

# Gölge Boks Hareket Sınıflandırması

Hakan Aytuğ Fırat

Kocaeli Üniversitesi

Bilişim Sistemleri Mühendisliği

[hakanaytugfirat@gmail.com](mailto:hakanaytugfirat@gmail.com) / 221307015

**Anahtar Kelimeler:** Gölge boks, hareket sınıflandırma, transformer modelleri, web scraping

## 1. GİRİŞ

Son yıllarda insan hareketlerinin otomatik olarak tanınması (Human Activity Recognition - HAR), bilgisayarla görü alanında büyük bir ilgi odağı haline gelmiştir. Bu teknolojinin spor analizi, sağlık takibi, güvenlik sistemleri ve insan-bilgisayar etkileşimi gibi birçok uygulama alanı bulunmaktadır. Bu çalışmada, dövüş sporlarında sıkça kullanılan bir antrenman yöntemi olan **gölge boks** üzerine odaklanılmıştır. Gölge boks, rakipsiz olarak yapılan hızlı ve çeşitli tekniklerin sergilendiği bir çalışmadır ve bu nedenle hareket sınıflandırması açısından oldukça zorlu bir veri kaynağıdır.

Bu proje kapsamında, ilk olarak yt-dlp ve selenium kütüphaneleri kullanarak veriler youtube üzerinden toplanmıştır daha sonra gölge boks yapan kişilerin hareketleri kısa aralıklara frame lere bölünmüştür. Frame lerden çıkarılan iskelet (pose) bilgileri üzerinden sınıflandırılmıştır. Amaç, altı farklı hareketin otomatik olarak tanınmasıdır: **aparkat, kroşe, direkt, eğilme (duck/slip), gard ve belirsiz**. İskelet verileri, video karelerinden elde edilen anahtar vücut noktaları yardımıyla oluşturulmuştur. Bu sayede modelin doğrudan görüntüdeki renk, arka plan ve aydınlatma gibi değişkenlerden etkilenmeden yalnızca vücut hareketlerine odaklanması sağlanmıştır.

Çalışmada, güncel görüntü tabanlı dönüştürücü (transformer) modeller olan **ViT, DeiT, Swin Transformer, BEiT** ve **ConvNeXt** karşılaştırmalı olarak değerlendirilmiştir. Modellerin başarımı; doğruluk (accuracy), hassasiyet (recall), özgüllük (specificity), F1 skoru ve AUC gibi performans metrikleri ile ölçülmüştür. Ayrıca eğitim-kayıp eğrileri, karışıklık matrisleri ve ROC eğrileri ile detaylı analizler sunulmuştur.

Bu çalışmanın başlıca katkıları şunlardır:

- Veriler web scraping yöntemiyle indirilip çeşitli işlemlerden geçtikten sonra kullanılabilir hale getirilmiştir.
- Gölge boks hareketlerine özel, poz (iskelet) verisine dayalı bir görüntü veri kümesi oluşturulmuştur.
- Beş farklı modern görüntü transformer modelinin performansları detaylı şekilde karşılaştırılmıştır.
- Tüm eğitim ve değerlendirme süreci Google Colab ortamında Python dili ile gerçekleştirilmiştir.

## 2. VERİ TOPLAMA (WEB SCRAPING)

Bu çalışmada kullanılan görüntü verileri, dövüş sporlarına ait çeşitli gölge boks videolarının YouTube üzerinden otomatik olarak toplanmasıyla elde edilmiştir. Veri toplama işlemi, Python programlama dili kullanılarak Selenium ve yt-dlp (YouTube video indirme aracı) kütüphaneleri ile gerçekleştirilmiştir.

Veri çekme süreci üç ana aşamadan oluşmaktadır:

**Arama ve sayfa tarama:** Selenium kullanılarak YouTube’da belirli anahtar kelimeler (örn. “*uppercut boxing*”, “*shadow boxing training*”) ile arama yapılmış ve sonuç sayfasında yer alan videoların bağlantıları tespit edilmiştir.

**Bağlantıların filtrelenmesi:** Video bağlantıları sistematik olarak taranmış, tekrar eden veya geçersiz bağlantılar elenmiştir.

**Video indirme:** Elde edilen geçerli bağlantılar yt-dlp kütüphanesi aracılığıyla doğrudan mp4 formatında belirtilen dizine indirilmiştir.

Kodun esnek yapısı sayesinde farklı hareketler için farklı anahtar kelimelerle aramalar yapıp, her bir sınıf için veri çeşitliliği sağlanmıştır. Komut satırı parametreleri kullanılarak indirilecek video sayısı (--limit) ve hedef dizin (--output) gibi değerler kolayca ayarlanabilmektedir.

Veri toplama işlemi sonunda indirilen videolar raw\_videos/ klasöründe toplanmış, sonraki adımda bu videolar karelerine (frame) ayrılarak görsel sınıflandırma için kullanılacak hale getirilmiştir.

```
python video_scraper.py --search "shadow boxing uppercut" --limit 10 --output raw_videos
```

Yukarıdaki kod betiğinde uygulamanın web scraping kısmının çalışma şartları belirlenmiştir. Limit kaç video indirileceği search hangi isimle video atılacağı output ise nereye indirileceğidir. Eğer limit yazılmazsa uygulama kapatılana kadar video indirilmeye devam edilir.

