



**MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**

**Video Verilerinden İnsan Aktivitesi Tanımaya Yönelik Yapay Zeka  
Uygulamaları**

**Emir Can Tanyıldız**

**2204040177**

**İstanbul, 2024**

## İÇİNDEKİLER

|   |     |
|---|-----|
| İçindekiler.....                                      | ii  |
| Özet.....   | iii |
| ABSTRACT.....   | iv  |
| 1. GİRİŞ VE AMAÇ .....                                | 1   |
| 2. Teorik Arka Plan.....                              | 2   |
| 2.1 Lojistik Regresyon .....                          | 2   |
| 2.1.1 En Küçük Kareler Yöntemi (Least Squares): ..... | 2   |
| 2.1.2 Maksimum Olabilirlik (Maximum Likelihood):..... | 2   |
| 2.1 SVM-SVC .....                                     | 3   |
| 2.2 Random Forest.....                                | 3   |
| 2.3 KNN.....  | 3   |
| 2.4 3D Convolutional Neural Network (Conv3D), .....   | 4   |
| 3. Kullanılan Metrikler.....                          | 5   |
| 3.1 Accuracy (Doğruluk) .....                         | 5   |
| 3.2 Precision (Kesinlik).....                         | 5   |
| 3.3 Recall (Duyarlılık).....                          | 5   |
| 3.4 F-1 Score: .....                                  | 6   |
| 3.5 ROC: .....  | 6   |
| 3.6 AUC:.....   | 7   |
| 4. Model Performanslarının Değerlendirilmesi .....    | 8   |
| 4.1 Veri Dağılımı: .....                              | 8   |
| 4.2 Lojistik Regresyon:.....                          | 9   |
| 4.3 SVC: .....  | 10  |
| 4.4 Random Forest:.....                               | 11  |
| 4.5 Random Forest:.....                               | 12  |
| 4.6 Conv 3D: .....                                    | 13  |

# ÖZET

## Video Verilerinden İnsan Aktivitesi Tanımaya Yönelik Yapay Zeka Uygulamaları

Bu çalışmada, insan aktivitelerinin sınıflandırılması amacıyla Kaggle Human Activity Recognition Video Dataset veri setindeki 7 farklı video kategorisi kullanılmıştır: Clapping, Meeting and Splitting, Sitting, Standing Still, Walking, Walking While Reading a Book, ve Walking While Using the Phone. Çalışmada, Lojistik Regresyon, SVM, Random Forest, KNN ve 3D Konvolüsyonel Sinir Ağı (Conv3D) gibi algoritmalar kullanılarak sınıflandırma testleri yapılmış ve bu algoritmaların performansları karşılaştırılmıştır. Çalışma, algoritmaların teorik temellerini açıklayarak, son kullanıcı ihtiyaçlarına uygun bir ürün geliştirmeyi hedeflemektedir.

### Anahtar Kelimeler:

İnsan Aktivite Tanıma, Video Sınıflandırma, Makine Öğrenmesi, Derin Öğrenme, Lojistik Regresyon, Destek Vektör Makineleri (SVM), Rastgele Orman, K-En Yakın Komşu (KNN), 3D Konvolüsyonel Sinir Ağı (Conv3D), Kaggle Veri Seti, Video Analizi, Aktivite Tanıma, Sınıflandırma Algoritmaları.

# **ABSTRACT**

## **Artificial Intelligence Applications for Recognizing Human Activity from Video Data**

In this study, 7 different video categories in the Kaggle Human Activity Recognition Video Dataset dataset were used to classify human activities: Clapping, Meeting and Splitting, Sitting, Standing Still, Walking, Walking While Reading a Book, and Walking While Using the Phone. In the study, classification tests were performed using algorithms such as Logistic Regression, SVM, Random Forest, KNN and 3D Convolutional Neural Network (Conv3D) and the performances of these algorithms were compared. The study aims to develop a product that suits end-user needs by explaining the theoretical foundations of algorithms.

### **Keyword:**

Human Activity Recognition, Video Classification, Machine Learning, Deep Learning, Logistic Regression, Support Vector Machines (SVM), Random Forest, K-Nearest Neighbor (KNN), 3D Convolutional Neural Network (Conv3D), Kaggle Dataset, Video Analysis, Activity Recognition, Classification Algorithms.

# 1. GİRİŞ VE AMAÇ

Bu çalışmanın temel amacı, Kaggle veri seti üzerinde yer alan 7 farklı video kategorisinin sınıflandırılmasına yönelik bir çözüm geliştirmektir. İlgili video kategorileri şunlardır: **Clapping** (El Çırpma), **Meeting and Splitting** (Buluşma ve Ayrılma), **Sitting** (Oturma), **Standing Still** (Duran Durum), **Walking** (Yürüyüş), **Walking While Reading a Book** (Kitap Okurken Yürüyüş) ve **Walking While Using the Phone** (Telefon Kullanarak Yürüyüş). Bu sınıflandırma süreci sırasında karşılaşılan zorluklar tespit edilerek, nihayetinde son kullanıcı ihtiyaçlarını karşılayabilecek bir ürün haline getirilmesi hedeflenmektedir.

Bu çalışmada, dört farklı makine öğrenme algoritması ve bir derin öğrenme modeli kullanarak testler gerçekleştirilmektedir.

Makine öğrenme algoritmaları arasında **Lojistik Regresyon** (Logistic Regression), **Destek Vektör Makineleri** (SVM), **Rastgele Orman** (Random Forest) ve **K-En Yakın Komşu** (KNN) yer almaktadır. Ayrıca, derin öğrenme modeli olarak **3D Konvolüsyonel Sinir Ağı** (Conv3D) uygulanmıştır. Çalışma kapsamında, bu algoritmaların sonuçları karşılaştırılmakta ve her birinin teorik arka planları detaylı bir şekilde açıklanmaktadır.

## 2. Teorik Arka Plan

Bu Bölümde bu çalışmada kullanılan Makine öğrenmesi ve Derin Öğrenme Modellerinin Matematiksel altyapısı anlatmak hedeflenmektedir

### 2.1 Lojistik Regresyon

Özellikle, sınıflandırma problemlerinde, örneğin bir görseldeki hayvanın kedi mi yoksa köpek mi olduğunu veya verilen bilgilerin erkeğe mi yoksa kadına mı ait olduğunu tahmin etme gibi iki sınıflı sınıflandırma görevlerinde sıkça kullanılmaktadır.

Lojistik regresyonun, lineer regresyon ile arasındaki en büyük fark, sınıflar arasındaki ayırıcı çizgiyi nasıl oluşturduğudur. Lineer regresyon, veriler arasındaki ilişkiyi modellemek için **En Küçük Kareler Yöntemi (Least Squares)** kullanırken, lojistik regresyon **Maksimum Olabilirlik (Maximum Likelihood)** yöntemini kullanarak sınıflar arasındaki sınırı belirler. Lojistik regresyon, doğrusal bir modelin çıktısını, her iki sınıf arasındaki olasılıkları 0 ile 1 arasında sıkıştırmak için bir logistik fonksiyon (**sigmoid fonksiyonu**) uygular. Bu sayede, sonuçlar kesin sınıf tahminlerine dönüşür.

