

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BASKETBOL OYUNCU TANIMLAMA VE SKOR TESPİTİ

YÜKSEK LİSANS TEZİ

Osman Murat TEKET

Elektrik ve Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. İmam Şamil YETİK

NİSAN 2021

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış, ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Osman Murat Teket

ÖZET

Yüksek Lisans Tezi

BASKETBOL OYUNCU TANIMLAMA VE SKOR TESPİTİ

Osman Murat Teket

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik ve Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. İmam Şamil Yetik

Tarih: Nisan 2021

Spor video analizi, görüntü işleme uygulamalarının popüler kullanım alanlarından biridir. Bu tür analizler profesyonel spor müsabakalarında ya da antrenman videolarında yapılmakta ve genel olarak skor ve oyuncu tanımlamaları için derin öğrenme yerine geleneksel yöntemler kullanılmaktadır. Bu tezin amacı ise bir basketbol antrenmanını gerçek zamanlı olarak analiz edebilen bir algoritma geliştirmektir. Yani, bu uygulama doğru oyuncuları tespit etmeli ve her bir oyuncu için skoru vermelidir. Bu amaçla, ilk olarak şutları ve bu şutların başarılı olup olmadığını tespit eden bir metod tanıtılmıştır. Spesifik olarak, basketbol potasının görüntüdeki yerini tespit etmek için bir derin öğrenme modeli verilmiş ve şut tespiti için arka plan çıkarımı kullanılmıştır. Ardından, şutun skor olup olmadığı farklı bir sinir ağıyla tanımlanmıştır. Şutun tanımlanmasından sonra, arabelleklene görüntüler içinde geriye gidilerek şutör tespiti yapılır. Oyuncu YOLOv3-tiny kullanılarak tespit edilir ve güdümsüz bir derin öğrenme modeliyle tanımlanır. Bu güdümsüz model yüksek performans vermesi için Üçlü Yitim ve bir bölütleme modelinin kullanımıyla eğitilmiştir. Gerçek zaman hedefine ulaşmak amacıyla, kullanılan bütün modeller için temel olarak mobil ağlardan farklı parametre ve eğitim yöntemleriyle yararlanılmıştır. Ek olarak, her seans sonunda bir kümeleme uygulaması çalıştırılarak tekrar oyuncu tanımlama yapılmış, ve sonucun iyileştirilmesi sağlanmıştır. Deneyler iki farklı antrenman videosu üzerinde yapılmıştır. Deney sonuçları skor tanımlamasında %95 doğruluk oranı; 2 oyuncu üzerinde gerçek zamanlı

olarak %92.5'e ve kümeleme ile 6 oyuncu üzerinde %86.7'ye varan doğruluk oranları ile tanıtılan metodun etkililiğini göstermektedir.

Anahtar Kelimeler: Oyuncu tanımlama, Skor tespiti, Derin öğrenme



ABSTRACT

Master of Science

PLAYER IDENTIFICATION WITH SCORING DETECTION

Osman Murat Teket

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. İmam Şamil Yetik

Date: April 2021

Sports video analysis is one of the most popular areas for image processing. These analyses are done on professional sports matches or training videos and generally employ traditional image processing methods instead of deep learning for scoring and player identification. The aim of this thesis is to obtain an algorithm to analyze a basketball training in real time. That is, the approach should be able to detect the correct players and scores for each player. For this purpose, we first propose a method to detect the shots and if they are makes or misses. Specifically, we use a deep learning model to detect the position of the hoop and background subtraction to detect the shot. Then, scoring is determined by another neural network using classification. After the determination of the shot, we go back in our buffered images to find the player who sent the shot. The player is detected by YOLOv3-tiny and identified by a deep unsupervised re-identification model. This unsupervised model is trained with Triplet Loss with the addition of a segmentation model to obtain high accuracy. To keep the real-time aim, all models employ mobile networks as base with different parameters and training methods. After each session, we run clustering to do the player re-identification again to improve on the result. The experiments were conducted on two training videos. The experiment results show the effectiveness of the current method with 95% accuracy on scoring identification and up to 92.5% overall accuracy on re-identification of 2 players in real-time, with up to 86.7% accuracy on 6 players with clustering.

Keywords: Player identification, Scoring detection, Deep learning

TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren hocam Doç. Dr. İmam Őamil YETİK'e, kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Elektrik Elektronik Mühendislięi Bölümü öğretim üyelerine ve destekleriyle her zaman yanımda olan aileme ve arkadaşlarıma, veri setiy ve teknik yardımlarından dolayı Rapsodo ve çalışma arkadaşlarıma teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	iv
ABSTRACT	vi
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ŞEKİL LİSTESİ	xi
ÇİZELGE LİSTESİ	xiii
KISALTMALAR	xiv
1. GİRİŞ	1
1.1 Literatür Aras tırması	2
1.1.1 Atıs, Tanımlama	2
1.1.2 Oyuncu Tanımlama	3
2. METODOLOJİ	7
2.1 Derin Öğrenme	7
2.1.1 Kullanılan Derin Öğrenme Modelleri	8
2.2 Veri Seti	11
2.2.1 Stil Transferi	11
2.2.2 Hareket Bulanıklığı Ekleme	13
2.2.3 Dolgulama	13
2.3 Atıs, Sınıflandırması.....	14
2.3.1 Atıs, Alanı Tespiti	15
2.3.2 Atıs, Tespiti	15
2.3.3 Skor Tanımlama	17
2.4 Oyuncu Tanımlama	18
2.4.1 Yitim Fonksiyonu	18
2.4.2 Arka Plan Silme	19
2.4.3 Şutör Tespiti	20
2.4.4 Oyuncu Tanımlama	20
2.5 Kullanılan Başarım Metriği	22
3. DENEY	25
3.1 Veri Seti	25
3.1.1 Atıs, Alanı Tespiti	25
3.1.2 Atıs, Tespiti	25
3.1.3 Skor Tanımlama	25
3.1.4 Oyuncu Tanımlama	26
3.2 Atıs, Sınıflandırması.....	27
3.2.1 Atıs, Alanı Tespiti	27

3.2.2 Atıs, Tespiti.....	27
3.2.3 Atıs, Sınıflandırma	27
3.3 Oyuncu Tanımlama	28
4. SONUÇ.....	35
KAYNAKLAR.....	36
ÖZGEÇMİŞ	41



ŞEKİL LİSTESİ

	Sayfa
Şekil 2.1: Tezde geliştirilen algoritmanın genel şeması.	7
Şekil 2.2: Sinir ağlarının basit bir gösterimi. Gizli katmanlar her bir girdi için farklı katsayılarla ve çıktılara sahiptir.	8
Şekil 2.3: Derinlemesine evrişim. $K \times K \times 1$ boyutundaki filtreler görüntünün her kanalına ayrı olarak uygulanır.	9
Şekil 2.4: Noktasal evrişim. Evrişim kerneli $1 \times 1 \times (\text{kanal sayısı})$ boyutundadır.	9
Şekil 2.5: Geleneksel evrişim algoritması.	10
Şekil 2.6: MobileNetV2 blok yapısı [35]. adım = 1 olan bloklardan sonra bir girdi ile çıktı arasında bir kısayol oluşturulmuştur. Dwise olarak belirtilen blok ise derinlemesine evrişimi göstermektedir.	11
Şekil 2.7: Stil transferi içerik ve stil görüntülerinin birleştirilmesiyle oluşturulur.	13
Şekil 2.8: 9×9 çekirdek kullanılarak ulaşılan hareket bulanıklığı sonuçları	14
Şekil 2.9: Oyuncu görüntüleri üzerinde kullanılan sıfırla dolgulama yöntemi	15
Şekil 2.10: File tespiti yapılmak üzere kırılan görüntü çıktısı	16
Şekil 2.11: Atış tespitinde kullanılan atış alanı	16
Şekil 2.12: Arka plan çıkarımı üzerine dilatasyon yapıldıktan sonra elde edilen öge	17
Şekil 2.13: Basketbol topu doğrulama modeli. Top atış alanından kırılarak modele 40×40 s, eklinde girdi olarak verilir	17
Şekil 2.14: Başarı-Iska tanımlaması. Aynı model 40 görüntü üzerinde birer birer çalıştırdıktan sonra 3'ten fazla 'In' sonucu almırsa atış, başarı kabul edilir.	18
Şekil 2.15: Kodlayıcı ve kodçözücünden oluşan DeepLabV3+ mimarisi. Kodlayıcı olarak modifiye edilmiş, DeepLabV3 kullanılırken, özel bir kodçözücü kullanılmıştır [42].	20
Şekil 2.16: Şutör tespit yöntemi. Oyuncu alanı tespiti arka plan çıkarımında yeterli hareketlilik bulunduktan hemen sonra yapılır.	21
Şekil 2.17: Oyuncu tanımlama algoritmasının genel işleyişi	22
Şekil 3.1: Basketbol topu onaylama modelinde kullanılan görüntüler. Top görüntüleri solda, arka plan görüntüleri sağda	25
Şekil 3.2: Skor tanımlaması veri setinden örnekler. Iska görüntüleri solda, başarı görüntüleri sağda	26
Şekil 3.3: Oyuncu tanımlama modelinin eğitiminde kullanılan veri setinden 2 örnek. (a) işlenmemiş, görüntüler, (b) arka plan silme yapılmış, görüntüler	26
Şekil 3.4: VID-1'deki oyuncular. (a) işlenmemiş görüntüler, (b) arka plan silme yapılmış, görüntüler	27
Şekil 3.5: Atış tanımlama model eğitim ve doğrulama setleri doğruluk oranları. Önceden eğitilmiş model altta, sıfırdan eğitilmiş model üstte	28

Şekil 3.6: Oyuncu tanımlama algoritmasının oyuncu bazında farklı videolar üzerinde K-Means ile elde edilen doğruluk sonuçları. (a)DeepLabV3+ kullanılmadan alınan sonuçlar, (b)DeepLabV3+ kullanımı ile arka plan silme yapılan görüntülerden alınan sonuçlar	31
Şekil 3.7: VID-2 görüntüleri üzerinde elde edilen vektörlerin 2 boyutlu gösterimi. (a) TL+K-Means, (b) TL+DeepLabV3++K-Means	32
Şekil 3.8: Oyuncu tanımlama algoritmasının oyuncu bazında tüm test seti üzerinde K-Means + Üçlü Yitim ile elde edilen doğruluk sonuçları. (a)DeepLabV3+ kullanılmadan alınan sonuçlar, (b)DeepLabV3+ kullanımı ile arka plan silme yapılan görüntülerden alınan sonuçlar	33
Şekil 3.9: Modellerden elde edilen öznitelik vektörlerinin 2 boyutlu gösterimi. (a) TL+K-Means, (b) TL+DeepLabV3++K-Means	34



ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 3.1: Gerçek zamanlı uygulamada oyuncu tanımlama doğruluk oranları.	29
Çizelge 3.2: K-Means kümeleme algoritmasıyla erişilen oyuncu tanımlama doğruluk oranları.	29
Çizelge 3.3: Farklı temel ağların VID-1 üzerinde gerçek zamanlı doğruluk performanslarının karşılaştırılması.	30
Çizelge 3.4: Sistemdeki tüm bileşenlerin VID-1 üzerindeki doğruluk oranları.	30

KISALTMALAR

- DL** : Derin Öğrenme (Deep Learning)
CNN : Evrişimli Sinir Ağı (Convolutional Neural Network)
KNN : En Yakın K Komsu (K-Nearest Neighbors)
TL : Üçlü Yitim (Triplet Loss)
BU : Çift Doğrusal Yukarı Örnekleme (Bilinear Upsampling)
IOU : Kesişim Birleşim Oranı (Intersection Over Union)
GT : Kesin Referans (Ground Truth)
BB : Sınırlayıcı Kutu (Bounding Box)
YN : Yanlış, Negatif (False Negative)
YNO : Yanlış, Negatif Oranı (False Negative Rate)
YP : Yanlış, Pozitif (False Positive)
YPO : Yanlış, Pozitif Oranı (False Positive Rate)
DN : Doğru Negatif (True Negative)
DNO : Doğru Negatif Oranı (True Negative Rate)
DP : Doğru Pozitif (True Positive Rate)
DPO : Doğru Pozitif Oranı (True Positive Rate)

1. GİRİŞ

Son yıllarda, derin öğrenme yöntemlerinin popülerliği ve etkinliğinin artmasının da etkisiyle spor videolarının analizi üzerine birçok çalışma ortaya konulmuştur. Bu çalışmalar çeşitli spor türlerinde farklı uygulamalar için yapılmıştır. Bu uygulamalar, profesyonel müsabaka analizleri ve akıllı antrenman uygulamaları olarak ikiye ayrılabilir. Müsabaka analizleri, koç ve antrenörler için istatistiksel veri çıkarılmasını sağlayarak daha hızlı taktik analizler yapılmasına yardımcı olmaktadır. Bu amaçla, araştırmacılar, sahada olanları otomatik olarak elde etmeye yönelik metotlar geliştirmişlerdir. Bu çalışmanın konusu olan basketbol düşünüldüğünde, bu metotlar önemli an tespiti [1, 2], oyuncu takibi ve tanımlanması [3, 4, 5, 6], atış, ve skor tanımlanması [7, 8] ile aksiyon sınıflandırma olmak üzere tanımlanabilir. Fakat, bu algoritmaların bazıları hâlâ yarı-otomatik olarak uygulanabilmektedir.

Akıllı antrenman uygulamalarına gelindiğinde ise, ShotTracker Player ve HomeCourt gibi ticari uygulamalar dışında, literatürde, sınırlı sayıda çalışma bulunmaktadır. Yine de [7]'de ortaya konulan uygulamada, yazarlar tek kameralı bir sistem üzerinde atışları tespit edip önemli an çıkarımı yapabilen bir model geliştirmişlerdir.

Bu tez, basketbol video analizi ve oyuncu tanımlama konularını birleştirerek çok oyunculu videolar üzerinde her oyuncu için skor analizi yapabilen bir akıllı antrenman uygulaması oluşturmayı amaçlamaktadır. Uygulamada, literatürdeki birçok metodun aksine her aşamada derin öğrenme (deep learning, DL) kullanılmıştır. Bunun nedeni, derin öğrenme uygulamalarının yüksek doğruluk oranını geleneksel uygulamalara göre çok daha kısa sürede elde edebilmesidir. Atış, ve oyuncu tanımlama işlemleri birleştirilmiş, olsa da ana odak oyuncu tanımlama ve takibidir.

Bu tez takiben dört bölümden oluşmaktadır. Metodoloji başlığı altında algoritmada kullanılan veri setleri ve yöntemler açıklanacaktır. Bu bölümde öncelikle atış, tanımlama algoritması alt başlıklarıyla açıklanacak, daha sonra oyuncu tanımlama metodu anlatılacaktır. Deney kısmında ise kullanılan metotların ne şekilde kullanıldıkları ve nasıl elde edildikleri ile bu metotların sonuçlarından bahsedilmiştir. Sonuç kısmında ise çalışmanın bulguları ve başarımları özetlenecek ve olası gelecek çalışmalar değerlendirilecektir.

1.1 Literatür Aras,tırması

Bu kısımda, öncelikle genel olarak makine öğrenmesi ve derin öğrenme uygulamalarının görüntü is,lemede kullanımları hakkında genel bilgi verildikten sonra bu tezin konusuna doğrudan bağlı araştırmalardan bahsedilecektir.

