T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME KULLANILARAK GÖRÜNTÜLERDEN İNSAN DURUŞ TESPİTİ

YÜKSEK LİSANS TEZİ

Firgat MURADLİ

Enstitü Anabilim Dalı	**:**	**BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ**
Tez Danışmanı	**:**	**Dr. Öğr. Üyesi Serap ÇAKAR**

Temmuz 2021

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME KULLANILARAK GÖRÜNTÜLERDEN İNSAN DURUŞ TESPİTİ

YÜKSEK LİSANS TEZİ

Firgat MURADLI

Enstitü Anabilim Dalı	**:**	**BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ**
Tez Danışmanı	**:**	**Dr. Öğr. Üyesi Serap ÇAKAR**

Bu tez 06.07.2021 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

Jüri Başkanı	**Üye**	**Üye**

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Firgat MURADLI

TEŞEKKÜR

Yüksek lisans eğitimim boyunca ve tez çalışmamın her aşamasında yönlendirmeleri, tez konumu belirlemede ve çalışmalarım sırasında fikirlerini ve bilgilerini paylaşan ve tavsiyeleri ile bana ışık tutan değerli danışman hocam Dr. Öğr. Üyesi Serap ÇAKAR'a sonsuz teşekkürler.

Ayrıca yaşamım boyunca arkamda duran, eğitim hayatımda kendilerinden aldığım destek ile bir adım ileriye gitmekte güç bulduğum babam Samad, annem Vefa, abim Aqil ve teyzem Gülane olmakla tüm aile bireylerime özel teşekkürü bir borç bilirim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

2D	: Two Dimensional-İki Boyutlu
3D	: Three Dimensional-Üç Boyutlu
ANN	: Artificial Neural Networks-Yapay Sinir Ağları
BatchNorm	: Batch Normalization-Toplu Normalleştirme
CNNs	: Convolutional Neural Networks-Evrişimli Sinir Ağları
ConvNet	: Convolutional Neural Network-Evrişimli Sinir Ağları
CPU	: Central Processing Unit-Merkezi işlem birimi
GPU	: Graphics Processing Unit-Grafik İşleme Ünitesi
K	: Çekirdek Boyutu
MAE	: Mean Absolute Error-Ortalama Mutlak Hata
MPİİ	: Max Planck İnstitut İnformatik-Veri Seti
N	: Çıktıların Sayısı
P ve W_{in}	: Sıfır Doldurma
PCK	: Probability Of Correct Key point-Doğru Anahtar Nokta Olasılığı
ReLU	: Rectified Linear Unit-Doğrultulmuş Doğrusal Birim
S	: Adım
SMPL	: Skinned Multi-Person Linear Model-Çok Kişili Doğrusa Model
W_{out}	: Çıktının Uzaysal Boyutu
J	: Kayıp Fonksiyonu
α	: Öğrenme Hızı
Θ	: Modelin Parametre Vektörü
y	: Nöronun Çıkışı
$\hat{y}$	: Öngörülen Çıktı

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: CNN, MPII Veri Seti, Keras

Son yıllarda insan pozu tahmini önemli ilerlemeler kaydetmiştir. Bununla birlikte, mevcut veri setleri, genel poz tahmin zorluklarını kapsama açısından sınırlıdır. Yine de bunlar sistemi değerlendirmek ve eğitmek için ortak kaynaklar olarak hizmet etmekte ve üzerinde farklı modeller karşılaştırılabilmektedir. Bu çalışmada Derin Öğrenme kullanılarak insan duruş tespiti gerçekleştirilmiştir. Veri seti olarak, çeşitlilik ve zorluk açısından önemli bir ilerleme sağlayan, insan vücudu modellerindeki gelecekteki gelişmeler için gerekli olabilecek "MPII İnsan Duruşu" kullanılmıştır.

Derin öğrenme modelleri, birçok bilim ve mühendislik alanında yaygın olarak kullanılmaktadır ve yüksek performans seviyelerine ulaşmaktadır. OpenCV ve Keras gibi açık kaynaklı yazılımların yaygınlaşması ile uygulamalarda kullanımı basitleştirilmiştir. Çalışmada açık kaynak kodu olan Opencv, Keras kütüphanesi ve Python programlama kullanılarak derin öğrenme modelleri uygulanmıştır. MPII veri seti kullanılarak derin öğrenme modelleri oluşturulmuştur. Oluşturulan derin öğrenme modeli eğitim ve test veri seti olarak ikiye ayrılmış ve kullanılmıştır. Modelin performansı, test setlerinin doğru sınıflandırma oranı ile ölçülmüştür.

HUMAN POSE DETECTION FROM IMAGES USING DEEP LEARNING

SUMMARY

Keywords: CNN, MPII dataset, Keras

Human pose prediction has made significant progress in recent years. However, the available datasets are limited in terms of covering common exposure estimation challenges. Yet these serve as common resources to evaluate, educate, and compare different models on it. In this article, we introduce a new "MPII Human Pose", a contribution that we think is necessary for future developments in human body models, making a significant advance in diversity and difficulty.

Deep learning models are widely used in many fields of science and engineering and reach high performance levels. With the widespread use of open source software such as Opencv and Keras, its use in applications has been simplified. In the study, deep learning models were applied using open source Opencv, Keras library and Python programming. Deep learning models were created using the MPII data set. The created deep learning model was divided into two as training and test data set and used. Training and test data sets will be obtained by using original images. The performance of the model will be measured by the correct classification rate of the test sets

BÖLÜM 1. GİRİŞ

Poz tahmin yöntemleri, karmaşık görünüm modelleri kullanır ve öğrenme algoritmalarına dayanarak eğitim verilerinden model parametrelerini tahmin eder. Bu yaklaşımların performansı büyük ölçüde şunlara bağlıdır: İnsan kıyafetleri, güçlü eklemlenme, kısmi (kendi kendine) tıkanmalar ve görüntü sınırlarında kesilmeyi temsil eden açıklamalı eğitim görüntülerinin mevcudiyeti. Spor sahneleri ve dik duran insanlar gibi özel senaryolar için eğitim setleri bulunmasına rağmen, bu kriterler temsil edilen faaliyetlerin kapsamı ve değişkenliği açısından hala sınırlıdır. Spor sahnesi veri kümeleri tipik olarak yüksek oranda eklemli pozlar içerir, ancak insanlar tipik olarak sıkı spor kıyafetleri giydiğinden görünüm çeşitliliği açısından sınırlıdır. Buna karşılık, "FashionPose" ve "Kolçaklar" gibi veri setleri, çeşitli farklı giyim türleri giyen kişilerin görüntülerini toplamayı amaçlayarak kesişimler ve kesmeleri içerir.

"MPII İnsan Duruşu" veri seti önerilmeden önce insan pozu tahmini için geniş bir zorluk yelpazesini kapsamayı amaçlayan daha temsili bir kıyaslama oluşturmak için hiçbir girişimde bulunulmamıştır. Bu veri seti karşılaştırmalı değerlendirmeler, görünüm değişkenliği ve karmaşıklığı açısından son teknolojiyi önemli ölçüde ilerletir ve 40.000'den fazla insan görüntüsünü içerir. Veri seti interneti veri kaynağı olarak kullanır ve 800'den fazla etkinliğin açıklamalarına dayalı sorguları kullanarak resimler ve resim dizilerini kapsar. Bu, yalnızca farklı etkinlikleri değil, aynı zamanda iç ve dış sahneleri ve farklı görüntüleme koşullarını kapsayarak çeşitli görüntülerle sonuçlanır. Böylece mevcut vücut poz tahmin tekniklerini incelememize ve bireysel başarısızlık şemalarını belirlememize olanak tanır.

1.1. Kaynak Araştırması

İnsan poz tahmini, bilgisayar vizyonu topluluğu için önemli bir araştırma konusudur [1]. Araştırmacılar ağırlıklı olarak, insan bilgisayar etkileşimi, aksiyon tanıma, gözetim, resim anlama, tehdit öngörüsü gibi çeşitli önemli alanlarda önemli uygulamaları sayesinde araştırma yapmışlardır. Uygulama alanlarının çeşitliliği nedeniyle bu alanın tüm yönlerini kapsamak zordur, bu nedenle bu inceleme, tek bir boyutlu görüntüden insan pozu tahmini yöntemlerindeki en önemli katkılara odaklanmaktadır. Modern yöntemler, derin öğrenme modüllerinin farklı mimarilerini kullanarak bazı yaygın veri setlerini eğitmeye, değerlendirmeye ve karşılaştırmaya dayanır. Bu nedenle, insan pozu tahmin etmeye yönelik ilk pratik modellerden başlayarak, bu en etkili yöntemlerin kısa bir analitik incelemesini yapabilmek için çeşitli derin öğrenme yöntemleri kullanılarak farklı çalışmalar yapılmıştır.

İnsanlar pozları insan vücudunun farklı yerlerinin ve konumlarının yerlerine bakarak algılayabilirler. İnsan Pozu Tahminini sorunu insan eklemlerinin yerelleştirilmesi sorunu olarak tanımlandığından, aynı temel kural bilgisayar ortamında da uygulanır. İnsan vücudu basit duruşlardan karmaşık duruşlara kadar değişir. Farklı pozların doğruluğu, vücut parçalarının tek bir görüntüde yer alması ve ışık, giysi, tek bir resimdeki birden fazla insan gibi bazı harici durumlar nedeniyle her zaman basit bir görev değildir ve bu farklı durumları tahmin etmek bazı sistematik süreçlere ihtiyaç duyar. Bu nedenle araştırmacılar tarafından ilginç bir konu olarak görülmektedir.

Literatürde 3B insan poz tahmininin ve ayrıca 3B ve 2B insan poz tahmininin bir arada yapıldığı çalışmalar mevcuttur.

1.1.1. 3B insan poz tahmini yaklaşımı

Tekin ve ark [2], yapmış olduğu çalışmada insanların 3B pozunu kurtarmak için video dizisinin arka arkaya gelen karelerinden hareket bilgilerini kullanmak için verimli bir yaklaşım önerilmiştir. Önceki yaklaşımlar genellikle adayların pozlarını bireysel

çerçevelerde hesaplar ve sonra belirsizlikleri çözmek için bir işlem sonrasında birbirine bağlar. Buna karşılık, sınırlayıcı kutuların uzamsal geçici hacminden merkezi çerçevedeki 3B pozuna doğrudan geri dönüş yapılmıştır. Ayrıca, bu yaklaşımın tam potansiyelini elde edebilmesi ve konunun merkezde kalması için birbirini takip eden çerçevelerdeki hareketi telafi etmenin şart olduğu gösterilmiştir. Çalışmalarında Human 3.6m ve KTH Multiview Football 3B veri setleri kullanılarak belirsizliklerin üstesinden etkin bir şekilde gelinmiş ve insan poz tahmin ölçütlerine göre büyük bir farkla en son teknolojiye ulaşılmıştır.

Pavlokos ve ark [3], yapmış olduğu çalışmada renkli tek bir görüntüden 3B insan poz tahmini sorusu ele alınmıştır. Uçtan uca öğrenme paradigmasının genel başarısına rağmen, en yüksek performanslı yaklaşımlar, 2B ortak yerelleştirme ve 3B pozu geri kazanmak için Çağdaş Ağ (ConvNet) bir sonraki optimizasyon adımından oluşan iki adımlı bir çözüm kullanmışlardır. Çalışmada, mevcut ConvNet yaklaşımlarıyla 3B poz sunumunu kritik bir konu olarak tanımlamışlardır ve bu görev için uçtan uca öğrenmenin değerini doğrulamak için iki önemli katkıda bulunmuşlardır. İlk olarak, konunun etrafında 3 boyutlu alanın hassas bir şekilde ayrıştırılması önerilmiştir ve her bir bağlantı için ses olasılıklarına göre tahmin etmek üzere bir ConNet'i eğitilmiştir. Bu 3B poz için doğal bir temsil oluşturulmuştur ve koordinatların doğrudan gerilemesine göre performans büyük ölçüde artırılmıştır. İkinci olarak, ilk tahminlerden daha da ilerlemek için, kaba-ince tahmin sistemi kullanılmışlardır. Bu adım çok boyutluluk artışını ele alır ve görüntü özelliklerinin tekrarlanan şekilde düzeltilmesini ve tekrardan işlenmesini sağlar. Önerilen yaklaşım, ortalama %30'dan fazla bir göreceli hata azalması elde ederek standart kıyaslamalarda en son teknolojiye sahip tüm yöntemleri aşmaktadır. Buna ek olarak, uçtan uca yaklaşıma göre optimum olmayan ilgili bir mimaride hacimsel temsilleri kullanarak araştırma yapılmıştır.

Tung ve ark [4], tarafından yapılan çalışmada, tek kamera girişi için öğrenme tabanlı bir hareket yakalama modeli önerilmiştir. Tek bir kamera videoda yapılan hareket yakalama için güncel son teknoloji çözümleri optimizasyon odaklıdır: 3B insan modelinin parametrelerini, projeksiyonunun videoda yapılan ölçümlerle eşleşmesi için optimize ederler (örn. kişi segmentasyonu, optik akış, anahtar noktası algılama vb.).

Optimizasyon modelleri yerel minimuma duyarlıdır. Bu darboğaz, yakalama sırasında arka planlar gibi temiz yeşil ekran, manuel başlatma veya giriş kaynağı olarak birden fazla kameraya geçiş gibi zorunlu kılınan darboğazdır. Model, kafes ve iskelet parametrelerini doğrudan optimize etmek yerine, tek bir RGB videoya sahip 3B şekil ve iskelet konfigürasyonlarını tahmin eden yapay ağ ağırlıklarını optimize eder. Model, sentetik verilerden güçlü bir denetim ve iskelet anahtar noktalarının farklı bir şekilde işlenmesinden, yoğun 3B şebeke hareketinden ve insan arka plan segmentasyonundan uçtan uca bir çerçevede kendi kendini denetleme kullanılarak eğitilmiştir. Deneysel olarak, modelin her iki gözetimli öğrenme ve test zamanı optimizasyonu bir araya getirdiği gözlemlenmiştir.

