



**“Ahşap Zeminler Üzerine Anomali Tespiti”**

**Projesi Final Raporu**

Eskişehir Osmangazi Üniversitesi

Neural Network Dersi

Baturhan Çağatay 152120211060

Emre Güner 152120211090

Doç. Dr. Eyüp Çınar

2024-2025

## İçindekiler

1.	Giriş.....	3
1.1	Proje Konusu ve Kapsamı .....	3
1.2	Problemin Tanımı .....	3
1.3	Projenin Amacı .....	4
1.4	Kullanım Alanları .....	4
2.	Yöntem.....	4
2.1	Kullanılan Veri Seti .....	4
2.2	Veri Seti Ön Hazırlık .....	4
2.3	Veri Arttırımı .....	7
2.4	Modellerin Değerlendirilme Metrikleri.....	10
2.5	Modellerin Eğitilmesi ve Sonuçların Analizi.....	11
2.5.1.	Efficient AD .....	11
2.5.2	Fast Flow .....	16
2.5.3	UniNet.....	20
2.5.4	PBAS.....	23
2.5.5	SimpleNet Modeli .....	27
2.5.6	INP-Former Modeli.....	33
2.5.7	Voting Modeli .....	38
2.6	Model Kıyaslama .....	40
3.	Sistem Mimarisi .....	43
3.1	Genel Bakış.....	43
3.2	Mimari Özellikleri.....	44
3.3	Kullanıcı Arayüz .....	44
4.	Gelecekte Yapılacak Çalışmalar .....	47
4.1	Model Ağırlıklarının Sıkıştırılması .....	47
4.2	Model Çalışma Süresinin Azaltılması.....	48
4.3	Gerçek Zamanlı Uygulamaya Uyumlaştırma.....	48
5.	Ekler ve Proje Erişim Bilgileri .....	48
5.1	Proje Tanıtım Videosu .....	48
5.2	GitHub ve Drive Bağlantıları .....	48
5.3	Sonuçların Tekrar Üretimi İçin Adımlar .....	49
6.	Sonuç.....	50
7.	KAYNAKÇA .....	50

## 1. Giriş

Endüstriyel üretim süreçlerinde ürün kalitesinin sürdürülebilirliği, hem rekabet avantajı sağlamak hem de müşteri memnuniyetini artırmak açısından kritik öneme sahiptir. Bu süreçlerin ayrılmaz bir parçası olan **görsel denetim**, üretim hattındaki kusurlu ürünlerin tespiti için hayati bir rol üstlenmektedir. Ancak özellikle **ahşap** gibi doğal ve yapısal olarak heterojen malzemelerde, geleneksel denetim yöntemleriyle kusur tespiti hem zaman alıcı hem de operatörün yorumuna bağlı olarak değişkenlik gösterebilen subjektif bir süreçtir.

Bu noktada, **bilgisayarla görme** (Computer Vision) ve **makine öğrenmesi** (Machine Learning) gibi yapay zekâ temelli yöntemler, üretim hatlarında **otomatik ve tutarlı kalite kontrol sistemlerinin** geliştirilmesine olanak sağlamaktadır. Özellikle son yıllarda yaygınlaşan **denetimsiz öğrenme (Unsupervised Learning)** tabanlı yaklaşımlar, yalnızca kusursuz (normal) örnekler üzerinden öğrenerek etiketli veriye olan ihtiyacı ortadan kaldırmakta ve gerçek dünya uygulamaları için pratik çözümler sunmaktadır.

### 1.1 Proje Konusu ve Kapsamı

Bu proje, endüstriyel üretim ortamlarında ahşap yüzeylerin kalite kontrolünde karşılaşılan zorluklara çözüm sunmak amacıyla geliştirilmiştir. Projenin temel konusu; denetimsiz öğrenme yaklaşımını kullanarak, yalnızca kusursuz ahşap görüntüleriyle eğitilen bir yapay zekâ modelinin, daha önce hiç karşılaşmadığı kusurları tespit edebilmesidir. Projede **MVTec Anomaly Detection (MVTec AD)** veri setinin **wood** (ahşap) alt kümesi kullanılmıştır. Model, hem görüntü sınıflandırma (anomalili/normal) hem de piksel düzeyinde segmentasyon (kusurlu bölgelerin belirlenmesi) gerçekleştirmektedir. Bu kapsamda geliştirilen sistem; veri ön işleme, model eğitimi, eşik belirleme, performans değerlendirmesi ve görselleştirme adımlarından oluşmaktadır.

### 1.2 Problemin Tanımı

Ahşap yüzeylerin doğal ve düzensiz yapısı, görsel denetim süreçlerinde klasik yöntemlerin yetersiz kalmasına neden olmaktadır. Özellikle farklı ışık koşulları, renk varyasyonları ve düzensiz desenler gibi faktörler, geleneksel görüntü işleme yöntemleriyle doğru ve tutarlı bir şekilde kusur tespiti yapılmasını zorlaştırmaktadır. Ayrıca manuel denetim süreçleri hem insan hatasına açıktır hem de büyük ölçekte sürdürülebilir değildir. Etiketli kusurlu veri elde etmenin zorluğu da denetimli makine öğrenmesi yöntemlerinin uygulanabilirliğini sınırlandırmaktadır. Bu nedenle, yalnızca normal örneklerle çalışabilen ve üretim hattına entegre edilebilecek **otomatik, esnek ve yorumlanabilir** bir anomali tespit sistemi büyük bir ihtiyaç haline gelmiştir.

### 1.3 Projenin Amacı

Projenin temel amacı, yalnızca kusursuz görüntülerle eğitilen bir yapay zekâ modelinin, test aşamasında kusurlu görüntülerle karşılaştığında **yüksek doğrulukla (özellikle F1 skor baz alınacak) anomali tespiti yapabilmesini sağlamaktır**. Bu doğrultuda, modelin hem **görüntü seviyesinde** sınıflandırma yapması hem de **piksel seviyesinde** kusurlu bölgeleri segmente edebilmesi hedeflenmektedir. Modelin gerçek zamanlı sistemlere entegre edilebilecek şekilde optimize edilmesi, eşik değerinin otomatik olarak belirlenmesi ve sonuçların hem sayısal hem görsel olarak değerlendirilebilmesi de projenin önemli hedefleri arasındadır.

### 1.4 Kullanım Alanları

Geliştirilen sistem, başta **mobilya, parke ve yapı malzemeleri üretimi** olmak üzere ahşap bazlı üretim yapan tüm endüstrilerde kullanılabilir. Bununla birlikte, benzer yüzey çeşitliliğine sahip malzemelerde (örneğin mermer, kumaş, deri) anomali tespiti yapılması gereken diğer üretim alanlarına da kolayca uyarlanabilir. Ayrıca sistemin, **manuel denetim süreçlerinin otomasyonu, kalite standardizasyonunun sağlanması, üretim sürecinde zaman/maliyet tasarrufu** gibi katkılar sunması hedeflenmektedir.

## 2. Yöntem

Bu bölümde, ahşap yüzey verilerinin hazırlanması, veri artırma teknikleri, kullanılan değerlendirme metrikleri ve farklı anomali tespit modellerinin eğitim süreçleri özetlenmiştir. Ayrıca, model performansları karşılaştırılmış ve toplu karar verme (voting) yaklaşımıyla genel başarı artırılmıştır.

### 2.1 Kullanılan Veri Seti

Bu çalışmada, endüstriyel yüzey denetimi için sıklıkla kullanılan **MVTec Anomaly Detection (MVTec AD)** veri setinin **wood (ahşap)** alt kümesi kullanılmıştır. Veri kümesinde **70 adet kusursuz (good)** ve **71 adet kusurlu (defect)** görüntü bulunmaktadır. Kusurlu görüntüler, yüzeydeki çatlak, renk bozulması ve lif kaymaları gibi farklı anomali türlerini içermektedir. Model yalnızca "good" sınıfıyla eğitilmiş, "defect" sınıfı test aşamasında değerlendirme için kullanılmıştır.

### 2.2 Veri Seti Ön Hazırlık

Modelin yalnızca anlamlı anomalilere odaklanabilmesi ve öğrenme sürecinin daha verimli hale getirilebilmesi amacıyla, **MVTec AD veri setinin wood (ahşap)** alt kümesindeki görüntüler üzerinde çeşitli **ön işleme (preprocessing)** adımları gerçekleştirilmiştir. Veri setindeki orijinal görüntüler RGB (renkli) formatındadır. Ancak

bu projede, renk bilgisinin model için ikincil öneme sahip olduğu, asıl belirleyicinin doku yapısı olduğu varsayılmıştır. Bu nedenle, tüm görüntüler **gri tonlamalı (grayscale)** forma dönüştürülerek işlenmiştir.

Bu yaklaşım sayesinde:

- Modelin işlem yükü azaltılmış,
- Üç kanal yerine tek kanal üzerinden daha yalın bir öğrenme sağlanmış,
- Renk varyasyonlarından kaynaklanan belirsizlikler elimine edilmiştir.

Böylece modelin **doğrudan yapısal bozulmalara** odaklanması ve daha tutarlı sonuçlar üretmesi hedeflenmiştir.

#### **Uygulanan Ön İşleme Adımları:**

##### **1. Görüntülerin Yüklenmesi:**

Google Drive'a yüklenen veri setine Google Colab üzerinden erişilmiş ve her bir RGB görüntü, cv2.IMREAD\_GRAYSCALE parametresiyle gri tonlamalı olarak yüklenmiştir.

##### **2. Gaussian Blur:**

Gürültüyü azaltmak ve daha homojen yüzeyler elde etmek amacıyla Gaussian bulanıklık filtresi uygulanmıştır.

##### **3. Otsu Eşikleme:**

Bulanıklık uygulanmış görüntülere Otsu yöntemiyle eşikleme yapılarak, potansiyel anomalilerin öne çıkarıldığı ikili (binary) maskeler elde edilmiştir.

##### **4. Bitwise Maskeleye:**

Elde edilen maskeler, gri tonlamalı görüntülere bitwise AND işlemiyle uygulanmış ve yalnızca **anomalilik potansiyeli taşıyan bölgeler** izole edilmiştir.

##### **5. Yeniden Boyutlandırma (Resize):**

Tüm görüntüler 256×256 piksel boyutuna ölçeklenmiştir.

- Eğitim görüntülerinde cv2.INTER\_AREA interpolasyonu kullanılarak kalite korunmuştur.

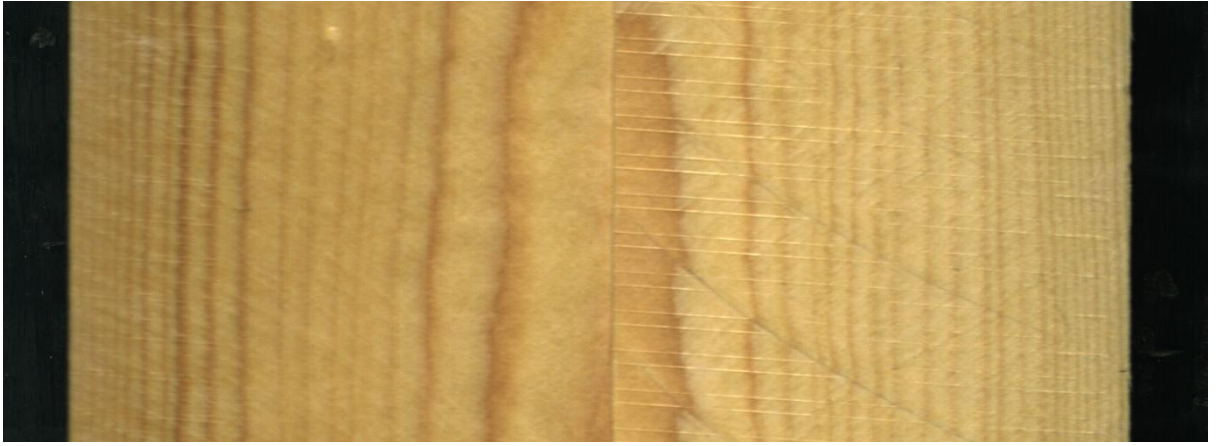
- Segmentasyon maskelerinde cv2.INTER\_NEAREST interpolasyonu kullanılarak sınıf bilgisi bozulmadan aktarım sağlanmıştır.

**6. Normalizasyon:**

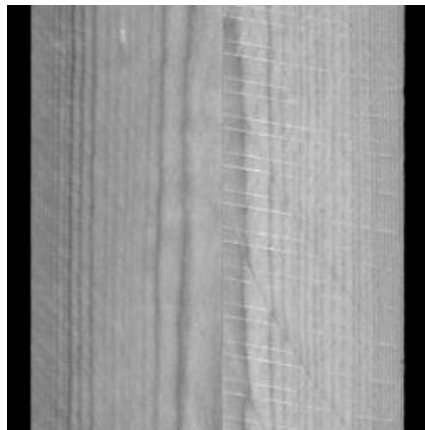
Piksel değerleri 0–1 aralığına normalize edilmiş, ardından görselleştirme ve kayıt için tekrar 0–255 aralığına dönüştürülmüştür.

**7. Kaydetme:**

İşlenmiş görüntüler ayrı bir klasöre aktarılmış ve eğitim için hazır hale getirilmiştir.



*Şekil 1 – Ön işleme yapılmamış – orijinal veri seti içinde train 69.jpg*



*Şekil 2 - Ön işleme yapılmış 69.jpg*

Bu işlemler sonucunda model, Şekil 1'deki orijinal görüntü yerine Şekil 2'deki sadeleştirilmiş ve anomali odaklı hale getirilmiş girdi ile eğitilerek daha kararlı ve başarılı sonuçlar verebilir hale gelmiştir.

### 2.3 Veri Arttırımı

Makine öğrenmesi modellerinin genelleme yeteneğini artırmak ve aşırı öğrenme (overfitting) riskini azaltmak amacıyla eğitim verisi üzerinde çeşitli **veri artırma (data augmentation)** teknikleri uygulanmıştır. Ahşap yüzey verileri gibi doğası gereği sınırlı çeşitliliğe sahip görsellerde, yapay olarak oluşturulmuş varyasyonlar sayesinde modelin farklı kusur türlerini tanıma kabiliyeti güçlendirilmiştir. Bu çalışmada, *wood* veri setinin train/good alt klasöründeki kusursuz görüntülere çeşitli dönüşümler uygulanarak yeni örnekler üretilmiştir.

İlk aşamada, her bir görüntüye temel dönüşümler uygulanarak varyasyonlar elde edilmiştir:

- **Yatay çevirme (Horizontal Flip)**

Görüntü yansıması üzerinden farklılık yaratmak amacıyla yatay ekseninde çevrilmiştir.

- **Rastgele döndürme (Random Rotation)**

Görüntüler  $\pm 30$  dereceye kadar döndürülerek farklı bakış açılarının modele kazandırılması hedeflenmiştir.

- **Renk bozulmaları (Color Jitter)**

Parlaklık, kontrast ve doygunluk gibi özellikler küçük oranlarda değiştirilmiştir (brightness=0.3, contrast=0.3, saturation=0.3). Bu sayede modelin ışık değişikliklerine karşı daha dayanıklı olması sağlanmıştır.

Bu dönüşümler her bir orijinal görsel üzerine uygulanarak yeni versiyonlar .aug\_ önekiyle kaydedilmiştir.

### Varyant Tabanlı Çoğaltma

İkinci aşamada, her görüntüye farklı kombinasyonlardan oluşan **beş ayrı augmentasyon varyantı** uygulanmıştır. Bu sayede her görselden beş farklı türev elde edilmiştir:

1. Yatay çevirme +  $15^\circ$  döndürme
2. Dikey çevirme + Parlaklık artırımı
3.  $25^\circ$  döndürme + Kontrast artırımı
4. Yatay çevirme + Doygunluk artırımı
5.  $\pm 30^\circ$  döndürme + Karma renk değişiklikleri

Bu çeşitlilik, modelin hem pozisyonel hem de renk varyasyonlarına karşı daha sağlam hale gelmesini amaçlamaktadır.

### İleri Seviye Dönüşümler

Üçüncü ve son aşamada, daha sofistike görüntü işleme teknikleri kullanılmıştır. Her görüntü için beş farklı dönüşüm aşağıda açıklanmıştır:

1. Yatay çevirme (hflip)
2. Hafif rastgele döndürme ( $\pm 10^\circ$ )
3. Parlaklık artırımı (brightness jitter)
4. Gaussian Bulanıklaştırma (Blur)

Görüntü üzerine radius=1 bulanıklık filtresi uygulanarak pürüzlü bölgeler yumuşatılmıştır.

5. Gaussian Gürültü Eklenmesi (Noise)

Görüntüye mean=0, std=0.05 parametreleriyle gürültü eklenmiş ve bu işlem `rch.Tensor` düzeyinde yapılmıştır.

Bu adım, modelin bozuk, bulanık ya da gürültülü görüntülerle karşılaştığında bile anomali tespit yeteneğini korumasını hedeflemektedir.

Bu üç aşamalı augmentasyon stratejisi ile:



- Eğitim veri sayısı artırılmış,
- Modelin genelleme yeteneği güçlendirilmiş,
- Işık, pozisyon, renk ve parazit değişimlerine karşı dayanıklı hale gelmesi sağlanmıştır.

Elde edilen tüm augmented görüntüler orijinal train/good klasöründe augX\_ örnekleriyle saklanarak eğitim sürecine dahil edilmiştir.

Efficient AD ile hem orijinal hem augmented edilmiş veri ile 140 epoch üzerinden eğitilmiştir.

*Tablo 1 Augmentation Sonuç Kıyas Tablosu*

Eğitim Verisi	Toplam Resim Sayısı	F1 Skoru (Image Level)	AUC Skor	Ortalama IOU
Orijinal	70	0.9103	0.9327	0.3175
350 Sanal + 70 Gerçek Resim	420	0.9054	0.9443	0.1308
70 Sanal + 70 Gerçek Veri	140	0.9252	0.9586	0.3090

**-350 augmentasyonlu görüntü kullanımı**, IoU gibi segmentasyon açısından kritik metrikleri ciddi şekilde düşürdüğü için tercih edilmemiştir. Yüksek sayıda yapay veri, modelin karar sınırlarını bulanıklaştırmıştır.

**-Orijinal veriyle eğitim**, düşük veri miktarına rağmen sağlam ve dengeli sonuçlar vermiştir.

**-70+70 senaryosu** ise en yüksek F1 ve AUC değerlerini üretmiş ancak IoU açısından orijinale göre biraz geride kalmıştır. Buna rağmen genel anlamda başarılı sayılabilir.

Tablo-1'deki sonuçlar incelendiğinde, veri artırımı (augmentation) uygulamalarının model performansı üzerinde sınırlı bir etki yarattığı görülmektedir. Özellikle -350 augmentasyonlu görüntü kullanımı, segmentasyon başarısını yansıtan IoU gibi kritik

metriklerde ciddi düşüşlere neden olmuş ve bu senaryo tercih edilmemiştir. Bu durum, aşırı sayıda yapay verinin modelin karar sınırlarını bulanıklaştırarak genelleme kabiliyetini zayıflattığını göstermektedir.

