Image&VideoAnalysis Project: Microexpressions detection with HOOF

Marioemanuele Ghianni
E-mail address
marioemanuele.ghianni@stud.unifi.it

Abstract

This project was made for the exam of the course Image and Video Analysis at University of Florence.
The aim of this project is to introduce and analyze a prototype of a method for detecting facial microexpressions (ME) in a high fps video.
The project is developed in Python using the opencv library and LibSVM for the classification part.

Future Distribution Permission

1. Introduction and basic concepts

FACIAL expressions (FE) are one of the major ways that humans convey emotions.
Aside from ordinary FEs that we see every day, under certain circumstances emotions can also manifest themselves in the special form of **microexpressions (ME)**. An ME is a very brief, involuntary FE which shows the emotion that people try to conceal and they are usually very fast and lasting less than half a second.
This makes the recognition task rather difficult for a person without an accurate and meticulous analysis of the video frames.
Detecting these micro-expressions can play a key role in various areas such as the psychological and investigative sector because it can lead to discover indications of non-verbal communication between patient and therapist or insecurities and emotions hidden during an interrogatory.
Although normal facial expression recognition is now considered a well-established and popular research topic with many good algorithms developed with accuracies exceeding 90%, in contrast the automatic recognition of MEs from videos is still a relatively new research field with many challenges and difficulties.
One of the challenges faced by this field is spotting the ME of a person accurately from a video sequence. As a ME is subtle and short, spotting of MEs is not an easy task.
Furthermore, spotting of MEs becomes harder if the video clip consists of spontaneous facial expressions and unrelated facial movements, i.e., eye-blinking, opening and closing of mouth, etc. On the other hand, other challenges of ME recognition include inadequate features for recognizing MEs due to its low change in intensity and lack of complete, spontaneous and dynamic ME databases.

2. State of art : features and classification approaches

The field of research on microexpressions has been experiencing great growth in particular in recent years; [16] the first complete and important studies are from the last decade and initially they were based on 3 different features : **3DHOG**, **HOOF** and **LBP-TOP**.
Polikovsky et al. [11] presented one of the first approach for facial microexpression recognition based on 3D histograms of oriented gradients.
They divided face into 12 regions selected

through manual annotation of points on the face and then a rectangle was centred on these points.

Then, 3DHOG features was used to recognise motion in each region.

Various HOG variants have subsequently been developed but in recent years research has focused on other types of features that guarantee greater accuracy.

Local Binary Pattern-Three Orthogonal Planes (LBP-TOP) it is one of the most used features starting from 2014, the year in which various works were published in this regard.

Guo et al. [4] used LBP-TOP features in their microexpression recognition experiment. To classify these features, they used the nearest neighbour method to compare the distance between unknown samples with entire known samples.

Euclidean distance has been used as distance measurement. This method was evaluated on SMIC database, one of the first spontaneous micro expression dataset which however contains a small number of examples.

The accuracy results were between 53 and 65%.

Currently, various LBP-TOP variants have been developed with progressively better results ; e.g. Wang et al.'s work [13], inspired two feature descriptors for micro-expressions recognition from the concept of LBP-TOP, LBP-Six Intersection Points (SIP) and LBP-Three Mean Orthogonal Planes (MOP).

LBP-SIP is an extension of LBP-TOP and more compact form. This compaction is based on the duplication in computing neighbour points through three planes.

Therefore, they only considered the 6 unique points on intersection lines of three orthogonal planes.

They claimed that these 6 points carry sufficient information to describe dynamic textures.

Vector dimensions in LBP-SIP is 20, in contrast LBP-TOP produce 48 dimensions.

The basic idea of LBP-MOP is to compute features of mean planes rather than all frames in the video.

Histogram of Oriented Optical Flow (HOOF) it is one of the most used features from 2017 onwards in this area with the contribution of Happy and Routray [5] and Liu et al.'s works [9].

In recent years approaches based on convolutional neural networks have been depopulated even though SVMs are still very much in use for classification.

