



**T.C.
TEKİRDAĞ NAMIK KEMAL ÜNİVERSİTESİ
ÇORLU MÜHENDİSLİK FAKÜLTESİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ BÖLÜMÜ**

NESNE TANIMA

YAPAY GÖRME VE ÖRÜNTÜ TANIMAYA GİRİŞ FİNAL ÖDEVİ RAPORU

**Hazırlayan
1150606019
Cem Kazım GENEL**

**Danışman
Doç. Dr. Rafet AKDENİZ**

**31/12/2020
Namık Kemal Üniversitesi Çorlu Mühendislik Fakültesi**

NESNE TANIMA

**Tekirdağ Namık Kemal Üniversitesi
Çorlu Mühendislik Fakültesi
Elektronik ve Haberleşme Mühendisliği Bölümü
Final Ödevi**

Final Ödevinin Yapıldığı Tarih: 31/12/2020

**Cem Kazım GENEL
1150606019**

Danışman: Doç. Dr. Rafet AKDENİZ

**31/12/2020
Namık Kemal Üniversitesi Çorlu Mühendislik Fakültesi**

FİNAL ÖDEVİ BEYANNAMESİ

“Nesne Tanıma” başlıklı bu final ödevine ait uygulama kodlarının hazırlanması, yürütülmesi, araştırmaların yapılması ve geliştirilmesi aşamalarında danışmanım Doç. Dr. Rafet AKDENİZ kontrolünde bilimsel etik ve akademik kurallara özen göstererek çalıştığımı; başka kaynaklardan aldığım verileri, bulguları ve materyalleri bilimsel etiğe uygun olarak metinde ve kaynakçada eksiksiz olarak gösterdiğimi ve aksi durumları kabul ettiğimi, beyan ederim.

Tarih: 31/12/2020

Öğrenci Adı Soyadı: Cem Kazım GENEL

İmza: cemkazimgenel

TEŞEKKÜR

Final ödevimi hazırlarken değerli desteklerini esirgemeyen danışman öğretmenim sayın Rafet AKDENİZ'e teşekkürlerimi saygılarımla taraflarına arz ederim.

31/12/2020

Cem Kazım GENEL

İÇİNDEKİLER

FİNAL ÖDEVİ BEYANNAMESİ	3
TEŞEKKÜR	4
İÇİNDEKİLER	5
KISALTMA LİSTESİ	7
ÖZET	8
MATLAB	1
Matlab Nedir?	1
GÖRÜNTÜ İŞLEME	2
Görüntü İşleme Nedir?	2
Görüntü İşlemenin Kullanım Alanları	3
DERİN ÖĞRENME	6
Derin Öğrenme Nedir?	6
Derin Öğrenme Kullanım Alanları	7
EVİRİŞİMSEL SİNİR AĞLARI (CNN)	10
Evrişimsel Sinir Ağları Nedir?	10
Evrişimsel Sinir Ağları Nasıl Çalışır?	11
ALEXNET	13
AlexNet Nedir?	13
AlexNet Kurulumu	13
NESNE TANIMA	15
Uygulamanın Giriş Ekranı	15
Axes Objesi	15
Resim Seç Butonu	15
Histogram Çizdir Butonu	15
Çıkış Butonu	15
Bu butona tıklanarak uygulamadan çıkış yapılır.	15
KULLANIM SENARYOSU	16

Ana Ekran	16
Resim Seçme Ekranı	17
Nesneyi Tanımlama Ekranı	18
Histogram Grafiği Ekranı	19
ÖDEVİN ÖNEMLİ KODLARI	20
OpeningFcn	20
Load Image	21
Detect Object	22
Histogrammed Image	23
PROJEDE KULLANILAN ALGORİTMALAR	24
AlexNet	24
imShow	24
imHist	24
KAYNAKLAR	25
ÖZGEÇMİŞ	26

KISALTMA LİSTESİ

UIGetFile : User Interface Get File (Kullanıcı Arayüzü ile Dosya Alma)

ÖZET

Final Ödevi

NESNE TANIMA

Cem Kazım GENEL

Danışman: Doç. Dr. Rafet AKDENİZ

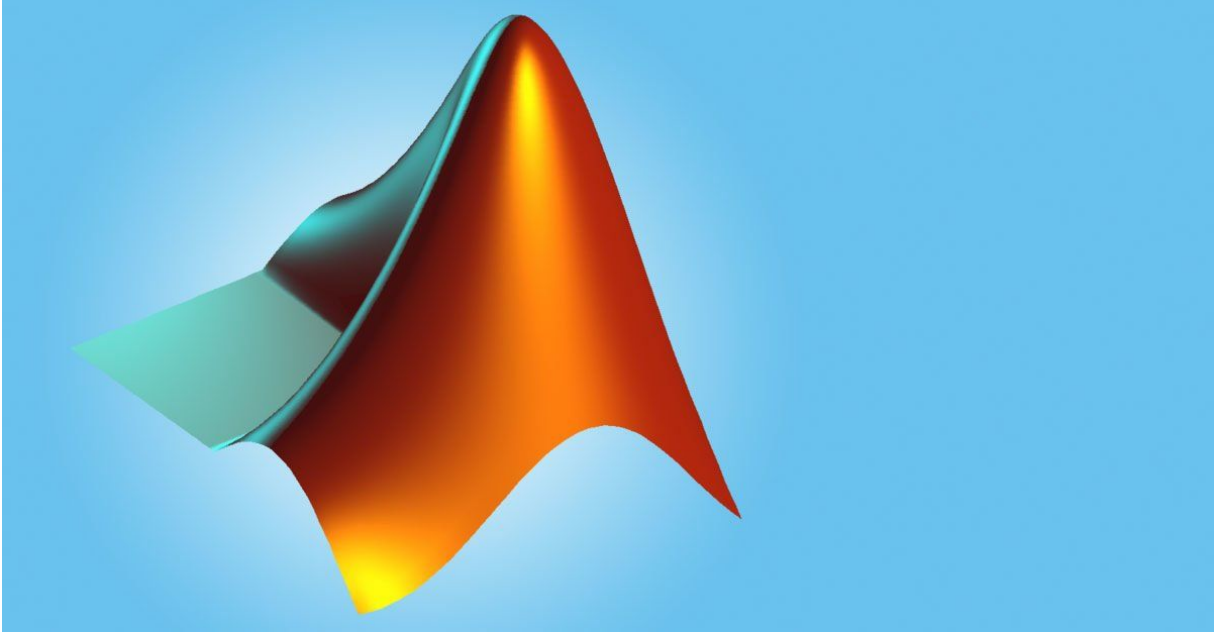
Bu uygulama, MATLAB ile yazılmış olup kullanıcının girdi olarak verdiği resimlerdeki nesneleri tanıyan bir görüntü işleme uygulamasıdır.

Anahtar Kelimeler: MATLAB, Nesne, Tanıma, Görüntü, İşleme

1. MATLAB

1.1. Matlab Nedir?

Matrix Laboratory kelimelerinin kısaltması olan **MATLAB**, sayısal hesaplama ve görselleştirme için kullanılan yüksek seviyeli bir programlama dilidir. Kontrol, görüntü işleme, istatistik optimizasyon, bulanık kontrol, sinir ağları, sayısal işaret işleme, güç sistemleri, filtre dizaynı, genetik algoritma, grafik, veri tabanı, web sunucusu, finans vb. gibi saymakla bitiremediğimiz bir çok alanda güvenli bir şekilde kullanılabilecek araçlar (toolbox) içerir.



Şekil.1.1.0.MATLAB

2. GÖRÜNTÜ İŞLEME

2.1. Görüntü İşleme Nedir?

Görüntü İşleme, görüntüyü dijital form haline getirmek ve bazı işlemleri gerçekleştirmek için geliştirilmiş, spesifik görüntü elde etmek veya ondan bazı yararlı bilgiler çıkarmak için kullanılan bir yöntemdir. Bu yöntemin girdisi video kesiti veya fotoğraf gibi bir görüntüdür. Çıktısı ise görüntünün istenilen ya da dikkat edilmesi gereken bölümüne karşılık gelir. Genellikle Görüntü İşleme sistemi, önceden belirlenmiş sinyal işleme (Signal Processing) yöntemlerini uygularken görüntüleri iki boyutlu sinyaller olarak ele alır.