Pavlokos ve ark [5], yapmış olduğu çalışmada, tek renkli görüntüden tüm gövde 3B insan poz ve şeklini tahmin etme sorunu ele alınmıştır. Bu, tekrarlanan optimizasyon tabanlı çözümlerin tipik olarak hâkim olduğu bir görevken, ConvNets eğitim verilerinin eksikliği ve düşük çözünürlüklü 3B tahminleri nedeniyle zarar görmüştür. Bu boşluğu kapatmayı hedefleyen çalışmalarında, ConNets'e dayalı etkin ve etkili bir doğrudan tahmin yöntemi önerilmektedir. Yaklaşımlarının temel kısmı, uçtan uca çerçevelere parametrik bir istatistiksel vücut şekli modelinin (SMPL) dahil edilmesidir. Bu sayede çok detaylı 3B kafes sonuçları elde edilmiştir. Ayrıca, sadece çok az sayıda parametre hesaplanması gerekmektedir. Bu da doğrudan ağ tahmini için kolay olmasını sağlar. İlginç bir şekilde, bu parametrelerin sadece 2B anahtar noktaları ve maskelerden güvenilir bir şekilde tahmin edilebileceğini göstermiştir. Bunlar, genel 2B insan analizinin tipik çıktılarıdır. Bu sayede, eğitimde 3B şekilli temel gerçekliği olan görüntülerin mevcut olması gerekliliği azaltılmıştır. Aynı zamanda, farklılığı koruyarak, eğitim zamanında tahmini parametrelerden 3B şebeke üretir ve 3B yüzey optimize edilmiştir. Son olarak 3B kafesini görüntüye yansıtmak için, 2B ek açıklamalarla (yani 2B anahtar noktaları veya maskeler) projeksiyonun tutarlılığını optimize ederek ağın daha da geliştirilmesini sağlayan bir ayrıştırılabilir oluşturucu kullanılmıştır. Önerilen yaklaşım, bu görevdeki önceki temel çizimleri aşarak ve tek renkli görüntüden 3B şeklin doğrudan tahmini için bir çözüm sunmuştur.

Sarafianos ve ark [6], tarafından yapılan çalışmada bir görüntü veya videoda verilen bir insanın 3 boyutlu pozunun tahmin edilmesi sorusu ele alınmıştır. Bu, son zamanlarda bilim camiasından büyük ilgi görmektedir. Bu eğilimin ana nedenleri, mevcut teknolojik gelişmeler tarafından yönlendirilen sürekli artan yeni uygulama yelpazesidir (örneğin, insan-robot etkileşimi, oyun, spor performans analizi). Son yaklaşımlar çeşitli zorluklarla başa çıkmış ve dikkate değer sonuçlar bildirmiş olsa da 3B poz tahmini büyük ölçüde çözülmemiş bir sorun olmaya devam etmektedir. Çünkü gerçek yaşam uygulamaları, mevcut yöntemlerle tam olarak ele alınmayan çeşitli zorluklar getirir. Örneğin dış mekân ortamında birden fazla kişinin 3B pozunu tahmin etmek büyük ölçüde çözülmemiş bir sorun olmaya devam etmektedir. Çalışmalarında, RGB görüntülerden veya görüntü dizilerinden 3B insan pozu tahminindeki son gelişmeler gözden geçirilmiştir. Girdiye (ör. Tek görüntü veya video, monoküler veya çoklu görünüm) dayalı yaklaşımların bir sınıflandırması önerilmiştir ve her durumda yöntemler temel özelliklerine göre sınıflandırılmıştır. Mevcut yeteneklere genel bir bakış sağlamak için, bu görev için özel olarak oluşturulan sentetik bir veri setinde son teknoloji yaklaşımların kapsamlı bir deneysel değerlendirmesi yapılmıştır.

Rhodin ve ark [7], tarafından yapılan çalışmada görüntülerden 3B insan pozu tahmini yöntemleri ve çözümü önerilmiştir. Bu çok büyük bir veri setine sahip olan gelişmiş derin ağ mimarileri ile mümkündür. Çalışmalarında, notların çoğunu birden fazla görünüm kullanarak, yalnızca eğitim sırasında değiştirme yöntemi önerilmiştir. Özellikle, sistemi tüm görünümlerde aynı pozu tahmin edecek şekilde eğitmişlerdir. Böyle bir tutarlılık kısıtlaması gereklidir, ancak doğru pozları tahmin etmek için yeterli değildir. Bu nedenle, küçük bir etiketli görüntü setinde doğru pozu tahmin etmeyi amaçlayan denetimli bir kayıpla ve ilk tahminlerden sapmayı önleyen bir düzenleme terimi ile tamamlamaktadır. Ayrıca, kamera pozunu insan pozuyla birlikte tahmin etmek için bir yöntem önerilmiştir, bu da kalibrasyonun zor olduğu çoklu görüntü çekimlerini kullanmaya olanak tanımaktadır. Yaklaşımın etkinliği, dönen kameralara ve uzman kayak hareketine sahip yeni bir Ski veri kümesinde gösterilmiştir.

1.1.2. 3B ve 2B insan poz tahmini yaklaşımı

Zhou ve ark [8], tarafından yapılan çalışmada, vahşi doğada üç boyutlu insan poz tahmini gerçekleştirilmiştir. Mevcut veri setleri ya 2B poz veren doğal görüntülerde ya da 3B poz veren laboratuvar görüntülerinde olduğu için, eğitim verilerinin eksikliği bu çalışmayı zorlaştırmıştır. Birleştirilmiş derin nötr bir ağda iki aşamalı basamaklı yapı sunan 2B ve 3B karma etiketler kullanan zayıf gözetimli bir aktarım öğrenme yöntemi önerilmiştir. Ağ, 3B derinlik regresyon alt ağı ile son teknoloji 2B poz tahmini alt ağını genişletmektedir. İki alt ağı sırayla ve ayrı eğiten önceki iki aşama yaklaşımın aksine, eğitme uçtan uca ve 2B poz ile derinlik tahmini alt görevleri arasındaki korelasyondan tam olarak yararlanmaktadır. Derin özelikler paylaşılan sunumlar aracılığı ile daha da iyi öğrenilmiştir. Bunu yaparken, vahşi doğadan alınmış görüntüler kontrollü laboratuvar ortamlarındaki 3B poz etiketine aktarılmıştır. Ayrıca, yeraltı derinlik etiketlerinin yokluğunda etkili olan 3B poz tahmini düzenlemek için 3B geometrik bir kısıtlama sunulmuştur. Çalışmanın sonunda hem 2B hem de 3B testlerinde rekabetçi sonuçlar elde edilmiştir.

Kanazawa ve ark [9], tarafından yapılan çalışmada, Human Mesh Recovery yöntemi kullanarak, tek bir RGB görüntüden bir insan vücudunun tam 3 boyutlu kafesini yeniden yapılandırmak için uçtan uca bir çerçeve tanımlamışlardır. 2B veya 3B bağlantı konumlarını hesaplayan mevcut yöntemlerin çoğunun aksine, şekil ve 3B bağlantı açılarıyla parametrelerden daha zengin ve daha kullanışlı bir kafes temsili üretilmiştir. Temel amaç, temel noktaların yeniden projeksiyon kaybını en aza indirmektir. Bu da modelin yalnızca iki boyutlu gerçek ek açıklamaları olan doğal ortamdaki görüntüler kullanılarak eğitilmesini sağlar. 2B anahtar noktası algılamalarına güvenmeyerek 3B poz ve şekil parametreleri doğrudan görüntü piksellerinden çıkartılmıştır. 3B kafeslerin çıktısını alan ve 3B ortak konum tahmini ve parça segmentasyonu gibi görevlerde rekabetçi sonuçlar veren, daha önce uygulanmış, doğada var olan ve dışarıda yapılan çeşitli optimizasyon temeli yöntemler konusunda yaklaşmaları göstermiştir.

Omran ve ark [10], tarafından yapılan çalışmada, 3B vücut duruşu ve şeklinin doğrudan tahmini, yüksek düzeyde parametreleştirilmiş derin öğrenme modelleri için bile zorluk çıkardığı ön görülmüştür. Bu çalışmada, yeni bir yaklaşım önerilmiştir. 2B görüntü uzayından tahmin uzayına eşleme yapmak zordur: perspektif belirsizlikleri kayıp işlevini gürültülü hale getirir ve eğitim verileri kısıtlıdır. Aşağıdan yukarıya semantik vücut parçası bölümlendirmesi ve yukarıdan aşağıya vücut modeli kısıtlamalarını kullanarak bir CNN içinde istatistiksel bir vücut modelini bütünleştirir. NBF (Natural Body Fitting) tamamen ayırt edilebilirdir ve 2B, 3B açıklamalar kullanılarak eğitilebilir. Ayrıntılı deneylerde, modelin bileşenlerinin performansı nasıl etkilediği analiz edilmiş, özellikle parça segmentasyonlarının açık ara temsil olarak kullanılması ve standart kıyaslamalarda rekabetçi sonuçlarla 2B görüntülerden 3B insan pozu tahmini için, verimli bir şekilde eğitilebilir bir çerçeve sunulmuştur.

Luvizon ve ark [11], tarafından yapılan çalışmada, kamera koordinatlarında, 2B açıklamalı veriler ve 3B pozların etkili bir kombinasyonunun yanı sıra basit bir çoklu görünüm genellemesine izin veren bir 3B insan pozu tahmin yöntemi önerilmiştir. 3B insan pozu tahmini, genellikle kök gövde eklemine göre 3B pozları tahmin etme görevi olarak görülür. Bu amaçla, sorun, görüntü düzleminde piksel cinsinden 3B pozların tahmin edildiği ve mutlak derinliğin milimetre cinsinden tahmin edildiği farklı bir perspektife dönüştürülmüştür. Buna dayanarak, tek bir monoküler eğitim prosedürü gerektiren kalibre edilmemiş görüntülerden çoklu görünüm tahminleri için fikir birliğine dayalı optimizasyon algoritması önerilmiştir. Kullandığı yöntem, iyi bilinen 3B insan pozu veri kümelerinde son teknolojiyi iyileştirerek, en yaygın karşılaştırmada tahmin hatasını %32 oranında azaltmıştır. Buna ek olarak, sonuçları, ortalama olarak monoküler tahminler için 80 mm ve çoklu görüntü için 51 mm'ye ulaşan mutlak pozisyon hatası olarak da rapor edilmiştir.

Luvizon ve ark [12], tarafından yapılan çalışmada, hareketsiz görüntülerden 2B ve 3B poz tahmini ve video sekanslarından insan eylemi tanıma için birlikte çok görevli bir çerçeve önerilmiştir. Eylem tanıma ve insan pozu tahmini yakından ilişkilidir, ancak her iki sorun da genellikle literatürde ayrı görevler olarak ele alınmaktadır. İki sorunu verimli bir şekilde çözmek için tek bir mimarinin kullanılabileceğini ve yine de en son

teknoloji sonuçlara ulaşılabileceğini ve ayrıca uçtan uca optimizasyonun ayrılmış öğrenmeye göre önemli ölçüde daha yüksek doğruluğa yol açtığını gösterilmiştir. Önerilen mimari, farklı kategorilerdeki verilerle aynı anda sorunsuz bir şekilde eğitilebilir. Dört veri setinde (MPII, Human3.6M, Penn Action ve NTU) alınan sonuçlar, yöntemin hedeflenen görevler üzerindeki etkinliğini göstermektedir.

Ramakrishna ve ark [13], tarafından yapılan çalışmada, görsel bellek için büyük bir hareket yakalama külliyatından yararlanarak, tek bir görüntüdeki anatomik işaretlerin 2B konumlarından bir insan figürünün 3B konfigürasyonunu kurtarmak için faaliyetten bağımsız bir yöntem sunulmuştur. Bir görüntünün projeksiyonlarından 3B noktalarının konfigürasyonunu yeniden inşa etmek, zor bir sorundur. Noktalar, bir vücut üzerindeki anatomik işaretler gibi anlamsal bir anlam taşıdığında, insan gözlemciler genellikle kapsamlı görsel hafızadan yararlanarak makul bir 3B konfigürasyon çıkarabilir. Yöntem, antropometrik olarak düzenli vücut pozunu çözer ve görüntü projeksiyonları üzerinde çalışan bir takip algoritması aracılığıyla kamerayı açıkça tahmin eder. Antropometrik düzenlilik oldukça bilgilendirici bir önsezidir, ancak bu tür kısıtlamaları doğrudan uygulamak zorludur. Bunun yerine, 3B'deki mantıksız konfigürasyonlardan vaz geçmek için kapalı formda çözülebilecek uzuv uzunluklarının karesi toplamına gerekli bir koşul uygulanmıştır. Yöntemin farklı bakış açılarından yakalanan çok çeşitli insan pozları üzerinde performansı değerlendirmiş ve yeni 3B konfigürasyonlara genelleme ve eksik verilere kadar dayanıklılık gösterilmiştir.

Zeng ve arka [14], tarafından yapılan çalışmada, UV (“U” ve “V” harfleri 2D dokunun eksenlerini belirtir) uzayındaki ağ ve yerel görüntü özellikleri (yani, 3B ağın doku haritalaması için kullanılan bir 2B alan) arasındaki yoğun uyumu açıkça kuran DecoMR adlı, modelsiz bir 3B insan ağ tahmin çerçevesi önerilmiştir. İnsan vücudunun 3B ağını tek bir 2B görüntüden tahmin etmek, artırılmış gerçeklik ve İnsan-Robot etkileşimi gibi birçok uygulamada önemli bir görevdir. Bununla birlikte, önceki çalışmalar, örgü yüzeyi ile görüntü pikselleri arasındaki yoğun yazışmaların eksik olduğu ve yetersiz bir çözüme yol açan CNN kullanılarak çıkarılan global görüntü özelliğinden 3B ağını yeniden yapılandırmıştır. DecoMR ilk olarak, yerel özellikleri

görüntü uzayından UV uzayına aktarılmış pikselden yüzeye yoğun yazışma haritasını (yani, IUV görüntüsü) öngörmüştür. Daha sonra aktarılan yerel görüntü özellikleri, aktarılan özelliklerle iyi hizalanmış bir konum haritasına getirilmek için UV alanında işlenmiştir. Son olarak, önceden belirlenmiş bir haritalama fonksiyonu ile konum haritasından 3B insan ağı yeniden yapılandırılmıştır. Ayrıca, mevcut süreksiz UV haritasının ağın öğrenilmesine yardımcı olmadığı da gözlemlenmiştir. Bu nedenle, orijinal ağ yüzeyindeki komşu ilişkilerin çoğunu koruyan yeni bir UV haritası önerilmiştir. Deneyler sonucunda, önerilen yerel özellik hizalamasının ve sürekli UV haritasının, birden fazla genel karşılaştırmada mevcut 3B ağ tabanlı yöntemlerden daha iyi performans gösterdiği gözlemlenmiştir.

