

MNIST Veri Seti Kullanılarak El Yazısı Rakam Tanıma: MLP ve Random Forest Yöntemlerinin Karşılaştırmalı Analizi

Sena ALTIPARMAK¹ , Fatma ULAŞ² 

¹223205039, 4.Sınıf, Normal Öğretim

²223205023, 4.Sınıf, Normal Öğretim

Received: .././.... Accepted: .././.... Published Online: .././....

Final Version: .././....

Özet

Bu çalışmanın temel amacı, el yazısı rakamların (0-9) bilgisayar destekli sistemler tarafından yüksek doğruluk oranıyla tanınmasını sağlayan bir yapay zeka modeli geliştirmektir. Proje kapsamında, makine öğrenmesi ve görüntü işleme literatürünün en yaygın kullanılan veri setlerinden biri olan MNIST (Modified National Institute of Standards and Technology) veri seti kullanılmıştır. Modelin eğitimi ve testi için Python programlama dili ve ilgili derin öğrenme kütüphanelerinden yararlanılmıştır. Çalışmada, görüntü verilerinin ön işleme adımları (normalizasyon, boyutlandırma) gerçekleştirilmiş ve ardından oluşturulan yapay sinir ağı (YSA) mimarisi ile model eğitilmiştir. Elde edilen bulgular, geliştirilen modelin el yazısı rakamları ayırt etmede yüksek başarı oranına sahip olduğunu ve temel görüntü sınıflandırma problemlerinde etkili bir çözüm sunduğunu göstermektedir.

Anahtar Kelimeler

El Yazısı Tanıma, MNIST, Yapay Sinir Ağları, Görüntü İşleme, Makine Öğrenmesi, Python.

1. Giriş

İnsanlar için el yazısı rakamları tanımak ve ayırt etmek son derece basit bir bilişsel süreçken, bilgisayarlar için bu işlem karmaşık algoritmalar ve örüntü tanıma teknikleri gerektiren zorlu bir problemdir. El yazısı rakam tanıma, bilgisayarlı görü (computer vision) alanının "Hello World"ü olarak kabul edilir ve daha karmaşık görüntü işleme projeleri için temel oluşturur.

Bu proje, farklı kişiler tarafından yazılmış rakamların dijital ortamda anlamlandırılmasını hedefler. Geleneksel programlama yöntemleriyle her bir el yazısı varyasyonunu (eğim, kalınlık, boyut farkı) kodlamak imkânsız olduğundan, veriden öğrenen makine öğrenmesi algoritmaları bu alanda standart çözüm haline gelmiştir.

Semboller ve Kısaltmalar

MNIST	<i>Modified National Institute of Standards and Technology</i>
YSA / ANN	<i>Yapay Sinir Ağları (Artificial Neural Networks)</i>
MLP	<i>Multi-Layer Perceptron (Çok Katmanlı Algılayıcı)</i>
CNN	<i>Convolutional Neural Networks (Evrişimli Sinir Ağları)</i>
ReLU	<i>Rectified Linear Unit</i>
LIME	<i>Local Interpretable Model-agnostic Explanations</i>

1.1. Projenin Amacı

Bu çalışmanın ana hedefi, MNIST veri setindeki el yazısı rakam görsellerini kullanarak, daha önce görmediği rakamları doğru sınıflandırabilen bir model eğitmektir. Bu süreçte verinin hazırlanması, modelin kurulması, eğitilmesi ve performansının değerlendirilmesi adımları izlenmiştir.

2.Literatür Araştırması

El yazısı rakam tanıma (Handwritten Digit Recognition), bilgisayarlı görü ve örüntü tanıma alanlarının en köklü problemlerinden biridir. Bu alandaki çalışmalar, geleneksel istatistiksel yöntemlerden derin öğrenme temelli yaklaşımlara doğru büyük bir evrim geçirmiştir.

2.1. MNIST Veri Setinin Kökeni ve Önemi

Literatürdeki çalışmaların büyük çoğunluğu, LeCun ve ark. (1998) tarafından NIST veri setinden türetilen MNIST veri setini standart kıstas (benchmark) olarak kabul eder.

2.2. Geleneksel Makine Öğrenmesi Yaklaşımları

Derin öğrenme öncesi dönemde, rakam tanıma için öznetelik çıkarımı (feature extraction) ve sınıflandırma aşamaları manuel olarak gerçekleştirilmekteydi.

