Deep Learning for Non Verbal Sentiment Analysis: Facial Emotional Expressions

Nour Meeki, Abdelmalek Amine, Mohamed Amine Boudia and Reda Mohamed Hamou

March 22, 2020

Deep Learning for non verbal sentiment analysis: facial emotional expressions

Nour MEEKI [*] [1], Abdelmalek AMINE [†2], Mohamed Amine BOUDIA [‡3], Reda Mohamed HAMOU [§4]

[1,2,3,4]GeCoDe Laboratory, Department of Computer Science, Tahar Moulay University of Saïda, Algeria

Abstract

People are of a lazy nature and always look for the easiest ways to express themselves and share their experiences and opinions. Due to the popularity of social networks, and to the images expressivity, people have the ability to express themselves throught their use. Our work is about non verbal sentiment analysis using on of the Deep Learning models: CNN (Convolutional Neural Networks). Specifically, we are interested in analyzing the sentiment expressed in facial expressions according to Kaggle's Dataset fer2013 for facial emotion recognition based on the emotions defined by the famous psychologist Ekman namely joy, anger, fear, disgust, sadness and surprise, neutrality is added to the six emotions. Thus, different proposed architectures are used and compared to determine the parameters that affect the results.
The best evaluation resulted in details of around 0,88 showing that the number of convolution layers, the batch_size, the dropout and the epoch number have an impact on the results. However using a CPU cost us a lot which proves that the use of a GPU when using huge amount of data is better and guarantee good results .

Key words: Deep Learning (DL), Sentiment Analysis (SA), Emotionnal Facial Expression(EFE), image classification, Convolutional Neural Network (CNN).

1 Introduction

Data on the web are very large in quantity and of different qualities. With the explosion of the internet and social networks has emerged the need to analyze millions of posts, tweets or opinions in order to know what Internet users think and try to identify the emotions in their messages. The recognition of human emotions has been studied for decades but still remains one of the most complex areas (Liu et al., 2003; Li et al., 2010). There are thre types of emotion recognition: 7% verbal (text), 55% non verbal (gesture, facial expressions) and 38% vocal (voice, intention) (Mehrabian and Wiener, 1967). Our study is focused on non verbal sentiment analysis, specifically facial expressions as the human face is the most expressive part, using one of the Deep Learning models: CNN (convolutional

[*]mekki.nouur@gmail.com

[†]abd_amine1@yahoo.fr

[‡]mamiamounti@yahoo.fr

[§]hamoureda@yahoo.fr

neural networks) which is widely used in the field of image processing and has given very good results. The dataset used is Kaggle's Dataset "fer2013" for facial emotion recognition based on the emotions defined by the famous psychologist Ekman namely joy, anger, fear, disgust, sadness and surprise. Neutrality is added to the six emotions. In ordre to determine the factors that affect the results different architectures are proposed and compared with other works done within the same framework.

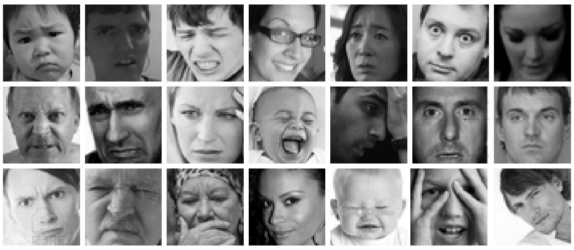


Figure 1: Images in fer2013 dataset

2 Related work

Convolutional Neural Networks (CNNs) are widely used in most of image processing applications, i.e. classifying images(Krizhevsky et al., 2012), grouping them by similarity, performing object recognition in scenes(Grangier et al., 2009). CNNs are typically constitute of input layer, convolution layers, fully connected layers and an output layer. Between the convolutions layers and fully connected layers, there may also be other layers such as pooling, dropout and normalization layers.

- The convolution layer allows to extract the features of an input image by applying a set of filters. It is defined by the following equation:

$$G[m,n] = (f*h)[m,n] = \sum_j \sum_k h[j,k] f[m-j, n-j] \tag{1}$$

- The pooling consists in reducing the dimensions of the images. Its goal is to keep as much relevant information as possible and reduce the number of parameters and calculations in the network.

- The ReLU (Rectified Linear Units) is a non-linear activation function, it is defined by:

$$Relu(x) = \begin{cases} 0 & if \quad x < 0 \\ x & if \quad x > 0 \end{cases} \tag{2}$$

- The dropout is proposed as a regularization method in order to avoid the overfitting problem.

