Facial Emotion Recognition

Anonymous CVPR submission

Paper ID

Abstract

Facial Emotion Recognition (FER) represents a pivotal aspect of computational vision, intertwining psychological understanding with technological advancement. The significance of FER spans a wide array of applications, from enhancing user experience design to bolstering security systems. This project is dedicated to delving into traditional computer vision techniques for effective FER, utilizing several datasets. The accent will be set on the feature extraction problem with the aim of defining which feature algorithm to use in priority when practising FER. A critical component of the project will therefore be the comparative analysis, where traditional computer vision methods are juxtaposed. This benchmarking endeavor seeks to illuminate the strengths and limitations of various approaches within the domain of facial emotion recognition.

1. Introduction

This project is about making a comparative review of traditional Facial Expression Recognition algorithm. Doing so we first want to explore the different algorithms that fall under the scope of ”traditional methods”, that is algorithms bulding hand-crafted features, that can be combined with machine learning techniques. The aim of this review is to have a better understanding of how those methods work and how they compare to each other. This is important because there is a multiplicity of ways a face can be presented in an image: it can be among other elements in the image, or a face alignment can differ from an image to another. The application of such algorithms are various in health for example when trying to assess patients well-being and providing personalized care, or treating mental disorders.

The project main focus will be on the feature extraction techniques of the traditional FER algorithms.

2. Problem

In traditional Face Expression Recognition, the results depend on manual feature engineering, relevant features need to be extracted from the images and a good classification model needs to be selected. In this project we focus on benchmarking some feature extraction strategies. The classification model that will be used for prediction is the same for every feature extraction strategy and the full focus of this review will be on the extraction strategies.

Figure 1. CK+ dataset emotion samples.

3. Methodology

3.1. Dataset

Two datasets used for this work are: the Extended Cohn-Kanade (CK+) dataset and the Japanese Female Facial Expression (JAFFE) dataset. Both are popular and widely used datasets in the field of facial expression analysis and emotion recognition, being designed to facilitate research on automatic facial expression recognition systems. We can find 6 different emotions represented in the images that are: anger, disgust, happiness, fear, sadness and surprise and a supplementary neutral that can also be referred to as contempt. The image acquisition is similar in both datasets, they have been acquired in controlled environments to maintain lightning and facial poses consistency.

The CK+ dataset is an extension of the CK dataset that contains 327 labeled facial videos, from which have been extracted the last three frames of each sequence. It contains a total of 981 facial expressions. The images have already been preprocessed, and cropped to 48x48 pixels images.

The JAFFE dataset consists of 213 images of different facial expressions from 10 different Japanese female subjects. Each subject was asked to do 7 facial expressions (6 basic facial expressions and neutral) and the images were annotated with average semantic ratings on each facial expression by 60 annotators. For each subject, there is 3 or 4 samples of each expression. The resolution is 256x256

Figure 2. JAFFE dataset emotion samples.

pixels.

3.2. Feature extraction

The feature extraction algorithm we investigated are the Histogram of Gradients (HoG), the Local Binary Pattern (LBP) and the Gabor filters.

3.2.1 Histogram of Gradients

The Histogram of Oriented Gradients (HOG) feature extraction algorithm has been introduced by Navneet Dalal and Bill Triggs in 2005. It is a fundamental technique in computer vision widely applied for object detection and recognition. Its efficacy lies in capturing crucial structural details within images.

HOG operates by dividing an image into small regions named cells. For each cell, gradients representing the magnitude and direction of pixel value changes are calculated. These gradients are then used to construct histograms of gradient orientations within each cell. Essentially, these histograms capture the distribution of gradient orientations, offering insights into the local structure of the image.

To enhance its robustness to lighting variations and contrast, HOG uses block normalization. Neighboring cells are grouped into blocks, and the histogram values within each block are normalized. This normalization process ensures invariance to local intensity changes.

An important characteristic of HOG is its use of overlapping cells, allowing for a more comprehensive representation of object boundaries. This overlapping nature enhances the algorithm's ability to capture intricate details.

The advantages of HOG are notable. Its features are robust to illumination changes, rendering it suitable for real-world scenarios. By analyzing local gradients, HOG captures essential information about the object's structure.

3.2.2 Local Binary Pattern

Local Binary Pattern (LBP) has been introduced in 1994 by Ojala, Pietikäinen, and Mäenpää. It is a texture descriptor widely employed in image analysis and computer vision. LBP provides an efficient means to represent local patterns and textures within an image. It was first dedicated to texture classification and segmentation problems but has then been successfully extended to face recognition. The method involves comparing the intensity of a central pixel with the intensities of its neighboring pixels in a defined circular or rectangular neighborhood, resulting in a binary code that encapsulates the pattern of intensity variations locally.

