



T.C.
SELÇUK ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ
BÖLÜMÜ

MAHMUT ULUDAĞ
1933302025

2016
KONYA

İÇİNDEKİLER

1. GİRİŞ	4
2. Sonar Nedir	4
3. SONAR ÇEŞİTLERİ.....	4
3.1. AKTİF SONAR	4
3.2. PASİF SONAR	4
3.3 SONAR GÖRÜNTÜ VERİ SETİ.....	5
4. YAPAY ZEKA	7
4.1 YAPAY ZEKANIN TANIMI	7
4.2. KERAS NEDİR	7
5. PYTHON	8
5.1. PYTHON NEDİR	8
5.2. PYCHARM KURULUM	9
6. PYTHON OPENCV	10
6.1. OPENCV NEDİR	10
6.2. OPENCV KURULUMU.....	10
7. PYTHON KERAS KURULUMU.....	11
8.OPENCV KOMUTLARI	12
9. GOOGLE COLAB İLE KERAS KULLANIMIN	15
9.1. FONKSİYONLAR VE İŞLEVLERİ	15
9.2. PROJE İÇİN HAZIRLANMIŞ FONKSİYONLAR	18
10. EĞİTİM İŞLEMİ SİZE VE BATCHSİZE.....	21
10.1 BATCHSİZE 500	21
10.2 BATCHSİZE 300	25
10.3 BATCHSİZE 150	29
10.4 BATCHSİZE 75	33
10.5 BATCHSİZE 50	37
10.6 BATCHSİZE 15	40

10.7 TEK KANALDA BATCHSİZE VE SİZE	44
10.8 ÜÇ KANALDA BATCHSİZE VE SİZE	44
10.9 ÜÇ Kanalda Filtre BATCHSİZE VE SİZE	44
11. EĞİTİM İŞLEMİ KONFEKSİYON MATRİX	45
11.1 TEK KANALDAKİ EN İYİ SONUÇ	45
11.2 ÜÇ KANALDAKİ EN İYİ SONUÇ.....	45
11.3 ÜÇ KANALDA FİLTREDEKİ EN İYİ SONUÇ	46
12. KAYNAKLAR	47

1. GİRİŞ

Bu yazıda sonar görüntü nedir ve sonar görüntü nasıl işlenir bunlara yapay zeka nasıl tanıtırız bunları görecez. Hangi yapay zekanın kullanıldığını ve hangi görüntü işleme metotları kullanıldığını göstermeye çalışacağız.

2. Sonar Nedir

Sonar aleti ses dalgaları ile çalışan ve denizin altında bulunan cisimlerin boyunu, uzaklığını ve sahip olduğu verileri tespit eden aletlerden biridir. Bununla beraber bu alet sayesinde kişiler su altında veya üzerinde gezmeyi, bilgi akışı sağlamayı ve su içinde bulunan cisimler hakkında bilgi toplamak için kullanılmaktadır. Sonar aleti ses dalgaları ile çalışmaktadır bununla beraber bu sistem ilk olarak denizaltı için çıkarılmıştır. Ses dalgaları sayesinde su altında ve üzerinde gezme mesafesi hesaplanmaktadır. Bununla beraber yunus, yarasa ve benzeri ses dalgaları ile iletişime geçen canlılarla iletişim için kullanılmaktadır.

3. SONAR ÇEŞİTLERİ

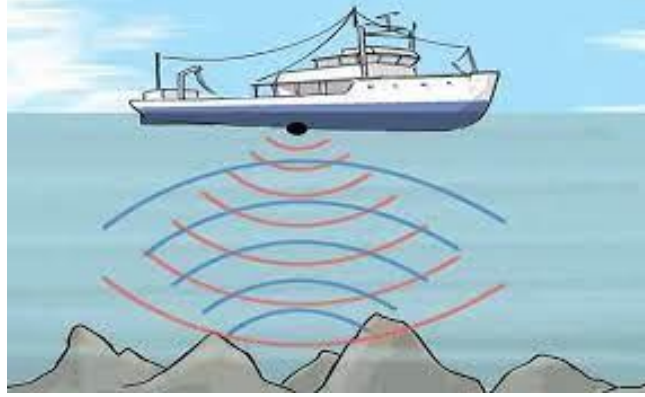
3.1. AKTİF SONAR

Aktif sonar transdüserleri, suya akustik bir sinyal veya ses darbesi yayar. Bir nesne ses darbesinin yolundaysa, ses nesneden zıplar ve sonar dönüştürücüye bir “eko” döndürür. Transdüserde sinyal alma özelliği varsa, sinyalin gücünü ölçer. Transdüser, ses darbesinin yayılması ve alımı arasındaki süreyi belirleyerek, nesnenin aralığını ve yönünü belirleyebilir.

3.2. PASİF SONAR

Pasif sonar sistemleri, öncelikle deniz cisimlerinden (denizaltılar veya gemiler gibi) ve balinalar gibi deniz hayvanlarından kaynaklanan gürültüyü tespit etmek için kullanılır. Aktif sonardan farklı olarak, pasif sonar bulunamayan askeri gemiler veya okyanusu sessizce “dinlemeye” odaklanan bilimsel görevler için bir avantaj olan kendi sinyalini vermez.

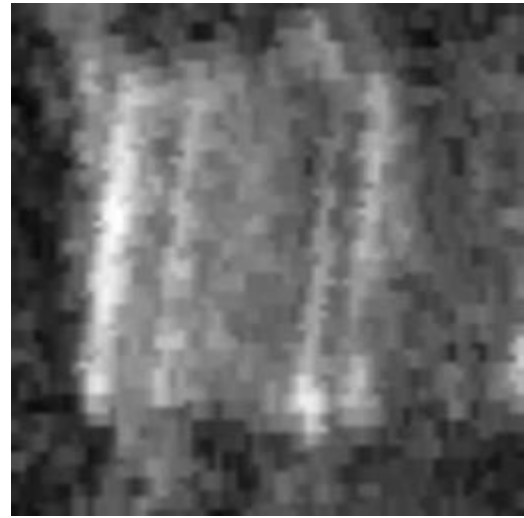
Aksine, yalnızca kendisine gelen ses dalgalarını algılar. Pasif sonar, diğer pasif dinleme cihazlarıyla birlikte kullanılmadığı sürece bir nesnenin aralığını ölçemez. Çoklu pasif sonar cihazları, bir ses kaynağının üçgenleştirilmesine izin verebilir.



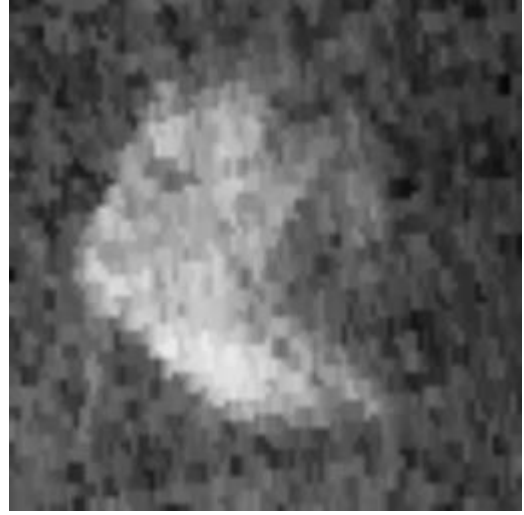
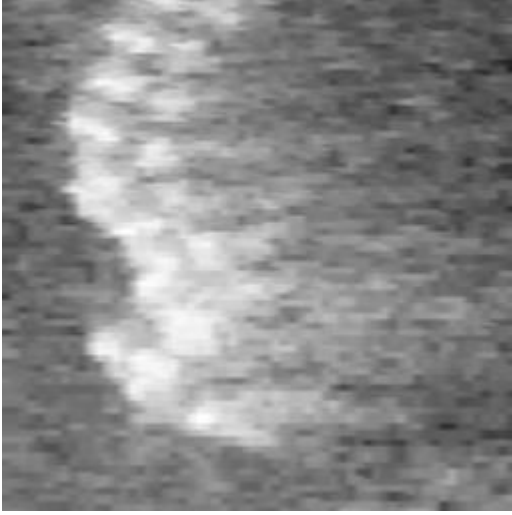
Şekil 3.1 Geminin sonar görüntü alma şekli

3.3 SONAR GÖRÜNTÜ VERİ SETİ

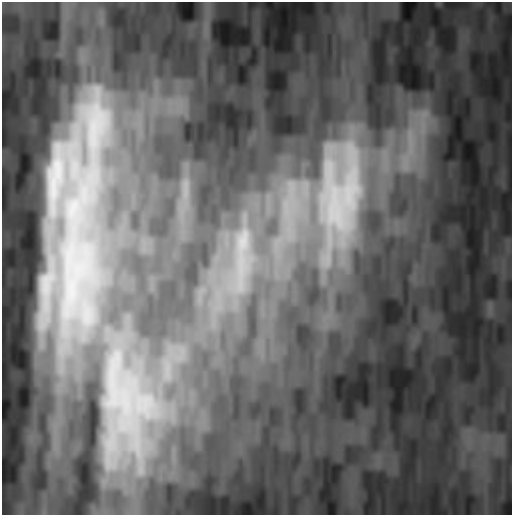
Sonar görüntü veri setimizde 10 adet sınıf bulunmaktadır bunlar şişe (bottle) , kutu (can) , zincir (chain) , içecek kartonu (drink carton) , kanca (hook), pervane (propeller) , şampuan şişesi (shampoo bottle), duran şişe (standing bottle), tekerlek (tire) , kapak (valve) 'dir.



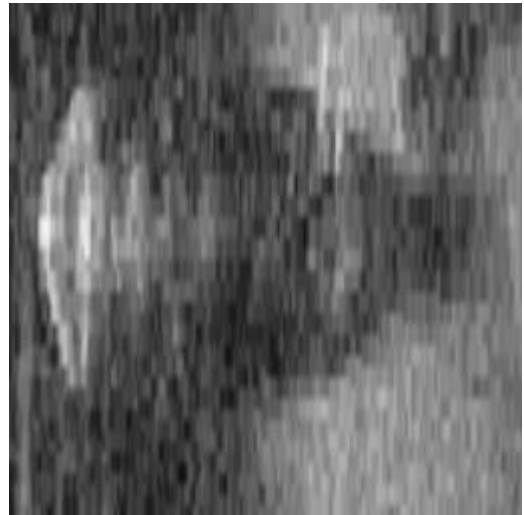
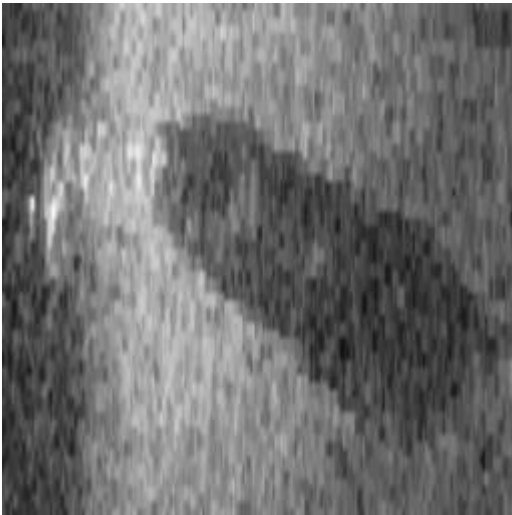
Şekil 3.2 Şişe (bottle) ve kutu (can)



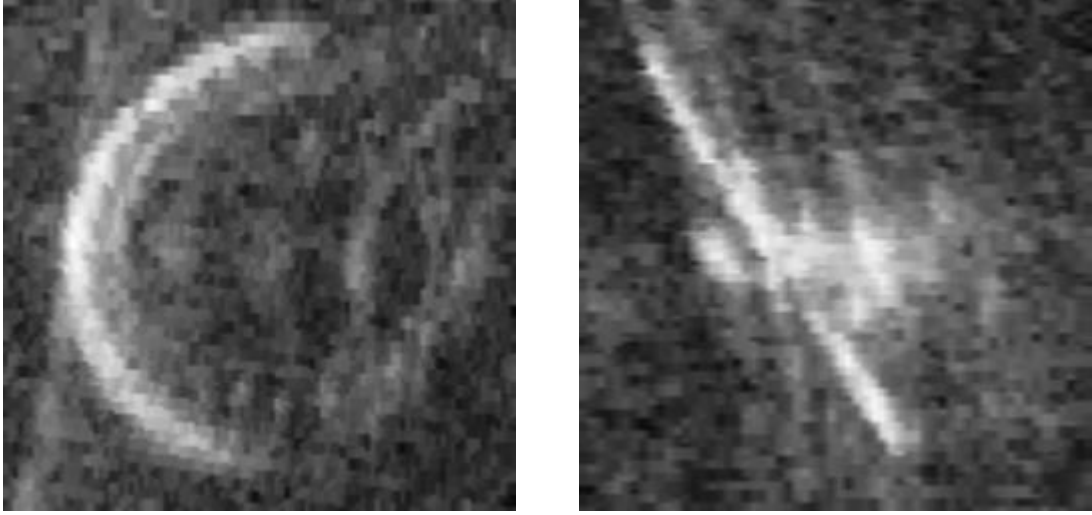
Şekil 3.3 Zincir (chain) ve içecek kartonu (drink carton)



Şekil 3.4 Kanca (hook) ve pervane (propeller)



Şekil 3.5 Şampuan şişesi (shampoo-bottle) ve duran şişe (standing-bottle)



Şekil 3.6 Tekerlek (tire) ve kapak (valve)

4. YAPAY ZEKA

4.1 YAPAY ZEKANIN TANIMI

Yapay Zeka ; öğrenme , gerekçeleme , problem çözme , yabancı bir dili alma vb. gibi insanoğlunun davranışlarını gösterebilen sistemlerle ilgilenen bir bilgisayar bilimidir.

Yapay Zeka'nın ana amacı insanların davranışlarının ve sezgisel yeteneklerinin bilgisayar üzerinde benzetimidir

Yapay Zeka'nın standart bir tanımı yapılamamakla beraber, yapılagelen tanımların ortak yönleri şunlardır; Yapay Zeka bir bilgisayar bilim dalıdır, Yapay Zeka bilgi ve davranışa dayanır, Yapay Zeka zeki davranışları araştırmaktadır . Zeka rakam ya da veriler yerine bilgiye dayalı mantıksal bir süreçtir. Bilgi ve bilginin işlenmesi ile zeki davranışlar ortaya çıkarılabilir.

