

# Derin öğrenme temelli nesne tespiti algoritmaları kullanılarak kişiye özgü reklam sunulması

Enes ULUTAŞ<sup>1</sup>, Hüseyin CENGİZ<sup>1</sup>, Musa Can YAZICIOĞLU<sup>1</sup>,  
Mustafa AKPINAR<sup>2\*</sup>

<sup>1</sup>Sakarya Üniversitesi Bilg. ve Bilişim Bil. Fak. Bilgisayar Müh. Böl., Esentepe kampüsü, Sakarya.

<sup>2</sup>Sakarya Üniversitesi Bilg. ve Bilişim Bil. Fak. Yazılım Müh. Böl., Esentepe kampüsü, Sakarya

Geliş Tarihi (Received Date): 11.02.2021

Kabul Tarihi (Accepted Date): 17.09.2021

## Öz

Günümüzde internet reklamları kişilerin çerez ve oturum bilgilerine erişerek kişiselleştirilmekte ve yüksek bir başarı elde etmektedir. Bu çalışmanın amacı internet reklamlarına benzer bir ortamın gerçek hayattaki reklamlar üzerinde uygulanmasıdır. Mağazaların giriş noktalarına veya ilan tahtalarına koyulacak bir kamera ve ekran ile gelen müşterilerin yaş, cinsiyet ve giyim tarzlarını inceleyerek kişiye özel reklamlar önerilmiştir. Böylelikle kullanıcıya beğenebileceği ürünleri gösterip kullanıcının ilgisini çekerek, satışların artırılması planlanmaktadır. Bir sonraki aşamada internette elde edilen görüntü verisetleri derin öğrenme algoritmaları ile incelenerek, görüntüdeki kişinin yaş, cinsiyet ve giyim tarzı analiz ve tespit edilmiştir. Giysi kısmında YOLOv3 algoritması kullanılmış olup, yaş ve cinsiyet kısmında önceden eğitilmiş olan bir model TensorFlow kütüphanesi yardımıyla tekrar eğitilerek kullanılmıştır. Eğitimler tamamlandıktan sonra elde edilen modellerin tahmin sonuçlarına göre bir öneri sistemi oluşturulmuştur. Örneğin gömlek ve etek giyen genç bir kadına, mağazanın reklam envanterinde, genç kadınlar için bulunan etek veya gömlek reklamı kişiye özgü olarak gösterilmektedir. Daha sonra çalışma bir kamera yardımıyla kişilerin görüntüsü alınarak önerilerde bulunmuş ve sonuçlar kabul edilebilir belirlenmiştir.

**Anahtar kelimeler:** Nesne tespiti, kişiye özgü reklam, derin öğrenme.

Enes ULUTAŞ, ulutasenes07@gmail.com, <http://orcid.org/0000-0001-7004-6213>

Hüseyin CENGİZ, b161210111@sakarya.edu.tr, <http://orcid.org/0000-0001-6950-6673>

Musa Can YAZICIOĞLU, musa.yazicioglu@ogr.sakarya.edu.tr, <http://orcid.org/0000-0003-0860-1110>

\*Mustafa AKPINAR, akpinar@sakarya.edu.tr, <http://orcid.org/0000-0003-4926-3779>

# Personalized advertisement using deep learning-based object detection algorithms

## Abstract

*Today, internet ads are personalized by accessing people's cookies and session information and achieving high success. This study aims to apply in an environment similar to real-life advertising on Internet ads. Personal advertisements were suggested by examining the age, gender, and dressing styles of customers who came to the locations with a camera and screen to be placed at the entry points or billboards of the stores. Thus, it is planned to increase sales by showing the products that the users may like and attracting the user's attention. Next, the image datasets obtained from the internet were examined with deep learning algorithms, and the age, gender, and clothing style of the person in the image were analyzed and determined. YOLOv3 object detection algorithm was used in the clothing, and a model that was previously trained in the age and gender section was retrained with the help of the TensorFlow library. A suggestion system was created according to the estimation results of the models found after the training was completed. For example, a young woman, who wears a shirt and a skirt, is shown exclusively in the advertising inventory of the store, with a skirt or shirt advertisement for young women. Then, the study made suggestions by taking images of the people with the help of a camera, and the results were determined as acceptable.*

**Keywords:** *Object detection, personalized advertisement, deep learning.*

## 1. Giriş

Günümüzde ürün satışlarının miktarını belirleyen bir numaralı etken reklamlardır. Bu yüzden reklamların şirketler için önemi oldukça artmıştır. Büyük e-ticaret siteleri de reklamları kullanarak kullanıcılara ürünlerini ulaştırmak için farklı yollar denemektedir. Bu yolların ortak amacı doğru ürünü doğru kullanıcıya ulaştırmaktır. Günümüzde bu çok sık karşımıza çıkmaktadır. Google, Facebook gibi şirketler bizim bilgilerimizi, tarayıcı geçmişlerimizi izleyerek bizim ilgi alanlarımızı belirlemekte ve buna göre kullanıcılara en doğru ürünlerin reklamlarını sunmaktadır. İnternet sektöründe reklamlar bu şekilde çalışmaktadır.

Türkiye’de 2019 yılının ilk 6 ayında reklam harcamaları 5 milyar Türk lirasını geçmektedir [1]. Şirketler artık reklamlara bu kadar çok para vermenin karşılığını yüksek verimlilikte almak istemektedir. Şu an internet sektöründe reklam sistemleri oldukça verimli çalışmaktadır ve günümüzde bazı şirketler reklamlar üzerine çalışmalar yapmaktadır. Bu şirketlerin çoğu Google, Facebook, Intel – Dentsu bilindik firmalardır. Bu firmalar dışında DeepAds, M&C Saatchi gibi firmalarda reklam üzerine çalışmaktadır. Kişilerin tarayıcı geçmişleri ve bilgilerini inceleyerek hangi kullanıcının hangi ürüne ihtiyaç duyduğunu analiz etmektedir. Bu analizleri şirketlerden aldıkları reklamlar üzerinde uygulamaktadırlar. Reklam almak ve vermek isteyen şirketler bu uygulamalarını sırasıyla Google AdSense ve Google Adwords üzerinde yürütmektedir. Facebook; Instagram ve Whatsapp uygulamalarının da sahibidir. Bizim bu uygulamalar üzerindeki hareketlerimizi izleyerek bizim ilgi duyduğumuz şeyleri analiz etmekte ve

bize buna göre reklam çıkartmaktadır. Bu mantık ile Google mantığı aynıdır. Facebook bu sistemi “Facebook for Business” üzerinden yürütmektedir. Dentsu Japonya merkezli bir reklam şirketidir. Intel ile yaptığı iş birliği ile öncelikle otobanlara koyulan reklam tabelalarına kamera entegre edip, ardından Intel tarafından geliştirilen bir yapay zeka ile araçların hangi sınıfta olduğunu tespit etmektedir. Sonuç olarak da arabaların bulunduğu sınıfa göre tabelalardaki reklamları değiştirmektedir. Örneğin lüks bir araç geçerse golf reklamı, aile aracı geçerse lunapark reklamı gibi reklamlar göstererek reklamların doğru kişilere ulaşmasını sağlamaktadır [2]. DeepAds girişiminde kameraları kullanarak insanların cinsiyetlerini ayırt etmekte ve buna göre reklamlar sunmaktadır. Sonraki hedeflerinde ise kıyafet, konum duygu gibi ifadeleri kullanarak reklam doğruluğunu artırmayı hedeflemektedir [3]. Hedefleri bizim çalışmamıza benzemektedir.

Bu şirketler sistemlerini internet üzerinden sürdürmektedir. Bu çalışmada ise kişiye özgü reklamlar gerçek hayata somut olarak gerçekleştirilmiştir. Günümüzde artık her sokak başında reklam tabelaları görülmektedir ve normal olarak algılanmaktadır. Sadece sokaklar değil mağazaların içlerinde veya alışveriş merkezleri (AVM) gibi kalabalık alanlarda da reklamlar çok yaygın kullanılmaktadır. Bu tarz yerlerde reklamlar kullanıcıya yüksek doğrulukta ulaşmamaktadır. Bu çalışmanın da amacı olan kişiye özgü reklamların geliştirmeye açık olan bir nokta olduğu görülmüştür. Sistem çalışma mantığı: Bir reklam tabelası, önünden geçen kullanıcıları analiz ederek kişilere bu analize göre reklam sunmaktadır. Bir görüntüdeki kişinin önce yaş ve cinsiyetini analiz edip ardından kullanıcının üzerindeki kıyafetleri de analiz ederek sınıflandırılacak. Bu sınıflandırmaları derin öğrenme mimarilerini kullanarak oluşturulacak.

Bu çalışmanın yenilikçi yönleri sırasıyla 17 farklı kıyafet türünün derin öğrenme modelleri ile belirlenmesi sağlanması, kişiye özgü reklamların gösterileceği bir mimarinin tasarlanması ile buradan elde edilen reklam tahminlerinin gösterimi ve kıyafet türlerinin model sonuçları üzerindeki etkisinin belirlenmesidir. Bu yenilikçi yönlerden kıyafet türlerinin model sonuçları üzerinde etkisinin belirlenmesi literatüre katkı sağlayacaktır.

