

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BİR GÖRÜNTÜDEKİ EN ÖNEMLİ NESNENİN
DOMİNANT RENGİNİN TESPİTİ

16011049 – Tarık ÇARLI

16011033 – Yusuf ANI

BİLGİSAYAR PROJESİ

Danışman
Doç. Dr. Mine Elif Karslıgil

Haziran, 2020

TEŞEKKÜR

Bu çalışmayı yapmamızda şüphesiz en büyük katkıyı yapan, anlamsız bile olan sorularımıza sabırla cevap veren, takıldığımız yerlerde bize yol gösteren Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü Doç. Dr. M. Elif Karslıgil Hocamıza canı gönülden teşekkürlerimizi sunarız.

Bizi hayatta desteklemekten vazgeçmeyen, her zaman yanımızda olduğunu bildiğimiz, daha iyi insanlar olabilmemiz için kendi hayatlarından fedakarlık eden ebeveynlerimize sonsuz hürmetlerimizi sunarız.

Tarık ÇARLI
Yusuf ANI

İÇİNDEKİLER

SİMGE LİSTESİ	v
KISALTMA LİSTESİ	vi
ŞEKİL LİSTESİ	vii
TABLO LİSTESİ	viii
ÖZET	ix
ABSTRACT	x
1 Giriş	1
2 Ön İnceleme	2
3 Fizibilite	3
3.1 Teknik fizibilite	3
3.1.1 Yazılım fizibilitesi	3
3.1.2 Donanım fizibilitesi	7
3.2 İş gücü ve zaman fizibilitesi	8
3.3 Yasal fizibilite	9
3.4 Ekonomik fizibilite	9
4 Sistem Tasarımı	10
4.1 Veri Kümesi Seçimi	11
4.2 Anlamsal Bölütleme	11
4.3 Anlamsal Bölütleme Modelinin Mimarisi	13
4.4 Bağlantılı Bileşen Etiketleme	17
4.5 Renk Niceleme	18
4.6 Görselleştirme, Histogram	19
4.7 Proje Kullanımı	20
5 Deneysel Sonuçlar	21

6 Test, Performans Analizi ve İyileştirme	30
6.1 Başarım Analizi	30
6.2 Hız Analizi	30
6.2.1 Eğitim Zamanı Hız Analizi	30
6.2.2 Test Zamanı Hız Analizi	31
7 Sonuç	32
Referanslar	33
Özgeçmiş	34

SİMGE LİSTESİ

k K-means küme sayısı

KISALTMA LİSTESİ

ROI	İlgi Bölgesi
CNN	Evrişimli sinir ağı
CCL	Bağlı Bileşen Etiketleme
JPEG	Birleşmiş Fotoğraf Uzmanları Grubu
PNG	Taşınabilir Ağ Grafikleri

ŞEKİL LİSTESİ

Şekil 3.1	Anlamsal bölütleme yöntemlerinin başarı durumları	6
Şekil 3.2	Projenin Gant diagramı	8
Şekil 4.1	Projenin Blok diagramı	10
Şekil 4.2	A: Giriş görüntüsüdür. B: Çıktı görüntüsüdür.	12
Şekil 4.3	A: Giriş görüntüsüdür. B: Çıktı görüntüsüdür.	12
Şekil 4.4	Farklı Mimarilerin Performansları	12
Şekil 4.5	DeepLabV3+ Modelinin Mimarisi	13
Şekil 4.6	Kodlayıcı-kod çözücü ağlar	14
Şekil 4.7	Atraus Evrişimi Formülü	15
Şekil 4.8	Derinlemesine evrişim ve Noktasal evrişim	15
Şekil 4.9	A: Giriş görüntüsüdür. B: Çıktı görüntüsüdür.	17
Şekil 4.10	A: Giriş görüntüsüdür. B: Çıktı görüntüsüdür.	18
Şekil 4.11	A: Giriş görüntüsüdür. B: Çıktı görüntüsüdür.	19
Şekil 4.12	A: Giriş görüntüsüdür. B: Çıktı görüntüsüdür.	19
Şekil 5.1	Projenin at görseli üzerindeki çıktısı	22
Şekil 5.2	Projenin uçak görseli üzerindeki çıktısı	23
Şekil 5.3	Projenin inek görseli üzerindeki çıktısı	23
Şekil 5.4	Projenin koltuk takımı görseli üzerindeki çıktısı	24
Şekil 5.5	Projenin tren görseli üzerindeki çıktısı	24
Şekil 5.6	Projenin insanların olduğu görsel üzerindeki çıktısı	25
Şekil 5.7	Projenin araba görseli üzerindeki çıktısı	25
Şekil 5.8	Projenin palyaço görseli üzerindeki çıktısı	26
Şekil 5.9	Projenin kuş görseli üzerindeki çıktısı	26
Şekil 5.10	Projenin kedi görseli üzerindeki çıktısı	27
Şekil 5.11	Projenin insanların olduğu görsel üzerindeki çıktısı	27
Şekil 5.12	Projenin insanların olduğu görsel üzerindeki çıktısı	28
Şekil 5.13	Projenin insanların olduğu görsel üzerindeki çıktısı	28
Şekil 5.14	Projenin atların olduğu görsel üzerindeki çıktısı	29
Şekil 5.15	Projenin arabaların olduğu görsel üzerindeki çıktısı	29

TABLO LİSTESİ

Tablo 3.1	Proje için Öne çıkan veri kümeleri	4
Tablo 3.2	Gider Tablosu	9
Tablo 6.1	Anket sonuçları	30
Tablo 6.2	Bileşenlerin Saniye Cinsinde Aldığı Süreler	31

Bir Görüntüdeki En Önemli Nesnenin Dominant Renginin Tespiti

Tarık ÇARLI

Yusuf ANI

Bilgisayar Mühendisliği Bölümü

Bilgisayar Projesi

Danışman: Doç. Dr. Mine Elif Karslıgil

Çalışmada bir görüntüdeki en önemli nesnenin dominant renginin tespiti amaçlanmıştır. Çalışmada günlük hayatta birçok yerde kullanılabilir. Bir şirketteki çalışanların kıyafetlerinde tercih ettiği renklerin tespit etmede, bir moda defilesindeki kıyafetlerin en popüler rengini tespit etmede veya bir bölgedeki en popüler araba rengi tercihi saptamada kullanılabilir.

Çalışmada görüntüye bir dizi farklı işlem sırasıyla uygulanarak amaçlanan sonuç elde edilir. Öncelikle giriş görüntüsü uygun formata çevirilerek derin öğrenme destekli anlamsal bölütleme yöntemiyle bölümlere ayırır. Bölümlere ayrılmış görüntüdeki en büyük nesnenin tespiti için bağlantılı bileşen etiketleme yöntemi kullanılır. Ardından sadece en büyük nesnenin piksel değerleri k-ortalama kullanarak renk niceleme fonksiyonuna gönderilir. Son olarak elde edilen renkleri indirgenmiş en büyük nesnenin pikselleri grafikler yardımıyla kullanıcıya gösterilir.

İşlemin başarısı anket yoluyla test edilmiştir. Modelimizin 20 görüntüye ait çıktıları, yaklaşık 20 kişiye gösterilerek, 1 ila 5 arasında puan vermeleri istenmiştir. Kullanıcıların verdiği puanların ortalaması alınarak sistemin başarısı gösterilmiştir.

Anahtar Kelimeler: Anlamsal Bölütleme, Bağlı bileşen etiketleme, Derin öğrenme, Renk niceleme, K-ortalama

ABSTRACT

Detection of the Dominant Color of the Most Important Object in an Image

Tarık ÇARLI

Yusuf ANI

Department of Computer Engineering

Computer Project

Advisor: Associate Prof. Dr. Mine Elif Karşılıgil

In the research, it was aimed to determine the dominant color of the most important object in an image. It can be used in many places in daily life in the study. It can be used to determine the colors preferred by the employees of a company in their clothes, to determine the dominant color of the clothes in a fashion show or to determine the dominant car color preference in a region.

In the research, the intended result is obtained by applying a series of different processes to the image. First of all, the introduction image is converted into a suitable format and it is divided into sections with semantic integration method supported by deep learning. The linked component labeling method is used to detect the largest object in the segmented image. Then, only the pixel values of the largest object are sent to the color quantization function using k-mean. Finally, the pixels of the largest object whose colors are reduced are displayed to the user with the help of graphics.

The success of the transaction has been tested through a questionnaire. The output of our model for 20 images was shown to about 20 people and asked to score between 1 and 5. The success of the system is shown by taking the average of the points given by the users.

Keywords: Semantic segmentation, Connected component labelling, Deep learning, Color quantization, K-means.

1

Giriş

Bu çalışma bir resmin içindeki en büyük nesnenin en popüler rengini tespit etmek için geliştirilen görüntü işleme teknikleri ve anlamsal bölütleme (semantic segmentation) kullanan yazılım uygulamasıdır. Bu çalışmayı gerçekleştirme motivasyonumuz birçok kullanım alanı olmasına rağmen bu konuda yapılan çalışmaların kapsamının dar tutulmasıdır. Resimdeki en büyük nesneyi bulacak ve resmin en popüler rengini bulacak çalışmalar vardır. Bu çalışma ayrı ayrı alanlarda kullanılmıştır. Biz bahsedilen iki çalışma konusunu birleştirerek tek bir çalışma içinde gerçekleştirdik. Bu çalışma hayatta birçok alanda kullanılabilir. Bu alanlardan bazıları örnek olarak aşağıda belirtilmiştir.

- **Defile fotoğraflarını kullanarak yılın popüler renklerini belirleme.** Defile fotoğraflarında manken görüntünün ilgi odağı olacağı için mankenin üzerindeki kıyafetlerin en popüler rengini bulacaktır.
- **Film görüntülerini kullanarak filmin ana karakterinin daha fazla kullandığı rengi tespit etme.** Filmin ana karakteri bir çok sahnede en büyük nesne olduğundan bu proje ile ana karakterin çok kullandığı renkleri bulabiliriz.
- **Ev görüntülerini kullanarak evlerde kullanılan popüler renkleri belirleme.** Ev görüntülerinde en büyük nesne ev olacağı için veri kümemizdeki ev görüntülerini kullanarak evlerde kullanılan popüler renkleri belirleyebiliriz.

2 Ön İnceleme

Bu bölümde GEBNRA projesinin veya benzerlerinin daha önce yapılan örnekleri incelenecektir. GEBNRA kapsamında yapılan araştırmalar sonucunda en büyük nesneyi bulmak ve en en popüler rengi tespit etmek için ayrı ayrı projeler bulunmuştur.

En büyük nesneyi bulmak için MIT tarafından uygulanmış [1] farklı mimarileri (RESNET , MOBILENET vb.) destekleyen bir yapıdadır. Projede veri kümesi olarak ADE20K veri kümesini kullanılmaktadır. ADE20K yaklaşık 23.000 adet resim ve karşılık gelen objelerin bilgileri yer almaktadır. Projede yapay zeka kütüphanesi olarak Pytorch kullanılmıştır. Projenin dökümanlarında yüksek başarı oranlarıyla anlamsal bölütleme yapıldığı görülmektedir fakat yapay zeka modelinin eğitimi için birden fazla grafik işlemci birimi desteği gerektiği ve çalışmamızın minimum gereksinimlerini karşılamadığı için model uygulanamamıştır.

Bir diğer anlamsal bölütleme modeli [2] ise kendilerinin hazırladığı bir veri kümesi kullanılmaktadır. Projede yapay zeka kütüphanesi olarak Keras kullanılmıştır. Projenin anlaşılır bir dökümantasyonu olmasına rağmen eğitim sonucundaki modelin başarı oranı istediğimiz oranlarda değildir.

Son anlamsal bölütleme modeli olarak Tensorflow'un kendi yayınladığı Deeplab kütüphanesi kullanılmıştır[3]. Projede farklı mimarilerde eğitilmiştir.Bu mimariler ResNet-v1-50,101 , PNASNet , MobileNetv2 ,Auto-DeepLab ve Xception mimarileri olarak sıralanabilir. Proje öğrenim aktarımı için belirtilen mimariler üzerinde Imagent , COCO veri kümelerinde eğitilmiş modeller sunmaktadır. Ayrıca projede öğrenim aktarımı için farklı veri kümeleri desteklenmektedir. Bu veri kümeleri PASCAL VOC 2012 , Cityscapes , ADE20K şeklindedir. Projede yapay zeka kütüphanesi olarak Tensorflow seçilmiştir.

