 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

**T.C.
GEBZE TEKNİK ÜNİVERSİTESİ**


Bilgisayar Mühendisliği Bölümü

**Arka Plan ve Manken
Görüntüsü Olan Görsellerden
Elbise Resminin İzole Edilmesi**

Değer MANDAL

**Danışman
Doç. Dr. Mehmet GÖKTÜRK**

**Haziran, 2021
Gebze, KOCAELİ**

 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

**T.C.
GEBZE TEKNİK ÜNİVERSİTESİ**

Bilgisayar Mühendisliği Bölümü

**Arka Plan ve Manken
Görüntüsü Olan Görsellerden
Elbise Resminin İzole Edilmesi**

Değer MANDAL

**Danışman
Doç. Dr. Mehmet GÖKTÜRK**

**Haziran, 2021
Gebze, KOCAELİ**

	<p>.....</p> <p>KILAVUZU</p>	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-


Bu çalışma/..../200.. tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Bölümü'nde Lisans Bitirme Projesi olarak kabul edilmiştir.

Bitirme Projesi Jürisi

Danışman Adı	Doç. Dr. Mehmet GÖKTÜRK	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

Jüri Adı	Prof. Dr. İbrahim SOĞUKPINAR	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

Jüri Adı	Doç. Dr. Erchan Aptoula	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

ÖNSÖZ

Bu kılavuzun ilk taslaklarının hazırlanmasında emeği geçenlere, kılavuzun son halini almasında yol gösterici olan Sayın Doç. Dr. Mehmet GÖKTÜRK hocama ve bu çalışmayı destekleyen Gebze Teknik Üniversitesi'ne içten teşekkürlerimi sunarım.

Ayrıca eğitimim süresince bana her konuda tam destek veren aileme ve bana hayatlarıyla örnek olan tüm hocalarıma saygı ve sevgilerimi sunarım.

Haziran, 2021

Değer MANDAL

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-


İÇİNDEKİLER

ÖNSÖZ	- 4 -
İÇİNDEKİLER.....	5
ŞEKİL LİSTESİ	6
ÖZET	8
SUMMARY	8
1. GİRİŞ.....	9
1.1. PROJE TANIMI.....	9
1.2. PROJENİN NEDEN VE AMAÇLARI.....	10
2. ARA RAPOR İÇERİĞİ.....	11
2.1. PROJE GEREKSİNİMLERİ.....	11
2.2. MALZEME VE YÖNTEM	11
2.2.1. VERİ SETİNİN ELDE EDİLMESİ.....	11
2.2.2. Görsellerin Etiketlenmesi ve Maskeleme	13
2.2.3. U-NET Modeli.....	16
2.2.4. Model Eğitimi	18
2.2.5. Maskeden Renklendirilmiş Görüntünün Elde Edilmesi.....	23
2.3. BAŞARI KRİTERLERİ	23
3. PROJE GÖRSEL SONUÇLAR.....	24
4. TARTIŞMA VE SONUÇ	27
4.1. EĞİTİM SONUCU	27
4.2. PROJENİN SONUCU.....	27
KAYNAKLAR.....	28

	<p>.....</p> <p>KILAVUZU</p>	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-


ŞEKİL LİSTESİ

Şekil 1	Projenin Genel Olarak Betimlenmiş Hali	10
Şekil 2	Modanisa Veri Seti Csv Dosyası İçeriği.....	12
Şekil 3	Csv Dosyası İçin Yazılan Python Kod İçeriği.....	12
Şekil 4	LabelMe Örnek Etiketleme	13
Şekil 5	Deeplabs İçin Örnek Görseller - 1	14
Şekil 6	Deeplabs İçin Örnek Görsel Visualization - 2	14
Şekil 7	Deeplabs İçin Örnek Görsel Renkli Maskeleme – 3	15
Şekil 8	LabelMe _io.py Değişikliği	15
Şekil 9	Deeplabs İçin Örnek Görsel Siyah Beyaz Maskeleme – 4	16
Şekil 10	Fully Convolutional Yapı	17
Şekil 11	U-NET Modeli	17
Şekil 12	Backbone Listesi	18
Şekil 13	İlk Eğitim Sonuç Grafiği	18
Şekil 14	İlk Eğitim Sonuç Görselleri	19
Şekil 15	İkinci Eğitim Sonuç Grafikleri	19
Şekil 16	İkinci Eğitim Sonuç Görselleri - 1	20
Şekil 17	İkinci Eğitim Sonuç Görselleri - 2	20
Şekil 18	Elbise Çeşitliliği	21
Şekil 19	Son Eğitim Sonuç Grafikleri	22
Şekil 20	Son Eğitim Sonuç Görselleri	22
Şekil 21	Renklendirilmiş Görüntü Google Colab Kodu	23
Şekil 22	Test Sonuçları - 1	24
Şekil 23	Test Sonuçları - 2	24
Şekil 24	Test Sonuçları - 3	25
Şekil 25	Streamlit Giriş Ekranı	25
Şekil 26	Streamlit Hakkında Ekranı	26
Şekil 27	Streamlit Örnek Ekran	26
Şekil 28	Streamlit Örnek Görsel Sonuçları	27

 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

KISALTMA LİSTESİ

CNN: CONVOLUTIONAL NEURAL NETWORK (KONVOLÜSYONEL SİNİR AĞI)

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

ÖZET


Bu proje ile birlikte, katalog çekimi görsellerinden arka planı izole ederek, kendi alışveriş sitelerine koymak isteyen website sahipleri yararlanabilecektir. Bu sayede katalog çekim fotoğraflarını rahatlıkla sitelerinde kullanabileceklerdir. Ayrıca bu izole edilmiş görüntüleri kullanarak istedikleri şekilde değişiklik yapabileceklerdir. Projenin amacı, elbisenin detaylarına da dikkat ederek en iyi şekilde görselin arka planını izole etmek ve izole edilmiş elbisenin çıktısını kullanıcıya sunmaktır.

İlk eğitimde maskelenmiş 108, ikinci eğitimde 380 son eğitim denemesinde ise 813 tane maskelenmiş görüntüyle eğitim yapılmıştır. Elde edilen en yüksek accuracy sonucu yüzde 61'dir. InceptionV3, Vgg16, Resnet34, Resnet50 gibi backbonelar ile eğitimler çeşitlendirilmiştir. En iyi sonuç Resnet34 modelinde elde edilmiştir.

SUMMARY

With this project, website owners who want to isolate the background from the catalog shooting images and put them on their own shopping sites will be able to benefit. In this way, they will be able to easily use catalog shooting photos on their sites. They will also be able to make changes as they wish using these isolated images. The aim of the project is to isolate the background of the image in the best way by paying attention to the details of the dress and to present the output of the isolated dress to the user.

Training was conducted with 108 masked images in the first training, 380 in the second training, and 813 masked images in the last training trial. The highest accuracy result obtained is 61 percent. Trainings are diversified with backbones such as InceptionV3, Vgg16, Resnet34, Resnet50. The best results were obtained in the Resnet34 model.


