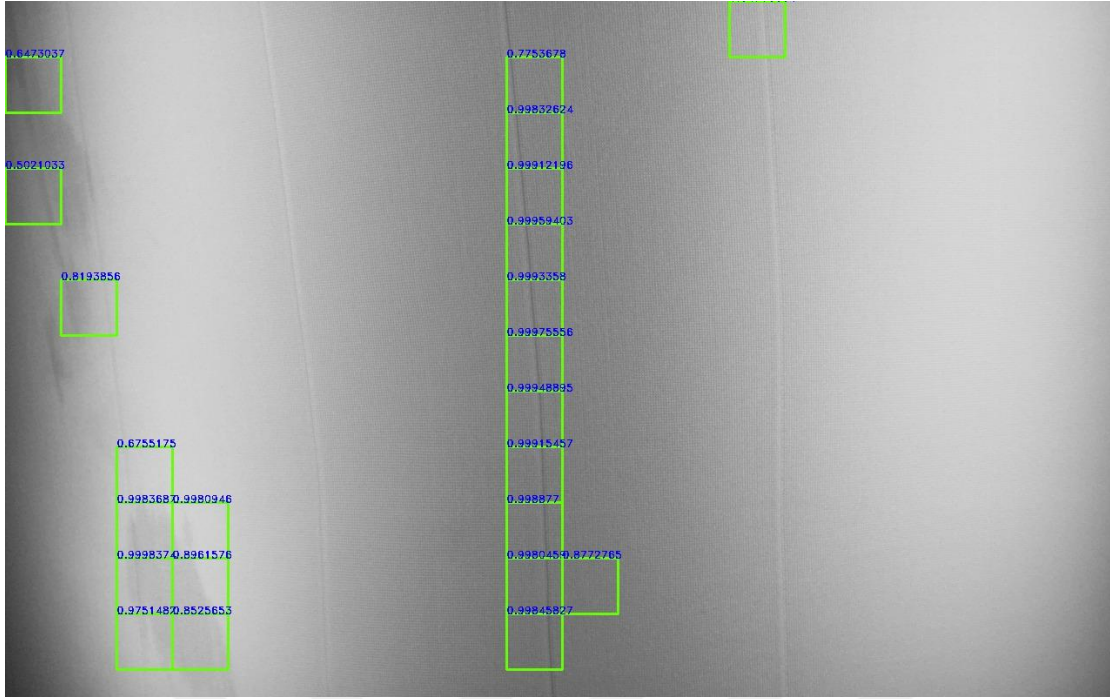


Şekil 4.11 : ESA ile hata tespiti (2).

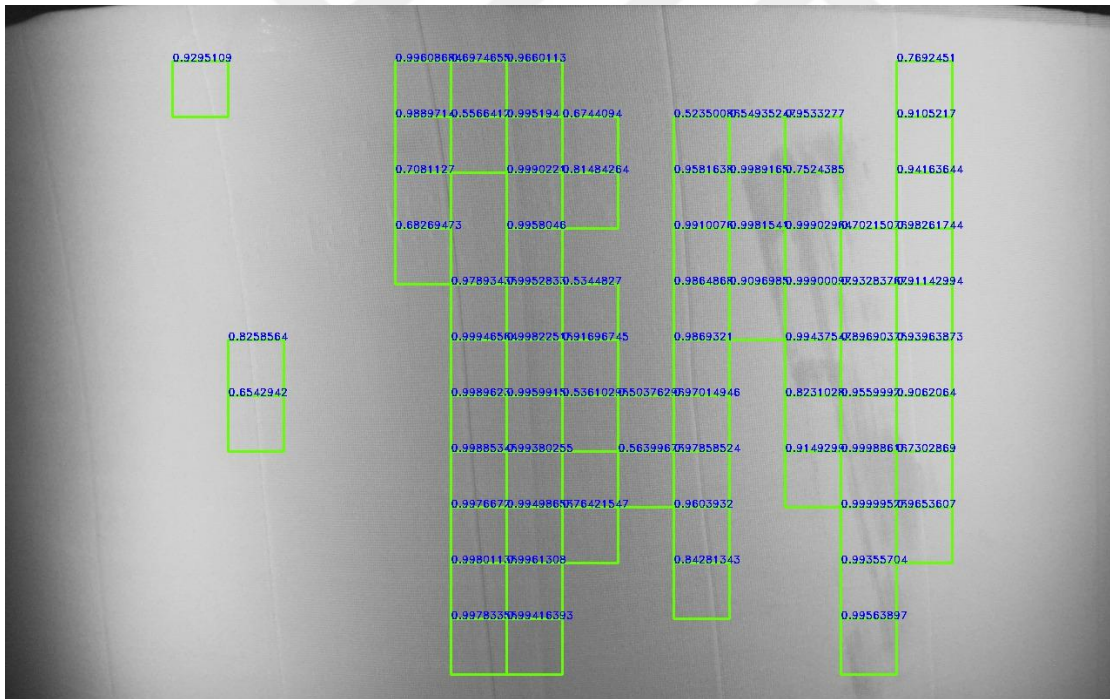


Şekil 4.12 : ESA ile hata tespiti (3).

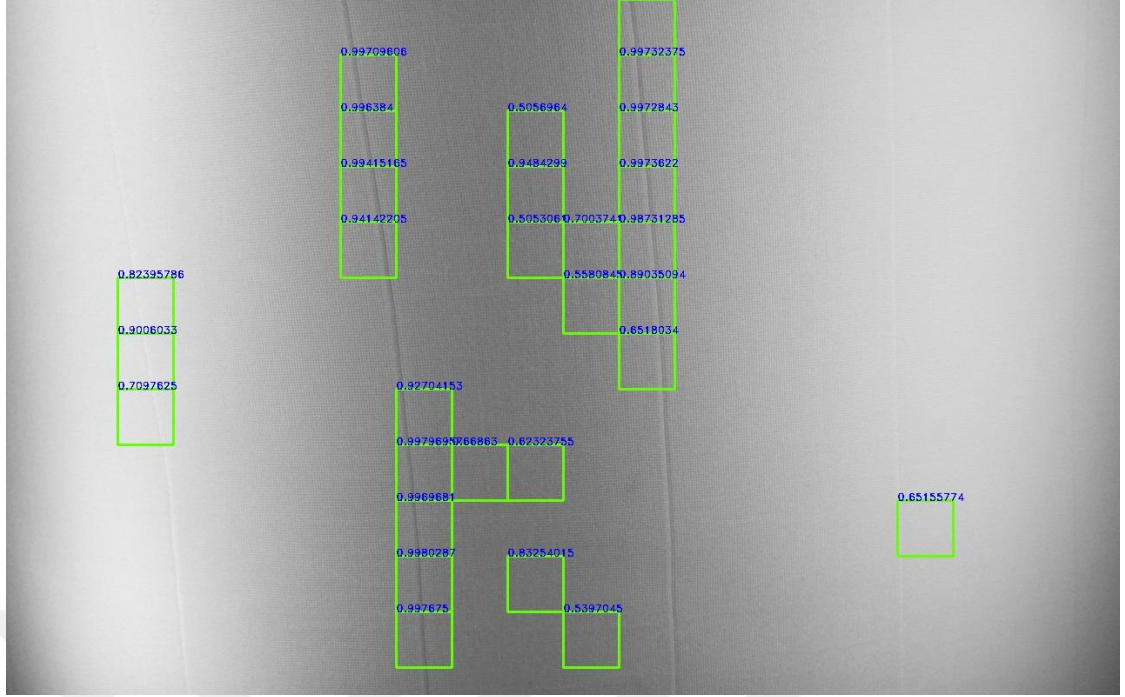




Şekil 4.13 : ESA ile hata tespiti (4).



Şekil 4.14 : ESA ile hata tespiti (5).



**Şekil 4.15 : ESA ile hata tespiti (7).**

#### **4.9.2 RestNet50 model uygulaması**

Bu bölümde, görüntü sınıflandırma görevinde kullanılacak olan derin öğrenme modelinin mimarisi, veri hazırlık süreci, eğitim aşamaları ve modelin performans değerlendirilmesi ayrıntılı olarak ele alınmıştır. Çalışmanın temelinde, ResNet50 mimarisi kullanılmaktadır. Bu model, transfer öğrenme (transfer learning) tekniği ile eğitilmekte, yani ResNet50 modeli, önceden ImageNet veri kümesi üzerinde eğitilmiş ve daha sonra belirli bir görev için özelleştirilmiş bir model olarak kullanılmaktadır. Modelin eğitim süreci, doğruluk ve kayıp gibi metrikler aracılığıyla değerlendirilecektir. Ayrıca, modelin tahmin yetenekleri ve genel başarısı test edilecektir [72,73].

##### **4.9.2.1 Veri hazırlığı**

Modelin doğru bir şekilde eğitilebilmesi için verilerin uygun şekilde hazırlanması gerekmektedir. Bu aşama, veri kümesinin yüklenmesi, görsellerin ön işlenmesi ve eğitim ile doğrulama kümesine ayrılması gibi adımları içermektedir. Veri kümesi, her biri farklı bir sınıfı temsil eden alt dizinlere ayrılmış olan görüntülerden oluşmaktadır. Verilerin uygun şekilde yüklenmesi için TensorFlow ve Keras kütüphanelerinin sunduğu `image_dataset_from_directory` fonksiyonu kullanılmaktadır. Bu fonksiyon,

dizindeki her alt klasörü bir sınıf olarak kabul etmekte ve her sınıf için görüntülerin bir veri kümesini oluşturmaktadır.

Eğitim ve doğrulama kümeleri, toplam veri kümesinin %60'ı ve %40'ı oranında ayrılmaktadır. Eğitim verisi, modelin öğrenme sürecini gerçekleştirdiği veri kümesi iken, doğrulama verisi modelin eğitim süreci sırasında eğitimin ne kadar genelleştirilebilir olduğunu kontrol etmek için kullanılmaktadır.

Veri kümesindeki her görüntü, modelin giriş boyutuna uygun hale gelmesi için yeniden boyutlandırılmaktadır. Bu çalışmada, her bir görüntü 180x180 piksel boyutlarına getirilmektedir. Görüntülerin boyutları yeniden ayarlandığında, aynı zamanda veri kümesinin batch\_size parametresi ile mini-batch'ler halinde işlenmesi gerekmektedir. Bu, verilerin küçük gruplar halinde modele beslenmesini ve modelin parametrelerinin adım adım güncellenmesini sağlamaktadır [74].

#### 4.9.2.2 Model mimarisi

Modelin temeli olarak ResNet50 (Residual Network) kullanılmaktadır. ResNet50, derin öğrenme alanında yaygın olarak kullanılan ve 50 katmanlı bir konvolüsyonel sinir ağı (CNN) modelidir. Bu model, residual connection adı verilen bir mekanizmayı kullanmaktadır. Bu, ağı daha derin katmanlarında meydana gelen kayıpların (vanishing gradients) üstesinden gelmeye yardımcı olmaktadır. Yani, katmanlar arasında doğrudan bağlantılar (skip connections) kullanılarak modelin daha derin katmanlarında öğrenme yapılması sağlanmaktadır [75].

ResNet50 modelinin, ImageNet gibi büyük veri kümeleri üzerinde önceden eğitilmiş ağırlıklarla yüklenmesi, transfer öğrenme (transfer learning) tekniğinin bir örneğidir. Bu, modelin genelleme yeteneğini artırır ve eğitim sürecinde büyük bir hız kazancı sağlamaktadır. ResNet50 modelinin en üst katmanları (include\_top=False) kaldırılarak sadece önceden öğrenilmiş özellik çıkarıcı katmanları kullanılmaktadır. Böylece, modelin daha genel özellikleri öğrenmesi sağlanmakta; kendi sınıflandırma katmanları sonradan eklenerek özelleştirilmektedir [76]. ResNet50 katmanları Şekil 4.16'da verilmektedir.

| Layer (type)              | Output Shape        | Param # |
|---------------------------|---------------------|---------|
| input_layer (InputLayer)  | (None, 180, 180, 3) | 0       |
| conv1_pad (ZeroPadding2D) | (None, 186, 186, 3) | 0       |

**Şekil 4.16:** ResNet50 modeli



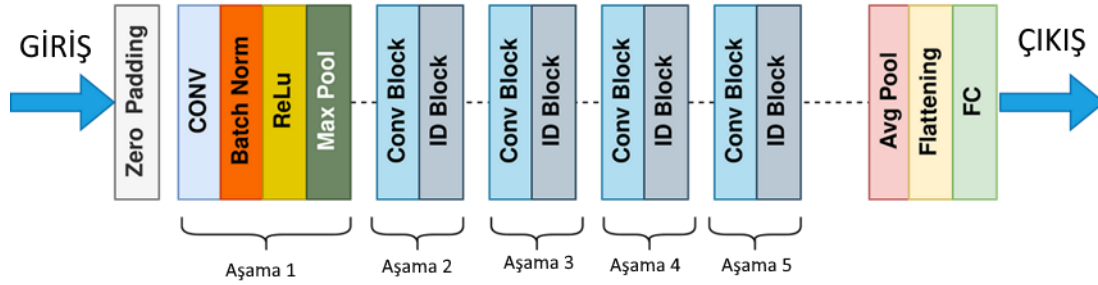
|                                             |                     |           |
|---------------------------------------------|---------------------|-----------|
| conv1_conv (Conv2D)                         | (None, 90, 90, 64)  | 9,472     |
| conv1_bn (BatchNormalization)               | (None, 90, 90, 64)  | 256       |
| conv1_relu (Activation)                     | (None, 90, 90, 64)  | 0         |
| pool1_pad (ZeroPadding2D)                   | (None, 92, 92, 64)  | 0         |
| pool1_pool (MaxPooling2D)                   | (None, 45, 45, 64)  | 0         |
| conv2_block1_1_conv (Conv2D)                | (None, 45, 45, 64)  | 4,160     |
| conv2_block1_1_bn (BatchNormalization)      | (None, 45, 45, 64)  | 256       |
| conv2_block1_1_relu (Activation)            | (None, 45, 45, 64)  | 0         |
| conv2_block1_2_conv (Conv2D)                | (None, 45, 45, 64)  | 36,928    |
| conv2_block1_2_bn (BatchNormalization)      | (None, 45, 45, 64)  | 256       |
| conv2_block1_2_relu (Activation)            | (None, 45, 45, 64)  | 0         |
| conv2_block1_0_conv (Conv2D)                | (None, 45, 45, 256) | 16,640    |
| conv2_block1_3_conv (Conv2D)                | (None, 45, 45, 256) | 16,640    |
| .                                           |                     |           |
| .                                           |                     |           |
| .                                           |                     |           |
| conv5_block3_add (Add)                      | (None, 6, 6, 2048)  | 0         |
| conv5_block3_out (Activation)               | (None, 6, 6, 2048)  | 0         |
| avg_pool (GlobalAveragePooling2D)           | (None, 2048)        | 0         |
| dense (Dense)                               | (None, 512)         | 1,049,088 |
| dense_1 (Dense)                             | (None, 5)           | 2,565     |
| =====                                       |                     |           |
| Total params: 24,639,365 (93.99 MB)         |                     |           |
| Trainable params: 1,051,653 (4.01 MB)       |                     |           |
| Non-trainable params: 23,587,712 (89.98 MB) |                     |           |

