THE 12th INTERNATIONAL SYMPOSIUM ON ADVANCED TOPICS IN ELECTRICAL ENGINEERING
March 25-27, 2021
Bucharest, Romania

Convolutional Neural Network Hyperparameters Optimization for Facial Emotion Recognition

Adrian Vulpe-Grigoraşi[1], Ovidiu Grigore[1]
[1]Faculty of Electronics, Telecommunications and Information Technology, University Politehnica of Bucharest, Bucharest, Romania
adrian.vulpe@upb.ro, ovidiu.grigore@upb.ro

2021 12th International Symposium on Advanced Topics in Electrical Engineering (ATEE) | 978-1-6654-1878-2/20/$31.00 ©2021 IEEE | DOI: 10.1109/ATEE52255.2021.9425073

Abstract **This paper presents a method of optimizing the hyperparameters of a convolutional neural network in order to increase accuracy in the context of facial emotion recognition. The optimal hyperparameters of the network were determined by generating and training models based on Random Search algorithm applied on a search space defined by discrete values of hyperparameters. The best model resulted was trained and evaluated using FER2013 database, obtaining an accuracy of 72.16%.**

Keywords: **hyperparameter optimization, convolutional neural networks, facial emotion recognition, Random Search, FER2013**

I. INTRODUCTION

Facial emotions play an important role in human interaction based on non-verbal communication. In non-verbal communication performed daily, the main method of expression and the easiest to observe is through facial emotions. Although facial emotions are relatively fairly easy for a person to detect and classify, for a machine, regardless of its complexity, detecting and recognizing emotions is indeed a challenge.

Facial emotion recognition is a field of study where significant progress has been made in terms of recognizing basic emotions such as anger, happiness, sadness or fear in controlled environments, yet still offers challenges when it comes to naturalistic, in the wild, emotions. Being able to recognize facial emotions in naturalistic environments is still an unsolved problem.

In recent years, with a growing interest in machine learning and deep networks, convolutional neural networks (CNN) have proved to be a possible solution when it comes to the task of facial emotion recognition under naturalistic terms. Works described in [2] to [11], although different by methods, prove that CNN can be applied with a high degree of success for facial emotion recognition. The aim of this paper is to improve the performance of convolutional neural networks by optimizing hyperparameters and architecture in order to classify human facial images into categories of discrete emotions. We have considered the use of Random Search algorithm as an automated method capable of generating models with various architectures and different configurations of hyperparameters from a discrete space of possible solutions that can be applied in the imposed problem. Random Search has proven to be an algorithm capable of providing solutions for determining the optimal hyperparameters of a model, while requiring a low computational cost compared to other search methods[12]. The performances obtained following the application of the strategy proposed in the paper are analyzed in the light of the results obtained on FER2013 database and also following a comparative analysis of the resulting model accuracy and the accuracies obtained in previous work on the same database.

II. RELATED WORK ON FER2013 DATABASE

FER2013 database contains images with facial emotions and was developed by Ian Goodfellow et al. for a Kaggle competition on facial emotion recognition in 2013. It includes a set of 35887 images in 8-bit grayscale format measuring 48x48 pixels, with facial emotions, divided into 3 categories: 28709 training data, 3589 test data and 3589 validation data. All pictures in the database are labeled so that each picture falls into one of the seven main categories of facial emotions: anger, disgust, fear, happiness, sad, surprised and neutral. Regarding the distribution of emotions, the 35887 images are divided as follows: anger - 4953 images, disgust - 547 images, fear - 5121 images, happiness - 8989 images, sad - 6077 images, surprised - 4002 images, and neutral - 6198 images [1]. Fig 1 shows an example of images from the FER2013 database.

Fig 1. Example of images contained in FER2013 database

FER2013 is a diverse database, includes facial emotions from each age category, containing data with various levels of

ISBN: 978-1-6654-1878-2/21/$31.00 ©2021 IEEE

Authorized licensed use limited to: East Carolina University. Downloaded on June 19,2021 at 05:11:40 UTC from IEEE Xplore. Restrictions apply.

exposure, illumination and intensity of expressions [2]. It is also worth mentioning that the human observer had an accuracy of about 65±5% in correctly recognizing the facial emotions presented in FER2013[1]. This is due to the diversity of the database in terms of the situations in which emotions were captured but also the presence in the database of images that show a wide range of emotions that can be recognized, despite pre-existing labeling [2].

Due to the popularity of FER2013 and also the free access to it, in recent years various methods have been applied to create a model with high accuracy in recognizing facial emotions.