	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

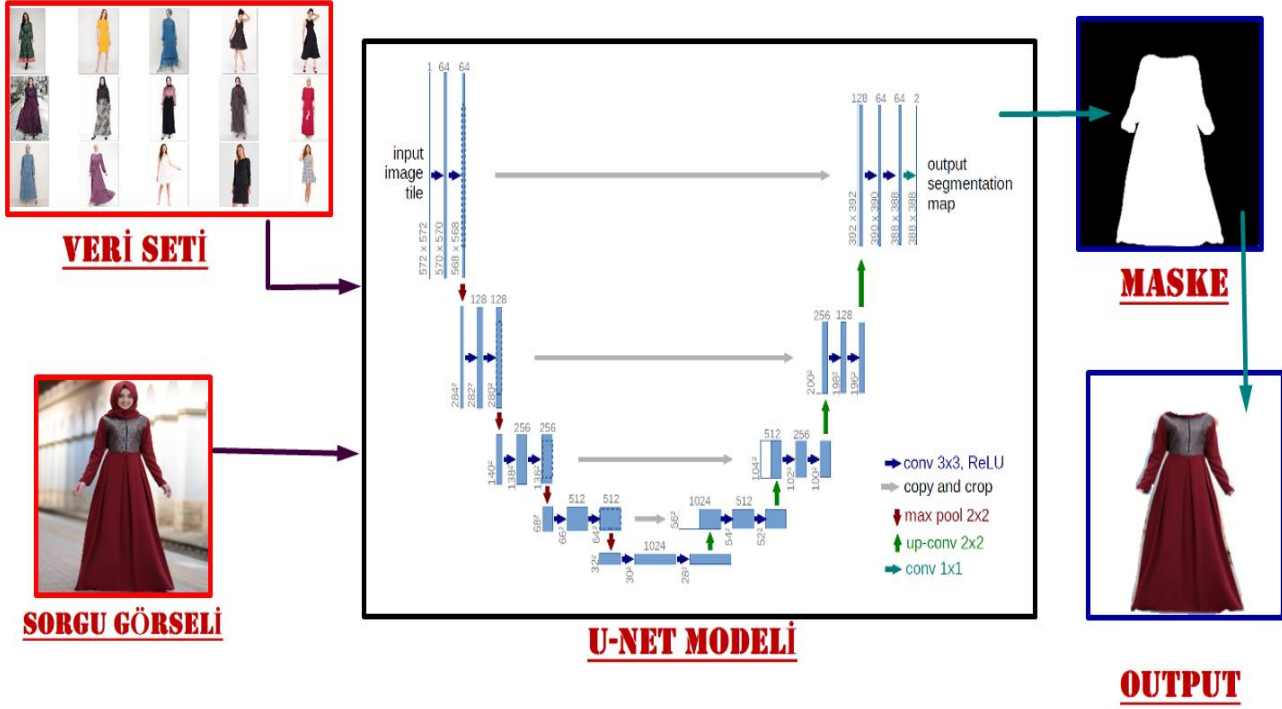
1. GİRİŞ

Katalog çekimleri bir kez yapıldığında bu resimlerin pazarlanması için bir çok ülke ve websitelerinde sergilenmesi gerekmektedir. Her website için resim üzerinde farklı değişiklikler yapılma ihtiyacı doğmaktadır. Ama elbise görüntüsü manken üzerinde olduğu için değişiklik yapabilme konusunda kısıtlamalar oluşmaktadır. Bu geliştirilen proje bu sorunu çözebilme aşamasında kullanılabilecek bir proje olmaktadır. Bu proje ile birlikte sadece elbise görüntüsü elde edilebilecektir. Bu da görsel üzerinde kolaylıkla değişiklik yapılabilmesine olanak sağlayacaktır. U-NET modeli kullanılarak girdi olarak verilen elbise görselinin maske görüntüsü elde edilmiştir. Daha sonra bu maske görüntüsü kullanarak istenilen çıktı elde edilmiştir.

1.1. PROJE TANIMI

Bu çalışmada derin öğrenme teknikleri kullanılarak bir elektronik ticaret giyim mağazası portföyündeki, manken görsellerinden sadece elbise görüntüsü elde edilmiştir. Özellikle farklı websiteler elbise görüntüsü üzerinde değişiklik yapabilmek istediğinde rahatlıkla yapabileceklerdir. Bu çalışmadaki U-NET modeli ile arka plan ve manken görüntüsü olan görsellerden elbise resminin izole edilmesi hedef alınmış ve yapılan çalışmada başarılı sonuçlara yakın sonuçlar elde edilmiştir. Projeyi betimleyen şekil Şekil 1’de betimlenmiştir.

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-



Şekil 1 Projenin Genel Olarak Betimlenmiş Hali

1.2. PROJENİN NEDEN ve AMAÇLARI

Farklı ülke ve e-ticaret sitelerinde katalog görsellerini değişiklik yapılabilir halde rahatlıkla kullanabilmek için bu proje geliştirilmiştir. Şu an proje sorgu olarak verilen elbise görselini almakta ve Modanisa'nın veri setiyle eğitilmiş olan U-NET modeli kullanılarak uygun maske oluşturmaktadır. Bu maskeyi de görüntü işleme teknikleriyle uygun arka planı izole edilmiş görüntü haline getirmektedir. İleriki aşamalarda herhangi bir e-ticaret sitesine entegre edilebilecek şekilde ilerlemek amaçlanmıştır.

Bu proje sayesinde:

- Katalog çekim görüntüleri rahatlıkla başka website ve ülke bazındaki sitelerde kullanılabilir.
- Elbise resimlerinin ayrıca çekimi yapılmak zorunda kalınmayacaktır. Katalog çekimindeki görseller kullanılabilir.
- Ülke bazında düşünüldüğünde, istenildiği gibi manken görüntüsü photoshop ile değiştirilebilir. Arka planda da ekleme yapılabilir.

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

2. ARA RAPOR İÇERİĞİ

"Labelme" toolu kullanılarak görseller etiketlenmiştir. Bu etiketli veriler ve "Deeplabs" modeli kullanılarak görsellerin maskeleri oluşturulmuştur. Maskeler ve "Modanisa" görselleri kullanılarak U-NET modeli eğitilmiştir. Eğitim sırasında hazır segmentasyon modeli kullanılmıştır. Birçok model denenmiştir decoder olarak, encoder kısmı için "Imagenet" kullanılmıştır. Decoder olarak ise en iyi sonuç "Resnet34" modeli ile elde edilmiştir. En son aşamada ise görüntü işleme teknikleri kullanılmıştır.

2.1. PROJE GEREKSİNİMLERİ

Proje Google Colab üzerinde hazırlandığı için herhangi bir donanım ihtiyacı duyulmamıştır.

Bu projede başarılması gerekenler:

- Modanisa veri setinden görsellerin indirilmesi ve uygun görsellerin veri seti olarak belirlenmesi
- Görsellerin etiketlenmesi
- Etiketli görsellerden sementik segmentasyon ile maske oluşturulması
- Maske ve görsellerin U-NET modelinde eğitilmesi
- Elde edilen maskenin görüntü işleme teknikleriyle istenilen renkli hale getirilmesi
- Sonuçların kullanıcıya daha iyi aktarılabilmesi için web arayüz tasarlanması
- Girdi olarak verilen katalog çekim görselinin, arka planı beyaz sadece elbise görüntüsü içeren bir çıktı görsel olarak sunulması

Bunların sağlanması için gerekli ihtiyaçlar:

- Modanisa veri seti
- Google Colab
- Python'ın Tensorflow, Keras gibi derin öğrenme kütüphaneleri
- Görüntü işleme için OpenCV ve Pillow kütüphaneleri

2.2. MALZEME VE YÖNTEM


2.2.1 Veri Setinin Elde Edilmesi

Modanisa

Modanisa database'ı bana csv dosyası olarak iletilmiştir. Bu csv dosyası 1,5 milyon tane veri içermekteydi. Bunların 10 bin tanesi elbise görseline aitti.

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

Csv Dosya İçeriği:

 product_images.csv

productid,category,url
1291981,Sandalet,https://fns.modanisa.com/r/pro2/2019/10/07/ayakland-100562-pudra-suet-7-cm-topuklu-bayan-sandalet-ayakkabi-pudra-ayakland-1291981-1.JPG
1493957,Pantolon,https://fns.modanisa.com/r/pro2/2020/02/27/kareli-pantolon-kirmizi-koton-1493957-1493957-1.jpg
54784,Kap / Pardes,https://fns.modanisa.com/r/pro2/2013/11/08/zimba-suslemeli-kapsonlu-kap--13107136--sax--scl-54784.jpg
632965,Kap / Pardes,https://fns.modanisa.com/r/pro2/2018/11/09/kurk-detayli-kap--camel--armine-632965-1.jpg
358078,Tesettür Elbise,https://fns.modanisa.com/r/pro2/2017/10/26/kadife-elbise--mor--ginezza-358078-358078-1.jpg
851008,Tesettür Elbise,https://fns.modanisa.com/r/pro2/2019/02/19/elbise-siyah-lc-waikiki-851008-1.jpg
679604,Tesettür Elbise,https://fns.modanisa.com/r/pro2/2018/11/23/elbise-siyah-lc-waikiki-679604-1.jpg
828624,Ärörp,https://fns.modanisa.com/r/pro2/2019/02/13/3lu-kisa-corap-seti-mavi-defacto-828624-1.jpg
1641013,Pantolon,https://fns.modanisa.com/r/pro2/2020/05/09/rahat-kesim-kapri-pantolon-turuncu-defacto-1641013-1.jpg

