Load data from internet:

from bing\_image\_downloader import downloader

# downloader.download("car png",limit=100,output\_dir='test\_data\_car',adult\_filter\_off=True)

# downloader.download("door png",limit=100,output\_dir='test\_data\_car',adult\_filter\_off=True)

# downloader.download("ice cream cone png",limit=100,output\_dir='test\_data\_car',adult\_filter\_off=True)

Training data file:

import os

import matplotlib.pyplot as plt

import numpy as np

from skimage.io import imread

from skimage.transform import resize

from sklearn.model\_selection import train\_test\_split

from sklearn.model\_selection import GridSearchCV

from sklearn import svm

# support vector machine

import pickle

target = []

# my image

images = []

#data after transform 2d metrics to 1d metrics for every photo

flatten\_data = []

directory = 'images'

CATEGORTES = ['car png'  , 'door png' , 'ice cream png']

for category in CATEGORTES:

  class\_num = CATEGORTES.index(category)

  path = os.path.join(directory,category)

  for img in os.listdir(path):

    img\_array =  imread(os.path.join(path,img))

    img\_resize = resize(img\_array,(150,150,3)) #normalize value from 0 to 1

    # list in python

    flatten\_data.append(img\_resize.flatten())

    images.append(img\_resize)

    target.append(class\_num)

# import numpy as np

flatten\_data  = np.array(flatten\_data)

target =  np.array(target)

images = np.array(images)

# import matplotlib.pyplot as plt

unique,count = np.unique(target,return\_counts=True)

plt.bar(CATEGORTES,count)

# split data to traning and testing

x\_train,x\_test , y\_train,y\_test = train\_test\_split(flatten\_data,target,test\_size=0.3,random\_state=109)

param\_grid = [

    {'C':[1,10,100,1000],

    'kernel':['linear']},

    {'C':[1,10,100,1000],

    'gamma':[0.001,0.0001],

    'kernel':['rbf']},

]

svc = svm.SVC(probability=True)

clf = GridSearchCV(svc,param\_grid)

clf.fit(x\_train,y\_train)

pickle.dump(clf,open('model.sav','wb'))

test data :

import numpy as np

from skimage.io import imread, imshow, show

from skimage.transform import resize

import pickle

CATEGORTES = ['car png'  , 'door png' , 'ice cream png']

model = pickle.load(open('model.sav', 'rb'))

while True:

    path = input('Enter your URL: ')

    if path.lower() == 'stop':

        break

    try:

        img = imread(path)

    except:

        print("Imgae not found!")

        continue

    img\_resize = resize(img,(150,150,3))

    flat\_data = [img\_resize.flatten()]

    flat\_data = np.array(flat\_data)

    y\_out = model.predict(flat\_data)

    y\_out = CATEGORTES[y\_out[0]]

    print(f'PREDICTED photo:{y\_out}')

    imshow(img)

    show()