#### 2.1.1 En Küçük Kareler Yöntemi (Least Squares):

$$y = aX + b$$

Basit bir örnek vermek gerekirse, aralarında doğrusal (lineer) bir bağlantı olan, X ve Y adında iki fiziksel büyüklük düşünelim. (Mesela, X belli bir ağaç türünün yaşı, Y aynı tür ağacın gövde çapı olabilir.) Y 'yi X 'in fonksiyonu olarak yazmak istiyoruz. Bu iki büyüklük arasındaki bağlantı doğrusal olduğuna göre, şöyle bir denklem halinde ifade edilebilir: Bizim aradığımız şey, bu denklemdeki a ve b sayıları için mümkün olan en doğru değerlerdir.

#### 2.1.2 Maksimum Olabilirlik (Maximum Likelihood):

Maksimum olasılık, verinin altında yatan dağılımın parametrelerini tahmin etmek için kullanılan güçlü bir yöntemdir. Bu metodun matematiksel temeli, verinin gözlemlenme olasılığını maksimum yapmak üzerine kuruludur ve genellikle log-likelihood fonksiyonuyla çalışılır.

$$\theta = \arg \max_{\theta} \log L(\theta)$$

## 2.1 SVM-SVC

**Destek Vektör Makineleri (SVM)**, sınıflandırma, regresyon ve aykırı değerlerin tespiti için kullanılan denetimli öğrenme (supervised learning) tekniklerinden biridir. SVM, özellikle **sınıflandırma** (classification) alanında gelişmiş bir algoritma olup, diğer sınıflandırma algoritmalarından bazı temel farklılıklar göstermektedir. Bu algoritma, verileri belirli sınıflara ayıran **en uygun sınırı** (decision boundary) bulma amacı güder.

## 2.2 Random Forest

Rastgele Orman algoritması, Makine Öğrenmesinde güçlü bir ağaç öğrenme tekniğidir . Eğitim aşamasında bir dizi Karar Ağacı oluşturarak çalışır . Her ağaç, her bölümdeki özelliklerin rastgele bir alt kümesini ölçmek için veri kümesinin rastgele bir alt kümesi kullanılarak oluşturulur. Bu rastgelelik, bireysel ağaçlar arasında değişkenlik yaratarak aşırı uyum riskini azaltır ve genel tahmin performansını iyileştirir.

## 2.3 KNN

kNN hassas ancak hesaplama açısından yoğun olduğundan büyük veri kümeleri için daha az uygundur

KNN Nasıl Çalışır:

1. Vektörleştirme: Veri kümesindeki her veri noktası, çok boyutlu bir alanda vektör olarak temsil edilir.
2. Uzaklık Hesaplaması: Yeni bir veri noktasını (sorgu noktası) sınıflandırmak için algoritma, uzaklık işlevini kullanarak sorgu noktası ile veri kümesindeki diğer tüm noktalar arasındaki uzaklığı hesaplar.
3. Komşuları Bulma: Algoritma, hesaplanan mesafelere göre sorgu noktasına en yakın k veri noktalarını (komşular) tanımlar. K değeri (komşu sayısı) çok önemlidir. Küçük bir k gürültüye duyarlı olabilirken, büyük bir k ayrıntıları düzeltebilir.
4. Tahminde Bulunma:
  - Sınıflandırma: Sınıflandırma görevleri için kNN, k komşular arasında en yaygın olan sorgu noktasına sınıf etiketini atar. Temelde, "çoğunluk oyu" gerçekleştirir.
  - Regresyon: Regresyon görevleri için kNN, sorgu noktasının değerini k komşularının değerlerinin ortalaması (veya bazen ağırlıklı ortalaması) olarak tahmin eder.

## 2.4 3D Convolutional Neural Network (Conv3D),

3D Convolutional Neural Network (Conv3D), özellikle video verisi gibi üç boyutlu (zaman ve uzay) verilerin işlenmesinde kullanılan bir derin öğrenme mimarisidir. Bu katman, zaman (z) boyutuna da sahip verilerle çalışmak için geliştirilmiş bir evrişim (convolution) yöntemidir. Conv3D, 2D evrişim katmanlarının uzantısıdır ve video verisi gibi üç boyutlu verileri işlerken hem uzamsal (x, y) hem de zaman (z) boyutlarını dikkate alır.

### 1. peki Evrişim Nedir?

Evrişim (Convolution), bir sinyalin veya verinin, başka bir fonksiyonla (genellikle kernel veya filtre) birleşerek yeni bir özellik haritası (feature map) oluşturması işlemidir. Evrişim işlemi, özellik çıkarımı yapmak amacıyla kullanılır. 2D Convolution, genellikle görüntülerdeki özellikleri öğrenmek için kullanılırken, 3D Convolution, zaman boyutu da içeren verilerle çalışmak için geliştirilmiştir (örneğin, videolar).

### 2. Conv3D Katmanı

Conv3D katmanı, geleneksel 2D Convolution katmanının zaman boyutunu içeren versiyonudur. Bu, 3D veriler üzerinde evrişim yapabilen bir katmandır. 3D veriler genellikle bir dizi 2D görüntüden (video karesi) oluşur, bu da her karedeki x ve y boyutlarına ek olarak, her video karesinin bir zaman boyutu (z) olduğunu gösterir.

Conv3D katmanında kullanılan kernel (filtre), 3D bir yapıdadır. 3D evrişim, yalnızca görüntünün iki uzamsal boyutunda (genişlik ve yükseklik) değil, aynı zamanda üçüncü boyut (zaman) boyunca da özellikleri öğrenir.

Örneğin, bir video verisi, 10 kareden oluşuyorsa, her bir kare 224x224x3 (yükseklik, genişlik, renk kanal sayısı) boyutlarına sahip olabilir. Bu durumda, video verisi (10, 224, 224, 3) boyutlarında bir veri setidir.



### 3. Kullanılan Metrikler

İlgili bölümde çalışmada kullanılan Makine öğrenmesi ve Derin Öğrenme Modellerini ölçmek için kullanılan metrikler hakkında bilgi verilmiştir

#### 3.1 Accuracy (Doğruluk)

makine öğrenimi modelinin sonucu ne sıklıkla doğru tahmin ettiğini ölçen bir ölçümdür. Doğruluğu, doğru tahmin sayısını toplam tahmin sayısına bölerek hesaplayabilirsiniz.