Şekil 2 Modanisa Veri Seti Csv Dosyası İçeriği

Elbise görsel urllerinin elde edilmesi için küçük bir python kodu yazılmıştır:

```

1 import io
2
3 f = io.open('product_images.csv', 'r', encoding = 'utf-8')
4 s = 'Elbise'
5 while True:
6     line = f.readline()
7     x = line.split(",")
8     if(x[1] == s):
9         print(x[2])
10    if len(line) == 0 and s not in line:
11        print('End of line. no more result')
12        break
13 f.close()
14
while True:
    if len(line) == 0 and s not in line:
        print('End of line. no more result')
        break
    else:
        print(line)

```

Run df

https://fns.modanisa.com/r/pro2/2014/05/26/leopar elbise grr medine tesettur 70941.jpg

https://fns.modanisa.com/r/pro2/2013/10/23/bebe-yaka-elbise-4148--murdum--ipekzade-53095.jpg

https://fns.modanisa.com/r/pro2/2016/04/01/jakarli-elbise--lacivert--moda-ozveri-196333-196333-1.jpg

https://fns.modanisa.com/r/pro2/2019/07/12/robada-tunik-ikili-takim--pudra--zehraca-1106385-28.jpg

https://fns.modanisa.com/r/pro2/2017/04/13/leoparli-elbise--taba--filizzade-289050-1.jpg

https://fns.modanisa.com/r/pro2/2017/12/15/karo-elbise--siyah--minel-ask-382606-1.jpg

https://fns.modanisa.com/r/pro2/2019/07/24/hakim-yaka-elbise--bej--mnatural-1162036-6.jpg

https://fns.modanisa.com/r/pro2/2018/02/06/cicek-desenli-elbise--lacivert--miss-cazibe-399701-1.jpg

https://fns.modanisa.com/r/pro2/2017/07/26/cicek-desenli-elbise--siyah--insirah-319819-319819-1.jpg

Şekil 3 Csv Dosyası İçin Yazılan Python Kod İçeriği

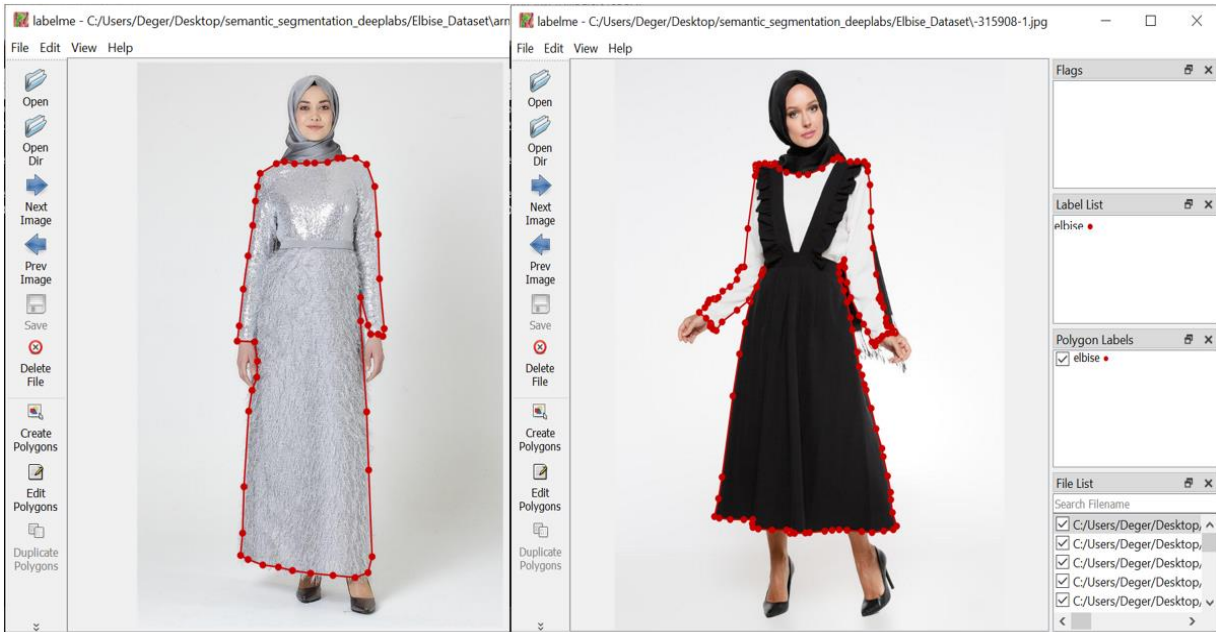
Görsellerin indirilmesi için Chrome Uzantısı olan, Tab Save Chrome uzantısı kullanılmıştır.

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

2.2.2 Görsellerin Etiketlenmesi ve Maskaeme

LabelMe

Görselleri etiketlemek için LabelMe tool'u kullanılmıştır. Tool etiklenecek görseli noktalar şeklinde üzerinden geçerek etiketleme yapmaktadır. Bunun sonucunda .json uzantılı dosya oluşturmaktadır. Etiketleme işlemi gerçekten uğraştırıcı ve zaman alan bir işlemdir. İlk sunum için 108, ikinci sunum için 380 ve en son sunum için 813 tane görsel etiketlenmiştir.