### 3. Videoların Kare Görüntülere Dönüştürülmesi (Frame Extraction)

Toplanan videolar, doğrudan modele girdi olarak kullanılamayacağı için öncelikle kare (frame) görüntülere bölünmüştür. Bu işlem, her bir videodan belirli aralıklarla görsel elde edilmesini sağlar. Projede bu işlem için her 10. frame alınacak şekilde bir strateji belirlenmiştir. Böylece hem veri çeşitliliği artırılmış hem de tekrar eden benzer karelerden kaçınılarak veri seti dengelenmiştir.

Frame çıkarımı için OpenCV kütüphanesi kullanılmış, .mp4, .mkv, .avi ve .mov formatındaki videolar desteklenmiştir. Her video işlenerek belirli aralıklarda görüntü alınmış ve JPEG formatında ayrı bir klasöre kaydedilmiştir. Bu görseller daha sonraki adımlarda iskelet çıkarımı (pose estimation) işlemi için temel veri olarak kullanılmıştır.

Bu yöntemle gölge boksu hareketlerinin zamansal yapısı parçalanmış ve görsel sınıflandırmaya uygun sabit formatta veri seti elde edilmiştir.

### 4. İskelet Verilerinin Elde Edilmesi (Pose Estimation)

Frame'lere ayrılan görüntüler üzerinden, her bir gölge boksu hareketinin insan vücudu üzerindeki temel noktaları çıkarılmış ve bu noktalardan oluşan iskelet yapıları oluşturulmuştur. Bu işlem için MediaPipe kütüphanesinin Pose modülü kullanılmıştır.

Ancak bu çalışmada, klasik pose çıkarımında elde edilen tüm noktalar yerine yalnızca hareketi anlamlandırmak için kritik olan noktalar dikkate alınmıştır. Bu sayede gereksiz verilerden kaçınılmış, veri setinin sınıflandırma için daha anlamlı hale gelmesi sağlanmıştır.

Kullanılan önemli vücut noktaları şunlardır: burun (kafa pozisyonu), iki omuz iki dirsek iki bilek iki kalça. Ayrıca, bu noktalar arasında hareket analizi için gerekli özel bağlantılar (örneğin kafa-omuz, omuz-dirsek, omuz-kalça gibi) tanımlanarak standart MediaPipe iskeletinden farklı olarak özelleştirilmiş bir iskelet sistemi oluşturulmuştur.

Pose çıkarımı sırasında vücut noktaları tespit edilemeyen (boş) görüntüler sistemden otomatik olarak ayıklanmış ve yalnızca net iskelet yapısı elde edilebilen görüntüler veri setine dahil edilmiştir. Bu adım, veri temizliğinin önemli bir parçası olmuştur.

Örneğin:



### 5. Görsel Verilerin Etiketlenmesi

İskelet görselleri oluşturulduktan sonra, bu görsellerin hangi hareket sınıfına ait olduğunu belirtmek amacıyla etiketleme işlemi gerçekleştirilmiştir. Bu adım, gölge boksu hareketlerinin sınıflandırılması için denetimli öğrenme (supervised learning) yaklaşımına uygun şekilde veri seti hazırlamak adına kritik önem taşımaktadır.

Veri etiketleme işlemi manuel olarak gerçekleştirilmiştir. Her bir iskelet görüntüsü kullanıcıya gösterilmiş ve klavye tuşları aracılığıyla doğru sınıfa taşınmıştır. Kullanıcıya kolaylık sağlamak amacıyla her sınıf için belirli bir tuş atanmış, örneğin q tuşu direkt vuruşu, w tuşu kroşeyi temsil etmektedir. Etiketleme sırasında ESC tuşu ile işlem sonlandırılabilen, geçersiz tuş basıldığında ise görsel atlanmaktadır.

Etiketleme sonucunda her sınıfa ait ayrı klasörler altında düzgün şekilde ayrılmış bir görsel veri seti elde edilmiştir. Bu yapılandırma, model eğitimi sürecinde sınıfların otomatik olarak tanınmasını ve dengeli biçimde işlenmesini sağlamaktadır.

### 6. Veri Kümesinin Eğitim ve Doğrulama İçin Bölünmesi

Etiketleme işlemi tamamlandıktan sonra, her sınıfa ait görüntüler ayrı klasörlerde olacak şekilde organize edilmiştir. Bu verilerin doğrudan model eğitiminde kullanılabilmesi için eğitim (train), doğrulama (validation) ve test (isteğe bağlı) olmak üzere bölünmesi gerekmektedir.

Bu projede, modelin doğruluğunu objektif şekilde değerlendirebilmek adına veri seti %80 eğitim ve %20 doğrulama oranında bölünmüştür. Böylece model, daha önce görmediği veriler üzerinde test edilerek aşırı öğrenme (overfitting) riski azaltılmıştır.

Bölme işlemi sırasında her sınıf kendi içinde dengeli biçimde ayrılmış; her hareket türünün eğitim ve doğrulama verileri aynı oransal dağılıma sahip olacak şekilde organize edilmiştir. Bu yapı, modelin tüm sınıfları adil şekilde öğrenmesini sağlamakta ve sınıf dengesizliğinden kaynaklanabilecek yanlışlıkların önüne geçmektedir.

Elde edilen bu yapı, PyTorch gibi kütüphanelerin ImageFolder yapısıyla doğrudan uyumludur ve model eğitim sürecine sorunsuz biçimde entegre edilmiştir.

