CNNSFR: A Convolutional Neural Network System for Face Detection and Recognition

Lionel Landry SOP DEFFO[1], Elie TAGNE FUTE[2]
Department of Computer Engineering, Department of Mathematics and Computer Science,
{University of Dschang, University of Buea} Cameroon

Emmanuel TONYE[3]
National Advanced School of Engineering, Department of Electrical Engineering
{University of Yaounde I} Cameroon

***Abstract*—In recent years, face recognition has become more and more appreciated and considered as one of the most promising applications in the field of image analysis. However, the existing models have a high level of complexity, use a lot of computational resources and need a lot of time to train the model. That is why it has become a promising field of research where new methods are being proposed every day to overcome these difficulties. We propose in this paper a convolutional neural network system for face recognition with some contributions. First we propose a CRelu module, second we use the module to propose a new architecture model based on the VGG deep neural network model. Thirdly we propose a two stage training strategy improved by a large margin inner product and a small dataset and finally we propose a real time face recognition system where face detection is done by a multi-cascade convolution neural network and the recognition is done by the proposed deep convolutional neural network.**

***Keywords*—Convolutional neural network; face recognition; VGG model; CReLU module; deep learning; architecture**

I. INTRODUCTION

High-quality cameras in mobile devices have made facial recognition a viable option for authentication as well as identification. However, the used multimedia computational devices cannot act as well human being does. That is why studies have tried to mimic the behavior of human brain to approximate artificially the results obtained by a human being: it is the notion of deep learning. In the mid-1960s, scientists began work on using the computer to recognize human faces. Since then, facial recognition software has come a long way.

In 1966, Bledsoe [1], [2] developed a system that could classify photos of faces by hand using what's known as a RAND tablet, a device that people could use to input horizontal and vertical coordinates on a grid using a stylus that emitted electromagnetic pulses

In 1987, Sirovich and Kriby [3], were able to show that feature analysis on a collection of facial images could form a set of basic features. They were also able to show that less than one hundred values were required in order to accurately code a normalized face.

In 1991, Turk and Pentland [4] expanded upon the Eigen face approach by discovering how to detect faces within images. This led to the first instances of automatic face recognition

From 1993 to 2000 the Defense Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology rolled out the Face Recognition Technology (FERET) program [5] which consists of creating a database of facial images. The database was updated in 2003 to include high-resolution 24-bit color versions of images. Included in the test set were 2,413 still facial images representing 856 people.

From 2005, the Face Recognition Grand Challenge (FRGC) [6] consisted of progressively difficult challenge problems was launched. It includes sufficient elements to overcome the lack of data. The set of defined experiments assists researchers and developers in making progress to meet the new performance goals.

The year 2010 was marked with a great change in the social media platforms all over the world and has leaded researchers to develop photo tagging feature for its user. However the accuracy was not that satisfying that is why technologies using deep learning such as deep face where born [7]. His tools identify human faces in digital images. It employs a nine layer neural network with over 120 million connection weights and was trained on four million images uploaded by Facebook users.

Many other models have been developed over years and two of the most popular are Facenet network [8] and VGG network [9]. They propose a deep architecture that is able to deal with the complexity of classification problem. However, these architectures generally need a very huge date set and a lot of iterations to have good results which if often difficult to have in some cases.

This paper presents a convolutional Neural Network System for Face Recognition based on VGG model and has four proposed contributions. First we propose a CRelu module that has proved to be efficient in enhancing computations; second we use the module to propose a new architecture of VGG network. Thirdly we propose training strategy that needs small dataset and we prove that it leads to good results and finally we propose a real time face recognition system where face detection is done by a multi-cascade convolution neural network and the recognition is done by the proposed deep convolutional neural network.

The rest of the paper is organized as follows: Section 2 presents the details on the proposed approach. In Section 3, the training methodology is presented. Section 4 presents the

implementation, analysis and results interpretations included. Finally, Section **5** concludes the work by doing an appraisal and by proposing amelioration perspectives.