Makine öğrenmesi uygulamalarının görüntü işleme üzerinde kullanımları bu tezle ilgili olarak öznelik çıkarımı, sınıflandırma ve nesne tespiti olarak tanımlanabilir. Öznelik çıkarımı, derin öğrenme dışında geleneksel uygulamalara karşı başarısız olmaktadır. Bu nedenle, makine öğrenmesi geleneksel yöntemlerden çıkarılan öznelikler ile nesne tespiti ve görüntü sınıflandırma gibi uygulamalarda bas,arıyla kullanılmış,tır. Yaygın nesne tespiti uygulamalarına örnek olarak Dalal ve çalı,s,ma arkadaş,larının Histogram of Oriented Gradients (HOG) ile öznelik çıkarımı yaptıktan sonra bir Support Vector Machine (SVM) [9] ile kis,i tespiti uygulaması verilebilir [10]. Burada, tüm görüntü üzerinde kis,i tespiti için kayan pencereler kullanılmış,tır. Bas,ka bir örnek olarak [11]'deki yüz tespiti uygulaması gösterilebilir. Bu uygulamada ise Haar benzeri öznelikler çıkarılmış ve AdaBoost [12] ile öğrenilmiştir. AdaBoost kullanımı sayesinde yüz buldurmeyen pencerelerin reddi hızlı ve etkili bir biçimde sağlanmış,tır. Ancak, bu tür nesne tespiti uygulamaları nesneye özel kalmakta ve bakış açısı deęişikliklerine göre düşük performans göstermektedir.

Sınıflandırma ve nesne tespiti uygulamalarında derin öğrenmenin etkisi ILSVRC yarış,masında açıkça görülmektedir. Yarış,maya 2012'de 60 milyon parametreye sahip bir derin öğrenme ağıyla katılan Krizhevsky ve çalı,s,ma arkadaş,ları [13] sınıflandırma kategorisinde yüksek farkla birinci olmuş,lardır [14]. Buna ek olarak, modifiye edilmiş bir AlexNet altyapısına sahip OverFeat [15] adlı derin öğrenme modeli, 2013 yılında nesne tespiti alanında birincilik kazanmış,tır. Bu yarış,ma için oluş,turulan verisetinin birçok farklı nesne ile binlerce görüntü içermesinden dolayı derin öğrenmenin genelleştirmeye daha açık olduğu görülmektedir.

1.1.1 Atıs, Tanımlama

Bu tezde bahsedilen atıs, tanımlama problemi literatürde çoęunlukla geleneksel yöntemler üzerinden incelenmiş,tır. Bu yöntemlerden ilki top tespiti ve takibidir. Aras,tırmacılar [16]'da top tespitini modifiye edilmiş, bir arka plan çıkarma ve dairesellik filtresi ile kullanarak sağlamıştır. Ratgeber ve çalı,s,ma arkadaş,ları [8] ise top ve fileyi renk tespiti ile elde etmişlerdir. Tespit edilen topları kullanarak atışın başladığı ve bittiği görüntüleri ayırt ettikten sonra, oyuncuların atışın başladığı görüntü üzerinden resimsel yapılar (pictorial structures) [17] kullanarak tespit ederek s,ütörü bulmuş,lardır. Bir bas,ka çalı,s,ma ise Mean-Shift ve Kalman filtre kullanarak hızlı ve yüksek doğruluk oranına sahip bir top tespiti algoritması tanıtmış,tır [18]. Yukarıda belirtilen çalı,s,malar tüm

çerçevede top tespitine odaklanmaktayken [7]'de bir atış, alanı tanımlanmış, tır. Burada, toplu kanal özellikleri Aggregated Channel Features (ACF) detektörü kullanılarak basketbol potası bulunmuş ve atış tespitinin ve tanımlanmasının yapıldığı bir atış alanı olarak tanımlanmış, tır. Atış, tespitinin ardından, arka plan çıkarımı yapılmış, görüntülerin özel bir şekilde birleştirilmesiyle elde edilen görüntüler üzerinde eğitilmiş bir derin öğrenme modeli atışın skor durumunu tespit etmek için kullanılmıştır. Bu yöntem, kısa süre gerektirip yüksek doğruluk oranına ulaşmaktadır.

1.1.2 Oyuncu Tanımlama

Bu tezde, oyuncu tanımlama iki şekilde kullanılabilir, şekilde tasarlanmıştır: sütor tanımlama ve oyuncuların bireysel takibi. Literatürde oyuncu tanımlaması genellikle bireysel oyuncu takibinde kullanılmış, tır. Bu uygulamalar oyuncuları video üzerinde tespit ederek bas, lamaktadırlar. Sonrasında, tespit edilen oyuncu görüntüleri üzerinde öznitelik çıkarımı yapılmakta ve çıkarılan öznitelik vektörleri daha sonraki tespit edilen oyuncu görüntülerinden çıkarılanlarla karşı, las, tırılmaktadır. Öznitelik çıkarımı için birçok farklı yöntem kullanılmış, tır. [5]'te tanıtılan sistemde, öznitelik çıkarımı SIFT (Scale-Invariant Feature Transform) [19] ile yapılmış, tır. Ek olarak, oyuncu forma numaraları kaba kuvvet eşleştirme ile elde edilmiştir. Oyuncu tanımlama doğruluğunu artırmak için bir oyuncu takibi algoritması kullanılmış, tır. Benzer bir yaklaş, ım olan [4]'te ise forma numaraları gradyan farkı ve Optik Karakter Tanıma (OCR) kullanılarak elde edilmiş ve oyuncu takibi parçacık filtreleri ile sağlanmıştır.

Birkaç uygulamada, oyuncuların ayırt edilmesinde oyuncu yüzleri kullanılmış, tır. [20]'de oyuncu yüzleri parçalara ayrıldıktan sonra SIFT kullanılarak es, les, tirilmiş, tir. Yüzlerin parçalara ayrılmasıyla SIFT ile alınan yanlış, tespitlerin önüne geçmek hedeflenmiş, tir. Benzer bir uygulamada, beyzbol oyuncularının tespiti, oyuncu ve yüz tespiti modifiye edilmiş, Viola-Jones [11] metodu kullanılarak yapılmış, tır [21]. Bu yöntemde ise elde edilen oyuncu yüzleri En Yakın Koms, u Sınıflandırıcı (Nearest Neighbour Classifier, NNC) kullanılarak oyuncu veritabanındaki diğer yüzler ile karşılaştırılmış ve tanımlama sağlanmıştır.

Yüksek doğruluk oranına ulaşan başka bir çalışmada ise; SIFT, MSER [22] ve RGB renk histogramlarının birles, tirilmesiyle olus, turulan öznitelik vektörleri bir sınıflandırıcıyı eğiterek oyuncu tanımlaması yapılmaktadır [6]. Bu yöntem bir NBA maç videosunda denenmiş ve %80 üzerinde doğruluk oranına ulaşmıştır; ancak, bu öznitelik vektörlerinin çıkarımı yüksek zaman istemesinden dolayı gerçek zamanlı kullanımı mümkün değildir. Derin öğrenmenin de oyuncu tanımlama uygulamalarına dahil olmasıyla çok daha yüksek performanslı yöntemlerin elde edilmesi mümkün olmuştur. [3]'te ortaya konulan çalış, mada, öznitelikler, SIFT gibi düşük seviye öznitelik çıkarıcıların aksine, bir CNN

kullanımı ile elde edilmiştir. Vücut parça görüntüleri ve çok ölçekli görüntülerden iki farklı öznitelik vektörü çıkarılmıştır. Vücut parçalarını çıkarmak için yazarlar Convolutional Pose Machines (CPM) [23] kullanarak vücuttaki anahtar noktalarını elde etmişlerdir. Her vücut parçası CNN'e girdi olarak verilmiş, ve çıktı vektörleri bir Fisher vektörüne eklendikten sonra çok ölçekli görüntülerin çıktısıyla birleştirilmiştir. Ardından, bu vektör bir SVM kullanımı ile sınıflandırılmıştır. Bu model çok yüksek doğruluk oranına sahip olsa da bu tezde ortaya konulan aksine güdümlü sınıflandırma yapmaktadır.

Yakın tarihte, oyuncu takibi ile birleştirilmiş komplike bir derin öğrenme metodu ortaya atılmıştır [24]. Bu sistem, CNN'den elde edilmiş, oyuncu öznitelik vektörleri ile forma numarası ve takım sınıflandırmasını birleştirerek oyuncunun kimliğini bulmaktadır. Ancak, bu model oyuncu özniteliklerini çıkarmak için oyuncu arka planını silmeyi ve oyuncu pozunu elde etmeyi sağlayan kompleks CNN mimarilerine ihtiyaç duymaktadır. Bu nedenle, modelin gerçek zamanlı bir uygulamada kullanılması mümkün değildir.

Bu tezde kullanılan oyuncu tanımlama algoritmasına benzer yöntemler ise oyuncu tanımlama işleminden daha yaygın olan kişi tanımlama uygulamalarında ortaya konulmuştur. Kişi tanımlama uygulamaları, genellikle insanların genel görünüşünün karşılaştırılmasıyla yapılmaktadır. Derin öğrenmenin yaygınlaşmasından önce, bu uygulamalar kişinin iki farklı görüntüsü arasında değişmeyen öznitelikleri bulmayı hedeflemişlerdir. Bu özniteliklerin en yaygınlarından biri kişiye ait renk bilgisidir [25, 26, 27]. Gray ve Tao [25] renk bilgisine ek olarak çesitli filtreler (Schmid ve Gabor) kullanarak elde ettikleri kaplama bilgisini de öznitelik olarak kullanmışlardır. [26]'da yazarlar, görüntülerdeki tutarlı kenar özelliklerini elde etmek için bir uzamsal ve zamansal bölütleme metodu geliştirmişlerdir. Diğer bir çalışmada ise 3 farklı renk uzayından (RGB, HSV ve YCrCb) alınan histogramlara ek olarak Yerel İkili Örüntü (Local Binary Patterns, LBP) histogramı kullanılmıştır [27].

Derin öğrenmenin kişi tanımlamadaki ilk uygulamalarından biri olan [28], siyam ağlarından (siamese networks) yararlanarak iki görüntü arasında etkili bir uzaklık metriği bulmak hedeflemiştir. Çalışmada, görüntüler üç parçaya ayrılmış, birbirine karşı gelen parçalar ayrı ayrı karşılaştırıldıktan sonra çıktıları birleştirilmiş, ve bir benzerlik metriğine verilmiştir. Sinir ağının eğitiminde yitim metriği olarak Binomial deviance kullanılmıştır. Yüksek performans ulaşılan bazı uygulamalarda ise kişi tanımlaması, sınıflandırma problemi olarak ele alınıp yitim çapraz düzensizlik (cross-entropy) ile hesaplanmıştır. [29]'da araştırmacılar bu yitim fonksiyonuna ek olarak Çekişmeli Üretici Ağlar (Generative Adversarial Networks, GAN) kullanarak yapay veri üreterek yüksek performansına ulaşmışlardır. Çalışmada, yapay olarak üretilen görüntüler üzerinde etiket yumuşatma yapılmış, ve böylece modelin genelleştirilmesi

sađlanmıřtır.

Tezin konusuna benzer bir alıřmada, retilen zneliklerin Geitli Tekrarlayan Birim'lere (Gated Recurrent Unit, GRU) [30] verilerek kis,iye ait znelik vektrleri ıkarılmıř [31]; bu vektrler daha sonra zel bir uzaklık metriđiyle karřılařtırılmıřtır.

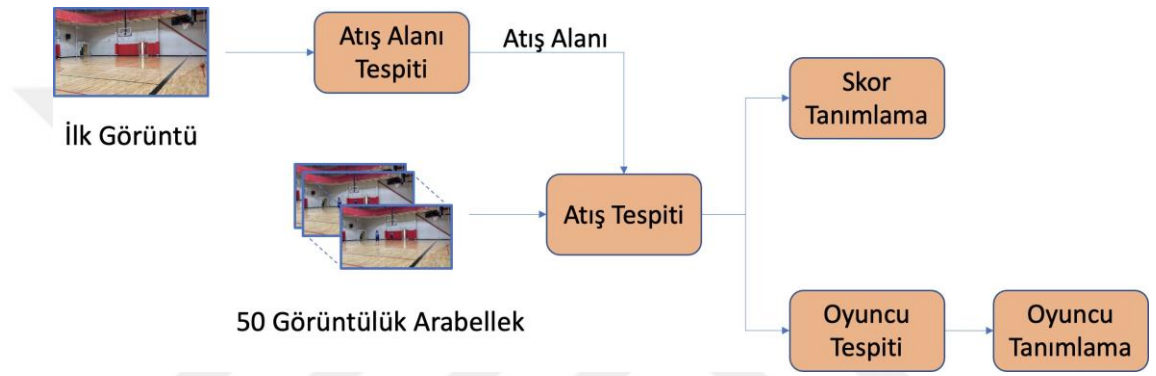
Tezin daha nceki uygulamalardan farkı ise, bir akıllı antrenman uygulaması tanıtmasının yanında bu uygulamanın bir video zerinde gerek zamanlı olarak oyuncu tanımlaması yapılmasına da olanak sađlamasıdır. Komple algoritmada srekli olarak řutların tespiti ve sınıflandırması yapıldıktan sonra oyuncu tespiti ve tanımlaması yapılır. Bylece, her bir oyuncu iin bađımsız olarak atıř performansı deđerlendirilmesi sađlanır. Bu tezin literatre en byk katkısı, derin đrenmenin yođun kullanılmasıyla hızlı ve dođruluđu yksek bir analiz uygulaması oluřturmasıdır.





2. METODOLOJİ

Spor video analizi, görüntü işleme alanında popülerliğini korumaktadır. Ancak, literatürde, basketbol videoları analizi için genellikle geleneksel yöntemler ile yayınlanmış, basketbol maçları kullanılmıştır. Bu tezde tanıtılan metot ise son zamanlarda derin öğrenmenin gösterdiği yüksek performansı göz önünde bulundurarak tamamiyle derin öğrenme bazlı bir sistem olarak geliştirilmiştir. Algoritmanın genel şeması Şekil 2.1’de verilmiştir.



Şekil 2.1: Tezde geliştirilen algoritmanın genel şeması.

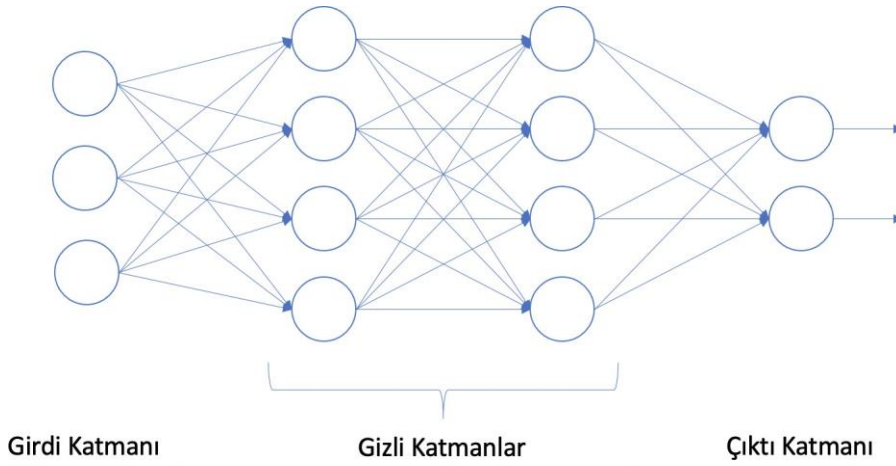
Görüldüğü gibi, atış alanı ilk görüntüden elde edilir ve atış tespiti güncel görüntü üzerinde bu atış, alanından yararlanılarak gerçek zamanlı olarak yapılır. Burada 50 görüntülük arabellek kullanılmasının sebebi ise oyuncu tanımlama için şutör tespittir. Algoritmanın hedefi, her bir oyuncu için skor tespiti yapılması ve oyuncuların birbirinden ayırt edilmesidir. Her bir atış, tespit edildikten sonra skor tanımlama ve oyuncu tanımlama paralel olarak çalışır.

Bu kısımda, öncelikle derin öğrenme ve kullanılan derin öğrenme metotları ile ilgili kısa bilgiler verilecek, ardından veri seti üzerinde yapılan işlemler anlatıldıktan sonra her bir aşama için metotlar açıklanacaktır.