**Şekil 4.16 : ResNet50 modeli. (devamı)**

Modelin son katmanlarında, ResNet50'nin çıkış katmanına bir Dense katmanı eklenmektedir. Bu katman, 512 nöronla donatılır ve ReLU aktivasyon fonksiyonu ile aktivasyon yapılmaktadır. Bu, modelin daha karmaşık özellikler öğrenmesini sağlamaktadır. Çıkış katmanında ise, modelin tahmin yapabilmesi için sınıflandırma yapılacak sınıfların sayısına bağlı olarak 5 sınıf için softmax aktivasyon fonksiyonu kullanılmaktadır. Softmax, çok sınıflı sınıflandırma problemlerinde yaygın olarak kullanılan bir aktivasyon fonksiyonudur ve her sınıf için bir olasılık değeri üretmektedir [77].

Ayrıca, ResNet50'nin önceden eğitilmiş ağırlıklarını kullanarak modelin transfer öğrenme aşamasında, bu ağırlıkların dondurulması sağlanabilmektedir. Yani, bu katmanların ağırlıkları eğitim sırasında güncellenmemektedir. Bu, modelin daha hızlı

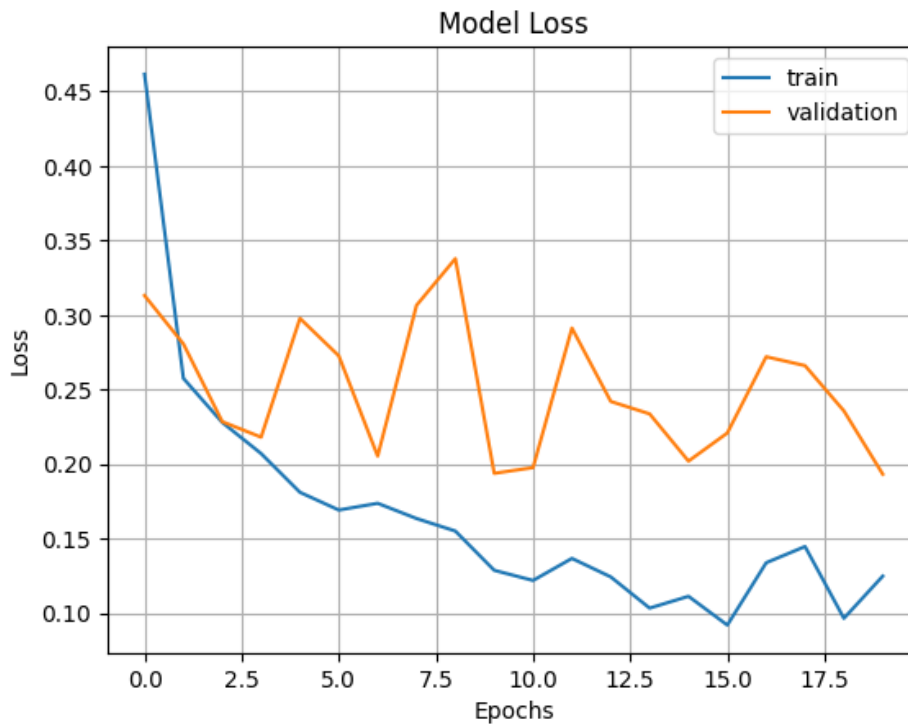
öğrenmesini ve daha az veriye ihtiyaç duymasını sağlamaktadır. RestNet50 mimarisi Şekil 4.17’de verilmektedir.



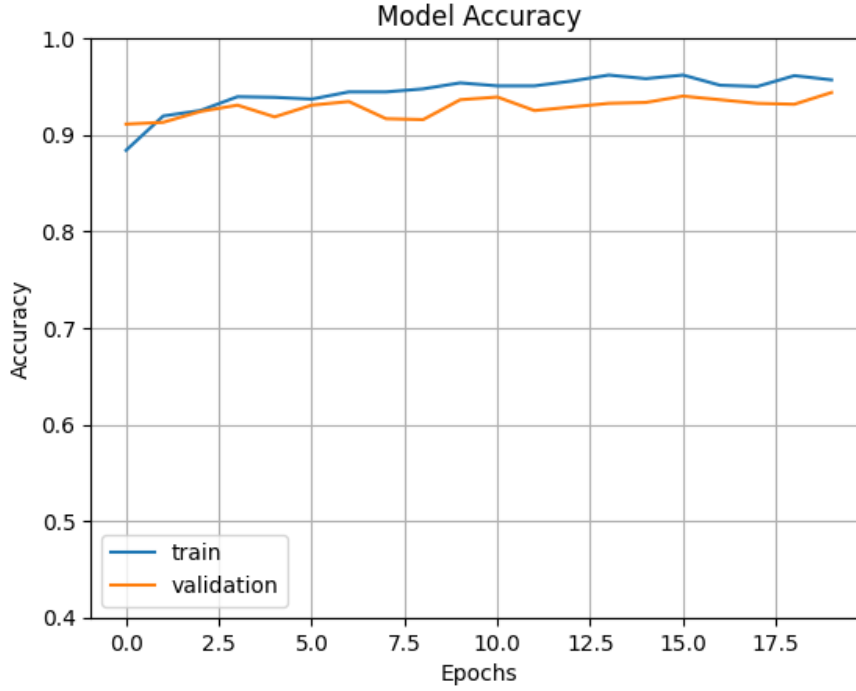
Şekil 4.17 : RestNet50 mimarisi.

#### 4.9.2.3 Modelin eğitimi

Modelin eğitim sürecinde, Adam optimizer kullanılmaktadır. Adam, öğrenme oranını her parametre için otomatik olarak ayarlayan bir optimizasyon algoritmasıdır ve genellikle derin öğrenme problemlerinde yaygın olarak tercih edilmektedir. Kayıp fonksiyonu olarak ise sparse categorical crossentropy kullanılmaktadır. Bu kayıp fonksiyonu, çok sınıflı sınıflandırma problemleri için uygun bir tercihtir, çünkü modelin tahmin ettiği sınıfın doğru sınıfla ne kadar örtüştüğünü ölçmektedir [78,79].



Şekil 4.18 : ResNet50 model kayıp grafiği.



**Şekil 4.19 :** RestNet50 model doğruluğu.

Modelin eğitimi 20 epoch boyunca yapılmaktadır. Bu süreçte, eğitim verileri modelin ağırlıklarını optimize ederken, doğrulama verileri modelin performansını değerlendirmek amacıyla kullanılmaktadır. Eğitim sırasında, modelin doğruluk (accuracy) ve kayıp (loss) metrikleri izlenmektedir. Bu metrikler, modelin eğitim sürecindeki başarısını görselleştirmekte ve modelin hangi epoch'ta daha iyi performans gösterdiğini anlamaya yardımcı olmaktadır. Eğitim sırasında, modelin doğruluk (accuracy) ve kayıp (loss) metrikleri değişimi Şekil 4.18 ve Şekil 4.19'de verilmektedir [79].

#### 4.9.2.4 Modelin değerlendirilmesi ve tahmin yapılması

Modelin eğitilme süreci tamamlandıktan sonra, test verileri üzerinde tahminler gerçekleştirilmektedir. Bu aşamada, belirli bir kusur sınıfına (örneğin, KIRIKIĞNE) ait rastgele seçilen bir görüntü modelin girişine verilerek tahmin süreci başlatılmaktadır. Model, bu görüntüye yönelik bir sınıflandırma işlemi gerçekleştirmekte ve en yüksek olasılığa sahip sınıfı tahmin olarak belirlemektedir. Tahmin süreci, ResNet-50 modelinin son çıkış katmanında yer alan softmax aktivasyon fonksiyonu aracılığıyla her bir sınıf için olasılık değerlerinin hesaplanmasıyla gerçekleştirilmektedir. Model, en yüksek olasılık değerine sahip

sınıfı tahmin olarak döndürerek giriş görüntüsünün ilgili kategoriye ait olduğunu belirlemektedir.

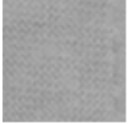
Modelin tahmin doğruluğu, gerçek etiketler ile karşılaştırılarak değerlendirilmektedir. Modelin sınıflandırma performansı, sınıflandırma doğruluğu, hata oranları ve diğer değerlendirme metrikleri kullanılarak ayrıntılı bir şekilde analiz edilmektedir [80].

#### 4.9.2.5 Sonuç ve ileri adımlar

Bu bölümde, görüntü sınıflandırma görevinde kullanılan bir derin öğrenme modelinin geliştirilmesi ve eğitilmesi süreci ayrıntılı olarak anlatılmıştır. ResNet50 modeli, transfer öğrenme yaklaşımı ile eğitilmiş ve modelin başarı oranı değerlendirilmiştir. Eğitim ve doğrulama süreçlerinde elde edilen doğruluk ve kayıp grafiklerinin görselleştirilmesi, modelin performansını değerlendirmemize olanak sağlamaktadır.

Hatasız sınıflandırılmış bir görüntünün testi ve sonucu Şekil 4.20’de verilmektedir.

```
7
5205_8.png
(1, 180, 180, 3)
1/1 ————— 1s 1s/step
[[3.0890510e-03 9.8514080e-01 2.0317656e-04 1.1555860e-02 1.1068521e-05]]
The predicted class is HATASIZ
```



**Şekil 4.20 :** RestNet50 ile hata sınıflandırma.

KIRIKIGNE sınıflandırılmış bir görüntünün testi ve sonucu Şekil 4.21’de verilmektedir.

```
37
2998_2.png
(1, 180, 180, 3)
1/1 ————— 0s 340ms/step
[[2.4482263e-06 8.4110281e-05 9.9989235e-01 2.1142954e-05 2.2094864e-11]]
The predicted class is KIRIKIGNE
```



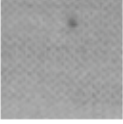
**Şekil 4.21 :** RestNet50 ile hata sınıflandırma.

DELIK sınıflandırılmış bir görüntünün testi ve sonucu Şekil 4.22’de verilmektedir.

```

120
2994_4.png
(1, 180, 180, 3)
1/1 _____ 0s 203ms/step
[[9.9980670e-01 4.4435041e-05 1.4880873e-04 9.7193329e-09 3.8280173e-12]]
The predicted class is DELIK

```



**Şekil 4.22 :** RestNet50 ile hata sınıflandırma.

Gelecek çalışmalarda, modelin doğruluğunu daha da artırmak için veri artırma (data augmentation) yöntemleri uygulaması düşünülmektedir. Bu, modelin daha fazla çeşitlilikle eğitilmesini sağlayarak, aşırı öğrenme (overfitting) riskini azaltacaktır. Ayrıca, modelin ince ayar (fine-tuning) yapılması, yani ResNet50'nin daha fazla katmanının eğitim sürecine dahil edilmesi, modelin daha iyi genelleme yapabilmesi için etkili bir yöntemdir. Bu tür iyileştirmeler, modelin daha karmaşık ve çeşitli veri kümeleri üzerinde başarılı performans göstermesini sağlayacaktır [81]. Çizelge 4.3'de karmaşıklık matrisi verilmiştir ve doğruluk oranı %96 hesaplanmıştır.

