Emotion Recognition

Edoardo De Matteis

February 3, 2021

Contents

1 Task and motivations

The goal of this project is to develop a facial emotion recognition system with some reservations. Facial emotion recognition [2] is the process of analyzing someone's human state from facial expressions, the human brain does it automatically by but software able to recognize human emotions has been and is been studied too. The technology is improving all the time and some are confident it will be able to read emotions as our brains do . Emotion AI has a wide range of uses from scanning for signs of terrorism [12] to emotion judgement for commercial purposes as *Disney* did, or at least wanted to do, for the release of *Toy Story 5* [7].

But all that glitters is not gold and Emotion AI could be a threat to freedom and equality [3], in China *Hanwang Technology* deployed a system that tracks students' behaviour during lessons and analyzes their actions to infer for example if they're keeping attention or not [5]. This software seems to be unefficient and based on pseudoscientific assumptions, these allegations didn't stop *Amazon*, *Microsoft* and *Google* from offering emotion recognition to their customers even though the first two companies claim that their product can't determine a person's internal emotional state from only facial expressions [9].

Other than the ethical dilemma that arises from these technologies is important to note how these are not real emotion recognition systems. In the same way this project will not recognize real emotions but rather expressions, to be more precise it aims to differentiate positive expressions from negative ones. With this abuse of terminology let's proceed describing the strategy adoptet against that task.

2 Strategy

To recognize a facial expression first is needed to find a face in an image, in the online demo it is achieved through Viola Jones while on training and test sets face recognition is automatically done by *Dlib*'s landmark detection. To extract features at first I considered using LBP histograms but since I was interested only in facial expressions seemed reasonable to use landmark points. The key idea is that assuming that similiar expressions have a similiar position on different faces then through an SVM classifier it could be possible to predict someone's "emotion" [1]. In figure 2.1 is displayed how landmark points are disposed in the used landmark pretrained model, for our purposes were not considered all points but only those defining eyes and the mouth (i.e. points from 37 to 68). Still is possible for the user to define an arbitrary list of points. I'm aware that this approach is not the state of the art and to get good results is reccomended to use neural networks [1], anyway I was interested in using SVM classifiers and how they would perform.

The adopted dataset is the *First Affect-in-the-Wild* (affwild) [8], it consists of 298 videos of which 252 for training and 46 for testing. I ignored videos in

[1] As said earlier someone's emotions are more deep than just their facial expression.

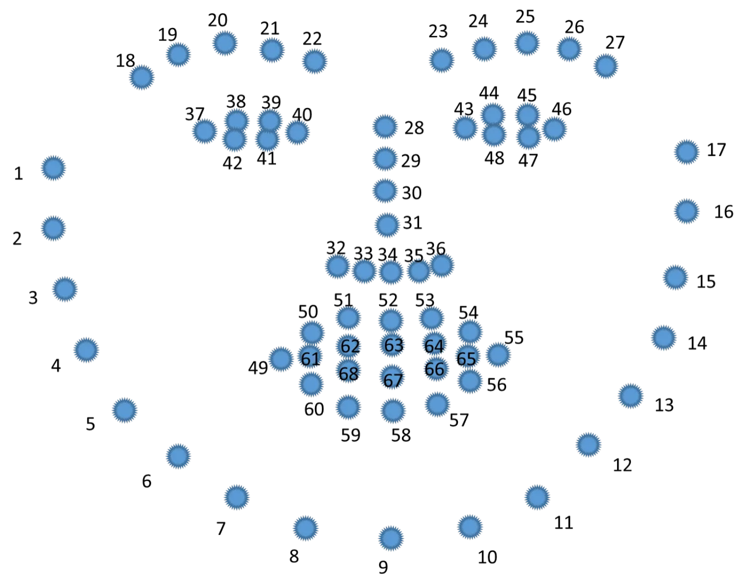

Figure 2.1: Face landmark points from [11].

the train set since there are no responses associated to them, these responses are associated to each frame in which a face was identified. In the dataset are stored relative bounding boxes too but since I was interested in landmark points and the detection is done by *Dlib* I just discarded them.

Emotions are defined as a points on a two dimensional cartesian plane where the abscissa is the *valence*, it expresses if an emotion is positive or negative, and the ordinate is the *arousal*, it discriminates animated emotions from languid ones. An example of this plane can be seen in figure 2.2, where are defined 4 different emotions but is possible to have more. To avoid possible performance reduction I decided to not discriminate between too many classes, and considering also that SVMs in their purest form are binary classifiers it came natural to me to just consider only one response. I chose valence since it may be more interesting to distinguish between positive and negative expressions rather than exciting or not.