2.1 Derin Öğrenme

Derin öğrenme, beyin yapısını taklit eden bir makine öğrenmesi metodu olarak gösterilebilir. Bu metot sinir ağlarından oluşmakta ve bu ağlar istenilen sonuç için girdiyi işlemektedir. Sinir ağları ise farklı işlemler yapılabilen farklı katmanlardan oluşur. Bu katmanlar: girdi katmanı, gizli katmanlar ve çıktı katmanı olarak gösterilebilir. Şekil 2.2’de verilen örnekte gizli katmanda girdiler farklı ağırlıklarla çarpılarak toplanır ve

çıktı olarak verilir. Derin öğrenmenin derin olmasını ise çok sayıda gizli katmanlardan oluşması sağlamaktadır.



Şekil 2.2: Sinir ağlarının basit bir gösterimi. Gizli katmanlar her bir girdi için farklı katsayılarla ve çıktılara sahiptir.

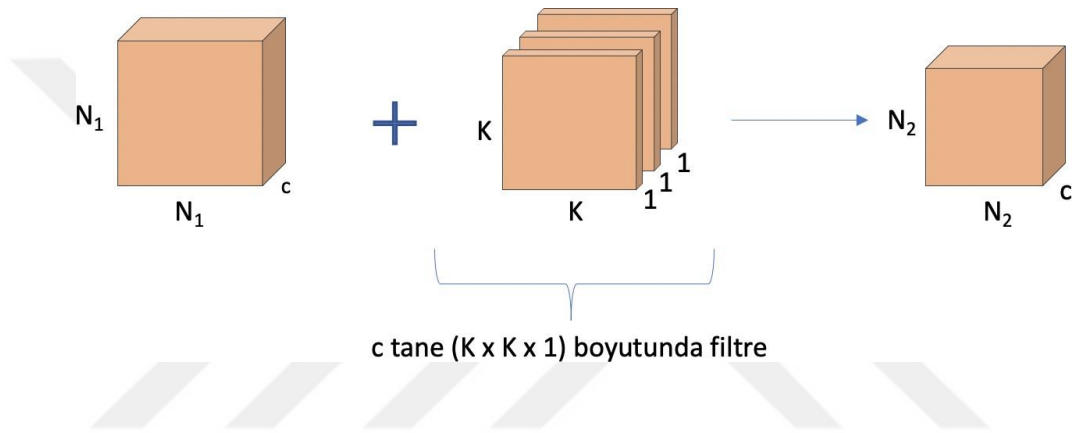
Görüntü işleme uygulamalarında, genellikle derin öğrenme modelleri evrişimli sinir ağlarından oluşur. Evrişimli sinir ağlarında, gizli katmanlar her girdi elementinin ağırlıklı çarpılması yerine girdi üzerinde evrişim işlemi için kullanılır. Bunun nedeni, her elemanın birbirlerine bağlı olduğu katmanların çok fazla parametreye sahip olmasıdır. Örnek olarak, $K \times K \times R$ büyüklüğünde bir girdiden $N \times 1$ bir çıktı istendiğinde bu katmanlar $(K^2RN + N)$ sayıda eğitilebilir parametre gerektirirken aynı girdiye d tane $m \times n$ evrişim katmanı uygulandığında $((m \cdot n \cdot R + 1) \cdot d)$ parametre kullanılmaktadır. Evrişim yapılan çekirdek büyüklüğü ($m \times n$) genel olarak küçük olduğu için evrişim katmanları çok daha az parametre ile çalışabilmektedir. Böylece, daha derin bir ağ yaratılarak görüntü gibi komplike girdilerden hızlı sonuç alınması sağlanır.

Derin öğrenme modellerinin eğitilmesi, yitim fonksiyonundan elde edilen yitimin azaltılması için geri yayılım (back-propagation) [32] yönteminin kullanılmasıyla sağlanır. Geri yayılım ile elde edilen gradyanlar bir eniyileştirici kullanılarak eğitim aşamalarında güncellenir. Bu kombinasyon ilk olarak [33]'te tanıtılmış ve geçerliliği kanıtlanmıştır.

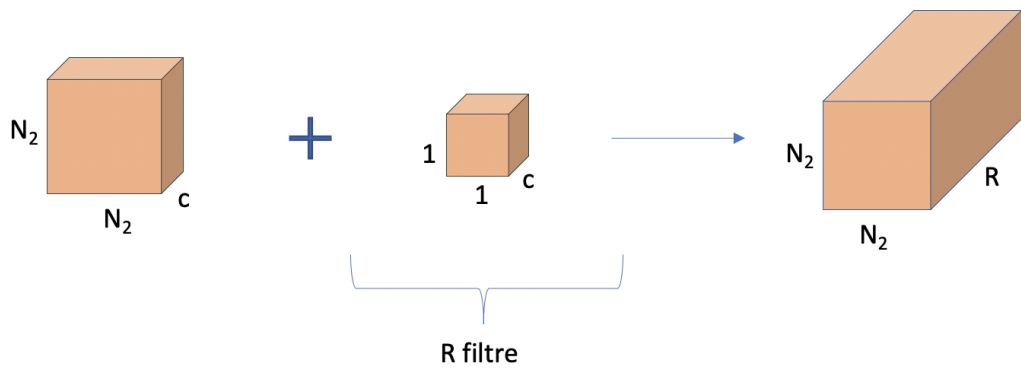
2.1.1 Kullanılan Derin Öğrenme Modelleri

Bu tezde açıklanan sistemin gerçek zamanlı uygulamalarda kullanabilmesi amacıyla ağ mimarisi olarak düşük çıkarım süreli derin öğrenme ağlarından yararlanılmıştır. Şütör tespitinde kullanılan YOLOv3-tiny [34] dışında, kullanılan tüm modeller için MobileNetV2 [35] baz alınmıştır. MobileNetV2 ağı 3.4 milyon parametreye sahip

olmasına rağmen sınıflandırma uygulamalarında tatminkar sonuçlar vermektedir. Bunun nedeni, mimarinin Derinlemesine Ayrılabilir Evrişim (Depthwise Seperable Convolution) kullanılarak daha hızlı sonuçlar almasıdır. Derinlemesine ayrılabilir evrişim, iki parçadan oluşur: derinlemesine evrişim ve noktasal evrişim. Derinlemesine evrişim, girdinin her kanalına ayrı ayrı uygulanan uzamsal evrişimlere verilen addır (Şekil 2.3). Noktasal evrişim ise $1 \times 1 \times N$ boyutunda tek bir süzgeç ile yapılır (Şekil 2.4). Bu kombinasyon, neredeyse aynı performansı vermesine rağmen çok daha hızlıdır. Şekil 2.5'te verilen standart evrişimin toplam işlem sayısı 2.1'deki gibi olurken, derinlemesine ayrılabilir evrişimin toplam işlem sayısı 2.2'deki gibidir. Bu işlemlerde c görüntü kanal sayısını, K filtre boyutunu, N_2 çıktı boyutunu ve R çıktı kanal sayısını göstermektedir.

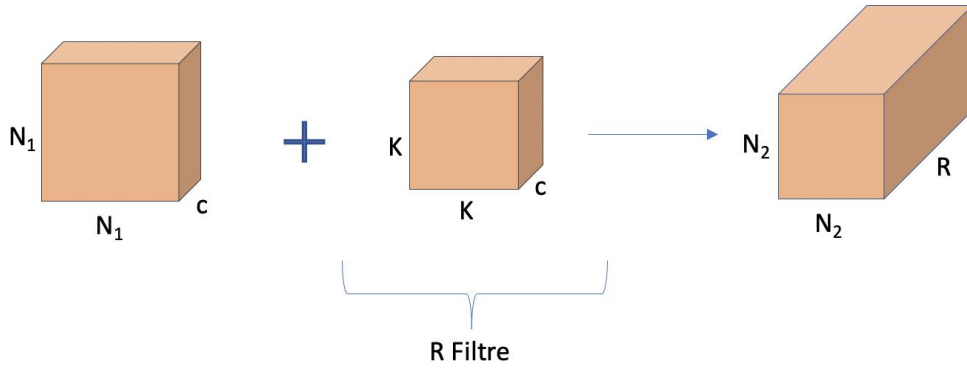


Şekil 2.3: Derinlemesine evrişim. $K \times K \times 1$ boyutundaki filtreler görüntünün her kanalına ayrı olarak uygulanır.



Şekil 2.4: Noktasal evrişim. Evrişim kerneli $1 \times 1 \times$ (kanal sayısı) boyutundadır.

$$cRK^2N_2^2 \quad (2.1)$$



Şekil 2.5: Geleneksel evrişim algoritması.

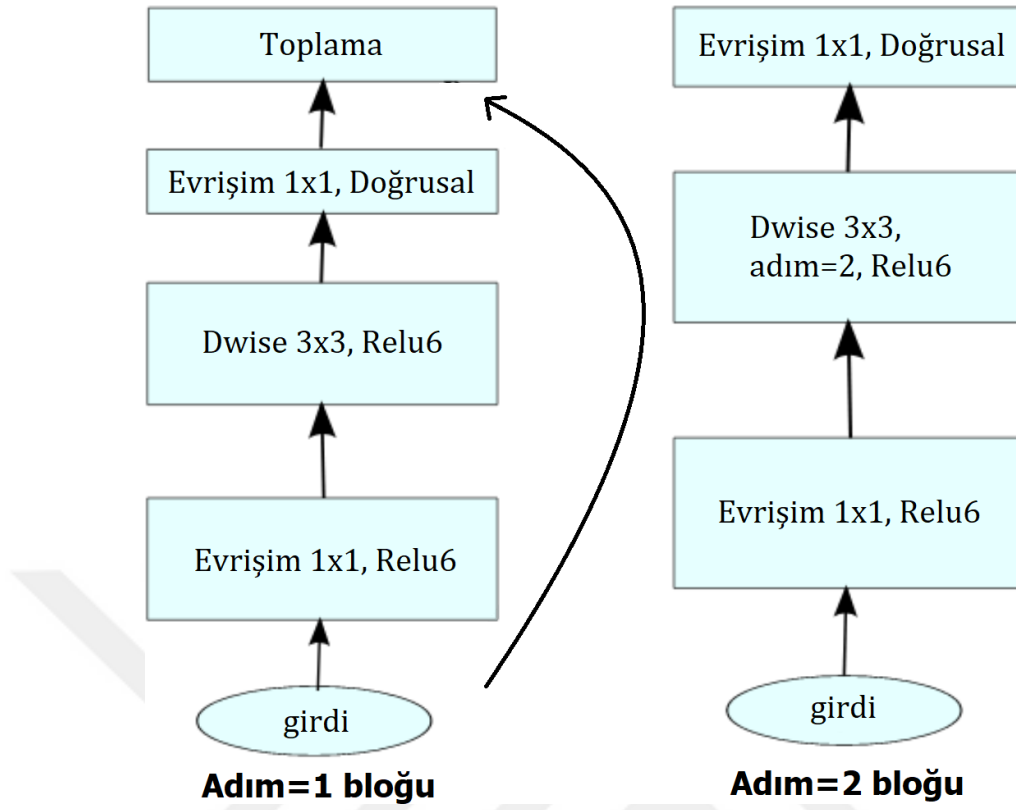
$$\frac{cK^2N^2}{2} + \frac{cRN^2}{2} \quad (2.2)$$

Derinlemesine ayrılabilir evrişime ek olarak, darboğaz katmanı (bottleneck layer) adı verilen katmanları arasında kısayollar oluşturularak daha detaylı öznitelikler elde edilmektedir (Şekil 2.6). Bunun sonucu olarak düşük çıkarım süresinde yüksek doğruluk oranına ulaşmaktadır.

Atış alanı tespitinde ise MobileNetV2, SSD (Single Shot Detector) [36] ile birlikte kullanılmıştır. SSD, çıktı kutuları uzayını farklı boyutlarda ve şekillerde varsayılan kutulara ayırarak öngörü zamanında her bir sınıf için varsayılan kutu ve kutudaki sapma miktarını verir. Aynı zamanda, farklı boyutlarda öznitelik vektörleri kullanılarak farklı boyutlardaki objelerin tespit edilmesi sağlanır. Modelin ilk uygulamasında kullanılan VGG-16 [37] aksine, daha tez sonuç alınabilmesi için öznitelik çıkarıcı olarak MobileNetV2 ağı baz alınmıştır.

Şutörlerin tespitinde kullanılan insan tespiti algoritması içinse YOLOv3-tiny modelinden yararlanılmıştır. YOLOv3-tiny hafif bir model olup 416x416 büyüklüğünde girdi almakta ve Titan Z üzerinde saniyede 220 kez çalışabilmektedir [34]. Bu sayede, oyuncu tespitinin tekrarlı yapılabilmesi sağlanmaktadır.

YOLOv3-tiny, 13 evrişim katmanından oluşmakta ve iki ayrı boyutta çıktı vermektedir. Bu çıktılar sınıfı, nesnenin bulunduğu ızgarayı (grid) ve ızgaraya yapılacak kaymaları içerir. Genel olarak, büyük veri setlerinde YOLOv3'ye karşı çok daha düşük doğruluk oranına ulaşsa da şut antrenmanında salon içerisinde çok fazla kişi olmayacağı için bu modelin kullanımı yeterli doğruluğa ulaşılmasını sağlamaktadır.



Şekil 2.6: MobileNetV2 blok yapısı [35]. adım = 1 olan bloklardan sonra bir girdi ile çıktı arasında bir kısayol oluşturulmuştur. Dwise olarak belirtilen blok ise derinlemesine evrişimi göstermektedir.

2.2 Veri Seti

Atış sınıflandırması için kullanılan veri setleri el ile, video ve fotoğraflar çekilerek oluşturulmuştur. Videolar ve fotoğraflar iPhone XS, XS Max ve iPad Pro 1st Gen kullanılarak 1920 x 1080 çözünürlükte alınmıştır. Antrenman videolarında bu cihazlar bir uçayak üzerinde 1.2 ve 1.6 metre arasında yükseklikte tutulmuştur. Uçayak ise basketbol kortunun ortasına yakın olacak şekilde farklı açılarda konumlandırılmıştır. Atış sınıflandırmasının her bir aşaması için farklı veri setleri kullanılmıştır. Ayrıca, veri setleri üzerinde birtakım zenginleştirmeler ve değişiklikler yapılmıştır. Bu bölümde yapılan bu değişikliklere değinilecektir.

2.2.1 Stil Transferi

Atış alanı tespitinde MobileNetV2+SSD [35][36] ağı kullanılmıştır. Bu ağın kullanımı ile basketbol filesini kırılan kare şeklindeki görüntüde bulunması sağlanmıştır. Sinir ağının eğitilmesi için farklı basketbol kortlarından çeşitli açılarda ve farklı ışıklandırma

koşullarıyla fotoğraflar alınmıştır. Basketbol filesi tespiti renkli (RGB) görüntüler üzerinde yapılır.

Veri seti üzerinde çeşitli görüntü çoğaltma yöntemleri rastgele uygulanmıştır. Bu yöntemler: tuz-biber gürültüsü ekleme, renk kanalı karıştırma, histogram dengeleme, devinim bulanıklığı ekleme ve yatay çevirme ve bunların çeşitli kombinasyonlarıdır. Ek olarak, görüntüler üzerinde veri seti zenginleştirme için stil transferi yapılmıştır.

Stil transferi, genellikle, belli bir stile sahip sanatsal görüntülerin tarzının girdi görüntülerinin içeriğiyle birleştirilmesi için kullanılır. Burada ise, farklı atış alanı tespiti görüntülerinin tarzlarının birbirleriyle değiştirilmesi için kullanılmaktadır. Bu değiştirme tespit modelinin eğitimi için yeni veriler çıkarılmasını sağlamaktadır. Stil transferi modeli olarak [38]'de ortaya konulan yöntem kullanılmıştır. Stil transferi, önceden eğitilmiş bir sinir ağının ara katmanlarından yararlanır. Bu katmanlar iki şekilde alınır: stil katmanları ve içerik katmanı. Stil katmanları, stili alınacak görüntünün modelde çalıştırılmasıyla elde edilir ve bir Gram matrisine eklenir. Sonrasında, beyaz gürültü görüntüsünden başlanarak gradyan iniş (gradient descent) ile gram matrisi stil görüntüsünününe benzeyen bir görüntüye ulaşılır. Bunu yaparken yitim metriği (L_{stil}) olarak 2.3'te görüldüğü gibi ortalama karesel uzaklık (mean square distance) kullanılır. Denklemde G ve A sırasıyla çıktı görüntüsünün Gram matrisini gösterirken N_k ve M_k karşılığı olan katmanın öznelik uzayının büyüklüğünü göstermektedir. ω_k ise stil katmanlarının birleştirilmesi için kullanılan ağırlığı göstermektedir.

