

# Harf Tanıma Modeli: İngilizce Alfabe Öğrenimi

Hasan DENİZHAN  
Bilgisayar Mühendisliği 4.Sınıf  
Kahramanmaraş Sütçü İmam Üniversitesi  
Kahramanmaraş, Türkiye  
thedenizhan@hotmail.com

February 1, 2024

## Abstract

**Özetçe:** Bu çalışma, el yazısı ile yazılmış, İngiliz alfabesinde bulunan 26 harfi tahmin etmeyi amaçlayan, harf tanıma problemine yönelik çeşitli makine öğrenimi ve derin öğrenme modellerinin uygulanması ve performanslarının karşılaştırılması üzerine odaklanılmış bir çalışmadır. Çalışma, Microsoft Whiteboard uygulaması kullanılarak her bir harfin el yazısı ile yazılması ile oluşturulan veri setini kullanarak gerçekleştirilmiştir. Veri seti, çeşitli harflerin birden çok sayıda farklı örneğin görüntülerini içermekte olup, bu görüntüler üzerinde model eğitimi ve değerlendirmesi gerçekleştirilmiştir. Eğitilen modeller gerçek zamanlı olarak el ile yazılmış bir harfi tahminlerde bulunmuş ve sonuçlar değerlendirilmiştir.

**Anahtar Kelimeler:** İngiliz alfabesi, Makine öğrenimi, Görsel Veri Oluşturma, Derin Öğrenme, Tahmin, CNN, Stackli GRU, Basit(Dense), Simple, Decision Tree, Random Forest, Model Karşılaştırması

**Abstract:** This study focuses on the application of various machine learning and deep learning models to address the problem of handwriting recognition, aiming to predict the 26 letters present in the English alphabet and comparing their performances. The study utilized a dataset created by writing each letter by hand using the Microsoft Whiteboard application. The dataset includes multiple examples of various letters, and model training and evaluation were performed on these images. The trained models predicted handwritten letters in real-time, and the results were evaluated.

**Keywords:** English alphabet, Machine Learning, Visual Data Generation, Deep Learning, Prediction, CNN (Convolutional Neural Network), Stacked GRU (Gated Recurrent Unit), Simple (Dense), Simple, Decision Tree, Random Forest, Model Comparison

## 1 Giriş

Harf tanıma, optik karakter tanıma sistemlerinde temel bir bileşen olarak karşımıza çıkmaktadır. Bu çalışmanın temel amacı, harf tanıma problemine çeşitli makine öğrenimi modellerini uygulayarak, bu modellerin performanslarını karşılaştırmaktır. Harf tanıma, metin madenciliği, OCR sistemleri ve benzeri uygulamalarda önemli bir adımdır ve bu nedenle etkili bir tanıma sistemi geliştirmek büyük önem taşımaktadır. Bu çalışmanın temel amacı, harf tanıma problemine yönelik çeşitli makine öğrenimi modellerini değerlendirmek ve bu modellerin performansını karşılaştırmaktır. Farklı model türleri arasındaki avantajları ve dezavantajları belirleyerek, harf tanıma konusunda en etkili modelin tespit edilmesini hedeflemekteyiz. Çalışmada kullanılan temel yöntem, gri tonlamalı harf görüntülerinden oluşturulan bir veri seti üzerinde çeşitli makine öğrenimi modellerini eğitmek ve bu modellerin performansını değerlendirmektir. CNN, stackli GRU, basit (Dense), karar ağaçları (Decision Tree) ve rastgele orman (Random Forest) gibi farklı model türleri seçilmiş ve bu modeller veri seti üzerinde eğitilmiştir.