İnsanoğlu esas olarak Bilgi'yi kullanmakta ve onu işlemektedir. Bu yüzden bilgi ve bilginin kullanımı Yapay Zeka'nın da anahtar karakteristikleridir

4.2. KERAS NEDİR

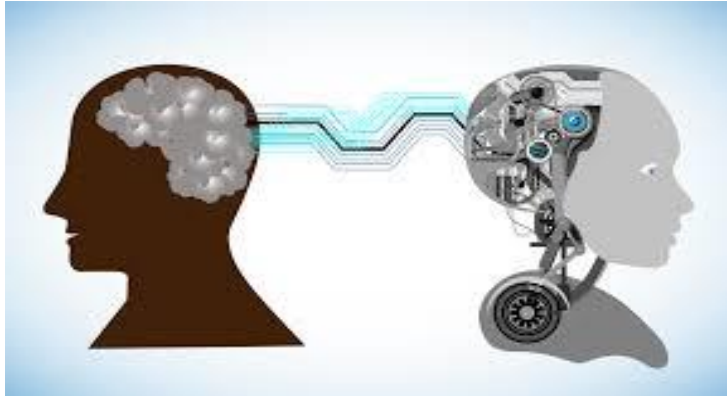
Keras, neredeyse her tür derin öğrenme modelini tanımlamak ve eğitmek için uygun bir yol sağlayan Python için bir derin öğrenme kütüphanesidir. Keras,

Tensorflow , Theano ve CNTK üzerinde çalışabilen Python ile yazılmış bir üst düzey sinir ağıları apisiidr.

İçerdiği çok fazla işlevsel fonksiyon sayesinde Keras kolayca bir derin öğrenme modeli oluşturmamızı ve onu eğitmemizi sağlıyor. Bu nedenle derin öğrenmeyi daha kolay öğreniyoruz.

Derin sinir ağıları ile hızlı deney yapabilmek için tasarlanmıştır. Kolay ve hızlı bir şekilde model oluşturulmasına olanak sağlarken deneme yanılma yaparak modelde neleri değiştirebileceğimize karar vermemizi kolaylaştırır. Bilgisayarlı görme modelleri için evrişimli sinir ağılarını (CNN), sürekli veriler içinse yinelemeli sinir ağılarını (RNN) destekler .

Keras 'ta model tanımlamanın iki yolu bulunmaktadır. Sequential ve Functional API . Sequential, en sık kullanılan olmakla beraber katmanları yığmak için kullanılır. Functional API ise çoklu çıktılı modeller gibi karmaşık model mimarileri tasarlamak için kullanılır



Şekil 4.1 yapay zeka ile ilgili görsel

5. PYTHON

5.1. PYTHON NEDİR

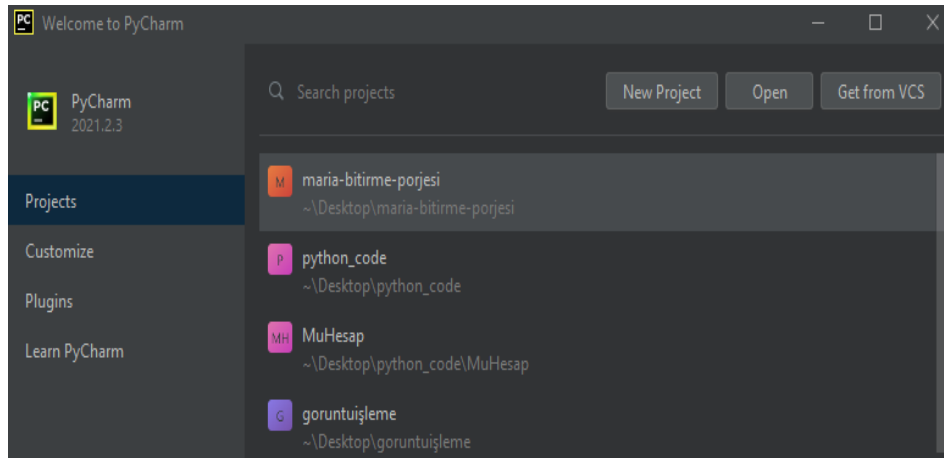
Python, nesne yönelimli, yorumlamalı, birimsel (modüler) ve etkileşimli yüksek seviyeli bir programlama dilidir.

Girintilere dayalı basit söz dizimi, dilin öğrenilmesini ve akılda kalmasını kolaylaştırır. Bu da ona söz diziminin ayrıntıları ile vakit yitirmeden programlama yapılmaya başlanabilen bir dil olma özelliği kazandırır.

Modüler yapısı, sınıf dizgesini (sistem) ve her türlü veri alanı girişini destekler. Hemen hemen her türlü platformda çalışabilir (Unix, Linux, Mac, Windows, Amiga, Symbian). Python ile sistem programlama, kullanıcı arabirimi programlama, ağ programlama, web programlama, uygulama ve veri tabanı yazılımı programlama gibi birçok alanda yazılım geliştirebilirsiniz. Büyük yazılımların hızlı bir şekilde prototiplerinin üretilmesi ve denenmesi gerektiği durumlarda da C ya da C++ gibi dillere tercih edilir.

5.2. PYCHARM KURULUM

Pycharm, python için oluşturulmuş IDE programıdır. Python piyasasında kullanılan en iyi IDE dir. Bu IDE ile çok hızlı ve kolay şekilde uygulama geliştirebilirsiniz . JetBrains orijinal sitesinde Community versiyonunu indirdik.



Şekil 5.1 Pycharm Giriş arayüz

Tabi bunun çalışabilmesi içinde Python kurulumuda yapmamız gerekdi . Python kendi sitesinde indirdikten sonra pycharm artık kullanabilir hale gelmiş oluyor.

Python kurulumu tamamlandıktan sonra istediğimiz kütüphaneleri yükleyebilecek hale gelmiş olur. Opencv ve Keras kütüphaneleri yükleyerek işe başlıyoruz.

6. PYTHON OPENCV

6.1. OPENCV NEDİR

OpenCV bilgisayarla görü, makine öğrenimi, görüntü işleme, video analizi gibi uygulamalar için kullanılan devasa bir açık kaynak kodlu kütüphanedir. Gerçek zamanlı işlemlerde oldukça önemli bir rol oynamaktadır. OpenCV kütüphanesi sayesinde web kameraları, video dosyaları veya diğer aygıt türleri tarafından bir bilgisayara bağlanan görsel bilgilerin yakalanmasını, analiz edilmesini ve değiştirilmesini destekleyen yüzlerce işlev içerir. Bir ekrana bir çizgi veya başka bir şekil çizmek için basit işlevler kullanılabilirken, kitaplığın daha gelişmiş kısımları yüzleri algılamak, hareketi izlemek ve şekilleri analiz etmek için algoritmalar içerir. Bu kitaplığın algoritmalarının çoğu, ürün incelemesi, tıbbi görüntüleme, robotik, yüz ve hareket tanıma ve insan-bilgisayar etkileşimi dahil olmak üzere bilgisayarla görmenin belirli kullanımlarıyla ilgilidir. OpenCV, hem ticari hem de hobi projelerinde oldukça az kısıtlama ile kullanılabilir.

OpenCV kütüphanesi sayesinde bir geliştirici, bilgisayarla görü işlevini güvenilir bir şekilde oluşturmaya yönelik karmaşık ve sıkıcı işlerin bir kısmını ortadan kaldıracak ve uygulamayı oluşturmaya odaklanabilir. Bir programcı, yüz tanıma ve benzeri için algoritmalar oluşturmak yerine, bir programın uygun kitaplık işlevine erişmesini sağlamak için yalnızca birkaç satır kod ekleyebilir. Aynı zamanda, bir programcının onu kullanan bir program oluşturmak için bilgisayar vizyonunun her yönüne hakim olması gerekmediği anlamına gelir.

6.2. OPENCV KURULUMU

OpenCV kurmak baya kolay bir iş. OpenCV kurmak için aşağıdaki adımları sırayla izleyebilirsiniz:

CMD(Komut İşleyicisi) açın. Windows arama kısmına cmd yazarak açabilirsiniz.

- pip install opencv-python yazdıktan sonra komutu gönderin.
- OpenCV kütüphanesi başarılı bir şekilde Python kütüphanesine eklenecektir.

```
Command Prompt
Microsoft Windows [Version 10.0.17134.829]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Waseem>pip install opencv-python
Collecting opencv-python
  Using cached https://files.pythonhosted.org/packages/99/ad/864b2f284517249a96190fdf6b21a291d9716115d6886938ccd86afaf821/opencv_python-4.1.0.25-cp37-cp37m-win32.whl
Requirement already satisfied: numpy>=1.14.5 in c:\users\waseem\appdata\local\programs\python\python37-32\lib\site-packages (from opencv-python) (1.16.4)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.1.0.25
```

Şekil 6.1 Opencv cmd ile kurulumu

OpenCV kurulumunu test etmek istiyorsanız etkileşimli oturumda aşağıdaki komutu çalıştırabilirsiniz.

```
Command Prompt - python
Microsoft Windows [Version 10.0.17134.829]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Waseem>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import cv2
>>> print(cv2.__version__)
4.1.0
>>>
```

Şekil 6.1 Opencv cmd ile versiyonunu görme

7. PYTHON KERAS KURULUMU

Keras kurmak için aşağıdaki adımları sırayla izleyebilirsiniz:

CMD(Komut İşleyicisi) açın. Windows arama kısmına cmd yazarak açabilirsiniz.

pip install keras yazdıktan sonra komutu gönderin.

Keras kütüphanesi başarılı bir şekilde Python kütüphanesine eklenecektir.

```

C:\Users\dataflair>pip3 install keras
Requirement already satisfied: keras in c:\users\shivam\appdata\local\programs\python\python37\lib\site-packages
(2.3.1)
Requirement already satisfied: numpy>=1.9.1 in c:\users\shivam\appdata\local\programs\python\python37\lib\site-p
ackages (from keras) (1.17.2)
Requirement already satisfied: keras-preprocessing>=1.0.5 in c:\users\shivam\appdata\local\programs\python\pytho
n37\lib\site-packages (from keras) (1.1.0)
Requirement already satisfied: pyyaml in c:\users\shivam\appdata\local\programs\python\python37\lib\site-package
s (from keras) (5.1.2)
Requirement already satisfied: h5py in c:\users\shivam\appdata\local\programs\python\python37\lib\site-packages
(from keras) (2.10.0)
Requirement already satisfied: six>=1.9.0 in c:\users\shivam\appdata\local\programs\python\python37\lib\site-pac
kages (from keras) (1.12.0)
Requirement already satisfied: scipy>=0.14 in c:\users\shivam\appdata\local\programs\python\python37\lib\site-pa
ckages (from keras) (1.3.1)
Requirement already satisfied: keras-applications>=1.0.6 in c:\users\shivam\appdata\local\programs\python\python
37\lib\site-packages (from keras) (1.0.8)
WARNING: You are using pip version 19.3.1; however, version 20.0.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
C:\Users\dataflair>_

```

Şekil 7.1 Keras cmd ile kurulumu

8.OPENCV KOMUTLARI

```

import cv2
image = cv2.imread("bottle-0.png", 0)
image=cv2.resize(image,(256,256))
cv2.imshow("resim",image);
cv2.waitKey(0)
cv2.destroyAllWindows()

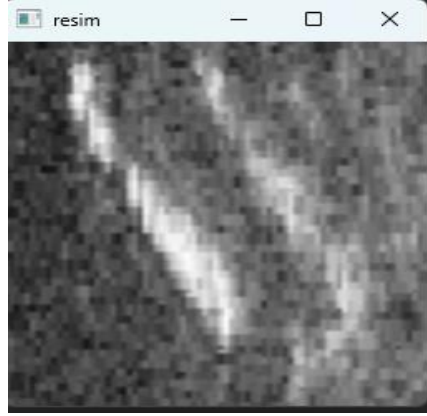
```

Şekil 8.1 Python opencv ile görsel okuma kodu

Burada yazdığımız kodda ilk başta import cv2 diyerek kurduğumuz opencv kütüphanesini aktif ediyoruz. Image diye bir değişkene resmin atamasını gerçekleştiriyoruz. Cv2.imread diyerek resmi alıyoruz imread fonksiyonun ilk aldığı parametre resmimizin bilgisayardaki konumu yani dosya yolunu girmemiz gerekir biz burada direk resmin ismini girme nedenimiz Python dosyamızla aynı klasörde olmasından dolayıdır. İkinci girdiğimiz parametre ise gri tonlamaya dönüştürmek içindi 0 yerine flags=cv2.IMREAD_GRAYSCALE şeklinde girsek de olurdu ikisi de aynı anlama geliyor sonuçta ama 0 yazmak daha kolay olduğundan biz 0 tercih ettik.

İkinci olarak yaptığımız işlem ise cv2.resize yani yeniden boyutlandırma resize fonksiyona biz iki parametre girdik bunlardan ilki imread ile okuduğumuz resmimizdir ikinci ise tuple türünde bir parametredir ilk değeri width yani genişlik ikinci değeri ise

height yani yükseklik değeri oluyor en son da bu yeni boyutlandırılmış görselimizi tekrardan image içine atıyoruz. Cv2.imshow kod parçasığında ise görseli ekranda göstermemize sağlıyorlar. Cv2.waitKey ve cv2.destroyAllWindows ise görüntünün biz kapatana kadar ekranda kalmasını sağlıyor.



Şekil 8.2 Sonar şişe görseli

Çıktısı da bu şekilde olur ve programı bitirmek için ise çarpıya basarak bitirebiliriz.

```
Tags = {
    0: 'Bottle',
    1: 'Can',
    2: 'Chain',
    3: 'Drink-carton',
    4: 'Hook',
    5: 'Propeller',
    6: 'Shampoo-bottle',
    7: 'Standing-bottle',
    8: 'Tire',
    9: 'Valve',
    10: 'Wall',
}

class GetImage:
    def __init__(self, path):
        self.path = path
        self.Images = self.Image(path)
    def Image(self, path):
        Imageslist = []
        dosyaisimleri = os.listdir(path)
```

Şekil 8.3 Python'da sınıfların listesi kodu

Yukardaki kod parçasığında ilk başta dict türünde bir değişken oluşturuyoruz daha sonra bu oluşturduğumuz değişkenin içerisine bizde bulunan görüntülerin isimlerini tag olarak kullanmak için sayısal değerlere atıyoruz bu işlem normalde diğer yazılım dillerinde enum olarak geçer ama burada bunu tanımlamak için ekstra bir şey yapmadan direk tanımlıyoruz.