Yang ve ark [15], tarafından yapılan çalışmada, Derin Evrişimsel Sinir Ağları (DCNN'ler) kullanarak monoküler görüntülerden 3D insan pozu tahmininde dikkate değer gelişmeler elde etmişlerdir. Kısıtlı laboratuvar ortamında toplanan büyük ölçekli veri kümelerindeki başarılarına rağmen, doğal görüntüler için 3B poz ek açıklamalarını elde etmek zordur. Bu nedenle, vahşi doğada 3B insan pozu tahmini hala zor bir problemdir. Çalışmalarında, tamamen açıklamalı veri kümesinden öğrenilen 3B insan pozu yapılarını, yalnızca 2B poz ek açıklamalarıyla doğal görüntülere dönüştüren rakip bir öğrenme çerçevesi önerilmiştir. Poz tahmin sonuçlarını sınırlandırmak için sabit kodlanmış kuralları tanımlamak yerine, tahmin edilen 3B pozları temel gerçeklerden ayırt etmek için yeni birçok kaynaklı ayırıcı tasarlayarak, bu, poz tahmincisinin vahşi ortamdaki görüntülerle bile antropometrik olarak geçerli pozlar oluşturmasını sağlamaya yardımcı olmuştur. Ayrıca, dskriminator için özenle tasarlanmış bir bilgi kaynağının performansı artırmak için gerekli olduğu gözlemlenmiştir. Böylece, dskriminator için yeni bir bilgi kaynağı olarak, vücut eklemleri arasındaki ikili göreceli konumları ve mesafeleri hesaplayan geometrik bir tanımlayıcı tasarlanmıştır. Karşı öğrenme çerçevesi yeni geometrik tanımlayıcı ile etkinliği, yaygın olarak kullanılan kamuya açık ölçütler üzerinde yapılan kapsamlı deneyler yoluyla kanıtlanmıştır. Yaklaşım, önceki son teknoloji yaklaşımlara kıyasla performansı önemli ölçüde artırmıştır.

Kaynak araştırılmasında yapılmış çalışmaların yöntem ve sonuçları Tablo 2.1.'de gösterilmiştir.

Tablo 1.1. Literatür taramasında yapılmış çalışmaların yöntem ve sonuçları

Araştırma	Yöntem	Veri tabanı	Görüntü Sayısı	Doğruluk
Ramakrishna ve ark [13]	PCA	CMU Motion Captur	2605	%99
Omran ve ark [10]	NBF + SMPL	UP-3d	8000	%98,5
Tung ve ark [4]	SFM	H3.6M	3600000	%98,4
Pavlokos ve ark [5]	SMPLify	H3.6M	3600000	%85,96
Tekin ve ark [2]	RSTV + KRR	HumanEva-I/II	3000	%85,36
Luvizin ve ark [11]	NTU RGB + D	Human3.6M	3600000	%85,5
Kanazawa ve ark [9]	SMPLify	MPI-INF-3DHP	2929	%82,5
Luvizion ve ark [12]	PVH + TSP	Human3.6M	3600000	%80,1
Sarafianos ve ark [6]	YOLOv4-P6	COCO	1500000	%75,4
Pavlokos ve ark [3]	RSTV+KDE	KTH Football II	800	%71,9
Rhodin ve ark [7]	MPJPE and NMPJPE	Human3.6M and Ski	3600000	%70,8
Zeng ve ark [14]	SMPL	Evalution on 3DPW	60	%68,5
Zhou ve ark [8]	3D+2D/wgeo	MPI-INF-3DHP	2929	%64,9
Yang ve ark [15]	SFM	Human3.6M	3,600,000	%58,6
Araştırma sonuçları	VGG16 ve Resnet50	MPII dataset	25000	%87

1.2. Amaç

Bu çalışmada, "MPII İnsan Duruşu" veri seti ve Evrişimsel Sinir Ağı (Convolutional Neural Network-CNN) kullanılarak insan duruş tespiti gerçekleştirilmiştir. Ağın tahminleri standart karşılaştırmalarla karşılaştırılmış ve ağı iyileştirmek için hangi optimizasyonların yapılabileceği ve ağın ana sınırlamalarının neler olduğu tartışılmıştır.

Çalışmanın amacı, Keras modeli adı verilen bir derin öğrenme modelinde son teknoloji bir ağ uygulayarak insan vücudunun duruşunu tahmin etmek için derin bir CNN

yapısının nasıl kullanılabileceğini keşfetmektir. Çalışmada aşağıdaki birkaç soruya cevap bulunmaya çalışılmıştır.

- Ağ arka plandaki, kapanıştaki, giysideki ve vücut ölçülerindeki farklı koşullarla ne kadar iyi başa çıkabilir?
- Eğitim verilerinin hazırlanması ağın performansını nasıl etkiler?
- Her pozu nadirliğine göre puanlayarak MPII veri setini analiz etmek, MPII veri kümesinde daha nadir bulunan pozları tahmin etmek daha mı zordur?

BÖLÜM 2. LİTERATÜR TARAMASI

2.1. 2B Açıklamalı Veri Kümeleri

MPII İnsan Duruşu veri seti [1], 2B vücut eklem açıklamalarına sahip 40 bin kişiyi içeren 25K görüntüye sahiptir. İnsan eklem noktaları hakkında 3B bilgi yoktur. Görüntüler, çok sayıda günlük aktiviteden toplanmıştır. Her görüntü bir Youtube videosundan alınmıştır.

Şekil 2.1. MPII veri kümesindeki bazı günlük insan etkinliklerinin örnek görüntüleri

2.2. İnsan Poz Tahmini

İnsan pozu tahmininin amacı değişebilir [16]. 3B vücut poz tahmini oluşturmak için tek bir 2B görüntünün ve bir 2B derinlik görüntüsünün kullanılması için çalışmalar yapılmıştır [17]. Bu tez çalışmasının odak noktası, bir insan vücudu pozunu tek bir 2B görüntüden tahmin etmektir. Vücut pozu vücuttaki ana eklemler (Şekil 2.1.), bilek, dirsek, omuz, ayak bileği, diz, kalça, boyun ve başın üst kısmı, göğüs kafesi ve pelvis (leğen kemiği) ile temsil edilir. Görev, bu eklemlerin görüntü koordinatlarını tahmin etmektir.

Şekil 2.2. Pozu oluşturan kilit noktalar

İnsan pozu tahmininin en zorlu yönlerine göre: (1) İnsanın görsel görünümlerindeki büyük değişkenlik (giysi, aksesuarlar, saç stilleri), (2) ışık koşullarındaki değişkenlik, (3) insan fiziğindeki değişkenlik (uzun, kısa, fazla kilolu, zayıf), (4) kendi kendine kapanma nedeniyle kısmi tıkanma veya sahnede nesnelerin katmanlaşması, (5) insan iskeletinin karmaşıklığı (insan vücudunun 230 eklemi vardır), (6) bu, pozun yüksek boyutluluğuna (244 derece serbestlik), (7) 3B kaybına yol açar [18]. 2B görüntünün görüntülenmesinden elde edilen bilgiler. Şekil 2.2.'deki görüntüler MPII veri setinden alınır ve insan pozu tahmininin zorlu koşullarını gösterir.

Şekil 2.3. MPII veri kümesinden, insan pozu tahmininin zorluğunu gösteren görüntü örnekleri

CNN'lerden önce insan poz tahminini çözme girişimleri genellikle soruna bütünsel bir bakış açısıyla yaklaşmamaktaydı ve CNN'lerin elde ettiği sonuçlara yakın sonuçlar üretememekteydi [19], [20]. İnsan pozu sorununu evrişimsel bir sinir ağıyla çözmeye yönelik ilk girişim 2013 yılında iki Google çalışanı tarafından yapılmıştır ve vücut eklemlerinin x-y koordinatlarına gerilemek için evrişimsel ve tamamen bağlantılı katmanlar kullanmışlardır [21]. Elde ettikleri sonuçlar o zamanlar son teknolojiydi ve CNN'lerin pozun bütünsel bir görünümünü elde edebileceklerini göstermiştir. O zamandan beri, vücut poz tahminlerinde en son teknoloji CNN'leri içermektedir.

Bu tez çalışması için kullanılan ağ düzeni, MPII İnsan Duruşu veri setinde en iyi sonuçları elde eden 2016 tarihli bir makaledeki düzeni yakından takip etmektedir [22]. Bu tür tamamen evrişimsel sinir ağı, poz tahmini için çok uygun görünmektedir. Bunun nedeni muhtemelen bir CNN'in hem yerel görüntü bilgilerini (parçaları, dirsekleri, bilekleri vb. algılama) hem de küresel görüntü bilgilerini (parçaları birbirine bağlama) işleme becerisidir.

2.2.1. Keras modeli

Keras, Python için derin bir öğrenme kütüphanesidir ve derin öğreneme modellerinin oluşturulması ve eğitimi için çok uygun bir ortam sağlar [23]. Keras başlangıçta araştırmacıların daha hızlı denemeler yapabilmeleri için geliştirilmiştir. Keras'ın öne çıkan özellikleri aşağıdaki gibidir;

- Kodu değiştirmeden hem CPU'da hem de GPU'da çalışmasını sağlar.
- Derin öğrenme modellerinin prototiplemesinin hızlıca yapılmasına imkân sağlayan kullanıcı dostu API'ye sahiptir.
- Evrişimsel ağlar (bilgisayarlı görü için), yinelemeli ağlar (zaman serisi işlemek için) ve her ikisinin beraber kullanımı için önceden tanımlı desteğe sahiptir.

Keras'da pek çok değişik ağ yapısının, çoklu girdi ya da çoklu çıktının, katman paylaşımının, model paylaşımının vb. uygulanabilmesi mümkündür. Bu da Keras'ı, çekişmeli üretici ağlardan sinirsel Turing makinesine kadar her türlü derin öğrenme modelinin oluşturulması için uygun hale getirmektedir.

Keras, MIT lisansı ile dağıtılmaktadır. Yani ticari projeler dahil her yerde serbest olarak kullanılabilmektedir. Python 2.7'den 3.6'ya kadar tüm versiyonları destekler.

2.2.2. Jupyter not defterleri

Jupyter not defterleri derin öğrenme denemelerini çalıştırmak için çok iyi bir yoldur [24]. Veri bilimi ve makine öğrenmesi topluluklarınca çok yaygın bir şekilde kullanılmaktadır. Not defteri Jupyter Notebook uygulaması tarafından oluşturulan ve internet tarayıcısında düzenleyebildiğimiz bir dosyadır. Python kodlarını çalıştırmanın yanı sıra ne yaptığımızı anlatabileceğimiz zengin bir metin düzenlemek mümkündür. Büyük kodlarımızı küçük parçalara ayırıp daha etkileşimli çalıştırabilir ve bir sorun olduğunda bütün kodu tekrar çalıştırmaya gerek kalmadan ilgili bölümü düzelterek yolumuza davam etme imkanı sağlar.

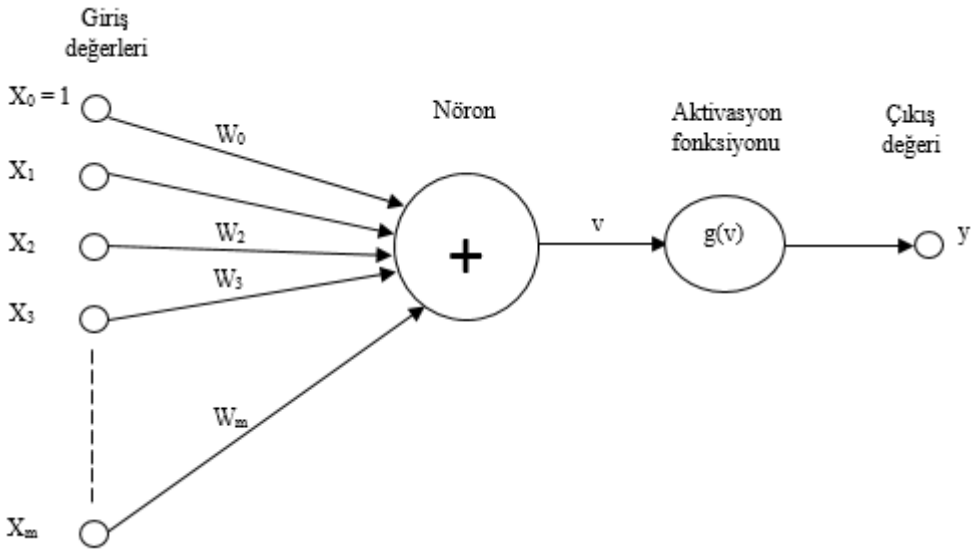
2.2.3. Evrişimli sinir ağlarına giriş-CNN'ler

Geçtiğimiz birkaç yıl içinde, CNN'ler insan pozu tahminine ve genel olarak bilgisayar görüşüne büyük katkılarda bulunmuştur. Bu bölüm, CNN'lerin nasıl çalıştığı ve bilgisayar görüşü için neden etkili oldukları hakkında kısa bir giriş yapmayı amaçlamaktadır. Bir sinir ağındaki temel hesaplama birimi, yapay nörondur. x den x_m'ye kadar olan girdi değerleri, w_0'dan w_m'ye kadar olan ağırlıklarıyla çarpılır ve sonuç

toplanır ve bir aktivasyon fonksiyonundan geçirilir (Şekil 2.4.). Nöronun çıkışı *y* aşağıdaki şekilde hesaplanır (Denklem 2.1):

$$Nöronun\ çıkışı\ y = g(\sum_{i=0}^{m} w_i\, x_i) \tag{3.1}$$

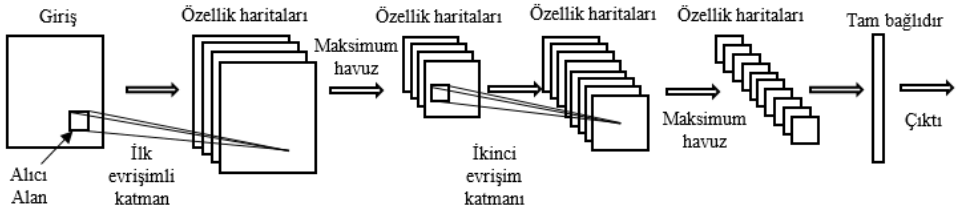
Yapay nöron, insanların beyninde bulunan nörondan esinlenmiştir. Girdiler dendritler olarak görülebilir ve aktivasyon fonksiyonu, aksonun (çıktı) bir sonraki nörona ne zaman ve ne kadar güçlü sinyal göndermesi gerektiğini kontrol eder. Tipik olarak ağın bir katmanda birçok nöronu vardır ve ilk katmandaki nöronlar, ikinci katmandaki nöronların girdisini oluşturur vb.



Şekil 2.4. Yapay bir nöron, x_0 önyargı olarak adlandırılır ve genellikle 1'e ayarlanır

CNN'ler, evrişimsel ve maksimum havuz katmanları tarafından oluşturulan ileri beslemeli, yapay sinir ağlarından oluşan bir gruptur. Bir CNN'de, bir nöronun önceki katmandan (alıcı alan) x_1'den x_m'e, çekirdek boyutu tarafından kontrol edilen sınırlı sayıda girdisi vardır. Nöronların etkin alıcı alanı daha derin katmanlarda büyür ve CNN'lere yerel özellikleri daha derin katmanlarda global özelliklerle birleştirme

yeteneği verir. Şekil 2.5., sınıflandırma için kullanılan bir CNN için basit bir yapıyı gösterir. Ağa girdi bir görüntüdür ve çıktı ise bir nesnenin tahmin edilen sınıfıdır.