2.1 Non verbal sentiment analysis through CNNs

The task of recognizing facial emotions is a required task for humans, but the transmission of this knowledge by machine is a challenge. Decades of time have been spent by engineering researchers writing computer programs that accurately recognize functionality. Thanks to Deep Learning methods, instead of programming a machine, we can teach it to recognize emotions with great precision. The following works proves this:

(Fathallah et al., 2017) is a study focused only on facial expression recognition that uses a CNN architecture with four convolution layers (with three maximum grouping layers) to extract entities hierarchically, followed by a fully connected layer and a softmax output layer indicating 6 expression classes to predict facial expression. The results and recognition rates have shown that the method used in the work surpasses the methods of the state of the art.

(Jindal and Singh, 2015) studied the sentiment analysis in images that exploits its average level attributes in addition to facial expression recognition. Experiments were conducted on a set of manually labelled Flickr image data, with a rich repertoire of images and associated tags reflecting the user's emotions. A CNN architecture composed of seven internal layers and a softmax layer. The hidden layers consist of five successive convolution layers, followed by two fully connected layers, to determine whether there are advantages when applying CNN to the visual sentiment analysis. Their results prove that CNNs can give good results for the problem of visual sentiment analysis.

(You et al., 2015) is another study where the same dataset as (Jindal and Singh, 2015) was used with half a million Flickr images(from SentiBank). They proposed a PCNN (Progressive Convolution Neural Network) network, as well as its training strategies, which have made it possible to further generalize the trained model and increase the accuracy. The results obtained in (Jindal and Singh, 2015) remain better among the other ones even after a new study made by (Gajarla and Gupta, 2015) on collected data from Flickr.

(Moran, 2019) : used the CNN architecture initially proposed by researcher Amogh Gudi (Gudi et al., 2015), they trained the network for 100 epochs on a set of 14,524 images. Validation was performed using 9000 of the remaining images from the FER2013 dataset. The network achieves optimal prediction accuracy (0,66), but strives to distinguish between the emotions of fear and sadness.

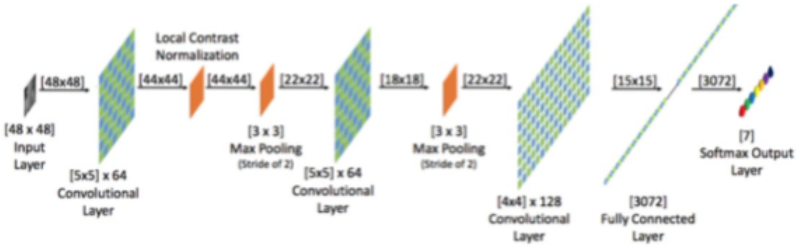

Local Contrast Normalization
[48x48]
[44x44]
[44x44]
[22x22]
[18x18]
[22x22]
[15x15]
[3072]
[48 x 48] Input Layer
[5x5] x 64 Convolutional Layer
[3 x 3] Max Pooling (Stride of 2)
[5x5] x 64 Convolutional Layer
[3 x 3] Max Pooling (Stride of 2)
[4x4] x 128 Convolutional Layer
[3072] Fully Connected Layer
[7] Softmax Output Layer

Figure 2: Architecture (Moran, 2019)

(Giannopoulos et al., 2018) : examined the performance of two known deep learning approaches (GoogLeNet and AlexNet) on facial expression recognition. The training process was designed to go through 5000 iterations on the GoogLeNet and AlexNet experiences. The average loss was presented every 10 iterations, while the accuracy of each network was presented every 500 iterations. The performance of GoogLeNet and AlexNet was studied on the FER-2013 dataset under three aspects, where each evaluates a specific functionality of the methods. In the first part of the study, the performance of networks was studied by recognizing or not the existence of emotional content in a facial expression and then, in the second part of the study, their performance was studied in specifying the exact emotional content of facial expressions. Finally, in the third part, the two methods of deep learning were trained on the emotional and neutral data studied. The results of the three parts are illustrated in figure 3.