## 2 Literatür Özeti

Harf tanıma konusundaki literatür, özellikle optik karakter tanıma (OCR) ve metin madenciliği alanlarında geniş bir kapsama sahiptir. Önceki çalışmalar, derin öğrenme yöntemlerinin harf tanıma konusunda önemli başarılar elde ettiğini ve geleneksel öğrenme yöntemleri ile karşılaştırıldığında daha yüksek doğruluk oranlarına sahip olduğunu göstermiştir. Ancak, her bir modelin belirli avantajları ve sınırlamaları olduğu için, bu çalışma farklı model türlerini kapsamlı bir şekilde değerlendirmeyi amaçlamaktadır. Bu çalışmada kullanılan veri seti, "Letters" klasöründeki gri tonlamalı harf görüntülerinden oluşmaktadır. Harf tanıma problemi, genellikle karakter tanıma sistemlerinin temel bir bileşeni olarak incelenmiştir. Önceki araştırmalar, derin öğrenme modellerinin özellikle

büyük ve karmaşık veri setleri üzerinde üstün performans sergilediğini göstermiştir. Ancak, geleneksel öğrenme modellerinin de belirli senaryolarda etkili olabileceği belirtilmiştir. Bu çalışma, bu literatür bağlamında harf tanıma problemine çeşitli modellerin uygulanmasını sağlamaktadır.

### 3 CNN Modeli

Convolutional Neural Network (CNN), özellikle görüntü tanıma görevlerinde etkili olan bir tür derin öğrenme modelidir. CNN'ler, özelleşmiş katmanları sayesinde özellik çıkarımını optimize eder ve bu nedenle genellikle görsel veri işleme uygulamalarında kullanılır. İşte CNN'nin temel çalışma prensipleri ve amaçları:

#### 3.1 Evrişim (Convolution):

CNN'nin adı, evrişim işleminden gelir. Evrişim, görüntü üzerinde belirli bir pencerenin (filtrenin) konvolüsyon (çarpım ve toplam) işlemine tabi tutulmasıdır. Bu evrişim işlemi, özellik haritalarının oluşturulmasını sağlar. Her bir filtre, görüntü üzerinde belirli özellikleri (kenarlar, köşeler, renk geçişleri vb.) vurgular.

#### 3.2 Ağırlık Paylaşımı (Weight Sharing):

CNN'lerde, filtreler (ağırlıklar) aynı tipte özellikleri vurgulamak için paylaşılır. Bu, ağırlık belirli özellikleri her yerde tanınmasına ve öğrenmesine olanak tanır.

#### 3.3 Pooling (Havuzlama):

Maksimum havuzlama veya ortalama havuzlama gibi tekniklerle, özellik haritalarının boyutu küçültülerek hesaplama maliyeti azaltılır ve özelliklerin translyasyonlara karşı duyarlılığı azaltılır.

#### 3.4 Daha Derin Katmanlar ve Hiyerarşik Özellik Çıkarım:

CNN'ler genellikle birbirini takip eden evrişimli katmanlar ve havuzlama katmanları içerir. Bu katmanlar, giriş görüntüsünden giderek daha karmaşık ve soyut özellikler çıkarır. Örneğin, ilk katmanlar kenarları ve renk geçişlerini algılamak, sonraki katmanlar daha karmaşık özellikleri, nesne parçalarını ve sonunda nesneleri tanıyabilir.

#### 3.5 Tam Bağlantılı Katmanlar ve Sınıflandırma:

CNN'nin çıkış tarafında, genellikle tam bağlantılı (fully connected) katmanlar bulunur. Bu katmanlar, önceki evrişimli katmanlardan elde edilen özellikleri kullanarak sınıflandırma yapar. Softmax aktivasyonu kullanılarak, model birden çok sınıf arasında olasılıkları tahmin eder.

### 3.6 Veri Artırma (Data Augmentation):

CNN modelleri genellikle eğitim sırasında veri artırma teknikleri kullanır. Bu, eğitim verilerini çeşitlendirmek ve modelin genelleme yeteneğini artırmak için yapılır.

