We choose to work with 2 datasets (section 2.1)

In preprocessing data, we decide to use image processing to enhance image and extract features in order to speed up our model.

Used methods:

* HOG “**histogram of gradients**”

It generalizes the object in the image in a way that makes the output as close as possible for the same object under different conditions and capture (like light and other noise), so it captures important actual features of the object.

We use **skimage** library to handle this because it gives us more control over HOG parameters than **openCV** equivalent function.

Our HOG parameters are:

- 8 bins for the histogram

- 12X12 pixels per cell

- 4X4 cells per block

- with disabled multichannel as we train the model on gray scale images.

All parameters are chosen according to this paper:

<https://link.springer.com/content/pdf/10.1007%2F978-3-642-02172-5_24.pdf>

this paper makes combinations of the whole parameters except normalization, so it gives us the best configuration.

In block normalization we choose **L2** method by try and error.

* **Face landmarks**

It extracts the key features of the face and its position, We use **DLIB** library with trained model to extract 68 face landmarks from each face we pass to it.

We have to choose this model because the size is important to us as its size is acceptable according to what it represents to us.

There are 2 models also, one with 5 landmarks and this isn’t enough for our model and one with 194 landmarks “this could be perfect, but it is size is too large”

*Important note* we use hog and face landmarks according to instructions in a paper

[Z. Zhang, P. Luo, C.-C. Loy, and X. Tang, “Learning Social Relation Traits from Face Images,” in Proc. IEEE Int. Conference on Computer Vision (ICCV), 2015, pp. 3631–3639.]

We get this from [state of art](https://arxiv.org/pdf/1612.02903.pdf?fbclid=IwAR0ajvGcS_-GKoZukSUJ-6jkLhUJjHp5YIxDLx1WjYotA4cIaWKxXjt1zCw) (paper compares papers in facial emotion recognition), this paper supposes that its accuracy 70% but after following it and only get 50-52 % with FER dataset.

With this challenging data set we had to apply some enhancement to remove the noise and remove images with distorted faces.

Used methods:

* For face detection:

We worked with pre-trained models like HAAR, LBP cascading classifiers and DLIB detector.

All of them works with frontal face and each one has its own advantages: HAAR is quite accurate but slow, LBP is faster than HAAR but less accurate and DLIB detector is more accurate than HAAR without any increase in detection time so our default classifier is DLIB detector.

* For image enhancement:

In the training phase we have to ensure that all the input image is gray scale and all of them is the same size.

For noise removal, we can’t cover all kinds of noise so after searching we found that the most common noise is gaussian and salt & pepper.

We remove gaussian noise by fastNLMeans algorithm and salt & pepper with median filter.

We find 3 combinations in out model architecture and here is the summary:

* For FER dataset.
  + Only CNN:
    - It takes – time in training and -- time in testing.
    - The model size was --.
    - Its training accuracy is – and testing accuracy is --.
  + Only HOG and face landmarks:
    - It takes – time in training and -- time in testing.
    - The model size was --.
    - Its training accuracy is – and testing accuracy is --.
  + Both:
    - It takes – time in training and -- time in testing.
    - The model size was --.
    - Its training accuracy is – and testing accuracy is --.
* For CK+ dataset.
  + Only CNN:
    - It takes – time in training and -- time in testing.
    - The model size was --.
    - Its training accuracy is – and testing accuracy is --.
  + Only HOG and face landmarks:
    - It takes – time in training and -- time in testing.
    - The model size was --.
    - Its training accuracy is – and testing accuracy is --.
  + Both:
    - It takes – time in training and -- time in testing.
    - The model size was --.
    - Its training accuracy is – and testing accuracy is --.