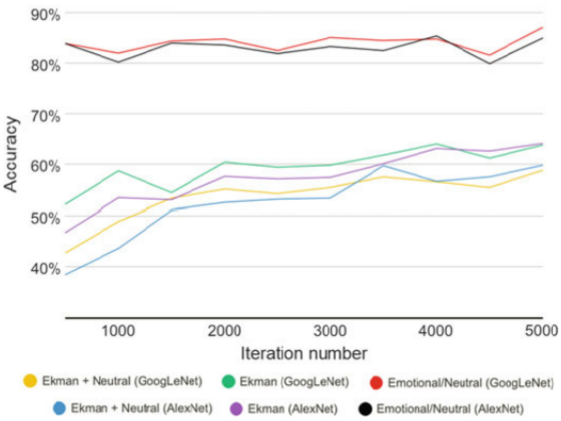


Figure 3: Accuracy of (Giannopoulos et al., 2018)

(Nishchal et al., 2018) : two different models were evaluated in their article according to their precision and the number of parameters. The initial architecture proposed is a standard CNN which includes 9 convolution layers, ReLU, batch normalization and Global Average Pooling, it contains on average 600,000 parameters. The evaluation of the proposed model made it possible to obtain details of approximately 0,66. The first model was given the name fully-CNN sequential. The second model is driven by the Xception (Chollet, 2017) architecture, this architecture has reached an accuracy of 0,95.

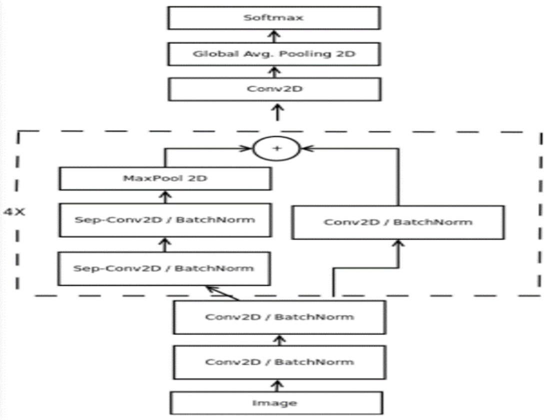


Figure 4: Xception's architecture (Chollet, 2017)

(Serengil, 2018): is another work carried out in the same framework, the constitution of their architecture is as follows: 5 layers of convolutions followed by 3 layers of pooling and 3 layers fully connected. The evaluation of their model made it possible to obtain an accuracy of 0,92 and a loss of 0,22 with a number of epochs equal to 100.

Table1 represents accuracy of the related works using other dataset than ours and table2 is about the related works using the same dataset as ours (fer2013).

Work	Method	Dataset	Accuracy
(Fathallah et al., 2017)	4 Conv + FC + Soft-Max	CK+/KDEF/RaFD/MUG	0.969
(Jindal and Singh, 2015)	5 Conv + 2FC	Flicker	0.535
(You et al., 2015)	PCNN(Progressive CNN)	Flicker	0.781
(Gajarla and Gupta, 2015)	ResNet-50	Flicker	0.733

Table 1: Related works accuracy 1

Architecture		**Accuracy**
(Serengil, 2018)		0.92
(Moran, 2019)		0.65
(Giannopoulos et al., 2018)	AlexNet	0.82
	GoogleNet	0.87
(Nishchal et al., 2018)	sequential fully-CNN	0.66
	Xception	0.95

Table 2: Related works accuracy 2

3 Our experimentation and results

The puropse of our study is to determine wich parameters affect the results of a model.

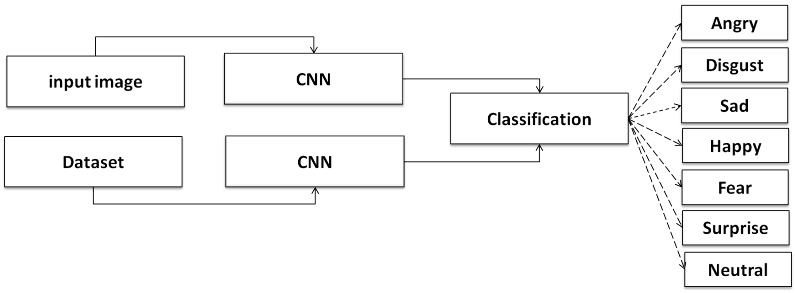


Figure 5: Emotional expression recognition system

3.1 Model evaluation metrics

To evaluate our models we used the following metrics:

- Accuracy: Calculates the average accuracy rate of all forecasts.

$$Accuracy = \frac{TruePositif + TrueNegatif}{TruePositif + TrueNegatif + FalsePositif + FalseNegatif} \tag{3}$$

- Loss: used to measure the inconsistency between the predicted value (p) and the actual wording (t).

$$Loss = -\sum_{j} t_{i,j} \log(P_{i,j}) \tag{4}$$

- Confusion Matrix: used to describe the performance of a classification model on a set of test data whose actual values are known.