3. Optical flow and HOOF in detail

The concept of optical flow was introduced by the American psychologist James J. Gibson in the 1940s to describe the visual stimulus provided to animals moving through the world.

Gibson stressed the importance of optic flow for affordance perception, the ability to discern possibilities for action within the environment.

Afterwards, the term optical flow is used in the field of robotics and surveillance and is closely related to the concept of motion detection, video compression and video stabilization.

[10]Optical flow is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of object or camera.

It is 2D vector field where each vector is a displacement vector showing the movement of points from first frame to second endowed with a magnitude and an orientation.

The optical flow works well when two fundamental assumptions occur:

- Color/brightness constancy

- Small motion

Consider a pixel $I(x, y, t)$ in first frame; It moves by distance (dx, dy) in next frame taken after dt time. So since those pixels are the same and intensity does not change, we can say,

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

Then take taylor series approximation of right-hand side, remove common terms and divide by dt to get the following equation:

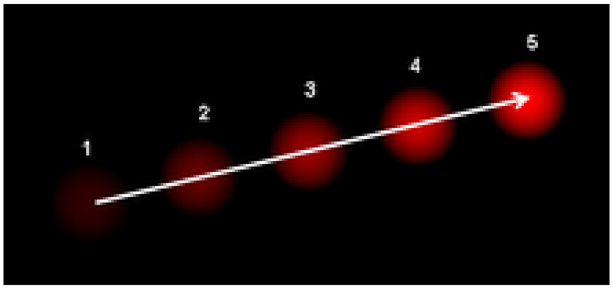

Figure 1: A ball moving in 5 consecutive frames. The arrow shows its displacement vector. Image by wikipedia optical flow's page.

$$f_x u + f_y v + f_t = 0 \; ;$$

where:

$$f_x = \frac{\partial f}{\partial x} \; ; \; f_y = \frac{\partial f}{\partial y}$$
$$u = \frac{dx}{dt} \; ; \; v = \frac{dy}{dt}$$

Above equation is called Optical Flow equation. In it, we can find f_x and f_y, they are image gradients.
Similarly f_t is the gradient along time. But (u, v) is unknown.
We cannot solve this one equation with two unknown variables.
So several methods are provided to solve this problem and one of them is Lucas-Kanade that computes optical flow for a sparse feature set.
Another method is based on Gunner Farneback's algorithm that computes the optical flow for all the points in the frame, the so-called dense optical flow which is the method actually used in this work.
Finally, from the calculation of the optical flow, the histogram of the oriented optical flow (HOOF) is constructed.
The histogram is calculated by binning flow vectors based on the orientation, and weighted by the magnitude of the vector.
So, the standard HOOF feature is a 8-dimension vector with the sums of the magnitudes of vectors in every bin, normalized based on the number of pixels.

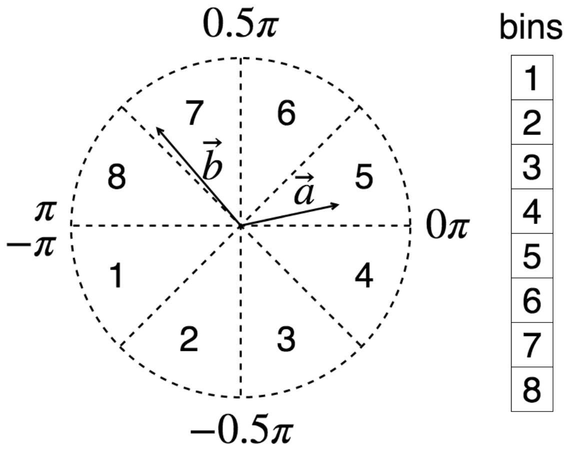

Figure 2: An example of an histogram construction process. flow vectors a and b are binned on 5 and 7 respectively based on their orientation