Support Vector Machines: Doğrusal olmayan kernel fonksiyonları ile %99.2 civarında doğruluk oranlarına ulaşılmıştır (Cortes & Vapnik, 1995).

k-Nearest Neighbors: Basit yapısına rağmen, uygun mesafe metrikleri kullanıldığında MNIST üzerinde etkili sonuçlar vermiştir.

Random Forest: Karar ağaçları topluluğu ile gürültülü verilere karşı dayanıklı modeller geliştirilmiştir.

2.3.Evrişimli Sinir Ağları (CNN) ve Derin Öğrenme

Günümüzde, CNN mimarileri şu tekniklerle geliştirilmiştir:

ReLU: Gradyan yok olması (vanishing gradient) problemini çözerek derin ağların eğitilmesini kolaylaştırmıştır.

Dropout ve Batch Normalization: Modelin aşırı öğrenmesini (overfitting) engelleyerek genelleme kapasitesini artırmıştır.

Veri Artırma (Data Augmentation): Görüntülerin döndürülmesi ve kaydırılmasıyla elde edilen sentetik veriler, modellerin hata oranını %0.2'nin altına çekmiştir.

2.4. Güncel Yöntemler: Transformer ve Dikkat Mekanizmaları

2020 sonrası literatürde, Doğal Dil İşleme alanından bilgisayarlı görüye aktarılan Vision Transformers modelleri dikkat çekmektedir. Saf CNN yapılarının aksine, pikseller arasındaki küresel ilişkileri modelleyen bu yapılar, MNIST üzerinde hata payını neredeyse sıfıra indiren "State-of-the-Art" sonuçlar üretmektedir.

3. Materyal ve Yöntem

Bu bölümde, projenin geliştirilmesinde kullanılan donanım, yazılım araçları ve veri seti detaylandırılmıştır.

3.1. Veri Seti (MNIST)

Projede kullanılan MNIST veri seti, makine öğrenmesi alanında en popüler veri setlerinden biridir. Veri seti toplamda 70.000 adet görüntüden oluşmaktadır:

Eğitim Verisi (Training Set): Modelin öğrenmesi için kullanılan 60,000 görüntü.

Test Verisi (Test Set): Modelin performansını ölçmek için kullanılan 10,000 görüntü.

Her bir görüntü 28x28 piksel boyutunda, gri tonlamalı (grayscale) ve 0 ile 9 arasındaki bir rakamı temsil eden etiketli verilerdir. Veriler model beslenmeden önce 0-255 aralığındaki piksel değerleri, işlem kolaylığı ve performans artışı için 0-1 aralığına normalize edilmiştir.

3.2. Kullanılan Teknolojiler

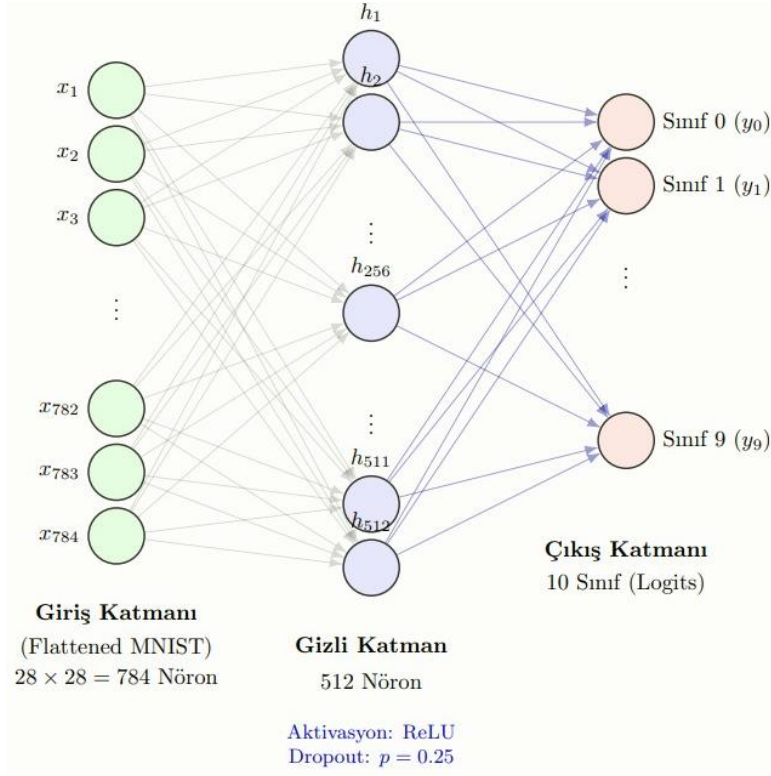
Projenin geliştirilmesinde Python programlama dili tercih edilmiştir. Veri manipülasyonu ve matematiksel işlemler için NumPy, Scikit-learn, TorchVision, modelin oluşturulması için PyTorch, sonuçların görselleştirilmesi için ise Matplotlib, Seaborn kütüphaneleri kullanılmıştır.

