Uppsala University

Intelligent Interactive Systems

1MD032 11619 HT2024

Assignment 1: Computer Vision

Author:
Abhinav Ramalingam

November 16, 2024

Contents

1 Introduction **2**

2 Objective **2**

3 PyFeat Visualization **2**

3.1 Do you agree with all its predictions? 4

3.2 What seems to confuse the system? 5

3.3 What are the cases tricky for a human observer? 6

4 AU Subset Selection **6**

4.1 Which AUs would you choose? 6

4.2 Problems of using too many features? 7

1 Introduction

Py-Feat is an advanced, open-source toolkit designed for efficient and accurate facial expression analysis, catering specifically to the needs of human behavior and computer vision researchers. It integrates state-of-the-art computer vision and deep learning algorithms to detect and interpret facial expressions, extracting crucial data such as Action Units, emotions, and facial landmarks from images and videos. By streamlining the process with intuitive and user-friendly features, Py-Feat addresses the limitations of existing tools, like OpenFace and iMotions-Affectiva, which may be costly, difficult to install, or lack transparency. Py-Feat empowers researchers with a simple interface for extracting and analyzing rich facial expression data, while also providing a platform for the integration and distribution of novel computer vision models [1].

2 Objective

The objective of this project is to use the Py-Feat facial expression analysis tool to extract and analyze facial data from a collection of images. Specifically, the goal is to:

1. **Image Analysis**: Use Py-Feat to analyze facial expressions in each image, overlay face bounding boxes and primary emotion, and save visualizations.

2. **Data Collection**: Extract AU activations for each face, store them in a CSV file with image and face identifiers.

3. **Data Analysis**: Separate AU data by valence (positive vs. negative), calculate the mean for each AU, and compute the absolute difference between positive and negative conditions.

4. **Visualization**: Sort AUs by the absolute difference of means and display the results in a graph, saving it as au_visualization.png.

3 PyFeat Visualization

The Py-Feat FEX detector uses a concept called Action Unit (AU) to detect Facial Expressions. The meanings of the AUs detected is based on the Facial

Action Coding System (FACS) by Paul Ekman [2]. Here's a brief overview of the AUs and their associated facial movements or expressions:

1. **AU01 (Inner Brow Raise)** – Raising of the inner part of the eyebrows.
2. **AU02 (Outer Brow Raise)** – Raising of the outer part of the eyebrows.
3. **AU04 (Brow Lowerer)** – Lowering of the eyebrows (frowning).
4. **AU05 (Upper Lid Raiser)** – Raising the upper eyelids.
5. **AU06 (Cheek Raiser)** – Lifting of the cheeks, often associated with a genuine smile.
6. **AU07 (Lid Tightener)** – Tightening of the eyelids.
7. **AU09 (Nose Wrinkler)** – Wrinkling of the nose, often associated with disgust.
8. **AU10 (Upper Lip Raiser)** – Lifting of the upper lip, often associated with disgust or contempt.
9. **AU12 (Lip Corner Puller)** – Pulling the corners of the lips upward, often associated with a smile.
10. **AU14 (Dimpler)** – The appearance of dimples on the cheek, usually associated with a smile.
11. **AU15 (Lip Corner Depressor)** – Pulling down the corners of the lips, often associated with sadness or anger.
12. **AU17 (Chin Raiser)** – Raising of the chin.
13. **AU20 (Lip Stretcher)** – Stretching of the lips horizontally.
14. **AU23 (Lip Tightener)** – Tightening of the lips horizontally, often associated with anger or disgust.
15. **AU24 (Lip Pressor)** – Pressing the lips together, often associated with discomfort or anger.

Based on the observed visualizations, multiple points are raised about the detection capabilities of the Py-Feat FEX Detector.

3.1 Do you agree with all its predictions?

The Py-Feat detector accurately predicted the facial expressions for most clear and easily identifiable cases. However, in some instances, it misclassified the emotions. For example, in a side-facing image, the system detected what was supposed to be a *'happy'* expression as *'anger'*, as shown in Figure 1.