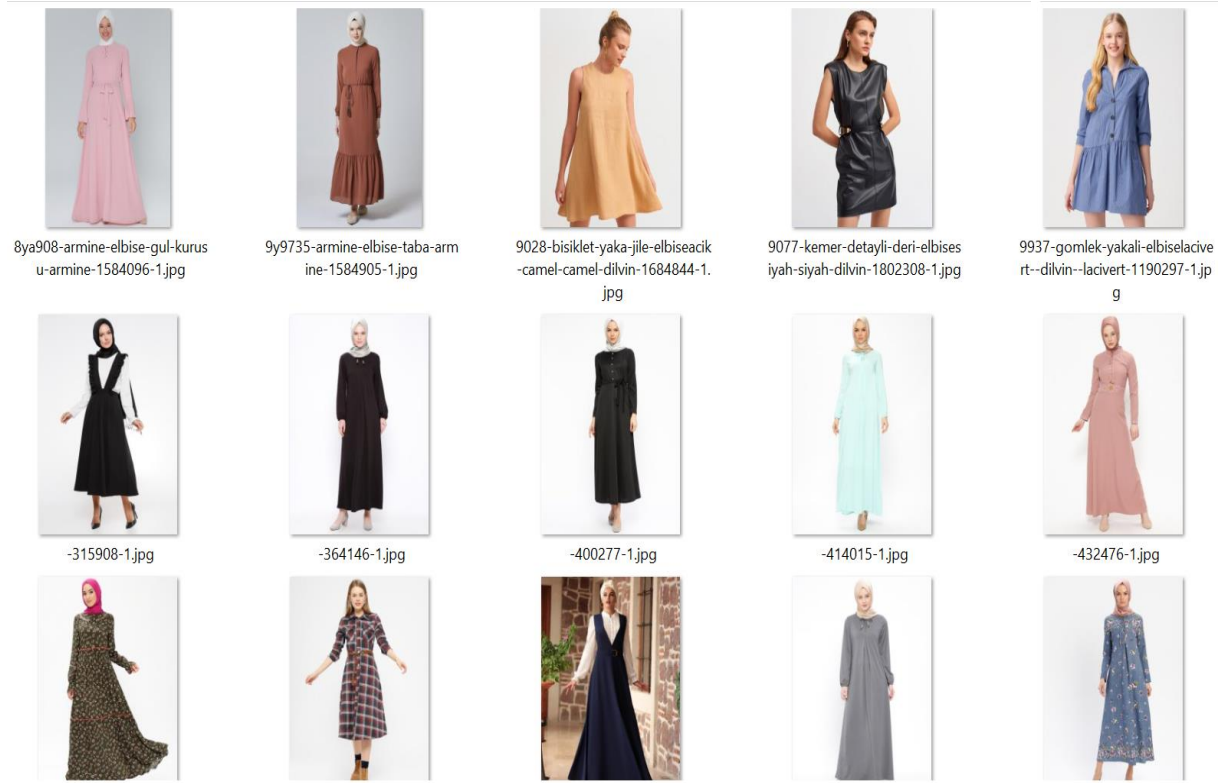
Şekil 4 LabelMe Örnek Etiketleme

Anlamsal Segmentasyon - Deeplabs Custom Training

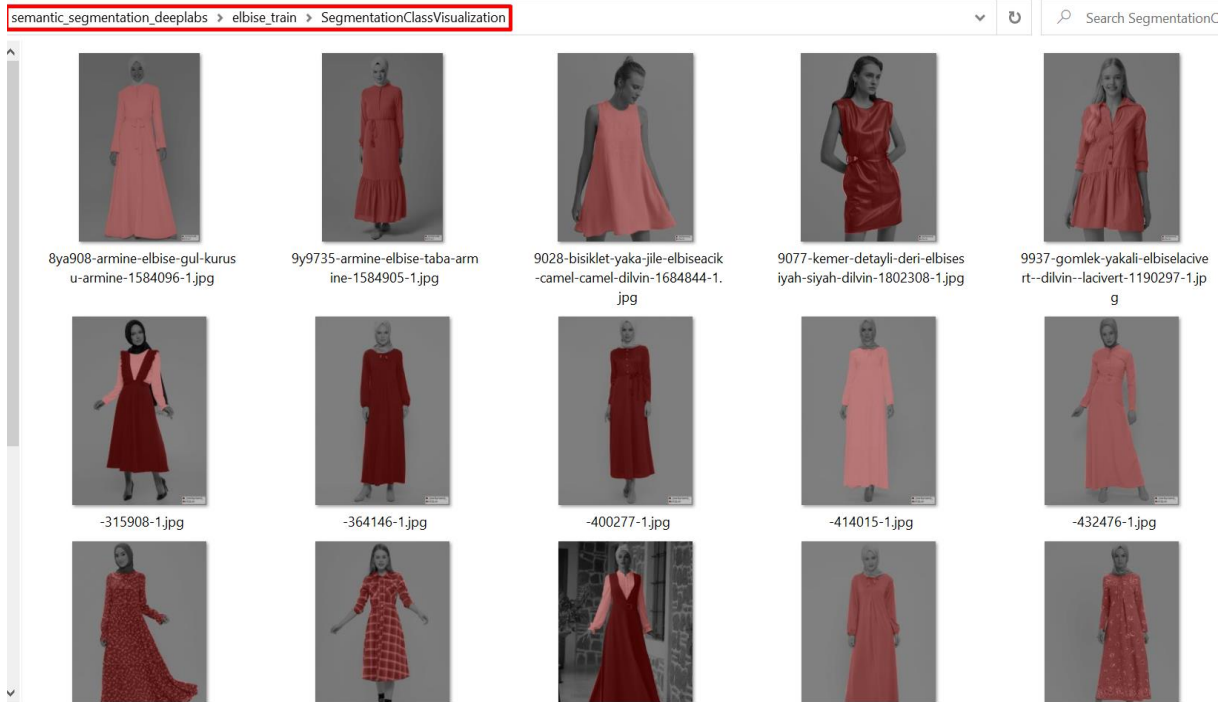
LabelMe sonucu elde edilen json dosyalarını kullanarak bu görsellerin maskelerini elde etmem gerekmekteydi. Bunun için semantik(anlamsal) segmentasyon yaptım ve bu işlem için Deeplabs modelini kullandım.

DeepLabs, Anlamsal Segmentasyon için güzel bir çözümdür. Kod, TensorFlow'da mevcuttur. Anlamsal bölümlleme, bir görüntüyü piksel düzeyinde anlamak ve ardından bir görüntüdeki her piksele bir etiket atamaktır, böylece aynı etikete sahip pikseller belirli özellikleri paylaşır. Örnek Görseller:

	<p>.....</p> <p>KILAVUZU</p>	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