3 Design

I decided to use the Python language with the external library *OpenCV* for its simplicity. Since I do not use the bounding boxes in the affwild dataset the first thing I need to get is a correspondence between landmark points and valence for each frame of each video, the execution took almost 21 hours so I made possible to save the results as text files.

Each valence ranges from -1 to 1 and values ranging between -0.5 and 0.5

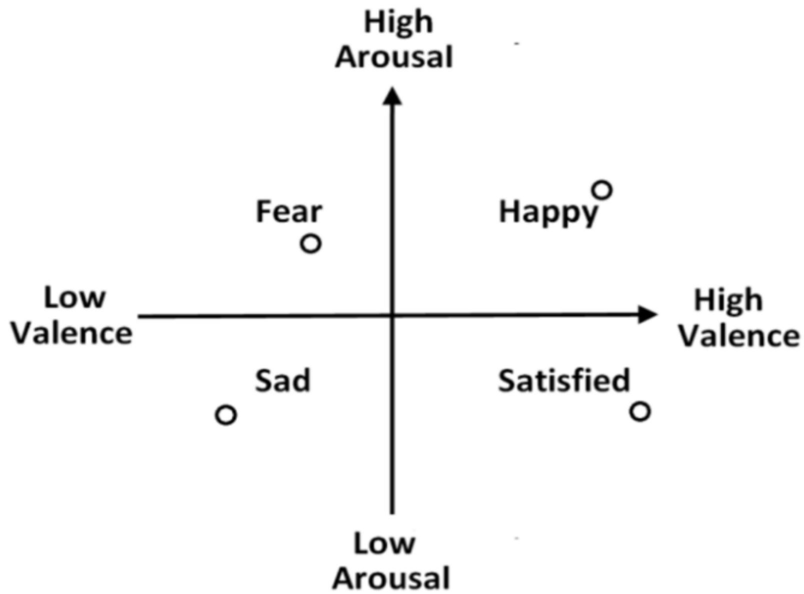

Figure 2.2: Emotion classification model from [13].

were explicitly ignored, it has been done to reduce noise and rule out uncertain valences. In the end were extracted 136909 landmark-valence tuples. Is possible for a single frame to contain more than one face as seen in figure 3.1 [2]. Since the affwild dataset collects videos with foreground faces is reasonable to assume that the face we're interested in is the widest one, hence I pick the landmark feature with the largest euclidean distance between the first and last point (respectively the lateral commissure of the viewer's left eye and the lower lip).

Valences are saved and read as an array of floats while landmarks are a two dimensional array. For each face dlib stores landmark points as an array of $[x, y]$ coordinates, to make things more readable (and possibily to make the code faster in corner cases due to caching) I decided to flatten it as a one dimensional vector. This different encoding should not change SVM's predictions.

Regarding SVM classification I first used both the OpenCV implementation and the *scikit-learn* one, the latter is way better than the first, for this reason the demo uses the scikit-learn SVM. During tests the dataset has been splitted as 80-20 percent for train and test respectively, it comes without saying, demo's SVM classifier is trained over the whole dataset.

As said before in the demo instead of directly detecting landmark points I first detect bboxes. This may be odd but I noticed how given an arbitrary frame there is more probability to detect a face with dlib's landmarks than with OpenCV's ViolaJones, detecting less landmarks makes the demo less precise but

[2] I find bounding boxes less invasive, in this case landmark detects two faces as well.

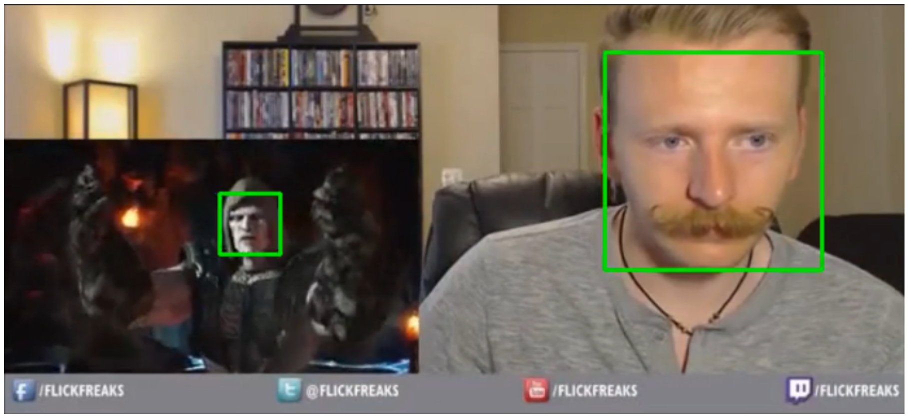

Figure 3.1: Two faces detected in the video 309.

also makes the video smoother for a presentation. Were used pretrained models both for bounding boxes [10] and landmark points [6].

