

# **DERİN ÖĞRENME – GÜZ DÖNEMİ 2025-2026**

## **FİNAL PROJE RAPORU**

**Ders/Dönem:** Derin Öğrenme – Güz Dönemi 2025-2026

**Proje Başlığı:** Çelik Yüzey Kusuru Sınıflandırması: Baseline CNN ve Gelişmiş Derin Öğrenme Modellerinin Karşılaştırılması

**Ekip Adı:** DevENG

**Ekip Üyeleri:**

23040301067 – Mustafa ÖZBEZEK – mustafaozbezek@stu.topkapi.edu.tr

23040301045 – Mehmet Kerem HAKAN – mehmetkeremhakan@stu.topkapi.edu.tr

**GitHub Bağlantısı:**

[https://github.com/keremh9/DevENG\\_DeepLearning\\_Project\\_FET312](https://github.com/keremh9/DevENG_DeepLearning_Project_FET312)

## **2. Problem Tanımı & Motivasyon**

### **2.1 İş / Bilimsel Soru**

Sıcak haddelenmiş çelik şeritlerin yüzeyinde oluşan kusurlar (çatlak, çizik, pitting vb.) ürün kalitesini düşürmekte ve üretim maliyetini artırmaktadır. Bu kusurların çoğu fabrikalarda hâlâ insan gözüyle tespit edilmeye çalışılmakta, bu da zaman alıcı, yorucu ve hataya açık bir süreç oluşturmaktadır.

Bu projede yanıtlamaya çalıştığımız temel soru şudur:

Çelik yüzey kusurları, NEU Surface Defect veri setindeki görüntülerden yola çıkarak, basit CNN modelleri ile otomatik ve güvenilir şekilde sınıflandırılabilir mi?

Amaç, üretim hattında kullanılabilecek, hızlı ve tutarlı otomatik bir karar destek sistemi için temel bir model geliştirmektir.

### **2.2 Görev Türü**

Bu çalışma, denetimli öğrenme kapsamında bir çok sınıfı görüntü sınıflandırma problemdir.

**Girdi:** Çelik yüzeyinden alınmış gri tonlu bir görüntü

**Cıktı:** İlgili kusur sınıfı (6 sınıfından biri)

Her grup üyesi, aynı sınıflandırma problemini çözen ancak farklı mimariye sahip basit bir CNN modeli geliştirecektir.

Final aşamasında ise, bu baseline performans seviyesinin üzerine çıkmak amacıyla daha derin ve önceden eğitilmiş (transfer learning tabanlı) modeller kullanılarak kapsamlı bir performans karşılaştırması gerçekleştirılmıştır.

### **2.3 Hedef Değişkenler**

Hedef değişkenimiz: Çelik yüzey kusuru sınıfı (kategori). NEU Surface Defect veri kümesinde her görüntü şu sınıflardan birine aittir:

- İnce Çatlaklar
- Yabancı Madde
- Yamalar
- Çukurlu Yüzey
- Haddelenmiş Pül
- Çizikler

Bu değişken kategoriktir ve model içinde 0-5 arası tamsayı etiketler ile temsil edilecektir.

### **2.4 Başarı Kriterleri**

Projenin başarısını, hem tek tek modellerin performansı hem de modeller arasındaki fark üzerinden değerlendireceğiz. Kullanılacak metrikler:

- Accuracy
- Makro Precision
- Makro Recall

-Makro F1-score

-Confusion matrix

Nicel hedeflerimiz:

Test accuracy'nin en az %75 olması

Makro F1-score'un en az 0,40 seviyesine ulaşması

### **3) Proje Yönetimi**

#### **3.1 Önemli Noktalar ve Zaman Çizelgesi**

Bu projede, vize yerine geçecek proje ilerleme raporunu hazırlarken aşağıdaki adımları izledik:

##### **Hafta 1:**

Proje konusunun belirlenmesi (çelik yüzey kusurlarının sınıflandırılması).

NEU Surface Defect veri setinin seçilmesi ve indirilmesi.

Sorumlu: Mustafa & Kerem (ortak).

##### **Hafta 2:**

Veri seti klasör yapısının düzenlenmesi.

Örnek görüntülerin incelenmesi ve sınıfların kontrol edilmesi.

Basit veri ön işleme adımlarının (gri tonlama, yeniden boyutlandırma, normalize etme) planlanması.

Sorumlu: Mustafa & Kerem (ortak).

##### **Hafta 3:**

Ortak train/validation/test ayrimının yapılması.

PyTorch Dataset ve DataLoader yapılarının oluşturulması.

Sorumlu: Kerem (ana), Mustafa (destek).

##### **Hafta 4:**

Kerem için basit CNN modeli mimarisinin tasarılanması ve ilk eğitim denemeleri.

Sorumlu: Kerem.

##### **Hafta 5:**

Mustafa ise farklı bir CNN mimarisi tabanlı bir model tasarlanması ve eğitilmesi.

Sorumlu: Mustafa.

##### **Hafta 6:**

Her iki modelin test edilmesi, accuracy, precision, recall, F1 ve confusion matrix sonuçlarının çıkarılması.

Modellerin performanslarının tabloda karşılaştırılması ve rapora yorumların yazılması.

Sorumlu: Mustafa & Kerem (ortak).

#### **Hafta 7:**

Proje İlerleme Raporu metninin yazılması, düzenlenmesi ve PDF hâline getirilmesi.

ALMS'e dijital yükleme ve basılı kopyanın hazırlanması.

Sorumlu: Mustafa & Kerem (ortak).

### **3.2 Roller ve Sorumluluklar**

#### **Mustafa ÖZBEZEK (23040301067)**

Farklı CNN tabanlı model mimarisini tasarlamak ve kodlamak.

Kendi modeline özel veri ön işleme adımlarını belirlemek.

Modelinin eğitim ve bağımsız test sonuçlarını hesaplayıp yorumlamak.

#### **Mehmet Kerem HAKAN (23040301045)**

Ortak veri yükleme, veri kümesinin bölünmesi ve PyTorch kurulumunu yapmak.

Kendi base modeli olan SimpleCNN\_Kerem mimarisini tasarlamak ve kodlamak.

Modelinin eğitim ve bağımsız test sonuçlarını hesaplayıp yorumlamak.

#### **Ortak Sorumluluklar (Mustafa & Kerem):**

Proje İlerleme Raporu'nun (PDF) birlikte yazılması ve düzenlenmesi.

Base modellerin performanslarının aynı tabloda karşılaştırılması ve aradaki farkların tartışılması.

GitHub deposunun oluşturulması ve kod dosyalarının yüklenmesi.

### **3.3 Çıktılar**

Proje kapsamında üretilen ve üretilerek başlıca çıktılar şunlardır:

#### **Ortak Proje İlerleme Raporu (PDF):**

Tüm bölümleri içeren, her iki base modelin de anlatıldığı ve karşılaştırıldığı rapor.

#### **Jupyter Notebook Kod Dosyaları:**

FET312\_23040301045\_DevENG\_1.ipynb → Kerem'in CNN modeli ve sonuçları.