II. METHOD

In this section, we present our proposed model for face recognition based on the VGG [10] deep convolutional neural network. It is a deep architecture that has been developed by the visual geometric group of the University of Oxford in 2015. It has proven to be very efficient in the image recognition task. In addition we have noticed that the deeper the network, the better are the results for more coefficients are used to compute the expected results. Also, we have noticed that the choice activation function is also crucial when designing the network and commonly for convolutional neural networks, the used function is ReLU (Rectified Linear Unit, Rectifier) which is an activation function for Neural Network, known as a ramp function and applied to computer vision and speech recognition. It has been used with some success in restricted Boltzmann machines for computer vision tasks [11].

Several variations have been proposed, like ELU [12] (Exponential linear unit), PReLU [13] (Parametric rectified linear unit), LReLU (Leaky ReLU) [14] and RReLU [15] (randomized ReLU). In contrast to ReLU, in which the negative part is totally dropped, leaky ReLU assigns a noon-zero slope to it .In PReLU, the slopes of negative part are learned from data rather than predefine and has prove to be a key factor of surpassing human-level performance on ImageNet classification task. ELU speeds up learning and alleviates the vanishing gradient problem however; it positive part has a constant gradient of one so it enables learning and does not saturate a neuron on that side of the function. In ReLU, the slopes of negative parts are randomized in a given range in the training, and then fixed in the testing and could reduce over-fitting due to its randomized nature.

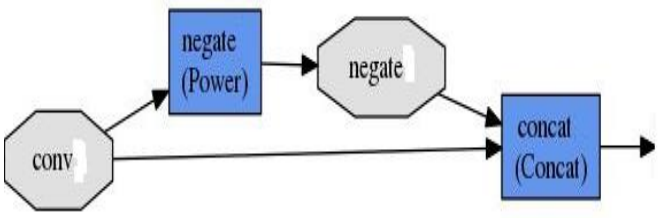

Fig 1. Proposed CRelu Module.

In this view, we propose a simple CReLU, where the idea in general is to concatenate a ReLU which selects only the positive part of the activation with a ReLU which selects only the *negative* part of the activation. Note that as a result this non-linearity doubles the depth of the activations [16]. This is with the knowledge that CReLU increases the quality of the result as proven in [17]. We therefore propose a simple CReLU shown in Figure 1.

We can see how we have connected the output of the convolution to the negation of the same output. The next step is to replace every activation functions, here is ReLU and PreLU essentially and it gives rise to a new architecture of the VGG model.

A. The Presentation of the VGG Model

It is a deep convolutional network for object recognition developed and trained by Oxford's renowned visual geometric group (VGG), which achieved very good performance on the ImageNet dataset. It is quite famous because not only it works well, but the Oxford team has made the structure and the weights of the trained network freely available on-line.

The idea of the VGG group members was to give an answer to "how to design the network structure". Among many choices, they has adopted the simplest. Only 3x3 convolutions and 2x2 pooling are used throughout the whole network. They have also used the fact that the depth of the network plays an important role. Deeper networks give better results.

Figure 2 gives the structure of the model, which takes input image of size 224 * 224 * 3 (RGB image), built using Convolutions layers (used only 3*3 size), max pooling layers (used only 2*2 size), a fully connected layers at end and has a of total 16 layers. Below is the description of each layer.

1) Convolution using 64 filters
2) Convolution using 64 filters + Max pooling
3) Convolution using 128 filters
4) Convolution using 128 filters + Max pooling
5) Convolution using 256 filters
6) Convolution using 256 filters
7) Convolution using 256 filters + Max pooling
8) Convolution using 512 filters
9) Convolution using 512 filters
10) Convolution using 512 filters + Max pooling
11) Convolution using 512 filters
12) Convolution using 512 filters
13) Convolution using 512 filters + Max pooling
14) Fully connected with 4096 nodes
15) Fully connected with 4096 nodes
16) Output layer with Softmax activation with 1000 nodes

B. The Proposed Model

We have already explained in details the proposed CRelu module and presented in details the architecture of the VGG chosen model. It is therefore important to mention that the model uses the ReLu activation function and is used 15 times in the network. Our proposed model will therefore replace these ReLu function by the the proposed module. Also in the last layer (layer 16) the softmax inner product is replaced by the combination of Large Margin Inner Product and Softmax with Loss. It is usually called L-Softmax loss [18] built for convolutional neural networks and this loss can greatly improve the generalization ability of CNNs, so it is very suitable for general classification, feature embedding and biometrics. This gives rise to the architecture presented on Figure 3.