Şekil.2.1.0.Görüntü İşleme

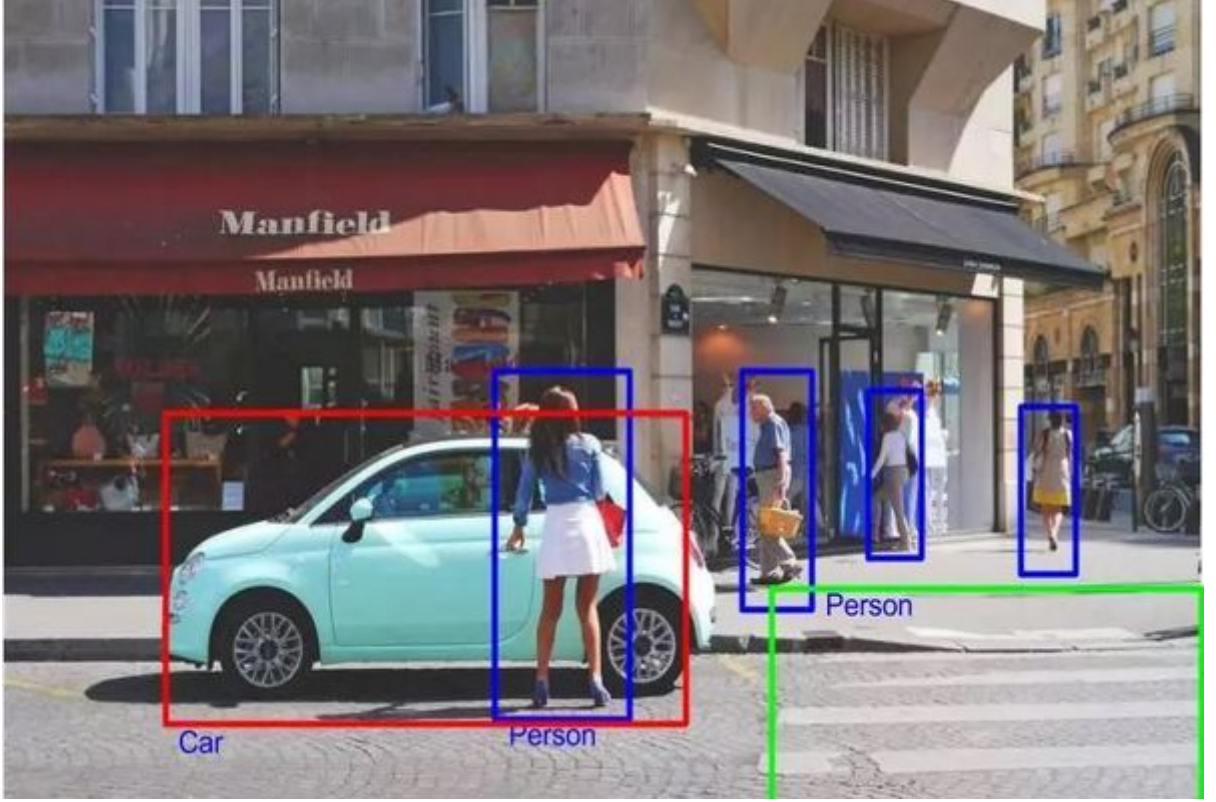
2.2. Görüntü İşlemenin Kullanım Alanları

Görüntü İyileştirme: Alınan görüntülerde gürültü olarak adlandırılan ve görüntü üzerinde bozulmalara neden olan bazı istenmeyen yapılar bulunabilir. Bu gürültülere örnek olarak tuz-biber gürültüsü, gauss gürültüsü, shot gürültüsü verilebilir. Görüntü işleme teknikleri içerisinde bulunan mean (ortalama) filtre, medium (ortanca) filtre gibi tekniklerle görüntü daha kaliteli ve gürültüsüz hale getirilebilir. Bu sayede görüntü üzerinde daha doğru sonuçlar elde edilecektir.



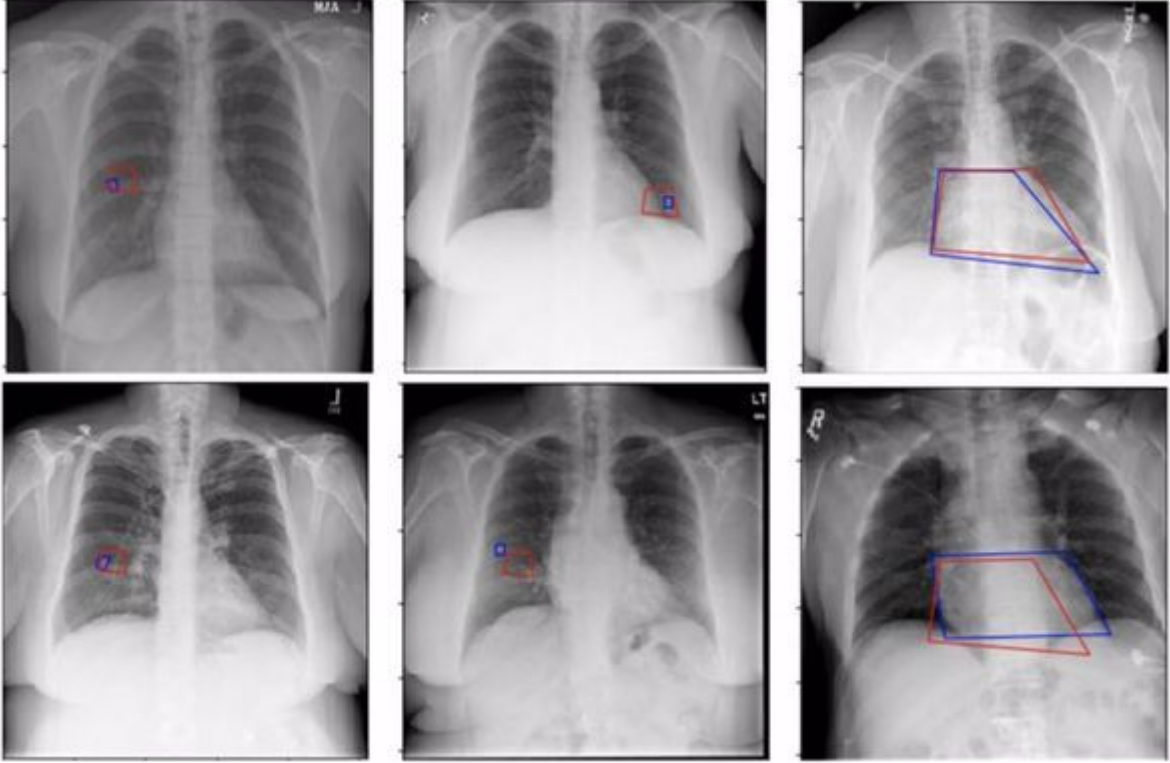
Şekil.2.2.0.Görüntü İyileştirme

Nesne Tanıma: Tespit edilecek cisme göre gerekli yöntemler ve algoritmalar kullanılarak görüntü üzerinden herhangi bir cismin tespiti ve takibi gerçekleştirilebilir. Örneğin yurt dışında bir çok ülkede suçluların tespiti bu yöntem ile gerçekleştirilmektedir. Mevcut olan kamera düzeneklerinden alınan görüntüler üzerinden herhangi bir insanın tespiti sağlanabilir. Bunun dışında trafik alanında da kullanımı mevcuttur. Trafik içerisinde bulunan araçları sayabilir ve araçların hızı ölçülebilir. Bu sayede trafik yoğunluğu olma veya aşırı hız yapma gibi durumların tespiti gerçekleştirilip merkeze gerekli bildirimler yapılabilir.



Şekil.2.2.1.Nesne Tanıma

Sağlık Sektörü: Görüntü işleme teknikleri sayesinde bir çok hastalığın teşhisi gerçekleştirilebilmektedir. Doğum öncesi fetüsün oluşumu ve takibi, tıbbi görüntülerin incelenmesi ,şüpheli dokuların belirgin hale getirilip uzmanlara doğru tanı koyabilme olanağı tanınması, meme kanserinin erken teşhisi gibi alanlarda görüntü işleme teknikleri kullanılmaktadır. Bunların yanı sıra beyin görüntüleme, kemik şeklinin ve yapısının analizi, kanser tanısı koyma ve tümörü fark etme gibi işlemlerde tıp biliminde kullanılabilmektedir.