Bu bölümde projenin yapılabilirliği değerlendirilmiştir. Projenin hedeflerine ulaşması için hangi teknolojilerin seçildiğinin nedenleri ve hangi kriterlere göre seçildiği belirtilmiştir. Projede hangi kaynakların ne kadar kullanılacağıda bu bölümde açıklanmıştır. Bu bölüm teknik fizibilite, iş gücü ve zaman fizibilitesi, yasal fizibilite ve ekonomik fizibilite olmak üzere 4 alt bölümde incelenmiştir.

Bu projede bir görüntüdeki en büyük nesnenin en popüler rengini bulmak için çalışma yapılmıştır. Görüntüdeki nesneleri tespit etmek için anlamsal bölütleme yapılması, ardından en büyük nesnenin bulunması gerekmektedir. Son olarak en büyük nesnenin renklerini belli bir sayıya kadar düşürülmesi gereklidir. Günümüzde anlamsal bölütleme için en başarılı modeller derin öğrenme tabanlı olduğu için derin öğrenme tabanlı anlamsal bölütleme kullanılmıştır. Anlamsal bölünme işleminden sonra en büyük nesneyi bulmak için bağlantılı bileşen etiketleme (connected component labelling) yöntemi uygulanmıştır. En büyük nesnenin toplam renk sayısını azaltmak için renk niceleme (color quantization) yöntemi uygulanmıştır. Ardından elde ettiğimiz veriler yardımcı kütüphaneler kullanılarak görselleştirilmiştir.

3.1 Teknik fizibilite

Bu bölümde projenin yazılımsal ve donanımsal gereksinimlerinin değerlendirilmiştir.

3.1.1 Yazılım fizibilitesi

Projenin gerçekleştirileceği programlama dili seçiminde kod yazımı kolay, kod okunurluğu yüksek, topluluğu gelişmiş ve kütüphaneler bakımından zengin olması dikkate alınmıştır. Modern programlama dillerinin hepsi belirli bir performansın üzerinde olduğu için ve projemizde akan veri (stream) üzerinde işlem yapmadığımız için performans dil seçimimizde bir kriter olarak değerlendirilmemiştir. Dil seçimi, bahsettiğimiz kriterler göz önüne alındığında en iyi seçenek olarak Python

programlama dili öne çıkmıştır. Pythonun alternatifi olarak C++ dili kullanılabilir. C++’da projeyi gerçekleştirmek için gerekli kütüphanelere sahiptir. Yukarıda bahsettiğimiz nedenlerden dolayı Python programlama dili tercih edilmiştir.

Projemizde kullanabileceğimiz veri kümeleri aşağıda tabloda belirtilmiştir.[4]

Tablo 3.1 Proje için Öne çıkan veri kümeleri

İsim	Amaç	Yıl	Sınıf	Örnek
PASCAL VOC	Genel	2012	21	2900
SBD	Genel	2011	21	11000
Microsoft COCO	Genel	2014	+80	120000
DAVIS	Genel	2016	4	6000
PASCAL-Context	Genel	2014	540	10000

Tabloda öne çıkan veri kümeleri sergilenmiştir. Bu veri kümeleri içinden diğerlerine göre küçük ve arka plan için ayrı bir sınıfı olduğundan PASCAL VOC 2012 veri kümesi seçilmiştir.

Derin öğrenme tabanlı uygulamalar popülerleştikten sonra anlamsal bölütlemedeki çalışmalar başarı oranlarının yükselmesi sebebiyle derin öğrenmeye kaymıştır. Başarı oranı sebebiyle bu çalışmada derin öğrenme kullanan yöntemlerden arasından yöntem seçilmiştir. Derin öğrenmeden önceki değerlendirilmeyen yöntemler kısaca aşağıda listelenmiştir. *Eski anlamsal bölütleme yöntemleri*

1. Eşikleme (Thresholding): Görüntüyü ön plan ve arka plan olarak ikiye böler. Belirtilen bir eşik değerini kullanarak nesneleri izole etmek için pikselleri iki düzeye ayırır. Eşikleme, gri tonlu görüntüleri ikili görüntülere dönüştürür.
2. K-kümeleme (K-means clustering): Verilerdeki grupları tanımlar ve K değişkeni grup sayısını temsil eder. Yöntem, özellik benzerliğine dayalı olarak her pikseli gruplardan birine atar.
3. Histogram tabanlı görüntü bölütleme (Histogram-based image segmentation): Pikselleri “gri düzeylere” göre gruplamak için bir histogram kullanır. Basit görüntüler bir nesne ve bir arka plandan oluşur. Arka plan genellikle bir gri düzeydir ve daha büyük bir varlıktır. Böylece, büyük bir tepe histogramdaki arka plan gri seviyesini temsil eder. Daha küçük bir tepe, başka bir gri seviye olan nesneyi temsil eder.
4. Kenar algılama (Edge detection): Parlaklıktaki keskin değişiklikleri veya süreksizlikleri tanımlar.

Yukarıdaki yöntemlerin bir çoğu projemizde kullanmaya uygun değildir, diğerleri ise başarı oranı bakımından derin öğrenmeli yöntemlere göre düşüktür.

Derin öğrenme ile birlikte anlamsal bölütleme alanında bir çok yeni yöntem çıkmıştır. Bu yöntemler aşağıda listelenmiştir.[5]

1. UWeakly- and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation
2. Fully Convolutional Networks for Semantic Segmentation
3. U-Net: Convolutional Networks for Biomedical Image Segmentation
4. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation
5. Multi-Scale Context Aggregation by Dilated Convolutions
6. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs
7. Rethinking Atrous Convolution for Semantic Image Segmentation
8. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation
9. FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation
10. Improving Semantic bölütleme via Video Propagation and Label Relaxation
11. Gated SCNN: Gated Shape CNNs for Semantic bölütleme

Method	mIOU
Adelaide_VeryDeep_FCN_VOC [85]	79.1
LRR_4x_ResNet-CRF [25]	79.3
DeepLabv2-CRF [11]	79.7
CentraleSupelec Deep G-CRF [8]	80.2
HikSeg-COCO [80]	81.4
SegModel [75]	81.8
Deep Layer Cascade (LC) [52]	82.7
TuSimple [84]	83.1
Large_Kernel_Matters [68]	83.6
Multipath-RefineNet [54]	84.2
ResNet-38_MS_COCO [86]	84.9
PSPNet [95]	85.4
IDW-CNN [83]	86.3
CASIA_IVA_SDN [23]	86.6
DIS [61]	86.8
DeepLabv3	85.7
DeepLabv3-JFT	86.9

Table 7. Performance on PASCAL VOC 2012 *test* set.

Şekil 3.1 Anlamsal bölütleme yöntemlerinin başarı durumları

Şekil 3.1’de modeller incelenmiştir. Günümüzde anlamsal bölütleme konusundaki çalışmalar baş döndürücü hızda ilerlemektedir. Bu kadar seçeneğin arasında projeye en uygun olanını belirlemek ve seçmek zor bir süreç olmuştur. Uzun incelemelerin sonucunda, seçtiğimiz veri setinde en iyi sonucu veren modellerden biri olan Deeplapv3 modeli tercih edilmiştir. Deeplapv3 modeli Tensorflow kütüphanesini kullanmaktadır. Günümüzde derin öğrenme konusunda en popüler kütüphanelerden birisidir. Python’da Tensorflow’un yanında Keras, Pytorch, Sci-kit learn gibi farklı kütüphaneler vardır. Anlamsal bölütleme sonucunda bölgelerin en büyüğünü bulmak için bağlı bileşen etiketleme[6] yöntemi kullanılmıştır

3.1.2 Donanım fizibilitesi

Projenin çalışması için bir işletim sistemine ihtiyaç duyulmaktadır. İşletim sistemi seçenekleri: Linux Mint, Elementary OS, Ubuntu OS ve Windows10 işletim sistemleridir. Kullanım kolaylığı, açık kaynak kodlu oluşu ve diğerlerine göre topluluğunun büyük ve yardımsever olması nedeniyle Ubuntu işletim sistemi tercih edilmiştir.

Ubuntu versiyon 18.04 minimum sistem gereksinimleri:

2 GHz çift çekirdek işlemci,

2 GB sistem belleği,

25 GB boş disk alanı,

ve işletim sistemini yüklemek için DVD sürücüsü veya usb port girişi.

Projeyi gerçekleştirme dili olarak seçilen Python için sistem gereksinimleri:

İşlemci: Intel® Core™ i5 işlemcisi 4300M at 2.60 GHz (2 çekirdek, her çekirdeğe iki thread) ve 5 GB boş disk alanı.

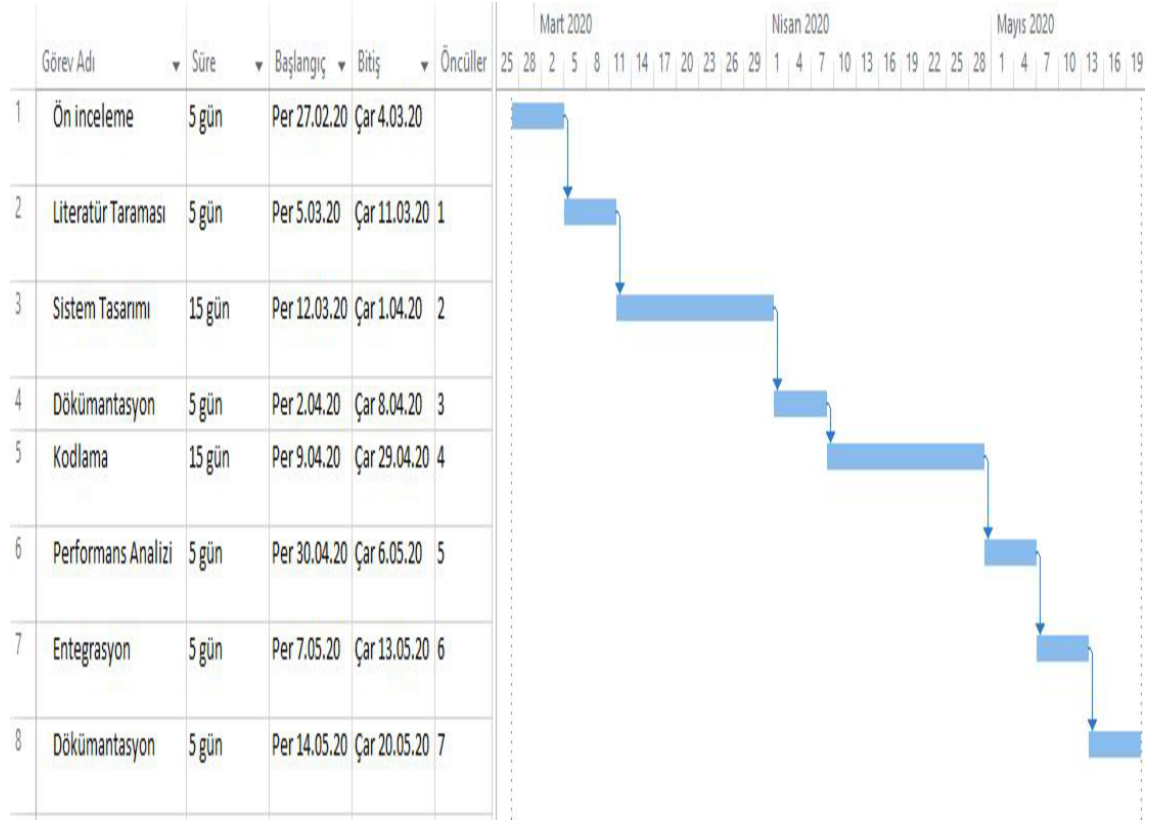
Geliştirme aşamasında modelin eğitimi için ek olarak güçlü bir ekran kartı olması, geliştirme aşamasını kısaltacaktır. Projenin modeli Google tarafından ücretsiz ve ücretli şekilde sunduğu Colab ortamında geliştirilmiştir. Şirket için ve kendi bilgisayarında geliştirecek kişiler için önerilen ekran kartı 2 Gb bellek kapasitesi ve 256 bit veri yoluna sahip olması uygun bulunmaktadır. Projenin canlı ortamı için bu gereksinime ihtiyaç yoktur.

Canlı proje için toplam da ihtiyaç duyulan bilgisayar gücü:

2 GHz çift çekirdek işlemci, 4 GB sistem belleği, 30 GB boş disk alanı, ve işletim sistemini yüklemek için DVD sürücüsü veya usb port girişidir .

3.2 İş gücü ve zaman fizibilitesi

Projede 2 kişi haftada 2 gün günde 8 saat olarak çalışmaktadır. Projenin planlamasına ait gant diagramı aşağıda verilmiştir.