The structure of classes and packages can be seen in the diagram in figure 3.2, it is all original code except the function `src.test.Test.plot_roc` used to plot ROC curves. It has been taken from [4] as commented in the source code.

Other code that is not original are clearly external functions, see table 1

Table 1: Imported libraries

os	To explore the filesystem.
re	For regular expressions.
fire	To implement command line arguments.
numpy	Arrays are essential thanks to the faster C implementation.
cv2	OpenCV, used for detection with Haar features and SVM.
dlib	Since in OpenCV landmark points detection is not implemented.
sklearn	Scikit-learn implements a more efficient SVM classifier.
joblib	Used to save scikit-learn's SVM model.

4 Performance evaluation

The SVM classifier implemented in the scikit-learn library uses a RBF kernel (equation 1), on the other hand in OpenCV is possible to choose which kernel to use and eventually the degree of the model.

$$K(x, x') = \exp\left(\frac{||x - x'||^2}{2\sigma^2}\right) \tag{1}$$

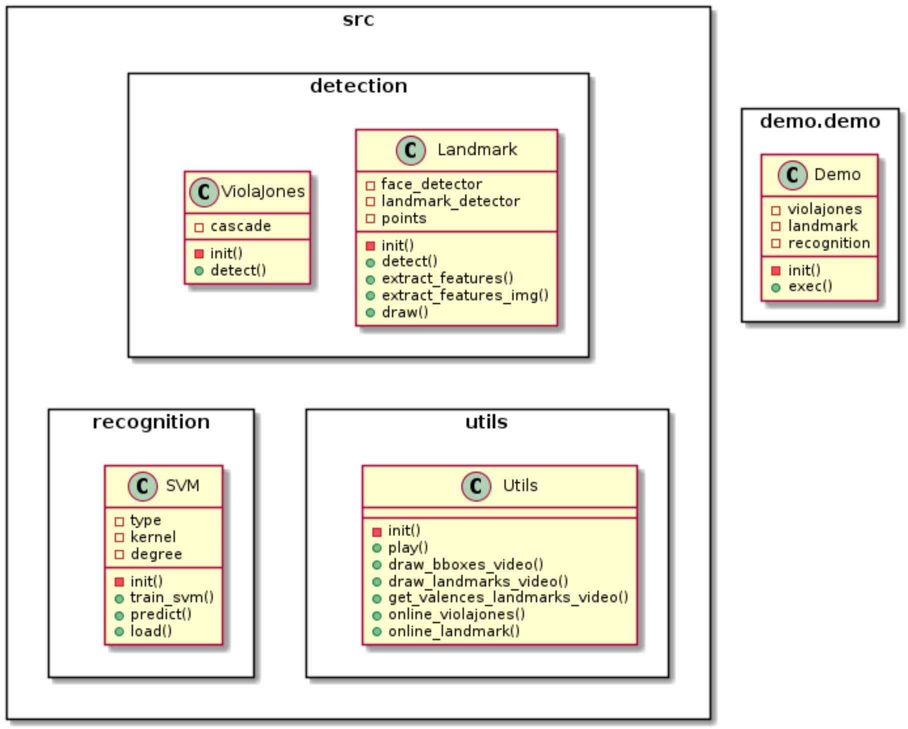

Figure 3.2: Diagram of the packages.

Comparing these two classifiers with the same kernel the scikit-learn's one offers better performances, in figure 4.1 are displayed the confusion matrices for both of them and in table 2 relative scores.

Table 2: Scores for the classifiers.

	scikit-learn	OpenCV
Accuracy	0.805	0.491
Precision	0.814	0.421
Recall	0.639	0.860
F1	0.358	0.283

They're not particularly good, except maybe accuracy and precisione, but is clear how scikit-learn's classifier is more balanced. Even in recall, where OpenCV performs better, the difference is less significant than when scikit-learn is better.

In image 4.2 the ROC curves for the scikit-learn SVM classifier and OpenCV ones are plotted. For OpenCV where considered three different classifiers: a linear one, a polynomial one with degree 3 and for comparision an RBF one.