GetImage adında bir class yani sınıf türetiyoruz önce sınıfın constructor yani sınıfın oluşturucusunu oluşturuyoruz. Oluşturduğumuz constructor içerisinde bir path yani dosya yolu belirttirilmesini istiyoruz ve bunu sınıfımızın path eşitliyoruz.

Image diye bir fonksiyon oluşturuyoruz ve fonksiyonumuz bir input yani girdisi bulunmaktadır.

Bu girdi bir tür str yani string Türkçe karşılığı söylemek gerekirse Türkçe metin anlamına geliyor yani bir metin ,isim ya da sözel bir ifade kullanmak istediğimizde işimizi çok kolaylaştıran bir değişkendir str . Fonksiyonda ilk başta Imagelist=[] diyerek Imagelist'in bir liste olduğunu belirttik bu liste sayesinde birden fazla olan verilerimiz birden fazla değişken yazmakla uğraşmadan tek değişkende tuttuk . Daha sonra Dosyaisimleri diye bir değişken daha oluşturduk bu değişkene os.listdir(path) diyerek bize verilen dosya yolunun altındaki dosya isimlerini almamıza yarıyor bizde verilen dosya yolunun altında bulunan bottle , can , vb dosya isimlerini alıyoruz.

```
class GetImage:
    def __init__(self, path):
        self.path=path
        self.Images=self.Image(path)
    def Image(self, path):
        Imageslist = []
        dosyaisimleri=os.listdir(path)
        i=0
        for dosyaismi in dosyaisimleri:
            altdosya=os.listdir(path+"\\ "+dosyaismi)
            tag=Tags[i]
            i+=1
            for image in altdosya:
                Imageslist.append((tag,cv2.imread(path+"\\ "+dosyaismi+"\\ "+image,0)))
        return Imageslist
GetImage("marine-debris-watertank-release\\class-image-crops")
```

Şekil 8.4 Python class yapısı ile toplu görsel okuma kodu

Bu çektiğimiz verileri tek tek ayırmak için bir for döngüsü kullanıyoruz bu sayede her dosya dosyaismi değişkeni olarak bize verecek. Altdosya diye bir değişken

oluşturuyoruz bu veri nesnesine dosyaismi altındaki dosyaların isimlerini atıyoruz yani bottle altında olan bottle-0.png isimlerini atıyoruz. Bir for döngüsü daha kullanarak altdosya dizinlerini tek tek ayırarak image diye tanımlanmış veri nesnesine atıyoruz . Daha önce bahsettiğimiz Imagelist şimdide ekleme yapacağız ekleme yapmak için append komutunu kullanacağız . Append komutunun içerisine bir tuple şeklinde veri ataması yapıyoruz bu tuple içerisinde bir tane tag ve imread ile okunmuş resim bulunuyor burada tag kullanmamızın sebebi görselin nereye ait olduğunu bilmemiz için , tag dediğimiz değişken Tags diye daha önce yazdığımız dict değişkenin bir elemanıdır yani örnek verecek olursak bir görsel okundu mesela bottle-154.png diye bunun bir Bottle tag taşımalarını sağlıyoruz. Bu sayede daha sonra yapacağımız görüntüyü işleme vb yerlerde kullanımız kolaylaştıracaktır.

9. GOOGLE COLAB İLE KERAS KULLANIMIN

9.1. FONSIYONLAR VE İŞLEVLERİ

Google colab yardımıyla GPU dolaylı ortaya çıkan sorunları ve hataları ortadan kaldırdık. Kısaca hazırladığınız fonsiyonları inceleyelim.

```
class TrainModel():
    def Model(self,channel,size):
        from keras.layers.convolutional import Convolution2D, MaxPooling2D
        from keras.models import Sequential
        from keras.layers.core import Dense, Dropout, Activation, Flatten
        model = Sequential()
        model.add(Convolution2D(32, (3, 3),
                                activation='relu',
                                input_shape=(channel, size, size),
                                data_format='channels_first'))
        model.add(Activation('relu'))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Dropout(0.2))
        model.add(Convolution2D(64, kernel_size=(3, 3)))
        model.add(Activation('relu'))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Dropout(0.2))
        model.add(Flatten())
        model.add(Dense(128))
        model.add(Activation('relu'))
        model.add(Dense(10))
        model.add(Dropout(0.2))
        model.add(Activation('softmax'))
        model.compile(loss='categorical_crossentropy',
                      optimizer='adam',
                      metrics=["binary_accuracy"])
        return model
```

Şekil 9.1 Python oluşturduğumuz keras modeli

TranModel isminde bir class oluşturduk. Bunun içerisinde 3 adet fonksiyon tanımladık ilk fonksiyonumuz “ Model ” bu fonksiyonda oluşturacağımız keras iç mimarisi oluşturduk. Öncelikle kullanacağımız kütüphaneleri tanımladık model=Sequential() Keras katmanlarını sıralı bir düzende düzenlemektir ve katmanları eklemeye başladık. Bu katmanlar kısaca aktivasyon fonksiyonları ve filtreler yer almaktadır.

```
def GetDatas(self,listimg,listlabel,num_class):
    import numpy as np
    from keras.utils import np_utils
    from sklearn.utils import shuffle
    img_data = np.array(listimg)
    img_data = img_data.astype('float32')
    img_data /= 255
    labels = np.array(listlabel)
    Y = np_utils.to_categorical(labels, num_class)
    x, y = shuffle(img_data, Y, random_state=30)
    return x,y

def MyModel(self,imgdata, labeldata, size=28, channel=1, batchsize=200,
            epochs=15, split=0.2, num_class=10):
    import numpy as np
    from sklearn.model_selection import train_test_split
    x,y=self.GetDatas(imgdata,labeldata,num_class)
    X_train, X_test, y_train, y_test = train_test_split(x,y,test_size=split,random_state=6)
    model=self.Model(channel,size)
    X_train = X_train.reshape(-1, channel, size, size)
    X_test = X_test.reshape(-1, channel, size, size)
    hist = model.fit(X_train,y_train,
                    batch_size=batchsize,
                    epochs=epochs,
                    verbose=1,
                    validation_data=(X_test, y_test))
    return model, hist
```

Şekil 9.2 Python modelin eğitilme kodu

Bir diğer fonksiyonumuz GetDatas fonksiyonudur. Bu fonksiyonda öğretilcek olan görsellerin ve bu görsellerin labelleri istenmektedir. Num_class olarak isten değişken ise kaç çeşit label olduğunu belirtir. Fonksiyonda ilk olarak kullanılacak kütüphaneler eklendi ve sonrasında img_data değişkenine görselleri bir array yani bir matriks olarak atadık matrikste görsellerin sıralı bir şekilde pixel değerleri tutulmaktadır. Bu tutulan veri sayısal olarak işleme yapıldığında çok büyük değerler ortaya çıkacağından dolayı öğrenme hızı düşecektir buna karşı önlem almak için tüm değerleri 255 böldük ve hepsi 0-1 arasında değerler almış oldu. Labels değişkenine gelen label değerlerini bir dizi formatına çevirdik bu diziyi hepsi sayısal olarak sınıflandırdık örnek verecek olursak mesele bottle (şişe) sınıflandırmadaki değeri [1 0

0 0 0 0 0 0 0 0] can ise [0 1 0 0 0 0 0 0 0] olarak sınıflandırır. Bu işlemlerden sonrada verileri kendi içerisinde 30 defa karıştırdıktan sonra verileri geri döndürüyoruz. Diğer fonksiyonumuz MyModel bu fonksiyonun için gereken değişkenler ; imgdata (eğitim için gereken görseller) , labeledata (görsellerin etiketleri) , size (görsellerin hangi boyutta oldukları) , channel (hangi kanalda oldukları yani gri ise 1 renkli ise 3 olarak girilmeli) , batchsize (veriler eğitilirken her grupta kaç tane veri olacağını belirtir) , epochs (eğitim kaç defa olacağını belirtir) , split (verilerin yüzde kaç test için kullanılacağını belirtir) ve num_class (kaç adet etiket olduğunu belirtir). Fonksiyonda kütüphaneleri tanımlayarak başlıyoruz tanımlar bittikten sonra verilerin düzenlenmesi için az önce bahsettiğimiz getdatas fonksiyonunu kullanarak düzenliyoruz.

Bu fonksiyonundan gelen değerleri x ve y diye değişkenler atıyoruz. Bu değişkenleri sklearn kütüphanesinden gelen train_test_split fonk. kullanarak verileri girilen yüzden değerine göre eğitim ve test için ayırıyoruz. Model=self.model() diyerek yazdığımız model mimarisini model değişkenine atıyoruz ve atama yaparken fonksiyonumuza bir channel değeri girilmesi lazım ve bu değeri giriyoruz. Eğitim ve test için ayırdığımız görsellerin matrislerini yeniden boyutlandırıyoruz bunu yapmamız nedeni görsellerimiz eğer tek channel(kanal) ise matris boyutlarında kanal değişkeni bulunmaz eğer 3 channel ise matris boyutlarında en sonda değer olarak gözükür. Bu konu hakkında şöyle kısaca bir örnek veriler bilir mesela tek kanallı 120 adet görselimiz bulunsun ve bunların ilk başta matrisleri boyutları (120, 32, 32) şeklinde olur ilk baştaki 120 kaç görsel olduğunu belirtir 2 değer ve 3 değer ise görsellerin boyutlarını belirtir. 3 channel(kanal) olsaydı matris boyutları çıktısı (120, 32, 32, 3) olurdu aslında tek kanal aynısı sadece dördüncü bir değer gelmiş oluyor bu değer ise 3 channel(kanal) olduğunu belirtiyor. Ama kerasda bizim model fonksiyonunda belirttiğimiz data_format= “ channels_first ” olduğundan dolayı ilk başta channel yani kanal değeri verilmelidir bizde bunu yeniden boyutlandırma (reshape) ile hızlıca gerçekleştiriyoruz. X_train.reshape(-1, channel, size, size) buradaki -1 görsellerin sayının geleceği yer bizim -1 dememizin nedeni ise matrisin içerisinde kaç adet varsa direk almasını söyledik ve bu şekilde daha sağlıklı olmuş olur. Diğerleri zaten adı üstünde kanal ve boyutların girildiği yerdir. Az önce bahsettiğimiz örnekleri bu şekilde boyutlandırırsak şöyle olur ; tek kanal için (120, 1, 32, 32) 3 kanallı ise (120, 3, 32, 32) bu şekilde olur. En sonda model.fit diyerek eğitimi başlatıyoruz tabi burada model.fit içerisinde yazılması gereken değişkenler var bunlar görseller, etiketler, batchsize, epochs ve test için kullanılacak değerlerin girilmesi gerekiyor ve model ve hist değerleri

döndürüyoruz. Burada şuna kadar anlatılan kısım keras ile nasıl bir model oluşturacağımız gösteriyor. Bu “ TrainModel ” classını kullanmak istersek eğer şu şekilde kullanabiliriz ;

```
images=[]
labels=[]
size=32
channel=1
model, hist = trainmodel.MyModel(images, labels, size, channel)
```

Şekil 9.3 Python kodu ile model class'ını çağırma

Images ve labels dediğimiz değişkenler birer list türünde olmalıdır. Bu listeler görseller ve bu karşılık gelen etiketler doğru bir şekilde aktarılmalıdır. Size resimlerin boyutlarını ifade eder ama burada dikkat etmemiz gereken size değişkeni görsellerin boyutları ne ise o girilmelidir yoksa model size ile resimleri size uyuşmaz ve modeli eğitemeyiz. Channel da resimlere göre doğru bir şekilde girilmeli yoksa size aldığımız aynı hatayı alırız.

9.2. PROJE İÇİN HAZILANMIŞ FONKSİYONLAR

Projemiz için bir takım fonksiyonlar gerekiyordu. İlk olarak bahsedeceğim fonksiyon test fonksiyonu yani eğitimi tamamladıktan sonra verileri test etmeye ihtiyacımız vardı bu ihtiyacı karşılamak için aşağıdaki gibi bir fonksiyon yazdık.

```
def Test(test_image,label,tmodel,channel=1,size=28):
    import cv2
    from matplotlib import pyplot
    import numpy as np
    sinif=np.zeros((11, 11)).astype(int)
    #test_image=cv2.resize(test_image,(size,size))
    test_image = np.array(test_image)
    test_image = test_image.astype('float32')
    test_image /= 255
    test_image = test_image.reshape(-1, channel ,size ,size)
    f=(tmodel.predict(test_image) > 0.5).astype("int32").tolist()
    liste=[]
    for ff in f:
        deger=0
        for ind,item in enumerate(ff):
            deger+=(ind+1)*item
        liste.append(deger)
    for ind,list in enumerate(liste):
        if(list==0):
            sinif[label[ind],10]+=1
            sinif[10,label[ind]]+=1
        else:
            sinif[label[ind],list-1]+=1
    return sinif;
```

Şekil 9.4 Python kodu ile model test etme kodu

Bu fonksiyonu kısaca özetlemek gerekirse ilk başta fonksiyonun çalışabilmesi için gereken bazı değişkenlerimiz mevcuttur. Bu değişkenler kısaca şöyledir test_image (test edilecek görseller),

label (görseller karşılık gelen etiketler), tmodel (eğitilmiş modelimiz), channel (görsellerin kanalı) ve size (görsellerin boyutları) . Fonksiyonumuzu kütüphaneleri tanımlayarak yazmaya başlıyoruz ve sonrasın daha önce bahsettiğimiz getdatas fonksiyonun yaptığı işlemleri yapıyoruz yani görselleri ve etiketleri eğitimiz modele göre düzenliyoruz. Burada sınıf diye bir matris oluşturduğum bu matrisin boyutlarını 111X11 yaptık. Tmodel.predict(test_image) diyerek model üzerinde görseli veya görselleri veriyoruz tmodel de çıktı olarak bize bazı değerler veriyor bunlar şöyledir ;

```
1/1 [=====] - 7s 7s/step
[[9.7582543e-01 1.0351008e-02 3.4721778e-04 8.6398731e-04 3.2846356e-04
 2.4433702e-03 9.7584445e-03 3.3604418e-05 4.8528400e-05 8.1578353e-09]
 [9.9950159e-01 1.6542827e-04 1.3317913e-04 2.1815781e-06 6.4281019e-05
 3.9443028e-05 8.0867714e-05 1.3692544e-06 1.1670115e-05 2.6806701e-10]]
```

Şekil 9.5 Modelin sonuç çıktısı

Burada çıktı olarak verdiği verilere bakınca çok bir şey anlaşılmıyor burada 0,5 büyük olan yer bizim sınıfımızın ne olduğunu belirtiyor. Burada matris ilk değerine bakacak olursanız 0.9'un farklı bir şekilde yazılmış halidir.

```
[[ True False False False False False False False False False]
 [ True False False False False False False False False False]]
```

Şekil 9.6 Model sonuç çıktısı

0.5 büyük olanlar True küçük olanlar yerine ise False yazılmıştır. Bu şekilde batkımızda artık bir anlam kazamamış oldu True değer index olarak 0 indexde olduğundan dolayı burada , ikisinde bottle(şişe)'dir. f=(Tmodel.predict(test_image)>0.5).astype("int32") dersek burada 0,5 büyük olanları 1 olarak küçük olanları ise 0 olarak çıktı verecektir. Şöyle gösterilebilir ;

```
[[1 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0]]
```

Şekil 9.7 Model sonuç çıktısı

Bu daha çok benzedi daha önce bahsettim labels değişkenine gelen label değerlerini bir dizi formatına çevirdik bu diziyi hepsi sayısal olarak sınıflandırdık demiştik ve orada bir örnek vermiştik bu örnekte bottle değerinin [1 0 0 0 0 0 0 0 0] olduğu söylemiştik ve burada çıktı aynı olduğundan modele gönderdiğimiz görsel bir bottledir dedik.