Öte yandan, yalnızca orijinal verilerle yapılan eğitim, veri sayısının sınırlı olmasına rağmen daha sağlam ve dengeli sonuçlar üretmiştir. 70 orijinal + 70 augment edilmiş görüntü ile yapılan eğitimde ise, en yüksek F1 skoru ve AUC değerlerine ulaşılmış; ancak segmentasyon açısından (IoU) orijinal veri setine kıyasla bir miktar geride kalmıştır. Buna rağmen bu senaryo, genel başarı düzeyi açısından en dengeli ve tercih edilebilir yapı olarak değerlendirilmiştir.

## 2.4 Modellerin Değerlendirilme Metrikleri

Anomali tespiti modellerinin başarımını değerlendirebilmek için hem **görüntü (image-level)** hem de **piksel (pixel-level)** düzeyde çeşitli performans metrikleri kullanılmaktadır. Bu metrikler, modelin yalnızca bir görüntünün anormal olup olmadığını tespit etme yeteneğini değil, aynı zamanda **anomalili bölgelerin konumunu ve sınırlarını ne kadar doğru belirleyebildiğini** de ölçmektedir. Bu kapsamda F1 skoru, IoU (Intersection over Union), AUC (Area Under Curve), precision, recall ve confusion matrix gibi ölçütler analiz edilmiştir.

### 2.4.1 Image-Level F1 Skoru

Image-level F1 Skoru, bir görüntünün genel olarak anomali içerip içermediğine yönelik yapılan ikili (binary) sınıflandırma performansını ölçer. Bu metrik, modelin anomaliyi “pozitif sınıf” olarak ele aldığı senaryolarda, doğru anomali tespitleriyle yanlış pozitif ve negatifleri dengeleyerek başarıyı değerlendirir. F1 skoru, precision (kesinlik) ve recall (duyarlılık) metriklerinin harmonik ortalaması olarak hesaplanır ve Denklem 1’de gösterilen formülle ifade edilir. Bu sayede, özellikle dengesiz sınıf dağılımına sahip veri kümelerinde adil ve dengeli bir performans ölçümü sağlar.

*Denklem 1: F1 Skoru Hesaplama*

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

#### 2.4.2 Pixel-Level F1 ve IoU Skorları

Modelin segmentasyon başarısını ölçmek için kullanılır. Her pikselin doğru sınıflanıp sınıflanmadığına bakılır.

- **Pixel-level F1:**

Her pikselin doğru/yanlış etiketlenmesine göre hesaplanan F1 skorudur.

- **Intersection over Union (IoU):**

Tahmin edilen maske ile gerçek (ground truth) maskenin ne kadar çakıştığını ölçer.

Bu metrikler özellikle kontur ve lokal kusur tespitinde kritiktir.

#### 2.4.3 AUC (Area Under Curve)

ROC eğrisinin altında kalan alanı ifade eder. Modelin farklı eşiklerdeki duyarlılığına dair genel bir başarı ölçüsüdür. 1'e ne kadar yakınsa, modelin genel ayrıştırma başarısı o kadar iyidir.

### 2.5 Modellerin Eğitilmesi ve Sonuçların Analizi

Bu bölümde, proje kapsamında kullanılan farklı anomali tespit modellerinin eğitim süreçleri detaylı şekilde ele alınmaktadır. Her modelin eğitiminde kullanılan mimari yapılar, veri işleme adımları, hiperparametre ayarları ve özel optimizasyon stratejileri açıklanmıştır. Ayrıca, yapılan kod değişiklikleri, kullanılan backbone ağlar ve her bir modelin eğitim performansı karşılaştırmalı olarak sunulmuştur. Aşağıdaki alt başlıklarda, EfficientAD, FastFlow, UniNet, PBAS, SimpleNet ve INP-Former gibi modellerin eğitim süreçleri ve elde edilen sonuçlar ayrı ayrı incelenmektedir.

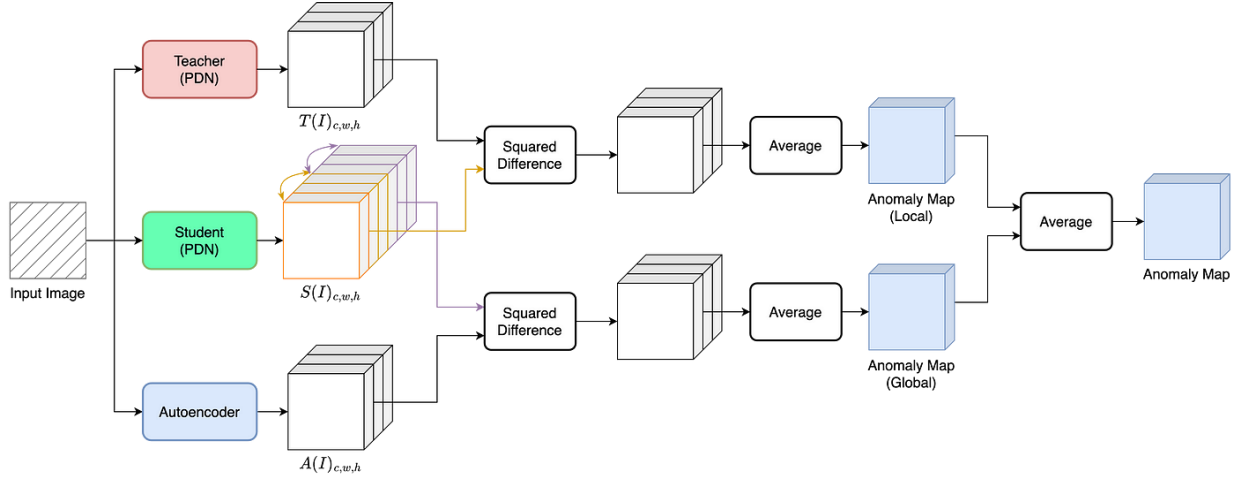
#### 2.5.1. Efficient AD

Bu bölümde Efficient AD modeli anlatılacaktır.

##### 2.5.1.1 Mimarisi

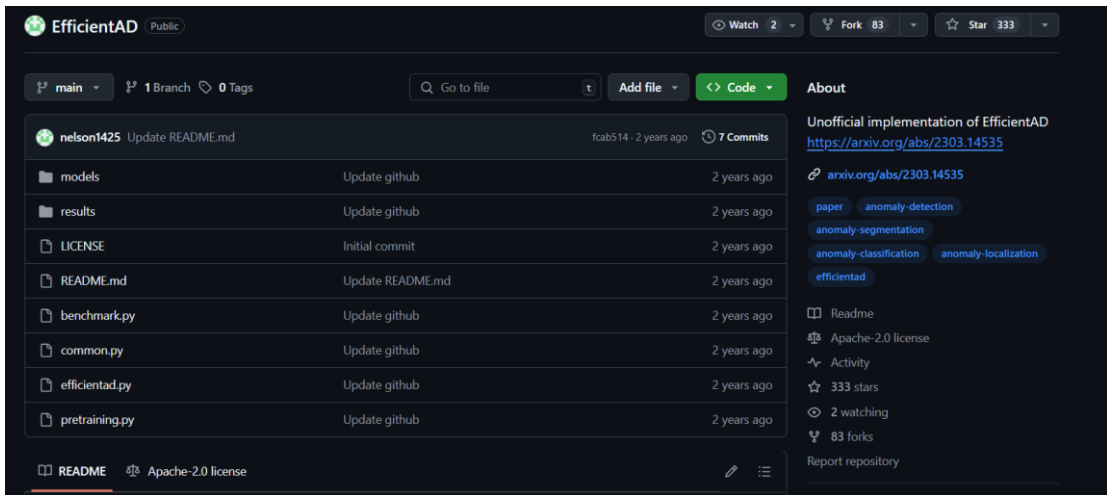
EfficientAD modeli, **öğrenci-öğretmen (student-teacher)** mimarisi temel alınarak tasarlanmıştır. Bu yapıda, öğretmen ağı yalnızca kusursuz (defect-free) görüntülerle eğitilirken, öğrenci ağı bu öğretmeni taklit etmeye çalışır. Ancak test

aşamasında, öğrenci ağı öğretmeni yeterince taklit edemediği bölgeler **anomali** olarak tanımlanır. Bu mimari yapının genel şematik gösterimi **Şekil 3**'te sunulmuştur



Şekil 3 Efficient AD Model Mimarisi

Modelin temel bileşenlerinden biri olan **Patch Description Network (PDN)**, görüntüyü küçük parçalara ayırarak hızlı ve etkili bir biçimde özellik çıkarımı gerçekleştirir. Buna ek olarak, modele entegre edilen **autoencoder** yapısı, görüntülerin hem yapısal hem de anlamsal özelliklerini öğrenerek, lokal ve global anomaliyi hassas biçimde tespit edebilir hale gelmesini sağlar. Bu iki bileşenin çıktıları birleştirilerek, **yüksek doğrulukta anomali tespiti** gerçekleştirilmektedir.



Şekil 4 Nelson- EfficientAD Modeli

Şekil 4 ile görebileceğiniz gibi Github platformu üzerinden EfficientAD modelini clone edip bu kodlar üzerinden değişimler yapıldı.

Değişim yapılan dosyalar “efficientad.py” ve “common.py” dosyalarıdır.

#### 2.5.1.2 Yapılan Değişimler:

- Her 10,000 iterasyonda validation ve ara test aşaması eklendi.
- AUC, F1, Precision, Recall ve Confusion Matrix gibi kapsamlı metrik hesaplamaları ve bu hesaplamalarının log olarak bastırılması eklendi.
- Normalize için quantile hesaplama ve ara/son kaydetme işlemleri eklendi.
- IoU hesaplamaları için ground-truth maskeler okunuyor.
- Dosya uzantıları kendi dosya uzantılarımız ile uyumlu hale getirildi.

#### 2.5.1.3 Sonuçlar ve Yorumlar:

Modelin eğitim süreci sonunda, 1500 epoch tamamlandığında elde edilen performans metrikleri Şekil-5’de sunulmuştur.

```
IoU Threshold Optimization Results:
Threshold: 0.10 | Mean IoU: 0.2724
Threshold: 0.20 | Mean IoU: 0.1989
Threshold: 0.30 | Mean IoU: 0.1219
Threshold: 0.40 | Mean IoU: 0.0626
Threshold: 0.50 | Mean IoU: 0.0306
Threshold: 0.60 | Mean IoU: 0.0134
Threshold: 0.70 | Mean IoU: 0.0065
Threshold: 0.80 | Mean IoU: 0.0025
Threshold: 0.90 | Mean IoU: 0.0010

✓ Best IoU: 0.2724 @ Threshold = 0.10

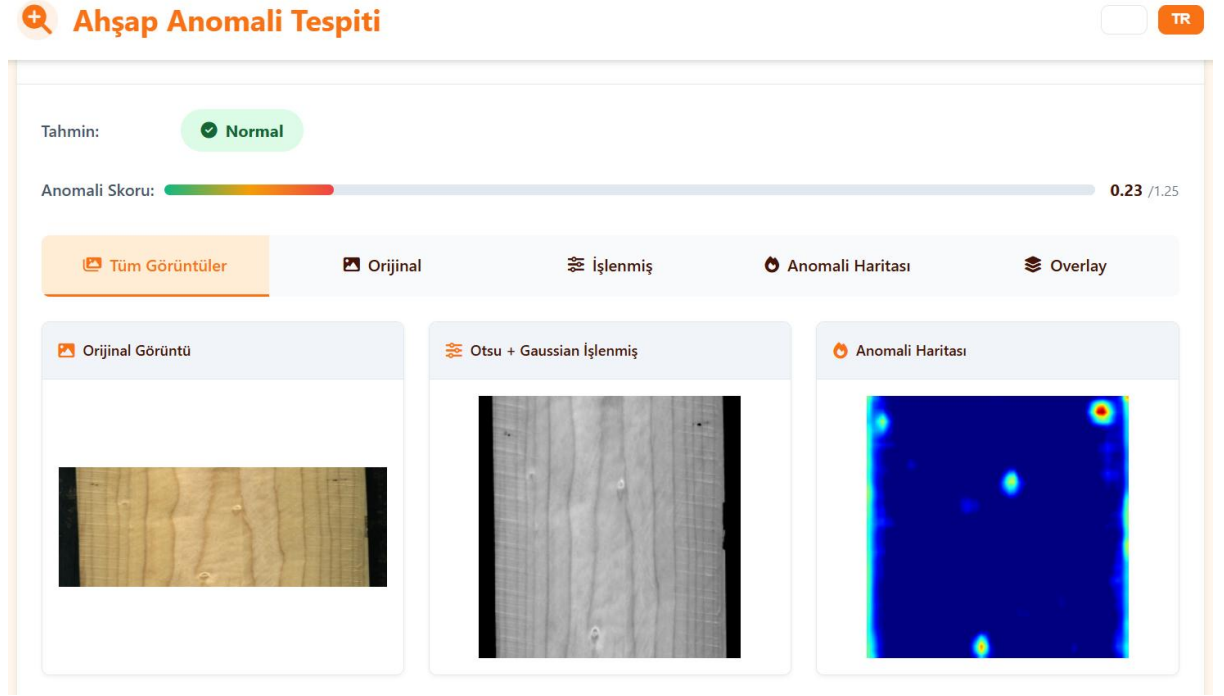
✓ Anomaly Detection Sonuçları
AUC Score      : 0.9171
F1 Score       : 0.8904
Precision      : 0.8553
Recall         : 0.9286
Optimal Threshold : 0.1834
Confusion Matrix:
[[59 11]
 [ 5 65]]

✓ Final Anomaly Detection Sonuçları
Final AUC Score      : 0.9171
Final F1 Score       : 0.8904
Final Precision      : 0.8553
Final Recall         : 0.9286
Final Confusion Matrix :
[[59 11]
 [ 5 65]]
```

Şekil 5 Efficient AD Performans Metrikleri

Modelin değerlendirme sonuçları, hem sınıflandırma hem de segmentasyon görevlerinde yüksek başarı sergilediğini ortaya koymaktadır. Image-level anomali tespitinde model, **%91.7 AUC**, **%89.0 F1 skoru**, **%85.5 precision** ve **%92.9 recall** ile güçlü ve dengeli

bir performans göstermiştir. Bu değerler, modelin hem anomaliyi kaçırma oranının düşük olduğunu hem de yanlış alarm üretme eğiliminin sınırlı olduğunu göstermektedir.

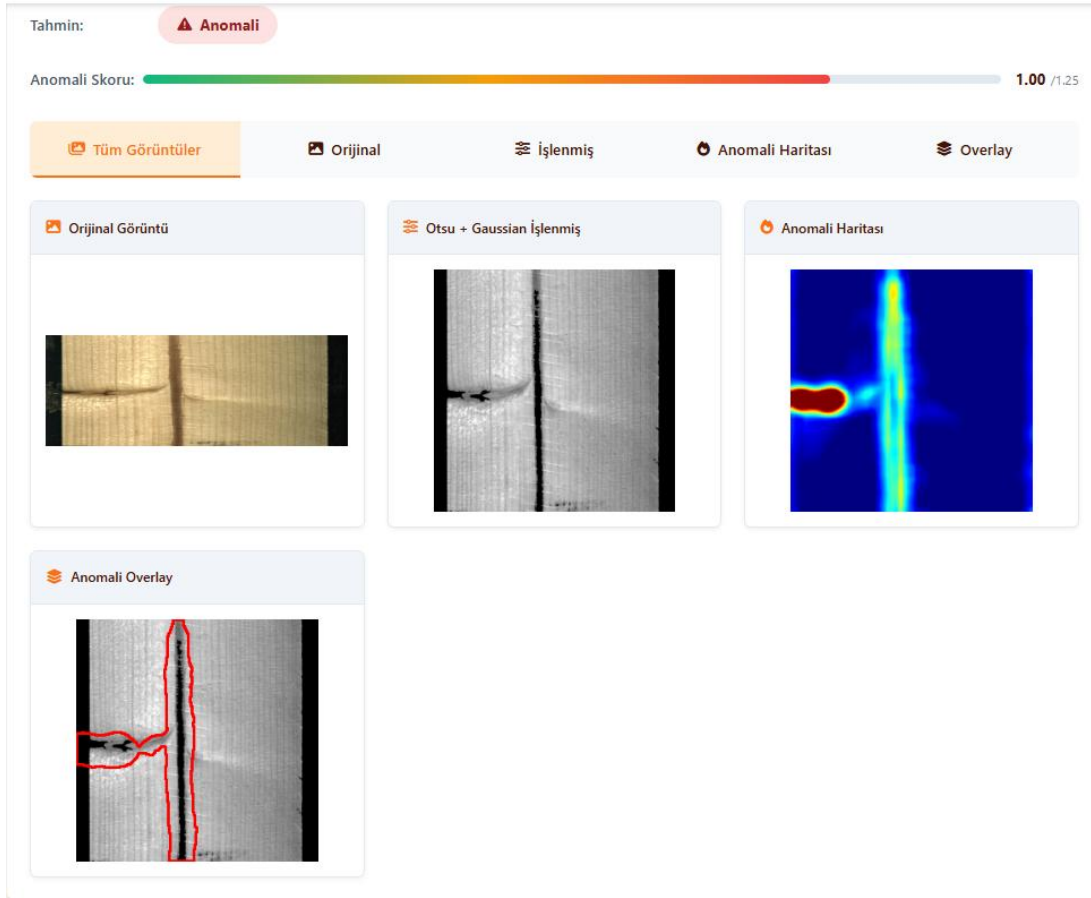


Şekil 6 EfficientAD Modelinin Arayüz Üzerinden Çıktısı-I

**Optimal threshold değeri 0.1834** olarak belirlenmiş ve bu eşik ile oluşturulan **confusion matrix**, 65 doğru pozitif ve sadece 5 yanlış negatif tespitle yüksek hassasiyeti desteklemektedir. Segmentasyon başarımına bakıldığında ise en yüksek **IoU değeri 0.2724**, **threshold = 0.10** için elde edilmiş; bu da düşük eşik seviyelerinde modelin anomalili bölgeleri daha iyi yakalayabildiğini göstermektedir. Sonuçlar, modelin anomali tespiti görevinde hem karar doğruluğu hem de lokalizasyon açısından tutarlı ve etkili bir performans sunduğunu kanıtlamaktadır. Overlay Thresholdu için 0.3825 değeri bulunmuştur.