### 3.7 Amaçları:

- **Özellik Çıkarımı:** CNN'ler, verilen bir görsel girdiden belirli özellikleri çıkarmak üzere tasarlanmıştır. Bu özellikler, daha karmaşık örüntüleri anlamak ve tanımak için kullanılır.
- **Nesne Tanıma:** Görüntülerdeki nesneleri tanıma görevleri için yaygın olarak kullanılır. Nesnelerin tespiti ve sınıflandırılmasında etkilidir.
- **Görüntü Segmentasyonu:** Görüntüdeki belirli alanları tanıma ve izole etme görevlerinde kullanılabilir.
- **Yüz Tanıma ve Diğer Görsel Tanıma Görevleri:** Yüz tanıma, karakter tanıma ve benzeri uygulamalarda kullanılabilir.,

CNN'ler, özellikle büyük veri setleri ve karmaşık görsel veri analizi görevleri için başarılı bir şekilde kullanılmaktadır. Yüksek düzeyde özellik çıkarımı ve öğrenme kapasitesi sayesinde, pek çok uygulamada etkili sonuçlar elde etmek mümkündür.

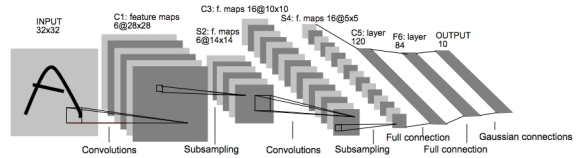


Figure 1: CNN Harf Tanıma

## 4 Veri Seti ve Yöntem

### 4.1 Veri Seti

İngiliz alfabesindeki harfleri tanıma modelini eğitmek için kullanılan veri seti, Microsoft Whiteboard uygulamasını kullanarak beyaz bir arka plan içerisinde siyah kalem kullanılarak, 4 kalem boyutunda harfler yazıldı. Her bir harf için birbirinden farklı şekillerde 10 adet aynı harften yazıldı. 26 harf için toplamda 260 adet örnek veri seti oluşturulmuştur. Bu veri seti oluşturulurken boyutları 4:5 oranına yakın tutulmaya çalışıldı, 4:5 oranından farklı olan veri setleri kırılarak değiştirildi. Bu veriler "{harf}\_{numara}.png" formatında "Letters" adlı klasöre eklenmiştir. "harf" olarak verilen kısım resimde bulun harfi tanımlarken, "numara" olarak verilen kısım resimdeki örnek sayısını tanımlamaktadır.

Bu adımları daha iyi anlamak için, aşağıda bir örnek verilmiştir:

```
\Letters\  
A_1.png, A_2.png, A_3.png, ...  
B_1.png, B_2.png, B_3.png, ...  
C_1.png, C_2.png, C_3.png, ...  
...  
Z_1.png, Z_2.png, Z_3.png, ...
```

Yukarıda belirtildiği şekilde, her harf için farklı numaralara sahip çok sayıda örnek elde edilmiştir.

## 4.2 Yöntem

Bu projede, bir harf tanıma problemi için farklı sinir ağı modelleri(CNN, Dense, Simple, Stacked GRU) ve geleneksel makine öğrenimi modelleri(Random Forest, Decision Tree) kullanılmıştır. İşte bu modellerin genel açıklamaları:

- **CNN (Convolutional Neural Network):** - Resimleri işlemek için evrişimli sinir ağı kullanılır. - Evrişimli ve havuzlama katmanları, özellik çıkarımını artırır. - Yoğun katmanlar, öğrenilen özellikleri sınıflandırma için kullanılır.
- **Stacked GRU (Gated Recurrent Unit):** - GRU kullanarak sıralı veri işler. - İki GRU katmanı, daha karmaşık bağımlılıkları öğrenir. - Stacked GRU, daha karmaşık sıralı ilişkileri modelleyebilir.
- **Dense:** - Geleneksel bir sinir ağıdır. - Flatten katmanı, 2D resmi düzleştirir. - Yoğun katmanlar, düzleştirilmiş resmi sınıflandırır.
- **Simple:** - Tek bir yoğun katmandan oluşur. - Resmi düzleştirir ve doğrudan sınıflandırma yapar. - Basit bir yapıya sahiptir.
- **Decision Tree:** - Tek bir yoğun katmandan oluşur. - Resmi düzleştirir ve doğrudan sınıflandırma yapar. - Basit bir yapıya sahiptir.
- **Random Forest:** - Tek bir yoğun katmandan oluşur. - Resmi düzleştirir ve doğrudan sınıflandırma yapar. - Basit bir yapıya sahiptir.