Şekil.2.2.2.Sağlık Sektörü

Savunma Sanayi: İnsansız hava araçları, görüntü ile hedef takibi yapan roketler gibi araçların bünyesinde bulunan donanımlar, görüntü işleme sonucu elde edilen veriler doğrultusunda hareket gerçekleştirir.

3. DERİN ÖĞRENME

3.1. Derin Öğrenme Nedir?

Derin öğrenme, verilen bir veri seti ile sonuçları tahmin eden birden fazla katmandan oluşan bir makine öğrenme yöntemidir. Derin öğrenme, makine öğrenmesi ve yapay zekâ birbirinden farklı anlamları olan terimlerdir. Derin öğrenme, makine öğrenmesinin; makine öğrenmesi ise yapay zekânın alt dalı olarak özetlenebilir.

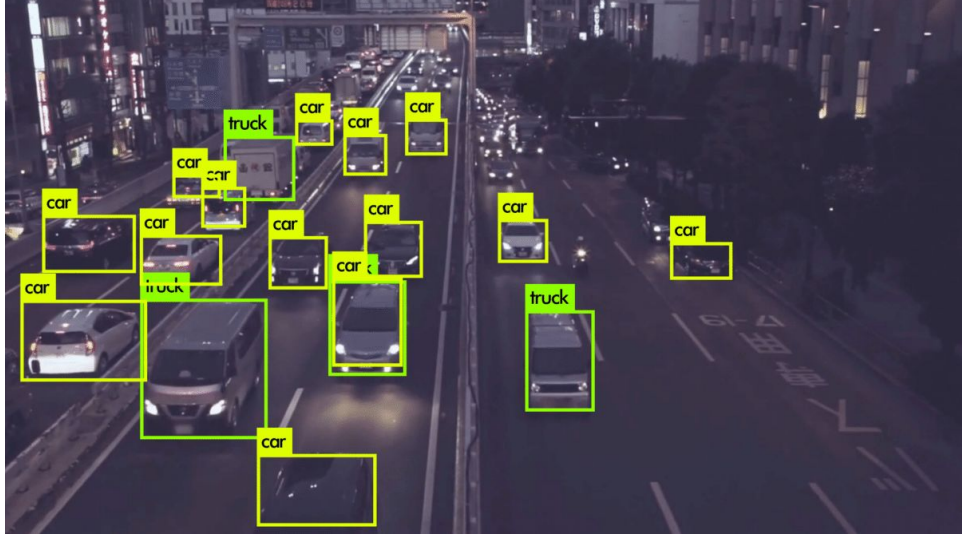


Şekil.3.1.0.Derin Öğrenme

3.2. Derin Öğrenme Kullanım Alanları

Derin öğrenme algoritmasının kullanım alanları aşağıda maddeler halinde belirtilmiştir.

1. **Nesne Tanıma:** Fotoğraf ve görüntülerde bulunan nesnelerin tanınması durumuna denir.



Şekil.3.1.1.Nesne Tanıma

2. **Ses Tanıma:** Konuşmacı tarafından söylenen sözleri tanımlamak için işitsel sinyallerin analizi işlemidir.



Şekil.3.1.2.Ses Tanıma

3. **Sürücüsüz (Otonom) Araç Kullanma:** Otonom araçlar derin öğrenme algoritmalarıyla yazılmış sistemler sayesinde bir sürücüye ihtiyaç duymadan yolu, trafik akışını ve çevresini algılayarak sürücünün müdahalesi olmadan seyir halinde gidebilen otomobillerdir.



Şekil.3.1.3.Sürücüsüz (Otonom) Araç Kullanma

4. **Yüz Tanıma:** Kişilerin yüz yapılarını şifreleme sistemi olarak değerlendiren ve bunu güvenlik amacı ile kullanan sistemlere yüz tanıma sistemleri adı verilir. Günümüzdeki son model iPhone telefonlarında FaceID adında yüz tanıma teknolojileri bulunmaktadır.



Şekil.3.1.4.Yüz Tanıma

5. **Savunma Sanayii:** Ülkemizde ODTÜ Görüntü Analizi Uygulama ve Araştırma Merkezi (OGAM) ve Aselsan Araştırma Merkezi Akıllı Veri Analitiği Araştırma Program Müdürlüğü tarafından derin öğrenme algoritmalarıyla türlü programlar geliştirilmektedir.



Şekil.3.1.2.Savunma Sanayii

6. **Sağlık Sektörü:** Kanser teşhisi, gen seçimi ve sınıflandırması, gen çeşitliliği, ilaç tasarımı, bileşim protein etkileşimi, RNA ile protein ilişkisi ve DNA metilasyonu gibi biyoinformatik uygulamalarında kullanılmaktadır.



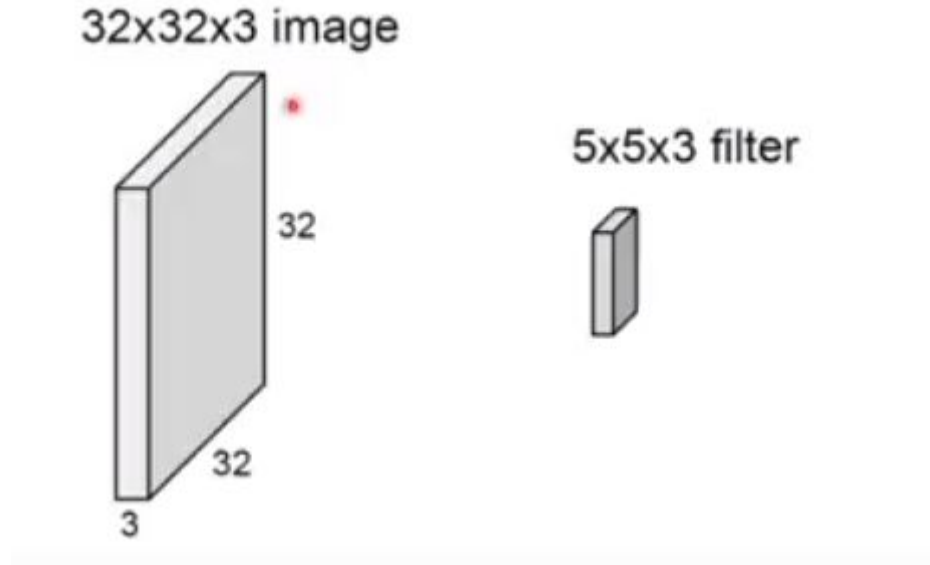
Şekil.3.2.0.Sağlık Sektörü

4. EVRİŞİMSSEL SİNİR AĞLARI (CNN)

4.1. Evrişimsel Sinir Ağları Nedir?

Bir evrişimsel sinir ağı (**ConvNet / Convolutional neural networks -CNN**), bir girdi görüntüsünü alıp, görüntüdeki çeşitli görüşleri/nesneleri birbirinden ayırabilen **derin öğrenme algoritmasıdır**.

