National College of Ireland

Light-Weight Convolutional Neural Network Model for Early Emotional Detection

MSc Research Project
MSc in AI for Business

Luis Antonio Reyna Torres
Student ID: x23148802

School of Computing
National College of Ireland

Supervisor: Rejwanul Haque

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Luis Antonio Reyna Torres
Student ID:	x23148802
Programme:	MSc in AI for Business
Year:	2018
Module:	MSc Research Project
Supervisor:	Rejwanul Haque
Submission Due Date:	20/12/2018
Project Title:	Light-Weight Convolutional Neural Network Model for Early Emotional Detection
Word Count:	XXX
Page Count:	17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Luis Antonio Reyna Torres
Date:	15th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Light-Weight Convolutional Neural Network Model for Early Emotional Detection

Luis Antonio Reyna Torres
x23148802

Abstract

This research addresses the development of a Light-Weight Convolutional Neural Network(CNN) Model and its implementation for early emotional detection in users based on images. Early detection of emotions is crucial to address problems such as stress, anxiety, and depression, which can negatively impact users' daily performance and quality of life. The research focuses on developing and training an efficient CNN model, optimized for mobile devices, that is able to identify emotions such as happy, sad, and angry from facial expressions captured in real-time. The Model receives supervised training with seven different emotions based on labeled images. Finally, the performance of the model is evaluated with a stack of algorithms to measure its accuracy and loss in classifying images based on the training provided. For the deployment of this model, a mobile app is developed using iOS technology like Swift and Core ML framework for Machine Learning Models, allowing its implementation on devices such as the iPhone and iPad to process and analyze images in real-time. The implementation of artificial intelligence in mobile applications offers a powerful tool for emotional monitoring and support, due to its portable use and wide data provided by users, contributing to the well-being of users.

1 Introduction

Early emotional detection in users is a research subject that has got more attention in recent years, due to its potential to enhance people's well-being and quality of life. Especially in the educational context, the ability to identify and address emotional issues such as stress, anxiety, and depression can have a meaningful impact on students' academic performance and mental health. This research focuses on the development of a lightweight Convolutional Neural Network(CNN) model for emotion detection and its implementation through a mobile application. Convolutional Neural Networks have proven to be very effective in computer vision tasks regarding image recognition, including facial expression identification. However, CNN models are often too large and require a large amount of computational resources, making them unable for mobile devices. This research addresses this limitation by developing a lightweight CNN model that keeps high accuracy in emotion detection while reducing computational complexity, allowing it to perform efficiently on mobile devices such as iPhone and iPad. The model proposed is trained to identify seven key emotions: happiness, anger, sadness, neutral, disgust, fear, and surprise. The choice of these emotions is based on their relevance in the educational context, where the appropriate conduct of these emotions can directly affect the students' well-being and performance. Implementing supervised learning techniques, the model is

trained with a dataset formed of labeled images to learn and identify patterns and specific characteristics of each emotion.

This study motivates to achieve the research question: How effectively a Convolutional Neural Network can detect users' emotions from an image? The study based on this research question aims to understand early emotional detection in students to obtain outcomes that encourage preventive actions against possible mental issues.

Also, this research pursues to achieve the next objectives involved in the research question:

- Implementation of CNN model for image classification and pattern recognition.
- Optimization of CNN for mobile devices.
- Development and deployment of a mobile app using Core ML and Swift in iOS.

The analysis of this research is conducted by the following sections: 1- Introduction, where a brief of this study is presented with the research questions and objectives. 2- Related Work, In this section previous researches related to this topic are studied and explained to conduct the analysis of the model. 3- Methodology, this section describes the methodology and its phases operated in this study to achieve the research question. 4- Design Specification, this section describes the techniques required for the solution proposed in this study. 5- Implementation, this section discusses the different scenarios where the solution proposed is deployed and also, the tools required and the outcomes provided. 6- Evaluation, this section provides an analysis and discussion of the results and main findings of the study. 7- Conclusion and Future Work, a deep analysis of the study is addressed in this section regarding the research question and objectives. Also, a proposal for future work and the improvement of the model is detailed as well.

2 Related Work

Emotions play a crucial role in people's daily lives, influencing their decisions, behaviors, and interpersonal relationships. Early detection of emotions allows for opportune intervention, providing useful support and resources to address negative emotions and promote a positive environment. However, the implementation of efficient and accessible emotional detection systems is a challenge, especially in terms of real-time processing and portable devices Jaiswal et al. (2020). In previous works Dang et al. (2020), they analyzed audio and textual data to provide sentiment analysis but this research focuses on image analysis to provide the emotional status of the user from the facial expression.

2.1 Affective Computing and Facial Expression Analysis

Picard (2000) describes in her book "Affective Computing" as the field that combines computer science and psychology to create systems and devices that are able to recognize, interpret, and analyze human emotions. Picard (2000) explains how emotions interact in the physical world through words, gestures, music, behavior, etc. These emotional patterns can be performed voluntarily or involuntarily and can be expressed with a smile or body behavior. Affective Computing considers that these affect expressions are related to some specific emotions as patterns of them so that these patterns can be represented

in a computer. Also, the researcher underlines the fact that still being debated about whether or not characteristic bodily patterns are related to emotions.