Bu çalışmada güncel derin öğrenme algoritmaları kullanılmıştır. Literatürde derin öğrenme bitkilerin tanımlanması [4–9], hastalık tespiti [10–14], görüntü işleme [15], cinsiyet tespiti [16] ve duygu analizi [17] gibi birçok alanda kullanılmıştır. Medikal görüntüler üzerine yapılmış derin öğrenme algoritmalarının derlemesi [18] çalışmasında gösterilmiştir. Bu çalışmalar dışında giyim türü tespiti için de çalışmalar bulunmaktadır. Bedeli ve arkadaşları yaptıkları çalışmada şehir gözetleme kameraları kullanılarak adli uygulamalarda kişilerin tespitinde giyim türü belirlenmeye çalışılmıştır [19]. Yaptıkları çalışmadaki test görüntülerinde %70 oranında etiketlemeyi doğru yaptığını belirlemişlerdir. Kullanılan Deep Fashion verisetinde doğrulama Loss değeri 6,03 olmuştur. Saxena ve arkadaşları da Giysi yardım görevi için giysi tanıma ve kavrama noktası algılama üzerine çalışmışlardır [20]. Yaptıkları çalışmada tam kollu üst, kolsuz üst ve kot pantolon olmak üzere üç tür kıyafet tespit edilmeye çalışılmıştır. Verisetinde tam kollu üst 194, kolsuz üst 180 ve kot pantolon 149 adet bulunmaktadır. Bu görüntüler 10 derecelik açılarla fotoğraflanmış ve 18.305 adet görüntü elde edilmiştir. Test verisetinde kolsuz üst giysilerin tekli ve çoklu görünümde sırasıyla %74 ve %96 doğrulukla tespit edildiği belirtilmiştir. Tam kollu üst giysileri tekli ve çoklu görünümde sırasıyla %40 ve %72 doğrulukla tespit edilmiştir. Tam kollu üst

giysilerinin kot pantolon ile karşılaştırıldığı görülmüştür. Dong ve arkadaşları da yaptıkları çalışmada giyim çoklu özelliklerini derin öğrenme algoritmasıyla belirlemeye çalışmışlardır [21]. Yaptıkları çalışmada eğitim verisiyle %65,96 ortalama doğruluk oranıyla kategorileri belirlenmiştir. Lin ve arkadaşları ise yaptıkları çalışmada Yahoo alışveriş sitelerine ait 161.234 elbise görüntüsünü kullanmışlar ve 15 kategoriye AlexNet modeli ile belirlemeye çalışmışlardır [22]. Cychnerski ve arkadaşları da evrişimli sinir ağlarından ResNet-50, SqueezeNet ve Single Shot MultiBox Detector (SSD) kullanarak elbise tespit etmeye çalışmışlardır [23]. Elde ettikleri modelin mAP değeri %42 olarak bulunmuştur. Liu ve arkadaşları tarafından DeepFashion veriseti kullanılarak FashionNet derin öğrenme mimarisi önerilmiştir [24]. FashionNet mimarisinin VGG-16 mimarisine benzer olduğunu söylemişler ve 50 ayrıntılı kategori tahmininde bulunmuşlardır. Han ve arkadaşları ise alışveriş web sitelerinden gelen yetersiz etiketlemeye sahip görüntü-metin verilerini kullanan nesne tespiti önermişlerdir [25]. Çalışmalarında elbise, üst giyim, pantolon, etek ve ceket kategorilerini diğer çalışmalar ile karşılaştırmışlardır. Gong ve arkadaşları ise çalışmalarında insan vücudunu ayırmaya çalışmışlardır [26]. Bu çalışmada uzuvların belirlenmesi dışında gözlük, üst giyim, elbise, şapka, kaban, çorap, pantolon gibi giyim kategorileri de belirlenmiştir. Literatürde konuyla alakalı yapılan çalışmalar konunun güncelliğini göstermektedir.

Bu bölümün başında reklamların günümüzdeki öneminden ve internet sektöründe nasıl çalıştığından bahsedilmiştir. Ardından Türkiye'nin reklam hacminden ve Dünya'daki reklamlar ile ilgili büyük çalışmalarda bulunan şirketlerden bahsedilmiştir. Mevcut sorundan bahsedilmiş ve bu sorunu nasıl çözüldüğü kısa biçimde aktarılmıştır. Son olarak da literatürde giyim kategorileme ile ilgili çalışmalar gösterilmiştir. Bu çalışmanın amacı derin öğrenme teknikleri kullanarak kişiye özgü reklam gösteriminin sağlanmasıdır. Bu amacı gerçekleştirmek için kullanılan çeşitli derin öğrenme algoritmalarından bir sonraki bölümde bahsedilmiştir. Üçüncü bölümde problem ve yaklaşım adım adım gösterilmiş olup, verisetleri ve hazırlanan modeller açıklanmıştır. Aynı bölümde eğitimde kullanılan parametreler de verilmiştir. Dördüncü bölümde kullanıcı arayüzü ve entegrasyonu anlatılmış servis mimarisi, veritabanı şeması ve kullanıcıya sunulan reklam gösterilmiştir. Beşinci bölümde verisetleri, hazırlanan modeller ve eğitim sonuçlarından bahsedilmiş, diğer çalışmalar ile karşılaştırılarak tartışılmıştır. Son bölümde sonuçlardan ve gelecek çalışmalardan bahsedilmiştir.

## 2. Derin öğrenme

Derin öğrenme, ham girdiden daha üst düzey özellikleri aşamalı olarak çıkarmak için birden çok katman kullanan bir makine öğrenme algoritmaları alt sınıfıdır. Örneğin, görüntü işlemede alt katmanlar kenarları tanımlayabilirken, yüksek katmanlar bir insanla ilgili rakamlar, harfler veya yüzler gibi kavramları tanımlayabilir.

Derin Öğrenme ilk defa 2012 yılında nesne sınıflandırma için yapılan, büyük ölçekli görsel tanıma (ImageNet) yarışmasında elde ettiği başarı ile dikkatleri üzerine çekmiştir [27]. Derin Öğrenmenin temelleri geçmişe dayansa da özellikle son yıllarda popüler olmasının iki sebebi büyük veri kaynaklarının ortaya çıkması ve bu işlemleri yapabilecek donanımsal gerekliliklerin karşılanabilir olmasıdır. Geleneksel makine öğrenmesi algoritmalarında sınıflandırma yapılırken ham verinin üzerinden manuel olarak özellik çıkartılması gerekir.

Konvolüsyonel Sinir Ağları (Convolutional Neural Network - CNN) günümüz derin öğrenme yaklaşımının temelini oluşturmaktadır [28]. Konvolüsyon işlemi konvolüsyon katmanının temelini oluşturmaktadır. Konvolüsyon işlemi veriye bağlı olarak bir, iki veya üç boyutlu olarak uygulanabilir. Bundan sonraki konvolüsyon işlemlerinin hepsi iki boyutlu olacaktır.  $n \times n$ 'lik bir görüntü,  $m \times m$ 'lik bir çekirdek (kernel) matrisi ile konvolüsyona sokulacak ve bir çıktı matrisi elde edilecektir. Burada görüntü matrisinin sol üst köşesinden başlayarak çekirdek matrisinin boyutlarındaki alan alınır ve çekirdek matrisi ile çarpılır. Bu değer çıktı matrisinin sol üst köşesinde bulunan ilk elemanı göstermektedir. Görüntü matrisi sağa ve aşağıya doğru iteratif olarak ilerler ve çekirdek matrisi ile çarpılır. Bu çarpımlar sonrası çıktı matrisinin boyutu  $(n-m+1) \times (n-m+1)$  kadar olacaktır. Çekirdek matrisinin boyutu ve elemanları isteğe göre seçilebilir. Literatürde genellikle kare matris tercih edilmektedir.

CNN algoritmasında kayan pencere yaklaşımı nesnelerin görsel üzerindeki yerlerini tespit etmede kullanılmaktadır. 2014 yılında tasarlanan R-CNN (Regional Based CNN) algoritmasında görsel bir ön işleme sokulmakta ve nesnelerin bölgeleri tespit edildikten sonra algoritmaya girdi olarak verilmektedir [29,30]. R-CNN mimarisinde nesne tespit ve eğitim süreçleri ardı ardına gerçekleştiği için uzun zaman almaktadır. Fast R-CNN mimarisi bu sorunların üstesinden gelmek için geliştirilmiştir [31]. R-CNN ile Fast R-CNN arasındaki temel fark CNN bir kez çalıştırılmakta ve nesne tespit önerileri arasından hesaplamaların paylaşımının yapılmasıdır. Faster R-CNN mimarisinde ise bölge önerisinde Region Proposal Network eklenmiştir [32].

### 3. Problem-yaklaşım ve hazırlanan modeller

#### 3.1. Problem-yaklaşım

Günümüzde internet reklamcılığında kullanılan kişisel öneri sisteminin yakaladığı başarının sokaklarda ve mağazaların önünde bulunan tabelalarda uygulanmamaktadır. Burada yer alan problem, oluşturulan reklamların hedef kitleye tam olarak ulaştırılmamasından dolayı istenilen ürünlerin hedef satış rakamlarının elde edilememesidir. Bu problemin çözümüne ilişkin yaklaşımımız aşağıdadır.