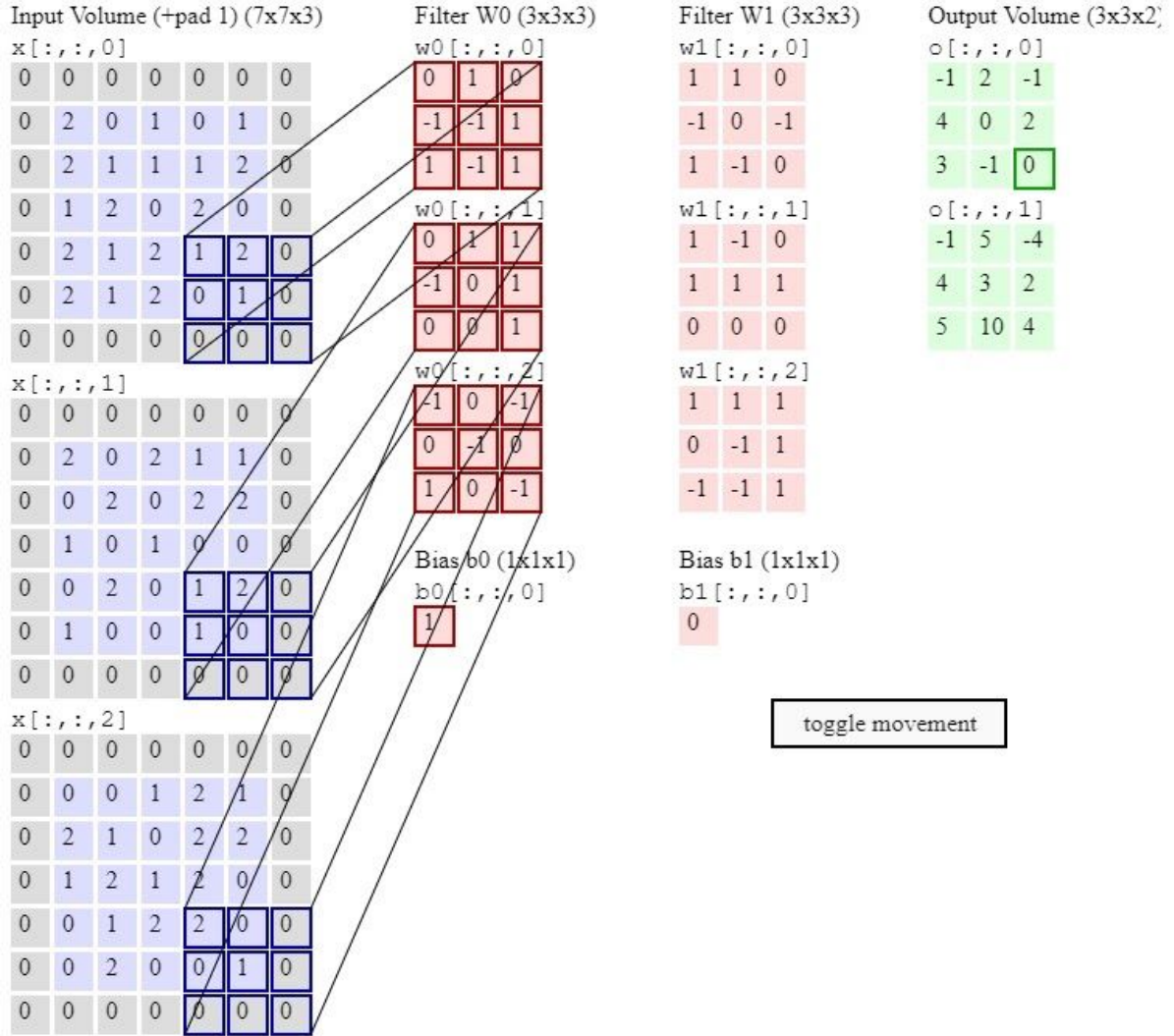
Evrişimsel sinir ağları, temel olarak görüntüleri sınıflandırmak (örneğin gördüklerini isimlendirmek), benzerlikle kümelemek (fotoğraf arama) ve sahnelerde nesne tanıma yapmak için kullanılan derin yapay sinir ağlarıdır.



Şekil.4.1.0.Evrişimsel Sinir Ağları Nedir?

4.2. Evrişimsel Sinir Ağları Nasıl Çalışır?

Evrişimli ağlar hakkında bilinmesi gereken ilk şey, görüntüleri insanlar gibi algılamamasıdır. Bu nedenle, kıvrımlı bir ağ tarafından beslenirken ve bu ağ tarafından işlenirken bir görüntünün ne anlama geldiğini farklı bir şekilde düşünmek gerekir.



Şekil.4.2.0.Evrişimsel Sinir Ağları Nasıl Çalışır?

Evrişimli ağlar, görüntüleri hacimler olarak algılar; yani, düz tuvaler yerine yalnızca genişlik ve yükseklik ile ölçülecek üç boyutlu nesneler. Bunun nedeni, dijital renkli görüntülerin kırmızı-mavi-yeşil (RGB) kodlamasına sahip olmasıdır ve bu üç rengi, insanların algıladığı renk spektrumunu oluşturmak için karıştırır. Evrişimli bir ağ, bu tür görüntüleri, biri diğerinin üzerine yığılmış üç ayrı renk katmanı olarak besler.

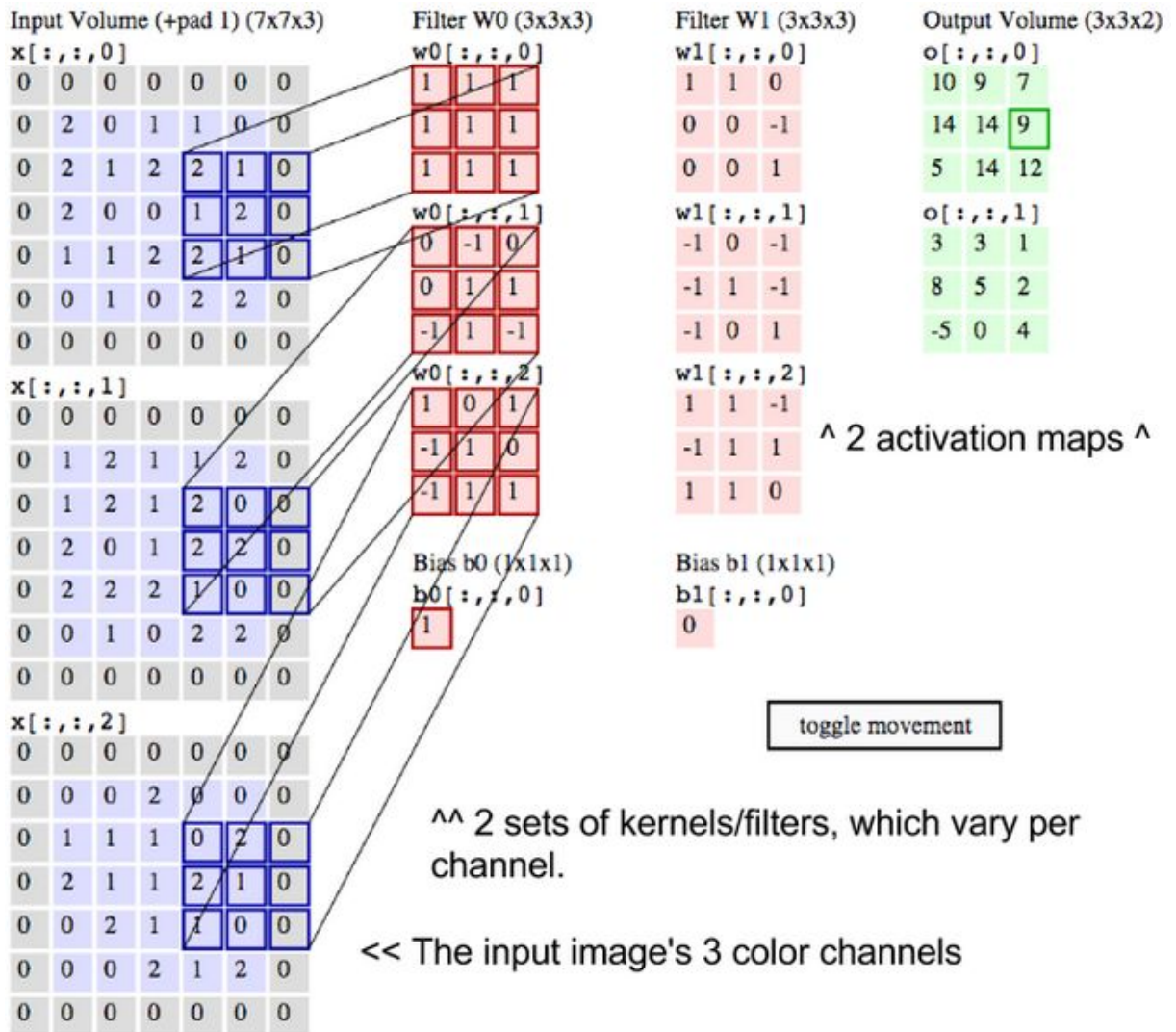
Dolayısıyla, evrişimli bir ağ, genişliği ve yüksekliği bu boyutlardaki piksel sayısı ile ölçülen ve derinliği RGB'deki her harf için bir tane olmak üzere üç katman derinliğinde olan dikdörtgen bir kutu olarak normal renkli bir görüntü alır. Bu derinlik katmanlarına kanallar denir.

Görüntüler evrişimli bir ağda hareket ettikçe, onları girdi ve çıktı hacimleri açısından tanımlayacağız ve bunları matematiksel olarak bu formdaki çok boyutlu matrisler olarak ifade edeceğiz: $30 \times 30 \times 3$. Katmandan katmana, boyutları aşağıda açıklanacak nedenlerle değişir.