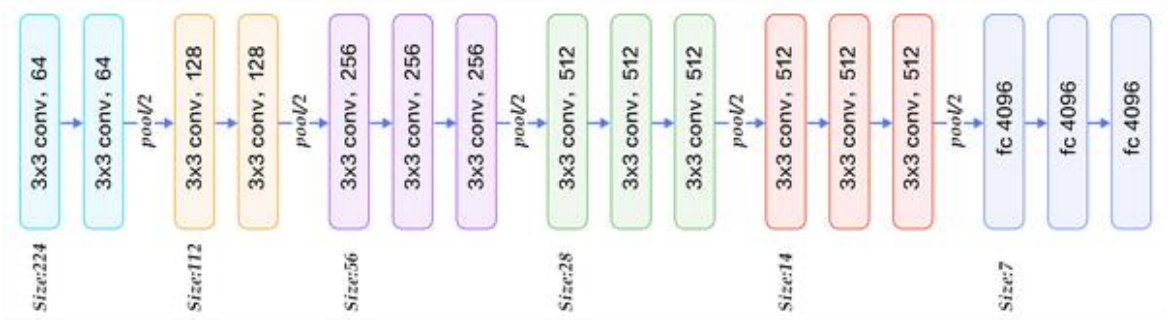

Fig 2. VGG Model.

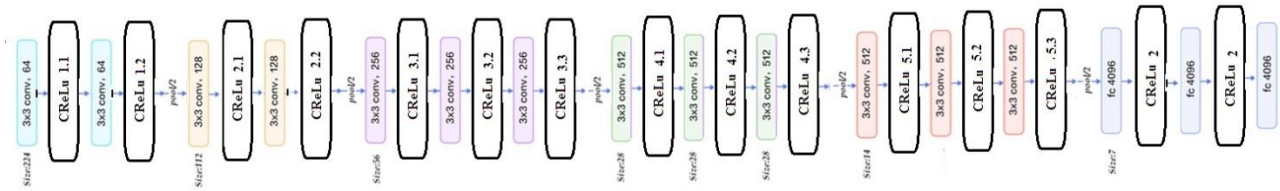

Fig 3. Proposed VGG Model.

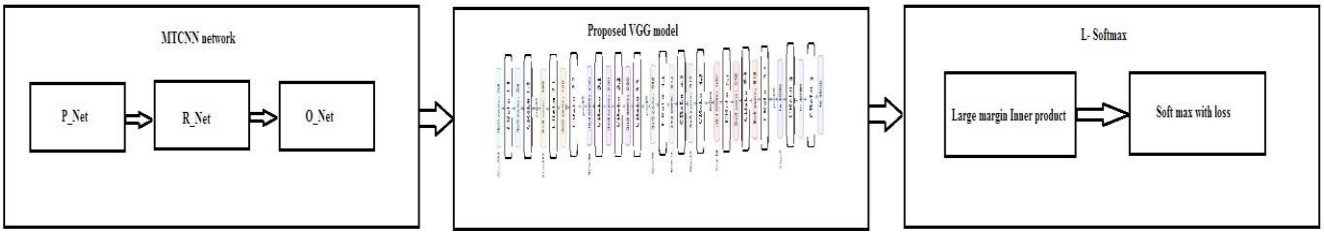

Fig 4. Final Architecture of the System.