Şekil 6 ve Şekil 7, EfficientAD modelinin test aşamasında elde ettiği sonuçların kullanıcı arayüzü üzerinde nasıl sunulduğunu göstermektedir. Şekil-6’da görebileceğimiz gibi örnek test görüntüsünde, modelin tahmini "**Normal**" olarak belirlenmiş ve görüntüye karşılık gelen **anomali skoru 0.23 / 1.25** olarak hesaplanmıştır. Bu skor, önceden belirlenen optimal eşik değerinin altında kaldığı için sistem doğru şekilde “normal” sınıfına

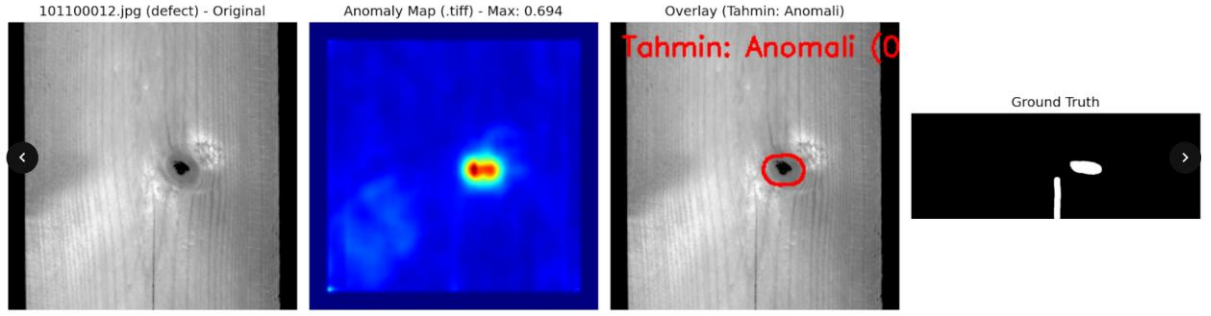
karar vermiştir. Ayrıca Şekil-7’de ise bu sefer anomali tespitinin nasıl yapıldığı gözlemlenmiştir.



Şekil 7 EfficientAD Modelinin Arayüz Üzerinden Çıktısı – II

Şekil 8’de, modelin ürettiği anomali haritası ile gerçek etiketler (Ground Truth) arasında yapılan karşılaştırma sunulmuştur. Bu kıyaslama, modelin yalnızca sınıflandırma düzeyinde değil, aynı zamanda **segmentasyon düzeyinde de başarılı ve tutarlı tahminlerde bulunduğunu** göstermektedir. Gerçek kusurlu bölgeler ile modelin vurguladığı bölgelerin çakışması, modelin anomaliyi doğru lokalize ettiğine ve sahte

pozitif/negatif oranlarının düşük tutulduğuna işaret etmektedir.



Şekil 8 EfficientAD Modeli Anomaly Map ile Ground Truth Kıyası

Sonuç olarak, EfficientAD modeli, hem sınıflandırma hem de segmentasyon görevlerinde yüksek performans sergilemekte, anomali tespitinde güvenilir ve etkili bir çözüm sunmaktadır.

### 2.5.2 Fast Flow

Bu bölümde Fast Flow modeli anlatılacaktır.

#### 2.5.2.1 Mimarisi

FastFlow modeli, denetimsiz anomali tespiti için tasarlanmış akış tabanlı bir derin öğrenme mimarisidir. Model, herhangi bir özellik çıkarıcıdan elde edilen temsilleri, ardışık  $3 \times 3$  ve  $1 \times 1$  konvolüsyonlardan oluşan flow adımları ile işler. Her flow adımı, giriş özelliklerini kanal bazında normalize edip ikiye böler; bir parçadan ölçek ve kaydırma parametreleri çıkarılırken diğerine affine dönüşüm uygulanır. Bu mimarinin görsel şeması Şekil-9'da sunulmuştur.



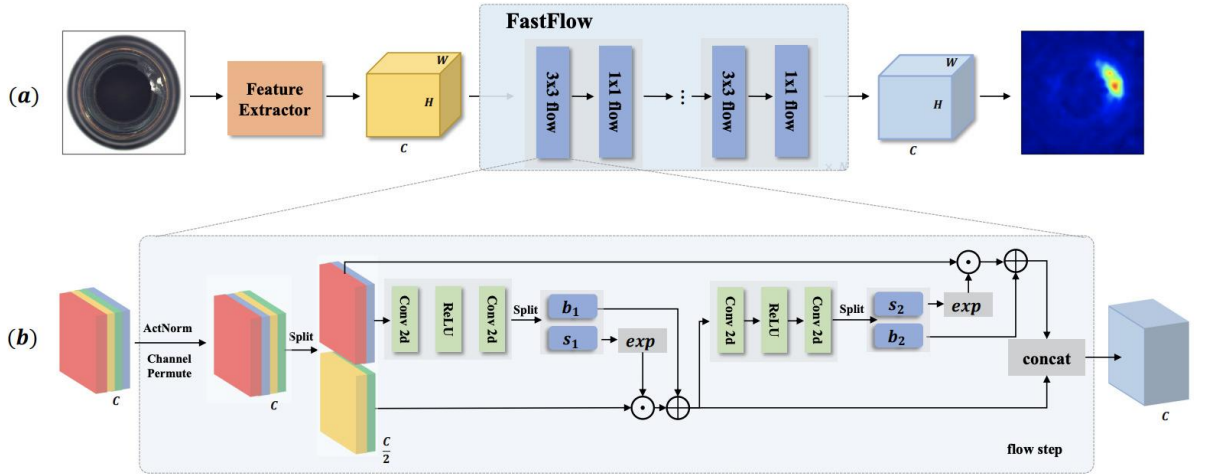
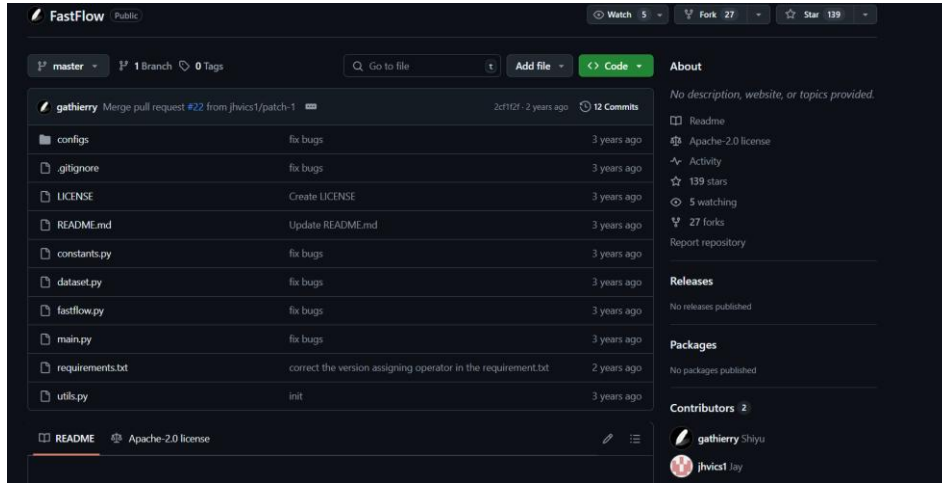


Figure 2: (a) the whole pipeline for unsupervised anomaly detection and localization in our method, which consists of a feature extractor and our FastFlow model. We can use an arbitrary network as the feature extractor such as CNN or vision transformer. FastFlow is alternately stacked by the “ $3 \times 3$ ” and “ $1 \times 1$ ” flow. (b) one flow step for our FastFlow, the “Conv 2d” can be  $3 \times 3$  or  $1 \times 1$  convolution layer for  $3 \times 3$  or  $1 \times 1$  flow, respectively.

### Şekil 9 FastFlow Model Mimarisi

Bu yapı, karmaşık veri dağılımlarını öğrenerek anormal örnekleri yüksek negatif log-olabilirlik değerleriyle tespit etmeyi sağlar. FastFlow, yerel ve küresel özellikleri etkili şekilde modelleyerek hem doğruluk hem de verimlilik sunar.



Şekil 10 FastFlow - FastFlow'un clone edildiği repo

Clone edilen modelin çalıştırılabilmesi için, modelin bağımlılıkları requirements.txt dosyasındaki paketler kullanılarak kurulmalıdır. Bu dosyada yer alan `git+https://github.com/VLL-HD/FrEIA.git` ifadesi, FrEIA kütüphanesinin doğrudan

GitHub deposundan indirilip kurulmasını sağlar. Ayrıca, pytorch-ignite gibi önemli kütüphaneler de yine bu dosya aracılığıyla yüklenir.

! Modelin dataseti içinde numaralandırılmasının düzenlenmesi lazım. Tüm ground truth ile tüm test defect dosyalarının eşlenmesi için.

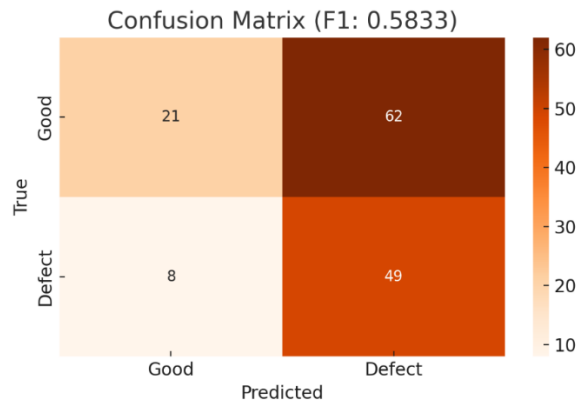
#### 2.5.2.2 Yapılan Değişimler:

En önemli değişimler “constant.py”, “dataset.py”, “fastflow.py” ve “main.py” dosyalarındadır.

- 1- DenseNet121 backbone’u eklenmiştir. – Görsel Özellik Çıkartma İçin-
- 2- Overfitting önleme için kullanılan düzenleme olan weighted decay değeri  $6e-5$  olarak değiştirilmiştir.
- 3- Dataset içinde uzantılar .png uzantısından .jpg uzantısına geçirilmiştir.
- 4- Mask threshold 0-1 float arasında iken threshold 0.7’den büyük olma durumu eklenmiştir.
- 5- ROC-AUC değerleri yanına F1 skoru, precision, recall curve, jaccard score performans matrikleri eklenmiştir.
- 6- Epoch sayısı overfitting olmasın diye 100 epoch ile sınırlandırılıp early stopping sayacı eklenmiştir.
- 7- Threshold bulma durumu find\_optimal\_threshold fonksiyonu ile sağlandı.

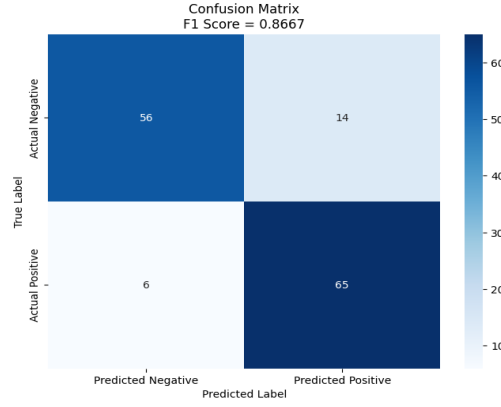
#### 2.5.2.3 Sonuçlar ve Yorumlar:

Model performanslarının, kullanılan farklı **backbone mimarileri** ve **eğitim parametrelerine** göre önemli ölçüde farklılık gösterdiği gözlemlenmiştir.



Şekil 11 WideResNet50 sonucu

Modelde WideResNet ile learning rate  $1e-4$  ile eğitildiğinde gözükten confusion matrix Şekil-11’de gözlemlenebilir.



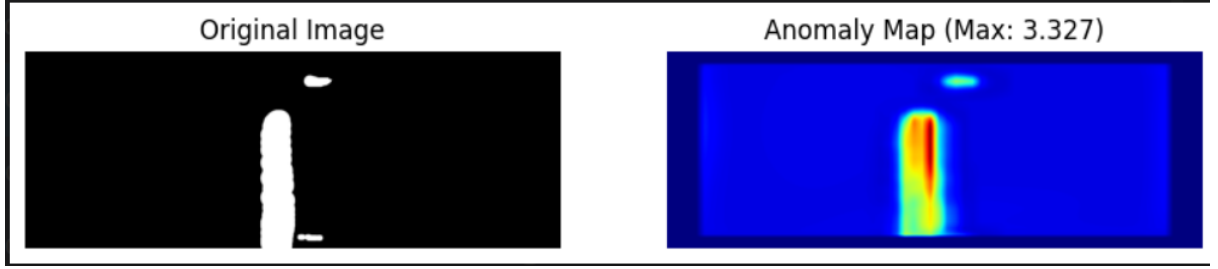
Şekil 12 DenseNet121 sonucu

Modelde DenseNet122 ile learning rate  $1e-3$  ve weigth decay  $6e-5$  olarak kabul edildiği zaman oluşan sonucun iyileştiği gözlemlenebiliyor. Model performansları, kullanılan farklı backbone mimarileri ve eğitim parametrelerine göre önemli ölçüde farklılık göstermiştir.

DenseNet121 tabanlı modelde, öğrenme oranı  $1e-4$  , weight decay  $6e-5$  olarak ayarlandığında, Confusion Matrix’te 56 doğru negatif ve 65 doğru pozitif sınıflandırma ile görece dengeli ve yüksek bir doğruluk sağlanmıştır. Bu yapı ile ortaya çıkan Doğruluk tablosu Şekil12’de gözlemlenebilir. Modelin F1 skoru 0.8667 gibi oldukça yüksek bir değerde seyretmiş ve bu durum modelin genel başarı seviyesini desteklemiştir. Buna karşın WideResNet50 backbone ile öğrenme oranı  $1e-3$  ve weight decay  $1e-5$  kullanıldığında, modelin performansında düşüş gözlenmiştir. Confusion Matrix sonuçları, özellikle "Good" sınıfının yanlış pozitif olarak yüksek oranda sınıflandırılması (62 adet) ile, modelin iyi sınıfı tespit etmede zorlandığını göstermektedir. Bu durum F1 skorunun 0.5833 gibi daha düşük bir değere inmesine neden olmuştur. Bu sonuçlar, düşük learning rate ve weight decay parametrelerinin WideResNet50 modeli üzerinde aşırı genelleme veya yetersiz öğrenmeye yol açabileceğine işaret etmektedir.

Sonuç olarak, model performansının backbone seçimi ve hiperparametre ayarlarına duyarlı olduğu, daha düşük learning rate ve düzenleme parametrelerinin

DenseNet üzerinde daha başarılı sonuçlar verdiği görülmektedir. Bu da anomali tespiti için model seçimi ve eğitim parametrelerinin dikkatli optimize edilmesi gerektiğini vurgulamaktadır.



*Şekil 13 FastFlow Anomaly Map ile Ground Truth Kıyası*

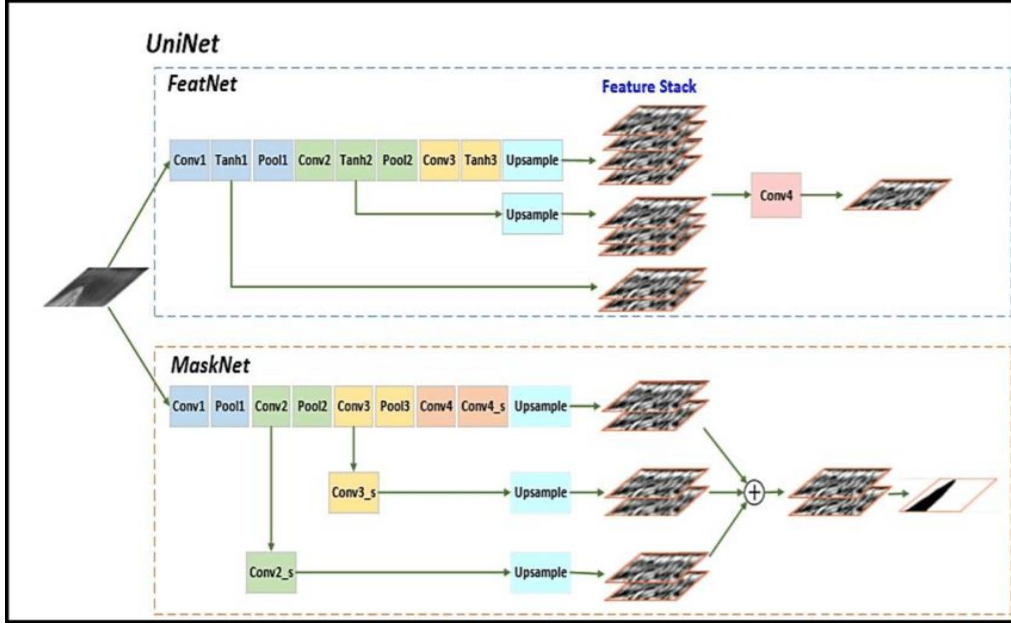
**Şekil 13’te** gösterildiği üzere, FastFlow modelinin ürettiği anomali haritası, gerçek kusur bölgeleriyle yüksek düzeyde çakışmakta ve kusurlu alanları başarılı şekilde vurgulamaktadır. Sol tarafta yer alan ground truth maskesiyle sağdaki anomaly map karşılaştırıldığında, modelin anomali yoğunluğunu doğru konumda ve güçlü sinyallerle (kırmızı bölgeler) tespit ettiği açıkça görülmektedir.

### 2.5.3 UniNet

Bu bölümde UniNet modeli anlatılacaktır.

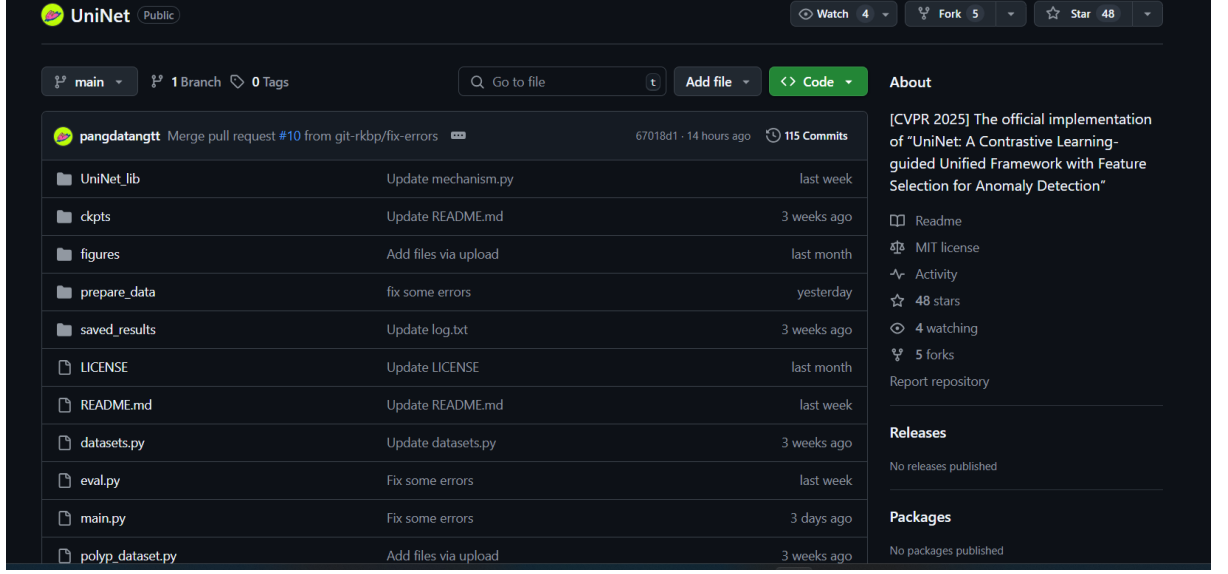
#### 2.5.3.1 Mimarisi

UniNet modeli, çok katmanlı bir encoder-decoder mimarisi üzerine kuruludur. Encoder kısmı, giriş görüntüsünden farklı seviyelerde özellikler çıkaran ardışık katmanlardan oluşur ve bu katmanlar giderek daha derin ve soyut temsiller üretir. Decoder kısmı ise encoder’dan gelen bu çok seviyeli özellikleri kullanarak, orijinal giriş boyutunda anlamlı çıktılar üretir.