Görüntüleri işlemek için kullanılan doğrusal cebir işlemlerinin temelini oluşturdukları için, görüntü hacminin her boyutunun kesin ölçülerine dikkat edilmesi gerekir.

Bir görüntünün her pikseli için, R, G ve B'nin yoğunluğu bir sayı ile ifade edilecek ve bu sayı, birlikte görüntü hacmini oluşturan üç yığılmış iki boyutlu matrislerden birinde bir eleman olacaktır. Bu sayılar, evrişimli ağı beslenen ilk, ham, duyuşal özelliklerdir ve ConvNets'in amacı, bu sayılardan hangisinin görüntüleri daha doğru bir şekilde sınıflandırmasına yardımcı olan önemli sinyaller olduğunu bulmaktır. (Tartıştığımız diğer ileri beslemeli ağlar gibi.)

Bir seferde bir piksele odaklanmak yerine, evrişimli bir ağ kare piksel yamalarını alır ve bunları bir filtreden geçirir. Bu filtre aynı zamanda görüntünün kendisinden daha küçük ve yama ile aynı boyutta bir kare matristir. Destek vektör makinelerine aşına olanlar için bir zil çalacak olan çekirdek de denir ve filtrenin görevi piksellerdeki desenleri bulmaktır.



Şekil.4.2.1.Evrişimsel Sinir Ağları Nasıl Çalışır?

5. ALEXNET

5.1. AlexNet Nedir?

AlexNet bir evrişimsel sinir ağı örneğidir. 2012 yılında LSVRC yarışmasını kazanan bir Evrişimli Sinir Ağı Mimarisine verilen isimdir. Bu algoritma, bu raporun 4.maddesinde bahsi geçen Evrişimsel Sinir Ağları algoritmasını temel alarak ve ana mantığını kullanarak kullanıcılara görüntü işleme (örn: nesne tanıma) alanlarında hizmet vermektedir.

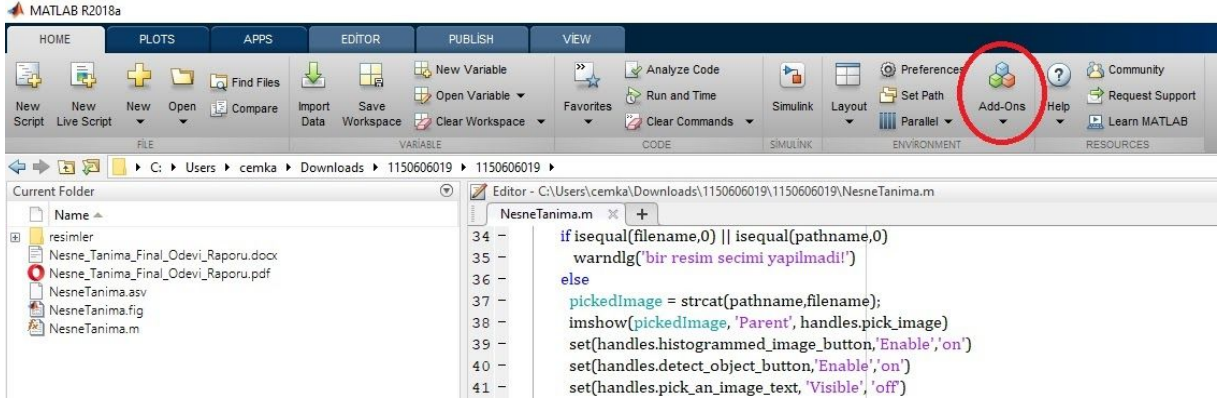
LSVRC (Büyük Ölçekli Görsel Tanıma Mücadelesi), araştırma ekiplerinin algoritmalarını büyük bir etiketli görüntü veri kümesi (ImageNet) üzerinde değerlendirdikleri ve çeşitli görsel tanıma görevlerinde daha yüksek doğruluk elde etmek için rekabet ettikleri bir yarışmadır. Bu, ekiplerin daha sonra tamamlamaya nasıl yaklaştığı üzerinde büyük bir etki yarattı.

AlexNet kütüphanesini **MATLAB** içerisine dahil edilerek, kullanıcı tarafından verilen fotoğraflardaki nesneleri tanımlanması sağlanır.

5.2. AlexNet Kurulumu

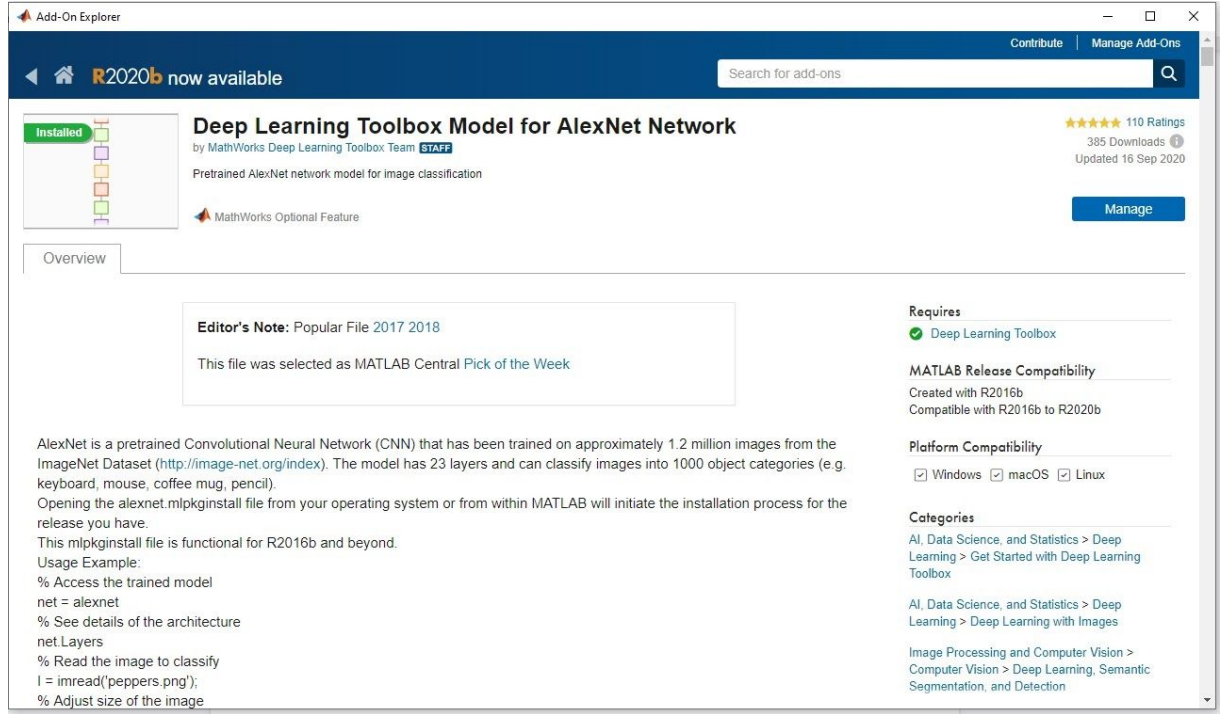
AlexNet kütüphanesinin **MATLAB** üzerine kurulumu aşağıda adımlar halinde gösterilmiştir.

1. **MATLAB** arayüzünde “**Home**” sekmesine gidilerek aşağıdaki resimde olduğu gibi kırmızı daire içerisinde bulunan “**Add-Ons**” butonuna tıklanır.



Şekil.5.2.0.AlexNet Kurulumu

2. Gelen “Add-On Explorer” pencereden arama kısmına “Deep Learning Toolbox Model for AlexNet Network” yazılarak arama işlemi yapılır.



Şekil.5.2.1.AlexNet Kurulumu

3. Bu pencereden “Install” yazan kısma tıklayarak bu kütüphanenin MATLAB’e kurulumu gerçekleştirilir.

6. NESNE TANIMA

Bu uygulamanın ana amacı kullanıcıdan alınan imajların içerisindeki nesnelerin tespitini sağlamaktır. Bu uygulamanın hata vermeden kullanabilmesi için 5.maddede belirtilen **AlexNet** algoritmasının kurulumu yapılması gerekmektedir. Uygulama içerisinde bulunan ekranlar aşağıda maddeler halinde gösterilmiştir.