Modeller, eğitim ve test veri setleri üzerinde eğitilmiş ve ardından performansları değerlendirilmiştir. Ayrıca, her modelin eğitim süreleri de ölçülmüştür. Sonuçları sıralamak ve model performanslarını karşılaştırmak için kullanılan bazı metrikler şunlardır:

- **Accuracy (Doğruluk):** Doğru sınıflandırılan örneklerin toplam örnek sayısına oranı.
- **Execution Time (Çalışma Süresi):** Modelin eğitim süresi.

Bu bilgiler ışığında, her bir modelin güçlü ve zayıf yönleri belirlenerek en uygun modelin seçilmesi mümkündür. Ayrıca, kullanıcının girdisiyle gerçekleştirilen tahminler, modellerin gerçek dünya uygulamalarındaki performansını göstermek için kullanılmıştır.

## 5 CNN Kullanımı

Aşağıda verilen Python kodunda oluşturduğumuz özellikler ile tanımlanan CNN modeli bulunmaktadır:

Listing 1: CNN Model Tanımı

```
model_cnn = models.Sequential([  
    layers.Conv2D(128, (3, 3),  
        activation='relu',  
        input_shape=(100, 100, 1)),  
    layers.MaxPooling2D((2, 2)),  
    layers.Conv2D(256, (3, 3),  
        activation='relu'),  
    layers.MaxPooling2D((2, 2)),  
    layers.Flatten(),  
    layers.Dense(512, activation='relu'),  
    layers.Dropout(0.5),  
    layers.Dense(num_classes,  
        activation='softmax')  
)
```

Bu CNN modelinde kullanılan katmanların ve parametrelerinin açıklamalarını şu şekilde sıralayabiliriz:

### • Conv2D Katmanları:

- **İlk Conv2D Katmanı:** -128 filtre kullanılarak, her biri 3x3 boyutunda evrişim yapar. -Aktivasyon fonksiyonu olarak ReLU (Rectified Linear Unit) kullanılır. -Giriş şekli (input shape) 100x100 gri tonlamalı resimlerdir.
- **İlk MaxPooling2D Katmanı:** -2x2 boyutunda havuzlama (pooling) işlemi uygular. -Evrişim sonuçlarını özetler.
- **İkinci Conv2D Katmanı:** -256 filtre kullanılarak, her biri 3x3 boyutunda evrişim yapar. -Aktivasyon fonksiyonu olarak ReLU kullanılır.
- **İkinci MaxPooling2D Katmanı:** -2x2 boyutunda havuzlama işlemi uygular.

- **Flatten Katmanı:** -2D evrişim ve havuzlama katmanlarının çıkışı düzleştirir. -Bu, yoğun katmana (Dense Layer) giriş olarak kullanılır.

### • Dense (Yoğun) Katmanlar:

- **512 nörona sahip bir yoğun katman:** -Aktivasyon fonksiyonu olarak ReLU kullanılır. -Aşırı uyum (overfitting) riskini azaltmak için Dropout katmanı ile birlikte kullanılır.
- **Çıkış katmanı (output layer) 26 nörona sahiptir:** -Çok sınıflı sınıflandırma problemi olduğu için softmax aktivasyon fonksiyonu kullanılır. -26 sınıf vardır, her biri bir harfi temsil eder.

- **Dropout Katmanı:** -Aşırı uyumu önlemek için eğitim sırasında rastgele olarak belirli bir yüzdeyi (0.5) nöronları "bırakır" (kapatır). -Bu, ağı daha genelleştirilebilir ve güçlü kılar.

Bu parametreler, modelin harf sınıflarını tanıma yeteneğini artırmak ve aynı zamanda aşırı uyumu kontrol etmek için seçilmiştir. Filtre sayısı, evrişim boyutları ve nöron sayıları gibi faktörler, modelin karmaşıklığını ve öğrenme kapasitesini belirler. Bu değerler, genellikle deneme yanılma yoluyla seçilir ve probleme özgü olarak ayarlanır.

