Training the Fer2013 Dataset with Keras Tuner

Benyoussef Abdellaoui[a], Aniss Moumen[b], Younes El Bouzekri El Idrissi[c] and Ahmed Remaida[d]

Laboratory of Engineering Sciences, National School of Applied Sciences, Ibn Tofaïl University, Kenitra, Morocco

Keywords: Keras tuner, fer2013, CNN.

Abstract: The emotional state of humans plays an essential role in the communication between humans and human-machine. The construction of a model capable of detecting emotions during a scene requires adjusting the model parameters. However, this adjustment is not easy. This article uses the Keras tuner module to find the hyperparameters during training the fer2013 dataset with the CNN algorithm. The use of the Keras tuner reduces the time and optimizes the model with the best parameters.

1 INTRODUCTION

The emotional state is inferred from visual expression, auditory expression, and physiological representation. The use of these and other techniques plays a significant role in the active life of humans and human-machine interactions. Many machine learning algorithms are used with varying degrees of accuracy, although areas require very high precision, such as health and autonomous driving. For this purpose, the algorithms used are trained on both private and public databases. In this work, we will use a very well-known database in emotional state detection from facial expressions, fer2013 ('Facial Expression Dataset Image Folders (Fer2013)' n.d.). The dataset contains photos of faces showing various types of emotions; the data is divided into training (80%), test (10%), and validation (10%) images, and represents seven emotional states: Anger, Disgust, Fear, Happy, Sad, Surprise, and Neutral. The following figure shows some photos from this dataset.

Figure 1: Some examples of images taken from the fer2013 dataset.

When building models, there are problems related to determining hyperparameters that will produce a better model that gives better accuracy. In this work, we will try to introduce a module called Keras-tuner (O'Malley et al. 2019), which has a role to help to determine the number ofhidden layers, the number of neurons in each layer, and the learning rate value. This paper presents the methodology used, the results found and ends with the discussion and conclusion.

2 RELATED WORK

Models that use facial emotion recognition (FER) with different algorithms have shown various potential in accuracy and performance computation. Researchers working in this area have experimented with several algorithms and databases. For most techniques, the big challenge is to adjust the hyperparameters and find a model with better experimental optimization. Several types of research have been done in this field. In this work, we focus on the following dataset fer2013. Based on this article

[a] https://orcid.org/0000-0002-5950-0187
[b] https://orcid.org/0000-0001-5330-0136
[c] https://orcid.org/0000-0003-4018-437X
[d] https://orcid.org/0000-0002-2981-8936

Abdellaoui, B., Moumen, A., El Bouzekri El Idrissi, Y. and Remaida, A.
Training the Fer2013 Dataset with Keras Tuner.
DOI: 10.5220/0010735600003101
In *Proceedings of the 2nd International Conference on Big Data, Modelling and Machine Learning (BML 2021)*, pages 409-412
ISBN: 978-989-758-559-3

(Khaireddin and Chen, n.d.)and others. Between the algorithms applied on this dataset, we find CNN, GoogleNet, VGGand SVM, Inception layer, ARM(ResNet-18), VGG, etc. We quote as an example the following works treating the performance on this dataset: Based on the convolutional neural network (CNN) technique, Kuang Liu et al. (Liu, Zhang, and Pan 2016) trained and evaluated their model to perform classification and recognition of images according to the emotional state. They claim to achieve an accuracy of 62.44%. P. Giannopoulos et al. (Giannopoulos, Perikos, and Hatzilygeroudis 2018) present deep learning approaches for facial emotion recognition using GoogleNet and AlexNet, and they achieve an accuracy of 65.2% by using GoogleNet. Mariana-Iuliana Georgescu et al. (Georgescu, Ionescu, and Popescu 2019). They tested several CNN architectures and pre-trained models, and they used a local learning framework to predict the class of the test images present in the dataset.

They obtained an accuracy of 66.51% using the VGG algorithm. Ali Mollahosseini et al. (Mollahosseini, Chan, and Mahoor 2016) try to resolve the FER problem; they proposed a CNN architecture followed by inception layers. This study is realized on seven public datasets. They have achieved an average accuracy of 66.4% for fer2013. Radu Tudor Ionescu et al.(Ionescu and Grozea 2013) proposed a new method for the classification of facial expressions from low-resolution images using the bag of words representation with an accuracy of 67.484%. Shervin Minaee et al. (Minaee, Minaei, and Abdolrashidi 2021) proposed an attentional convolutional network capable of focusing on essential parts of the human face, and achieving a significant performance improvement compared to previous models tested on several datasets, notably Fer2013, CK+, FERG, and JAFFE. They have reached an average accuracy of 70.02% for fer2013. Yichuan Tang (Tang 2013) used the CNN and replaced the softmax layer with a linear support vector machine (SVM). They have accomplished an average accuracy of 71.2% for fer2013. Jiawei Shi et al. (Shi, Zhu, and Liang 2021) addressed the problem caused by convolution padding, which causes a degradation of the feature map. The output feature map (albino feature) weakens the representation of the facial expression. They proposed an alternative to the pooling structure layer, called ARM (Agent Representation Module). ARM combined with ResNet-18 boosted the performance of FER(71.38). Christopher Pramerdorfer et al. (Pramerdorfer and Kampel 2016) achieve a FER2013 test accuracy of 75.2%. Several datasets represent images of people with different emotions. Some of these databases are public like MMI, DISFA, CK+, MultiPIE, SFEW, FERA, and FER2013 (Kabir et al. 2020) and others are private. We review some state-of-the-art examples of the accuracy obtained for FER2013 using different algorithms. Given the difficulties researchers face in finding hyperparameters. We are conducting experiments on our FER2013 dataset with models found on Kaggle. To tackle this problem, we used the Keras tuner module, which is used to simplify this task. In the next chapter, we present our results and conclude with a discussion.