4. ME database : CASME2

The first studies and researches in this area have been developed on datasets with non spontaneous microexpressions where the participants were asked to perform various basic emotions with low muscle intensity and moving back to neutral as fast as possible.
Famous datasets in this context are **YorkDDT** and **Polikovsky Dataset** but they are not publicly accessible.
Anyway, Posed facial expressions have been found to have significant differences to spontaneous expressions [1], therefore the micro-expressions in these datasets are not representative of natural human behaviour and highlights the requirement for expressions induced naturally.
Developing micro-expression spontaneous datasets is one of the biggest challenges faced in this research area.
It is difficult to elicit micro-expressions because they are difficult to fake, so we need to get the true emotion while the person try to hide it.
Nowadays, there are very few spontaneous micro-expression datasets that are valid, large

and freely accessible and this constitutes a problem in the development of research in this field.

One of the few valid and publicly accessible datasets with various subjects and videos of fully labeled microexpressions is undoubtedly the Chinese Academy of Sciences Micro-expression database : **CASME2**.

Participants of the database were asked maintain a neutral face (neutralization paradigm) in the study while a series of videos are shown to arouse various emotions. Therefore, microexpressions captured in CASME2 are relatively "pure and clear", without noises such as head movements and irrelevant facial movements.

[3] The CASME2 database has the following characteristics:

- The samples are spontaneous and dynamic micro-expressions. Baseline (usually neutral) frames are kept before and after each micro-expression, making it possible to evaluate different detection algorithms.
- The recordings have high temporal resolution (200 fps) and relatively higher face resolution at 280×340 pixels.
- Micro-expression labeling is based on FACS investigator's guide and Yan et al.'s findings (Yan et al., 2013) that is different from the traditional 6 categories on ordinary facial expression.
- The recordings have proper illumination without lighting flickers and with reduced highlight regions of the face.
- Some types of facial expressions are difficult to elicit in laboratory situations, thus the samples in different categories distributed unequally. In CASME2, the examples are more balanced.

5. Preprocessing phase

A fundamental point of the preprocessing phase is the highlighting and extraction of regions of interest (**ROIs**) from the various video frames.

The motivation for extracting regions is to eliminate data that does not add enough value to the task of spotting micro-expressions and thus reducing the complexity of the problem.

In our case we will select the regions corresponding to the mouth and the right and left eyebrows because they are the regions where most of the facial microexpressions are concentrated.

This selection also allows you to not consider parts of the face that are sources of false positives, in fact, as Liong et al.'s work [14] suggests it is better to exclude the eye region because eye blinks cause a lot of false positives.

Also, these ROIs allow a direct connection with the AUs noted in the casme2 dataset thus allowing to isolate specific micro-expressions of a region (for example: left eyebrow) and therefore to be able to consider only that region.

5.1. Face alignment

The first step is the face alignment; to achieve this we start first with the identification of landmarks via the imutils package [12] which localizes the facial landmarks on the frame using a pre-trained facial landmark detector within the Dlib library.

A convenience function then calculates the center points of both eyes using the detected landmarks. Subsequently, the face is aligned such that the eyes are on a horizontal line, as shown in the bottom left of Figure 3.

5.2. Face detection and cropping

The next step is face detection from the video frame.

For this purpose it has been used the default Dlib face detector [6] that works almost real-time on CPU.

Once the face region has been identified, cropping is performed and a reduced-size frame image is obtained which does not contain irrelevant elements for analysis such as the background, hair and so on.