Tablo 1. Model Mimarisi

Katman Tipi	İşlem	Çıkış Boyutu	Parametre Sayısı
Giriş Katmanı (Input)	Flatten(28x28)	(None, 784)	0
Gizli Katman (Hidden)	Linear + ReLu	(None, 512)	401,920
Düzenleştirme (Regularization)	Dropout(p=0.25)	(None, 512)	0
Çıkış Katmanı (Output)	Linear	(None, 10)	5,130
Toplam			407,050

Tablo1'de modelin katman yapısı ve parametre sayıları verilmiştir. Model, giriş katmanında 28x28 boyutundaki görüntü matrisini vektörel hale getirerek 784 girişli bir yapı oluşturmaktadır. Gizli katmanda 512 adet nöron kullanılmış ve aktivasyon fonksiyonu olarak ReLU tercih edilmiştir. Aşırı öğrenmeyi (overfitting) engellemek amacıyla bu katmana %25 oranında Dropout uygulanmıştır.

Çıkış katmanında ise 10 farklı sınıfı temsil eden Linear katman bulunmaktadır. Modelin bu katmanda ürettiği ham değerler (logits), eğitim sırasında kullanılan CrossEntropyLoss fonksiyonu içerisinde işlenerek olasılık dağılımına dönüştürülmektedir. Bu nedenle çıkış katmanında harici bir aktivasyon fonksiyonu tanımlanmamıştır.



Şekil 1. Model Mimarisi

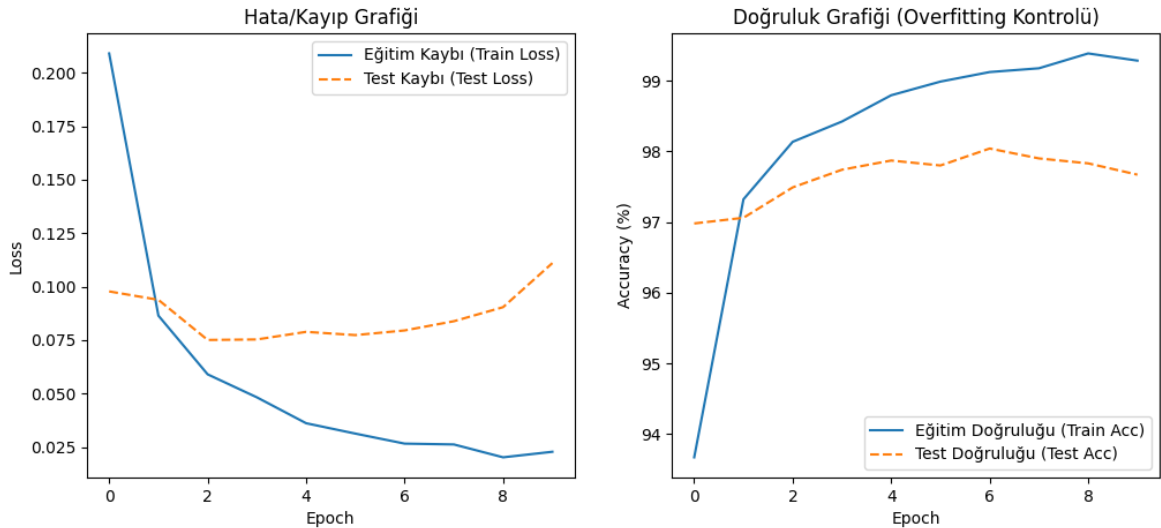
4. Deneysel Sonuçlar

Bu bölümde, modelin eğitimi sırasında gerçekleştirilen deneyler, karşılaşılan problemler (aşırı öğrenme vb.) ve bu problemlere yönelik uygulanan çözüm yöntemleri detaylandırılmıştır.

