Livia Biggi 1793434 Eugenio Bonifazi 1540750 Adriano Fragomeni 1396786

Statistical Learning Project: Emotion Classification

Introduction

This project aimed at creating a programme that would recognise and classify seven different types of emotions (expressions), given as input a dataset of photos of men and women.
The analysis employed different techniques that determined the orientation of the face (i.e. the pose of the subject), and the facial expression.

The dataset used is the Karolinska Directed Emotional Faces (KDEF) (Lundqvist et al., 1998) consisting of 4900 photos, though 78 photos had been manually removed beforehand, as they were ruined and/or included too much noise. Thus, our analysis included only 4822 of the original dataset photos.

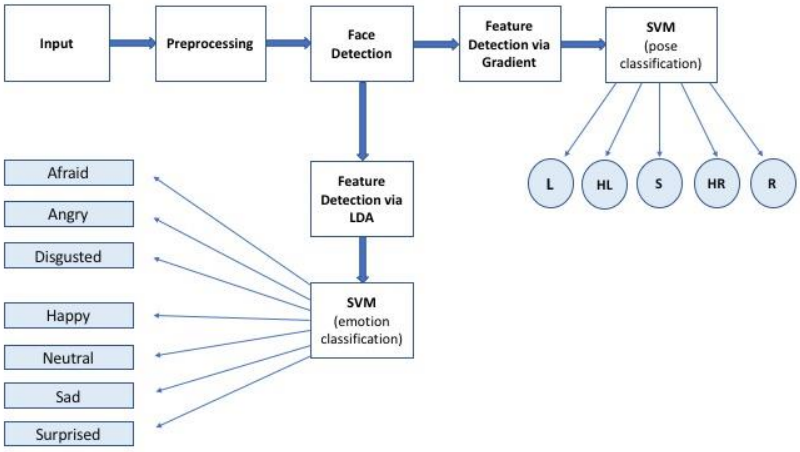
The photos depict 70 individuals (35 females and 35 males) displaying seven different emotional expressions (i.e. afraid, angry, disgusted, happy, neutral, sad, surprised), as shown below.

Each of these expressions is viewed from five different angles (full left profile, full right profile, frontal and three quarters – left and right). The subjects do not present any beards, moustaches, earrings or glasses and no visible make-up.

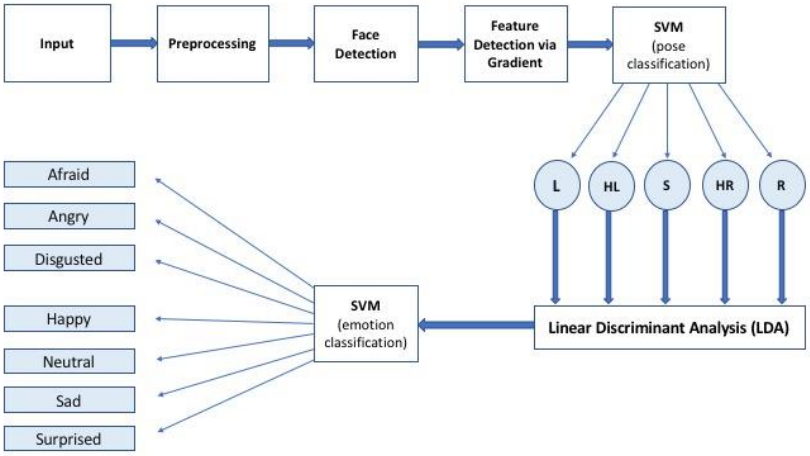
The dataset has been split into a training and test sets, accounting for 70- and 30%, respectively.

The pipeline used presents differences between the training and the test sets, as we have chosen to avoid propagating the classification errors when fitting the data on our model (first on the pose classification, and later, on the emotion classification). In the training set, pose classification and emotion classification are both computed on the detected faces, whereas in the test set the classification of the pose is computed on the detected faces, and its output is then passed as input to the emotion classification. The two pipelines are presented below:

Training Set Pipeline



Test Set Pipeline



The structure of this report follows the structure of the test-set pipeline.

