

```
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
#Load the VGG model

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

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

img_width, img_height = 256, 256
image_size = img_height

vgg_conv = VGG16(weights='imagenet', include_top=False, input_shape=(img_width, img_height))

# Freeze the layers except the last 4 layers
for layer in vgg_conv.layers[:-2]:
    layer.trainable = False

# Check the trainable status of the individual layers
for layer in vgg_conv.layers:
    print(layer, layer.trainable)

# Create the model
model = models.Sequential()

# Add the vgg convolutional base model
model.add(vgg_conv)

# Add new layers
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'))
```

```
# Show a summary of the model. Check the number of trainable parameters
model.summary()

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)

acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(acc))

plt.plot(epochs, acc, 'b', label='Training acc')

```

```
plt.plot(epochs, val_acc, 'r', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()

plt.figure()

plt.plot(epochs, loss, 'b', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()

plt.show()
```

```
#####
# insert the testing code
#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
```

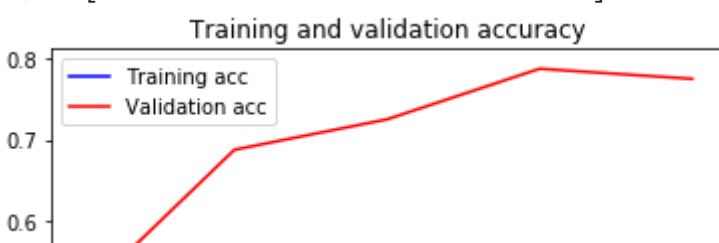
C→

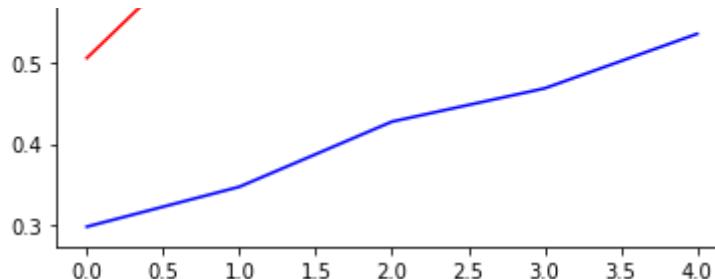
```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
<keras.engine.input_layer.InputLayer object at 0x7ff970bb54e0> False
<keras.layers.convolutional.Conv2D object at 0x7ff970bb54a8> False
<keras.layers.convolutional.Conv2D object at 0x7ff970bb5780> False
<keras.layers.pooling.MaxPooling2D object at 0x7ff970bbc5f8> False
<keras.layers.convolutional.Conv2D object at 0x7ff970bafcd8> False
<keras.layers.convolutional.Conv2D object at 0x7ff970b40e10> False
<keras.layers.pooling.MaxPooling2D object at 0x7ff970b44cf8> False
<keras.layers.convolutional.Conv2D object at 0x7ff970b4fbe0> False
<keras.layers.convolutional.Conv2D object at 0x7ff970b54438> False
<keras.layers.convolutional.Conv2D object at 0x7ff970b5b198> False
<keras.layers.pooling.MaxPooling2D object at 0x7ff970b60c50> False
<keras.layers.convolutional.Conv2D object at 0x7ff970b6db00> False
<keras.layers.convolutional.Conv2D object at 0x7ff970b75358> False
<keras.layers.convolutional.Conv2D object at 0x7ff970b7b278> False
<keras.layers.pooling.MaxPooling2D object at 0x7ff970b7ab70> False
<keras.layers.convolutional.Conv2D object at 0x7ff970b0ea20> False
<keras.layers.convolutional.Conv2D object at 0x7ff970b172b0> False
<keras.layers.convolutional.Conv2D object at 0x7ff970b1e5c0> True
<keras.layers.pooling.MaxPooling2D object at 0x7ff970b22a20> True
Model: "sequential_3"
```

