Project Instructions

This project shows how advancement of data science deprecating security technologies. Specifically, we will using deep learning to break a CAPTCHA system called [Really Simple CAPTCHA](), which is a WordPress.org plugin. Try a demo here: https://contactform7.com/captcha/

CAPTCHA stands for "completely automated public Turing test to tell computers and humans apart", which is used to verify that a user is not a bot.

We require you to have some background knowledge in (1) Python programming and (2) Jupyter Notebook. If you do not already have the background, please check the following links for self-study:

- A quick tutorial on Python: http://cs231n.github.io/python-numpy-tutorial/
- A quick tutorial on Jupyter Notebook: http://cs231n.github.io/ipython-tutorial/
- How to set up your Python and Jupyter Notebook environment: http://cs231n.github.io/setup-instructions/

In this project, you will learn and use Python libraries including OpenCV, Scikit-learn, TensorFlow and Keras. OpenCV is used for image processing, Keras on top of TensorFlow is used for building deep learning models, and Scikit-learn is a machine learning library and we will use its preprocessing module.

Please open assignment.ipynb in the assignment folder, and following the instructions step-by-step to complete this notebook file. Most of the codes are provided but you are required to complete the code where you see

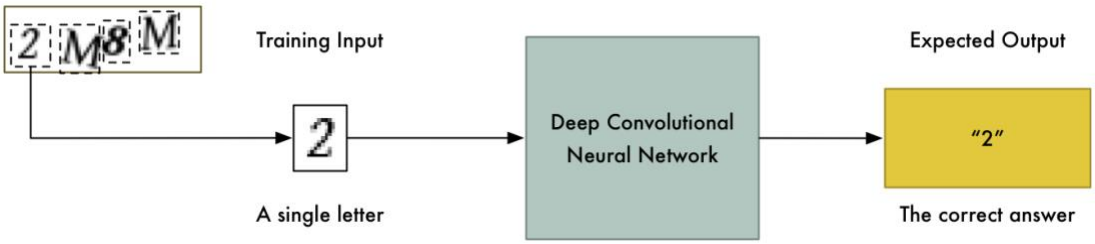
```
#########################################################################
# TODO: your code here ...                                              #
#########################################################################
```

You may also see some questions that you need to answer, marked by:

Type Your Answer Here:

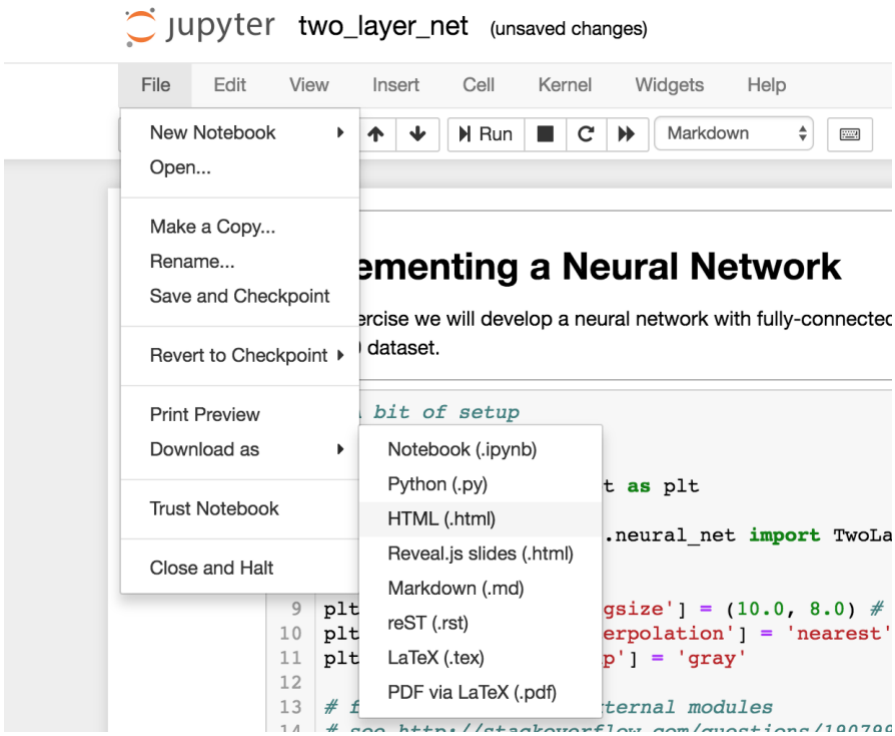
The notebook file provides instructions that are self-explanatory, and can be divided into 3 steps:

1. Step 1: Extract single letters from the training CAPTCHA images provided;



2. Step 2: Train the neural network to recognize single letters, where the network is built using Keras;
3. Step 3: Use the model to break CAPTCHA images not seen before.

You are expected to answer all questions and complete all codes to fill in, and save the notebook file as an HTML file:



Errata (!!! Please read !!!)