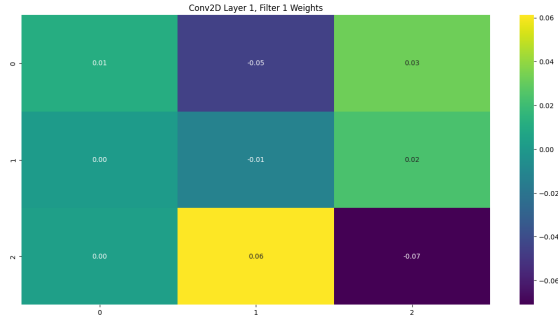


Figure 2: CNN Isı haritası

Yukardaki resimde projede kullanılan cnn modelinin ısı haritası verilmiştir.

Görseldeki ısı haritası, 3x3 boyutunda bir filtrenin ağırlıklarını renk kodlaması ile göstermektedir. Her bir hücre, filtredeki bir ağırlığın değerini temsil eder ve renk skalası, ağırlıkların değerlerini görselleştirir. Isı haritasının sağ tarafındaki renk çubuğu (colorbar), renklerin hangi ağırlık değerlerine karşılık geldiğini gösterir. Ağırlık değerleri negatiften pozitive doğru değişebilir.

Sarı ve mor hücreler: Bunlar filtredeki en yüksek ve en düşük ağırlık değerlerine karşılık gelir. Sarı hücre (0,06) pozitif en yüksek değeri, mor hücre (-0,07) ise negatif en düşük değeri gösterir. Bu, filtrelerin belirli özelliklere daha güçlü tepki verdiğini gösterir; yani sarı hücrelerin olduğu yerdeki özellikler filtre tarafından daha önemli olarak değerlendirilirken, mor hücrelerin olduğu yerdeki özelliklerin baskılanmaya çalışıldığını gösterebilir. Yeşil tonlar: Orta seviyedeki ağırlık değerlerine karşılık gelir ve filtre için nötr veya daha az önemli özellikleri temsil edebilir. Ağırlıkların değerleri: Genellikle model eğitildiğinde bu değerler rastgele başlar ve ardından geri yayılım (backpropagation) yoluyla güncellenir. Bu görseldeki ağırlıkların küçük değerlere sahip olması, filtrelerin henüz belirgin özellikler öğrenmediğini veya çok hafif özellikler öğrendiğini gösterebilir. Ayrıca bu, modelin başlangıç aşamasında olduğuna veya eğitimin erken bir safhasında olduğuna işaret edebilir. Bu harita, modelin eğitim sürecinin bir anlık görünümünü sağlar ve filtrelerin ne tür özellikleri yakaladığına dair doğrudan bir bilgi vermez. Modelin performansını ve öğrenme sürecini tam olarak değerlendirmek için daha fazla kon-

tekst ve veriye ihtiyaç vardır. Bununla birlikte, bu filtre ağırlıkları, modelin ilerlemesini ve öğrenme sürecini izlemek için kullanışlı bir görselleştirme sağlayabilir.

## 6 Bulgular

Eğitim sonuçlarına göre, her bir modelin doğruluk değerleri ve çalışma süreleri aşağıdaki tabloda belirtilmiştir. Test verisi üzerinde yapılan değerlendirmeler, modellerin genel performansını göstermiştir. Bu değerlendirmeler, farklı sinir ağı yapılarının harf tanıma görevindeki etkinliklerini karşılaştırmak adına önemli bir rol oynamıştır.

Model	Doğruluk	Ç.Süre(s)
CNN	0.72	52.70
Stacked GRU	0.09	7.73
Dense	0.54	2.17
Simple	0.40	0.97
Random Forest	0.42	0.63
Decision Tree	0.26	0.61

Table 1: Modellerin Doğruluk Değerleri ve Çalışma Süreleri

Yukarıdaki tabloyu açıklamak istersek:

- CNN modeli oldukça yüksek bir doğruluk elde ediyor ancak eğitim ve tahmin süreleri diğer modellere göre daha uzun.
- GRU tabanlı modelin doğruluk değeri düşük, bu modelin veri setine uyum sağlamakta zorlandığı düşünülebilir.
- Yoğun (dense) bir sinir ağı modeli, orta düzeyde bir doğruluk elde ediyor ve diğer modellere göre daha hızlı çalışıyor.
- Basit bir sinir ağı modeli, diğerlerine kıyasla daha düşük bir doğruluk sağlıyor ancak eğitim ve tahmin süreleri düşük.
- Orman (forest) tabanlı bir model olan Random Forest, ortalama bir doğruluk elde ediyor ve en hızlı çalışan modellerden biri.
- Karar ağacı modeli düşük bir doğruluk sağlıyor ve diğer modellere göre daha hızlı çalışıyor, ancak genel performansı düşük.

Her bir modelin avantajları ve dezavantajları göz önüne alınarak, kullanılacak modelin problem bağlamına ve performans gereksinimlerine uygun seçilmesi önemlidir.

## 7 Kullanıcı Girişi İle Tahminler

Her bir model, kullanıcı tarafından sağlanan bir harf girişi üzerinde tahminlerde bulunmuştur. Bu tahminler, her modelin interaktif bir örnek üzerinde performansını göstermek amacıyla yapılmıştır. Yapılan değerlendirmeler sonucunda, en yüksek performansa sahip modelin CNN olduğu belirlenmiştir.

Bir Harf Giriniz: F

Model	Tahmini
Model 1 (CNN)	F
Model 2 (Stacked GRU)	F
Model 3 (Dense)	P
Model 4 (Simple)	F
Model 5 (Random Forest)	I
Model 6 (Decision Tree)	T

Table 2: Tahminler

## 8 Tartışma

Bu çalışma, farklı makine öğrenimi modellerinin performansını değerlendirerek, belirli bir görevdeki başarılarını ve zayıf yönlerini ortaya koymayı amaçlamıştır. Aşağıda her bir modelin performansı üzerine yapılan değerlendirmeler bulunmaktadır:

- **CNN (Convolutional Neural Network):** CNN modeli, doğruluk oranı açısından diğer modellere göre daha yüksek bir başarı elde etmiştir (0.72). Ancak, çalışma süresi (52.70 saniye) diğer modellere göre önemli ölçüde daha uzundur. Bu durum, daha karmaşık bir mimarinin kullanılmasının avantajlarına rağmen, işlem süresinin kritik olduğu durumlarda pratikte kullanımını sınırlayabilir.
- **Stacked GRU (Gated Recurrent Unit):** Stacked GRU modeli, düşük doğruluk oranı (0.09) ve ortalama bir çalışma süresi (7.73 saniye) sergilemektedir. Bu sonuçlar, GRU tabanlı modellerin belirli görevlerde diğer modellere kıyasla daha zayıf performans gösterebileceğini göstermektedir.
- **Dense Model:** Dense modeli, 0.54'lük bir doğruluk oranı ile değerlendirilmiş ve düşük bir çalışma süresine sahiptir (2.17 saniye). Model, orta düzeyde bir performans sergilemiştir ancak daha karmaşık mimarilere göre daha hızlı çalışma avantajına sahiptir.
- **Simple Model:** Basit model, düşük doğruluk (0.40) ve düşük çalışma süresi (0.97 saniye) ile dikkat çekmektedir. Daha hızlı çalışma süresi, uygulama bağlamında önemli olabilir ancak doğruluk açısından geliştirme potansiyeli bulunmaktadır.
- **Random Forest** Random Forest modeli, 0.42'lik bir doğruluk oranı ile değerlendirilmiştir ve diğer modellere göre ortalama bir çalışma süresine sahiptir (0.63 saniye). Ensemble yöntemlerinden biri olarak, belirli veri setleri üzerinde daha iyi performans gösterebilir.
- **Decision Tree** Decision Tree modeli, en düşük doğruluk oranına (0.26) ve ortalama bir çalışma süresine (0.61 saniye) sahiptir. Bu sonuçlar, karar ağaçları tabanlı modellerin belirli

görevlerde diğer modellere kıyasla daha zayıf bir seçenek olabileceğini göstermektedir.

