

```
import tensorflow as tf
import random as rn
import os, cv2
import numpy as np

os.environ['PYTHONHASHSEED'] = '0'
# Setting the seed for numpy-generated random numbers
np.random.seed(37)
# Setting the seed for python random numbers
rn.seed(1254)
# Setting the graph-level random seed.
tf.set_random_seed(89)
from keras import backend as K
session_conf = tf.ConfigProto(intra_op_parallelism_threads=1, inter_op_parallelism_threads=1)

#Force Tensorflow to use a single thread
sess = tf.Session(graph=tf.get_default_graph(), config=session_conf)
K.set_session(sess)
import glob

from sklearn.utils import shuffle

from sklearn.model_selection import train_test_split

import re
from keras.models import Sequential
from keras import models
from keras import layers
from keras import optimizers
from keras.applications import VGG16
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.layers import Dense, Conv2D, Flatten, MaxPooling2D, Dropout
from sklearn.utils import shuffle
from PIL import Image
from google.colab.patches import cv2_imshow
import matplotlib.pyplot as plt
import glob
import numpy as np
import os, cv2

def gen_image(img):
    plt.imshow(img)
    return plt

#from sklearn.cross_validation import train_test_split

from google.colab import drive
drive.mount('/content/drive')

PATH = os.getcwd()
from google.colab import drive
drive.mount('/content/drive')
https://colab.research.google.com/drive/1fNLnI9eL8rDzRIP2nEkS-FES75QuTgRw#printMode=true
```

```
train_dir = '/content/drive/My Drive/Colab Notebooks/cats_dogs_horse_humans/train'
validation_dir = '/content/drive/My Drive/Colab Notebooks/cats_dogs_horse_humans/validation'

#Create model
model = Sequential()
#Add model layers

model.add(Conv2D(128, kernel_size=3,strides=1, activation='relu',input_shape=(256,256,3)))
    # 64 are the number of filters, kernel size is the size of the filters example 3*3
model.add(Conv2D(128, kernel_size=3,strides=1, activation='relu'))
model.add(MaxPooling2D(pool_size=(3,3)))
model.add(Conv2D(256, kernel_size=3,strides=1, activation='relu'))
model.add(Conv2D(256, kernel_size=3,strides=1, activation='relu'))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(4, activation='softmax'))

image_size=256
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')

validation_datagen = ImageDataGenerator(rescale=1./255)

# Change the batchsize according to your system RAM
train_batchsize = 50
val_batchsize = 10

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(image_size, image_size),
    batch_size=train_batchsize,
    class_mode='categorical')

validation_generator = validation_datagen.flow_from_directory(
    validation_dir,
    target_size=(image_size, image_size),
    batch_size=val_batchsize,
    class_mode='categorical',
    shuffle=False)

# Compile the model
model.compile(loss='categorical_crossentropy',
              optimizer=optimizers.RMSprop(lr=1e-4),
              metrics=['acc'])
```

```
# Train the model
history = model.fit_generator(
    train_generator,
    steps_per_epoch=train_generator.samples/train_generator.batch_size ,
    epochs=5,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples/validation_generator.batch_size,
    verbose=1)

'''#TESTING
test_dir='/content/drive/My Drive/images'
test_datagen = ImageDataGenerator(rescale=1./255)
test_generator = test_datagen.flow_from_directory( test_dir, target_size=(256, 256),batch_
test_generator=test_generator.reshape(10,256,256,3)

ytested = model.predict_classes(test_generator)
labels=['human','horse','dogs','cats']
for i in range(len(ytested)):
    #print("The Predicted Testing image is =%s verify below" ,labels[ytested[i]])
    print(np.argmax(ytested[i]))
    gen_image(test_generator[i]).show() '''

#TESTING
data_path = '/content/drive/My Drive/Colab Notebooks/cats_dogs_horse_humans/test/all_data'
img_data_list=[]
img_list = glob.glob(data_path+'/*.jpg')
for img in img_list:
    input_img=cv2.imread(img,1 )
    input_img_resize=cv2.resize(input_img,(256,256))
    img_data_list.append(input_img_resize)

img_data = np.array(img_data_list)
img_data = img_data.astype('float32')
x_test = shuffle(img_data, random_state=2)
x_test=x_test/255
#Nv_test=x_test.shape[0]
x_test = x_test.reshape(40,256,256,3)
label=['humans','horse','dogs','cats']
ytested = model.predict_classes(x_test)
for i in range(40):
    print("The Predicted Testing image is =%s verify below" %label[ytested[i]])
    gen_image(x_test[i]).show() # printing image vs the predicted image below
```

