

DERİN ÖĞRENME – GÜZ DÖNEMİ 2025-2026

PROJE İLERLEME RAPORU

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

Proje Başlığı: Çelik Yüzey Kusuru Sınıflandırması: İki Farklı Basit CNN Modelinin 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

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.

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

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 konvolusyon 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 çalışmalardan 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.

Kaynak: NEU Surface Defect Database

Format: JPEG görüntüler

Lisans/Kullanım: Akademik araştırma amaçlı kullanım için uygundur.

Görüntü İçeriği: Metal yüzeylerde 6 farklı kusur sınıfı bulunmaktadır.

5.2 Şema (Features / Attributes)

Modelde kullanılan her görüntü aşağıdaki ön işleme adımlarından geçirilir:

Giriş: 224×224 çözünürlük

Kanal: 1 (grayscale)

Özellikler: Piksel yoğunluk değerleri ($0-255 \rightarrow$ normalize $0-1$)

Beklenen aralık: Normalized mean = 0.5, std = 0.5

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

5.3 Veri Seti Boyutu:

Toplam Görüntü Sayısı: 1800

Sınıf Sayısı: 6

Her Sınıftaki Görüntü Sayısı: 300

Sınıflar dengeli olduğu için model performansında dengesiz veri kaynaklı hata beklenmez.

Görüntü Formatı: Gri tonlu (1 kanal)

PNG/JPG formatları

Yaklaşık 200×200 piksel çözünürlük

Model eğitimi sırasında veri kümesi şu şekilde bölünmüştür:

%70 Eğitim

%15 Doğrulama %15 Test Bu oranlar, hem literatürde yaygın kullanılan dağılıma uygundur hem de sınıf dengesi korunarak uygulanmıştır.

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 görüntüleriidir.

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 Yöntemler

Bu projede çelik yüzey kusurlarını görüntülerden otomatik olarak sınıflandırmak için basit (shallow) 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, 224×224 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

6.2 Veri Ön İşleme ve Eğitim Süreci

Boyutlandırma: Tüm görüntüler 224×224 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 katmanlarından oluşur:

Conv2D (16 filtre, 3x3) + ReLU + MaxPool(2x2)

Conv2D (32 filtre, 3x3) + ReLU + MaxPool(2x2)

Conv2D (64 filtre, 3x3) + ReLU + MaxPool(2x2)

Flatten

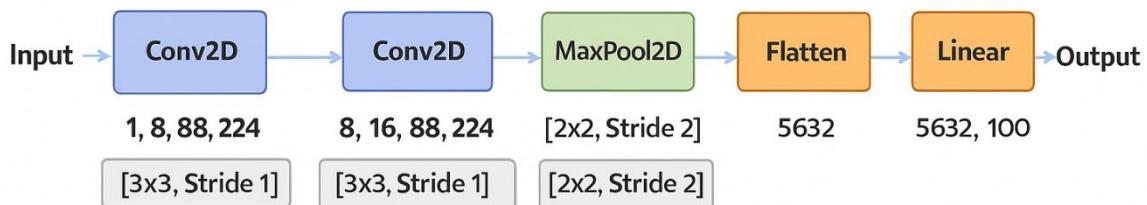
Fully Connected (128 nöron) + ReLU + Dropout(0.3)

Çıkış Katmanı: Linear(128 → 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

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 5x5 kernel kullanan, iki konvolüsyon bloğundan oluşan hafif bir CNN yapısıdır.

Mimari:

Conv2D (32 filtre, 5×5) + ReLU
MaxPool (2×2)
Conv2D (64 filtre, 5×5) + ReLU
MaxPool (2×2)
Flatten
Dense (128 nöron) + ReLU + Dropout(0.3)

Neden bu mimari?

5×5 kernel → yüzey dokusunu daha geniş alanda analiz eder.
Az katman → daha hızlı eğitim.
Dropout → 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 1: Basit CNN Modeli-Temel Performans Deneyi

Ana Amaç

Bu deneyin amacı, en basit CNN mimarisinin NEU Surface Defect veri setinde ne kadar iyi performans gösterebileceğini ölçmektedir. Model, birkaç konvolüsyon bloğu ve bir tam bağıntılı katmandan oluşan basit bir CNN mimarisiidir. Bu model, proje için bir baseline oluşturmayı hedeflemektedir.

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 birbirile karmaşma durumlarını gösterir

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

Deney 2: Geliştirilmiş CNN Modeli-Karşılaştırmalı Model Deneyi

Ana Amaç

Bu deneyin amacı, Basit CNN modeline göre biraz daha geniş özellik çıkarımı yapabilen, bir gizli tam bağlı katmanı içeren ve farklı bir augmentation stratejisi kullanan bir CNN mimarisinin performansını ölçmek ve baseline modelle karşılaştırmaktır.

Bu deney, mimari farkların performansa olan etkisini test etmek için yapılmıştır.

Train / Test Ayrımı

Aynı eğitim protokolü kullanılmıştır:

%70 Train – %15 Validation – %15 Test

Aynı seed ile rastgele bölünmüştür

Bu sayede iki model tamamen aynı veri üzerinde adil şekilde karşılaştırılır

Hiperparametre Ayarları

Epoch: 15

Learning rate: 0.0005 (gizli katman olduğu için daha düşük)

Optimizer: Adam

Loss: CrossEntropyLoss

Batch size: 32

Augmentation: flip + rotation + random crop

Değerlendirme Kriterleri

Aynı metrikler kullanılmıştır:

Accuracy

Precision (macro)

Recall (macro)

F1-score (macro)

Confusion matrix

Modeller Arası Değerlendirme ve Başarı Sıralaması

Bu projede iki farklı CNN modeli geliştirilmiştir:

SimpleCNN_Kerem (baseline)

SimpleCNN_Mustafa (gizli katmanlı, kernel 5×5)

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.

Beklenen Sonuç

Gizli katman ve daha geniş kernel kullanan Mustafa'nın modeli (SimpleCNN_Mustafa)

→ daha yüksek feature extraction kapasitesine sahip olduğu için

→ baseline modele göre biraz daha yüksek F1-score elde etmesi beklenmektedir.

8) 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

9) Kaynaklar

1-Kaggle, "NEU Surface Defect Database."

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

2-X. Song, Y. Zhang ve H. Wang, "Çelik yüzey kusurlarının derin evrişimsel sinir ağları ile sınıflandırılması," 2019 IEEE International Conference on Robotics and Automation (ICRA), 2019.

3-J. Li ve Y. He, "Sığ evrişimsel ağlar ile yüzey kusuru tespiti," Applied Surface Science, c. 503, 2020.

4-PyTorch Geliştiricileri, "PyTorch Dokümantasyonu." Erişim adresi: <https://pytorch.org>

[5] Scikit-learn Geliştirici Ekibi, "Scikit-learn Makine Öğrenimi Kütüphanesi."

Erişim adresi: <https://scikit-learn.org>

[6] Python Software Foundation, "Python 3 Resmî Dokümantasyonu."

Erişim adresi: <https://www.python.org>