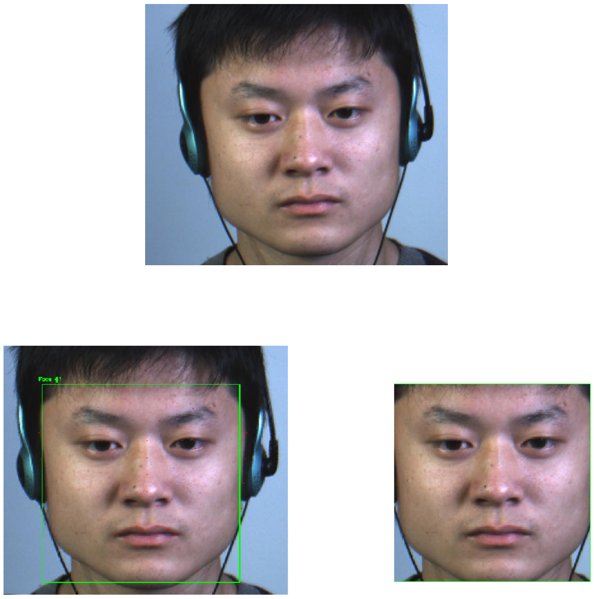

Figure 3: An example of a face alignment, detection and cropping. On top the starting image frame, bottom left the aligned one and right the final cropped frame

5.3. ROIs extraction

From the last frame obtained from the previous phases, the ROIs are then extracted which will follow 3 distinct and parallel processes of processing and features extraction.

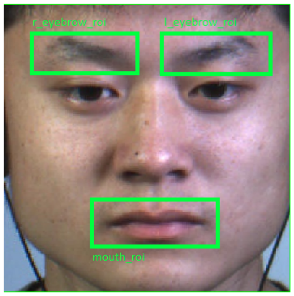

Figure 4: An example of ROIs extraction.

6. Sliding window

Since we are dealing with long and high fps videos, to analyze the video over time a sliding window has been set with a certain overlap.
The upper bound established for the microexpressions corresponds to about 0.5 seconds which in a 200 fps video correspond to 100 consecutive frames.
The sliding window it is therefore set to contain 100 frames with a overlap set at 40 frames (then the window proceeds from 0-100 and then from 60-160 frames and so on).
We also need to somehow ensure that the face maintains the same orientation throughout the video window so the first frame of the window is a sort of **reference frame** and determines the alignment of all subsequent frames of the window.
In fact, the first frame of the sliding window is used to localize the landmarks and to align the face, and consequently, the same transformation is applied to the remaining frames in the sliding window.
Additionally, the regions of interest are extracted based on the first frame of the window.

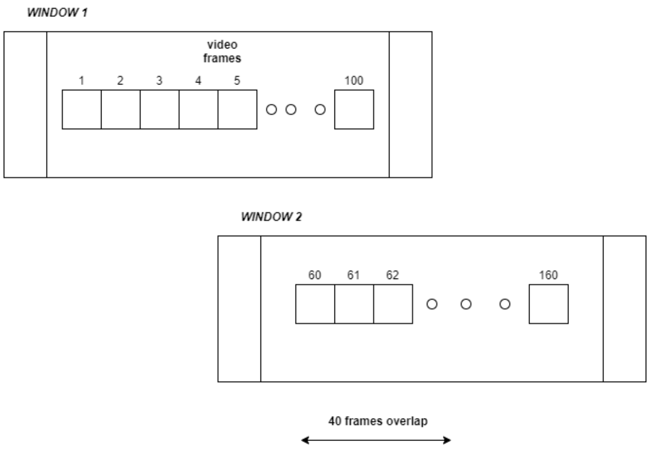

Figure 5: Sliding window scheme.

7. True positive definition

Of course, given the nature of the sliding window, the microexpression is not always completely contained within the window so we need a clear definition of true positive for both training and recognition.
Therefore a threshold is set according to which above is to be considered true microexpression contained in the window and was initially set at 0.8.
This means that if at least 80% of the frames indicated as microexpression are present in the window, we have a true positive and our program must recognize the microexpression.

8. Feature extraction

We have already seen that the ultimate goal is to extract a vector of features that correspond to HOOFs.
This vector is the combination of the optical flow histograms obtained by comparing the various current frames with the reference frame of the window so, the reference frame is stored and used also for the optical flow.
Once the histograms have been extracted from all the frames of the window, we move on to the next with a new reference frame and a new vector of features.

