UNIK4690 Project

Akhsarbek Gozoev - akhsarbg
Sadegh Hoseinpoor - sadeghh
Key Long Wong - keylw

May 19, 2018

1 Project description

The purpose of the software is to recognise text from any surface with uneven lighting. Hence this falls under the "Optical character recognition" (OCR) problem

As OCRs are still a challenging task even for companiese like Google, ref. reader to Googles OCR translator application on smartphones; "Transalte", drawbacks such as; difficulty finding all the text on the photo becasue of lighting, noise etc., therefore we will have to limit our software significantly.

1.1 Initial limitations

- English alphabet + numbers [0-9]
- Homogeneous background
- Computer printed text

1.2 Project components

The group have come to the conclusion that the OCR software has 3 main components to it.

1. *Text segmentation*
   - Finding text on an image and returning the text segments
2. *Preprocessing*
   - Do preprocessing on the segmented text; rotation, symbol segmentation, etc.. (preprocessing from its definition, should be done first, however because of simplification we assume we manage to segment out text first.)
3. *Classification*
   - Classification of the symbols

Additionally there is one more very important component for this OCR software to work, ***labled data***. Even though one might not need to code for this part, a good pool of labeled data is needed to be able to classify symbols. More on this later.
(4. *Data classification - Gathering labeled data to train a classification algorithm*)

1.3 Component description

This section we will describe our thoughts on how we plan to solve each component, in the form of algorithms, APIs, and datasets. Note that this is our initial thoughts, not necessary the solution we will end up implementing.

As we want to test proof-of-concept while we are coding, we will be making several simplifications. These simplifications will be described further under each component.

1.3.1 Text segmentation

Key fyll ut dette her
.
.
.
.
.
.

1.3.2 Preprocessing

Description

1.3.3 Classification

Description
At this point because of avalible knowledge and the intrest in Convolutional Neural network (CNN), we ended up trying to solve this part both with a CNN, and a Multilayer Peceptron (MLP or Deep neural network (DNN)).

Deep Neural Network
Multilayer peceptron neural networks are relatively straight forward to code, however the challinging part is to decide on good hyperparameters and to not overfit our network.
hyperparameters

- Number of Layers
- Number of nodes in each layer
- Activation functions
- Loss function

- Optimization function
- Learning rate
- Initialization of the weights and biases
- Number of epochs

Convolutional Neural Network

Limitation - proof of concept

1.3.4 Labled Data

Description
Labled data is needed because our classifiers need to be trained to understand the difference between the charecters. This is usually done by training a classifier with a set of training data, labels are needed in our case, since it is a supervised machine learning algorithm we want to use. As the training data is used to train the software, we will need data to test our software as well, hence the need for test data. The test data is used to get a measure of what the error rate of our software is, based on the results we can then tune the hyperparameters to get a better/smaller error rate. Lastly we will need validation data. This is an independent dataset that the software is not familiar with. The accuracy of the software on the validation set will then be a measure of how good the software can classify the characters.

Limitation - proof of concept
As we have limited us to the English alphabet and numbers ranging from [0-9], we will need labeled data for each of these 36 characters; training, test and validation sets. As the concept of classifying only numbers vs all 36 charecters does not differ that much, we will first see if we can solve the OCR problem with just numbers. Therfore we only need a dataset containing numbers at first. Thereafter we will search for a dataset containing all the characters we need.

Dataset

MNIST
This is a dataset containing handwritten numbers [0-9]. It has a training set of 60.000 examples and a test set of 10.000 examples. (ref. reader to http://yann.lecun.com/exdb/mnist/).

Report

Week 1 19.04.18

- Feedback on project proposal
- Overview of project
  - simplification
  - binary image → numbers → straight text → Classify
- init; github - atom
- first test of charcter Segmentation

Week 2 26.04.18

- Charcter Segmentation - Projection Histograms - OpenCV
  - By projection the histogram of the binary image on the Y-axis, we can find where the sentences/lines of text appears. Following, a projection histogram on the X-axis can discover where the charecters appear.

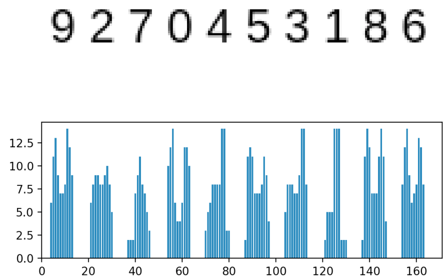
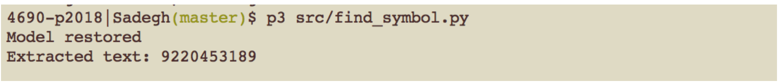


Figure 1: [0-9] segmented with projection histogram

- Classification - Perceptron neural network - TensorFlow
  - MNIST dataset - Datasett consisting of several thousand handwritten labeled numbers
    - ∗ Numbers ranging from [0-9]
    - ∗ Images are 28x28pixels
  - Hyperparameter tuneing
    - ∗ Activation function
    - ∗ Number of hidden layers
    - ∗ Nodes in hidden layers
    - ∗ Cost function
    - ∗ Optimazation function
    - ∗ Learning rate
  - Theoretic accuracy of the network with 2 hidden layers 98%
    - ∗ Measured accuracy 97%



```
4690-p2018|Sadegh(master)$ p3 src/find_symbol.py
Model restored
Extracted text: 9220453189
```

Figure 2: First output with classification. input see Figure 1

Week 3 03.05.18

- Rotation of text
  - Hough transform
  - *cv2.minAreaRect()*
- How to distinguish between upside-down, and verticle vs horisontal text segments
  - Classifiy in all 4 rotations, and choose the classification with highest avrage confidence
- Classification - Perceptron neural network - Error
  - Error rate too high, test-set accuracy 97%, validation set accuracy $< 50\%$
  - CNN - TensorFlow Estimator API
    - Challenging documantation; load/save models
  - Dataset - FNIST - Group contribution
    - Dataset including several fonts
    - English alphabet, and numbers [0-9]

Week 4 17.05.18

- ——
  - ——
  - ——
- ——