Layer (type)	Output Shape	Param #
<hr/>		
vgg16 (Model)	(None, 8, 8, 512)	14714688
flatten_3 (Flatten)	(None, 32768)	0
dense_7 (Dense)	(None, 64)	2097216
dropout_5 (Dropout)	(None, 64)	0
dense_8 (Dense)	(None, 64)	4160
dropout_6 (Dropout)	(None, 64)	0
dense_9 (Dense)	(None, 4)	260
<hr/>		
Total params: 16,816,324		
Trainable params: 4,461,444		
Non-trainable params: 12,354,880		

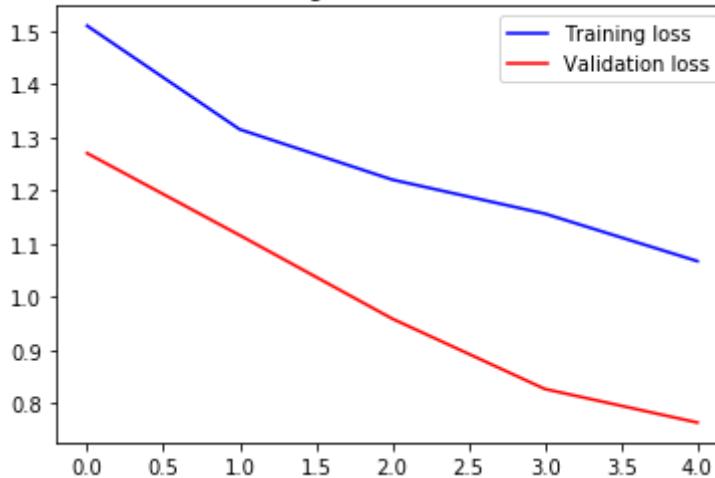
---

```
Found 608 images belonging to 4 classes.
Found 160 images belonging to 4 classes.
Epoch 1/5
13/12 [=====] - 12s 923ms/step - loss: 1.5013 - acc: 0.287
Epoch 2/5
13/12 [=====] - 10s 794ms/step - loss: 1.3094 - acc: 0.356
Epoch 3/5
13/12 [=====] - 11s 815ms/step - loss: 1.2177 - acc: 0.432
Epoch 4/5
13/12 [=====] - 10s 804ms/step - loss: 1.1500 - acc: 0.470
Epoch 5/5
13/12 [=====] - 11s 811ms/step - loss: 1.0769 - acc: 0.534
```