#### 3.2 Precision (Kesinlik)

Positive olarak tahminlediğimiz değerlerin gerçekten kaç adedinin Positive olduğunu göstermektedir.

$$TP / (TP+FP)$$

Kesinlik değeri özellikle False Positive tahminlemenin maliyeti yüksek olduğu durumlarda çok önemlidir. Örneğin mail kutunuza gelmesi gereken mailleri eğer modeliniz spam olarak işaretlerse (FP) bu durumda almanız gereken önemli mailleri görememiş olur ve sizin için kayıp yaratan bir durumda kalırsınız. Bu durumda **Kesinlik** değerinin yüksek olması bizim için model seçiminde önemli bir kriterdir.

#### 3.3 Recall (Duyarlılık)

Positive olarak tahmin etmemiz gereken işlemlerin ne kadarını Positive olarak tahmin ettiğimizi gösteren bir metriktir.

$$TP / (TP+FN)$$

Duyarlılık değeri de False Negative olarak tahminlemenin maliyetinin yüksek olduğu durumlarda bize yardımcı olacak bir metriktir. Mümkün olduğunca yüksek olması gereklidir. Örneğin Fraud Detection için oluşturduğumuz bir model Fraud olan bir işlemi non-fraud olarak işaretlerse böyle bir durumun sonuçları bir banka için sorun teşkil edecektir.

### 3.4 F-1 Score:

Score değeri bize Kesinlik (Precision) ve Duyarlılık (Recall) değerlerinin harmonik ortalamasını göstermektedir.

Basit bir ortalama yerine harmonik ortalama olmasının sebebi ise uç durumları da gözardı etmememiz gerektiğidir. Eğer basit bir ortalama hesaplaması olsaydı Precision değeri 1 ve Recall değeri 0 olan bir modelin F1 Score'u 0.5 olarak gelecektir ve bu bizi yanıltacaktır.

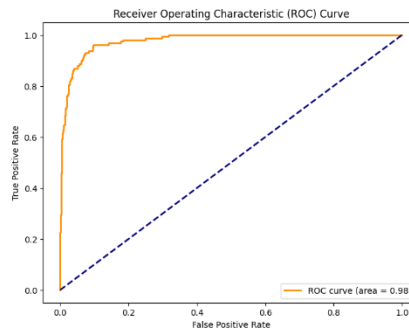
$$2 * ((\text{precision} * \text{recall}) / (\text{precision} + \text{recall}))$$

Doğruluk (Accuracy) yerine F1 Score değerinin kullanılmasının en temel sebebi eşit dağılmayan veri kümelerinde hatalı bir model seçimi yapmamaktır. Ayrıca sadece False Negative ya da False Positive değil tüm hata maliyetlerini de içerecek bir ölçme metriğine ihtiyaç duyulduğu içinde F1 Score bizim için çok önemlidir.

### 3.5 ROC:

ROC eğrisi sınıflandırma problemleri için çok önemli bir performans ölçümüdür. ROC bir olasılık eğrisidir ve altında kalan alan olan AUC ayrılabirliğin derecesini veya ölçüsünü temsil eder.

ROC eğrisinde X ekseninde FPR(Yanlış Pozitif Oran) ve Y ekseninde ise TPR (Gerçek Pozitif Oranı) bulunmaktadır.



### 3.6 AUC:

**AUC** (Area Under the Curve), bir **performans değerlendirme metriği** olup, özellikle sınıflandırma problemlerinde modelin başarısını ölçmek için kullanılır. AUC, özellikle **ROC (Receiver Operating Characteristic)** eğrisinin altında kalan alanı temsil eder. ROC eğrisi, modelin **doğru pozitif oranı** (True Positive Rate - TPR) ile **yanlış pozitif oranı** (False Positive Rate - FPR) arasındaki ilişkiyi gösteren bir grafik oluşturur.

AUC, bu eğrinin altındaki alanı hesaplar ve modelin genel doğruluğunu özetler. AUC değeri **0 ile 1 arasında** değişir:

- **AUC = 1.0:** Model mükemmel bir şekilde sınıflandırma yapıyordur. Yani, pozitif ve negatif sınıfları mükemmel şekilde ayırabiliyor.
- **AUC = 0.5:** Model tamamen rastgele tahminler yapıyor demektir. Model, pozitif ve negatif sınıfları ayırt edemiyor.
- **AUC < 0.5:** Modelin performansı kötü. Modelin sınıfları ters şekilde ayırdığı anlamına gelebilir. (Yani, model yanlış sınıflandırma yapıyor.)

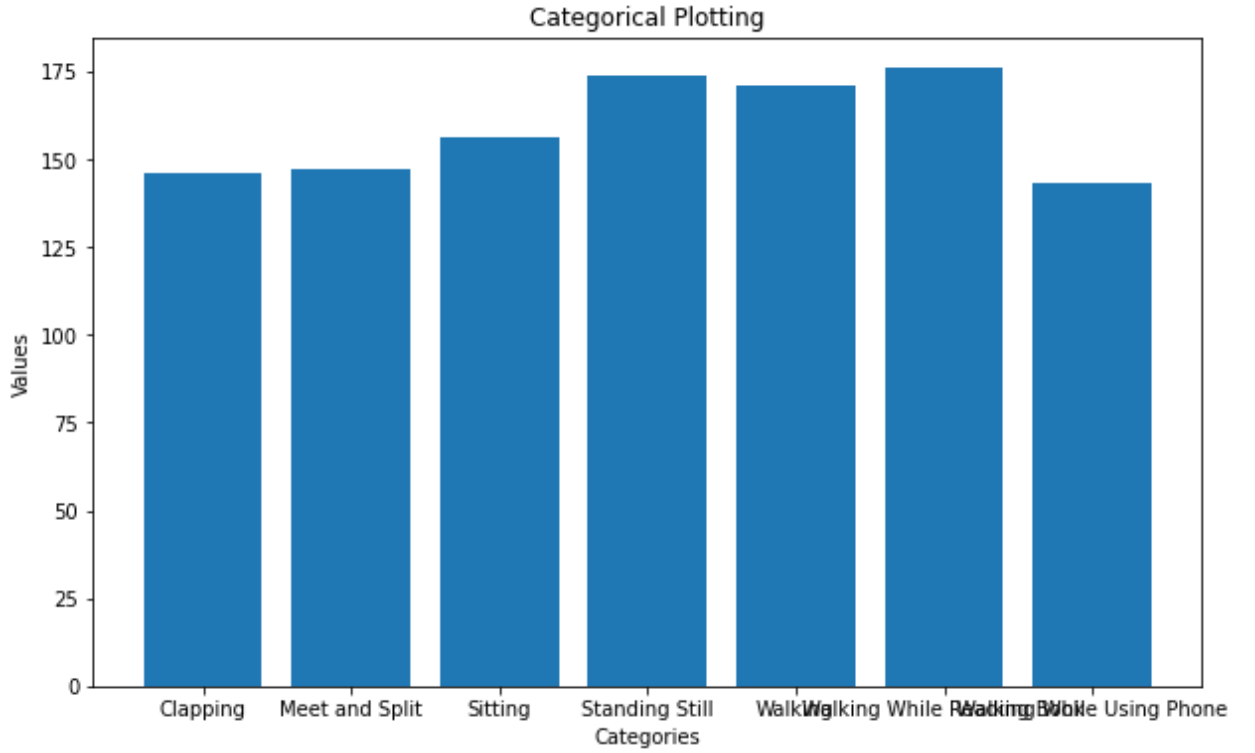
## 4. Model Performanslarının Değerlendirilmesi

İlgili bölümde çalışmada kullanılan Makine öğrenmesi ve Derin Öğrenme Modellerini Başarı

Oranları Verilmiştir

### 4.1 Veri Dağılımı:

Veri Dağılımı Grafik1.1'deki Gibidir:



Grafik1.1

## 4.2 Lojistik Regresyon:

- **Parametreler**

- **max\_iter=10000:**

- **Accuracy (Doğruluk):**

- **0.2893**

Model, sınıflandırma doğruluğu açısından %28,93'lük bir başarı elde etmiştir..

