Real-Time CNN Based Facial Emotion Recognition Model for a Mobile Serious Game

Carolain Anto-Chavez, Richard Maguiña-Bernuy and Willy Ugarte[a]

Universidad Peruana de Ciencias Aplicadas, Lima, Peru

Keywords: Facial, Emotion, Expression, Recognition, Machine Learning, Real-Time, Mobile, FER.

Abstract: Every year, the increase in human-computer interaction is noticeable. This brings with it the evolution of computer vision to improve this interaction to make it more efficient and effective. This paper presents a CNN-based emotion face recognition model capable to be executed on mobile devices, in real time and with high accuracy. Different models implemented in other research are usually of large sizes, and although they obtained high accuracy, they fail to make predictions in an optimal time, which prevents a fluid interaction with the computer. To improve these, we have implemented a lightweight CNN model trained with the FER-2013 dataset to obtain the prediction of seven basic emotions. Experimentation shows that our model achieves an accuracy of 66.52% in validation, can be stored in a 13.23MB file and achieves an average processing time of 14.39ms and 16.06ms, on a tablet and a phone, respectively.

1 INTRODUCTION

Computer vision has been evolving in recent years, and it brings with it a lot of beneficial uses in human-computer interactions (Zarif et al., 2021). Automatic facial emotion recognition (FER) is one of the fields in computer vision that is growing and it's being applied in the gaming industry, medical care, education, security, and so on.

For example, nowadays, cameras are able to detect a smile on the frame and automatically take a picture without pressing any button (Zhou et al., 2021). Various methods are used to recognize the emotions expressed by people in photos and videos. However, some of these are not capable of running in real time, which prevents a fluent human-computer interaction. Also, others are often very large models, which complicates their integration into devices that have limited disk space.

In 1971, Ekman (Ekman and Friesen, 1971) defined the seven basic emotions: angry, disgust, fear, happy, neutral, sad and surprise. Since then, research has focused on the detection of these emotions automatically by computer (Zhou et al., 2021)(Minaee et al., 2021). Following this path, the goal of our work is to implement a lightweight emotion facial recognition model, capable of being executed in real time and that can be integrated into a serious game for mobile devices without internet connection.

Currently, advanced image classification methods are based on Convolutional Neural Networks (CNN). For example, some models are built based on pre-constructed architectures such as MobileNet (Nan et al., 2022) and EfficientNet (Wahab et al., 2021). Although these are models that achieve high accuracy in some tasks, due to the depth of their network, the processing required for the images is not optimal for some devices, especially mobile devices. Similarly, the use of ResNet-50, VGG-19 and Inception-V3 (Ullah et al., 2022) are applied for this task. However, the high density of these convolutional networks requires at least 500MB of disk space, which is not at all convenient, especially for its integration in a video game. Other approaches consist of extracting geometric features from the face (Murugappan and Mutawa, 2021) or creating a graph based on face landmarks (Farkhod et al., 2022) before sending the obtained information to a classification model , which increments the time needed for the emotion classification proccess, due to the image pre-processing work.

The key components of our resarch are the use of Tensorflow to implement the architecture of our designed CNN model. Then, we use the FER-2013[1] dataset to train the model and the Tensorflow Lite li-

[a] https://orcid.org/0000-0002-7510-618X

[1] https://www.kaggle.com/datasets/msambare/fer2013

Anto-Chavez, C., Maguiña-Bernuy, R. and Ugarte, W.
Real-Time CNN Based Facial Emotion Recognition Model for a Mobile Serious Game.
DOI: 10.5220/0012683800003699

In *Proceedings of the 10th International Conference on Information and Communication Technologies for Ageing Well and e-Health (ICT4AWE 2024)*, pages 84-92
ISBN: 978-989-758-700-9; ISSN: 2184-4984

brary to reduce its disk weight. Furthermore, we use the Unity Engine to develop a serious game where we integrate the FER model for an emotion imitation activity.

The main contributions of this work are list as follows:

- We are implementing a lightweight CNN model that achieves great accuracy for facial emotion classification.
- We integrate the model into a serious game developed in the Unity Engine aimed for children with autism in a emotion imitation activity.
- We validate the accuracy and low process cost of the implemented model during real-time tests in different mobile devices.

