Natural Language Processing
Project 1: OCR Error Correction using Language Modeling

The objective of this project is to apply Language Modelling on improving the output quality of an optical character recognition (OCR) system.

The OCR baseline system

The IAM Handwriting Database contains forms of handwritten English text which can be used to train and test handwritten text recognizers and to perform writer identification and verification experiments.

We implemented a Neural Neatwork (NN) OCR system which maps an image (or matrix) M of size W×H to a character sequence (c1, c2, ...) with a length between 0 and L. The text is recognized on character-level, therefore words or texts not contained in the training data can be recognized too (as long as the individual characters get correctly classified).

$$NN: \underbrace{M}_{W\times H} \rightarrow \underbrace{(C_1, C_2, \ldots, C_{1n})}_{0\leq n\leq L}$$

Equation 1: The NN written as a mathematical function which maps an image M to a character sequence (c1, c2, …).

You can download the model from Blackboard where you can find the file Project1NLP_Model.zip that contains a script “run.py”, a folder “model” and some test files.

To use the pre-trained model you need first to pre-process the input images to transform it to IAM format where you need to:
- crop the image;
- increase its contrast and
- thicken the lines

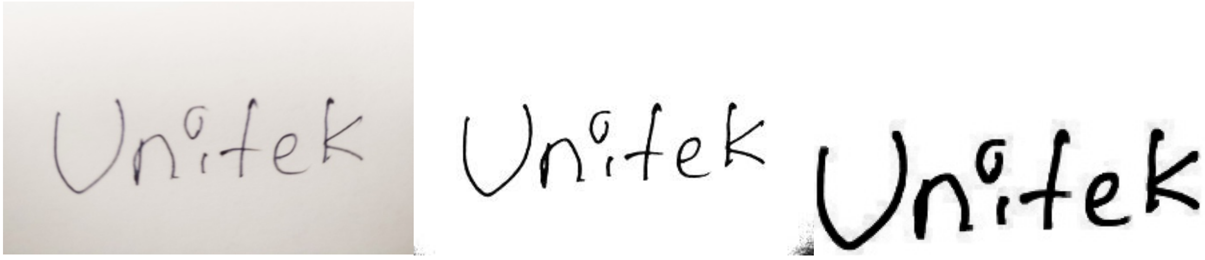
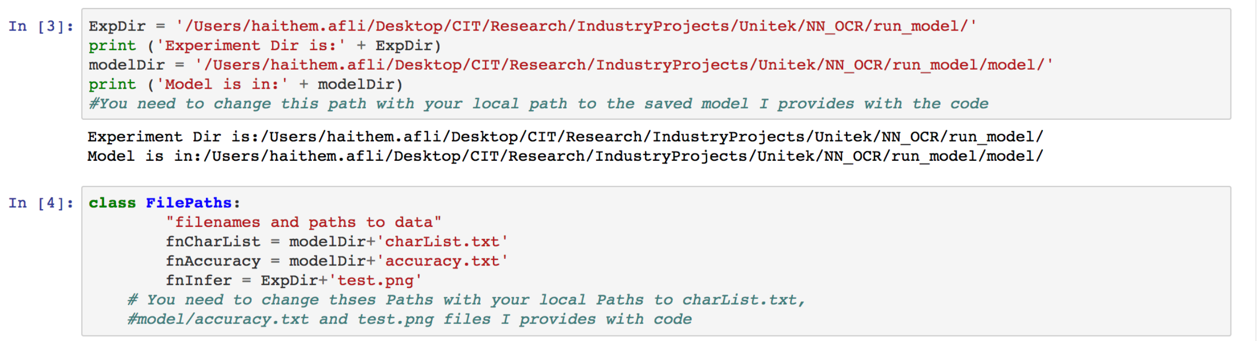


Figure 1: Pre-processing steps of an input image.

Once your input image is ready, you can use the given model by running the following command: **Python run.py**

You need to make sure that you are successfully updated your paths for the working directorie "ExpDir" and "modelDir", and if necessary the files paths as you can see in the following figure.