Şekil 2.5. İki evrişimli katmana sahip basit bir CNN mimarisi

2.2.4. Evrişimli katman

Bir CNN'nin ana yapı taşı, öğrenilebilir filtrelere sahip evrişimsel bir katmandır. Katman, giriş ve filtreler arasında evrişimsel bir işlem gerçekleştirir ve sonucu bir sonraki katmana aktarır. Evrişimin sonucu, özellik haritası adı verilen 2B bir düzlemdir. Normalde, evrişimsel katmanın çıktısı 3B hacim olacak şekilde birkaç öğrenilebilir filtre kullanılır, burada H ve W uzamsal boyutlardır ve D özellik haritalarının sayısıdır (kullanılan filtre sayısı). Örneğin Şekil 2.4.'teki ağın ilk katmanda 4 filtre ve ikinci katmanda 8 filtre vardır. Çıktının uzamsal boyutu W_{out}, aşağıdaki formüle göre filtre çekirdek boyutu (K), adım (S), sıfır doldurma (P) ve W_{in} ile belirlenir (Denklem 2.2):

$$Çıktının\ uzamsal\ boyutu\ W_{out} = \frac{W_{in}-K+2P}{S} + 1 \qquad (2.2)$$

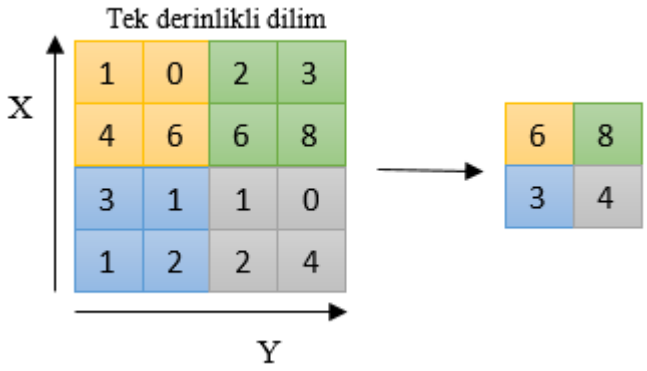
Bu çalışmada uygulanan ağdaki hemen hemen tüm evrişimsel katmanlar, 3 × 3 çekirdek boyutu, 1 sıfırlama ve 1 adım kullanır, böylece uzamsal boyutlar değişmez [25].

Evrişimsel katmanın tasarımı biyolojiden ve kedinin görsel korteksinin nasıl çalıştığından esinlenmiştir [26]. Görsel kortekteki her hücre, görsel alanın alıcı alan adı verilen küçük bir bölgesine bakar. Hücreler birlikte tüm görsel alanı kaplar ve her hücre, kenarlar gibi özellikleri algılayarak yerel bir filtre gibi davranır. İlk evrişimsel

katmanlar, kenarlar ve dokular gibi yerel özellikleri algılar [27]. Daha sonra ağdaki evrişimsel katmanlar, bu yerel özellikleri daha yüksek seviyeli özelliklerle birleştirir. Eğitim sırasında ağ güncellenir ve görev için en kullanışlı özellikleri çıkaran filtre parametrelerini öğrenir. CNN'in bu filtreleri el işi yapmak yerine öğrenmesine izin vermek hem zamandan tasarruf sağlar hem de daha iyi sonuçlar verir.

2.2.5. Maksimum havuzlama katmanı

Maksimum havuzlama işlemi bir çekirdekteki en büyük değeri döndürür ve diğer değerleri atar. Şekil 2.5., en yaygın maksimum havuzlama türünü gösterir. Maksimum paylaşım katmanı, uzamsal çözünürlüğü kademeli olarak düşürmek için evrişimli katmanlar arasına yerleştirilir. Bu, parametre miktarını azaltma etkisine sahiptir ve böylece ağdaki hesaplama miktarı azalır [28]. Ağdaki parametrelerin miktarının azaltılması aynı zamanda düzenleyici bir etkiye sahiptir ve ezberlemeyi azaltır.



Şekil 2.6. 2 adım ile çekirdek boyutu 2x2 olan maksimum havuzlama işlemi

2.2.5.1. Dropout katmanı

Ağ içindeki bazı bağlantıların kaldırılmasıyla eğitim performansının artacağı varsayılmaktadır. Dropout katmanına 0'dan büyük 1'den küçük bir oran verilmektedir. Böylece eğitim esnasında bu oran miktarındaki bağlantıyı rastgele kapatmaktadır. Dropout Katmanı, eğitim verilerinde birlikte uyarlamaları önleyerek yapay sinir ağlarında aşırı uyumu azaltmak için kullanılan bir düzenleme tekniğidir. Dropout

terimi ağırlıkların incelmesi anlamına gelir. Dropout katmanı, bir sinir ağının eğitim süreci sırasında ünitelerin (hem gizli hem de görünür) rastgele "bırakılması" veya çıkarılması anlamına gelir. Hem ağırlıkların incelmesi hem de birimlerin düşmesi, aynı tip düzenlileştirmeyi tetikler ve ağırlıkların seyreltilmesi söz konusu olduğunda sıklıkla Dropout katmanı kullanılır.

Dropout katmanı genellikle zayıf ve güçlü katman olarak ikiye bölünür. Zayıf katman, kaldırılan bağlantıların sonlu fraksiyonunun küçük olduğu süreci tanımlar ve güçlü katman, bu fraksiyonun büyük olduğu zamanları ifade eder. Güçlü ve zayıf katman arasındaki sınırın nerede olduğu konusunda net bir ayrım yoktur ve kesin çözümlerin nasıl çözüleceğine dair sonuçları olsa da çoğu zaman bu ayrım anlamsızdır.

Bazen girişlere sönümleme gürültüsü eklemek için dropout kullanılır. Bu durumda, zayıf katman az miktarda sönümleme gürültüsü eklemeyi ifade ederken, güçlü katman daha fazla miktarda sönümleme gürültüsü eklemeyi ifade eder. Her ikisi de ağırlık dropout katmanları olarak yeniden yazılabilir.

2.2.5.2. Ağı eğitme

Eğitim sırasında, CNN'e bir görüntü girilir ve maksimum havuzlama katmanları çıktılarını hesaplar ve bir öngörü çıkana kadar bunları bir sonraki katmana besler ve bu sürece ileri yayılma denir. Denetimli öğrenmede, tahmin edilen sonuç bir temel gerçekle karşılaştırılır, bir kayıp işlevi uygulanır ve tahmin temel gerçeğe ne kadar yakınsa, kayıp o kadar küçük olur. Bu çalışmadaki ağ, bir öklid (L2 olarak da adlandırılır) kayıp işlevi kullanır ve aşağıdaki tanıma sahiptir. Burada *N*, çıktıların sayısı, *y* temel gerçek ve $\hat{y}$, öngörülen çıktıdır (Denklem 2.3).

$$\text{Öklid} = Loss = \frac{1}{2N}\sum_{i=1}^{N}(y_i - \hat{y}_i))^2 \quad (2.3)$$

Kayıp hesaplandıktan sonra, her bir nöronun kayba ne kadar katkıda bulunduğunu bulmak için geri yayılma algoritması kullanılır [29]. Bunu, ağ parametrelerine göre kayıp fonksiyonunun türevini alarak yapar. Θ modelin parametre vektörü, α öğrenme

hızı ve $J(\Theta)$ kayıp ise, i'inci parametrenin güncellenmesi şu şekilde yazılabilir (Denklem 2.4):

$$Modelin\ parametre\ vektörü = \Theta_i = \Theta_i - \alpha \frac{\partial}{\partial \Theta_i} \mathrm{J}(\Theta) \tag{2.4}$$

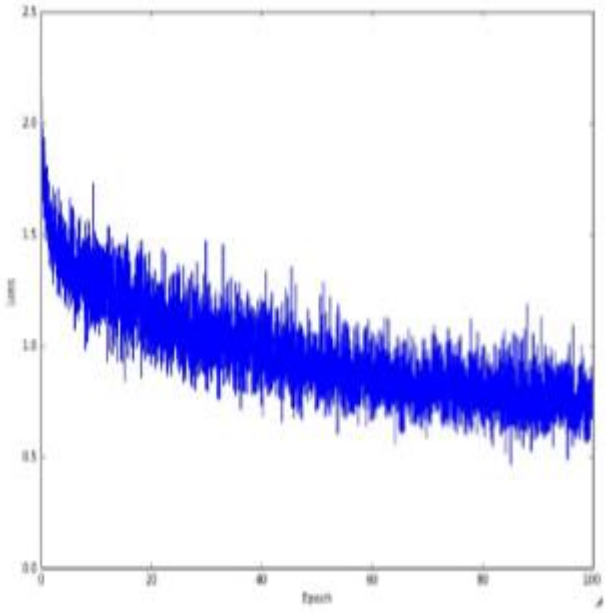
Hata katkısı daha sonra ağ üzerinden geriye doğru yayılır ve kayıpları en aza indirmek için parametreler güncellenir. Öğrenme hızı, modelin bir meta parametresidir ve belirli görev için iyi ayarlanmalıdır. Şekil 2.7., denetimli öğrenmenin temel adımlarını göstermektedir.

Şekil 2.7. Denetimli öğrenmenin bir tekrarı

Eğitimi hızlandırmak için, bir eğitim tekrarlaması sırasında çeşitli eğitim örnekleri (mini yığın olarak adlandırılır) genellikle paralel olarak işlenir. Verilerini eğitim ve test seti olarak iki bölüme ayırmak yaygın bir uygulamadır. Eğitim seti, ağı eğitmek için kullanılır ve ağın doğrulama seti üzerinde tahminler yapmasına izin vermek için eğitim periyodik olarak kesilir.

Test setindeki tahminler, eğitimin nasıl ilerlediğini takip etmek için çizilen bir kayıpla sonuçlanacaktır. Test setindeki kayıp azaldığı sürece eğitim devam eder. Ağ, eğitim setindeki tüm eğitim örneklerini bir kez gördüğünde, buna epok denir ve Şekil 2.8., epok sayısına göre çizildiğinde eğitim sırasındaki kaybı gösterir. Grafiğin pürüzlü olmasının nedeni, ağın stokastik gradyan inişiyle eğitilmiş olmasıdır. Kayıp azalmak için durduğunda, ağ test seti üzerinde test edilir [30]. Ağ, test setinde iyi ancak eğitim

setinde kötü performans gösteriyorsa, modelin eğitim setini aştığı ve iyi genelleme yapamadığı anlamına gelir.



Şekil 2.8. Epok sayısının bir fonksiyonu olarak çizilen kayıp

2.2.5.3. Toplu normalleştirme

Bu tezde uygulanan ağın tamamında toplu normalleştirme tutarlı bir şekilde kullanılmıştır. Toplu normalleştirme, derin sinir ağlarını eğitirken yaygın bir soruna hitap eder, bu erken katmanlardaki parametre değişiklikleri, giriş-üstü katmanların dağılımını büyük ölçüde değiştirebilir [31]. Bu sorunu çözmek ve sonraki katmanlar için öğrenmeyi daha kolay hale getirmek için toplu normalleştirme, belirli bir mini yığın için tüm girdileri bir katmana normalleştirir (eğitim sırasında paralel olarak işlenen bir dizi eğitim örneği). Eğer x bir katmana girdi ise, $B = (x_1 \ldots m)$ bir toplu işteki x'in tüm girdileridir ve y_i toplu normalleştirilmiş çıktıysa, toplu normalleştirme aşağıdaki formüllerle açıklanır (Denklem 2.5,2.6,2.7):

$$Mini\ yığın\ ortalaması = \ \mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \qquad (2.5)$$

$$Varyans = \sigma_B = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \quad (2.6)$$

$$Tüm\ Girdiler = \widehat{x_i} = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \theta}} \quad (2.7)$$

$$Toplu\ Normalleştirilmiş\ Çıktı = Y_i = \Upsilon\widehat{x_i} + \beta N_{\Upsilon,\beta}(x_i) \quad (2.8)$$

Ağın temsil gücünü korumak için γ ve β parametreleri tanıtılır ve eğitim sırasında öğrenilir. Mini yığın ortalaması ve varyans, μ_B ve σ_B yalnızca eğitim sırasında kullanılır. Çıkarım için eğitimli ağı kullanırken, tüm eğitim setinin genel ortalaması ve varyansı kullanılır. Mini yığın boyutu ne kadar büyük olursa, parti normalizasyonu o kadar iyi performans gösterir, çünkü ortalama ve varyans tahminleri daha az gürültülü hale gelir. Normalizasyondan sonra katmanlar sıfır ortalamaya ve bir standart sapmaya sahip girdilere sahip olacak ve bu daha öngörülebilir dağılım, ağın daha yüksek bir öğrenme oranıyla eğitilmesine izin verecektir. Toplu normalleştirme ayrıca bir düzenleyici etkiye sahiptir ve ağ parametrelerinin nasıl başlatıldığına karşı ağı daha az hassas hale getirir.

2.2.5.4. Evrişimli katmanlar ile tam bağlantılı katmanlar

Tamamen bağlı katmanlardan oluşan sinir ağları ile karşılaştırıldığında, CNN'ler çok daha az ağırlığa ihtiyaç duyar ve birbirine yakın piksel değerlerinin birbirinden uzak piksel değerlerinden daha fazla ilişkili olduğu özelliğinden daha iyi yararlanırlar. Tamamen bağlı katmanlardan oluşan sinir ağları, daha büyük görüntülere iyi ölçeklenememe eğilimindedir. Üç renk kanalı olan 32 × 32 boyutundaki bir görüntü için, tamamen bağlı tek bir nöronun ağırlığı 32 × 32 × 3 = 3072 olacaktır. Görüntülerin boyutu 256 × 256 × 3'e çıkarsa, tamamen bağlı tek bir nöronun ağırlığı 196608 olacaktır. Normalde bir katmanda birkaç nöron olur, bu da ağırlık sayısının hızla arttığı anlamına gelir. Çok sayıda ağırlık, eğitim sırasında çok fazla bellek gerektirir ve tamamen bağlı ağı aşırı yüklenmeye yatkın hale getirir [28].