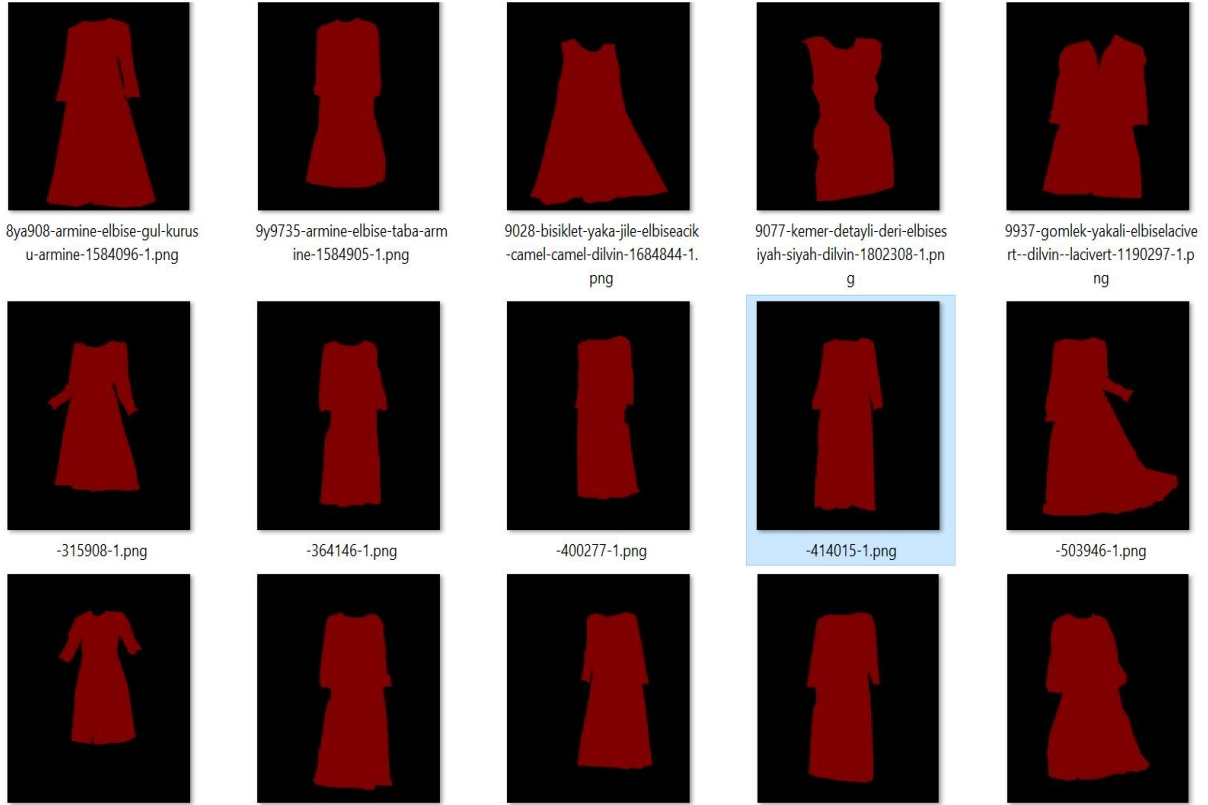


Şekil 5 Deeplabs İçin Örnek Görseller - 1



Şekil 6 Deeplabs İçin Örnek Görsel Visualization - 2

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-



Şekil 7 Deeplabs İçin Örnek Görsel Renkli Maskeleme – 3

Benim elde etmek istediğim maskeler siyah beyaz olmalıydı bu yüzden LabelMe toolunun _io.py dosyasında değişiklik yaptım.

```

_io.py x labelme2voc.py x labels.txt x untitled
import os.path as osp

import numpy as np
import PIL.Image

def lblsave(filename, lbl):
    import imgviz

    if osp.splitext(filename)[1] != ".png":
        filename += ".png"
    # Assume label ranges [-1, 254] for int32,
    # and [0, 255] for uint8 as VOC.
    if lbl.min() >= -1 and lbl.max() < 255:
        lbl_pil = PIL.Image.fromarray(lbl.astype(np.uint8), mode="P")
        colormap = imgviz.label_colormap()
        #lbl_pil.putpalette(colormap.flatten())

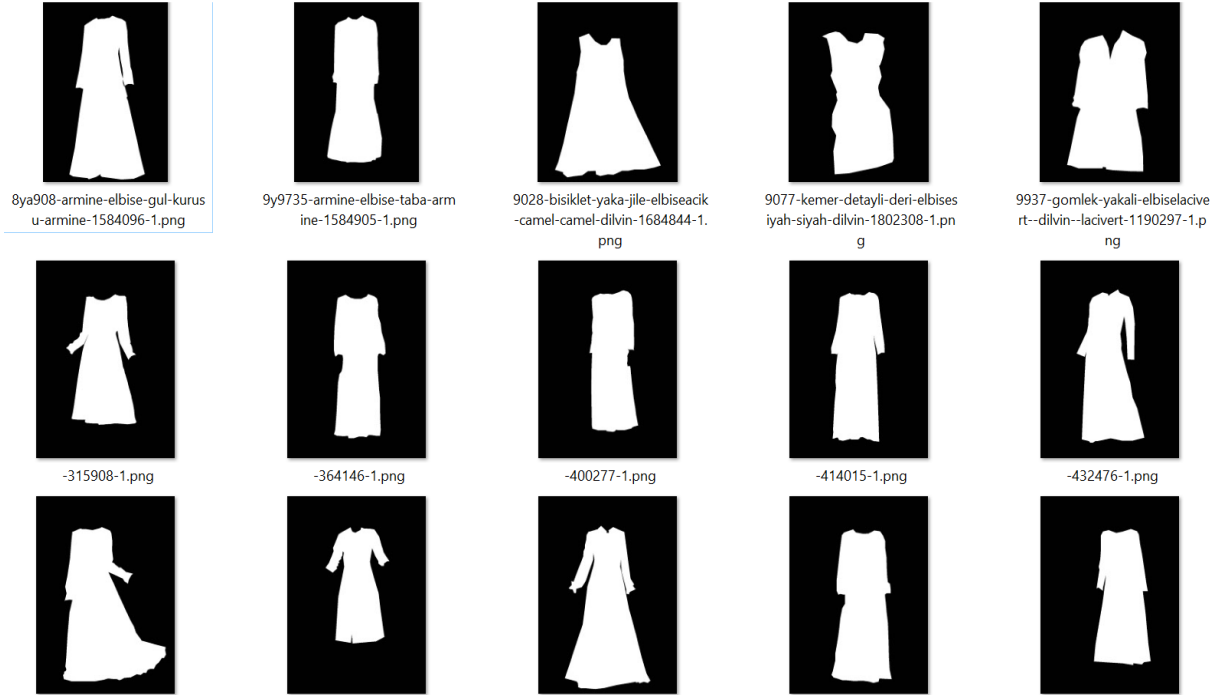
        colormap = np.ones((255, 3), dtype=float)
        colormap[0] = [255, 255, 255]
        lbl_pil.putpalette((colormap * 255).astype(np.uint8).flatten())

        lbl_pil.save(filename)
    else:
        raise ValueError(
            "[%s] Cannot save the pixel-wise class label as PNG. "
            "Please consider using the .npy format." % filename
        )

```

Şekil 8 LabelMe _io.py Değişikliği

	<p>.....</p> <p>KILAVUZU</p>	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-




Şekil 9 Deeplabs İçin Örnek Görsel Siyah Beyaz Maskeleye – 4

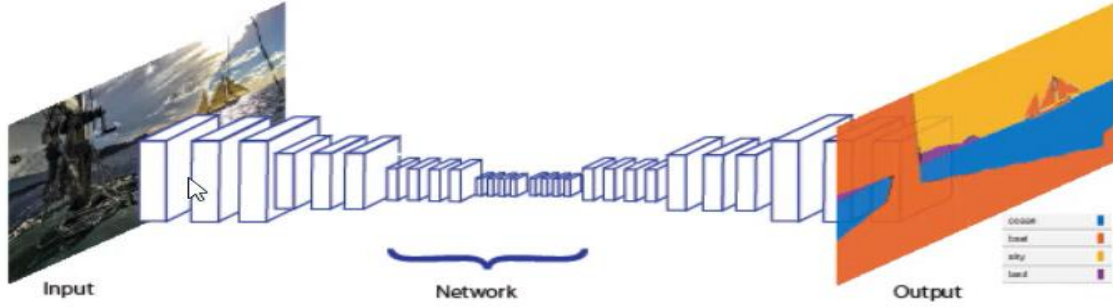
2.2.3 U-NET Modeli

U-NET

Geleneksel modellerde, katmanlar genellikle bir sonrakine bağlanır. Daha fazla maxpooling katmanı girişi geçerken, daha fazla özellik kaybolur. U-NET bu sorunu akıllıca çözmektedir. Encoder katmanından decoder katmanlarına doğrudan çıktılar ekler, böylece decoder katmanı daha fazla ayrıntı kullanabilir. U-NET, merdiven benzeri bir yapı oluşturmak için encoder katmanı özellik haritalarını her aşamada decoder katmanından örneklenmiş özellik haritalarına birleştirir. U-NET mimarisi, decoder katmanının her aşamada encoder katmanında toplandığında kaybolan ilgili özellikleri geri öğrenmesine olanak tanır.

U-NET encoder ve decoder olmak üzerine iki kısımdan oluşmaktadır. Bu yapı Fully Convolutional yapı oluşturmaktadır. Bunun anlamı encoder kısmında daralan daha sonra ise decoder kısmında genişleyen bir yapı oluşturmaktadır. Aşağıdaki görsel Fully Convolutional yapıyı temsil etmektedir.

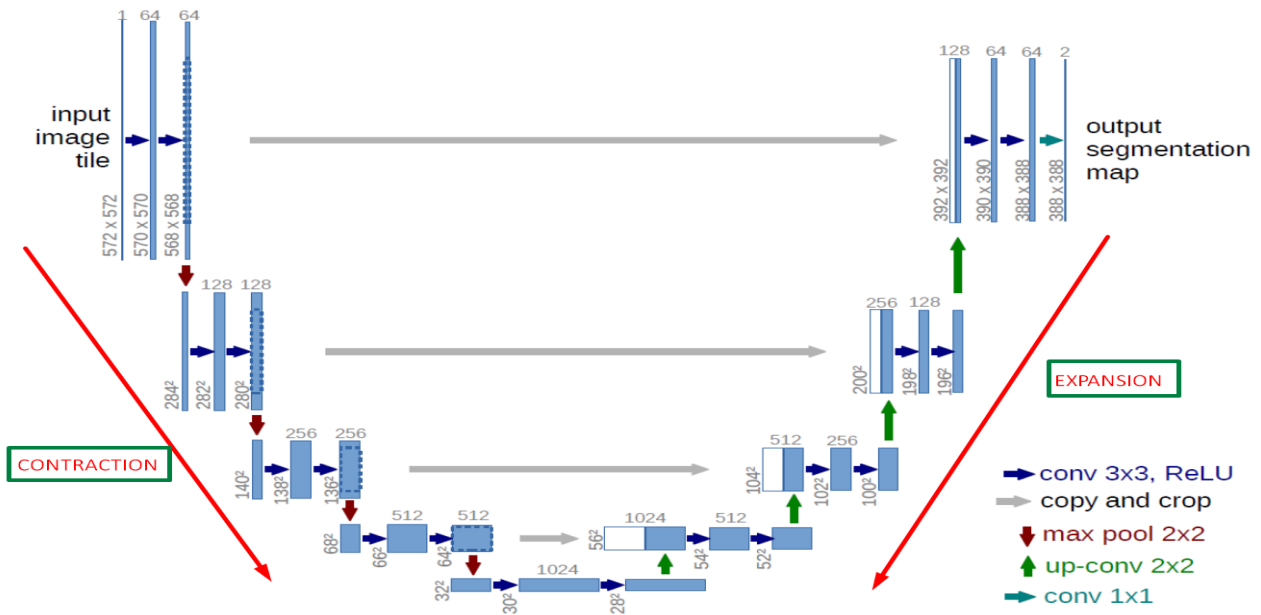
	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-



Şekil 10 Fully Convolutional Yapı

Encoder – Decoder

Bu yapı aşağıda görselleştirilmiştir. Contraction, encoder kısmını temsil etmektedir. Expansion ise decoder kısmını temsil etmektedir. İkisinin oluşturduğu yapı U-NET modelinin yapısıdır.



