SUR project

Attila Kovacs (xkovac60)

Project Overview

The aim of this project was to create a system for person identification based on a head shot and a short recording of the person answering questions in English. Due to the limited ammount of annotated data avalable for each person, and the limitation that models trained on data other than the provided can't be used, this assigment was a challenging task.

Audio Classification System for Speaker identification

My first step was to listen to some of the recordings. My first note was that the microphone gain was low or the microphone was too far from the speaker. So as the first step I've decided to increase the amplitude of the recorded signal. I've also noted that all of the recordings have a lot of silence, in some of the recording less than 2s of the signal is from the speaker really talking. This is pretty normal for these datasets and there are tools which can remove the silent parts.

After cleaning up the recordings I've fitted a gausian mixture model(GMM) for each speaker. This aproach had an accuracy on the test set of around 14%. This was better than random choice (~3%), but still not great. Instead of using the audio samples I've decided to use Mel-frequency cepstral coefficients to extract specific features from the audio signal. This approach had an accuracy on the test set of ~30%.

My next step was to somehow augment the dataset to create more instances for each speaker. For this I've decided to segment the non-silent parts of the recordings into 3s chunks, for recordings where no such segment exists the recording is added as is and an augmented variant is also added(to try to balance the distribution). Because by using this extraction method the ammount of available data was higher I had enough data to create an universal background model (UBM) and then fine-tune it for each person.

Using Exhaustive Grid Search the hyperparameters were fine-tuned. With this approcach I was able to achieve an accuracy of 91% on the test set.

Image Classification System for person identification

When I first started thinking about this part I've thought that this would be easy. I just have to model a ResNet or similar CNN and train it. This unfortunately was not the case. Even with fine-tuning the hyper parameters the resulting models were not performing well. The accuracy was around 15%, I expected much more.

I suspect that the convolutional neural networks were not performing well due to the low ammount of annotated input data. Even with augmentations we can't increase the dataset size to the same degree as with the audio recordings.

My next implementation was a simple SVM on the raw image data based on https://www.geeksforgeeks.org/image-classification-using-support-vector-machine-svm-in-python/ (https://www.geeksforgeeks.org/image-classification-using-support-vector-machine-svm-in-python/)

This implementation had an even worse accuracy than the CNN. So the last thing that remained was to somehow extract the necessary features from the image. For this I've utilized Principal Component Analysis (PCA). This resulted in a model with an accuracy of 40%. After researching a bit I stumbled upon Histogram of Oriented Gradients (HOG). Both of the feature extraction methods alone resulted in the same accuracy of 40%, but by combining them I was able to reach an accuracy of 71% on the test dataset. To increase the number of images in the dataset I've decided to also use the test data during the training step, so I don't know the final accuracy of the model, but it should be around 70%.

Usage

- The scripts were developed using Python 3.11.12 other versions were not tested
- To not mess up the system packages creating a virtual environment is recommended (`python3 -m venv venv && . venv/bin/activate`)
- install required packages using pip: `pip install -r requirements.txt`
- Put the train and eval datasets into the SRC directory, or modify the configuration in each file accordingly.

Audio

The audio classification consists of 2 python scripts:

- **train_audio.py**: Trains a Universal Background Model (UBM) and Gaussian Mixture Models (GMMs) for speaker identification.

- **eval_audio.py**: Evaluates the trained models on query files.

  Assuming that the train and test datasets are located in ./data/train and ./data/dev directories, respectively, and the sound files to be evaluated are located in ./eval directory, you can run the scripts as follows:

```
python train_audio.py
python eval_audio.py --model_dir saved_models --query_dir eval --output audio_results.txt
```

if this is not the case you may need to modify the configuration in `train_audio.py` accordingly:

```
TRAIN_DIR = "data/train"
TEST_DIR = "data/dev"
MODEL_DIR = "saved_models"
```

Image

The image classification consists of only 1 python script:

- **image.py**: Trains a Support Vector Machine (SVM) classifier using Histogram of Oriented Gradients (HOG) and Principal Component Analysis (PCA).

  Assuming that the train and test datasets are located in ./data/train and ./data/dev directories, respectively, you can run the script as follows:

```
python image.py
```

The script will output the results to a file named `image_res.txt`. If the above assumption is not the case you once again have to edit the configuration in `image.py`:

```
TRAIN_DIR = "data/train"
TEST_DIR = "data/dev"
EVAL_DIR = 'eval'
OUTPUT = 'image_res.txt'
```

Fusion

A script is also included which takes the results from the classificators above and using z scores, combines them into a final result.

- **fusion.py**: Combines the results of multiple classifiers using z-scores. To run this script you need to have the output files from the previous scripts (e.g. `image_res.txt`, `audio_res.txt`). This script creates 3 fused variants of the results:
  - **fusion_1img_1aud.txt**: Combines the image and audio results using z-scores, without aplying any weights.
  - **fusion_1img_2aud.txt**: Combines the image and audio results using z-scores with weights applied in a way that the scores from the audio results have a bigger impact on the final score.
  - **fusion_2img_1aud.txt**: ombines the image and audio results using z-scores, with weights applied in a way that the scores from the image results have a bigger impact on the final score.

Conclusion and further improvements

I really enjoyed working on this assignment, and to learn and try out how the methods presneted on the lectures are used in the real world. I think it was a great experience for me. The end results might not reflect it but I've spent a considerable amount of time on this project. To further improve the accuracy of the identifier I think the next step would be to use a pretrained base model and fine-tune it on my dataset, this method could improve the accuracy a lot.