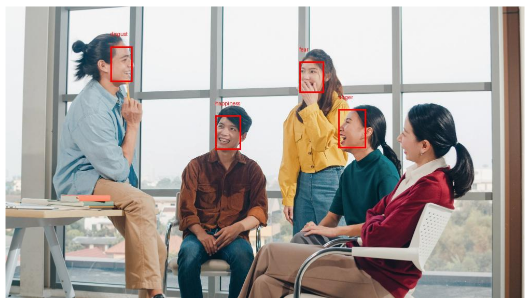

Figure 1: Detected a 'happy' face as 'anger'

Additionally, when the image was upside down, the detector wrongly classified a *'happy'* face as *'sadness'*, as shown in Figure 2. There was also a case where the system incorrectly identified a non-facial object as a face, highlighting a limitation in its detection capabilities.

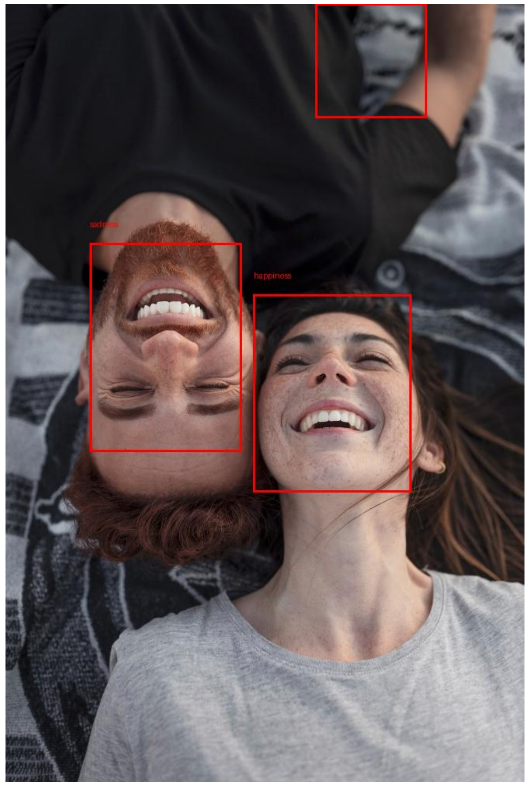

Figure 2: Detected a part of a shirt as a face, and Detected a ‘happy’ face as ‘sadness’

3.2 What seems to confuse the system?

The system seems to fail primarily when faced with unconventional face orientations and partial visibility. For instance, when the face is turned sideways or upside down, the detector sometimes misinterprets emotions, such as classifying a *‘happy’* face as *‘sadness’*, as shown in Figure 2.

Additionally, when parts of the face are obscured, the detector struggles to correctly identify expressions. In one case, the system completely ignored a face because the person was partially covering it with a hand, as shown in Figure 3. This highlights that the model may need improvements in handling non-frontal orientations and occlusions.

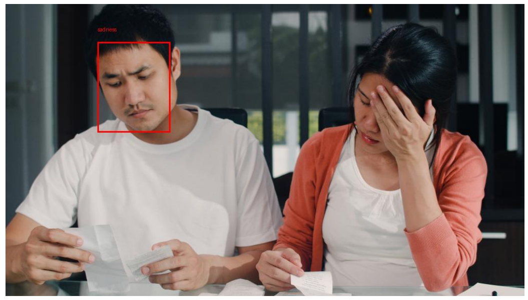

Figure 3: Failed to detect a face

3.3 What are the cases tricky for a human observer?

Cases involving mixed emotions, such as sarcasm or a difficult smile combined with sadness, would be challenging for both the system and a human observer to accurately interpret. Similarly, neutral expressions, where subtle emotional cues are present, can be particularly hard to distinguish.

4 AU Subset Selection

Based on the results of the AU analysis, the next objective is to choose a subset of the AUs as inputs for a predictive algorithm.

