Stanford University

CS229 : Machine Learning techniques

Project report

Emotion Classification on face images

Authors:
Mikael Jorda
Nina Miolane

Instructor
Andrew Ng

December 12, 2015

1 Introduction

Humans can recognize intuitively emotions on people's faces, but computers or robots? While the Microsoft Project Oxford announces on November 11th 2015 the release of its machine learning tools identifying facial expressions, we also tackle the problem of emotion recognition on face images. In our case, emotion recognition is treated as a supervised classification problem. We use a subset of the CK+ (expanded Cohn-Kanade) database [4], with 327 face images of 640x400 pixels across 123 subjects. As a pre-processing, the images are cropped around the faces and intensity-normalized. Each image has a label from a set of 7 emotions: 1=anger, 2=contempt, 3=disgust, 4=fear, 5=happy, 6=sadness and 7=surprise. Our goal is to train a classifier that, given a new image, automatically annotates it with the corresponding emotion.

As we have a small data set, we implement methods traditionally used on such small datasets (see Section 2), as well as a deep learning method but adapted to small datasets (Section 3). We compare and discuss the results obtained in Section 4 and 5.

Figure 1: Examples of face images. From left to right: Disgust (label 3), Happy (label 5) and Sadness (label 6).

2 Featurization of the images with traditional methods

We build an image representation, i.e. an appropriate featurization of the image with respect to the emotion classification problem. In the literature, authors have developed efficient featurizations that perform very well for face recognition or object recognition on small databases [7][8]. We have focused on the Bag-of-Words (BoW) and Fisher Vector (FV) representations, which we have adapted here to the emotion classification problem.

2.1 SIFT and dense SIFT local descriptors of images

Both BoW and FV featurizations rely on a beforehand computation of local descriptors of images. We choose to consider the very popular SIFT (Scale-Invariant Feature Transform) descriptors or dense SIFT descriptors. Their computations rely on the Gaussian scale space of the image, and the corresponding Difference of Gaussians (DoG) that detects image's keypoints like edges. The SIFT is defined as a 128-dimensional vector describing the image intensity by computing local gradient patches at these keypoints. In contrast, dense SIFT compute a 128-dimensional descriptor at each point of a regular grid predefined on the image.

Very interesting properties of these descriptors with respect to our problem are their invariance to affine transformations, and robustness to changes in illumination, noise, and small changes in view point. Thus they may capture facial expression even if the subject's head appears with some angle. We used the functions vl_sift and vl_dsift, of the VLFEAT library [5] to implement SIFT and dense SIFTs. One controls the average number of detected SIFT by varying the Peak and Edge thresholds, which respectively eliminate too small peaks of the DoG scale space, and peaks whose curvature is too small. One controls the number of detected dense SIFT by refining the grid step on the image.

As shown on Fig. 2.1, SIFT descriptors seem relevant to our approach: they detect emotion-related keypoints, like forehead folding for Disgust and Sadness, but not for Happiness.

2.2 Principal Component Analysis on the local descriptors

Studies have shown that gradient patches surrounding SIFT keypoints are highly structured, and therefore easier to represent using Principal Component Analysis (PCA) [2]. The first components of the PCA subspace are claimed to be sufficient to encode the variations in the gradient patch caused by the identity of the keypoint. In contrast, the later components shall represent details in the gradient that are not necessarily useful.

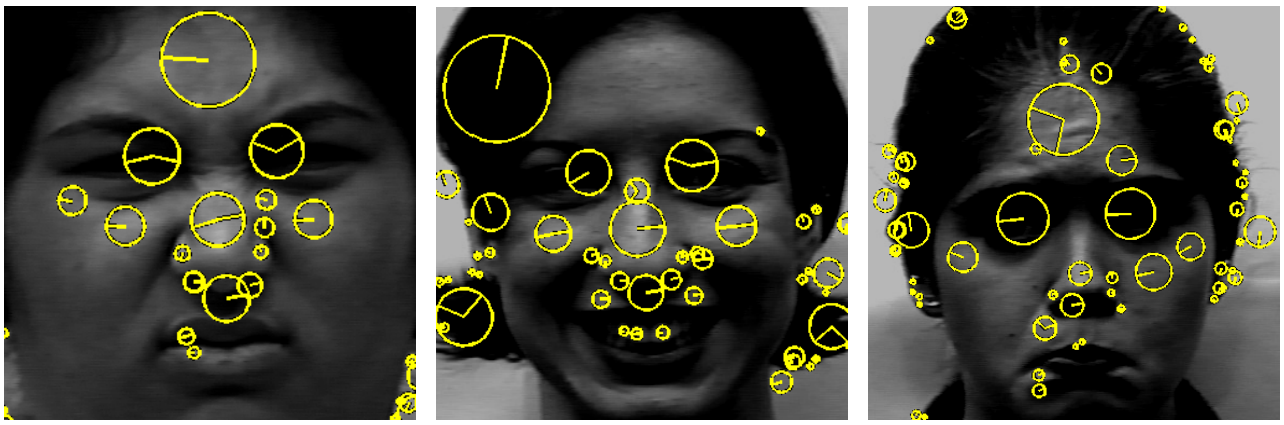

Figure 2: Examples of SIFT descriptors of our images.

Thus we perform a Principal Component Analysis on the SIFT or dense SIFT descriptors with the idea of getting more distinctive and more compact descriptors. Following recommendations in the literature [7], we select the first principal components to project our descriptors.

2.3 Pooling of local descriptors: Bag of Words and Fisher vector

On Figure 2.1, some descriptors are attached to keypoints that are irrelevant for emotions classification, such as those attached to hairs for example. Such keypoints are also often subject-specific and as such, rare among all local descriptors. To remove these unsuitable descriptors, we perform another step in our featurization procedure. In the literature, this step is known as the coding and pooling step. We consider two strategies: Bag-of-Words and Fisher vector.

2.3.1 Bag-of-Words representation

The Bag-of-Words (BoW) featurization works as follows. First, in the coding step, one learns a visual dictionnary of the local descriptors projections. This is done by training a K-means algorithms on the total set of SIFT (or dense SIFT) descriptors projections, whose K centers define the K "visual words".

Second, in the pooling step, each descriptor projection of a given image is hardly assigned to a cluster. The final BoW representation is an histogram counting the occurrences of each visual word in the image. The image is represented by a single vector of length K. We normalize it successively through Power normalization ($\alpha = 1/2$) and L2 normalization, techniques that have been shown to improve the featurization [6]. The size of the vocabulary K is selected by cross validation as we shall see in the results section.

2.3.2 Fisher vector representation

The Fisher vector featurization is very close to Bag-of-Words, being even considered as its generalization [7]. It works as follows. First, in a coding step, one learns a generative model of the descriptors projections. This model is taken to be a Gaussian Mixture Model (GMM), as such models may approximate any continuous distribution with arbitrary precision. It accounts for example for the probability of descriptors around the eyes, the mouth, or the folding of the forehead. The Maximum Likelihood estimates of the GMM parameters are learned on a training set of descriptor projections using the Expectation-Maximization algorithm (EM), implemented in the function vl_gmm of VLFeat.

Second, in a pooling step, the FV representation characterizes the image by the deviation of its SIFT descriptor projections from the SIFT generative model (see [7] for the interpretation of the Fisher Vector in the context of Information Geometry).

In practice, we use the function vl_fisher of the VLFeat library to compute the FV of a new image. The image is represented by a single vector of length 2*64*M, where we recall 64 is the number of principal components selected for the descriptors projections and M is the number of Gaussians. We also normalize this vector successively through Power normalization ($\alpha = 1/2$) and L2 normalization [6]. The number M is selected by cross validation as we shall see in the results section.

3 Featurization of the images through a Deep Learning Method

Lately, deep convolutional neural networks (CNN) have been shown to be extremely powerful for image recognition, in particular for faces recognition [9]. But these networks usually require a huge amount of training data (several Millions) in order to be trained properly and to be efficient. As our database contains only 327 images, we do not train such a network ourselves.

Rather, we use the Deep Face net from [9], which has been pre-trained on 2.6M face images. However, as the goal of [9] was face recognition and not emotion recognition, we choose to cut the last layers of the CNN which are the most specialized. Then we process our images with this cut neural network: the hidden units at the chosen layer form a vector which we normalize and use as our feature vector. The number L of layers that we cut is selected through cross validation.

4 Classification via Support Vector Machine

We use Support Vector Machine with the "one vs the rest" strategy for multi-class classification. More precisely, we train 7 classifiers to separate one emotion from the rest, solving the primal problem for L2-loss with L2-regularization. In practice, we use liblinear library for the implementation [1]. The results below show the 10-fold cross validation accuracy on our whole dataset. The regularization parameter C is automatically determined to give the highest accuracy.

4.1 First results

For each of the featurization methods, we vary a parameter to achieve the highest cross validated accuracy. For Fisher Vectors, we vary the number of Gaussians in the GMM (M); for Bag of Words, the number of clusters (K); and for the CNN method, the number of removed layers (L). The results are summarized in Table 1.

Table 1: 10-fold accuracy percentage on the whole dataset (327 images)

Bag of Words		Fisher Vector		CNN	
K = 256	73.39 %	M = 30	78.59 %	L = 1	61.80 %
K = 400	75.84 %	M = 60	81.35 %	L = 2	58.71 %
		M = 265	78.29 %	L = 3	64.53 %
				L = 4	63.91 %
				L = 5	66.36 %

The Bag of Words featurization performs quite well with a cross-validated average accuracy of 75.84% but the best featurization is the Fisher Vector, with a cross-validated average accuracy of 81.35%. This seems natural as the Fisher vector is a generalization of Bag of Words, containing more information on the images.

The CNN approach performs less well. This means that the results on the hidden units of our pre-trained CNN may not be a good featurization for our emotion classification problem. The CNN pre-trained for face recognition may not be adapted for emotion classification if the goal is to reach an outstanding accuracy. Note that training our own CNN for emotion recognition would require at least 10 or 100 times more images that we have. Thus we discard this option.

4.2 Introducing the Spatial Pyramid Kernel

The Spatial Pyramids approach consists in placing a sequence of increasingly coarser grids (a "pyramid") on the images, and computing the feature vectors on each of the zones [3]. As such, it incorporates spatial information: the location of the descriptor projections in each of the zones, which was lost in the featurization process.

We decide to consider only one grid and split the image in 9 zones, which are defined as shown in Figure 4.2. Taking advantage of the images cropping, these zones are chosen to be pertinent for emotion classification: one zone contains the mouth, another the nose, two others the eyes etc. Thus, the featurization with Spatial Pyramid compares mouth descriptor projections with mouth descriptor projections, nose descriptor projections with noise descriptor projections etc. The new feature vector (FV or BoW) is defined as the concatenation of the 9 feature vectors per zone. This amounts to run SVM with a Mercer Kernel called the Spatial Pyramid Kernel [3].

Adding the Spatial Pyramid Kernel completes our pipeline (Featurization and Classification), which is summarized on Figure 4.

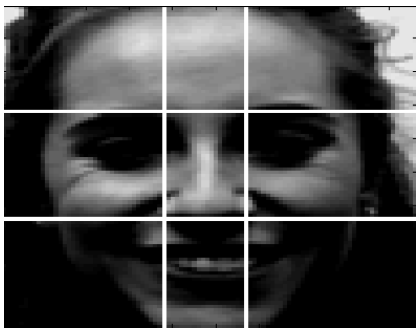

Figure 3: The 9 zones considered for the Spatial Pyramid Kernel.

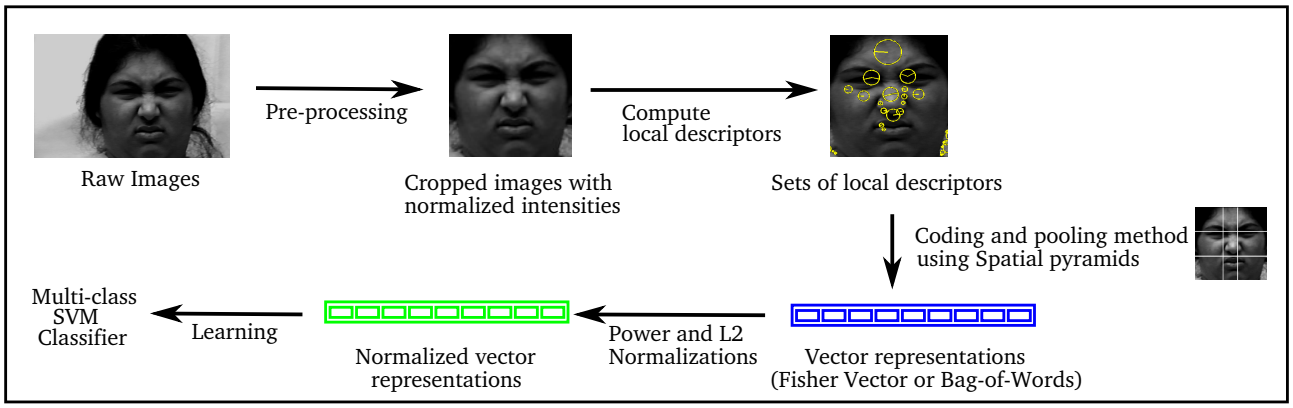

Figure 4: Pipeline of our learning algorithm.

4.3 Results with the Spatial Pyramid Kernel

Using Spatial Pyramid kernel, the average accuracy is significantly increased both for Bag of Words and Fisher Vectors. Figure 4.3 shows the results for these two methods for different parameters. Cross-validating on the hyper-parameters M and K, we now obtain 84.71% for Bag of Words (K = 400) and 85.32% for Fisher vectors (M = 15). We select the corresponding classifiers.

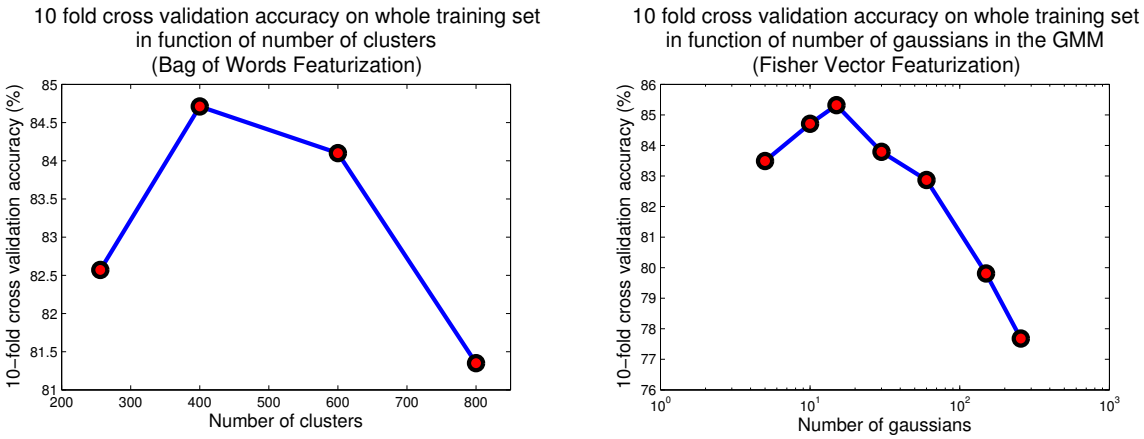

Figure 5: Accuracy of SVM with the two featurization methods.

Ultimately, we compute a better estimate of the generalization error on the selected classifiers. We divide our dataset in two parts: train and test sets. We train the two best classifiers (M = 15 for FV and K = 400 for BoW) and adjust the regularization parameter C with a 5-fold cross validation on the training set (first half). Then, we test the two classifiers on the test set (second half). We get the following results for the estimates of the generalization error (to be exhaustive, we also include the result for CNN):

- 71.34% for BoW,
- 73.78% for FV.

- 59.15% for the CNN (without Spatial Pyramid kernel, L = 5)

Our method manages to classify 7 emotions with a reasonable accuracy for such a small dataset of 327 images.

5 Analysis and conclusions

A main source of errors come from the fact that we may recognize faces instead of emotions. We remark that if we train our classifier on some images and try to predict the emotion of a new face of a subject absent from the training set, the success rate is high. But if we try to predict the emotion of someone who was in the training set, we tend to fail. With more images from more people, we would densify the space of face geometry while keeping the space of emotion at the same size. Therefore the separation of the emotions would be more "visible" than the separation of face geometries and this source of errors would be reduced: with millions of images, we would always have someone resembling our subject in the training database *for each emotion* so this problem would disappear.

More images would also allow us to to train a full neural network and to use deep learning techniques for emotion classification. With the success of these techniques on images, it would certainly be interesting and efficient.

We also notice the great improvement in accuracy due to the introduction of the spatial pyramid kernel. This was expected since this method allows to compare mouths with mouths, noses with noses and so on. Therefore, we expect our algorithm to be able to detect more easily the facial expression (local information) compared to the face geometry (global information)

Then, and in regards of concrete applications, we would like to be able to recognize emotions from images taken from different point of view and in different environments. This would mean, again, to have more training images, but now taken in a lot of different orientations. Of course, our SIFT based representation would be naturally suited for this with its robustness to light conditions and affine transformations of the image.

Ultimately, a substantial adding to our algorithm would be the possibility to rank all 7 emotions, using SVM-ranking methods for example. Indeed, facial expressions are much more complex than the label of a single emotion. Capturing more complex facial behavior, with for example a percentage of each emotion presence, would make our algorithm much finer and closer to human intelligence.

References

[1] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang and Chih-Jen Lin: LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research 9 (2008).

[2] Ke, Y. and Sukthankar, R.: PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. Proceedings of the Conference on Computer Vision and Pattern Recognition (2004).

[3] Lazebnik, S., Schmid C. and Ponce J.: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. Proceedings of the International Conference on Computer Vision and Pattern Recognition (2006).

[4] Lucey P., Cohn J.F., Kanade T., Saragih J., Ambadar Z. and Matthews I.: The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. Proceedings of the Computer Vision and Pattern Recognition Workshops (2010).

[5] Vedaldi A., Fulkerson B.. Vlfeat: An Open and Portable Library of Computer Vision Algorithms. Proceedings of the International Conference on Multimedia (2010).

[6] Perronnin F., Sánchez J. and Mensink T.: Improving the Fisher Kernel for Large-Scale Image Classification. European Conference of Computer Vision. 105 (3) (2010)

[7] Sánchez J., Perronnin F., Mensink T., and Verbeek J.: Image Classification with the Fisher Vector: Theory and Practice. International Journal of Computer Vision. 105 (3) (2013).

[8] Ionescu B., Benois-Pineau J., Piatrik T., Quénot G.: Bag-of-Words Image Representation: Key Ideas and Further Insight. Advances in Computer Vision and Pattern Recognition (2014).

[9] Parkhi O. M., Vedaldi A. and Zisserman A.: Deep face recognition. Proceedings of the British Machine Vision Conference (2015).