### 7. Veri İşleme

Veriler işlendikten sonra sınıf başına veri adetleri şu şekildedir:

<b>Belirsiz</b>	train: 448	val: 112
<b>Aparkat</b>	train 368	val: 92
<b>Direkt</b>	train 332	val: 84
<b>Eğilme</b>	train 226	val: 56
<b>Kroşe</b>	train 382	val: 93
<b>Gard</b>	train 358	val: 90

### 8.1 Vision Transformer (ViT) Modeli

Bu çalışmada ilk olarak, Transformer mimarisini doğrudan görsel verilere uygulayan **Vision Transformer (ViT)** modeli kullanılmıştır. Model, timm kütüphanesi üzerinden vit\_base\_patch16\_224 yapılandırmasıyla çağırılmış ve önceden ImageNet üzerinde eğitilmiş ağırlıklar kullanılarak transfer öğrenme uygulanmıştır. Çıkış katmanı, projedeki 6 sınıfa göre yeniden yapılandırılmıştır.

#### Eğitim Süreci

Veri seti 32'lik mini-batch'lerle 50 epoch boyunca eğitim ve doğrulama adımlarına tabi tutulmuştur. Eğitimde Adam optimizasyon algoritması ve CrossEntropyLoss kayıp fonksiyonu kullanılmıştır. Eğitim süreci boyunca random horizontal flip ve rotation gibi basit artırma teknikleri uygulanmıştır.

- **Toplam eğitim süresi:** 1318.76 saniye
- **Tek görüntü için çıkarım süresi:** ~12.11 ms
- **En düşük doğrulama kaybı:** 0.3926
- **En düşük eğitim kaybı:** 0.0793

#### Performans Sonuçları

Classification Report:				
	precision	recall	f1-score	support
direct	0.84	0.92	0.88	84
gard	0.84	0.70	0.76	90
hook	0.84	0.89	0.86	93
slip	1.00	0.81	0.89	47
uncertain	0.79	0.89	0.84	112
uppercut	0.91	0.86	0.88	92
accuracy			0.85	518
macro avg	0.87	0.84	0.85	518
weighted avg	0.85	0.85	0.85	518

Modelin başarı düzeyini değerlendirmek için kullanılan sınıflandırma raporunda dört temel metrik yer almaktadır: **precision**, **recall**, **f1-score** ve **support**.

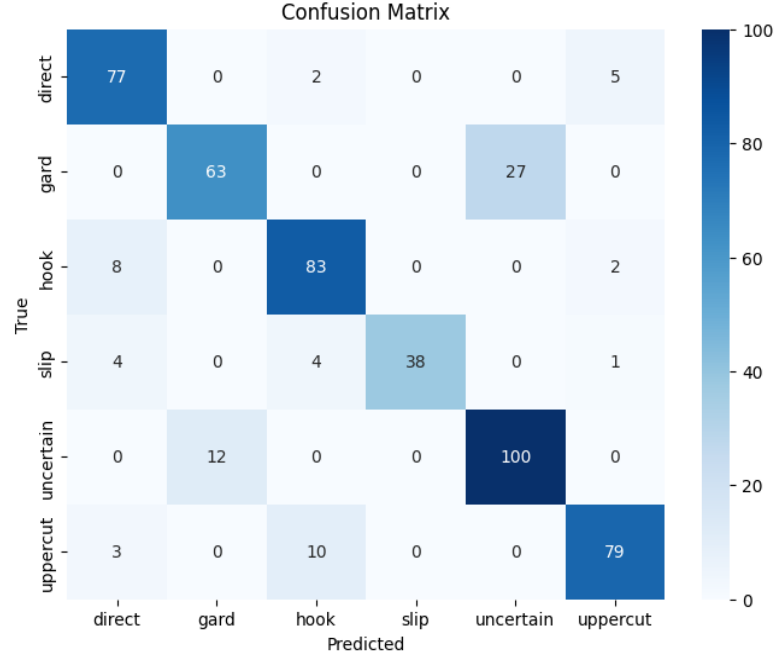
- **Precision (Kesinlik):** Modelin bir sınıfa ait olarak tahmin ettiği örneklerin gerçekten doğru olma oranıdır. Örneğin, model "uppercut" dediğinde gerçekten uppercut olma olasılığı.
- **Recall (Duyarlılık):** Gerçekten bir sınıfa ait olan verilerin model tarafından doğru bir şekilde tanınma oranıdır. Örneğin, tüm "slip" hareketlerinin kaç tanesinin doğru şekilde tespit edildiği.
- **F1-Score:** Precision ve recall değerlerinin harmonik ortalamasıdır. Bu metrik, özellikle veri dağılımı dengesiz olduğunda modelin genel başarımını daha adil bir şekilde gösterir.
- **Support:** Her bir sınıfa ait test verisi sayısını ifade eder. Modelin hangi sınıf üzerinde ne kadar veriyle test edildiğini gösterir.

Ayrıca tabloda:

- **Accuracy**, genel doğruluk oranını belirtir (modelin toplam başarı oranı).
- **Macro Avg**, her sınıfın metrik ortalamalarının eşit ağırlıklı ortalamasıdır.

- **Weighted Avg**, sınıfların veri sayısına göre ağırlıklandırılmış ortalamalarıdır.

Bu metrikler, modelin hangi sınıflarda daha başarılı olduğunu tespit etmeye ve genel performans analizine olanak sağlar.