**Çizelge 4.3 :** Restnet50 karmaşıklık matrisi.

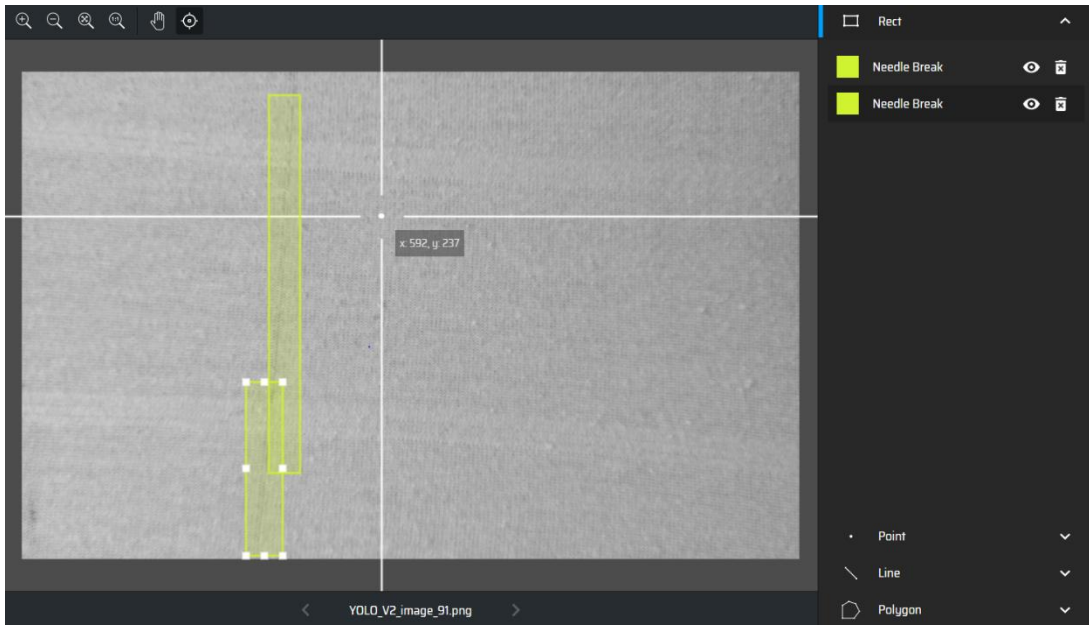
|        |          | Tahmin     |           |        |
|--------|----------|------------|-----------|--------|
|        |          | Kusursuz   | Kusurlu   | Toplam |
| Gerçek | Kusursuz | 148<br>%65 | 4<br>%2   | 152    |
|        | Kusurlu  | 3<br>%2    | 71<br>%31 | 74     |
|        | Toplam   | 151        | 75        | 226    |

#### 4.9.3 YoloV5 uygulaması

(You Only Look Once) algoritması, nesne tespiti için kullanılan, regresyon tabanlı bir algoritmadır. Algoritma, görüntünün bir bölümünü seçmek yerine, tüm görüntü üzerinde tek bir çalıştırmada sınıf ve sınırlayıcı kutu tahmini yapmaktadır. Bu yönüyle, R-CNN gibi bölge bazlı algoritmalarından ayrılmaktadır. R-CNN algoritmaları, önce nesnelerin bulunabileceği bölgeleri belirlemekte, ardından bu bölgelerde ayrı ayrı Evrişimsel Sinir Ağları (CNN) kullanarak sınıflandırma işlemi gerçekleştirmektedir. Ancak, bu yöntem resmin iki ayrı işlemden geçmesine neden olmakta ve düşük FPS (Frame Per Second) oranlarıyla sonuçlanarak gerçek zamanlı uygulamalar için yetersiz kalmaktadır [82].

#### 4.9.3.1 Veri setini hazırlama

YOLOv5 ile model eğitimi için veri seti oluşturmak kritik bir adımdır. Bu süreçte kullanılacak görüntüler üzerinde hataların işaretlenmesi gerekmektedir. Bunun için LabelImg veya Makesense.ai gibi araçlar kullanılarak, görüntü üzerindeki hatalı bölgeler seçilmekte ve hata tiplerine göre etiketlenmektedir. Bu işlem sayesinde iğne kırığı, delik, likra kaçağı, may izi ve yağ izi gibi farklı hata türleri tanımlanabilmektedir. Örnek bir hata etiketleme işlemi Şekil 4.23’de gösterilmiştir [83].



Şekil 4.23 : YOLOv5 ile hata etiketleme.

Etiketleme tamamlandıktan sonra veri seti, eğitim (%70) ve test (%30) olarak iki gruba ayrılmaktadır. Her grup için ayrı klasörler oluşturulmakta ve bu klasörler içinde de images (görüntüler) ve labels (etiketler) adında alt klasörler eklenmektedir. Görüntüler images klasörüne, etiket dosyaları ise labels klasörüne kaydedilmektedir.

#### 4.9.3.2 Ortam kurulumu

Modelin eğitileceği ortam, ihtiyaca göre bir bilgisayar, bulut sunucusu veya geliştirme kartları üzerinde hazırlanabilmektedir. Ortamın kolay yönetimi ve taşınabilirliği için Docker imajları kullanılabilir. YOLOv5’in eğitim ve test aşamalarında GPU kullanımını etkinleştirmek için PyTorch ve OpenCV kütüphanelerinin GPU sürümleri kurularak test edilmektedir. Eğitim sırasında sınıf türleri, data.yaml dosyasında belirtilmektedir. Bu sınıflar, etiketleme sırasında kullanılan sırayla tanımlanmaktadır.

#### 4.9.3.3 Modelin eğitimi

Görüntü işleme modelini eğitmek için hazırlanan eğitim verisi kullanılmaktadır. Görüntülerin boyutları, renk kanalları ve formatı uygun şekilde ayarlanması gerekmektedir. Ayrıca görüntülerdeki nesnelerin konum ve tür bilgilerini içeren etiketleme işlemi doğru bir şekilde hazırlanmalıdır.

Model eğitimi için temel parametrelerin (örneğin, görüntü boyutu, batch size, epoch sayısı, kullanılan modelin önceden eğitilmiş ağırlık dosyası yolu gibi) ayarlanması gerekmektedir. Parametrelerin tanımlanmasının ardından eğitim işlemi başlatılmaktadır. Bu süreçte model, eğitim verisini kullanarak kendini optimize etmektedir. Modelin performansı doğruluk oranı (accuracy), IoU (Intersection over Union) gibi metriklerle değerlendirilmektedir. Eğitim tamamlandığında, model dosyası gerçek zamanlı ya da offline işlemlerde kullanılabilir hale gelmektedir [84].

#### 4.9.3.4 Eğitilmiş model ile kusur tanıma

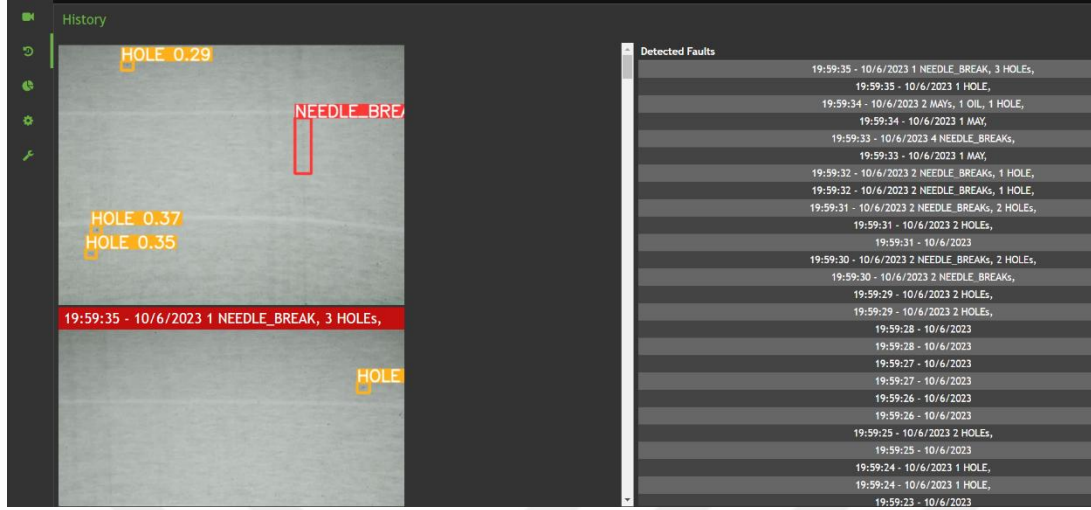
Kumaş hatalarını tespit etmek için YOLOv5 ile birlikte gelen detect.py dosyası güncellenerek kullanılabilir. Bu dosya, gerçek zamanlı kamera görüntüleri, videolar, resimler, online video bağlantıları ve farklı akış protokolleri gibi çeşitli giriş türlerini desteklemektedir. Çizelge 4.4’de karmaşıklık matrisi verilmiştir ve doğruluk oranı %95 hesaplanmıştır.

**Çizelge 4.4 : YOLOv5 karmaşıklık matrisi.**

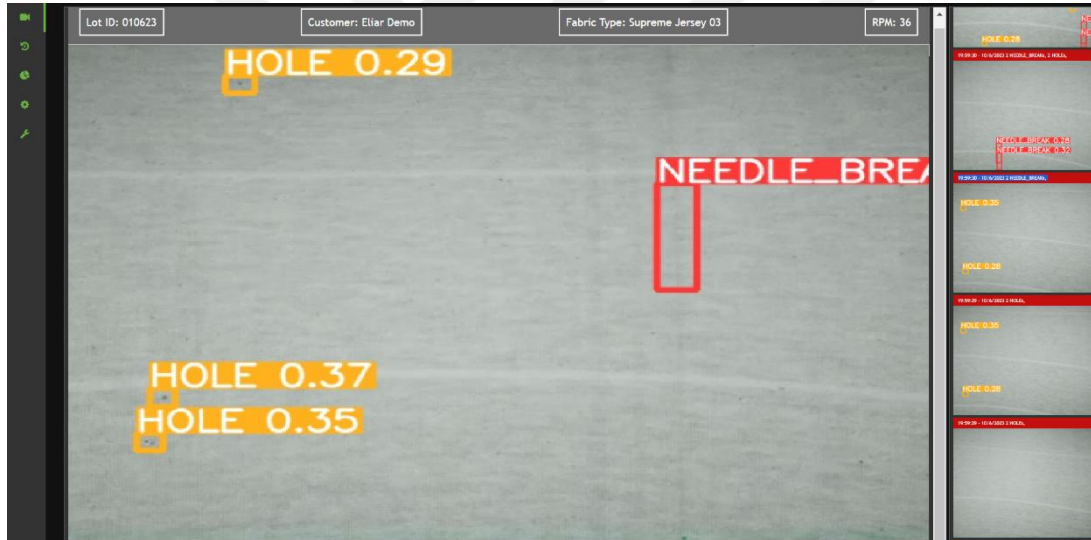
|        |          | Tahmin     |           |        |
|--------|----------|------------|-----------|--------|
|        |          | Kusursuz   | Kusurlu   | Toplam |
| Gerçek | Kusursuz | 146<br>%65 | 6<br>%3   | 152    |
|        | Kusurlu  | 5<br>%2    | 69<br>%30 | 74     |
|        | Toplam   | 151        | 75        | 226    |

GPU kullanılarak yapılan tespit işlemi, donanım özelliklerine bağlı olarak yüksek performanslı ve gerçek zamanlı sonuçlar sunmaktadır. Örneğin, tespit edilen hataların konumu, türü ve zamanı gibi bilgiler anlık olarak ekranda gösterilmekte ve bir veri tabanına kaydedilmektedir. Şekil 4.24, bu işlemin bir örneğini göstermektedir. Şekil 4.25’de veri tabanındaki kayıtlar kullanılarak geçmişteki hataların raporlanması ve

analizi yapılmaktadır. Bu analizler, kumaş üretim süreçlerinde kalite kontrolü artırmak için kullanılmaktadır.



Şekil 4.24 : YOLOv5 ile hata tespiti.



Şekil 4.25 : YOLOv5 ile hata sonuçları ve geçmiş hata raporu.

#### 4.9.4 Otoenkoder anomali tespiti uygulaması

Anomali tespiti, son yıllarda derin öğrenme alanında yoğun olarak araştırılan konular arasında yer almaktadır. Görüntülerdeki anomalilerin tespiti amacıyla Autoencoder (Otoenkoder) tabanlı bir model ele alınmıştır. Anomali tespiti, belirli bir veri kümesindeki normal örneklerden farklı olan örneklerin belirlenmesi işlemidir. Bu model, her bir görüntüyü normal veya anormal olarak sınıflandırmak için yeniden yapılandırma hatası ve çekirdek yoğunluk tahmini (Kernel Density Estimation - KDE) yöntemlerini kullanmaktadır. Anomali tespiti eğitimi, hazırlanan veri seti üzerinde



gerçekleştirilmekte olup, kullanılan yöntem farklı görüntü kümelerine de uygulanabilmektedir [85].

#### **4.9.4.1 Veri kümesi ve eğitim**

Modelin eğitimi için test sisteminden elde edilen veri seti kullanılmıştır ve bu veri seti, her bir görselin ya normal (hatasız) ya da anormal (parazitli) olduğunu belirten etiketleri bulunmaktadır. Veri kümesi, üç ana alt gruba ayrılmaktadır:

- Eğitim veri kümesi (train\_HATASIZ),
- Test veri kümesi (test\_HATASIZ),
- Anomali (hatalı) veri kümesi (DELIK,KIRIKIGNE).

Veri setinden elde edilen görseller, modelin eğitim sürecinde kullanılacak şekilde 64x64 piksel boyutunda yeniden boyutlandırılmıştır. ImageDataGenerator sınıfı, verilerin ölçeklendirilmesi ve düzgün bir şekilde beslenmesi için kullanılmıştır [86].

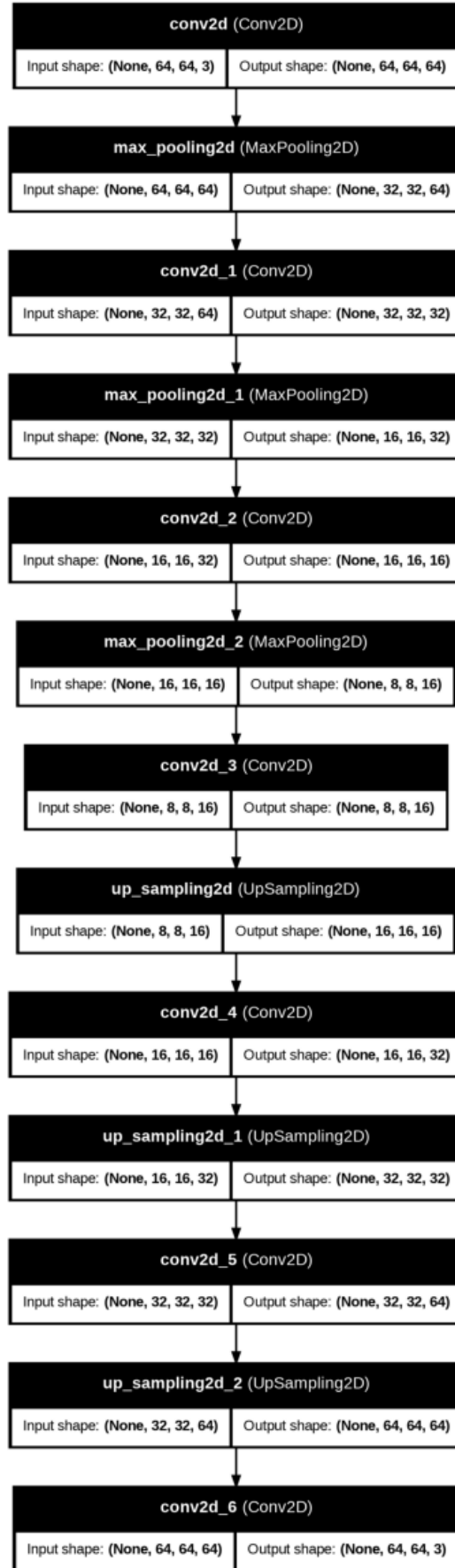
#### **4.9.4.2 Modelin yapılandırılması**

Model, Autoencoder mimarisi ile tasarlanmıştır. Autoencoder, bir görüntüyü sıkıştırarak (encode) daha küçük bir boyuta indirgemeyi ve ardından bu sıkıştırılmış veriyi kullanarak (decode) orijinal görüntüyü yeniden oluşturmayı amaçlamaktadır. Bu modelin başarısı, özellikle bottleneck (şişe boynu) katmanındaki sıkıştırılmış özelliklerin doğru şekilde çıkarılmasıyla ölçülmektedir [87].