CNN'lerde, bir filtrenin ağırlıkları görüntü üzerinde paylaşılır. Ağırlıkların ve küçük filtre boyutlarının paylaşılması, CNN'lerin tamamen bağlı katmanlardan çok daha az ağırlıklara sahip olmasını sağlar. 256×256×3 boyutunda bir görüntü örneğinde, Filtre boyutu 3×3 ve 128 özellikli bir evrişimli katman, 3×3×3×128 = 3456 ağırlıklara sahip olacaktır [32]. Bu, tamamen bağlı katmandan önemli ölçüde daha azdır. CNN'lerin daha az bellek kullanması, derin CNN'ler olarak adlandırılan birçok katmanın birbiri ardına istiflenmesini mümkün kılar [33]. Ağın daha soyut ve karmaşık görevleri öğrenmesi gerektiğinde bunun etkili olduğu kanıtlanmıştır. Örneğin, derin CNN'ler, Go ve ImageNet yarışmasında dünyanın en iyi oyuncusunu yenmeyi başaran Alphago'da başarıyla kullanılmıştır [34].

BÖLÜM 3. DERİN ÖĞRENME KULLANARAK İNSAN HAREKET TESPİTİ

3.1. MPII Veri Seti

Bu çalışmada "MPII İnsan Duruşu" veri seti [1] kullanılmıştır. Veri seti, eklemli insan pozu tahminlerini değerlendirmek için son teknoloji ürünü bir karşılaştırma ölçütüdür. Görüntüler Youtube videolarından alınmıştır ve insan pozları, arka planları, kıyafetleri, vücut ölçüsü, açıklamalı kişiye olan mesafesi ve açısı bakımından büyük farklılıklar gösterir. Veri seti, açıklamalı vücut eklemlerine sahip 40 binden fazla kişiyi içeren yaklaşık 25 bin görüntüden oluşmaktadır. Veri setinin boyutu görüntüler için 12,9 GB ve ek açıklamalar için 12,5 MB'dır. Ek açıklamalar bir Anoconda Jupiyter yapısında sağlanır ve görüntü başına bilgi aşağıda listelenmiştir.

1. Ek açıklamalı resim listesi
- Görüntü adı
- Resimdeki her kişi için gövde ek açıklamaları
    - *x1, y1, x2, y2*-baş dikdörtgenin koordinatları
    - Ölçek-kişi ölçeği w.r.t. 200 piksel yükseklik
    - Nesne konumu-görüntüdeki kaba insan konumu
    - Açıklamalı anahtar noktalar-kişi merkezli vücut eklemi açıklamaları
    - *x, y*-bir eklemin koordinatları
    - ·İd-eklem kimliği (0-r ayak bileği, 1-r diz, 2-r kalça, 3-l kalça, 4-l diz, 5-1 ayak bileği, 6-pelvis, 7-göğüs, 8-üst boyun, 9-baş üstü, 10-r bilek, 11-r dirsek, 12-r omuz, 13-l omuz, 14-l dirsek, 15-l bilek)
    - Görünür-ortak görünürlük
2. Eğitim / test görüntü ataması listesi

- Tek kişi-yeterince ayrılmış bireylerin kimliğini içerir

Şekil 3.1., MPII veri kümesindeki bir görüntü için açıklamalı verileri gösterir.

Şekil 3.1. MPII veri kümesindeki ek açıklama. Mavi dikdörtgen: baş dikdörtgeni, sarı dikdörtgen: sınırlayıcı kare, kırmızı daire: açıklamalı kişinin merkezi, yeşil daireler: anahtar noktalar

Sınırlayıcı kare, nesne konumu (nesneler) ve ölçek ile hesaplanabilir, ancak Şekil 3.2., (a)'da gösterildiği gibi genellikle çok küçüktür. Sık sık ayak bilekleri kesilir ve bu nedenle eğitim verilerinin dışında bırakılır. Bu sorunu çözmek için, *y* koordinatı 15 piksel ve ölçek 1,25 kat artırılır. Şekil 3.2., (b), ayarlanmış sınırlayıcı kareyi göstermektedir. Nesne konumu ve MPII veri kümesinden gelen ölçekle, ayarlanmış sınırlayıcı kareyi aşağıdaki gibi hesaplamak önemsizdir (*x, y* sol üst köşedir) (Denklem 3.1,3.2,3.3):

$$Sınırlayıcı\ Kare = \ Side = scale\ x\ 200\ x1.2 \tag{3.1}$$

$$Ayarlanmış\ Sınırlayıcı\ Kare = x = objpos.x - \frac{side}{2} \tag{3.2}$$

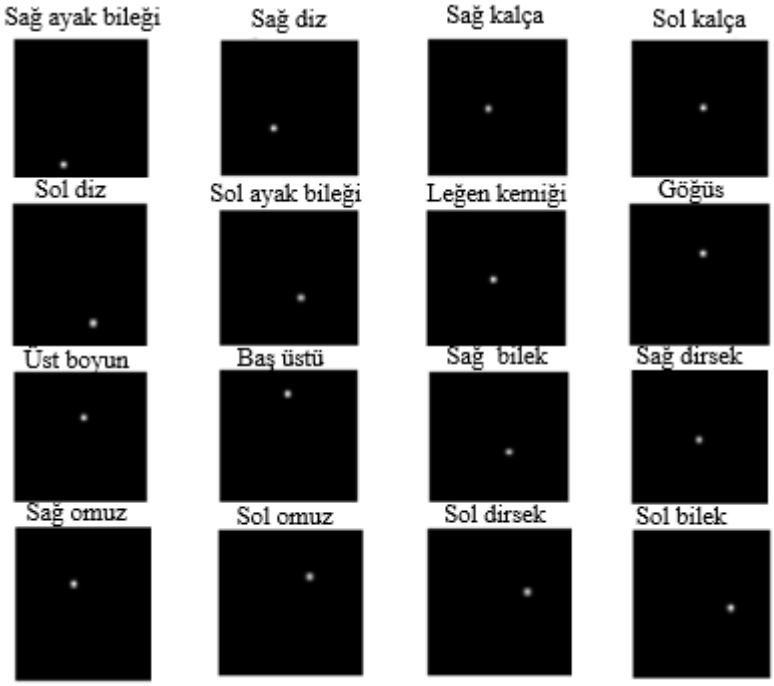
$$Ayarlanmış\ Sınırlayıcı\ Kare = y = objpos.y - \frac{side}{2} + 15 \tag{3.3}$$

(a) MPII sınırlayıcı kare (b) Ayarlanmış sınırlayıcı kare

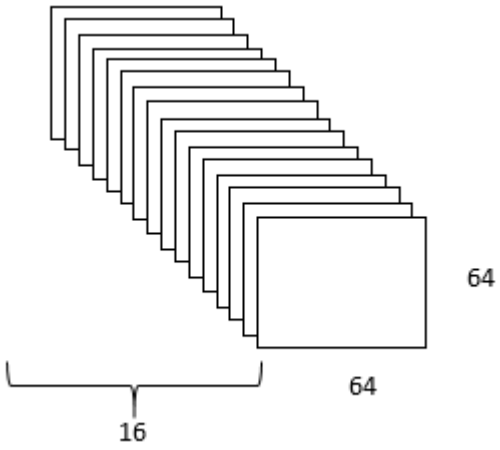
Şekil 3.2. Ayarlamadan önce ve sonra sınır karesi

3.1.1. Eğitim verileri

Eğitim verilerinde yalnızca yeterince ayrılmış kişilerin ek açıklamaları kullanılır ve toplamda yaklaşık 24 bin ek açıklama vardır. Bu görüntülerden kabaca 20000 tanesi eğitim için 4000 tanesi test için kullanılmıştır. Eğitim başlatılmadan önce, MPII veri setindeki her görüntünün notlarının 64 × 64 boyutunda 16 etiket görüntüsüne (her açıklamalı eklem için bir tane) dönüştürülmesi gerekmektedir. Her etiket görüntüsüne karşılık gelen eklemin x-y koordinatında bir 2B Gauss tepesi (7 piksel çapında ve 1 standart sapması) yerleştirilmiştir (Şekil 3.3.). Etiketli görüntüler, her eğitim görüntüsü için 16x64x64 hacimde etiket oluşturmak için istiflenmiştir (Şekil 3.4.). Etiketli görüntü yığını, eğitim sırasında temel gerçek olarak kullanılmıştır.



Şekil 3.3. Şekil 3.1.'deki sarı karede bulunan anahtar noktalar için 16 kesin referans görüntüsü



Şekil 3.4. Yığılanmış kesin referans görüntüleri

MPII veri setinde, tıkalı eklemlerin temel gerçeği verilir ve eğitim verilerine dahil edilebilir. Bununla birlikte, görüntüde eksik olan veya münferit olarak tıkanmış olan eklemlerin MPII veri kümesinde kesin referans açıklaması yoktur. Bu durumda, eğitim verileri olarak sıfırların temel gerçeği kullanılır.

Eğitim verilerini genişletmek için veri arttırma kullanılır ve görüntü dikey eksen etrafında birleştirilir. Veri büyütme ile birlikte, kabaca toplam 48 bin eğitim görüntüsü ve 768 bin etiket görüntüsü vardır.

3.1.2. Eğitim verilerinin ön işlenmesi

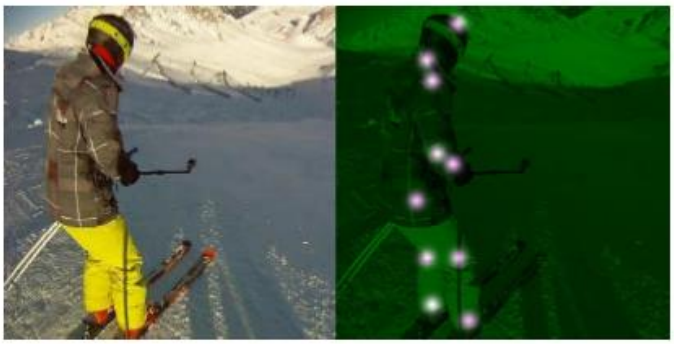
MPII veri kümesindeki görüntülerin çözünürlüğü birbirinden farklıdır ve CNN girdi olarak 256×256 boyutunda bir görüntü aldığından, orijinal görüntünün CNN'e beslenmeden önce hem kırpılması hem de yeniden boyutlandırılması ihtiyacı vardır. Orijinal görüntüyü kırpmanın ve yeniden boyutlandırmanın farklı yolları vardır ve ayrıca eğitim verilerine kapatılmış anahtar noktaları dahil edip etmeme seçeneği de vardır. CNN'in performansını iyileştirmeye çalışma sürecinde, üç farklı ön işleme yöntemi uygulanmış ve değerlendirilmiştir. Tüm ön işleme yöntemleri, en boy oranının korunması için görüntüyü kırpmak için bir sınırlayıcı kare kullanır. Ön işleme, yazılım paketi Anoconda Jupiyter kullanılarak yapılmıştır.

3.1.2.1. İlk ön işleme yöntemi-M1

Şekil 3.5., bir kayakçının ön işlemden önceki orijinal görüntüsünü göstermektedir. İlk ön işleme yönteminde sınırlayıcı karenin orijinal görüntünün dışında olmasına izin verilmez. Sınırlayıcı karenin kenarı görüntüden daha büyükse, kenar minimuma (yükseklik, genişlik) ayarlanır. Görüntüyü bu şekilde kırpmak, açıklamalı kişinin Şekil 3.6.'da görüldüğü gibi görüntünün merkezinde olmasını sağlamaz.

Şekil 3.5. Bir kayakçının ön işlemden önceki görüntüsü

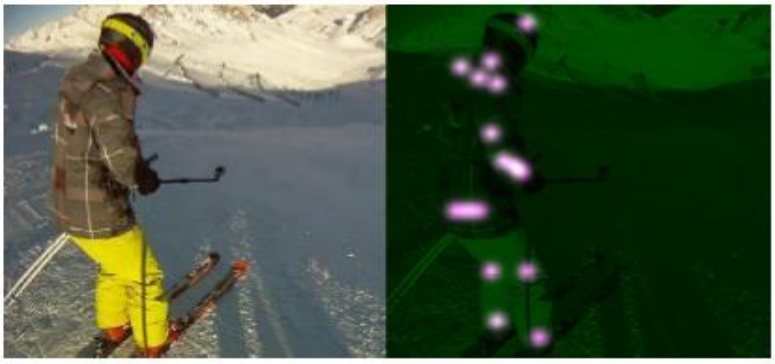
Eğitim verilerinin ilk versiyonu yalnızca görünür eklemler için kesin referans notları içermekteydi, tıkalı eklemler için temel gerçek göz ardı edildi ve sıfıra ayarlandı. Şekil 3.6., bu tür bir ön işlemi göstermektedir. Tıkanan anahtar noktaların kaldırılmasının nedeni, görünmeyen anahtar noktaların ağa poz hakkında bilgi verememesidir, dolayısıyla eğitim verilerinden çıkarılabilmektedir. Bu yaklaşım, ağa eğitim sırasında daha az anahtar nokta ek açıklaması sağlar ve ayrıca ağın tıkanan anahtar noktaların konumunu tahmin etmeyi öğrenmesini engeller.



Şekil 3.6. Anahtar noktaları kapatılmadan ortalanmamış eğitim verileri

3.1.2.2. İkinci ön işleme yöntemi-M2

Eğitim verilerinin ikinci versiyonunda, temel gerçeğe sahip tüm kilit noktalar eğitim için kullanılmıştır (Şekil 3.7.). MPII veri kümesinde, çoğu anahtar noktanın, görüntünün içinde oldukları tahmin edildiği sürece görünür olmasalar bile açıklamaları vardır. Yalnızca açıkça görüntüde olmayan veya ciddi şekilde kapatılmış anahtar noktalarda açıklama yoktur. Bu temel noktalar görünmese bile, onları eğitim sırasında görmenin ağ için faydalı olacağı düşünülmektedir. İkinci eğitim veri seti ile, ağın, tıkanmış olsalar bile konum kilit noktalarını tahmin etmeyi öğrenip öğrenemeyeceği araştırılabilir. Eğitim verilerinin ikinci versiyonu, ilk versiyondakiyle aynı tipte merkezlenmemiş sınırlayıcı kare ve kırpma kullanır.



Şekil 3.7. Kilitlenen anahtar noktaları içeren merkezlenmemiş eğitim verileri

3.1.2.3. Üçüncü ön işleme yöntemi-M3

Eğitim verilerinin üçüncü ve son sürümü, kapatılmış anahtar noktaları içerir, ancak açıklamalı kişi her zaman görüntünün ortasına yerleştirilmiş ve görüntüyü kırpmak için farklı bir yol kullanılmıştır. Şekil 3.8., üçüncü ön işleme yöntemini göstermek için kullanılmıştır.