- Python script yardımıyla Webcam üzerinden alınan çerçeveler (frame) önceden eğitilmiş olan yaş, cinsiyet ve kıyafet modelleri üzerinden geçirilerek tahmin sonuçları alınır.
- 10 saniyede bir toplanan tahminlerde, insanların üzerinde en çok görülen kıyafet, cinsiyet ve insanların ortalama yaş filtreleri çıkarılır.
- Bu çıkarılan filtreler socket bağlantısıyla Node.js sunucusuna iletilir. React.js uygulaması socket bağlantısıyla Node.js sunucusundan filtreleri alır.
- React.js uygulamasında alınan filtreler .Net Core Web Application Programming Interface'ye (API) parametre olarak gönderilir. Web API'de ise bu filtrelere uygun kıyafetin resimleri React.js uygulamasına cevap (response) olarak döndürülür.
- Cevap olarak alınan reklam afişleri React.js uygulamasında kullanıcıya sunulur.

#### 3.2. Hazırlanan Modeller

Başlangıç olarak çalışmamızda tespit edilecek nesnelerin belirlenmesi gerekmektedir. Günümüzde bir kişi sokağa çıktığında çoğunlukla ayakkabı, alt giyim ve üst giyim olmak üzere bu üç kombin seçeneklerine karar vererek dışarıya, gezintiye, çıkmaktadır. Bu yüzden nesne tespiti yaparken ana sınıfları buna dikkat ederek seçilmiştir. Bir kişiye

sadece bu ana sınıflardan reklam önerisi yaparsak çok genel olacaktır. Bu yüzden bu üç ana sınıflara göre kırılımlar oluşturmasının uygun olduğu düşünülmüştür. Örnek olarak ayakkabı ana sınıfını ele alınacak olursa; bu ana sınıf için alt kırılım oluşturulmak istenirse aklımıza gelen sınıflar bot, günlük ayakkabı, topuklu ayakkabı, terlik gibi alt sınıflar olacaktır. Bu kırılımları oluşturarak kişiye özel gelişmiş bir reklam öneri sistemimiz olacaktır.

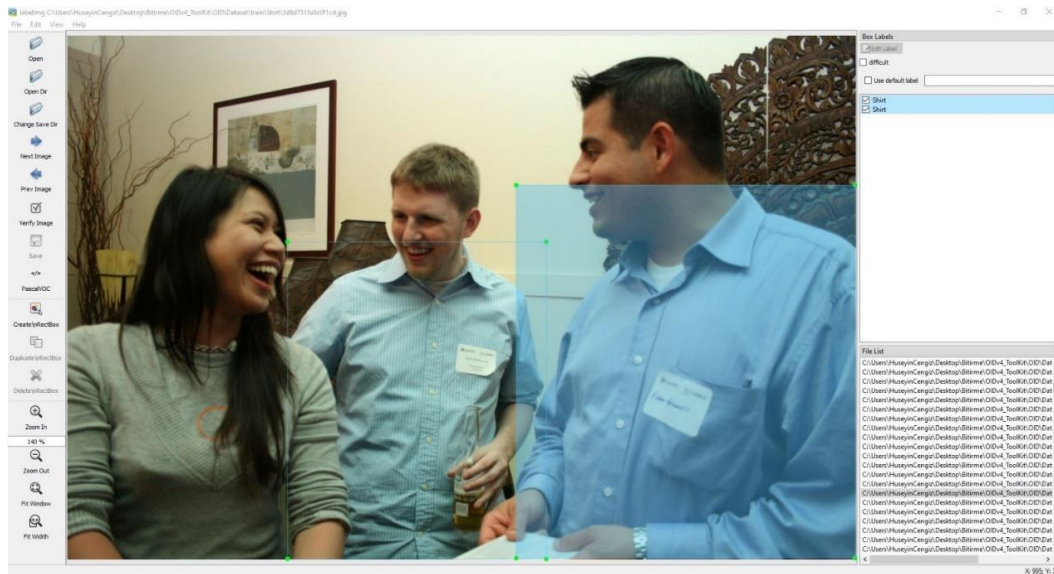
### 3.2.1 Verisetinin hazırlanması

Bu çalışmada üç ana sınıf (üst giyim, alt giyim ve ayakkabı), on yedi alt sınıfa bölünmüştür. Bu alt sınıflar; Bot, Günlük Ayakkabı, Topuklu Ayakkabı, Terlik, Kot Pantolon, Kumaş Pantolon, Kısa Pantolon, Etek, Elbise, T-Shirt, Gömlek, Kazak, Ceket, Kaban, Gözlük, Güneş Gözlüğü ve Şapka olarak sıralanmıştır (Tablo 1).

Alt sınıflara böldükten sonra sıra bu sınıflara ait verisetlerini oluşturmaya gelmiştir. Her bir sınıf için 1.000 etiketlenmiş fotoğraf elde etme hedefi koyulmuştur. T-Shirt, Kazak ve Günlük Ayakkabı sınıfları için önceden etiketlenmiş veri seti bulamamıştır. Diğer on dört sınıf için ise Google'ın Open Images Dataset V5 servिसinden yararlanarak etiketlenmiş halde her biri için 1.000 fotoğraf elde edilmiştir. Geri kalan etiketlenmemiş üç sınıf için ise 20 bin fotoğraftan oluşan bir albüm indirilmiştir. Bu albümde resimler etiketlenmiş şekilde bulunmamaktadır. Öncelikle bu 3 sınıfın her biri için en uygun olan 1000 tane fotoğraf seçilmiştir. Bu fotoğrafların etiketleme işlemi Python programlama dilinde yazılmış olan LabelImage uygulamasını kullanarak elle gerçekleştirilmiştir (Şekil 1).

Tablo 1. Kıyafetlerin türleri ve alt sınıfları.

Ana Sınıflar	Üst Giyim		Alt Giyim		Ayakkabı	
Alt Sınıflar	Güneş Gözlüğü	Gözlük	Kısa Pantolon	Kot Pantolon	Bot	Günlük Ayakkabı
	Gömlek	Kazak	Etek	Kumaş Pantolon	Terlik	Topuklu Ayakkabı
	T-Shirt	Ceket				
	Elbise	Kaban				
	Şapka					



Şekil 1. LabelImage uygulaması.

Elimizde olan her sınıf için 1.000'er tane toplam 17 bin fotoğraf modeli eğitmek için yeterli olmadığı görülmüştür. Fotoğraf sayımızı arttırmak için Python programlama dilinde yazılmış olan Impy paketini kullanarak ilk önce fotoğraflardaki sınıflara ait nesnelere eğer boyutu uygunsa 1.024x1.024 şeklinde kesip alınmıştır. Bu işlemden sonra elde edilen fotoğrafların üzerinde veri büyütme (data augmentation) tekniği uygulanarak yani fotoğraflar üzerinde bükme, eskitme, dikey döndürme, yatay döndürme, ölçeklendirme gibi işlemler yapılarak her sınıf için 5.000 etiketlenmiş fotoğraf elde edilmiştir. Her sınıftan 4.000 tane fotoğraf eğitim, 1.000 fotoğraf ise test işlemi için ayrıldı. Böylelikle eğitim için toplam 68.000, test için ise 17.000 etiketlenmiş fotoğraf elde edilmiştir.

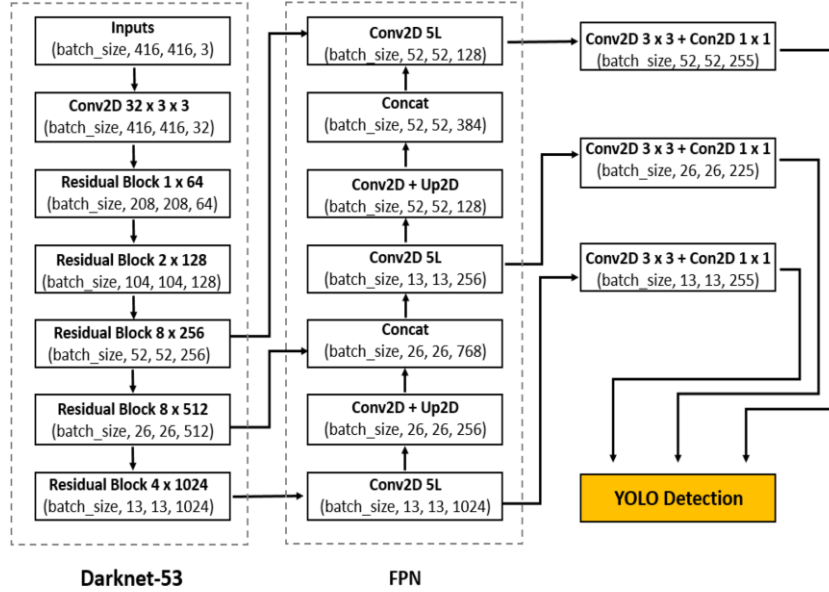
### 3.2.2 Eğitim

Kullanılacak veri seti hazırlandıktan sonra yapılacak ilk adım projeye uygun hızlı ve istikrarlı bir şekilde çalışan nesne tespit tekniğinin belirlenmesidir. Belirlenen nesne tespit tekniklerinde kullanılacak olan omurga CNN mimarisi seçildikten sonra uygun hiper parametreleri yerleştirilip eğitimin gerçekleştirilmesi gerekir. Her teknik için eğitim yapıldıktan sonraki aşama, modellerin vereceği çıktıları değerlendirilip uygun olan seçilir. Sonraki adım ise uygun olan modelin uygulamada kullanımı şeklindedir.