Cacioppo et al. (1992) mentioned cases where the collection of specific data and analysis have made a difference in getting accurate outcomes of finding physiological patterns for specific emotions. This does not mean that the problem is easy to solve but some pattern expressions are better at expressing emotions than others and also, in most cases it depends on factors such as the intensity of expression and the person expressing it. Moreover, is important to define the successful result that we expect, at the end of the day a computer can perfectly recognize all your feelings. A system is considered as an outside observer with limited access to your mind and emotions. Picard (2000) mentions the possibilities that some systems are able to perform better than the user, for example, wearable and mobile devices where the systems are constantly analyzing physiological patterns and biosignals. The difference is that systems are more capable of processing patterns, although humans are more capable of interpreting patterns. Spiers (2016) on his research about facial recognition using deep learning models mentions the importance of emotions in decision-making and daily tasks, where the brain tests each possible option in a specific scenario but it is biased by emotion to quickly make a decision. He emphasizes the term empathy as a human capacity that can be enhanced by systems that are capable of measuring the emotional state of the user leading to a better understanding of their behavior.

Some of the ways to analyze emotions from users mentioned by Garcia-Garcia et al. (2017) are facial expressions, voices, body gestures and movements, text, etc. They explain in their research a tech view of tools and algorithms implemented to analyze patterns through these different emotional expressions. For purposes of this research, emotions from facial expressions are addressed to achieve the objectives of this study. Nowadays, the accelerated growth of technology and the inclusion of Artificial Intelligence(AI) conduct to the development of many sources helps to achieve a better study of facial recognition. We will take a deep understanding of these algorithms in the next Chapter 2.2.

In her book, Picard (2000) mentions how other theorists have changed the categories for describing emotional states. She mentioned that Tomkins in 1962 suggested eight basic emotions: fear, anger, joy, anguish, disgust, surprise, shame, and interest. Then, Ortony et al. (2022) collected lists of basic emotions, they got the most common four emotions from these lists combining near synonyms resulting in fear, sadness, anger, and joy. The next most common two are disgust and surprise. Over the years, researchers have suggested a variety of basic emotions but the most accurate definition was given by Ekman (1992), who defined six basic emotions with the criteria of those who are linked by a distinctive universal facial expression, resulting in fear, sadness, happiness, anger, disgust, and surprise. For purposes of this study, an emotional classifier model is developed which considers facial expression to determine users' emotions. This model follows the theory of Bassili (1979) which describes the motion patterns of the face, this means, the emotional interpretation of the user is directly related to the motion and deformation of facial features, due to motion in the image of the face would allow identifying emotions. Figure 1 represents Bassili's study on motion-based signals for facial expressions, he identifies particularly facial motions for each of the six basic emotions that provide meaningful signals to the agents to identify facial expressions.

Based on these six emotions, the model proposed in this research aims to classify seven emotions from images: happiness, anger, sadness, neutral, disgust, fear, and sur-

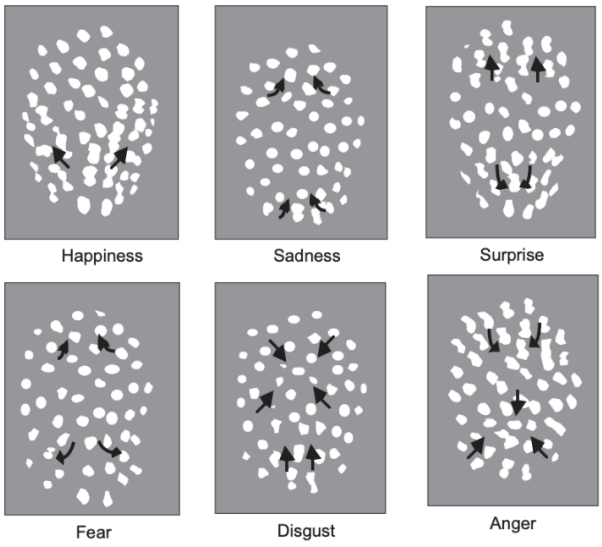

Figure 1: Bassili's model of motion signals for facial expression

prise. Considering the study of Bassili, the data based on images used to train the model strongly represents the motion signals for each emotion where it is notorious which emotion is related to. Some Machine Learning(ML) models are proposed by previous researchers to identify and classify emotions for facial recognition, these models are analyzed in Chapter 2.2 of this research.

2.2 Artificial Intelligence for Facial Expression

Artificial Intelligence(AI) is a field of study that focuses on the development of systems able to perform tasks without human intervention but they achieve these tasks with the same quality or better than humans, for example, decision-making, language translation, voice recognition, and image processing.

Machine Learning(ML) is detailed by Jaiswal et al. (2020) as a sub-field of AI that provides systems the ability to learn without any explicit coding added. This means, that ML creates models based on input data then these models provide outputs which are usually a set of predictions or decisions. Then, when a new requirement arises, the model is able to provide an answer based on the previous training data without the intervention of a new code.

Facial expressions are a key way of communication not verbal which demonstrate emotions such as happiness, sadness, and anger. Systems able to analyze and classify facial expressions require a process to be developed including fields like computer vision, psychology, and cognitive science. ML is often divided into three classes: supervised, unsupervised, and reinforcement. This section analyses some supervised ML models implemented in related state-of-the-art facial expressions to compare the limitations and characteristics required for this research.

2.3 Support Vector Machines for Motion Facial Detection

The model suggested by many researchers for facial recognition is Support Vector Machines(SVM) due to its ability to handle classification tasks with high accuracy. Michel

and El Kaliouby (2003) implemented this model for real-time expression detection arguing that it helps for a long range between classes to define each emotion from the user in continuous movement. Based on the six basic emotions, the researcher employed a face template to locate the position of 22 facial features and track their position through subsequence frames for each expression. This practice is aligned to Bassili (1979) theory where locating a motion signals of a facial expression since a neutral to a peak frame representative of the expression, it is possible to calculate the range accurately of each emotion taking the distance between feature locations. Figure 2 shows the frames captured to identify the location of facial features for each emotion.

