Affective Computing

Introduction to exercises

matti.matilainen@oulu.fi

General information

- Individual work
- First two exercises 10 points each, the third exercise is worth 20 points.

Date	Published	Return deadline
12.9.2025	Exercise 1 – Data annotation	
19.9.2025	Exercise 2 (Python) – Facial expression recognition	Exercise 1 – Data annotation
26.9.2025	Exercise 3 (Python) – Multimodal emotion recognition	Exercise 2 (Python) – Facial expression recognition
10.10.2025		Exercise 3 (Python) – Multimodal emotion recognition

Exercise 1 – Data annotation

- The contact person: Fan Kang (fang.kang@oulu.fi)
- Idea in a nutshell: Use a dataset annotation tool to record micro gesture time stamps from video files.
- Everyone who has enrolled to the course has already been assigned several videos they must annotate. Go and see your ID from Moodle. You can find your videos with your ID from https://drive.google.com/drive/folders/1cb9oxjj1WI-ir1DSXkHVBtOlT5hDyUm2
  - You can also find the examples of all the Micro gesture classes from this google drive link
- If you want to read more about the dataset: https://arxiv.org/pdf/2107.00285

Exercise 1 – Data annotation

- Annotate the videos, export the data as txt files
- Merge the annotation txt files with a python tool (available in Moodle) to a csv file.
- Take part in a Kaggle data annotation contest. For this you need to create a Kaggle account if you already dont have one.
  - You can try several times
  - If you get 60 points from the contest, you get 4 points for this course. The top three in the Kaggle contest get extra points.
- Extra task: Try to do the data annotation automatically by using an open source model for activity segmentation.
  - OpenGVLab InternVideo / InternVideo2 (video understanding foundation models).
  - VideoMAE (masked autoencoder for video).
  - ActionFormer / SlowFast (gesture/action recognition baselines).
  - GPT-4V / LLaVA / Video-LLaMA (multimodal LLMs with video input, if available).

Exercise 1 – Data annotation

A micro gesture in the dataset is something that the person does actively. For example one of the gestures is caled ”crossing the fingers”.

- You should mark the beginning time stamp for this gesture when the fingers start moving closer together.
- **The ending time stamp should be marked when the fingers stop moving and not when the fingers are no longer crossed**
  - In practice this means that the micro gestures are short. For example, the person might cross their fingers and leave them crossed for a long time. We are not interested in how long they keep their fingers crossed.
- The same rule applies to all micro gestures
- When a person performs a gesture that is not a micro gesture we are interested in, you must give it label 99 ”Non micro gesutre”

Python exercises – setting up the environment

- Contact person: matti.matilainen@oulu.fi
- The both exercises are implemented using python and jupyter notebook.
- Two ways to setup your system
  - The easy way: use Google Colab
  - The other way:
    - Install miniconda (https://www.anaconda.com/docs/getting-started/miniconda/install)
    - Open a miniconda prompt and create an environment with python 3.7
      - Others will likely work too but you have to have a version that works with the dlib library you are using
    - Install the following libraries with pip: jupyter numpy matplotlib scikit-image scikit-learn
    - Install dlib. You can find pre-built wheel files from https://github.com/z-mahmud22/Dlib_Windows_Python3.x
      - This site contains wheel files for Windows, if you have any other OS, you have to find the correct files or compile them from source yourself.

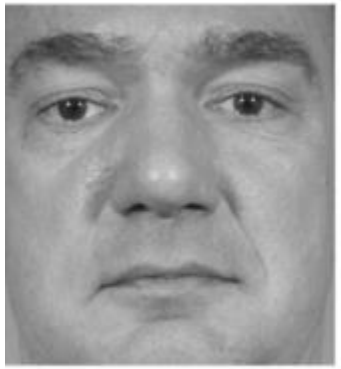
Python exercises – setting up the environment

```
conda create --name yourname python=3.7.*
conda activate yourname
pip install jupyter numpy matplotlib scikit-image scikit-learn
pip install dlib-19.22.99-cp37-cp37m-win_amd64.whl
jupyter notebook
```

Note: the dlib-19.22.99-cp37-cp37m-win_amd64.whl is the precompiled wheel file that you have downloaded using the insturctions from the previous slide.

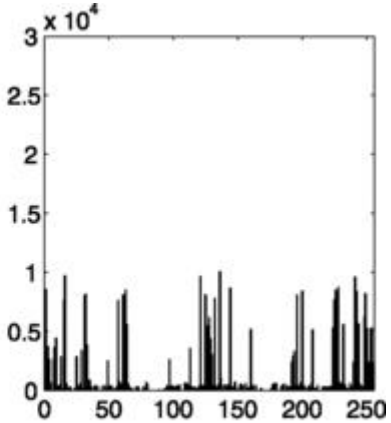
Exercise 2 – Facial expression recognition

- If you run this on Colab make sure that you upload all the necessary files. There are several of them.
- Find face landmarks with dblib (shown in the image)
- Study face normalization
- Build a program that recognizes facial expression from videos. Use the SVM classifier and LBP-TOP features.