C. The Real Time System

Now that we have proposed the recognition model we combine it to a detection model to produce our final framework. It is well known that a face recognition system passes through a detection phase before recognition. However the proposed approaches in the literature usually use face cascade detection which is relatively old. We have decided to use the MTCNN (multi cascade convolutional neural network). It is based on:

- A Proposal Network (P-Net) used to obtain the candidate facial windows and their bounding box regression vectors. Then candidates are calibrated based on the estimated bounding box regression vectors. Finally a non-maximum suppression (NMS) is employed to merge highly overlapped candidates.
- A Refine Network (R-Net), which further rejects a large number of false candidates, performs calibration with bounding box regression, and conducts NMS.
- An O-Net network which aims to identify face regions with more supervision. In particular, the network will output five facial landmarks positions.

We combine this multi cascade neural network with the proposed VGG model and finally we add an L-softmax module which is as stated earlier composed of a large margin inner product and a softmax with loss. Not that this module replaces the last inner product layer of the proposed VGG model. This give rise to the architecture presented on Figure 4.

III. Training Methodology

To train our model, we perform the following steps:

- We gather images that will be used for training and divide them into train set and test set with the ratio 2/3 and 1/3. These images are usually taken from public datasets where each identity has at least 80 images.
- Each identity is assigned a label, consequently all the images of one identity is assign a unique number. This leads to the creation for each identity a file containing the names of all it images zit hot corresponding label
- We gather into one file all the names and label and shuffle the obtained results such a way that all images names of one identity should not be adjacent. Note that it is better for each name of image to be written with it absolute path

- Divide the images into train set and test set with ratio 2/3 and 1/3 indicating that the number of images for one identity in the training set should be the double of the one present in the test set.
- When all this are done you can use that information to train your network. But first of all a training is done using the model and no initial weigh value then the, obtained weigh values are used to fine-tuned the same work. In our case we have decided to take 1000 iterations for the first training and 9000 iterations to fine-tune. This has proven that it is more efficient than the one using the previous approach.

IV. Results

A. Training Results

We choose Caffe [19] to implement our solution. It is Caffe a deep learning framework made with expression, speed, and modularity in mind. It is developed by Berkley AI Research (BAIR) and by community contributors. The choice is motivated by: Expressive architecture, Extensible code, Speed and Community.

It is written in C/C++ and has a python interface. The parameters used to train our model are listed in table 1.

Since we are working in CPU mode, it was almost impossible to work on large dataset for the resources are limited in that mode. The GPU memory of the machine used is only of 2GB so was unable to do more than 10 steps with a memory dump. For that reason we have chosen 7 identities from the pub83 [20] dataset and the last one is that of the second author. Each of these identities has at least 80 pictures for the training set and at least 20 pictures for the testing set which gives a total of about 100 images per persons.

We had the following results during training.

Figure 5 shows the variation of loss during training as well of that of the accuracy. We can notice that the accuracy tend to increase and decrease later on. For the lost it seems to increase only.

TABLE I. Parameters used in the Training Process

Parameters	Values
Number of iterations	10 000
Initialization method	Xavier method
Propagation algorithm	Stochastic gradient descent (SGD)
momentum	0.9
Weigh decay	0.0005
Batch size	8
Learning rate	0.001
Test interval	20
Step size	2000

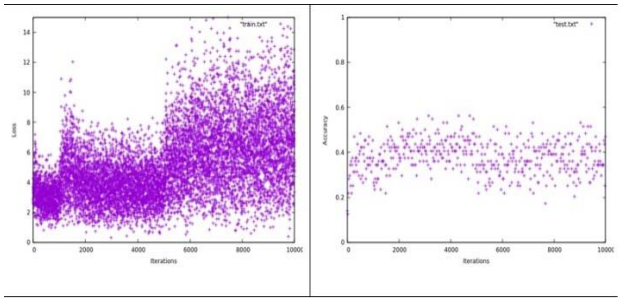

Fig 5. Lost and Accuracy Evolution during Training with the Original Model.