Uygulamada aranan en yüksek öncelikli özellik Raspberry Pi üzerinde çalışacak kadar hızlı olmalıdır. Çünkü bu model Raspberry Pi'in içine gömüldüğünde, insanlar geçerken hızlı bir şekilde görüntüdeki nesnelere tespit edip kişiye uygun reklam sunulmalıdır. İkinci özellik ise tespit işleminin projede beklenen çıktıyı verebilmesidir. Bu iki kısıt ile beraber beklenen çıktıyı makul bir şekilde vermesine rağmen çok büyük işlem gücü gerektirdiği için hızdan feragat etmemizi sağlayan Region Proposal temelli teknikler (R-CNN ve türevleri) eleştirilmiştir. YOLO (You Only Look Once) algoritması ise CNN algoritmaları ile karşılaştırıldığında daha hızlı çalışmaktadır [33–36]. YOLO dışında da hızlı çalışan algoritmalar olmasına karşın mAP (mean absolute precision) değerleri yetersiz kalmaktadır [36–38]. YOLO hem gerçek zamanlı tespit yapabilmekte hem de ortalama kesinlik (mAP) değerleri yeterli seviyededir.

Nesne tespitindeki diğer önemli bir nokta eğitimin başlangıcıdır. Günümüzde derin öğrenme algoritmalarının sıfırdan eğitime başlanması çok maliyetli bir işlemdir. Bu işlemler için grafik işlemciler (GPU) sıklıkla kullanılmakta ve genellikle eğitimler günler ve haftalar almaktadır. Eğer yüksek kapasite veya uzun bir zamana ihtiyacınız varsa fakat yeterli kaynağınız bulunmuyorsa, transfer öğrenmesi (transfer learning) gerçekleştirilmektedir. Transfer öğrenmesinde daha önce eğitilmiş bir ağ alınarak, istenen nesnenin tanıtılması için eğitilmesi gerçekleştirilmektedir. Örneğin ImageNet veraseti ile eğitilmiş olan AlexNet modelini alarak eğitim süresi kısaltılabilmektedir. Daha önce belirli nesnelere tanıyabilen model, istenilen nesneyi de tanımaya başlayacaktır. Burada önemli olan diğer bir ayar, çıkış sayısının tanınacak nesne çeşidi olmasıdır. On nesne tanıyacak sistem için son katmanda on adet çıkış olmalıdır. Tensorflow'un SSD de eğitilmiş olan mimarileri arasında Mobilenet v2 ve Inception v2 seçilmiştir. Darknet'in YOLO da eğitilmiş olan mimarileri arasında ise YOLOv3-416 seçilmiştir (Şekil 2). Bu mimarilerin COCO veri setinde eğitilmiş hazır ağırları TensorFlow ve DarkNet kütüphaneleri tarafından sağlanmaktadır. COCO; sınıflandırma, tespit ve takip etme, segmentasyon görevleri için hazırlanmış içinde 80 farklı nesneyi barındıran 200.000 görüntü bulunan açık kaynak bir veri setidir.

SSD Inception v2 ve SSD Mobilenet v2 için optimizasyon algoritması olarak gradyan temelli RMS\_PROP\_OPTIMIZER seçilmiştir. Stabil learning rate tercih edilmemiş onun yerine eksponansiyel olarak azaltma tekniği kullanılmıştır. İki mimaride de 0,004 ile başlanmış 0,95 faktörüyle azaltılmıştır. Minibatch ölçüğü 32 olarak belirlenmiştir. Eğitim 100.000 adım olarak planlanmıştır.



Şekil 2. YOLOv3-416 derin öğrenme mimarisi\* [39].

Yolov3-416 mimarisi için optimizasyon algoritması olarak rasgele gradyan inişi (stochastic gradient descent) seçilmiştir. Öğrenme katsayısı (Learning rate) olarak 0,001 kullanılmıştır. Batch olarak 64 değeri kullanılmıştır. Batch size'a 64 değeri vermek demek bir iterasyonda nöral ağınlık ağırlıklarını güncellerken 64 resmin kullanıldığını belirtir. Subdivisions değeri olarak 16 kullanılmıştır. Nöral ağını eğitirken batch size değerini 64 olarak kullanmak isteyebilirsin ama bazen batch size değerini 64 olarak kullandığında ekran kartının yeterli bellek değeri olmayabilir. Böyle zamanlarda ekran kartı herhangi bir zamanda batch / subdivision sayısı kadar resmini işleyecektir. Bu bölme işlemi sonucundaki değer batch boyutuna ulaştığında bir iterasyon tamamlandı sayılır. Momentum değeri 0,9 olarak kullanılmıştır. İterasyonlar arasındaki büyük ağırlık değişikliklerini cezalandırmak için momentum parametresi kullanılır. Decay değeri 0,0005 olarak kullanılmıştır. Tipik bir sinir ağı milyonlarca ağırlığa sahiptir ve bu nedenle eğitim verisinden herhangi birini kolayca ezberleyebilirler. Ezberlemek, basitçe eğitim verilerinde çok iyi ama test verilerinde kötü çıktı vermesi anlamına gelir. Ağ eğitim setindeki tüm fotoğrafları ezberlemiş yani çıktıyı genelleymemiştir. Bu sorunu azaltmanın yollarından biri ağırlıklar için büyük değeri cezalandırmaktır. Decay parametresi bu ceza değerini tutar.

Eğitim NVIDIA GTX 1080 TI 11 GB GPU, 16 GB RAM ve i7-6700HQ Intel işlemciye sahip bilgisayarda yapılmıştır. İlk olarak SSD MobileNet v2 modelini eğitirken mini batch sayısı 32 olacak şekilde planlanmıştır, ancak mini batch sayısı için 32 değerini

\*Bu görsel M. Laval tarafından yapılmış "Tomato detection based on modified YOLOv3 framework" çalışmasından alınmıştır (açık erişim (CC) lisanslıdır), M. Laval, "Tomato detection based on modified YOLOv3 framework", Scientific Reports, vol. 11, no. 1, 2021. Available: 10.1038/s41598-021-81216-5 [Accessed 8 September 2021].



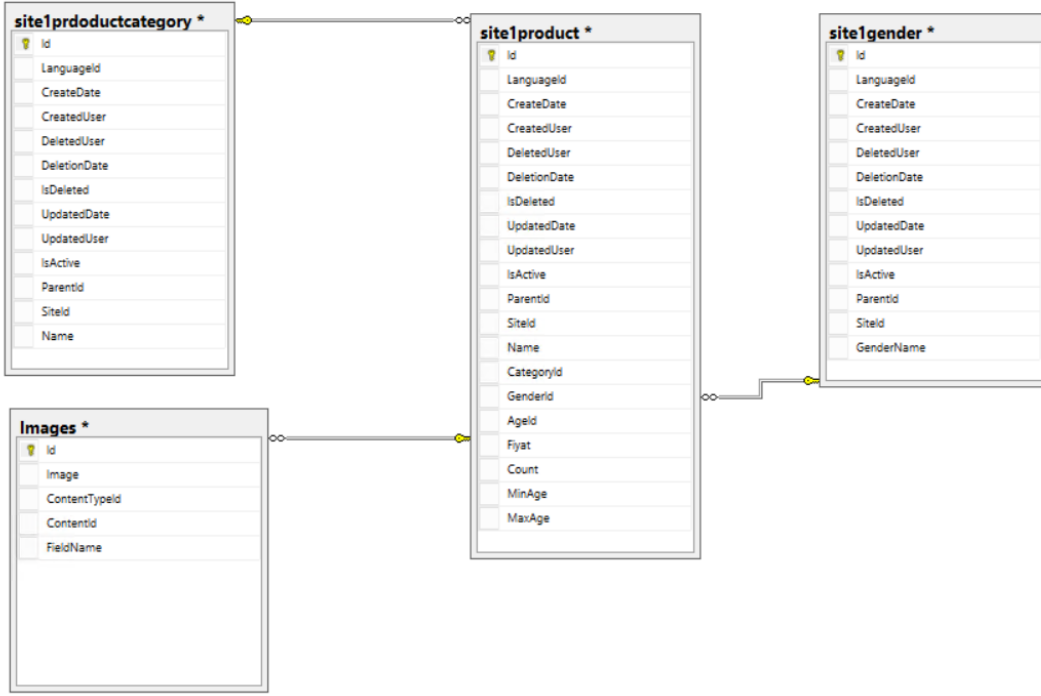
kullanmak 11 GB GPU belleği için fazla gelmiştir. Bundan dolayı “out of memory” hatası alınmıştır. Bu bilgisayarda eğitimin gerçekleştirilebilmesi için farklı minibatch sayıları denenmiştir ve 8 uygun değer olarak seçilmiştir. Bu minibatch sayısı ile eğitim gerçekleştirilmiştir. Optimum batch sayısını olarak belirlediğimiz 8 değerini kullanarak aynı şekilde SSD Inception v2 modeli eğitilmiştir. Diğer model, YOLOv3-416, için bir üst paragrafta bahsedilen parametreler kullanılarak model eğitilmiştir.