4.1. Deneysel Süreç ve Optimizasyon

Modelin en yüksek başarımlarına ulaşması ve genelleme yeteneğinin artırılması amacıyla üç aşamalı bir deneysel süreç izlenmiştir.

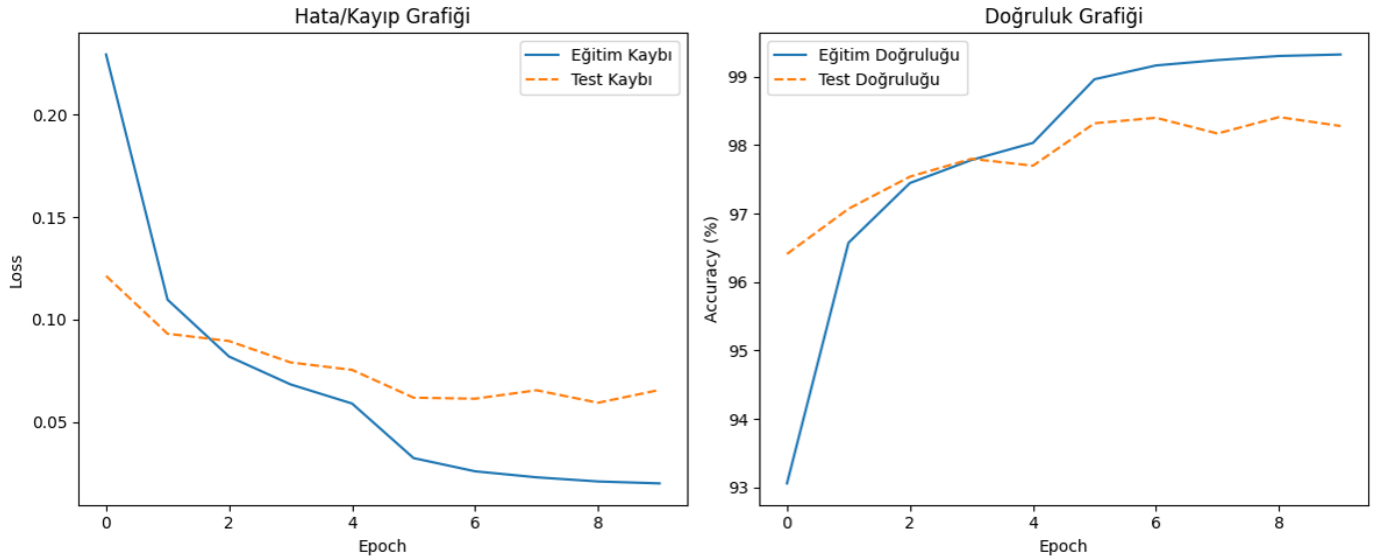
1. Aşama (Taban Model Denemesi): İlk deneylerde, model 8 epoch (eğitim döngüsü) boyunca herhangi bir düzenleme (regularization) tekniği kullanılmadan eğitilmiştir. Bu süreçte eğitim setindeki doğruluk oranı hızla artarken, test setindeki başarımın aynı oranda artmadığı ve modelin veriyi ezberlediği (overfitting) gözlemlenmiştir.



Şekil 2. Taban Model Eğitimi Grafiği

2. Aşama (Dropout Entegrasyonu): Aşırı öğrenme problemini çözmek amacıyla gizli katmana %25 oranında "Dropout" uygulanmıştır. 5 epoch süren bu deney sonucunda ezberleme sorunu ortadan kalkmış olsa da modelin yakınsama süresinin uzadığı ve doğruluk oranının %93 seviyelerinde takılı kaldığı tespit edilmiştir. Bu durum, modelin daha uzun süre eğitilmesi gerektiğini göstermiştir.