3.2 Proposed architectures

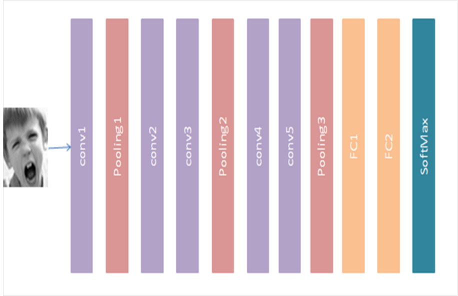
We have proposed 3 different architectures described bellow.

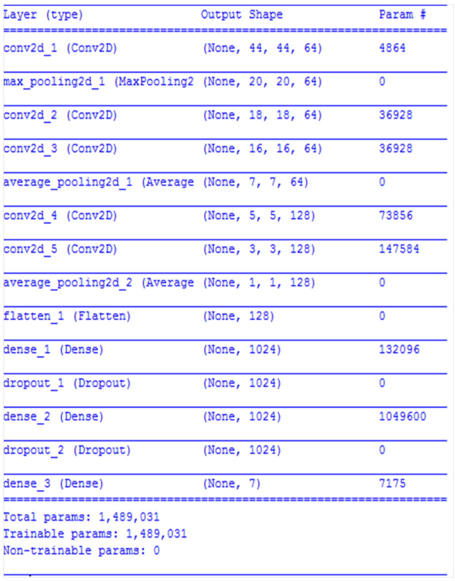
1. Architecture 01: Consists of 5 convolution layers, 3 pooling layers and 3 fully connected layers. The input image is (48*48), the image first goes to the first convolution layer. This layer is composed of 64 filters of size (5*5), each of our convolutional layers is followed by a function of activation ReLU this function forces the neurons to return positive values, after this convolution 64 features maps of size (44*44) will be created, then a Maxpooling with cells of size (5*5) is applied. Its function is to reduce the spatial size of the incoming entities and thus contributes to reducing the number of parameters and calculations in the network, thus helping to reduce over-learning. At the end of the Maxpooling layer, we will have 64 feature maps of size (20*20). The 64 feature maps obtained are input to the second convolution layer which is also composed of 64 (3*3) size filters. The 64 feature maps will serve as input for the third convolutional layer that is similar to the second layer in its constitution and are followed by an AveragePooling layer with(3*3) cells. The fourth and fifth convolutional layers have 128 filters of size (3*3) and the activation function chosen is the same as that of the other layers. They are followed by an AveragePooling layer. At the exit of the AveragePooling layer, we will have 128 feature maps of size (1*1). The feature vector resulting from the convolutions has a dimension of 128. To finish the construction of the architecture, we use 3 fully connected layers. The first 2 fully-connected layers calculate a vector of size 1024, and are each followed by a ReLU layer and a dropout equal to 0.2. The last layer returns the vector of probabilities of size 7 (the number of classes) by applying the softmax function. The given summary includes informations about : the layers and their order in the model, the output shape of each layer, the number of parameters (weight) in each layer, and the total number of parameters (weight) in the model. Parameters in convolutional layers are calculated by the given formular:

$$Parameter = (filtreSize \times inputFeatureMaps + 1) \times outputFeatureMaps \tag{5}$$

Parametrs in fully connected layers are calculated by the given formular:

$$Parameter = (input + 1) \times output \tag{6}$$





```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 44, 44, 64)        4864
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 20, 20, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 18, 18, 64)        36928
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 64)        36928
_________________________________________________________________
average_pooling2d_1 (Average (None, 7, 7, 64)          0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 5, 5, 128)         73856
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 3, 3, 128)         147584
_________________________________________________________________
average_pooling2d_2 (Average (None, 1, 1, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              132096
_________________________________________________________________
dropout_1 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dropout_2 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_3 (Dense)              (None, 7)                 7175
=================================================================
Total params: 1,489,031
Trainable params: 1,489,031
Non-trainable params: 0
_________________________________________________________________
```