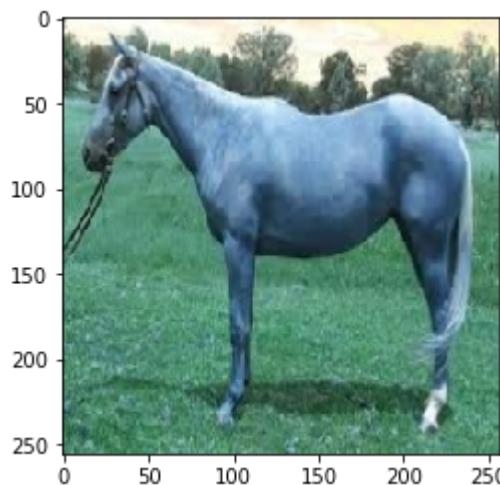




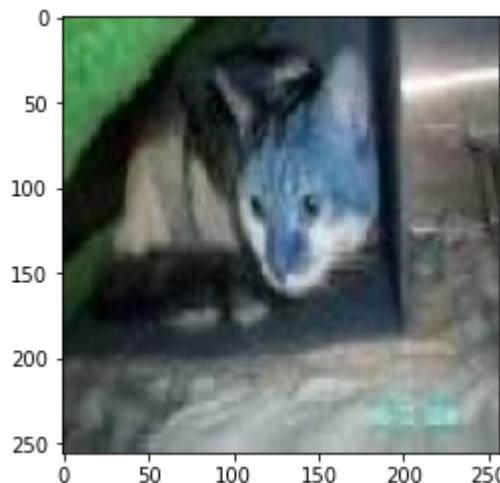
Training and validation loss



The Predicted Testing image is =cats verify below



The Predicted Testing image is =horse verify below

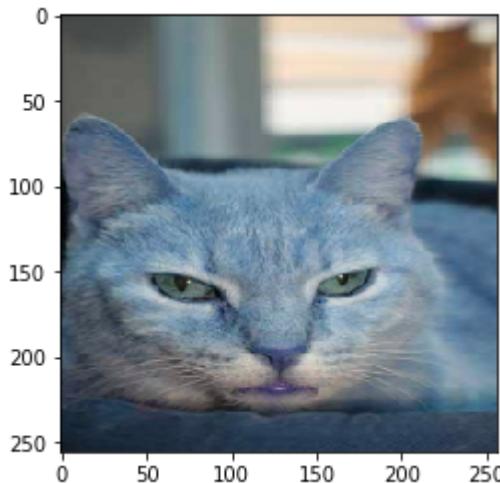


The Predicted Testing image is =dogs verify below

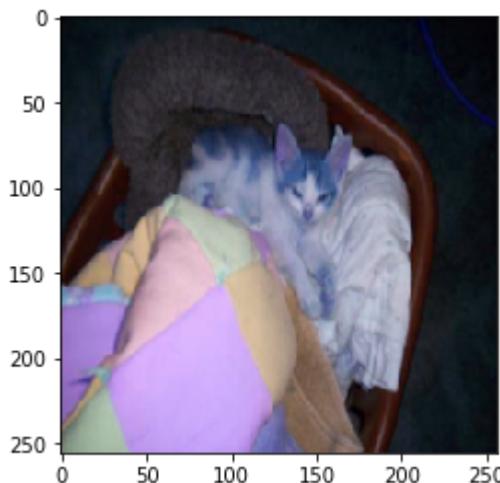




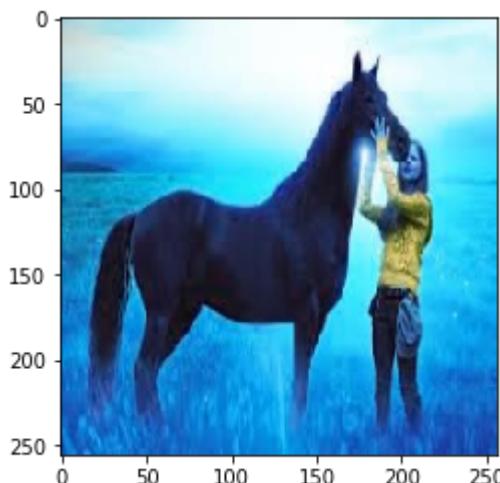
The Predicted Testing image is =horse verify below



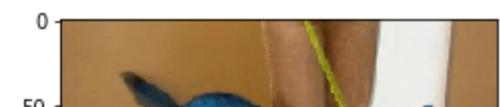
The Predicted Testing image is =horse verify below