The computation process begins with the selection of a neighborhood around each pixel in the image. Then the intensity values of the neighbors are compared to that of the central pixel, resulting in a binary pattern through thresholding. The binary codes obtained from these comparisons are concatenated in a specific order to form a unique binary pattern, capturing the local texture around the central pixel. The distribution of these patterns is then represented in a histogram.

One key characteristic of LBP is its rotation invariance. The binary pattern remains unchanged even when the entire pattern is rotated. Additionally, LBP exhibits gray-scale invariance, making it robust to changes in illumination as it only considers relative differences in pixel intensities. However, it has limitations. It primarily focuses on local patterns and may not capture global spatial information.

3.2.3 Gabor filters

Gabor filters are mathematical functions designed for texture analysis and edge detection, by being sensitive to specific frequencies and orientations. Their strength lies in capturing spatial frequency information, making them particularly well-suited for the nuanced analysis of facial features. In the context of FER, Gabor filters are applied to facial images to capture important texture information.

Gabor filters are defined as a combination of a sinusoidal grating modulated by a Gaussian function.

In the context of FER, Gabor filters act as convolutional kernels when applied to facial images. They respond strongly to specific patterns, contributing to the extraction of relevant features associated with facial expressions.

4. Evaluation

In order to evaluate the performance of each feature extraction algorithms, we evaluated the accuracy of support vector classfiers to which were inputted the features derived during extraction. For each couple of feature extractor and classiffier we searched the the best hyperparameters of the extractor.

4.1. Evaluation metric

The evaluation metric plays a vital role during the training process and the selection of which is an important key for discriminating and obtaining the optimal classifier. Facial Emotion Recognition is naturally a multi-class classification problem. the accuracy, that is the proportion of the samples that are correctly classified is a direct performance evaluation metric.

In order to comprehensively take the recognition effect for each category of expression into consideration, the final

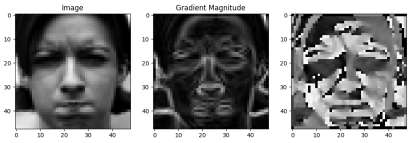

Figure 3. Example of gradients for an image of CK+ dataset.

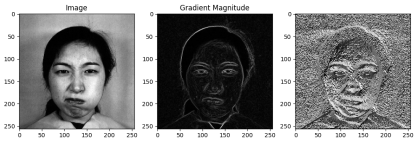

Figure 4. Example of gradients magnitude and direction for a JAFFE dataset sample.

accuracy can also be defined as the average of the recognition accuracy of each category of expression. These two methods of accuracy calculation are called overall accuracy and average accuracy, respectively. In general, higher accuracy stands for better classification performance.

The definition of accuracy is given by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively.

Some metrics for binary classification can be extended for multi-class classification evaluations like precision, recall, and F-measure. Those metrics, despite not being presented, have been subject to systematic checks during evaluation.

4.1.1 HOG

In order to compute the HOG features of an image, we first compute its gradients. As per the original article simple 1-dimensional $[-1, 0, 1]$ mask work best, we get the gradients by filtering the image with these kernels. We can then derive the magnitude and the direction of gradients with the following formulas:

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

After conversion from radian to degrees, the direction values of the gradients lie between 0 and 360 (signed gradients) and we convert them to unsigned gradients that lie between 0 and 180 as the paper mentions that performances decrease with signed gradients even when the number of bins is also doubled to preserve the original orientation resolution.

Figure 3 and Figure 4 show the gradients for samples of each dataset.

After having computed the gradients, we divide the image into overlapping cells and compute the histogram of oriented gradients. Figure 5 shows approximately how the cell

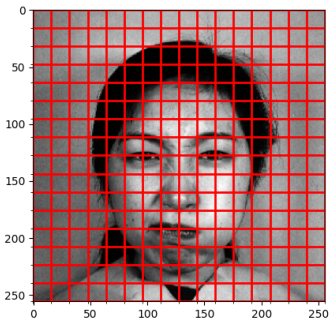

Figure 5. View of image division in cells for a JAFFE dataset sample.

division is applied, however it doesn't explicit the notion of overlapping cells. Cells are more a sliding window (without padding) than a grid applied to the sample and the figure is only for understanding the idea.

In each cell, gradient magnitudes of each pixel are gathered according to the pixel gradient direction to form a histogram. The 0-180 segment is splitted in bins, and pixel contribute to the bins depending on their magnitude. The vote (the value that goes into a bin) is selected based on closeness of a pixel direction to the bin. For example, supposing the 0-180 segment is divided in 9 bins (0, 20, ...) a pixel with magnitude 10 and direction 20 degrees, will add 10 to the second bin. Under same assumptions a pixel with magnitude 20 and direction 10 degrees, since it is halfway between the first and second bins, will contribute half in the first bin and half in the second. After this exercise, we derive Figure 6 and 7.