The rest of this paper is distributed as follows: In Section 2, we review related work of different implementations of FER models and their integration in some systems. Then, in Section 3, we mention important concepts for this work and describe the details of our main contributions. The setup, experiments performed and results of this work are presented in Section 4. Finally, in Section 5, we show the conclusions and discuss about recomendations for future work.

2 RELATED WORKS

Facial Emotion Recognition (FER) is a topic that has gained more relevance in recent years. It is used in areas of great importance such as education, health, security, among others. A variety of algorithms and artificial intelligence techniques have been proposed to improve the results of these models. In this section, we present related works that had achieved very good results in the practice of this field, as well as examples of its use in serious games and teaching methodologies for children with autism.

In (Zhou et al., 2021), the authors proposed a lightweight CNN for real-time facial emotion detection. Instead of using OpenCV, their work uses multitask cascaded convolutional networks (MTCNN) for face detection. The obtained face image is sent for classification to their proposed model, which was a CNN based on the Xception architecture. The authors use Global Average Pooling to remove the fully connected layer at the end. The model was tested on the FER-2013 dataset and achived 67% of accuracy. In the same way, we build our own CNN model and train it using the FER-2013 dataset to achieve similar accuracy, but opted to use OpenCV for face detection.

The authors in (Murugappan and Mutawa, 2021) proposed an emotion expression classification based of geometric features extracted from the face. Their method consisted of forming five triangles based on eight points marked on key parts of the face. From there, the inscribed circle area of the triangles are extracted as features to categorize emotions using machine learning methods. The Random Forest classifier got the best results with a 98.17% accuracy during training. Instead of extracting features and then classifying them, we use a CNN to send an image directly to the classification process.

In (Farkhod et al., 2022), the authors opted to use a graph-based method for emotion recognition. Face detection was done using Haar-Cascade, for then creating landmarks through a media-pipe face mesh model, and use those key points to train an implemented graph neural network (GNN). Using the FER-2013 dataset, the proposed model achieved an accuracy of 91.2%. The authors use transfer learning techniques to make the model able to recognize emotions on masked faces, which is also able to work in real time. Similarly, we use the FER-2013 dataset to train our model and aim for it to process images in real time to be able to integrate it in a videogame.

The authors in (Vulpe-Grigorasi and Grigore, 2021) presenten a method to optimize the hyperparameters of CNNs to increase accuracy for facial emotion recognition. In their work, they described the maximum number of convolutional layers, the number of kernels to apply in each convolutional layer and the recommended dropout in convolutional and fully-connected layer. A proposed model trained with the FER-2013 dataset obtained an accuracy of 72.16%. Our work follows some of their recommendations to build an optimized CNN to accomplish similar accuracy in the same dataset.

To develop a Serious Game for people with autism, the authors in (Dantas and do Nascimento, 2022) implemented their own FER model. To achieve this, they used the Adaboost algorithm to determine the regions of interest (ROI) from a face image. Dlib library was used to draw facial keypoints for each ROI. The histogram of oriented gradient (HOG) method was then processed to the ROIs and their keypoints. The result HOG image is sent for final emotion classification with a CNN. 98.84% accuracy was achieved using the CK+, FER-2013, RAF-DB and MMI Facial Expression datasets. We also use CNN for emotion classification, but avoid doing much image pre-processing to speed up the results.

The authors in (Garcia-Garcia et al., 2022) uses the capacities of facial emotion recognition to develop a serious game about teaching children with autism how to identify and express emotions. The proposed videogame is aim for children between the ages of 6 and 12 with emotion disability. The interaction with the implemented system is based of tangible user interfaces (TUIs). Three sections were developed for emotional training, achieving positive results. The main activity was the imitation of emotions presented on the screen, which uses the FER model. This solution inspired us to create a serious game in a form of a mobile application for children with autism and emotion disability, although our target audience will be children between the ages of 5 and 11 in Peru.

In (Pavez et al., 2023), the authors planned to develop a tool called ”Emo-mirror” to help children with autism recognize and understand the emotions of others. It uses augmented reality and facial recognition as part of the inmersive learning experience. The children use the mirror to choose an emotion and try to imitate it based on an image displayed. A FER model was implemented by the authors which uses the Viola-Jones algorithm for face detection and a CNN based on ResNet50 for emotion classification. The model and the intelligent mirror achieved great results during the experimental phase. Our FER model also uses CNN for emotion classification and we integrate it in an activity for the children to express emotions shown on the screen. However, it will not be the only excersise in the serious game, as we create a complete set of activities for emotion education.