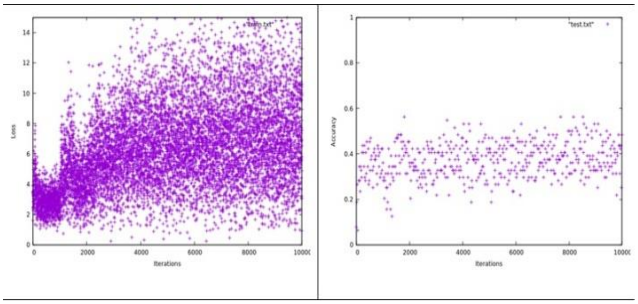

Fig 6. Lost and Accuracy Evolution during Training with the Proposed Model.

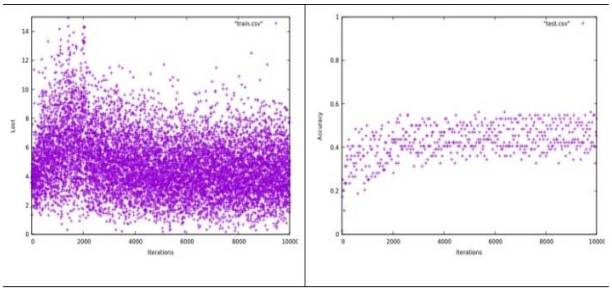

Fig 7. Effect of Large Margin on Training.

TABLE II. Results Obtained during the Test Process

Identity	Out put probability of the original model	Out put probability of the proposed model
SOP DEFFO	0.99	0.99
Angeline Jolie	0.958	0.959
Barack Obama	0.013	0.997
Beyonce Knowles	0.0003	0.2
Brad Pitt	0.044	0.05
Christina Ricci	0.0001	0.03
Georges Clooney	0.0003	0.001
Halle Berry	0.0019	0.17

A different observation can be made on figure 6. The loss in increasing and when closer to the end of the training it starts decreasing. For the accuracy, it increases gradually which is a good result.

Finally on Figure 7, the convergence of loss is more visible thanks to the large margin module. In addition, the evolution of accuracy is more perceptible which means the result is getting better.

After these observations during training let us see the effect on the output probabilities for each identity presented in table II

We see that our results are better than the one obtained using the original VGG model. With the original model, we observe a high output probability for only 2 identities and a very low one for the other but with our model we have high values for 3 identities and acceptable one for the rest.

B. Real Time Detection and Recognition

Figure 8 and Figure 9 present a real time detection and recognition by the proposed system. It can be seen how the face is first of all detected by the bounding box then recognized later on. The label of the detected person is written on top of the image. This shows that the system is really working.

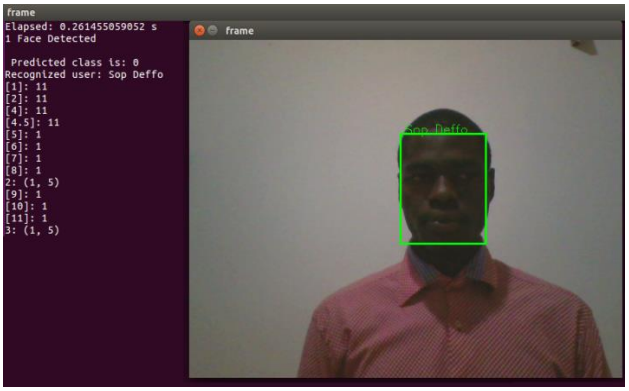

Fig 8. Detection and Recognition of Identity Sop Deffo.

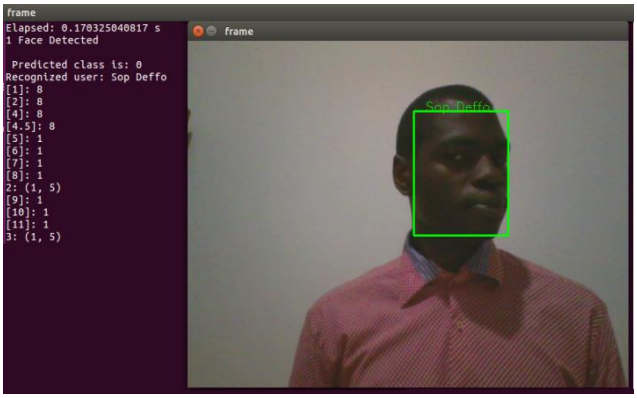

Fig 9. Detection and Recognition of identity Sop Deffo.