### **3.3. Yaş ve cinsiyet için uygun modelin bulunması**

Yaş ve cinsiyetin belirlenmesinde iki hazır model kullanılmıştır. Bunun sebebi, eğitim için harcanacak zaman ile gücün çok yüksek olmasıdır ve donanım olarak aynı anda bir tane eğitim cihazı (bilgisayar) olmasıdır. Eldeki bu bilgisayar kıyafet eğitimi için kullanılmıştır. Hazır model kullanılmasındaki diğer sebep ise yaş ve cinsiyet daha önceden çok fazla eğitimlerin yapılışlarıdır. Bu modeller yüksek doğrulukta sonuçlar bulmaktadır. Burada öncelikle Viola Jones metodunu kullanarak geliştirilmiş bir model test edilmiştir [40]. Bu test sonucunda çıktılar her ne kadar hızlı olsa da kişinin başı sadece dik pozisyonlardayken yaş ve cinsiyet analizi yapabildiğini göstermiştir. Bu noktada derin öğrenme ile oluşturulmuş modellerin bizim için daha iyi sonuç vereceğini düşünülmüştür. Bu hazır modelde Adience dataseti kullanılmıştır [41, 42].

## **4. Kullanıcı Arayüzünün Oluşturulması ve Entegrasyonu**

Bu bölümde veritabanı yapısı, kullanıcı paneli ve sistemin model çerçevesi gösterilmiştir. İlişkisel veri tabanı olarak MSSQL kullanılmıştır. Sistem dört temel tablodan oluşmaktadır. Bunlar; ürünler, ürünlerin kategorileri, cinsiyet ve resimlerin bulunduğu tablolardır (Şekil 3). Kullanıcının sisteme reklamları girebilmesi için de panel hazırlanmıştır (Şekil 4). Kullanıcı ekleyeceği ürünleri ve kategorilerini bu panelden yapmaktadır. Bu Panelin ön yüzü React.js kullanılarak hazırlanmıştır. Panelin bağlı olduğu bir .NET Core API bulunmaktadır. Bu API ön yüzden gelen verileri veri tabanına kaydederken (Object–relational mapping) ORM olarak Dapper kullanılmaktadır. Kullanıcılar bu sayfada ürünün genel bilgilerini, resimlerini ve ürünün sahip olduğu yaş aralığı gibi bilgileri girerek ürünü özelleştirebilmektedir. Kullanıcılar bu ekranda önceden eklediği içerikleri görebilir. İsterse güncelleyip silebilir.



Şekil 3. Veritabanı şeması.

Hazırlamış olduğumuz Python script 10 saniye boyunca çerçeve (frame) almaya devam etmektedir (Şekil 5). Script aldığı frameleri eğittiğimiz modele input olarak verir. Model verilen framelere göre en çok bulunan kıyafet, yaş ve cinsiyeti dönmektedir. Ardından modelin döndüğü bu verileri filtre haline getirip schedule paketi sayesinde 10 saniyede bir Node.js servere socket bağlantısı ile göndermektedir. Node.js server bu filtreyi alarak React.js uygulamasına iletir. React.js uygulaması bu filtreyi alarak, hazırlamış olduğumuz .NET Core API'ye göndermektedir. API bu filtreye göre, kullanıcının eklemiş olduğu ürünler içerisinde eşleşen ürünlerin resimlerini React.js uygulamasına geri dönmektedir (Şekil 6). Ardından React.js uygulaması bu resimleri ekranda slider şeklinde göstermektedir.

İÇERİK EKLE : PRODUCT

Türkçe İngilizce İspanyolca

Name:

Category:

Gender:

Images:

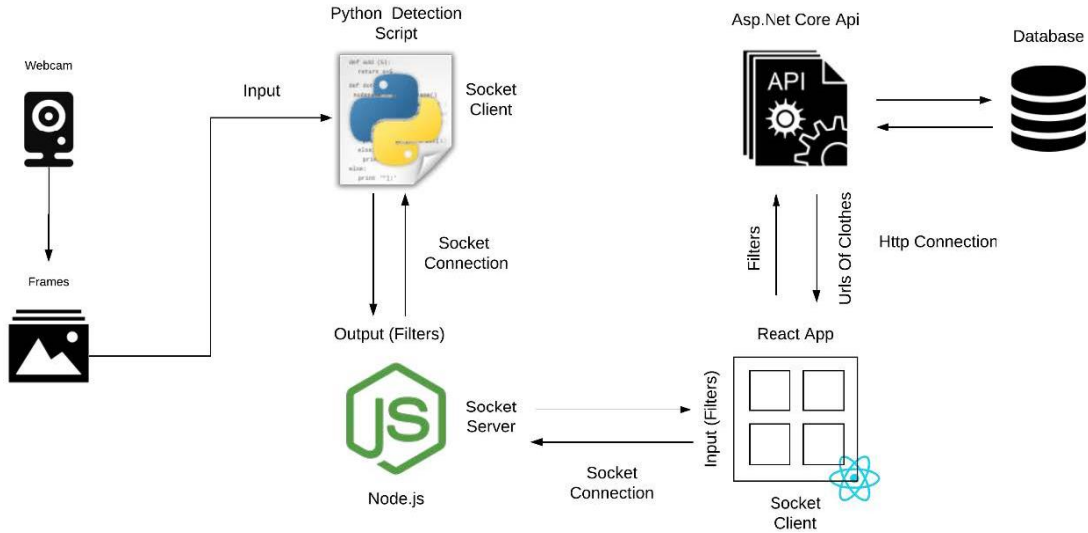
Ürün Fiyat:

Görüntülenme Sayısı:

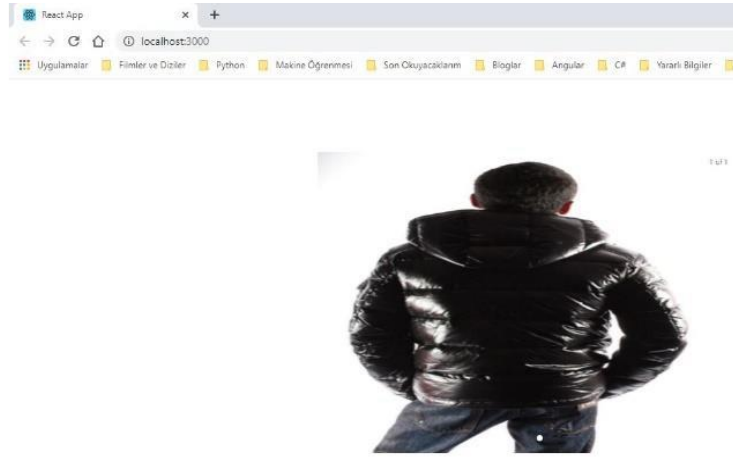
Minimum Yaş:

Maximum Yaş:

Şekil 4. Veritabanı şeması.



Şekil 5. Sistem çalışma diyagramı.



Şekil 6. API'den filtreye uygun dönen resmin gösterilmesi.

## 5. Bulgular ve Tartışma

Bu çalışmada yaş ve cinsiyet belirlenmesinde hali hazırda yüksek doğruluğa sahip ve önceden eğitilmiş olan Caffe mimarisini ile hazırlanmış [41] nolu çalışmadaki model kullanılmıştır. Yapılan çalışmada cinsiyet için AdienceFaces verisetini kullanmıştır [43] ve 19.487 adet görüntü bulunmaktadır. Cinsiyet tahmin modelinin doğruluğu %86,8 olarak  $\pm 1,4$  standart hatada Caffe mimarisi kullanılarak belirlenmiştir [41]. Yaş tahmininde de Adience veriseti kullanılmıştır [43] ve 26.000 görüntü bulunmaktadır. Yaş tahmininde aralık tahmini gerçekleştirilmiş olup yaş aralıkları 8 aralıkta belirlenmiştir [41]. Yaş tahmin modelinin doğruluğu %84,7 olarak  $\pm 2,2$  standart hatada Caffe mimarisi kullanılarak belirlenmiştir [41]. Çalışmanın ana konusu kişiye özgü reklam gösterimi olduğundan dolayı kıyafet türünün belirlenmesi daha önemli olmuştur ve bu çalışmadaki temel konu olarak ele alınmıştır. Bu çalışmadaki özgünlük kişilerin giydikleri kıyafetlerin türlerinin belirlenmesidir. Böylece giyim tarzı, yaş ve cinsiyet kullanılarak kişiye özgü reklam gösterimi yapılacaktır. Yaş ve cinsiyet reklamları belirli aralıkta kısıtlarken, giyim tarzına göre gösterilen reklam değişebilmektedir. Örneğin 30 civarı yaşlarda bir erkeğin takım elbisesinin belirlenmesi

ile kendisine profesyonel iş yaşamına uygun reklamlar gelebilecekken, aynı kişinin izin gününde şort ve tişört ile görülmesiyle bilgisayar oyunları, sportif giyim reklamları gibi reklamlar gösterilebilir. Bu çalışmada ilk olarak SSD Inception v2 [44], SSD Mobilenet v2 [44] modelleri çalışılmıştır. Daha sonra burada elde edilen sonuçlar sonrasında YOLO v3-416 modeli [45] kullanılmış olup, sonuçları gösterilmiştir. Bu üç model de aynı verisetiyle eğitilmiş olup, sonuçlar buna göre karşılaştırılmıştır.