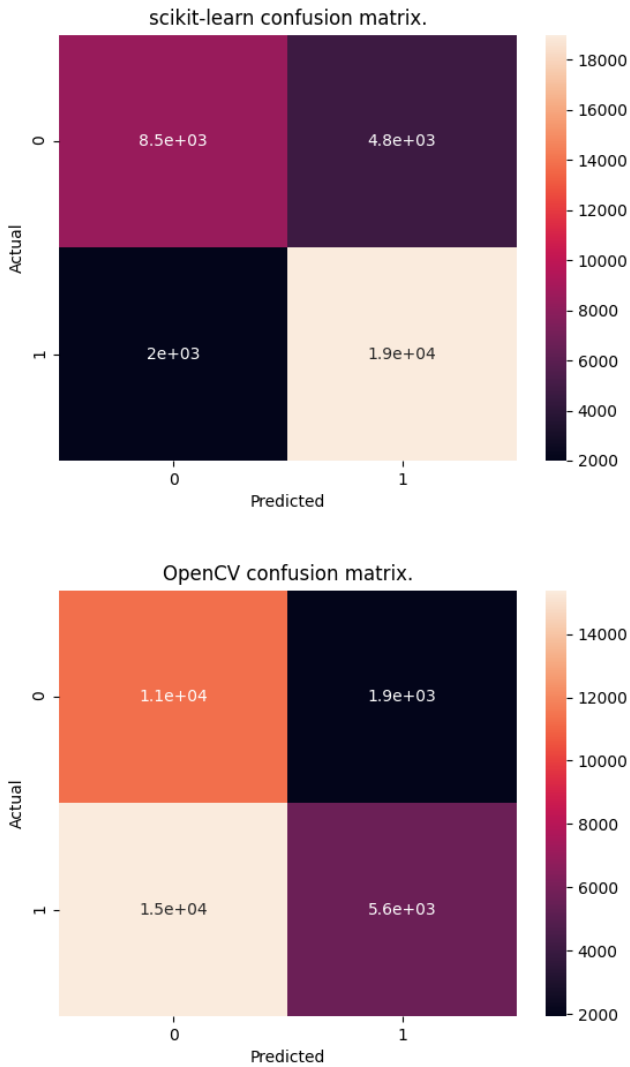

Figure 4.1: Confusion matrices.

The polynomial kernel one has terrible performances, is like throwing chances [3], while the linear and RBF ones overlap each other with quite better performances. Again the best performing classifier is the scikit-learn with a RBF kernel, the AUCs are displayed in table 3.

Also the DET curves for the scikit-learn and OpenCV, both with the RBF kernel, were plot in figure 4.3 and again the scikit-learn model is better than

[3] With an AUC of 0.5

Table 3: Scores for the classifiers.

	scikit	OpenCV
AUC	0.774	0.560

the OpenCV one.

On figure 4.4 are plotted the false positive (i.e. false genuine) error rate and the false negative (i.e. false rejection) one, the intersection point is the equal error rate (EER) and the ideal threshold seems to be 1.25. Keep in mind that these two error rates are plotted against only three thresholds so this value does not necessarly match the best threshold. The EET is identified only for the scikit-learn SVM since it is the best model so far.

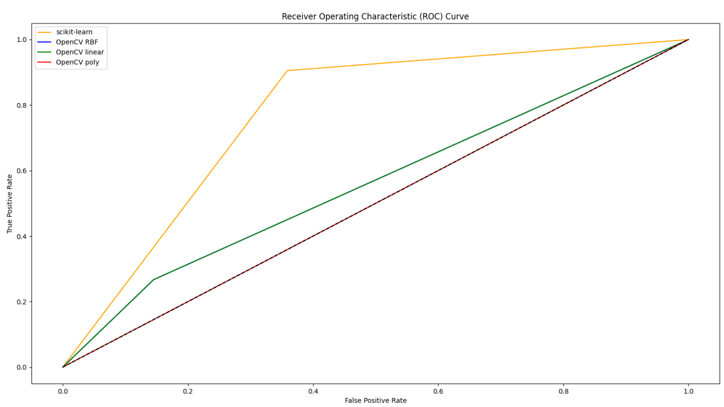

Figure 4.2: ROC.

5 Conclusions

Starting from the fact that I did not expect too high performances a 76% AUC is not too bad, unfortunately I did not find any notebook or any other data that uses the same dataset as mine to compare my results with. I expected the polynomial kernel to be better than the linear one and I was surprised of how a degree of 3 makes the model overfit.