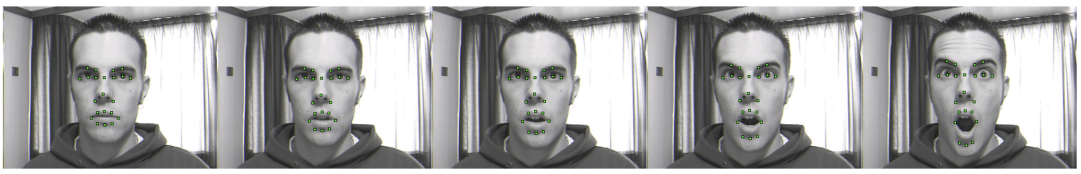

Figure 2: The sequence of frames to identify the motion cues on expression since neutral to the peak of emotion

One of the benefits of the implementation of SVM Michel and El Kaliouby (2003) explains in classification tasks the possible usage of different kernel functions such as linear, radial, and polynomial allowing flexibility in modeling complex relationships in data. This feature allows to the model define facial expression with non-linear variations. Ghimire et al. (2017) also implements SVM to locate feature in a face model achieving an accurate classification of each emotion during the training phase, one problem faced in their research was poor data implemented in the training phase. This is a common problem faced in image multi-classification tasks due to the extensive data and high compute performance required. SVM deals with noisy data and overfitting on a multi-class classification by a cascade of binary classifiers with a voting scheme been successfully employed for classification task such as text categorization and face detection.

2.4 Convolutional Neural Networks for Image Classification

Convolutional Neural Networks (CNN) has become one of the most representative Neural Networks in the field of deep learning (Li et al. (2021)). Computer vision implements CNN for visual tasks such as face recognition, autonomous vehicles, self-service supermarkets, etc. In the image-processing field, CNNs are effectively useful due to their matrix structure to analyze images through pixels with kernels. They achieve computer vision tasks such as pattern detection, image classification, and facial expression detection. Their principal difference from an Artificial Neural Network(ANN) is the structure through convolutional layers. Figure 3 shows a basic architecture for CNN, where as Li et al. (2021) explains it consists in Convolutional Layers, where kernels are applied to the input data(images) to create feature maps like shapes and patterns. Activation Layers, where activation functions like ReLu are implemented to enhance pattern learning. Pooling Layers, which help to reduce the number of parameters or features using functions like max-pooling, and finally fully connected layers that perform classification tasks.

Challenges about the implementation of CNN in image process detailed by Jaiswal et al. (2020) emotion facial detection is the quantity of data to train the model, due to CNN requires a long quantity of information to perform an accurate classification. An-

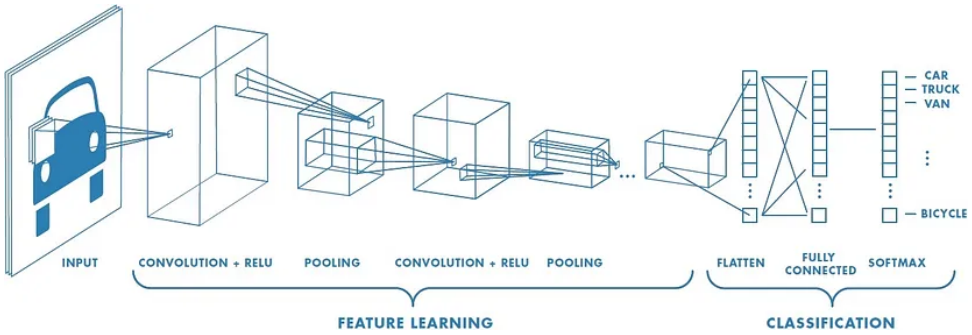

Figure 3: Basic Convolutional Neural Network architecture

other implication is the insensitive computer performance during the training phase which involves software and hardware specialized for its training. However, they highly the use of CNN for the automated processing of main features from images without manual task to define these features. Ali et al. (2020) also implements CNN for facial emotional detection supporting the implementation through three phases of emotion classification: Face detection, Facial feature extraction, and Facial classification. They suggest the use of the Viola-Jomes algorithm which uses the Haar basis feature to detect relevant features for face detection in this case: eyes brown, noise and mouth. Achieving accuracy of 95% with the implementation of CNN.

2.5 Light-Weight Models for Facial Expression(AWS, Google Cloud, and iOS)

In the study of facial expression recognition, modern cloud-based tools have risen, leveraging the computational capacity and infrastructure of cloud services like AWS and Google Cloud, which offer modern pre-trained models and a capable environment to develop and train custom models. Also, the implementation of these models into mobile technologies like iOS allows real-time facial expression analysis on a device. The implementation of CNN into mobile devices preserving its powerful performance for multi-class classification is known as the Light-Weight Model because it does not require a high compute performance to operate.

Amazon Rekognition[1] is the service provided by Amazon Web Services(AWS) which allows a powerful analysis of images and videos. it can identify objects, people, text, and emotion detection through tags leading to its easy integration with other systems and developing languages like Python for image processing tasks.

Google Cloud Vision API[2] is the service provided by Google Cloud that offers a powerful image analysis including emotion recognition and facial detection. it is capable of detecting emotions such as anger, surprise, and joy by analyzing facial landmarks and expressions. The easy integration of this tool into applications via REST API provide accurate result of image processing tasks.

Core ML[3] is the Machine Learning framework provided by Apple to integrate and develop ML models into iOS mobile applications. This framework allows real-time analysis and reduces the time of response since the user makes a request, this is because

[1] `https://aws.amazon.com/es/rekognition/`
[2] `https://cloud.google.com/vision/`
[3] `https://developer.apple.com/machine-learning/core-ml/`

models can be deployed directly into devices. Pre-trained models developed with TensorFlow or any other library from Python can be converted into Core ML models to be used in a mobile app.