Çalışmada kullanılan ilk model SSD MobileNet v2 olmuştur. Bu modelin eğitimi 1,5 gün sürmüştür. Bu modelin L1 ve L2 loss değerleri olan sınıflandırma ve yerel loss (classification ve localization loss) değerleri Tablo 2’de gösterilmiştir. 30.000 iterasyonda her iki Loss değeri de düşüş eğiliminde olmuştur. L2 değeri 10.000 iterasyon sonrasında fazla değişmemekle birlikte, L1 değeri 26.000 iterasyondan sonra fazla değişmemiştir. Bu ağıın GeForce GTX 765m GPU, 8 GB RAM ve Intel i7-4700MQ donanıma sahip bir bilgisayar ortamındaki bir çerçeveyi işleme hızı 39 ms olmuştur. Bu ortamda gerçek zaman performansı vermektedir. Hız konusunda istenilen sonuç alınsa da doğruluk kısmında yeterli gelmemiştir. Çalışmadaki ikinci model olan SSD Inception v2 eğitimi ise 2 gün sürmüştür. Bu modelde L1 ve L2 değerleri sırasıyla ilk 5.000 ve 6.000 iterasyonda düşmüş, sonrasında değişmemiştir. Bu sebeple eğitim 23.000 iterasyonda kesilmiştir. Bu ağıın GeForce GTX 765m GPU, 8 GB RAM ve Intel i7-4700MQ donanıma sahip bir bilgisayar ortamındaki bir çerçeveyi işleme hızı 45 ms olmuştur. Bu ortamda gerçek zaman performansı vermektedir. Raspberry Pi3 Model B+ ortamında bir çerçeveyi işleme hızı 1600–1900 ms arasında değişmektedir. Bu ortamda ise gerçek zaman performansı alınamamaktadır. SSD Inception v2 modelinde de istenen doğruluk oranına ulaşılamamıştır. Test sonuçlarında ağıın renkli veriseti üzerinde ezber yaptığı görülmüş ve hatalı çıktılar verdiği elde edilmiştir (her gösterilen kıyafete “kaban” çıktısı vermesi). Faster R-CNN modelinde ise eğitim 2,5 gün sürmüştür. Bu modelde L1 ve L2 değerleri ilk 3.000 iterasyonda düşmüş, sonrasında değişmemiştir. Bu sebeple eğitim 12.000 iterasyonda kesilmiştir. Bu ağıın GeForce GTX 765m GPU, 8 GB RAM ve Intel i7-4700MQ donanıma sahip bir bilgisayar ortamındaki bir çerçeveyi işleme hızı 80-100 ms olmuştur. Bu ortamda gerçek zaman performansı verememektedir. Yine model Raspberry Pi3 Model B+ ortamında bir çerçeveyi işleme hızı 2500 – 3000 ms arasında değişmektedir. Her üç model incelendiğinde iterasyonların yarıda kesilmesine rağmen eğitim süresinin gitgide arttığı görülmüştür. Bu 3 modelin çalışmada kullanılmasının uygun olmadığı görüldüğünden yeni bir derin öğrenme mimarisine yönelinmiştir.

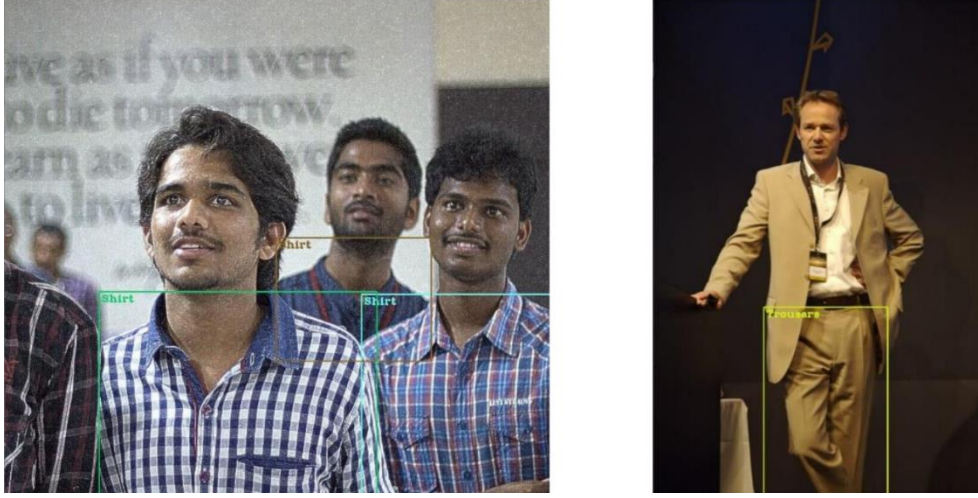
Tablo 2. Kıyafetlerin Türlerinin Belirlenmesinde Modellerin Hata Değerleri.

Model	L1	L2
SSD MobileNet v2	5,154	0,846
SSD Inception v2	5,286	0,856
Faster R-CNN	0,312	0,227

Bu aşamada SSD mimari ile kıyaslandığında YOLO v3 modelinin kullanılmasının uygun olduğu görülmüştür. Burada ilk olarak 3 sınıf üzerinden eğitim gerçekleştirilmiş, sonuçların yeterli görülmesinden sonra 17 sınıflı model oluşturularak eğitim gerçekleştirilmiştir.

YOLO v3 modelinde ilk olarak Shirt, Pantolon ve Bot olarak üç sınıf belirlenmiştir ev 1 günlük eğitim ile model sonuçları alınmıştır. “YOLO v3 - M1” olarak adlandırılan bu

modelin hata değeri (loss) 0,425 ve mAP değeri %69,36 olmuştur (Şekil 7). Bu modelde



Şekil 7. YOLO v3 - M1 sonuçları.

6.000 iterasyon gerçekleştirilmiş olup %33 mAP değerine 900 iterasyonda, %53 mAP değerine 1600 iterasyonda, %59 mAP değerine 2400 iterasyonda ulaşılmıştır. Loss değeri ise 5 değerinden başlamış ve 600 iterasyonda 1,3 değerine, 1200 iterasyonda 0,8 değerine, 1800 iterasyonda 0,6 değerine, 3600 iterasyonda ise 0,5 değerine ulaşmıştır. Genel olarak mAP ve Loss değerleri 2000 iterasyona kadar modelin performansının hızlıca geliştiğini, sonrasında ise iyileştirmeler yaptığını göstermiştir. YOLO v3 - M1 modeli YOLO v3'ün uygunluğunu sınamak için gerçekleştirdiği için eğitim 6.000 iterasyonda sınırlandırılmıştır. Bir sonraki model olan YOLO v3 – M2'de iterasyon sayısı arttırılmıştır.

Bir sonraki eğitimde 3.2.1 bölümünde belirtilmiş olan 17 sınıf (kıyafet türü) için hazırlanan “YOLO v3 – M2” olarak adlandırılan modelin eğitimi 3 gün sürmüştür (Tablo 3). Bu eğitimde mAP@0.5 değeri 68.000 iterasyonda %63,76 olmuş, ortalama Loss değeri ise 0,2705 olmuştur. Toplam true-positive (TP) sayısı 15.822, false-positive (FP) sayısı 6.444, false-negative (FN) 8.015 olarak belirlenmiştir. Intersection Over Union (IoU) değeri %56,3 olmuştur. YOLO v3 – M2 eğitimi hem Loss hem de mAP üzerinde üç bölümde kendini göstermiştir. mAP değeri başlangıçtan 4.080 iterasyon arasında %57 değerine çıkarken, 4.080 – 51.000 iterasyon arasında %4 artışla %61 olurken, 51.000 – 68.000 iterasyon arasında yaklaşık %4 artarak %63,76 olmuştur.

Çalışmada 3 ve 17 kıyafet türü üzerinden elde edilen sonuçlar YOLO v3 modellerinin kıyafet türü tespitinde sırasıyla kabul edilebilir hata değerleri olan %69,36 ve %63,76 ortalama mAP değeri elde etmesi, geliştirilmiş modeller kullanılabileceğini göstermiştir. Bu aşamadan sonra üç farklı YOLO v3 modeli oluşturulmuştur. Bu yaklaşımda üç parça olarak YOLO v3 modeli eğitilmiştir. Burada 17 kıyafet türü üzerinden elde edilen doğruluğu sınıfları 3 modele ayırarak arttırılması planlanmıştır. Modelleri “üst giyim”, “alt giyim” ve “ayakkabı” olarak ayrılması önerilmiştir. Gözlük, güneş gözlüğü ve şapka bu yaklaşımda eklenmemiştir. Bu modeller “üst giyim”, “alt giyim” ve “ayakkabı” için sırasıyla YOLO v3 - M3, YOLO v3 – M4 ve YOLO v3 - M5 olarak adlandırılmıştır ve bu modellerin değerleri Tablo 4’de gösterilmiştir.

Tablo 3. YOLO v3 – M2 modelindeki 17 sınıfın hata değerleri.