- **F1 Score:**

- **0.2346**

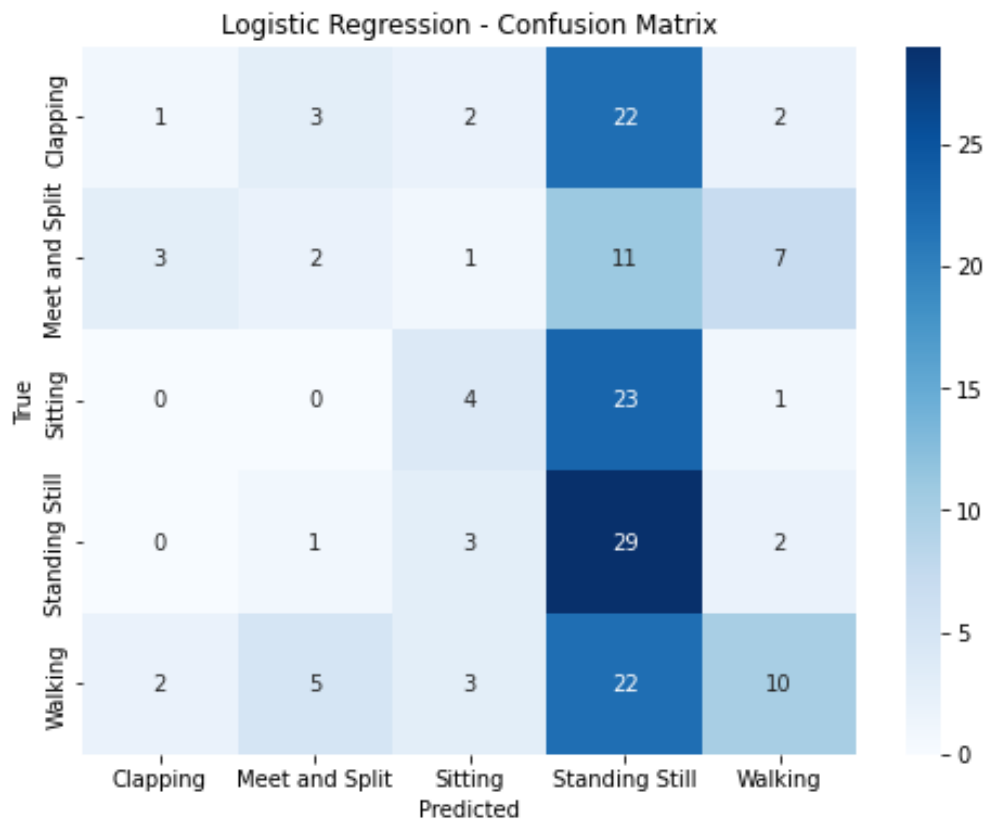
- **Precision (Kesnlik):**

- **0.2928**

Modelin, pozitif olarak sınıflandırılan örneklerin ne kadarının gerçekten pozitif olduğunu gösterir

- **Recall (Duyarlılık):**

- **0.2893**



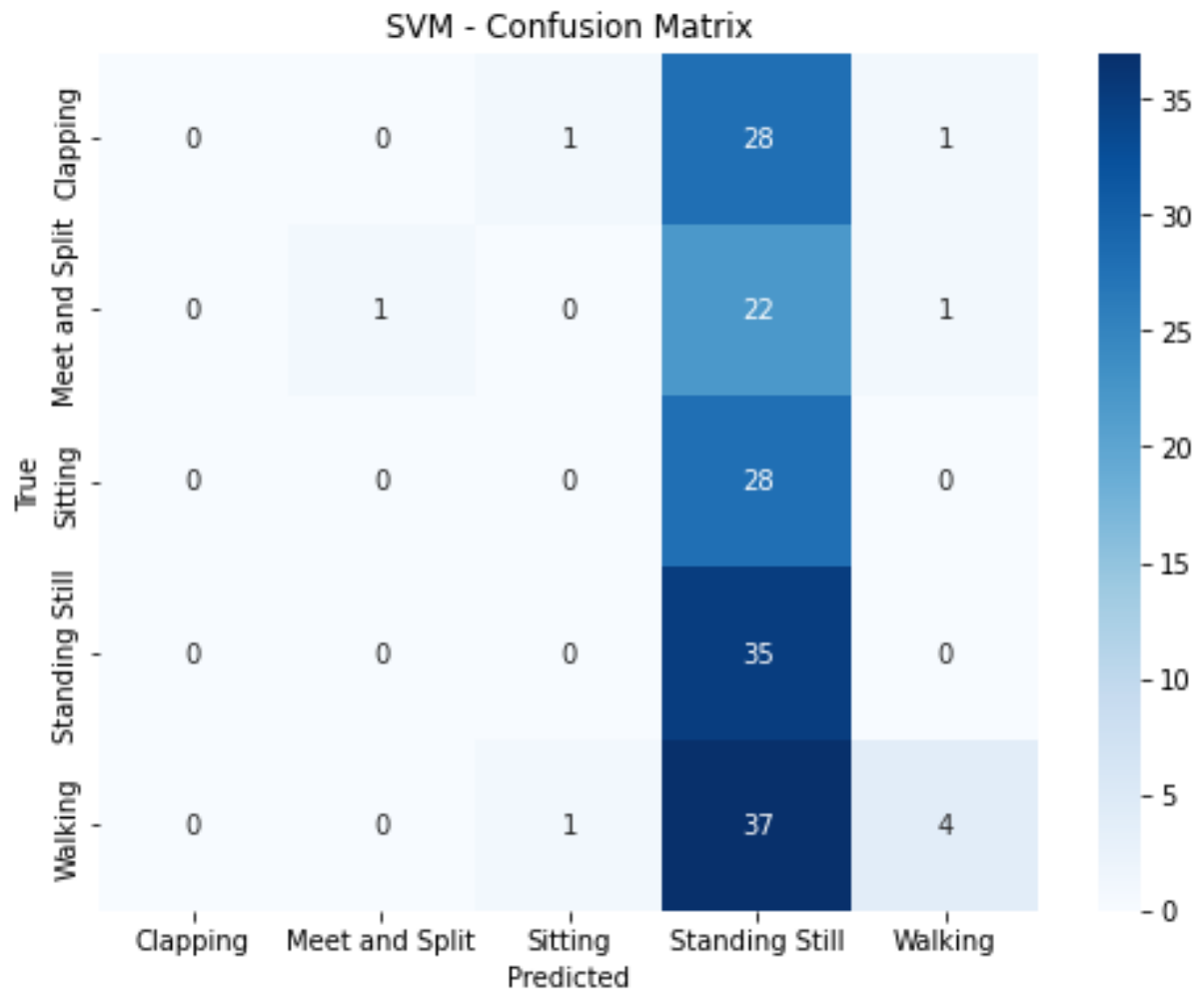
### 4.3 SVC:

- **Parametre Ayarları:**

- **kernel='linear':** Doğrusal sınıflandırma için bir çekirdek fonksiyonu kullanılmıştır.
- **probability=True:** Modelin tahmin olasılıklarını döndürmesi sağlanmıştır.