Şekil 3.2 Projenin Gant diagramı

3.3 Yasal fizibilite

Projeyi gerçekleştirmek için herhangi bir yasal koşul yoktur ve etik kurul onayı gerekmemektedir. İzinsiz olarak bir veri kümesi kullanılmamıştır. Projede kullandığımız üçüncü parti hizmet veren yazılımlar açık kaynak kodludur ve sahip olduğu lisanslar aşağıda belirtilmiştir.

- Python: PSF lisansına sahiptir.
- matplotlib: PSF lisansına sahiptir.
- Deeplabv3: MIT lisansına sahiptir.
- Scikit-learn: BSD lisansına sahiptir.
- PIL: The open source PIL Software lisansına sahiptir.
- Pascal Voc: lisans belirtilmemiştir.

Python yazılım vakfı(PSF) BSD stili bir yazılımdır. GNU GPL ile uyumlu bir lisanstır. Yukarıdaki lisansı belirtilen yazılımların projemizde kullanmak için yasal bir engel yoktur.

3.4 Ekonomik fizibilite

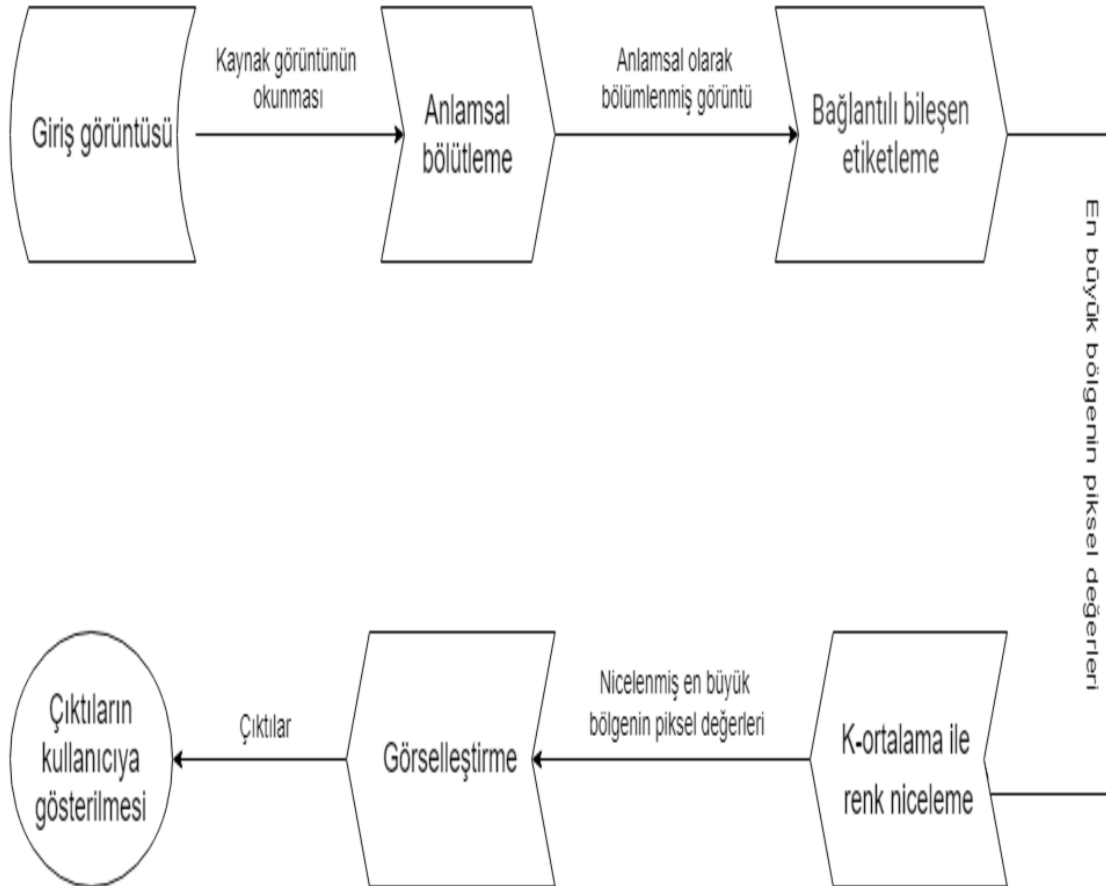
Toplam Gider						
Çalışan/Ay	Çalışan Sayısı	Ay	Toplam Çalışan/Ay	Aylık Çalışan Ücreti	Donanım Gideri	Toplam Gider
0,40	2	3	2.4	15000 TL	3000 TL	39000 TL

Tablo 3.2 Gider Tablosu

Projede 2 kişi, haftada 2 gün ve günde 8 saat olarak çalışmaktadır. 1 kişi haftada 2 gün çalışıyorsa bu kişi 1 ay boyunca toplam 0.4 adam-ay çalışmıştır. Projenin tamamlanma süresi 3 aydır. Bu kişi, proje üzerinde 3 ayda toplam 1.2 adam-ay çalışmış demektir. Proje üzerinde toplam 2 kişi çalıştığından dolayı projenin çalışan ücreti toplam 2.4 adam-ay demektir. Çalışanların birbirine göre üstünlükleri olmadığından dolayı ikisi de aynı ücreti alabilir. Günümüzde bir çalışanın ortalama maaşı 15000 TL dir. Toplam çalışma süresi 2.4 ay olduğuna göre projenin toplam çalışan ücreti 36000 TL dir. Proje Google bulut hizmeti olarak sunduğu Colab ortamının ücretsiz versiyonu üzerinde geliştirilebilir. İnternete bağlanacak orta özellikte bir bilgisayar ortalama 3000 TL dir. Projenin Toplam maliyeti 39000 Tl olarak hesaplanmıştır.

4 Sistem Tasarımı

Sistemin çalışma şeklinin görselleştirilmiş hali aşağıda blok diagramı olarak rapora eklenmiştir. Dosya sisteminden alınan görüntülere modellenin uygulanması için yeniden boyutlandırma ön işlemi uygulanmıştır.



Şekil 4.1 Projenin Blok diagramı

4.1 Veri Kümesi Seçimi

Veri kümesi seçimi anlamsal bölütlemenin başarısını dolayısıyla da projenin başarısını etkileyeceği için çok önemlidir. Seçilecek veri kümesi temel olan nesneleri tespit edebiliş arkaplan olan kısımlarını da işaretlemelidir. Bu şartları göz önüne aldığımızda en uygun seçeneğin Pascal VOC 2012 olduğu ortaya çıkmıştır. Pascal Voc 2012 veri kümesinde toplam 4 ana sınıf ve bunların 20 tane alt sınıfı işaretlenmiştir. Pascal VOC 2012 veri kümesindeki sınıflar aşağıda listelenmiştir.

1. İnsan: İnsan.
2. Hayvan: Kuş, Kedi, İnek, Köpek, At, Koyun.
3. Vasıta: Uçak, Bisiklet, Tekne, Otobüs, Araba, Motorsiklet, Tren.
4. Kapalı alan: Şişe, Sandalye, Yemek masası, Saksı bitkisi, Koltuk takımı, Televizyon, Monitör.

Bu sınıflar dışında Pascal Voc 2012 veri kümesinin bize sunduğu bir diğer avantaj, eğer fotograftaki herhangi bir piksel sınıflarımızdan biriyle ilişkilendirilemezse arka plan olarak sınıf ataması yapıyor. Bu sayede sadece gerçekten ilgilenilen sınıfların nesnelere bağlantılı bileşem etiketleme yöntemi uygulanıp daha hızlı sonuç alınabilir.

4.2 Anlamsal Bölütleme

Yapılan çalışmalarda görüntüde ön planda olabilecek nesne veya nesneleri tanımlamak gerekmektedir. Bu sayede görüntünün en popüler rengini bulurken görüntünün ilgilenmediği kısımlardaki arkaplan renklerinin hesaplamaya dahil edilmeyerek başarının artması planlanmıştır. Bunu yapabilmek için görüntüdeki nesneleri tespit edip onları bölütlemek gerekmektedir. Son zamanlarda yapılmış görüntü bölütleme çalışmalarda derin öğrenme tabanlı bölütlemelerin yapılması ve başarı oranlarının diğer yöntemlere göre daha yüksek olması sebebiyle bölütleme derin öğrenme tabanlı seçilmiştir.

Bölütlemenin birden fazla çeşiti vardır. Bunlardan bazıları anlamsal bölütleme, nesne tespiti, örnek bölütlemesidir. Çalışmada anlamsal bölütlemenin seçilmesinin sebebi aynı türdeki nesneleri aynı bölümlerde göstererek, görüntüdeki ilgili nesne hakkında daha fazla bilgi toplamaktır. Örnek olarak sokak fotoğrafını içeren bir görüntüde birden fazla araba varsa bizim için tüm arabaların rengi bilgi değeri taşımaktadır. Örnek olarak anlamsal bölütleme bileşeninin girdi ve çıktı görüntüleri aşağıda verilmiştir.



Şekil 4.2 A: Giriş görüntüsüdür. B: Çıktı görüntüsüdür.



Şekil 4.3 A: Giriş görüntüsüdür. B: Çıktı görüntüsüdür.

		Top-1 accuracy	Top-5 accuracy
VGGNet – 1 st Runner Up in ILSVRC 2014	VGG-16	0.715	0.901
ResNet – Winner in ILSVRC 2015	ResNet-152	0.770	0.933
Inception-v3 – 1 st Runner Up in ILSVRC 2015	Inception V3	0.782	0.941
	Xception	0.790	0.945

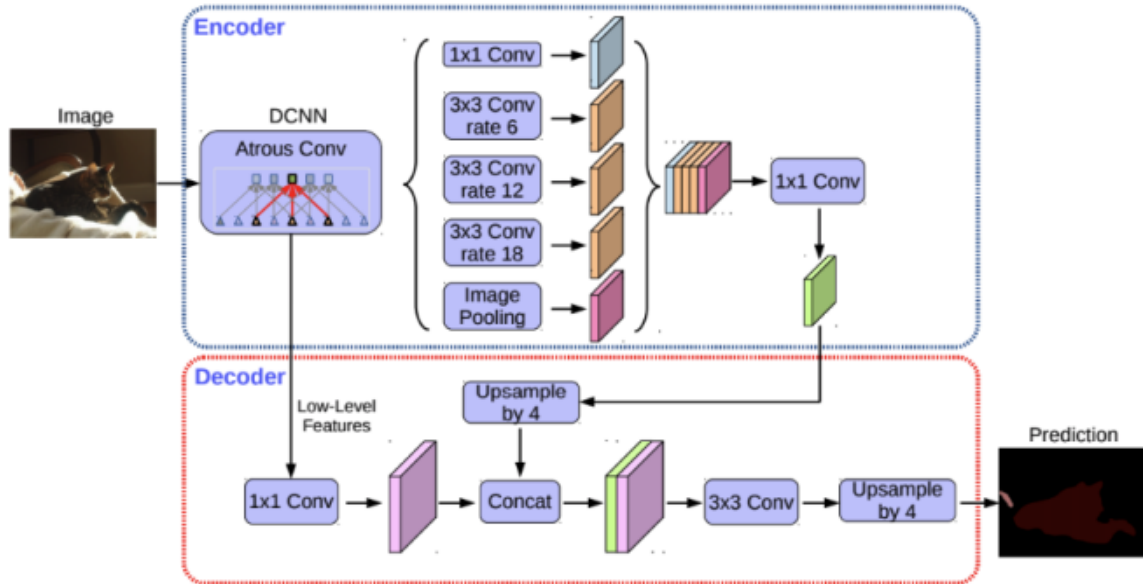
ImageNet: Xception has the highest accuracy

Şekil 4.4 Farklı Mimarilerin Performansları

4.3 Anlamsal Bölütleme Modelinin Mimarisi

DeepLab modeli genel olarak iki adımdan oluşur:

1. Kodlayıcı aşama: Bu aşamanın amacı görüntüden gerekli bilgileri elde etmektir. Bu, önceden eğitilmiş bir Evrişimsel Sinir Ağı (CNN) kullanılarak yapılır. Görüntü sınıflandırması için yapılmış çalışmalarda CNN kullanılmasının temel sebebi, evrişimsel katmanların bir görüntüdeki farklı özellikleri araması ve bu bilgileri sonraki katmanlara aktarmasıdır. Temel bilgileri içeren bölütleme görevi için, görüntüdeki mevcut nesneleri ve konumları kolayca bulunabilir çünkü CNN ağının sınıflandırmada iyi sonuç vermektedir.
2. Kod çözücü aşama: Kodlama aşamasında çıkarılan bilgiler burada uygun boyutlardaki çıktıları yeniden yapılandırmak için kullanılır.