FET312\_23040301067\_DevENG\_2.ipynb → Mustafa'nın CNN modeli ve sonuçları.

#### **GitHub Deposu (tek ortak repo):**

[https://github.com/keremh9/DevENG\\_DeepLearning\\_Project\\_FET312](https://github.com/keremh9/DevENG_DeepLearning_Project_FET312)

### **4.1 Literatürdeki Temel Çalışmalar**

[1] X. Song et al., “Steel Surface Defect Classification Using Deep Convolutional Neural Networks,” 2019.

Kapsam: Çelik yüzey kusurlarını CNN tabanlı modeller ile sınıflandırmayı amaçlar.

Yöntem: 4 konvolüsyon bloğu içeren orta derinlikte bir CNN modeli.

Veri Seti: NEU Surface Defect Dataset (Bizimle aynı veri seti.)

Metrikler: Accuracy ve F1-score.

Sonuçlar: %92–93 doğruluk seviyelerine ulaşılmıştır.

Bizim farkımız:

Bu çalışmada kullanılan model daha derin bir yapıya sahipken, biz daha basit iki farklı CNN modeli geliştirerek, daha düşük karmaşıklıkla kabul edilebilir performans elde etmeyi hedefliyoruz.

## [2] J. Li and Y. He, “Surface Defect Detection Based on Shallow Convolutional Networks,” 2020.

Kapsam: Daha hafif CNN modellerinin endüstriyel yüzeylerde nasıl performans gösterdiğini inceler.

Yöntem: 2–3 konvolüsyon katmanlı basit CNN modelleri.

Veri Seti: Endüstriyel yüzey veri setleri (NEU ile benzer yapı)

Metrikler: Accuracy, Precision, Recall.

Sonuçlar: Hafif modellerin hızlı eğitilebildiği ve gerçek zamanlı üretim hatlarında daha uygun olduğu gösterilmiştir.

Bizim farkımız:

Bu makalenin önerdiği yaklaşım bize benzer olsa da, biz aynı veri üzerinde iki farklı hafif CNN mimarisini karşılaştırarak, hangi mimari tasarımının daha etkili olduğunu ortaya koyuyoruz.

## [3] Kaggle Benchmark Çalışmaları – NEU Surface Defect Dataset

Kapsam: Kaggle üzerinde NEU veri seti ile yapılan kullanıcı projeleri.

Yöntem: Çoğunlukla transfer learning, bazı basit CNN modelleri.

Veri Seti: NEU (6 sınıf).

Sonuçlar: Transfer learning modelleri %95+ doğruluğa ulaşırken, basit CNN modellerinin çoğu %80–88 arası sonuçlar vermektedir.

Bizim farkımız:

Transfer learning yerine, Yıldız hocamızın istediği gibi tamamen kendi tasarladığımız iki basit CNN modeli ile ilerledik ve model karmaşıklığı düşük olmasına rağmen makul performans elde etmeyi hedefledik.

Bu projede yaptığımız çalışma, literatürdeki mevcut çalışmalarдан şu yönleriyle farklılaşmaktadır:

Aynı veri seti üzerinde iki farklı CNN mimarisinin sistematik olarak karşılaştırılması.

Eğitimi kolay, parametre sayısı düşük modeller ile bir temel seviye benchmark oluşturulması.

Transfer learning yerine tamamen sıfırdan tanımlanan basit derin öğrenme mimarilerinin performansının değerlendirilmesi.

Final aşamasında kullanılacak gelişmiş modeller için referans/karşılaştırma zemini yaratılması.

Bu yönleriyle projemiz, NEU veri setinde basit yapıdaki CNN modellerinin performansını belgeleyen, kısa ve anlaşılır bir temel çalışma niteliği taşımaktadır.

## 5) Veri Açıklaması ve Yönetimi

### 5.1 Veri Kümesi Açıklaması (Dataset Description)

Bu projede NEU-DET (Northeastern University Surface Defect Dataset) kullanılmıştır.

Veri kümesi, metal yüzeylerdeki kusurları sınıflandırmak amacıyla hazırlanmış görüntülerden oluşur. *NEU-DET veri seti, çelik yüzey kusurlarının sınıflandırılmasında yaygın olarak kullanılan bir benchmark veri setidir (Song ve ark., 2013).*

NEU-DET veri seti, sınıflar arası dengeli bir dağılıma sahip olacak şekilde önceden eğitim ve test kümelerine ayrılmıştır. **StratifiedKFold** mantığına uygun olarak, veri seti sınıf dağılımı korunacak şekilde (stratified hold-out) ayrılmıştır.

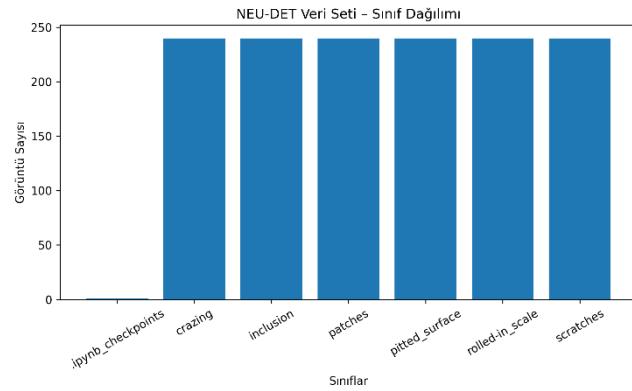
Kaynak: <https://www.kaggle.com/datasets/kaustubhdikshit/neu-surface-defect-database>

Format: JPEG görüntüler

Lisans/Kullanım: Akademik araştırma amaçlı kullanım için uygundur.

Görüntü İçeriği: Metal yüzeylerde 6 farklı kusur sınıfı bulunmaktadır.

Bu **histogram**, NEU-DET veri setinde bulunan altı farklı yüzey kusuru sınıfına ait görüntü sayılarının her sınıf için eşit olacak şekilde dengeli dağıldığını göstermektedir.



### 5.2 Şema (Features / Attributes)

Modelde kullanılan her görüntü aşağıdaki ön işleme adımlarından geçirilir:

Giriş: 224×224 çözünürlük

Kanal: 1 [Transfer learning modellerinde, gri tonlu görüntüler 3 kanala kopyalanarak (grayscale → pseudo-RGB) ağa verilmiştir.]

Özellikler: Piksel yoğunluk değerleri (0–255 → normalize 0–1)

Beklenen aralık: Normalized mean = 0.5, std = 0.5

Ekstra metinsel veya sayısal özellik bulunmamaktadır; yalnızca görüntü temelli bir veri küməsidir.