Modelin temel katmanları şu şekilde yapılandırılmıştır:

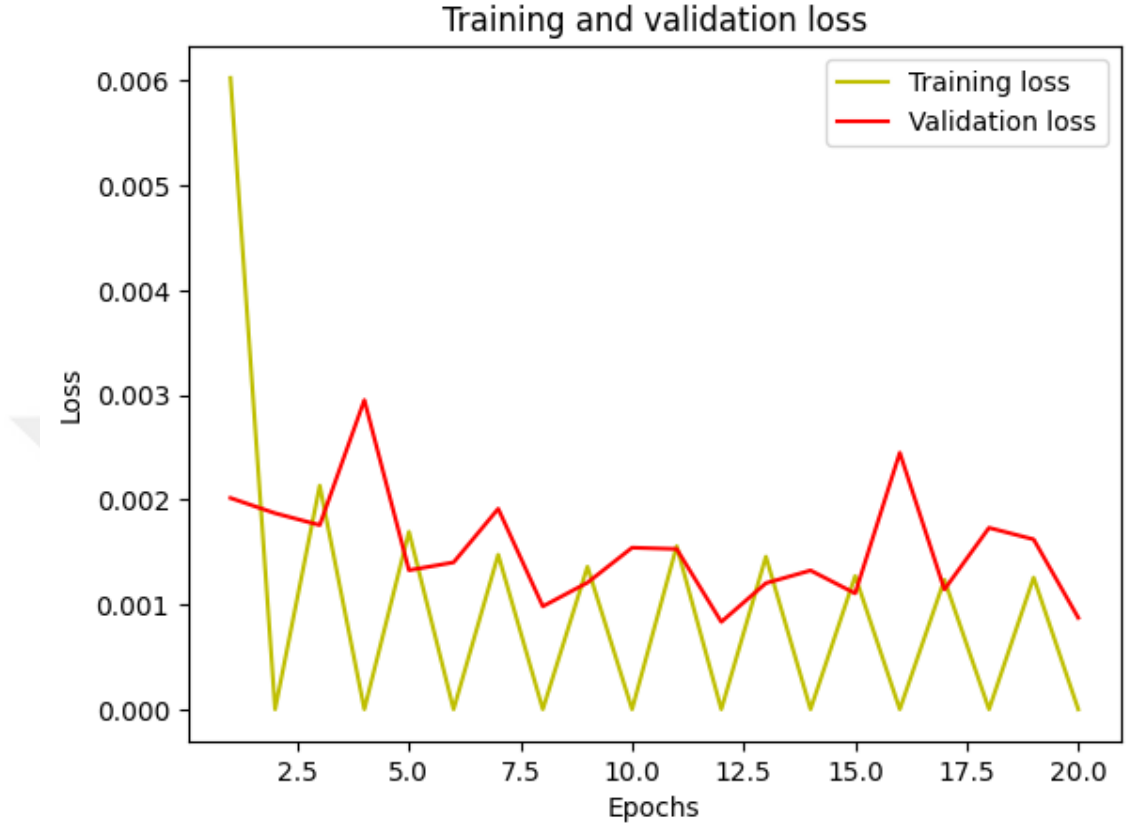
- Encoder: Konvolüsyonel katmanlar, her seferinde özelliklerin sayısını azaltarak daha küçük boyutlarda sıkıştırılmış bir temsil (latent space) oluşturmaktadır. Her konvolüsyonel katmandan sonra MaxPooling2D katmanları ile boyut küçültme işlemi yapılmaktadır.
- Decoder: Encoder'dan alınan sıkıştırılmış veriler, tersine dönüştürülerek orijinal görüntünün yeniden oluşturulmasını sağlamaktadır. Burada UpSampling2D katmanları kullanılarak görüntü boyutları büyütülmektedir.

Katmanları açıklanan bu sequential model Şekil 4.26'de verilmektedir. Bu modelin toplam parametresi 52.067(203.39 KB), eğitilebilir parametresi 52.067(203.39 KB) ve eğitilemez parametresi 0'dır.



Şekil 4.26 : Anomali tanıma modeli.

Model, Adam optimizasyon algoritması ve mean squared error (MSE) kayıp fonksiyonu ile eğitilmiştir. Eğitim sırasında, modelin doğruluğu ve kaybı izlenmiş ve öğrenme süreci Şekil 4.27’da görselleştirilmektedir.



Şekil 4.27 : Eğitim model doğruluğu ve kaybı.

#### 4.9.4.3 Anomali tespiti için latent alanın kullanımı

Modelin en kritik bileşeni, encoder katmanının çıktısı olan latent space (gizli uzay)’dır. Bu latent alan, her görüntü için yüksek boyutlu bir vektör olarak temsil edilmekte olup, anomali tespiti sürecinde temel bir rol oynamaktadır. Anomali tespiti için, bu vektörlerin yoğunluk dağılımı incelenmektedir. Normal ve anormal görüntülere ait vektörlerin yoğunlukları arasında belirgin bir fark gözlemlenmekte olup, bu fark anomali tespiti için ayırt edici bir özellik olarak kullanılabilir [88].

#### Çekirdek yoğunluk tahmini (KDE)

Kernel Density Estimation (KDE), bir veri kümesindeki her bir örneğin yoğunluğunu modellemek amacıyla kullanılan istatistiksel bir tekniktir. Bu çalışmada, scikit-learn kütüphanesinde yer alan KernelDensity sınıfı kullanılarak latent alan vektörlerinin yoğunluğu tahmin edilmektedir. Eğitim verisi üzerinde KDE modeli eğitildikten sonra,

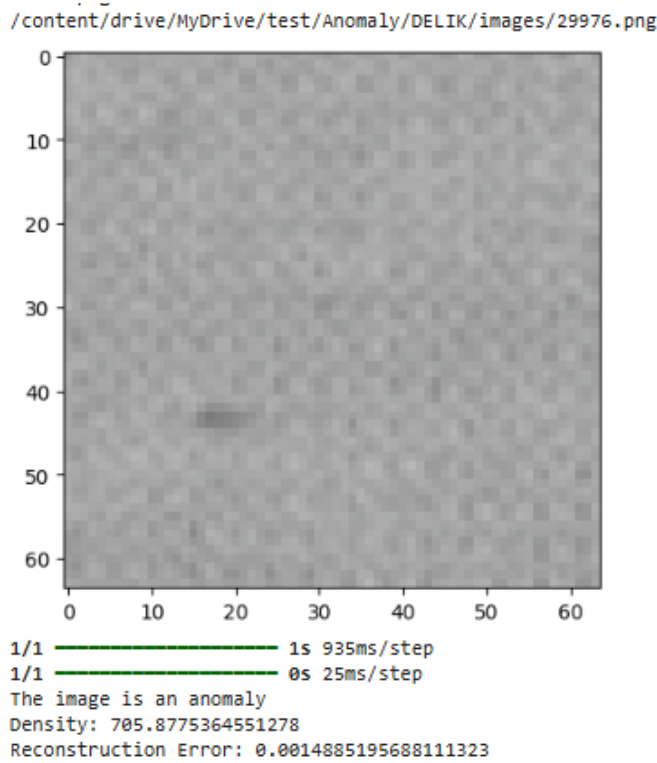
test verisine ait her bir görüntünün karşılık gelen yoğunluğu hesaplanmaktadır. Anomali tespiti sürecinde, normal görüntüler genellikle daha yüksek yoğunluk değerlerine sahipken, anormal görüntüler düşük yoğunluk değerleriyle tanımlanmaktadır [88].

### Yeniden yapılandırma hatası

Bir diğer önemli metrik yeniden yapılandırma hatasıdır. Bu hata, model tarafından yeniden oluşturulan görüntü ile orijinal görüntü arasındaki farkı ifade etmektedir. Anormal bir görüntü, modelin orijinal halini daha düşük doğrulukla yeniden oluşturmasına neden olmakta ve bu durum yüksek yeniden yapılandırma hatası ile sonuçlanmaktadır [89,90].

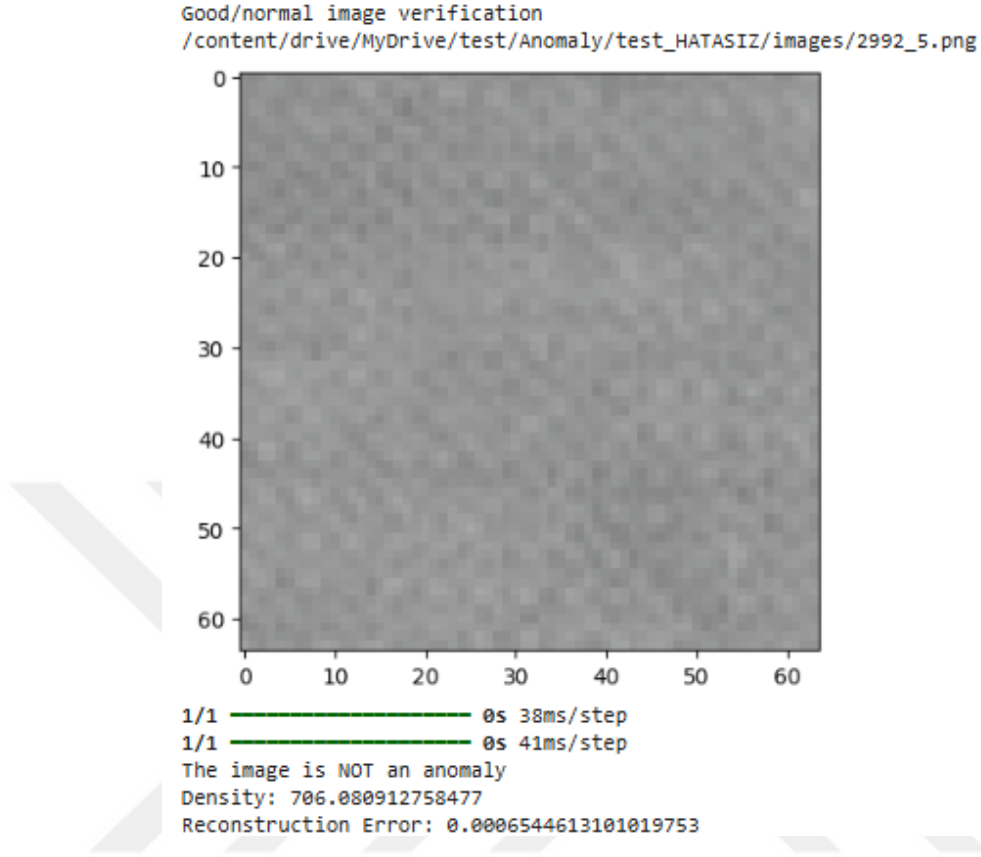
### Anomali tespiti

Görüntülerin anomali olup olmadığını belirlemek için hem yeniden yapılandırma hatası hem de yoğunluk bilgileri birlikte kullanılmaktadır. Belirli bir eşik değeri (threshold) ile bu iki metrik karşılaştırılmaktadır. Eğer bir görüntü, her iki metrik açısından da belirli eşiklerin dışındaysa, bu görüntü anormal olarak sınıflandırılmaktadır. Şekil 4.28’de delik tipi hatalı bir görüntü anomali tespiti yapılmış ve anomali olarak tespit edilmiştir.



Şekil 4.28 : Anomali hata tespiti.

Şekil 4.29’de hatasız bir görüntü anomali tespiti yapılmış ve anomali değil olarak tespit edilmiştir.



Şekil 4.29 : Anomali olmayan hata tespiti.

Modelin başarısını değerlendirmek için hem normal veri kümesi hem de anomali veri kümesi üzerindeki yeniden yapılandırma hatası hesaplanmaktadır. Sonuçlar, modelin normal görüntüleri ne kadar doğru bir şekilde yeniden yapılandırıldığını ve anormal görüntüleri doğru bir şekilde tespit edip etmediğini göstermektedir.

Aşağıdaki adımlar uygulanarak modelin doğruluğu değerlendirilmiştir:

Eğitim ve doğrulama kaybı: Eğitim sırasında modelin kaybı izlenmiş ve eğitim sürecinde hangi epoch'ta modelin daha iyi sonuçlar verdiği görselleştirilmiştir.

Anomali ve normal görüntülerin doğruluğu: Modelin doğruluğu, rastgele seçilen anormal ve normal görüntüler üzerinde test edilmiştir. Bu görüntüler, model tarafından anomali veya normal olarak doğru bir şekilde sınıflandırılmıştır [91].