3 EXPERIMENTS AND RESULTS

We have reviewed the state of the art of research and competitions that have used the fer2013 dataset. We used the Kaggle platform on which we reused a public code; this model trained on 200 epochs has given the results as indicated in the following table:

Table 1: Metrics obtained without Kuras tuner

Metric	Value
Loss	0.1223
Accuracy	0.9436
Mean_squared_error	0.0094
Mean_absolute_error	0.0187
Val_loss	2.2965
Val_accuracy	0.8176
Time of running	0:02:59.98

The following figure shows training loss vs validation loss.

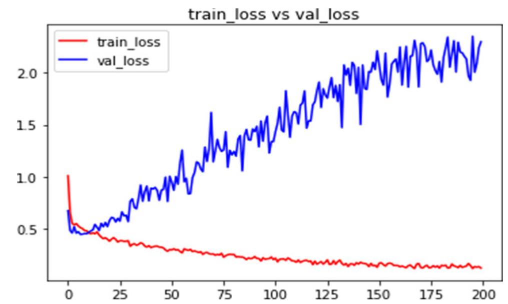

Figure 2: Training loss vs validation loss.

The following figure shows Training accuracy vs validation accuracy.

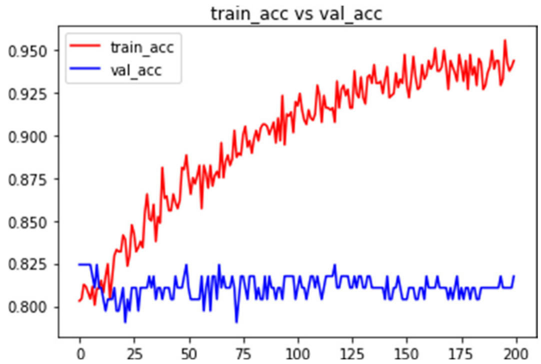

Figure 3: Training accuracy vs validation accuracy.

Let’s see the results of applying the Keras tuner model optimizer for this fer2013 dataset. In the following table, wesummarize the structure of our network.

Table 2: The structure of our network after obtaining parameters by applying the Keras tuner module.

Layer (type)	Output Shape	Parameters
Conv2d (Conv2D)	(None, 46, 46, 32)	320
Conv2d_1 (Conv2D)	(None, 44, 44,32)	9248
Max_pooling2d (MaxPooling2D)	(None, 22, 22,32)	0
Dropout (Dropout)	(None, 22, 22,32)	0
Conv2d_2 (Conv2D)	(None, 20, 20,32)	9248
Conv2d_3 (Conv2D)	(None, 18, 18,32)	9248
Max_pooling2d_1 (MaxPooling2)	(None, 9, 9, 32)	0
Dropout_1(Dropout)	(None, 9, 9, 32)	0
flatten (Flatten)	(None, 2592)	0
dense (Dense)	(None, 448)	1161664
dropout_2 (Dropout)	(None, 448)	0

We got a total parameter is 1192871, trainable parameters are 1192871, and non-trainable parameters are null. With this structure, the best hyperparameters obtained are shown below for 50 epochs.

Table 3: The best hyperparameters obtained.

Hyperparameters	Value
Number of filters	32
Dropout_1	0.05
Dropout_2	0.4
Number of layers	1
Units_0	448
Dense_activation	tanh
Dropout_0	0.0
Learning_rate	0.0050647069 02828907
Units_1	320

The next table shows the hyperparameters for the five best trials.

Table 4: The hyperparameters for the five best trials.

Hyper	1	2	3	4	5
Input units	128	192	64	128	224
N layers	2	3	2	4	1
Conv_0_un	160	32	128	128	224
conv_1_uni	160	64	192	32	128
conv_2_un	160	96	224	32	192
conv_3_un	224	32	224	32	96
Score	0.51	0.51	0.5	0.5	0.47

The training accuracy is about 0.95%, and the best validation accuracy obtained is 0.53%.

The following figure shows the loss and the accuracy plot.