C→

The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.
We recommend you [upgrade](#) now or ensure your notebook will continue to use TensorFlow 1.x via the %ter
Jsing TensorFlow backend.
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=94731

Enter your authorization code:
.....

Mounted at /content/drive

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorf

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorf

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorf

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorf

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorf
Instructions for updating:

Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`
Found 608 images belonging to 4 classes.

Found 160 images belonging to 4 classes.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/optimizers.py:793

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorf

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_core/python/
Instructions for updating:

Use tf.where in 2.0, which has the same broadcast rule as np.where

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorf

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorf

Epoch 1/5

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorf

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorf

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorf

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorf

L3/12 [=====] - 221s 17s/step - loss: 1.7662 - acc: 0.2818

Epoch 2/5

L3/12 [=====] - 7s 546ms/step - loss: 1.3769 - acc: 0.2716

Epoch 3/5

L3/12 [=====] - 13s 1s/step - loss: 1.3905 - acc: 0.2853 -

Epoch 4/5

L3/12 [=====] - 13s 1s/step - loss: 1.3920 - acc: 0.2856 -

Epoch 5/5

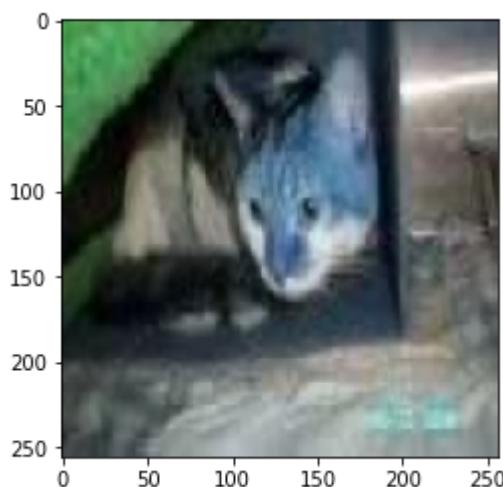
L3/12 [=====] - 13s 1s/step - loss: 1.3838 - acc: 0.2796 -

The Predicted Testing image is =humans verify below

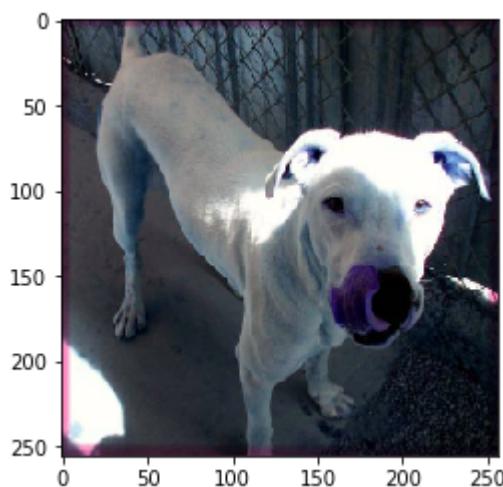




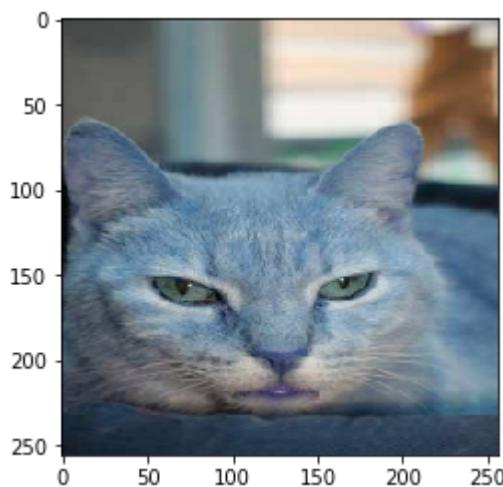
The Predicted Testing image is =humans verify below