V. Conclusion

We presented in this paper a convolutional neural network system for face detection and recognition. In this system the detection is done by a multi cascade convolutional neural network system and recognition by deep proposed neural network architecture. The proposed model is based on the deep VGG neural network, a large margin inner product and a proposed CRelu function. The results obtained have proven to be better than the one obtained using the original model. For future work we intend to find mechanism to increase the size of the dataset in order to be able to recognize many persons.

References

[1] Bledsoe, W.W,"The model method in facial recognition," Panoramic Research Inc., Palo Alto, CA, Rep. PRI:15, August 1966

[2] Bledsoe, W.W, "Man machine facial recognition, Panoramic Research Inc., Palo Alto, CA, Rep. PRI:22, August 1996

[3] L. Sirovich and M. Kirby Low-Dimensional procedure for the characterization of human faces. Journal of optical society of America Vol 4 page 519 March 1987

[4] Mattew A Turk and Alex P. Pentland, Face recognition using Eigen faces, vision and modeling group, The media laboratory , Massachusetts Institute of Technology, 1991

[5] Jonathon Phillips, Patrick J. Rauss, and Sandor Z. De, FERET (Face Recognition Technology) Recognition Algorithm Development and Test Results, Army Research Laboratory (ARL), October 1996

[6] P. Jonathon Phillips, Patrick J. Flynn Todd Scruggs Kevin W. Bowyer, William Worek, Preliminary Overview of the Face Recognition Grand Challenge, IEEE Conference on Computer Vision and Pattern Recognition 2005

[7] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, Deep Face Recognition, Visual Geometry Group, Department of Engineering Science, University of Oxford

[8] West, J (2017) History of Face Recognition – Facial recognition software [online] FaceFirst Face Reconition facial recognition software available on https://www.facefirst.com/blog/brief-of-face recognition-software/ [Accessed 15 Oct. 2018

[9] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet large scale visual recognition challenge CoRR, abs/1409.0575, 2014

[10] Karen Simonyan and Andrew Zisserman, very deep convolutional network for large-scale image recognition, ICLR conference 2015

[11] Vinod Nair and Geoffrey Hinton, Rectified linear Units improve Reststricted Boltzmann Machines. ICML 2010

[12] Clevert D.A., Unterthiner T. &Hochreiter S. 2016. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)

[13] Maas, Andrew L, Hannun, Awni Y, and Ng, Andrew Y. Rectifier nonlinearities improve neural network acoustic models. In ICML, volume 30, 2013

[14] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. arXiv preprint arXiv:1502.01852, 2015.

[15] Wang, Naiyan, Li, Siyi, Gupta, Abhinav, and Yeung, Dit-Yan. Transferring rich feature hierarchies for robust visual tracking. arXiv preprint arXiv:1501.04587, 2015

[16] Wenling Shang, KihyukSohn, Diogo Almeida, Honglak Lee, Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units, arXiv:1603.05201

[17] Shifeng Zhang Xiangyu Zhu Zhen, Lei Hailin, Shi Xiaobo and Wang Stan Z. Li FaceBoxes: A CPU Real-time Face] Detector with High Accuracy, arXiv: 1708.05234v2 [cs.CV] 19 Aug 2017.

[18] Weiyang Liu, Yandong Wen, Zhiding Yu and Meng Yang Large-Margin Softmax Loss for Convolutional Neural Networks Proceedings of The 33rd International Conference on Machine Learning. 2016: 507-516.

[19] Jia, Yangqing and Shelhamer, Evan and Donahue, Jeff and Karayev, Sergey and Long, Jonathan and Girshick, Ross and Guadarrama, Sergio and Darrell, Trevor, Caffe: Convolutional Architecture for Fast Feature Embedding, arXiv preprint arXiv:1408.5093, 2014

[20] Becker, B. C. and Ortiz, E.G. "**Evaluating Open-Universe Face Identification on the Web**," In CVPR 2013, Analysis and Modeling of Faces and Gestures Workshop.