*Şekil 14 UniNet Modeli Mimarisi*

Encoder ve decoder katmanları arasında bulunan “skip connection” yapıları, özellikle yüksek çözünürlüklü detayların korunması için kritik öneme sahiptir. Bu atlamalar sayesinde, encoder’dan decoder’a doğrudan bilgi akışı sağlanarak detay kayıplarının önüne geçilir ve modelin daha hassas segmentasyon veya anomali haritaları üretmesi mümkün olur. UniNet’in bu yapısı, çok seviyeli bilgi bütünleşmesi ve detay odaklı geri besleme mekanizmaları ile anomali tespiti gibi karmaşık görevlerde yüksek performans sağlar.



Şekil 15 pangdatangtt - UniNet Modeli Repo

Şekil 15’te gösterilen GitHub deposu, UniNet modeline ait açık kaynak kodları içermektedir. Kendi modelimizi eğitmek için bu repo kullanılmıştır.

#### 2.5.3.2 Yapılan Değişimler:

Yapılan değişimler “dataset.py” , “eval.py” , “main.py”, “train.py” dosyalarıdır.

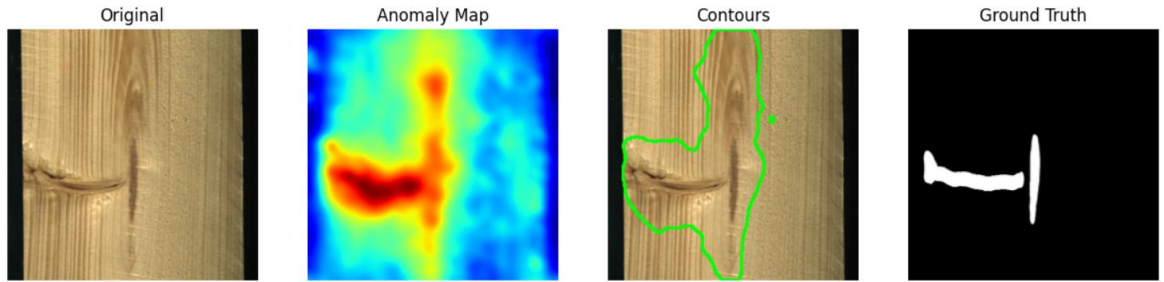
- 1- Precision-Recall eğrisi kullanılarak, modelin performansını en iyi yansıtan eşik değeri (threshold) otomatik olarak belirlenmiştir.
- 2- F1 skorunun maksimum olduğu optimal eşik değeri tespit edilmiştir.
- 3- Derin öğrenme modüllerinde kullanılmak üzere, özel bir Gaussian filtre çekirdeği (kernel) oluşturma fonksiyonu geliştirilmiştir.
- 4- Model değerlendirmesinde, Ortalama Doğruluk (Average Precision - AP), Jaccard indeks, doğruluk (accuracy) ve karışıklık matrisi (confusion matrix) gibi çok çeşitli performans metrikleri hesaplanmıştır.
- 5- Değerlendirme sürecinde, kare başına düşen işleme süresi (frame per second - fps) ve gecikme analizleri gerçekleştirilmiştir.
- 6- Elde edilen değerlendirme sonuçlarının anlaşılır biçimde sunulabilmesi için tablo formatında raporlama imkanı sağlanmıştır.
- 7- Veri setleri ve eğitim dosyalarında, kullanılan veri yolları ve dosya uzantıları düzenlenerek veri yönetimi optimize edilmiştir.

#### 2.5.3.3 Sonuçlar ve Yorumlar:

Model, BEST\_P\_PRO.pth ağırlıkları kullanılarak MVTec AD veri setinin ahşap (wood) sınıfı üzerinde değerlendirilmiştir. Görüntü seviyesi değerlendirmede, modelin F1 skoru 0.898 ve Intersection over Union (IoU) değeri 0.8148 olarak elde edilmiştir. Confusion Matrix sonuçları, 60 doğru negatif ve 66 doğru pozitif sınıflandırmanın yanı sıra, düşük yanlış pozitif ve yanlış negatif değerlerini göstermektedir; bu da modelin hem anormal hem de normal görüntüleri yüksek doğrulukla ayırt ettiğini göstermektedir.

Tabloda ayrıca, modelin ahşap veri sınıfı için görüntü seviyesi AUROC (Area Under Receiver Operating Characteristic) skoru 93.6%, piksel seviyesi AUROC skoru 96.7% ve piksel seviyesi ortalama precision (AUPRO) skoru ise 88.6% olarak raporlanmıştır. Bu sonuçlar, modelin özellikle anomali segmentasyonu görevinde yüksek performans sergilediğini, hem görüntü hem de piksel seviyesinde güvenilir tespitler yaptığını ortaya koymaktadır.

Genel olarak, modelin belirlenen eşik değeri (threshold) 0.5 olarak optimize edilmiş ve anomali skor aralığı 1.43 ile 2.73 arasında değişmektedir. Bu performans değerleri, modelin ahşap kusur tespitinde sağlam ve etkili bir çözüm olduğunu göstermektedir.



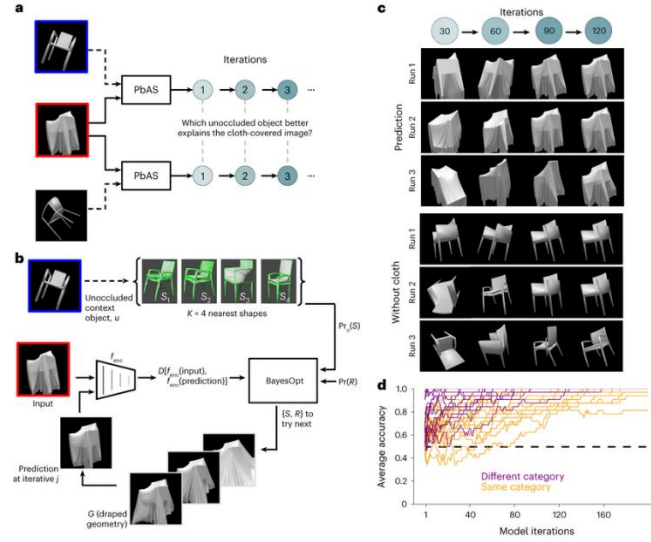
Şekil 16 UniNet Anomaly Map ile Ground Truth Kıyası

#### 2.5.4 PBAS

Bu bölümde PBAS modeli anlatılacaktır.

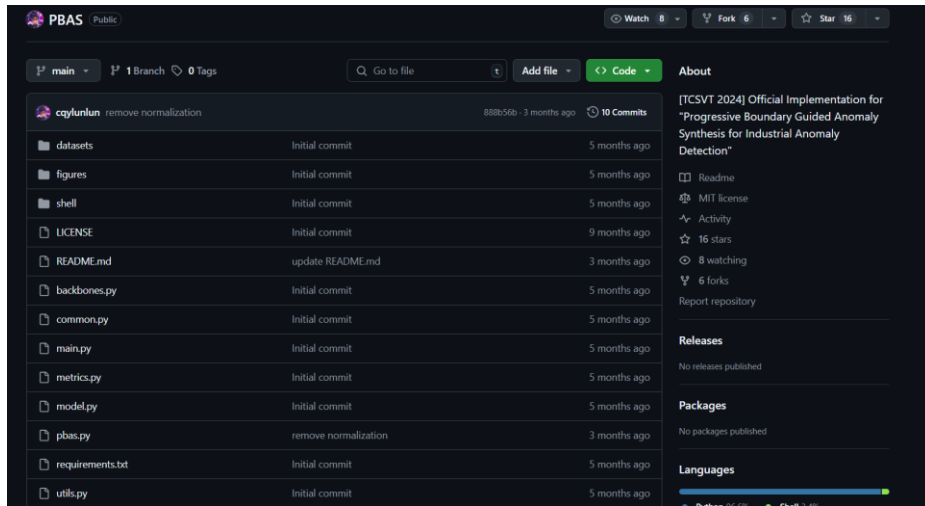
### 2.5.4.1 Mimarisi

PBAS (Patch-Based Anomaly Scoring) modeli, yalnızca kusursuz görüntülerle eğitilen, yamalara (patch) dayalı denetimsiz bir anomali tespit mimarisidir. Giriş görüntüsünden çıkarılan özellikler, önceden oluşturulmuş normal embedding'lerle karşılaştırılır



Şekil 17 PBAS Model Mimarisi

Her bir yamanın bu referanslarla olan uzaklığı hesaplanarak anomali skoru elde edilir; uzaklık arttıkça anomalilik olasılığı da artar. Bu skorlar birleştirilerek anomaly map oluşturulur ve eşikleme yöntemiyle segmentasyon maskeleri elde edilir.



Şekil 18 cqylunlun - PBAS Repo

Şekil 18'de gösterilen GitHub deposu, PBAS modeline ait açık kaynak kodları içermektedir. Kendi modelimizi eğitmek için bu repo kullanılmıştır.



#### 2.5.4.2 Yapılan Değişimler:

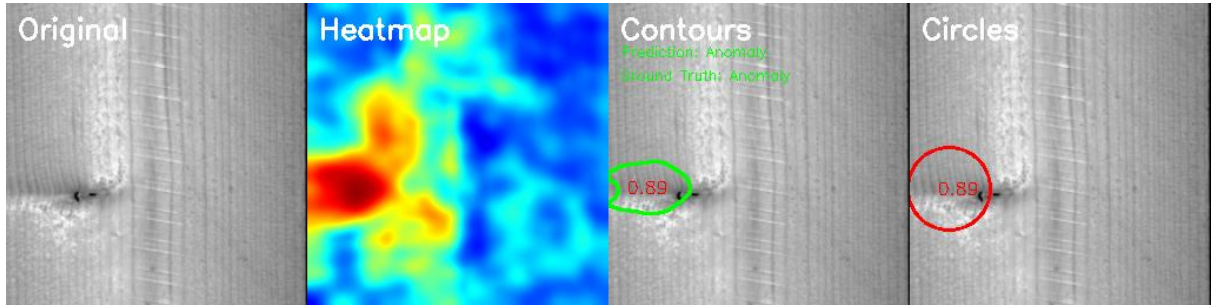
1. F1 score, Precision score, Recall score, Confusion matrix gibi sınıflandırma metrikleri eklendi.
2. Threshold ile Heatmap ve Overlay çizimi eklendi.
3. Kontur merkezine anomali skoru yazdırıldı.
4. Anomali minimum çaplı çember ile çevreleyen fonksiyon eklendi.
5. Her konturun “Alanı, Merkez koordinatları, Ortalama anomaly skoru, F1, precision, recall skorları” .txt dosyasına yazıldı.
6. Yeni görsel kayıt formatları eklendi.

#### 2.5.4.3 Sonuçlar Ve Yorumlar:

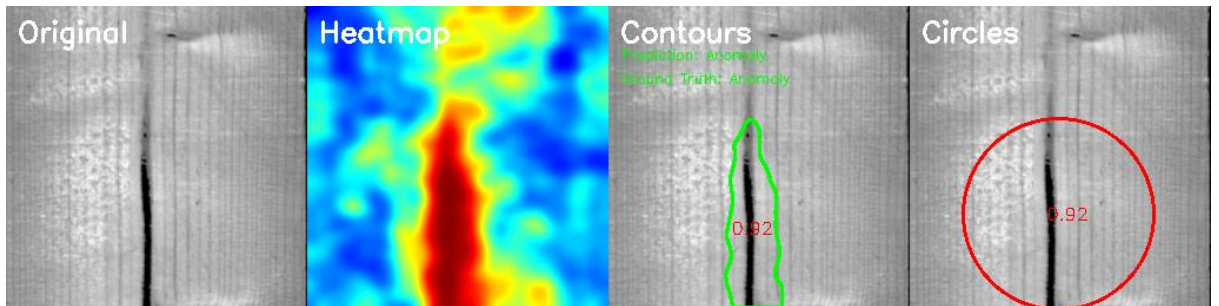
PBAS (Patch-Based Anomaly Scoring) modeli üzerinde gerçekleştirilen yapısal geliştirmeler, modelin yalnızca sayısal başarı metrikleriyle değil, aynı zamanda görsel ve kontur düzeyinde yorumlanabilirlik açısından da önemli ölçüde güçlenmesini sağlamıştır. Uygulanan değişiklikler kapsamında, image-level anomaly skorları için optimum threshold değeri otomatik olarak belirlenmiş; precision, recall ve F1 skorları hesaplanarak sınıflandırma başarımı detaylı biçimde değerlendirilmiştir. Elde edilen sonuçlara göre, modelin **F1 skoru 0.8889**, **precision 0.8767** ve **recall 0.9014** olarak hesaplanmış; bu da PBAS'ın dengeli bir şekilde hem anomaliyi yakalama (yüksek recall) hem de yanlış pozitifleri azaltma (yüksek precision) kapasitesine sahip olduğunu göstermektedir. Ayrıca modelin **image-level AUROC skoru %91.91** ve **Average Precision (AP) skoru %85.54** ile sınıflar arası ayırım gücünün yüksek olduğu görülmektedir.

PBAS modelinde anomali sınıflandırması için eşik değeri belirleme süreci, image-level anomaly skorlarının doğrulukla ikili (binary) sınıflandırmaya dönüştürülebilmesini sağlamak amacıyla optimize edilmiştir. Model, tüm test örnekleri için üretilen sürekli skorlar üzerinde minimum ve maksimum değerler arasında 100 farklı eşik değeri oluşturarak (0.01-0.99 gibi), her biri için tahmin edilen sınıf etiketlerini hesaplamaktadır. Bu tahminler, gerçek etiketlerle karşılaştırılarak F1 skorları elde edilmekte ve en yüksek F1 skorunu sağlayan eşik değeri “optimal threshold” olarak seçilmektedir. Bu yöntem, modelin precision ve recall arasındaki dengeyi en iyi sağlayacak eşiği otomatik olarak belirlemesine olanak tanımakta; sonrasında bu eşik değeri ile elde edilen nihai precision, recall ve F1 metrikleri, modelin anomali tespitteki genel başarımını somut olarak

yansıtmaktadır. Ayrıca, segmentasyon düzeyinde gerçekleştirilen kontur tabanlı analizlerde de benzer şekilde lokal F1 skorları hesaplanarak, modelin hem görüntü hem de piksel düzeyinde tutarlı ve yüksek doğrulukta tahminler sunduğu detaylı biçimde raporlanabilmektedir. Ek olarak, genişletilen evaluate() fonksiyonu ile segmentasyon maskeleri üzerinden kontur bazlı analiz yapılmış, her anomalinin alanı, ortalama skoru ve lokal F1 değeri hesaplanarak sonuçlar hem görsel hem metinsel olarak kayıt altına alınmıştır. Overlay haritalar, binary maskeler, contour çizimleri ve heatmap görselleştirmeleri sayesinde model çıktıları daha anlaşılır hale getirilmiş ve karar destek süreci güçlendirilmiştir. Çıktı örnekleri Şekil-19 ve Şekil-20’de sunulmuştur. Tüm bu yenilikler, PBAS modelini sadece etkili bir anomaly detection aracı değil, aynı zamanda yorumlanabilirliği ve denetlenebilirliği yüksek bir sistem haline getirmiştir.



Şekil 19 PBAS ile Anomali Tespiti – I



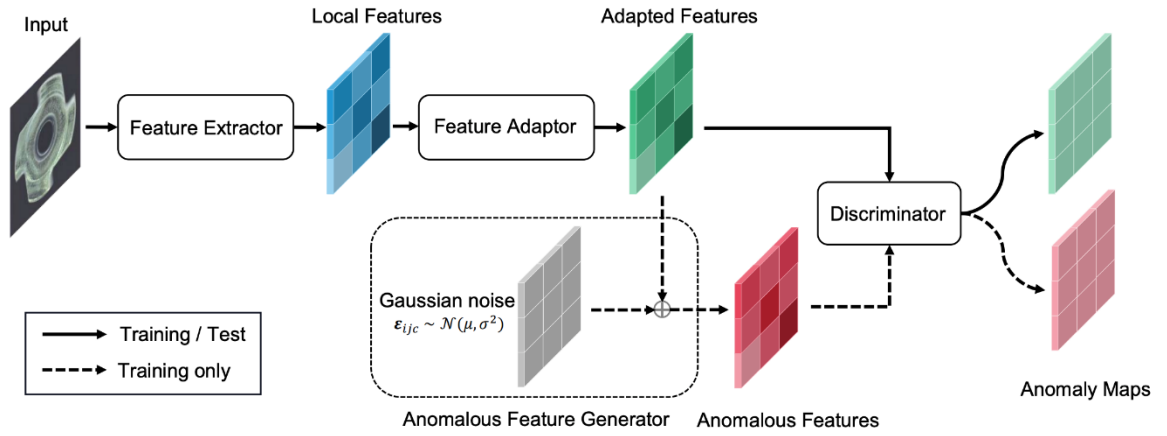
Şekil 20 PBAS ile Anomali Tespiti – II

### 2.5.5 SimpleNet Modeli

Bu bölümde SimpleNet modeli anlatılacaktır.

