Handbook

Source Code Files

1. Src

1.1. App.py

This script is responsible for the whole web interface interaction. It can be thought of as the main script through which all other scripts are called. The functions in the script are called by the frontend files (html files). This script has to called to run the flask framework responsible for hosting the webpage on the local host. The port is defined within this script.“Usage —> python app.py”

1.2. Segmentation.py

This script is responsible for segmentation/extraction of words from a given image. It takes the input image of text and outputs each extracted word in a separate image in the Segmented folder in the dataset. "Usage --> python segmentation.py <ImgName> --optional<inverted?> --optional<height> --optional<width> --optional<pad size> “

1.3. Augmentation.py

This script is responsible for generating a copy of the input image where one of the augmented operations are performed. Taking as an input the source image, the script creates edited image in the augmented folder in the dataset. “Usage —> python augmentation.py <ImgName> <Operation>”.

1.4. DataGenerator.py

The dataGenerator.py script is responsible for generating multiple augmented images through performing the aumgentation.py script multiple times with different operations. “Usage —> python dataGenerator.py <ImgName>”

1.5. dbwriter.py

This script is responsible for converting the already generated database into a specific format (IAM database format). The images are renamed and added to a folder named “Sub” in the dataset. additionally, a text file named “words.txt” is generated which contains all the entries in the dataset along their target text.

1.6. Main.py

This is the main script for the training and recognition process. It purpose of using it can be defined by its call. For training: “Usage —> python main.py - - train”, As for recognition: “Usage —> python main.py - -wordbeamsearch ”. WordBeamSearch is one of the decoding algorithms, explained in more details in the thesis.

1.7. Model.py

This script defines the specifications for the model used in the training process (cnn, rnn, etc..). The script is fully documented for more details. This script is called by the main.py script.

1.8. Output.txt

This text file is used to save the output log from the terminal. This content is then used to be displayed in the web interface to enable users to monitor the running processes without looking through the actual terminal console. The following command has to be add to any python call to add its print log into the output.txt “| tee -a output.txt”

1.9. Templates

This folder contains the html images rendered by the flask framework through the app.py script. Any html page created should be placed inside this folder.

1.10. Static

This folder contains all the style files used for the web interface run by flask. Any css or js files should be placed inside this folder.

1.11. Ops

This folder includes scripts for all the implemented operations that can be used for data augmentation. These scripts are used by the augmentation.py script.

2. Dataset

2.1. Segmented

This folder contains the images segmented by the segmentation script. This folder contains sub folder, each of a name that represents all the images inside. For example all images containing the word “cheese” will be in a sub folder called “cheese”. This folder and all other sub folders can be generated if they are not created yet i.e. user doesn’t have to create them in advance.

2.2. Augmented

This folder is similar to Segmented folder in logic; however, it contains the augmented images generated by the dataGenerator.py script.

2.3. Raw

This folder contains all images uploaded through the web interface either for recognition or training.

2.4. RawGS

This folder contains all images ,converted to grayscale, uploaded through the web interface either for recognition or training. It is mainly used for debugging process.

2.5. words.txt

This files contains all the words in the database that are used in the training process. This file is auto generated during the database format conversion.

3. Model

This folder contains the files generated at the training process. If you want to train the system from scratch, you have to manually delete all files in that folder. If not deleted, the files are reused in the next training process which is useful if you want to retrain the system after adding new data to the already existing dataset.

4. Venv

This folder is generated by the virtual environment that contains all the libraries and dependencies used in the project. This enables the system to run on the operating system without any prior setups. It has to be run before the usage of the system scripts.

“Usage : $ source ./venv/bin/activate # sh, bash, ksh, or zsh\\n”