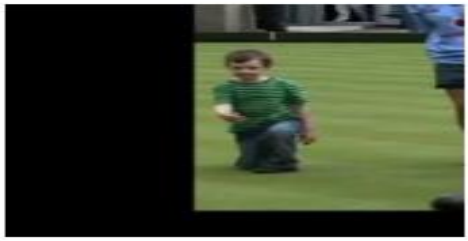
Şekil 3.8. Sol alt köşedeki çocuk kenara yakındır

İkinci ön işleme yöntemi, görüntüyü kırpmak için sarı sınırlayıcı kutuyu kullanır ve ardından görüntüyü 256x256 olarak yeniden boyutlandırır. Açıklamalı kişi Şekil 3.8.'deki gibi kenara yakın olduğunda, genişletilmiş bir görüntü ve değişen bir en-boy oranı ile sonuçlanacaktır (Şekil 3.9.).

Şekil 3.9. İkinci ön işleme yöntemi görüntüleri uzatır ve çocuk ortada değildir

Sıfır dolgu, açıklamalı kişiyi ortalamak ve görüntünün esneme sorununu önlemek için kullanılır. Sıfır dolgusunda, sınırlayıcı kutunun görüntünün kenarlarını aştığı yerde görüntüye sıfırlar eklenir. Şekil 3.10., üçüncü ön işleme yönteminden sonraki sonucu göstermektedir.

Görüntüye açıklama eklemek için kişinin etrafında ortalamadan, görüntüde birbirine nispeten yakın birden fazla kişi olduğunda ağ için zor olurdu, çünkü kime açıklama ekleyeceği belirsiz olurdu. Eğitim setindeki bu değişikliğin amacı, ağa not ekleyecek kişinin merkezinin her zaman görüntünün merkezinde olduğunu varsayabilirse, performansın ne kadar iyileştirilebileceğini incelemektir. Bu, kişinin halihazırda bulunduğu yerde daha basit bir sorunu temsil eder, kalan görev pozu tahmin etmektir.



Şekil 3.10. Çocuğu ortalamak ve görüntünün gerilmesini önlemek için sıfır dolgu kullanılmıştır

3.2. Ağı Eğitme

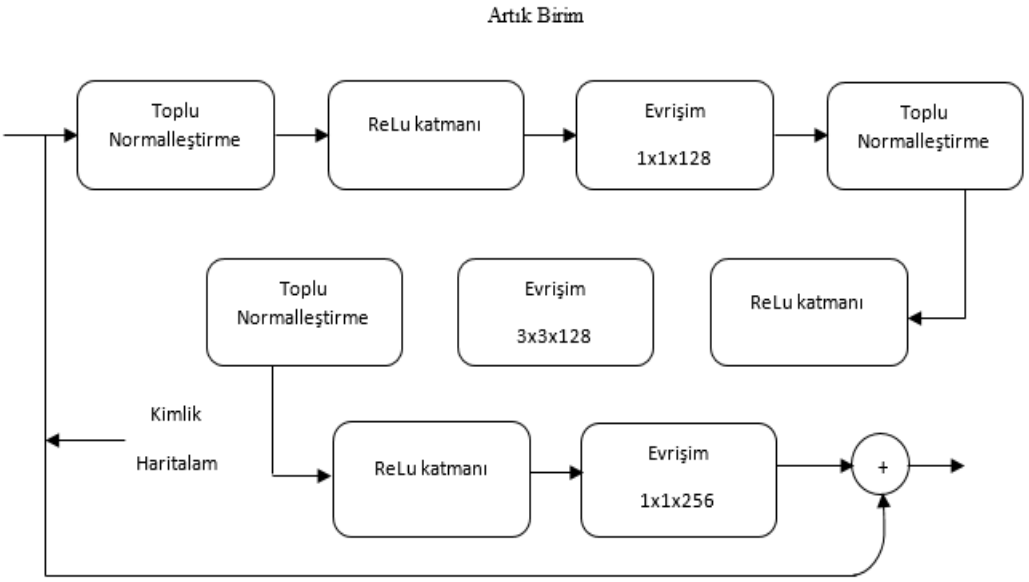
Ağın ana bölümü, 64×64 boyutunda bir görüntü ve giriş olarak 256 özellik alan Sequential modüllerinden oluşur. Ağın ilk bölümü giriş görüntüsünü, Sequential modülüne beslenebilmesi için önceden işler. Bu, evrişimsel katmanlar ve maksimum havuzlama katmanları ile yapılır ve bu ilk adımın ayrıntıları Tablo 3.1.'de gösterilmektedir.

Tablo 3.1. İlk sequential modülünü görüntüye hazırlayan ağın ilk bölümü

Layer	Kernel	Stride	Padding	Output
Giriş Görüntüsü				256x256x3
Convolution	7	2	3	128x128x64
Convolution	3	1	0	128x128 x128
Max pooling	2	2	0	64x64x128
Convolution	3	1	0	64x64x256

3.2.1. Artık birim (residual unit)

Artık birimler, sequential ağının ana yapı taşıdır. Son birkaç yılda derin sinir ağlarında yapılan son gelişmeleri birleştirmektedir. Şekil 3.11., artık ünitenin genel diyagramını göstermektedir.

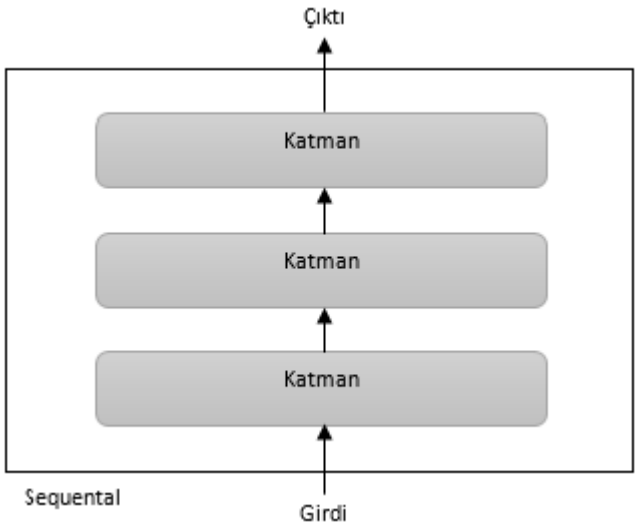


Şekil 3.11. Geleneksel öğrenme ve transfer öğrenme

Artık birimlerin kimlik eşlemesi, son evrişimli katmanın çıktısına elemanlar halinde eklenir. Bu tür atlama katmanı, derin sinir ağlarını optimize etmeye ve eğitmeye yardımcı olur [33]. Artık birimdeki her evrişimden sonra parti normalizasyonu gerçekleştirilir. Toplu normalizasyonun derin sinir ağlarının eğitim süresini büyük ölçüde azalttığı ve düzenleyici bir etkiye sahip olduğu gösterilmiştir. Kalan üniteye giren ve çıkan özelliklerin sayısı ağ genelinde 256'dır [32]. 3x3 evrişimden önce ve sonra 1x1 evrişimin nedeni, artık birim için saklanması gereken ağırlıkların sayısını azaltmaktır. 256 özelliğe sahip 3x3 evrişim, 256×9×256×590000 ağırlık gerektirir. Özellikleri azaltmak için 1x1 evrişim, ardından 3x3 evrişim ve 256 özelliğe geri dönmek için 1x1 evrişim gerektirir: 256×1×128+128×9×128+128×1×256≈213000 ağırlık. Özelliklerin yarısının kullanılması, bellekte kaydedilen parametre miktarını tam özelliklerle 3x3 evrişime kıyasla yaklaşık %64 azaltır. GPU'da bellek tasarrufu, daha derin ağlara ve daha büyük parti boyutlarına olanak tanıdığından, bu da eğitimi ve toplu normalleştirme verimliliğini hızlandırdığı için önemlidir.

3.2.2. Sequential modeli

Sequential model ağın bir girdisi ve bir çıktısı varsayımıyla katmanlar doğrusal olarak dizilmişlerdir (Şekil 3.12.). Bu genelde doğru bir varsayımdır ve pratik uygulamalarda şu ana kadar Sequential modeli kullanılmıştır.

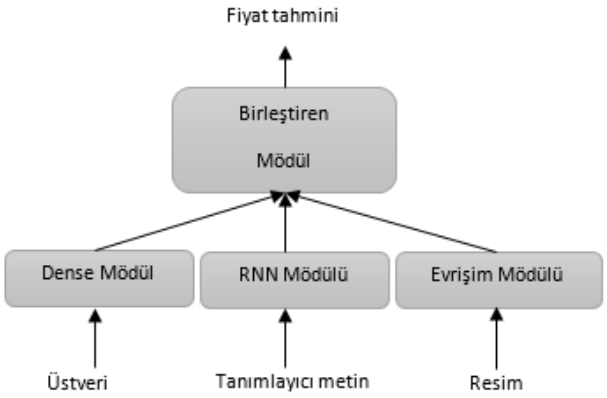


Şekil 3.12. Sequential model: Katmanların doğrusal dizilimi

Ama bu yaklaşım bazı durumlarda yeterince esnek değildir. Bazı ağlar birden fazla bağımsız girdi gerektirebilir, bazılarının birden fazla çıktısı olabilir ve bazı ağlardaysa katmanlar arasında farklı dolanmalar olabilir ve doğrusal üst üste katmanlar yerine çizge gibi görünürler.

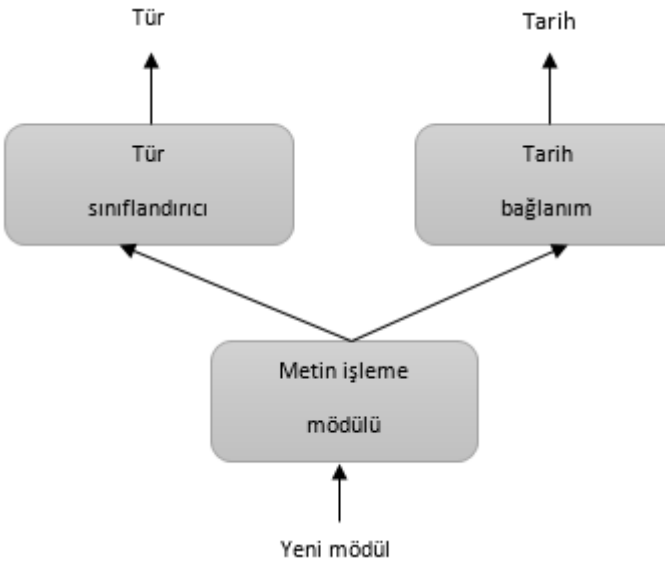
Mesela bazı görevler çoklu model girdiler gerektirir: Farklı kaynaklardan gelen girdileri birleştirip her veriyi farklı katmanlarda işlerler. İkinci el kıyafetlerin piyasa fiyatlarını tahmin etmeye çalışan bir model düşünün ve girdi olarak kullanıcıdan üst veri (ürünün markası, üretim yılı vb.), kullanıcı tanımlı metin ve ürünün resmini alsın. Sadece üst veri olsaydı bir elemanı kodlayıp tamamen bağlı bir ağla fiyat tahmin edilebilirdi. Sadece kullanıcı tanımlı metin olsaydı RNN veya 1B evrişimli sinir ağı kullanılabilir. Sadece resim olsa 2B evrişimli sinir ağı kullanılabilir. Bu üçü aynı anda nasıl kullanılabilir? Basit bir yaklaşım olarak üç ayrı model eğitilip, tahminlerinin ağırlıklı ortalaması alınabilir. Ama bu yaklaşım en iyi sonucun gerisinde kalabilir

çünkü modelin çıkardığı bilgiler gereksiz olabilir. Daha iyi bir yaklaşım bu modelleri aynı anda beraber eğitmek olabilir (Şekil 3.13.).



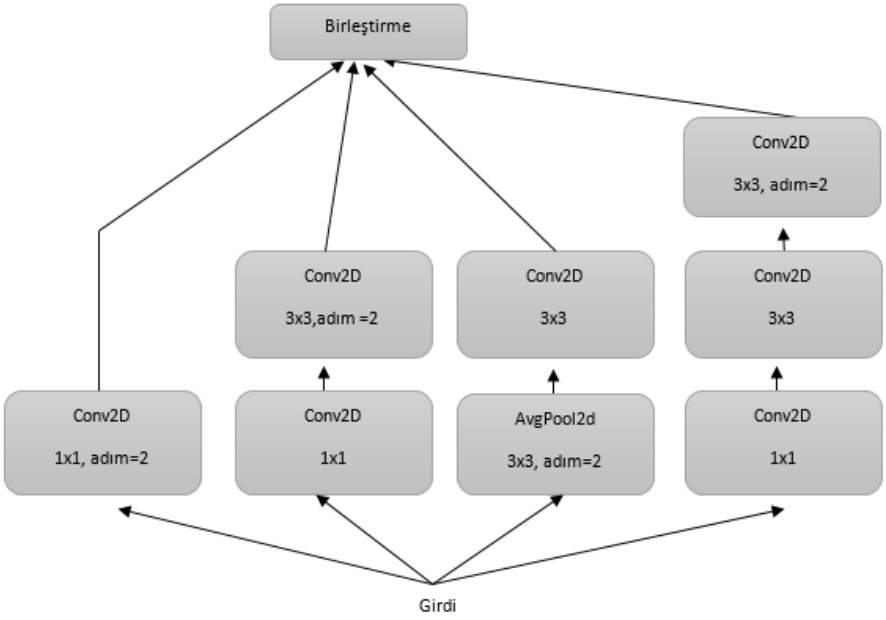
Şekil 3.13. Çok girdili model

Benzer şekilde, bazı görevler aynı girdiden farklı hedef çıktıları gerektirebilir. Kısa bir hikâye ve masalın metninden türünü (romantik, korku vb.) ve yaklaşık olarak yazıldığı tarihi tahmin etmek isteyebiliriz. Biri türünün, diğeri tarihinin tahmini için iki farklı model eğitebiliriz. Bu özellikler istatiksel olarak bağımsızdır ve ikisini beraber öğrenen bir model eğitebiliriz. Böyle bir modelin iki çıktısı ya da başı (Şekil 3.14.) olacaktır. Tür ve tarih arasındaki korelasyonu düşünerek tarih bilgisini bilmek ağın türler uzayında daha zengin ve iyi gösterimler öğrenmesine yardım edebilir veya tam tersi geçerlidir.