#### 2.5.5.1 Mimarisi

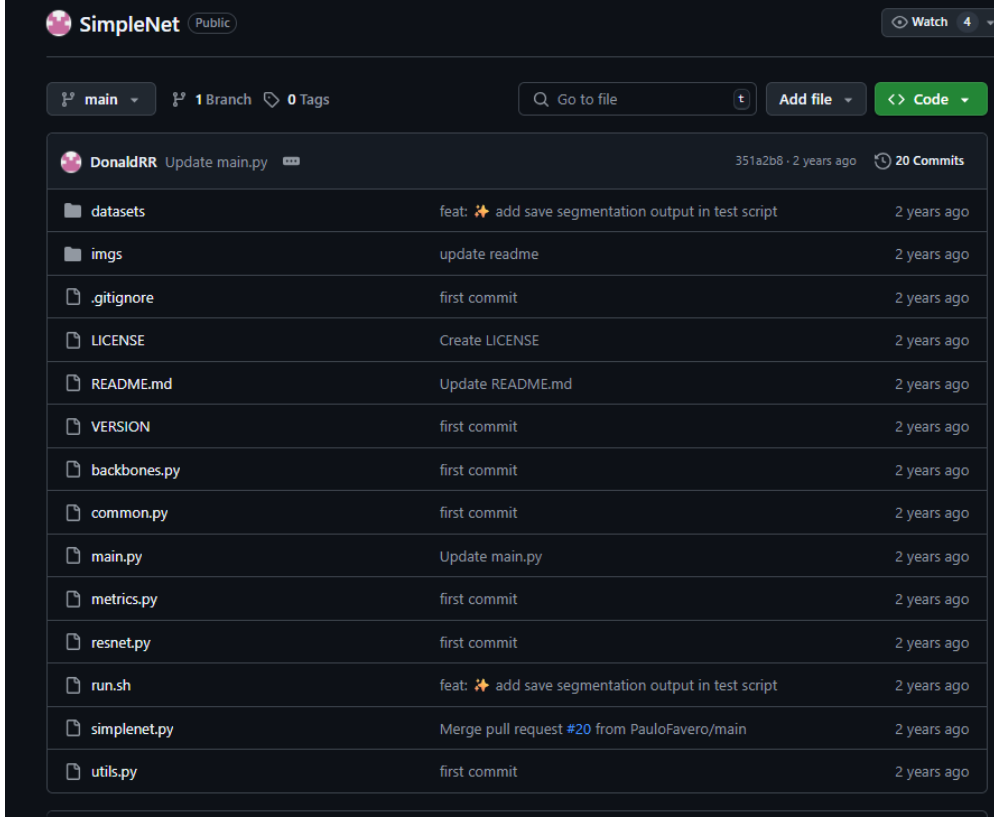
**SimpleNet**, denetimsiz görüntü tabanlı anomali tespiti için geliştirilen, sade ama etkili bir derin öğrenme mimarisidir. Model, yalnızca "normal" örneklerle eğitilerek, test aşamasında karşılaştığı anomalili bölgeleri yüksek hassasiyetle belirlemeyi hedefler. Girdi görüntüleri önce bir backbone (örneğin WideResNet-50) ile **yerel özneliklere** dönüştürülür, ardından bu öznelikler **Feature Adaptor** modülü üzerinden normalize edilir ve daha ayrık bir temsile aktarılır. Şekil-21'de bahsedilen mimari şeması sunulmuştur.



Şekil 21 SimpleNet Model Mimarisi

Bu çalışmada, görüntü tabanlı anomali tespiti için denetimsiz bir yaklaşım sunan SimpleNet mimarisi esas alınarak, adaptif karar mekanizmalarıyla güçlendirilmiş bir sistem geliştirilmiştir. Model mimarisi; bir giriş görüntüsünden başlamak üzere sırasıyla **özellik çıkarımı**, **özellik adaptasyonu**, **anomali skoru üretimi** ve **anomaly map oluşturulması** aşamalarından oluşmaktadır. İlk olarak, giriş görüntüsü bir backbone ağı (bizim çalışmamızda **WideResNet-50** backbone ağı kullanıldı) aracılığıyla yerel özelliklere dönüştürülür. Bu yerel özellikler, Feature Adaptor modülü üzerinden daha ayrık ve normalize edilmiş bir uzaya taşınarak adapte edilmiş özelliklere çevrilir. Eğitim süreci sırasında yalnızca normal veriler kullanılarak, bu özellikler üzerine **Gaussian gürültüsü** eklenerek yapay anomaliler üretilir ve bu anomaliler ile gerçek normal özellikler **Discriminator** ağına verilerek ayrıklaştırma (discrimination) yeteneği

kazandırılır. Discriminator, hem normal hem de anormal özellikleri sınıflandırarak, her bir piksel için bir anomali skoru oluşturur ve bu skorlar nihai **anomaly map**'i üretir. Bu yapı, herhangi bir doğrudan etiket ihtiyacı olmaksızın anomalili bölgeleri yüksek hassasiyetle tespit edebilme kapasitesi sunar.



Şekil 22 SimpleNet donaldr repo

Şekil 22’de gösterilen GitHub deposu, SimpleNet modeline ait açık kaynak kodları içermektedir. Kendi modelimizi eğitmek için bu repo kullanılmıştır.

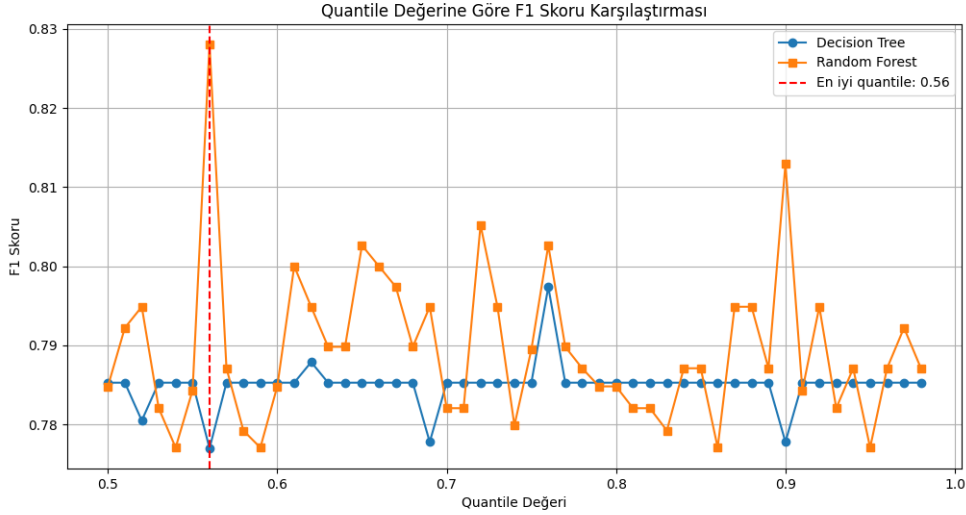
#### 2.5.5.2 Yapılan Değişimler:

##### 1. Eşik Optimizasyonu ve Özellik Tabanlı Modelleme:

Bu çalışmada, SimpleNet modelinden elde edilen heatmap çıktılarının daha etkili bir şekilde sınıflandırılabilmesi amacıyla, eşik tabanlı karar analizleri ile klasik makine öğrenmesi algoritmaları entegre edilmiştir. İlk aşamada, her bir test görseli için farklı quantile eşik değerleri denenmiş; bu eşikler yardımıyla oluşturulan binary maskelerden anomaly\_ratio, mean\_intensity, topk\_mean, std\_intensity, entropy gibi istatistiksel ve bölgesel öznitelikler çıkarılmıştır.

## 2. Model Kıyaslamaları ile En Uygun Sınıflayıcının Seçilmesi:

Bu öznitelikler Decision Tree ve Random Forest gibi algoritmalarla sınıflandırılmış, her quantile değeri için ayrı ayrı F1 skorları hesaplanmıştır. Şekil-23'te gözlemlenebileceğine göre, yapılan işlemler sonucunda optimum eşik değeri quantile = 0.56 olarak belirlenmiş ve bu noktada en yüksek başarı sağlayan model türü tanımlanmıştır.



Şekil 23 SimpleNet Optimal Eşik Değer

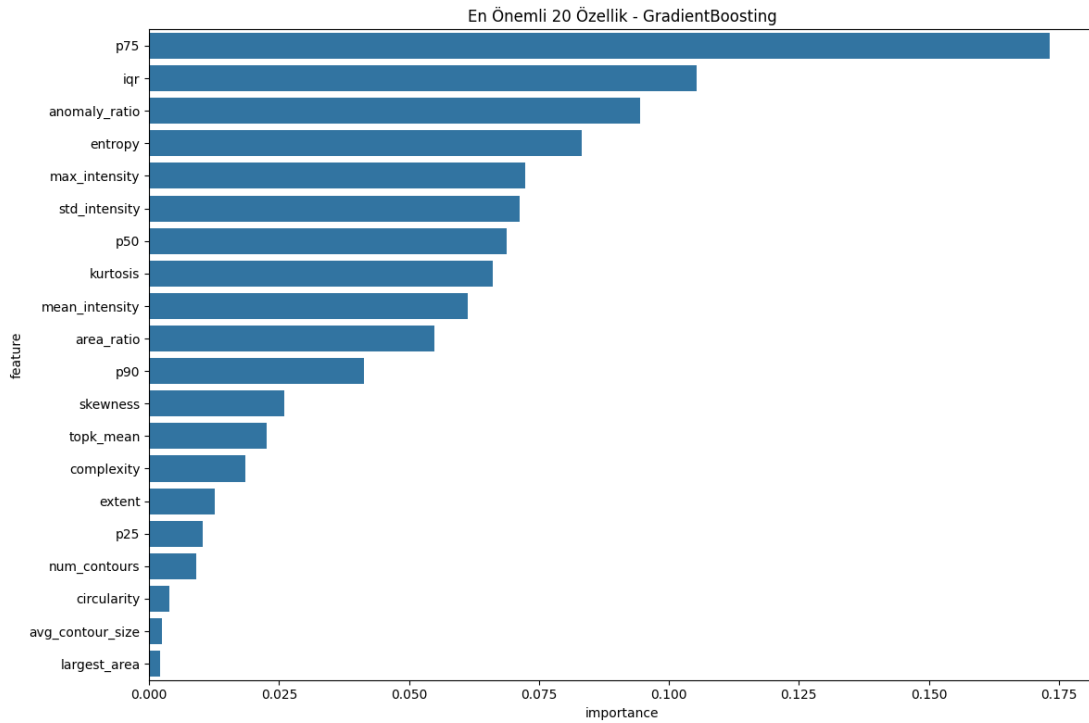
## 3. Genişletilmiş Model Havuzu ve Hiperparametre Optimizasyonu:

İkinci aşamada ise çıkarılan 20'den fazla özneliğe dayalı olarak Gradient Boosting, XGBoost, AdaBoost, SVM gibi modeller GridSearchCV ve Optuna kütüphaneleri kullanılarak detaylı hiperparametre optimizasyonuna tabi tutulmuştur.

## 4. Özellik Önem Sıralaması ile Yorumlanabilirlik:

Model performansı sadece metriklerle değil, aynı zamanda açıklanabilir yapay zeka yaklaşımıyla da desteklenmiştir. p75, iqr, anomaly\_ratio, entropy ve max\_intensity gibi özniteliklerin karar mekanizmasındaki ağırlıkları grafiklerle görselleştirilmiştir. Şekil 24'teki Özellik Önem Grafiği, Gradient Boosting modelinin anomaly heatmap'lerden çıkarılan istatistiksel ve geometrik öznitelikler üzerinden nasıl karar verdiğini görselleştirmektedir. Bu öznitelikler, doğrudan birer "threshold" (eşik değeri) olarak kullanılmaz; bunun yerine model, her bir örnek için bu özniteliklerin değerlerini girdi

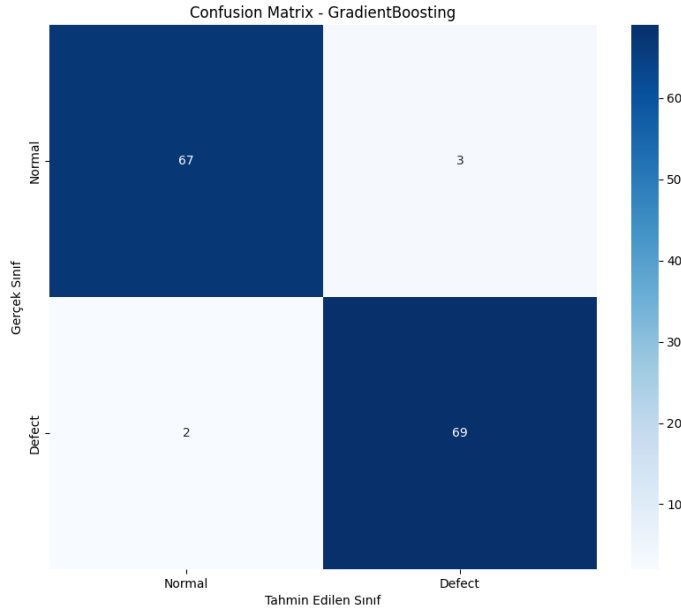
olarak alır ve bu çok boyutlu özellik uzayında sınıflandırma yapar. Örneğin, **p75** (75. yüzdelik dilim) özelliği bir görseldeki anomaly skorlarının üst çeyreğini temsil ederken, model bu değerin diğer özniteliklerle birlikte nasıl bir örüntü oluşturduğunu öğrenir. **iqr**, **entropy**, **anomaly\_ratio**, **std\_intensity**, gibi özellikler ise anomaly map'ın dağılımını, varyansını ve düzensizliğini yansıtarak kusurlu ve kusursuz yüzeyleri ayırt etmeye katkı sağlar. Dolayısıyla bu grafik, modelin karar verirken hangi istatistiksel temsillere daha çok "ağırlık" verdiğini, yani sınıflandırma sırasında hangi özniteliklerin daha fazla bilgi taşıdığını ortaya koyar. Bu yapı, sabit bir threshold belirlemekten çok daha esnek ve genellenebilir bir çözüm sunar.



Şekil 24 Özellik Önem Grafiği

## 5. Tüm Sürecin Görselleştirilmesi ve Kayıt Edilmesi:

Model çıktıları confusion matrix (Şekil 25), sınıflandırma raporu ve feature importance görselleri (Şekil 24) ile desteklenmiş, prediction CSV ve optimizasyon log dosyaları proje klasöründe sistematik biçimde kaydedilmiştir.



Şekil 25 SimpleNet Karmaşıklık Matrisi

#### 2.5.5.3 Sonuçlar ve Yorumlar:

Bu modelde, diğer yaklaşımlardan farklı olarak makine öğrenmesi algoritmaları, heatmap'lerden çıkarılan özelliklerin en uygun eşik değerlerini belirlemek amacıyla girdi olarak kullanılmış ve hiperparametre optimizasyonu ile en yüksek doğrulukta sınıflandırma hedeflenmiştir. Gerçekleştirilen çalışmada, anomaly map'lerden elde edilen istatistiksel ve bölgesel özellikler üzerinden eğitilen **Gradient Boosting** sınıflayıcı, modelleme sürecinde en başarılı performansı göstermiştir. Sınıflandırma raporuna göre modelin genel doğruluk oranı **%96** olarak hesaplanmış; bu da modelin hem kusurlu hem de kusursuz yüzeyleri yüksek isabet oranıyla ayırt edebildiğini ortaya koymuştur. Özellikle **Normal** sınıfı için **precision: %97**, **recall: %96** ve **F1 skoru: %96**, **Defect** sınıfı için ise **precision: %96**, **recall: %97** ve **F1 skoru: %97** olarak elde edilmiştir. Bu sonuçlar, modelin dengeli, tutarlı ve güvenilir bir sınıflayıcı olduğunu açıkça göstermektedir. Ayrıca makro ve ağırlıklı ortalama metriklerin de **0.96** seviyelerinde olması, modelin genel sınıflandırma başarımını pekiştirmektedir. Bu sonuçları Şekil-26 ile de görsel olarak sunulabildi. Model, hem hatalı alarm oranını minimize etmekte hem

de gerçek kusurları yüksek başarıyla tespit ederek endüstriyel uygulamalar için oldukça güçlü bir çözüm sunmaktadır.

```
Sınıflandırma Raporu:
```

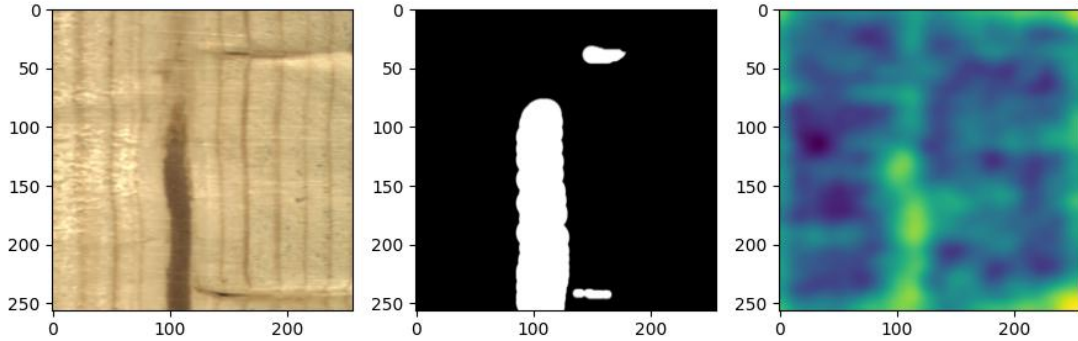
	precision	recall	f1-score	support
Normal	0.97	0.96	0.96	70
Defect	0.96	0.97	0.97	71
accuracy			0.96	141
macro avg	0.96	0.96	0.96	141
weighted avg	0.96	0.96	0.96	141

```
En önemli 10 özellik:
```

1. p75: 0.1733
2. iqr: 0.1053
3. anomaly\_ratio: 0.0944
4. entropy: 0.0832
5. max\_intensity: 0.0724
6. std\_intensity: 0.0713
7. p50: 0.0688
8. kurtosis: 0.0661
9. mean\_intensity: 0.0613
10. area\_ratio: 0.0548

✓ Tüm sonuçlar '/content/eval\_outputs' klasörüne kaydedildi.  
Tahminler: /content/eval\_outputs/predictions\_GradientBoosting.csv  
Confusion Matrix: /content/eval\_outputs/conf\_matrix\_GradientBoosting.png  
Özet rapor: /content/eval\_outputs/optimization\_summary.txt

Şekil 26 SimpleNet Değerlendirmesi



Şekil 27 SimpleNet Örnek Output

Şekil 27’de ise, SimpleNet modelinin gerçek bir test görüntüsü üzerinde ürettiği anomaly map, ground truth maskesi ile birlikte sunulmuştur. Modelin kusurlu alanları başarıyla tespit ettiği, anomaly map’te yüksek skorların (sarı-kırmızı bölgeler) gerçek kusurlarla çakıştığı net bir şekilde gözlemlenmektedir. Bu görseller, modelin hem sınıflandırma hem de lokalizasyon düzeyinde yüksek doğrulukla çalıştığını somut olarak ortaya koymaktadır.

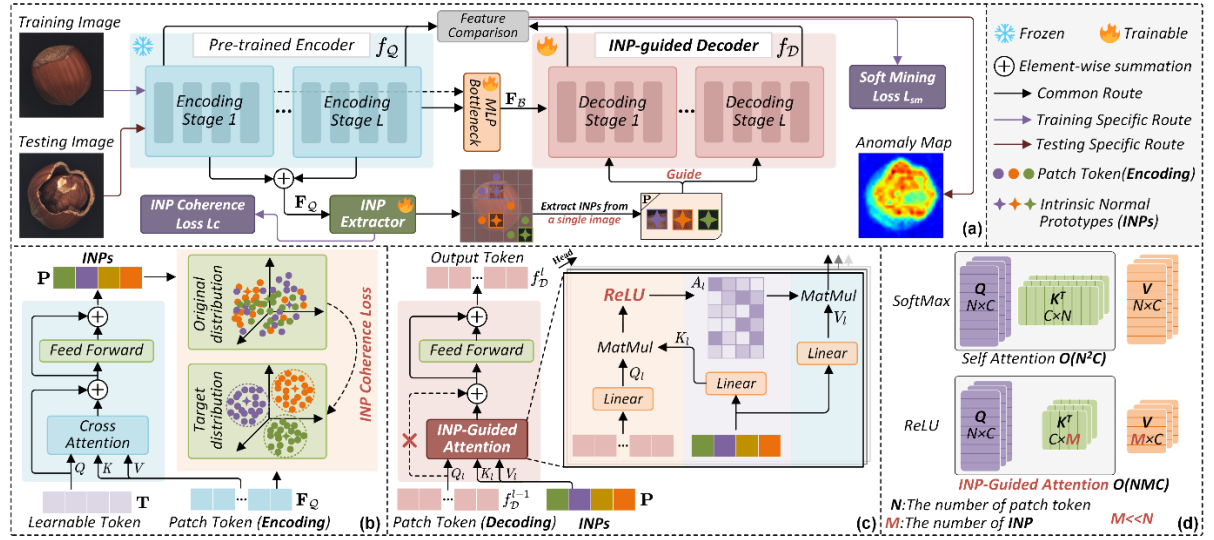


### 2.5.6 INP-Former Modeli

Bu bölümde INP-Former modeli anlatılacaktır.