Şekil 4.5 DeepLabV3+ Modelinin Mimarisi

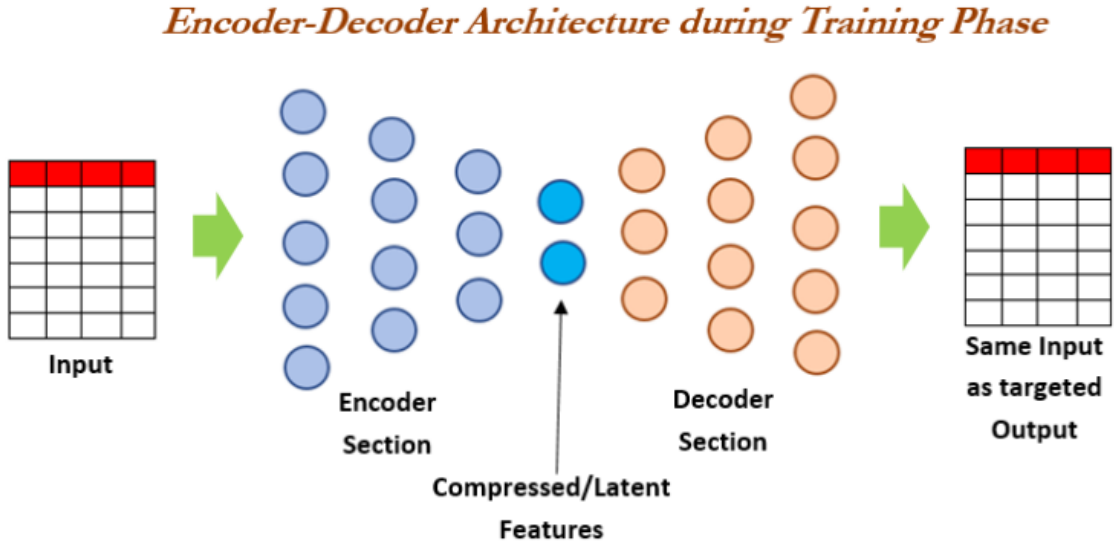
DeepLab mimarisi, iki popüler sinir ağı mimarisinin birleşimine dayanmaktadır:

1. Uzaysal Primit Ortaklama
2. Kodlayıcı-Çözücü Ağlar

CNN'lerle çalışırken modelimizin nesnelerin boyutundaki değişikliklere karşı sağlam olmalıdır. Bunun nedeni, modelimiz yalnızca küçük nesnelerin görüntüleri kullanılarak eğitilmişse, giriş görüntülerinin ölçekli versiyonlarında iyi performans göstermeyebilir. Bu sorun, uzamsal piramit ortaklama ağları kullanılarak çözülebilir. Bunlar, eğitim için girişin birden çok ölçekli versiyonunu kullanır ve böylece çok ölçekli

bilgileri yakalar. Uzaysal piramit ortaklama ağları çok ölçekli bağlamsal bilgileri kodlayabilir. Bu, gelen özellikleri veya ortaklama işlemlerini birden fazla oranda ve etkili bir görüş alanı ile inceleyerek yapılır. Uzaysal piramit ortaklama ağları genellikle farklı ölçeklerdeki girdileri eğitmek ve özellikleri daha sonraki bir adımda birleştirmek için aynı temel ağın paralel sürümlerini kullanır.

Girdi görüntüsünde bulunan her bilgi bizim için değerli değildir. Değerli olmayan bilgiler gürültü olarak adlandırılır. Bu gürültüyü yok etmek konusunda kodlayıcı-Kod çözücü ağları iyi performans göstermektedir. Girdiyi, tüm girdi bilgilerini temsil etmek için kullanılabilecek yoğun bir forma dönüştürmeyi öğrenirler.



Şekil 4.6 Kodlayıcı-kod çözücü ağlar

Girdinin farklı boyutlardaki temsillerini almak hem çok zaman alan hem de çok fazla donanımsal yapıya ihtiyaç duymasından dolayı DeepLab, evrişim işleminin genelleştirilmiş bir biçimi olan atrous evrişim kavramını tanıttı. Atrous evrişimi, resmin değişik boyutlardaki görüntülerini oran (r) parametresi vererek elde etmemizi sağlar. Normal evrişim, $r = 1$ olan Atrous Evrişimi durumudur. Atrous Uzaysal Piramit Ortaklama (ASPP), Uzaysal Piramit Ortaklama ile Atrous Evrişiminin birlikte kullanılmasıdır. Burada ASPP 4 paralel işlem kullanır: 1×1 evrişim ve 3×3 Atrous evrişimi oranları 6, 12, 18 olmak üzere, evrensel ortalama ortaklaması ile görüntü seviyesi özellikleri ekler ve Bilineer yukarı örnekleme özellikleri doğru boyutlara ölçeklemek için kullanılır.

Derinlemesine evrişim konvolüsyonlar: Derinlemesine evrişimler, standart evrişim işleminden daha az sayıda hesaplama ile evrişim gerçekleştirmek için kullanılan bir

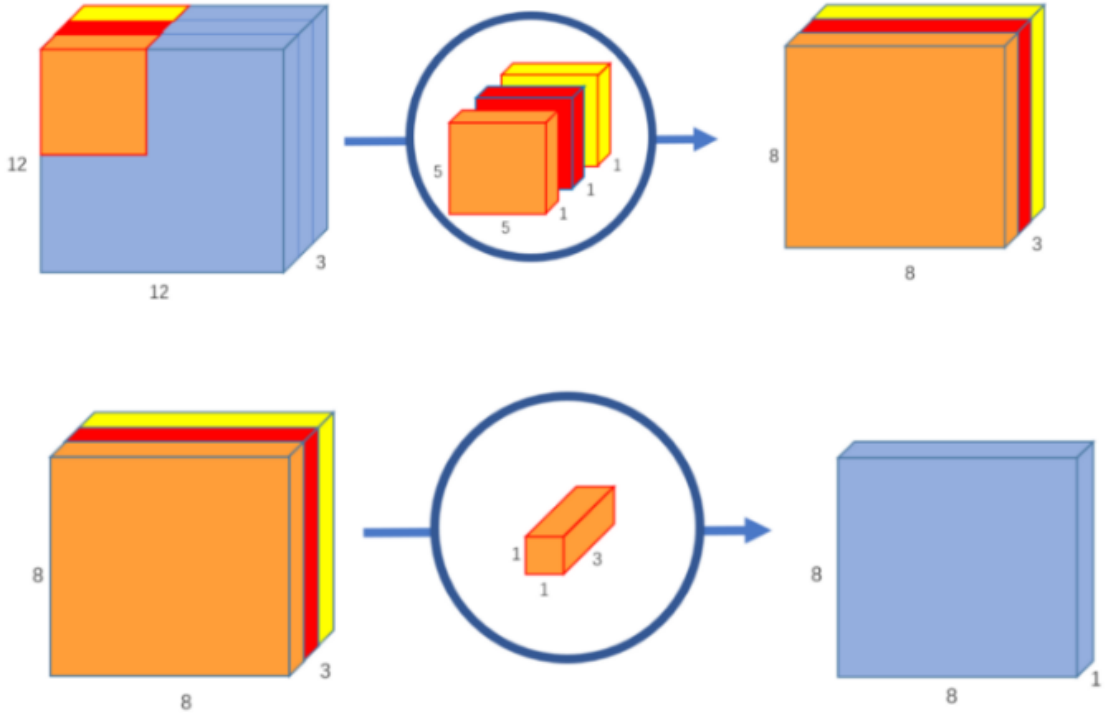
$$y[i] = \sum_k x[i + r \cdot k]w[k]$$

Şekil 4.7 Atrous Evrişimi Formülü

tekniktir. Bu, evrişim işleminin iki aşamaya bölünmesini içerir:

Derinlemesine evrişim: Bu ilk adımda, bize 8 x 8 x 3 boyutunda bir çıktı veren 5 x 5 x 1 şeklinde tek bir çekirdeğe sahip bir evrişim uyguluyoruz..

Noktasal evrişim: Kanal sayısını artırmak istiyoruz. Giriş görüntüsünün derinliğine uyan bir derinliğe sahip 1 x 1 çekirdek kullanacağız (bizim durumumuzda 3). Bu 1 x 1 x 3 kıvrım, 8 x 8 x 1 şeklinde bir çıktı verir. Kanal sayısını artırmak için gerektiği kadar 1 x 1 x 3 kıvrım kullanabiliriz.



Şekil 4.8 Derinlemesine evrişim ve Noktasal evrişim

DeepLab Model Mimarisini Anlamak: DeepLab V3, ana özellik çıkarıcısı olarak ImageNet veri kümesinden oluşturulmuş Resnet-101'i Atrous evrişim ile birlikte kullanır. Değiştirilmiş ResNet modelinde, son ResNet bloğu farklı dilatasyon hızlarına

sahip Atrous evriřimi kullanır. Deęiřtirilmiř ResNet bloęunun stndeki kod zc modl iin Atrous uzaysal piramit ortaklama ve iki izgili yukarı rnekleme kullanır.

DeepLab V3 +, ařaęıdaki zelliklerle birlikte ana zellik ıkarıcı olarak Aligned Xception zellięini kullanır: Tm maksimum ortaklama iřlemlerinin yerini uzun adımlarla derinlemesine ayrılabilir evriřimler alır. Her 3 x 3 derinlemesine evriřimden sonra ekstra parti normalizasyonu ve ReLU aktivasyonu eklenir Giriř akıřı aę yapısını deęiřtirmeden modelin derinlięi artırılır.

Xception Modeli Francois Chollet tarafından nerilmiřtir. Xception, standart Inception modln derinlemesine ayrılabilir konvolsyonlarla deęiřtiren Inception mimarisi'nin bir uzantısıdır. Derinlemesine ayrılabilir konvolsyon'u anlamak iin nce uzaysal ayrılabilir konvolsyon'u deęinmemiz gerekmektedir. Uzaysal ayrılabilir konvolsyon'da normal $n \times n$ 'lik bir kernel yerine $n \times 1$ ve $1 \times n$ olmak zere sırayla 2 farklı iřlemden geirerek, daha az iřlem yapmamıza raęmen yakın performans elde edilmektedir. Derinlemesine ayrılabilir konvolsyonlar'da ise kernel $n \times n \times 1$ boyutunda alınarak elde edilmiř grnty $1 \times 1 \times 3$ 'lk bir kernel iřlemden geirerek, derinlięi kullanmıř ve daha fazla hız elde edilmiř olunur.

4.4 Bağlantılı Bileşen Etiketleme

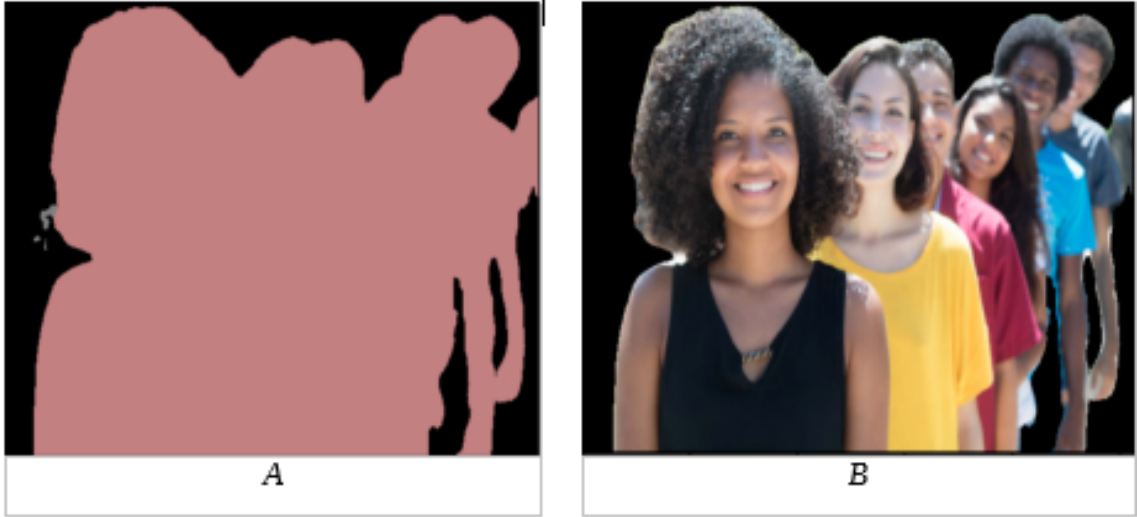
Bağlantılı bileşen etiketleme bölgelere ayrılmış görüntüdeki bir sınıfın birbirine bağlı olan en büyük yüzey alanının hesaplanması için geliştirilen bir yöntemdir. Bağlantılı bileşen etiketleme sadece ikili görüntülerde çalışabilir onun için bağlantılı bileşen etiketleme yönetimini görüntüdeki en büyük nesneyi bulmak için görüntüdeki bölge sayısı kadar görüntü ayarlanıp bu yöntem uygulanır. Bağlantılı bileşen analizinin 8 komşu yön ve 4 komşu yön olmak üzere 2 farklı şekilde yapılabilir. Bu çalışmada 4 komşu yöne gidebilecek şekilde bağlantılı bileşen analizi kullanılmıştır. Bağlantılı bileşen etiketleme kullanan fonksiyonumuzun çalışma adımları 3 bölgeye sahip bir görüntü için aşağıda belirtilmiştir.

