

# KAVGA TESPİTİ

Berkant Şimşek  
22040101038

berkantsimsek@stu.topkapi.edu.tr

İlkay Gökbudak  
22040101047

ilkaygokbudak@stu.topkapi.edu.tr

Ömer Avcı  
22040101024

omeravci@stu.topkapi.edu.tr

## I. ÖZET

Bu projede, videolardan veya anlık görüntülerden kavga anlarının tespit edilmesi amacıyla derin öğrenme teknikleri kullanılmaktadır. Kavga tespiti sistemi, güvenlik kameralarından alınan görüntüler veya videolar üzerinde anormal davranışları (örneğin, kavga, şiddet) tespit etmek için geliştirilmiştir. Bu tespitler, güvenlik güçlerine veya diğer otoritelere erken uyarılar sağlayabilir. Böylelikle önceden tedbir alarak suçların bir nebze de olsa önüne geçmeyi amaçlamaktayız.

## II. GİRİŞ

Kavga tespiti, gözetleme sistemleri için önemli bir konudur. Ancak, gözetleme videolarında yüksek performansla kavga tespit edebilen bir algoritma oluşturmada çok az başarı elde edilmiştir. Bu çalışmada, gözetleme videolarında kavga tespiti görevi için yeni bir yöntem kullandık. Sistem canlı görüntüyü alıp çeşitli algoritmalarından geçirip bize tehlikeli bir durum olup olmadığının sonuçları veriyor. Bu sonuç doğrusunda karar veriyor.

## III. YÖNTEMLER

### .Veri Toplama ve Hazırlık

Sonuçlarımızı sınıflandırmak için 2 ayrı sınıf oluşturduk bunlar 'fight' ve 'no fight' olmak üzere. 'fight' -> 1 olarak sembolize edilirken 'nofight' -> 0 olarak sembolize edilmektedir. Ayriyeten canlı olarak modelimiz ile birlikte bir sonuç çıkarıp o sonuca göre de bir yargıya ulaşabilmekteyiz. Github'dan veri setlerini aldık [1]. Bu veri setinde yaklaşık 75 tanesi savaş içerikli video, 75 tanesi de savaş içermeyen video olmak üzere toplamda yaklaşık 150 tane veri kullanarak modelimizi eğittik. Boyutları farklı olan verilerimizde modelimizde daha net ve doğru sonuçlar elde edebilmek için 224x224 boyutlarına çevrildi.

### .Veri Ön İşleme

data\_x\_ext.npy: Modelin girdi özelliklerini içerir. (Özellik haritaları veya zaman serisi veri)

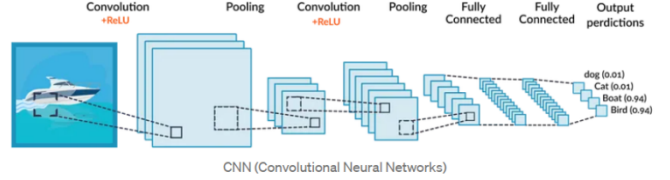
data\_y.npy: Modelin hedef kitlelerini içerir. (Çıkış etiketleri: 'fight' veya 'noFight'). Veri, np.load ile numpy dizileri olarak yüklenir.

Transpose işlemi: x\_train üzerindeki her örnek için zaman serisi transpoz ediliyor. Bu, her örneğin T zaman adımı ve F özellik boyutlarının F zaman adımı ve T özellik boyutlarına dönüştürülmesi anlamına gelir.

Sonuç olarak modelin girdi şekli (40,2048) olur. 40 zaman adımı iken her adımda 2048 özellik bulunmaktadır.

### .Modelin Tanımlanması

#### CNN MODELİ:



[7]

#### ResNet152:

ResNet152, derin öğrenme araştırmalarında yaygın olarak kullanılan ve çok güçlü bir Convolutional Neural Network (CNN) modelidir. ResNet, özellikle derin ağlarda karşılaşılan vanishing gradient (kaybolan gradyan) problemini çözmek için tasarlanmış bir yapıdır ve residual connections (artık bağlantılar) kullanarak daha derin ağlar inşa edebilmek için kullanılır. Bu model, ImageNet veri seti üzerinde önceden eğitilmiş olup, görüntü sınıflandırma görevlerinde oldukça başarılıdır.