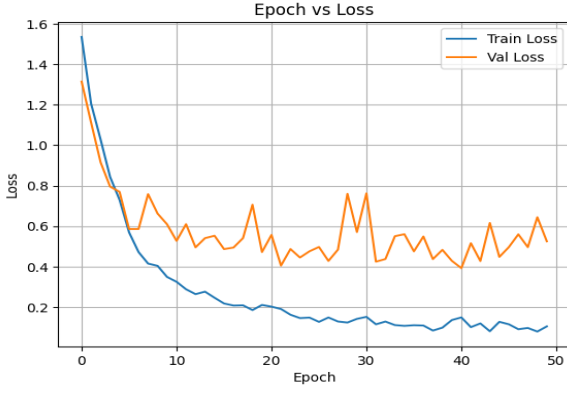


Modelin sınıflandırma performansını daha detaylı şekilde değerlendirmek amacıyla kullanılan confusion matrix, her sınıf için gerçek etiketlerle modelin tahminlerini karşılaştırır. Matrisin satırları **gerçek sınıfları**, sütunları ise **modelin tahmin ettiği sınıfları** temsil eder.

Bu matrise göre bazı önemli gözlemler şunlardır:

- **Direct** sınıfındaki 84 örneğin 77 tanesi doğru sınıflandırılmış, sadece birkaç örnek **hook** ve **uncertain** sınıflarıyla karıştırılmıştır.
- **Gard** sınıfında 63 doğru tahmin varken, 27 örnek model tarafından **uncertain** olarak hatalı sınıflandırılmıştır.
- **Hook** sınıfında 83 örnek doğru tahmin edilmiş, 8 örnek **direct**, 2 örnek ise **uppercut** olarak karıştırılmıştır.
- **Slip** sınıfında 38 doğru tahmin yapılmış, 4'er örnek **direct** ve **hook** olarak yanlış tahminlenmiştir.
- **Uncertain** sınıfı yüksek doğruluk göstermektedir: 112 örneğin 100'ü doğru tahminlenmiş, 12 örnek yanlışlıkla **gard** sınıfına atılmıştır.
- **Uppercut** sınıfında ise 79 doğru tahmine karşılık, 10 örnek **hook** ve 3 örnek **direct** olarak karıştırılmıştır.

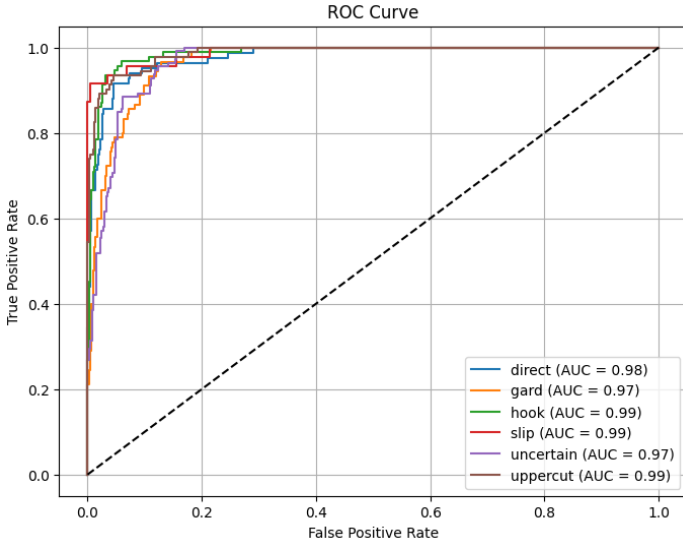
Bu grafik sayesinde modelin hangi sınıflar arasında daha çok hata yaptığını, hangi hareketleri birbirine karıştırdığını analiz etmek mümkündür. Özellikle gard ve uppercut gibi bazı sınıflar diğerleriyle karıştırılmaya daha yatkındır.



Yukarıdaki grafikte modelin eğitim sürecinde **eğitim (train) ve doğrulama (validation) loss değerlerinin** epoch (devir) bazında değişimi gösterilmiştir.

- **Mavi çizgi**, eğitim verisi üzerindeki hatayı (train loss) temsil eder. Eğitim süreci ilerledikçe bu hata düzenli şekilde düşmüş ve model veriye daha iyi uyum sağlamıştır.
- **Turuncu çizgi**, modelin doğrulama verisi üzerindeki hatasını (val loss) göstermektedir. Bu çizgi genellikle dalgalı bir yapı gösterse de genel olarak düşüş eğilimindedir.

Genel olarak, eğitim ve doğrulama hatalarının düşmesi modelin hem öğrenme hem de genelleme açısından başarılı olduğunu göstermektedir. Ancak doğrulama hatasında zaman zaman gözlemlenen dalgalanmalar, verideki çeşitliliğe veya sınıflar arası ayrım zorluklarına işaret edebilir. Yine de modelin overfitting (aşırı öğrenme) yapmadığı, tutarlı ve dengeli bir öğrenme süreci sergilediği görülmektedir.



Yukarıdaki grafik, modelin sınıflandırma başarısını sınıf bazında değerlendirmek amacıyla oluşturulan **ROC eğrilerini** (Receiver Operating Characteristic Curve) göstermektedir. Her bir renkli çizgi, bir sınıfa ait ROC eğrisini temsil eder.