Modelin tahmin ettiği ve gerçek sınıfları karşılaştırarak sonuçlar elde edilmiştir. Anomali görüntüler için model, yüksek bir yeniden yapılandırma hatası ve düşük bir yoğunluk skoru üretmiştir. Normal görüntüler ise düşük bir yeniden yapılandırma

hatası ve yüksek bir yoğunluk skoru ile tanımlanmıştır. Modelin doğruluğu, eşik değerlerine göre değişkenlik göstermektedir ve bu eşikler deneysel olarak ayarlanabilmektedir [92].

Bu çalışmada kullanılan Autoencoder tabanlı model, farklı uygulamalarda da kullanılan bir yöntemdir. Gelecekte, modelin daha geniş veri kümeleri üzerinde test edilmesi ve daha fazla parametre ile optimize edilmesi ile daha başarılı sonuçlar alınması hedeflenmektedir.



## 5. DENEYSEL DÜZENEK

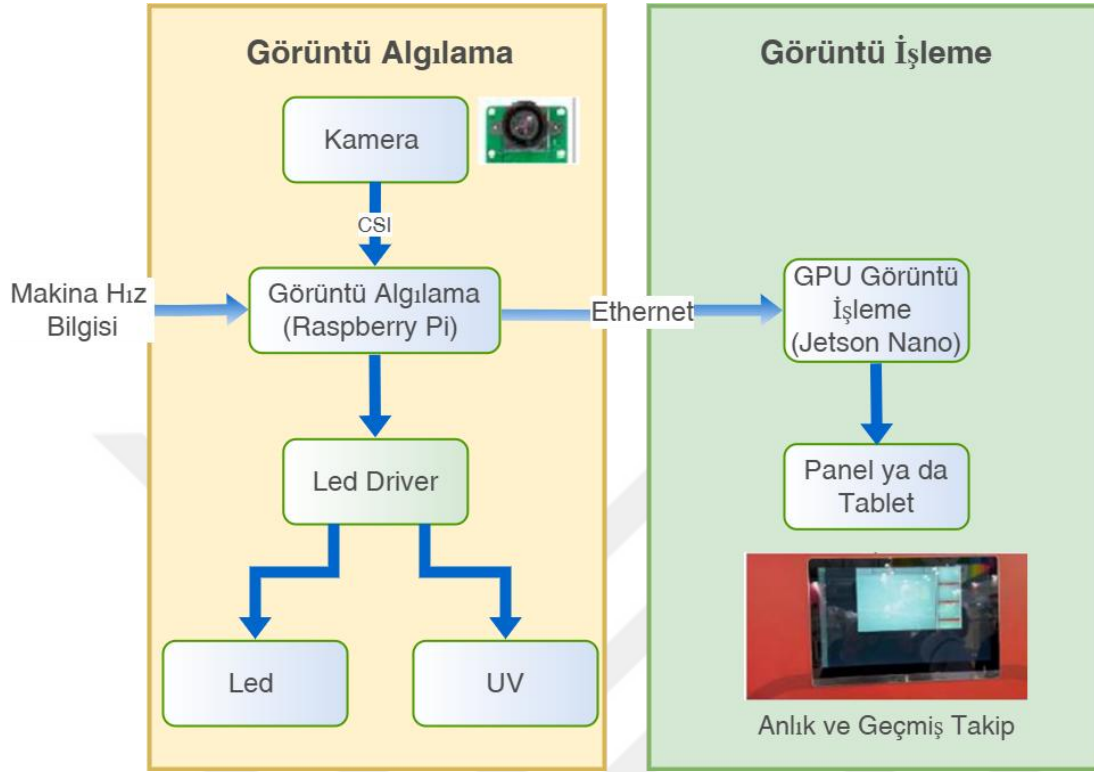
Sistemin en iyi şekilde çalışması için önerilen yöntemler performans, işleme zamanı, hedeflenen başarı oranına göre optimum çalışacak bir model oluşturulması hedeflenmektedir. Gerçek zamanlı çalışacak bu sistem ön işleme adımı ile hatalı olan bölgeleri tespit ederek eleme yapacak, hata skorları yüksek bölgeler bu yöntemler ile işlenerek sınıflandırma yapılmaktadır.

Hata tespiti sürecinde ön işleme (preprocessing) adımı, veri setinin kalitesini artırmak ve hata tespit algoritmalarının daha etkili çalışmasını sağlamak için kritik bir öneme sahiptir. Ön işleme, veri setindeki gürültüyü azaltma, özellik çıkarma, normalizasyon gibi işlemleri içermektedir. Preproses işlemleri, hata tespit sistemlerinin güvenilirliğini ve performansını artırarak endüstriyel uygulamalarda daha etkili ve kesin sonuçlar elde edilmesine olanak tanımaktadır. Raspberry Pi 4, Camera Serial Interface (CSI) protokolü aracılığıyla kamera modülü üzerinden görüntü elde ederek, bu verileri Ethernet üzerinden Jetson Nano'ya aktarmaktadır. Jetson Nano üzerinde, CUDA destekli sinir ağı modelleri kullanılarak elde edilen görüntüler işlenmekte ve hata tespiti ile sınıflandırma işlemleri gerçekleştirilmektedir. CPU tabanlı çalışma sırasında 1 frame işleme süresi 1800ms zaman alırken CUDA desteği ile beraber işlemesi 140ms'ye kadar düşürülmüştür. Sonraki aşamada Jetson Nano ile görüntüler haberleşme ile alınarak bu işlemler çalışılmıştır.

Saha denemeleri kapsamında, öğrenme tabanlı hata tespit algoritması farklı kumaş tipleri üzerinde test edilerek, çeşitli kumaş yapıları ve hata türlerine adaptasyon kapasitesi değerlendirilmiştir. Bu denemeler, algoritmaların kumaş türlerine göre optimizasyonunu ve çeşitli hata türlerini (Delik, İğne Kırığı, Likra Kaçığı, May Çizgisi, Yağ Lekesi) yüksek oranda başarıyla tespit edebilmeyi sağlamıştır.

Algoritmaların saha denemeleri sırasında toplanan veriler, gerçek çalışma koşulları altında algoritma performansının değerlendirilmesine ve gerekli tuning işlemlerinin yapılmasına olanak tanımıştır. Bu süreç, algoritma parametrelerinin gerçek zamanlı çalışma koşullarına en uygun şekilde ayarlanarak, sistem performansının maksimize edilmesini sağlamıştır. Sistem saha testleri sonucunda: 0.2 mm/piksel çözünürlüğünde

saniyede 8-9 frame işleyebilir hale gelmiştir. Bu sonuçlar 30 rpm ile dönen ve 3 m/dk ile akan kumaş üzerinde çalışabilme imkânı sağlamıştır. Bu sistem modeli genel görünümü Şekil 5.1’de verilmektedir.



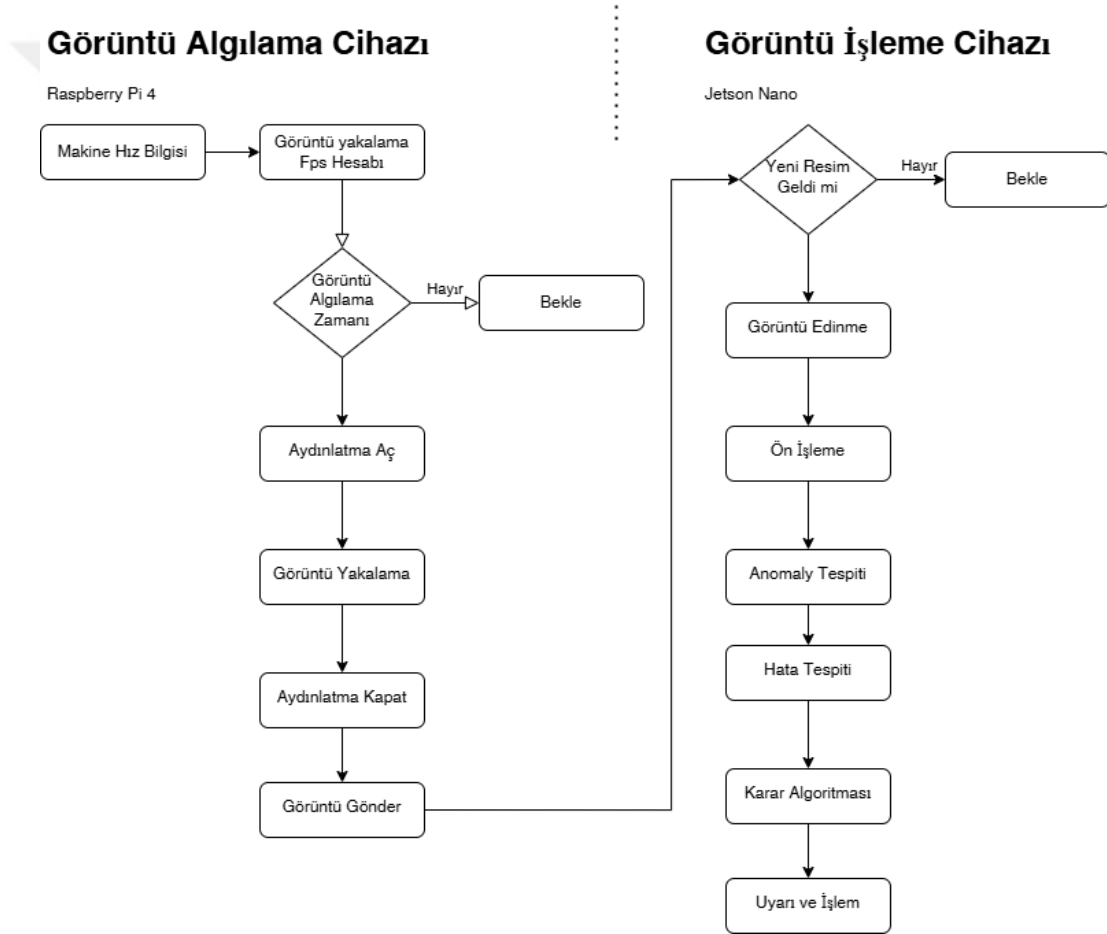
**Şekil 5.1 : Sistem modeli.**

Algoritmaların saha denemeleri sırasında toplanan veriler, gerçek çalışma koşulları altında algoritma performansının değerlendirilmesine ve gerekli tuning işlemlerinin yapılmasına olanak tanımıştır. Bu süreç, algoritma parametrelerinin gerçek zamanlı çalışma koşullarına en uygun şekilde ayarlanarak, sistem performansının maksimize edilmesini sağlamıştır.

Şekil 5.2, bir görüntü algılama ve işleme sisteminin işleyişini iki temel bileşen üzerinden açıklamaktadır: görüntü algılama cihazı ve görüntü işleme cihazı. Görüntü algılama cihazı, Raspberry Pi 4 kullanılarak çalışmakta olup, makinenin hız bilgisine dayalı olarak görüntü yakalama işlemini gerçekleştirmektedir. Bu bileşen, makinenin hızına bağlı olarak görüntü yakalama frekansını (FPS) hesaplamakta ve belirlenen zaman aralıklarında görüntü kaydetmektedir. Görüntü algılama süreci, yakalama işlemi başlamadan önce aydınlatma sisteminin aktif hale getirilmesi ile başlamakta, görüntü kaydı tamamlandıktan sonra ise sistem kapatılmaktadır. Algılanan görüntüler, daha ileri düzeyde işlenmek üzere Görüntü İşleme Cihazına iletilmektedir.



Görüntü İşleme Cihazı, Jetson Nano donanımı kullanılarak iletilen görüntülerin analizinden sorumludur. Bu süreçte, sistem öncelikle yeni bir görüntünün alınıp alınmadığını kontrol etmektedir. Yeni bir görüntü algılandığında, ön işleme aşaması uygulanmaktadır. Ön işleme, görüntüdeki gürültüyü azaltmayı ve analize uygun hale getirmeyi amaçlamaktadır. Ön işleme aşamasının ardından, işlenen görüntü üzerinde anomali tespiti gerçekleştirilmekte ve bu anomaliler arasından gerçek hatalar belirlenmektedir. Hata tespit sürecini takiben, bir karar algoritması devreye girerek uygun aksiyonların alınmasını sağlamaktadır. Son aşamada, kullanıcıya gerekli uyarılar iletilmekte ve durumun gerektirdiği şekilde makinenin durdurulması veya düzeltici işlemlerin başlatılması gibi müdahaleler gerçekleştirilmektedir.



**Şekil 5.2 : Sistem Çalışma Diyagramı.**



## 6. SONUÇLAR VE GELECEK ÇALIŞMALAR

Gerçekleştirilen çalışmalar sonucunda, kullanılan yöntemlerin performansı, hassasiyeti ve uygulanabilirliği kapsamlı bir şekilde değerlendirilmiş, bu doğrultuda belirli koşullar altında hangi yöntemin daha etkili olduğu analiz edilmiştir. Aşağıda, hata matrisi temel alınarak hesaplanan performans skorları ve işleme süreleri sunulmaktadır. Kusurlu kumaşların görüntüleri, gelişmiş kamera-optik sistemleri aracılığıyla elde edilmiş, tanımlama algoritmaları ise çeşitli yaklaşımlar kullanılarak geliştirilmiş ve test edilmiştir. Deneysel veriler doğrultusunda farklı yöntemlerin performansları karşılaştırmalı olarak incelenmiştir.

Örme fabrikalarından toplanan kusurlu kumaşlar, iş yeri test düzeneğine entegre edilerek üretim sistemleri ile benzer bir çalışma ortamı oluşturulmuştur. Yakalanan görüntülerde iğne kırığı, kelebek ve yağ lekesi gibi çeşitli fiziksel kusurlar tespit edilmiştir. Her hata türüne ilişkin sonuçlar, tüm modeller için Çizelge 6.1’de sunulmakta olup, elde edilen bulgular doğrultusunda belirli hata türlerinin hangi model ile daha etkin bir şekilde tespit edilebileceği ortaya konulmaktadır.