Finally, we define a block, that is the a window of cells to be considered during a normalization process. The block is to the cell the equivalent of what the cell are to pixels. It is a sliding window on cells in which the histogram of the center cell is normalized with the information of all the cell in the block. This allows HOG to be less sensible to light and contrast variations.

For the CK+ dataset best results were obtained when

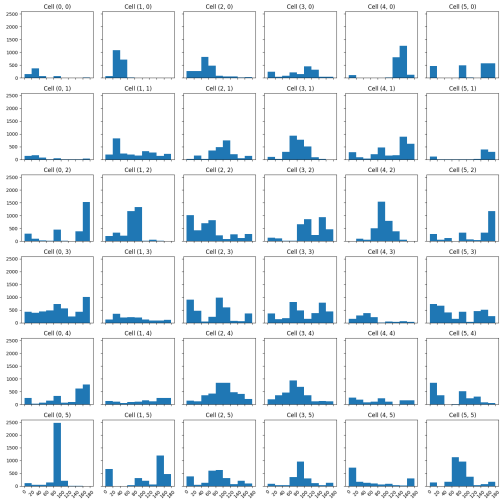

Figure 6. Example of gradients magnitude and direction for an CK+ dataset sample.

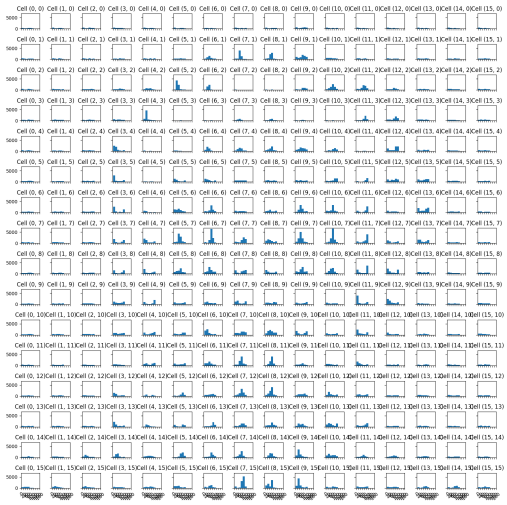

Figure 7. Example of histogram of oriented gradients for an image of JAFFE dataset.

considering 9 bins, 8×8 pixels per cell and 2×2 cells per block. For the JAFFE dataset the best results were obtained with no resize of the images, considering 6 bins, 16×16 pixels per cells and 9×9 cells per block.

4.1.2 Local Binary Pattern

In order to compute features with the Local Binary Pattern extraction method, we first splitted images into non-overlapping blocks. We can refer to Figure 5 to have this procedure in mind. It is interesting to remark that differently to the HOG method, the blocks do not overlap resulting in much more local features. Due to their size, the search for best block size yielded different results on the CK+ dataset

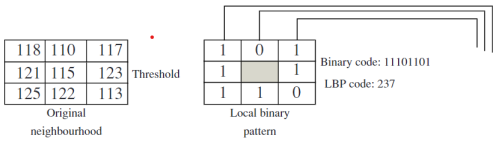

Figure 8. Example of Local Binary Pattern with the 8 pixels of a neighborhood in a radius 1 around the reference pixel.

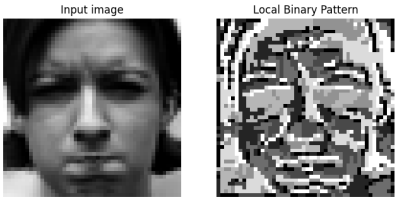

Figure 9. Example of Local Binary Pattern for a CK+ dataset sample.

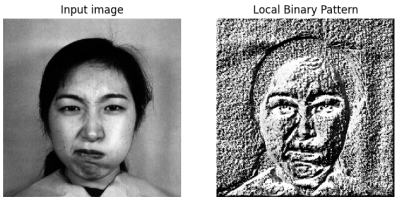

Figure 10. Example of Local Binary Pattern for a JAFFE dataset sample.

and the JAFFE dataset. FOr the CK+ dataset, we chose a block size of 6×6 pixels and for the JAFFE dataset we chose 12×12 pixels.

The second step was to compute the Local Binary Pattern of the regions corresponding to blocks. In order to do for a every pixel in the region, for a given radius (parameter) and a given number of points (parameter) in the neighborhood correspong to this radius, neighobring pixels are assigned a label depending on wether they exceed the threshold defined by the value of the center pixel or no. The label then define a number in binary that is converted to decimal. A visualization for the concept is given by Figure 8. Figure 9 and Figure **??** show local binary patterns for sample from the datasets.

The search for best parameters that CK+ dataset classification was efficient when setting the radius parameter to 3 and the number of point parameter to 10. For the JAFFE dataset the optimal parameters were found to be a radius of 6 and a neighborhood of 12 pixels.

Once LBP of all pixels have been computed for all re-

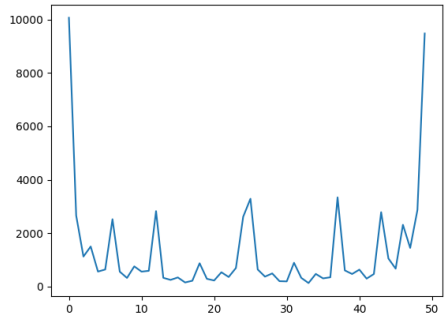

Figure 11. Example of Local Binary Pattern Histogram for a JAFFE dataset sample.

gions, an histogram of the LBP codes is computed for each region. These histograms, once concatenated correspond to the features to be used in the classifying part of the project. The final number of features for a given image corresponds to the number of blocks needed to cover it multiplied by the number of bins of the histogram. Figure 11 shows an example of histogram found in a region of an image from JAFFE dataset.

4.1.3 Gabor filters

The Gabor wavelets kernel can be defined as

$$G(x, y; f, \theta, \lambda, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda} + \theta\right) \quad (1)$$

A 2-d Gabor function is a plane wave with wave-factor k, restricted by a Gaussian envelope function with relative width :

$$\psi(k, x) = \frac{k^2}{\sigma^2}\exp\left(-\frac{k^2x^2}{2\sigma^2}\right)\left[\exp(ik \cdot x) - \exp\left(-\frac{\sigma^2}{2}\right)\right]$$

In order to compute features from Gabor filters, we first created a bank of filters with different orientations and frequency. Figure 12 shows a bank of this kind. In our study we set ψ to 0. And we have looked for interesting parameters. Six distinct orientations from 0 degrees to 180 degrees, differing in 30 degrees steps have been considered. These Gabor filters are then applied to each of the images and concatenate to make a features. Several methods imply to proceed to a PCA afterwards in order to reduce dimensionality. In this work we didn't proceed to a PCA but rather resized our images so that the number of features would not explode.

For the CK+ dataset, the best results were found with a kernel size of 11, a σ of 1.5, γ of 1.2, λ of 3, as mentionned above ψ was considered 0 and the orientations as defined

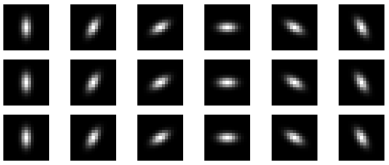

Figure 12. Example of Gabor filters bank with different orientations and frequencies.

Method	Accuracy	Sigma
HOG	99.18%	0.865%
LBP	95.61%	0.608%
Gabor	98.36%	1.290%

Table 1. Extractor review on CK+ dataset

Method	Accuracy	Sigma
HOG	88.25%	1.630%
LBP	78.41%	2.721%
Gabor	82.13%	3.221%

Table 2. Exctractor review on JAFFE dataset

above. For the JAFFE dataset, best results were found after resizing with a window of 11, σ of 3, γ of 3 and λ of 5, the same than above applies for ψ and orientations.

4.1.4 Results

The results are evaluated with a K-fold cross-validation for which the parameter K has been set to 4. The model used was a Support Vector Classifier. The results are presented in Tabe 1 and 2

It is not suprising to see that HOG methods always yield better results that LBP method in our review. The HOG method, because of its overlapping cells and normalization procedure is able to reduce the locality of the features whereas LBP regions do not overlap.

It is also interesing to point that classificaiton with Gabor filters was the heaviest interms of computational resources for our computers.

Finally, we notice that the results in term of variance were the biggest for the GAbor filters.

5. Conclusion

We were able in this project to investigate and discover different feature extraction procedure in the Facial Emotion Recognition field. From the results we could derive the most interesting procedure for our data.

Additionally, the comparison of the two datasets, CK+ and JAFFE, provided insights into the generalizability of

the models across different datasets. The ability of the models to perform well on both datasets signifies their potential applicability to diverse scenarios and datasets.

However, it is essential to note that despite being two separate datasets, CK+ and JAFFE have many similarities that may explain why the standing of the feature extraction methods was the same for both. The choice of the most suitable method depends on the specific requirements of the application, the characteristics of the dataset, and the computational resources available. For images in which the subject has the head turned to the right or the left, results are note ensured by the work in this project.

Future work could involve the exploration of more advanced feature extraction techniques, the integration of deep learning approaches for end-to-end feature learning, and the utilization of larger and more diverse datasets to further enhance the robustness and generalizability of facial expression recognition systems.

In conclusion, this FER project has contributed to a deeper understanding of the strengths and limitations of HOG, LBP, and Gabor filters in conjunction with SVM for facial expression classification.

References