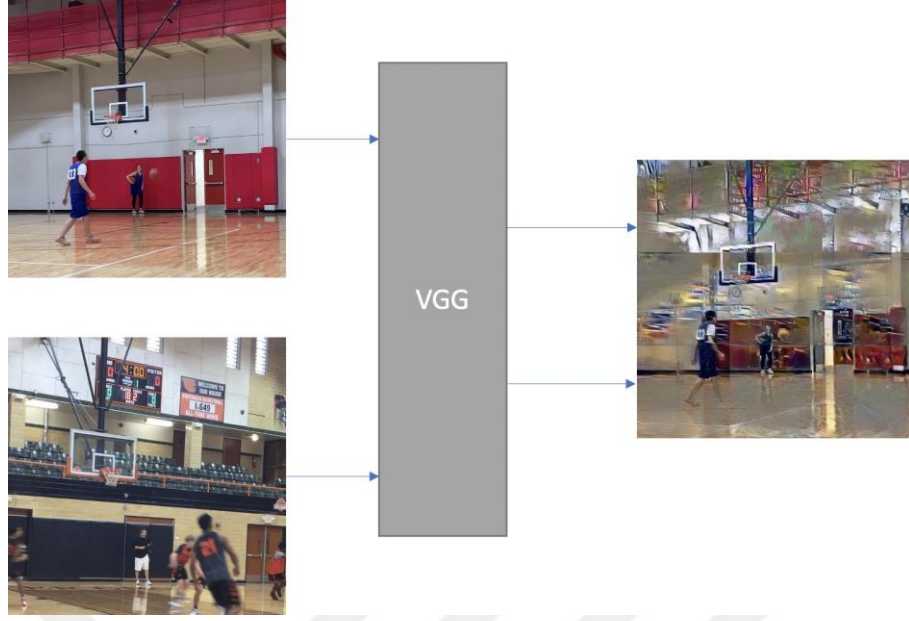
$$L_{stil} = \sum_k \omega_k \sum_{i,j} \frac{1}{4N_k^2 M_k^2} (G_{i,j} - A_{i,j})^2 \quad (2.3)$$

Benzer şekilde, içerik katmanı da içerik görüntüsüyle alınan öznelik vektörleri ile gradyan inişinde aynı şekilde kullanılır. Bu sayede, iki yitimin birleşimiyle stil görüntüsünün stiline ve içerik görüntüsünün içeriğine sahip bir görüntü elde etmek amaçlanır. İçerik yitimi (L_{icerik}) ve stil yitimi (L_{stil}) 2.4'teki gibi birleştirilir [38]. Denklemde α ve β istenen sonuca göre ayarlanmış katsayılardır.

$$L = \alpha L_{stil} + \beta L_{icerik} \quad (2.4)$$

Bu uygulamada stil katmanları olarak [38]'deki gibi ImageNet [39] üzerinde eğitilmiş VGG-16 [37] 'block1_conv1', 'block2_conv1', 'block3_conv1', 'block4_conv1' ve 'block4_conv1' eşit ağırlıkta alınmıştır. İçerik katmanı olarak ise aynı ağın 'block5_conv2' katmanı kullanılmıştır.

Zenginleştirme işlemi öğrenme oranı 0.02 olan Adam iyileştiricisi kullanılarak 1000 adımda yapılmıştır. Yitim metriği içinse $\alpha = 10^{-1}$ ve $\beta = 10^4$ kullanılmıştır (Şekil 2.7).



Şekil 2.7: Stil transferi içerik ve stil görüntülerinin birleştirilmesiyle oluşturulur.

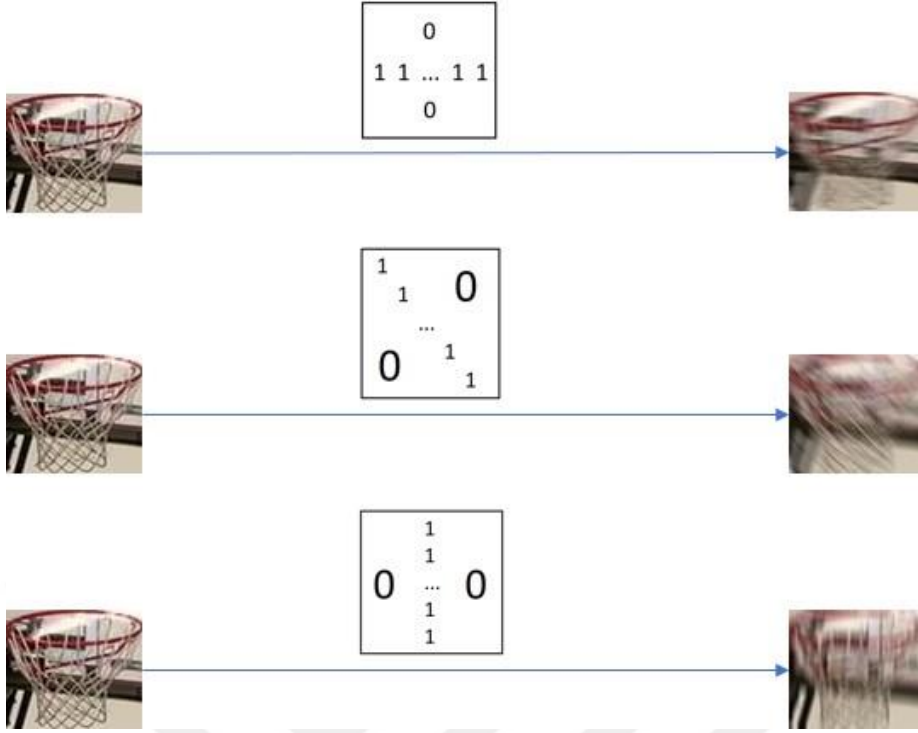
2.2.2 Hareket Bulanıklığı Ekleme

Algoritmanın skor tanımlama kısmında basketbol filesi görüntüleri kullanıldığı için bazı atışlardan ve oyuncu hareketlerinden dolayı görüntülerde gürütü oluşmaktadır. Skor tanımlama modeli, bu durumlarda kötü performans göstermeye eğilimlidir. Bu durumun önüne geçebilmek için skor tanımlamada eğitim verilerine yapay hareket bulanıklığı eklenmiştir.

Hareket bulanıklığı eklemek için 3 farklı filtreden yararlanılmıştır. Bu filtreler yatay, dikey ve çapraz olarak 3 şekilde ve 5x5, 7x7 ve 9x9 boyutundadırlar. Veri seti görüntülerinden %10'u rastgele biçimde filtrelenmiş ve eğitime eklenmiştir. Filtrelenmiş birkaç örnek Şekil 2.8'de verilmiştir.

2.2.3 Dolgulama

Derin öğrenme algoritmalarında çoğunlukla kare şeklinde görüntüler kullanılmaktadır. Bunun birçok nedeninden biri de önceden eğitimde kullanılan veri setlerinin kare görüntülerden oluşması olarak verilebilir. Bu tezde kullanılan önceden eğitilmiş DL ağlarının hepsi kare görüntü kullanmaktadır. Atış alanı tespiti ve basketbol topu doğrulama modellerine verilen görüntülerin elde edilme şekillerine bağlı olarak kare olması garantiliyken skor ve oyuncu tanımlarında böyle bir durum mümkün değildir. Bu görüntüler ağlara iki şekilde verilebilir:



Şekil 2.8: 9x9 çekirdek kullanılarak ulaşılan hareket bulanıklığı sonuçları

- Görüntüler modelin istediği boyuta direkt olarak boyutlandırılır.
- Sıfırla dolgulama yapılarak görüntülerin kare hale getirilmesi sağlanır.

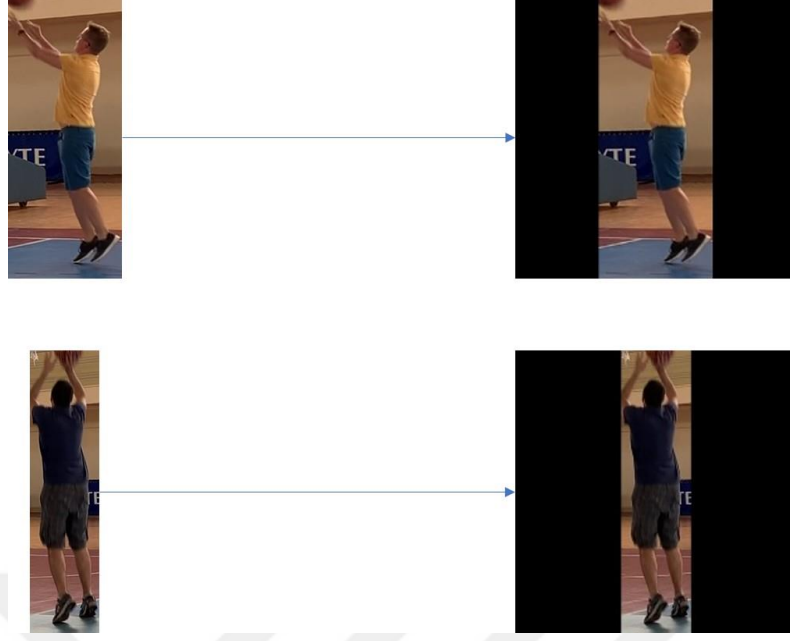
Skor tanımlama bir sınıflandırma modeli üzerinden yapıldığından ve basketbol filesi imgelerinin karesel olmasından dolayı direkt olarak boyutlandırma yapılmıştır. Ancak, oyuncu tanımlama algoritmasında oyuncuların vücut boyutları ve yüzleri gibi özellikleri tanımlamada kullanılması gerekliliği nedeniyle oyuncu görüntülerinde kare şekile ulaşılması için sıfırla dolgulama yapılmıştır (Şekil 2.9). Ayrıca, oyuncu görüntülerinin genişliği oyuncuların uzaklığına ve boyuna göre değişmektedir.

2.3 Atış, Sınıflandırması

Atış, sınıflandırması üç farklı aşamada incelenebilir:

- Atış, Alanı Tespiti
- Atış, Tespiti
- Skor Tanımlama

Atış, alanı tespiti, atış, tespiti yapılırken sürekli top tespiti yapılması gerekliliğinin kaldırılması amacıyla eklenmiştir.



Şekil 2.9: Oyuncu görüntüleri üzerinde kullanılan sıfırla dolgulama yöntemi

2.3.1 Atış, Alanı Tespiti

Atış tespitine başlamadan önce, pota arkalığının ve potanın içinde bulunduğu bir alan tanımlanması gerekmektedir. Bu alanın bulunabilmesi için MobileNetV2+SSD derin öğrenme modeli kullanılmaktadır. Bu model, MobileNetV2 ve SSD ağlarının birleşimi ile oluşturulmuş, ve gerçek zamanlı nesne tespiti için kullanılmaktadır. Bu tezde, 2.10'de kırılmış, görüntü bu modele girdi olarak verilmekte ve çıktı olarak basketbol filesini gösteren bir sınırlayıcı kutu (bounding box) alınmaktadır.

File tespiti yapıldıktan sonra, atış tespiti yapılması için Şekil 2.11'de görülen atış alanı tanımlanır. Atış, alanı, bulunan file sınırlayıcı kutusunun merkezinden yatay olarak her iki yönden 112 piksel ve dikey olarak 40 piksel aşağı ve 184 piksel yukarı olarak kırılarak elde edilir. Bu alan daha sonra geri plan çıkarımı ile atış, tespitinde kullanılacaktır. Ayrıca, elde edilen file daha sonra skor tanımlamada kullanılacaktır.

2.3.2 Atış, Tespiti

Atış, alanı tespitinden sonra, sürekli olarak geri plan çıkarımı yapılarak atış, alanındaki değişimler incelenir. Her geri plan çıkarımı işleminden sonra, elde edilen görüntü üzerinde eliptik yapısal elementler bulunarak dilatasyon uygulanır. Bu uygulama ile basketbol topuna benzeyen öğeler bulunur (Şekil 2.12). Ancak, bu öğeler bazı



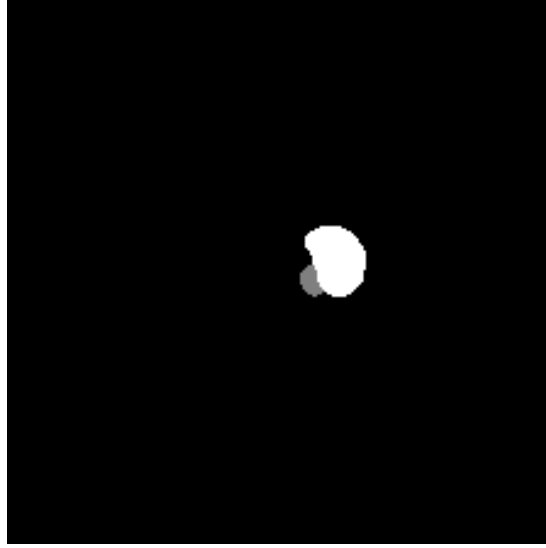
Şekil 2.10: File tespiti yapılmak üzere kırılan görüntü çıktısı



Şekil 2.11: Atış tespitinde kullanılan atış alanı

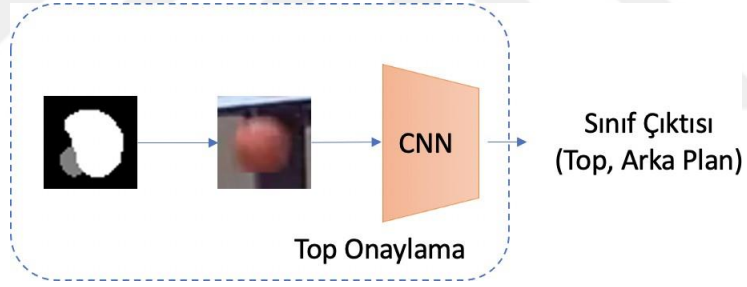
durumlarda oyuncu hareketleri gibi basketbol topu dışındaki nedenlerden de kaynaklanabilmektedir.

Yanlış pozitiflerin önüne geçmek için, öge tespitinden sonra, ögenin bir top olduğunu doğrulamak üzere MobileNetV2 bazlı başka bir doğrulama modeli kullanılır. Bu modele girdi olarak, arka plan çıkarımından elde edilen öge merkezde olacak şekilde tüm atış



Şekil 2.12: Arka plan çıkarımı üzerine dilatasyon yapıldıktan sonra elde edilen öge

alanından kesilmiş 40x40 boyutunda görüntüler verilir (Şekil 2.13). Bu CNN'in çıktısı Top, Arka plan olarak sınıflandırma sonucudur.

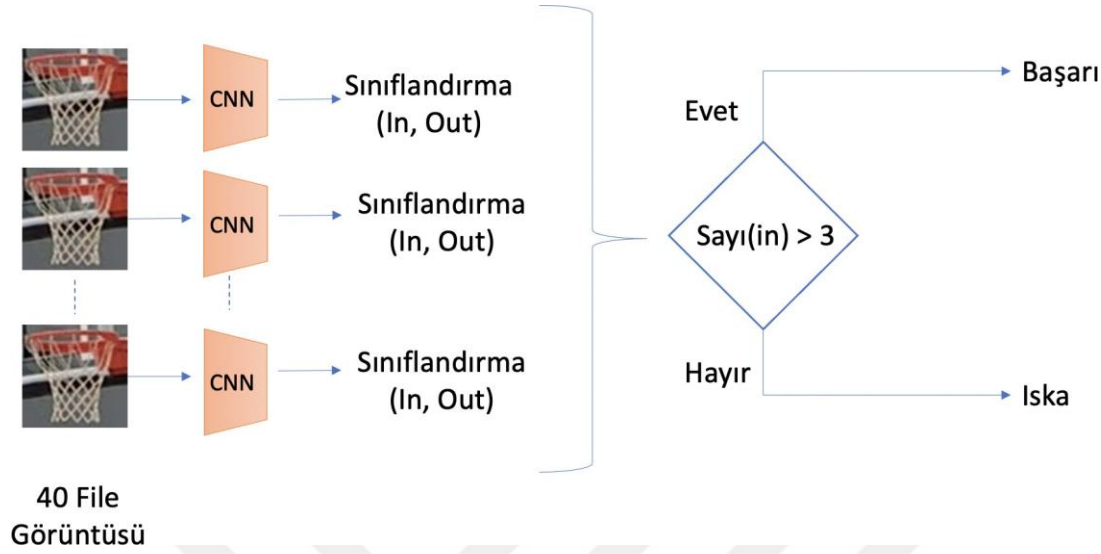


Şekil 2.13: Basketbol topu doğrulama modeli. Top atış alanından kırılarak modele 40x40 s,eklinde girdi olarak verilir.