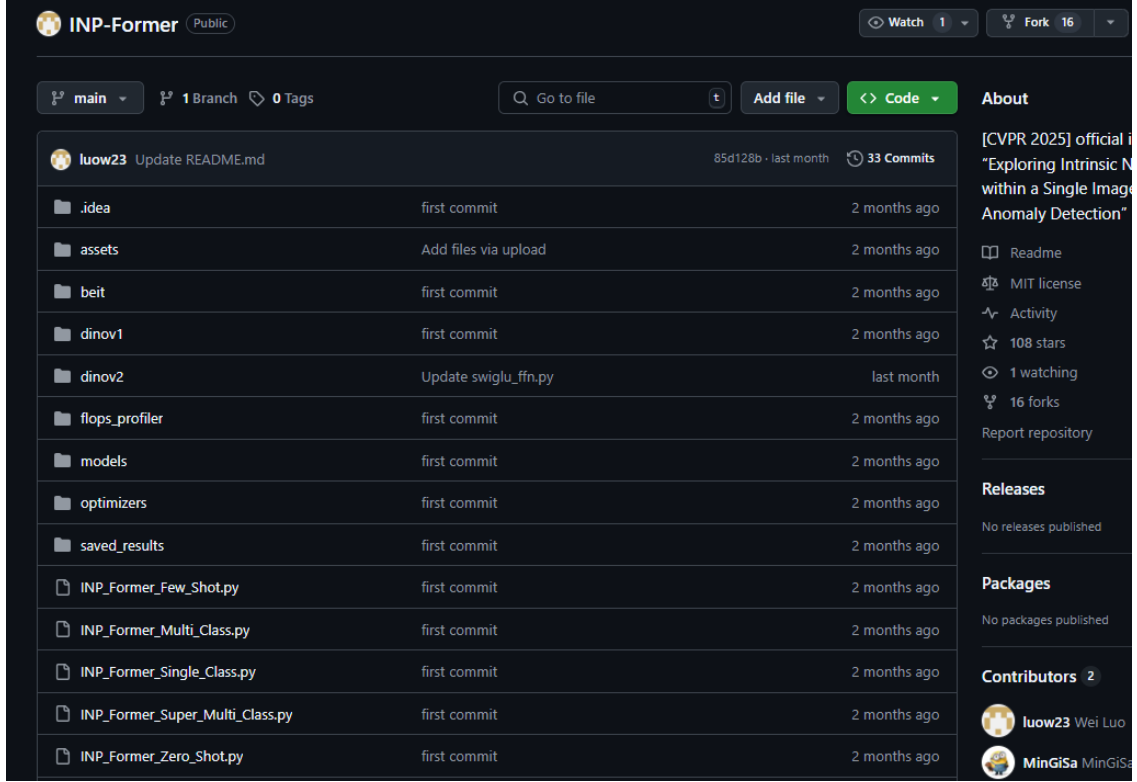
#### 2.5.6.1 Mimarisi

INP-Former modeli, yalnızca kusursuz (normal) görüntülerle eğitilen ve test aşamasında daha önce görülmemiş anomalileri tespit etmeyi amaçlayan, prompt tabanlı bir anomaly detection mimarisidir. Modelin temel bileşeni olan transformer yapısı, DINOv2 gibi güçlü bir görsel encoder ile elde edilen özellik haritalarını, her test görüntüsüne özgü olarak üretilen öğrenilebilir prompt'larla birleştirerek çalışır. Bu prompt'lar, görüntüdeki anomalilere duyarlı bölgeleri vurgulamak için cross-attention mekanizması ile kullanılır.



Şekil 28 INP-Former Modeli Mimarisi

Transformer katmanlarından elde edilen çıktı, anomaly map oluşturmak için kullanılır ve genellikle cosine similarity gibi bir uzaklık metriğiyle piksel düzeyinde anomaly skorları hesaplanır. Böylece INP-Former, hem görsel detaylara duyarlı hem de örnek bazlı dikkat mekanizmasıyla yüksek doğrulukta sınıflandırma ve segmentasyon yeteneği sunar. Mimari şeması Şekil-28'de sunulmuştur.



Şekil 29 INP-Former luow23 repo

Şekil 29’de gösterilen GitHub deposu, INP-Former modeline ait açık kaynak kodları içermektedir. Kendi modelimizi eğitmek için bu repo kullanılmıştır.

#### 2.5.6.2 Yapılan Değişimler:

##### 1- Validation aşaması için özel kayıp hesaplama fonksiyonu eklenmiştir.

Modelin eğitim sürecinin daha sağlıklı izlenebilmesi amacıyla evaluate\_val\_loss adlı bir fonksiyon geliştirilmiş ve eğitim sırasında her epoch sonunda doğrulama verisi üzerinde cosine embedding loss hesaplanması sağlanmıştır. Bu fonksiyon, encoder ve decoder çıkışlarının benzerliğini ölçerek modelin temsil öğrenme başarımını doğrudan değerlendirmeye imkân tanımaktadır. Orijinal çalışmada bu tarz bir doğrulama izleme mekanizması yer almamaktadır; dolayısıyla bu katkı modelin aşırı öğrenmeye (overfitting) karşı izlenebilirliğini artırmıştır.

##### 2. Eğitim sırasında doğrulama kaybı (validation loss) yazdırılarak izlenebilirlik artırılmıştır.

Eğitim döngüsüne her epoch sonunda modelin evaluate\_val\_loss fonksiyonu ile test verisi

üzerinde değerlendirilmesini içeren bir adım eklenmiştir. Bu sayede, eğitim süreci boyunca modelin genel performans trendi anlık olarak takip edilmekte ve gerektiğinde erken durdurma (early stopping) gibi stratejilerin uygulanmasına zemin hazırlanmaktadır. Orijinal kodda bu mekanizma bulunmadığından modelin aşırı öğrenip öğrenmediği yalnızca final metriklerle bakılarak değerlendirilebiliyordu.

### **3. Eğitim setindeki örnek sayısı, log dosyalarına açıkça yazdırılmıştır.**

Modelin eğitildiği görsel sayısı `print_fn(f'train image number: ...')` komutu ile kayda geçirilmiş ve bu bilgi her çalışma için kayıt altına alınmıştır. Bu, deney tekrarlanabilirliği açısından kritik bir adımdır çünkü kullanılan veri miktarı, modelin başarımıyla doğrudan ilişkilidir. Orijinal sürümde bu tür bir bilgi çıktısı sunulmamaktadır ve bu durum deney analizlerinde belirsizlik yaratabilmektedir.

### **4- Komut satırı argümanlarının varsayılan değerleri uygulama senaryosuna uygun olarak özelleştirilmiştir.**

Kodun belirli bir veri seti ve deney ortamında kullanılabilirliğini artırmak amacıyla `data_path`, `item`, `encoder`, `input_size`, `crop_size`, `batch_size`, `total_epochs` gibi parametrelerin varsayılan değerleri kullanıcıya özgü bir biçimde tanımlanmıştır. Böylece modelin farklı ortamlarda yeniden yapılandırılma ihtiyacı azaltılmış, deney tutarlılığı sağlanmıştır.

### **5- Kayıt dizinleri parametrik olarak adlandırılmış ve deney bilgileriyle ilişkilendirilmiştir.**

Kaydedilen model sonuçları, kullanılan veri kümesi, encoder tipi, giriş boyutları ve INP sayısı gibi hiperparametre bilgilerini içerecek şekilde dinamik olarak adlandırılmıştır. Bu yapılandırma sayesinde her bir deney çalışmasına ait çıktıların içerikleri kolaylıkla ayırt edilebilmekte ve sonuçların sistematik olarak organize edilmesi mümkün hale gelmektedir.

#### *2.5.6.3 Sonuçlar ve Yorumlar:*

Bu çalışmada, INP-Former modelinin ahşap yüzeylerdeki anormallikleri tespit etme performansı değerlendirilmiştir. Test sürecinde model, `dinov2reg_vit_large_14` tabanlı bir encoder ve dikkatle yapılandırılmış INP modülleri ile çalıştırılmış; her test görseli için anomaly map oluşturulmuş, ardından bu haritalar üzerinden anomaly skorları

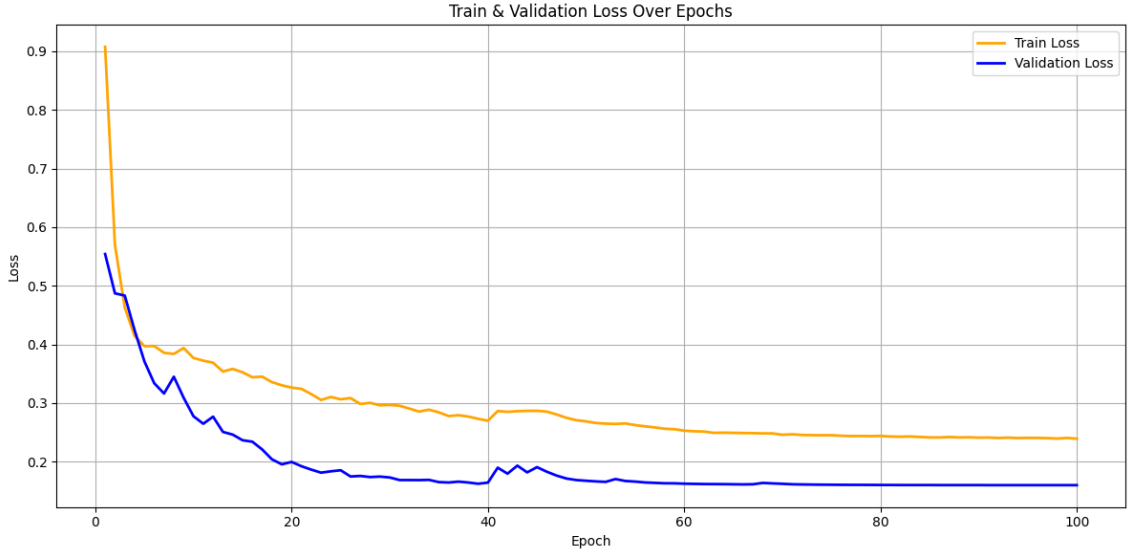
hesaplanmıştır. Skorların dağılımına göre optimal bir eşik değeri (threshold) tespit edilerek her görüntü “anormal” veya “normal” olarak sınıflandırılmıştır. Bu yaklaşım sayesinde modelin gerçek-pozitif ve yanlış-pozitif oranları arasındaki denge sağlanmış, karar doğruluğu en üst seviyeye çıkarılmıştır.

Eşik değerinin belirlenmesinde, anomaly skorlarının dağılımı üzerinde yapılan yoğun tarama (sweep) süreci sonucunda F1 skorunun maksimize edildiği değer temel alınmıştır. Bu değer etrafında yapılan tahminlerin sonuçları incelenmiş, F1 skoru, IoU değeri ve confusion matrix gibi metriklerle desteklenmiştir. Elde edilen sonuçlara göre model, hem normal hem de kusurlu görselleri oldukça başarılı bir şekilde ayırt etmiş; overlay kontur çizimleri ve görselleştirmelerle bu başarı görsel olarak da ortaya konulmuştur.

Özellikle segmentasyon maskelerinden elde edilen IoU değerleri, modelin anormalliklerin sadece varlığını değil, aynı zamanda yerini de isabetli biçimde tahmin edebildiğini göstermektedir. Anomaly map’lerin üzerine çizilen tahmini ve gerçek konturlar, modelin hata yapma eğilimlerini görsel düzeyde analiz etme olanağı sunmuştur. Bu süreç, modelin hassasiyetini (recall) artırırken, yanlış alarm oranını da düşük tutabilme potansiyelini ortaya koymuştur.

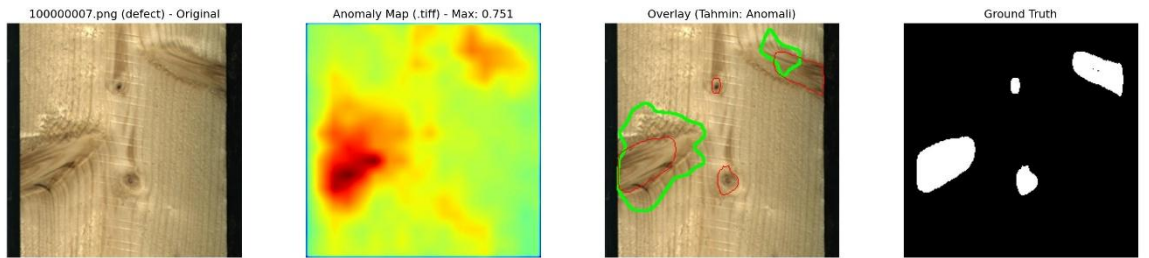
Modelin eğitim sürecine ait kayıp eğrileri **Şekil 30’da** sunulmuştur. Grafikte görüldüğü üzere, hem **eğitim (train)** hem de **doğrulama (validation)** kayıpları ilk 20 epoch boyunca hızlı bir şekilde azalmış ve ardından belirli bir seviyede stabil hale gelmiştir. Bu durum, modelin erken dönemlerde hızlı bir öğrenme gerçekleştirdiğini ve başlangıçtaki yüksek hata oranlarının kısa sürede düştüğünü göstermektedir. Özellikle **40. epoch** sonrasında doğrulama kaybının düşük seviyelerde sabitlenmesi, modelin **aşırı öğrenmeye (overfitting)** düşmeden genel geçer örüntüleri öğrenebildiğine işaret etmektedir. Ayrıca, eğitim kaybının doğrulama kaybına kıyasla bir miktar daha yüksek seyretmesi, modelin eğitim sırasında verideki gürültüye karşı dayanıklı olduğunu ve

karar sınırlarını sağlam bir şekilde öğrenebildiğini ortaya koymaktadır.

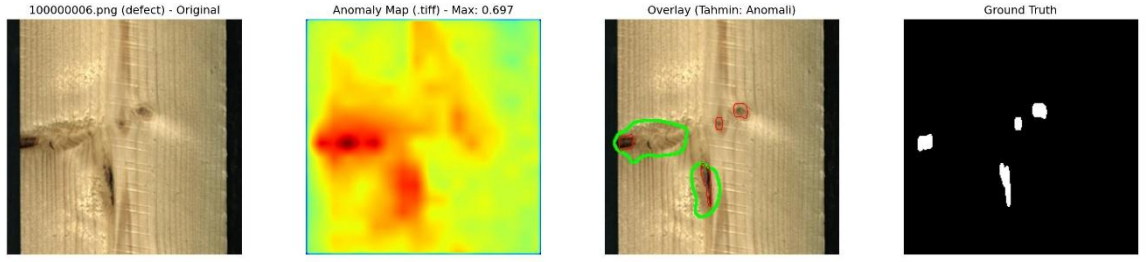


Şekil 30 Validation-Train Loss İlişkisine Göre Overfitting Kontrolü

Şekil 31 ve Şekil 32, INP-Former modelinin test görüntüleri üzerinde gerçekleştirdiği anomali tespiti ve segmentasyon performansını örneklerle göstermektedir. Her iki şekilde de sırasıyla orijinal görüntü, üretilen anomaly map, modelin tahminine göre oluşturulan overlay görüntü ve ground truth maskesi sunulmuştur. INP-Former modeli, anomali haritalarında yüksek skorlar (kırmızı bölgeler) üreterek kusurlu alanları doğru şekilde vurgulamış ve overlay görüntülerde bu bölgeleri yeşil konturlar ile başarılı biçimde işaretlemiştir.



Şekil 31 INP\_Fomer örnek output – I



Şekil 32 INP-Former örnek output – II

Sonuç olarak, INP-Former modeli, wood veri kümesinde detaylı bir görsel ve sayısal analiz süreciyle değerlendirilmiş; hem bölgesel hem de küresel doğruluk kriterlerinde tatmin edici performans sergilemiştir. Segmentasyon tabanlı değerlendirme, anomaly skorlarının eşiklenmesi ve overlay görselleştirme gibi modern yöntemlerin entegre edilmesiyle, bu modelin gerçek dünyadaki endüstriyel uygulamalara uygunluğu güçlü bir şekilde desteklenmiştir.

### 2.5.7 Voting Modeli

Voting (oylama) modeli, birden fazla modelin çıktılarının birleştirilerek nihai kararın verildiği toplu karar mekanizmasıdır. Bu projede kullanılan Voting yapısı, **EfficientAD**, **UniNet** ve **FastFlow** modellerinin çıktılarına dayanmakta ve her bir modelin test görüntüsü üzerinde yaptığı tahmini dikkate alarak **demokratik etiketleme (majority voting)** yaklaşımını uygulamaktadır.

Bu yöntemde her model, ilgili test görüntüsünü "**Anomali**" ya da "**Normal**" olarak sınıflandırır. Ardından, üç modelin verdiği kararlar toplanarak aşağıdaki şekilde oylanır:

- En az **iki model aynı kararı verirse**, bu karar **nihai karar** olarak kabul edilir.
- Örneğin, iki model görüntüyü “anomali” olarak sınıflandırmışsa, genel sonuç da “anomali” olur.

**Bu yöntemin temel avantajları şunlardır:**

- **Model Bağımsızlığı:** Her model kendi yapısı ve güçlü yönleriyle değerlendirme yapar.
- **Hata Telafisi:** Bireysel model hataları, diğer modellerin katkısıyla dengelenir.

- **Genel doğruluk artışı:** Özellikle tek bir modelin yetersiz kaldığı durumlarda daha dengeli ve sağlam bir karar mekanizması sunar.

Bu yaklaşım, özellikle heterojen yapıdaki anomali türlerini tanımakta zorlanan bireysel modellerin zayıf yönlerini kompanse etme potansiyeline sahiptir. Örneğin, **EfficientAD** küçük yapısal anomalilerde, **FastFlow** yoğun yüzey bozulmalarında ve **UniNet** ise detaylı segmentasyon görevlerinde başarılıdır. Bu modellerin birlikte kullanılmasıyla, genel sistem performansında artış gözlemlenmiştir.