3 CONTRIBUTION

In this section, we describe some preliminary concepts, our methods, construction stages and contribution of this project.

3.1 Preliminary Concepts

Now, we will present the main concepts used in our work. We aim to teach children with autism to recognize and express emotions through a serious game and facial emotion recognition.

Definition 1 (Affective Computing (AC)(Garcia-Garcia et al., 2022))**.** *It is any form of computing that is related to emotions. One of the most popular lines of work in this subject is automatic emotion detection.*

Nowadays, AC and human-computer interaction are being combined to create applications or systems that can detect how the user is feeling and make a decision based on that.

Definition 2 (Facial Emotion Recognition (FER)(Dantas and do Nascimento, 2022))**.** *Is a computational technique based in computer vision and image processing to detect a person's emotion from and image or in real time from a video camera.*

Example 1. *In Figure 1 we can see some examples of face detection and emotion classification done by a computer.*

Figure 1: Examples of Facial Emotion Classification (Zhou et al., 2021).

Definition 3 (Autism Spectrum Disorder (ASD) (Garcia-Garcia et al., 2022))**.** *It is a neurodevelopmental disorder characterized by deficits in social interaction and communication in different contexts.*

Emotional disability are also inherit in people with ASD, with the severity of it varying depending on the type of ASD a person has. It is known that therapy helps people with autism improve these skills.

Definition 4 (Facial Expressions of People with Autism (Dantas and do Nascimento, 2022))**.** *Children with ASD tend to be distracted, and the visual inattention compromises their social activities during the early learning age. For this reason, express and detect emotions is hard for them.*

New software is being develop to contribute in the skills training of children with ASD. Serious Games are one of these tools and are defined as follows.

Definition 5 (Serious Game (SG) (Dantas and do Nascimento, 2022))**.** *These are games with the objective to teach and develop skills. They combine common game characteristics such as fun and entertainment, but are primaly for educational purposes.*

Example 2. *We can see a Serious Game that teaches emotions in Figure 2.*

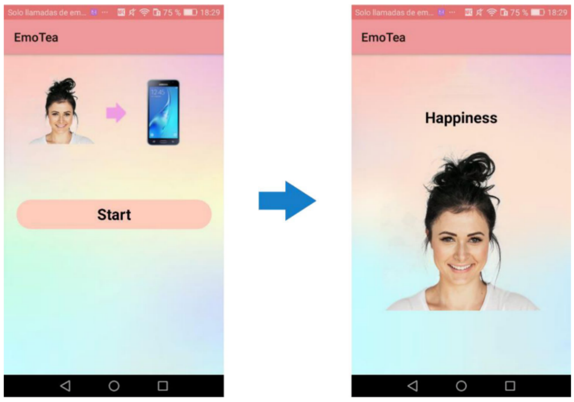


Figure 2: EmoTea - A Serious Game about emotions (Garcia-Garcia et al., 2022).

3.2 Method

In this section, the main contributions proposed will be presented.

3.2.1 Facial Emotion Recognition Model

The first contribution of this research is the application of Affective Computing, specifically in the construction of a FER model, able to recieve a frame captured by a camera video, process it and return the prediction of the emotion in real time.

Our goal is to integrate the model into a mobile Serious Game, and to achieve this is necessary to come up with a lightweight file that can be load inside the application. For this end, the TensorFlow library is used to create and train the emotion classificator. Afterwards, the model is converted into a TensorFlow Lite file, which helps us save a lot of disk space and is still capable of predicting emotions accurately in real time.

A Convolutional Neural Network (CNN) is used for this work. The designed architecture for this model can be seen in Figure 3. For the feature extraction of the model we use four phases of convolutions where 3x3 Kernels are applied to the input image. Max Pooling, Batch Normalization and Dropout operations are used between the convolution processes and for the activation function, RelU is selected for this work. After the last Max Pooling operation, we added a Flatten layer to start the classification part of the model, where a Dense layers is added before the final seven neurons layer which uses the Softmax activation function for emotion detection.

The dataset selected for this project was FER-2013, which can be found in the Kaggle repository. It consists of 28,709 face images for training and 7,178 in the test set. The dataset is organized in seven directories representing the basic emotions: angry, disgust, fear, happy, neutral, sad and surprise. We opted to separate 20% of the train set for validation purposes during the model training, and use the test set for the evaluation. Also, an augmentation process was applied to reduce the validation loss and avoid overfitting.