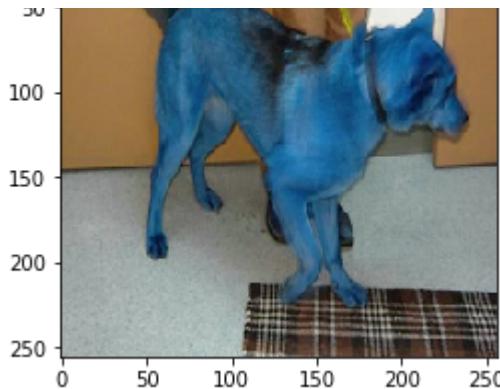


The Predicted Testing image is =cats verify below

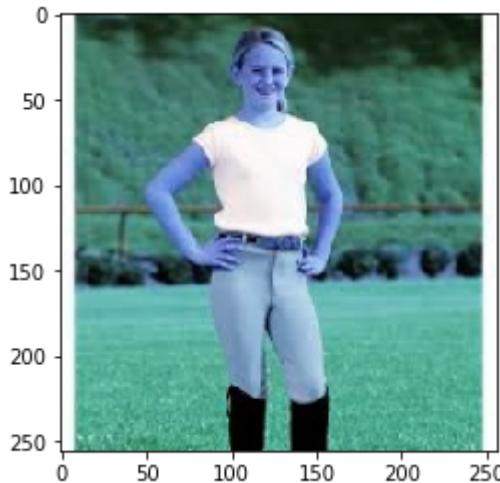


The Predicted Testing image is =dogs verify below

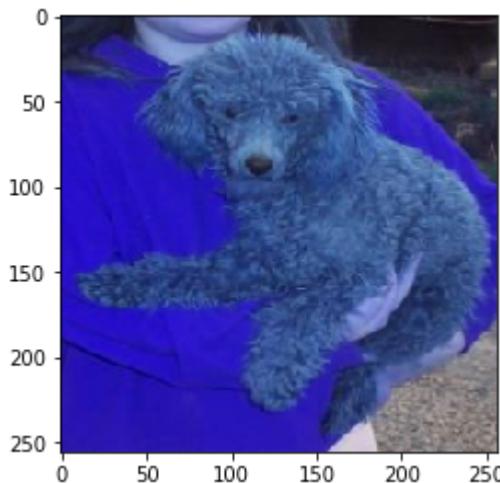




The Predicted Testing image is =humans verify below



The Predicted Testing image is =horse verify below

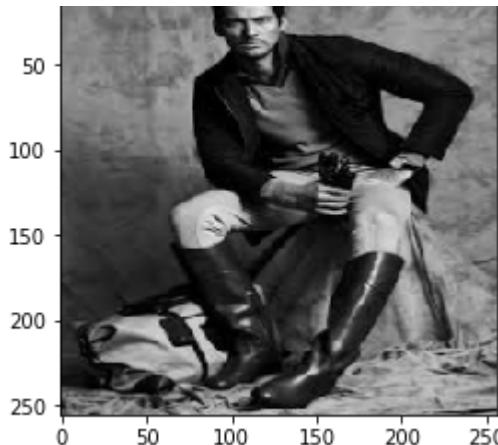


The Predicted Testing image is =humans verify below

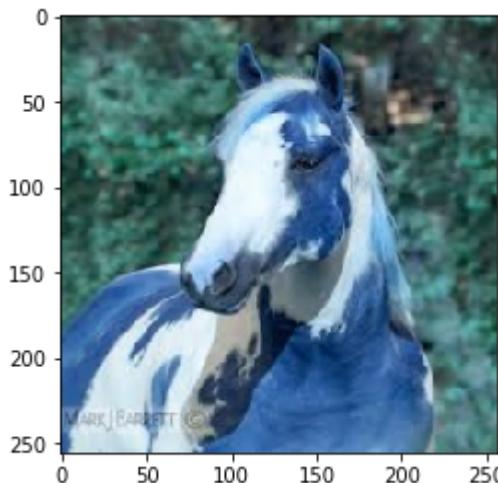


The Predicted Testing image is =humans verify below

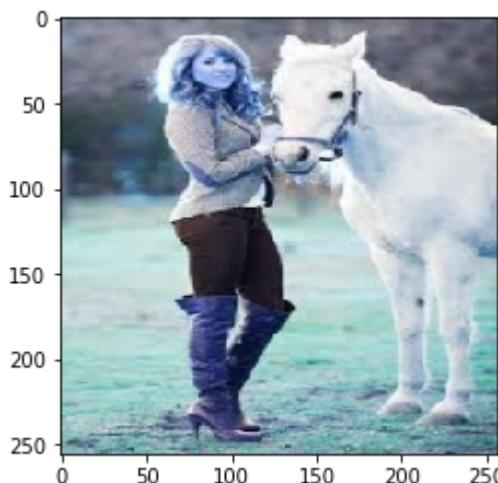




The Predicted Testing image is =dogs verify below



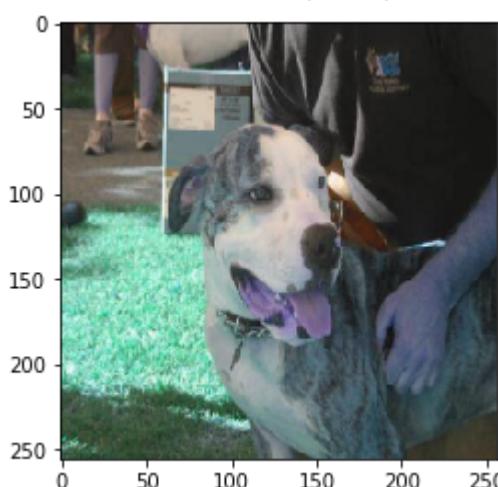
The Predicted Testing image is =humans verify below