#### Model Yapısı

ve

#### Kullanımı:

Kodda, ResNet152 modelinin üst katmanları kaldırılmış (include\_top=False), yani sadece görüntülerin özelliklerini (feature) çıkarmak için kullanılan temel katmanlar ve bloklar kullanılmıştır. Model, giriş olarak verilen görüntüleri işleyip, bunlardan öznitelik vektörleri (feature vectors) çıkarır. Çıkarılan özellikler, daha sonra bir RNN modeline (decoder) aktarılmaktadır.

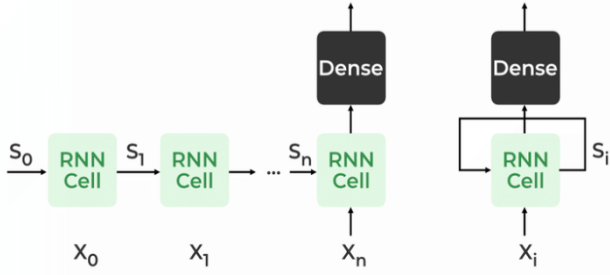
#### Özellik

#### Çıkartımı:

Görüntü, önce cv2.resize fonksiyonu ile 224x224 boyutlarına getirilir ve sonra preprocess\_input fonksiyonu ile uygun bir şekilde ön işleme tabi tutulur. Bu işlem, ResNet modeline uygun giriş formatına dönüştürme işlemini gerçekleştirir. Bu görüntü, daha sonra ResNet152'nin katmanlarından geçirilir ve encoder\_model (ResNet152) tarafından özellik çıkarımı yapılır. Bu özellikler, 40 karelik bir pencere üzerinden sıralı olarak bir RNN modeline (decoder) aktarılır.

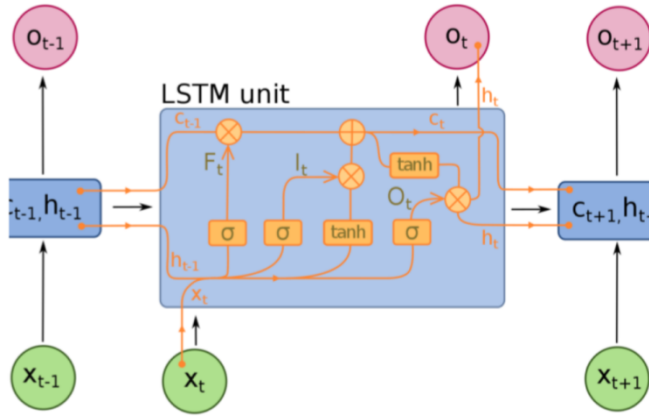
#### RNN MODELİ:

## RECURRENT NEURAL NETWORKS



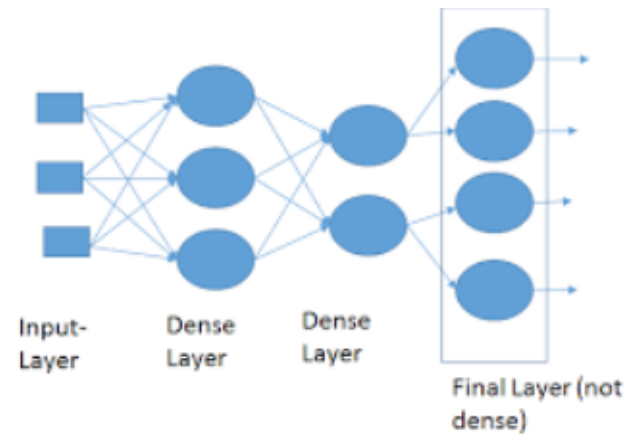
[5]Giriş Katmanı: 40 zaman adımına sahip ve her adımda 2048 özellik olan bir giriş şekli kabul edilir.