3.2.2 Serious Game

The second contribution consists of the development of a SG which aims to support the emotional learning of children with autism. This game will integrate the FER model as one of his core characteristics to create more dynamic activities.

The SG will be developed using the Unity Engine because of its capacity and ease of creating 2D videogames for mobiles. It will also make the application scalable, as the engine makes it very easy to transition the development for computers and also consoles.

It will feature three main activities, including a section where the children can learn about the seven basic emotions. The first one, will help the children build their expression recognition skills by selecting the correct emotion based of an image shown on the screen. The second exercise will also improve the recognition skills, with the difference that we will present, through a sentence, contexts in which a person expresses a feeling and the children have to select the emotion that best suits the situation. The final activity, and the one that uses the FER model, is the emotion imitation. Here, the children will see a picture of a person expressing an emotion and they have to imitate it as best as they can to improve their emotion expression skills. This section uses the mobile camera to capture the face of the kid that is using the application and it will be sent to the loaded classification model, which returns the emotion expressed at that moment. The diagram in Figure 4 shows the sections the SG will present and how it integrates the FER model.

The last feature of the SG will be the generation of reports at the end of each activity session. This will help the parents or specialists to follow the kid's progress in their development of the emotional ability. For the purpose of this work, a prototype capable of integrating the proposed FER model and being executed on mobiles has been developed.

4 EXPERIMENTS

In this section, we discuss about the experiments done for this project, as well as the setup used and results obtained during this process.

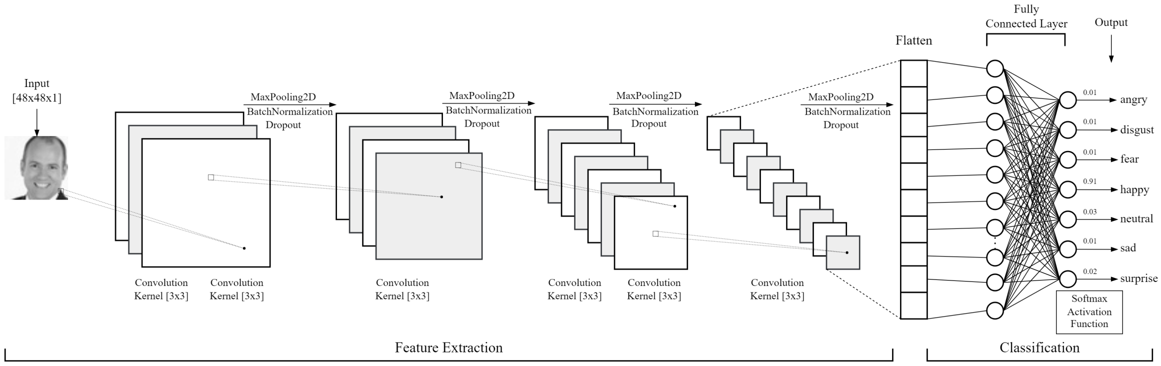


Figure 3: CNN Architecture.