9. SVM classification and training

Each feature vector therefore represents an example to be classified.
For this purpose, a support vector machine (SVM) is used; in particular the LibSVM library by Chih-Chung Chang and Chih-Jen Lin [2] was used for the implementation.
LIBSVM is an integrated software for support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM). It supports also multi-class classification.
The SVM model is being trained with a dataset consisting of a series of examples with label +1 in the case of microexpressions and with label +2 otherwise.
Having divided the analysis into 3 ROIs we will therefore have that, for example, an example relating to the region of the mouth that contains a microexpression relative to the mouth will have label +1 while it will have label +2 if the microexpression is relative to the eyebrows (it can be deduced from the AUs associated with the microexpression and reported by the dataset).
This division is also reflected in having 3 distinct specialized datasets and 3 distinct models for prediction.
The dataset is constructed by balancing the positive and negative examples extracted from various subjects (19 of the 24 subjects are used) while the test set consists of the remaining subjects.
As regards the training phase, executing the frame by frame program generates 3 ”datax” files relating to the 3 ROIs (datax0 -⟶ lips/mouth datax1 - → left eyebrow datax2 - → right eyebrow).
These ”raw” files represent, in a format very close to the libsvm format, the descriptors obtained for each window; each descriptor is divided by an ”AAA” placeholder character and each subfolder by an ”END” placeholder.
These placeholders must be eliminated during the labeling phase which must be carried out manually given the difficulty of the task.
Manually it is necessary to consult the excel file of the dataset, the table of online AUs and, based on the threshold set, determine if the window relating to a ROI really corresponds to a microexpression; if yes, the label must be changed from +2 to +1.
Once the 3 files relating to a subject are obtained and labeled appropriately, data is collected for various subjects and finally 3 datasets corresponding to the various ROIs will be created from the combination of the various files of the various subjects.
Datasets must be constructed in such a way as to be balanced between positive and negative examples avoiding a degenerate model that predicts only one class.
Finally, the SVM_model() function of svm.py

allows, given the input dataset, to obtain the relevant model to be used for subsequent predictions.

This function, basically, through LibSVM, scales the input dataset generating a file range and performs a cross validation through gridsearch for model tuning obtaining the best parameters and the related model file.

To use this model it is necessary to enter the name or possibly the path, if located outside the source folder, of the previously generated range file and the model file in the SVM_predict function in svm.py.

As for predictions, if at least one of the three models relating to the 3 ROIs reports a microexpression, then it is generally considered a positive prediction even if the microexpression is not related to the ROI in question so we need a sort of ”merge” of predictions executed by the ”merge_predict” function in svm.py

This is because the goal of the project is to make a detection and it also helps us to be able to compare this method with others.

10. Tests results

The tests were conducted, as previously specified, on the CASME2 data set dividing it into training subjects and test subjects trying to balancing.

In fact, given the limited number of data, some particular microexpressions are only performed by one or two subjects which, if inserted in the test set, certainly lead to a non-recognition of these.

For this reason, the training and test subjects have been carefully selected but some variations have also been made to check the progress of the predictions as the training and test set vary.

With the variations, the average recall values are around 0.40 with an average F1-score of 0.45 and an average precision of 0.56.

The best results on the dataset are listed below in a table and a confusion matrix.

Where :

$$Recall = TP/TP + FN$$

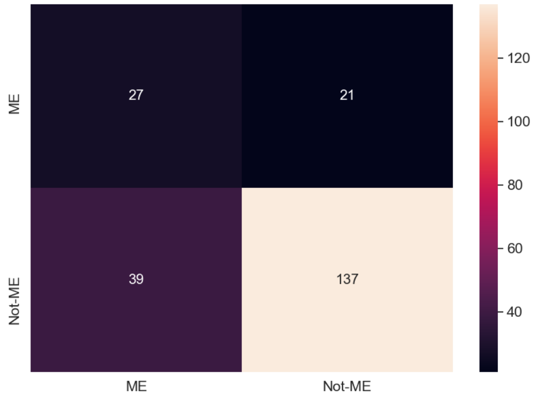

Figure 6: Confusion matrix of test results.