As these technologies are new modern pre-trained models, there's not too much influence from previous researchers about their implementation for facial expression analysis. This challenge and opportunity to integrate modern powerful technologies in image analysis motivates this research to combine the performance of CNN multi-classification and mobile capability pursuing a high accuracy with less compute performance required.

3 Methodology

In order to answer the research question and achieve the goals planted in this study, the methodology proposed by Olson and Delen (2008) is the Cross-Industry Standard Process for Data Mining (CRISP-DM) which consists of six phases to conduct the development of a CNN multi-class classification model based on emotional expressions from images. Figure 4 shows the composition of this methodology.

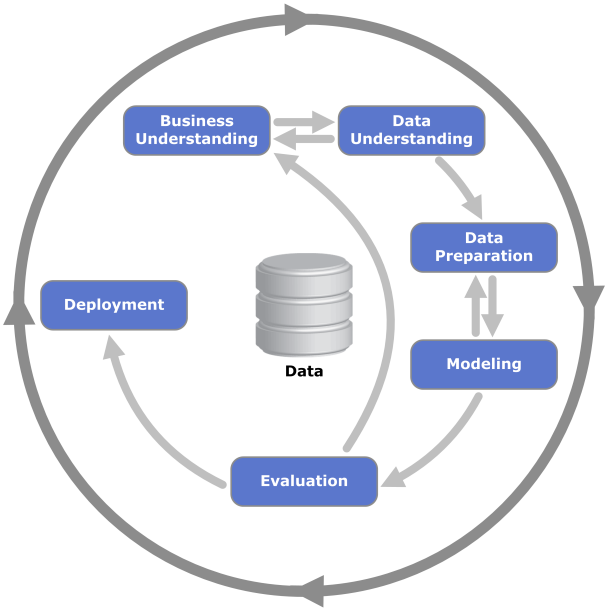

Figure 4: Phases of CRISP-DM methodology

3.1 Business Understanding

This study focuses on the development of a lightweight Convolutional Neural Network(CNN) model for the six basic emotional facial expressions (happy, sad, angry, disgust, surprise, and fear) using images. The CNN is considered a supervised model because the data used for training is labeled, this means each image used to train the model is already labeled to a specific emotion.

This model expects to achieve an accurate detection and classification of emotions in facial expressions from an image. Also, as part of its implementation is to be optimized to perform properly on mobile devices.

3.2 Data Understanding

The dataset used to train the model is FER2013 (Facial Expression Recognition 2013 Dataset)[4] which contains 30,000 facial images of the basic facial expressions such as angry, distrust, fear, happy, sad, surprise, and neutral with size restricted to 48x48. The dataset contains nearly 5,000 samples for each facial expression from a variety of populations regarding sex, age, and race. Figure 5 shows an example of the images in the dataset Fer2013. The dataset is already divided into two datasets, one is provided for training containing 28709 examples and the second one is provided for testing containing 3589. This previous data preparation is meaningful for the training phase of our model. The implementation of these two datasets will be explained in the next Chapter 3.3.

Figure 5: Example of images in Fer2013

This dataset has a free licence which means it does not require any other permission to be implemented for educational purposes as this research requires.

This dataset implemented is composed of 'in the wild' emotions which can be difficult for the training and interpretation of the model. However, due to its rich variety of populations can be beneficial to interpret different situations of facial expressions. The images within the dataset display no more faces from people in 48x48 grayscale color. The challenge to use this dataset is the model will classify the emotions regarding the expression shown in the images but there's no range that define the emotion from neutral expression until the peak of the expression.

3.3 Data Preparation

The data management phase is required in this research because the dataset implemented is stored online. FER2013 is a public dataset that contains near of 30000 samples of images classified into seven emotions. The structure of this dataset is a column for emotion and another column for the image related to that emotion (happy, sad, surprise, disgust, fear, neutral, and angry). The principal benefit of implementing this dataset is the saving of time and resources to train the model especially compute performance due to the images being collected when the model is trained and not from the local host.

First, a request is created to fetch the dataset using the library *deeplake* by Python. Then, each column is stored in a numpy array to be able to its management through the

[4] `https://paperswithcode.com/dataset/fer2013`

CNN. As the dataset is already divided into two datasets for each purpose, this action was done twice to create a request for each dataset: ds_train and ds_test. The use of both data sets in our model is influential because it ensures a diversity of samples for each emotion avoiding the use of the same image for training and test phases avoiding potential biases in the classification of the model expecting some specific characteristics of each emotion. One important consideration for data preparation is the division of the dataset into train and test values for the model. Due to our dataset being already divided into two datasets for a specific phase, this step is not necessary to be repeated in this work.

Also, previous researchers suggested the implementation of data augmentation which is the random expansion of the dataset creating a modified version of the images stored in the dataset. This method helps to avoid an underfitting issue in the model because an insufficient data to learn, resulting in a poor performance of the model especially on neural networks. This method is not implemented in our model on its first train because the dataset already has a rich diversity of samples and it is possible to achieve the main purposes of this research. Table 1 shows a resume of the steps followed in the data preparation phase.

1	Request fetch of training dataset
2	Request fetch of testing dataset
3	Create numpy arrays to store images and labels from dataset
4	Normalization of the images to the range [0,1]

Table 1: Data preparation steps applied into dataset

4 Design Specification

The use of Convolutional Neural Networks(CNN) in deep learning tasks to classify images helps to find patterns in an easy way using Kernel or filters. As an image is represented with pixels in a computer where a certain group of pixels can define a pattern within the image, a convolution multiplies a group of pixels with a filter matrix and sums the values. Then the convolution layer slides over the next pixels and performs the same process until all the pixels have been covered.