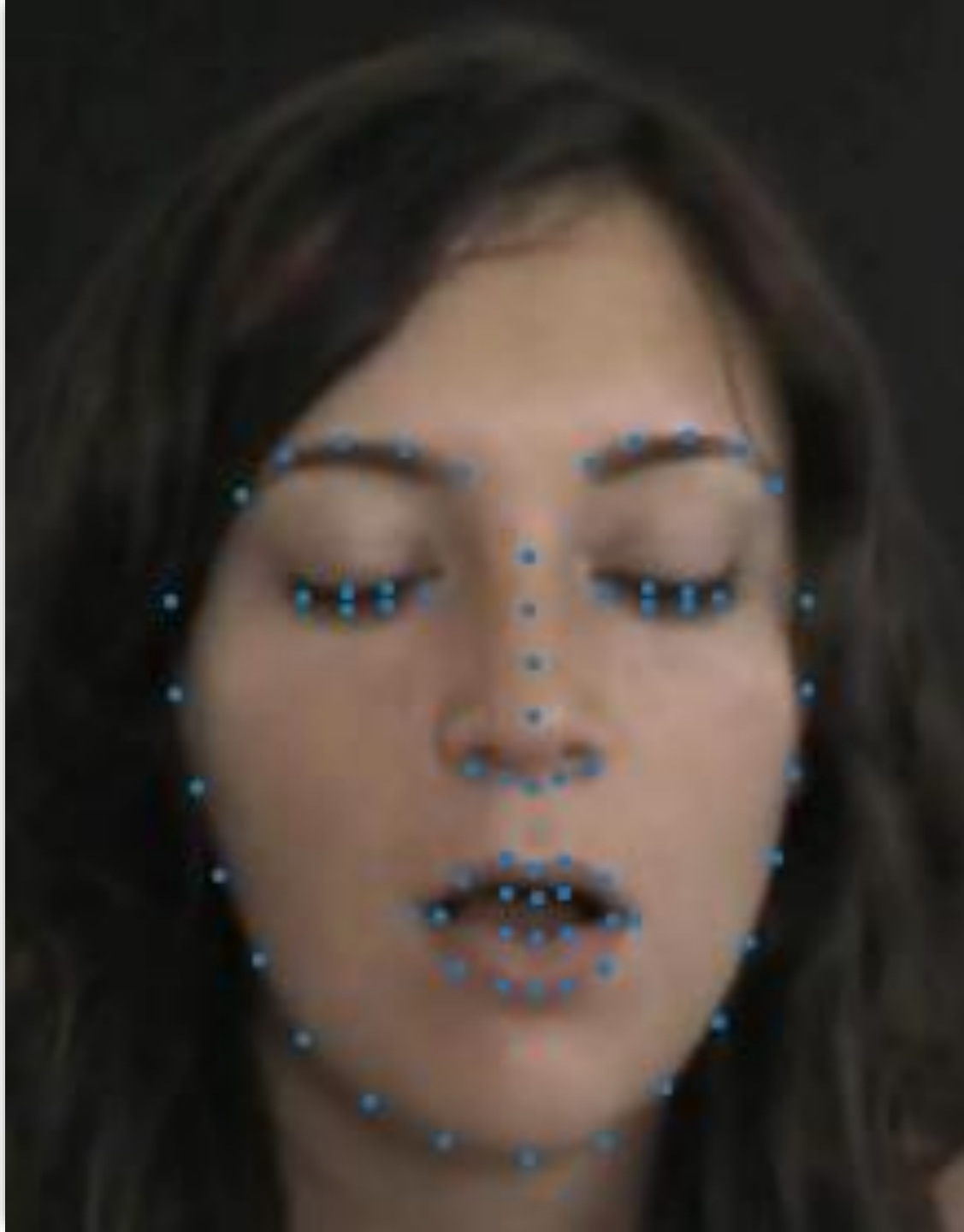


Input image

LBP image



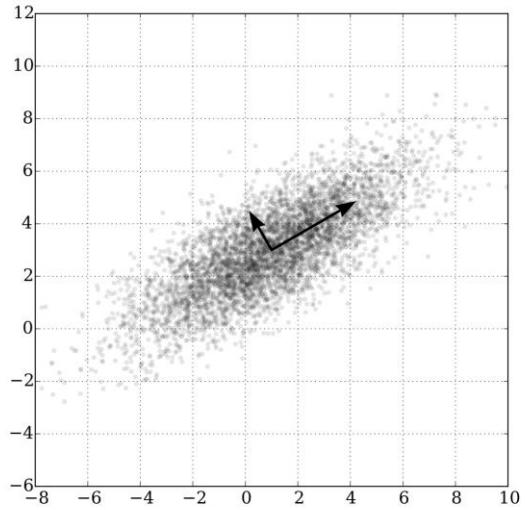
LBP histogram



Exercise 3 – Multimodal emotion recognition

- Data modalities: video and audio.
  - LBP-TOP features (the same features as in the previous exercise) for video and Prosodic features (the pitch, how the pitch changes, tempo, intensity = ways of speaking that make our voices unique) for audio.
- Goal: to build a classifier that recognizes emotions

- In this exercise we first process both of the modalities separately with PCA and then concatenate the features to build a classifier.
- In the second part, we use the features calculated with PCA. But before combining the features we use a method called CCA to reduce the dimensionality of both video and audio features to 15.
  - In the notebook you are asked to study what is the effect of CCA
- Train an SVM model for classification



PCA