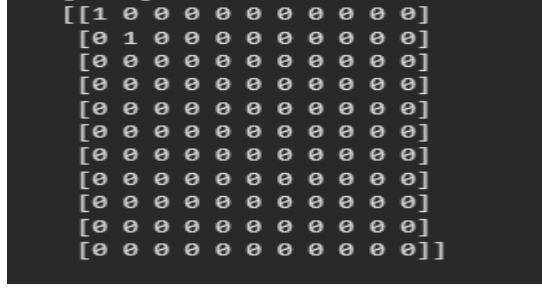
```
f=(tmodel.predict(test_image) > 0.5).astype("int32").tolist()
liste=[]
for ff in f:
    deger=0
    for ind,item in enumerate(ff):
        deger+=(ind+1)*item
    liste.append(deger)
for ind,list in enumerate(liste):
    if(list==0):
        sinif[label[ind],10]+=1
        sinif[10,label[ind]]+=1
    else:
        sinif[label[ind],list-1]+=1
return sinif;
```

Şekil 9.8 Model çıktısı alma kodu

Burada modelimizden çıkan sonuçlar bir liste halinde olduğundan dolayı liste for ff in f diyerek elemanlarına ayırıyoruz daha sonra ff dediğimiz değişken bir liste türündendir ff elemanlarına ayırarak değer değişkenimize atıyoruz.

Bunu kısa bir örnekle açıklayacak olursak şöyle açıklarız ;

Ff her çıktısı [1, 0, 0, 0, 0, 0, 0, 0, 0] bu formatta olur bu da bir liste olduğundan dolayı bunu tekrar bir for ile ayırıyoruz. For gelen ind değeri for kaçınıcı döngüde olduğunu yani listenin hangi index geldiğini ifade eder item ise listeden gelen değeri ifade ediyor yani sırasıyla item 1,0,0,0, şeklinde değer alacaktır. Deger+=(ind+1)*item yani gelen index değeri bir artır item değeri ile çarp mesele sırasıyla 1,0,0 gelsin 1 index'si 0'dır deger+=(0+1)*1 yani deger değişkenin yeni değeri 1 olur sonra 0 index'si 1'dır geldiğinde deger+=(1+1)*0=0 olur 1+0=1 olur bu şekilde devam eder. Oluşturduğumuz bu listeyi tekrar ayrıştırarak oluşturduğumuz matrikse yazıyoruz. Ve sonuç çıktısı şöyle olur ;



Şekil 9.9 Çıktının Confusion matrix

Bu bir “ confusion matrix ” çıktısıdır.

10. EĞİTİM İŞLEMİ SİZE VE BATCHSİZE

Eğitimi kısaca şöyle yapıyoruz ;

```
images, labels = getDatas(size, channel)
trainmodel = TrainModel()
model, hist = trainmodel.MyModel(images, labels, size, channel)
ind = Test3(images, labels, model, channel, size)
```

Şekil 10.1 Modelin çalıştırma kodu

İlk başta veri setimiz çağırıyoruz daha sonra bu veri setini oluşturduğumuz class ile eğitiyoruz ve test koyuyoruz.

10.1 BATCHSİZE 500

Eğitimi yaparken ilk başta tek kanalda batchsize 500 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.1 Tek kanalda 16 ve 28 görüntü boyutunun 500 batchsize doğruluk çıktısı

Tek Kanal						Tek Kanal					
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	340	109	449	76%	24%	bottle	378	71	449	84%	16%
can	175	192	367	48%	52%	can	330	37	367	90%	10%
chain	63	163	226	28%	72%	chain	127	99	226	56%	44%
drink-carton	289	60	349	83%	17%	drink-carton	286	63	349	82%	18%
hook	46	87	133	35%	65%	hook	93	40	133	70%	30%
propeller	0	137	137	0%	100%	propeller	116	21	137	85%	15%
shampoo-bottle	0	99	99	0%	100%	shampoo-bottle	30	69	99	30%	70%
standing-bottle	0	65	65	0%	100%	standing-bottle	51	14	65	78%	22%
tire	267	64	331	81%	19%	tire	313	18	331	95%	5%
valve	198	10	208	95%	5%	valve	201	7	208	97%	3%
toplam	1378	986	2364	58,29%	41,71%	toplam	1925	439	2364	81,43%	18,57%

Çizelge 10.2 Tek kanalda 32 ve 64 görüntü boyutta 500 batchsize doğruluk çıktısı

Tek Kanal						Tek Kanal					
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	356	93	449	79%	21%	bottle	402	47	449	90%	10%
can	326	41	367	89%	11%	can	350	17	367	95%	5%
chain	129	97	226	57%	43%	chain	176	50	226	78%	22%
drink-carton	312	37	349	89%	11%	drink-carton	316	33	349	91%	9%
hook	83	50	133	62%	38%	hook	98	35	133	74%	26%
propeller	109	28	137	80%	20%	propeller	126	11	137	92%	8%
shampoo-bottle	42	57	99	42%	58%	shampoo-bottle	72	27	99	73%	27%
standing-bottle	42	23	65	65%	35%	standing-bottle	53	12	65	82%	18%
tire	313	18	331	95%	5%	tire	311	20	331	94%	6%
valve	199	9	208	96%	4%	valve	201	7	208	97%	3%
toplam	1911	453	2364	80,84%	19,16%	toplam	2105	259	2364	89,04%	10,96%

Çizelge 10.3 Tek kanalda 92 görüntü boyutta 500 batchsize doğruluk çıktısı

Tek Kanal					
92X92	Train	Loss	Toplam	%Train	%loss
bottle	406	43	449	90%	10%
can	302	65	367	82%	18%
chain	140	86	226	62%	38%
drink-carton	320	29	349	92%	8%
hook	110	23	133	83%	17%
propeller	128	9	137	93%	7%
shampoo-bottle	55	44	99	56%	44%
standing-bottle	57	8	65	88%	12%
tire	316	15	331	95%	5%
valve	201	7	208	97%	3%
toplam	2035	329	2364	86,08%	13,92%

Burada görüldüğü üzere tek kanalda 500 batchsize en iyi size 64 çıktı.

Eğitimi yaparken ikinci olarak 3 kanalda batchsize 500 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.4 Üç kanalda 16 ve 28 görüntü boyutta 500 batchsize doğruluk çıktısı

3 Kanal						3 Kanal					
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	371	78	449	83%	17%	bottle	396	53	449	88%	12%
can	189	178	367	51%	49%	can	331	36	367	90%	10%
chain	84	142	226	37%	63%	chain	126	100	226	56%	44%
drink-carton	297	52	349	85%	15%	drink-carton	305	44	349	87%	13%
hook	18	115	133	14%	86%	hook	84	49	133	63%	37%
propeller	42	95	137	31%	69%	propeller	104	33	137	76%	24%
shampoo-bottle	6	93	99	6%	94%	shampoo-bottle	35	64	99	35%	65%
standing-bottle	0	65	65	0%	100%	standing-bottle	47	18	65	72%	28%
tire	314	17	331	95%	5%	tire	307	24	331	93%	7%
valve	193	15	208	93%	7%	valve	202	6	208	97%	3%
toplam	1514	850	2364	64,04%	35,96%	toplam	1937	427	2364	81,94%	18,06%

Çizelge 10.5 Üç kanalda 32 ve 64 görüntü boyutta 500 batchsize doğruluk çıktısı

3 Kanal						3 Kanal					
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	351	98	449	78%	22%	bottle	283	166	449	63%	37%
can	291	76	367	79%	21%	can	150	217	367	41%	59%
chain	126	100	226	56%	44%	chain	35	191	226	15%	85%
drink-carton	280	69	349	80%	20%	drink-carton	248	101	349	71%	29%
hook	87	46	133	65%	35%	hook	0	133	133	0%	100%
propeller	101	36	137	74%	26%	propeller	0	137	137	0%	100%
shampoo-bottle	56	43	99	57%	43%	shampoo-bottle	0	99	99	0%	100%
standing-bottle	22	43	65	34%	66%	standing-bottle	0	65	65	0%	100%
tire	315	16	331	95%	5%	tire	309	22	331	93%	7%
valve	202	6	208	97%	3%	valve	186	22	208	89%	11%
toplam	1831	533	2364	77,45%	22,55%	toplam	1211	1153	2364	51,23%	48,77%

Çizelge 10.6 Üç kanalda 92 görüntü boyutta 500 batchsize doğruluk çıktısı

3 Kanal					
92X92	Train	Loss	Toplam	%Train	%loss
bottle	0	449	449	0%	100%
can	0	367	367	0%	100%
chain	0	226	226	0%	100%
drink-carton	0	349	349	0%	100%
hook	0	133	133	0%	100%
propeller	0	137	137	0%	100%
shampoo-bottle	0	99	99	0%	100%
standing-bottle	0	65	65	0%	100%
tire	0	331	331	0%	100%
valve	0	208	208	0%	100%
toplam	0	2364	2364	0,00%	100,00%

Burada görüldüğü üzere 3 kanalda 500 batchsize en iyi size 28 çıktı ve size 92 olduğunda %100 kayıp gözüküyor.

Eğitimi yaparken 3 kanalda filtre uygulayarak batchsize 500 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar

Çizelge 10.7 Üç kanalda Filtre 16 ve 28 görüntü boyutta 500 batchsize doğruluk çıktısı

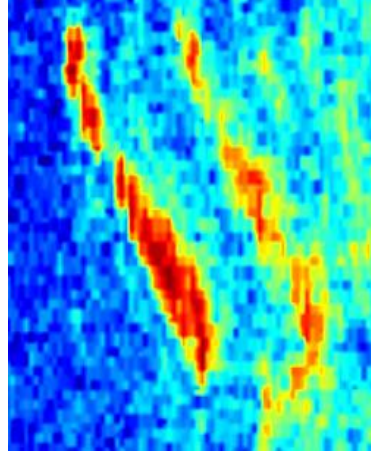
3F Kanal						3F Kanal					
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	273	176	449	61%	39%	bottle	342	107	449	76%	24%
can	129	238	367	35%	65%	can	316	51	367	86%	14%
chain	0	226	226	0%	100%	chain	63	163	226	28%	72%
drink-carton	267	82	349	77%	23%	drink-carton	289	60	349	83%	17%
hook	0	133	133	0%	100%	hook	61	72	133	46%	54%
propeller	20	117	137	15%	85%	propeller	112	25	137	82%	18%
shampoo-bottle	0	99	99	0%	100%	shampoo-bottle	21	78	99	21%	79%
standing-bottle	0	65	65	0%	100%	standing-bottle	0	65	65	0%	100%
tire	146	185	331	44%	56%	tire	252	79	331	76%	24%
valve	106	102	208	51%	49%	valve	187	21	208	90%	10%
toplam	941	1423	2364	39,81%	60,19%	toplam	1643	721	2364	69,50%	30,50%

Çizelge 10.8 Üç kanalda Filtre 32 ve 64 görüntü boyutta 500 batchsize doğruluk çıktısı

3F Kanal						3F Kanal					
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	350	99	449	78%	22%	bottle	259	190	449	58%	42%
can	283	84	367	77%	23%	can	5	362	367	1%	99%
chain	118	108	226	52%	48%	chain	0	226	226	0%	100%
drink-carton	292	57	349	84%	16%	drink-carton	214	135	349	61%	39%
hook	76	57	133	57%	43%	hook	0	133	133	0%	100%
propeller	101	36	137	74%	26%	propeller	0	137	137	0%	100%
shampoo-bottle	48	51	99	48%	52%	shampoo-bottle	0	99	99	0%	100%
standing-bottle	3	62	65	5%	95%	standing-bottle	0	65	65	0%	100%
tire	295	36	331	89%	11%	tire	7	324	331	2%	98%
valve	191	17	208	92%	8%	valve	1	207	208	0%	100%
toplam	1757	607	2364	74,32%	25,68%	toplam	486	1878	2364	20,56%	79,44%

Çizelge 10.9 Üç kanalda Filtre 92 görüntü boyutta 500 batchsize doğruluk çıktısı

3F Kanal					
92X92	Train	Loss	Toplam	%Train	%loss
bottle	326	123	449	73%	27%
can	129	238	367	35%	65%
chain	51	175	226	23%	77%
drink-carton	291	58	349	83%	17%
hook	3	130	133	2%	98%
propeller	13	124	137	9%	91%
shampoo-bottle	0	99	99	0%	100%
standing-bottle	0	65	65	0%	100%
tire	221	110	331	67%	33%
valve	171	37	208	82%	18%
toplam	1205	1159	2364	50,97%	49,03%



Şekil 10.1 Sonar görüntüye uygulanan filtre

Burada görüldüğü üzere 3 kanalda 500 batchsize en iyi size 32 çıktı. Ve üst tarafta görsele uygulana filtrenin çıktısı olarak bir örnek verilmektedir.