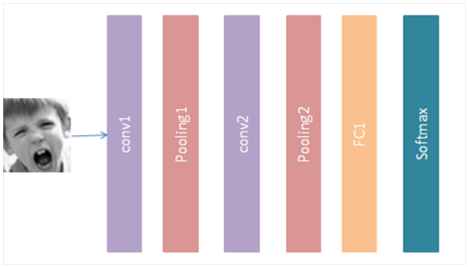
Figure 6: First model architecture and summary

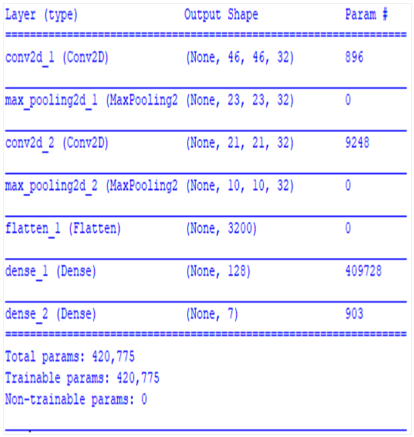
From the confusion matrix presented in table 3, we deduce that among a set of images consisting of 3589 a number of 1997 images have been well classified. The model made a good learning of the classes of emotions expressing: happiness, surprise and neutral.

	Anger	Disgust	Fear	Happiness	Sad	Surprise	Neutral
Anger	161	4	75	65	76	10	76
Disgust	9	19	6	5	10	0	7
Fear	41	1	187	49	91	47	80
Happiness	26	0	33	709	36	16	75
Sad	46	2	81	79	274	21	150
Surprise	16	1	46	37	7	292	16
Neutral	37	0	43	83	71	18	355

Table 3: First model confusion's matrix

2. Architecture 02: Consists of 2 convolution layers, 2 pooling layers and 2 fully connected layers. The input image has a size of (48 * 48), the two convolution layers have a set of 32 (3*3) size filters and a Relu activation function. Each layer is followed by a MaxPooling layer with (2*2) size windows. At the end of the MaxPooling layer, we will have 32 feature maps of size (10*10). The feature vector resulting from the convolutions therefore has a dimension of 3200. The fully connected first layer calculates a vector of size 128 and followed by a layer Relu and a dropout which is equal to 0.2. The fully connected second layer returns a vector of probabilities of size 7 (the number of classes) and the softmax function is applied to it.





```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 46, 46, 32)        896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 23, 23, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 21, 21, 32)        9248
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 10, 10, 32)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 3200)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               409728
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 903
=================================================================
Total params: 420,775
Trainable params: 420,775
Non-trainable params: 0
_________________________________________________________________
```

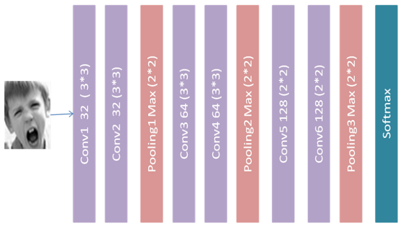
Figure 7: Second model architecture and summary

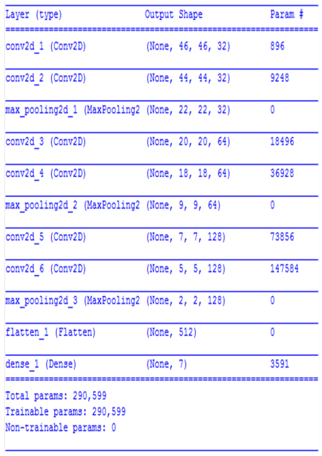
For this model two representations were used. In the first respresentation a batch_size of size 16 was used while a batch_size of 96 was used in the second one.

Architecture 02	batch_size	Accuracy	Loss
First representation	16	0.718	0.757
second representation	96	0.76	0.63

Table 4: A comparaison between the two representations of the second architecture

3. Architecture 03: Consists of 6 convolution layers, 3 pooling layers and one fully connected layer. Same as the two previous architectures, the input image is 48 * 48. The first two convolution layers have a set of 32 (3*3) size filters and a Relu activation function, followed by a MaxPooling with 2 * 2 size windows. The convolution layers 3 and 4 have a set of 64 (3*3) size filters and the Relu activation function is used in both layers. They are also followed by a MaxPooling layer with (2*2) size windows. The only difference with layers 5 and 6 is that the applied filter set is 128. 128 feature maps of size (2*2) are output from the third layer of MaxPooling. A vector of dimension 512 is obtained. For the last layer, the fully connected layer, the softmax function is applied, and the returned probability vector is of size 7 (the number of classes).





```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 46, 46, 32)        896
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 44, 44, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 22, 22, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 20, 20, 64)        18496
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 18, 18, 64)        36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 9, 9, 64)          0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 7, 7, 128)         73856
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 5, 5, 128)         147584
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 2, 2, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 3591
=================================================================
Total params: 290,599
Trainable params: 290,599
Non-trainable params: 0
_________________________________________________________________
```