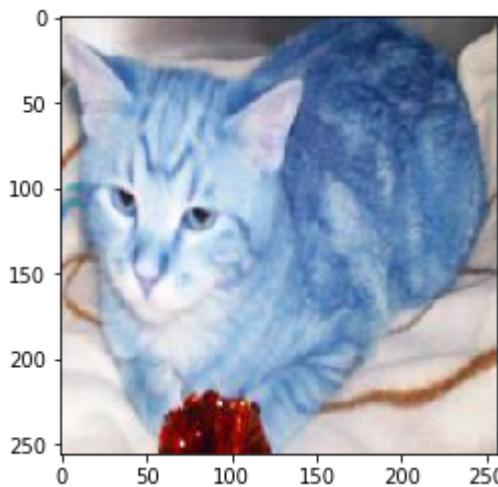
The Predicted Testing image is =horse verify below



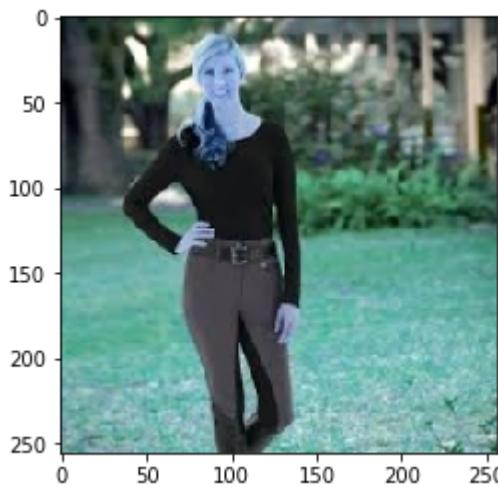
The Predicted Testing image is =dogs verify below



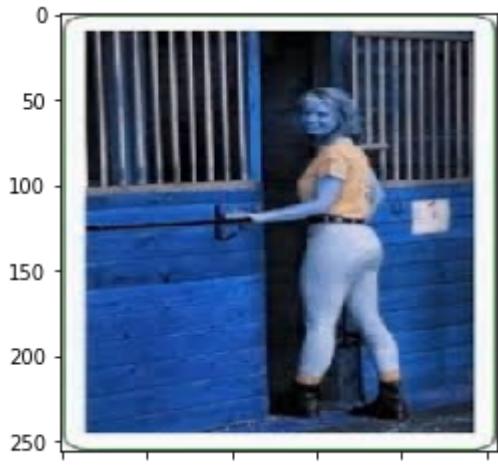
The Predicted Testing image is =horse verify below



The Predicted Testing image is =humans verify below

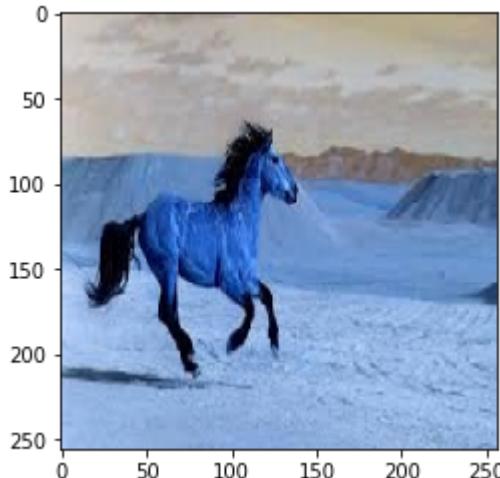


The Predicted Testing image is =humans verify below



0 50 100 150 200 250

The Predicted Testing image is =cats verify below



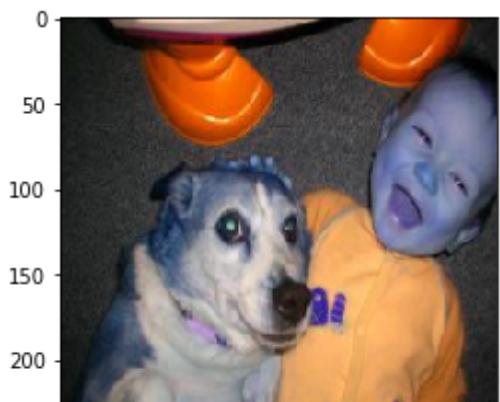
The Predicted Testing image is =horse verify below