The Predicted Testing image is =humans verify below



The Predicted Testing image is =horse verify below



The Predicted Testing image is =humans verify below

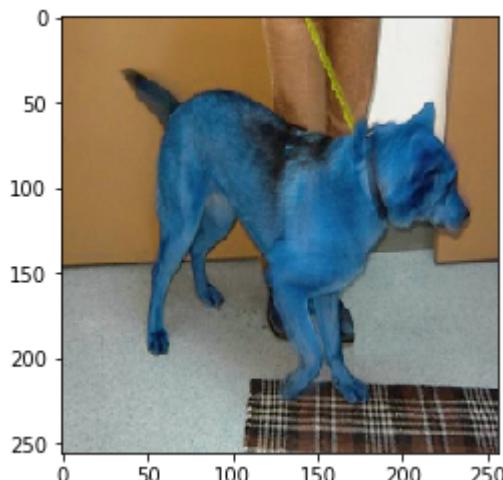




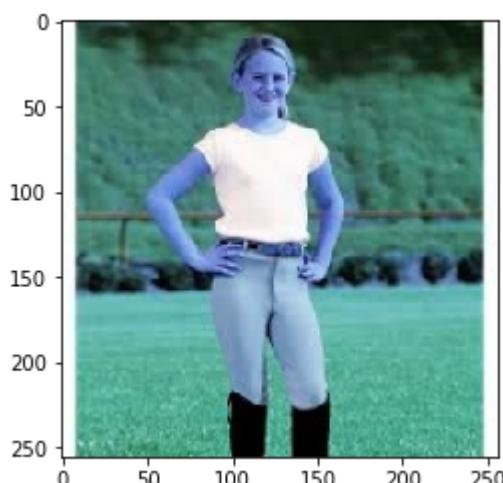
The Predicted Testing image is =humans verify below