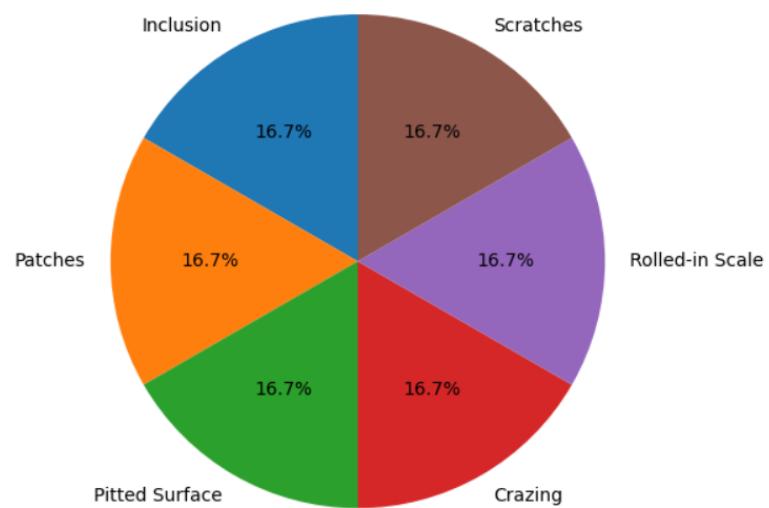
### 5.3 Veri Seti Boyutu:

Bu çalışmada kullanılan NEU Surface Defect veri seti toplam 1800 adet görüntünden oluşmaktadır. Veri seti 6 farklı yüzey kusuru sınıfı içermekte olup, her sınıfın ait 300 adet görüntü bulunmaktadır. Veri setinin sınıf dağılımı dengelemiştir, her bir sınıfın veri kümesi içinde eşit sayıda temsil edilmektedir.

Şekil X, veri setindeki sınıfların dağılımını göstermektedir.

Dengeli sınıf dağılımı sayesinde model performansı, sınıf dengesizliğinden kaynaklı bir yanlışlığa maruz kalmamaktadır.

Class Distribution of NEU Surface Defect Dataset



### 5.3 Veri Bölme (Train–Validation–Test Split)

Model eğitimi ve değerlendirmesi için veri seti, sınıf dağılımı korunacak şekilde stratified olarak bölünmüştür.

Veri seti %70 eğitim, %15 doğrulama ve %15 test olacak biçimde ayrılmıştır. Bu bölme işlemi sırasında her sınıfın eğitim, doğrulama ve test kümelerinde oransal olarak eşit temsil edilmesine dikkat edilmiştir.

Bu yaklaşım sayesinde modeller, tüm sınıflar için adil ve dengeli bir şekilde eğitilmiş ve değerlendirilmiştir.

### 5.4 Etik, Gizlilik ve Önyargı Değerlendirmesi

NEU Surface Defect veri kümesi:

Kişisel veri içermez.

İnsan yüzü, biyometrik veri, kişisel kimlik bilgisi yoktur.

Endüstriyel görüntülerden oluşur.

Çelik yüzeylerinin mikroskopik kusurlarını gösteren toplu üretim görüntüleridir.

Gizlilik riski yoktur.

Belli bir fabrika, kişi veya kuruma ait tanımlayıcı bilgi bulunmamaktadır.

Bu veri seti tamamen çelik yüzey fotoğraflarındanoluğu için herhangi bir kişisel veri veya gizlilik riski yoktur.

Bu nedenle veri seti, etik veya gizlilik açısından ek bir risk barındırmayan, tamamen endüstriyel ve güvenli bir veri kümesidir.

## 6) Yöntemler ve Mimari

### 6.1 Baseline Modeller (Vize Aşaması)

Bu çalışmanın vize aşamasında, NEU Surface Defect veri seti üzerinde temel bir referans performans seviyesi oluşturmak amacıyla, tamamen sıfırdan tasarlanmış iki farklı basit CNN mimarisi geliştirilmiştir.

Bu baseline modellerin temel amacı, transfer learning veya önceden eğitilmiş derin mimariler kullanılmadan, düşük hesaplama maliyetine sahip CNN yapılarının yüzey kusuru sınıflandırma problemindeki başarısını değerlendirmektir.

Geliştirilen bu modeller, final aşamasında uygulanacak daha gelişmiş mimariler ile karşılaştırılmak amacıyla kullanılmıştır.

### 6.2 Yöntemler

Bu projede çelik yüzey kusurlarını görüntülerden otomatik olarak sınıflandırmak için basit Convolutional Neural Network (CNN) modelleri kullanılmıştır.

Amaç: Kolay eğitilen, düşük hesaplama maliyetli iki farklı CNN mimarisi geliştirmek,

Aynı veri üzerinde performanslarını karşılaştırmak,

Final aşaması için temel bir referans (benchmark) oluşturmaktır.

İzlediğimiz genel adımlar:

Görselleri gri tonlu,  $224 \times 224$  boyutuna getirmek

Normalize etmek

Hafif veri artırma (augmentation) uygulamak

Eğitim / doğrulama / test bölmesi yapmak

İki farklı basit CNN modeli eğitmek

Accuracy, Precision, Recall, F1 ve Confusion Matrix ile değerlendirmek

Sonuçları karşılaştırarak yorumlamak

*Vize aşamasında geliştirilen baseline CNN modelleri, düşük karmaşıklıklı mimarilerin performans sınırlarını ortaya koymuştur. Final aşamasında ise bu sınırların aşılması ve daha güçlü temsil öğrenimi sağlanması amacıyla, daha derin ve önceden eğitilmiş mimariler tercih edilmiştir.*

### 6.3 Gelişmiş Modeller Hyperparameter Tuning ve Eğitim Stratejileri

Final aşamasında kullanılan tüm modellerin performansını artırmak amacıyla, eğitim sürecinde çeşitli hiperparametreler üzerinde manuel deneme–yanılma temelli hiperparametre ayarlamaları (manual hyperparameter tuning) gerçekleştirilmiştir. Bu kapsamda, özellikle learning rate, epoch sayısı, batch size ve optimizer seçimi gibi temel hiperparametreler farklı kombinasyonlar hâlinde denenmiştir. Modellerin doğrulama kümesi üzerindeki performansları dikkate alınarak en uygun parametreler belirlenmiştir.

Transfer learning kullanılan modellerde, önceden eğitilmiş ağların ilk katmanları başlangıçta dondurulmuş (freeze) ve yalnızca sınıflandırma katmanları eğitilmiştir. Daha sonraki

aşamalarda ise ağıın son konvolüsyon blokları kademeli olarak açılarak (unfreeze) fine-tuning uygulanmıştır.

Eğitim sürecinde aşırı öğrenmeyi (overfitting) önlemek amacıyla erken durdurma (early stopping) ve uygun öğrenme oranı planlayıcıları (learning rate schedulers) kullanılmıştır. Tüm bu ayarlamalar sonucunda, **her model için doğrulama performansı en yüksek olan hiperparametre kombinasyonu** seçilerek test aşamasına geçilmiştir.

## 6.4 Veri Ön İşleme ve Eğitim Süreci

Boyutlandırma: Tüm görüntüler  $224 \times 224$  px'e ölçeklendi.

Kanal: Görüntüler gri tonlu (1 kanal) olarak işlendi.

Normalize: Pixel değerleri  $[-1, 1]$  aralığına getirildi.