Modelin çeşitli metriklerle ölçülen sonuçları:

- **Accuracy (Doğruluk):** 0.2516
- **F1 Score:** 0.1394
- **Precision (Kesinlik):** 0.3784
- **Recall (Duyarlılık):** 0.2516



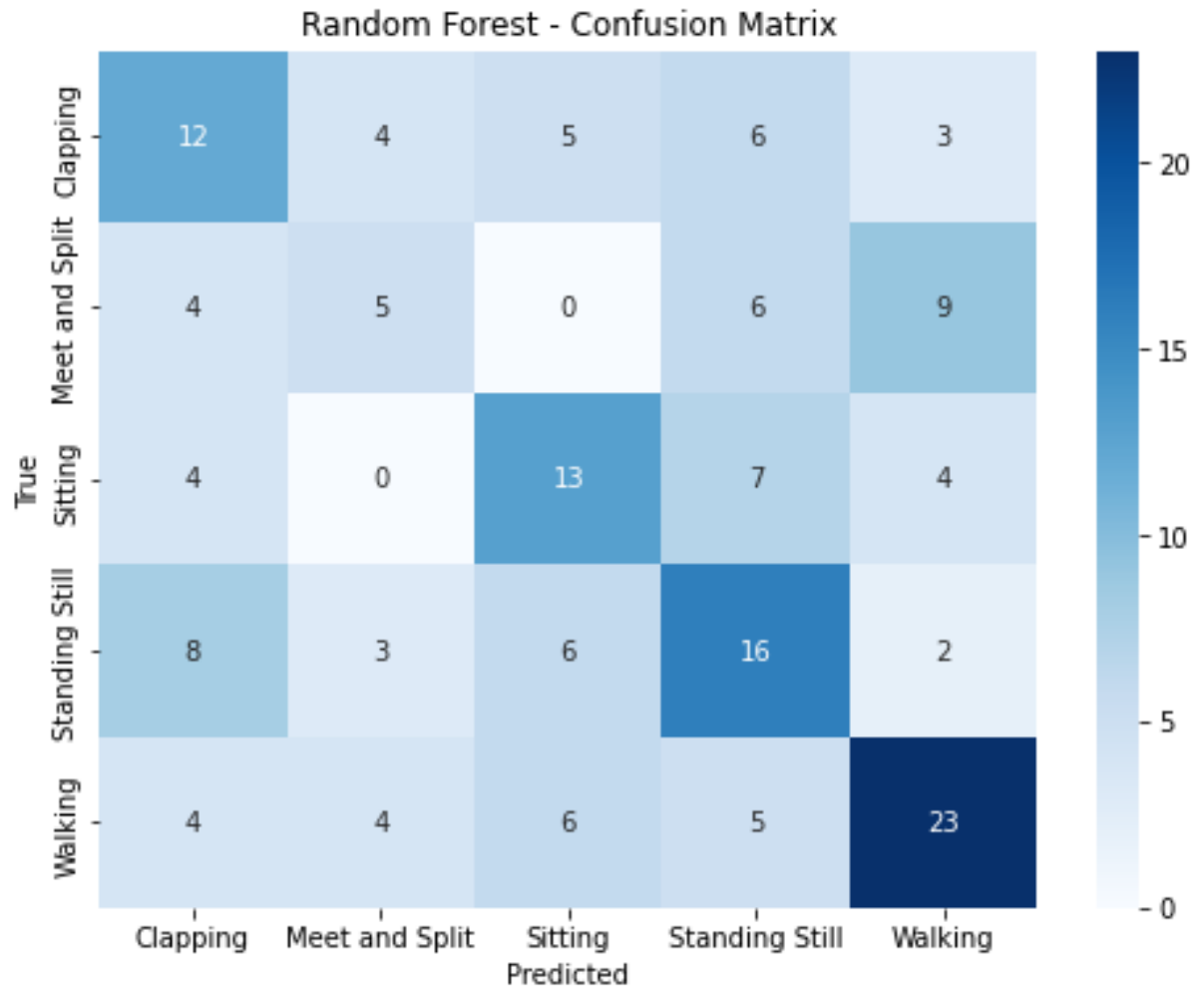
#### 4.4 Random Forest:

- **Parametre Ayarları:**

- **n\_estimators=300:** Modelde 300 adet karar ağacı kullanılmıştır. Bu, daha sağlam ve dengeli bir sınıflandırma performansı sağlamayı amaçlamaktadır.

Modelin çeşitli metriklerle ölçülen sonuçları:

- **Accuracy (Doğruluk):** 0.4340
- **F1 Score:** 0.4300
- **Precision (Doğruluk):** 0.4305
- **Recall (Duyarlılık):** 0.4340