1. bir bölge seçilir, seçilen bölgenin değerleri 1 ve diğer tüm bölgeler 0 olarak ayarlanır.
2. Seçilen bölgenin en büyük bağlantılı alanı bulunur.
3. Seçilen bölgenin en büyük bağlantılı alanının ne kadar piksel içerdiği hesaplanır.
4. Aynı işlemler diğer bölgeler için de yapılır.
5. En büyük alan tespit edilir ve o bölgenin gerçek resimdeki pikselleri dizi yapılarak geri döndürülür.

Bu bileşenin örnek girdi ve çıktı görüntüleri aşağıda verilmiştir.



Şekil 4.9 A: Giriş görüntüsüdür. B: Çıktı görüntüsüdür.



Şekil 4.10 A: Giriş görüntüsüdür. B: Çıktı görüntüsüdür.

4.5 Renk Niceleme

Renk niceleme için Scikit-learn kütüphanesini kullandık. K-ortalama algoritmasını kullanarak fotoğrafın genel görüntü kalitesini en az seviyede düşüşerek fotoğraftaki renk sayısını belirlediğimiz k sayısına indirir. Performans kazanımı için sadece en büyük segmetin piksellerine renk niceleme uygulanmıştır. Çoğu standart teknik, renk nicemlemesini, noktaların orijinal görüntüde bulunan renkleri temsil ettiği ve üç eksenin üç renk kanalı olarak temsil edildiği üç boyutlu uzayda bir kümelendirme problemi olarak ele alır. Renk nicemlemesine hemen hemen her üç boyutlu kümeleme algoritması uygulanabilir, bunun tersi de geçerlidir. Renk nicelemesini uygulayabilmek için ana işlem adımları aşağıda belirtilmiştir.

1. İndirgenecek renk sayısı kadar nokta(c) 3 boyutlu uzayda ayrı ayrı ve aynı piksel değeri olmamak şartı ile rasgele olarak piksel değerlerinden birine atanır.
2. Sırası ile her pikselin c noktalarına olan uzaklığı ölçülür. En kısa olan mesefadaki c noktasının veri yapısına bu değer yerleştirilir. 3 boyutlu uzayda 2 nokta arasındaki mesafeyi ölçmek için Öklid mesafesi kullanılır.
3. Ayrı ayrı c noktalarının kendine ait veri yapısındaki piksel değerlerinin ortalaması alınır ve yeni c noktası olarak atanır.
4. C noktaları ideal yerlerine gelene kadar adım 2 ile adım 3 tekrar tekrar uygulanır.

Bu bileşenin örnek girdi ve çıktı görüntüleri aşağıda verilmiştir.



Şekil 4.11 A: Giriş görüntüsüdür. B: Çıktı görüntüsüdür.



Şekil 4.12 A: Giriş görüntüsüdür. B: Çıktı görüntüsüdür.

4.6 Görselleştirme, Histogram

Histogram, gruplandırılmış bir veri dağılımının sütun grafiğiyle gösterimidir. Diğer bir ifadeyle, tekrarlı sayılardan oluşan verilerin grafiğe aktarılması, yani veri gruplarının grafiğinin dikdörtgen sütunlar halinde gösterilmesidir. Elimizde dizi halinde en büyük nesnenin belli bir renge indirgenmiş olarak piksel değerleri olduğundan dolayı bu değerlerin histogramı çıkartılır. Histogram sonucuna göre en popüler olan 4 renk, alanı kendi rengi olacak şekilde yuvarlak grafik ve sütun grafik yapılarak son kullanıcıya gösterilir. Histogram çıkarma işlemi python sözlük(dictionary) veri yapısı kullanarak yapılmıştır. Pikseller anahtar(key) değeri olarak kullanılmıştır. Anahtara karşılık gelen değerler(value) tekrar sayısıdır.

4.7 Proje Kullanımı

Proje Jupiter notebook üzerinde çalışan, kullanıcı arayüzü olmayan bir programdır. Projede kullanıcı rolleri yoktur. Tek bir kullanıcı vardır. Program girdi olarak 2 tane komut parametresi alır. Birinci parametre opsiyoneldir program içindeki k sayısını değiştirmeye yarar. 2. parametre ise zorunludur ve programın girdisi olacak resmin dosya sistem üzerindeki göreceli veya gerçek yeri verilmelidir.

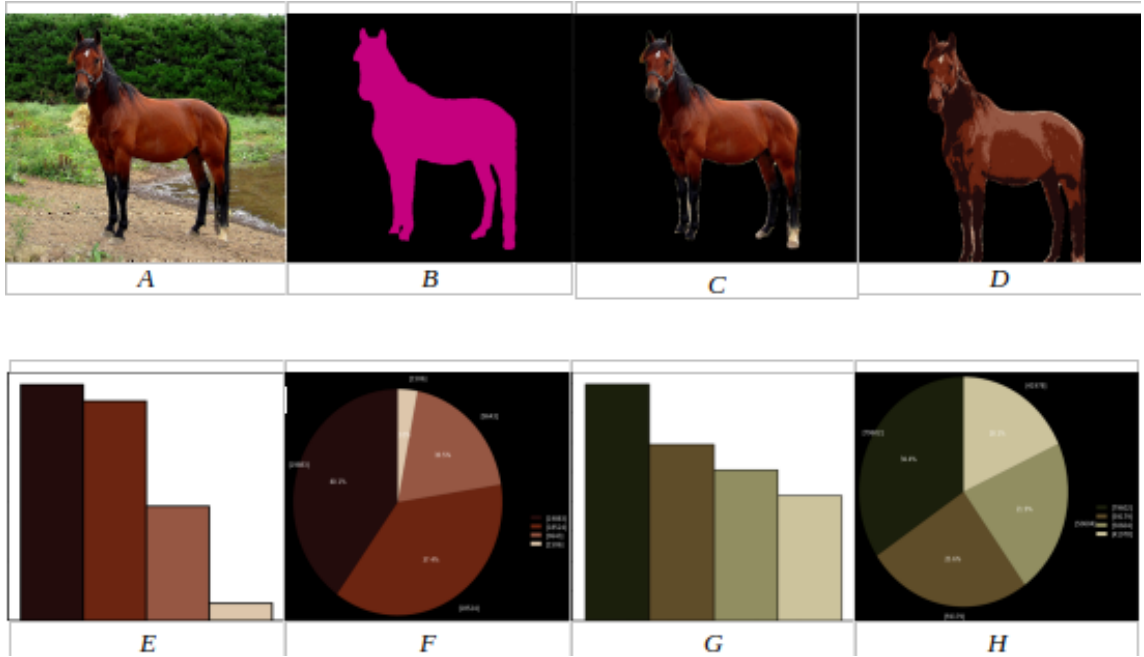
5 Deneysel Sonular

Bu blmde projenin alıřan hali, 15 tane nceden seilmiř grnt kullanılarak test edilecektir. Bu grntler eřit sayıda grnt iericek řekilde 3 ana kategoriye ayrılmıřtır. Birinci kategoride en popler rengi belli, tek nesne ve arka plandan oluřan grntler, ikinci kategoride ok renkli, tek nesne ve arka plandan oluřan grntler, nc kategoride ok nesneli ve arka plandan oluřan grntler vardır. Projenin son ıktısı ve ara ıktılar her bir rnek grnt iin ayrı ayrı eklenmiřtir. Okuyucuların karřılařtırma yapabilmeleri iin tm resmin renk histogramı, deęerlendirme iin ekran grntlerine eklenmiřtir. Girdi olarak verilen grntnn tm grselleřtirilmiř ıktılarını inceleyenler daha kolay karřılařtırma yapabilsenler diye tek bir ıktı grnts olarak birleřtirilmiřtir. ıktı grnts 8 tane grntnn birleřiminde oluřur. Her grntnn bařlıęı vardır ve bu grntnn bařlıęına gre ne anlama geldięi ařaęıdaki listede aıklanmıřtır.

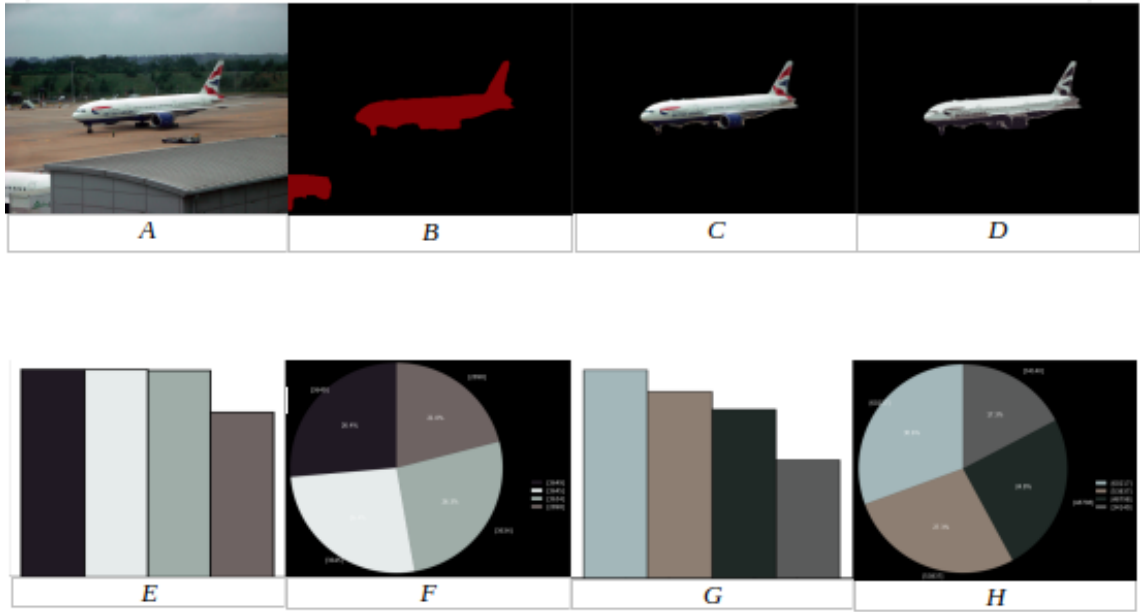
- **A:** Girilen grntdr.
- **B:** Anlamsam btnleme sonucu oluřan grntdr.
- **C:** Baęlı bileřen etiketleme sonucu en byk nesnenin grntsdr.
- **D:** Renk niceleme kullanılarak en byk nesnenin 4 renge indirgenmiř grntsdr.
- **E:** En byk nesnenin renk stn grafięidir.
- **F:** En byk nesnenin renk pasta grafięidir.
- **G:** Btn grntnn renk stn grafięidir.
- **H:** Btn grntnn renk pasta grafięidir.