A Convolutional Neural Network is developed using Python as a core language and the library TensorFlow. The CNN is type Secuencial formed by one first layer shaped for 48 neurons as the images in the dataset are 48x48 pixels so they didn't require any pre-processing alteration on their measure. Then, two convolutional layers are added to the neural network with 32 nodes in the first layer and 64 nodes in the second layer, these numbers can be adjusted to be more or less depending on the size of the dataset. They have a kernel size of (3,3) which means the size of a 3x3 kernel matrix or filter also a ReLu activation function is implemented on each of the layers and a MaxPooling function to reduce the size of pixels matrix of each layer just keep important information with parameters of (2,2) to slide over the feature map. Then a Flatten() function is called to create a flatten layers which is the connection between the convolutional layers and dense layers. In the end, two dense layers were added to the CNN the first one with 128 neurons and ReLu activation function. The second layer or output layer with 7 neurons which means the number of categories or emotions to classify in our multi-class model.

The second layer has softmax as an activation function, this is because the model needs a probability distribution among the classes to calculate the accuracy and loss. Then the model makes a decision based on the highest probability from the classes. For the compile function 'Adam' compiler is suggested by previous researchers in more cases as it manages the learning rate throughout the training. The learning rate determines how fast the optimal weights are calculated. As a smaller learning rate more accurate weights but the time to compute the weights will be longer. Figure 6 shows the architecture of the CNN developed for this research. For classification models, the most common choice for loss function is categorial cross entropy, and 'accuracy' is used as a parameter to interpret the accuracy score by the test set. On the fit function for the CNN, it uses 30 epochs which means the times the model will cycle through the data and batch size which is the number of samples the model processes. Our model starts with a 32 batch size but it was too low at the training phase so it was increased to 64.

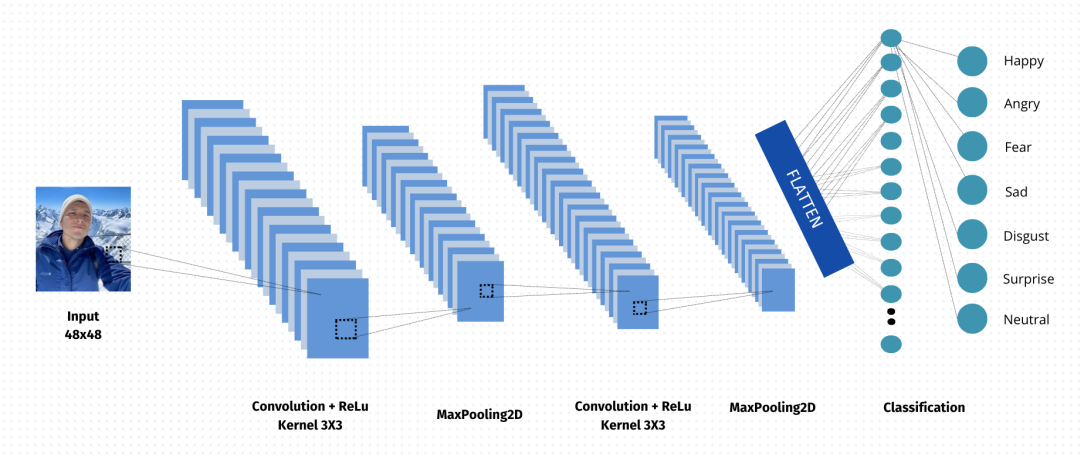

Figure 6: Architecture of CNN for Facial Expression Classification.

5 Implementation

Three phases are suggested for emotional detection: Facial detection which is provided by image pre processing, facial feature extraction like eyebrows, mouth, and nose, and finally emotion classification. For purposes of this research, the images implemented as input data to predict the emotion throughout our CNN must accomplish some features such as 48x48 size measure, the face is displayed in most areas of the picture and it is grayscale colored.

After the training phase of our CNN model, it is implemented in two different scenarios to measure it accurately to facial emotional detection: the first scenario is a Python interface that receives a local image to deliver the facial expression. The second scenario is a mobile application developed in iOS that captures a face emotion through the camera and instantly categorizes the face detection. These scenarios are detailed below with their specifications regarding sources implemented and the process taken to achieve an image classification about the face expression. Two study cases were selected in order to compare the performance of the CNN model in two different environments, expecting the same accuracy in both scenarios regarding its predictions.

5.1 Case Study 1 - CNN with Python

Having done the training of our model, its first implementation consists in an interface developed with Python language. The class named predictionEmotion.py receives a local image that displays the face emotion to predict in more of the image area. Considering that the model could receive any type of image (.jpg, .jpeg, .png) with any measure of size and it could be colored or not colored, the Python interface first realizes a pre-process of the image in order to achieve the requirements for the CNN model. First, after receive the image as input data in the class, it converts the image into grayscale color using the python library *opencv*, then the image is resized to 48x48 pixels as the first layer of our CNN is 48 neurons to create the kernel matrix and find the features related to the emotion displayed. Finally, the image is computed normalized to the range of [0,1]. Figure 7 shows the process mentioned to digitalize an image in our model.

Figure 7: Pre-process input image in the CNN model

A series of local images were implemented as input data in this scenario in which different facial expressions display their 'peak' of the expression. The result of the classification was plotted using the library matplotlib by Python to compare the similarity found in each of the seven emotions by the CNN model. In order to improve the classification in our model, some of the images to classify contain not only users' faces or the face is just a partial portion of the image, there are other cases where the eyes are covered by sunglasses or other parts of the face like eyebrows and mouth using a face mask. The results obtained from this scenario are detailed in figure 8 where accuracy was obtained for the different emotions achieving a classification from the emotion with the most similarity calculated. The images are random situations with different facial expressions, they were obtained from the internet in order to implement the model with different images from the dataset.