The Predicted Testing image is =humans verify below

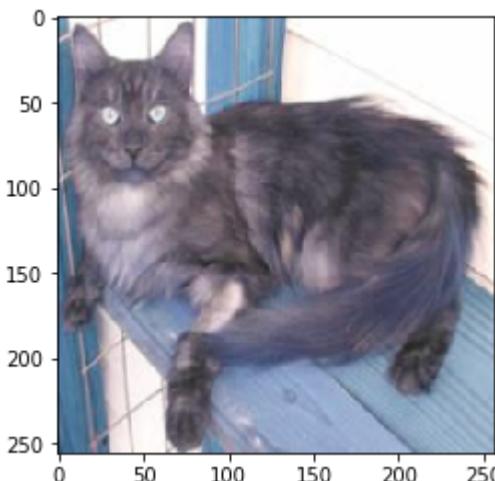


The Predicted Testing image is =horse verify below

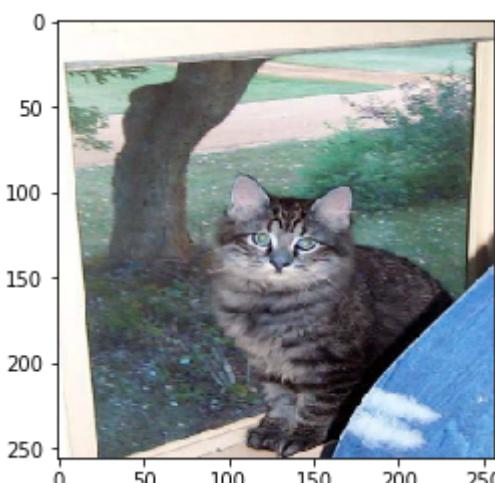




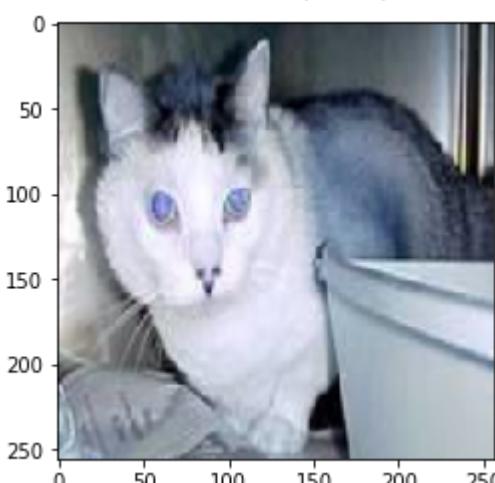
The Predicted Testing image is =horse verify below