The Predicted Testing image is =cats verify below

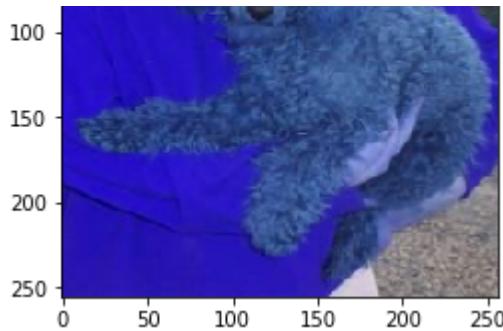


The Predicted Testing image is =humans verify below

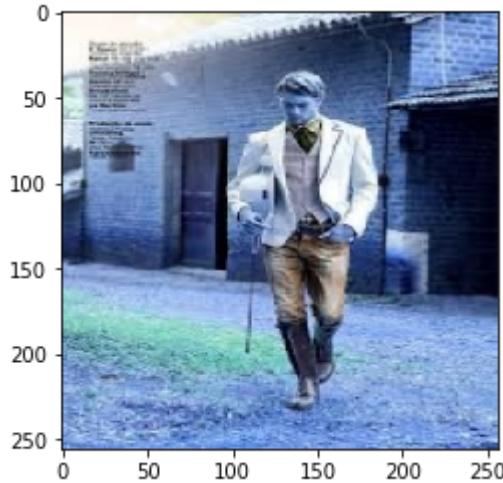


The Predicted Testing image is =humans verify below

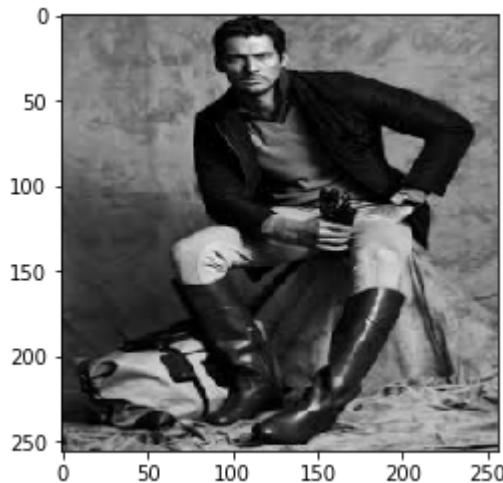




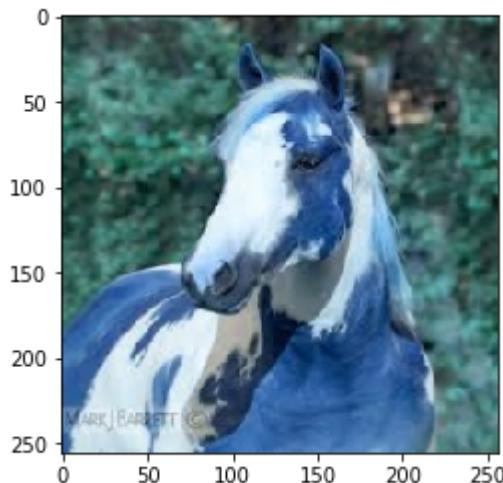
The Predicted Testing image is =humans verify below



The Predicted Testing image is =humans verify below

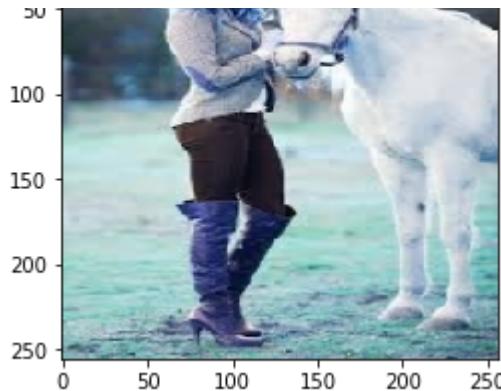


The Predicted Testing image is =humans verify below



The Predicted Testing image is =humans verify below

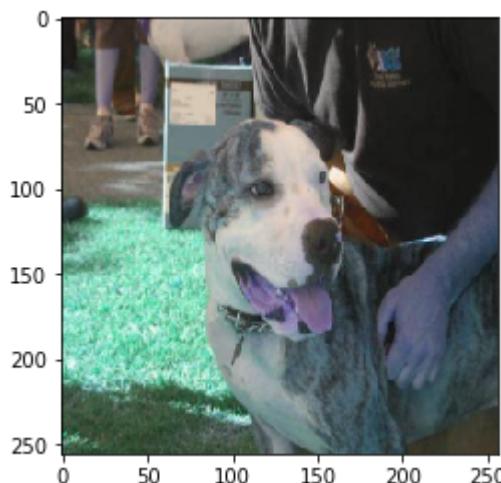




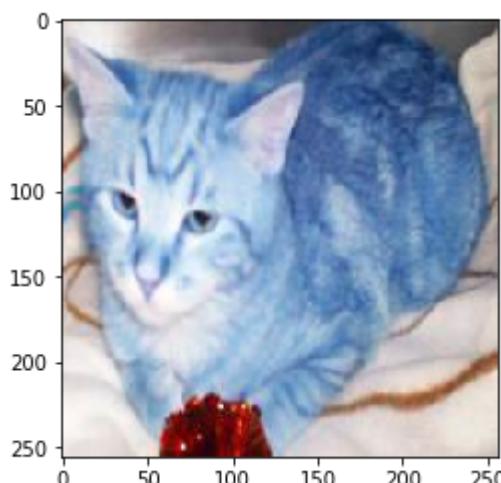
The Predicted Testing image is =cats verify below