Projenin ıktılarının doęruluk oranı 20 kiřinin zerinde anket yoluyla test edilmiřtir. Ankete katılanlara, projenin ařaęıda olan ıktı grnts formatında grntler

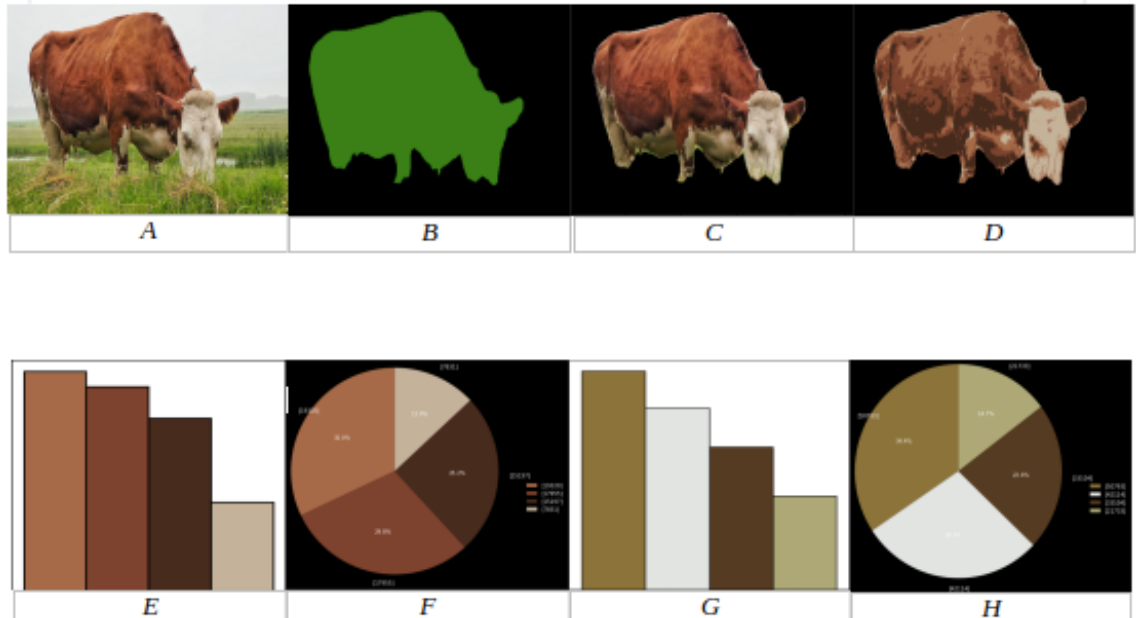
gösterilmiştir ve doğruluk oranına 1 ile 5 arasında puan verilmesi istenmiştir. İlk kategorideki görüntülerde projemizin ankette aldığı ortalama doğruluk değeri 4.6'dır. İlk kategorideki görüntüler tek renk ve tek nesneden oluştuğu için projemizin katılımcılara göre başarısı yüksektir. İkinci kategorideki görüntülerde projenin ankette aldığı ortalama doğruluk değeri 4.4'dir. İkinci kategorideki görüntüler birden fazla renk ve tek nesneden oluştuğu için projemizin katılımcılara göre başarısı ilk kategoriye göre düşmüştür. Bunu nedeni renk niceleme algoritmamız en büyük nesnedeki renkleri 4 renge indirgemeye çalıştığı için eğer en büyük nesne fazla renge sahipse renklerde bozulmalar meydana gelmektedir. Üçüncü kategorideki görüntülerde projemizin ankette aldığı ortalama doğruluk değeri 4.25'dir. projede anlamsal bütünleme tekniği ve bağlantılı bileşen analizi tekniği kullanmamız sebebi ile eğer görüntüde aynı türden farklı iki nesne bağlantılı bir şekilde duruyorsa bu iki nesneyi projemiz tek nesne olarak algılayarak istenilen performansın anlatında kalmaktadır. Bu duruma çözüm olarak örnek bölütleme(Instance segmentation) tekniği anlamsal bölütleme yerine kullanılabilir.



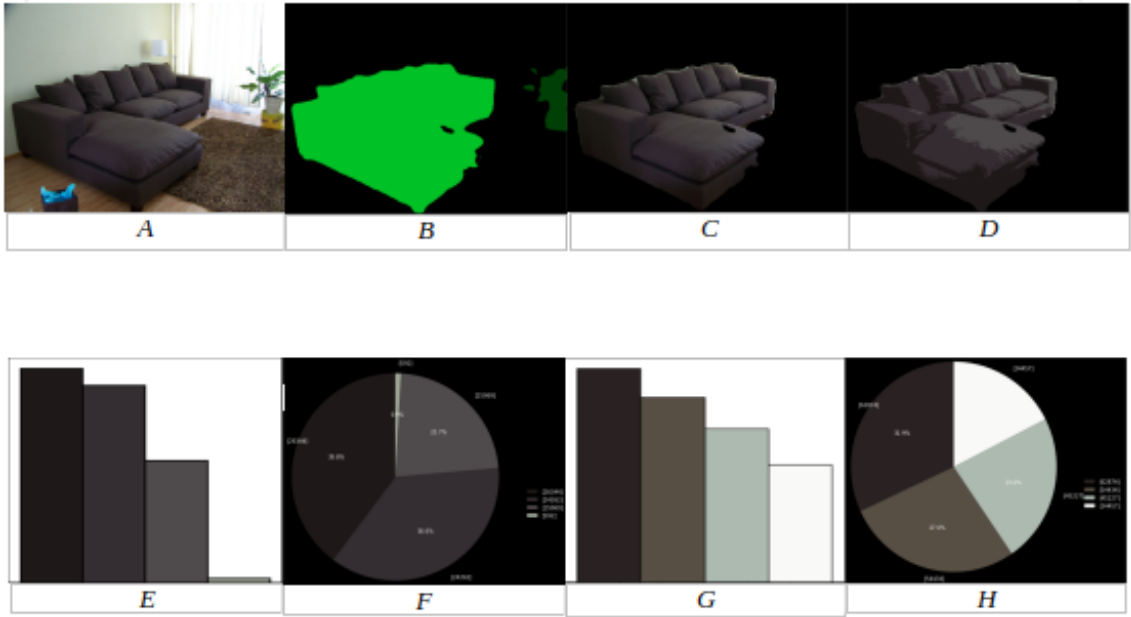
Şekil 5.1 Projenin at görseli üzerindeki çıktısı



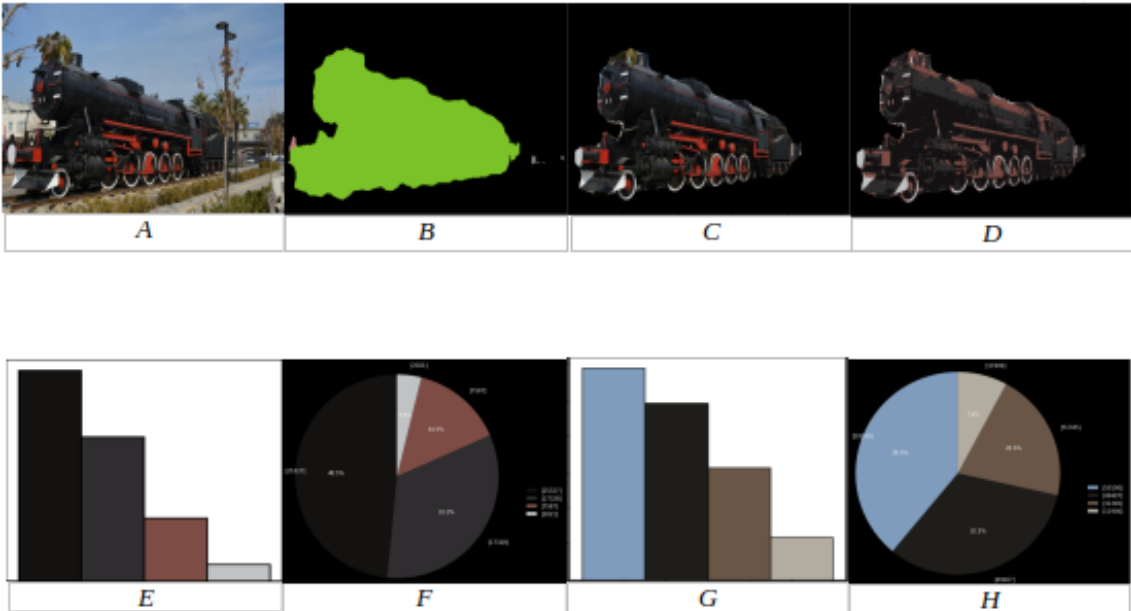
Şekil 5.2 Projenin uçak görseli üzerindeki çıktısı



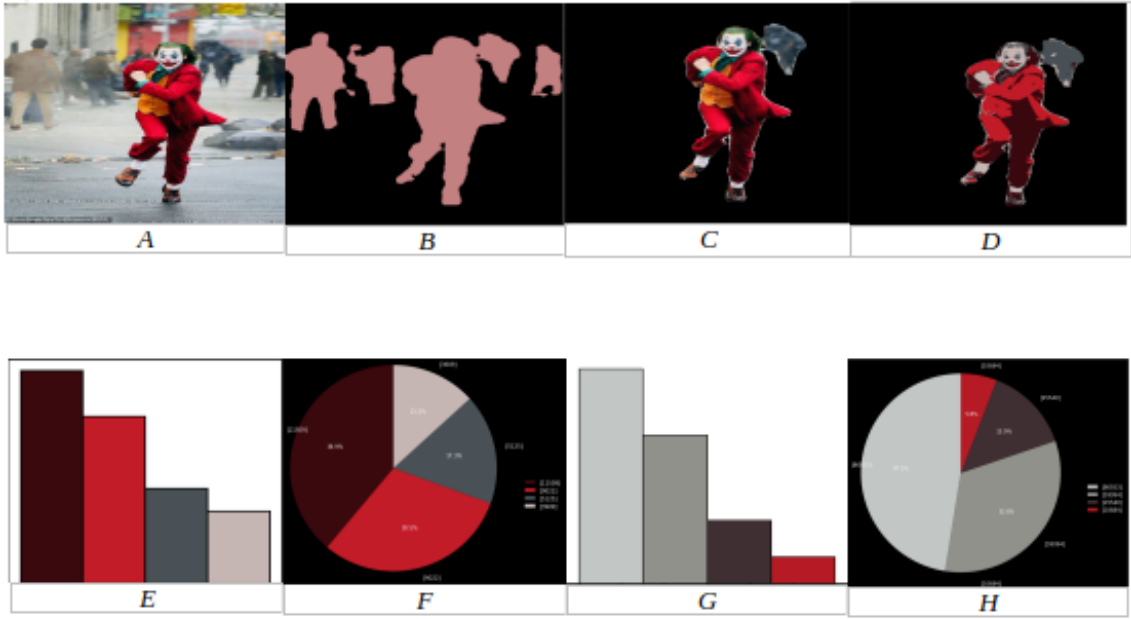
Şekil 5.3 Projenin inek görseli üzerindeki çıktısı



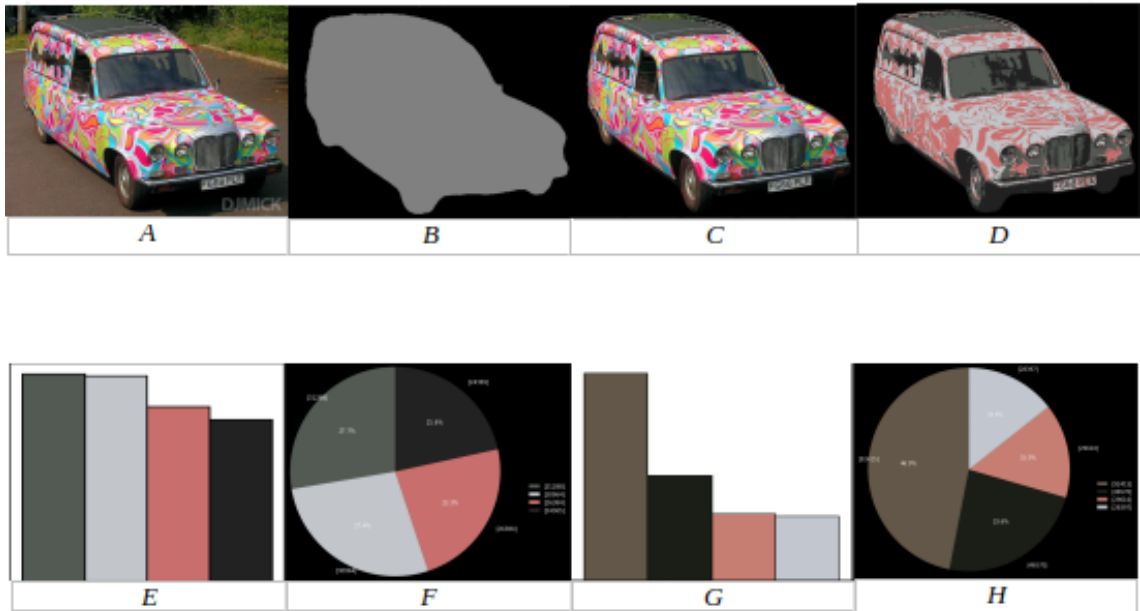
Şekil 5.4 Projenin koltuk takımı görseli üzerindeki çıktısı