10.2 BATCHSIZE 300

Eğitimi yaparken ilk başta tek kanalda batchsize 300 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.10 Tek kanalda 16 ve 28 görüntü boyutta 300 batchsize doğruluk çıktısı

Tek Kanal	BatchSize300					Tek Kanal	BatchSize300					Tek Kanal
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss	32X32
bottle	330	119	449	73,50%	26,50%	bottle	412	37	449	91,76%	8,24%	bottle
can	285	82	367	77,66%	22,34%	can	332	35	367	90,46%	9,54%	can
chain	90	136	226	39,82%	60,18%	chain	130	96	226	57,52%	42,48%	chain
drink-carton	305	44	349	87,39%	12,61%	drink-carton	287	62	349	82,23%	17,77%	drink-carton
hook	77	56	133	57,89%	42,11%	hook	96	37	133	72,18%	27,82%	hook
propeller	91	46	137	66,42%	33,58%	propeller	126	11	137	91,97%	8,03%	propeller
shampoo-bottle	19	80	99	19,19%	80,81%	shampoo-bottle	60	39	99	60,61%	39,39%	shampoo-bottle
standing-bottle	6	59	65	9,23%	90,77%	standing-bottle	65	0	65	100,00%	0,00%	standing-bottle
tire	297	34	331	89,73%	10,27%	tire	314	17	331	94,86%	5,14%	tire
valve	196	12	208	94,23%	5,77%	valve	204	4	208	98,08%	1,92%	valve
toplam	1696	668	2364	71,74%	28,26%	toplam	2026	338	2364	85,70%	14,30%	toplam

Çizelge 10.11 Tablo Tek kanalda 32 ve 64 görüntü boyutta 300 batchsize doğruluk çıktısı

Tek Kanal	BatchSize300					Tek Kanal	BatchSize300				
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	394	55	449	87,75%	12,25%	bottle	426	23	449	94,88%	5,12%
can	336	31	367	91,55%	8,45%	can	324	43	367	88,28%	11,72%
chain	159	67	226	70,35%	29,65%	chain	192	34	226	84,96%	15,04%
drink-carton	294	55	349	84,24%	15,76%	drink-carton	278	71	349	79,66%	20,34%
hook	110	23	133	82,71%	17,29%	hook	94	39	133	70,68%	29,32%
propeller	128	9	137	93,43%	6,57%	propeller	124	13	137	90,51%	9,49%
shampoo-bottle	54	45	99	54,55%	45,45%	shampoo-bottle	53	46	99	53,54%	46,46%
standing-bottle	58	7	65	89,23%	10,77%	standing-bottle	15	50	65	23,08%	76,92%
tire	314	17	331	94,86%	5,14%	tire	298	33	331	90,03%	9,97%
valve	200	8	208	96,15%	3,85%	valve	194	14	208	93,27%	6,73%
toplam	2047	317	2364	86,59%	13,41%	toplam	1998	366	2364	84,52%	15,48%

Çizelge 10.12 Tek kanalda 92 görüntü boyutta 300 batchsize doğruluk çıktısı

Tek Kanal	BatchSize300				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	439	10	449	97,77%	2,23%
can	285	82	367	77,66%	22,34%
chain	158	68	226	69,91%	30,09%
drink-carton	338	11	349	96,85%	3,15%
hook	106	27	133	79,70%	20,30%
propeller	128	9	137	93,43%	6,57%
shampoo-bottle	73	26	99	73,74%	26,26%
standing-bottle	61	4	65	93,85%	6,15%
tire	322	9	331	97,28%	2,72%
valve	195	13	208	93,75%	6,25%
toplam	2105	259	2364	89,04%	10,96%

Tek kanalda 300 batchsize en verimli boyut 92 pixeldir . %98.04 doğruluk oranı vardır

Eğitimi yaparken ikinci olarak 3 kanalda batchsize 300 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.13 Üç kanalda 16 ve 28 görüntü boyutta 300 batchsize doğruluk çıktısı

3Channel	BatchSize300					3Channel	BatchSize300				
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	362	87	449	80,62%	19,38%	bottle	410	39	449	91,31%	8,69%
can	227	140	367	61,85%	38,15%	can	301	66	367	82,02%	17,98%
chain	112	114	226	49,56%	50,44%	chain	105	121	226	46,46%	53,54%
drink-carton	264	85	349	75,64%	24,36%	drink-carton	314	35	349	89,97%	10,03%
hook	10	123	133	7,52%	92,48%	hook	92	41	133	69,17%	30,83%
propeller	49	88	137	35,77%	64,23%	propeller	102	35	137	74,45%	25,55%
shampoo-bottle	0	99	99	0,00%	100,00%	shampoo-bottle	52	47	99	52,53%	47,47%
standing-bottle	4	61	65	6,15%	93,85%	standing-bottle	26	39	65	40,00%	60,00%
tire	283	48	331	85,50%	14,50%	tire	297	34	331	89,73%	10,27%
valve	181	27	208	87,02%	12,98%	valve	195	13	208	93,75%	6,25%
toplam	1492	872	2364	63,11%	36,89%	toplam	1894	470	2364	80,12%	19,88%

Çizelge 10.14 Üç kanalda 32 ve 64 görüntü boyutta 300 batchsize doğruluk çıktısı

3Channel	BatchSize300					3Channel	BatchSize300				
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	376	73	449	83,74%	16,26%	bottle	408	41	449	90,87%	9,13%
can	312	55	367	85,01%	14,99%	can	328	39	367	89,37%	10,63%
chain	189	37	226	83,63%	16,37%	chain	193	33	226	85,40%	14,60%
drink-carton	316	33	349	90,54%	9,46%	drink-carton	342	7	349	97,99%	2,01%
hook	76	57	133	57,14%	42,86%	hook	110	23	133	82,71%	17,29%
propeller	112	25	137	81,75%	18,25%	propeller	121	16	137	88,32%	11,68%
shampoo-bottle	62	37	99	62,63%	37,37%	shampoo-bottle	78	21	99	78,79%	21,21%
standing-bottle	43	22	65	66,15%	33,85%	standing-bottle	56	9	65	86,15%	13,85%
tire	293	38	331	88,52%	11,48%	tire	314	17	331	94,86%	5,14%
valve	199	9	208	95,67%	4,33%	valve	204	4	208	98,08%	1,92%
toplam	1978	386	2364	83,67%	16,33%	toplam	2154	210	2364	91,12%	8,88%

Çizelge 10.15 Üç kanalda 92 görüntü boyutta 300 batchsize doğruluk çıktısı

3Channel	BatchSize300				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	343	106	449	76,39%	23,61%
can	353	14	367	96,19%	3,81%
chain	179	47	226	79,20%	20,80%
drink-carton	319	30	349	91,40%	8,60%
hook	83	50	133	62,41%	37,59%
propeller	97	40	137	70,80%	29,20%
shampoo-bottle	63	36	99	63,64%	36,36%
standing-bottle	37	28	65	56,92%	43,08%
tire	307	24	331	92,75%	7,25%
valve	198	10	208	95,19%	4,81%
toplam	1979	385	2364	83,71%	16,29%

3 kanalda 300 batchsize en verimli boyut 64 pixeldir . %91.12 oranında doğruluk payı vardır.

Eğitimi yaparken 3 kanalda filtre uygulayarak batchsize 300 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.16 Üç kanalda Filtre 16 ve 28 görüntü boyutta 300 batchsize doğruluk çıktısı

3fChannel	BatchSize300					3fChannel	BatchSize300				
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	411	38	449	91,54%	8,46%	bottle	443	6	449	98,66%	1,34%
can	306	61	367	83,38%	16,62%	can	352	15	367	95,91%	4,09%
chain	114	112	226	50,44%	49,56%	chain	194	32	226	85,84%	14,16%
drink-carton	320	29	349	91,69%	8,31%	drink-carton	341	8	349	97,71%	2,29%
hook	85	48	133	63,91%	36,09%	hook	108	25	133	81,20%	18,80%
propeller	91	46	137	66,42%	33,58%	propeller	123	14	137	89,78%	10,22%
shampoo-bottle	44	55	99	44,44%	55,56%	shampoo-bottle	88	11	99	88,89%	11,11%
standing-bottle	38	27	65	58,46%	41,54%	standing-bottle	64	1	65	98,46%	1,54%
tire	311	20	331	93,96%	6,04%	tire	326	5	331	98,49%	1,51%
valve	199	9	208	95,67%	4,33%	valve	207	1	208	99,52%	0,48%
toplam	1919	445	2364	81,18%	18,82%	toplam	2246	118	2364	95,01%	4,99%

Çizelge 10.17 Üç kanalda Filtre 32 ve 64 görüntü boyutta 300 batchsize doğruluk çıktısı

3fChannel	BatchSize300					3fChannel	BatchSize300				
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	405	44	449	90,20%	9,80%	bottle	288	161	449	64,14%	35,86%
can	354	13	367	96,46%	3,54%	can	164	203	367	44,69%	55,31%
chain	172	54	226	76,11%	23,89%	chain	113	113	226	50,00%	50,00%
drink-carton	310	39	349	88,83%	11,17%	drink-carton	231	118	349	66,19%	33,81%
hook	97	36	133	72,93%	27,07%	hook	0	133	133	0,00%	100,00%
propeller	122	15	137	89,05%	10,95%	propeller	0	137	137	0,00%	100,00%
shampoo-bottle	79	20	99	79,80%	20,20%	shampoo-bottle	0	99	99	0,00%	100,00%
standing-bottle	58	7	65	89,23%	10,77%	standing-bottle	0	65	65	0,00%	100,00%
tire	318	13	331	96,07%	3,93%	tire	314	17	331	94,86%	5,14%
valve	199	9	208	95,67%	4,33%	valve	0	208	208	0,00%	100,00%
toplam	2114	250	2364	89,42%	10,58%	toplam	1110	1254	2364	46,95%	53,05%

Çizelge 10.18 Üç kanalda Filtre 92 görüntü boyutta 300 batchsize doğruluk çıktısı

3fChannel	BatchSize300				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	402	47	449	89,53%	10,47%
can	297	70	367	80,93%	19,07%
chain	148	78	226	65,49%	34,51%
drink-carton	288	61	349	82,52%	17,48%
hook	106	27	133	79,70%	20,30%
propeller	87	50	137	63,50%	36,50%
shampoo-bottle	59	40	99	59,60%	40,40%
standing-bottle	46	19	65	70,77%	29,23%
tire	328	3	331	99,09%	0,91%
valve	200	8	208	96,15%	3,85%
toplam	1961	403	2364	82,95%	17,05%

Üç kanalda filtreli görsellerde 300 batchsize en verimli boyut 28 pixeldir .
%95,02 oranında doğruluk payı vardır.

10.3 BATCHSİZE 150

Eğitimi yaparken ilk başta tek kanalda batchsize 150 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.19 Tek kanalda 16 ve 28 görüntü boyutta 150 batchsize doğruluk çıktısı

Tek Kanal	BatchSize150					Tek Kanal	BatchSize150				
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	407	42	449	90,65%	9,35%	bottle	412	37	449	91,76%	8,24%
can	291	76	367	79,29%	20,71%	can	333	34	367	90,74%	9,26%
chain	73	153	226	32,30%	67,70%	chain	142	84	226	62,83%	37,17%
drink-carton	317	32	349	90,83%	9,17%	drink-carton	332	17	349	95,13%	4,87%
hook	93	40	133	69,92%	30,08%	hook	115	18	133	86,47%	13,53%
propeller	117	20	137	85,40%	14,60%	propeller	128	9	137	93,43%	6,57%
shampoo-bottle	37	62	99	37,37%	62,63%	shampoo-bottle	77	22	99	77,78%	22,22%
standing-bottle	40	25	65	61,54%	38,46%	standing-bottle	56	9	65	86,15%	13,85%
tire	323	8	331	97,58%	2,42%	tire	319	12	331	96,37%	3,63%
valve	196	12	208	94,23%	5,77%	valve	204	4	208	98,08%	1,92%
toplam	1894	470	2364	80,12%	19,88%	toplam	2118	246	2364	89,59%	10,41%

Çizelge 10.20 Tek kanalda 32 ve 64 görüntü boyutta 150 batchsize doğruluk çıktısı

Tek Kanal	BatchSize150					Tek Kanal	BatchSize150				
	Train	Loss	Toplam	%Train	%loss		Train	Loss	Toplam	%Train	%loss
32X32						64X64					
bottle	405	44	449	90,20%	9,80%	bottle	446	3	449	99,33%	0,67%
can	361	6	367	98,37%	1,63%	can	360	7	367	98,09%	1,91%
chain	176	50	226	77,88%	22,12%	chain	210	16	226	92,92%	7,08%
drink-carton	329	20	349	94,27%	5,73%	drink-carton	344	5	349	98,57%	1,43%
hook	113	20	133	84,96%	15,04%	hook	123	10	133	92,48%	7,52%
propeller	131	6	137	95,62%	4,38%	propeller	133	4	137	97,08%	2,92%
shampoo-bottle	73	26	99	73,74%	26,26%	shampoo-bottle	98	1	99	98,99%	1,01%
standing-bottle	62	3	65	95,38%	4,62%	standing-bottle	64	1	65	98,46%	1,54%
tire	324	7	331	97,89%	2,11%	tire	330	1	331	99,70%	0,30%
valve	203	5	208	97,60%	2,40%	valve	207	1	208	99,52%	0,48%
toplam	2177	187	2364	92,09%	7,91%	toplam	2315	49	2364	97,93%	2,07%

Çizelge 10.21 Tablo Tek kanalda 92 görüntü boyutta 150 batchsize doğruluk çıktısı .

Tek Kanal	BatchSize150				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	446	3	449	99,33%	0,67%
can	362	5	367	98,64%	1,36%
chain	213	13	226	94,25%	5,75%
drink-carton	344	5	349	98,57%	1,43%
hook	128	5	133	96,24%	3,76%
propeller	135	2	137	98,54%	1,46%
shampoo-bottle	89	10	99	89,90%	10,10%
standing-bottle	65	0	65	100,00%	0,00%
tire	330	1	331	99,70%	0,30%
valve	206	2	208	99,04%	0,96%
toplam	2318	46	2364	98,05%	1,95%

Tek kanalda filtreli görsellerde 150 batchsize en verimli boyut 92 pixeldir . %98,05 oranında doğruluk payı vardır.