1. If a cell containing the following line reports error:
contours = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[1]

Replace it with the following line and see if the error is fixed (this is due to OpenCV version issue):

```
contours = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[0]
```

2. If the cell containing “plot_model” reports error, note that you need to install graphviz on your OS. For example, for mac you run:

```
!brew install graphviz
```

Also make sure that the related python packages are installed:

```
!conda install pydot -y
!conda install graphviz -y
```

After these operations, if it still does not work, try to replace

```
from tensorflow.keras.utils import plot_model
```

with

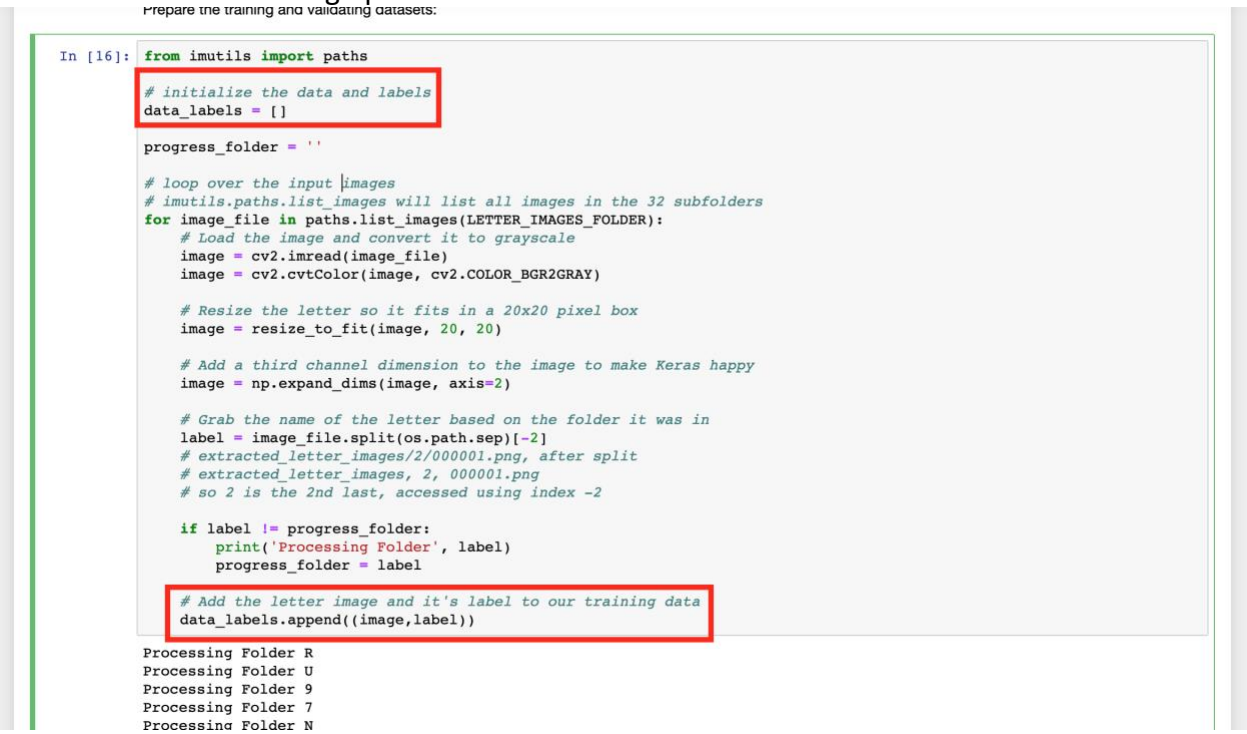
```
from keras.utils import plot_model
```

3. Note that we need to shuffle the data and labels before model.fit(.), since otherwise, only the early letters are in the training data, and validation is on later letters.

```
model. fit(x=None, y=None, batch_size=None, epochs=1, verbose=1, callbacks=None, validation_split=0.0, validation_data=None, shuffle=True, class_weight=None, sample_weight=None, initial_epoch=0, steps_per_epoch=None, validation_steps=None, validation_freq=1, max_queue_size=10, workers=1, use_multiprocessing=False)
```

The ‘shuffle’ argument does not work on validation data here

Please make the following updates:



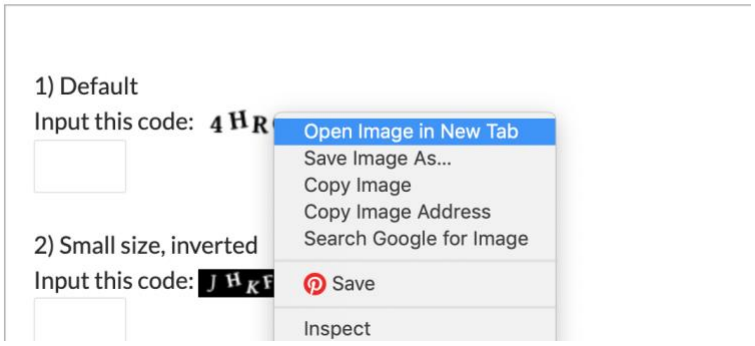
```
Processing Folder K
Processing Folder L
Processing Folder 2
Processing Folder Y
Processing Folder 5
Processing Folder P
Processing Folder W
```

```
In [17]: # shuffling training data, as we will use part of it for validation and we need labels to be balanced

         import random

         random.shuffle(data_labels)
         data = [x[0] for x in data_labels]
         labels = [x[1] for x in data_labels]
```

Convert lists into NumPy arrays as required by Keras for input + data normalization:

```
In [18]: # scale the raw pixel intensities to the range [0, 1] (this is important!!!)
         data = np.array(data, dtype="float") / 255.0

         # you may also use std rather than 255, and center the data by mean as follows:
         #data = (np.array(data, dtype="float") - data.mean()) / data.std()
```

4. If the image to test is not available, go to https://contactform7.com/captcha/



Get the URL there for use. You can get any number of images in this way.
If wget does not work, just download to local disk folder, or if you are using mac, try:
!brew install wget