UNIVERSITY OF BUEA

P.O Box 63,
Buea, South West Region

REPUBLIC OF CAMEROON

PEACE-WORK-FATHERLAND

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

COURSE CODE: CEF 444

COURSE TITLE: AI AND MACHINE LEARNING

TRAINING AN ANIMAL CLASSIFIER MODEL USING KERAS

GROUP MEMBERS

NDZO DANIEL UGHE (FE18A045)
OMBUCHUM MICAH ANJI (FE18A059)
CHI CLAUDETTE MAH (FE18A018)
BANTI JEANIFAR NAHBILA (FE18A015)

Course Instructor: **Dr SOP DEFFO LIONEL**

What is Keras

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. It is user friendly, easily extensible, and works with python.

Overview of the project

This animal classifier is designed to classifier 5 different animals, that is; tiger, camel, sheep, pig and horse. The five different types of animals were downloaded in bulk separately. The images where then converted into a format that was suitable for the model. After the conversion, the images were divided into test and training dataset (90% was used for training and 10% used for testing). After some analysis, the model was compiled, trained, tested, and did predictions on some images.

Dataset

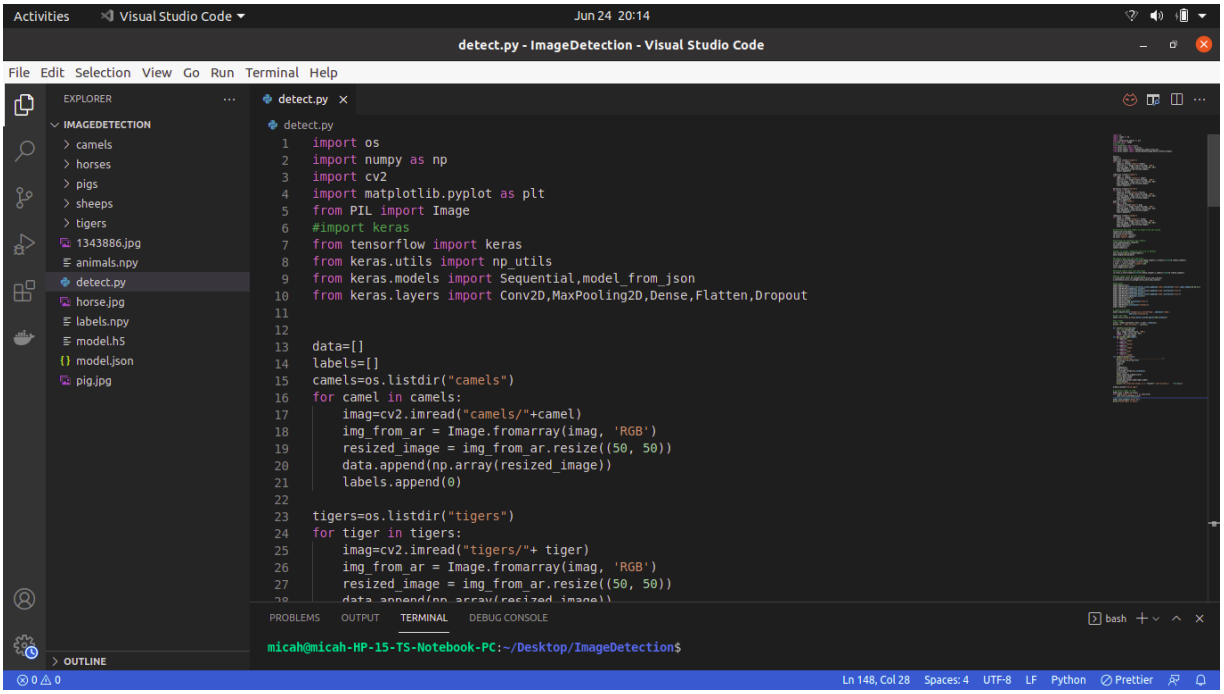
- 100 photos of tiger
- 100 photos of pigs
- 100 photos of sheep
- 90 photos of camel
- 90 photos of horse
- 90% of each animal was used for training and 10% for testing.