Eğitimi yaparken ikinci olarak 3 kanalda batchsize 150 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.22 Üç kanalda 16 ve 28 görüntü boyutta 150 batchsize doğruluk çıktısı

3Channel	BatchSize150					3Channel	BatchSize150				
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	399	50	449	88,86%	11,14%	bottle	433	16	449	96,44%	3,56%
can	316	51	367	86,10%	13,90%	can	323	44	367	88,01%	11,99%
chain	108	118	226	47,79%	52,21%	chain	210	16	226	92,92%	7,08%
drink-carton	322	27	349	92,26%	7,74%	drink-carton	325	24	349	93,12%	6,88%
hook	68	65	133	51,13%	48,87%	hook	110	23	133	82,71%	17,29%
propeller	87	50	137	63,50%	36,50%	propeller	107	30	137	78,10%	21,90%
shampoo-bottle	54	45	99	54,55%	45,45%	shampoo-bottle	65	34	99	65,66%	34,34%
standing-bottle	31	34	65	47,69%	52,31%	standing-bottle	51	14	65	78,46%	21,54%
tire	304	27	331	91,84%	8,16%	tire	325	6	331	98,19%	1,81%
valve	197	11	208	94,71%	5,29%	valve	202	6	208	97,12%	2,88%
toplam	1886	478	2364	79,78%	20,22%	toplam	2151	213	2364	90,99%	9,01%

Çizelge 10.23 Üç kanalda 32 ve 64 görüntü boyutta 150 batchsize doğruluk çıktısı

3Channel	BatchSize150					3Channel	BatchSize150				
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	384	65	449	85,52%	14,48%	bottle	448	1	449	99,78%	0,22%
can	363	4	367	98,91%	1,09%	can	353	14	367	96,19%	3,81%
chain	182	44	226	80,53%	19,47%	chain	219	7	226	96,90%	3,10%
drink-carton	336	13	349	96,28%	3,72%	drink-carton	348	1	349	99,71%	0,29%
hook	113	20	133	84,96%	15,04%	hook	130	3	133	97,74%	2,26%
propeller	123	14	137	89,78%	10,22%	propeller	124	13	137	90,51%	9,49%
shampoo-bottle	90	9	99	90,91%	9,09%	shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	48	17	65	73,85%	26,15%	standing-bottle	65	0	65	100,00%	0,00%
tire	299	32	331	90,33%	9,67%	tire	327	4	331	98,79%	1,21%
valve	195	13	208	93,75%	6,25%	valve	207	1	208	99,52%	0,48%
toplam	2133	231	2364	90,23%	9,77%	toplam	2320	44	2364	98,14%	1,86%

Çizelge 10.24 Üç kanalda 92 görüntü boyutta 150 batchsize doğruluk çıktısı

3Channel	BatchSize150				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	382	67	449	85,08%	14,92%
can	275	92	367	74,93%	25,07%
chain	135	91	226	59,73%	40,27%
drink-carton	319	30	349	91,40%	8,60%
hook	55	78	133	41,35%	58,65%
propeller	93	44	137	67,88%	32,12%
shampoo-bottle	40	59	99	40,40%	59,60%
standing-bottle	60	5	65	92,31%	7,69%
tire	288	43	331	87,01%	12,99%
valve	190	18	208	91,35%	8,65%
toplam	1837	527	2364	77,71%	22,29%

3 kanalda filtreli görsellerde 150 batchsize en verimli boyut 64 pixeldir .
%98,14 oranında doğruluk payı vardır.

Eğitimi yaparken 3 kanalda filtre uygulayarak batchsize 150 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.25 Üç kanalda filtre 16 ve 28 görüntü boyutta 150 batchsize doğruluk çıktısı

3fChannel	BatchSize150					3fChannel	BatchSize150				
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	431	18	449	95,99%	4,01%	bottle	439	10	449	97,77%	2,23%
can	321	46	367	87,47%	12,53%	can	359	8	367	97,82%	2,18%
chain	178	48	226	78,76%	21,24%	chain	209	17	226	92,48%	7,52%
drink-carton	316	33	349	90,54%	9,46%	drink-carton	347	2	349	99,43%	0,57%
hook	101	32	133	75,94%	24,06%	hook	123	10	133	92,48%	7,52%
propeller	119	18	137	86,86%	13,14%	propeller	134	3	137	97,81%	2,19%
shampoo-bottle	92	7	99	92,93%	7,07%	shampoo-bottle	53	46	99	53,54%	46,46%
standing-bottle	61	4	65	93,85%	6,15%	standing-bottle	65	0	65	100,00%	0,00%
tire	321	10	331	96,98%	3,02%	tire	328	3	331	99,09%	0,91%
valve	205	3	208	98,56%	1,44%	valve	204	4	208	98,08%	1,92%
toplam	2145	219	2364	90,74%	9,26%	toplam	2261	103	2364	95,64%	4,36%

Çizelge 10.26 Üç kanalda filtre 32 ve 64 görüntü boyutta 150 batchsize doğruluk çıktısı

3fChannel	BatchSize150					3fChannel	BatchSize150				
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	448	1	449	99,78%	0,22%	bottle	438	11	449	97,55%	2,45%
can	357	10	367	97,28%	2,72%	can	355	12	367	96,73%	3,27%
chain	217	9	226	96,02%	3,98%	chain	214	12	226	94,69%	5,31%
drink-carton	344	5	349	98,57%	1,43%	drink-carton	343	6	349	98,28%	1,72%
hook	129	4	133	96,99%	3,01%	hook	116	17	133	87,22%	12,78%
propeller	137	0	137	100,00%	0,00%	propeller	128	9	137	93,43%	6,57%
shampoo-bottle	95	4	99	95,96%	4,04%	shampoo-bottle	88	11	99	88,89%	11,11%
standing-bottle	65	0	65	100,00%	0,00%	standing-bottle	62	3	65	95,38%	4,62%
tire	331	0	331	100,00%	0,00%	tire	327	4	331	98,79%	1,21%
valve	207	1	208	99,52%	0,48%	valve	207	1	208	99,52%	0,48%
toplam	2330	34	2364	98,56%	1,44%	toplam	2278	86	2364	96,36%	3,64%

Çizelge 10.27 Üç kanalda filtre 92 görüntü boyutta 150 batchsize doğruluk çıktısı

3fChannel	BatchSize150				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	446	3	449	99,33%	0,67%
can	349	18	367	95,10%	4,90%
chain	219	7	226	96,90%	3,10%
drink-carton	332	17	349	95,13%	4,87%
hook	129	4	133	96,99%	3,01%
propeller	129	8	137	94,16%	5,84%
shampoo-bottle	92	7	99	92,93%	7,07%
standing-bottle	64	1	65	98,46%	1,54%
tire	329	2	331	99,40%	0,60%
valve	207	1	208	99,52%	0,48%
toplam	2296	68	2364	97,12%	2,88%

3 kanalda filtreli görsellerde 150 batchsize en verimli boyut 32 pixeldir .
%98,56 oranında doğruluk payı vardır.

10.4 BATCHSIZE 75

Eğitimi yaparken ilk başta tek kanalda batchsize 75 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.28 Tek kanalda 16 ve 28 görüntü boyutta 75 batchsize doğruluk çıktısı

Tek Kanal	BatchSize75					Tek Kanal	BatchSize75					Tek Kanal
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss	32X32
bottle	416	33	449	92,65%	7,35%	bottle	438	11	449	97,55%	2,45%	bottle
can	332	35	367	90,46%	9,54%	can	353	14	367	96,19%	3,81%	can
chain	148	78	226	65,49%	34,51%	chain	209	17	226	92,48%	7,52%	chain
drink-carton	338	11	349	96,85%	3,15%	drink-carton	337	12	349	96,56%	3,44%	drink-carton
hook	99	34	133	74,44%	25,56%	hook	125	8	133	93,98%	6,02%	hook
propeller	127	10	137	92,70%	7,30%	propeller	131	6	137	95,62%	4,38%	propeller
shampoo-bottle	62	37	99	62,63%	37,37%	shampoo-bottle	95	4	99	95,96%	4,04%	shampoo-bottle
standing-bottle	65	0	65	100,00%	0,00%	standing-bottle	60	5	65	92,31%	7,69%	standing-bottle
tire	313	18	331	94,56%	5,44%	tire	325	6	331	98,19%	1,81%	tire
valve	206	2	208	99,04%	0,96%	valve	207	1	208	99,52%	0,48%	valve
toplam	2106	258	2364	89,09%	10,91%	toplam	2280	84	2364	96,45%	3,55%	toplam

Çizelge 10.29 Tek kanalda 32 ve 64 görüntü boyutta 75 batchsize doğruluk çıktısı

BatchSize75					Tek Kanal	BatchSize75				
Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
442	7	449	98,44%	1,56%	bottle	448	1	449	99,78%	0,22%
358	9	367	97,55%	2,45%	can	364	3	367	99,18%	0,82%
214	12	226	94,69%	5,31%	chain	222	4	226	98,23%	1,77%
339	10	349	97,13%	2,87%	drink-carton	346	3	349	99,14%	0,86%
118	15	133	88,72%	11,28%	hook	133	0	133	100,00%	0,00%
131	6	137	95,62%	4,38%	propeller	137	0	137	100,00%	0,00%
83	16	99	83,84%	16,16%	shampoo-bottle	94	5	99	94,95%	5,05%
65	0	65	100,00%	0,00%	standing-bottle	65	0	65	100,00%	0,00%
322	9	331	97,28%	2,72%	tire	330	1	331	99,70%	0,30%
194	14	208	93,27%	6,73%	valve	208	0	208	100,00%	0,00%
2266	98	2364	95,85%	4,15%	toplam	2347	17	2364	99,28%	0,72%

Çizelge 10.30 Tek kanalda 92 görüntü boyutta 75 batchsize doğruluk çıktısı

Tek Kanal	BatchSize75				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	446	3	449	99,33%	0,67%
can	365	2	367	99,46%	0,54%
chain	225	1	226	99,56%	0,44%
drink-carton	345	4	349	98,85%	1,15%
hook	131	2	133	98,50%	1,50%
propeller	137	0	137	100,00%	0,00%
shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	65	0	65	100,00%	0,00%
tire	330	1	331	99,70%	0,30%
valve	208	0	208	100,00%	0,00%
toplam	2351	13	2364	99,45%	0,55%

Tek kanalda 75 batchsize en verimli boyut 92 pixeldir . %99,45 doğruluk oranı vardır

Eğitimi yaparken ikinci olarak 3 kanalda batchsize 75 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.31 Üç kanalda 16 ve 28 görüntü boyutta 75 batchsize doğruluk çıktısı

3Channel	BatchSize75					3Channel	BatchSize75				
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	437	12	449	97,33%	2,67%	bottle	446	3	449	99,33%	0,67%
can	292	75	367	79,56%	20,44%	can	355	12	367	96,73%	3,27%
chain	202	24	226	89,38%	10,62%	chain	212	14	226	93,81%	6,19%
drink-carton	322	27	349	92,26%	7,74%	drink-carton	348	1	349	99,71%	0,29%
hook	86	47	133	64,66%	35,34%	hook	127	6	133	95,49%	4,51%
propeller	121	16	137	88,32%	11,68%	propeller	127	10	137	92,70%	7,30%
shampoo-bottle	75	24	99	75,76%	24,24%	shampoo-bottle	98	1	99	98,99%	1,01%
standing-bottle	56	9	65	86,15%	13,85%	standing-bottle	64	1	65	98,46%	1,54%
tire	323	8	331	97,58%	2,42%	tire	321	10	331	96,98%	3,02%
valve	203	5	208	97,60%	2,40%	valve	206	2	208	99,04%	0,96%
toplamlam	2117	247	2364	89,55%	10,45%	toplamlam	2304	60	2364	97,46%	2,54%

Çizelge 10.32 Tablo Tek kanalda 32 ve 64 görüntü boyutta 75 batchsize doğruluk çıktısı

3Channel	BatchSize75					3Channel	BatchSize75				
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	427	22	449	95,10%	4,90%	bottle	445	4	449	99,11%	0,89%
can	361	6	367	98,37%	1,63%	can	357	10	367	97,28%	2,72%
chain	214	12	226	94,69%	5,31%	chain	226	0	226	100,00%	0,00%
drink-carton	349	0	349	100,00%	0,00%	drink-carton	348	1	349	99,71%	0,29%
hook	123	10	133	92,48%	7,52%	hook	132	1	133	99,25%	0,75%
propeller	127	10	137	92,70%	7,30%	propeller	134	3	137	97,81%	2,19%
shampoo-bottle	98	1	99	98,99%	1,01%	shampoo-bottle	84	15	99	84,85%	15,15%
standing-bottle	62	3	65	95,38%	4,62%	standing-bottle	65	0	65	100,00%	0,00%
tire	325	6	331	98,19%	1,81%	tire	331	0	331	100,00%	0,00%
valve	204	4	208	98,08%	1,92%	valve	207	1	208	99,52%	0,48%
toplamlam	2290	74	2364	96,87%	3,13%	toplamlam	2329	35	2364	98,52%	1,48%

Çizelge 10.33 Tablo Tek kanalda 92 görüntü boyutta 75 batchsize doğruluk çıktısı

3Channel	BatchSize75				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	447	2	449	99,55%	0,45%
can	342	25	367	93,19%	6,81%
chain	225	1	226	99,56%	0,44%
drink-carton	348	1	349	99,71%	0,29%
hook	133	0	133	100,00%	0,00%
propeller	128	9	137	93,43%	6,57%
shampoo-bottle	98	1	99	98,99%	1,01%
standing-bottle	65	0	65	100,00%	0,00%
tire	325	6	331	98,19%	1,81%
valve	205	3	208	98,56%	1,44%
toplamlam	2316	48	2364	97,97%	2,03%

3 kanalda 75 batchsize en verimli boyut 64 pixeldir . %98.52 oranında doğruluk payı vardır.