4.1 Which AUs would you choose?

Based on the analysis performed, I would choose the top 11 AUs as inputs for the predictive algorithm. These AUs are selected based on the cumulative explained variance (Figure 4) from the PCA analysis, where we observed that the explained variance levels off after the 11th AU. This indicates that adding more AUs beyond this point does not significantly improve the ability to explain the variation in the data. The chosen AUs are: `'AU07', 'AU12', 'AU06', 'AU10', 'AU20', 'AU25', 'AU24', 'AU14', 'AU23', 'AU09', and 'AU02'`. These AUs were identified because they collectively capture the majority of the variance in the dataset, making them the most informative features for the model, while avoiding the inclusion of redundant or less impactful features.

4.2 Problems of using too many features?

Using too many features increases the risk of **overfitting**, where the model captures noise rather than underlying patterns, leading to poor generalization on unseen data. It also raises **model complexity**, making the model harder to train, interpret, and optimize. This increases computational costs and time, reducing efficiency. Reducing features simplifies the model, improving both performance and computational efficiency.

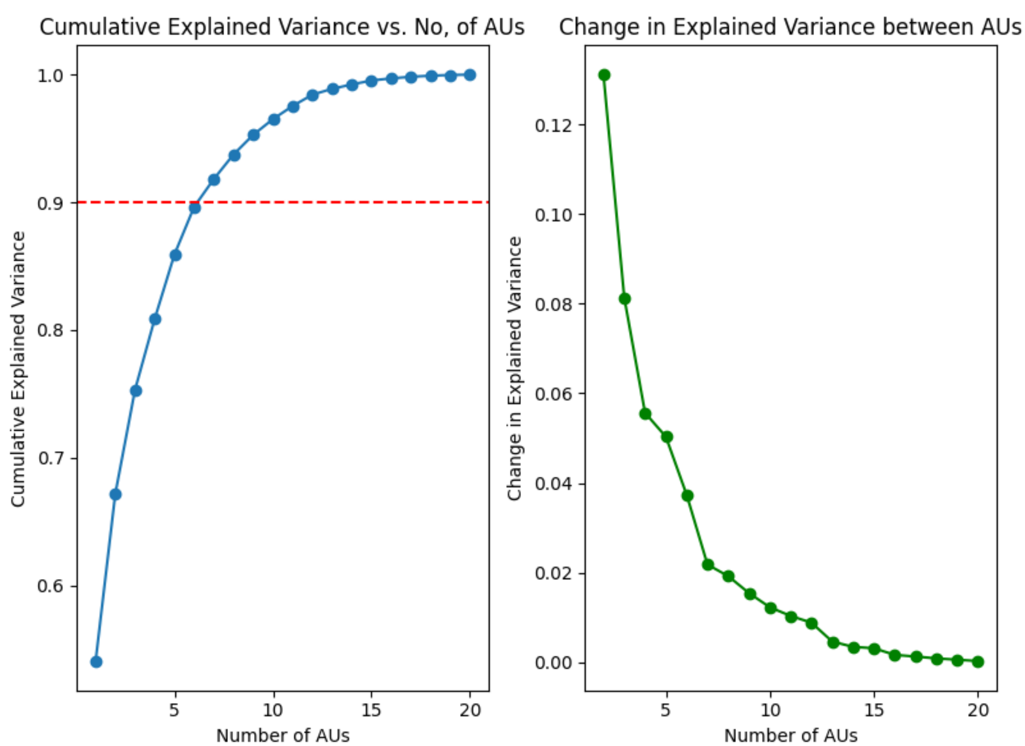

Figure 4: AU Analysis - Cumulative Explained Variance

References

[1] Jolly, E., Cheong, J. H., Xie, T., & Chang, L. J. (2022). Py-Feat: Facial expression analysis toolbox. Retrieved November 14, 2024, from https://py-feat.org/pages/intro.html

[2] Ekman, P., & Friesen, W. V. (1978). Facial action coding system. *Environmental Psychology & Nonverbal Behavior*.