LSTM Katmanları: İki adet LSTM katmanı kullanılır: İlk LSTM katmanı 1024 birimle, çıkışı zaman dizisi şeklinde (return\_sequences=True) verir. İkinci LSTM katmanı 512 birimle, çıktısı yalnızca son zaman adımında verir (return\_sequences=False).



[4]

Dense ve Batch Normalization Katmanları: Sonrasında relu aktivasyon fonksiyonu kullanan iki Dense katmanı gelir. Aralarındaki Batch Normalization, öğrenmeyi stabilize



etmek için kullanılır.

[6]

Çıkış Katmanı: Sigmoid aktivasyon fonksiyonu ile, son sınıflandırma yapısı (0 veya 1, yani kavg yok veya kavg).

Adam Optimizasyonu: Modelin eğitiminde kullanılan optimizasyon algoritması olarak Adam seçilmiştir:

CNN ve RNN Kombinasyonu:

CNN (ResNet152) görüntülerden uzun vadeli bağımlılıkları ve özellikleri çıkarırken, RNN (Decoder kısmı) bu özellikleri zaman içinde işleyerek sıralı veriyi anlamaya çalışır. Bu kombinasyon, özellikle video analizi gibi sıralı veri işleme görevlerinde yaygın olarak kullanılır.

.Modelin Eğitilmesi

Modelin Eğitilmesi: Eğitim verileri (x\_train ve y\_train) kullanılarak modelin eğitimine başlanır.

Epoch Sayısı: Eğitim 50 epoch boyunca yapılır.

Batch Size: Her iterasyonda 32 örnek ile eğitim yapılır.

Validation Split: Eğitim verilerinin %20'si doğrulama seti olarak ayrılır.

Callback: Eğitim sırasında öğrenme oranını azaltmak için

ReduceLROnPlateau callback kullanılır (eğer doğrulama kaybı 5 epoch boyunca iyileşmezse, öğrenme oranı azaltılır.).

.Modelin Kaydedilmesi

- **train.py** dosyasındaki model dosyasındaki model eğitildikten sonra `model.save()` ile kaydedilir.
- **main.py** dosyasında ise bu kaydedilen model `load_model()` ile yüklenir ve canlı video akışında tahminler yapılır.
- **handle\_features.py** dosyasında, video karelerinden çıkarılan özellikler `np.save()` ile kaydedilir ve daha sonra eğilmek için kullanılır.

.Modelin Değerlendirilmesi

Doğruluk (Accuracy): Model, doğruluk ve kayıp (loss) metrikleri ile değerlendirilmiştir.

Eğitim Değerlendirmesi: Eğitim ve doğrulama doğruluğu/kayıp grafiklerinin çizilmesi, modelin zaman içindeki yakınsamasını ve performansını görsel olarak sunuyor.

Olası

Performans

Sorunları:

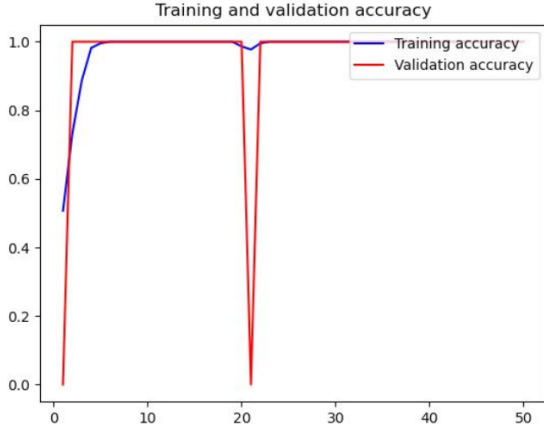
Sınıf Dengesizliği: Eğer veri setinizde dövüş olmayan sınıfların sayısı dövüş sınıflarından fazla ise, sınıf ağırlıkları veya aşırı örnekleme (oversampling) gibi teknikler kullanmak faydalı olabilir.

Aşırı Öğrenme (Overfitting): Modelin eğitim setine aşırı

öğrenme yapmadığından emin olun. Bunun için düzenleme stratejileri faydalı olabilir.