**Çizelge 6.1 :** Hata tiplerinde başarı oranı. (Eğrilik Algoritması, Modifiye Eğrilik Algoritması, Gabor Dönüşümü, Curvelet Dönüşümü, Hough Dönüşümü, Evrişimli Sinir Ağı, RestNet50, YoloV5).

| Yöntem    | Delik | İğne Kırığı | May İzi | Likra K. | Yağ İzi |
|-----------|-------|-------------|---------|----------|---------|
| EA        | 0,32  | 0,39        | 0,56    | 0,49     | 0,29    |
| MDEA      | 0,78  | 0,82        | 0,74    | 0,81     | 0,60    |
| GD        | 0,81  | 0,99        | 0,99    | 0,85     | 0,78    |
| CD        | 0,85  | 0,78        | 0,75    | 0,70     | 0,84    |
| HD        | 0,92  | 0,78        | 0,71    | 0,85     | 0,78    |
| ESA (CNN) | 0,91  | 0,90        | 0,88    | 0,82     | 0,92    |
| RestNet50 | 0,96  | 0,95        | 0,92    | 0,87     | 0,95    |
| YoloV5    | 0,94  | 0,95        | 0,92    | 0,86     | 0,94    |

Son olarak, farklı yaklaşımlar için karışıklık matrisi ve skorlar elde edilmiştir. Deneysel test sonuçları, doğruluk ve hassasiyet değerlerini karşılaştırmak amacıyla Çizelge 6.2’de sunulan hata başarı oranı verileri, farklı yöntemlerin hata tespiti performanslarını karşılaştırmalı olarak göstermektedir. Sonuçlara göre, geleneksel yöntemler arasında en yüksek doğruluk oranı (%91) ve en iyi F1 skoru (%94) Gabor

Dönüşümü (GD) yöntemiyle elde edilmiştir. Ancak, işlem süresi açısından değerlendirildiğinde GD yalnızca 110 ms ile oldukça hızlı bir performans sergilemektedir. Derin öğrenme tabanlı yöntemler arasında ise en yüksek başarı oranı RestNet50 (%96 doğruluk, %95 F1 skoru) ile elde edilirken, işlem süresi 1340 ms ile diğer yöntemlere kıyasla daha uzun sürmektedir. YoloV5 yöntemi, %95 doğruluk ve %93 F1 skoru ile yüksek performans sunarken, 127 ms işlem süresi ile hızlı bir alternatif oluşturmaktadır. Geleneksel yöntemler arasında Hough Dönüşümü (HD), %87 doğruluk ve 38 ms işlem süresi ile düşük işlem maliyeti avantajına sahiptir. Genel olarak, derin öğrenme tabanlı yöntemler daha yüksek doğruluk oranları sağlarken, işlem süresi açısından bazı geleneksel yöntemlere kıyasla daha fazla hesaplama maliyeti gerektirdiği görülmektedir.

**Çizelge 6.2 :** Hata başarı oranı. (Eğrilik Algoritması, Modifiye Eğrilik Algoritması, Gabor Dönüşümü, Curvelet Dönüşümü, Hough Dönüşümü, Evrişimli Sinir Ağı, RestNet50, YoloV5).

| Yöntem    | Acc  | F1   | Precision | Recall | Process Time |
|-----------|------|------|-----------|--------|--------------|
| EA        | 0,32 | 0,39 | 0,56      | 0,29   | 203ms        |
| MDEA      | 0,78 | 0,82 | 0,74      | 0,92   | 295ms        |
| GD        | 0,91 | 0,94 | 0,89      | 0,99   | 110ms        |
| CD        | 0,80 | 0,65 | 0,81      | 0,72   | 1150ms       |
| HD        | 0,87 | 0,77 | 0,85      | 0,84   | 38ms         |
| ESA (CNN) | 0,93 | 0,89 | 0,91      | 0,89   | 1080ms       |
| RestNet50 | 0,96 | 0,95 | 0,95      | 0,95   | 1340ms       |
| YoloV5    | 0,95 | 0,93 | 0,92      | 0,93   | 127ms        |

Makinenin tespit edebileceği hataların en az 1 mm boyutunda olması durumunda, doğru bir algılama gerçekleştirebilmek için en az 4 piksel çözünürlüğünde görüntü alınması gerekmektedir. Bu gereksinim, makinenin günlük kapasitesi ile birlikte değerlendirildiğinde, günlük olarak yaklaşık 50-60 GB veri işlenmesi ve depolanması gerekliliğini ortaya koymaktadır. Gerçek zamanlı çalışma hedefi göz önünde bulundurulduğunda, önerilen kesik eğrilik algoritmasının, işlenen ve saklanan veri miktarı açısından öğrenme tabanlı algoritmalara kıyasla daha avantajlı olduğu sonucuna varılmıştır.

Hatalı görüntüler üzerinde gerçekleştirilen test sonuçları incelendiğinde, 96×200 piksel boyutundaki bir görüntünün 57.600 bayt veri içerdiği, buna karşın 638 noktada çıkarılan özelliklerin yalnızca 5.100 bayt veri gerektirdiği belirlenmiştir. Bu durum, veri sıkıştırma oranını iyileştirerek sistemin gerçek zamanlı çalışma yeteneğini artırmaktadır.

Hatalı kumaş tespiti için önerilen eğrilik algoritması, bu sorun için özel olarak yapılan geliştirmelerle birlikte hatalı kumaş görüntüsü üzerinde uygulanmış ve elde edilen sonuçlar sunulmuştur. Her nokta için eğim alınarak, konveks veya konkav olup olmadığı kontrol edilmiştir ve akıllı seçim algoritması ile güncellenerek bir eğrilik fonksiyonu elde edilmiştir. Elde edilen dış yarıçap ve açı değerleri ile hata tespiti için orta seviye doğrulukla kullanılabileceği gösterilmiştir.

Gabor dönüşümü, hata türüne özgü parametre ayarları ile dikey ve yatay hataların tespitinde yüksek başarı oranlarına ulaşabilmektedir. Bununla birlikte, dikey ve yatay hatalardaki bu başarılı sonuçlara rağmen, çapraz ve diğer türdeki hataların tespitinde beklenen performansı sergileyememektedir. Farklı kumaş türleri için benzer başarı düzeyine ulaşabilmek amacıyla, parametrelerin güncellenmesi ve optimizasyonu gerekmektedir.

Curvelet Dönüşümü, yağ lekeleri kusurlarının tespitinde güçlü bir performans sergilemiş ve %84 doğruluk oranına ulaşarak delik tespiti için gösterdiği başarıya yakın bir sonuç elde etmiştir. Bu durum, Curvelet Dönüşümünün özellikle ince dokuların ve desenlerin kritik olduğu kusur türlerinde etkili bir yöntem olduğunu göstermektedir. Bununla birlikte, yöntemin görüntüye dönüşüm uygulayıp ardından yeniden oluşturması nedeniyle yüksek doğruluk oranı sağlasa da gerçek zamanlı uygulamalar için yeterli hızda çalışmasının zor olduğu değerlendirilmektedir.

Bununla birlikte, Hough Dönüşümü, delik tespitinde en yüksek doğruluk oranını elde ederek en iyi performansı göstermiş, ancak diğer kusur türlerinde orta seviyede doğruluk sağlamıştır. Çoğu kusur türünde öğrenme tabanlı modeller diğer yöntemleri geride bıraksa da Curvelet Dönüşümü yağ kusurlarının tespitinde öne çıkmaktadır. Bu durum, belirli kusur analizlerinin gerekli olduğu uygulamalarda Curvelet Dönüşümünün avantaj sağladığını göstermektedir. Bu yöntemlerin bir araya getirilmesiyle daha güçlü bir hata tespit mekanizması oluşturulabilir. Curvelet Dönüşümü, görüntüyü ön işleyerek kenarları ve eğrileri belirginleştirirken, Hough Dönüşümü şekil tespiti aşamasında daha yüksek doğruluk sağlayabilmektedir. Curvelet Dönüşümü ve Hough Dönüşümü, görüntü işleme alanında iki farklı ancak güçlü paradigmayı temsil etmektedir. Curvelet Dönüşümü, eğri özelliklerin çok ölçekli ve yönlü analizine olanak tanırken, Hough Dönüşümü parametrik şekillerin tespitinde üstün performans göstermektedir. Bu iki yöntemin birleştirilmesi, özellikle kusur

tespiti gerektiren endüstriyel uygulamalarda hata analizinin doğruluğunu arttırılabilmektedir.

Öğrenme tabanlı kumaş kusur tespit yöntemleri, diğer geleneksel yaklaşımlara kıyasla yüksek doğruluk oranları sağlamakla birlikte, büyük miktarda eğitim verisi ve yüksek işlem gücü gerektirmektedir. Yeterli miktarda eğitim verisinin bulunmadığı durumlarda, en yüksek başarı oranları genellikle ResNet-50 gibi önceden eğitilmiş derin öğrenme modelleri kullanılarak elde edilmiştir. Yerel hata tespiti için yüksek performanslı donanımın kullanılması gerekmektedir. Bu donanım gereksinimleri sağlandığında, yüksek doğruluk oranı sunan öğrenme tabanlı yaklaşımlar etkin bir şekilde uygulanabilecektir. Gerçekleştirilen tüm hesaplama ve testler, gerçek zamanlı hata tespiti için donanım seçiminin kritik bir öneme sahip olduğunu ortaya koymaktadır. Çizelge 6.2'de sunulan işlem süresi sonuçları, Raspberry Pi gibi standart bir geliştirme kartı ve Google Colab ortamında hesaplanmıştır. Ardından, tüm testler GPU destekli bir gömülü sistem üzerinde gerçekleştirilmiş ve bu sistemin performansı diğer donanımlarla karşılaştırılmıştır. Yapılan testler sonucunda, gömülü sistemin daha yüksek performans sunduğu ve gerçek zamanlı işleme kriterlerine daha yakın sonuçlar sağladığı belirlenmiştir. GPU üzerinde çalışan öğrenme tabanlı hata tespit yöntemi, tek bir görüntü için yaklaşık 110-130 ms sürede işlem yapmaktadır. Elde edilen bu sonuçlar, neredeyse gerçek zamanlı bir gömülü sistem uygulaması geliştirmek için yeterli işlem hızına ulaşıldığını göstermektedir.

Farklı tekniklerin birleştirilmesiyle, geniş bir kusur yelpazesine uyum sağlayabilen, yüksek doğruluk ve performans sunan entegre bir sistem geliştirilmiştir. İlk aşamada, görüntülerin ön işlenmesi sırasında detayların belirginleştirilmesi sağlanmış, sonrasında şekil ve kusur tespiti için hassas analizler gerçekleştirilmiştir. Bunun üzerine, ileri düzey algoritmaların kullanımıyla, farklı kusur türlerinde etkili bir şekilde çalışan ve yüksek doğruluk sağlayan bir yapı oluşturulmuştur. Donanım üzerinde yapılan testler, sistemin gerçek zamanlı çalışmaya uygun olduğunu ve yüksek başarı kriterlerini karşıladığını göstermiştir. Bu yaklaşımlar bir araya getirilerek, kusur tespiti ve sınıflandırması için etkin ve uygulanabilir bir ürün ortaya konmuştur.

Geliştirilen sistemin başarısı, yalnızca kullanılan algoritmaların doğruluğuna değil, aynı zamanda maliyet ve performans dengesine de dayanmaktadır. Sistem, yüksek performanslı donanım gereksinimini azaltmak için ön işleme aşamasında detayları optimize eden tekniklerden faydalanmaktadır. Bu, görüntülerin doğru bir şekilde

analiz edilmesini sađlarken iřlem ykn dřrr ve ađ maliyetini azaltmaktadır. zellikle đrenme tabanlı yntemler, eđitim iin byk veri setleri ve yksek iřlem gc gerektirdiđinden, n iřleme yntemleri hem verimliliđi artırabilir hem de daha dřk maliyetle uygulanabilirlik sađlayabilmektedir. Ayrıca, ađ yapısının karmařıklıđı ve byklđ, sistemin maliyetini ve gerek zamanlı performansını dođrudan etkilemektedir. Bu nedenle, dođru n iřleme tekniklerinin seimi, donanım ve ađ maliyeti zerindeki yk azaltırken, yksek dođruluk ve hızlı sonular elde edilmesini mmkn kılmaktadır. Bu yaklařım, maliyet, performans ve esneklik aısından sistemin endstriyel uygulanabilirliđini artırmaktadır.

### **6.1 Gelecek Uygulama Alanı**

Kumař hata tespitinde yapay zeka ve makine đrenimi teknolojilerinin daha yaygın ve etkin bir řekilde kullanılması beklenmektedir. Bu bađlamda, yerli ve uluslararası birok giriřim, geliřmiř grnt iřleme ve analiz teknikleri ile hataları daha hızlı ve dođru bir řekilde tespit etmeye odaklanmaktadır. Trkiye'deki ve dnya genelindeki yeniliki řirketler, tekstil fabrikalarına entegre edilebilen akıllı kamera sistemleri geliřtirerek retim hatlarında gerek zamanlı hata tespiti gerekleřtirebilmektedir. Bu sistemler, retim srelerinde verimliliđi artırırken aynı zamanda maliyetleri dřrmekte ve insan hatasını minimize ederek kalite kontrol srelerini optimize etmektedir. Bu tr teknolojiler, kumař kalitesinde standartları ykselterek mřteri memnuniyetini artırmakta ve tekstil endstrisinin dijital dnřmn hızlandırmaktadır.

Bu teknolojik ilerlemeler, aynı zamanda uluslararası rekabet ortamında nemli bir avantaj sađlamaktadır. zellikle, hata tespit sistemlerinin makine đrenimi algoritmalarıyla desteklenmesi, karmařık hataların bile yksek dođrulukla tespit edilmesini mmkn kılmaktadır. Bunun yanı sıra, sistemlerin srekli olarak đrenme ve geliřme kapasitesi sayesinde, tekstil endstrisinde kalite kontrol sreleri daha esnek ve dinamik bir yapıya kavuřacaktır. Bu tr yeniliki zmler, yalnızca verimliliđi artırmakla kalmayıp, srdrlebilir retim hedeflerine ulařılmasına da katkı sađlamaktadır. Sonu olarak, kumař hata tespitinde yapay zekâ ve makine đrenimi tabanlı zmler, tekstil endstrisinin geleceđini řekillendirerek, sektrde kresel standartların belirlenmesinde nc bir rol oynayacaktır.