Kıyafet Türü	mAP (%)	TP	FP
Bot	60,88	985	391
Günlük Ayakkabı	46,76	2.039	1.095
Topuklu Ayakkabı	42,92	822	648
Terlik	48,06	743	463
Kot Pantolon	48,54	807	384
Kumaş Pantolon	56,00	715	327
Kısa Pantolon	62,40	1.084	386
Etek	80,37	903	161
Elbise	69,5	864	343
Tişört	92,90	940	92
Gömlek	74,31	759	154
Kazak	92,00	926	79
Ceket	51,43	847	562
Kaban	58,01	811	446
Gözlük	61,56	794	317
Güneş Gözlüğü	62,51	854	353
Şapka	75,79	989	243

M3, M4 ve M5 modellerinde ortalama Loss değeri 5 ile başlamıştır ve modellerin eğitimi 3 gün sürmüştür. YOLO v3 - M3 modelinde ortalama Loss 0,4943 değerine 16.000 iterasyonda ulaşmıştır. Bu model eğitiminde mAP@0.5 değeri 16.000 iterasyonda %45,13, ortalama IoU %44,95, precision 0,6, recall 0,56, F1-score 0,58 olmuştur. YOLO v3 – M4 modelinde de ortalama Loss 0,3204 değerine 16.000 iterasyonda ulaşmıştır. Bu model eğitiminde mAP@0.5 değeri 16.000 iterasyonda %58,02, ortalama IoU %53,26, precision 0,69, recall 0,59, F1-score 0,64 olmuştur.

Tablo 4. YOLO v3 – M3, M4 ve M5 modelindeki 3 sınıfın hata değerleri.

Kıyafet Sınıfı	Kıyafet Türü	mAP (%)	TP	FP
Ayakkabı YOLO v3 - M3	Bot	61,85	951	407
	Günlük Ayakkabı	52,65	2.395	1.381
	Topuklu Ayakkabı	20,57	427	689
	Terlik	45,45	729	477
Alt Giyim YOLO v3 – M4	Kot Pantolon	33,68	553	501
	Kumaş Pantolon	55,02	678	313
	Kısa Pantolon	64,16	1.070	422
	Etek	79,21	920	216
Üst Giyim YOLO v3 – M5	Elbise	47,30	579	407
	Tişört	93,95	938	69
	Gömlek	72,54	786	156
	Kazak	41,94	318	236
	Ceket	29,18	527	587
	Kaban	40,64	537	370

YOLO v3 – M5 modelinde de ortalama Loss 0,2817 değerine 24.000 iterasyonda ulaşmıştır. Diğerlerine göre burada 8.000 iterasyon fazla işlem yapılmasının sebebi 6 farklı kıyafet türünün olmasıdır. Her kıyafet türü için 4.000 iterasyon olduğu düşünülerek diğer modeller 16.000 iterasyonda sonlandırılmıştır. Bu model eğitiminin sonunda mAP@0.5 değeri %54,26, ortalama IoU %53,71, precision 0,67, recall 0,54, F1-score 0,60 olmuştur.

17 sınıf (kıyafet türü) için hazırlanan YOLO v3 – M2 ile üç farklı kıyafet sınıfı için hazırlanan YOLO v3 – M2, YOLO v3 – M3 ve YOLO v3 – M4 karşılaştırıldığında M2 modelinin daha uygun olduğu görülmüştür. Gruplandırılmış verilerde farklı türlerde de kıyafetlerin bulunmasının modellerin performansını düşürebileceğini göstermiştir. Örneğin 17 sınıflı modelde gözlük, gömlek, kot pantolon ve bot giymiş bir kişinin giydiklerinin tespiti bölgesel olarak sıralama yapılmış biçimde (yukarıdan aşağıya) olduğundan tahmin performansını arttırmıştır. Diğer tarafta M3 modelinde giyilmiş olan şapka, kısa pantolon veya kazak gibi kıyafetlerin de ayakkabı sınıfındaki bir tür olarak belirlenebilmesine olanak tanımıştır. Bu durum hataların daha fazla olmasına sebep olmuştur.

Çalışma literatürle karşılaştırıldığında sonuçların geliştirilmiş olduğu görülmüştür (Tablo 5). Bedeli ve arkadaşlarının yaptıkları çalışmada giysileri dört kategoriye ayırdıkları görülmüştür. Bu çalışmada eğitim Loss değerinin 0,07, doğruluk değerinin %75,3 olduğunu belirtmişlerdir [19]. Düşük olması istenen Loss değeri çalışmamızda eğitim aşamasında en düşük 0,227 olmuş, ortalama doğruluk değeri ise %64 olmuştur. Bedeli ve arkadaşlarının yaptıkları dört kategori seçildiğinde ise %72,05 olmuştur. Bu çalışmada bulunan 17 kategori bulunan model sonucunun 4 kategori bulunan Bedeli ve arkadaşlarının çalışması ile karşılaştırıldığında uygun olduğu görülmektedir. Saxena ve arkadaşları ise tam kollu tişört, kolsuz tişört ve kot pantolon olmak üzere üç tür kıyafet tespitinde test verisetinde tekli görünümde doğrulukları sırasıyla %40, %74, %100 olarak belirlemişlerdir [20]. Bizim çalışmamızda ise kazak (uzun kollu tişört yerine), tişört ve kot pantolon doğrulukları sırasıyla %92, %92,90, %48,54 olmuştur. Kot pantolon dışında sonuçlarımızın daha iyi olduğu görülmüştür. Dong ve arkadaşları da yaptıkları çalışmanın bir kısmında giyim kategorilerini de belirlemişlerdir [21]. Kategoriler hakkında detaylı bilgi olmayıp yün mont ile pamuk mont gibi detaylı kategorilerin olduğunu belirtmişlerdir. Yaptıkları çalışmada eğitim verisetiyle %65,96 ortalama doğruluk oranıyla belirledikleri kategoriler, bizim çalışmamızda test verisetinde %63,76 olmuştur. Dong ve arkadaşları, Liu ve arkadaşları tarafından DeepFashion veriseti kullanılarak önerdikleri FashionNet derin öğrenme mimarisi [24] ile kendi çalışmalarını karşılaştırmış ve Liu ve arkadaşlarının çalışmalarındaki kategori belirleme doğruluğunu %55,85 olarak ve 50 kategori için göstermişlerdir. Bizim çalışmamızın Liu ve arkadaşlarından daha yüksek doğrulukta olduğu da görülmüştür. Cychnerski ve arkadaşları da elbise tespitinde ortalama doğruluğu %42 bulmuşlardır [23] ve çalışmamızın sonuçlarının daha yüksek doğrulukta olduğu görülmüştür. Literatürdeki çalışmalar ile çalışmamız karşılaştırıldığında kabul edilebilir sonuçlar elde edildiği görülmektedir.

Tablo 5. Çalışmanın diğer çalışmalar ile kıyaslanması.

Yapılan Çalışmalar	Loss	Doğruluk	Eğitim Doğruluğu	Tam kollu tişört	Kolsuz tişört	Kot pantolon
[19] – 4 kategori	0,07	%75,3	-	-	-	-
[20]	-	-	-	%40	%74	%100
[21]	-		%65,96			
[23]	-	%42	-	-	-	-
[24] – 50 kategori	-	%55,85	-	-	-	-
Bu çalışma – 17 kategori	0,227	%64	%63,76	%92	%92,90	%48,54

## 6. Sonuçlar

Bu çalışmada YOLO v3 modelinde ilk olarak belirlenen Shirt, Pantolon ve Bot olarak üç sınıf model sonuçları hız ve doğruluk açısından tatmin eden bir sonuç elde etmiştir. Elde edilen sonuçlar, bir sonraki aşamada 17 sınıfı olan “YOLO v3 – M2” modelinin geliştirilmesini olanak tanımıştır. M2 modelinde de elde edilen sonuçların kabul edilebilir olması, daha yüksek doğruluğun elde edilebileceği düşünülen 3 parçalı 3 model ile tekrar eğitime başlanmasını sağlamıştır. Üç parçalı modeldeki modeller olan M3, M4 ve M5; M2 ile karşılaştırıldığında daha düşük doğruluğa sahip olduğu görülmüştür. Kişiye özgü reklam gösterimlerinde M2 modelinin kullanımının uygun olduğu görülmüştür. Modellerin eğitimi tamamlandıktan sonra GeForce GTX 765m GPU, 8 GB RAM ve Intel i7-4700MQ donanıma sahip bir bilgisayar modeller çalıştırılmıştır ve gerekli olan saniyedeki çerçeve sayısının (FPS - frames per second) performansı sistem tarafından karşılanmamıştır. Sistemin özelliklerinden dolayı çerçeveyi işleme hızı bizim için yeterli gelmemiştir. GeForce GTX 1080TI, RTX 2080 SUPER gibi yüksek seviye ekran kartları kullanılarak performans artırımı gerçekleştirilebilir. Eğer Raspberry Pi kullanılırsa modellerin Coral, Intel Neural Compute Stick gibi Universal Serial Bus (USB) ivmelendiricilerin Tensor Processing Unit’lerinden (TPU) yararlanılarak istenilen hızı alabileceği öngörülmektedir. Çalışmanın bundan sonraki kısmında diğer derin öğrenme algoritmaları, eğitim ve test verisetinin genişletilmesi gerçekleştirilecektir. Çalışma çerçeve hızını arttırmak için ise USB ivmelendiricilerin kullanılması planlanmaktadır.