TABLE I
SUMMARY OF RESULTS ON FER2013

Model	Network Type	Accuracy%
Khanzada Amil et al. [3]	Ensemble CNN	75.8
Georgescu et al. [8]	Fusion CNN + Bag of visual words (BOVW)	75.42
Pramerdorfer et al. [2]	Ensemble CNN	75.2
Zhang et al.[2],[10],[11]	Multitask network	75.1
Kim et al. [10]	Ensemble CNN	73.73
Connie et al. [6]	Hybrid CNN–Scale Invariant Feature Transform (SIFT)	73.4
VGG (Visual Geometry Group CNN architecture) [2], [14]	CNN	72.7
ResNet [2]	Residual	72.4
Hua et al. [7]	Ensemble CNN	71.91
Inception [2], [13]	Deep CNN	71.6
Tang [1], [5]	CNN + SVM	71.16
ResNet152[9]	Residual	69.7
VGG-16 [9], [14]	CNN	68.2
Ionescu et al. [4]	BOVW	67.48

From the results presented in table 1, of interest for this paper are only the ones that used convolutional neural networks in their architecture and their hyperparameters. Table 2 presents the unique hyperparameters of the models in question. It should be noted that Pramerdorfer et al. [2], Georgescu et al. [8], VGG-16 [9] use in their proposed methods VGG models or VGG model variants and have the same hyperparameters as VGG and have been assimilated with VGG. Khanzada Amil et al. [3] use in their model VGG models or variants but their ensemble networks also make use of models with different hyperparameters and only those hyperparameters have been included.

TABLE II
HYPERPARAMETER VALUES

Model	Hyperparameters
Khanzada Amil et al. [3]	Convolutional layers: 4 Kernels: 32, 64 Kernel size: 3x3 Dropout: 0.2 Dense: 1024, 4096
Kim et al. [10]	Convolutional layers: 3 Kernels: 32, 64 Kernel size: 4x4, 5x5 Dense: 1024
Connie et al. [6]	Convolutional layers: 6 Kernels: 32, 64, 128 Kernel size: 3x3 Dropout: 0.1, 0.4. 0.5 Dense: 2048
VGG [2], [14]	Convolutional layers: 8, 16 Kernels: 32, 64, 128, 256, 512 Kernel size: 3x3, 5x5 Dropout: 0.5 Dense: 1000, 4096
Hua et al. [7]	Convolutional layers: 3, 4, 5 Kernels: 32, 64, 128 Kernel size: 3x3 Dense: 2048
Inception [2]	Convolutional layers: 22 Kernels: 16 to 384 Dropout 0.4

III. HYPERPARAMETER OPTIMIZATION

In the previous section we have seen the different hyperparameters used in Convolutional Neural Networks for facial emotion recognition. Based on those hyperparameters we define our discrete space of solution from which to generate models using Random Search. The space is bounded as followes:

1. Number of kernels in the first convolutional layer minimum 32 and maximum 256 with a step of 32.
2. Maximum number of convolutional layers beside the first layer: 4
3. Number of kernels in the convolutional layers: minimum 64 and maximum 512 with a step of 64.
4. Dropout in convolutional layers: minimum 0.1, maximum 0.4, with a step of 0.1.
5. Dropout in the fully-connected layer: minimum 0.1, maximum 0.4, with a step of 0.1.

For the solution bounded by the proposed search-space, the learning rate was set at 0.001, the optimization algorithm applied being Adam [15]. Batch size was set to 128, the size of the kernels in the convolutional layers was set to 3x3 and the activation function used was a Rectified Linear Unit (ReLU). For classification 2 fully-connected layers were

Authorized licensed use limited to: East Carolina University. Downloaded on June 19,2021 at 05:11:40 UTC from IEEE Xplore. Restrictions apply.

used, the first having 256 neurons and the second 7 neurons, one for each emotion in the database. In the initial setup the total number of models to be generated using Random Search in the proposed bounded space was limited to 500. After the generation phase, each model is submitted for training for 20 epochs in order to check its accuracy. Fig 2. displays the validation accuracy of the best model and of the next five best models generated.