Figure 8: Accuracy obtained in seven emotions for each image

5.2 Case Study 2 - Mobile Application with iOS

Regarding the goals to achieve in this study, the implementation of our CNN model in a mobile device is required. For this, an application is developed in iOS environment using Swift as language coding. Xcode is the IDE by Apple to code mobile applications in iOS, for this case study we work with Xcode version 15.2.

Having trained our CNN in Python language, the next step is to transform this code into a CoreML framework to be used in an iOS application. First, we import the library *coremltools* by Python. Then, the class loads our model saved with extension .h5 which means created with TensorFlow. The function convert() transforms the CNN into light-weight CNN receiving in this case two parameters the first one is our model loaded and the second one is the type of input to receive with the characteristics needed for our model, for example, type image input with 48x48 of size in grayscale color. Finally, it creates a new framework type .mlpackage which is the extension for CoreML frameworks to use in our application.

In XCode, a blank project is created just to implement the essential environment for our ML model. It is created the UI view where the user will interact with the application and receive the action to perform the CNN, for this, an image view is added at the top of the view to display an example of our image to analyze, one button which will trigger the analysis of the image displayed when the user tap on it and a text label which will display the results provided by our CNN model. Then, a controller class named ViewController is created to –receive the actions from the UI View and make the tasks required for the facial expression analysis such as image preprocessing, request to the light-weight CNN, and send the results to the UI View. The controller has two functions, when the user taps on the button to predict the emotion from the image shown, one function receives the action and captures the image view in the UI, then obtains the metadata from the image and pre-process the image to transform it based on the requirements from our CNN model such as 48x48 size and grayscale color. For this preprocess, one extension of the element UIImage was created and applied to the image received from the interface view (these changes are not visualized for the user). The second function receives the image preprocessed and performs the analysis with our lightweight CNN with CoreML to obtain the probabilities for each emotion being the maximum of them the classification of the facial expression in the image. At the end, the results are shown in the interface view to the user. Figure 9 shows an illustration of this process.

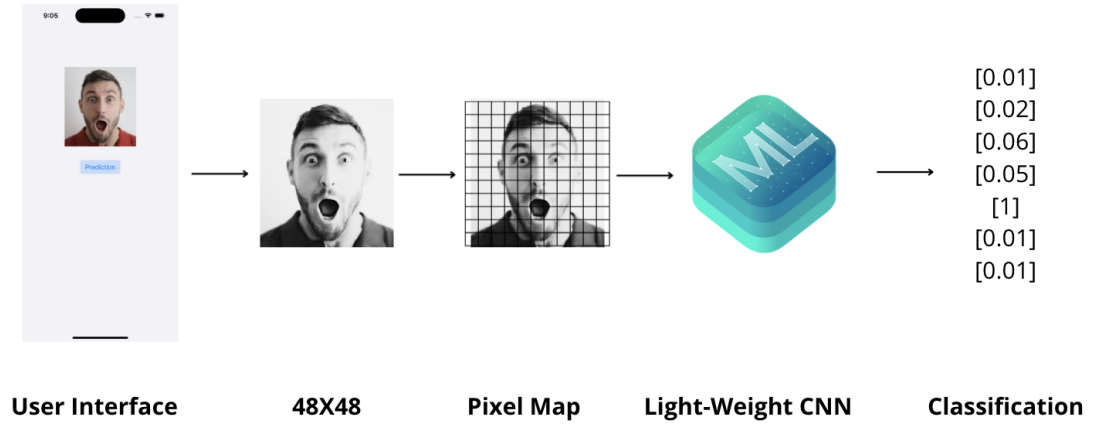

Figure 9: Process with CoreML to facial expression analysis

As part of the implementation, the same bunch of images from case study 1 are analyzed to finally compare the results in different environments. Figure 10 shows the results obtained from the CNN in CoreML. Considering the same CNN in both scenarios and the same database for the training phase, the results can vary between both implementations because the weights were optimized and could be changed during the transformation of the CNN to lightweight. However, the results are expected not to have a wide difference between them and deliver the same facial expression decision for both studies.

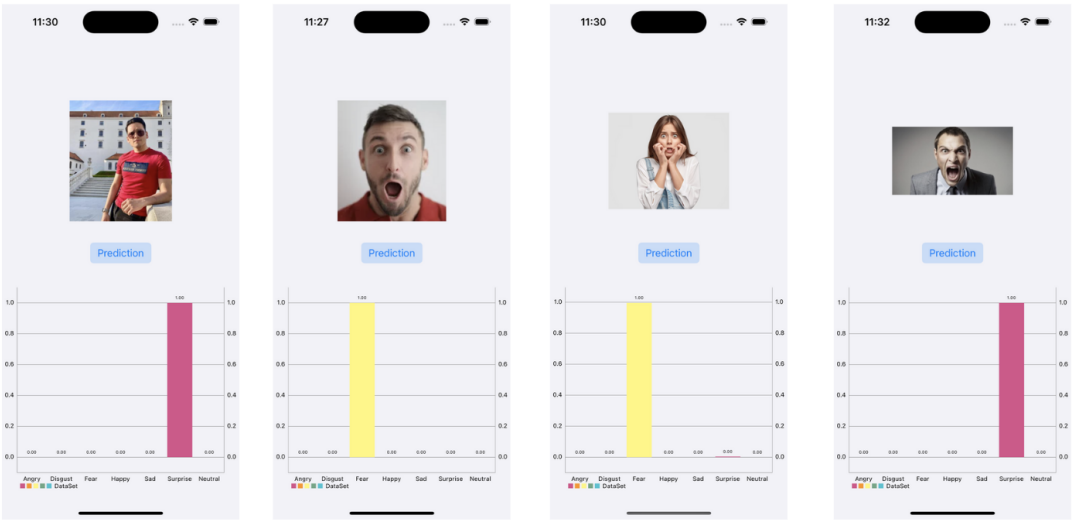

Figure 10: Accuracy obtained in seven emotions for each image with CoreML

6 Evaluation

For this section, the CNN model is evaluated with accuracy and loss during the training phase, also, a confusion matrix is provided to analyze the true and false predictions for each emotion by the CNN model. Then we will analyze the results provided by each case study and compare them to define challenges and discuss the different scenarios for CNN performance.