Augmentation:

Kerem: Sadece horizontal flip

Mustafa: Flip + rotation + random crop

Eğitim Süreci:

Optimizer: Adam

Epoch: 15

Batch size: 32

Loss fonksiyonu: CrossEntropyLoss

## 6.3 Kerem'in Modeli

Bu model, yüzey kusurlarını sınıflandırmak için tasarlanmış hafif ve temel bir CNN mimarisidir. Model şu katmanlardan oluşur:

Conv2D (16 filtre,  $3 \times 3$ ) + ReLU + MaxPool( $2 \times 2$ )

Conv2D (32 filtre,  $3 \times 3$ ) + ReLU + MaxPool( $2 \times 2$ )

Conv2D (64 filtre,  $3 \times 3$ ) + ReLU + MaxPool( $2 \times 2$ )

Flatten

Fully Connected (128 nöron) + ReLU + Dropout(0.3)

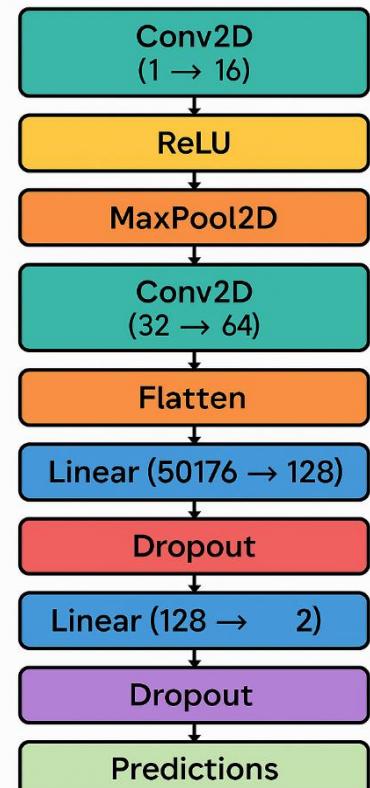
Çıkış Katmanı: Linear( $128 \rightarrow 6$  sınıf)

Bu yapı, düşük hesaplama maliyeti ile temel kenar, doku ve bölgesel özelliklerin aşamalı olarak

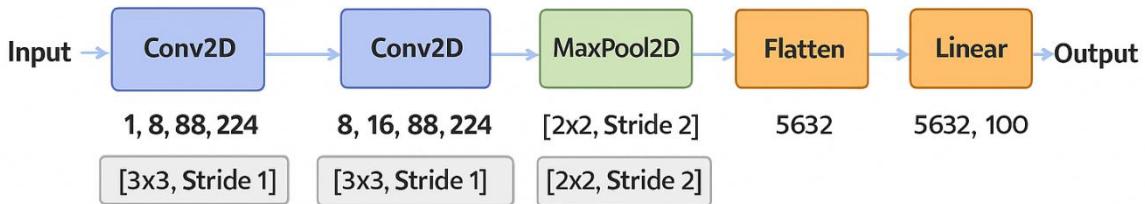
çıkarılmasını sağlar. Projede "Kerem modeli" olarak kullanılan bu CNN, daha küçük ve hızlı bir mimari olup karşılaştırma amacıyla seçilmiştir.

## 6.4 Mustafa'nın Modeli

### Simple CNN – Kerem



# Mustafa'nın Basit CNN Mimarisi



Bu model Kerem'in modeline göre bir tık daha gelişmiş olup 1 adet gizli katman içerir.

Bu model, kusurları daha geniş bir alanda yakalamak için  $5 \times 5$  kernel kullanan, iki konvolüsyon bloğundan oluşan hafif bir CNN yapısıdır.

## Mimari:

Conv2D (32 filtre,  $5 \times 5$ ) + ReLU

MaxPool ( $2 \times 2$ )

Conv2D (64 filtre,  $5 \times 5$ ) + ReLU

MaxPool ( $2 \times 2$ )

Flatten

Dense (128 nöron) + ReLU + Dropout(0.3)

## Neden bu mimari?

$5 \times 5$  kernel  $\rightarrow$  yüzey dokusunu daha geniş alanda analiz eder.

Az katman  $\rightarrow$  daha hızlı eğitim.

Dropout  $\rightarrow$  aşırı öğrenmeyi azaltır.

## Avantajlar:

Geniş kernel ile doku detaylarını iyi yakalar.

Eğitim süresi hızlıdır.

## Dezavantajlar:

Derin modeller kadar güçlü değildir.

Dengesiz veri setinde hassasiyet düşebilir.

## 7) Deney Tasarımı

Bu projede, iki farklı CNN mimarisinin performansını karşılaştırmak amacıyla iki temel deney tasarlanmıştır. Her deney, aynı veri kümesi üzerinde çalışacak şekilde planlanmıştır ve adımlar yeniden uygulanabilir olacak biçimde açıklanmıştır.

### **Deney: Basit CNN Modeli-Temel Performans Deneyi**

#### **Ana Amaç**

Bu deneyde, vize aşamasında geliştirilen basit CNN mimarisinin NEU Surface Defect veri seti üzerindeki temel performansı ölçülmüştür.

Amaç, daha gelişmiş modellerle yapılacak karşılaştırmalar için bir referans (baseline) performans seviyesi oluşturmaktır.

#### **Train / Test Ayrımı**

Veri seti %70 eğitim, %15 doğrulama, %15 test olacak şekilde bölünmüştür.

Bölme işlemi sabit bir seed ile yapılmıştır (seed=42).

Eğitim sırasında validation seti her epoch sonunda değerlendirme için kullanılmıştır.

#### **Hiperparametre Ayarları**

Epoch sayısı: 15

Learning rate: 0.001

Optimizer: Adam

Loss fonksiyonu: CrossEntropyLoss

Batch size: 32

#### **Değerlendirme Kriterleri**

Bu model şu metriklerle değerlendirilmiştir:

**Accuracy:** Genel başarı oranı

**Precision (macro):** Sınıflara eşit önem verildiği için seçildi

**Recall (macro):** Her sınıfın ne kadar doğru yakalandığını ölçer

**F1-score (macro):** Dengesizlik yok ama daha stabil bir metrik sağlar

**Confusion matrix:** Sınıfların birbirileyle karışma durumlarını gösterir

Bu deney sonunda elde edilen metrikler, ikinci modelin sonuçlarıyla karşılaştırılmıştır.

### **Gelişmiş Derin Öğrenme Modelleri (Final Aşaması)**

Final aşamasında, vize çalışmasında elde edilen baseline performans seviyesinin üzerine çıkmak amacıyla, daha derin ve modern derin öğrenme mimarileri kullanılarak ek deneyler gerçekleştirilmiştir.

Gelişmiş modellerin tamamı, adil bir karşılaştırma sağlamak amacıyla aynı veri bölmesi, aynı veri ön işleme adımları ve aynı performans metrikleri kullanılarak eğitilmiş ve değerlendirilmiştir.

Bu çalışmada Mehmet Kerem HAKAN tarafından ikinci bir farklı mimari yerine, aynı mimarinin (ResNet18) iki farklı transfer learning stratejisine (Feature Extraction vs. Fine-Tuning) verdiği tepkiyi ölçmek hedeflenmiştir. Literatürde ResNet18'in endüstriyel yüzey kusurlarında sıkça kullanılması nedeniyle, bu mimarinin 'dondurulmuş ağırlıklar' ile 'yeniden eğitilen ağırlıklar' arasındaki performans farkının analizi, farklı bir mimari denemekten daha kritik bir bulgu olarak değerlendirilmiştir. Bu sayede modelin veri setine adaptasyon yeteneğini izole edilerek incelenmiştir.