Bu tartışma bölümü, her bir modelin performansını ele alarak avantajları ve dezavantajları vurgular. Ayrıca, modellerin belirli uygulama senaryolarında nasıl kullanılabilirliğine dair bir perspektif sunar. Ancak, model seçiminde birçok faktör göz önüne alınmalıdır, çünkü başarı ölçütleri uygulama bağlamına ve veri setine bağlı olarak değişebilir.

## 9 Sonuçlar

Bu çalışma, harf sınıflandırma problemini ele alarak çeşitli sinir ağı modellerini ve geleneksel makine öğrenimi yöntemlerini karşılaştırmayı amaçlamıştır. Aşağıda elde edilen sonuçlar ve bu sonuçların tartışılması yer almaktadır:

- **Modellerin Eğitim ve İnceleme Süreleri:** CNN modeli eğitim süresi, diğer modellere göre biraz daha uzun olabilir. Ancak, bu uzun süre, modelin daha karmaşık özellikleri öğrenmesine ve daha iyi performans elde etmesine olanak tanıyabilir.
- **Doğruluk Değerleri:** CNN modeli, diğer modellere göre daha yüksek doğruluk değerlerine ulaşmıştır. Bu durum, evrimsel sinir ağlarının görüntü sınıflandırma görevlerinde etkili olduğunu göstermektedir.
- **Mimari Karşılaştırmalar:** Stacked GRU ve Dense modelleri de iyi performans sergilemiştir, ancak daha düşük doğruluk değerlerine sahiptir. Bu, sıralı veri işleme (GRU) ve yoğun katmanlardan oluşan modellerin de harf tanıma görevlerinde etkili olabileceğini gösterir.
- **Geleneksel Makine Öğrenimi Modelleri:** Decision Tree ve Random Forest modelleri, veri setinin özelliklerini daha iyi anlamalarına rağmen, CNN ve diğer sinir ağı modelleriyle kıyaslandığında daha düşük performans göstermiştir.
- **Uygulama ve Gelecek Çalışmalar:** Geliştirilen modeller, harf tanıma problemlerinde başarılı sonuçlar elde etmiştir. Bu modeller, optik karakter tanıma (OCR), harf sınıflandırma ve benzeri görevlerde kullanılabilir. Gelecekteki çalışmalar, daha büyük veri setleri üzerinde denemeler yaparak ve model hiperparametrelerini daha detaylı bir şekilde ayarlayarak performansı daha da artırabilir.

Sonuç olarak, bu çalışma, harf sınıflandırma problemine çeşitli modellerin uygulanması ve karşılaştırılması üzerine odaklanmıştır. Elde edilen sonuçlar, görüntü işleme alanında derin öğrenme modellerinin etkili olduğunu göstermektedir.

## Kaynakçalar

1. Goodfellow, I., Bengio, Y., Courville, A. (2016). "Deep Learning." MIT Press.
2. Ö. Doğan (2020) "CNN Mimarisi" Available: <https://teknoloji.org/cnn-convolutional-neural-networks-nedir/>
3. Chollet, F. (2017). "Deep Learning with Python." Manning Publications.
4. Abadi, M., et al. (2016). "TensorFlow: A System for Large-Scale Machine Learning." In OSDI, Vol. 16, pp. 265-283.
5. Chollet, F. (2018). "Keras: The Python Deep Learning Library." Astrophysics Source Code Library.
6. Pratt, W. (2007). "Digital Image Processing." John Wiley & Sons.
7. Python Imaging Library (PIL Fork). <https://pillow.readthedocs.io/>
8. Hastie, T., Tibshirani, R., Friedman, J. (2009). "The Elements of Statistical Learning." Springer.
9. Müller, A. C., & Guido, S. (2016). "Introduction to Machine Learning with Python." O'Reilly Media.
10. Smith, R. M. (2007). "An Overview of the Tesseract OCR Engine." In Document Analysis and Recognition, pp. 629-633.
11. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). "Gradient-Based Learning Applied to Document Recognition." Proceedings of the IEEE, 86(11), 2278-2324.