The Predicted Testing image is =humans verify below

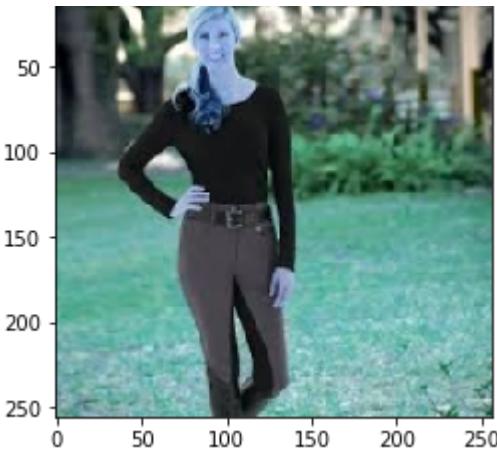


The Predicted Testing image is =cats verify below

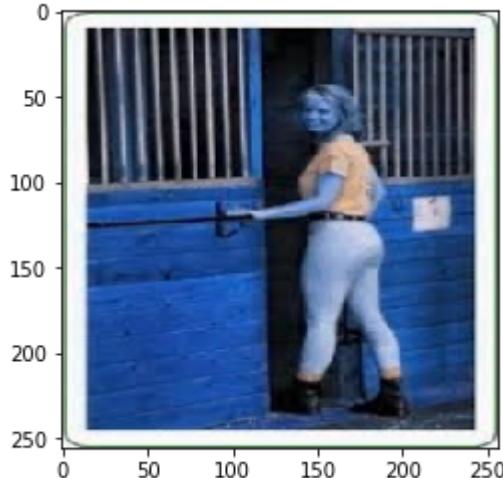


The Predicted Testing image is =humans verify below

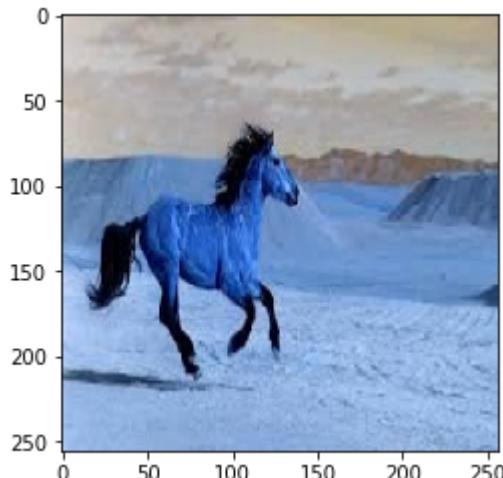




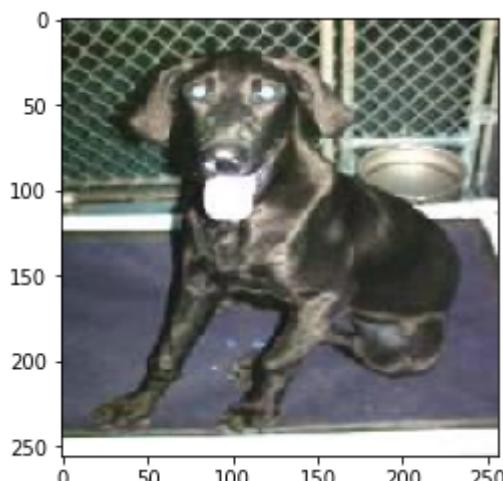
The Predicted Testing image is =humans verify below