As mentioned, the CNN model was trained with 30 epochs which is the number of cycles training over the data and a function loss of categorical cross entropy, which allows to calculate the accuracy and loss on each epoc during the training and test phase. Figure 11 shows the evolution of training throughout the epochs.

```
Epoch 7/30
449/449 ━━━━━━━━━━━━━━━━━━━━ 63s 139ms/step - accuracy: 0.4912 - loss: 1.3218 - val_accuracy: 0.5038 - val_loss: 1.2999
Epoch 8/30
449/449 ━━━━━━━━━━━━━━━━━━━━ 63s 141ms/step - accuracy: 0.4969 - loss: 1.3054 - val_accuracy: 0.5013 - val_loss: 1.2884
Epoch 9/30
449/449 ━━━━━━━━━━━━━━━━━━━━ 69s 153ms/step - accuracy: 0.5131 - loss: 1.2669 - val_accuracy: 0.5130 - val_loss: 1.2748
Epoch 10/30
449/449 ━━━━━━━━━━━━━━━━━━━━ 58s 130ms/step - accuracy: 0.5174 - loss: 1.2429 - val_accuracy: 0.5208 - val_loss: 1.2611
Epoch 11/30
449/449 ━━━━━━━━━━━━━━━━━━━━ 63s 141ms/step - accuracy: 0.5210 - loss: 1.2240 - val_accuracy: 0.5227 - val_loss: 1.2631
Epoch 12/30
449/449 ━━━━━━━━━━━━━━━━━━━━ 62s 139ms/step - accuracy: 0.5378 - loss: 1.1846 - val_accuracy: 0.5171 - val_loss: 1.2621
Epoch 13/30
449/449 ━━━━━━━━━━━━━━━━━━━━ 67s 149ms/step - accuracy: 0.5510 - loss: 1.1563 - val_accuracy: 0.5280 - val_loss: 1.2536
Epoch 14/30
449/449 ━━━━━━━━━━━━━━━━━━━━ 61s 136ms/step - accuracy: 0.5630 - loss: 1.1281 - val_accuracy: 0.5249 - val_loss: 1.2500
Epoch 15/30
449/449 ━━━━━━━━━━━━━━━━━━━━ 56s 124ms/step - accuracy: 0.5676 - loss: 1.1031 - val_accuracy: 0.5269 - val_loss: 1.2867
Epoch 16/30
449/449 ━━━━━━━━━━━━━━━━━━━━ 58s 128ms/step - accuracy: 0.5759 - loss: 1.0900 - val_accuracy: 0.5255 - val_loss: 1.2730
Epoch 17/30
449/449 ━━━━━━━━━━━━━━━━━━━━ 65s 146ms/step - accuracy: 0.5824 - loss: 1.0568 - val_accuracy: 0.5272 - val_loss: 1.2642
```

Figure 11: Calculate of accuracy and loss for each epoch in training phase

The image above shows an increment of accuracy and decrement of loss in the training phase, however, the validation accuracy starts to has an inconsistent growth in epoch number 11 and the loss as well. From this information, we can anticipate an inconsistency in our multi class classification model. This inconsistency can be produced by some factors such as: unbalanced data, due to, there is more element from one emotion than other in the dataset, emotions can be very similar each other to classify, difficult to interpret some images or even not enough data in the dataset. Calculating an accuracy of 0.53 and loss of 1.57. A decision matrix is calculated based on the evaluation results, table 2 details the decision matrix for each emotions.

	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	196	0	67	44	96	10	54
Disgust	26	10	6	3	8	0	3
Fear	56	0	190	38	114	40	58
Happy	59	0	46	640	65	14	71
Sad	90	0	102	53	294	8	106
Surprise	21	0	56	23	18	275	22
Neutral	56	0	71	59	128	7	286

Table 2: Confusion Matrix of Training and Evaluation Phase

On the table above each emotion is represented by columns and rows, where columns represent the predictions by the model during the valuation phase and rows represent the true label for each emotion. For example, in the first row for Angry facial expression there were 196 elements true positives predicted as angry facial expression, 0 instances of angry were predicted as disgust , 67 angry instances were incorrectly predicted as fear, 96 angry instances were incorrectly predicted as sad and so on for all the facial expressions. The diagonal values represent the correct predictions for each emotion so a higher value indicate better model performance for that emotion. We notice an inconsistency in Disgust emotion in particular where there were more incorrected predictions as angry that the correct predictions for the emotion itself. This can be produce future issue predictions for this emotion in particular. After these results, we expect a well prediction for happy but a several confusion between disgust and angry.

6.1 Experiment / Case Study 1

For this case study, table a details the performance of the CNN during the training and test phase for each emotion, the performance is calculated with 3 different functions such as precision, recall and f1-score.