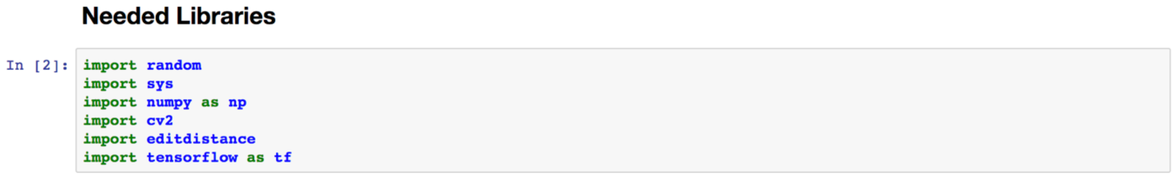
```
In [3]: ExpDir = '/Users/haithem.afli/Desktop/CIT/Research/IndustryProjects/Unitek/NN_OCR/run_model/'
        print ('Experiment Dir is:' + ExpDir)
        modelDir = '/Users/haithem.afli/Desktop/CIT/Research/IndustryProjects/Unitek/NN_OCR/run_model/model/'
        print ('Model is in:' + modelDir)
        #You need to change this path with your local path to the saved model I provides with the code

        Experiment Dir is:/Users/haithem.afli/Desktop/CIT/Research/IndustryProjects/Unitek/NN_OCR/run_model/
        Model is in:/Users/haithem.afli/Desktop/CIT/Research/IndustryProjects/Unitek/NN_OCR/run_model/model/

In [4]: class FilePaths:
                "filenames and paths to data"
                fnCharList = modelDir+'charList.txt'
                fnAccuracy = modelDir+'accuracy.txt'
                fnInfer = ExpDir+'test.png'
            # You need to change thses Paths with your local Paths to charList.txt,
            #model/accuracy.txt and test.png files I provides with code
```

Figure 2: Paths you need to modify before running the pre-trained model.

You need, also, to install the following python libraries if they are not already in your python3 environment.

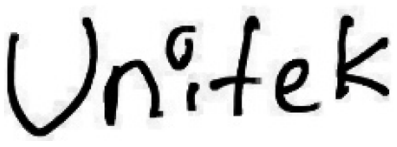


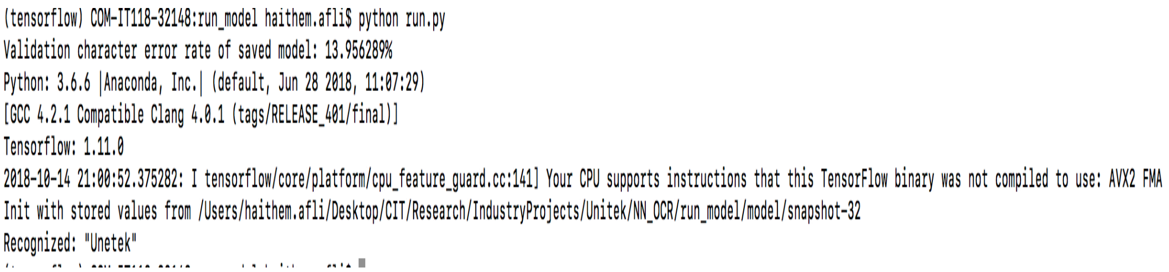
Needed Libraries

```
In [2]: import random
        import sys
        import numpy as np
        import cv2
        import editdistance
        import tensorflow as tf
```

Figure 3: Python libraries you need to install before running run.py script.

Executing run.py script will give the following result if you use the given "Unitek" image.





```
(tensorflow) COM-IT118-32148:run_model haithem.afli$ python run.py
Validation character error rate of saved model: 13.956289%
Python: 3.6.6 |Anaconda, Inc.| (default, Jun 28 2018, 11:07:29)
[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)]
Tensorflow: 1.11.0
2018-10-14 21:00:52.375282: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
Init with stored values from /Users/haithem.afli/Desktop/CIT/Research/IndustryProjects/Unitek/NN_OCR/run_model/model/snapshot-32
Recognized: "Unetek"
```

As you can see the system recognises the word "**Unetek**" from the input image. It's a good result but we still can improve it by trying to correct the misrecognised **i** (recognised as **e**).

Your task in this project is to add a new module that takes the recognised word and try to correct it using a character/word based language models.
This includes detecting characters given an image containing a line of text and denoising the output of the characters by using a language model.