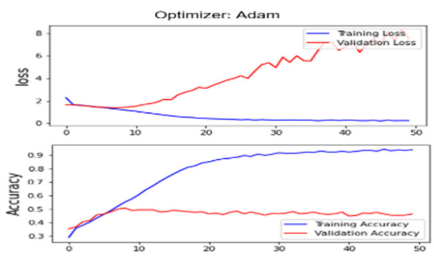

Figure 4: The loss and accuracy plot.

4 DISCUSSION

On the other hand, when we use the Keras tuner, we have two main tasks: searching for hyperparameters, building the network model with the best hyperparameters, and running it on the data. In our experience, we have chosen 50 epochs. As seen in the previous chapter, our model built without the Keras tuner module gives an accuracy of 94.36% in the training phase and 81.76% in the validation phase. It provided an accuracy of 93.91% in the learning phase and an accuracy of 46.34 % in the testing phase. We note that the accuracies obtained are different; the network gives the best accuracy obtained in the training step without Keras tuner; this value of accuracy is close to that obtained with Keras tuner. The same thing about precision in the testing phase, the model without Keras tuner surpasses the values cited in the literature seen in this work and the model built with Keras tuner.

5 CONCLUSION

In this paper, we have described the dataset fer2013. We present some works that have used this dataset and briefly describe the results found. Our article was to introduce the Keras tuner module that allows the automation of hyperparameters of the models. We have presented the results. We found out that further tuning would give better results. We intend to improve these settings to use them on other datasets of the domain or different domains as we can also think of using them in other problems like a regression to improve the achievements.

REFERENCES

'Facial Expression Dataset Image Folders (Fer2013)'. n.d. Accessed 27 June 2021. https://kaggle.com/astraszab/facial-expression-dataset-image-folders-fer2013.

Georgescu, Mariana-Iuliana, Radu Tudor Ionescu, and Marius Popescu. 2019. 'Local Learning with Deep and Handcrafted Features for Facial Expression Recognition'. IEEE Access 7: 64827–36. https://doi.org/10.1109/ACCESS.2019.2917266.

Giannopoulos, Panagiotis, Isidoros Perikos, and Ioannis Hatzilygeroudis. 2018. 'Deep Learning Approaches for Facial Emotion Recognition: A Case Study on FER-2013'. In Advances in Hybridization of Intelligent Methods: Models, Systems and Applications, edited by Ioannis Hatzilygeroudis and Vasile Palade, 1–16. Smart Innovation, Systems and Technologies. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-66790-4_1.

Ionescu, Radu Tudor, and C. Grozea. 2013. 'Local Learning to Improve Bag of Visual Words Model for Facial Expression Recognition'. 2013. https://www.semanticscholar.org/paper/Local-Learning-to-Improve-Bag-of-Visual-Words-Model-Ionescu-Grozea/97088cbbac03bf8e9a209403f097bc9af46a4ebb.

Kabir, Md. Mohsin, Farisa Benta Safir, Saifullah Shahen, Jannatul Maua, Iffat Ara Binte Awlad, and M. F. Mridha. 2020. 'Human Abnormality Classification Using Combined CNN-RNN Approach'. In 2020 IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET), 204–8. Charlotte, NC, USA: IEEE. https://doi.org/10.1109/HONET50430.2020.9322814.

Khaireddin, Yousif, and Zhuofa Chen. n.d. 'Facial Emotion Recognition: State of the Art Performance on FER2013', 9.

Liu, Kuang, Mingmin Zhang, and Zhigeng Pan. 2016. 'Facial Expression Recognition with CNN Ensemble'. In 2016 International Conference on Cyberworlds (CW), 163–66. https://doi.org/10.1109/CW.2016.34.

Minaee, Shervin, Mehdi Minaei, and Amirali Abdolrashidi. 2021. 'Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network'. Sensors 21 (9): 3046. https://doi.org/10.3390/s21093046.

Mollahosseini, Ali, David Chan, and Mohammad H. Mahoor. 2016. 'Going Deeper in Facial Expression Recognition Using Deep Neural Networks'. In 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), 1–10. https://doi.org/10.1109/WACV.2016.7477450.

O'Malley, Tom, Elie Bursztein, James Long, François Chollet, Haifeng Jin, and Luca Invernizzi. 2019. 'Keras Tuner'. Retrieved May 21: 2020.

Pramerdorfer, Christopher, and Martin Kampel. 2016. 'Facial Expression Recognition Using Convolutional Neural Networks: State of the Art'. ArXiv:1612.02903 [Cs], December. http://arxiv.org/abs/1612.02903.

Shi, Jiawei, Songhao Zhu, and Zhiwei Liang. 2021. 'Learning to Amend Facial Expression Representation via De-Albino and Affinity'. ArXiv:2103.10189 [Cs], June. http://arxiv.org/abs/2103.10189.

Tang, Yichuan. 2013. 'Deep Learning Using Linear Support Vector Machines', June. https://arxiv.org/abs/1306.0239v4.