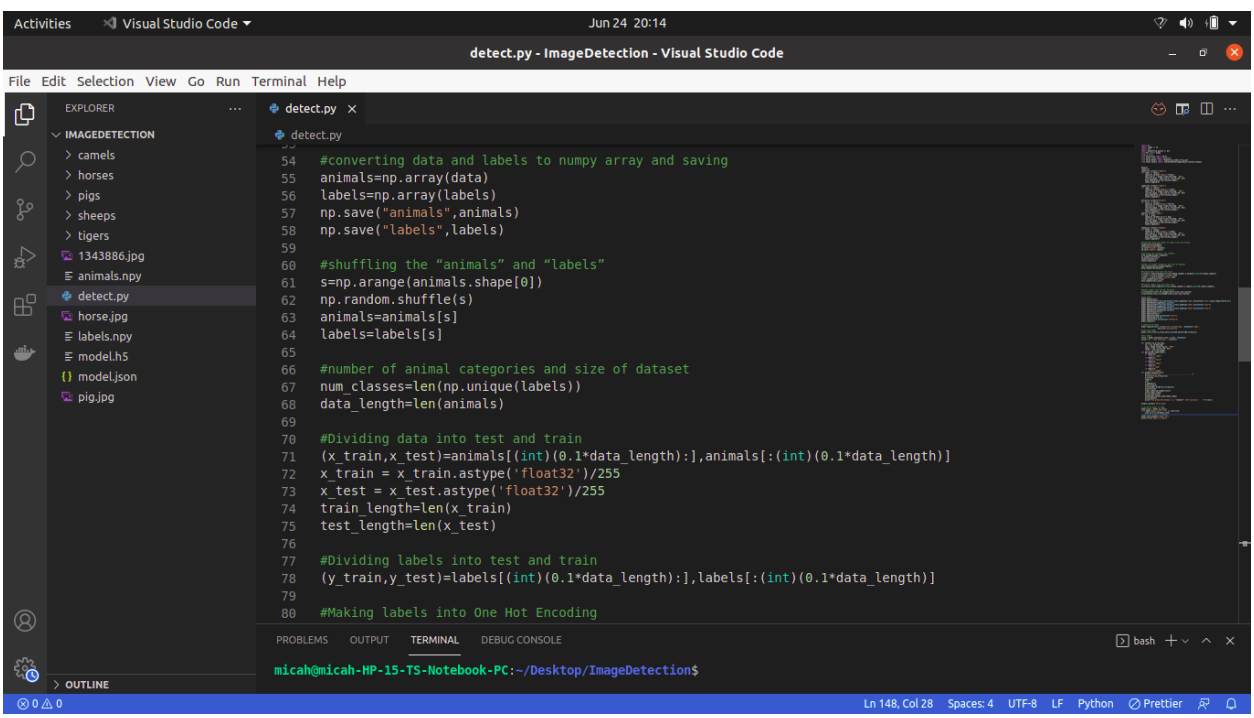
Libraries used

- Tensorflow (for building models. Keras runs on top of it)
- Keras(Developing and evaluating deep learning models)
- OpenCv (For computer vision)
- Numpy(For working with arrays)
- Pillow(For image manipulation)
- Matplotlib(Plotting library for pyhton)

Screenshots of the working classifier

Screenshots of the source code





Activities Visual Studio Code Jun 24 20:15

detect.py - ImageDetection - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

IMAGEDETECTION

- camels
- horses
- pigs
- sheeps
- tigers
- 1343886.jpg
- animals.npy
- detect.py
- horse.jpg
- labels.npy
- model.h5
- model.json
- pig.jpg