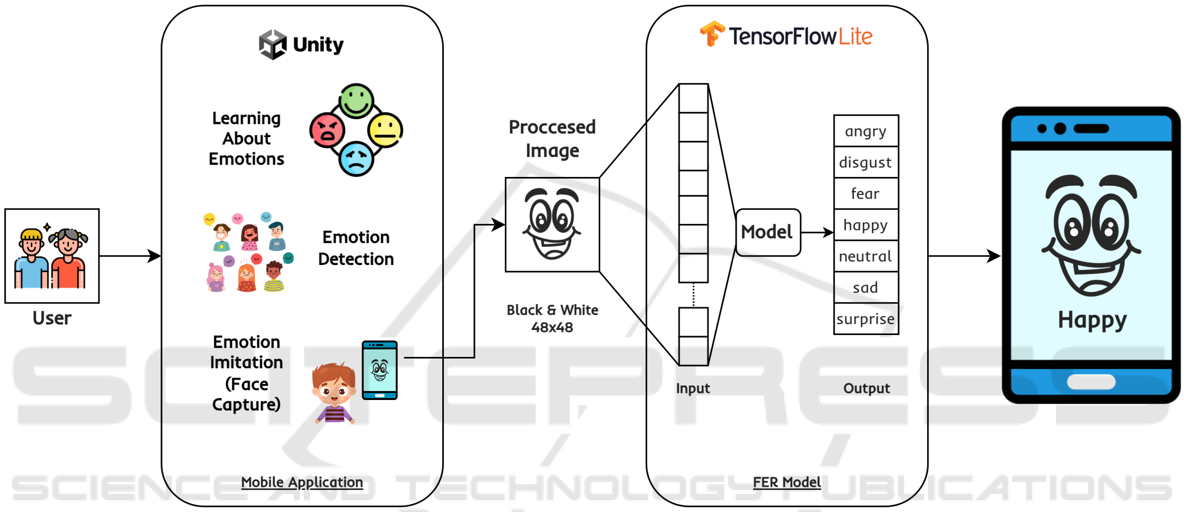


Figure 4: Emotion Game Diagram.

4.1 Experimental Protocol

This subsection explains the configuration of the environment where the experiments were performed. Local hardware, applications and frameworks used are detailed here.

The development of this work was done in a computer with an AMD Ryzen 7 3700X CPU, 16GB of DDR4 RAM memory, and an NVIDIA GeForce RTX 3070 GPU. The implementation of the FER model was carried out in a personal Anaconda environment where Python 3.9.18, CUDA 11.2 and cuDNN 8.1.0 versions were installed. Additionally, TensorFlow version 2.10.1 is used for the model building and training. The source code of the model implementation is available at https://github.com/EmotionGame-PRY20232001/EmotionGame-FER-Model

As mentioned in the previous section, the FER-2013 dataset is used in this work. It can be found in the Kaggle repository and consists in a total of 35,887 face images organized in the seven basic emotions, 28,709 for training and the rest for test evaluations. After taking a look of the image directories, we noticed several images where no faces were shown and decided to exclude them from the set, leaving a total of 28,635 images for training and 7,164 for testing. Also, we opted to use 20% of the training set for validation purposes and leave the test set for the evaluation of the model.

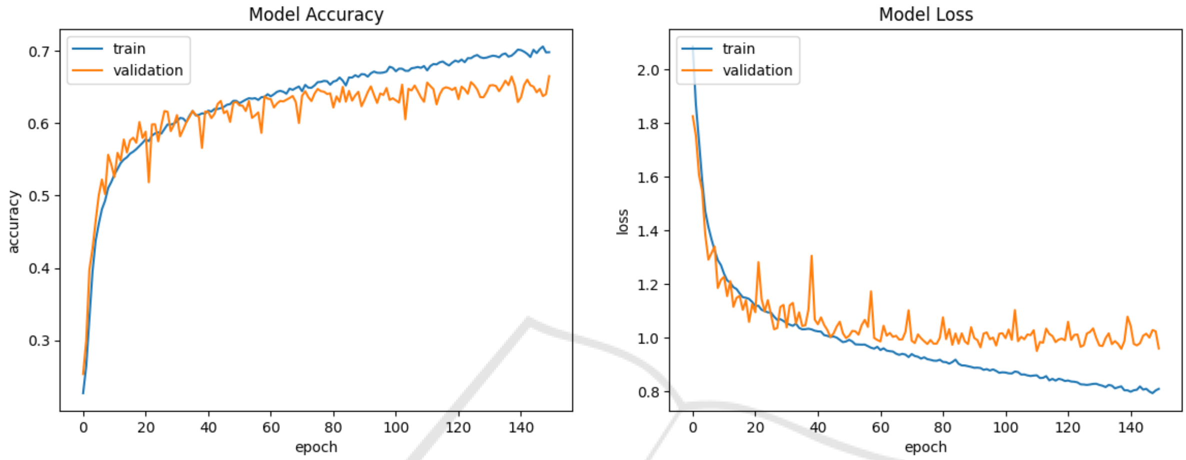
The application prototype where we tested the functionality of the lightweight model was developed in the Unity Engine 2021.3.29f1 version. To integrate the TensorFlow Lite model, a third-party library called "tf-lite-unity-sample[2]" is used, which helped us load and execute the model in Unity. Also, the OpenCV Plus Unity[3] free package is used to implement a face recognition system. The source code of the application proto-

[2]https://github.com/asus4/tf-lite-unity-sample

[3]https://assetstore.unity.com/packages/tools/integration/opencv-plus-unity-85928

Table 1: Devices Specifications.