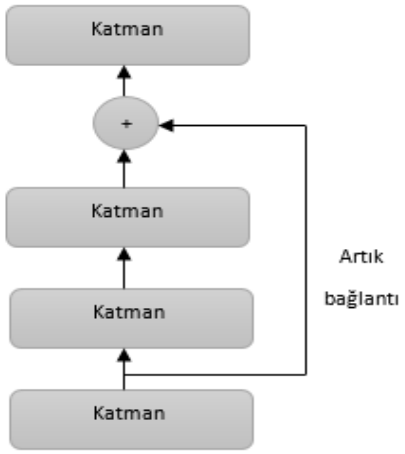


Şekil 3.14. Çoklu çıktılı (ya da çok başlı) model

Ek olarak çoğu yeni sinir ağı mimarisi doğrusal olmayan ağ topolojisi gerektirmektedir. Ağ yapısı yönlü çevrimsiz çizgiseldir. Inception (Google'da Szegedy vd. tarafından geliştirilmiştir) ailesi, girdisini paralel evrişim dallarında işleyip çıkışta tekrar bir tensör hâline getiren ve Inception adı verilen modüllere dayanır (Şekil 3.15.). Ayrıca yine son zamanlarda ResNet ailesiyle başlayan (Microsoft'tan He vd.) artık bağlantı kullanan modeller yaygınlaşmaktadır. Artık bağlantı (Şekil 3.16.) önceki katmanların gösterimlerini sonraki katmanların veri akışına eklenmesiyle yapılır. Bunu yapmak için önceki katmanın çıktı tensörü sonraki katmanın çıktı tensörüne eklenir. Böylece veri işleme akışında olası bilgi kayıplarının önüne geçilmesine yardımcı olur. Çizgesel ağların daha birçok örneği mevcuttur.



Şekil 3.15. Inception modülü: Birçok paralel dalda evrişim işlemi

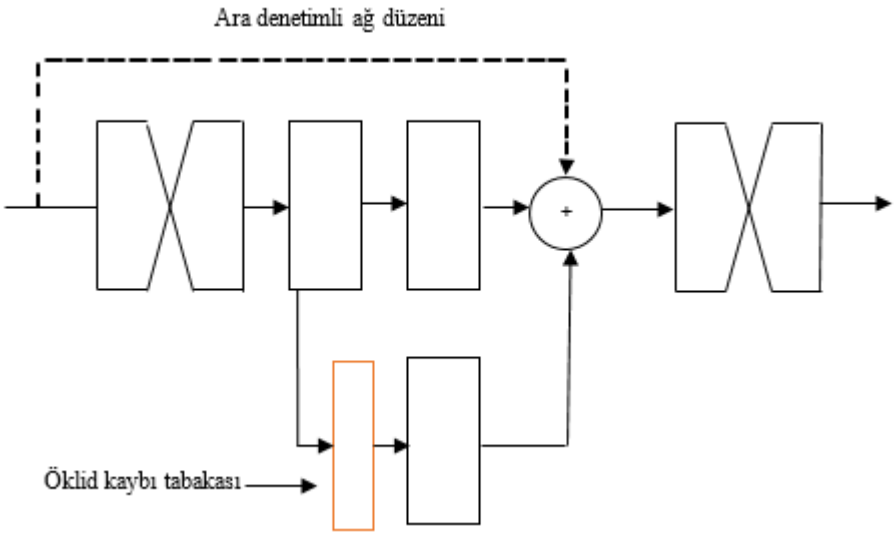


Şekil 3.16. Artık bağlantı: Önceki bilgiyi giden çıktıya eklemek

3.2.3. Bağlantı katmanı

Ağ, bir ara denetimle bağlantılı iki sequential modülünden oluşur. Bu bağlantı katmanı Şekil 3.17.'te görülebilir. İlk sequential modülünün çıktısı, ikinci sequential modülünden önce element bazında tekrar toplanmadan iki yöne bölünür. Bir yol iki artık birimden geçer ve diğeri 16 kilit nokta için tahminlerde bulunur ve bir öklid kaybı

işlevi uygulanır. Tüm artık birimlerin bir atlama katmanı olduğundan, ilk sequential modülüne yapılan orijinal girdi de ikinci sequential modülüne yapılan girdinin bir parçasıdır.



Şekil 3.17. Bağlantı katmanı

Şekil 3.18., bağlantı katmanındaki tahminlerin ve son tahminin nasıl olduğunu göstermektedir. Her sequential modülü tarafından tahminin kademeli olarak yeniden tanımlandığı görülebilir.

(a) Sağ dirsek (b) Sağ bilek

Şekil 3.18. Orta ve son tahminler

3.2.4. Eğitim ayrıntıları

Eğitim, Windows 8.1 işletim sistemine sahip bir bilgisayarda yapılmıştır. Eğitim için seçilen çerçeveye Keras adı verilir. Tablo 3.2., eğitim sırasında kullanılan meta parametreleri göstermektedir.

Tablo 3.2. Eğitim için kullanılan parametreler

Optimization algorithm	SGD
Learn rate	1 e-5
Momentum	0,95
Batch size	16

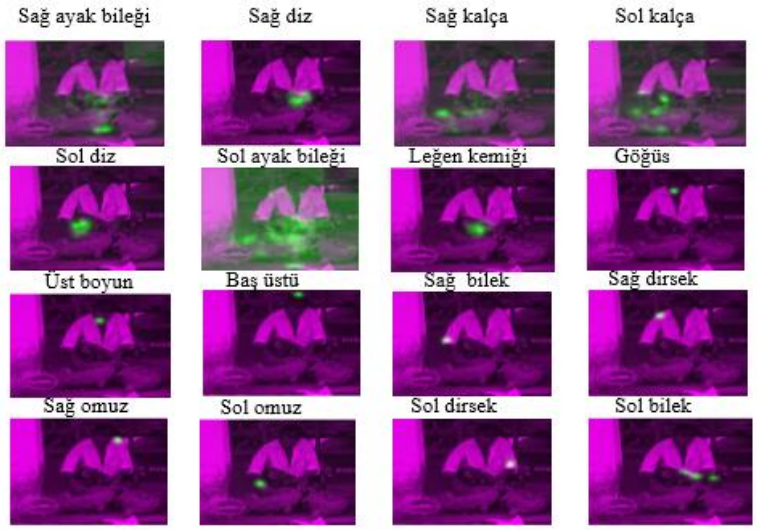
3.3. Görsel Değerlendirme

Ağın tahminlerini görselleştirmenin bir yolu, tahmin edilen ısı haritalarını orijinal görüntünün üzerine yerleştirmektir. Bu, hangi tür eklemlerin zor olduğuna ve ağın ne tür hatalar yapmaya eğilimli olduğuna dair bir fikir verir ve özellikle testin erken aşamasında yardımcı olur. Şekil 3.19., tahmin yapmak için kullanılan bir test görüntüsünü göstermektedir.

Şekil 3.19. Tahmin yapmak için kullanılan görüntü

Şekil 3.20.'de, ağın her bir anahtar nokta için tahmininde ne kadar isabetli olduğunu görmek mümkündür. Ayak bileklerinin tahminleri, ağın bu anahtar noktaların konumu

ile karıştırıldığını ve bunun da tıkanmadan kaynaklandığını göstermektedir. Tıkanmış anahtar noktaların tahminlerinin gücü, görünür anahtar noktalara kıyasla düşüktür.



Şekil 3.20. Ağ tahminlerini görselleştirmenin daha kolay bir yolu, her bir anahtar noktanın yalnızca maksimum değerini göstermek ve bunları bir iskelete bağlamaktır. Şekil 3.21.'de bir örnek gösterilmiştir. Çok düşük kaliteli tahminleri atmak için 0.1 eşiği kullanılır.

Şekil 3.21. Dizlerin ve ayak bileklerinin tahminleri eşiğin üzerinde değildir ve dahil edilmemiştir

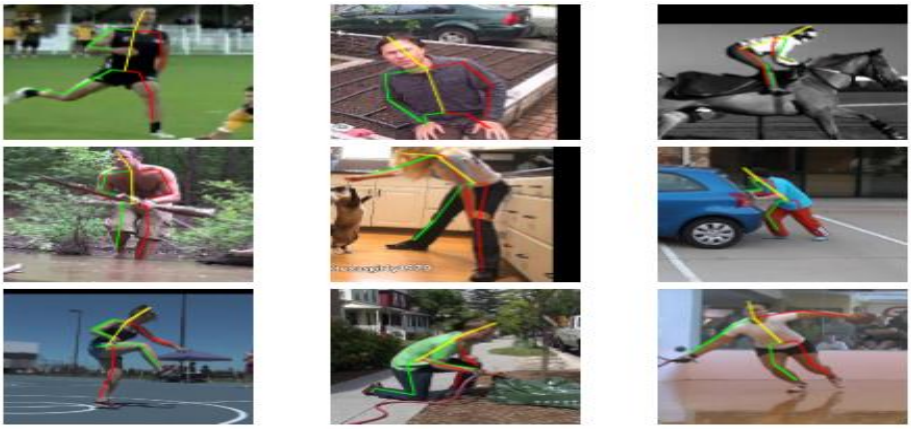
3.3.1. PCKh kullanarak değerlendirme

Bir deneyin ağın performansını iyileştirip iyileştirmediğini değerlendirmek için, bir tür değerlendirme ölçütü gereklidir. Bu, tercihen deney sırasında geliştirilebilecek tek bir sayıdır. MPII insan pozu veri kümesinin değerlendirilmesi için PCK (doğru anahtar

nokta olasılığı) ölçüsü sıklıkla kullanılır. MPII, çeşitli mesafeler için PCK değerini hesaplamak için bir Jupiyter araç takımı sağlar.

BÖLÜM 4. UYGULAMA VE SONUÇLAR

Şekil 4.1.'deki örnek tahminlerde görülebileceği gibi, ağ çeşitli arka planları, pozları ve kıyafetleri işleyebilir. MPII veri kümesinde her görüntü için bu parametreler hakkında bilgi bulunmadığından, arka plandaki ve giysideki değişikliklerin tam olarak ne kadarının tahmin doğruluğunu etkilediğini ölçmek zordur. Bununla birlikte, tahminleri incelerken, ağın bu koşullardaki değişikliklere göre oldukça değişmez olduğu görülmektedir. Ağ, tıkanmamış ve normdan çok fazla sapmayan bir poz tahmin ederken genellikle birkaç hata yapar.



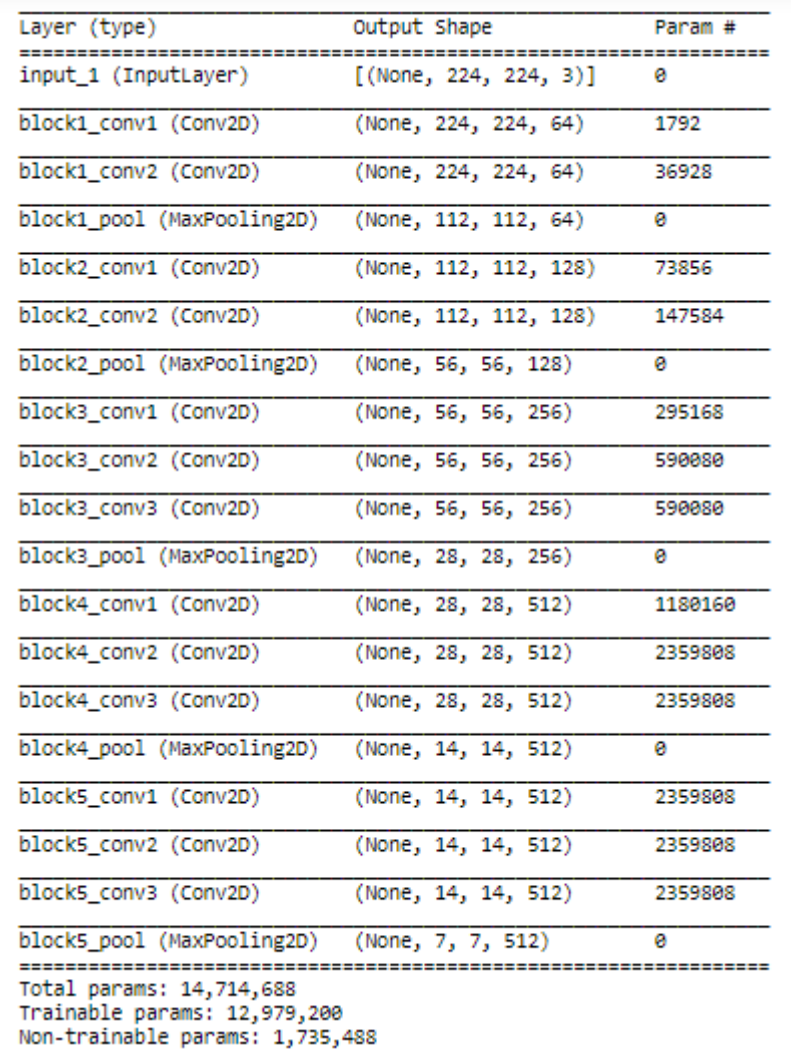
Şekil 4.1. MPII test setiyle ilgili tahminler

Kayıp, eğitim ve doğrulama ile hesaplanır ve yorumu, modelin bu iki set için ne kadar iyi performans gösterip göstermediği ile elde edilir. Doğruluktan farklı olarak, kayıp bir yüzde değildir. Eğitim veya test setlerinde her örnek için yapılan hataların toplamıdır. Aşağıdaki şemalarda, iki farklı modelin kayıplarını temsil eden iki grafik vardır, sol grafiğin yüksek kaybı ve sağ grafiğin düşük kaybı göstermektedir (Şekil 4.1.).

Şekil 4.2. Öğrenme oranının kayıp üzerine etkisi

4.1. Sonuçlar

Modelimizin Eğitim ve test sonuçlarını alabilmemiz için VGG16 ve Resnet50 olmak üzere iki model kullanılmıştır. Önce VGG16 modeli ile deneme yapılmıştır. Kullandığımız VGG16 evrişim tabanının katman listesi Şekil 4.3.'deki gibidir.



Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0

Total params: 14,714,688
Trainable params: 12,979,200
Non-trainable params: 1,735,488

Şekil 4.3. Kullanılan modellerden birinin sonuçlarına bir örnek

Son nitelik haritasının şekli (7,7,512) şeklindedir. Bunun üzerine tam bağlantılı ağ katmanını ekleyebiliriz. Bu noktada iki farklı yol seçebiliriz.

Evrişimsel tabanı kendi veri sitemizde çalıştırıp sonuçlarını Numpy dizisi olarak diske kaydedip bunu daha sonra kendi başına bir girdi olarak düşünüp tamamen bağlı bir sınıflandırıcıya gönderebiliriz. Bu çözüm basit ve hızlı olacaktır. Çünkü evrişimsel taban bu sürecin en maliyetli parçası ve her resim evrişimsel tabandan sadece bir kere geçecektir. Fakat aynı sebepten dolayı bu teknik veri seti çeşitlendirmesini kullanmamıza imkân tanımamaktadır.

Modelimi sahip olduğumuz tabanın (conv_base) üzerine Dense katman eklenerek ve tüm girdilerle en baştan çalıştırılarak denenmiştir. Bu şekilde veri seti çeşitlendirmeyi kullanmak da mümkündür. Fakat tüm resimler her epokta evrişimsel tabandan geçeceğinden çok daha maliyetli olacaktır.

Bu çalışmada iki teknik de incelenmiştir ve işletim süresi daha fazla sürmesine rağmen, sonuçlarımız ikinci yönteme göre alınmıştır.