6.1. Uygulamanın Giriş Ekranı

Bu ekranda seçilen imajları gösterebilen bir Axes, GUI ile callback bağlantıları yapılmış ve isimleri “**Resim Seç**”, “**Nesneyi Tanımla**”, “**Histogram Çizdir**” ve “**Çıkış**” butonları bulunmaktadır.

6.1.1. Axes Objesi

Bu obje, kullanıcının girdi olarak sağladığı imajın bu alanda görüntülenmesini sağlamaktadır.

6.1.2. Resim Seç Butonu

Bu butonuna tıklanarak kullanıcıdan bir resim seçilmesi istenmesi için bir **UIGetFile** penceresi açılır. Açılan pencereden tüm desteklenen resim formatlarının uzantılarına sahip imajların seçimi yapılabilir. Seçim yapıldıktan sonra seçilen imaj ekrandaki Axes kısmına konulur.

6.1.3. Histogram Çizdir Butonu

Bu butona tıklandığında yeni bir figür açılarak yeni açılan pencerede seçim yapılmış olan imajın histogram grafiği görüntülenmektedir.

6.1.4. Çıkış Butonu

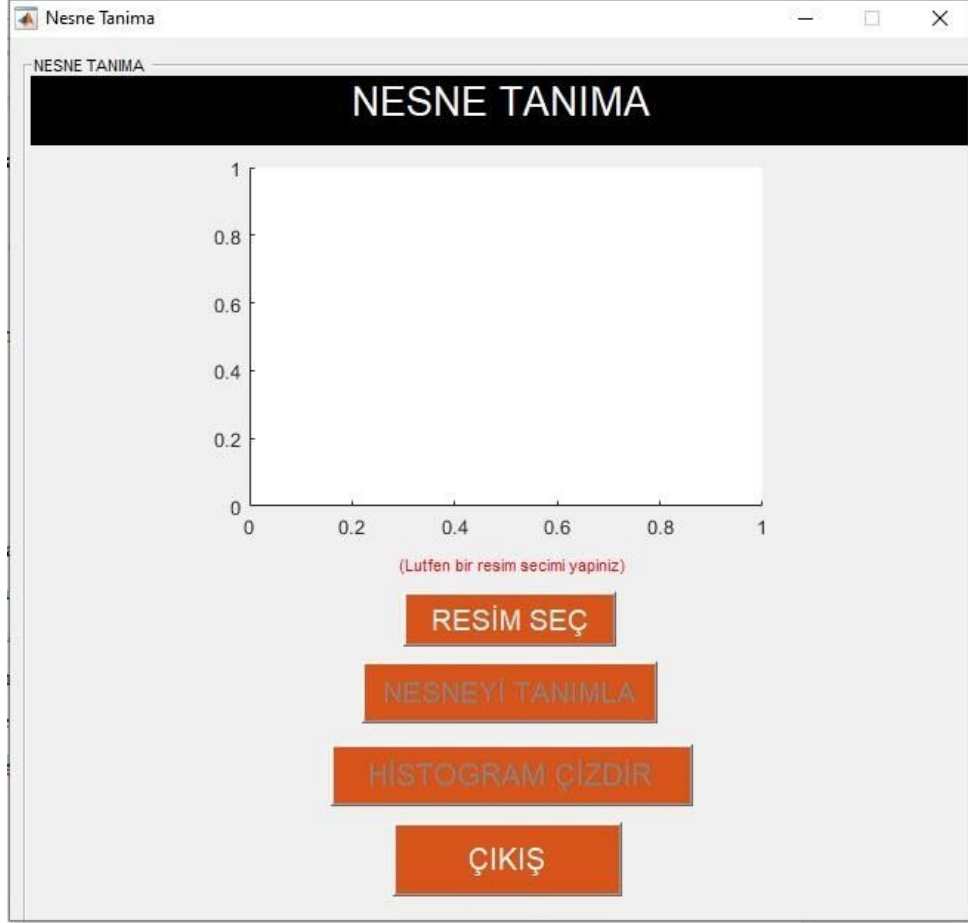
Bu butona tıklanarak uygulamadan çıkış yapılır.

7. KULLANIM SENARYOSU

Bu uygulamanın kullanıcı tarafından gerçekleştirilecek kullanım senaryosu aşağıda maddeler halinde gösterilmiştir.

7.1.1. Ana Ekran

Uygulama ilk açıldığında aşağıdaki ekranla karşılaşılır.

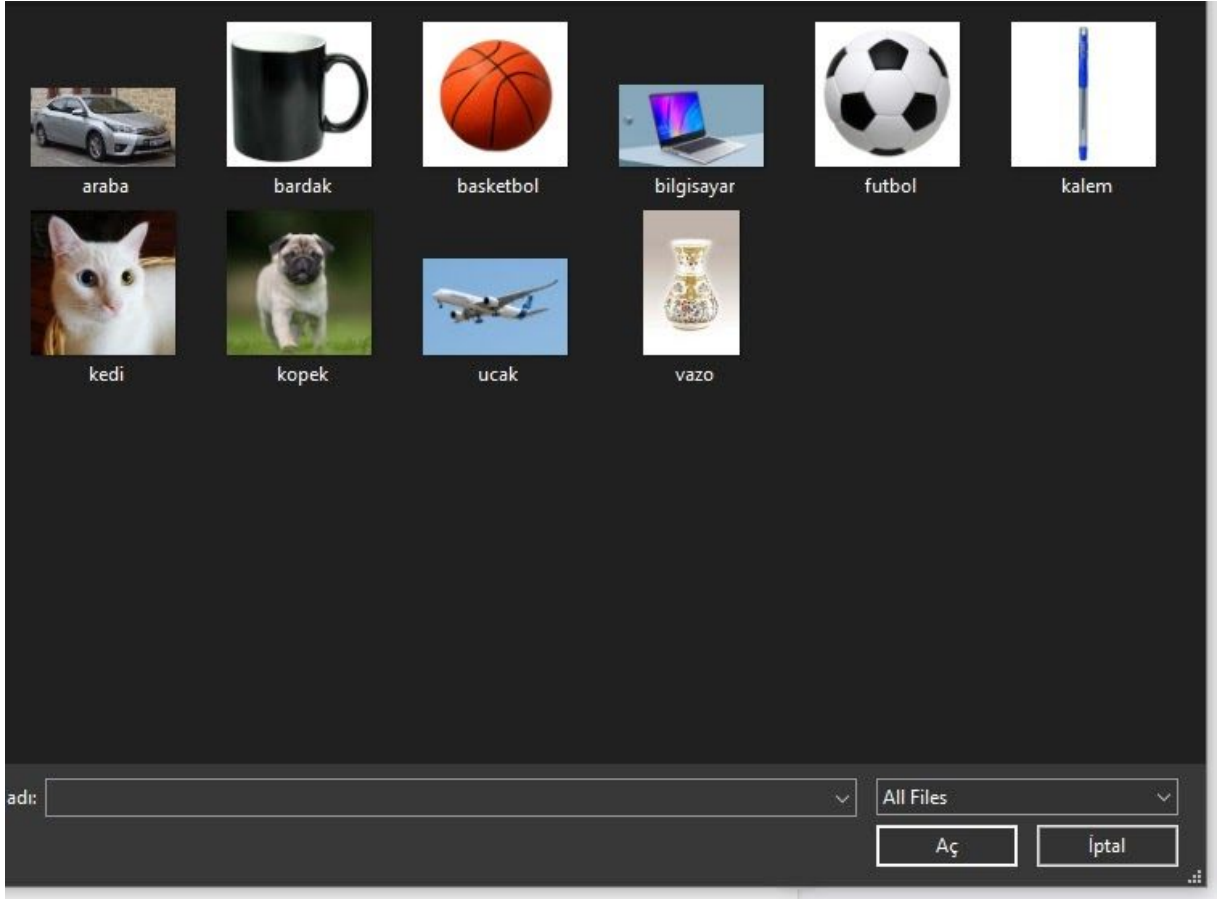


Şekil.7.1.1.0.Ana Ekran

“**Resim Seç**” butonuna tıklanıp imaj seçilmediği sürece uygulama ilk açıldığı andan itibaren “**Nesneyi Tanımla**” ve “**Histogram Çizdir**” butonları aktif olmayacaktır. Aynı zamanda “**Resim Seç**” butonu üzerindeki text objesi de imaj seçimi yapılmadığı sürece görünür olacaktır. İmaj seçimi yapıldıktan sonra “**Nesneyi Tanımla**” ve “**Histogram Çizdir**” butonları aktif olurken “(Lütfen bir resim seçimi yapınız)” değerine sahip text objesi görünmez olacaktır.