The Predicted Testing image is =humans verify below



The Predicted Testing image is =humans verify below

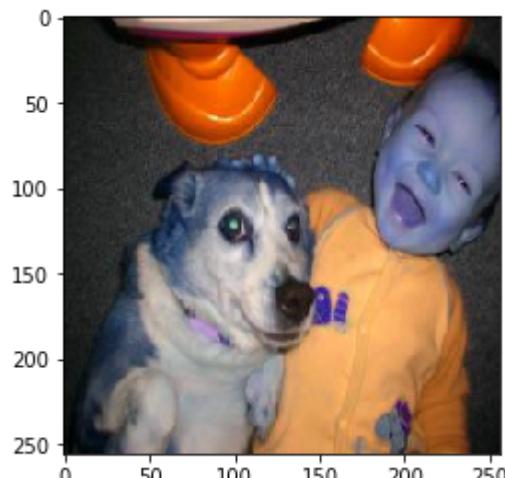


The Predicted Testing image is =humans verify below

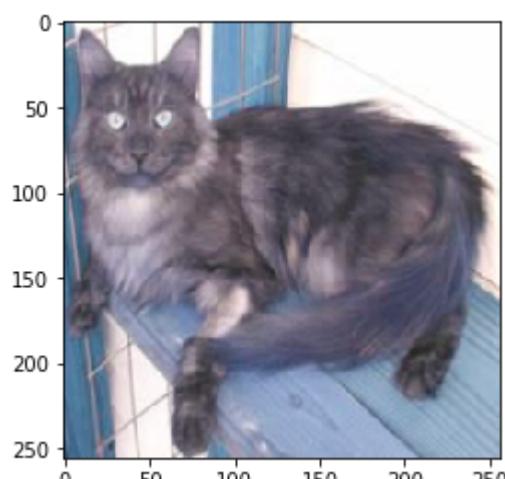
The Predicted Testing image is =humans verify below



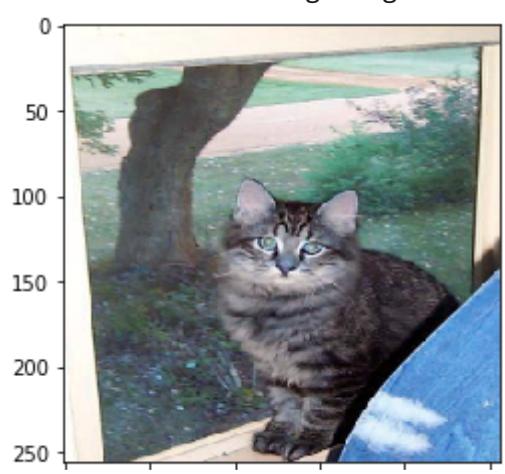
The Predicted Testing image is =cats verify below



The Predicted Testing image is =cats verify below

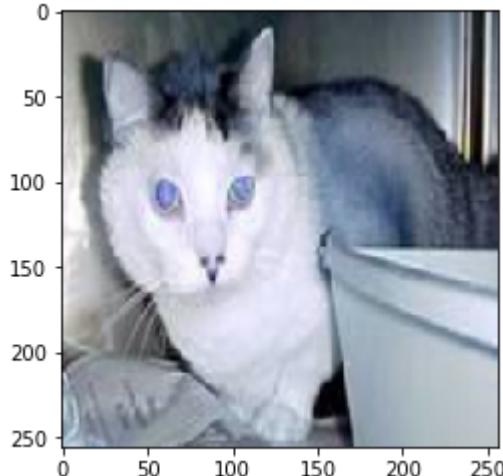


The Predicted Testing image is =humans verify below



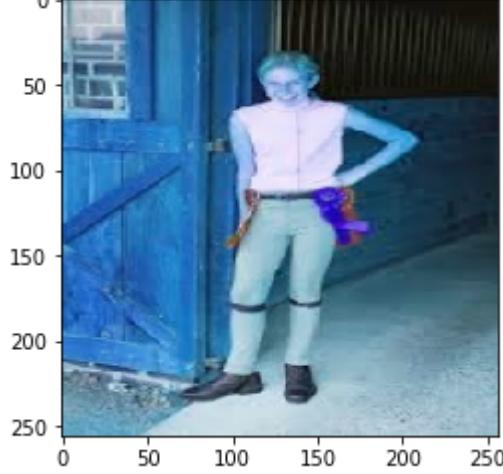
0 50 100 150 200 250

The Predicted Testing image is =humans verify below



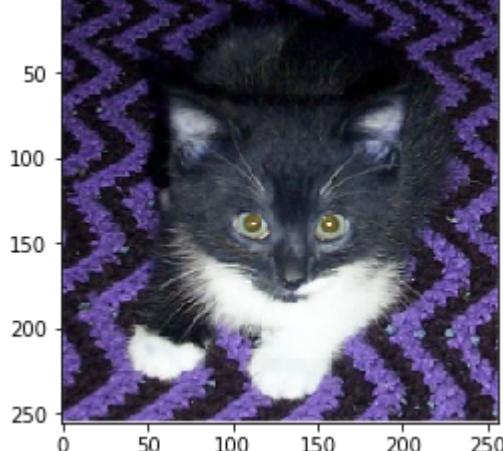
0 50 100 150 200 250

The Predicted Testing image is =humans verify below



0 50 100 150 200 250

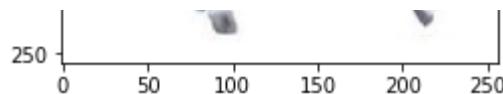
The Predicted Testing image is =humans verify below