Figure 8: Third model architecture and summary

The table 5 summarizes the results obtained for the three proposed models. From these results, we notice that the performance of a model depend on the number of convolution layers used (the higher the number, the more accurate the model is) and the size of the batch_size used (the larger it is, the better we have a good training model). The execution time depends on the complexity of the model (the more fully connected layers we have, the more time it takes to execute).
Despite the complexity of the first model, its results are not satisfactory because we have a considerable loss of information (0,70). So, we can see that the third model is the most reliable of the three models proposed since we have an accuracy of 0,889 and a loss of 0,29.

Architecture	Accuracy	Loss	Execution time
Architecture 01	0.77	0.70	120 h
Architecture 02 R1	0.718	0.757	72 h
Architecture 02 R2	0.76	0.63	48 h
Architecture 03	0.889	0.29	72 h

Table 5: Table comparision between the proposed models

The following figures represent predictions made on a few images from FER2013's private test devoted to the evaluation. The first model predictions were not that succeful as the predictions made by the two other models.

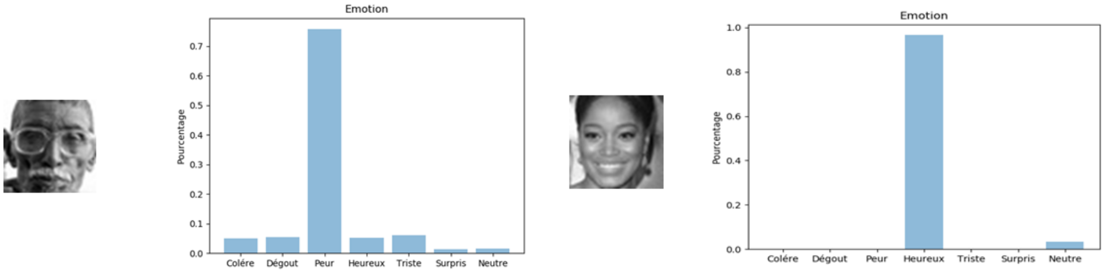


Figure 9: Predictions of the first model

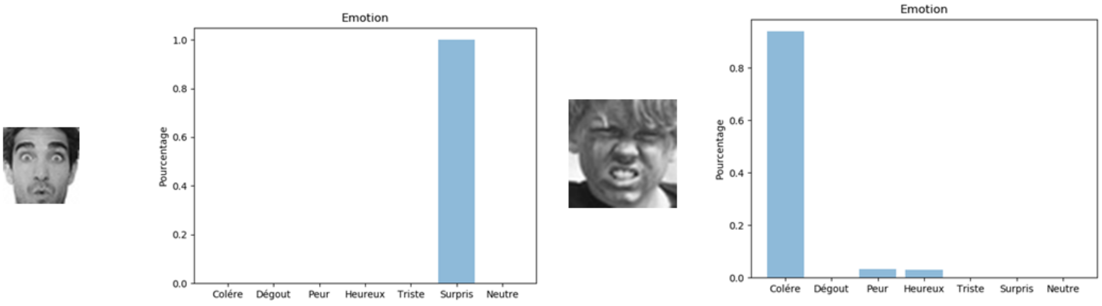


Figure 10: Predictions of the second model

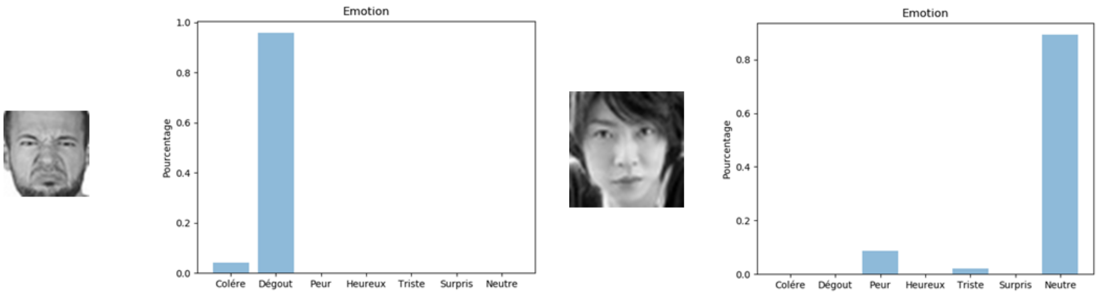


Figure 11: Predictions of the third model

From the comparison of our results Table 5 with the related works using the same dataset "fer2013" Table 2, we can conclude that the obtained results are acceptable since the accuracy values are included between the two architectures proposed in (Nishchal et al., 2018), the Xception architecture that gave very good results (accuracy= 0,95) and the sequential architecture fully-CNN which has an accuracy value of 0,66.

4 Conclusion

A picture is worth a thousand words. This work have been interested in analyzing non verbal sentiment, especially emotional facial expressions. To solve our problem, we used Convolutional Neural Networks (CNNs) as a deep learning architecture. We presented three different models in order to determine the factors that give good results in terms of accuracy and loss. Thus, we found that the more we have: convolution layers, a good dropout, a large size of batch_size, and a large number of epochs, the more the results are satisfactory. The third model showed the best results. The number of convolution layers and the size of the batch_size reflect these good results, but the execution time was expensive. The use of a CPU during the training phase has cost precious time. This amounts to the large size of the dataset, which requires the use of a GPU instead.