OUTLINE

detect.py

```python

#make model
model=Sequential()
model.add(Conv2D(filters=16,kernel_size=2,padding="same",activation="relu",input_shape=(50,50,3)))
model.add(MaxPooling2D(pool_size=2))
model.add(Conv2D(filters=32,kernel_size=2,padding="same",activation="relu"))
model.add(MaxPooling2D(pool_size=2))
model.add(Conv2D(filters=64,kernel_size=2,padding="same",activation="relu"))
model.add(MaxPooling2D(pool_size=2))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(500,activation="relu"))
model.add(Dropout(0.2))
model.add(Dense(5,activation="softmax"))
model.summary()

# compile the model
model.compile(loss='categorical_crossentropy', optimizer='adam',
              metrics=['accuracy'])
#train the model
model.fit(x_train,y_train,batch_size=50,epochs=100,verbose=1)

#test model
score = model.evaluate(x_test, y_test, verbose=1)
print('\n', 'Test accuracy:', score[1])

def convert_to_array(img):
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE bash

```
micah@micah-HP-15-TS-Notebook-PC:~/Desktop/ImageDetection$
```

0 0 Ln 148, Col 28 Spaces: 4 UTF-8 LF Python Prettier

Activities Visual Studio Code Jun 24 20:15

detect.py - ImageDetection - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

IMAGEDETECTION

- camels
- horses
- pigs
- sheeps
- tigers
- 1343886.jpg
- animals.npy
- detect.py
- horse.jpg
- labels.npy
- model.h5
- model.json
- pig.jpg

OUTLINE

detect.py

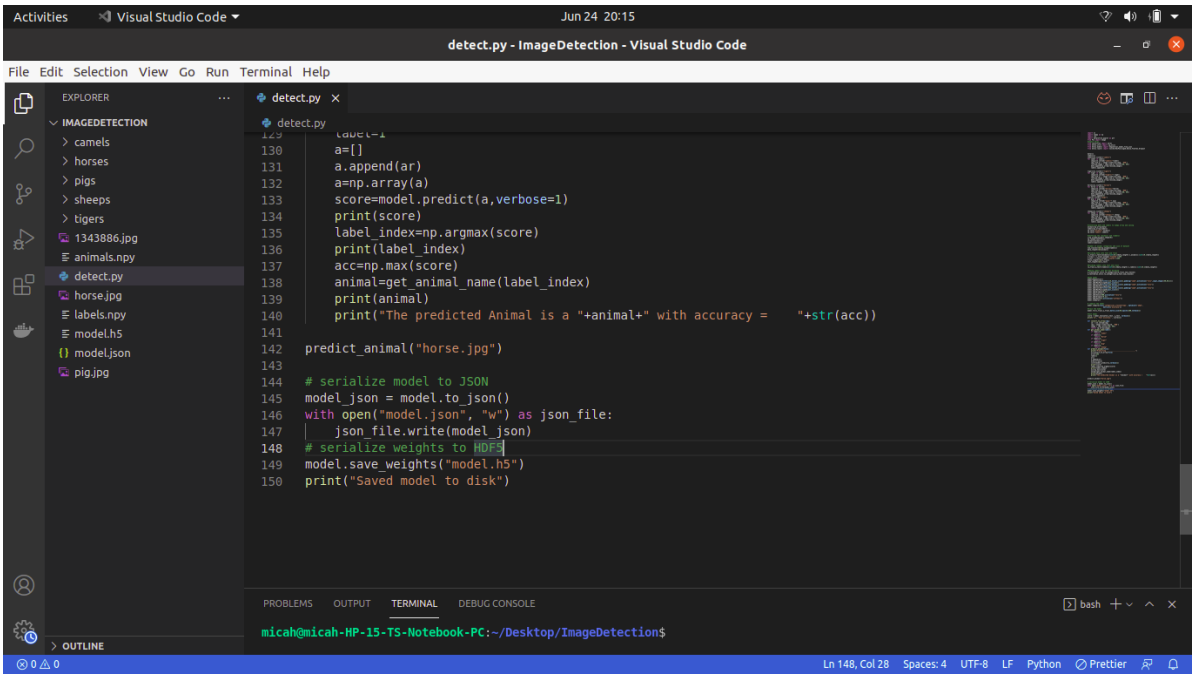
```python