Stats on CASMEII dataset	
$Recall$	0.41
$F1 - score$	0.47
$Precision$	0.56
TP	27
FP	21
FN	39
TN	137
TOT	224

$$F1Score = 2 * (Recall * Precision)/(Recall + Precision)$$

$$Precision = TP/TP + FP$$

Having a direct comparison with other state-of-the-art works is not easy as both as a choice of databases and of metrics and definitions, it is difficult to find similar works.

In any case, even if we adopt a similar definition of true positive,this method outperforms heavily the baseline method via LBP and LTP presented by Li et al.[7].

The algorithm procedure is also very similar to

Michiel Verburg and Vlado Menkovski's work albeit with a different calculation of the features and a different definition of the reference frame. However, using another reference dataset, the results are not fully comparable even if there is a substantial difference in terms of results in favor of this project.

Tests were also carried out by varying the definition of true positive and the threshold; reducing the threshold from 90% to 20%, true positive rates and false positive rates were calculated.

Finally, the ROC curve was built from these.

The estimated AUC stands at around **63.8%** a value very similar to the result obtained by Xiaobai Li et al's work [8] which still uses HOOF as features but uses different preprocessing techniques.

As expected, we can see that by lowering the threshold we can identify more microexpressions but there are also always more false positives so the threshold value depends a lot on the type of application.

If you need to have the lowest number of false positives you will opt for a particularly high threshold, vice versa if instead you want to identify more microexpressions.

A threshold set at 80% is however a good compromise.

11. Discussion of failure cases

As previously mentioned, many classification errors, in particular false negatives, are largely due to the small dataset which does not allow a great generalization starting from the examples.

A solution could be to merge two datasets but the latter must still be similar; different approaches to capturing microexpressions, different ethnicities of subjects, different frame rates and substantially different illuminations can generate problems.

As for false positives, there have been some cases (6 out of 21) in which maybe a microexpression was detected in the window before or after the actual one; these can be called ”less serious” errors and may perhaps be reduced by adopting a more specific definition of true positives that perhaps takes more into account the peak of the microexpression.

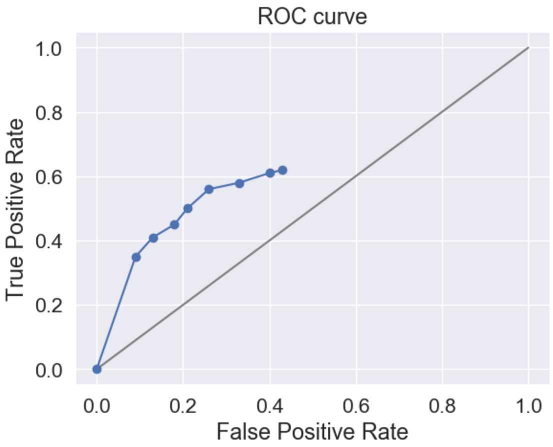

Figure 7: ROC curve as the definition threshold of true positive changes.

The other errors are mainly due to the regions relating to the eyebrows; by using only the mouth region to classify mouth-related microexpressions, higher recall and precision values are obtained.

This is probably due to the complexity and variety of the microexpressions involving the eyebrows and perhaps also the eye-blink partially influences a movement of the eyebrows generating noise.

12. Conclusions and future developments

The detection/recognition of microexpressions is a very complex task and a research field still open; in this project we tried to show a basic approach for detection.

The results are quite good but there is great room for improvement given the complexity of the problem.