Last the online demo seems to work better under precise conditions: is reccomended to make the face visible for face detection and glasses can compromise the expression identification. Brightness is crucial, not only for the emotion recognition phase but if the background features clutter for the face detection too. After some experiments I noticed how the ideal distance is along 22 centimeters from the webcam, this could be due to two differents reasons:

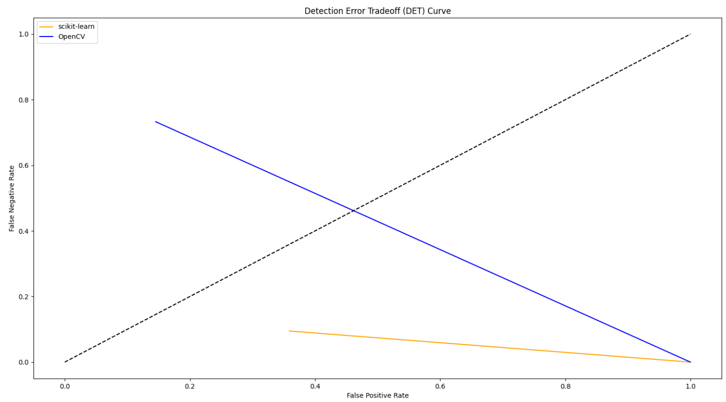

Figure 4.3: DET.

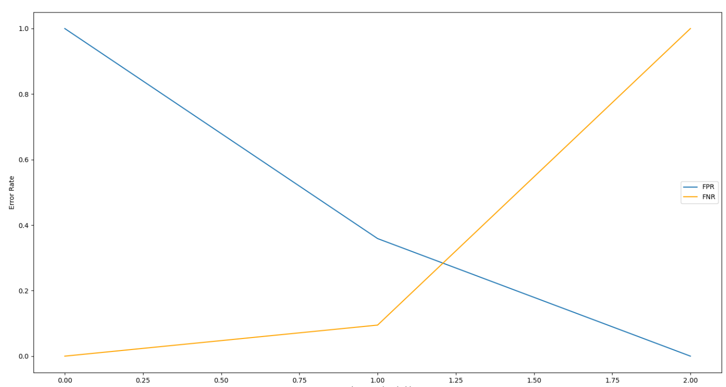

Figure 4.4: EET.

if the distance is shorter there can be false positive face detections, while if it is shorter or higher then the emotion recognition tends to fail and stick to an emotion. Another interesting point is that three-quarter poses tend to be recognized as positive emotions even when they're not, this could be probably due to a selection bias in the dataset.

5.1 Future works

It could be possible to implement a classifier that predicts both valence and arousal, it could be done with a SVM multiclass classifier that uses softmax regression or with two binary SVMs. Another possible improvement could be implementing the emotion detection with a nerual network and see how results compare.

References

[1] Bleed AI. Emotion/facial expression recognition with opencv. https://bleedai.com/facial-expression-recognition-emotion-recognition-with-opencv/, 2020.

[2] Algorithmia. Introduction to facial emotion recognition. https://algorithmia.com/blog/introduction-to-emotion-recognition, February 2018.

[3] Article19. Emotion recognition technology: A threat to free speech, equality and privacy. https://www.article19.org/resources/emotion-recognition-technology/, January 2021.

[4] CodeSpeedy. How to plot roc curve using sklearn library in python. https://www.codespeedy.com/how-to-plot-roc-curve-using-sklearn-library-in-python/.

[5] The Disconnect. Camera above the classroom. https://thedisconnect.co/three/camera-above-the-classroom/, 2019.

[6] dlib. shape_predictor_68_face_landmarks. http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2.

[7] Gizmodo. Disney is building facial recognition to figure out when you'll laugh during *Toy Story 5*. https://gizmodo.com/disney-is-building-facial-recognition-to-figure-out-whe-1797267294, July 2017.

[8] ibug: Intelligent Behaviour Understanding Group. First affect-in-the-wild challenge. https://ibug.doc.ic.ac.uk/resources/first-affect-wild-challenge/, 2017.

[9] Rest of World. China is home to a growing market for dubious "emotion recognition" technology. https://restofworld.org/2021/chinas-emotion-recognition-tech/, January 2021.

[10] Lalitha Rajesh. Haarcascades. https://www.kaggle.com/lalitharajesh/haarcascades.

[11] Sefiks. Facial landmarks for face recognition with dlib. https://sefiks.com/2020/11/20/facial-landmarks-for-face-recognition-with-dlib/, November 2020.

[12] Ars Technica. The premature quest for ai-powered facial recognition to simplify screening. https://arstechnica.com/information-technology/2017/06/security-obsessed-wait-but-can-ai-learn-to-spot-the-face-of-a-liar/, February 2017.

[13] Şule Şahaboğlu. A research on the machine learning algorithms' applications on brain data analysis. 06 2019.