Şekil 11 U-NET Modeli

Projede encoder kısmı için “Imagenet” decoder kısım için ise birçok model denenmiştir. En iyi sonuç “Resnet34” modelinde elde edilmiştir. Aşağıdaki görsel backbone listesidir. Backbone decoder kısmında kullanılabilecek modeller olarak düşünülebilir.

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

Backbones

Type	Names
VGG	'vgg16' 'vgg19'
ResNet	'resnet18' 'resnet34' 'resnet50' 'resnet101' 'resnet152'
SE-ResNet	'seresnet18' 'seresnet34' 'seresnet50' 'seresnet101' 'seresnet152'
ResNeXt	'resnext50' 'resnext101'
SE-ResNeXt	'seresnext50' 'seresnext101'
SENet154	'senet154'
DenseNet	'densenet121' 'densenet169' 'densenet201'
Inception	'inceptionv3' 'inceptionresnetv2'
MobileNet	'mobilenet' 'mobilenetv2'
EfficientNet	'efficientnetb0' 'efficientnetb1' 'efficientnetb2' 'efficientnetb3' 'efficientnetb4' 'efficientnetb5' 'efficientnetb6' 'efficientnetb7'

Şekil 12 Backbone Listesi

2.2.4 Model Eğitimi


108 Görsel İle Olan Model Eğitimi

Elde edilen sonuçlar hiç iyi değildir. Veri setinin çok küçük olmasından dolayı grafikler zikzaklarla doludur.

Elde edilen sonuçlar:



Şekil 13 İlk Eğitim Sonuç Grafiği

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-



Şekil 14 İlk Eğitim Sonuç Görselleri

380 Görsel İle Olan Model Eğitimi

Bu aşamada eğitim doğruluğunu arttırabilmek için birçok parametre ile test edilmiştir.

Bu parametreler şu şekildedir:

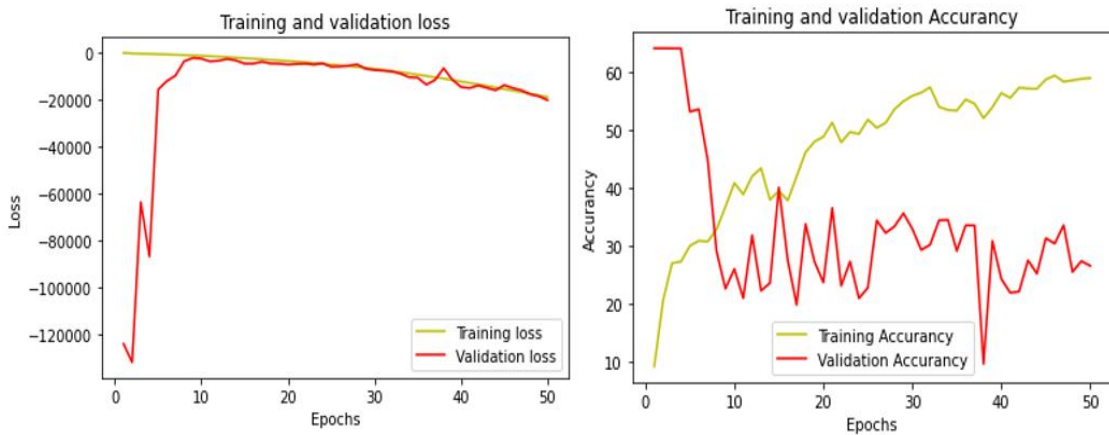
Görsel piksel: 128, 256

Model: Resnet34, Resnet50, Vgg16, InceptionV3


Batch Size: 64, 32, 16, 8

Epoch Sayısı: 100, 85, 65, 50

En iyi sonuç 256 piksel, Resnet34, 32 ve 50 epoch sonucunda elde edilmiştir.



Şekil 15 İkinci Eğitim Sonuç Grafikleri


	<p>.....</p> <p>KILAVUZU</p>	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-



Şekil 16 İkinci Eğitim Sonuç Görselleri - 1



Şekil 17 İkinci Eğitim Sonuç Görselleri - 2

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

Accuracy değeri sadece 0.60 değerinde kaldı. Bunun asıl nedeni üzerinde çalıştığım şeyin elbise görseli olmasıdır. Elbise görselleri çok çeşitli olmaktadır ve etiketlemesi çok zordur.

Accuracy değerinin az olmasının nedenleri:

- Veri setimin az olması
- Elimde yeterli verinin olmaması
- Veri setimin çok çeşitli olması