Eğitimi yaparken 3 kanalda filtre uygulayarak batchsize 75 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.34 Üç kanalda filtre 16 ve 28 görüntü boyutta 75 batchsize doğruluk çıktısı

3fChannel	BatchSize75					3fChannel	BatchSize75				
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	437	12	449	97,33%	2,67%	bottle	449	0	449	100,00%	0,00%
can	353	14	367	96,19%	3,81%	can	362	5	367	98,64%	1,36%
chain	199	27	226	88,05%	11,95%	chain	219	7	226	96,90%	3,10%
drink-carton	333	16	349	95,42%	4,58%	drink-carton	344	5	349	98,57%	1,43%
hook	128	5	133	96,24%	3,76%	hook	132	1	133	99,25%	0,75%
propeller	120	17	137	87,59%	12,41%	propeller	133	4	137	97,08%	2,92%
shampoo-bottle	89	10	99	89,90%	10,10%	shampoo-bottle	98	1	99	98,99%	1,01%
standing-bottle	65	0	65	100,00%	0,00%	standing-bottle	65	0	65	100,00%	0,00%
tire	324	7	331	97,89%	2,11%	tire	331	0	331	100,00%	0,00%
valve	203	5	208	97,60%	2,40%	valve	207	1	208	99,52%	0,48%
toplam	2251	113	2364	95,22%	4,78%	toplam	2340	24	2364	98,98%	1,02%

Çizelge 10.35 Üç kanalda filtre 32 ve 64 görüntü boyutta 75 batchsize doğruluk çıktısı

3fChannel	BatchSize75					3fChannel	BatchSize75				
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	448	1	449	99,78%	0,22%	bottle	449	0	449	100,00%	0,00%
can	366	1	367	99,73%	0,27%	can	365	2	367	99,46%	0,54%
chain	217	9	226	96,02%	3,98%	chain	213	13	226	94,25%	5,75%
drink-carton	347	2	349	99,43%	0,57%	drink-carton	349	0	349	100,00%	0,00%
hook	131	2	133	98,50%	1,50%	hook	132	1	133	99,25%	0,75%
propeller	136	1	137	99,27%	0,73%	propeller	135	2	137	98,54%	1,46%
shampoo-bottle	98	1	99	98,99%	1,01%	shampoo-bottle	98	1	99	98,99%	1,01%
standing-bottle	65	0	65	100,00%	0,00%	standing-bottle	65	0	65	100,00%	0,00%
tire	325	6	331	98,19%	1,81%	tire	327	4	331	98,79%	1,21%
valve	207	1	208	99,52%	0,48%	valve	205	3	208	98,56%	1,44%
toplam	2340	24	2364	98,98%	1,02%	toplam	2338	26	2364	98,90%	1,10%

Çizelge 10.36 Üç kanalda filtre 92 görüntü boyutta 75 batchsize doğruluk çıktısı

3fChannel	BatchSize75				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	449	0	449	100,00%	0,00%
can	367	0	367	100,00%	0,00%
chain	222	4	226	98,23%	1,77%
drink-carton	347	2	349	99,43%	0,57%
hook	132	1	133	99,25%	0,75%
propeller	137	0	137	100,00%	0,00%
shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	65	0	65	100,00%	0,00%
tire	331	0	331	100,00%	0,00%
valve	208	0	208	100,00%	0,00%
toplam	2357	7	2364	99,70%	0,30%

3 kanalda filtreli görsellerde 300 batchsize en verimli boyut 92 pixeldir .
%99,70 oranında doğruluk payı vardır.

10.5 BATCHSİZE 50

Eğitimi yaparken ilk başta tek kanalda batchsize 50 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.37 Tek kanalda 16 ve 28 görüntü boyutta 50 batchsize doğruluk çıktısı

Tek Kanal	BatchSize50					Tek Kanal	BatchSize50				
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	428	21	449	95,32%	4,68%	bottle	434	15	449	96,66%	3,34%
can	348	19	367	94,82%	5,18%	can	365	2	367	99,46%	0,54%
chain	165	61	226	73,01%	26,99%	chain	215	11	226	95,13%	4,87%
drink-carton	338	11	349	96,85%	3,15%	drink-carton	345	4	349	98,85%	1,15%
hook	112	21	133	84,21%	15,79%	hook	129	4	133	96,99%	3,01%
propeller	129	8	137	94,16%	5,84%	propeller	134	3	137	97,81%	2,19%
shampoo-bottle	81	18	99	81,82%	18,18%	shampoo-bottle	90	9	99	90,91%	9,09%
standing-bottle	64	1	65	98,46%	1,54%	standing-bottle	64	1	65	98,46%	1,54%
tire	322	9	331	97,28%	2,72%	tire	323	8	331	97,58%	2,42%
valve	205	3	208	98,56%	1,44%	valve	207	1	208	99,52%	0,48%
toplam	2192	172	2364	92,72%	7,28%	toplam	2306	58	2364	97,55%	2,45%

Çizelge 10.38 Tek kanalda 32 ve 64 görüntü boyutta 50 batchsize doğruluk çıktısı

Tek Kanal	BatchSize50					Tek Kanal	BatchSize50				
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	444	5	449	98,89%	1,11%	bottle	449	0	449	100,00%	0,00%
can	364	3	367	99,18%	0,82%	can	366	1	367	99,73%	0,27%
chain	219	7	226	96,90%	3,10%	chain	222	4	226	98,23%	1,77%
drink-carton	347	2	349	99,43%	0,57%	drink-carton	346	3	349	99,14%	0,86%
hook	127	6	133	95,49%	4,51%	hook	133	0	133	100,00%	0,00%
propeller	137	0	137	100,00%	0,00%	propeller	136	1	137	99,27%	0,73%
shampoo-bottle	95	4	99	95,96%	4,04%	shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	64	1	65	98,46%	1,54%	standing-bottle	65	0	65	100,00%	0,00%
tire	328	3	331	99,09%	0,91%	tire	331	0	331	100,00%	0,00%
valve	206	2	208	99,04%	0,96%	valve	208	0	208	100,00%	0,00%
toplam	2331	33	2364	98,60%	1,40%	toplam	2355	9	2364	99,62%	0,38%

Çizelge 10.39 Tek kanalda 92 görüntü boyutta 50 batchsize doğruluk çıktısı

Tek Kanal	BatchSize50				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	448	1	449	99,78%	0,22%
can	363	4	367	98,91%	1,09%
chain	224	2	226	99,12%	0,88%
drink-carton	347	2	349	99,43%	0,57%
hook	133	0	133	100,00%	0,00%
propeller	137	0	137	100,00%	0,00%
shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	64	1	65	98,46%	1,54%
tire	331	0	331	100,00%	0,00%
valve	208	0	208	100,00%	0,00%
toplam	2354	10	2364	99,58%	0,42%

Tek kanalda 50 batchsize en verimli boyut 64 pixeldir . %99.62 doğruluk oranı vardır

Eğitimi yaparken ikinci olarak 3 kanalda batchsize 50 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.40 Üç kanalda 16 ve 28 görüntü boyutta 50 batchsize doğruluk çıktısı

3Channel	BatchSize50					3Channel	BatchSize50				
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	444	5	449	98,89%	1,11%	bottle	445	4	449	99,11%	0,89%
can	319	48	367	86,92%	13,08%	can	366	1	367	99,73%	0,27%
chain	199	27	226	88,05%	11,95%	chain	222	4	226	98,23%	1,77%
drink-carton	340	9	349	97,42%	2,58%	drink-carton	349	0	349	100,00%	0,00%
hook	130	3	133	97,74%	2,26%	hook	133	0	133	100,00%	0,00%
propeller	126	11	137	91,97%	8,03%	propeller	131	6	137	95,62%	4,38%
shampoo-bottle	82	17	99	82,83%	17,17%	shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	63	2	65	96,92%	3,08%	standing-bottle	65	0	65	100,00%	0,00%
tire	323	8	331	97,58%	2,42%	tire	328	3	331	99,09%	0,91%
valve	206	2	208	99,04%	0,96%	valve	208	0	208	100,00%	0,00%
toplam	2232	132	2364	94,42%	5,58%	toplam	2346	18	2364	99,24%	0,76%

Çizelge 10.41 Üç kanalda 32 ve 64 görüntü boyutta 50 batchsize doğruluk çıktısı

3Channel	BatchSize50					3Channel	BatchSize50				
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	446	3	449	99,33%	0,67%	bottle	449	0	449	100,00%	0,00%
can	359	8	367	97,82%	2,18%	can	363	4	367	98,91%	1,09%
chain	222	4	226	98,23%	1,77%	chain	224	2	226	99,12%	0,88%
drink-carton	347	2	349	99,43%	0,57%	drink-carton	349	0	349	100,00%	0,00%
hook	131	2	133	98,50%	1,50%	hook	133	0	133	100,00%	0,00%
propeller	135	2	137	98,54%	1,46%	propeller	136	1	137	99,27%	0,73%
shampoo-bottle	99	0	99	100,00%	0,00%	shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	65	0	65	100,00%	0,00%	standing-bottle	65	0	65	100,00%	0,00%
tire	329	2	331	99,40%	0,60%	tire	331	0	331	100,00%	0,00%
valve	205	3	208	98,56%	1,44%	valve	208	0	208	100,00%	0,00%
toplam	2338	26	2364	98,90%	1,10%	toplam	2357	7	2364	99,70%	0,30%

Çizelge 10.42 Üç kanalda 92 görüntü boyutta 50 batchsize doğruluk çıktısı

3Channel	BatchSize50				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	447	2	449	99,55%	0,45%
can	364	3	367	99,18%	0,82%
chain	224	2	226	99,12%	0,88%
drink-carton	348	1	349	99,71%	0,29%
hook	131	2	133	98,50%	1,50%
propeller	136	1	137	99,27%	0,73%
shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	65	0	65	100,00%	0,00%
tire	330	1	331	99,70%	0,30%
valve	207	1	208	99,52%	0,48%
toplam	2351	13	2364	99,45%	0,55%

3 kanalda 50 batchsize en verimli boyut 64 pixeldir . %99.70 oranında doğruluk payı vardır.

Eğitimi yaparken 3 kanalda filtre uygulayarak batchsize 50 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.43 Üç kanalda filtre 16 ve 28 görüntü boyutta 50 batchsize doğruluk çıktısı

3fChannel	BatchSize50					3fChannel	BatchSize50				
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	443	6	449	98,66%	1,34%	bottle	448	1	449	99,78%	0,22%
can	360	7	367	98,09%	1,91%	can	361	6	367	98,37%	1,63%
chain	210	16	226	92,92%	7,08%	chain	224	2	226	99,12%	0,88%
drink-carton	345	4	349	98,85%	1,15%	drink-carton	349	0	349	100,00%	0,00%
hook	127	6	133	95,49%	4,51%	hook	132	1	133	99,25%	0,75%
propeller	131	6	137	95,62%	4,38%	propeller	137	0	137	100,00%	0,00%
shampoo-bottle	95	4	99	95,96%	4,04%	shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	65	0	65	100,00%	0,00%	standing-bottle	65	0	65	100,00%	0,00%
tire	329	2	331	99,40%	0,60%	tire	331	0	331	100,00%	0,00%
valve	206	2	208	99,04%	0,96%	valve	208	0	208	100,00%	0,00%
toplam	2311	53	2364	97,76%	2,24%	toplam	2354	10	2364	99,58%	0,42%

Çizelge 10.44 Tablo 3 kanalda filtre 32 ve 64 görüntü boyutta 50 batchsize doğruluk çıktısı

3fChannel	BatchSize50					3fChannel	BatchSize50				
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	449	0	449	100,00%	0,00%	bottle	447	2	449	99,55%	0,45%
can	365	2	367	99,46%	0,54%	can	366	1	367	99,73%	0,27%
chain	219	7	226	96,90%	3,10%	chain	226	0	226	100,00%	0,00%
drink-carton	349	0	349	100,00%	0,00%	drink-carton	348	1	349	99,71%	0,29%
hook	133	0	133	100,00%	0,00%	hook	133	0	133	100,00%	0,00%
propeller	135	2	137	98,54%	1,46%	propeller	137	0	137	100,00%	0,00%
shampoo-bottle	99	0	99	100,00%	0,00%	shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	65	0	65	100,00%	0,00%	standing-bottle	65	0	65	100,00%	0,00%
tire	331	0	331	100,00%	0,00%	tire	331	0	331	100,00%	0,00%
valve	207	1	208	99,52%	0,48%	valve	208	0	208	100,00%	0,00%
toplam	2352	12	2364	99,49%	0,51%	toplam	2360	4	2364	99,83%	0,17%

Çizelge 10.45 Üç kanalda filtre 92 görüntü boyutta 50 batchsize doğruluk çıktısı

3fChannel	BatchSize50				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	447	2	449	99,55%	0,45%
can	361	6	367	98,37%	1,63%
chain	220	6	226	97,35%	2,65%
drink-carton	347	2	349	99,43%	0,57%
hook	130	3	133	97,74%	2,26%
propeller	136	1	137	99,27%	0,73%
shampoo-bottle	91	8	99	91,92%	8,08%
standing-bottle	64	1	65	98,46%	1,54%
tire	328	3	331	99,09%	0,91%
valve	205	3	208	98,56%	1,44%
toplam	2329	35	2364	98,52%	1,48%

3 kanalda filtreli görsellerde 50 batchsize en verimli boyut 28 pixeldir . %98,83 oranında doğruluk payı vardır.