All stages can potentially be improved; there are various papers showing advanced preprocessing

techniques that can benefit this task such as motion magnification [15] to enhance the differences of MEs and TIM (Temporal Interpolation Model) [17] for low-fps databases. Another substantial improvement can be obtained by switching to a fuzzy histogram configuration as proposed by Happy et al.[5] which allows a greater tolerance to small variations that can however distort the histogram. Also as regards the features, variations can be made; LBP-TOP it is a very valid alternative and in Li et al's work [8] it has been shown how it is much more effective than HOOF on CASME2 in fact from the ROC curve image we can see the difference reflected also in a much higher AUC value.
Another approach widely used lately concerns the extraction of features via CNN and the use of long short-term memory (LSTM) recurrent neural network, where the temporal characteristics of the data are analysed.
The algorithm is also not suitable for a real time execution given the processing times.
A further step forward, once excellent results are obtained, can be to streamline the procedure and make it suitable for a real time application.

References

[1] Shazia Afzal and Peter Robinson. "Natural Affect Data- Collection Annotation in a Learning Context". In: Oct. 2009, pp. 1–7. DOI: 10.1109/ACII.2009.5349537.

[2] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM – A Library for Support Vector Machines". In: (2019). URL: https://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[3] Prof. Xiaolan Fu's. *CASME II Database*. 2006. URL: http://fu.psych.ac.cn/CASME/casme2-en.php.

[4] Y. Guo et al. "Micro-expression recognition based on local binary patterns from three orthogonal planes and nearest neighbor method". In: *2014 International Joint Conference on Neural Networks (IJCNN)*. 2014, pp. 3473–3479.

[5] S. L. Happy and A. Routray. "Fuzzy Histogram of Optical Flow Orientations for Micro-Expression Recognition". In: *IEEE Transactions on Affective Computing* 10.3 (2019), pp. 394–406.

[6] Davis E. King. "Dlib-Ml: A Machine Learning Toolkit". In: *J. Mach. Learn. Res.* 10 (Dec. 2009), pp. 1755–1758. ISSN: 1532-4435.

[7] J. Li et al. "Spotting Micro-Expressions on Long Videos Sequences". In: *2019 14th IEEE International Conference on Automatic Face Gesture Recognition (FG 2019)*. 2019, pp. 1–5.

[8] Xiaobai Li et al. "Reading Hidden Emotions: Spontaneous Micro-expression Spotting and Recognition". In: *Arxiv* (Nov. 2015).

[9] Yong-Jin Liu et al. "A Main Directional Mean Optical Flow Feature for Spontaneous Micro-Expression Recognition". In: *IEEE Transactions on Affective Computing* 7 (Jan. 2015), pp. 1–1. DOI: 10.1109/TAFFC.2015.2485205.

[10] "Opencv : optical flow tutorial". In: (2019). URL: https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html.

[11] Senya Polikovsky, Yoshinari Kameda, and Yuichi Ohta. "Facial micro-expressions recognition using high speed camera and 3D-gradient descriptor". In: Jan. 2010, pp. 1–6. DOI: 10.1049/ic.2009.0244.

[12] A. Rosebrock. "A series of convenience functions for image processing". In: (2019). URL: https://github.com/jrosebr1/imutils.

[13] Yandan Wang et al. "Efficient Spatio-Temporal Local Binary Patterns for Spontaneous Facial Micro-Expression Recognition". In: *PloS one* 10 (May 2015), e0124674. DOI: 10.1371/journal.pone.0124674.

[14] Liong J. See K. Wong and R. C.W. Phan. "Automatic microexpression recognition from long video using a single spotted apex". In: *Springer International Publishing* (2017), pp. 345–360.

[15] Hao-Yu Wu et al. "Eulerian Video Magnification for Revealing Subtle Changes in the World". In: *ACM Transactions on Graphics - TOG* 31 (July 2012). DOI: 10.1145/2185520.2185561.

[16] Walied Merghani Adrian K. Davison IEEE Moi Hoon Yap. "A Review on Facial Micro-Expressions Analysis: Datasets, Features and Metrics". In: (2018).

[17] Ziheng Zhou, Guoying Zhao, and Matti Pietikäinen. "Towards a practical lipreading system". In: June 2011, pp. 137–144. DOI: 10.1109/CVPR.2011.5995345.