Data Pre-processing

As the first step of both pipelines, the pre-processing proved crucial in altering specific parameters and facilitated the subsequent detection of the face. The Image Pre-Processing method implemented in our analysis has been inspired from a photo-processing algorithm (Cao et al., 2018; Pythontic.com), which is divided into Contrast Stretching and Gamma Enhancement.

Contrast Stretching is a form of image normalisation that is applied directly on the image and modifies the pixel values in such a way that the intensities are transformed into a bigger range. The value of the new pixel is calculated as a linear operation, meaning that no kernel is used, and thus that each pixel value is not determined using the values of the neighbouring pixels. The final pixel values are calculated using the formula:

$$I_o = (I_i - min_i) * \frac{max_o - min_o}{max_i - min_i} + min_o$$

where I_o is the output pixel, I_i is the input pixel (i.e. the original value of the pixel), min_i and max_i are the minimum and maximum values that a pixel can take in the input image, and min_o and max_o are the minimum and maximum values that a pixel can take in the output image (i.e. 0 and 255).

The *Gamma Enhancement* method, on the other hand, is an alteration of the gamma parameter of a photo, and it is implemented through the Adaptive Gamma Correction (AGC) algorithm.
The AGC pre-processing method is based on the formula

$$T(l) = round[l_{max}(\frac{l}{l_{max}})^{\gamma(l)}]$$

Where *T(l)* indicates the magnitude of the individual pixels, $\gamma = 1 - c(l) = 1 - \sum_{x=0}^{l} p(x)$ and *c(l)* is the CDF of the grey levels of the input image and $l = 0,1,\ldots,255$ *p(x)* denotes the normalised grey level histogram, and l_{max} is the maximum pixel intensity, i.e. 255 for 8-bit greyscale images.

Rather than using a simple probability distribution, *p(x)* is considered a weighted probability $p_w(l)$, which is calculated as

$$p_w(l) = p_{max}(\frac{p(l) - p_{min}}{p_{max} - p_{min}})^{\alpha}$$

Where $p_{max} = \max_l(p(l))$, $p_{min} = \min_l(p(l))$ α is the adjusted parameter and takes the values of 0.25 in bright images, and 0.75 in dimmed images.

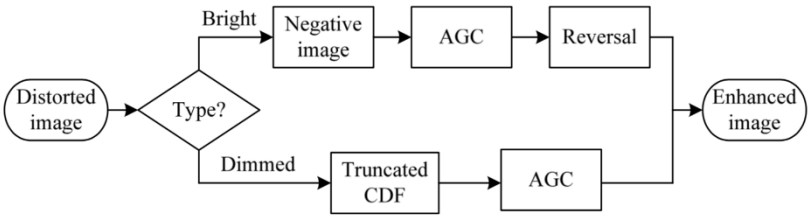
Differently from other implementations, the paper by Cao et al. (2018), applies the AGC algorithm depending on the properties of the individual images. The algorithm distinguishes between *bright* or *dimmed* photos according to the formula

$$t = \frac{m_I - T_t}{T_t}$$

Where $m_I = \sum_x \sum_y \frac{I(x,y)}{MN}$ and $I(x,y)$ is the input image with $x = 1,2,\ldots.M, y = 1,2,\ldots..N$

T_t is defined as the expected global average brightness for normal natural images, which we set as 112, consistently with the paper by Cao et al. (2018).
A photo is defined *dimmed* if $t < -\tau_t$ and *bright* if $t > \tau_t$, with $\tau_t = 0.3$

The rationale behind the distinction between bright and dimmed pictures relies on the fact that the values of the pixels of a photo negative are symmetric to those of the original photo ($I'(x,y) = 255 - I(x,y)$), meaning that the distribution of the values of pixels will also be symmetric to the distribution of the value of the pixels in the original photo. Hence, by working on the photo negatives of bright images (i.e. considering the bright images as *dimmed*), it is possible to apply meaningful transformations to the pictures.



The picture above is taken from the paper by Cao et al. (2018), and shows the different steps taken when pre-processing the images, depending on their properties (i.e. *bright* or *dimmed*). If a photo is considered *bright*, the AGC algorithm is applied to its negative, otherwise, the pre-processing is done on the truncated CDF, i.e. the cumulative distribution function of the pixel grey levels within an image. By working on the truncated CDF, it is possible to set a threshold on the maximum value that γ can take, thus controlling for detail loss in the bright elements in the image.

$$\gamma'_w(l) = max(\tau, 1 - c_w(l)) \qquad \text{where} \quad \tau = 0.5$$

Algorithm-1: Negative-image-based AGC Algorithm

Step-1. Obtain the negative image $\mathbf{I}'$ of the input image according to Eq. (4).

Step-2. Obtain the gray level histogram $p(l)$ of $\mathbf{I}'$, and compute $p_w(l)$ via Eq. (2).

Step-3. Compute $\gamma_w(l)$=1-$c_w(l)$, where $c_w(l)$ is the CDF derived from normalized $p_w(l)$.

Step-4. Apply pixel value transformation to $\mathbf{I}'$ according to Eq. (1) and yield $\mathbf{I}_e'$.

Step-5. Output the enhanced image $\mathbf{I}_e$=*round*$[255-\mathbf{I}_e']$, where *round*$[\cdot]$ is rounding operation.

Algorithm-2: CDF-truncated AGC Algorithm

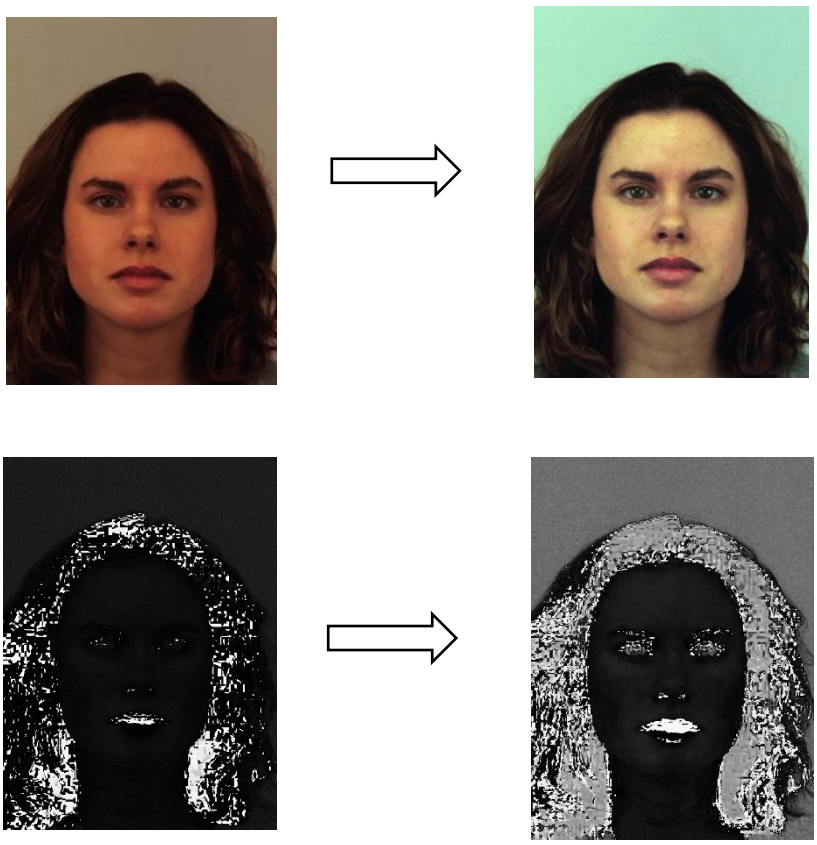
Step-1. Obtain gray level histogram $p(l)$ of the input image $\mathbf{I}$.

Step-2. Compute $p_w(l)$ according to Eq. (2).

Step-3. Compute $\gamma'_w(l)$ according to Eq. (5).

Step-4. Output the enhanced image $\mathbf{I}_e$ by transforming $\mathbf{I}$ according to Eq. (1).

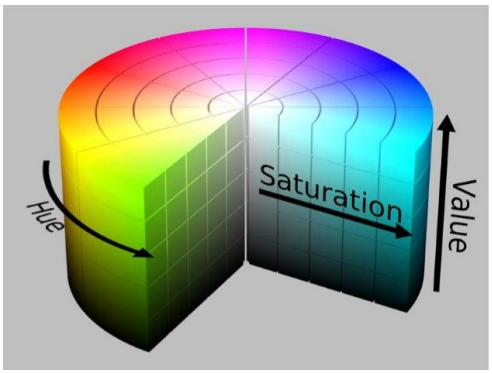
The choice of the algorithm strongly lies on the fact that the images in the KDEF dataset tend to be quite dark, thus interfering with our skin-detection method for face detection. The Adaptive Gamma Correction has a better performance than other algorithms when dealing with darker images (dimmed images) and thus is efficient in adjusting and correcting the photos of the dataset.
This is exemplified by the set of images below, showing the original and pre-processed images, and their respective Hue component (which is later used for face detection).



Face Detection

After pre-processing the images, the goal is to detect the skin region of the pictures (the faces of the subject). The face detection method used in our analysis is based on a skin-colour-based algorithm (Tayal et al., 2012).

The whole dataset is converted from the RGB model, where colours are defined in terms of a combination of primary colours, to the HSV model, where colours are defined in terms of hue, saturation and value.



From the paper by Tayal et al. (2012) the H-component is used to detect skin and non-skin regions, such that all pixels lying in the interval

$$19 < H \leq 240$$

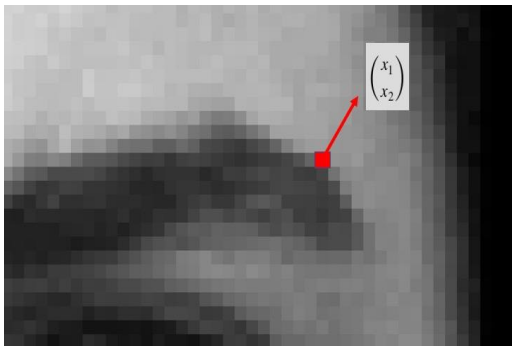
are considered non-skin features, whereas the pixels outside the interval are interpreted as skin.

Given this interval, the V-component is modified by setting all pixels whose H-value falls within the specified range to 0 (i.e. they are blackened) and preserving the original value of all other pixels. This process results in a non-homogeneous blob, to which morphological processes (first dilation and then erosion) are applied to smooth the detected skin.
The blob is then transformed into a binarized image to remove smaller groups of pixels that are inconsistent with the neighbouring pixels in the picture. The resulting image comprises a white portion representing the skin region, and a black portion representing the non-skin region.
The image binarization method used implements a simple binarization (with threshold equal to 128) and the Otsu's Binarization (which is based on the minimization of the weighted within-class variance of the pixels), depending on the picture. The rationale behind this is that the binary threshold is not always efficient in our analysis, and the Otsu's method performs better where the simple binary threshold fails.
Finally, a rectangle is drawn around the detected skin regions.
The final output of our implemented face detection is thus a picture consisting of the region inside the rectangle, followed by the application of a padding, which introduces new (black) pixels around the edges of an image. The padding allows us to use the images of the detected faces in the subsequent steps of the analysis, as they are cropped in different ways (depending on the skin regions detected) and hence present different dimensions.

Pose Classification

In order to classify the different poses we proceeded using the Histogram of Oriented Gradient (HOG) analysis that removes noisy information such that the images are more easily classifiable. The analysis has been taken from the paper by Murphy-Chutorian et al. (2007), and from the website Mccormickml.com (2018).

This reduction of useless information is achieved through the use of histograms computed on the frequencies of bi-dimensional oriented vectors. For each pixel these vectors are obtained by computing the two differences with respect to the pixels on both sides and the ones above and below.

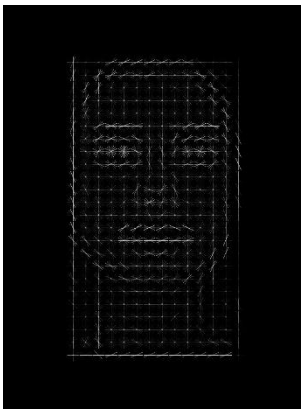


Each vector has the following features:

- $Angle = arctan(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix})$
- $Magnitude \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \sqrt{x_1^2 + x_2^2}$

The Histogram of Oriented Gradient operates through groups of pixels (i.e. cells) which are composed of an arbitrary number of normalised gradient vectors; a histogram of the degree of the vectors' angles is computed for each cell. The contribution of each vector to this histogram is given by the vector's magnitude, and distributed among the appropriate bins.

The algorithm takes into account blocks of cells taken in a discretionary amount. For each of these blocks a vector composed by all the histograms' bins contained in the blocks' cells is taken and normalized.



Naturally, all these operations are computed on the whole image with the same number of pixels per cell, and the same number of cells per block (i.e. a group of cells). Finally, the number of resulting values is equal to *the number of cells in the height of matrix × the number of cells in the length of the matrix × number of bins of the histogram × number of cells per block.*

The various trials we performed gave us the following (best) parameters: 24x24 pixels per cell, 8 bins per histogram, 4 cells for each block, i.e. 2x2 blocks.
Prior to the transformation via gradient, the 762x562 images comprise 428244 parameters, whereas after the dimensionality reduction implemented using the HOG method, the final parameters are reduced to 23552.

After the implementation of the HOG algorithm, the pictures are classified with respect to the subjects' pose, using the Support Vector Machine classifier. The chose parameters of the SVM have been defined using the Grid Search and Cross Validation methods (cv=10), and produced the results for C and γ equal to 0.9 and 0.005, respectively. The kernel function used is the Radial Basis Function (RBF) computed as

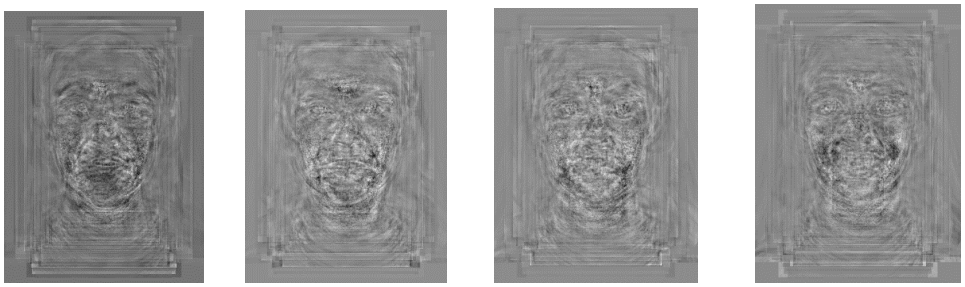
$$\exp(-\gamma \parallel x - x' \parallel^2) \cdot \gamma \qquad \text{where } \gamma > 0$$

Emotion Classification

Once the images have been classified according to the subjects' poses they are passed on to the next step of the pipeline that implements Emotion Classification.

This step comprises the Linear Discriminant Analysis (LDA) and the Support Vector Machine (SVM) methods, which are applied to the five pose groups. The former is used as a dimensionality reduction method and finds the most important components, while the latter takes as input the LDA-modified vectors of images and performs a supervised classification of the emotion.

We decided to take the first four components, as they explained the 90% of variance in each of the five sub-datasets. The components are illustrated below



The values of the SVM parameters C and γ have been chosen using Grid Search and Cross Validation (cv=10), and produced the following results:

	Left	Half Left	Frontal	Half Right	Right
C	0.2	0.9	0.05	1.13	0.22

γ	0.35	0.01	0.15	0.145	0.027

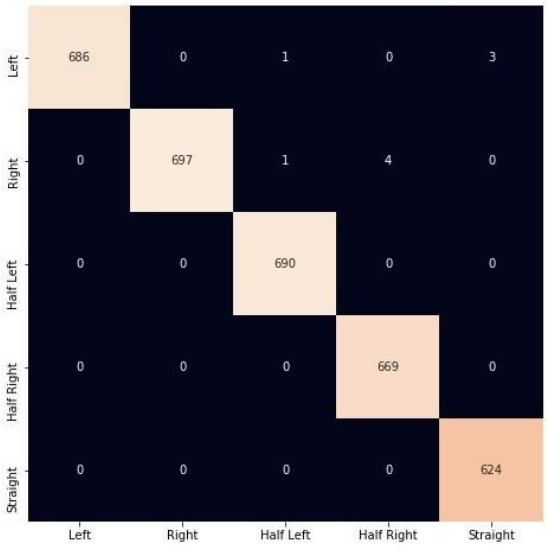
The kernel used for the SVM classification is the RBF kernel already mentioned in the *Pose Classification* section.

Results and Discussion

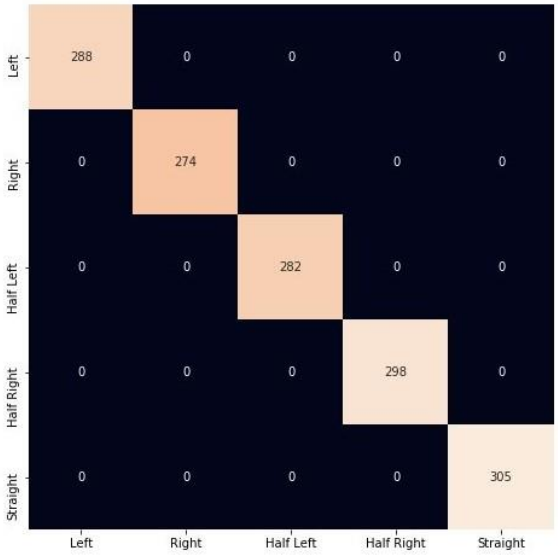
Although our programme tried to minimise the errors by fitting the classification methods on the same object (i.e. the detected faces), rather than on the results of the previous classification, mis-classifications still arose, and errors accumulated in the test set pipeline. These errors stem from the mis-interpretation of the skin-regions (though quite infrequent), on the pose classification and finally on the emotion classification. As a result, without taking into account the issue of overfitting, the precision and accuracy levels obtained in the train are higher than those obtained in the test.

Concerning the Pose Classification, the results obtained from the training set and those obtained from the test set are identical: the HOG+SVM classifier obtained an accuracy score = 0.99 on the training set, and an accuracy = 1 on the test set, as shown by the confusion matrices below.

Training set:



Test set:

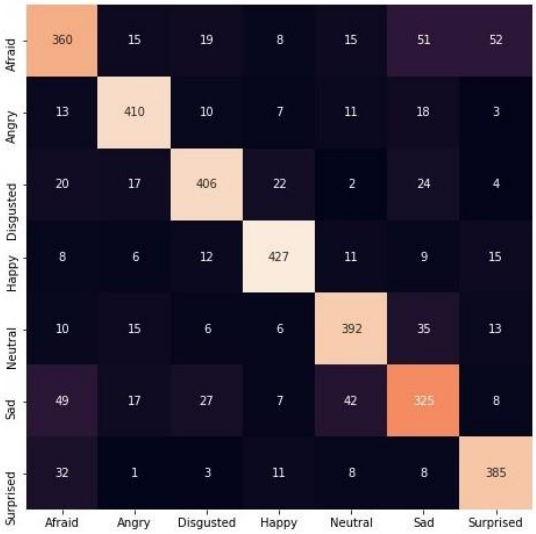


On the other hand, the results are particularly different with regards to the Emotion Classification. The accuracy scores present evident differences, specifically, the accuracy obtained on the training set (condensing all the results from the five different face orientations) is 0.8014, whereas that obtained on the test set is 0.3835.

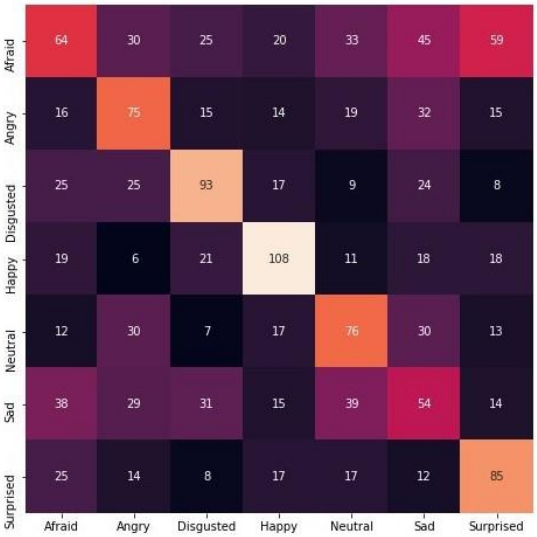
As mentioned above, such a low score might be the result of multiple factors; for instance, a strong overfitting (as the Support Vector Machine method tends to be susceptible to overfitting), as well as the presentation of the data themselves. Although the results on the two sets have been condensed, it important to highlight the important role played by the face orientation when classifying an expression, especially since two of those poses are profiles (left and right), and thus present a lot less features than the corresponding straight (frontal) photos. Therefore, the condensed confusion matrices of the performance on the training and test sets are presented below, as well as the confusion matrices of the individual orientations (on the test set).

Livia Biggi 1793434 Eugenio Bonifazi 1540750 Adriano Fragomeni 1396786

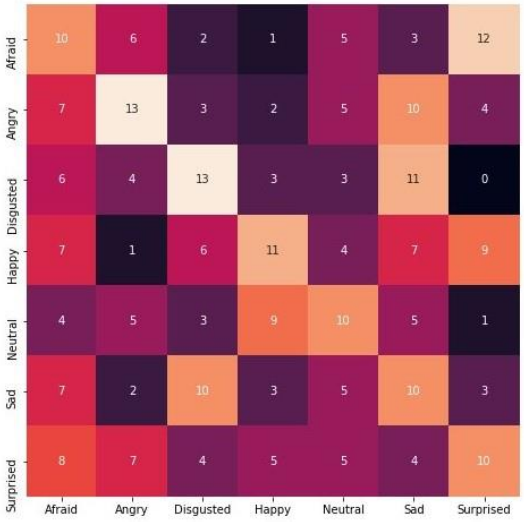
Emotion Classification results on training set:



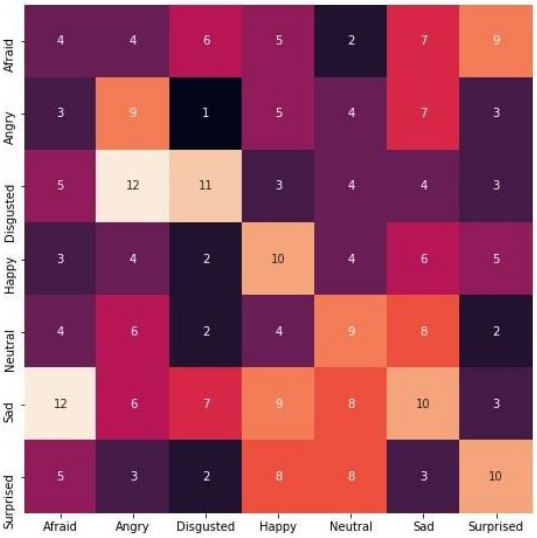
Emotion Classification results on test set:



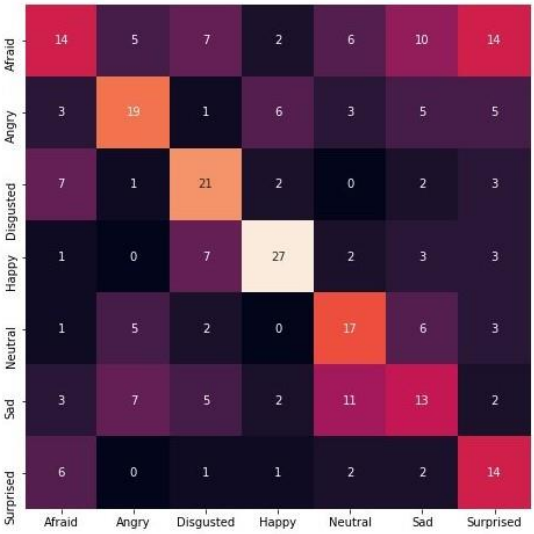
Left (accuracy=0.267)



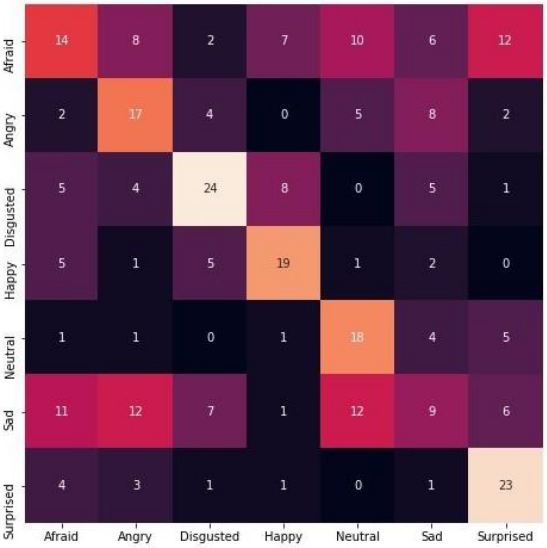
Right (accuracy=0.229)



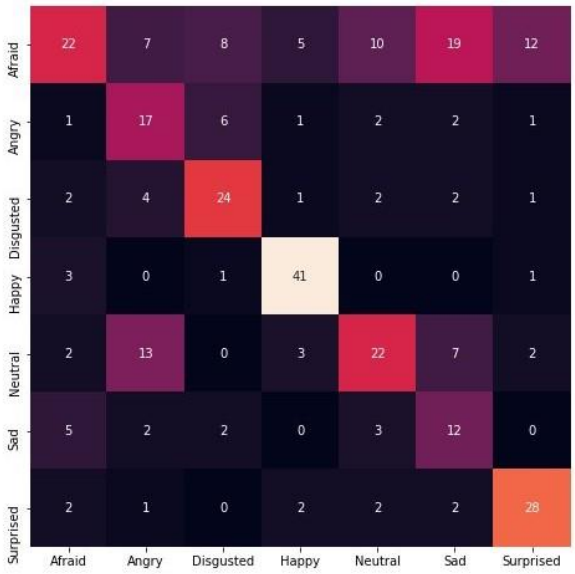
Half Left (accuracy=0.44)



Half Right (accuracy=0.416)



Straight (accuracy=0.54)



It is indeed evident that, depending on the pose, the classifier does perform better – though not enough for it to be considered as efficient as the pose classifier.

Conclusion and Remarks

The analysis carried out aimed at classifying over 4,800 pictures of both men and women, by focusing first on the orientation of their face in the photo, and then on their expression. The classifications methods used include the Histogram of Oriented Gradient, as well as supervised-learning methods, such as the Linear Discriminant Analysis and the Support Vector Machine method. The implementation of the pipeline defined at the beginning of this paper has held optimal results in the classification of the pose, and poor results concerning the emotion of the classification. In general, it has been shown that the accuracy of the Emotion Classifier on the test set varies between 22- and 54%, depending on the pose.

Some of the problematic aspects of this analysis may concern the addition of the padding – the introduction of black pixels around the edges of the images, which might be interpreted by the classifiers as important trends (as the size of the padding is different for every photo). As previously mentioned, the padding is introduced as a means to control the dimensions of the photos of the dataset after cropping the skin regions from the pre-processed image; though this might generate useless noise and might cause the classifier to pick-up uninteresting features and mis-classify the photos based on non-existent trends in the data. An alternative to the padding may be the resizing and rescaling of the pictures, which were discarded as they happened to also affect the skin-region detection and cut out important features from the rescaled image (e.g. the noise or the region below the noise). The padding and other 'noisy' elements of this analysis might be interesting to examine more in-depth in future projects.

References

Cao, G., Huang, L., Tian, H., Huang, X., Wang, Y. and Zhi, R. (2018). Contrast enhancement of brightness-distorted images by improved adaptive gamma correction. *Computers & Electrical Engineering*, 66, pp.569-582.

Contrast stretching using Python and Pillow | Pythontic.com. URL: https://pythontic.com/image-processing/pillow/contrast%20stretching

Lundqvist, D., Flykt, A., & Öhman, A. (1998). The Karolinska Directed Emotional Faces – KDEF, CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, ISBN 91-630-7164-9.

Mccormickml.com. (2018). *HOG Person Detector Tutorial · Chris McCormick*. [online] Available at: http://mccormickml.com/2013/05/09/hog-person-detector-tutorial/

Murphy-Chutorian, E., Doshi, A. and Trivedi, M. (2007). Head Pose Estimation for Driver Assistance Systems: A Robust Algorithm and Experimental Evaluation. *Intelligent Transportation Systems Conference*, Conference.

Tayal, Y., Lamba, R. and Padhee, S. (2012). Automatic Face Detection Using Color Based Segmentation. International Journal of Scientific and Research Publications, 2(6).