- ROC eğrisi, modelin pozitif sınıfları ne kadar iyi ayırt edebildiğini gösterir.
- Eğrinin altında kalan alan **AUC (Area Under Curve)** olarak adlandırılır ve 1'e yaklaştıkça modelin sınıfı doğru ayırt etme yeteneği artar.
- Bu çalışmada:
  - **direct**: AUC = 0.98
  - **gard**: AUC = 0.97
  - **hook**: AUC = 0.99
  - **slip**: AUC = 0.99
  - **uncertain**: AUC = 0.97
  - **uppercut**: AUC = 0.99

AUC değerlerinin tüm sınıflarda **0.97 ve üzeri** olması, modelin sınıfları yüksek doğrulukla ayırdığını ve genel performansının oldukça başarılı olduğunu göstermektedir. ROC eğrisinin köşeye yakın seyretmesi, yanlış pozitif oranının düşük, doğru pozitif oranının yüksek olduğunu ifade eder. Bu da modelin karar eşiklerinde güçlü bir ayrım yapabildiğini gösterir.

## 8.2 Data-efficient Image Transformer (DeiT) Modeli

Bu çalışmada ViT modeline alternatif olarak, görsel veriler üzerinde daha az veriyle daha etkili eğitim yapabilme kapasitesine sahip olan **DeiT (Data-efficient Image Transformer)** modeli kullanılmıştır. DeiT modeli, Vision Transformer (ViT) yapısına dayalı olmakla birlikte, özel bilgi verici (distillation) token'ı ve veri artırımı stratejileri ile daha düşük veri miktarlarında bile yüksek performans gösterebilecek şekilde optimize edilmiştir.

Model eğitimi sırasında daha önce hazırlanan iskelet görüntüleri kullanılmış, giriş boyutu olarak 224x224 piksel çözünürlüğünde normalize edilmiş görüntüler tercih edilmiştir. Eğitimde kullanılan parametreler, ViT modeline benzer şekilde iskelet optimizasyon algoritması ve CrossEntropyLoss fonksiyonu ile yapılandırılmıştır. Model, 50 epoch boyunca eğitim ve doğrulama adımlarından geçirilmiştir.

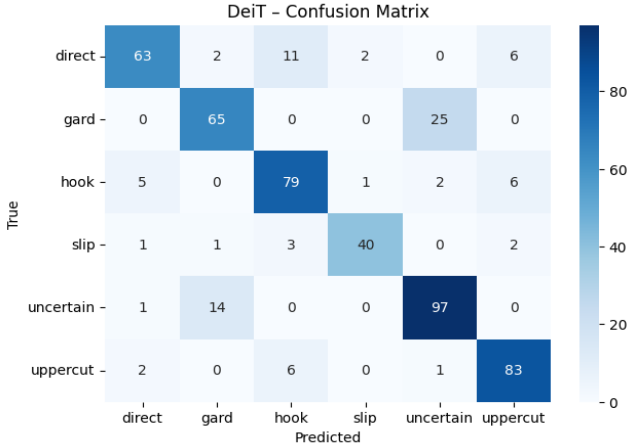
DeiT modeli, sınıflar arası ayrımda yüksek doğruluk sağlamış ve ROC, F1-score, precision-recall gibi temel sınıflandırma metriklerinde güçlü sonuçlar vermiştir. Özellikle sınıflar arası belirsizliğin yoğun olduğu durumlarda dahi DeiT modelinin genel başarımı dikkat çekicidir.

Modelin çıktılarına ait grafiksel sonuçlar (Confusion Matrix, ROC Curve, Loss Graph vb.) aşağıdaki bölümlerde verilmiştir.

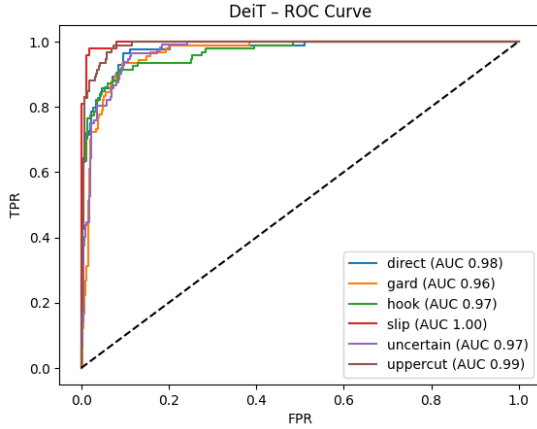
### Sınıflandırma Çıktısı:

Classification Report:				
	precision	recall	f1-score	support
direct	0.88	0.75	0.81	84
gard	0.79	0.72	0.76	90
hook	0.80	0.85	0.82	93
slip	0.93	0.85	0.89	47
uncertain	0.78	0.87	0.82	112
uppercut	0.86	0.90	0.88	92
accuracy			0.82	518
macro avg	0.84	0.82	0.83	518
weighted avg	0.83	0.82	0.82	518

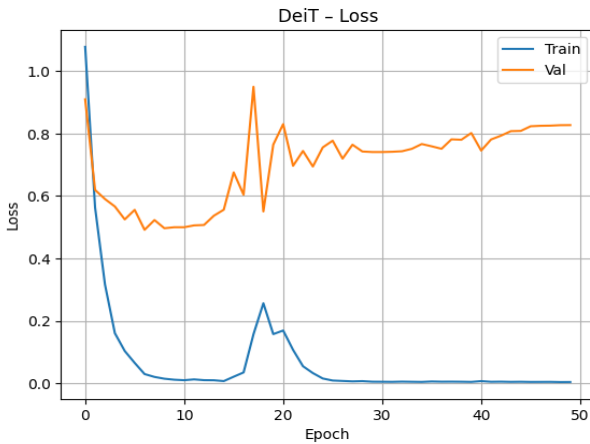
### Confusion Matrix:



### Roc Curve Grafiği:



### Loss Grafiği:



### 8.3 Swin Modeli

Bu çalışmada geleneksel ViT mimarisinden farklı olarak, görsel veriyi hiyerarşik pencereler (sliding window) üzerinden analiz eden Swin Transformer modeli kullanılmıştır. Swin Transformer, görüntüyü küçük parçalar (patch'ler) halinde analiz etmekle kalmayıp, bu parçalar arasındaki yerel ilişkileri çok katmanlı biçimde modelleyerek daha detaylı ve konuma duyarlı bir öğrenme sağlar.

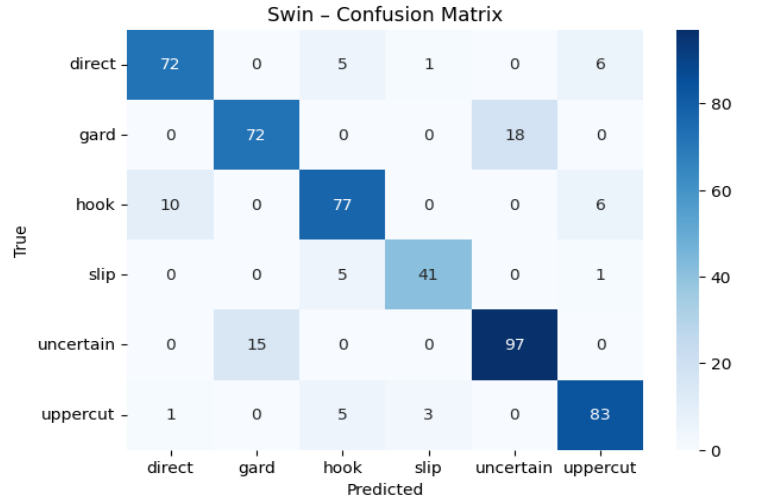
Model, daha önceki aşamalarda hazırlanmış iskelet temelli görüntülerle eğitilmiş, giriş boyutu olarak 224x224 normalize edilmiş görseller kullanılmıştır. Eğitimde ViT ve DeiT modellerindeki gibi aynı veri kümesi ve sınıf yapısı korunarak, sonuçların doğrudan karşılaştırılabilir olması sağlanmıştır. Eğitim süreci yine 50 epoch boyunca yürütülmüş, optimizasyon için Adam algoritması ve CrossEntropyLoss fonksiyonu tercih edilmiştir.

Swin Transformer modeli özellikle detayları koruyan yapısı sayesinde gard ve hook gibi birbirine benzer hareketler arasındaki farkları daha isabetli şekilde ayırt edebilmiştir. Bu durum, confusion matrix ve ROC eğrisi analizlerinde net biçimde gözlenmektedir.

### Sınıflandırma Çıktısı:

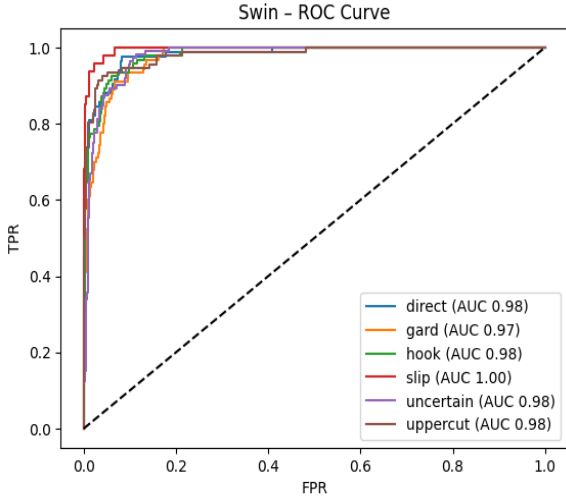
Classification Report:				
	precision	recall	f1-score	support
direct	0.87	0.86	0.86	84
gard	0.83	0.80	0.81	90
hook	0.84	0.83	0.83	93
slip	0.91	0.87	0.89	47
uncertain	0.84	0.87	0.85	112
uppercut	0.86	0.90	0.88	92
accuracy			0.85	518
macro avg	0.86	0.85	0.86	518
weighted avg	0.85	0.85	0.85	518