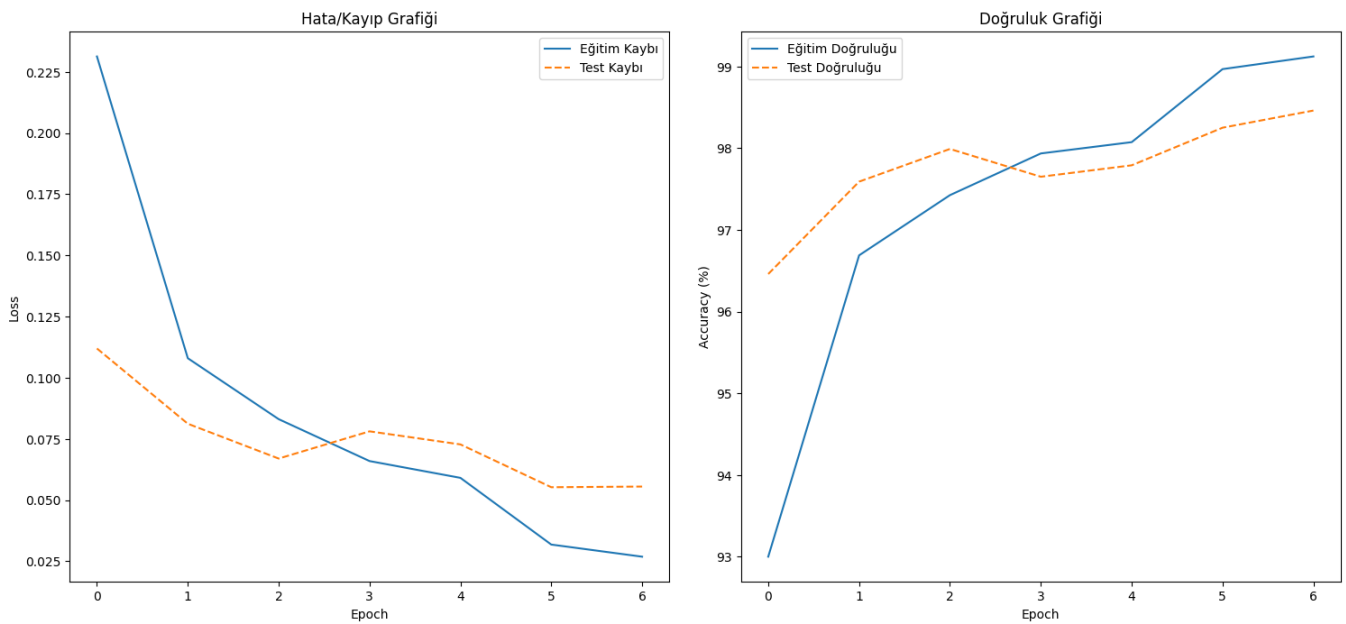
3. Aşama (LR Scheduler ve Epoch Artırımı): Öğrenme sürecini daha kararlı hale getirmek için "Learning Rate Scheduler" (Öğrenme Oranı Planlayıcı) sisteme dahil edilmiş ve eğitim süresi 10 epoch'a çıkarılmıştır. Yapılan gözlemlerde, modelin 7. epoch'a kadar istikrarlı bir şekilde öğrendiği, ancak son 3 epoch'ta test hatasının (loss) tekrar artış eğilimine girdiği, yani modelin tekrar ezberlemeye başladığı fark edilmiştir.



Şekil 3. 10 Epoch Eğitim Grafiği

4.2. Nihai Model Kararı

Elde edilen bu deneysel veriler ışığında, modelin hem ezberlemesini önlemek hem de en yüksek doğruluğa ulaşmasını sağlamak amacıyla eğitim süreci 7 epoch ile sınırlandırılmıştır. Bu konfigürasyon ile eğitilen final modeli, test veri seti üzerinde yüksek bir doğruluk oranına ulaşmış ve görmediği veriler üzerinde başarılı bir genelleme performansı sergilemiştir.