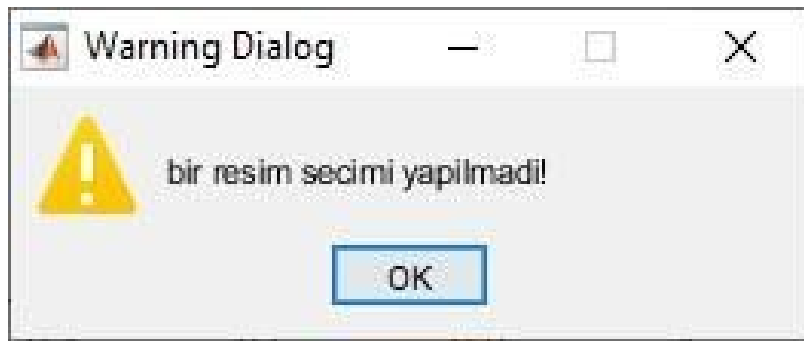
7.1.2. Resim Seçme Ekranı

Bu ekrandan istenilen uzantıya sahip herhangi bir imaj seçimi yapılabilir.



Şekil.7.1.2.0.Resim Seçme Ekran

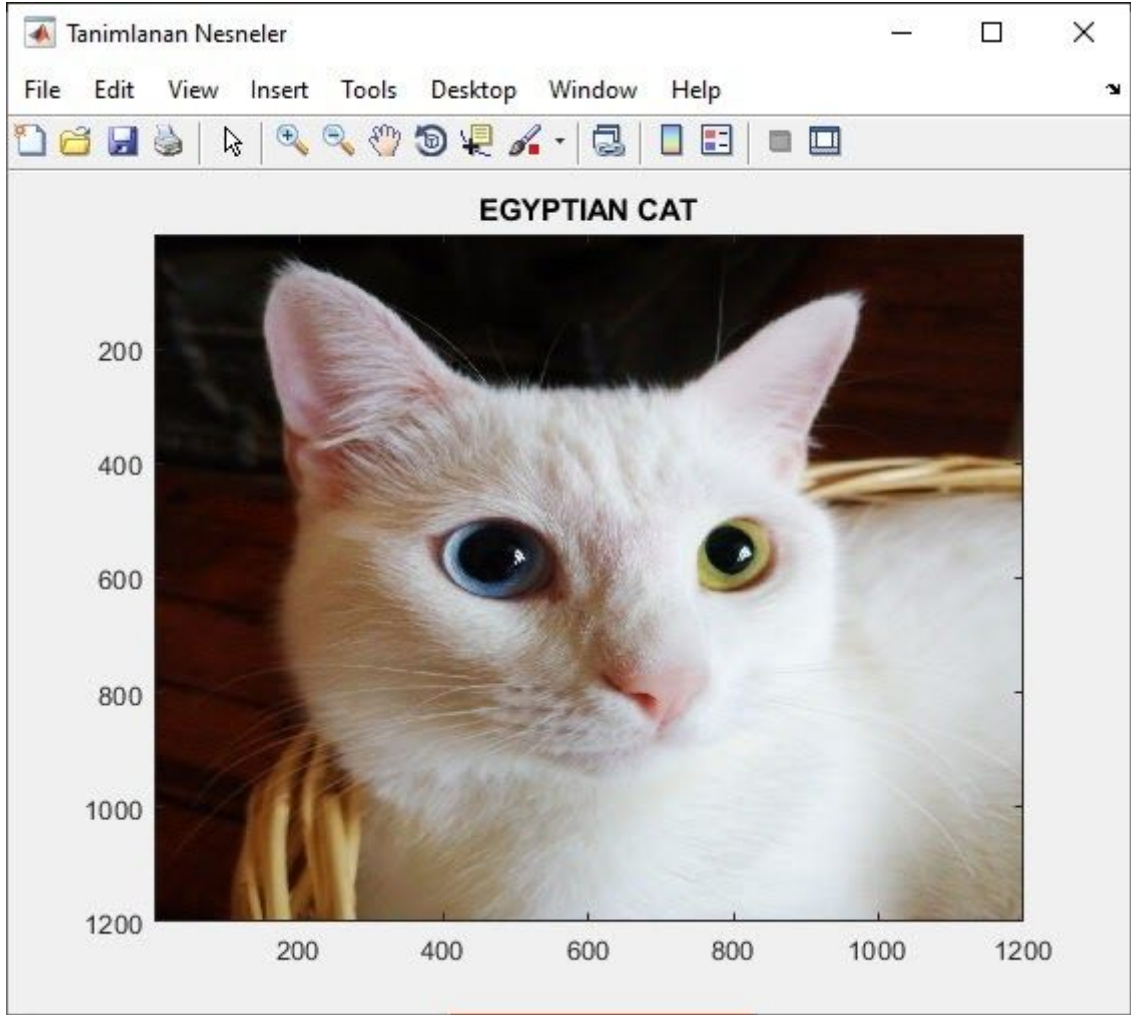
Eğer seçim yapılmadan bu ekran kapatılırsa kullanıcıya aşağıda pop-up'ın gösterimi sağlanır.



Şekil.7.1.2.1.Bir Resim Seçilmedi PopUp'ı

7.1.3. Nesneyi Tanımlama Ekranı

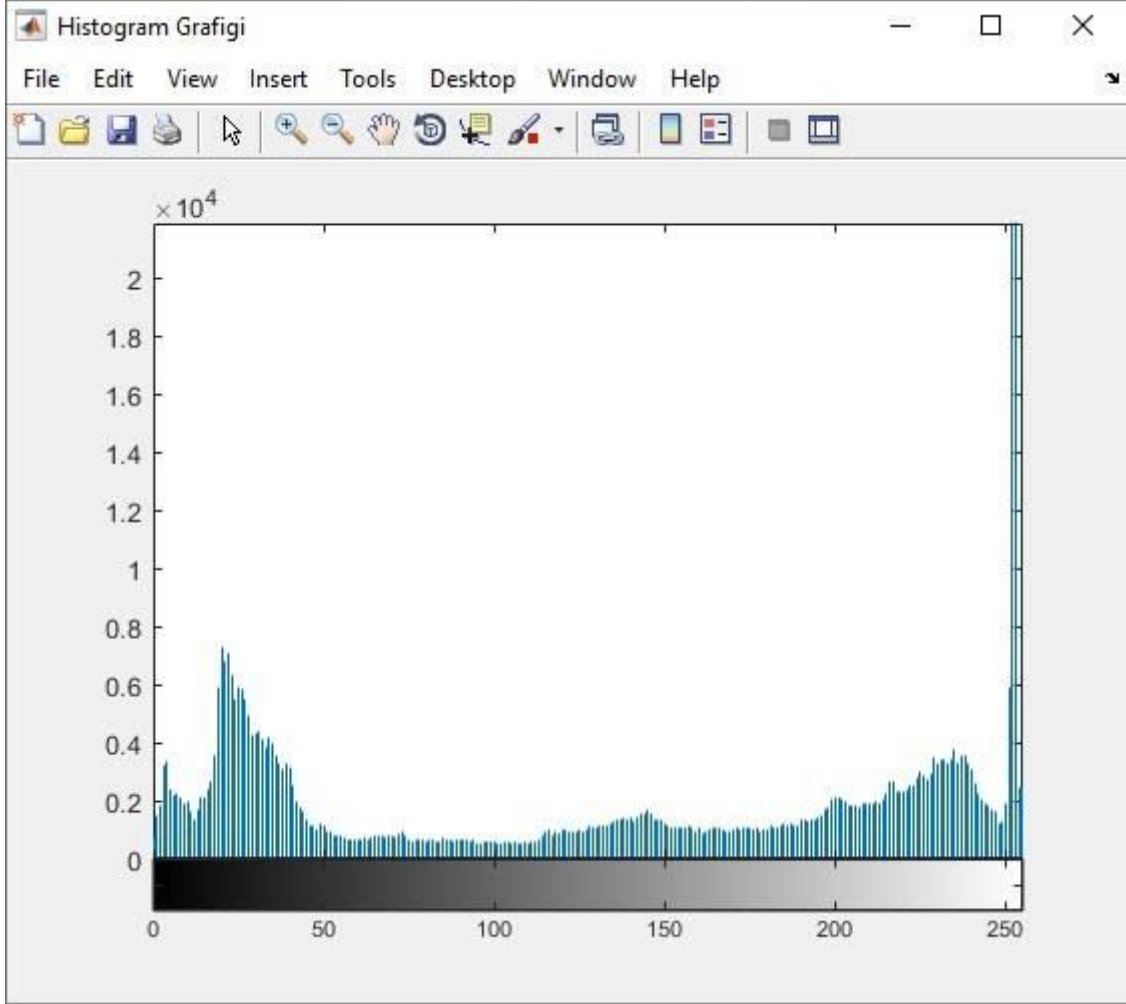
Kullanıcı bir resim seçimi yaptıktan sonra uygulamanın giriş ekranındaki “**Nesneyi Tanımla**” butonuna tıkladığında karşısına aşağıdaki gibi yeni figür penceresi açılacaktır. Bu pencerede resimdeki nesnelerin algılanması sağlanacaktır. Algılanan nesnenin ismi imajın üzerindeki başlıkta yer alacaktır.