Gelecekte, kumaş hata tespitinde generative AI ve büyük dil modelleri (LLM'ler) önemli bir rol alacağı beklenmektedir. Generative AI, doğal dil işleme ve öğrenme yetenekleriyle, hata tespit sistemlerine daha akıllı analiz ve raporlama yetenekleri kazandırabilir. Özellikle, kumaş üzerinde tespit edilen kusurların sınıflandırılması, bu kusurların oluşma nedenlerinin analiz edilmesi ve çözüm önerilerinin sunulması gibi işlemler, generative AI'nin sağladığı gelişmiş bağlam anlama ve veri üretme yetenekleri sayesinde daha hızlı ve etkili hale gelebilir. Gerçek zamanlı veri analitiğiyle, generative AI, üretim süreçlerinden gelen büyük miktarda veriyi işleyebilir, hata türleri hakkında detaylı raporlar oluşturabilir ve bu bilgileri operatörlere açıklayıcı bir şekilde sunabilir. Böylece, hata tespit ve müdahale süreçleri daha bilinçli ve proaktif bir şekilde yürütülebilir.

Ayrıca, generative AI, kumaş üretiminde sürdürülebilirlik ve verimliliği artırma potansiyeline sahiptir. Üretim hattında biriken geçmiş verileri analiz ederek kusur oluşumunun nedenlerini önceden tahmin edebilir ve iyileştirme önerileri sunabilir. Bunun yanında, üretim sürecinde oluşan hataların detaylı sınıflandırması, kusurlu kumaşların geri dönüşüm veya yeniden işleme süreçlerine yönlendirilmesini sağlayabilir. Generative AI, dijital ikiz teknolojileri ve artırılmış gerçeklik (AR) ile entegre edildiğinde, operatörlere hataların görselleştirilmesi ve çözüm önerilerinin interaktif bir şekilde sunulması gibi yenilikçi yöntemler sunabilir. Bu gelişmeler, tekstil endüstrisinde kaliteyi artırırken, atık oranlarını azaltarak çevresel etkilerin de minimuma indirilmesine katkı sağlayacaktır.



## KAYNAKLAR

- [1] Hanbay K., Talu M. F., Özgüven F. (1998). *Fabric defect detection systems and methods—a systematic literature review*. *Optik* 127(24), 11960–11973. <https://doi.org/10.1016/j.ijleo.2016.09.110>
- [2] Kumar A. (2011). *Computer vision-based fabric defect analysis and measurement*, 45–65. <https://doi.org/10.1533/9780857093608.1.45>
- [3] Harwood J., Davis G., Wolfe P. (2003). *Automated Fabric Inspection Systems: Evaluation of Commercial Products*. *Text. Res. J.* 73, 702–709.
- [4] Tsai D. M., Hsieh C. Y. (1995). *Automated Surface Inspection Using Gabor Filters*. *Int. J. Adv. Manuf. Technol.* 10, 419–426.
- [5] Garbowski T., Maier G., Novati G. (2012). *On calibration of orthotropic elastic-plastic constitutive models for paper foils by biaxial tests and inverse analyses*. *Struct. Multidiscip. Optim.* 46, 111–128.
- [6] Asada H., Brady M. (1986). *The Curvature Primal Sketch*. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8*, 2–14.
- [7] Karataş İ., Soyaslan D. D. (2016). *Tekstil Endüstrisinde Yapay Görme ile Hata Tespit Uygulamaları*. *Mehmet Akif Ersoy Üniversitesi Fen Bilimleri Enstitüsü Dergisi, Özel Sayı 1*, 51–57.
- [8] VA Imaging. *How to select a machine vision camera interface: USB3, GigE, 5GigE, 10GigE Vision*. VA Imaging. Available: <https://va-imaging.com/blogs/machine-vision-knowledge-center/how-to-select-a-machine-vision-camera-interface-usb3-gige-5gige-10gige-vision>. [Erişim: Aralık. 2024].
- [9] Ahmed S. A. (2018). *Comparative Study Among Sobel, Prewitt and Canny Edge Detection Operators Used in Image Processing*. *J. Theor. Appl. Inf. Technol.* 96(19), 6517–6526.
- [10] Roerdink J. B. T. M. (1998). *An Introduction to Digital Image Processing*. Department of Mathematics and Computing Science, University of Groningen, Groningen, The Netherlands.
- [11] Gonzalez R. C., Woods R. E. (2008). *Digital Image Processing*, 1–976.
- [12] Zhu Y., Er M. J. (2000). *Moment-Based Analysis of Closed Contours for Texture and Shape Classification*. *IEEE Trans. Syst. Man Cybern. Part Syst. Hum.* 30, 525–533.
- [13] Li H. (2015). *Image Contour Extraction Method based on Computer Technology*. In *Proceedings of the 2015 4th National Conference on Electrical, Electronics and Computer Engineering*, Xi'an, China, 12–13 December 2015; Atlantis Press: Paris, France, pp. 1185–1189.

- [14] **Ciomaga A., Monasse P., Morel J.-M.** (2017). *The Image Curvature Microscope: Accurate Curvature Computation at Subpixel Resolution*. *Image Process. Line* 7, 197–217.
- [15] **Tu L. W.** (2017). *Differential Geometry: Connections, Curvature, and Characteristic Classes*. In *Graduate Texts in Mathematics*; Springer: Cham, Switzerland; ISBN 978-3-319-55082-4.
- [16] **Sonka M., Hlavac V., Boyle R.** (2008). *Image Processing, Analysis, and Machine Vision*, 3rd ed.; Cengage Learning: Boston, MA, USA; pp. 223–230.
- [17] **Subrahmonia J. D., Lakshmanan S., Goldberg M.** (1996). *Edge and curvature extraction for real-time road boundary detection*. *IEEE Trans. Veh. Technol.* 45, 774–784.
- [18] **Canny J.** (1986). *A computational approach to edge detection*. *IEEE Trans. Pattern Anal. Mach. Intell.* 8, 679–698. [PubMed]
- [19] **Suzuki S., Abe K.** (1985). *Topological structural analysis of digital binary images by border following*. *Comput. Vision Graph. Image Process.* 30, 32–46.
- [20] **Papoulis A.** (2000). *Probability, Random Variables, and Stochastic Processes*, 4th ed.; McGraw-Hill: New York, NY, USA; pp. 315–318.
- [21] **Bovik A. C., Clark M., Geisler W. S.** (1990). *Multichannel texture analysis using localized spatial filters*. *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 55–73.
- [22] **Dunn D., Higgins W., Wakeley J.** (1994). *Texture segmentation using 2-D Gabor elementary functions*. *IEEE Trans. Pattern Anal. Mach. Intell.* 16, 130–149.
- [23] **Jain A. K., Farrokhnia F.** (1991). *Unsupervised texture segmentation using Gabor filters*. *Pattern Recognit.* 24, 1167–1187.
- [24] **Manjunath B. S., Ma W. Y.** (1996). *Texture features for browsing and retrieval of image data*. *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 837–852.
- [25] **Vautrot P., Bonnet N., Herbin M.** (1996). *Comparative study of different spatial/spatial-frequency methods (Gabor filters, wavelets, wavelets packets) for texture segmentation/classification*. In *Proceedings of the 3rd IEEE International Conference on Image Processing*, Lausanne, Switzerland, 19 September 1996; Volume 3, pp. 145–148.
- [26] **Bigun J., du Buf J. M. H.** (1994). *N-folded symmetries by complex moments in Gabor space and their application to unsupervised texture segmentation*. *IEEE Trans. Pattern Anal. Mach. Intell.* 16, 80–89.
- [27] **Teuner A., Pichler O., Hosticka B. J.** (1995). *Unsupervised texture segmentation of images using tuned matched Gabor filters*. *IEEE Trans. Image Process.* 4, 863–873.
- [28] **Bodnarova A., Bennamoun M., Latham S.** (2002). *Optimal Gabor filters for textile flaw detection*. *Pattern Recognit.* 35, 2973–2991.
- [29] **Clausi D. A., Jernigan M. E.** (2000). *Designing Gabor filters for optimal texture separability*. *Pattern Recognit.* 33, 1835–1849.

- [30] Candes E. J., Donoho D. L. (2000). *Curvelets—A surprisingly effective nonadaptive representation for objects with edges*. *Curves and Surfaces*, 105–120.
- [31] Zhang Y., Li T. L., Li Q. (2013). *Defect detection for tire laser shearography image using curvelet transform-based edge detector*. *NDT E Int.* 55, 12–19. <https://doi.org/10.1016/j.ndteint.2013.01.005>
- [32] Starck J., Candes E. J., Donoho D. L. (2002). *The curvelet transform for image denoising*. *IEEE Trans. Image Process.* 11(6), 670–684. <https://doi.org/10.1109/TIP.2002.1034998>
- [33] Do M. N., Vetterli M. (2005). *Fast discrete curvelet transforms*. *IEEE Trans. Image Process.* 14(12), 2080–2093. <https://doi.org/10.1109/TIP.2005.85937115>
- [34] Demanet L., Ying L. (2005). *Multiscale geometric image processing using curvelets*. *IEEE Trans. Image Process.* 14(6), 934–945. <https://doi.org/10.1109/TIP.2005.847031>
- [35] Pennec E. L., Mallat S. (2005). *Directional multiscale modeling of images using curvelets*. *IEEE Trans. Image Process.* 14(4), 439–450. <https://doi.org/10.1109/TIP.2005.844103>
- [36] Khoje S. A., Adsul S. K. B. (2018). *Automated skin defect identification system for fruit grading based on discrete curvelet transform*. *Int. J. Image Process.* 12(4), 197–205. <https://doi.org/10.1109/IJIP.2018.1234567>
- [37] Deng Y., Ding K., Yaguang D. (2011). *Wavelets and curvelets transform for image denoising to damage identification of thin plate*. *J. Sound Vib.* 330(22), 5291–5303. <https://doi.org/10.1016/j.jsv.2011.05.012>
- [38] Candes E. J., Donoho D. L. (2003). *Curvelets and optimal representations of multi-dimensional signals*. *Multiscale Model. Simul.* 1(2), 249–262. <https://doi.org/10.1137/S1540345903425686>
- [39] Vafaie S., Salajegheh E. (2020). *A comparative study of shearlet, wavelet, laplacian pyramid, curvelet, and contourlet transform to defect detection*. *J. Appl. Remote Sens.* 14(2), 1–15. <https://doi.org/10.1117/1.JRS.14.022201>
- [40] Miri M. S., Mahloojifar A. (2011). *Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction*. *IEEE Trans. Biomed. Eng.* 58(5), 1183–1192. <https://doi.org/10.1109/TBME.2010.2095023>
- [41] Xu K., Ding Y. (2014). *Application of shearlet transform to classification of surface defects for metals*. *J. Mater. Process. Technol.* 214(4), 795–802. <https://doi.org/10.1016/j.jmatprotec.2013.11.011>
- [42] Zhang H., Tian X. M., Tian Y. (2018). *An image fusion method based on curvelet transform and guided filter enhancement*. *IEEE Access* 6, 26550–26559. <https://doi.org/10.1109/ACCESS.2018.2831412>
- [43] IEEE International Conference on Image Processing (ICIP). (2005). *Proceedings of the IEEE International Conference on Image Processing*, pp. 157–160. <https://doi.org/10.1109/ICIP.2005.1530044>

- [44] Pennec E. L., Mallat S. (2005). *Fast curvelet transform for image analysis*. *Multiscale Model. Simul.* 4(3), 992–1032. <https://doi.org/10.1137/04060439016>
- [45] P. A., Sabeenian R. S. (2018). *Fabric defect detection using discrete curvelet transform*, 1–6. [https://doi.org/10.1007/978-3-030-13435-3\\_12](https://doi.org/10.1007/978-3-030-13435-3_12)
- [46] Pennec E. L., Mallat S. (2005). *Directional multiscale modeling of images using curvelets*. *IEEE Trans. Image Process.* 14(4), 439–450. <https://doi.org/10.1109/TIP.2005.844103>
- [47] Hough P. (1962). *Method and means for recognizing complex patterns*. US Patent 3,069,654.
- [48] Ballard D. H. (1981). *A modified Hough transform for detecting circles and ellipses*. *Pattern Recognit.* 13(2), 111–122. [https://doi.org/10.1016/0031-3203\(81\)90009-1](https://doi.org/10.1016/0031-3203(81)90009-1)
- [49] Duda R. O., Hart P. E. (1972). *The Hough transform for line detection*. *Commun. ACM* 15(1), 11–15. <https://doi.org/10.1145/361237.361242>
- [50] Jain A. K. (1992). *Generalizing the Hough transform to detect arbitrary shapes*. *Pattern Recognit.* 25(3), 297–308. [https://doi.org/10.1016/0031-3203\(92\)90092-O](https://doi.org/10.1016/0031-3203(92)90092-O)
- [51] Kimme C., Sklansky D. B., J. (1981). *Robust shape detection using the Hough transform*. *Comput. Graph. Image Process.* 15(3), 235–246. [https://doi.org/10.1016/0146-664X\(81\)90054-7](https://doi.org/10.1016/0146-664X(81)90054-7)
- [52] Zhou Q., Zhou L. Z., Zhou S. (2010). *Statistical detection of defect patterns using Hough transform*. *IEEE Trans. Semicond. Manuf.* 23(3), 345–354. <https://doi.org/10.1109/TSM.2010.2052354>
- [53] Wang J., Gao P. F., R. X. (2020). *Machine vision intelligence for product defect inspection based on deep learning and Hough transform*. *J. Manuf. Syst.* 54, 271–278. <https://doi.org/10.1016/j.jmsy.2020.06.008>
- [54] Ge Q., Xu M. F., J. (2024). *Defect detection of industrial products based on improved Hough transform*. *J. Ind. Intell. Inf.* 3(4), 1–7.
- [55] Ajmi C., Jerbi S. E. F. A. (2020). *Automatic detection of weld defects based on Hough transform*. *J. Manuf. Process.* 48, 340–348. <https://doi.org/10.1016/j.jmapro.2020.03.005>
- [56] Davies R., Keller A. (2016). *Efficient Hough transform for circle detection in real-time applications*. *J. Real-Time Image Process.* 11(1), 25–33. <https://doi.org/10.1007/s11554-015-0487-6>
- [57] Keller J., Varghese T. (2006). *Robust edge detection using a novel Hough transform approach*. *Comput. Vis. Image Underst.* 102(2), 147–158. <https://doi.org/10.1016/j.cviu.2006.03.006>
- [58] Liu G. (2020). *Surface Defect Detection Methods Based on Deep Learning: A Brief Review*. In *Proceedings of the 2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, Guangzhou, China, 18–20 December 2020; pp. 200–203.
- [59] Zhang Y., Li G., Yang J., Wei X. (2021). *A review of machine learning-based defect detection methods*. *J. Manuf. Process.* 66, 481–495.