### Confusion Matrix:

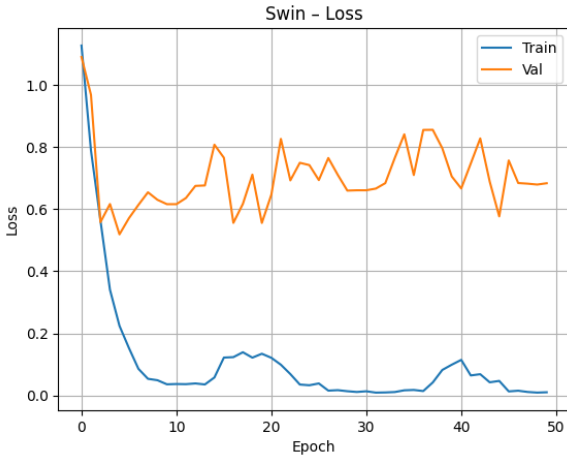




### Roc Curve Grafiği:



### Loss Grafiği:



### 8.4 Beit Modeli

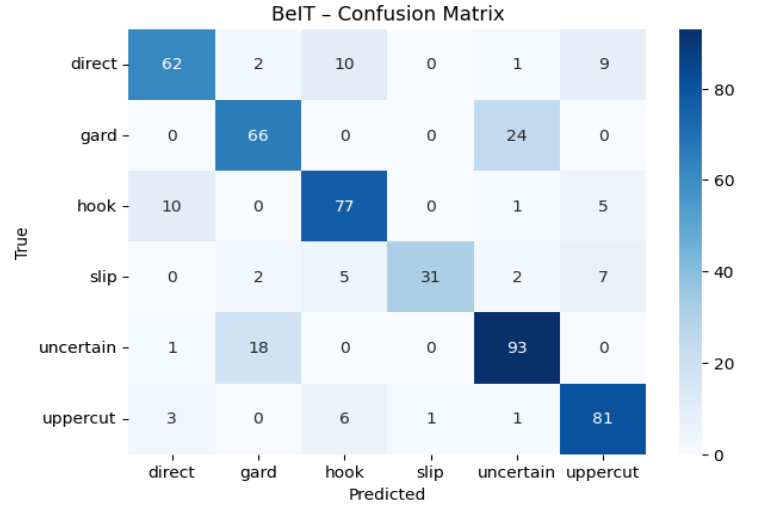
Bu modelde, önceden maskelenmiş görüntü modelleme stratejisiyle eğitilmiş olan BEiT (Bidirectional Encoder representation from Image Transformers) modeli kullanılarak gölge boksu hareketlerinin sınıflandırılması gerçekleştirilmiştir. BEiT, BERT tarzı bir ön eğitim yaklaşımını görüntü verisine uygulayarak görsel temsilleri daha güçlü şekilde öğrenmeyi amaçlamaktadır.

Modelin eğitimi için, daha önce iskelet çizimleriyle oluşturulan görseller, 224x224 boyutuna yeniden ölçeklendirilip normalize edilerek giriş verilmiştir. Kullanılan sınıflar ViT, DeiT ve Swin modelleriyle birebir aynı olacak şekilde korunmuş, böylece modeller arası karşılaştırma sağlıklı hale getirilmiştir. Eğitim süreci 50 epoch boyunca yürütülmüş, Adam optimizyer ve CrossEntropyLoss ile öğrenme işlemi gerçekleştirilmiştir.

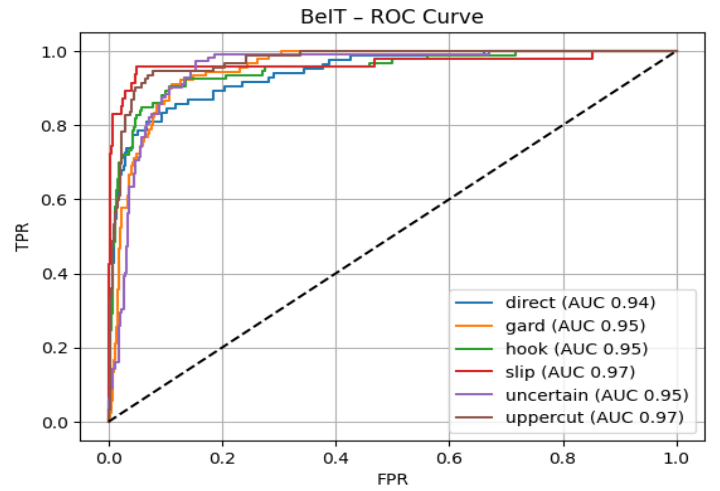
### Sınıflandırma Çıktısı:

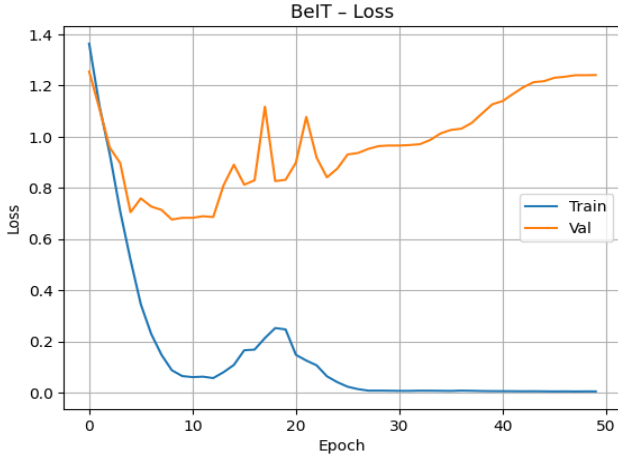
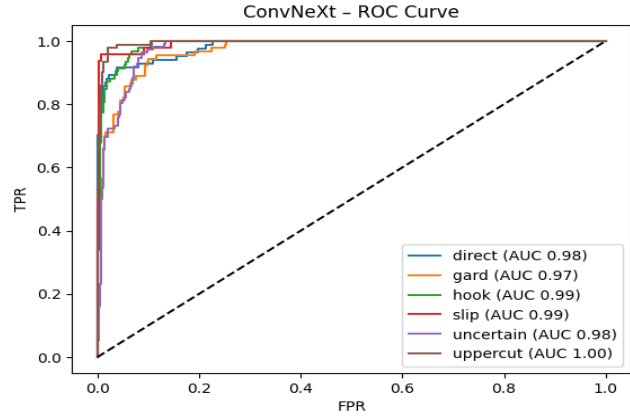
	precision	recall	f1-score	support
direct	0.82	0.74	0.78	84
gard	0.75	0.73	0.74	90
hook	0.79	0.83	0.81	93
slip	0.97	0.66	0.78	47
uncertain	0.76	0.83	0.79	112
uppercut	0.79	0.88	0.84	92
accuracy			0.79	518
macro avg	0.81	0.78	0.79	518
weighted avg	0.80	0.79	0.79	518