During the implementation of this case study as figure 8 details, a bunch of images was analyzed with out model. The images were obtained from internet to have different factor from the dataset. For images with surprise facial expression the models tends to interpret it as fear emotion and the same happen with fear facial expression which is interpreted as surprise emotion. This can be produced because the images used in the dataset are very similar between facial features each other such as eyebrows, eyes, nose, and mouth.

	precision	recall	f1-score
Angry	0.39	0.42	0.40
Disgust	1	0.18	0.30
Fear	0.35	0.38	0.37
Happy	0.74	0.72	0.73
Sad	0.41	0.45	0.43
Surprise	0.78	0.66	0.72
Neutral	0.48	0.47	0.47

Table 3: Caption

6.2 Experiment / Case Study 2

The second case study as it is noticed in figure 10 have a strong bias to fear and surprise emotion. Due to the convert function reduces weights in our CNN to optimize the performance on mobile devices, some classification can pretend to loss representation in the classification model. However, the initial CNN model was developed to perform in other environment than mobile with more compute performance, so this probability issue must be solved if the CNN is developed and trained for a mobile device since the beginning considering performance and capabilities of CoreML and devices. Also, a balanced dataset with high quantity of images and a variety expressions of each facial expression will perform a better prediction for each emotion.

7 Conclusion and Future Work

In conclusion for this research, it is important to evaluate the research question and objectives proposed at the beginning of this study. The investigation of topics such as Affective Computing, Convolutional Neural Networks, and CoreML for mobile technologies implemented in previous art-related for other researchers conducted to achieve the research question How effectively a Convolutional Neural Network can detect users' emotions from an image? This research addresses the architecture and functionality of a CNN, as well as the parameters required to make an accurate prediction and classification of facial expressions from images. The model proposed achieves an image classification considering a dataset composed of labeled images to distinguish each facial expression, in order to enhance this accuracy there are some challenges to be considered after the study of this research and that can be implemented in future work. Moreover, regarding the objective of this study, the implementation of a CNN model for image classification succeeded in both study cases, a deep understanding of its architecture allowed us to make the proper adjusts for each environment and optimize the use of computer performance for mobile devices using native languages such as Python and Swift.

7.1 Challenges

Throughout the study of this research, some challenges and limitations were faced especially during the training phase because the Convolutional Neural Network was developed once time for both study cases. This drove to identify stable versions of frameworks such as TensorFlow, Python, coremltools, and CoreML that perform between each other for both scenarios. To compare both study cases was necessary to develop and train only one

CNN for both study cases which one of them was limited to perform over the classification of facial expressions. Another challenge considered is the multi-class classification because this research worked with 7 classes and there are facial expressions quite similar between two or three of these classes. The limited implementations of CNN in mobile devices motivate the study of this research and it led to considerable time and effort to investigate the conversion of a lightweight CNN and its implementation for facial expression classification in a mobile device.

7.2 Future Implementations

The combination of AI technologies and mobile devices conducts beneficial implementations for any kind of industry. This model was designed to be implemented in the educational domain where through mobile applications implemented in schools, the system performs a facial expression analysis of the student, and through historical data, it can deliver an early detection of mental issues such as stress and anxiety. This analysis helps to take action on this mental situation. For improvements in this model, it must be considered a larger database with a balanced image of each facial expression, not only in their peak of emotion but also some variations to increase the probability and diversity for each one. Also, the development of a CNN especially to perform on mobile environment with CoreML is considered to achieve a better accuracy in classification tasks.

References

Ali, M. F., Khatun, M. and Turzo, N. A. (2020). Facial emotion detection using neural network, *the international journal of scientific and engineering research* .

Bassili, J. N. (1979). Emotion recognition: the role of facial movement and the relative importance of upper and lower areas of the face., *Journal of personality and social psychology* **37**(11): 2049.

Cacioppo, J. T., Uchino, B. N., Crites, S. L., Snydersmith, M. A., Smith, G., Berntson, G. G. and Lang, P. J. (1992). Relationship between facial expressiveness and sympathetic activation in emotion: a critical review, with emphasis on modeling underlying mechanisms and individual differences., *Journal of personality and social psychology* **62**(1): 110.

Dang, N. C., Moreno-García, M. N. and De la Prieta, F. (2020). Sentiment analysis based on deep learning: A comparative study, *Electronics* **9**(3): 483.

Ekman, P. (1992). An argument for basic emotions, *Cognition & emotion* **6**(3-4): 169–200.

Garcia-Garcia, J. M., Penichet, V. M. and Lozano, M. D. (2017). Emotion detection: a technology review, *Proceedings of the XVIII international conference on human-computer interaction*, pp. 1–8.

Ghimire, D., Jeong, S., Lee, J. and Park, S. H. (2017). Facial expression recognition based on local region specific features and support vector machines, *Multimedia Tools and Applications* **76**: 7803–7821.

Jaiswal, A., Raju, A. K. and Deb, S. (2020). Facial emotion detection using deep learning, *2020 International Conference for Emerging Technology (INCET)*, IEEE, pp. 1–5.

Li, Z., Liu, F., Yang, W., Peng, S. and Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects, *IEEE transactions on neural networks and learning systems* **33**(12): 6999–7019.

Michel, P. and El Kaliouby, R. (2003). Real time facial expression recognition in video using support vector machines, *Proceedings of the 5th international conference on Multimodal interfaces*, pp. 258–264.

Olson, D. L. and Delen, D. (2008). *Advanced data mining techniques*, Springer Science & Business Media.

Ortony, A., Clore, G. L. and Collins, A. (2022). *The cognitive structure of emotions*, Cambridge university press.

Picard, R. W. (2000). *Affective computing*, MIT press.

Spiers, D. L. (2016). Facial emotion detection using deep learning.