2.3.3 Skor Tanımlama

Daha önce belirtildiği gibi, skor tanımlama elde edilen file görüntüleri üzerinde uygulanmaktadır. Şut sonucunu elde etmek amacıyla top doğrulama modeline benzer olarak, MobileNetV2 sınıflandırma ağı kullanılmaktadır. Ancak, top doğrulama modelinden farklı olarak bu model atış, tespitinden sonraki 40 görüntüde birer birer kos,maktadır. Burada 40 görüntü, topun fileden sektikten sonra tekrar fileye girme ihtimali kayda alınarak seçilmiştir,tir. Görüntülerde kos,an sınıflandırma CNN'i çıktı olarak başarı ve ıska (in, out) vermektedir. 40 görüntünün çıktıları Şekil 2.14'te gösterildiği

gibi karşılaştırılır ve başarı çıktısı 3'ten fazla kez elde edilirse atışın başarılı olduğu sonucuna ulaşılır, ancak daha az kez elde edilirse atışın iska olduğu çıkarımı yapılır.



Şekil 2.14: Başarı-Iska tanımlaması. Aynı model 40 görüntü üzerinde birer birer çalıştırıldıktan sonra 3'ten fazla 'In' sonucu alınırsa atış başarılı kabul edilir.

2.4 Oyuncu Tanımlama

2.4.1 Yitim Fonksiyonu

Oyuncu tanımlama işleminde, yitim fonksiyonu olarak Üçlü Yitim (Triplet Loss, TL) kullanılmıştır. Oyuncu tanımlaması siyam ağlarından faydalanılarak yapıldığı için çıkarılan öznelik vektörleri arasındaki Öklid uzaklığının belirli derecede ayırt edilebilir olması son derece önemlidir. 2.5'te görüldüğü gibi Öklid uzaklığı, $d(\mathbf{p}, \mathbf{q})$, iki vektörün (\mathbf{p} ve \mathbf{q}) her bir elemanının (p_i, q_i) arasındaki farkın toplamının karekökünün alınmasıyla elde edilir. TL fonksiyonu modelin farklı oyuncuların öznelik vektörleri arasındaki uzaklığın optimizasyonunu hedeflemektedir. Derin öğrenmede kullanımı olarak [40]'da ortaya atılan bu metrik birçok insan tanımlama uygulamasında da kullanılmış, ve başarılı sonuçlara ulaşmıştır [41].

Oyuncu tanımlamada kullanılan üçlü yitim 2.6'da verilmiştir. Denklemde, $d(\mathbf{p}, \mathbf{a})$ pozitif örnekle (\mathbf{p}) baz alınan örnek (\mathbf{a}) arasındaki Öklid uzaklığını gösterirken $d(\mathbf{n}, \mathbf{a})$ negatif örnekle (\mathbf{n}) baz alınan örnek (\mathbf{a}) arasındaki Öklid uzaklığını göstermektedir. α ise uzaklıklar arasındaki farkı ayarlamak için kullanılır. Görüldüğü gibi, yitim ikisi arasındaki uzaklığı α değerinden büyük tutmayı hedeflemektedir.

$$d(\mathbf{p}, \mathbf{q}) = \frac{\sum_{i=1}^n |p_i - q_i|}{n} \quad (2.5)$$

$$L_{triplet} = \max(0, d(\mathbf{p}, \mathbf{a}) - d(\mathbf{n}, \mathbf{a}) + \alpha) \quad (2.6)$$

Bu amaçla, önceden ImageNet ile eğitilmiş MobileNetV2 modeli, sınıflandırma katmanları çıkarıldıktan sonra 1x1280 boyutunda öznitelik vektörleri elde edilecek şekilde eğitilmiştir.

2.4.2 Arka Plan Silme

Oyuncu tanımlama yönteminde çeşitli oyuncu görüntülerinde düşük performans alındığı gözlemlenmiştir. Bu görüntüler genellikle aynı oyuncunun arka planı birbirinden farklı olan görüntüleridir. Bu nedenle, arka planın çıkarılması için çeşitli yöntemler test edilmiştir. Öncelikle, geleneksel arka plan çıkarımı yöntemleri denenmiştir, ancak şutör tespitinin sadece atışlardan sonra yapılması gerekliliğiyle birlikte tek görüntü üzerinde çalıştırılabilecek bir yöntem tercih edilmiştir. Bu yöntem, önceden eğitilmiş DeepLabV3+ [42] modelinin kullanımındadır.

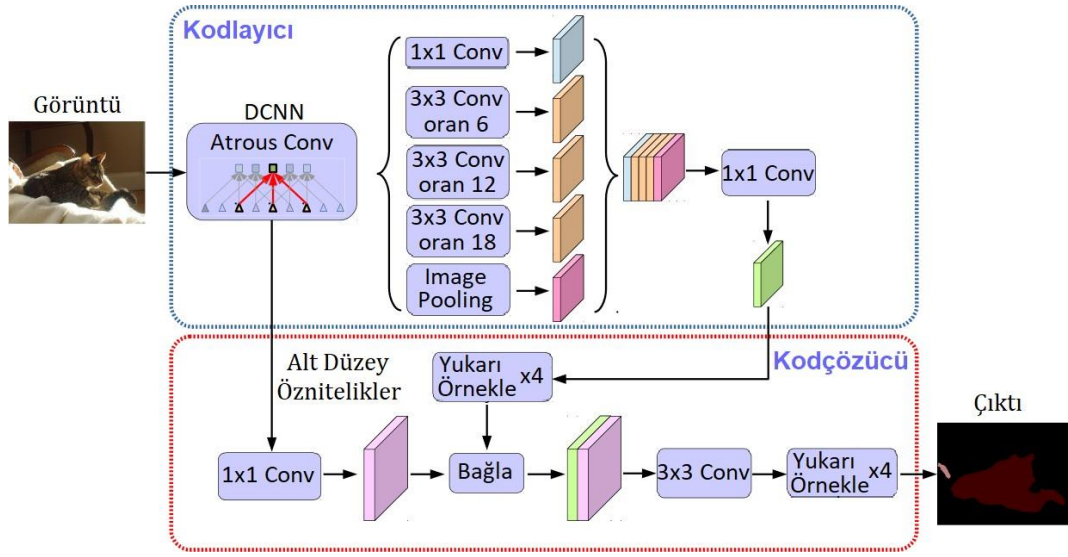
DeepLabV3+, kodlayıcı-kodçözücü (encoder-decoder) ağı olarak bölütleme işlemi gerçekleştirmektedir. Daha önceki uygulama olan DeepLabV3 [43] üzerine özel bir kodçözücü mimarisi eklenerek daha yüksek performans elde edilmesi sağlanmıştır.

DeepLabV3 ve DeepLabV3+'ta Atrous Convolution adı verilen evrisim yöntemi kullanılarak mimaride derinlere gidildiğinde oluşan görüntü çözünürlüğü azalmasının önüne geçilmiştir. Bu yöntem 2.7'deki denklemlerle açıklanabilir. Denklemde, x girdiyi, y çıktıyı, ω filtreyi ve r evrisimin oranını göstermektedir. Mimaride derinlere inildikçe r 'nin artırılması ile görüntü çözünürlüğü istenilen düzeyde tutulurken uzun mesafe özelliklerinin çıkarımı mümkün olmaktadır.

$$y[i] = \sum_k x[i + rk] \omega[k] \quad (2.7)$$

Aynı zamanda, DeepLabV3+ modelinin DeepLabV3'ye göre daha hızlı çalışabilmesi için MobileNetV2'ye benzer olarak Derinlemesine Ayrılabilir Evrisim kullanan Xception [44] mimarisi temel ağ olarak kullanılmıştır. Atrous Convolution işleminin Derinlemesine Ayrılabilir Evrisim üzerine eklenmesiyle Atrous Seperable Convolution adı verilen katmanlar elde edilmiştir [42].

DeepLabV3+ mimarisi Şekil 2.15'te verilmiştir. Görüldüğü gibi, kodçözücü olarak DeepLabV3'de kullanılan Çift Doğrusal Yukarı Örnekleme (Bilinear Upsampling, BU) metodu yerine kodçözücüde öncelikle 4 faktörlü BU kullanılır ve elde edilen çıktı



Şekil 2.15: Kodlayıcı ve kodçözücünden oluşan DeepLabV3+ mimarisi. Kodlayıcı olarak modifiye edilmiş, DeepLabV3 kullanılırken, özel bir kodçözücü kullanılmıştır [42].

kodlayıcı modülünden çıkan düşük seviyeli özneteliklerle birleştirilir. Ardından, bu özneteliklerin boyutunu düşürmek için evrisim katmanı uygulandıktan sonra tekrar 4 faktörlü BU ile çıktı elde edilir.

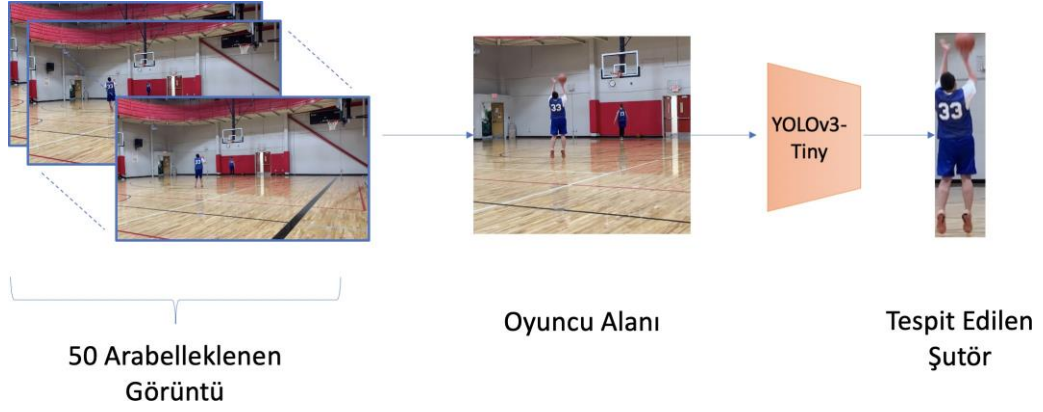
DeepLabV3+ modeli PASCAL VOC 2012 test veri setinde %89'a varan performans göstermektedir [42]. Bu tezde DeepLabV3+ modeli PASCAL 2012 veri setinde önceden eğitilmiş olarak kullanılmıştır.

2.4.3 Şutör Tespiti

Şutör tespiti için, atış tespit edildikten sonra atış tespit algoritmasına benzer şekilde arka plan çıkarımı yapılmaktadır. Farklı olarak, arka plan çıkarımı 50 arabelleklenen görüntü üzerinde yapılır. Burada, arabelleklenen görüntüler üzerinde topun atış, tespiti görüntüsünden itibaren takip edilmesiyle zamanda geriye doğru gidilir. Bu işlem görüntülerde topa yakın bir alanda büyük hareketlilik olduğu ana kadar devam eder. Hareketlilik tespit edildiğinde, görüntüden 768x768 boyutunda bir parça kırılarak YOLOv3-tiny ile kişi tespiti çalıştırılır. Tespit edilen kişiler arasında topa en yakın olan oyuncu şutör olarak kabul edilir (Şekil 2.16).

2.4.4 Oyuncu Tanımlama

Şutörü her atış için ayrı ayrı tanımlamak amacıyla bir güdümsüz öğrenme modeli geliştirilmiştir. Genelde, insan tanımlama uygulamalarında kullanılan metodlar yüksek



Şekil 2.16: Şütör tespit yöntemi. Oyuncu alanı tespiti arka plan çıkarımında yeterli hareketlilik bulunduktan hemen sonra yapılır.

çıkartım süresi ve kapsamlı veri setleri gerektirmektedir. Ancak, oyuncu tanımlama algoritması atış tespitiyle birlikte çalışacağı için, MobileNetV2 bazlı bir model kullanılmıştır. Bu modelin kullanımıyla hızlı sonuçlar alınması hedeflenmiştir. MobileNetV2 mimarisi her oyuncu görüntüsünden öznitelik çıkarımı amacıyla Siyam Ağı olarak kullanılmıştır (Şekil 2.17). Şütör görüntüsü modele girdi olarak verildikten sonra, çıktı olarak alınan öznitelik vektörü diğer oyuncuların alınan öznitelik vektörleri ile Öklid Uzaklığı (2.5) kullanılarak karşılaştırılır ve birbirlerine olan uzaklığı düşük olan vektörlerin aynı oyuncuya ait olduğu çıkarımı yapılır. Algoritmanın güdümsüz olması ise daha önce modelin eğitilmediği oyuncuların karşılaştırılmasına izin vermektedir. Bu amaçla, Öklid Uzaklığı metriğine uygun olarak Üçlü Yitim (Triplet Loss, TL) [40] kullanılmıştır. TL ile eğitme verileri ve test verileri tamamiyle farklı oyuncuların oluşabilmektedir.

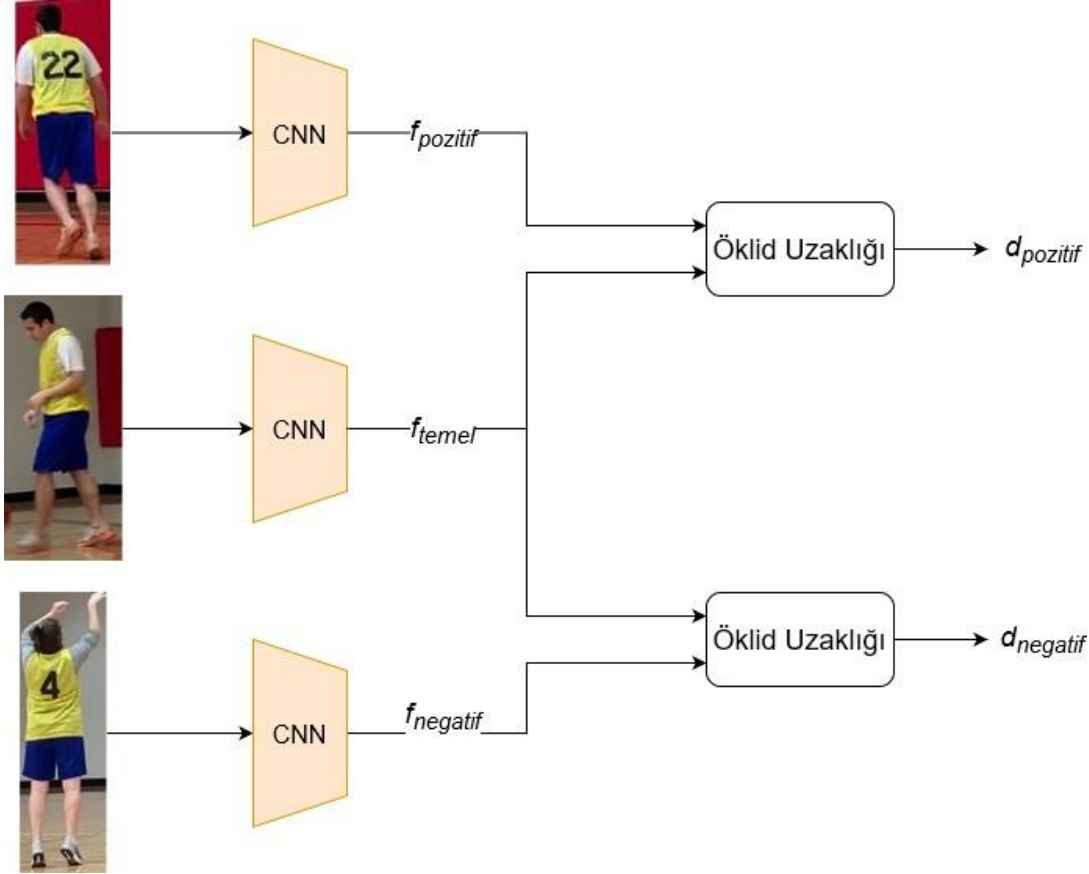
Bazı durumlarda, oyuncu görüntülerindeki arka plan değişimlerinin tanımlama işlemine olumsuz etki ettiği gözlemlenmiştir. Bu nedenle, oyuncuları arka plandan ayırmak amacıyla bir görüntü bölütleme ağı olan DeepLabV3+ kullanılmıştır. Bölütlenen oyuncu görüntüleri daha sonra MobileNetV2 bazlı siyam ağına eğitim verisi olarak verilmiştir. Eklenen bu model doğruluk oranını yüksek ölçüde artırmaktadır. DeepLabV3+'ın hesaplama süresi kısa olduğundan gerçek zamanlı uygulamada MobileNetV2 ile kullanımı mümkündür.