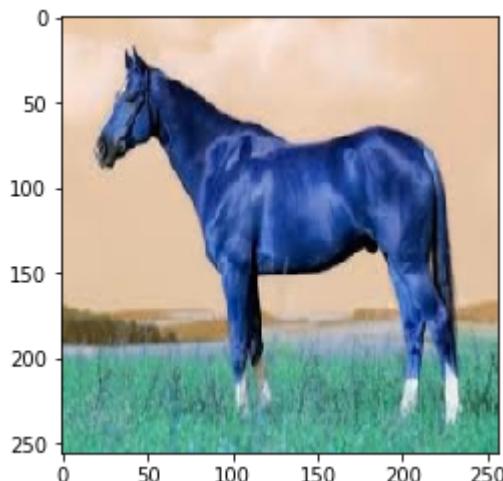
0 50 100 150 200 250

The Predicted Testing image is =cats verify below

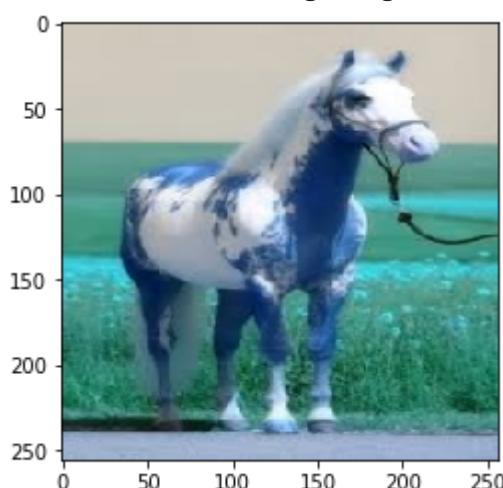




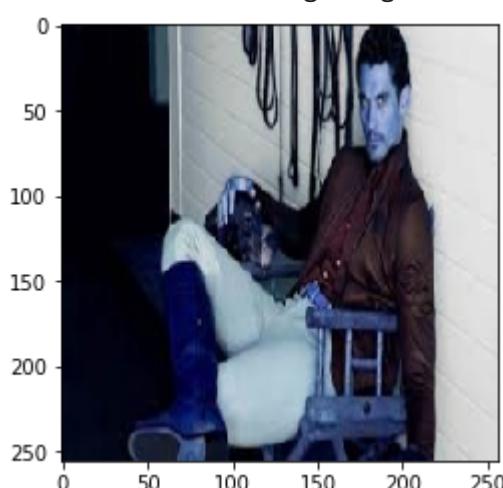
The Predicted Testing image is =humans verify below