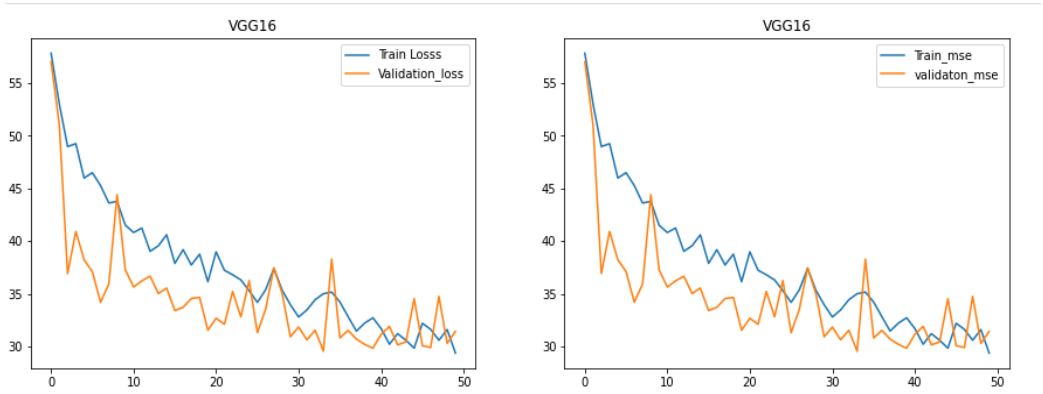
4.2. VGG16 Model Deneme Sonuçları

Uygulama VGG16 model üzerinde test edilmiş ve bu deneme süresi altı günde tamamlanmıştır. Yalnızca yeterince ayrılmış kişi ek açıklamaları kullanılmış ve toplamda yaklaşık 24.000 ek açıklama vardır. VGG16 modeli için kabaca 24000 görüntü kullanılmıştır, bunların 20000'i eğitim için ve geri kalan 4000'i test içindir. Modelimiz 10 epok olarak çalıştırılmış ve yüzde 87 başarıyla sonuçlanmıştır.

```
Epoch 00005: val_loss did not improve from 35.34702
Epoch 6/10
20000/20000 [==============================] - 611s 254ms/step - loss: 17.8991 - mean_absolute_error: 17.8991 - val_loss:
19.9042 - val_mean_absolute_error: 19.9042

Epoch 00006: val_loss did not improve from 35.34702
Epoch 7/10
20000/20000 [==============================] - 3793s 2s/step - loss: 14.9169 - mean_absolute_error: 14.9169 - val_loss: 1
7.8837 - val_mean_absolute_error: 17.8837

Epoch 00007: val_loss did not improve from 35.34702
Epoch 8/10
20000/20000 [==============================] - 640s 267ms/step - loss: 17.3158 - mean_absolute_error: 17.3158 - val_loss:
19.1339 - val_mean_absolute_error: 19.1339

Epoch 00008: val_loss did not improve from 35.34702
Epoch 9/10
20000/20000 [==============================] - 613s 255ms/step - loss: 15.1196 - mean_absolute_error: 15.1196 - val_loss:
20.3745 - val_mean_absolute_error: 20.3745

Epoch 00009: val_loss did not improve from 35.34702
Epoch 10/10
20000/20000 [==============================] - 2610s 1s/step - loss: 10.8319 - mean_absolute_error: 10.8319 - val_loss: 1
3.6980 - val_mean_absolute_error: 13.6980

Epoch 00010: val_loss did not improve from 35.34702
```

Şekil 4.4. VGG16 model deneme sonuçları

4.3. VGG16 Model Grafik Sonuçları



Şekil 4.5. VGG16 modelinin dönem ve kayıp grafikleri

Bir modelin kayıp düşüşlerinin hızlı bir şekilde alındığı açıkça görülmektedir ve makul doğrulukla çıkarımda oldukça hızlıdır. Bu, aktarım öğrenmesinin ne kadar güçlü, ancak basit olabileceğinin açık bir örneğidir.

4.4. Değerlendirme

Eğitim ve doğrulama performansı oldukça iyidir, ancak görünmeyen verilerdeki performans belirsizdir. Orijinal veri setimiz iki ayrı bölüme ayırmıştık. Burada hatırlanması gereken önemli nokta, test veri kümesinin eğitim veri kümesiyle benzer ön işleme tabi tutulması gerektiğidir. Bunu hesaba katmak için, işleve beslemeden önce test veri kümesini de ölçeklendiririz.

4.5. Resnet50 Modeli

Keras Uygulamalarından önceden eğitilmiş bir model, tahmin yapmak için önceden kalibre edilmiş ağırlıkları kullanmanıza izin verme avantajına sahiptir. Bu durumda, Imagenet'in ağırlıklarını kullanılabilir ve ağ bir ResNet 50 modeli olur. Include_top = False seçeneği, Dense katmanlarını kaldırarak özellik çıkarılmasına izin verir.

Kullanılan ResNet 50 modelinin evrişim tabanının katman listesi Şekil 4.6.'da gösterilmiştir.

```
Model: "resnet50"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_3 (InputLayer)            [(None, 224, 224, 3) 0
__________________________________________________________________________________________________
conv1_pad (ZeroPadding2D)       (None, 230, 230, 3)  0           input_3[0][0]
8, 28, 128)  512         conv3_block3_2_conv[0][0]
__________________________________________________________________________________________________
conv3_block3_2_relu (Activation (None, 28, 28, 128)  0           conv3_block3_2_bn[0][0]
__________________________________________________________________________________________________
conv3_block3_3_conv (Conv2D)    (None, 28, 28, 512)  66048       conv3_block3_2_relu[0][0]
__________________________________________________________________________________________________
conv3_block3_3_bn (BatchNormali (None, 28, 28, 512)  2048        conv3_block3_3_conv[0][0]
__________________________________________________________________________________________________
conv5_block3_add (Add)          (None, 7, 7, 2048)   0           conv5_block2_out[0][0]
                                                                 conv5_block3_3_bn[0][0]
__________________________________________________________________________________________________
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0           conv5_block3_add[0][0]
==================================================================================================
Total params: 23,587,712
Trainable params: 3,415,552
Non-trainable params: 20,172,160
__________________________________________________________________________________________________
None
```

Şekil 4.6. Kullanılan Resnet50 modelinin sonuçlarına bir örnek.

İki modelimiz olan Resnet50 modeli için yine 24000 ek açıklama kullanılmıştır. Bunların 20000'i eğitim, geri kalanı test içindir. Modelimiz yine 10 epok olarak çalıştırılmış ve bu defa modelimiz yüzde 67 başarıyla sonuçlanmıştır. Böylece VGG16 modelinin Resnet50 modeline göre daha iyi sonuçlar verdiği görülmüştür.

BÖLÜM 5. SONUÇLAR VE GELECEK ÇALISMALAR

Bu çalışmada kullanılan MPII veri kümesinin analizi yeni yaklaşımlardan biridir. Önceki yöntemler, pozların veri kümesinde ne kadar yaygın olduğuna veya normdan ne kadar saptıklarına odaklanmamıştır. MPII veri setindeki pozların önceki analizleri, pozları farklı vücut pozu ve görüş noktası kümelerine ayırmaya odaklanmıştır. Bu, farklı ağların farklı poz ve bakış açıları üzerinde nasıl çalıştığını değerlendirmek için yapılır. Sonuçlar, bu tezdeki bulgularla tutarlıdır ve yüksek poz skoruna sahip pozların tahmin edilmesi daha zordur.

Ağın performansı daha sıkı bir veri büyütme ile geliştirilebilir. Bunu yapmanın bir yolu, görüntüyü rastgele bir açıyla döndürmek ve eğitim sırasında görüntüyü ağa beslemeden önce rastgele bir faktörle yakınlaştırmaktır. Bu, eğitim verilerindeki kilit noktaların daha geniş bir dağılımına yol açacaktır. Başka bir yöntem olarak eğitim ve test verilerini artırarak daha iyi sonuçlar elde edebiliriz. Bu, bilgisayarın çalışma gücüne bağlı olarak değişir.

Ayrıca, ağın MPII veri kümesindeki pozların dağılımına aşırı öğrenme sorununu da azaltacaktır. Aşırı öğrenme sorunu, daha çeşitli pozlarla tercih edilen eğitim verilerini artırarak da ele alınabilir.

Ağın iyi çalışması için görüntünün merkezinde açıklama yapılacak kişiye sahip olması gerekir ve bu bilgiler MPII veri kümesinden sağlanır. Açıklama yapılacak kişinin konumunun bilinmediği gelecekteki bir uygulamada, görüntüdeki kişiyi bulmak için bir R-CNN (bölge önerilerini bulmak için bir CNN kullanan bir yöntem [32], [33]) ağı kullanılabilir. Önerilen bölge daha sonra tam pozu bulmak için poz tahmin ağına beslenebilir.

KAYNAKLAR

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 3686–3693, 2014, doi: 10.1109/CVPR.2014.471.

[2] B. Tekin, A. Rozantsev, V. Lepetit, and P. Fua, "Direct prediction of 3D body poses from motion compensated sequences," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 991–1000, 2016, doi: 10.1109/CVPR.2016.113.

[3] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pose," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1263–1272, 2017, doi: 10.1109/CVPR.2017.139.

[4] H. Y. F. Tung, H. W. Tung, E. Yumer, and K. Fragkiadaki, "Self-supervised learning of motion capture," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5237–5247, 2017.

[5] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to Estimate 3D Human Pose and Shape from a Single Color Image," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 459–468, 2018, doi: 10.1109/CVPR.2018.00055.

[6] N. Sarafianos, B. Boteanu, B. Ionescu, and I. A. Kakadiaris, "3D Human pose estimation: A review of the literature and analysis of covariates," *Comput. Vis. Image Underst.*, vol. 152, pp. 1–20, 2016, doi: 10.1016/j.cviu.2016.09.002.

[7] H. Rhodin *et al.*, "Learning Monocular 3D Human Pose Estimation from Multi-view Images," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, no. January 2019, pp. 8437–8446, 2018, doi: 10.1109/CVPR.2018.00880.

[8] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, “Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach,” *Proc. IEEE Int. Conf. Comput. Vis., vol. 2017-Octob, pp. 398–407, 2017, doi: 10.1109/ICCV.2017.51.*

[9] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, “End-to-End Recovery of Human Shape and Pose,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 7122–7131, 2018, doi: 10.1109/CVPR.2018.00744.

[10] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, “Neural body fitting: Unifying deep learning and model based human pose and shape estimation,” *Proc. - 2018 Int. Conf. 3D Vision, 3DV 2018*, pp. 484–494, 2018, doi: 10.1109/3DV.2018.00062.

[11] D. C. Luvizon, H. Tabia, and D. Picard, “Consensus-based optimization for 3D human pose estimation in camera coordinates,” *arXiv*, 2019.

[12] D. C. Luvizon, D. Picard, and H. Tabia, “2D/3D pose estimation and action recognition using multitask deep learning,” *arXiv*, 2018.

[13] V. Ramakrishna, T. Kanade, and Y. Sheikh, “Reconstructing 3D human pose from 2D image landmarks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7575 LNCS, no. PART 4, pp. 573–586, 2012, doi: 10.1007/978-3-642-33765-9_41

[14] W. Zeng, W. Ouyang, P. Luo, W. Liu, and X. Wang, “3D human mesh regression with dense correspondence,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 7052–7061, 2020, doi: 10.1109/CVPR42600.2020.00708.

[15] W.Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang, “3D Human Pose Estimation in the Wild by Adversarial Learning,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, no. April 2020, pp. 5255–5264, 2018, doi: 10.1109/CVPR.2018.00551.

[16] H. Yasin, U. Iqbal, B. Kruger, A. Weber, and J. Gall, “A dual-source approach for 3D pose estimation from a single image,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, no. October, pp. 4948–4956, 2016, doi: 10.1109/CVPR.2016.535.

[17] J. Shotton *et al.*, “springerreference_978-0-387-30771-8.pdf,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2821–2840, 2013, doi: 10.1109/TPAMI.2012.241.

[18] A. R. Guide, *Computer vision 3*. 2016.

[19] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition,” *Int. J. Comput. Vis.*, vol. 61, no. 1, pp. 55–79, 2005, doi: 10.1023/B:VISI.0000042934.15159.49.

[20] D. Ramanan, “Learning to parse images of articulated bodies_Unknown_Unknown_Ramanan.pdf.”

[21] A. Toshev and C. Szegedy, “DeepPose: Human pose estimation via deep neural networks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1653–1660, 2014, doi: 10.1109/CVPR.2014.214.

[22] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9912 LNCS, pp. 483–499, 2016, doi: 10.1007/978-3-319-46484-8_29.

[23] F. Chollet, *Deep Learning with Python, Manning*. 2018.

[24] N. Ketkar and J. Moolayil, *Deep Learning with Python*. 2021.

[25] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” pp. 1–31, 2016, [Online]., Erişim Tarihi: 03.04.2021.

[26] D. Hubel and T. Wiesel, “<Jphysiol01104-0228.Pdf>,” *J. Physiol.*, pp. 215–243, 1968, [Online]., Erişim Tarihi: 09.04.2021.

[27] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8689 LNCS, no. PART 1, pp. 818–833, 2014, doi: 10.1007/978-3-319-10590-1_53.

[28] F. Li, J. Johnson, and S. Yeung, “Lecture 1 : Introduction Welcome to CS231n,” pp. 1–48, 2017.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.

[30] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” *Proc. COMPSTAT 2010 - 19th Int. Conf. Comput. Stat. Keynote, Invit. Contrib. Pap.*, pp. 177–186, 2010, doi: 10.1007/978-3-7908-2604-3_16.

[31] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 1, pp. 448–456, 2015.

[32] C. Szegedy *et al.*, “Going deeper with convolutions,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

[33] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[34] D. Silver *et al.*, “Mastering the game of Go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, 2016, doi: 10.1038/nature16961.

ÖZGEÇMİŞ

Adı Soyadı : Firgat Muradli

ÖĞRENİM DURUMU

Derece	Eğitim Birimi	Mezuniyet Yılı
Yüksek Lisans	Sakarya Üniversitesi / Fen Bilimleri Enstitüsü / Bilgisayar ve Bilişim Mühendisliği	Devam ediyor
Lisans	Azerbaycan Teknoloji Üniversitesi / Otomasyon, Telekomünikasyon ve Enformasyon Fakültesi / Bilgi Teknolojileri ve Sistemleri Mühendisliği	2017
Lise	Nazim Haciyev Lisesi	2013

YABANCI DİL

Rusça

İngilizce

Türkçe

ESERLER

1. Human Pose Estimation Using Deep Learning

HOBİLER

Kitap okumak

Hatıra paraları toplamak