### .Modelde Eğitim Sonuçlarının Görselleştirilmesi

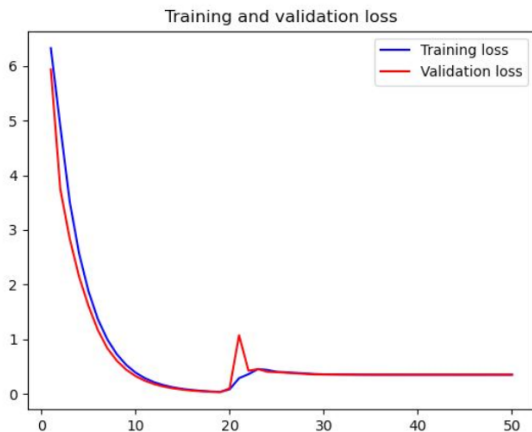
Grafik2:Eğitim ve Doğrulama Kaybı



Grafik1:eğitim ve doğrulama doğruluğu

Grafikte, eğitim mavi ve doğrulama kırmızı renkte gösterilmektedir değerler. zamanla nasıl değiştiği gösterilmektedir.

1. **Yüksek Doğruluk:** Hem eğitim hem de doğrulama doğruluğu 1.0 (yüzde yüz) seviyesine çok yakın veya tam olarak oraya ulaşmış. Bu, modelin verilere oldukça iyi uyum sağladığını gösteriyor.
2. **Aşırı Uyum (Overfitting):** Eğitim ve doğrulama doğrulukları arasında belirgin bir fark yok, ancak doğrulama doğruluğu belirli bir noktada düşüş gösteriyor. Bu, modelin bazı veriler üzerinde aşırı uyum sağladığını işaret eder.
3. **Dengeli Performans:** İki eğilimin birbirine çok yakın olması, modelin genel performansının tutarlı olduğunu ve test verisi üzerinde de iyi sonuçlar verebileceğini düşündürür.



Grafik, eğitimin mavi ve doğrulama kaybının kırmızı renkte gösterilmektedir. Grafikte zamanla verilerin nasıl değiştiğini göstermektedir.

1. **Düşen Kayıp Değerleri:** Her iki kayıp eğrisi de başlangıçta yüksek değerlerden başlayıp zamanla önemli ölçüde düşmüştür. Bu modelin öğrenme sürecinin etkili olduğunu ve zamanla performansının arttığını göstermektedir.
2. **Çizgilerin Yakınlığı:** Eğitim ve doğrulama kayıpları birbirine oldukça yakın seyrettiği için modelin hem eğitim hem de doğrulama setleri üzerinde tutarlı bir performans sergilediği söylenebilir.
3. **Düşük Kayıp ve Genelleme:** Kayıp değerlerinin oldukça düşük seviyelerde kalmış olması, modelin genelleme kabiliyetinin iyi olduğunu ve test verisi üzerinde de başarılı performans gösterebileceğini yorumlayabiliriz.
4. **Ani Dalgalanmalar:** 20. iterasyondan sonra kayıplarda dikkat çeken küçük bir dalgalanma gözlemlenmektedir. Bu öğrenme sürecinin henüz tam olarak stabil olmadığını göstermektedir. Bu tür dalgalanmalar, öğrenme oranının çok yüksek olmasından veya başka parametrelerin iyi ayarlanmaması nedeniyle olabilir.

### IV. SONUÇ:

Bu projede, videolardan veya anlık görüntülerden kavga anlarını tespit etmek amacıyla derin öğrenme teknikleri kullanıldı. Özellikle, ResNet152 ve LSTM tabanlı bir model tasarlandı ve bu model ile güvenlik kameralarından alınan görüntülerdeki kavga ve şiddet gibi anormal davranışların tespiti hedeflendi. Veriler, 'fight' ve 'no fight' olmak üzere iki sınıfa ayrılarak modelin eğitimine sunuldu. Bu sınıflandırma ile güvenlik güçlerine erken uyarılar sağlanarak suçların önüne geçilmesi amaçlanmıştır.