The Predicted Testing image is =horse verify below



The Predicted Testing image is =horse verify below

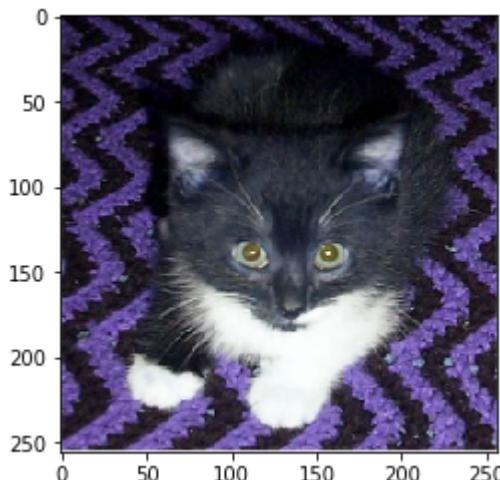


The Predicted Testing image is =humans verify below

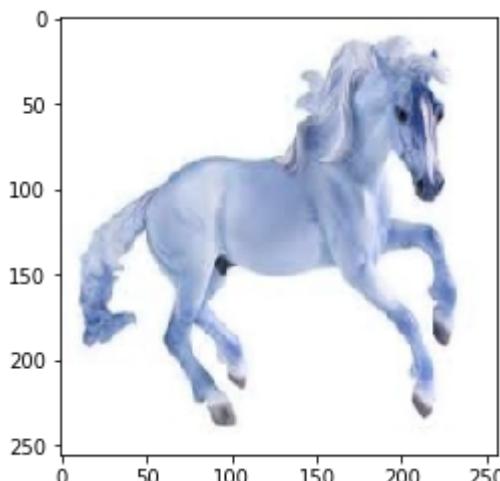




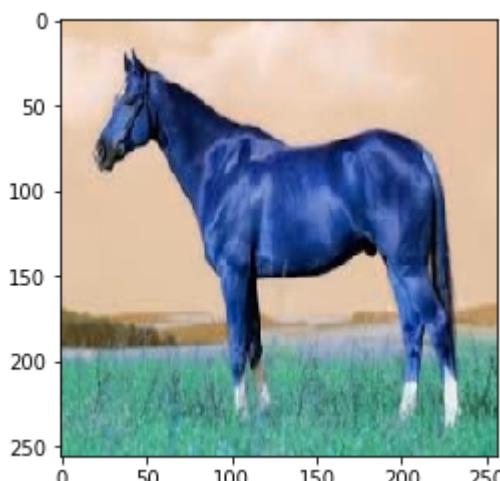
The Predicted Testing image is =horse verify below



The Predicted Testing image is =cats verify below

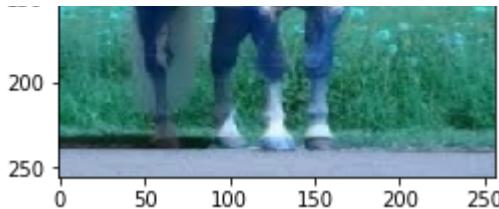


The Predicted Testing image is =cats verify below

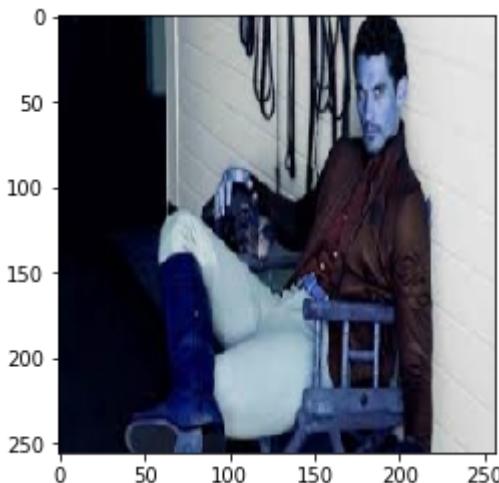


The Predicted Testing image is =cats verify below

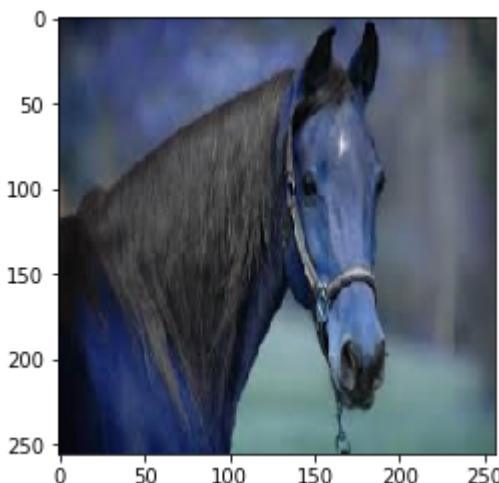




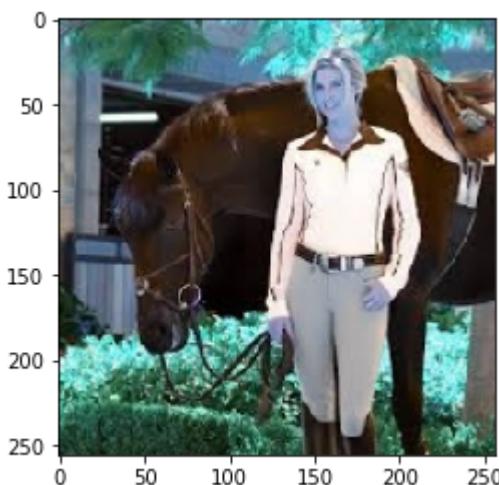
The Predicted Testing image is =humans verify below