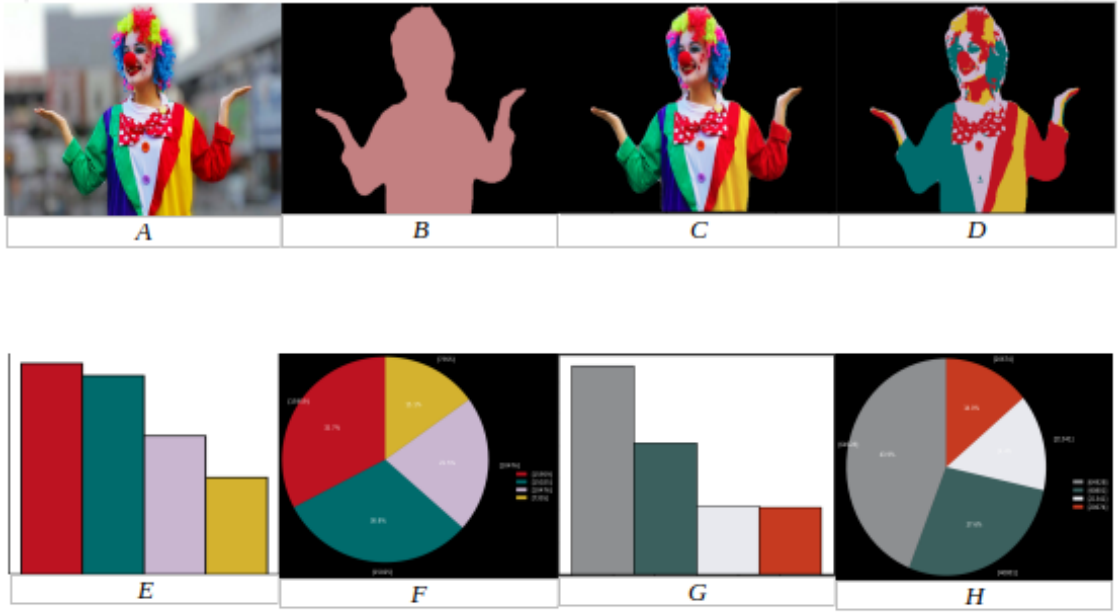
Şekil 5.5 Projenin tren görseli üzerindeki çıktısı



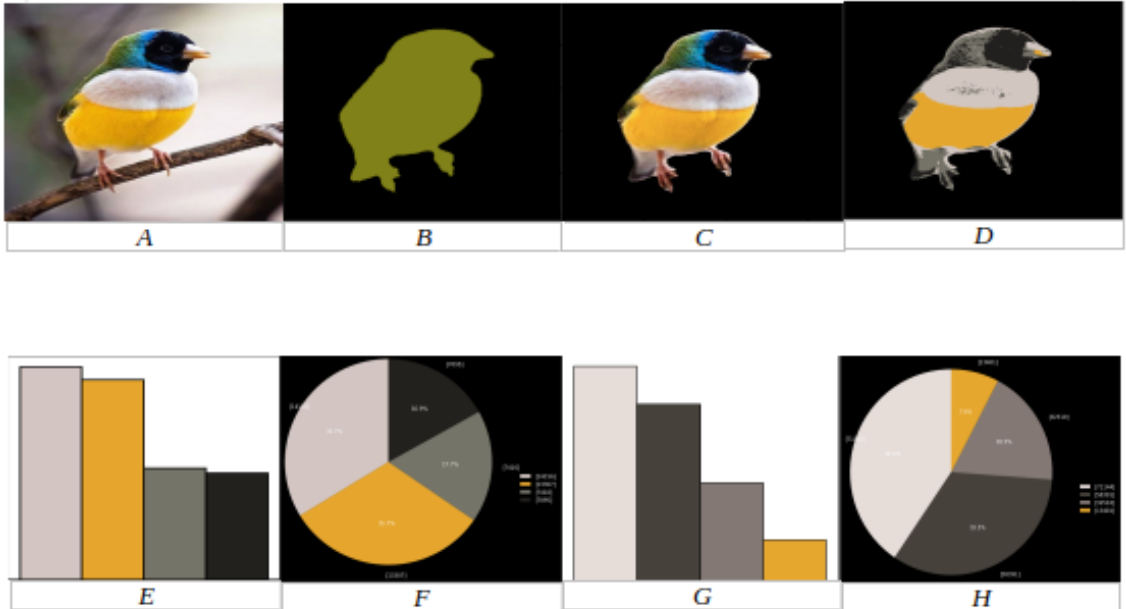
Şekil 5.6 Projenin insanların olduğu görsel üzerindeki çıktısı



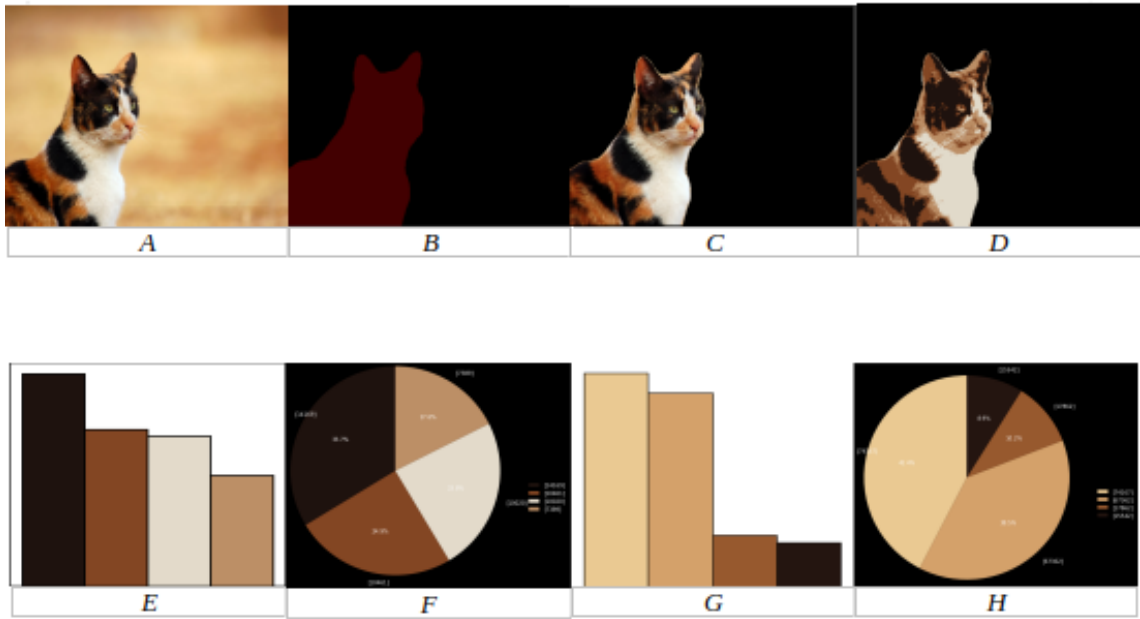
Şekil 5.7 Projenin araba görseli üzerindeki çıktısı



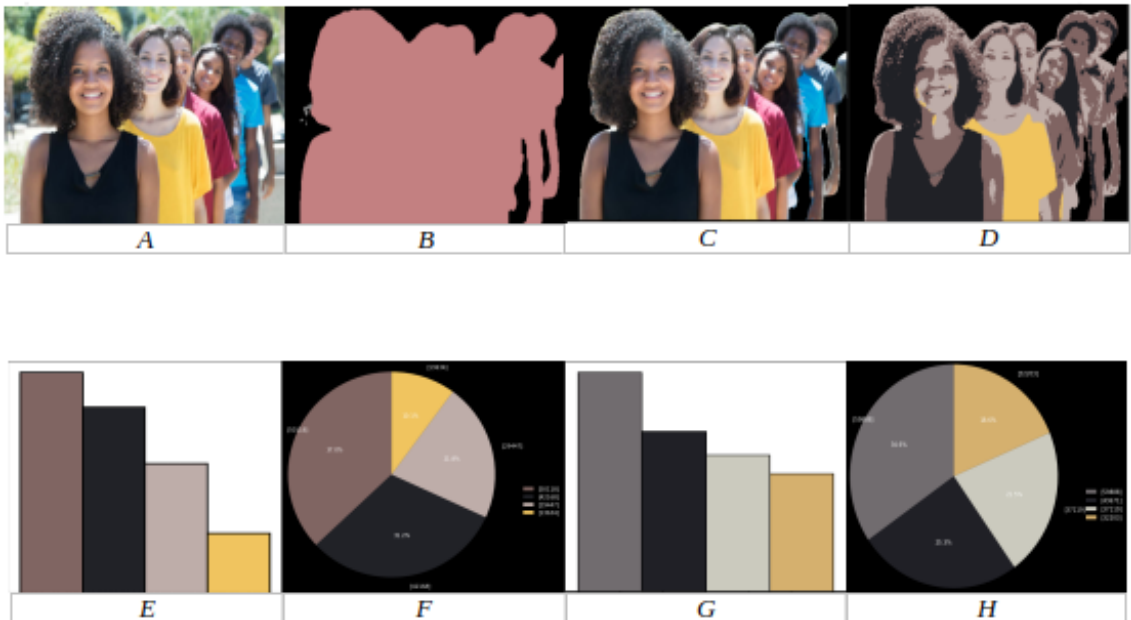
Şekil 5.8 Projenin palyaço görseli üzerindeki çıktısı



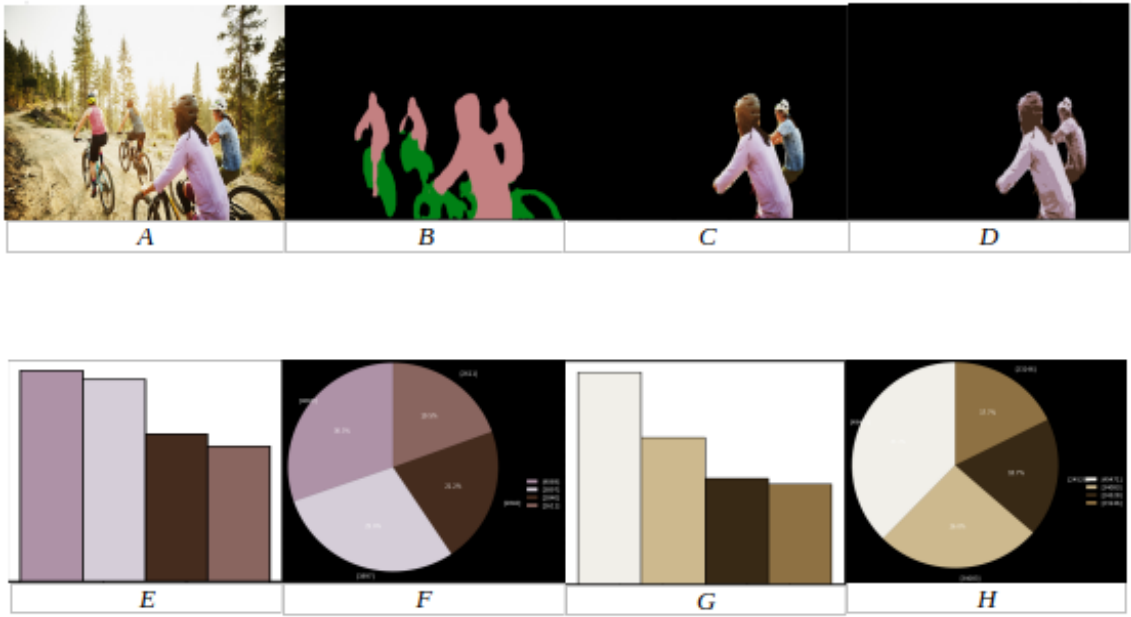
Şekil 5.9 Projenin kuş görseli üzerindeki çıktısı



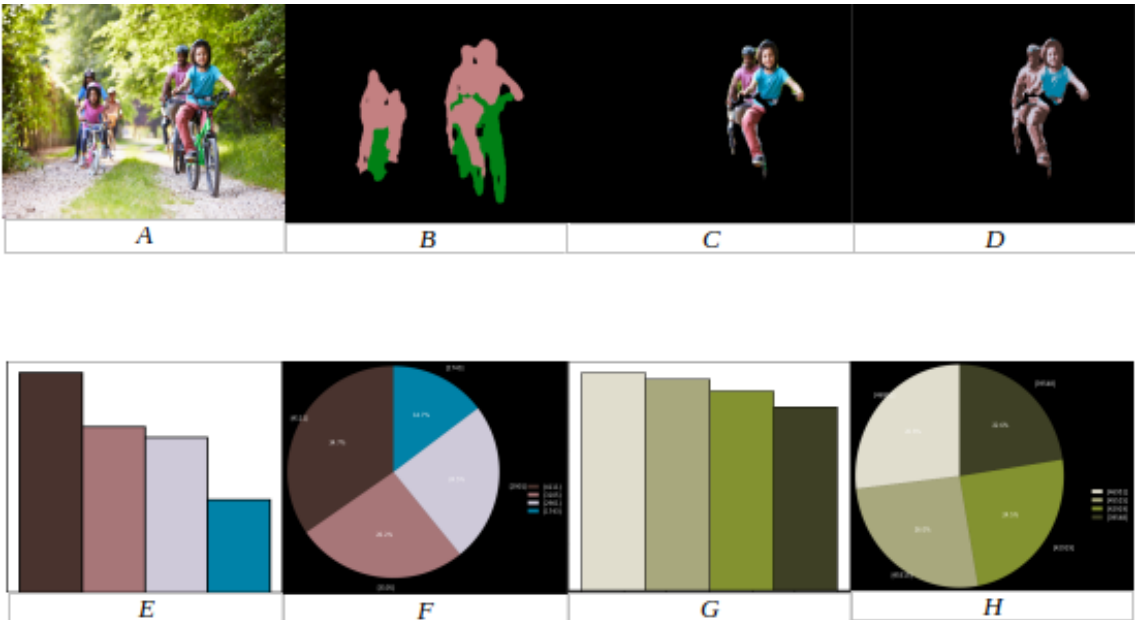
Şekil 5.10 Projenin kedi görseli üzerindeki çıktısı