Device	Type	Processor	Memory	OS Version
1	Phone	Octa-Core 2GHz	4GB	12
2	Tablet	Octa-Core 2.3GHz	4GB	13
3	Phone	Octa-Core 3.36GHz	8GB	13



(a) Accuracy plot.

(b) Loss plot.

Figure 5: Train and validation plot of the proposed model.

type is available at https://github.com/EmotionGame-PRY20232001/FER-Test-Project

Finally, three Android devices are use to test the application that integrates the FER model and measure the time, in miliseconds, that takes to execute the model. In Table 1, you can see the specifications of the devices used for this work.

4.2 Results

In this subsection, the results obtained during the experiment phase are detailed.

4.2.1 Model Training Performance

As we mentioned, the implementation and training of the FER model is done in a Anaconda environment. We use data augmentation for the train set, where we applied horizontal flip, 5 rotations, 20% of image zoom out, a width and height shift range of 10%, and seed 200 is used. In Table 2 you can see the final model architecture summary. The Adam optimizer with a learning rate of 0.001 was set for compiling, as well as the Categorical Crossentropy loss function. Finally, the model was sent for fitting for 150 epochs and a batch size of 32.

After the training process, the obtained results were quite satisfactory. The training accuracy after 150 epochs was 69.83% with a loss of 0.81. Similarly, the final validation accuracy achieved was 66.52% with a 0.97 loss. Figures 5a and 5b show the accuracy-epoch and loss-epoch plots, respectively, for the training process of the proposed model.

4.2.2 Model Accuracy Comparison

For this work, we selected some proposed models that are also based on CNNs and used the FER-2013 dataset for training. For example, some authors presented their own version of a CNN with different hyperparameters configuration (Zhou et al., 2021)(Vulpe-Grigorasi and Grigore, 2021)(Singh and Nasoz, 2020). Others implemented different CNN subnets to integrate them in a full model (Zeng et al., 2018)(Chuanjie and Changming, 2020). Also, Deep Neural Networks were constructed for emotion classification (Verma and Rani, 2021)(Mollahosseini et al., 2016).

Our model was sent for evaluation with the test set of the FER-2013 dataset, where additional data augmentation was applied. The evaluation ended with an accuracy of 66.50% and 0.97 loss. The accuracy comparison with other models are presented in Table 3.

Table 2: Model Summary.

Layer	Output Shape	Param #
Conv2D	(None, 48, 48, 32)	320
Conv2D	(None, 48, 48, 64)	18,496
BatchNormalization	(None, 48, 48, 64)	256
MaxPooling2D	(None, 24, 24, 64)	0
Dropout	(None, 24, 24, 64)	0
Conv2D	(None, 24, 24, 64)	36,928
BatchNormalization	(None, 24, 24, 64)	256
MaxPooling2D	(None, 12, 12, 64)	0
Dropout	(None, 12, 12, 64)	0
Conv2D	(None, 12, 12, 128)	73,856
Conv2D	(None, 12, 12, 256)	295,168
BatchNormalization	(None, 12, 12, 256)	1,024
MaxPooling2D	(None, 6, 6, 256)	0
Dropout	(None, 6, 6, 256)	0
Conv2D	(None, 6, 6, 256)	590,080
BatchNormalization	(None, 6, 6, 256)	1,024
MaxPooling2D	(None, 3, 3, 256)	0
Dropout	(None, 3, 3, 256)	0
Flatten	(None, 2304)	0
Dense	(None, 1024)	2,360,320
BatchNormalization	(None, 1024)	4,096
Dropout	(None, 1024)	0
Dense	(None, 7)	7,175
Total params:	3,388,999	
Trainable params:	3,385,671	
Non-trainable params:	3,328	

4.2.3 Performance of Model in Real-Time

As mentioned before, we integrated the model into a prototyped developed in Unity, which we tested in different mobile devices. The final weight in disk after convertion into a TensorFlow Lite model was of 13.23MB, making it proper to use in mobiles. In this prototype, we set up the model to be executed in intervals of 0.2 seconds for 15 seconds. We built the project and installed it in the three devices. In Table 4, we recorded the average process time of the model execution in those devices.

4.3 Discussion

In this subsection, we discussed about the results obtained in the previous section.