Kullanıcı arayüzü üzerinden **Voting modeli** seçildiğinde, her bir modelin bireysel tahmini kullanıcıya ayrı ayrı sunulmakta ve sistemin verdiği **nihai karar** açık biçimde görüntülenmektedir. Bu sayede, karar süreci **şeffaflaştırılmış** ve kullanıcıya her bir modelin çıktısını inceleyerek **model bazlı analiz yapma imkânı** tanınmıştır. **Şekil 33** ve **Şekil 34**'te, bu yapının uygulama sonuçları örnek olarak sunulmakta; her modelin verdiği sınıflandırma kararı ile sistemin birleşik (oylama temelli) nihai sonucu görsel olarak karşılaştırılabilmektedir.

Sonuç olarak, voting yaklaşımı bu projede hem **güvenilirlik**, hem de **yorumlanabilirlik** açısından fayda sağlamış; modeller arası kolektif değerlendirme sayesinde daha dengeli sonuçlar elde edilmesine olanak tanımıştır.

## Analiz Sonuçları

[← Yeni Analiz](#)

Tahmin:

▲ Anomali

### Modellerin Kararları:

EfficientAD

Anomali

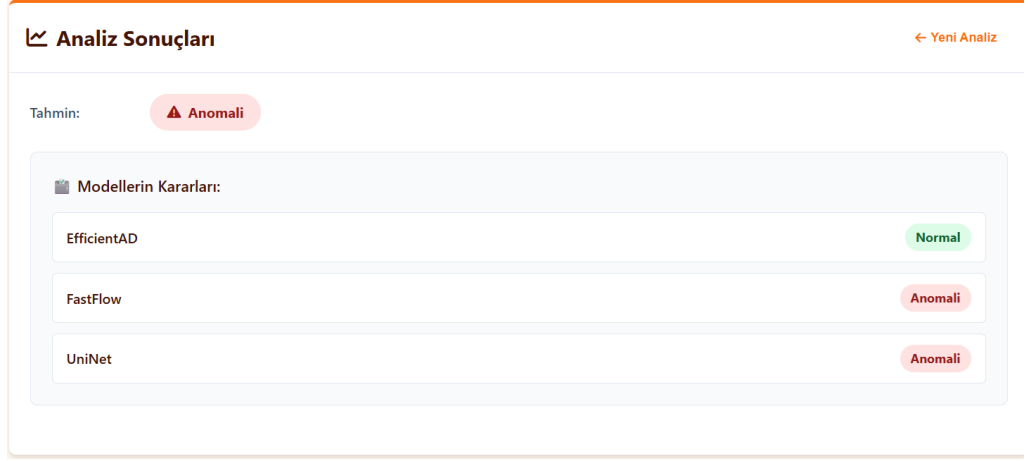
FastFlow

Anomali

UniNet

Anomali

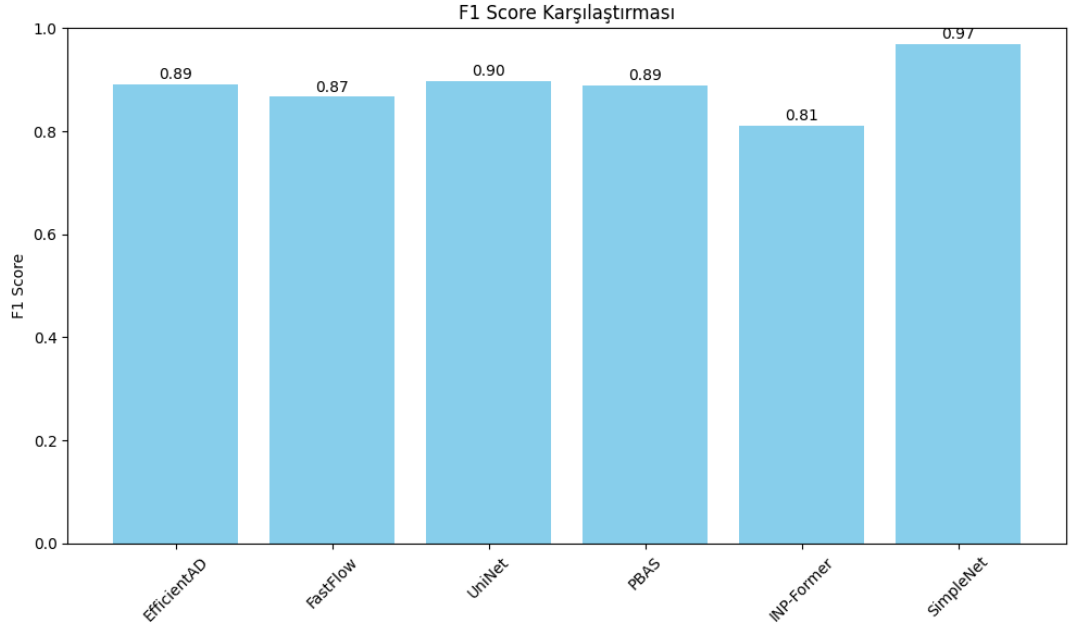
Şekil 33 Voting Yaklaşımı İle Örnek Çıktı – I



Şekil 34 Voting Yaklaşımı İle Örnek Çıktı - II

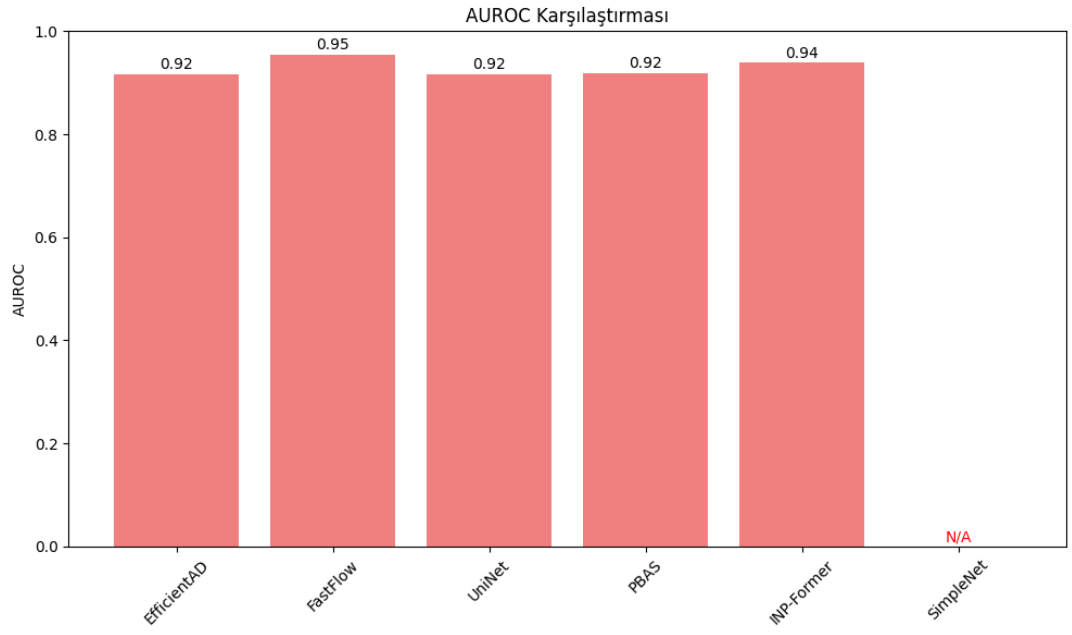
## 2.6 Model Kıyaslama

Bu bölümde, proje kapsamında kullanılan farklı anomaly detection modelleri, sayısal başarı metrikleri üzerinden karşılaştırmalı olarak değerlendirilmektedir. Karşılaştırmada temel olarak **F1 skoru**, **AUROC (Area Under ROC Curve)**, **precision** ve **recall** değerleri esas alınmış; her modelin sınıflandırma doğruluğu ve anomali tespiti başarısı bu metrikler ışığında analiz edilmiştir. Böylece, hem genel doğruluk hem de duyarlılık açısından en güçlü ve dengeli performans sergileyen modeller belirlenmiştir.

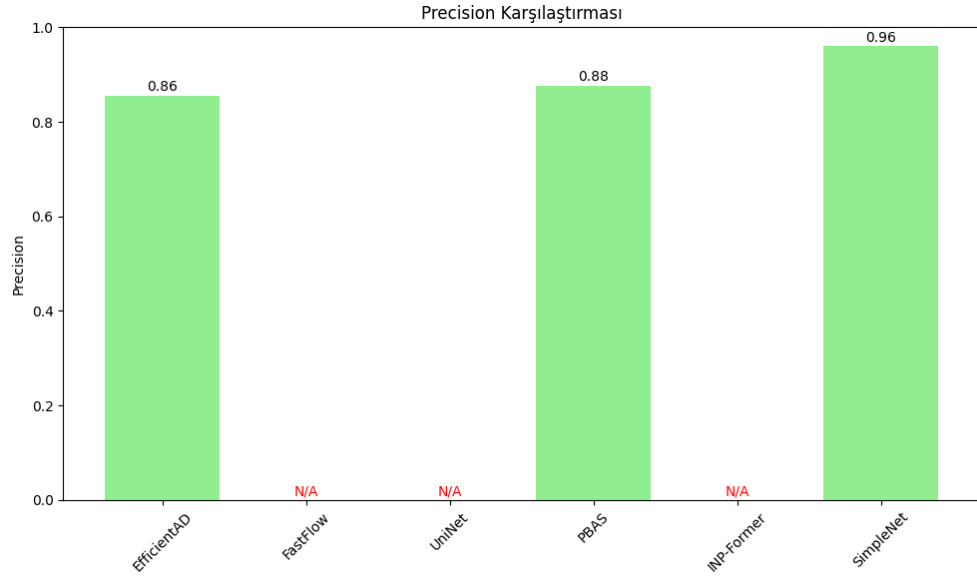


Grafik 1 F1 Skorları Kıyaslama

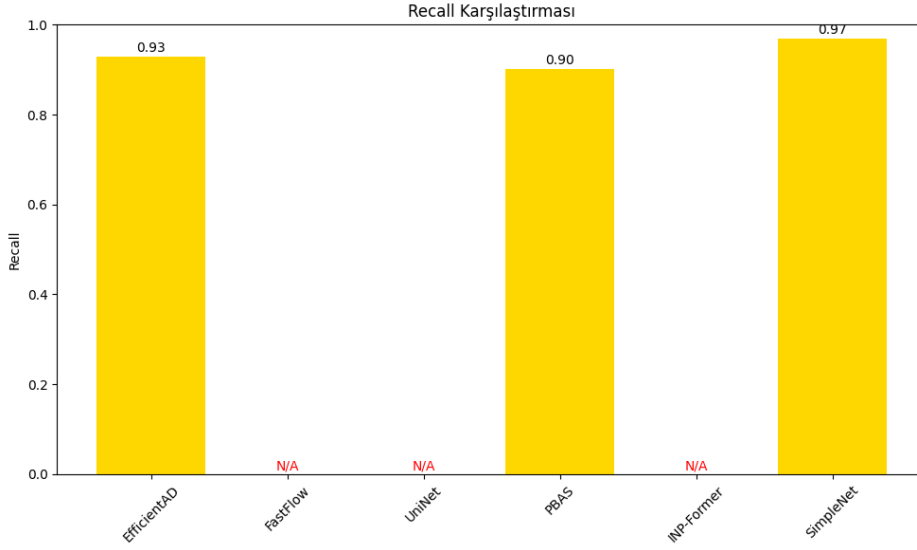




Grafik 2 AUROC Skor Kıyaslama



Grafik 3 Precision Skor Kıyaslaması



*Grafik 4 Recall Skoru Kıyaslama*

Yapılan kapsamlı karşılaştırma sonucunda, **SimpleNet modeli, %97 F1 skoru** ve **%96 doğruluk oranı** ile görüntü düzeyinde en yüksek sınıflandırma başarısını sergilemiştir. Bu durum, modelin hem anomalili hem de kusursuz görselleri yüksek isabet oranıyla ayırt edebildiğini göstermektedir (**Grafik 1**). Öte yandan, **UniNet modeli, %81.48 IoU skoru** ile özellikle **segmentasyon** görevlerinde, yani kusurlu bölgelerin lokal olarak belirlenmesinde en iyi sonuçları vermiştir. Bu başarı, UniNet’in çok katmanlı encoder-decoder mimarisi ve detay koruyucu “skip connection” yapıları sayesinde elde edilmiştir.

**FastFlow modeli, %95.44 AUROC skoru** ile sınıflar arası ayırım gücü açısından dikkat çekmektedir (**Grafik 2**). Modelin akış tabanlı yapısı, veri dağılımlarındaki ince farklılıkları başarılı şekilde modelleyerek, yüksek ayırım kabiliyetine sahip güvenilir anomaly skorları üretmesini sağlamıştır. **PBAS modeli** ise **%87.67 precision** ve **%90.14 recall** oranlarıyla dengeli bir sınıflandırma performansı ortaya koymuştur (**Grafik 3 ve Grafik 4**). Bu model, hem doğru anomali tespitinde yüksek başarı sağlarken, hem de hatalı pozitiflerin (false positive) azaltılmasında güçlü bir yapı sergilemiştir. Ayrıca, kontur bazlı değerlendirme ve görsel açıklanabilirlik gibi nitelikleri sayesinde karar destek sistemleri açısından da önemli avantajlar sunmaktadır.

Bazı modellerde **precision, recall veya AUROC** gibi metriklerin eksik olması, ilgili model mimarisinin çıktılarının **kod entegrasyonu sürecinde ortak bir değerlendirme formatına dönüştürülememesi** veya bazı metriklerin doğrudan hesaplanamamasıyla ilgilidir. Bu durum, özellikle farklı kaynaklardan klonlanan modellerin veri formatı ve değerlendirme yöntemlerindeki çeşitliliğinden kaynaklanmıştır.

Sonuç olarak, her modelin belirli metriklerde öne çıktığı gözlemlenmiştir. Bu da, anomali tespiti gibi çok boyutlu bir problemde model seçiminin yalnızca tek bir başarı metriğine değil; **kullanım senaryosu, sistem gereksinimi ve yorumlanabilirlik düzeyi gibi kriterlere göre** dikkatle yapılması gerektiğini açıkça ortaya koymaktadır. Görüntü düzeyi sınıflandırma başarısı, segmentasyon doğruluğu, sınıf ayırım yeteneği ve model çıktılarının yorumlanabilirliği gibi farklı boyutlar, uygun model mimarisinin belirlenmesinde belirleyici rol oynamaktadır.

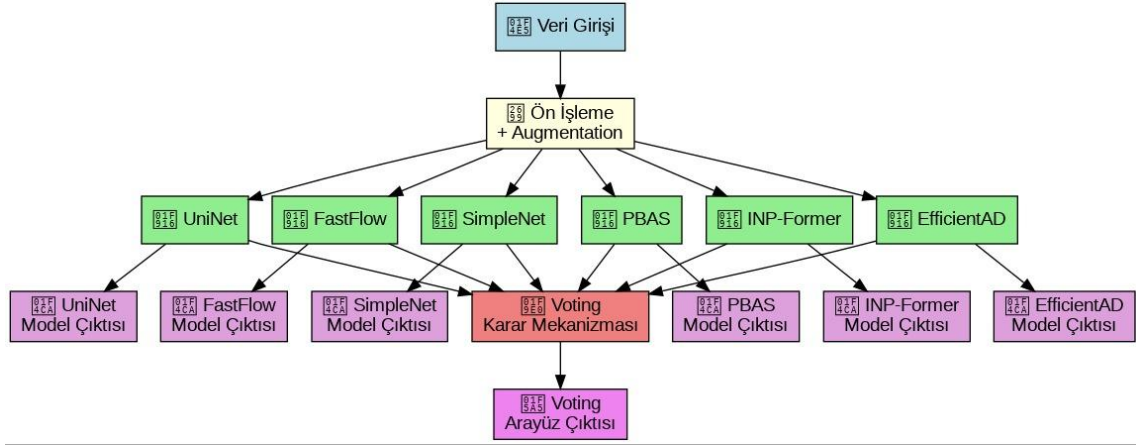
### 3. Sistem Mimarisi

Bu bölümde, geliştirilen yapay zeka tabanlı anomali tespit sisteminin genel çalışma yapısı, bileşenleri ve bu bileşenler arasındaki veri akışı detaylı olarak açıklanmaktadır. Sistem; veri girişi ve ön işleme adımlarından başlayarak, farklı anomaly detection modelleri üzerinden veri işlenmesini, modellerin bireysel veya toplu (voting) karar üretmesini ve son olarak sonuçların kullanıcı arayüzüne sunulmasını kapsayan **katmanlı ve modüler bir mimariye** sahiptir. Ayrıca, hem bağımsız model çıktıları hem de oylama tabanlı nihai kararların arayüzde nasıl görselleştirildiği şematik olarak sunulmuştur.

#### 3.1 Genel Bakış

Bu projeye ait genel sistem mimarisi, katmanlı ve modüler bir yapı temelinde tasarlanmıştır. Şekil-35'teki gibi, ilk katman olarak veri girişi (ham test veya eğitim görselleri) yer almakta, bunu takip eden aşamada veri ön işleme ve augmentation modülü ile veriler modele uygun hale getirilmektedir. İkinci katmanda, her biri bağımsız çalışan farklı anomaly detection modelleri (EfficientAD, UniNet, FastFlow, SimpleNet, PBAS, INP-Former) paralel yapıda konumlandırılmıştır. Bu modellerden elde edilen sonuçlar, üçüncü katmanda bulunan karar mekanizması (voting sistemi) üzerinden birleştirilerek ya da tek modelin sonucu üzerinden nihai sınıflandırma kararı üretilmektedir. Son olarak, tüm süreç çıktıları (anomaly map, overlay, skorlar) kullanıcıya görsel olarak sunulmak

üzere arayüz katmanına iletilmektedir. Bu yapı, sistemin modüler, genişletilebilir ve yorumlanabilir şekilde işlemlerini sağlamaktadır.



Şekil 35 Geliştirilen Anomali Tespit Sisteminin Genel Mimari Yapısı

### 3.2 Mimari Özellikleri

Geliştirilen sistem, görsel tabanlı anomali tespiti görevini uçtan uca gerçekleştiren, modüler ve genişletilebilir bir mimari üzerine kurulmuştur. İlk aşamada, sistem MVTec AD veri setinin wood alt kümesini temel alarak, görüntüleri ön işleme adımlarından (grayscale dönüşüm, Gaussian blur, Otsu eşikleme vb.) geçirir ve eğitim-test ayrımı yaparak uygun formatta modele aktarır. Bunu takiben, eğitim verilerine çeşitli **data augmentation** teknikleri uygulanarak modelin genelleme kapasitesi artırılır.

Veriler, paralel olarak yapılandırılmış EfficientAD, FastFlow, UniNet, PBAS, SimpleNet ve INP-Former modellerine yönlendirilir. Her model kendi iç yapısına uygun şekilde eğitilir ve test verisi üzerinde bireysel tahminler üretir. Bu tahminler; F1, AUROC, IoU, precision ve recall gibi metriklerle değerlendirilir. Sistem, model çıktılarının kullanıcıya iki farklı şekilde sunulmasına olanak tanır: Modeller tek başlarına sınıflandırma yapabildiği gibi, alternatif olarak bir **voting mekanizması** ile tüm modellerin kararları birleştirilerek nihai karar da üretilebilir.