- [60] **Abbineni J., Thalluri O.** (2018). *Software defect detection using machine learning techniques*. In *Proceedings of the 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 11–12 May 2018; pp. 471–475.
- [61] **LeCun Y., Bengio Y., Hinton G.** (2015). *Deep learning*. *Nature* 521, 436–444.
- [62] **Krizhevsky A., Sutskever I., Hinton G. E.** (2012). *ImageNet classification with deep convolutional neural networks*. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA; pp. 1097–1105.
- [63] **Pan S. J., Yang Q.** (2010). *A survey on transfer learning*. *IEEE Trans. Knowl. Data Eng.* 22, 1345–1369.
- [64] **Yosinski J., Clune J., Bengio Y., Lipson H.** (2014). *How transferable are features in deep neural networks?* In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA; pp. 3320–3338.
- [65] **Howard J., Ruder S.** (2018). *Universal language model fine-tuning for text classification*. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA; pp. 328–339.
- [66] **Keshavan M. S., Janik P.** (2019). *Transfer learning for improving performance of deep learning models in medical image analysis*. *J. Med. Imaging* 6, 041201.
- [67] **Chen Y., Ding Y., Zhao F., Zhang E., Wu Z., Shao L.** (2021). *Surface defect detection methods for industrial products: A review*. *Appl. Sci.* 11, 7657.
- [68] **Iandola F. N., Han S., Moskewicz M. W., Ashraf K., Dally W. J., Keutzer K.** (2016). *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size*. *arXiv*, arXiv:1602.07360.
- [69] **Chen M., Yu L., Zhi C., Sun R., Zhu S., Gao Z., Ke Z., Zhu M., Zhang Y.** (2022). *Improved faster R-CNN for fabric defect detection based on Gabor filter with Genetic Algorithm optimization*. *Comput. Ind.* 134, 103551.
- [70] **Yu W., Lai D., Liu H., Li Z.** (2021). *Research on CNN algorithm for monochromatic fabric defect detection*. In *Proceedings of the 2021 6th International Conference on Image, Vision and Computing (ICIVC)*, Qingdao, China, 23–25 July 2021; pp. 20–25.
- [71] **Liu Q., Wang C., Li Y., Gao M., Li J.** (2022). *A fabric defect detection method based on deep learning*. *IEEE Access* 10, 4284–4296.
- [72] **Lee Y. J., Grauman K.** (2011). *Learning the easy things first: Self-paced visual category discovery*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, USA, 20–25 June 2011; pp. 1721–1730.
- [73] **Wei Y., Sun Q., Wang L., Liu C.** (2020). *Few-shot learning for surface defect classification*. *IEEE Access* 8, 32728–32736.

- [74] Redmon J., Farhadi A. (2018). *YOLOv3: An incremental improvement*. *arXiv*, arXiv:1804.02767.
- [75] He K., Zhang X., Ren S., Sun J. (2016). *Deep residual learning for image recognition*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
- [76] Simonyan K., Zisserman A. (2014). *Very deep convolutional networks for large-scale image recognition*. *arXiv*, arXiv:1409.1556.
- [77] Huang G., Liu Z., Van Der Maaten L., Weinberger K. Q. (2017). *Densely connected convolutional networks*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
- [78] Liu S., Jiang X., Yang J., Chen B. (2019). *A novel defect detection method based on feature extraction and recognition*. *Pattern Recognit. Lett.* 119, 158–165.
- [79] Zhang Y., Wallace B. (2015). *A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification*. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 397–409.
- [80] Bhuiyan Z. W., Haider S. A. R., Haque A., Uddin M. R., Hasan M. (2024). *IoT-based meat freshness classification using deep learning*. *IEEE Access* 12, 196047–196069.
- [81] Kingma D. P., Ba J. (2015). *Adam: A method for stochastic optimization*. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 7–9 May 2015.
- [82] Goodfellow I., Bengio Y., Courville A. (2016). *Deep learning*; MIT Press: Cambridge, MA, USA.
- [83] He Z., Shao H., Chen X. (2020). *A deep learning-based approach for defect detection in manufacturing*. *J. Intell. Manuf.* 31, 399–411.
- [84] Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. (2016). *Rethinking the inception architecture for computer vision*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
- [85] Grandini M., Bagli E., Visani G. (2020). *Metrics for multi-class classification: An overview*. *arXiv*, arXiv:2008.05756.
- [86] Lu X., Li X. (2016). *Transfer learning with convolutional neural networks for remote sensing image classification*. *IEEE Trans. Geosci. Remote Sens.* 54(10), 5763–5772. <https://doi.org/10.1109/TGRS.2016.2569839>
- [87] Tajbakhsh C., Gurudu J. Y. S. (2016). *Fine-tuning convolutional neural networks for biomedical image analysis: Actively and incrementally*. *IEEE Trans. Med. Imaging* 35(5), 1182–1195. <https://doi.org/10.1109/TMI.2016.2517689>

- [88] Zoph B., Vasudevan V., Shlens J. (2018). *Learning transferable architectures for scalable image recognition*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8697–8710. <https://doi.org/10.1109/CVPR.2018.00907>
- [89] He K., Zhang X., Sun J. (2016). *Deep residual learning for image recognition*. *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- [90] Wang Y., Li S. S., Li J. (2019). *Fabric surface defect detection based on convolutional neural networks*. *Text. Res. J.* 89(13), 2454–2466. <https://doi.org/10.1177/0040517519840706>
- [91] Talu M. F., Hanbay K., Varjovi M. H. (2024). *CNN-based fabric defect detection system on loom fabric inspection*. *Tekstil ve Konfeksiyon* 32(3). <https://doi.org/10.32710/tekstilvekonfeksiyon.1032529>
- [92] Vasagam S. N., Sornam M. (2022). *Intermittent leather defect detection based on ensemble algorithms derived from black hat transformation and Hough transformation*. In *Proceedings of the [Conference Name]*, pp. 35–45. [https://doi.org/10.1007/978-981-16-5655-2\\_418](https://doi.org/10.1007/978-981-16-5655-2_418)
- [93] Yıldız K. (2024). *Kumaş Hatalarının Isıl Görüntüleme ve Görüntü İşleme Teknikleri ile Tespit Edilmesi*. Doktora tezi, Marmara Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü.
- [94] Hanbay K. (2024). *Yuvarlak Örgü Makineleri için Görüntü İşleme Tabanlı Kumaş Hatası Tespit Sistemi*. Doktora Tezi, İnönü Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği.
- [95] Nawaf A. M. (2019). *Development of a New Software for Fabric Defect Detection and Classification Using Image Processing and Machine Learning Methods*. Yüksek Lisans Tezi, Çukurova Üniversitesi.
- [96] Raheja J. L., Ajay B., Chaudhary A. (2013). *Real-time fabric defect detection system on an embedded DSP platform*. *Optik* 124(21), 5280–5284. <https://doi.org/10.1016/j.ijleo.2013.03.038>
- [97] Wu Y., Zhang X., Fang A. F. (2020). *Automatic fabric defect detection using cascaded mixed feature pyramid with guided localization*. *Sensors (Basel)* 20(3), 871. <https://doi.org/10.3390/s20030871>
- [98] Ngan H. Y. T., Pang G. K. H., Yung N. H. C. (2008). *Motif-based defect detection for patterned fabric*. *Pattern Recognit.* 41(6), 1878–1894. <https://doi.org/10.1016/j.patcog.2007.11.014>
- [99] Zhao Y., Hao K., He H., Tang X., Wei B. (2020). *A visual long-short-term memory-based integrated CNN model for fabric defect image classification*. *Neurocomputing* 380, 259–270. <https://doi.org/10.1016/j.neucom.2019.10.067>
- [100] Mak K. L., Peng P., Yiu K. F. C. (2009). *Fabric defect detection using morphological filters*. *Image Vis. Comput.* 27(10), 1585–1592. <https://doi.org/10.1016/j.imavis.2009.03.007>
- [101] Li Y., Zhang D., Lee D. J. (2019). *Automatic fabric defect detection with a wide-and-compact network*. *Neurocomputing* 329, 329–338. <https://doi.org/10.1016/j.neucom.2018.10.070>

- [102] Bissi L., Baruffa G., Placidi P., Ricci E., Scorzoni A., Valigi P. (2013). *Automated defect detection in uniform and structured fabrics using Gabor filters and PCA*. *J. Vis. Commun. Image Represent.* 24(7), 838–845. <https://doi.org/10.1016/j.jvcir.2013.05.011>
- [103] Habib M. T., Faisal R. H., Rokonzaman M., Ahmed F. (2024). *Automated fabric defect inspection: A survey of classifiers*. *Int. J. Found. Comput. Sci. Technol. (IJFCST)* 4(1).
- [104] Rasheed A., Zafar B., Rasheed A., Ali N., Sajid M., Dar S. H., Habib U., Shehryar T., Mahmood M. T. (2020). *Fabric defect detection using computer vision techniques: A comprehensive review*. <https://doi.org/10.1155/2020/8189403>
- [105] Böttger T. (2016). *Real-time texture error detection on textured surfaces with compressed sensing*. *Pattern Recognit. Image Anal.* 26, 88–94. <https://doi.org/10.1134/S1054661816010053>
- [106] Ouyang W., Hou J., Xu B. (2019). *Fabric defect detection using activation layer embedded convolutional neural network*. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2019.2913620>

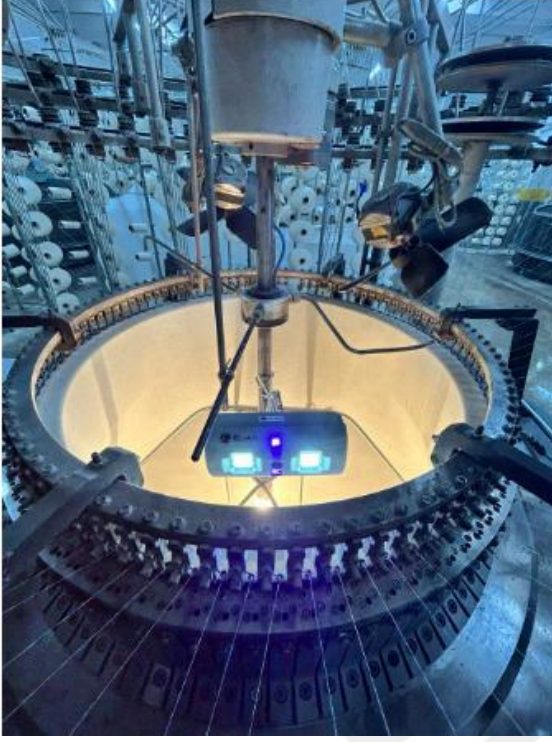


## **EKLER**

### **EK A: Saha Test Görüntüleri**



## EK A: Saha Test Görüntüleri



(a)



(b)



(c)



(d)

Şekil A.1 : Saha Test Görüntüleri: (a)Tek Plaka. (b)Çift Plaka. (c)Üstten Bağlantı (d) tek plaka

## ÖZGEÇMİŞ

**Ad-Soyad : Mehmet ERDOĞAN**

### ÖĞRENİM DURUMU:

- **Lisans** : 2014, Yıldız Teknik Üniversitesi, Elektrik Elektronik Fakültesi, Elektrik Mühendisliği Bölümü
- **Yükseklisans** : 2017 , İstanbul Teknik Üniversitesi, Elektrik Elektronik Fakültesi, Elektrik Mühendisliği Bölümü

### DOKTORA TEZİNDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Erdoğan M., Doğan M. (2022).** Ayrık Eğrilik Algoritması ile Kumaş Kalite Kontrol Sistemi Geliştirilmesi. 23. *Otomatik Kontrol Türk Milli Komitesi Kongresi (TOK)*, 15–18 Eylül, 2022 Elazığ, Türkiye.
- **Erdoğan M., Doğan M. (2023).** Kumaş Kalite Kontrol İşleminde Gabor-Eğrilik Algoritması ve Yolov5 ile Karşılaştırılması. 24. *Otomatik Kontrol Türk Milli Komitesi Kongresi (TOK)*, 14–16 Eylül, 2023 İstanbul, Türkiye.
- **Erdoğan, M.; Doğan, M.** Enhanced Curvature-Based Fabric Defect Detection: A Experimental Study with Gabor Transform and Deep Learning. *Algorithms* 2024, 17, 506. <https://doi.org/10.3390/a17110506>

### DİĞER YAYINLAR, SUNUMLAR VE PATENTLER:

- 1501 Teydeb 3220359 ‘Örme Kumaş Makinelerinde Kumaş Hataları ve Anomalileri Gerçek Zamanlı Tespit Edebilen Görüntü İşleme Tabanlı Kalite Kontrol Sisteminin Geliştirilmesi’