Atış, sınıflandırmasını da içeren komple uygulamada; oyuncu tanımlama, atış, tespitinden sonra şütörün saptanması ve şütörün saptanmış oyuncunun hanesine yazılması amacıyla çalışmaktadır.

Oyun seansı boyunca ayrı, tırma gerçek zamanlı olarak yapılır. Burada, oyuncuların ana öznitelik vektörleri, şütör görüntüsü üzerinden elde edilir. Her şütörün sonra elde edilen oyuncu vektörleri ana özniteliklerle karşılaştırıldıktan sonra, en yakın vektörler aynı

oyuncuya ait kabul edilir.

Ek olarak, seans sonunda elde edilen tüm vektörler tespit edilmiş, oyuncu sayısı göz önüne alınarak kümeleme işlemi kullanılmaktadır. Burada kümeleme işlemi için K-Means algoritmasından faydalanılmıştır. Bu işlemle birlikte seans sonunda elde edilen doğruluk oranı artırılmak hedeflenmiştir. K-Means algoritması k değerine ihtiyaç duyduğu için, seanstaki oyuncu sayısının doğruluğu büyük önem taşımaktadır. Bu nedenle, oyuncu sayısının model tarafından hesaplanmak yerine, bilinen bir parametre olduğu varsayılmıştır.



Şekil 2.17: Oyuncu tanımlama algoritmasının genel işleyişi

2.5 Kullanılan Başarım Metriği

Çalışmanın her aşaması ayrı ayrı inceleneceği için farklı başarımleri kullanımı gerekmektedir. Özellikle, atış alanının bulunabilmesi için nesne tespiti modeli kullanılmıştır. Bu nedenle, Intersection Over Union (IOU) metriği kullanılmıştır. IOU 2.8'deki gibi tanımlanabilir. Denklem payı, tahmin kutusu (BB) ile kesin referans (ground truth) (GT) kesişimiyken paydası ise iki kutunun birleşimidir. Bu metrik, modelin genel performansını ölçmede yarar sağlasa da, tezde kullanılan komple algoritmadaki doğruluğu

gösterme konusunda yanıltıcı olabilmektedir.

$$IOU = \frac{BB \cap GT}{BB \cup GT} \quad (2.8)$$

Çalışmanın diğer aşamaları, sınıflandırma modelleriyle gerçekleştirilmiştir. Bu nedenle, doğruluk ve duyarlılık metrikleri kullanılmıştır. Doğruluk (2.9) ve duyarlılık (2.10) ile hesaplanır. Bu denklemlerde DP doğru pozitif, YP yanlış pozitif, DN doğru negatif, YN ise yanlış, negatif göstermektedir.

$$\frac{DP + DN}{DP + YP + DN + YN} \quad (2.9)$$

$$\frac{DP}{DP + YN} \quad (2.10)$$



3. DENEY

Bu bölümde deney kısmında kullanılan veriler ve deney çıkarımları açıklanacaktır.

3.1 Veri Seti

3.1.1 Atış Alanı Tespiti

Atış alanı tespiti için oluşturulan veri seti 21 farklı konumdan alınan 851 görüntüden oluşmaktadır. Bu görüntülerden 15 konumdan alınan 651 tanesi sinir ağının eğitiminde, 6 konumdan elde edilen 200'ü ise ağın testinde kullanılmıştır.

3.1.2 Atış Tespiti

Atış tespiti algoritmasında, veri setine sadece basketbol topu bulma modelinde ihtiyaç duyulmaktadır. Bu nedenle, top ve arkaplan görüntülerini ayırt etmek için yaklaşık 8000 top ve arka plan görüntüsü oluşturulmuştur. Bu görüntülerden yaklaşık 7000'i sınıflandırma modelinin eğitiminde, 1231 tanesi ise testinde kullanılmıştır. Veri setinden örnekler Şekil 3.1'de verilmiştir.

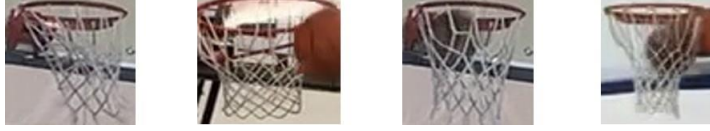


Şekil 3.1: Basketbol topu onaylama modelinde kullanılan görüntüler. Top görüntüleri solda, arka plan görüntüleri sağda

3.1.3 Skor Tanımlama

Skor tanımlama için basketbol filesi görüntüleri kullanılmıştır. Bu görüntüler, atış alanı ve atış tespitinde kullanılan videolardan elde edilmiştir. Model eğitiminde kullanılan veri seti, 2536 basketbol ve 2600 arka plan gösteren 5136 toplam görüntüden oluşmaktadır. Ek olarak, test için 634 basketbol ve 650 arka plan görüntüden oluşan 1284 görüntü oluşturulmuştur. Veri setinden örnekler Şekil 3.2'de verilmiştir.

Daha önce de belirtildiği gibi, bu kısımda da model eğitiminin verimliliğini artırmak amacıyla çeşitli zenginleştirme yöntemleri rastgele uygulanmıştır. Bu yöntemler içerisinde,

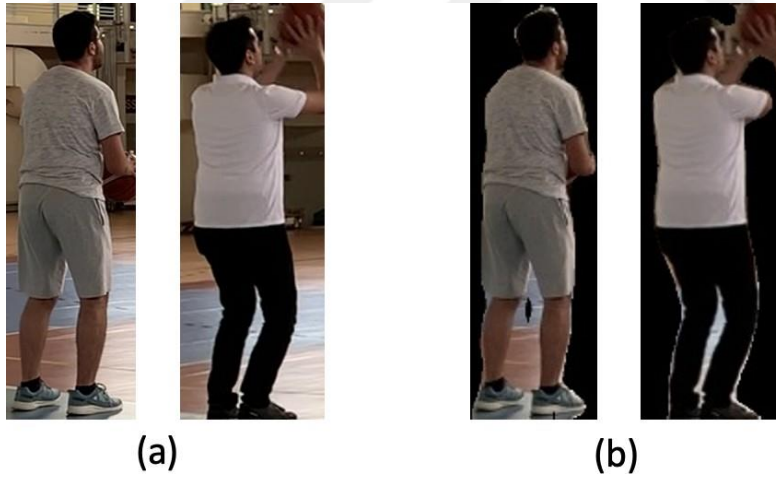


Şekil 3.2: Skor tanımlaması veri setinden örnekler. İska görüntüleri solda, başarı görüntüleri sağda

yatay çevirme, kanal karıştırma, boyut oranı değiştirme ve tuz-biber görüntüsü ekleme sayılabilir.

3.1.4 Oyuncu Tanımlama

Oyuncu tanımlama için 22 farklı oyuncu ve 5214 görüntüden oluşan bir veri seti oluşturulmuştur. Bu oyuncular arasında aynı renkte formalar ve farklı giysiler giyenler bulunmaktadır. Veri setinden birkaç örnek Şekil 3.3'te görülebilir. Modelin eğitiminde, 22 oyuncu 14 ve 8 olarak ayrılmış, 14 kişi (4132 görüntü) eğitimde, 8 kişi (1082 görüntü) doğrulamada kullanılmıştır. Kişi bazında görüntü sayısı 30 ile 1000 arasında değişmektedir. Modelin test edilmesi içinse komple algoritmanın testinde kullanılan 2 farklı antrenman videosundan çıkarılan oyuncu görüntülerinden faydalanılmıştır.



Şekil 3.3: Oyuncu tanımlama modelinin eğitiminde kullanılan veri setinden 2 örnek. (a) islenmemiş, görüntüler, (b) arka plan silme yapılmış, görüntüler

1. video (VID-1) aynı formayı giyen iki oyuncudan oluşmaktadır. Oyuncular toplam 519 görüntüye (283+236) sahiptir (Şekil 3.4). 2. video (VID-2) ise 3'ü aynı formayı giyen 4 oyuncudan (154+138+147+105 = 544 görüntü) oluşmaktadır.

VID-2 seti kümeleme işleminin sağlamlığını test etmek amacıyla, VID-1 ile aynı



Şekil 3.4: VID-1'deki oyuncular. (a) işlenmemiş görüntüler, (b) arka plan silme yapılmış, görüntüler

çözünürlüğe sahip olmasına rağmen daha düşük görüntü kalitesine sahip bir kameradan alınmıştır.

3.2 Atış, Sınıflandırması

3.2.1 Atış, Alanı Tespiti

Atış alanı tespit performansı file tespit modelinin performansına bağlıdır. File tespit modeli test veri seti içinde IOU = 0.76 vermektedir. Ancak, bu oran atış, alanı tespit performansını tam olarak göstermemektedir. Zira, file tespiti fileyi içine aldığı sürece elde edilen kutunun büyüklüğü belli bir orana kadar kabul edilebilir olmaktadır. Bu nedenle, atış alanı tespiti doğruluk oranı %91.4'ü bulmaktadır.

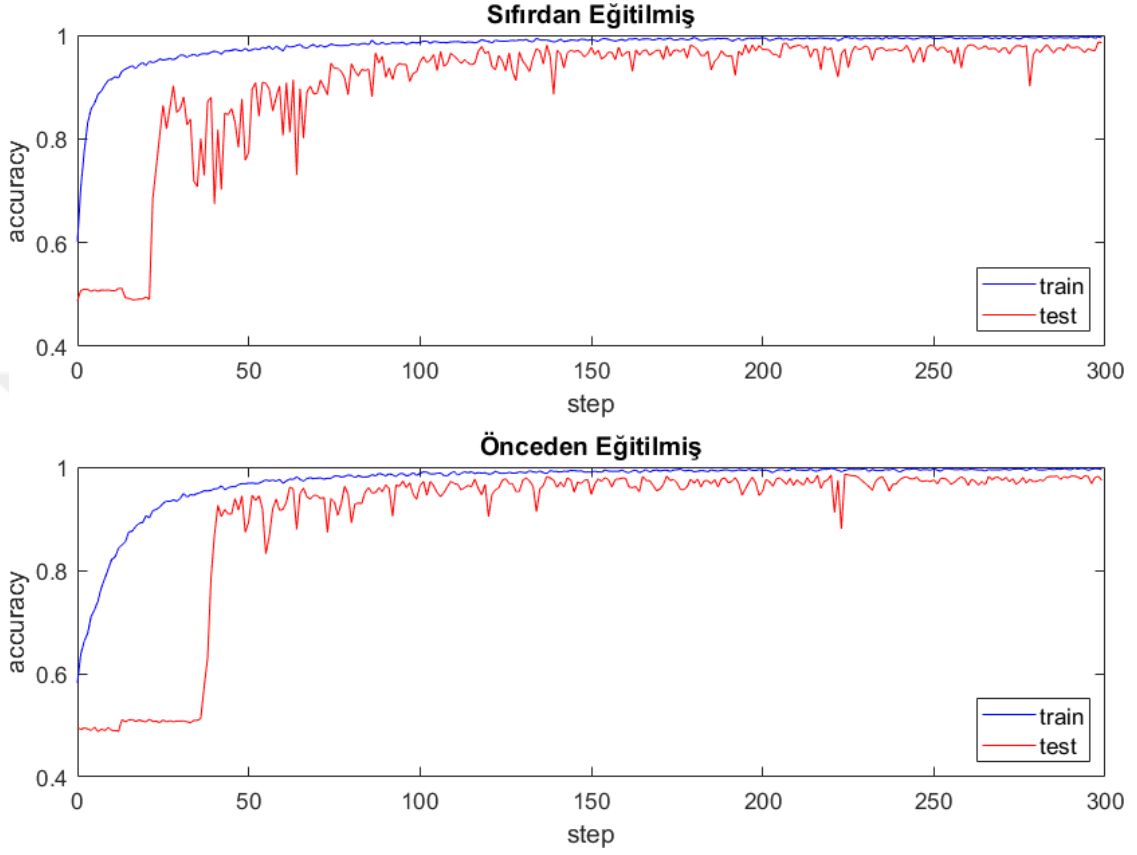
3.2.2 Atış, Tespiti

Kamera belirtilen sınırlar içine koyulduğu sürece atış tespiti algoritmasının arka plan çıkarımı kısmı her aşamada doğru çalışmaktadır. Ancak, burada da top onaylama modelinin çıktısı belirleyici olmaktadır. Bu modelin ortalama doğruluk oranı, kendi başına test seti üzerinde %92'dir. Modelin her atış adayında tekrar çalıştırıldığı düşünüldüğünde ilk videoda (VID-1) %97'ye varan doğruluk oranına ulaşılmaktadır.

3.2.3 Atış, Sınıflandırma

Atış, sınıflandırması için daha önce belirtilen MobileNetV2 sınıflandırma sonucu verecek şekilde eğitilmiştir. Küme büyüklüğü (batch size) olarak 32, eniyileştirici olarak öğrenme hızı 0.0001 olan Adam kullanılmıştır. Bu aşamada, yeterince veri

olmasına rağmen, model önceden ImageNet üzerinde eğitilmiştir. Önceden eğitilmiş ve eğitilmemiş modelin eğitim boyunca doğruluk karşılaştırmaları Şekil 3.5'te verilmiştir. Görüldüğü gibi sonuç olarak aynı doğruluk oranlarına ulaşılsa da daha önce eğitilmiş model daha stabil olmaktadır.



Şekil 3.5: Atış tanımlama model eğitim ve doğrulama setleri doğruluk oranları. Önceden eğitilmiş model altta, sıfırdan eğitilmiş model üstte

3.3 Oyuncu Tanımlama

Oyuncu tanımlama sonuçları iki farklı şekilde incelenmiştir. Öncelikle, model gerçek zamanlı algoritma düşünülerek test edilmiş, ve bazı vektörleri o şekilde seçilmiştir. Bu seçimlerde oyuncu sayısının önceden belirlenmiş olduğu varsayımı yapılmıştır.

Modelin kümelenme yapılmadan elde edilen doğruluk oranı Çizelge 3.1'de verilmiştir. Çizelgede TL kullanılarak eğitilen oyuncu tanımlama modeli aynı veriler ile fakat sınıflandırma yitimiyle eğitilen model ile karşılaştırılmıştır. Görüldüğü gibi, TL, sınıflandırma modelinden çok daha yüksek doğruluk oranına ulaşmaktadır. DeeplabV3+ eklenmesiyle yapılan arka plan silme ise doğruluk oranını gözle görülür ölçüde artırmaktadır.

Çizelge 3.1: Gerçek zamanlı uygulamada oyuncu tanımlama doğruluk oranları.