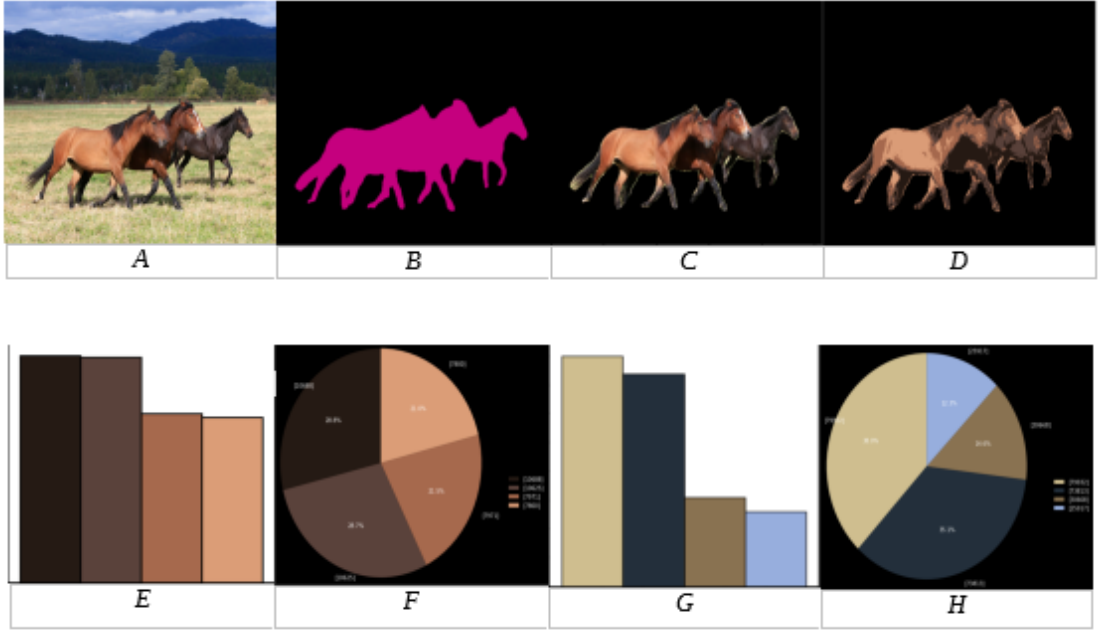
Şekil 5.11 Projenin insanların olduğu görsel üzerindeki çıktısı



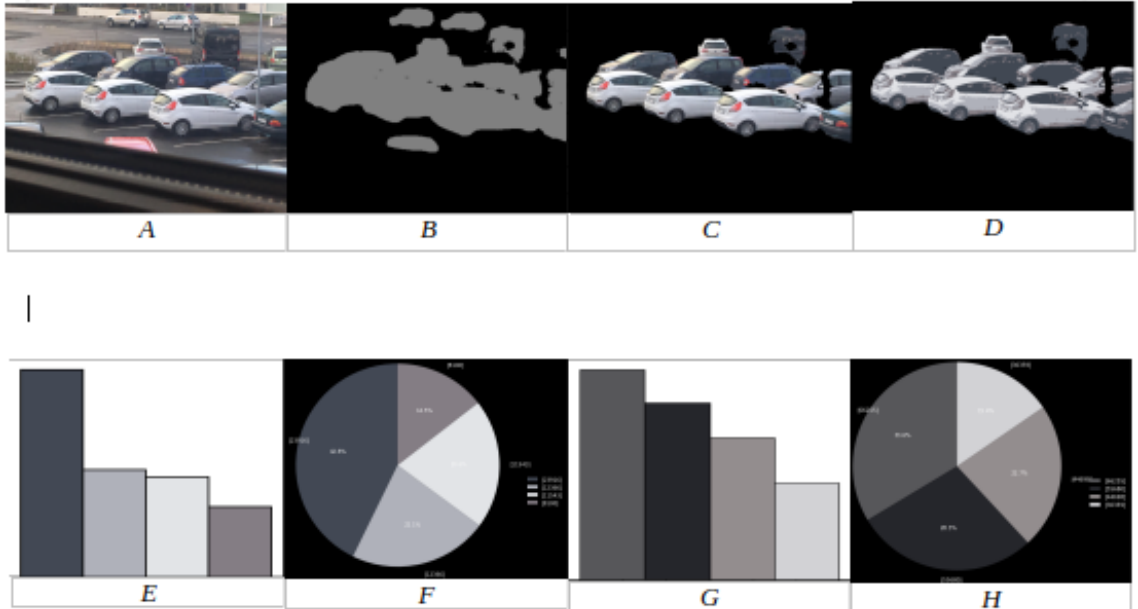
Şekil 5.12 Projenin insanların olduğu görsel üzerindeki çıktısı



Şekil 5.13 Projenin insanların olduğu görsel üzerindeki çıktısı



Şekil 5.14 Projenin atların olduğu görsel üzerindeki çıktısı



Şekil 5.15 Projenin arabaların olduğu görsel üzerindeki çıktısı

6

Test, Performans Analizi ve İyileştirme

6.1 Başarım Analizi

Tablo 6.1 Anket sonuçları

	S-1	S-2	S-3	S-4	S-5	S-6	S-7	S-8	S-9	S-10	S-11	S-12	S-13	S-14	S-15	S-16	S-17	S-18	S-19	S-20	Tüm -S
Çok kötü (1)	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
Kötü (2)	0	1	2	1	0	0	2	0	1	0	0	0	0	1	0	0	1	0	0	0	
Ortalama (3)	4	3	3	3	2	1	4	3	3	0	3	5	0	2	1	1	3	3	1	1	
İyi (4)	6	6	7	1	7	6	5	3	4	1	7	6	5	4	3	7	6	6	6	5	
Çok iyi (5)	10	11	8	15	11	13	9	14	12	19	9	9	15	13	16	12	10	11	13	14	
Ortalama puan	4.3 ~ %86.0	4.35 ~ %87.0	4.05 ~ %81.0	4.5 ~ %90.0	4.45 ~ %89.0	4.6 ~ %91.9	4.05 ~ %81.0	4.55 ~ %91.0	4.35 ~ %86.9	4.95 ~ %99.0	4.15 ~ %83.0	4.2 ~ %84.0	4.75 ~ %95.0	4.45 ~ %89.0	4.75 ~ %95.0	4.55 ~ %91.0	4.25 ~ %85.0	4.45 ~ %89.0	4.44 ~ %88.0	4.65 ~ %93.00	4.43 ~ %88.79

Performans analizi için 20 farklı kişiye 20 farklı görüntü ve bu görüntüye ait en popüler renkler verilmiştir. Kullanıcılara verilen renklerin resmin gerçek en popüler renklerini ne kadar iyi yansıttığı sorulmuştur. Değerlendirme kriterleri olarak Çok kötü (1) , Az kötü(2) , Ortalama(3) , İyi(4) , Çok iyi(5) verilmiştir.

Sorulara verilen cevaplardan ortalamalar alınmıştır. Sistemin tüm sorulardaki ortalama başarıları %88.79'dur.

6.2 Hız Analizi

6.2.1 Eğitim Zamanı Hız Analizi

Projede anlamsal bölütleme için eğitilen DeeplabV3 modelinin , Pascal VOC 2012 için eğitim süresi yaklaşık 9.5 saattir. Bahsedilen eğitim süresi Tesla V100 ekran kartı üzerinde ölçülmüştür. Daha iyi donanımlarla bu süre kısaltılabilir.

6.2.2 Test Zamanı Hız Analizi

Tablo 6.2 Bileşenlerin Saniye Cinsinde Aldığı Süreler

	Anlamsal Bölütleme	Bağlantılı bileşen etiketleme	Renk Niceleme	Toplam
Zaman (saniye)	8.96	0.04	0.44	9.45

6.2’de de görüldüğü üzere anlamsal bölütleme yaklaşık 9 saniye sürerken diğer bileşenler toplam yarım saniye sürüyor. Bu ölçümler, girdi/çıkıtlı işlemlerinden bağımsız olarak, Intel(R) Xeon(R) CPU @ 2.30GHz işlemcisinde, 12 GB bellekle beraber çalıştırıldığında çıkan sonuçlardır. Daha iyi donanımlarda daha iyi sonuçların elde edilmesi beklenmektedir.

7 Sonuç

Bizim bu çalışmadaki hedefimiz bir görüntüdeki en büyük nesnenin en popüler rengini tespit etmektir. Görüntü ilk olarak derin öğrenme destekli anlamsal bölütleme yöntemi kullanarak bölümlere ayrılır. Bölümlere ayrılmış görüntüdeki en büyük neyneyi bulmak için bağlı bileşen etiketleme yöntemi kullanılır. Ardından sadece en büyük nesnenin piksel değerleri renk niceleme yöntemi ile belirli sayıda renge indirgenir. Son olarak elde ettiğimiz sonuçlar görselleştirilerek kullanıcıya sunulur.

Modelimiz anlamsal bölütleme kullandığımız için aynı türden bağlı nesneler görüntüde var ise bunları tek nesne olarak kabul eder. Deneysel sonuçlar kategori 3 kısımdaki görüntülerde görebileceğiniz gibi bu durumda modelimizin başarı puanı düşüyor. Bunun önüne geçmek için anlamsal bölütleme yerine, örnek bölütleme yöntemi kullanılarak aynı türden nesneler de farklı bölgelere ayrılabilir. İkinci geliştirme ise k-ortalama algoritması eğer görüntü çok renkten oluşuyor ve az renge nicelenmeye çalışılıyorsa, görüntünün renk kalitesini bozabilmektedir. Renk kalitesini koruyarak veya k sayısını kendi seçen bir algoritma ile renk nicelemesi yaparken renk kalitesi korunmaya çalışılabilir.

Modelimiz günümüzde birçok alanda kullanılabilir. Örneğin, bir okuldaki kameralarda öğrencilerin kıyafetlerinde en çok hangi renkleri tercih ettiğinin tespitinde, moda defilelerinde yılın popüler renklerinin tespit etmede, bir ülkede veya şehirde en çok tercih edilen araba renklerinin tespitinde kullanılabilir.

- [1] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *International Journal on Computer Vision*, 2018. [Online]. Available: <https://github.com/CSAILVision/semantic-segmentation-pytorch>.
- [2] D. Gupta. (2019). Implementation of segnet, fcn, unet , pspnet and other models in keras., [Online]. Available: <https://github.com/divamgupta/image-segmentation-keras>.
- [3] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018. [Online]. Available: <https://github.com/tensorflow/models/tree/master/research/deeplab>.
- [4] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. G. Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *CoRR*, vol. abs/1704.06857, 2017. arXiv: 1704 . 06857. [Online]. Available: <http://arxiv.org/abs/1704.06857>.
- [5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *CoRR*, vol. abs/1606.00915, 2016. arXiv: 1606 . 00915. [Online]. Available: <http://arxiv.org/abs/1606.00915>.
- [6] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao, "The connected-component labeling problem: A review of state-of-the-art algorithms," *Pattern Recognition*, vol. 70, pp. 25–43, 2017, ISSN: 0031-3203. DOI: <https://doi.org/10.1016/j.patcog.2017.04.018>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0031320317301693>.

BİRİNCİ ÜYE

İsim-Soyisim: Tarık ÇARLI
Doğum Tarihi ve Yeri: 16.02.1998, Kocaeli
E-mail: 11116049@std.yildiz.edu.tr
Telefon: 0530 792 13 71
Staj Tecrübeleri: TUBITAK BILGEM BTE Ofisi

İKİNCİ ÜYE

İsim-Soyisim: Yusuf ANI
Doğum Tarihi ve Yeri: 05.10.1998, Bursa
E-mail: 11116033@std.yildiz.edu.tr
Telefon: 0531 409 69 30
Staj Tecrübeleri: TURKGEN Yazılım ofisi

Proje Sistem Bilgileri

Sistem ve Yazılım: Ubuntu İşletim Sistemi, Python
Gerekli RAM: 4GB
Gerekli Disk: 30GB