The Predicted Testing image is =humans verify below



The Predicted Testing image is =humans verify below

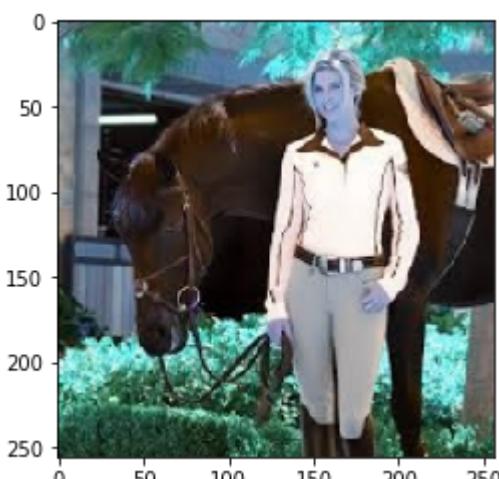


The Predicted Testing image is =humans verify below





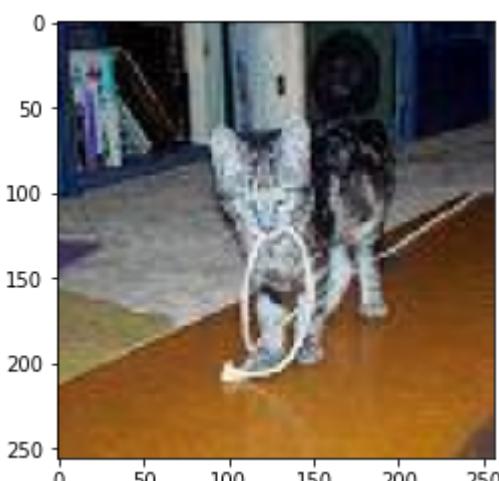
The Predicted Testing image is =humans verify below



The Predicted Testing image is =humans verify below

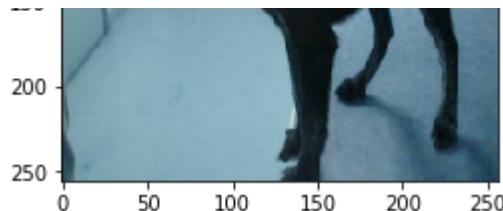


The Predicted Testing image is =cats verify below

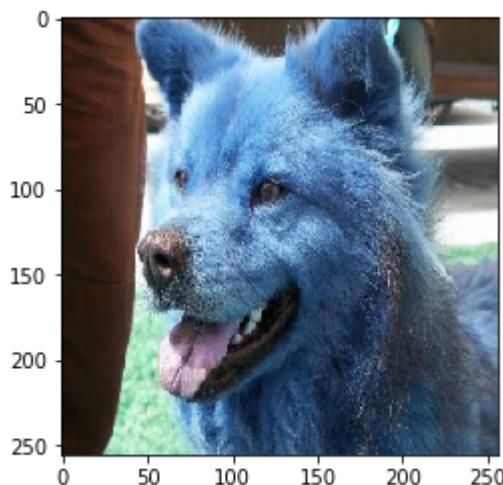


The Predicted Testing image is =humans verify below

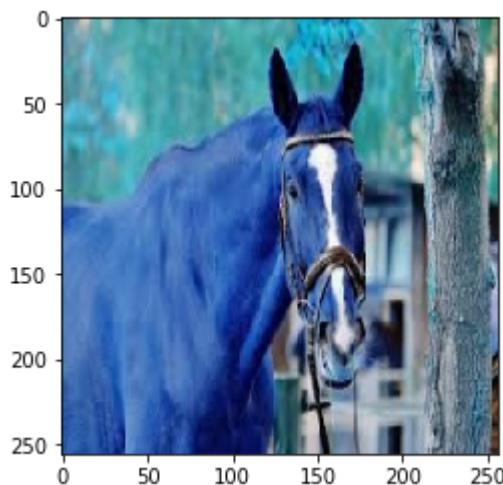




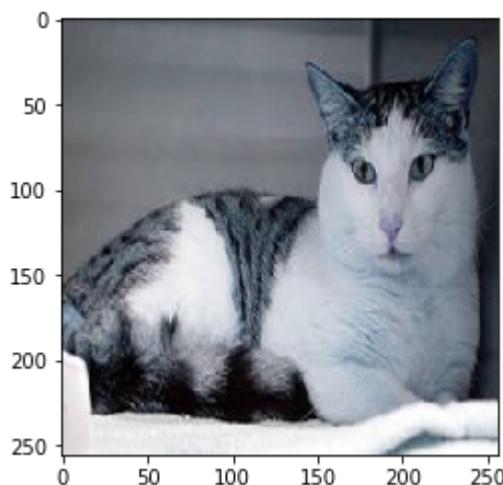
The Predicted Testing image is =humans verify below



The Predicted Testing image is =humans verify below



The Predicted Testing image is =humans verify below



The Predicted Testing image is =humans verify below