10.6 BATCHSİZE 15

Eğitimi yaparken ilk başta tek kanalda batchsize 15 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.46 Tek kanalda 16 ve 28 görüntü boyutta 15 batchsize doğruluk çıktısı

Tek Kanal	BatchSize15					Tek Kanal	BatchSize15				
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	448	1	449	99,78%	0,22%	bottle	449	0	449	100,00%	0,00%
can	352	15	367	95,91%	4,09%	can	363	4	367	98,91%	1,09%
chain	217	9	226	96,02%	3,98%	chain	221	5	226	97,79%	2,21%
drink-carton	344	5	349	98,57%	1,43%	drink-carton	349	0	349	100,00%	0,00%
hook	119	14	133	89,47%	10,53%	hook	132	1	133	99,25%	0,75%
propeller	131	6	137	95,62%	4,38%	propeller	136	1	137	99,27%	0,73%
shampoo-bottle	97	2	99	97,98%	2,02%	shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	65	0	65	100,00%	0,00%	standing-bottle	65	0	65	100,00%	0,00%
tire	328	3	331	99,09%	0,91%	tire	331	0	331	100,00%	0,00%
valve	207	1	208	99,52%	0,48%	valve	207	1	208	99,52%	0,48%
toplam	2308	56	2364	97,63%	2,37%	toplam	2352	12	2364	99,49%	0,51%

Çizelge 10.47 Tek kanalda 32 ve 64 görüntü boyutta 15 batchsize doğruluk çıktısı

Tek Kanal	BatchSize15					Tek Kanal	BatchSize15				
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	448	1	449	99,78%	0,22%	bottle	448	1	449	99,78%	0,22%
can	365	2	367	99,46%	0,54%	can	366	1	367	99,73%	0,27%
chain	225	1	226	99,56%	0,44%	chain	223	3	226	98,67%	1,33%
drink-carton	349	0	349	100,00%	0,00%	drink-carton	349	0	349	100,00%	0,00%
hook	132	1	133	99,25%	0,75%	hook	133	0	133	100,00%	0,00%
propeller	137	0	137	100,00%	0,00%	propeller	137	0	137	100,00%	0,00%
shampoo-bottle	93	6	99	93,94%	6,06%	shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	65	0	65	100,00%	0,00%	standing-bottle	64	1	65	98,46%	1,54%
tire	331	0	331	100,00%	0,00%	tire	331	0	331	100,00%	0,00%
valve	208	0	208	100,00%	0,00%	valve	208	0	208	100,00%	0,00%
toplam	2353	11	2364	99,53%	0,47%	toplam	2358	6	2364	99,75%	0,25%

Çizelge 10.48 Tek kanalda 92 görüntü boyutta 15 batchsize doğruluk çıktısı

Tek Kanal	BatchSize15				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	447	2	449	99,55%	0,45%
can	366	1	367	99,73%	0,27%
chain	225	1	226	99,56%	0,44%
drink-carton	347	2	349	99,43%	0,57%
hook	133	0	133	100,00%	0,00%
propeller	137	0	137	100,00%	0,00%
shampoo-bottle	98	1	99	98,99%	1,01%
standing-bottle	65	0	65	100,00%	0,00%
tire	331	0	331	100,00%	0,00%
valve	208	0	208	100,00%	0,00%
toplam	2357	7	2364	99,70%	0,30%

Tek kanalda 15 batchsize en verimli boyut 64 pixeldir . %99.75 doğruluk oranı vardır

Eğitimi yaparken ikinci olarak 3 kanalda batchsize 15 aldık ve size değerleri değiştirerek sonuçları gözlemledik sonuçlar ;

Çizelge 10.49 Üç kanalda 16 ve 28 görüntü boyutta 15 batchsize doğruluk çıktısı

3Channel	BatchSize15					3Channel	BatchSize15				
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	446	3	449	99,33%	0,67%	bottle	449	0	449	100,00%	0,00%
can	357	10	367	97,28%	2,72%	can	367	0	367	100,00%	0,00%
chain	217	9	226	96,02%	3,98%	chain	226	0	226	100,00%	0,00%
drink-carton	332	17	349	95,13%	4,87%	drink-carton	349	0	349	100,00%	0,00%
hook	132	1	133	99,25%	0,75%	hook	133	0	133	100,00%	0,00%
propeller	131	6	137	95,62%	4,38%	propeller	136	1	137	99,27%	0,73%
shampoo-bottle	90	9	99	90,91%	9,09%	shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	65	0	65	100,00%	0,00%	standing-bottle	65	0	65	100,00%	0,00%
tire	327	4	331	98,79%	1,21%	tire	329	2	331	99,40%	0,60%
valve	207	1	208	99,52%	0,48%	valve	207	1	208	99,52%	0,48%
toplam	2304	60	2364	97,46%	2,54%	toplam	2360	4	2364	99,83%	0,17%

Çizelge 10.50 Üç kanalda 32 ve 64 görüntü boyutta 15 batchsize doğruluk çıktısı

3Channel	BatchSize15					3Channel	BatchSize15				
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	449	0	449	100,00%	0,00%	bottle	449	0	449	100,00%	0,00%
can	365	2	367	99,46%	0,54%	can	359	8	367	97,82%	2,18%
chain	222	4	226	98,23%	1,77%	chain	226	0	226	100,00%	0,00%
drink-carton	348	1	349	99,71%	0,29%	drink-carton	349	0	349	100,00%	0,00%
hook	133	0	133	100,00%	0,00%	hook	133	0	133	100,00%	0,00%
propeller	136	1	137	99,27%	0,73%	propeller	137	0	137	100,00%	0,00%
shampoo-bottle	99	0	99	100,00%	0,00%	shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	65	0	65	100,00%	0,00%	standing-bottle	65	0	65	100,00%	0,00%
tire	331	0	331	100,00%	0,00%	tire	331	0	331	100,00%	0,00%
valve	208	0	208	100,00%	0,00%	valve	208	0	208	100,00%	0,00%
toplam	2356	8	2364	99,66%	0,34%	toplam	2356	8	2364	99,66%	0,34%

Çizelge 10.51 Tablo 3 kanalda 92 görüntü boyutta 15 batchsize doğruluk çıktısı

3Channel	BatchSize15				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	449	0	449	100,00%	0,00%
can	365	2	367	99,46%	0,54%
chain	226	0	226	100,00%	0,00%
drink-carton	332	17	349	95,13%	4,87%
hook	133	0	133	100,00%	0,00%
propeller	136	1	137	99,27%	0,73%
shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	65	0	65	100,00%	0,00%
tire	331	0	331	100,00%	0,00%
valve	208	0	208	100,00%	0,00%
toplam	2344	20	2364	99,15%	0,85%

3 kanalda 15 batchsize en verimli boyut 28 pixeldir . %99.83 oranında doğruluk payı vardır.

Eđitimi yaparken 3 kanalda filtre uygulayarak batchsize 15 aldık ve size deęerleri deęiřtirerek sonuęları gözlemledik sonuęlar ;

Çizelge 10.52 Üç kanalda filtre 16 ve 28 görüntü boyutta 15 batchsize doğruluk çıktısı

3fChannel	BatchSize15					3fChannel	BatchSize15				
16X16	Train	Loss	Toplam	%Train	%loss	28X28	Train	Loss	Toplam	%Train	%loss
bottle	449	0	449	100,00%	0,00%	bottle	449	0	449	100,00%	0,00%
can	362	5	367	98,64%	1,36%	can	367	0	367	100,00%	0,00%
chain	224	2	226	99,12%	0,88%	chain	223	3	226	98,67%	1,33%
drink-carton	349	0	349	100,00%	0,00%	drink-carton	349	0	349	100,00%	0,00%
hook	132	1	133	99,25%	0,75%	hook	131	2	133	98,50%	1,50%
propeller	134	3	137	97,81%	2,19%	propeller	137	0	137	100,00%	0,00%
shampoo-bottle	99	0	99	100,00%	0,00%	shampoo-bottle	98	1	99	98,99%	1,01%
standing-bottle	65	0	65	100,00%	0,00%	standing-bottle	65	0	65	100,00%	0,00%
tire	330	1	331	99,70%	0,30%	tire	331	0	331	100,00%	0,00%
valve	208	0	208	100,00%	0,00%	valve	208	0	208	100,00%	0,00%
toplam	2352	12	2364	99,49%	0,51%	toplam	2358	6	2364	99,75%	0,25%

Çizelge 10.53 Üç kanalda filtre 32 ve 64 görüntü boyutta 15 batchsize doğruluk çıktısı

3fChannel	BatchSize15					3fChannel	BatchSize15				
32X32	Train	Loss	Toplam	%Train	%loss	64X64	Train	Loss	Toplam	%Train	%loss
bottle	449	0	449	100,00%	0,00%	bottle	449	0	449	100,00%	0,00%
can	360	7	367	98,09%	1,91%	can	365	2	367	99,46%	0,54%
chain	222	4	226	98,23%	1,77%	chain	224	2	226	99,12%	0,88%
drink-carton	348	1	349	99,71%	0,29%	drink-carton	347	2	349	99,43%	0,57%
hook	133	0	133	100,00%	0,00%	hook	133	0	133	100,00%	0,00%
propeller	137	0	137	100,00%	0,00%	propeller	137	0	137	100,00%	0,00%
shampoo-bottle	88	11	99	88,89%	11,11%	shampoo-bottle	99	0	99	100,00%	0,00%
standing-bottle	65	0	65	100,00%	0,00%	standing-bottle	65	0	65	100,00%	0,00%
tire	331	0	331	100,00%	0,00%	tire	331	0	331	100,00%	0,00%
valve	208	0	208	100,00%	0,00%	valve	208	0	208	100,00%	0,00%
toplam	2341	23	2364	99,03%	0,97%	toplam	2358	6	2364	99,75%	0,25%

Çizelge 10.54 Üç kanalda filtre 92 görüntü boyutta 15 batchsize doğruluk çıktısı

3fChannel	BatchSize15				
92X92	Train	Loss	Toplam	%Train	%loss
bottle	449	0	449	100,00%	0,00%
can	344	23	367	93,73%	6,27%
chain	225	1	226	99,56%	0,44%
drink-carton	349	0	349	100,00%	0,00%
hook	132	1	133	99,25%	0,75%
propeller	137	0	137	100,00%	0,00%
shampoo-bottle	92	7	99	92,93%	7,07%
standing-bottle	65	0	65	100,00%	0,00%
tire	331	0	331	100,00%	0,00%
valve	208	0	208	100,00%	0,00%
toplam	2332	32	2364	98,65%	1,35%

3 kanalda filtreli görsellerde 15 batchsize en verimli boyut 28 ve 64 pixelleridir . %99,75 oranında doğruluk payı vardır.

10.7 TEK KANALDA BATCHSİZE VE SİZE

Çizelge 10.55 tek kanalda batchsize ve size doğruluk yüzdeleri

Batchsize Görsel Boyutu	15	50	75	150	300
16	97,63%	92,72%	89,08%	80,11%	71,74%
28	99,49%	97,54%	96,44%	89,59%	85,70%
32	99,53%	98,60%	95,85%	92,08%	86,59%
64	99,74%	99,61%	99,28%	97,92%	84,51%
92	99,70%	99,57%	99,45%	98,05%	89,04%

En iyi sonuç size 64 batchsize 15 olup %99.74 olandır.

10.8 ÜÇ KANALDA BATCHSİZE VE SİZE

Çizelge 10.55 Üç kanal görüntü için doğruluk yüzdeleri

Batchsize Görsel Boyutu	15	50	75	150	300
16	97,46%	94,41%	89,55%	79,78%	63,11%
28	99,83%	99,23%	97,46%	90,98%	80,11%
32	99,66%	98,90%	96,86%	90,22%	83,67%
64	99,66%	99,70%	98,51%	98,13%	91,11%
92	99,15%	99,45%	97,96%	77,70%	83,71%

En iyi sonuç size 28 batchsize 15 olup %99.83 olandır.

10.9 ÜÇ Kanalda Filtre BATCHSİZE VE SİZE

Çizelge 10.56 Üç kanal filtrelenmiş görüntü için doğruluk yüzdeleri

Batchsize	15	50	75	150	300
Görsel Boyutu					
16	99,49%	97,75%	95,21%	90,73%	81,17%
28	99,74%	99,57%	98,98%	95,64%	95,00%
32	99,02%	99,49%	98,98%	98,56%	89,42%
64	99,74%	99,83%	98,90%	96,36%	46,95%
92	98,64%	98,51%	99,70%	97,12%	82,95%

En iyi sonuç size 64 batchsize 50 olup %99.83 olandır.

11. EĞİTİM İŞLEMİ KONFEKSİYON MATRİX

11.1 TEK KANALDAKİ EN İYİ SONUÇ

Çizelge 11. Tek kanal görüntü için en iyi sonucu veren 15 batchsize ve 28 size da elde edilen karmaşıklık matrisi

1Channel	BatchSize=15									
64X64	bottle	can	chain	drink-c	hook	propeller	shampoo-b	standing-b	tire	valve
bottle	448	0	0	0	0	0	0	0	0	0
can	0	366	1	0	0	0	0	0	0	0
chain	0	0	223	0	0	0	0	0	0	0
drink-c	0	0	0	349	0	0	0	0	0	0
hook	0	0	0	0	133	0	0	0	0	0
propeller	0	0	1	0	0	137	0	0	0	0
shampoo-b	0	0	1	0	0	0	99	1	0	0
standing-b	0	0	0	0	0	0	0	64	0	0
tire	0	0	0	0	0	0	0	0	331	0
valve	0	0	0	0	0	0	0	0	0	208

11.2 ÜÇ KANALDAKİ EN İYİ SONUÇ

Çizelge 11. Üç kanal görüntü için en iyi sonucu veren 15 batchsize ve 28 size da elde edilen karmaşıklık matrisi

3Channel	BatchSize=15									
28X28	bottle	can	chain	drink-c	hook	propeller	shampoo-b	standing-b	tire	valve
bottle	449	0	0	0	0	0	0	0	0	0
can	0	367	0	0	0	0	0	0	0	0
chain	0	0	226	0	0	0	0	0	0	0
drink-c	0	0	0	349	0	0	0	0	0	1
hook	0	0	0	0	133	0	0	0	0	0
propeller	0	0	0	0	0	136	0	0	0	0
shampoo-b	0	0	0	0	0	0	99	0	0	0
standing-b	0	0	0	0	0	0	0	65	1	0
tire	0	0	0	0	0	0	0	0	329	0
valve	0	0	0	0	0	0	0	0	0	207

11.3 ÜÇ KANALDA FİLTREDEKİ EN İYİ SONUÇ

Çizelge 11.3 Üç kanal filtrelenmiş görüntü için en iyi sonucu veren 15 batchsize ve 28 size da elde edilen karmaşıklık matrisi

3FChannel	BatchSize=50									
64X64	bottle	can	chain	drink-c	hook	propeller	shampoo-b	standing-b	tire	valve
bottle	447	0	0	0	0	0	0	0	0	0
can	0	366	0	0	0	0	0	0	0	0
chain	0	0	226	0	0	0	0	0	0	0
drink-c	0	0	0	348	0	0	0	0	0	0
hook	0	0	0	0	133	0	0	0	0	0
propeller	0	0	0	0	0	137	0	0	0	0
shampoo-b	0	0	0	0	0	0	99	0	0	0
standing-b	0	0	0	0	0	0	0	65	0	0
tire	0	1	0	0	0	0	0	0	331	0
valve	2	0	0	1	0	0	0	0	0	208

12. KAYNAKLAR

<https://keras.io/api/>

https://github.com/neeraj95575/object_detection_using_keras

<https://github.com/anushaagrawal/Multi-class-object-detection>

<https://pyimagesearch.com/2018/06/04/keras-multiple-outputs-and-multiple-losses/>

<https://www.youtube.com/watch?v=hraKTseOuJA>

<https://pyimagesearch.com/2018/05/07/multi-label-classification-with-keras/#pyis-cta-modal>