def convert_to_array(img):
    im = cv2.imread(img)
    img = Image.fromarray(im, 'RGB')
    image = img.resize((50, 50))
    return np.array(image)
def get_animal_name(label):
    if label==0:
        return "camel"
    if label==1:
        return "horse"
    if label==2:
        return "tiger"
    if label==3:
        return "pig"
    if label==4:
        return "sheep"
def predict_animal(file):
    print("Predicting ...............................")
    ar=convert_to_array(file)
    ar=ar/255
    label=1
    a=[]
    a.append(ar)
    a=np.array(a)
    score=model.predict(a,verbose=1)
    print(score)
    label_index=np.argmax(score)
```

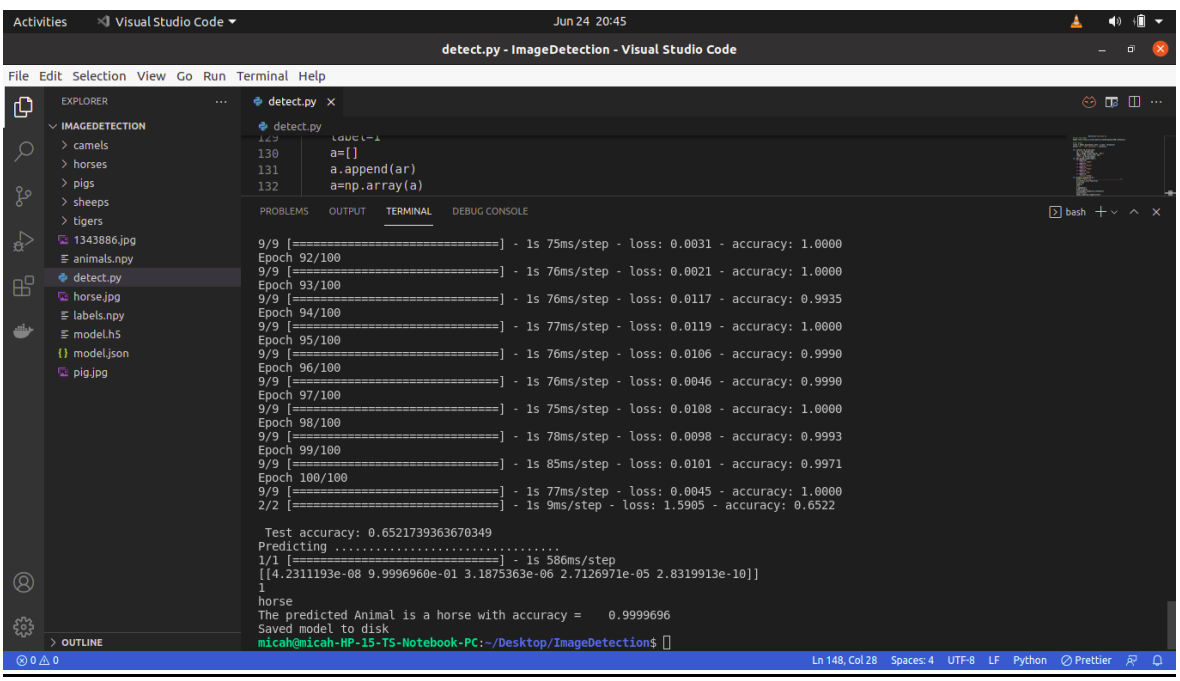
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE bash

```
micah@micah-HP-15-TS-Notebook-PC:~/Desktop/ImageDetection$
```

0 0 Ln 148, Col 28 Spaces: 4 UTF-8 LF Python Prettier



Output of the code



Conclusion

The model created is around 80% accurate because it was not fed with sufficient data. But it is still able to classify the different types of animals it was trained with.