Bu yapı sayesinde sistem, hem esnek hem de yorumlanabilir bir anomali tespit çözümü sunar ve kullanıcıya model bazlı veya toplu değerlendirme sonuçlarını ayrı ayrı analiz etme imkânı tanır.

### 3.3 Kullanıcı Arayüz

Projenin kullanıcı etkileşimini mümkün kılan en önemli bileşeni, web tabanlı olarak geliştirilen kullanıcı arayüzüdür. Flask tabanlı backend yapısı ve HTML/CSS/JavaScript destekli frontend yapısı sayesinde, kullanıcıların herhangi bir

teknik bilgiye ihtiyaç duymadan ahşap yüzey görüntülerini analiz edebilmeleri sağlanmıştır.

Ahşap Anomali Tespiti

TR

Ahşap Uzmanı

Özellikle ahşap malzemeler için optimize edilmiş, çatlak ve leke gibi kusurları tespit eden AI sistemleri.

Detaylı Analiz

Isı haritaları ile ahşap yüzeydeki anomalilerin görsel olarak belirlenmesi ve sınıflandırılması.

Hızlı Tarama

Saniyeler içinde sonuç veren hızlı işlem süresi ile üretim hatlarında kalite kontrolü için ideal çözüm.

Görüntü Yükleme

Görüntünüzü buraya sürükleyip bırakın  
veya yüklemek için [dosya seçin](#)

Tespit Modeli:

EfficientAD

EfficientAD, ahşap yüzey kusur tespiti için yüksek doğruluk ve makul işlem süresi ile optimize edilmiştir.

Tespit Hassasiyeti:

Orta (dengeli)

Hassasiyeti özel ahşap kalite gereksinimlerinize göre ayarlayın.

Görüntüyü Analiz Et

Analiz Sonuçları

← Yeni Analiz

Tahmin: 

Anomali

Anomali Skoru: 

0.61 / 1.25

Tüm Görüntüler

Orjinal

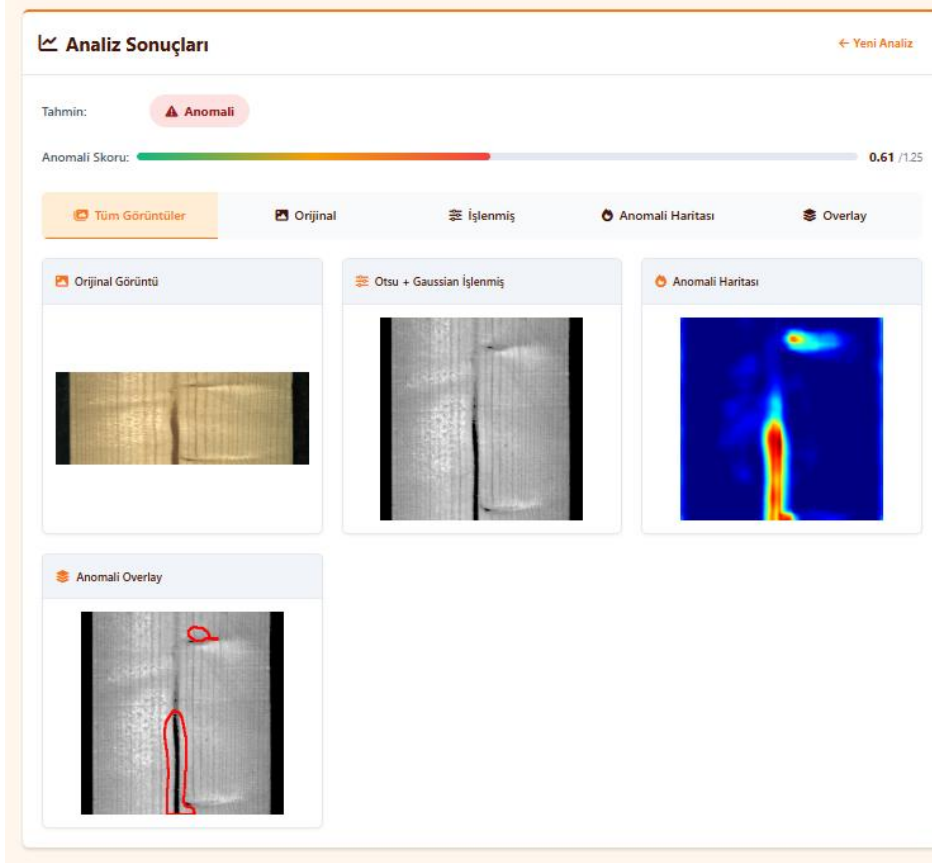
İşlenmiş

Anomali Haritası

Overlay

Şekil 36 Ana Ekran

Şekil-35'te sunulduğu gibi, ana sayfa (index.html), kullanıcıların analiz etmek istedikleri ahşap yüzey görsellerini yükleyebilmeleri için sürükle-bırak destekli bir "dropzone" bileşeni içermektedir. Kullanıcı, bu alana bir görsel bırakarak veya doğrudan sistemden seçerek yükleme yapabilmektedir. Ardından, kullanıcıya sistemde kullanılabilir beş farklı modelden biriyle (EfficientAD, UniNet, FastFlow, PBAS, INP-Former) veya toplu karar (voting) mekanizması ile analiz yapma seçeneği sunulmaktadır. Ayrıca, kullanıcılar "Düşük", "Orta" ve "Yüksek" olmak üzere üç farklı hassasiyet (threshold) seviyesi arasından seçim yaparak tespit algoritmasının duyarlılığını özelleştirebilmektedir.



Şekil 37 Analiz Sonucu Ekranı

Yükleme sonrası analiz işlemleri, Flask uygulaması tarafından app.py dosyasında tanımlı /upload endpoint'i aracılığıyla gerçekleştirilir. Burada, görselin ID'si sistemdeki mevcut görsellerle çakışmayacak şekilde dinamik olarak belirlenir ve ardından seçilen modele uygun run\_inference\_\*.py fonksiyonu çağrılarak analiz yapılır. Her model için ayrı eşik değerleri tanımlanmış olup, bu değerler get\_overlay\_threshold() fonksiyonu aracılığıyla modele ve seçilen hassasiyet seviyesine göre otomatik olarak belirlenmektedir.

Analiz sonuçları kullanıcıya hem sayısal hem de görsel olarak sunulmaktadır. Sayısal olarak, tespit edilen anomali skoru kullanıcıya "Anomali Skoru" başlığı altında, dinamik olarak büyüyen renkli bir çubuk (score-bar) eşliğinde gösterilir. Skorun yanında sınıflandırma sonucu ("Normal" ya da "Anomali") renklendirilmiş bir etiketle sunulur. Görsel olarak ise orijinal görüntü, ön işlenmiş görüntü (Otsu + Gaussian Blur), anomali haritası (heatmap) ve anomali bölgelerinin konturla işaretlendiği üst üste bindirilmiş

(overlay) görüntü ayrı kartlar halinde kullanıcıya sunulmaktadır. Şekil-36’da sunulmuştur. Bu görseller üzerine tıklanarak modal yapı içerisinde büyük önizleme de yapılabilmektedir. Eğer kullanıcı “Voting” modelini seçmişse, her bir modelin verdiği karar ayrı ayrı bir liste halinde gösterilmektedir. Bu sayede, hangi modelin anomaliyi tespit ettiği veya etmediği net biçimde analiz edilebilir. Bu mekanizma, karar verme sürecinin şeffaflığını artırmakta ve kullanıcıya daha fazla kontrol imkânı sunmaktadır. Kullanılan sistem bu sayede yalnızca bir sınıflandırma motoru değil, aynı zamanda karar destek sistemi olarak da işlev görmektedir. Ayrıca, sistem içerisinde yer alan aboutus.html dosyası ile proje geliştirme ekibi, kullanılan teknolojiler ve model performans metrikleri detaylı biçimde sunulmuştur. Bu sayfa, hem teknik kullanıcılar hem de potansiyel sistem entegratörleri için kapsamlı bir tanıtım alanı sağlamaktadır. Kullanıcılar, ekip üyelerine ait LinkedIn ve GitHub profillerine doğrudan erişim sağlayabilir, kullanılan teknolojik yığın (stack) ile ilgili bilgi sahibi olabilir. Sayfa ayrıca her bir modelin AUROC, F1, precision gibi metriklerini kutucuklar halinde sergileyerek sistemin başarısını somut verilerle desteklemektedir.

Sonuç olarak, geliştirilen web tabanlı arayüz; kullanıcı dostu, görsel açıdan zengin ve teknik olarak özelleştirilebilir bir yapı sunarak projenin pratik uygulanabilirliğini üst düzeye taşımaktadır. Arayüzün esnek yapısı sayesinde ileride yeni modellerin sisteme entegre edilmesi veya kullanıcı tercihlerine göre sistem parametrelerinin dinamik olarak ayarlanması kolaylıkla mümkündür. Bu yönüyle sistem, hem akademik çalışmalar hem de endüstriyel uygulamalar açısından yüksek potansiyele sahiptir.

#### 4. Gelecekte Yapılacak Çalışmalar

Bu bölümde, sistemin performansını artırmaya ve kullanıcı deneyimini iyileştirmeye yönelik gelecekte planlanan geliştirme adımları ele alınmaktadır. Özellikle model boyutlarının küçültülmesi ve sistemin yanıt süresinin azaltılması gibi konular, sistemin daha hızlı, hafif ve erişilebilir hale getirilmesi açısından öncelikli hedefler arasında yer almaktadır.

##### 4.1 Model Ağırlıklarının Sıkıştırılması

Projede kullanılan bazı modellerin ağırlık dosyaları oldukça büyük olup, bu durum hem yükleme süresini uzatmakta hem de sistem kaynaklarını zorlamaktadır. Gelecekteki çalışmalarda, model boyutlarının küçültülmesi hedeflenmektedir. Bunun için, modellerin daha verimli formatlarda kaydedilmesi, gereksiz parametrelerin

azaltılması ve sıkıştırma tekniklerinin kullanılması planlanmaktadır. Bu sayede, hem sistem belleği daha verimli kullanılacak hem de arayüzdeki gecikmeler azaltılacaktır.

#### 4.2 Model Çalışma Süresinin Azaltılması

Arayüzde yapılan analizlerin daha hızlı sonuç vermesi için, modelin çalışma süresinin kısaltılması önemli bir hedeftir. Bu kapsamda, daha hafif yapıya sahip modellerin tercih edilmesi veya mevcut modellerin optimize edilmesi planlanmaktadır. Ek olarak, arka planda yapılan işlemlerin daha verimli hale getirilmesiyle birlikte, kullanıcı deneyiminin iyileştirilmesi amaçlanmaktadır. Bu geliştirme ile analiz süreleri kısılacak, sistem daha akıcı ve tepkisel hale gelecektir.

#### 4.3 Gerçek Zamanlı Uygulamaya Uyumlaştırma

Sistemin endüstriyel ortamlarda doğrudan üretim hatlarına entegre edilebilmesi için, gerçek zamanlı çalışmayı destekleyen bir yapıya dönüştürülmesi hedeflenmektedir. Bu kapsamda, veri akışının anlık olarak işlenebilmesi, modelin giriş çıktılarında minimum gecikme ile tahmin yapabilmesi ve sonuçların anında görselleştirilmesi gerekmektedir. Bunun için hem model seviyesinde düşük gecikmeli (low-latency) mimariler tercih edilecek, hem de arayüz ve backend bileşenleri daha hızlı yanıt verebilecek şekilde optimize edilecektir. Ayrıca, edge cihazlarda çalışabilecek hafif model versiyonlarının geliştirilmesi de bu hedef doğrultusunda değerlendirilmektedir.

### 5. Ekler ve Proje Erişim Bilgileri

Bu bölümde, projeye ait tanıtım videosu, GitHub deposu, model ağırlıkları ve veri seti klasörü yer almaktadır. Projeyi çalıştırmak isteyen kullanıcılar aşağıdaki bağlantılar ve talimatları takip ederek sistemi kolayca kullanabilir.

#### 5.1 Proje Tanıtım Videosu

Proje tanıtım videosu için aşağıdaki bağlantıyı açabilirsiniz:

[https://drive.google.com/drive/u/0/folders/1gumPqSOsk2fXWoXKZrED\\_01gs2WUBCcz](https://drive.google.com/drive/u/0/folders/1gumPqSOsk2fXWoXKZrED_01gs2WUBCcz)

#### 5.2 GitHub ve Drive Bağlantıları

GitHub Linki:

<https://github.com/emreguener/AnomalyDetection-DeepLearning>



Model ağırlıkları ve veri seti:

<https://drive.google.com/drive/u/0/folders/1cTF4XmDJ4EKme7sKN01J4aIPPwtxAr6a>

### 5.3 Sonuçların Tekrar Üretimi İçin Adımlar

#### 1. Arayüz kısmı için:

Gerekli adımlar ve kullanım talimatları GitHub sayfasında ReadMe kısmında yer almaktadır. Eğer arayüzü kullanmak istiyorsanız şu adımları izleyiniz:

GitHub reposunu klonladıktan sonra:

- a) backend/models/ klasörü altına her modele ait .pth ağırlık dosyalarını Drive'daki yapıya uygun şekilde yerleştiriniz:
  - Örneğin: PBAS için ckpt\_best\_19.pth ve center.pth dosyaları
  - Aynı mantıkla: UniNet, FastFlow, INP-Former, EfficientAD gibi klasörler için de ilgili .pth dosyaları mevcuttur.
- b) Projenin kök dizininde terminalden Şekil-37'deki gibi komutları çalıştırarak Flask arayüzünü başlatabilirsiniz:

```
bash
cd backend
python app.py
```

Şekil 38 Arayüzü çalıştırma komutu

#### 2. Modellerin Bağımsız Eğitimi:

GitHub reposunda, her model klasörünün altında bulunan \*\_Run.ipynb uzantılı Jupyter notebook dosyaları sayesinde, ilgili modeli **arayüzden bağımsız olarak** eğitebilirsiniz.

Örneğin:

PBAS/PBAS\_Run.ipynb  
UniNet/UniNet\_Run.ipynb

Bu notebook dosyalarında, eğitim ayarlarını ihtiyacınıza göre değiştirerek modeli sıfırdan eğitebilir veya varsayılan konfigürasyonlarla çalıştırıp çıktılarınızı oluşturabilirsiniz.

## 6. Sonuç

Bu proje kapsamında, ahşap yüzeylerdeki yapısal bozuklukların tespiti için farklı mimarilere sahip denetimsiz öğrenme tabanlı anomaly detection modelleri detaylı şekilde incelenmiş, karşılaştırılmış ve uygulamaya entegre edilmiştir. Geliştirilen sistem, yalnızca kusursuz görüntülerle eğitilen yapay zekâ modelleri sayesinde etiketli veriye olan ihtiyacı ortadan kaldırmakta, hem görüntü düzeyinde sınıflandırma hem de piksel düzeyinde segmentasyon başarılarını yüksek doğrulukla gerçekleştirmektedir. Tüm modellerin farklı güçlü yönleri olduğu görülmüş; bazıları sınıflandırma başarısı, bazıları segmentasyon doğruluğu, bazıları ise açıklanabilirlik açısından öne çıkmıştır. Bu farklılıklar, sistemin çok boyutlu analiz yapabilmesini mümkün kılmıştır. Ayrıca, kullanıcı dostu ve yorumlanabilir bir arayüz ile görsel analiz süreçleri sadeleştirilmiş ve karar destek süreci güçlendirilmiştir. Genel olarak proje, endüstriyel kalite kontrol sistemlerinde kullanılabilecek, esnek, yorumlanabilir ve yüksek performanslı bir çözüm olarak değerlendirilmiştir.

## 7. KAYNAKÇA

Bu bölümde, proje boyunca kullanılan kaynaklar IEEE kaynakça formatında yazılmıştır.

[1] L. Heckler-Kram, J.-H. Neudeck, U. Scheler, R. König, and C. Steger, "The MVTec AD 2 Dataset: Advanced Scenarios for Unsupervised Anomaly Detection," *arXiv preprint arXiv:2503.21622*, Mar. 2025. [Online]. Available:

<https://arxiv.org/abs/2503.21622>

[2] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, C. Steger, "MVTec Anomaly Detection Dataset," MVTec Software GmbH, 2021. [Online]. Available:

<https://www.mvtec.com/company/research/datasets/mvtec-ad>

[3] S. Jeong, S. Park, T. Kim, and J. Choi, "UniNet: A Contrastive Learning Guided Unified Framework for Anomaly Detection and Localization," 2023. [Online].

Available: <https://paperswithcode.com/paper/uninet-a-contrastive-learning-guided-unified>

[4] J. Luo, X. Zhang, H. Wang, J. Pang, and Y. Zhang, "Exploring Intrinsic Normal Prototypes within Transformers for Unsupervised Anomaly Detection," *Papers with Code*, 2023. [Online]. Available: <https://paperswithcode.com/paper/exploring-intrinsic-normal-prototypes-within>

[5] Q. Wu, W. Liang, Z. Li, and Z. Lin, "Progressive Boundary-Guided Anomaly Synthesis for Anomaly Detection and Segmentation," *Papers with Code*, 2023.

[Online]. Available: <https://paperswithcode.com/paper/progressive-boundary-guided-anomaly-synthesis>

- [6] F. Yu, J. Li, and H. Wang, "EfficientAD: Accurate Visual Anomaly Detection at Computational Efficiency," *Papers with Code*, 2023. [Online]. Available: <https://paperswithcode.com/paper/efficientad-accurate-visual-anomaly-detection>
- [7] R. Liu, H. Xu, and P. Luo, "SimpleNet: A Simple Network for Image Anomaly Detection and Localization," *Papers with Code*, 2023. [Online]. Available: <https://paperswithcode.com/paper/simplenet-a-simple-network-for-image-anomaly>
- [8] T. Ge, C. Lu, Z. Liu, and C. Shen, "FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows," *Papers with Code*, 2022. [Online]. Available: <https://paperswithcode.com/paper/fastflow-unsupervised-anomaly-detection-and>
- [9] GitHub - pangdatangtt, "UniNet," 2023. [Online]. Available: <https://github.com/pangdatangtt/UniNet.git>
- [10] GitHub - cqylunlun, "PBAS: Progressive Boundary-guided Anomaly Synthesis," 2023. [Online]. Available: <https://github.com/cqylunlun/pbas.git>
- [11] GitHub - luow23, "INP-Former," 2023. [Online]. Available: <https://github.com/luow23/INP-Former.git>
- [12] GitHub - gathierry, "FastFlow," 2022. [Online]. Available: <https://github.com/gathierry/FastFlow.git>
- [13] GitHub - nelson1425, "EfficientAD," 2023. [Online]. Available: <https://github.com/nelson1425/EfficientAD>
- [14] GitHub - donaldr, "SimpleNet: A Simple Network for Image Anomaly Detection and Localization," 2023. [Online]. Available: <https://github.com/donaldr/simplenet>