Şekil 4. Nihai Model Eğitim Grafiği

Confusion Matrix										
0	973	1	1	0	0	2	1	1	1	0
1	0	1126	2	1	0	0	1	1	4	0
2	3	1	1010	3	1	0	0	7	7	0
3	0	0	2	999	0	2	0	2	3	2
4	0	0	2	1	969	0	4	0	1	5
5	2	0	0	8	1	876	2	0	2	1
6	2	3	0	1	3	4	944	0	1	0
7	1	4	6	3	0	0	0	1008	1	5
8	2	0	2	5	3	3	0	4	953	2
9	1	2	0	4	8	2	1	3	0	988
	0	1	2	3	4	5	6	7	8	9
Tahmin Edilen										

Şekil 5. Nihai Model Confusion Matrisi

4.3. Alternatif Yöntemle Karşılaştırma: Random Forest

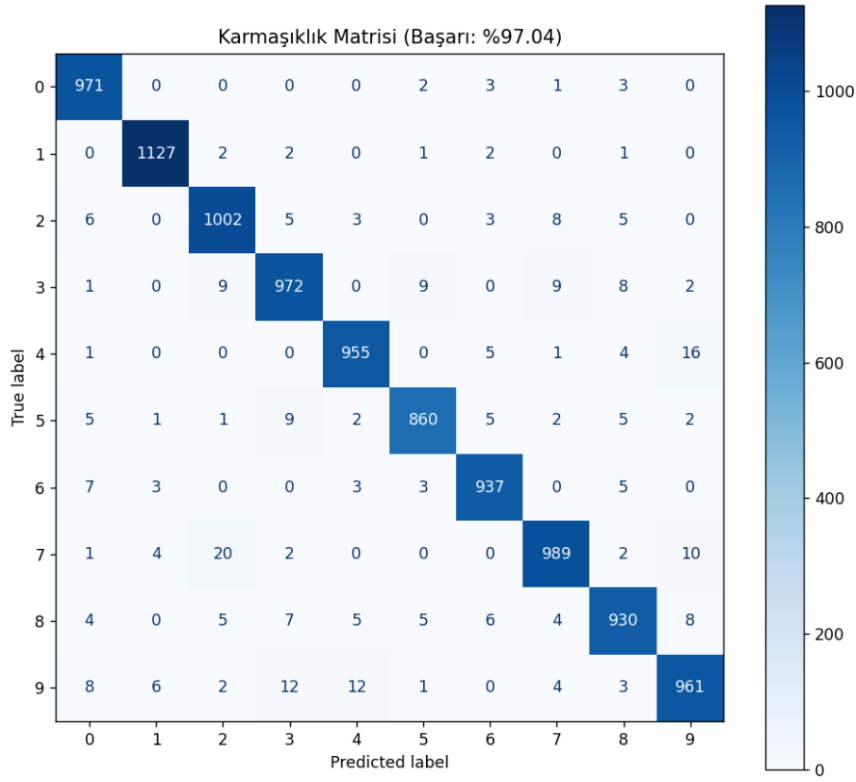
Geliştirilen MLP (Çok Katmanlı Algılayıcı) modelinin performansını objektif bir şekilde değerlendirmek amacıyla, aynı veri seti üzerinde geleneksel makine öğrenmesi algoritmalarından biri olan Random Forest algoritması ile karşılaştırmalı bir analiz yapılmıştır.

Random Forest, görüntü verilerindeki pikselleri ($28 \times 28 = 784$ öznitelik) birer sayısal giriş olarak kabul etmiş ve birden fazla karar ağacı (decision tree) oluşturarak sınıflandırma yapmıştır. Bu kıyaslamada Random Forest algoritması, derin öğrenme modellerinin aksine GPU yerine CPU üzerinde çalıştırılmış ve Scikit-learn kütüphanesi kullanılmıştır.

Tablo 2. MLP ve Random Forest Modellerinin Performans Karşılaştırması

Kriter	MLP	Random Forest
Mimari Yapısı	Yapay Sinir Ağı (YSA)	Karar Ağaçları Topluluğu
Eğitim Süresi	Orta (GPU Destekli)	Düşük/Hızlı (CPU)
Parametre/Ayar	7 Epoch, Dropout (0.25)	n_estimators (Ağaç Sayısı): 100
Test Doğruluk Oranı	%98.46	%97.04

Yapılan testler sonucunda, Random Forest algoritmasının kurulumunun daha basit ve eğitim süresinin daha kısa olmasına rağmen; ham piksel verileri arasındaki karmaşık örüntüleri ve mekansal ilişkileri öğrenmede MLP modelinin daha başarılı olduğu görülmüştür. Özellikle birbirine benzeyen rakamların ayırt edilmesinde, MLP mimarisindeki gizli katmanın ve aktivasyon fonksiyonlarının sağladığı esneklik, doğruluk oranını artırıcı bir faktör olmuştur. Sonuç olarak, el yazısı tanıma gibi algısal görevlerde Derin Öğrenme tabanlı yaklaşımların geleneksel yöntemlere kıyasla daha yüksek performans sunduğu deneysel olarak doğrulanmıştır.



