THE README FILE OF THE AU DETECTOR

by Evangelos Sariyanidi

August 2, 2016

1 Which AUs are detected

Currently the following AUs are detected:
1/2/4/6/12/15/20/24/25/26/27/45
According to my personal observations the following AUs are detected better than the others:
1/2/4/12/15/25/26/45
However AU1+4 is not detected very well.

2 Head pose information

2.1 For Live AU detection (with visual output)

When running on a live stream, the interface outputs colors to steer the user toward a frontal appearance, where AUs are recognised more easily. When the user is away from a frontal look, the colors become red, otherwise the colors become green. The color bar at top warns the user about the PITCH angle whereas the color bar at the side warns the user about the YAW angle.

2.2 For offline AU detection

For offline processing we provide values that indicate how much a face has deviated from a frontal appearance (see Section 4). By comparing those values with a threshold, one can ignore the faces that are not frontal. We have two thresholds; one for the pitch angle and one for the yaw angle. Currently those thresholds are set as 0.55 and 1.05 based on simple visual inspection.

3 Run arguments

3.1 For Live AU detection

When running the program for live AU detection, one or two arguments should be provided in the form of: 1) configuration file 2) [OPTIONAL] session name. If session name is not provided, it will be decided automatically based on timestamp. The list of arguments looks like this:

```
./LIVE config-file.cfg session-name
```

3.2 For AU detection on offline files

When running the AU detector for offline AU detection, three arguments should be provided in the form of: 1) configuration file 2) video to process AND the calibration file (calibration file is optional) separated by semicolon, 3) name of the session (i.e. the name of the folder under which recordings will be provided). Then the list of arguments looks like this:
`./LIVE config-file.cfg video-file.avi;optional-calibration-file.avi session-name`

4 Where is the output of AU detector? What is the form of the AU output?

The output of the AU detector is always stored under
`#AUdetectorPath#/data/RECORDINGS`
Each session is stored into a different folder, for example session1 is:
`#AUdetectorPath#/data/RECORDINGS/session1`
The following are stored (all optional), the text within the brackets shows the subdir and the extension with which the files are stored:

1. AU results: [AUresults/*_results.txt]
2. AU results for each feature type (#feature_type#) separately: [AUresults/*_results_#feature_type#.txt]
3. Head Pose Yaw (this is NOT the pose in angles but a value that increases as a face in a given image deviates from the frontal YAW angle): [AUresults/*_headposeyaw.txt]
4. Head Pose Pitch (this is NOT the pose in angles but a value that increases as a face in a given image deviates from the frontal PITCH angle): [AUresults/*_results_headposepitch.txt]
5. Raw Frames [frames/*_raw.png]
6. Processed Frames [frames/*_appearance.png]

All the above are stored in a one-file-per-frame basis. Each filename starts with a unique timestamp.

5 The feature types and classifiers fusion

5.1 Feature types

Currently there are four different feature types employed during AU detection. The list is below. The values within brackets indicate how those features are referred to within the config file and the software.

1. . [appearance] Appearance features: QLZM
2. . [shape] Shape features: x,y coordinates of landmarks. Coordinates are normalised to eliminate in-plane rotation, scaling and translation within the image.
3. . [ndiff-appearance] Appearance features that describe information that is differential w.r.t. neutral frame. Currently these features encode the motion between a neutral frame (collected during calibration) and a given frame (i.e. the most recent frame during live AU detection). **These features REQUIRE calibration (see Section 6 below.)**

4. . [ndiff-shape] Shape features that describe information that is differential w.r.t. neutral frame. Currently these features encode the difference (i.e. subtraction) between facial landmark coordinates of a neutral frame (collected during calibration) and a given frame (i.e. the most recent frame during live AU detection). **These features REQUIRE calibration (see Section 6 below.)**

5.2 Classifiers and fusion

There is an SVM classifier trained for each AU separately with each of the feature types below. The fusion of the classifiers is done with a method that is simple and yet effective: An AU is deemed to be detected if all classifiers of that AU (i.e. classifiers trained with different feature types) agree.

There is an optional 2-stage classification for each feature type. Because AU detection is a largely imbalanced-data problem (i.e. there are few positives and many negatives for each AU class), we came up with a 2-stage classification strategy. For each feature type, we train two classifiers. First we train a classifier with all data, and then change the threshold so that the classifier has a very high True Positive (TP) rate (e.g. 0.95), even though the False Positive (FP) rate can also go up. Then we train a second classifier, but this time the negatives are only selected from the FPs of the first classifier. In other words, the second classifier is trained with difficult negatives only. Then, the two-stage classifier of a feature type deems that an AU is detected if both classifiers agree that the AU is detected.

The two-stage classification is optional, and can be turned off (i.e. use only first classifiers output) from the config file. See next section.

6 The CONFIG file

6.1 An exemplar file

Config files are easily parsed by a human. Each line in the file indicates the value for an option. More detail on each option as well as the default value for each option will be provided in the following section.

```
visualiseOutput=1
visualiseDetailed=1
sssAUlist=45
AUlist=1/2/4/6/12/15/20/24/25/26/27/43/45
storeRawFrames=0
use2stageClassification=1
storeProcessedFrames=0
storeCalibrationData=0
storeAUresults=1
storeDetailedAUresults=0
calibration=1
cameraId=0
featureTypes=appearance/shape/ndiff-appearance/ndiff-shape
```

6.2 Details on each option

Below is a detailed discussion on each option.

```
//! Whether to show the processed window during AU detection (must be 0 for processing with NAO)
bool visualiseOutput = 1;
```

```
//! Show details during visualisation:  perf. of each classifier individually
bool visualiseDetailed = 1;

//! A string that contains the list of AUs to recognise. E.g. "1/2/4/6"
std::string AUlist;

//! Whether to store raw (i.e. unprocessed) frames
bool storeRawFrames = 0;

//! Use both stages of each classifier
bool use2stageClassification = 0;

//! Whether to store  processed frames (i.e. frames with  AU results)
bool storeProcessedFrames = 0;

//! Whether to store calibration data
bool storeCalibrationData = 0;

//! Whether to store AU results (a text file for each frame)
bool storeAUresults = 0;

//! Whether to store detailed AU results, i.e. the performance of each classifier individually
bool storeDetailedAUresults = 0;

//! Calibration type = { 0: No calibration, 1: Simple calibration (i.e. first frame of sequence),
//2: Regular calibration (i.e. rotate head on all angles etc.)
int calibration = 0;

//! Camera (i.e. dev id)
int cameraId = 0;

//! The list of feature types to use for AU detection, separated by '/'.
//! E.g.: "appearance/shape/ndiff-appearance/ndiff-shape"
std::string featureTypes = ""
```

The features that start with "ndiff" can be used only if calibration is '1' or '2'.