### Confusion Matrix:



### Roc Curve Grafiği:

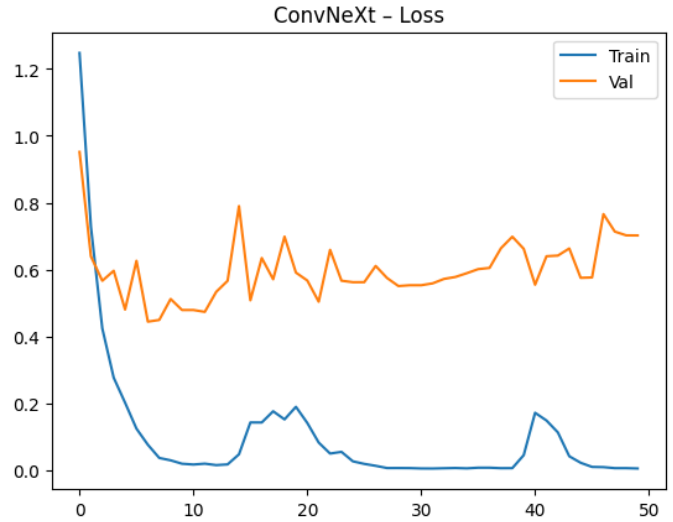


**Loss Grafiği:****Roc Curve Grafiği:**

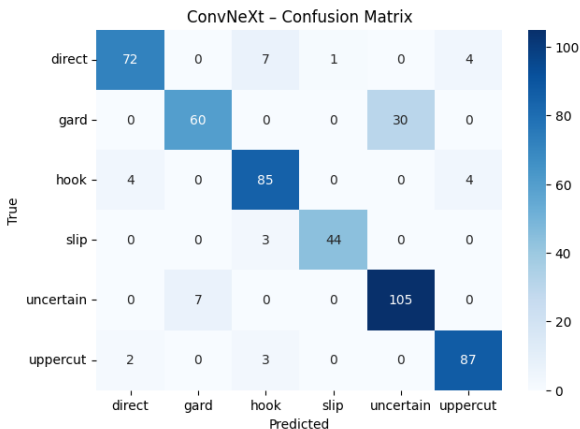
## 8.5 ConvNeXtModeli

Bu çalışmada, modern CNN mimarilerinden biri olan ConvNeXt modeli ile gölge boksu hareket sınıflandırması gerçekleştirilmiştir. ConvNeXt, klasik konvolüsyonel sinir ağlarının (CNN) gücünü korurken, Vision Transformer (ViT) mimarisinden ilham alarak daha derin, geniş ve verimli bir yapı sunar. Bu yönüyle, hem ViT tabanlı modellerle hem de geleneksel CNN yapılarıyla başarılı şekilde rekabet edebilmektedir.

Modelin eğitimi sırasında, daha önce çizilmiş iskelet temsillerinden oluşan görüntüler 224x224 boyutuna yeniden boyutlandırılarak normalize edilmiştir. Eğitim, daha önceki modellerde olduğu gibi 50 epoch boyunca sürdürülmüş, eğitim ve doğrulama setleri aynı tutulmuştur. Adam optimizasyon algoritması ve CrossEntropyLoss kullanılarak model optimize edilmiştir.

**Loss Grafiği****Sınıflandırma Çıktısı:**

	precision	recall	f1-score	support
direct	0.92	0.86	0.89	84
gard	0.90	0.67	0.76	90
hook	0.87	0.91	0.89	93
slip	0.98	0.94	0.96	47
uncertain	0.78	0.94	0.85	112
uppercut	0.92	0.95	0.93	92
accuracy			0.87	518
macro avg	0.89	0.88	0.88	518
weighted avg	0.88	0.87	0.87	518

**Confusion Matrix:**

## 9.Linkler ve Kaynakça

### Github:

<https://github.com/hakan8755/golgeBoksuHareketSiniflandirmasi>

### Drive:

<https://drive.google.com/drive/folders/154nXITtVANFSXy5MiLkt6crkPWz-nB4I?usp=sharing>

## Kaynaklar

ChatGpt 4.5

<https://medium.com/@RobuRishabh/vision-transformer-vit-39f627d04b2a>

<https://sh-tsang.medium.com/review-deit-data-efficient-image-transformer-b5b6ee5357d0>

<https://www.geeksforgeeks.org/computer-vision/swin-transformer/>

<https://sh-tsang.medium.com/review-beit-bert-pre-training-of-image-transformers-c14a7ef7e295>

<https://www.kungfu.ai/blog-post/convnext-a-transformer-inspired-cnn-architecture>