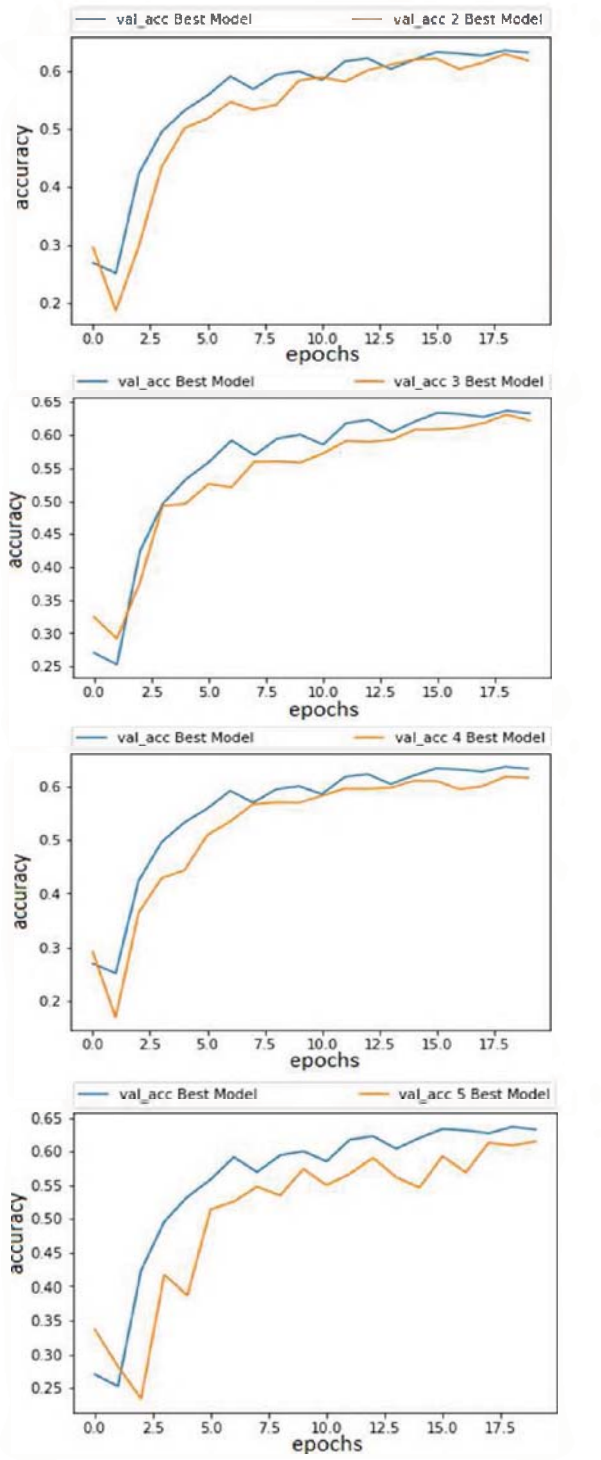

Fig 2. Validation accuracy of the best model and next best five models.

It should be noted that best solution out of 500 generated models exhibited a validation accuracy of 63.22% after 20 training epochs. The Random Search generation phase and training have been implemented in Google Colaboratory using a Tesla T4 GPU. The layout of the best model is presented in Fig. 3

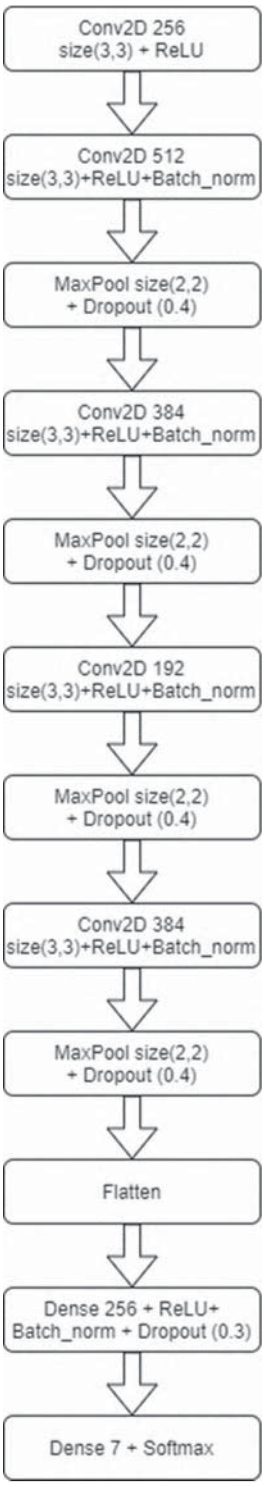

Fig 3. Model architecture.

Authorized licensed use limited to: East Carolina University. Downloaded on June 19,2021 at 05:11:40 UTC from IEEE Xplore. Restrictions apply.