Şekil.7.1.3.0.Nesneyi Tanımlama Ekranı

7.1.4. Histogram Grafiđi Ekranı

Kullanıcı seđtiđi imajın histogramını istemesi durumunda uygulamanın ana ekranında bulunan “**Histogram izdir**” butonuna tıklayarak yeni aılan pencerede ařađıdaki gibi grafiđini grntlemesini sađlayabilir.



řekil.7.1.4.0.Histogram Grafiđi Ekranı

8. ÖDEVİN ÖNEMLİ KODLARI

Uygulamanın inşa edilme aşamasında projeyi özel kılabilcek kodlar aşağıda maddeler halinde gösterilmiştir.

8.1.1. OpeningFcn

Bu fonksiyonda ekran ilk açıldığında çalışması istenilen aşağıdaki kodlar bulunmaktadır.

```
function NesneTanima_OpeningFcn(hObject, ~, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
set(handles.histogrammed_ball, 'Enable', 'off')
set(handles.detect_object, 'Enable', 'off')
set(handles.pick_an_image_text, 'Visible', 'on')
```

Şekil.8.1.1.0.OpeninFcn

Bu fonksiyon “**set(handles.histogrammed_ball, ‘Enable’, ‘off’)**” yazan kodda “**Histogram Çizdir**” butonu ekran ilk açıldığında inaktif olmasını sağlamaktadır. Bu kodun bir altındaki “**set(handles.histogrammed_ball, ‘Enable’, ‘off’)**” yazan kod “**Nesneyi Tanımla**” butonu için inaktiflik sağlamaktadır. En son satırda “**set(handles.pick_an_image_text, ‘Enable’, ‘off’)**” yazan kod ise “**(Lütfen bir resim seçimi yapınız)**” yazan text objesinin ekran ilk açıldığında görünür olmasını sağlamaktadır.

8.1.2. Load Image

```
function load_image_Callback(~, ~, handles)
    global pickedImage
    [filename, pathname] = uigetfile({'*.*'});
    if isequal(filename,0) || isequal(pathname,0)
        warndlg('bir resim secimi yapılmadi!')
    else
        pickedImage = strcat(pathname,filename);
        imshow(pickedImage, 'Parent', handles.pick_image)
        set(handles.histogrammed_ball,'Enable','on')
        set(handles.detect_object,'Enable','on')
        set(handles.pick_an_image_text, 'Visible', 'off')
    end
```

Şekil.8.1.2.0.Load Image

Bu fonksiyon “**filename, pathname**” yazan kısımda **UIGetFile** methodunu kullanarak kullanıcıya bir seçim ekranı açmayı ve ekrandan seçilen imajın bulunduğu yolu alma görevi görmektedir. Kullanıcı seçim yapmazsa bir pop-up gösterimi sağlanır, seçim yaparsa seçilen imaj Axes’te görüntülenir. Aynı zamanda bu görevlerle birlikte “**Nesneyi Tanımla**” ve “**Histogram Çizdir**” butonları aktifleştirilirken, “**(Lütfen bir resim seçimi yapınız)**” text objesi görünmez olur.

8.1.3. Detect Object

```
44 - function detect_object_button_Callback(~, ~, handles)
45 -     global pickedImage
46 -     detecting_image(pickedImage, handles)
47
48 - function detecting_image(file_name, ~)
49 -     detectedImage = imread(file_name);
50 -     nnet = alexnet;
51 -     figure('Name', 'Tanimlanan Nesneler', 'NumberTitle', 'off');
52 -     image(detectedImage);
53 -     picture = imresize(detectedImage, [227,227]);
54 -     label = classify(nnet, picture);
55 -     title(upper(char(label)));
```

Şekil.8.1.3.0.Detect Object

Bu fonksiyon “**Nesneyi Tanımla**” butonuna tıklandığında “**handles**” argümanı ile imajın bulunduğu yolu “**detecting_image**” fonksiyonuna gönderilmesini sağlar. “**detecting image**” isimli fonksiyonda “**nnet = alexnet;**” komutuyla AlexNet kütüphanesine erişim sağlanır ve bu kütüphaneye bağlı “**nnet**” adında bir varlık oluşturulur. Daha sonra “**Tanımlanan Nesneler**” adında bir figür açılması sağlanır. Bu figürde ana ekrandaki imajın gösterimi yapılır. Bir sonraki adımda “**picture**” adında yeniden boyutlandırılmış “**detectedImage**” için yeni bir varlık oluşturulur. Daha sonra aynı figür penceresinde resimde tanınan objelerin isimlerinin gösterilmesi için “**label = classify(nnet, picture);**” komutuyla AlexNet kütüphanesi tarafından resimdeki objeye verilen ismin “**label**” değişkenine atanması sağlanır. Son adımda “**title(upper(char(label)));**” komutuyla yeni figür penceresindeki resmin üstünde bir başlık oluşturulur ve bu başlığa “**label**” değişkeninin AlexNet kütüphanesinden öğrenmiş olduğu obje ismi yazdırılır.

8.1.4. Histogrammed Image

```
function histogrammed_image_Callback(~, ~, ~)
    global pickedImage;
    pureImage = imread(pickedImage);
    figure('Name','Histogram Grafiği','NumberTitle','off');
    imhist(pureImage)
```

Şekil.8.1.4.0.Histogrammed Image

Bu fonksiyon kullanıcının seçmiş olduğu imaj için “Histogram Çizdir” butonuna tıklandığında tetiklenir. Bunun sonucunda bir pencere (figür) açılır ve seçilen imajın histogram grafiği bu pencerede görüntülenir.

9. PROJEDE KULLANILAN ALGORİTMALAR

Bu projenin yapım aşamasında ihtiyaç duyulan ve inşa edilen algoritmalar aşağıda maddeler halinde gösterilmiştir.

9.1.1. AlexNet

Görüntü işleme teknolojisinde kullanılan bir evrimsel sinir ağı metodudur.

9.1.2. imShow

Görüntülerin kullanıcı ekranında görüntülenmesini sağlar.

9.1.3. imHist

Görüntülerin histogram grafiğini çizdirir.

10. KAYNAKLAR

D: “**Udentify**”, son güncelleme 31 Aralık 2020 <https://www.udentify.co>

D: “**MesutPiskin**”, son güncelleme 31 Aralık 2020 <https://mesutpiskin.com>

D: “**MathWorks**”, son güncelleme 31 Aralık 2020 <https://www.mathworks.com>

D: “**Medium**”, son güncelleme 31 Aralık 2020 <https://ayyucekizrak.medium.com>

D: “**Pathmind**”, son güncelleme 31 Aralık 2020 <https://wiki.pathmind.com>

D: “**TowardDataScience**”, son güncelleme 31 Aralık 2020 <https://towardsdatascience.com>

D: “**Wikipedia**”, son güncelleme 31 Aralık 2020 <https://www.wikipedia.org>

D: “Çeyrek Mühendis”, son güncelleme 31 Aralık 2020 <https://www.ceyrekmuhendis.com>

ÖZGEÇMİŞ



Kişisel Bilgiler

Ad-Soyad : Cem Kazım GENEL

Doğum Tarihi : 21.12.1996

Doğum Yeri : İstanbul

Eğitim Bilgileri

Lise : 2010 - 2014

Üniversite : 2015 - ...