4.3.1 Model Training Performance

When dealing with neural networks, the accuracy and loss of the training and validation process are expected to maintain similar values at each epoch. Otherwise, we would be dealing with a case of overfitting, which occurs when a model is accurate with train-

Table 3: Accuracy comparison with other models.

Model	Valid accuracy	Test accuracy	Test loss
Lightweight CNN(Zhou et al., 2021)	67.00%	67.00%	0.98
Optimized CNN(Vulpe-Grigorasi and Grigore, 2021)	69.96%	72.16%	0.97
Proposed CNN(Singh and Nasoz, 2020)	-	61.70%	-
Fusion Network(Zeng et al., 2018)	-	61.86%	-
Subnets Integration(Chuanjie and Changming, 2020)	-	70.10%	-
Deep Neural Network(Verma and Rani, 2021)	70.15%	70.15%	-
Deep Neural Network(Mollahosseini et al., 2016)	-	66.40%	-
Our model	**66.52%**	**66.50%**	**0.97**

Table 4: Process time of the model in three devices.

Device	Type	Average Process Time
1	Phone	21.87ms
2	Tablet	14.39ms
3	Phone	16.06ms

ing data but not with new data. For this work, the use of data augmentation was of great support, as it helped up avoid this problem. In Figure 5a, you can see how the accuracy per epoch stays very close in train and validation, ending with a result of 69.83% and 66.52%, respectively. Similarly, in Figure 5b, the loss per epoch maintained constant proximity during the entire training. This proves that our model can be reliable for use in real cases with high accuracy.

4.3.2 Model Accuracy Comparison

As mentioned in the results section, our model was set for comparison with other proposes that are based con CNNs and were trained with the FER-2013 dataset. At first view in Table 3, we can see that our model beats some of the models in the list in test accuracy and remains close to the others. However, the subnets in (Chuanjie and Changming, 2020) and the deep neural network in (Verma and Rani, 2021) require more process time to do predictions as they are bigger architectures than our model. In our opinion, the lightwight CNN model in (Zhou et al., 2021) is one of the best proposes for real-time execution, it achives 67.00% and was stored in a 872.9KB file. It beats our model by 0.5% but our loss was slightly less than theirs.

4.3.3 Performance of Model in Real-Time

The results obtained in the tests of our model on mobile devices were quite satisfactory. As shown in Table 4, the average process time on a tablet was 14.39ms and on the best phone model we obtained a time average of 16.06ms. Models with deep architectures can take a significant amount of time to run, especially on mobile devices, which are less powerful than desktop computers or laptops. For instance, the authors in (Hua et al., 2019) mentioned that their proposed integration of subnetworks needs 2.518 seconds to predict the emotion in one picture without using a PC, which is not optimal when needing fast results. This is why developing an emotion facial recognition model with high accuracy and capable of running in less than 100ms is a great achievement, as it allows to create a more fluid human-computer interaction.

5 CONCLUSIONS AND PERSPECTIVES

This paper focuses on developing a FER model that achieves good validation accuracy and is capable of being integrated and executed in a mobile application in real-time. Although different methods have been used to try to achieve high accuracy in this area of affecting computing, not all of them are optimal for obtaining fast results. For this reason, we have implemented a lightweight CNN that was trained with the FER-2013 dataset. As seen in the first results, the use of data augmentation allowed us to achieve good results, reaching an accuracy of 66.52% in validation and avoiding overfitting.

The second achievement of this work was to integrate the implemented model into an application that can be run on mobile devices. As mentioned, TensorFlow Lite was the tool that allowed us to reduce the disk size of the FER model, resulting in a 13.23MB file. Also, as can be seen in the third part of the results, the tests on different devices were satisfactory. Here you can see how the model manages to process a prediction in an average of 14.39ms and 16.06ms, on a tablet and a phone, respectively. This proves how a low-density model can achieve high accuracy and predictions in an instant.

For our future work, two points are in mind: First, we will aim to obtain better accuracy in our model training by adjusting the hyperparameters that have been used (Leon-Urbano and Ugarte, 2020) or for other applications (Cornejo et al., 2021). Similarly, we think that using larger images, instead of the 48x48 ones, could help with this goal (Lozano-Mejía et al., 2020). Second, we plan to use the potential of automatic facial expression recognition in a serious game about emotions for children with autism. This will allow us to increase the dynamism of the activities and demonstrate the capability of artificial intelligence in human-computer interaction.