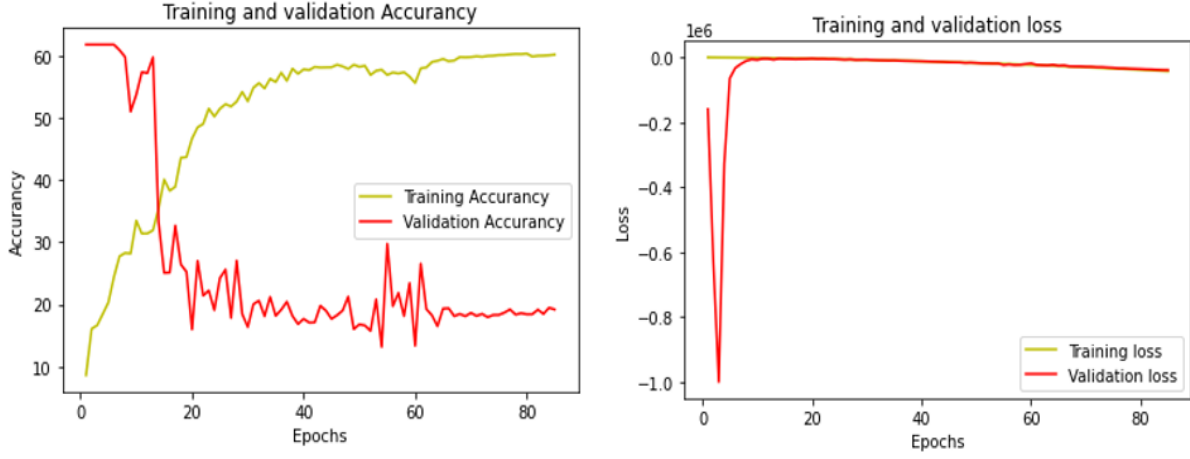


Şekil 18 Elbise Çeşitliliği

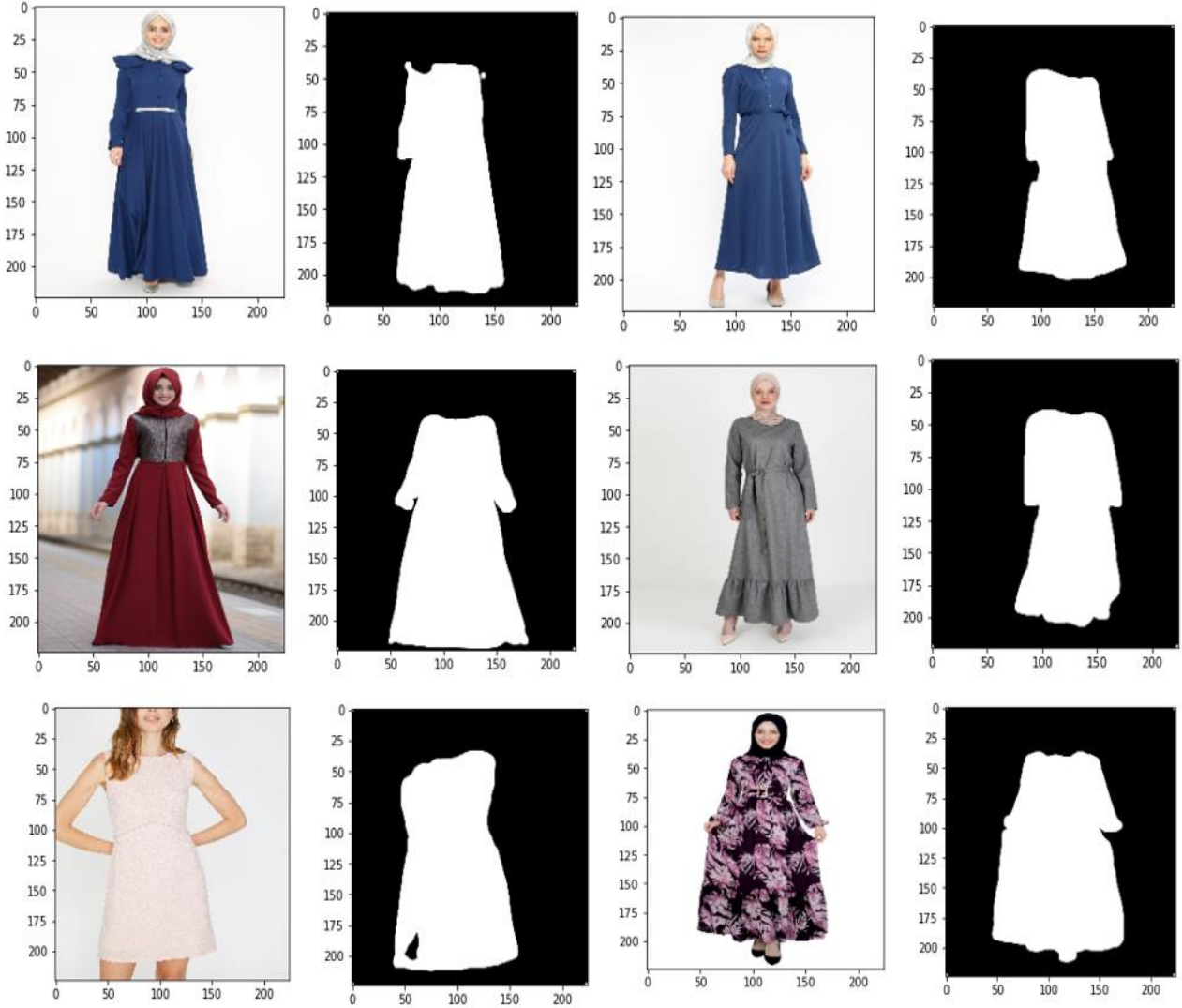
813 Görsel İle Olan Model Eğitimi

Bu aşamada da birçok parametre denemiştir. En iyi sonuç elde edilen parametreler:

256 piksel, Resnet34, 64 ve 85 epoch sonucunda elde edilmiştir. Accuracy oranı %61'dir.



Şekil 19 Son Eğitim Sonuç Grafikleri



Şekil 20 Son Eğitim Sonuç Görselleri

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

2.2.5 Maskeden Renklendirilmiş Görüntünün Elde Edilmesi

Maske görüntüsü ve asıl manken görüntüsünü reshape ederek kaydetmem gerekti. prediction_image olarak isimlendirilen değişken maske görüntüsüdür. Original olarak isimlendirilen değişken ise asıl manken görüntüsüdür. Daha sonra, elbise görüntüsü asıl rengine arkaplan ise beyaz olacak şekilde değiştirilmiştir.

```
[ ] prediction_image = prediction.reshape(224,224)
#prediction = model(image)
pred = np.dstack([prediction_image, prediction_image, prediction_image])
pred = (pred * 255).astype(np.uint8)
cv2.imwrite('/gdrive/My Drive/Segmentation/pred99.png', pred)
```

```
[ ] from PIL import Image
original = Image.open('/gdrive/My Drive/Segmentation/test3.jpg')
original = original.resize((224,224), Image.ANTIALIAS)
original.save('/gdrive/My Drive/Segmentation/original99.png')
```

```
import cv2
mask = cv2.imread('/gdrive/My Drive/Segmentation/pred99.png')
dress = cv2.imread('/gdrive/My Drive/Segmentation/original99.png')


_, mask = cv2.threshold(mask, thresh=180, maxval=255, type=cv2.THRESH_BINARY)
# copy where we'll assign the new values
background = np.copy(dress)
# boolean indexing and assignment based on mask
background[(mask==0).all(-1)] = [255,255,255]

fig, ax = plt.subplots(1,2,figsize=(12,6))
ax[0].imshow(cv2.cvtColor(dress, cv2.COLOR_BGR2RGB))
ax[1].imshow(cv2.cvtColor(background, cv2.COLOR_BGR2RGB))
```

Şekil 21 Renklendirilmiş Görüntü Google Colab Kodu

2.3. BAŞARI KRİTERLERİ

- Minimum başarı oranının %60 olması
- Tek bir görsel için maksimum 30 saniye içinde çıktı görsel elde edilmesi
- 300 görsel ve üzeri model eğitimi

	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

3. PROJE GÖRSEL SONUÇLAR


Projemi test ettiğimde aldığım sonuçlar aşağıdaki gibidir:



Şekil 22 Test Sonuçları - 1



Şekil 23 Test Sonuçları - 2

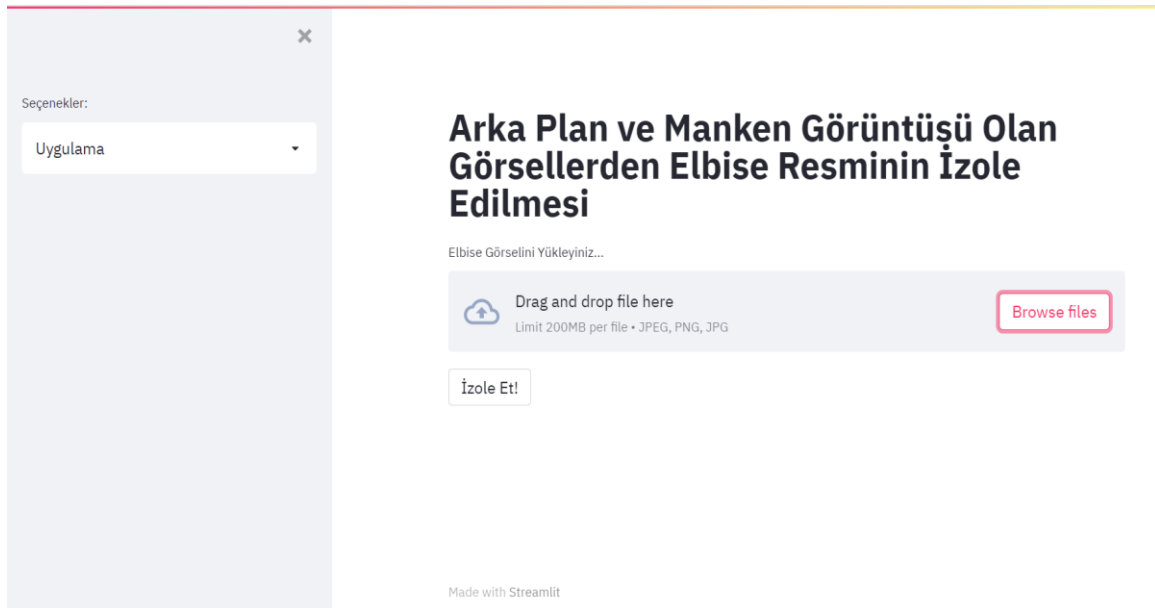
	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-




Şekil 24 Test Sonuçları – 3

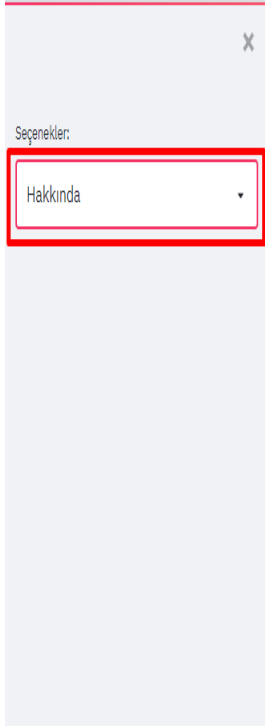
Elde edilen sonuçlar iyi değildir bunun nedeni aldığım accuracy değerinin düşük olmasıdır. Accuracy değerinin düşük olmasının nedeni de veri seti yani elbise görsellerinin çeşitliliği ve etiketlenme işlemidir.