## Kaynaklar

- [1] Deloitte, Reklamcılar Derneği, Türkiye’de Tahmini Medya ve Reklam Yatırımları: 2019 ilk 6 Ay Raporu, Technical Report, İstanbul, (2019).
- [2] Clouidian, Intel, Dentsu, Deep Learning Enables Intelligent Billboard for Dynamic, Targeted Advertising on Tokyo Expressway, Technical Report, USA, (2017).
- [3] Sternson, T., Wang, Q., Gulmammadova, W., Sun G. ve Kenzie J., AWS DeepAds Advertising, [https://aws.amazon.com/tr/deeplens/community-projects/DeepAds\\_Advertising/](https://aws.amazon.com/tr/deeplens/community-projects/DeepAds_Advertising/), (06-Feb-2021).
- [4] Doğan, F. ve Türkoğlu, İ., Derin Öğrenme Algoritmalarının Yaprak Sınıflandırma Başarımlarının Karşılaştırılması,” **Sakarya University Journal of**



- Computer and Information Sciences**, 1, 1, 10–21, (2018).
- [5] Fan, S., Li, J., Zhang, Y., Tian, X., Wang, Q., He, X., Zhang, C. and Huang, W., On line detection of defective apples using computer vision system combined with deep learning methods, **Journal of Food Engineering**, 286, 110102, (2020).
- [6] Kayaalp, K. ve Metlek, S., Classification of Robust and Rotten Apples by Deep Learning Algorithm, **Sakarya University Journal of Computer and Information Sciences**, 3, 2, 112–120, (2020).
- [7] Momeny M., Jahanbakhshi A., Jafarnezhad K. ve Zhang, Y.-D., Accurate classification of cherry fruit using deep CNN based on hybrid pooling approach, **Postharvest Biology and Technology**, 166, 111204, (2020).
- [8] Uğuz, S., Automatic Olive Peacock Spot Disease Recognition System Development by Using Single Shot Detector, **Sakarya University Journal of Computer and Information Sciences**, 3, 3, 158–168, (2020).
- [9] Adak, M. F., Identification of Plant Species by Deep Learning and Providing as A Mobile Application, **Sakarya University Journal of Computer and Information Sciences**, 3, 3, 231–238, (2020).
- [10] Uçar, M. ve Uçar, E., Computer-Aided Detection of Lung Nodules in Chest X-Rays using Deep Convolutional Neural Networks, **Sakarya University Journal of Computer and Information Sciences**, 2, 1, 1–8, (2018).
- [11] Anthimopoulos, M., Christodoulidis, S., Ebner, L., Christe, A., ve Mougiakakou, S., Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network, **IEEE Transactions on Medical Imaging**, 35, 5, 1207–1216, (2016).
- [12] Akyol, F. B. ve Altun, O., Detection of Cervix Cancer from Pap-smear Images, **Sakarya University Journal of Computer and Information Sciences**, 3, 2, 99–111, (2020).
- [13] Khamparia, A., Gupta, D., de Albuquerque, V. H. C., Sangaiah, A. K. ve Jhaveri, R. H., Internet of health things-driven deep learning system for detection and classification of cervical cells using transfer learning, **The Journal of Supercomputing**, 76, 11, 8590–8608, (2020).
- [14] Erdem, E. ve Aydın, T., Detection of Pneumonia with a Novel CNN-based Approach, **Sakarya University Journal of Computer and Information Sciences**, 4, 1, 26–34, (2021).
- [15] Çelik, G. ve Talu, M. F., Çekişmeli üretken ağ modellerinin görüntü üretme performanslarının incelenmesi, **Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi**, 22, 1, 181–192, (2020).
- [16] Gündüz, G. ve Cedimoğlu, İ. H., Derin Öğrenme Algoritmalarını Kullanarak Görüntüden Cinsiyet Tahmini, **Sakarya University Journal of Computer and Information Sciences**, 2, 1, 9–17, (2019).
- [17] Bilgin, M. ve Şentürk, İ. F., Danışmanlı ve yarı danışmanlı öğrenme kullanarak doküman vektörleri tabanlı tweetlerin duygu analizi, **Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi**, 21, 2, 822–839, (2019).
- [18] Erdoğan, P., Deep Learning Performance on Medical Image, Data and Signals, **Sakarya University Journal of Computer and Information Sciences**, 2, 1, 28–40, (2019).
- [19] Bedeli, M., Geradts, Z. ve van Eijk, E., Clothing identification via deep learning: forensic applications, **Forensic Sciences Research**, 3, 3, 219–229, (2018).
- [20] Saxena, K. ve Shibata, T., Garment Recognition and Grasping Point Detection for Clothing Assistance Task using Deep Learning, **IEEE/SICE International**

- Symposium on System Integration (SII)**, 632–637, Paris, (2019).
- [21] Dong, Q., Gong, S. ve Zhu, X., Multi-task Curriculum Transfer Deep Learning of Clothing Attributes, **2017 IEEE Winter Conference on Applications of Computer Vision (WACV)**, 520–529, California, (2017).
- [22] Lin, K., Yang, H.-F., Liu, K.-H., Hsiao, J.-H. ve Chen C.-S., Rapid Clothing Retrieval via Deep Learning of Binary Codes and Hierarchical Search, **Proceedings of the 5th ACM on International Conference on Multimedia Retrieval**, 499–502, Shanghai, (2015).
- [23] Cychnerski, J., Brzeski, A., Boguszewski, A., Marmolowski, M. ve Trojanowicz, M., Clothes detection and classification using convolutional neural networks, **2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)**, 1–8, Limassol, Cyprus, (2017).
- [24] Liu, Z., Luo, P., Qiu, S., Wang, X. ve Tang, X., DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations, **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 1096–1104, Nevada, (2016).
- [25] Han X., Wu Z., Huang, P.X., Zhang, X., Zhu, M., Li, Y., Zhao, Y. ve Davis L.S., Automatic Spatially-Aware Fashion Concept Discovery, **IEEE International Conference on Computer Vision (ICCV)**, 1472–1480, Venice, (2017).
- [26] Gong, K., Liang, X., Zhang, D., Shen, X. ve Lin, L., Look into Person: Self-Supervised Structure-Sensitive Learning and a New Benchmark for Human Parsing, **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 6757–6765, Hawaii, (2017).
- [27] Krizhevsky A., Sutskever I. ve Hinton G. E., ImageNet Classification with Deep Convolutional Neural Networks, **Advances in Neural Information Processing Systems 25 (NIPS 2012)**, 1097–1105, Nevada, (2012).
- [28] Bouwmans, T., Javed, S., Sultana, M. ve Jung, S.K., Deep neural network concepts for background subtraction: A systematic review and comparative evaluation, **Neural Networks**, 117, 8–66, (2019).
- [29] Girshick, R., Donahue, J., Darrell, T. ve Malik, J., Rich feature hierarchies for accurate object detection and semantic segmentation, **2014 IEEE Conference on Computer Vision and Pattern Recognition**, 580-587, Ohio, 2014.
- [30] Ipek, B. ve Akpinar, M. Application of Deep Learning Based Object Detection on Unmanned Aerial Vehicle, **2020 IEEE 5th International Conference on Computer Science and Engineering (UBMK)**, 1-5, Diyarbakır, (2020).
- [31] Girshick, R., Fast R-CNN, **2015 IEEE International Conference on Computer Vision (ICCV)**, 1440-1448, Santiago, (2015).
- [32] Ren, S., He, K., Girshick, R. ve Sun, J., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 39, 6, 1137-1149, (2017).
- [33] Redmon, J., Divvala, S., Girshick, R. ve Farhadi A., You Only Look Once: Unified, Real-Time Object Detection, **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 779-788, Nevada, (2016).
- [34] Chen, W., Huang, H., Peng, S., Zhou, C. ve Zhang, C., YOLO-face: a real-time face detector, **The Visual Computer**, (2020).
- [35] Du, J., **Understanding of Object Detection Based on CNN Family and YOLO**, **Journal of Physics: 2nd International Conference on Machine Vision and Information Technology (CMVIT 2018)**, 012029, Hong Kong, (2018).
- [36] Tao, J., Wang, H., Zhang, X., Li, X. ve Yang, H., An object detection system

- based on YOLO in traffic scene, **6th International Conference on Computer Science and Network Technology (ICCSNT)**, 315–319, Dalian, (2017).
- [37] Buric, M., Pobar, M. ve Ivasic-Kos, M., Ball Detection Using Yolo and Mask R-CNN,” **2018 International Conference on Computational Science and Computational Intelligence (CSCI)**, 319–323, Nevada, (2018).
- [38] Redmon, J. ve Farhadi, A., YOLO9000: Better, Faster, Stronger, **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 6517–6525, Hawaii, (2017).
- [39] Laval, M., Tomato detection based on modified YOLOv3 framework, **Scientific Reports**, 11, 1, 1447, 1-11, (2021).
- [40] Viola, P. ve Jones, M. J., Robust Real-Time Face Detection, **International Journal of Computer Vision**, 57, 2, 137–154, (2004).
- [41] Levi, G. ve Hassner, T., Age and gender classification using convolutional neural networks,” **2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)**, 34–42, Massachusetts, (2015).
- [42] Hassner, T., Harel, S., Paz, E. ve Enbar, R., Effective face frontalization in unconstrained images, **2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 4295–4304, Massachusetts, (2015).
- [43] Eidingen, E., Enbar, R., ve Hassner, T., Age and Gender Estimation of Unfiltered Faces, **IEEE Transactions on Information Forensics and Security**, 9,12, 2170–2179, (2014).
- [44] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S. ve Murphy, K., “Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors,” in **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 3296–3297, Hawaii, (2017).
- [45] Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E. ve Liang, Z., Apple detection during different growth stages in orchards using the improved YOLO-V3 model, **Computers and Electronics in Agriculture**, 157, 417–426, (2019).