REFERENCES

Chuanjie, Z. and Changming, Z. (2020). Facial expression recognition integrating multiple cnn models. In *ICCC*, pages 1410–1414.

Cornejo, L., Urbano, R., and Ugarte, W. (2021). Mobile application for controlling a healthy diet in peru using image recognition. In *FRUCT*, pages 32–41. IEEE.

Dantas, A. C. and do Nascimento, M. Z. (2022). Recognition of emotions for people with autism: An approach to improve skills. *Int. J. Comput. Games Technol.*, 2022:6738068:1–6738068:21.

Ekman, P. and Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17(2):124–129.

Farkhod, A., Abdusalomov, A. B., Mukhiddinov, M., and Cho, Y. (2022). Development of real-time landmark-based emotion recognition CNN for masked faces. *Sensors*, 22(22):8704.

Garcia-Garcia, J. M., Penichet, V. M. R., Lozano, M. D., and Fernando, A. (2022). Using emotion recognition technologies to teach children with autism spectrum disorder how to identify and express emotions. *Univers. Access Inf. Soc.*, 21(4):809–825.

Hua, W., Dai, F., Huang, L., Xiong, J., and Gui, G. (2019). HERO: human emotions recognition for realizing intelligent internet of things. *IEEE Access*, 7:24321–24332.

Leon-Urbano, C. and Ugarte, W. (2020). End-to-end electroencephalogram (EEG) motor imagery classification with long short-term. In *SSCI*, pages 2814–2820. IEEE.

Lozano-Mejía, D. J., Vega-Uribe, E. P., and Ugarte, W. (2020). Content-based image classification for sheet music books recognition. In *EirCON*, pages 1–4. IEEE.

Minaee, S., Minaei, M., and Abdolrashidi, A. (2021). Deep-emotion: Facial expression recognition using attentional convolutional network. *Sensors*, 21(9):3046.

Mollahosseini, A., Chan, D., and Mahoor, M. H. (2016). Going deeper in facial expression recognition using deep neural networks. In *WACV*, pages 1–10.

Murugappan, M. and Mutawa, A. (2021). Facial geometric feature extraction based emotional expression classification using machine learning algorithms. *PLOS ONE*, 16(2):e0247131.

Nan, Y., Ju, J., Hua, Q., Zhang, H., and Wang, B. (2022). A-mobilenet: An approach of facial expression recognition. *Alexandria Engineering Journal*, 61(6):4435–4444.

Pavez, R., Díaz, J., Arango-López, J., Ahumada, D., Méndez-Sandoval, C., and Moreira, F. (2023). Emo-mirror: a proposal to support emotion recognition in children with autism spectrum disorders. *Neural Comput. Appl.*, 35(11):7913–7924.

Singh, S. and Nasoz, F. (2020). Facial expression recognition with convolutional neural networks. In *CCWC*, pages 0324–0328.

Ullah, Z., Ismail Mohmand, M., ur Rehman, S., Zubair, M., Driss, M., Boulila, W., Sheikh, R., and Alwawi, I. (2022). Emotion recognition from occluded facial images using deep ensemble model. *Computers, Materials & Continua*, 73(3):4465–4487.

Verma, V. and Rani, R. (2021). Recognition of facial expressions using a deep neural network. In *SPIN*, pages 585–590.

Vulpe-Grigorasi, A. and Grigore, O. (2021). Convolutional neural network hyperparameters optimization for facial emotion recognition. In *ATEE*, pages 1–5.

Wahab, M. N. A., Nazir, A., Ren, A. T. Z., Noor, M. H. M., Akbar, M. F., and Mohamed, A. S. A. (2021). Efficientnet-lite and hybrid CNN-KNN implementation for facial expression recognition on raspberry pi. *IEEE Access*, 9:134065–134080.

Zarif, N. E., Montazeri, L., Leduc-Primeau, F., and Sawan, M. (2021). Mobile-optimized facial expression recognition techniques. *IEEE Access*, 9:101172–101185.

Zeng, G., Zhou, J., Jia, X., Xie, W., and Shen, L. (2018). Hand-crafted feature guided deep learning for facial expression recognition. In *FG 2018*, pages 423–430.

Zhou, N., Liang, R., and Shi, W. (2021). A lightweight convolutional neural network for real-time facial expression detection. *IEEE Access*, 9:5573–5584.