The Predicted Testing image is =horse verify below



The Predicted Testing image is =humans verify below

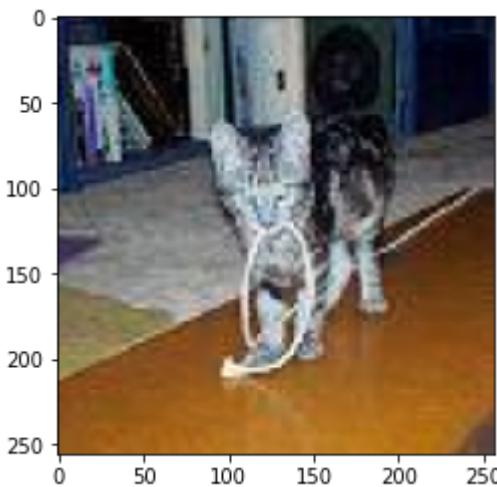


The Predicted Testing image is =horse verify below

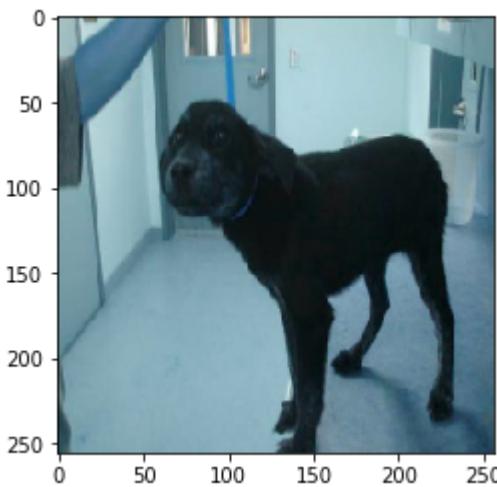




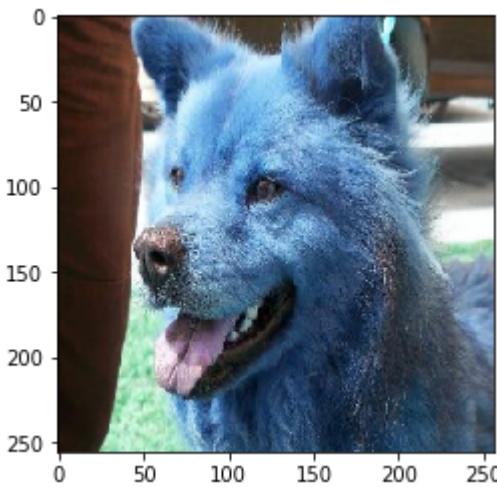
The Predicted Testing image is =horse verify below



The Predicted Testing image is =horse verify below



The Predicted Testing image is =horse verify below

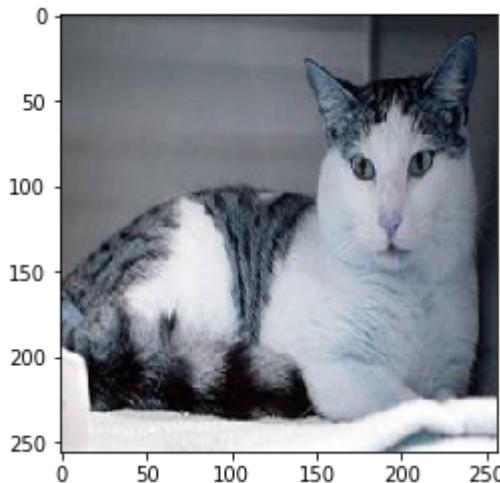


The Predicted Testing image is =horse verify below





The Predicted Testing image is =horse verify below



The Predicted Testing image is =horse verify below