	VID-1	VID-2
Sınıflandırma	0.63	0.54
TL	0.871	0.623
TL + Arkaplan Silme	0.925	0.763

Oyuncu bazında sonuçlar VID-2 üzerinden elde edilen bütün oyuncular üzerinden incelenmiştir. Şekil 3.6’da görüldüğü gibi, iki model de benzer derecede doğruluk oranına sahiptir. Burada TL + K-Means modeli ortalama %80.6 doğruluk verirken, arka plan silmenin eklenmesiyle bu oran %78.3’e düşmüştür. Ancak, P6 oyuncusunun diğer oyunculara kıyasla farklı renkte forma giydiği düşünülünce DeepLabV3+ ile arka plan silmenin daha stabil bir model oluşturduğu çıkarımı yapılabilir. Bu çıkarım Şekil 3.8’de görülmektedir. Şekil 3.7’de de görülebileceği üzere arka plan silme yapılmadan önceki sonuçlar değişken olurken DeepLabV3+ kullanımıyla doğruluk oranında ve sonuçların kararlılığında artış olmaktadır.

Kümeleme ile yapılan testlerden alınan doğruluk sonuçları Çizelge 3.2’de verilmiştir.

Çizelge 3.2: K-Means kümeleme algoritmasıyla erişilen oyuncu tanımlama doğruluk oranları.

	VID-1	VID-2	VID-1+VID-2
TL+K-Means	0.827	0.806	0.696
TL+DeepLabV3++K-Means	0.983	0.783	0.867

Arka plan silmenin kümeleme işlemine etkisini görselleştirebilmek amacıyla 3.9’da iki farklı modelin 2 boyutlu gösterimi verilmiştir. Şekil 3.9a’da görüldüğü gibi arka plan silme yapılmadan elde edilen öznitelik vektörlerini birbirlerine yakın olmaktadır. Arka plan silme eklendiğinde ise oyuncu vektörleri, kümeleme işlemi kolaylaştıracak biçimde ayrılmıştır (Şekil 3.9b).

Temel ağ performansını karşılaştırmak amacıyla model, MobileNetV2 dışında farklı ağlar kullanılarak aynı parametrelerle tekrar eğitilmiştir. Bu karşılaştırma VID-1’in gerçek zamanlı performansı üzerinden arka plan silme yapılmış, görüntüler ile yapılmıştır. Karşılaştırma sonuçları Çizelge 3.3’te verilmiştir. Karşılaştırma ilk MobileNet [45], MobileNetV2, MobileNetV2x0.75 (parametre sayısı dörtte üçüne indirilmiş MobileNetV2) ve NASNetMobile [46] üzerinde yapılmıştır. Görüldüğü gibi, MobileNetV2 hem en yüksek doğruluk oranına ulaşmakta, hem de düşük parametre sayısı ile daha hızlı sonuç verebilmektedir.

Sistemin genel performansı ise yine VID-1 üzerinde ölçülmüş, ve Çizelge 3.4’te verilmiştir. Daha önce de bahsedildiği gibi, top onaylama modeli birden fazla kez

Çizelge 3.3: Farklı temel ağların VID-1 üzerinde gerçek zamanlı doğruluk performanslarının karşılaştırılması.

	Doğruluk Oranı	Parametre Sayısı
MobileNet	0.924	3.23M
MobileNetV2	0.925	2.26M
MobileNetV2x0.75	0.782	1.38M
NASNetMobile	0.795	4.27M

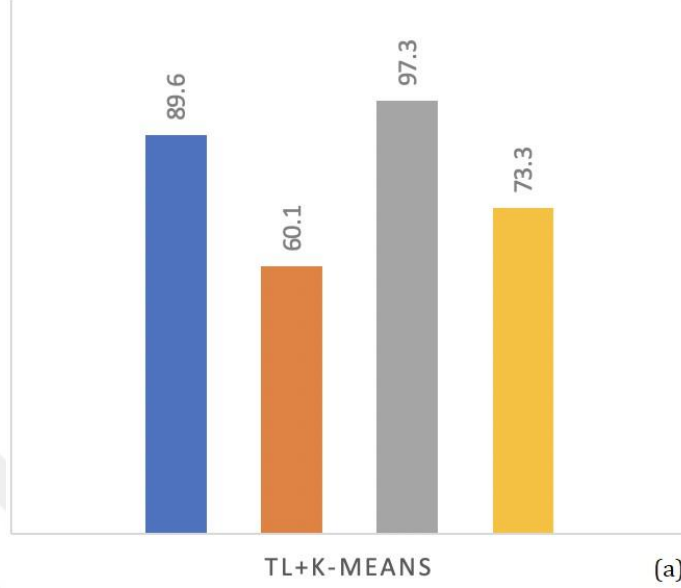
çalıştırıldığı ve sadece tek bir pozitif yeterli olduğu için performansı yüksek olmaktadır.

Çizelge 3.4: Sistemdeki tüm bileşenlerin VID-1 üzerindeki doğruluk oranları.

	Bileşen
Atıs, Alanı Tespiti	0.914
Atıs, Tespiti	0.97
Atıs, Sınıflandırma	0.95
Oyuncu Tanımlama	0.925

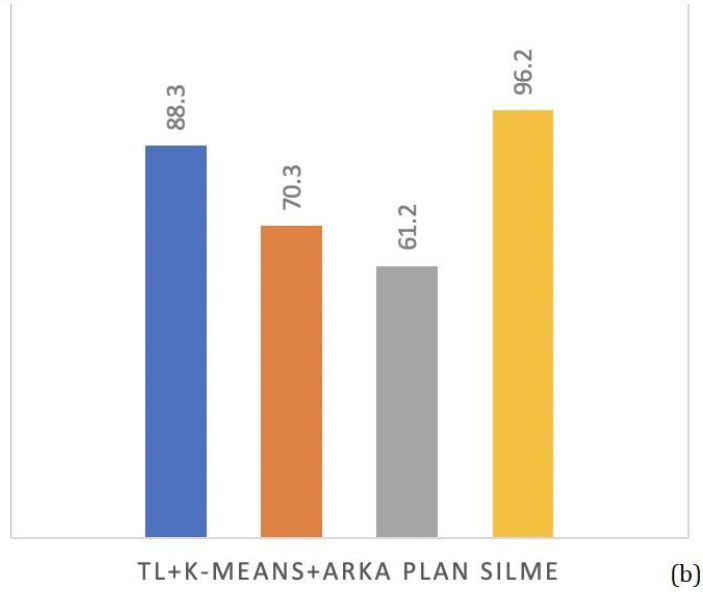
VID-2

■ P3 ■ P4 ■ P5 ■ P6

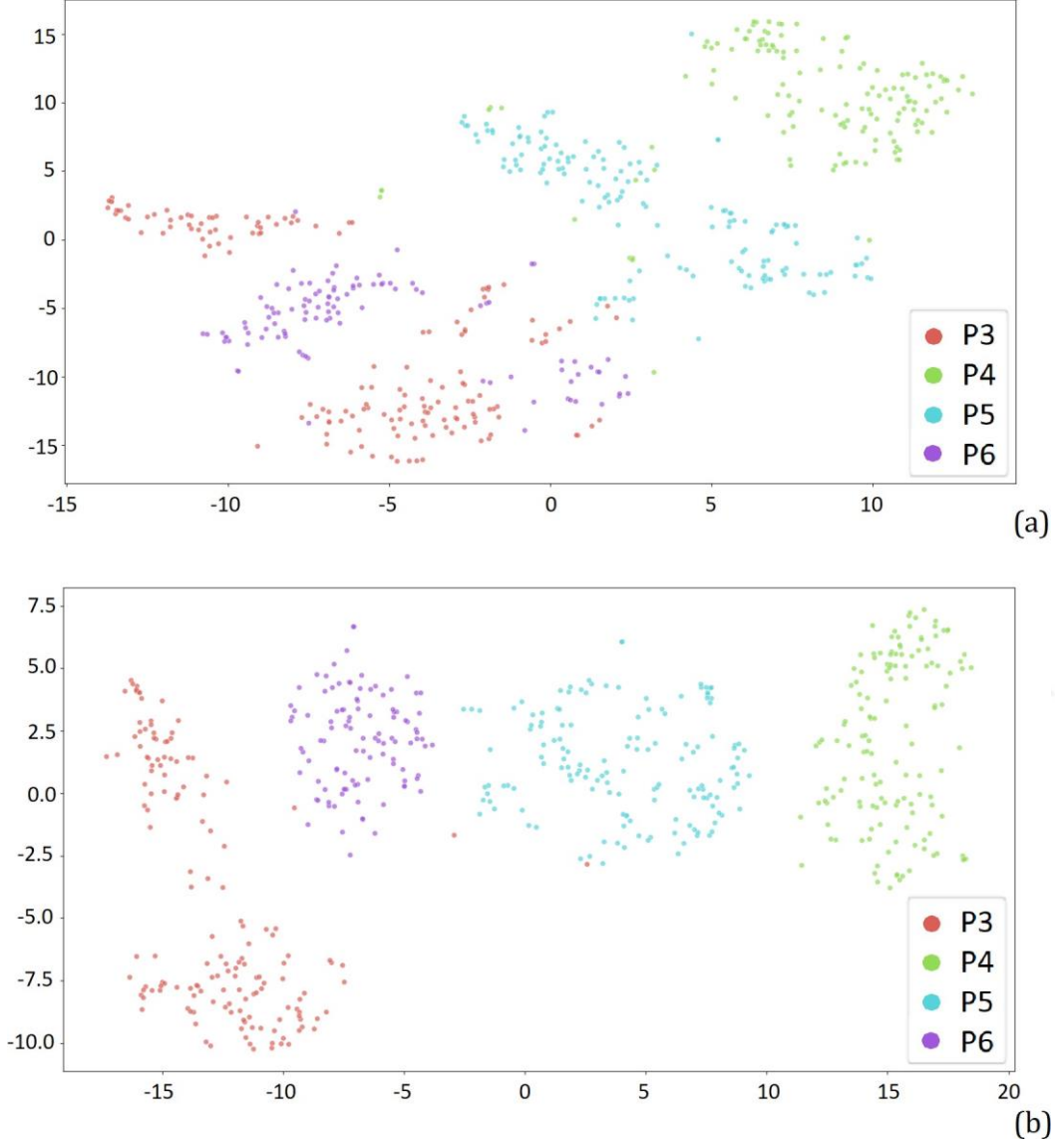


VID-2

■ P3 ■ P4 ■ P5 ■ P6



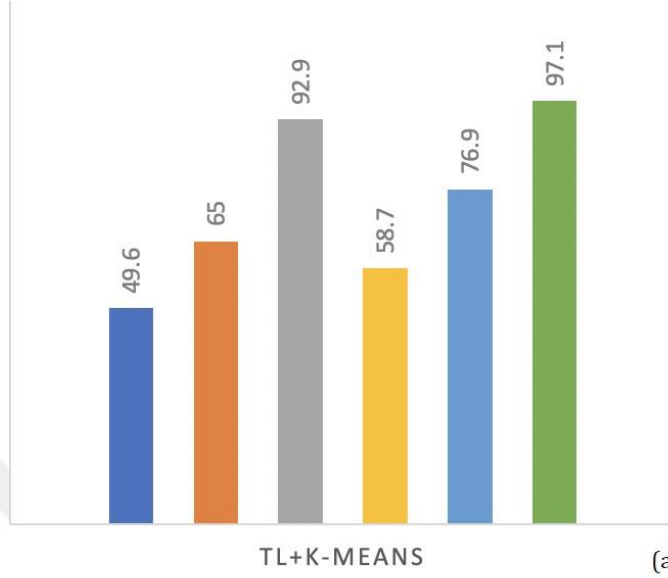
Şekil 3.6: Oyuncu tanımlama algoritmasının oyuncu bazında farklı videolar üzerinde K-Means ile elde edilen doğruluk sonuçları. (a)DeepLabV3+ kullanılmadan alınan sonuçlar, (b)DeepLabV3+ kullanımı ile arka plan silme yapılan görüntülerden alınan sonuçlar



Şekil 3.7: VID-2 görüntüleri üzerinde elde edilen vektörlerin 2 boyutlu gösterimi. (a) TL+K-Means, (b) TL+DeepLabV3++K-Means

VID-1 + VID-2

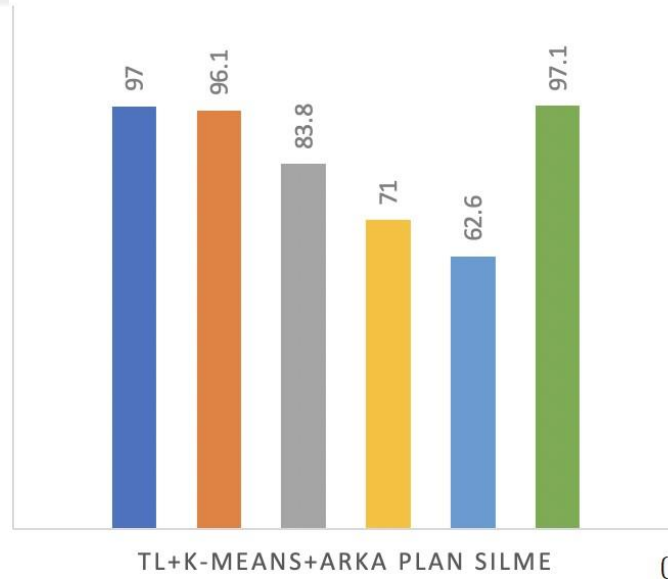
■ P1 ■ P2 ■ P3 ■ P4 ■ P5 ■ P6



(a)

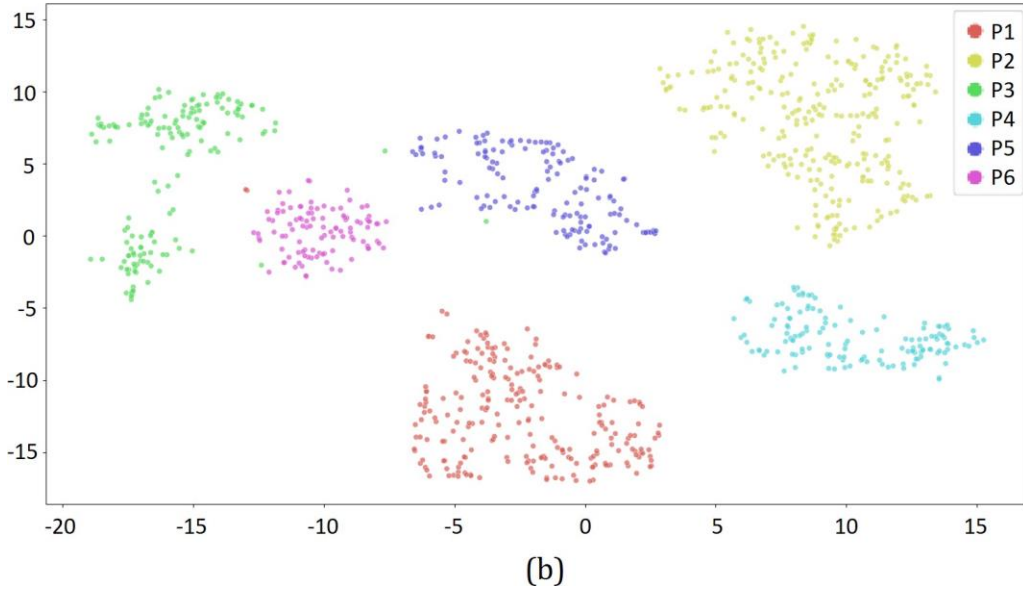
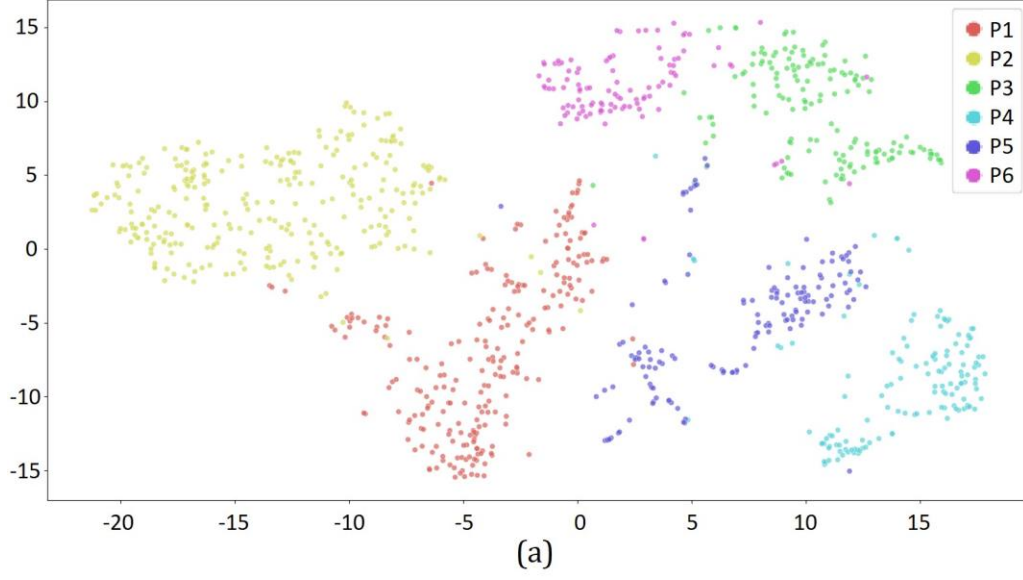
VID-1 + VID-2