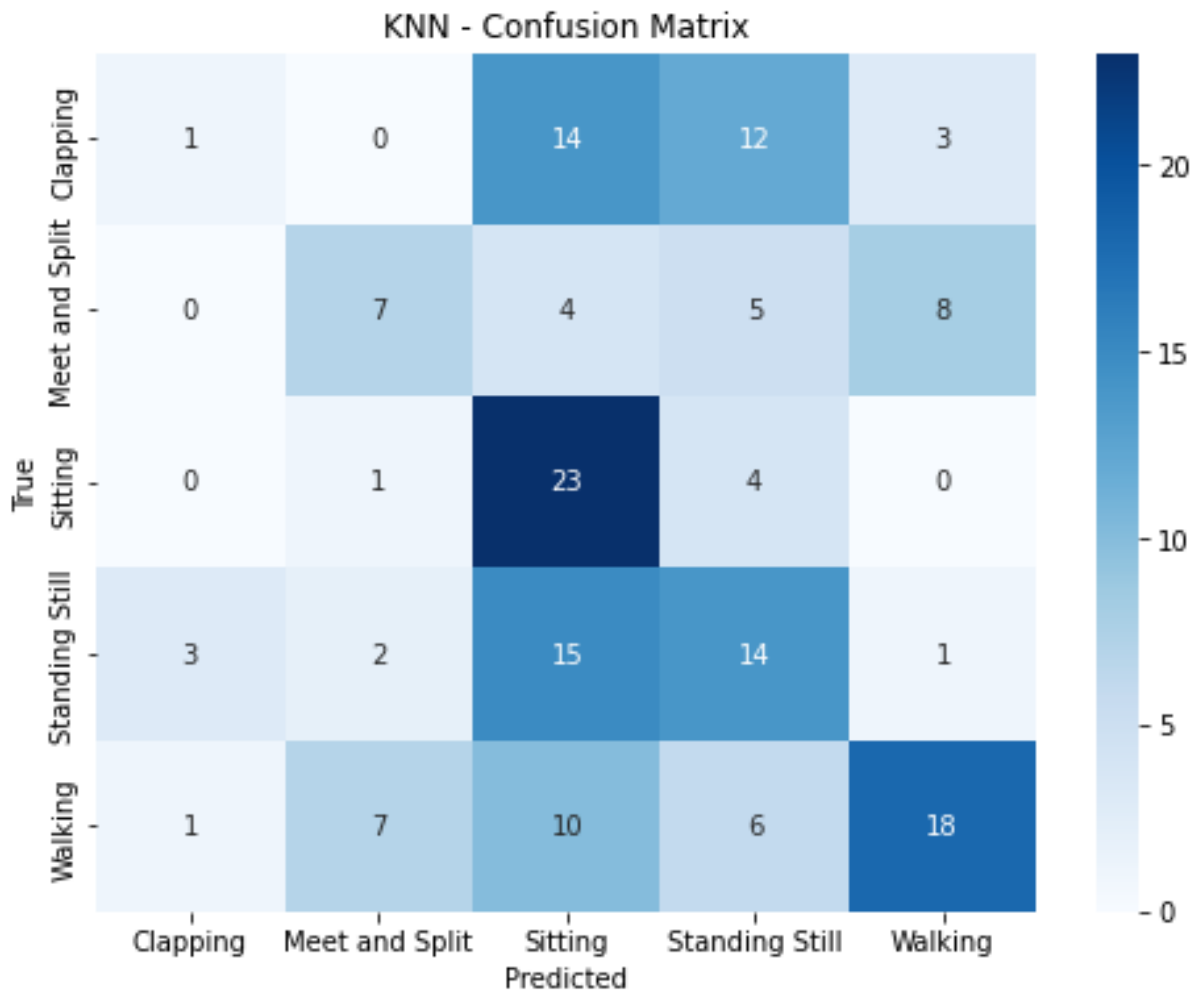
## 4.5 Random Forest:

- **Parametre Ayarları:**

- **n\_neighbors=55:** Model, her veri noktası için sınıf tahminlerini belirlemek amacıyla en yakın 55 komşuyu dikkate almıştır.

Modelin çeşitli metriklerle ölçülen sonuçları:

- **Accuracy (Doğruluk):** 0.3962
- **F1 Score:** 0.3617
- **Precision (Doğruluk):** 0.3949
- **Recall (Duyarlılık):** 0.3962





## 4.6 Conv 3D:

### - Model Mimarisi:

- **Girdi Şekli:** (10, 224, 224, 3)
- **Katmanlar:**
  - Conv3D (32 filtre, çekirdek boyutu 3, "same" padding, ReLU aktivasyonu)
  - MaxPooling3D
  - Conv3D (64 filtre, çekirdek boyutu 3, "same" padding, ReLU aktivasyonu)
  - MaxPooling3D
  - Conv3D (128 filtre, çekirdek boyutu 3, "same" padding, ReLU aktivasyonu)
  - MaxPooling3D
  - Dropout (oran: 0.3)
  - GlobalAveragePooling3D
  - Dense (kategoriler kadar çıktı, softmax aktivasyonu)