### **Ortak Deney Protokolü**

Tüm deneylerde, modellerin adil ve karşılaştırılabilir sonuçlar üretmesi amacıyla ortak bir deney protokolü uygulanmıştır.

Bu protokol kapsamında, tüm modeller aynı stratified veri bölmesi üzerinde eğitilmiş, aynı giriş görüntü boyutu ( $224 \times 224$ ), aynı veri artırma stratejileri ve aynı değerlendirme metrikleri kullanılmıştır.

Bu yaklaşım sayesinde, elde edilen performans farklarının veri veya eğitim koşullarından değil, doğrudan model mimarilerinden kaynaklandığı garanti altına alınmıştır.

### **Modeller Arası Değerlendirme ve Başarı Sıralaması**

Bu projede iki farklı CNN modeli geliştirilmiştir:

SimpleCNN\_Kerem (baseline)

SimpleCNN\_Mustafa (gizli katmanlı, kernel  $5 \times 5$ )

### **Başarı Sıralaması Nasıl Belirlenecek?**

Modeller aynı test seti üzerinde test edilmiştir.

Her model için accuracy, precision, recall ve F1-score değerleri karşılaştırılacaktır.

Ek olarak confusion matrix üzerinden yanlış sınıflandırmalar analiz edilecektir.

En yüksek F1-score ve Accuracy değerine sahip model "en başarılı" kabul edilecektir.

## **8) Sonuçlar**

Bu bölümde, vize aşamasında geliştirilen baseline CNN modelleri ile final aşamasında uygulanan gelişmiş derin öğrenme mimarilerinin performans sonuçları sunulmakta ve karşılaştırılmaktadır.

Modellerin başarımı, test veri seti üzerinde hesaplanan accuracy, macro precision, macro recall ve macro F1-score metrikleri kullanılarak değerlendirilmiştir. Ayrıca, sınıf bazlı performans farklılıklarını incelemek amacıyla sınıf bazlı metrikler ve confusion matrix sonuçları analiz edilmiştir.

### Tüm Modeller İçin Genel Performans Karşılaştırması (Test Seti)

Model	Accuracy	Macro Precision	Macro Recall	Macro F1-score
SimpleCNN – Kerem	0.8309	0.7627	0.5229	0.5744
SimpleCNN – Mustafa	0.8199	0.6955	0.5063	0.5478
ResNet18 (Frozen) – Kerem	0.9583	0.9608	0.9583	0.9575
ResNet18 (Fine-tune) – Kerem	1.0000	1.0000	1.0000	1.0000
EfficientNet-B0 (Frozen) – Mustafa	0.9778	0.9799	0.9778	0.9775
MobileNetV3-Small (Frozen) – Mustafa	0.9333	0.9442	0.9333	0.9337

### Modellerin Sınıf Bazlı Macro F1-score Karşılaştırması (Test Seti)

Sınıf	ResNet18 (Frozen) – Kerem	ResNet18 (Fine-tune) – Kerem	EfficientNet-B0 (Frozen) – Mustafa	MobileNetV3-Small (Frozen) – Mustafa
Crazing	0.9836	<b>1.0000</b>	1.0000	0.9744
Inclusion	0.9189	<b>1.0000</b>	0.9286	0.8491
Patches	0.9917	<b>1.0000</b>	1.0000	0.9744
Pitted Surface	0.9508	<b>1.0000</b>	0.9917	0.8613
Rolled-in Scale	0.9916	<b>1.0000</b>	1.0000	1.0000
Scratches	0.9600	<b>1.0000</b>	0.9449	0.9431

Deney sonuçları incelendiğinde, genel perforans metrikleri açısından en başarılı modelin **ResNet18 (Fine-tune)** olduğu görülmektedir. Bu model, özellikle Inclusion ve Pitted Surface gibi sınıflandırılması gereke zor yüzey kusurlarında elde ettiği yüksek macro F1-score değerleri ile dikkat çekmektedir.

Bu başarımın temel nedenleri arasında, modelin daha derin bir mimariye sahip olması, önceden eğitilmiş ağırlıklar sayesinde daha güçlü ve ayırt edici özellikler çıkarılabilmesi ve uygulanan **fine-tuning** stratejisinin modele veri setine özgü temsil öğrenme yeteneği kazandırması gösterilebilir.

Baseline CNN modelleri ile karşılaştırıldığında, gelişmiş derin öğrenme modellerinin özellikle karmaşık ve birbirine benzer yüzey kusurlarını ayırt etme konusunda daha başarılı olduğu gözlemlenmiştir.

Modellerin sınıf bazlı hata dağılımlarını daha ayrıntılı incelemek amacıyla, her model için test seti üzerinde elde edilen confusion matrix sonuçları Ekler bölümünde sunulmuştur.

## **9) Kullanılan Araçlar ve Frameworkler**

Bu projede derin öğrenme modellerini geliştirmek için PyTorch kütüphanesi kullanılmıştır. PyTorch, CNN modellerinin kolayca tanımlanmasını, eğitilmesini ve değerlendirilebilmesini sağlamıştır.

### **Kütüphaneler ve Ortam**

Python: 3.10

PyTorch: 2.x

Torchvision: 0.x

NumPy: Görüntü verisi işlemleri

Pandas: Küçük ölçekli veri düzenleme

Scikit-learn: Accuracy, Precision, Recall, F1-score, confusion matrix hesaplama

Matplotlib: Performans grafiklerinin çizimi

Seed: Tüm deneylerde tekrar edilebilirlik için seed=42 kullanılmıştır.

Donanım:

GPU varsa CUDA kullanılabilir

CPU ortamında da çalışabilir.

### **Geliştirilen Kodlar**

Tüm Jupyter Notebook dosyaları ve proje kodları ortak GitHub deposunda paylaşılacaktır:

GitHub Repo:

[https://github.com/keremh9/DevENG\\_DeepLearning\\_Project\\_FET312](https://github.com/keremh9/DevENG_DeepLearning_Project_FET312)

## **10) Kaynaklar**

[1] Kaggle, "NEU Surface Defect Database," 2019. Erişim adresi:

<https://www.kaggle.com/datasets/kaustubhdikshit/neu-surface-defect-database> (Erişim Tarihi: Ocak 2026).

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep residual learning for image recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016, <https://doi.org/10.1109/CVPR.2016.90>.

[3] Junyed Li, Yigang He, "Surface defect detection based on shallow convolutional networks," Applied Surface Science, Volume 503, 144366, 2020, <https://doi.org/10.1016/j.apsusc.2019.144366>.

[4] Kechen Song, Yunhui Yan, "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects," Applied Surface Science, Volume 285, pp. 358-368, November 2013, <https://doi.org/10.1016/j.apsusc.2013.08.072>.

[5] Scikit-learn Developers, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, Volume 12, pp. 2825–2830, 2011. Erişim adresi: <https://scikit-learn.org>.

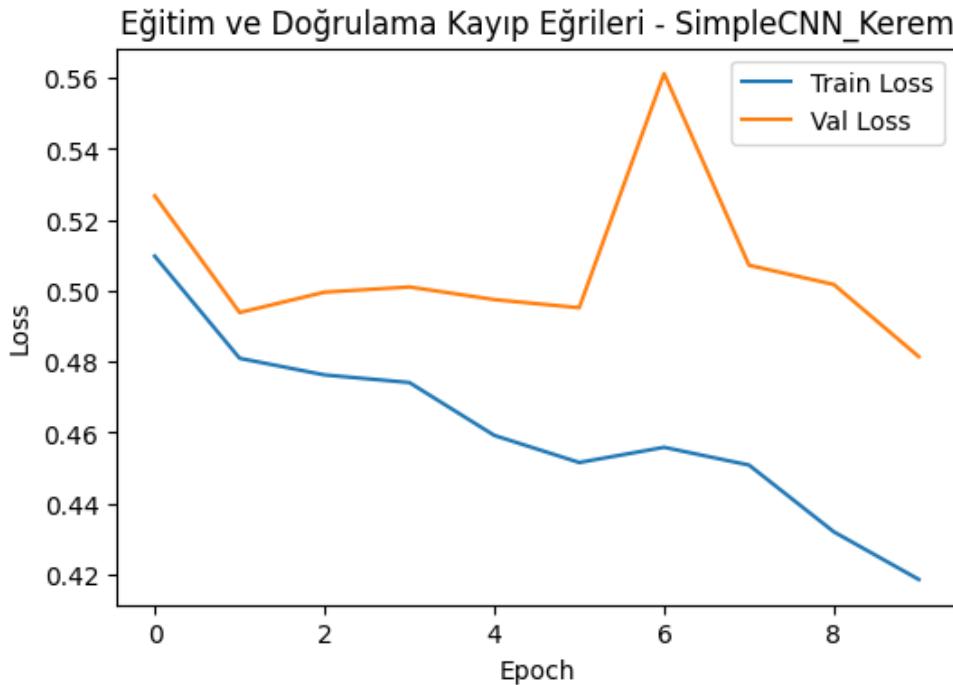
[6] Mingxing Tan, Quoc V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," International Conference on Machine Learning (ICML), pp. 6105-6114, 2019. Erişim adresi: <http://proceedings.mlr.press/v97/tan19a.html>.

## 11) Appendix(Ek)

### Baseline Modeller

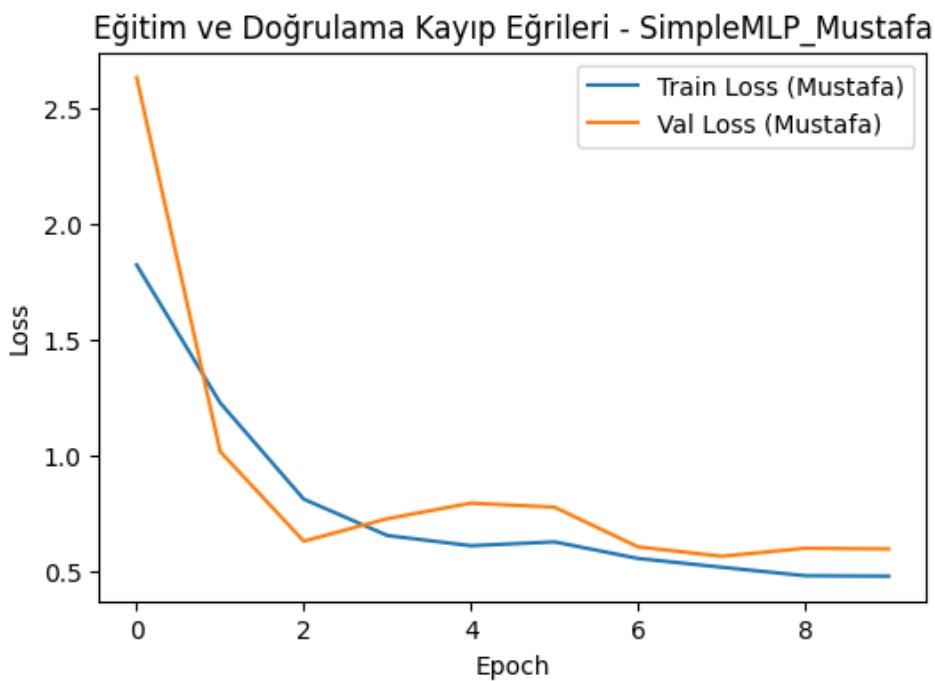
#### Kerem – Eğitim/Doğrulama Kayıp Eğrisi

CNN modelinin eğitim sürecindeki kayıp değişimlerini göstermektedir.



#### Mustafa – Eğitim/Doğrulama Kayıp Eğrisi

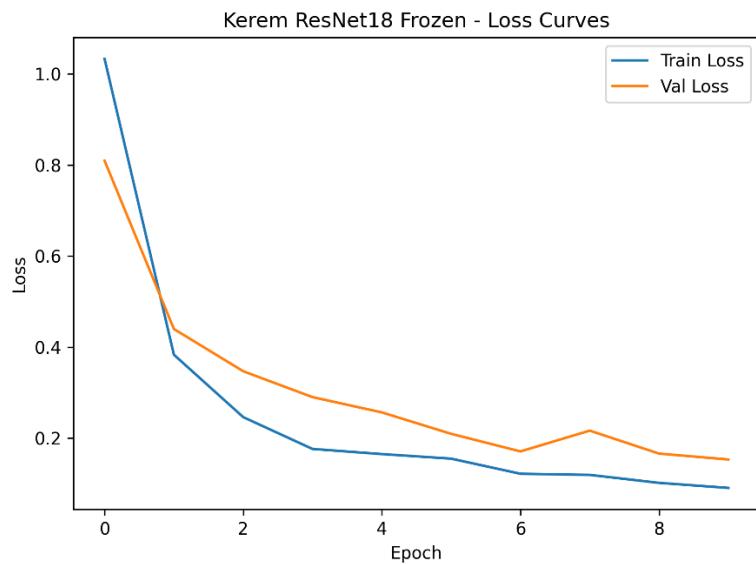
Modelin epoch boyunca eğitim ve doğrulama kayıp değişimini göstermektedir.



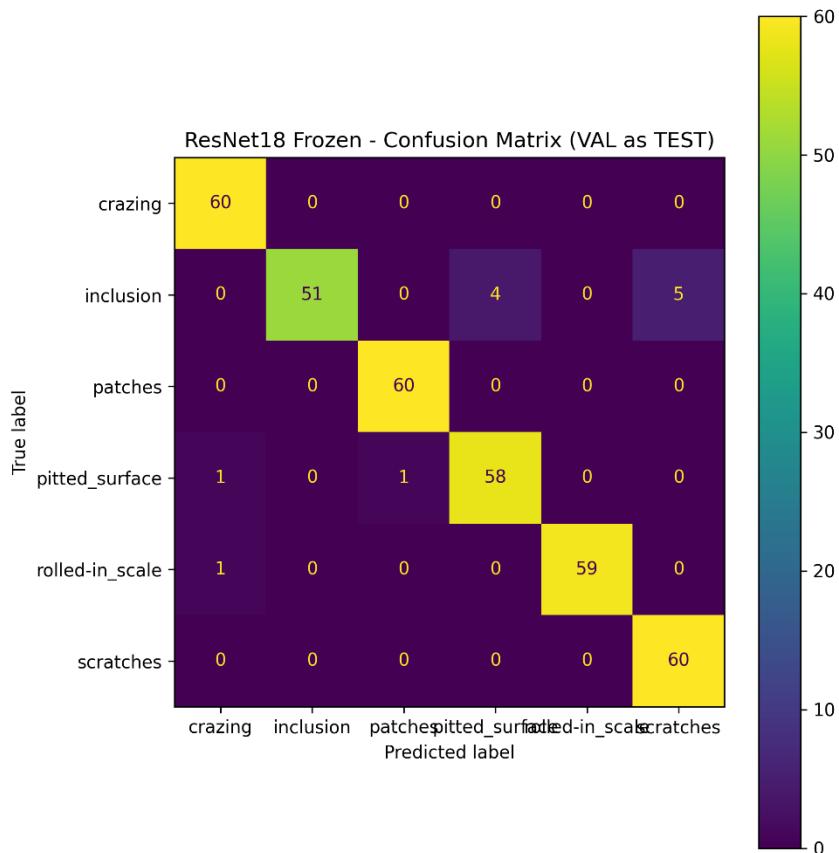
## Gelişmiş Modeller

### ResNet18 (Frozen) – Kerem

ResNet18 (Frozen) modeline ait eğitim ve doğrulama kayıp (loss) eğrileri.

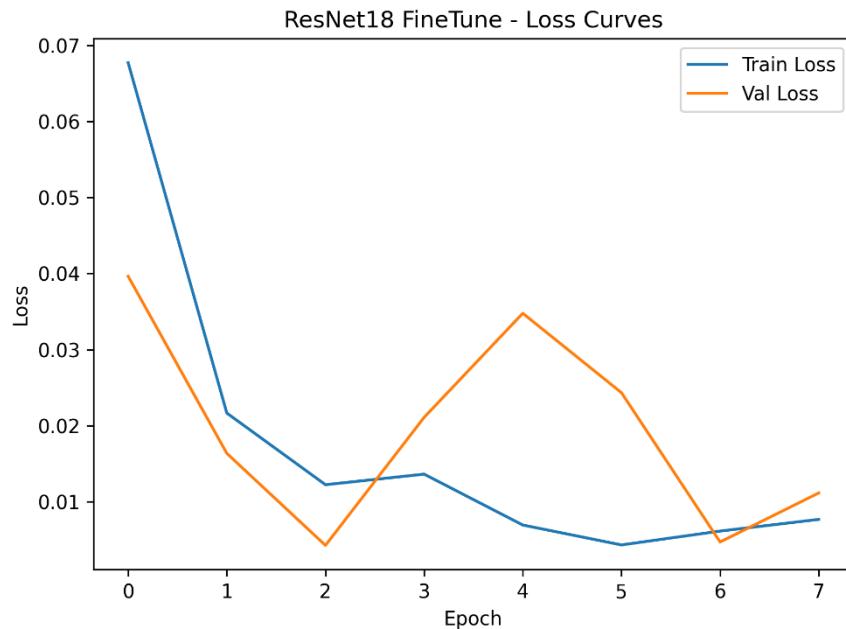


ResNet18 (Frozen) modelinin test seti üzerindeki confusion matrix sonucu.

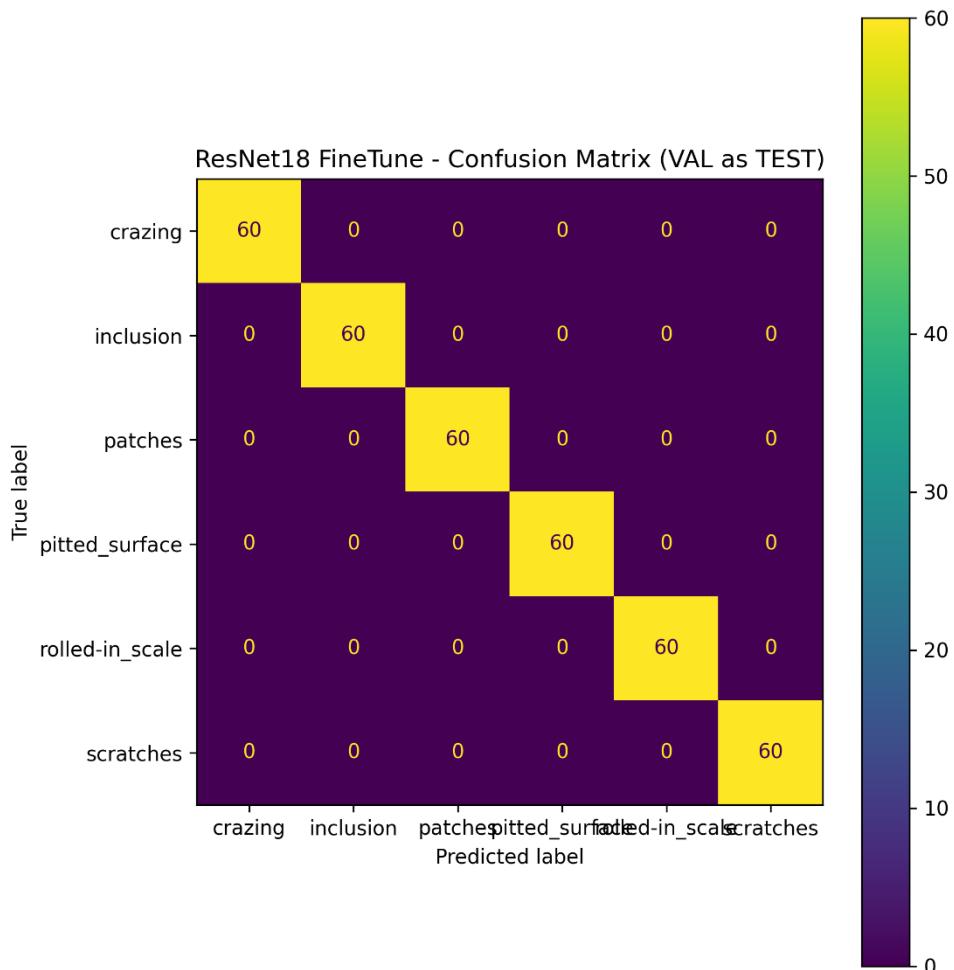


## ResNet18 (Fine-tune) – Kerem

ResNet18 (Fine-tune) modeline ait eğitim ve doğrulama kayıp (loss) eğrileri.