Kullanılan yöntemlerin başarıyla entegre edilmesi sonucunda model, videolar üzerinde kavga tespiti yapabilme konusunda önemli bir başarı göstermiştir. Özellikle CNN ve RNN yapılarını birleştirerek, modelin hem görüntüden özellik çıkarımı yapabilmesi hem de bu özellikleri zaman boyutunda işleyebilmesi sağlanmıştır. Eğitim süreci boyunca elde edilen doğruluk ve kayıp grafikleri, modelin iyi bir şekilde eğitim aldığını ve genelleme yeteneğinin güçlü olduğunu göstermektedir.

Modelin sonuçları, erken uyarı sistemleri ve güvenlik uygulamaları için potansiyel olarak büyük bir katkı sağlayabilir. Ayrıca, eğitim sırasında karşılaşılan aşırı öğrenme (overfitting) ve sınıf dengesizliği gibi sorunlar, çeşitli tekniklerle minimize edilmiştir.

Sonuç olarak, geliştirilen kavga tespiti sistemi, güvenlik alanında önemli bir uygulama olma potansiyeline sahiptir ve bu teknolojinin daha geniş ölçekli güvenlik sistemlerinde uygulanabilirliği araştırılabilir. Gelecekte, daha büyük veri setleriyle modelin doğruluğu artırılabilir ve daha kapsamlı analizler yapılabilir.

#### REFERENCES

- [1] Seymanur Aktı, "Dataset title if known," GitHub, [Online]. Available: <https://github.com/seymanurakti/dataset-repository>. [Accessed: Jan. 16, 2025]. . (references)
- [2] Bharati, Puja, and Ankita Pramanik. "Deep learning techniques—R-CNN to mask R-CNN: a survey." Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019 (2020): 657-668.
- [3] Sherstinsky, Alex. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." Physica D: Nonlinear Phenomena 404 (2020): 132306.
- [4] "Yinelenen Sinir Ağları Nedir?" *Datakapital Blog*, [Online]. Available: <https://datakapital.com/blog/yinelenen-sinir-aglari-nedir>. [Accessed: Jan. 16, 2025].
- [5] "Recurrent Neural Networks," *GeeksforGeeks*, [Online]. Available: [https://media.geeksforgeeks.org/wp-content/uploads/20231204131544/recurrent\\_neural\\_networks-768.png](https://media.geeksforgeeks.org/wp-content/uploads/20231204131544/recurrent_neural_networks-768.png). [Accessed: Jan. 16, 2025].
- [6] "Working of Dense Layer," *Data Science Stack Exchange*, [Online]. Available: <https://datascience.stackexchange.com/questions/103183/working-of-dense-layer>. [Accessed: Jan. 16, 2025].
- [7] "CNN (Convolutional Neural Networks) Nedir?" *Medium*, [Online]. Available: <https://bartubozkurt35.medium.com/cnn-convolutional-neural-networks-nedir-a5bafc4a82a1>. [Accessed: Jan. 16, 2025].



İlkyay GÖKBUDAK, C, C#, Java, Javascript, Python dilleri üstünde çalışmıştır. TCM bilişim şirketinde staj yapmıştır. Önemli projeleri arasında JavaScript ile 3D Araba yarışı, Java ile fizik simülasyonu, Java Swing ile Depo kayıt sistemi ve Nextjs ile E-Ticaret web sitesi yer alır.



Berkant ŞİMŞEK;  
C, C++,html,Python,PHP, Java ve sql dilleri üzerinde yoğunlaşmış ve kendni geliştirmiştir. Önemli projeleri arasında C ile uçak kayıt sistemi, Java ile Otel rezervasyon sistemi ve JavaScript ,PHP ile Takı sitesi,X-O Game projeleri yer alır.



Ömer AVCI;  
C, C++, Java,JS, PHP(Laravel) ve Python dillerinde kendini geliştirmiştir.Grandoria Turizm ve Airport.Online şirketinde IT Intern olarak çalışmakta. Önemli projeleri arasında Kişisel blog sitesi, Turizm Acentası Web Sitesi ve C ile uçak kayıt sistemi yer alır