Şekil 6. Random Forest Confusion Matrisi

4.4. Random Forest Açıklanabilirliği: LIME Analizi

Çalışmanın karşılaştırma aşamasında kullanılan Random Forest algoritmasının, ham pikselleri nasıl değerlendirdiğini görselleştirmek ve veri setindeki özneliliklerin (piksellerin) model üzerindeki etkisini doğrulamak amacıyla, bu modele özgü olarak LIME analizi uygulanmıştır.

Elde edilen LIME çıktıları şunları göstermiştir:

Pozitif Katkı Sağlayan Pikseller: Model, tahmin ettiği rakamın yapısal hatlarını oluşturan pikselleri "doğru sınıfı destekleyen kanıt" olarak değerlendirmiştir.

Negatif/Nötr Pikseller: Rakamın dışında kalan veya başka bir rakamı andıran bölgeler ise tahmin olasılığını düşüren veya etkilemeyen alanlar olarak belirlenmiştir.

LIME analizi, Random Forest modelinin karar verirken rastgele gürültülere değil, rakamın bulunduğu anlamlı bölgelere odaklandığını doğrulamıştır. Ancak, modelin bu pikselleri mekansal bir bütün (bir eğri, çizgi veya döngü) olarak yorumlamaktan ziyade; hangi koordinattaki pikselin hangi sayısal değere sahip olduğuna odaklanan ayrık bir algı yapısına sahip olduğu tespit edilmiştir.

5. Tartışma

Eğitim süreci boyunca modelin performans gelişimi "Loss" (Kayıp) ve "Accuracy" (Doğruluk) metrikleri üzerinden takip edilmiştir.

Model, belirlenen epoch (döngü) sayısı boyunca eğitildikten sonra test veri seti üzerinde değerlendirilmiştir. Eğitim setinde görülen doğruluk oranı ile test setindeki doğruluk oranı birbirine yakın çıkmıştır, bu da modelin ezberleme (overfitting) yapmadığını ve genelleme yeteneğinin yüksek olduğunu göstermektedir.

Elde edilen sonuçlara göre model, el yazısı rakamları %98'in üzerinde bir doğrulukla tanıyabilmektedir. Karmaşıklık matrisi (Confusion Matrix) incelendiğinde, modelin en çok birbirine benzeyen rakamları (örneğin 4 ve 9 veya 2 ve 7) karıştırma eğiliminde olduğu gözlemlenmiştir.

6.Sonuç

Bu projede, el yazısı rakam tanıma problemi ele alınmış ve makine öğrenmesi yöntemleriyle başarılı bir sınıflandırma modeli geliştirilmiştir. Çalışma, ham piksellerden anlamlı bilgiler çıkarılabileceğini ve yapay zekanın görüntü işleme alanındaki etkinliğini kanıtlar niteliktedir. Gelecek çalışmalarda, daha karmaşık mimariler (örneğin CNN) kullanılarak doğruluk oranı artırılabilir veya model gerçek zamanlı kamera görüntülerinden rakam tanıyacak hale getirilebilir.

Kaynaklar

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

LeCun, Y., Cortes, C., & Burges, C. J. (2010). MNIST handwritten digit database. <http://yann.lecun.com/exdb/mnist/>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. The Journal of Machine Learning Research, 12, 2825-2830.

Python Software Foundation. (2025). Python Language Reference, version 3.x. <https://www.python.org>

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135-1144).

Savaş, S., & Buyrukoğlu, S. (Eds.). (2022). Teori ve Uygulamada Makine Öğrenmesi. Nobel Akademik Yayıncılık.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958.

Ekler:

Proje Sunumu: https://prezi.com/view/bUhgbTYD5A4RCE1IA8Wk/?referral_token=WyvE-9lnB3FN

Proje Kodları: https://drive.google.com/drive/folders/1iwXwpTAbspFjMJKugi5Qgk61e_0HskEJ?usp=drive_link