For our next work, we plan to do a general non verbal sentiment analysis (images and videos), in order to be able to determine the feeling released in. We also plan to do a hybrid sentiment analysis between the verbal, non verbal and introduce the vocal eventually.

References

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258.

Fathallah, A., Abdi, L., and Douik, A. (2017). Facial expression recognition via deep learning. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pages 745–750. IEEE.

Gajarla, V. and Gupta, A. (2015). Emotion detection and sentiment analysis of images. *Georgia Institute of Technology.*

Giannopoulos, P., Perikos, I., and Hatzilygeroudis, I. (2018). Deep learning approaches for facial emotion recognition: A case study on fer-2013. In *Advances in Hybridization of Intelligent Methods*, pages 1–16. Springer.

Grangier, D., Bottou, L., and Collobert, R. (2009). Deep convolutional networks for scene parsing. In *ICML 2009 Deep Learning Workshop*, volume 3, page 109. Citeseer.

Gudi, A., Tasli, H. E., den Uyl, T. M., and Maroulis, A. (2015). Deep learning based facs action unit occurrence and intensity estimation. *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG).*

Jindal, S. and Singh, S. (2015). Image sentiment analysis using deep convolutional neural networks with domain specific fine tuning. In *2015 International Conference on Information Processing (ICIP)*, pages 447–451. IEEE.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Li, G., Hoi, S. C., Chang, K., and Jain, R. (2010). Micro-blogging sentiment detection by collaborative online learning. In *2010 IEEE International Conference on Data Mining*, pages 893–898. IEEE.

Liu, B., Dai, Y., Li, X., Lee, W. S., and Philip, S. Y. (2003). Building text classifiers using positive and unlabeled examples. In *ICDM*, volume 3, pages 197–188. Citeseer.

Mehrabian, A. and Wiener, M. (1967). Decoding of inconsistent communications. *Journal of personality and social psychology*, 6(1):109.

Moran, J. L. (2019). Classifying human emotion using convolutional neural networks. *UC Merced Undergraduate Research Journal*. https://escholarship.org/uc/item/1t89f7xk.

Nishchal, P. C., Chengappa, P., Raman, T., Pandey, S., and Shyam, G. K. (2018). Emotion identification and classification using convolutional neural networks. *International Journal of Advanced Research in Computer Science*, 9(Special Issue 3):357.

Serengil, S. I. (2018). Facial expression recognition with keras. https://sefiks.com/2018/01/01/facial-expression-recognition-with-keras/?fbclid=IwAR2sJwE1ydZmpErqyBEzFaZNY9crXz-g-oDuYZJ_YmRf7RoSUjxLzmbbhCo Consulté le 20/04/2019.

You, Q., Luo, J., Jin, H., and Yang, J. (2015). Robust image sentiment analysis using progressively trained and domain transferred deep networks. In *Twenty-Ninth AAAI Conference on Artificial Intelligence.*