Streamlit App Test Sonuçları



Şekil 25 Streamlit Giriş Ekranı

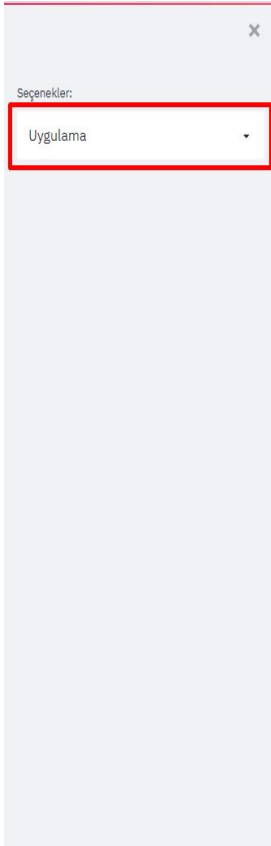
	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-



Şekil 26 Streamlit Hakkında Ekranı

Arka Plan ve Manken Görüntüsü Olan Görsellerden Elbise Resminin İzole Edilmesi

Katalog çekimleri bir kez yapıldığında bu resimlerin pazarlanması için çok ülke ve websitelerinde sergilenmesi gerekmektedir. Her website için resim üzerinde farklı değişiklikler yapılma ihtiyacı doğmaktadır. Ama elbise görüntüsü manken üzerinde olduğu için değişiklik yapabilmek konusunda kısıtlamalar oluşmaktadır. Bu geliştirilen proje bu sorunu çözebilmek aşamasında kullanılabilecek bir proje olmaktadır. Bu proje ile birlikte sadece elbise görüntüsü elde edilebilecektir. Bu da görsel üzerinde kolaylıkla değişiklik yapılabilmesine olanak sağlayacaktır. U-NET modeli kullanılarak girdi olarak verilen elbise görselinin maske görüntüsü elde edilmiştir. Daha sonra bu maske görüntüsü kullanarak istenilen çıktı elde edilmiştir.



Şekil 27 Streamlit Örnek Ekran

Arka Plan ve Manken Görüntüsü Olan Görsellerden Elbise Resminin İzole Edilmesi

Elbise Görselini Yükleyiniz...



Drag and drop file here
Limit 200MB per file • JPEG, PNG, 3PG

Browse files

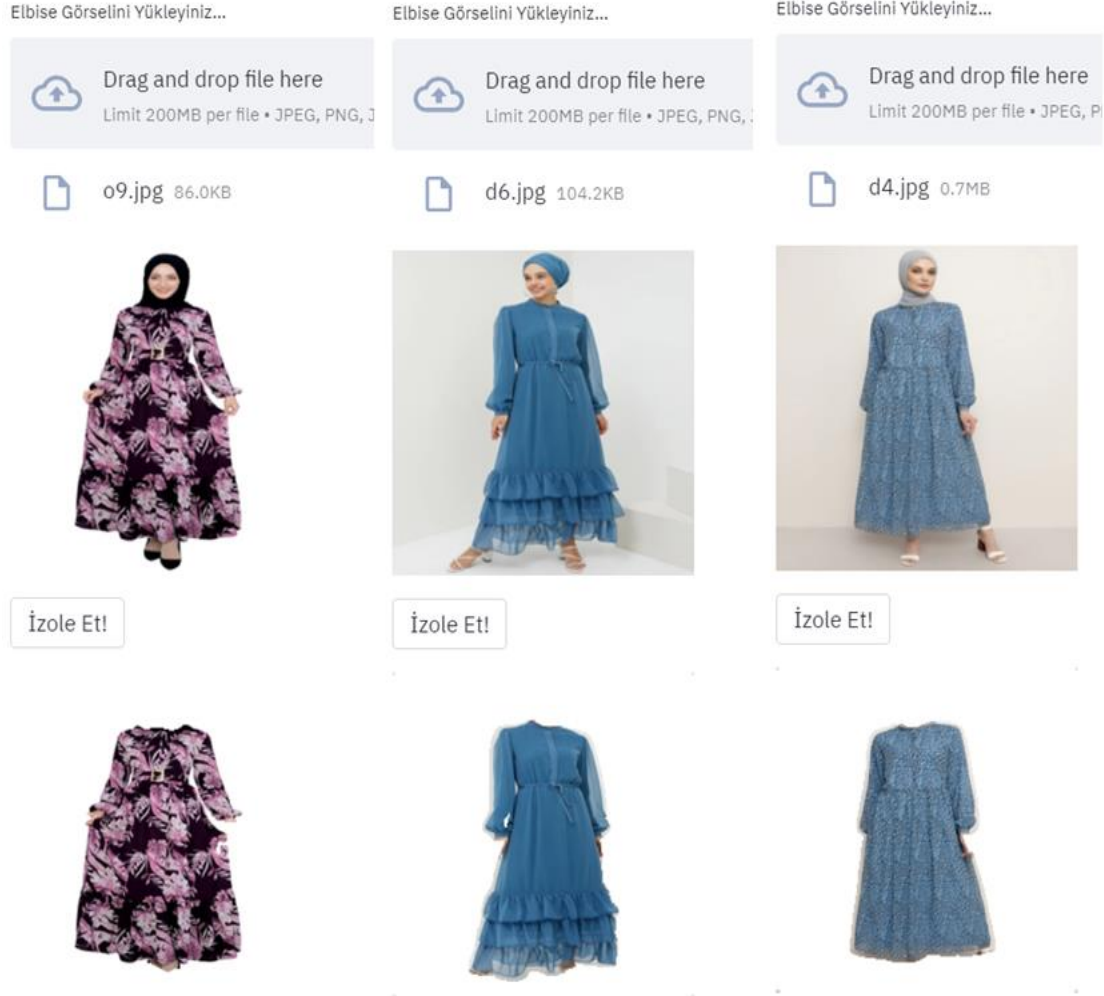
o9.jpg 86.0KB



İzole Et!



	KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-



Şekil 28 Streamlit Örnek Görsel Sonuçları

4. TARTIŞMA VE SONUÇ

4.1. EĞİTİM SONUCU

85 epoch olana kadar geçen süre yaklaşık olarak 20 dakika olarak görülmüştür. Alınan Accuracy değeri %61'i geçememektedir. Elbise görsel sayısı iki katına (380 görselden 813 görsel) çıkarılmasına rağmen sadece %2 oranında bir artış gözlenmiştir.

4.2 PROJENİN SONUCU

Çalışma sonucunda elde edilen yaklaşım ile website sahipleri bazı photoshop işlemleri ve projeyi kullanarak rahatlıkla elbise görsellerini websitelerinde kullanabilecektir. Halihazırdaki photoshop işlemleri daha kolay ve kullanışlı hale gelecektir. Bu proje bu işlemleri kolaylaştıracaktır. Daha sonraki aşamalar için başlangıç proje niteliğindedir.

 KILAVUZU	Doküman No:	KL-0003
		Yayın Tarihi:	31.05.2018
		Değ.No:	0
		Değ.Tarihi:	-

KAYNAKLAR

[1]Xuebin Qin, Zichen Zhang, Chenyang Huang, “U2-Net: Going Deeper with Nested U-Structure for Salient Object Detection”, University of Alberta, Canada, 5 Aug 2020.

[2]<https://manishsinghrajput96.blogspot.com/2020/08/how-to-dataset-annotations-for%20%20instance.html>

[3]<https://medium.com/free-code-camp/how-to-use-deeplab-in-tensorflow-for-object-segmentation-using-deep-learning-a5777290ab6b>

[4]https://github.com/qubvel/segmentation_models

[5]Evan Shelhamer, Jonathan Long, and Trevor Darrell, Member, IEEE. "Fully Convolutional Networks for Semantic Segmentation." arXiv:1605.06211v1 [cs.CV] 20 May 2016.

[6]Pongsate Tangseng, Zhipeng Wu, and Kota Yamaguchi. "Looking at Outfit to Parse Clothing." arXiv:1703.01386v1 [cs.CV] 4 Mar 2017.