The model has a total of 5173959 parameters and a size of 59MB. As a comparison VGG with a total number of parameters ranging from 133 to 144 million parameters and a size of 528 MB achieved 72.7% accuracy on FER2013 database [14]. Also it should be noted that Georgescu et al. [8], VGG-16 [9] and Khanzada Amil et al. [3] made use of VGG models and variants and as such the total number of parameters for their models is in range or higher than 133-144 million.

IV. Experimental Results

In order to test the overall performance, the proposed model was trained for 750 epochs on FER2013 database. For this final training the database was augmented in form of horizontal mirroring, ±10° rotations, ±10% image zooms, and ±10% horizontal and vertical shifting. In this stage, the learning rate was set the optimizer chosen was Adam with a learning rate of 0.001 and batch size was set to 128.

Following the training, the model obtained an accuracy of 69.96% with 1.08 loss on the validation data and 72.16% accuracy with 0.97 loss on the test data from FER2013. It should be noted that the test data was not used in the training or selection stages.

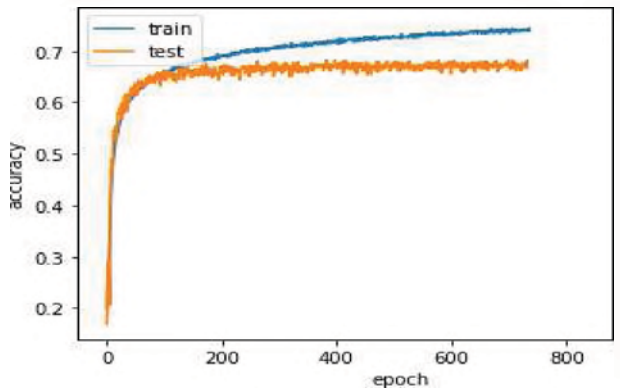

Fig 4. Train and test accuracy plot for the proposed model.

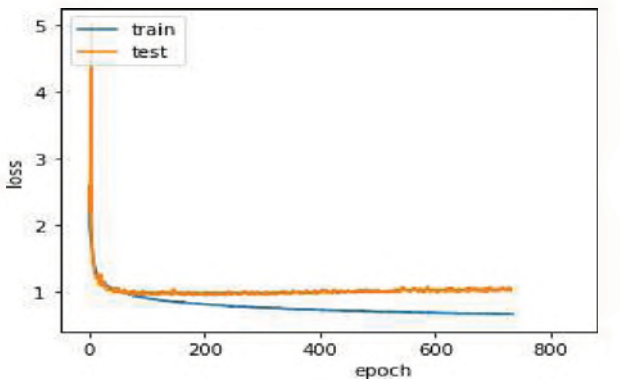

Fig 5. Train and test loss plot for the proposed model.

Based on these results, Tables III and IV show a performance comparison between the proposed model and the models discussed in this paper.

TABLE III
RESULTS AND RANKING OF THE PROPOSED MODEL

Model	Network Type	Accuracy%
Khanzada Amil et al. [3]	Ensemble CNN	75.8
Georgescu et al. [8]	Fusion CNN + BOVW	75.42
Pramerdorfer et al. [2]	Ensemble CNN	75.2
Zhang et al.[2],[10],[11]	Multitask network	75.1
Kim et al. [10]	Ensemble CNN	73.73
Connie et al. [6]	Hybrid CNN–SIFT	73.4
VGG [2]	CNN	72.7
ResNet [2]	Residual	72.4
Proposed model	CNN – Hyperparameter optimization	72.16
Hua et al. [7]	Ensemble CNN	71.91
Inception [2], [13]	Deep CNN	71.6
Tang [1], [5]	CNN + SVM	71.16
ResNet152[9]	Residual	69.7
VGG-16 [9]	CNN	68.2
Ionescu et al. [4]	BOVW	67.48

TABLE IV
RESULTS AND RANKING OF THE PROPOSED MODEL CNN ONLY

Model	Accuracy%
VGG [2]	72.7
Proposed model	72.16
Fine-tuned VGG in fusion CNN [8]	72.11
Best individual model in ensemble [10]	71.86
Inception [2]	71.6
Tang [1], [5]	71.16
CNN only model [6]	70.8
VGG-16 [9]	68.2
Best individual model in ensemble [7]	68.18

From both tables it can be seen that the difference in accuracy between the proposed model and VGG is of 0.54%, VGG having 130-144 million parameters while the proposed model has only 5173959 parameters. Also, the difference in accuracy between the top two models: Khanzada Amil et al. [3] and Georgescu et al. [8], and the proposed model is of 3.26 - 3.64%. It should be noted that both solutions make use of more parameters than the proposed model and also their network type is different and more complex than CNN network used in our model.