### - Eğitim Yapılandırması:

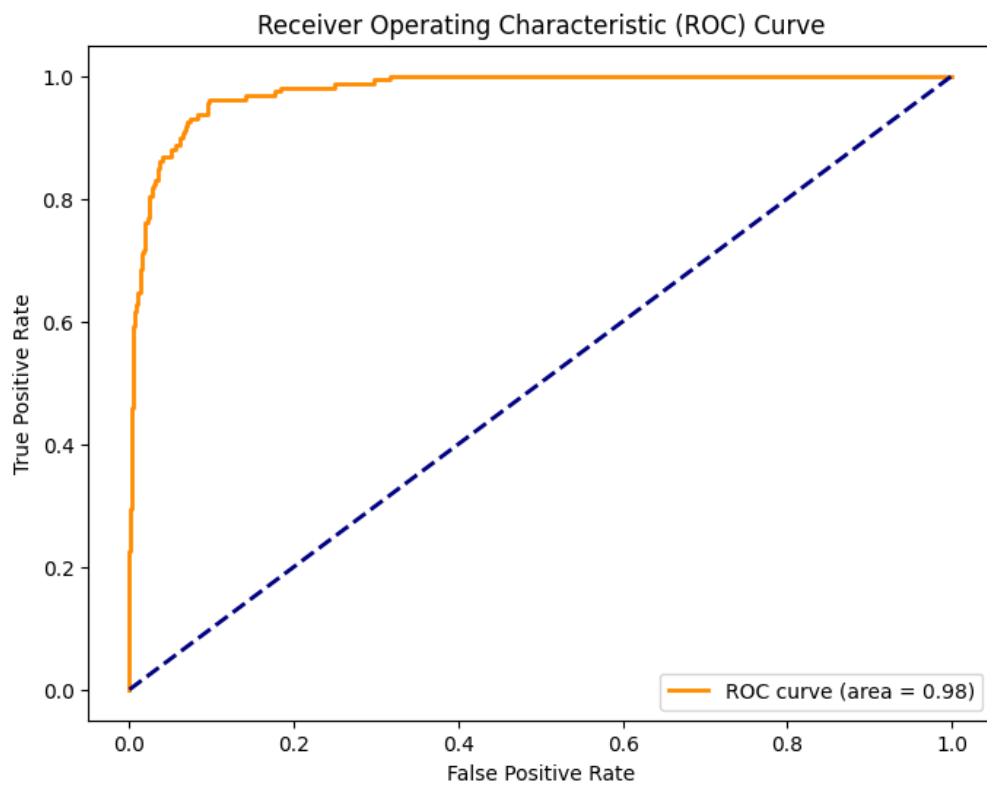
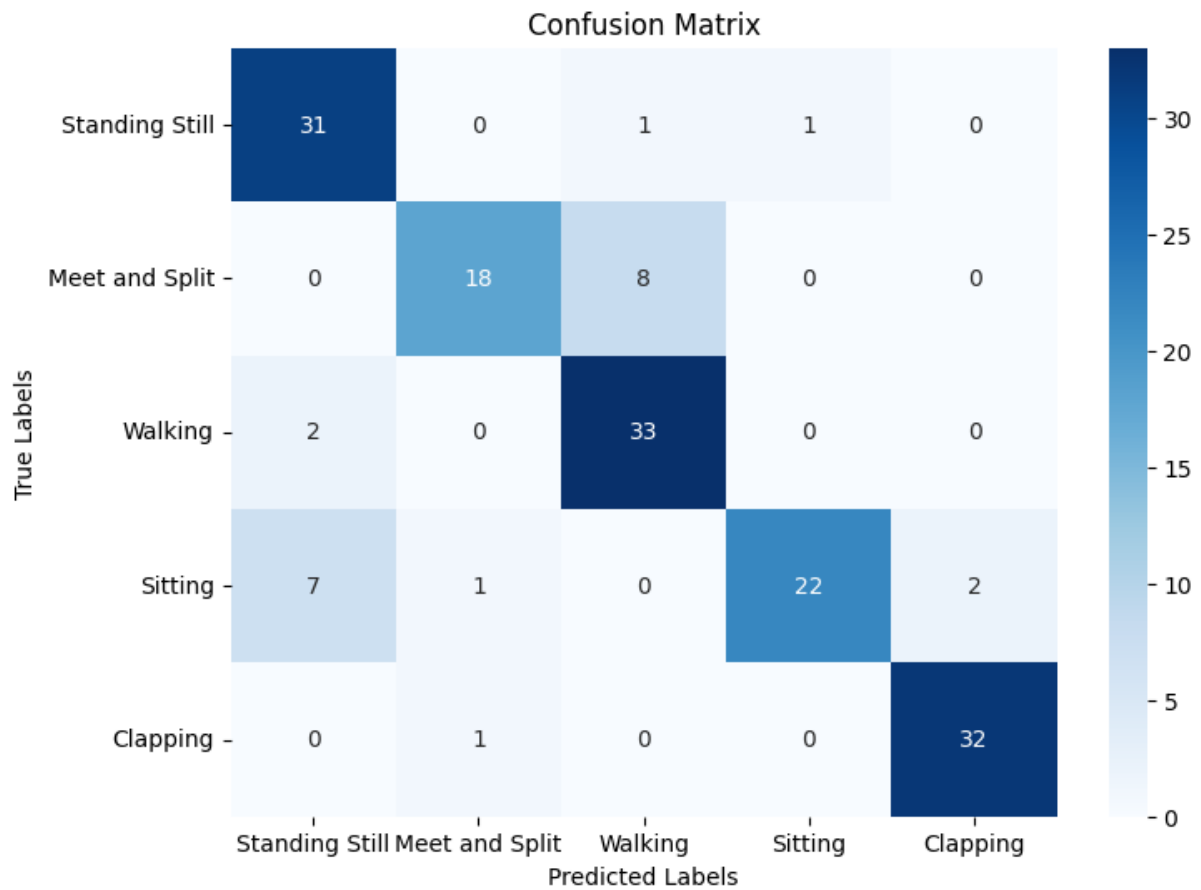
- **Loss Fonksiyonu:** SparseCategoricalCrossentropy
- **Optimizasyon:** Adam (öğrenme oranı: 0.001)
- **Değerlendirme Metrikleri:** Accuracy
- **Epoch Sayısı:** 30
- **Batch Boyutu:** 32

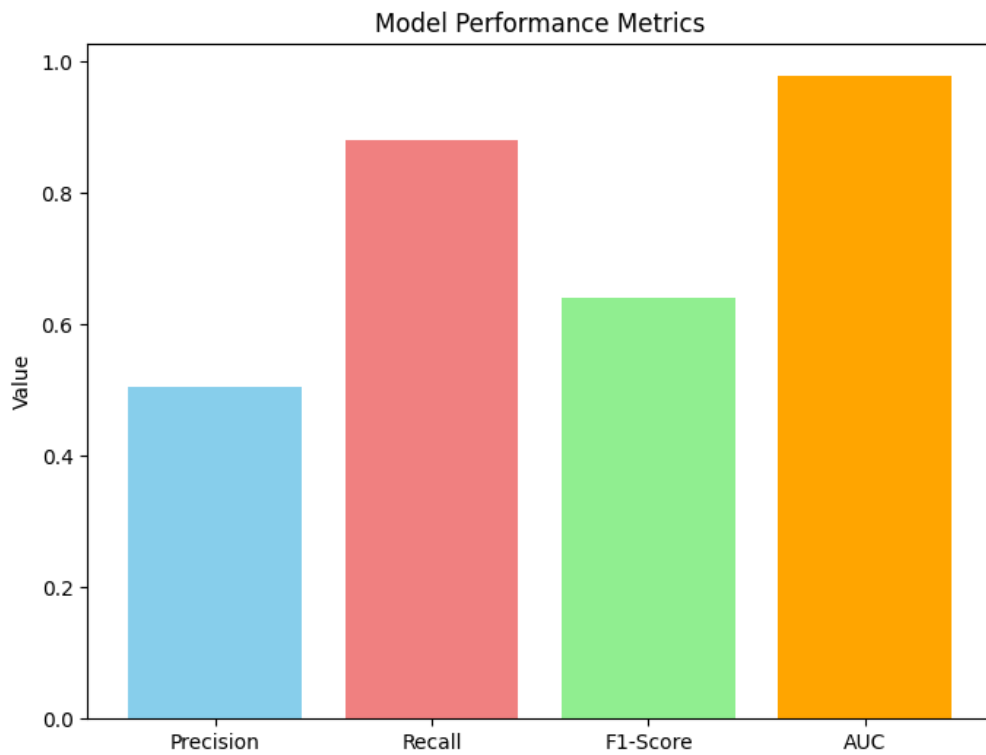
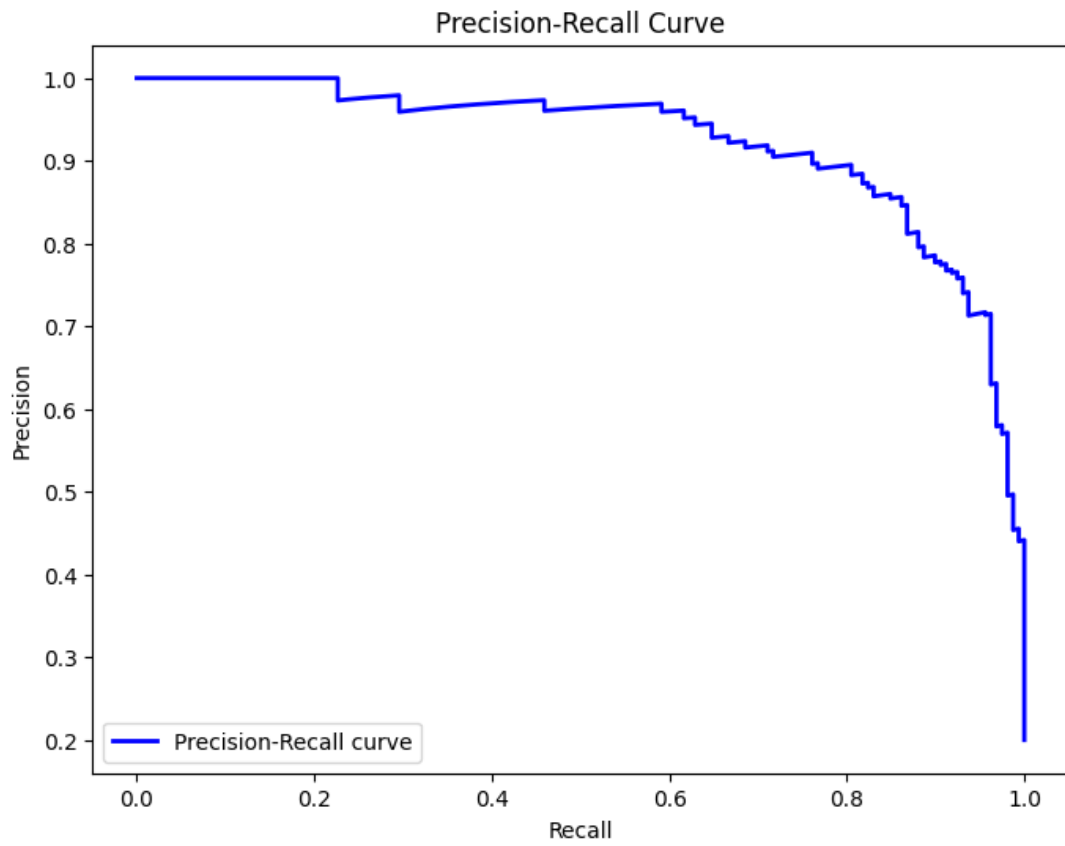
### - Callbacks:

- **ModelCheckpoint:**
  - En iyi modelin validation accuracy'e göre kaydedilmesi.
  - En iyi ağırlıkların geri yüklenmesi sağlandı.

### - Eğitim Sonuçları:

- **Eğitim Kaybı (Loss):** 0.4435
- **Eğitim Doğruluğu (Accuracy):** 0.8239
- **Doğrulama Kaybı (Validation Loss):** 0.4435
- **Doğrulama Doğruluğu (Validation Accuracy):** 0.8239





| Classification Report: |           |        |          |         |
|------------------------|-----------|--------|----------|---------|
|                        | precision | recall | f1-score | support |
| Standing Still         | 0.78      | 0.94   | 0.85     | 33      |
| Meet and Split         | 0.90      | 0.69   | 0.78     | 26      |
| Walking                | 0.79      | 0.94   | 0.86     | 35      |
| Sitting                | 0.96      | 0.69   | 0.80     | 32      |
| Clapping               | 0.94      | 0.97   | 0.96     | 33      |
| accuracy               |           |        | 0.86     | 159     |
| macro avg              | 0.87      | 0.85   | 0.85     | 159     |
| weighted avg           | 0.87      | 0.86   | 0.85     | 159     |

## 5. Model Değerlendirmesi ve Sonuç

Bu çalışmada değerlendirilen beş modelden, **Conv3D tabanlı derin öğrenme modeli** en yüksek doğruluk ve metrik değerlerine ulaşarak diğer modellerden üstün performans göstermiştir. Logistic Regression, SVM, Random Forest ve KNN gibi klasik makine öğrenme algoritmalarına kıyasla, Conv3D modeli özellikle karmaşık video verilerinin özelliklerini öğrenmede daha etkili olmuş ve aşağıdaki sonuçları elde etmiştir:

- **Eğitim Doğruluğu (Accuracy):** 0.8239
- **Doğrulama Doğruluğu (Validation Accuracy):** 0.8239
- **Eğitim Kaybı (Loss):** 0.4435
- **Doğrulama Kaybı (Validation Loss):** 0.4435

Bu sonuçlar, Conv3D modelinin video sınıflandırma görevlerinde derin öğrenme tabanlı yaklaşımların klasik yöntemlere kıyasla üstün olduğunu açıkça ortaya koymaktadır. Özellikle 3D verilerin zaman ve mekan boyutlarındaki ilişkilerini anlamadaki başarısı, Conv3D modelini bu problem için en uygun çözüm haline getirmiştir.

# Kaynakça

## Kaynakça

1. Fakca, M. *Lojistik Regresyon Nedir? Nasıl Çalışır?* Medium. <https://mfakca.medium.com/lojistik-regresyon-nedir-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-4e1d2951c5c1>
2. Wikipedia. *En Küçük Kareler Yöntemi*. [https://tr.wikipedia.org/wiki/En\\_k%C3%BC%C3%A7%C3%BCk\\_kareler\\_y%C3%B6ntemi](https://tr.wikipedia.org/wiki/En_k%C3%BC%C3%A7%C3%BCk_kareler_y%C3%B6ntemi)
3. Fatihto, M. *Support Vector Machine Algoritması - Makine Öğrenmesi*. Medium. <https://mfatihto.medium.com/support-vector-machine-algoritmas%C4%B1-makine-%C3%B6%C4%9Frenmesi-8020176898d8>
4. GeeksforGeeks. *Random Forest Algorithm in Machine Learning*. <https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/>
5. Microsoft Learn. *KNN vs ANN*. <https://learn.microsoft.com/tr-tr/azure/cosmos-db/gen-ai/knn-vs-ann>
6. Evidently AI. *Accuracy, Precision, Recall*. <https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall#:~:text=Accuracy%20is%20a%20metric%20that,often%20the%20model%20is%20right%3F>
7. Ögün, G. *Doğruluk (Accuracy), Kesinlik (Precision), Duyarlılık (Recall) ya da F1-Score?* Medium. <https://medium.com/@gulcanogundur/do%C4%9Fruluk-accuracy-kesinlik-precision-duyarl%C4%B1%C4%B1k-recall-ya-da-f1-score-300c925feb38>
8. Ögün, G. *ROC ve AUC*. Medium. <https://medium.com/@gulcanogundur/roc-ve-auc-1fefcfc71a14>
9. TensorFlow. *Video Classification Tutorial*. [https://www.tensorflow.org/tutorials/video/video\\_classification](https://www.tensorflow.org/tutorials/video/video_classification)
10. TensorFlow. *SparseCategoricalCrossentropy Loss Function*. [https://www.tensorflow.org/api\\_docs/python/tf/keras/losses/SparseCategoricalCrossentropy](https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy)
11. Kaggle. *Video Classification Notebook*. [https://www.kaggle.com/code/lonnieqin/video-classification?source=post\\_page-----36309d1fa87c-----](https://www.kaggle.com/code/lonnieqin/video-classification?source=post_page-----36309d1fa87c-----)
12. TensorFlow. *SparseCategoricalCrossentropy Loss Function Documentation*. [https://www.tensorflow.org/api\\_docs/python/tf/keras/losses/SparseCategoricalCrossentropy](https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy)