VISION
Implementation of a cascade of regressions for facial landmarks localisation

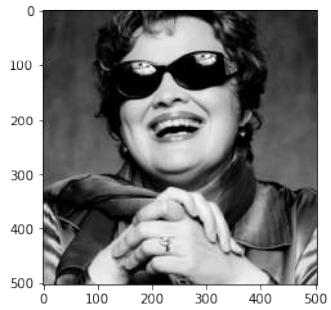
Lucrezia Tosato & Marie Diez

Contents

1 Data Preprocessing

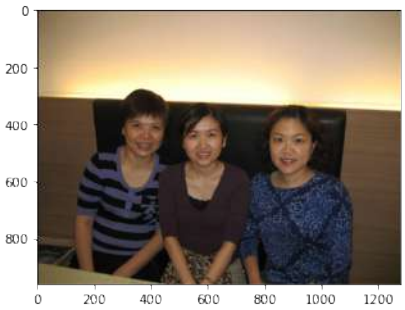
1.1 Data Visualisation

The data were downloaded from the two links with the command **!wget**, to open the .zip files the library **zipfile** was imported and all files were opened.
For parsing the data, the command **.readlines()** was used and all the images were saved in a list.
You can find here some examples of images of the training set :



(a) image 0

(b) image 1



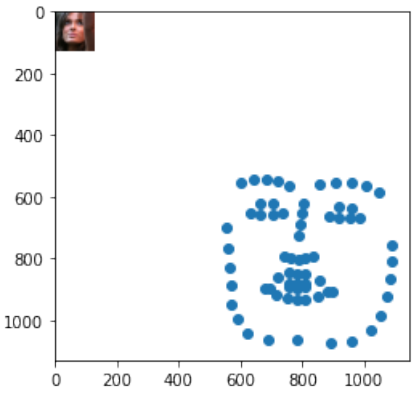
(c) image 11

Figure 1: plot of random 12 images

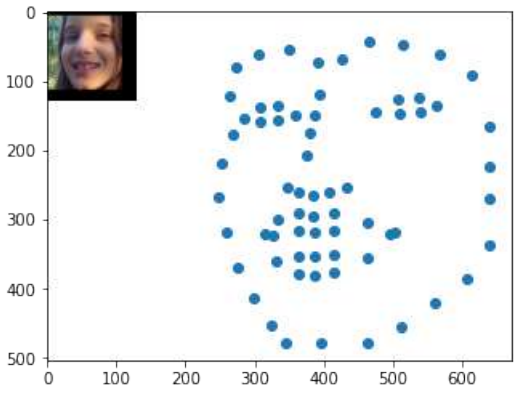
1.2 Data augmentation

To increase the performance of the regression we can use data augmentation, for that we have to :

1. Compute the bounding box
2. Widen the bounding box by 30% and crop the image with this new dimensions, then we resize the image in 128*128



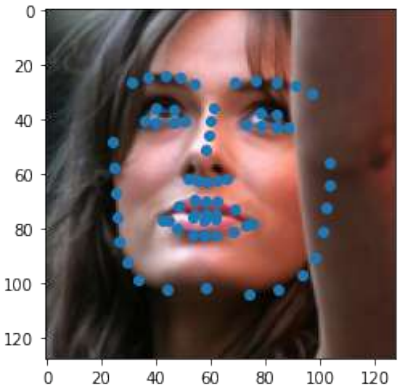
(a) Resized image and associated landmarks without adaptation



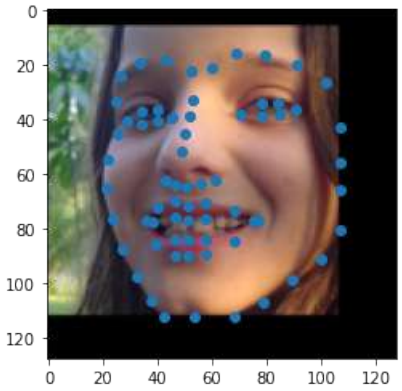
(b) Resized image and associated landmarks without adaptation

Figure 2

3. Compute the new landmark coordinates for this resized image. That will be the ground truth



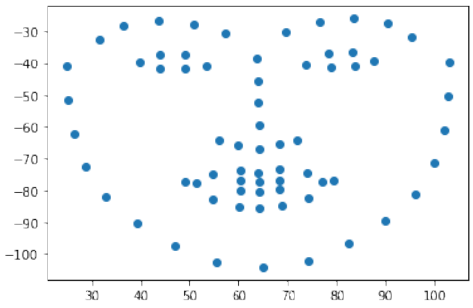
(a) Resized image and associated landmarks with adaptation



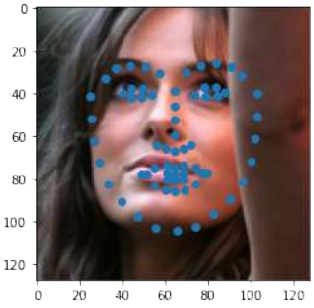
(b) Resized image and associated landmarks with adaptation

Figure 3

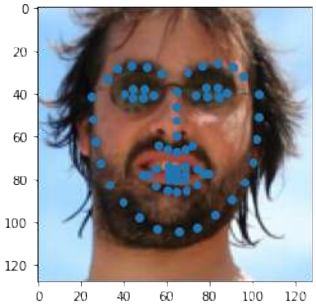
4. Compute the mean position with the **np.mean()**, it's necessary to invert the y values to have an appropriate representation.
   We can see some picture with the mean value of the landmarks applied.



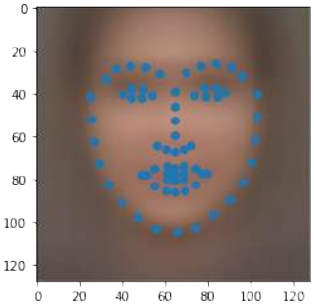
(a) Mean shape



(b) Mean shape on image 1



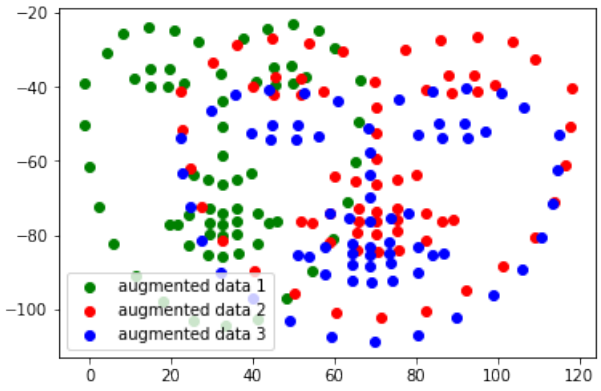
(c) Mean shape on image 2



(d) Mean shape on mean image

Figure 4

5. Generate 10 random perturbations around the mean position (in translation and scaling) around the mean position (Fig. 5a). The amplitude of these perturbations is b±20% and±20 px for scaling and translation respectively.



(a) Randoms perturbations on mean shape

Figure 5

We obtain a clean dataset to train and evaluate our model. We will do the same (without the perturbations) for the test set.

Why do we generate these perturbations? How could they be estimated automatically? The purpose of perturbations is to make learning more robust by data augmentation. Indeed, the model will be able to find the best solution even for an initialization a little bit worse than the mean shape (thus a mean shape with perturbations), the perturbed parameters approach the ground truth. We can use random value between thresholds to estimated automatically some perturbations. Instead of intentional

perturbations to construct an aligned shape model, we can apply the variation within the landmarks themselves.

2 Training a single regressor

Here we want to estimate a displacement of the landmarks, starting from a mean model.

2.1 Feature extraction

The first step is to extract local representation around each landmark.

1. Why do we not directly use the raw value of the image pixels as a representation ? We do not directly use the raw value of the image pixels as a representation since in this way we would also take in consideration external characteristics not proper to facial expression, we have to aligned the data, and remove rotation,translation an scaling with sift features extractor which is invariant for them.

2. We create for each current landmark of the mean model a keypoint from Opencv, we compute a SIFT on each keypoint patch.

3. What is the dimension of each feature? The dimension of each feature is 128 because the sift cut in 4 squares the patch and compute 8 bins histogram on that (4*8=128). For each landmark we have an associated feature of dimension 128.

4. What is the dimension of this feature vector? In 1 image we have 68 features of dimension 128 which means 8704 features. For each image, with the train dataset we have 10 pertubations for each image (3148*10 = 31480 images). So we have 31480 images who each have an associated features vector of dimension 68*128. The dimension of this descriptors matrix will be : (31480, 8704).

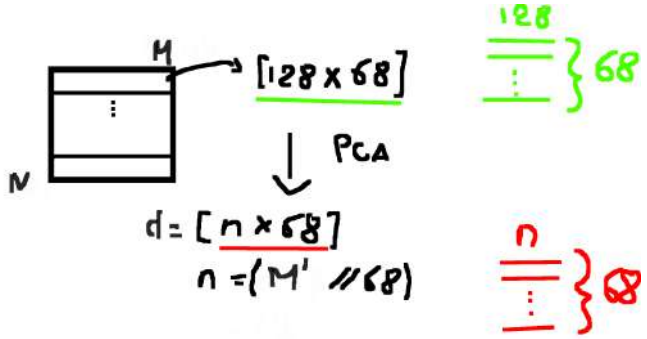
2.2 Dimensionality reduction

The second step is to reduce the dimension of the obtained descriptors.

1. What is the main interest of this dimensional reduction? Could you cite some other dimensional reduction methods for machine learning? The main interest of this dimensional reduction is to keep only the features that have an important information, for that we keep only the dimensions that have a certain amount of the total variance, we can find the number with cumulative variances and with a threshold, here it set to 98%.
Other dimensional reduction methods for machine learning are:

- Non-negative matrix factorization (NMF)
- Linear discriminant analysis (LDA)
- Generalized discriminant analysis (GDA)
- Missing Values Ratio
- Low Variance Filter
- High Correlation Filter
- Backward Feature Elimination
- Forward Feature Construction
- Random Forests/Ensemble Trees

2.



(a) Dimensionality reduction

Figure 6

As the computation time is very long for the whole dataset, we train on the first 1000 images, we want to keep 98% of total variance, that correspond to M'=372 vectors.

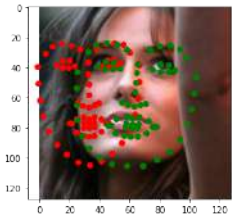
3. What are the dimensions of the new resulting matrix $\tilde{X}_0$? The dimension of the resulting matrix $\tilde{X}_0$ is : (10000, 372).

2.3 Displacement estimation

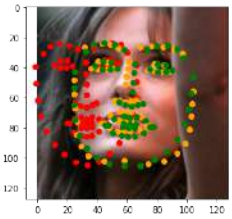
1. Now that we have our reduced features, we compute the estimation of landmark displacement by computing R_0 and b_0 together by adding 1 column to R_0, with least square estimation between the ground truth δ_s^*.

2. To see if R_0 is correct we can compare the predicted result $R_0 X_0$ with δ_s^*. The predicted accuracy is 53%.

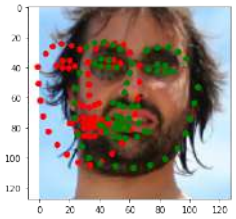
Here are presented some results for different initialisation in red for different images. In green you can find the predict result and in orange the optimal predicted result, on the first 5 images of 300w_train_image :



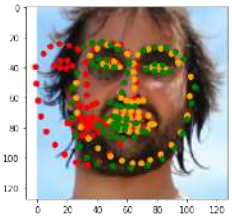
(a) train image 0



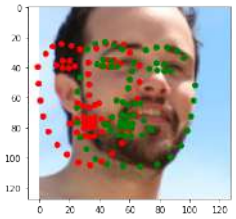
(b) train image 0 with ground truth



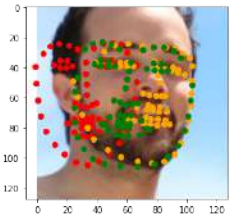
(c) train image 1



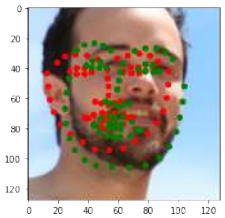
(d) train image 1 with ground truth



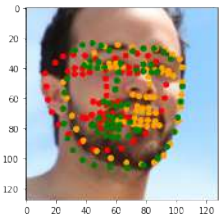
(e) train image 2



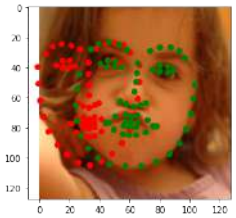
(f) train image 2 with ground truth



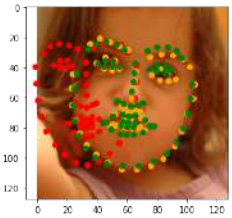
(g) train image 2 with another initialisation



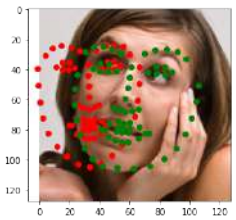
(h) train image 2 with another initialisation with ground truth



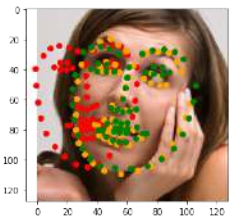
(i) train image 3



(j) train image 3 with ground truth



(k) train image 4



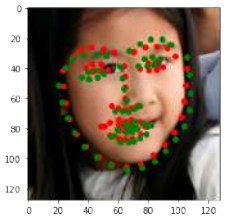
(l) train image 4 with ground truth

Figure 7

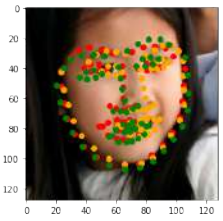
We can see that the predicted results are good.

3. Why this prediction error is not relevant to evaluate our methods ? This prediction error is based on the train dataset, which is not relevant to evaluate our methods, we have to use a test set.

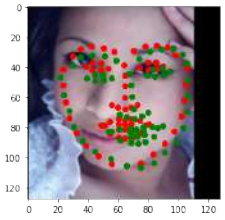
4. We clean the test dataset helen and we compute sift and pca, we get a matrix of dimension (330, 372). Here the results of the first 5 images :



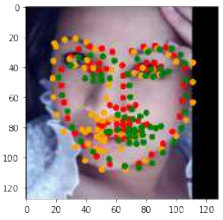
(a) test image 0



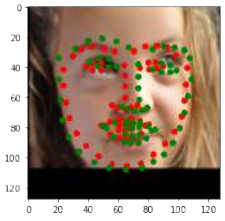
(b) test image 0 with ground truth



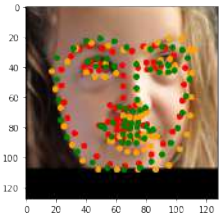
(c) test image 1



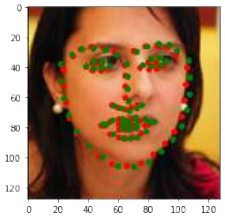
(d) test image 1 with ground truth



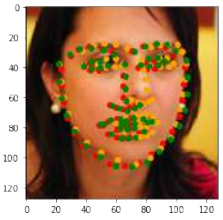
(e) test image 2



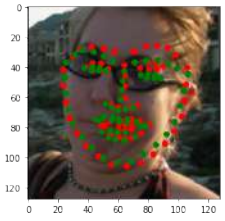
(f) test image 2 with ground truth



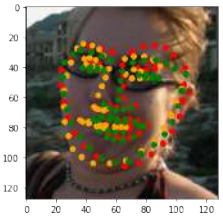
(g) test image 3



(h) test image 3 with ground truth



(i) test image 4

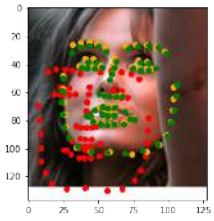


(j) test image 4 with ground truth

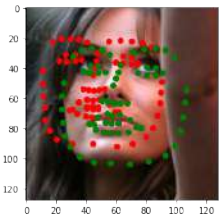
Figure 8

We can see that our predicted results on the test dataset are very bad, the predicted accuracy is : 0.5% We don't find the reason why we have so bad result in test. We tried to use variation of landmarks for the data augmentation and not the variation of the mean shape, with that we obtain 87% of accuracy in train and 81% in test (you can find at the end of the notebook this part).

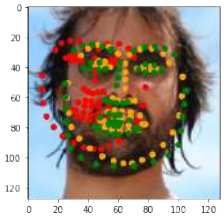
Here are some result with this augmentation :



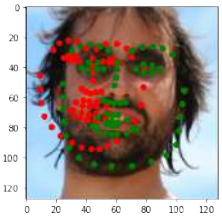
(a) train image 0



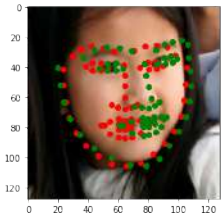
(b) train image 0 with ground truth



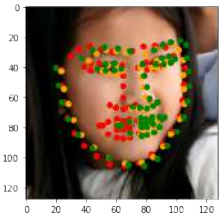
(c) train image 1



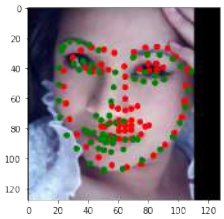
(d) train image 1 with ground truth



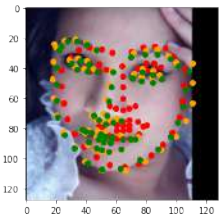
(e) test image 0



(f) test image 0 with ground truth



(g) test image 1



(h) test image 1 with ground truth

Figure 9

To improve our performance we could finish the cascade regression.

References

[1] Rosaria Silipo and Maarit Widmann, 3 New Techniques for Data-Dimensionality Reduction in Machine Learning, The NewStack, 2019.

[2] Benjamin Johnston and Philip de Chazal, A review of image-based automatic facial landmark identification techniques, EURASIP Journal on Image and Video Processing, Article number: 86, 2018.