■ P1 ■ P2 ■ P3 ■ P4 ■ P5 ■ P6



(b)

Şekil 3.8: Oyuncu tanımlama algoritmasının oyuncu bazında tüm test seti üzerinde K-Means + Üçlü Yitim ile elde edilen doğruluk sonuçları. (a)DeepLabV3+ kullanılmadan alınan sonuçlar, (b)DeepLabV3+ kullanımı ile arka plan silme yapılan görüntülerden alınan sonuçlar



Şekil 3.9: Modellerden elde edilen öznelik vektörlerinin 2 boyutlu gösterimi. (a) TL+K-Means, (b) TL+DeepLabV3++K-Means

4. SONUÇ

Bu tezde, gerçek zamanlı akıllı bir basketbol antrenman uygulaması geliştirilmesi amaçlanmıştır. Daha önceki yöntemler incelendiğinde, gerçek zamanlı metodların derin öğrenme uygulamalarından yararlanmayarak geleneksel yöntemler kullandığı gözlemlenmiştir. Ancak, derin öğrenme kullanılan uygulamaların geleneksel olanlara kıyasla çok daha yüksek doğruluk oranı verdiği sonucuna ulaşılmıştır. Bu nedenle, komple algoritmanın her aşamasında derin öğrenme yöntemlerinden yararlanılmıştır.

Uygulama, her bir oyuncunun atış, sayısını ve bu atışlardaki başarı oranlarını hesaplamayı amaçlamaktadır. Basketbol antrenmanı videosu üzerinde atış, tespiti ve oyuncu tanımlama olmak üzere 2 farklı kısımdan oluşmaktadır. Gerçek zamanlı çalışma hedefini gerçekleştirmek için derin öğrenme modelleri olarak düşük çıkarım süreli mobil ağlar temel ağ olarak tercih edilmişlerdir.

Atış tespitinin gerçek zamanlı olarak yapılabilmesi için derin öğrenme ile yapılan tespit işlemi sürekli yapılmamaktadır. Bu nedenle, atış, tespiti için bir alan belirlenmiş, ve bu alan üzerinde arka plan çıkarımı ile hareket bulunmuştur. Alanın çıkarımı MobileNetV2 ve SSD bazlı bir DL ağı ile sağlanmış ve veri setindeki yetersizliğin önüne geçmek için stil transferi kullanılmıştır. Atış, alanı tespitinden sonra arka plan çıkarımı ile elde edilen hareketler ise yine DL ile onaylanarak yanlış, pozitiflerin önüne geçilmiştir. Skor tanımlama içinse MobileNetV2 sınıflandırma modeli olarak kullanılmıştır. Modelin her atış, tespitinden sonra 40 görüntü üzerinde birer birer çalıştırılması ve sınıflandırma sonuçlarının esiklenmesiyle sonuca ulaşılır. İlk deneylerde, atış, sınıflandırması modelinin gürültülü ortamlarda düşük performans gösterdiği gözlemlenmiştir. Bunun önüne geçmek amacıyla skor tanımlama veri setine rastgele hareket bulanıklığı eklenmiştir.

Oyuncu tanımlama algoritması için, her bir atıştan sonra bulunan şutör görüntüsü üzerinde MobileNetV2 bazlı bir öznetelik çıkarıcı kullanılmıştır. Bu çıkarıcı farklı yitim fonksiyonları ve ön işleme metodlarıyla denenmiş, TL yitim fonksiyonu ve DeepLabV3+ arka plan çıkarımının model performansına yüksek katkıda bulunduğu gözlemlenmiştir. Oyuncu tanımlama test edilirken aynı formayı giyen 2 oyuncudan oluşan bir video ile 3'ü aynı 1'i farklı formayı giyen 4 oyuncudan oluşan bir diğer video kullanılmıştır.

Gerçek zamanlı çalışabilen komple algoritma, oyuncu görüntülerinin engellenmesine ve görüntü kalitesine karşı hassas olsa da genel olarak yüksek doğruluk oranı vermektedir. Doğruluk oranı iki farklı yöntemde hesaplanmıştır. Bunlardan ilki olan gerçek zamanlı yöntemde Üçlü Yitimli model ilk videoda %92.5, ikinci videoda ise %76.3 doğruluk

oranına ulaşmaktadır. Diğer yöntemde ise bir kümeleme uygulaması olan K-Means kullanılmıştır. Böylece, ilk videoda %98.3, ikinci videoda %78.3, toplam oyunculara ise %86.7 doğruluk oranları elde edilmiştir.

Yapılan çalışma aşağıda belirtilen yöntemlerle daha da ileriye taşınabilir.

- Atış alanı tespiti için üretilen veri setinin büyüklüğü artırılarak modelin farklı koşullarda da yüksek performans vermesi sağlanabilir.
- Skor tanımlaması için kullanılan sınıflandırma metoduna ek olarak [7]'deki metod çoklu girdi alan tek bir mobil ağda birleştirilebilir.
- Oyuncu görüntülerinin başka oyuncular ve objeler tarafından örtülmesi sorununu önlemek için Rastgele Silme Veri Artırma (Random Erasing Augmentation, REA) [47, 48] kullanılabilir.
- Oyuncu tanımlama metodu görüntü kalitesine duyarlı olduğu için çoklu ölçekli bir model oluşturulabilir. Ayrıca, Süper Çözünürlük (Super Resolution) modellerine benzer bir yapı kullanılarak oyuncu görüntülerinin çözünürlük olarak birbirlerine yaklaşması sağlanabilir. Aynı şekilde, oyuncu tanımlama için CPM benzeri bir yöntem kullanılarak parça modeli oluşturulabilir.
- Oyuncu tanımlama metodunun ayrıştırma performansını artırmak için model eğitiminde [49]'da tanıtılan Quadruplet Loss (Dörtlü Yitim) ve normalizasyon yöntemi kullanılabilir.
- Kisi tespiti modeli sürekli olarak çalıştırılarak oyuncu tanımlama modelini Kalman filtre ile güçlendirilebilir. YOLOv3-tiny yerine daha hızlı ve doğruluğu daha yüksek bir mimari olan YOLOv4-tiny [50] kullanılarak gerçek zamanlı kisi tespiti tüm çerçeve üzerinde yapılabilir; bu sayede Kalman filtre kullanımı ile oyuncu tanımlama algoritması doğruluğu artırılabilir. Aynı model top tespiti de yapacak şekilde eğitilirse tek bir model üzerinden hem şutör tespiti hem de oyuncu tespiti yapılabilir.
- Son olarak, uygulamada oyuncuların temel vektörlerinin ilk atışlarından sonra oluşturulması ve sabit kalması yerine, tüm oyuncular tespit edildikten sonrasında her atış için K-Means çalıştırılarak her oyuncu için küme merkezi temel vektör olarak alınabilir. Bu sayede, başlangıçta oluşturulabilecek yanlış pozitiflerin önüne geçilebilir.

KAYNAKLAR

- [1] **Chauhan, D., Patel, N. M., and Joshi, M.** (2016). Automatic summarization of basketball sport video. In: *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pp. 670–673.
- [2] **Lee, G., Kim, H., and Kim, W.** (2009). Highlight generation for basketball video using probabilistic excitement. In: *2009 IEEE International Conference on Multimedia and Expo*, pp. 318–321.
- [3] **Senocak, A.** et al. (June 2018). Part-Based Player Identification Using Deep Convolutional Representation and Multi-Scale Pooling. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- [4] **Lu, C.** et al. (2013a). Identification and Tracking of Players in Sport Videos. In: *Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service. ICIMCS '13*. Huangshan, China: Association for Computing Machinery, pp. 113–116.
- [5] **Yamamoto, T.** et al. (2013). Multiple players tracking and identification using group detection and player number recognition in sports video. In: *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, pp. 2442–2446.
- [6] **Lu, W.** et al. (2013b). Learning to Track and Identify Players from Broadcast Sports Videos. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.7, pp. 1704–1716.
- [7] **Liu, W.** et al. (2017). Deep learning based basketball video analysis for intelligent arena application. In: *Multimed Tools Appl* 76, pp. 24983–25001.
- [8] **Ratgeber, L.** et al. (2019). Video Mining in Basketball Shot and Game Analysis. In: *Acta Polytechnica Hungarica* 16.1, pp. 7–27.
- [9] **Vapnik, V. and Cortes, C.** (1995). Support-vector networks. In: *Machine Learning* 20 (3), pp. 273–297.
- [10] **Dalal, N. and Triggs, B.** (2005). Histograms of oriented gradients for human detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1, 886–893 vol. 1.
- [11] **Viola, P. and Jones, M.** (2001). Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the 2001 IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition. CVPR 2001. Vol. 1, pp. I–I.

- [12] **Freund, Y. and Schapire, R. E.** (Aug. 1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. In: *J. Comput. Syst. Sci.* 55.1, pp. 119–139.
- [13] **Krizhevsky, A., Sutskever, I., and Hinton, G. E.** (2012). ImageNet Classification with Deep Convolutional Neural Networks. In: *Advances in Neural Information Processing Systems*. Ed. by **Pereira, F.** et al. Vol. 25. Curran Associates, Inc., pp. 1097–1105.
- [14] **Russakovsky, O.** et al. (2015). ImageNet Large Scale Visual Recognition Challenge. In: *Int. J. Comput. Vis.* 115.3, pp. 211–252.
- [15] **Sermanet, P.** et al. (2014). OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. arXiv: 1312.6229 [cs.CV].
- [16] **Chakraborty, B. and Meher, S.** (2013). A real-time trajectory-based ball detection-and-tracking framework for basketball video. In: *Journal of Optics* 42, pp. 156–170.
- [17] **Fischler, M. A. and Elschlager, R. A.** (1973). The Representation and Matching of Pictorial Structures. In: *IEEE Transactions on Computers* C-22.1, pp. 67–92.
- [18] **Liu, Y., Liu, X., and Huang, C.** (2011). A new Method for Shot Identification in Basketball Video. In: *J. Softw.* 6, pp. 1468–1475.
- [19] **Lowe, D. G.** (2004). Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision* 60, pp. 91–110.
- [20] **Ballan, L.** et al. (2007). Soccer Players Identification Based on Visual Local Features. In: *Proceedings of the 6th ACM International Conference on Image and Video Retrieval. CIVR '07*. Amsterdam, The Netherlands: Association for Computing Machinery, pp. 258–265.
- [21] **Mahmood, Z.** et al. (2015). Automatic player detection and identification for sports entertainment applications. In: *Pattern Analysis and Applications* 18 (4), pp. 971–982.
- [22] **Matas, J.** et al. (2004). Robust wide-baseline stereo from maximally stable extremal regions. In: *Image and Vision Computing* 22.10. British Machine Vision Computing 2002, pp. 761–767.
- [23] **Wei, S.** et al. (2016). Convolutional Pose Machines. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4732.
- [24] **Zhang, R.** et al. (June 2020). Multi-Camera Multi-Player Tracking with Deep Player Identification in Sports Video. In: *Pattern Recogn.* 102.C.
- [25] **Gray, D. and Tao, H.** (2008). Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features. In: *Computer Vision – ECCV 2008*. Ed. by

- Forsyth, D., Torr, P., and Zisserman, A.** Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 262–275.
- [26] **Gheissari, N., Sebastian, T. B., and Hartley, R.** (2006). Person Reidentification Using Spatiotemporal Appearance. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2, pp. 1528–1535.
- [27] **Mignon, A. and Jurie, F.** (2012). PCCA: A new approach for distance learning from sparse pairwise constraints. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2666–2672.
- [28] **Yi, D.** et al. (2014). Deep Metric Learning for Person Re-identification. In: *2014 22nd International Conference on Pattern Recognition*, pp. 34–39.
- [29] **Zheng, Z., Zheng, L., and Yang, Y.** (Oct. 2017). Unlabeled Samples Generated by GAN Improve the Person Re-Identification Baseline in Vitro. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
- [30] **Cho, K.** et al. (Oct. 2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1724–1734.
- [31] **Wu, L.** et al. (2019). Where-and-When to Look: Deep Siamese Attention Networks for Video-Based Person Re-Identification. In: *IEEE Transactions on Multimedia* 21.6, pp. 1412–1424.
- [32] **Rumelhart, D. E., Hinton, G. E., and Williams, R. J.** (1986). “Learning Internal Representations by Error Propagation”. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, pp. 318–362.
- [33] **LeCun, Y.** et al. (1990). Handwritten Digit Recognition with a Back-Propagation Network. In: *Advances in Neural Information Processing Systems*. Ed. by **Touretzky, D.** Vol. 2. Morgan-Kaufmann, pp. 396–404.
- [34] **Redmon, J. and Farhadi, A.** (2018). YOLOv3: An Incremental Improvement. arXiv: 1804.02767 [cs.CV].
- [35] **Sandler, M.** et al. (June 2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [36] **Liu, W.** et al. (2016). SSD: Single Shot MultiBox Detector. In: *Lecture Notes in Computer Science*, pp. 21–37.
- [37] **Simonyan, K. and Zisserman, A.** (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv: 1409.1556 [cs.CV].
- [38] **Gatys, L. A., Ecker, A. S., and Bethge, M.** (2015). A Neural Algorithm of Artistic Style. arXiv: 1508.06576 [cs.CV].

- [39] **Deng, J.** et al. (2009). ImageNet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255.
- [40] **Schroff, F., Kalenichenko, D., and Philbin, J.** (2015). FaceNet: A unified embedding for face recognition and clustering. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823.
- [41] **Hermans, A., Beyer, L., and Leibe, B.** (2017). In Defense of the Triplet Loss for Person Re-Identification. arXiv: 1703.07737 [cs.CV].
- [42] **Chen, L.-C.** et al. (Sept. 2018). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*.
- [43] **Chen, L.-C.** et al. (2017a). Rethinking Atrous Convolution for Semantic Image Segmentation. arXiv: 1706.05587 [cs.CV].
- [44] **Chollet, F.** (July 2017). Xception: Deep Learning With Depthwise Separable Convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [45] **Howard, A. G.** et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv: 1704.04861 [cs.CV].
- [46] **Zoph, B.** et al. (2018). Learning Transferable Architectures for Scalable Image Recognition. arXiv: 1707.07012 [cs.CV].
- [47] **Zhong, Z.** et al. (Apr. 2020). Random Erasing Data Augmentation. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.07, pp. 13001–13008.
- [48] **Luo, H.** et al. (2019). Bag of Tricks and a Strong Baseline for Deep Person Re-Identification. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1487–1495.
- [49] **Chen, W.** et al. (July 2017b). Beyond Triplet Loss: A Deep Quadruplet Network for Person Re-identification. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 1320–1329.
- [50] **Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M.** (2020). Scaled-YOLOv4: Scaling Cross Stage Partial Network. arXiv: 2011.08036 [cs.CV].