Authorized licensed use limited to: East Carolina University. Downloaded on June 19,2021 at 05:11:40 UTC from IEEE Xplore. Restrictions apply.

Furthermore, the proposed model was able to predict the test labels in 11.13 seconds for all 3589 test images. The reported measurement has been obtained in Google Colaboratory on a Tesla T4 GPU.

V. CONCLUSIONS

The paper aimed to present a method to increase the accuracy of a convolutional neural network model by optimizing hyperparameters and its architecture with the use of Random Search algorithm as an automated method capable of generating models with various architectures and different configurations of hyperparameters from a discrete space of possible solutions. The experimental results have shown that by using this method a compact model based only on CNN with an accuracy of 72.16% can be obtained. The superficial hyperparameter optimization aimed to show that an efficient solution can be achieved in a search space in which previous results are considered to be local minima.

The result is a satisfactory one considering the constraints applied in terms of optimization, this being done on a small number of hyperparameters. Also, the optimization of the model architecture was done only on the convolutive layers, the classification layers not being involved in this process. At the same time, it should be mentioned that the result was obtained by limiting the search space for possible solutions to only 500 elements.

Further research will consider the increase of the solution space enabling the Random Search algorithm to discover new architectures and combinations of hyperparameters in an attempt to achieve models with accuracies that exceed the presented model.

REFERENCES

[1] Goodfellow, Ian J et al. "Challenges in representation learning: a report on three machine learning contests." Neural networks: the official journal of the International Neural Network Society vol. 64 (2015): 59-63. doi:10.1016/j.neunet.2014.09.005

[2] C. Pramerdorfer and M. Kampel, "Facial expression recognition using convolutional neural networks: state of the art," arXiv preprint arXiv:1612.02903, 2016

[3] Khanzada, Amil et al. "Facial Expression Recognition with Deep Learning." ArXiv abs/2004.11823 (2020).

[4] R. T. Ionescu, M. Popescu, and C. Grozea. Local Learning to Improve Bag of Visual Words Model for Facial Expression Recognition. Proceedings of ICML Workshop on Challenges in Representation Learning, 2013

[5] Y. Tang. Deep Learning using Linear Support Vector Machines. Proceedings of ICML Workshop on Challenges in Representation Learning, 2013

[6] T. Connie, M. Al-Shabi, W. P. Cheah, and M. Goh. Facial Expression Recognition Using a Hybrid CNN–SIFT Aggregator. Proceedings of MIWAI, volume 10607, pp. 139– 149. Springer, 2017.

[7] W. Hua, F. Dai, L. Huang, J. Xiong, and G. Gui. HERO: Human Emotions Recognition for Realizing Intelligent Internet of Things. IEEE Access, 7:24321–24332, 2019.

[8] M. Georgescu, R. T. Ionescu and M. Popescu, "Local Learning With Deep and Handcrafted Features for Facial Expression Recognition," in IEEE Access, vol. 7, pp. 64827-64836, 2019, doi: 10.1109/ACCESS.2019.2917266.

[9] A. O. Vorontsov, A. N. Averkin, "Comparison of different convolution neural network architectures for the solution of the problem of emotion recognition by facial expression", Proceedings of the 8th International Conference 'Distributed Computing and Grid-technologies in Science and Education', Dubna, Moscow region, Russian, 10-14 September 2018, pp. 342-345.

[10] B.-K. Kim, S.-Y. Dong, J. Roh, G. Kim, and S.-Y. Lee, "Fusing aligned and non-aligned face information for automatic affect recognition in the wild: A deep learning approach," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 48–57.

[11] S. Li and W. Deng, "Deep Facial Expression Recognition: A Survey," in IEEE Transactions on Affective Computing, doi: 10.1109/TAFFC.2020.2981446.

[12] Bergstra, J. and Yoshua Bengio. "Random Search for Hyper-Parameter Optimization." The Journal of Machine Learning Research 13 (2012): 281-305.

[13] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.

[14] Simonyan, K. and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." CoRR abs/1409.1556 (2015)

[15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". CoRR abs/1412.6980 (2014)

Authorized licensed use limited to: East Carolina University. Downloaded on June 19,2021 at 05:11:40 UTC from IEEE Xplore. Restrictions apply.