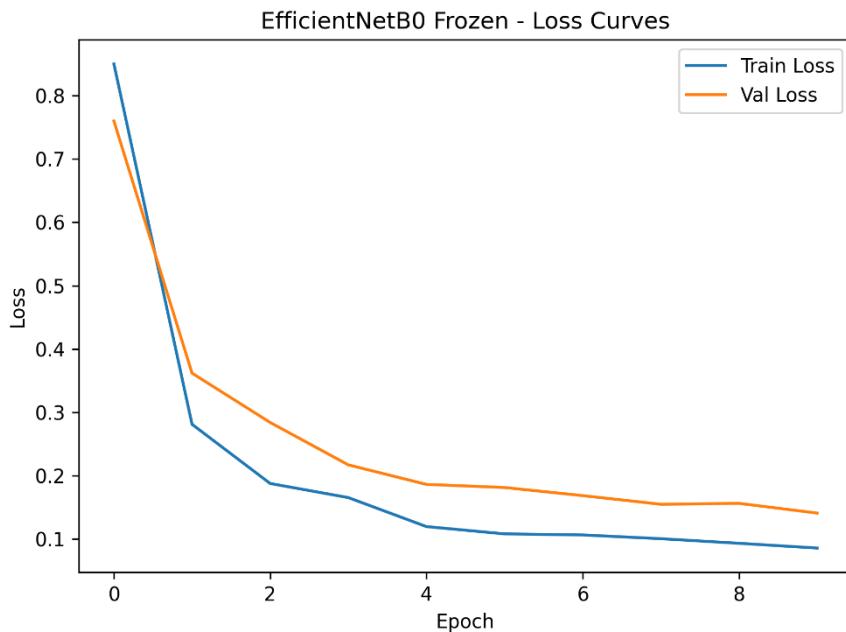


ResNet18 (Fine-tune) modelinin test seti üzerindeki confusion matrix sonucu.

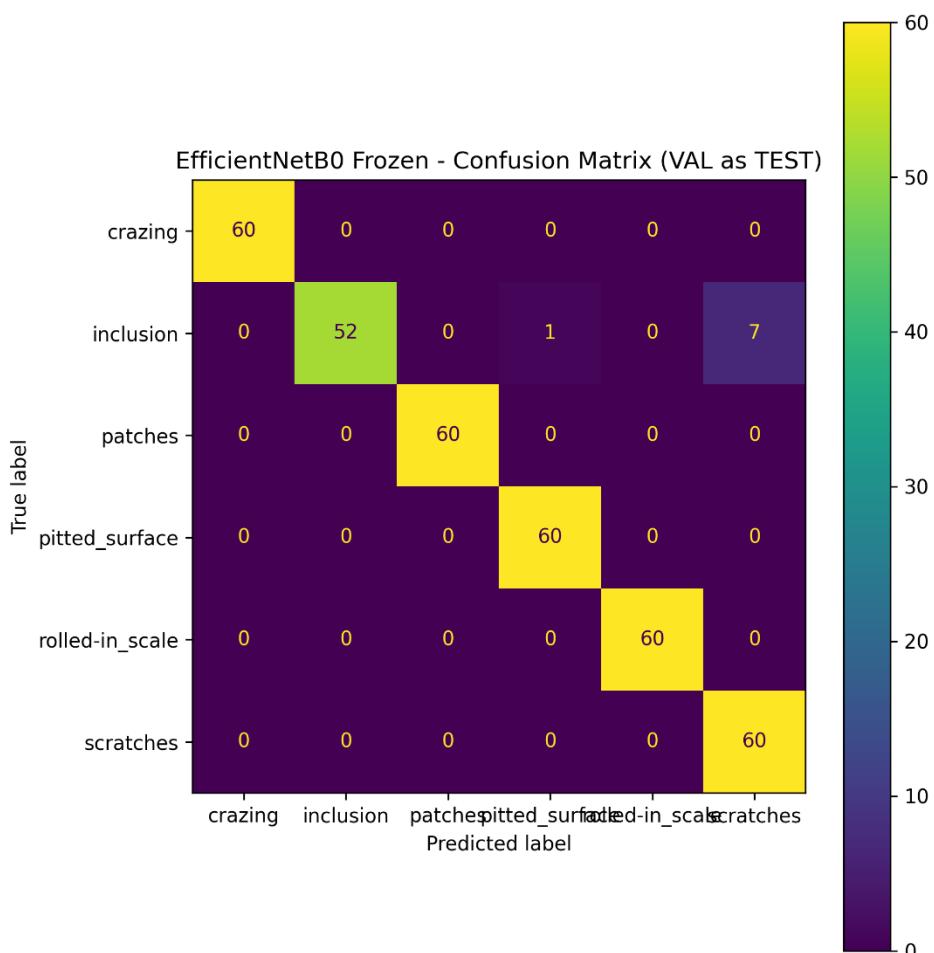


## EfficientNet-B0 (Frozen) – Mustafa

EfficientNet-B0 (Frozen) modeline ait eğitim ve doğrulama kayıp (loss) eğrileri.

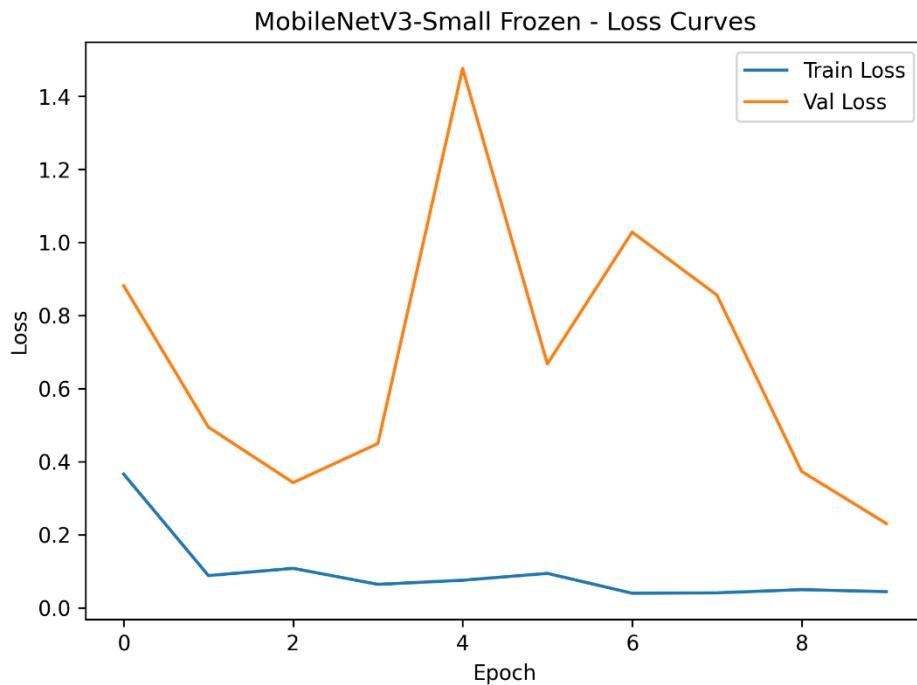


EfficientNet-B0 (Frozen) modelinin test seti üzerindeki confusion matrix sonucu.



## MobileNetV3-Small (Frozen) – Mustafa

MobileNetV3-Small (Frozen) modeline ait eğitim ve doğrulama kayıp (loss) eğrileri.



MobileNetV3-Small (Frozen) modelinin test seti üzerindeki confusion matrix sonucu.

