BABEŞ BOLYAI UNIVERSITY, CLUJ NAPOCA, ROMÂNIA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

# MOE: classroom assistant

– MIRPR report –

**Team members**
Raicu Robert, Software Engineering, 258/2

2019

## Abstract

Working with young pupils is not easy, any teacher can tell you that. It's a period of life when they still develop and sometimes get easily discouraged or distracted especially when working on computers. Because of this, there is a need for classroom tools to allow only certain applications and monitor the emotional state of children as one teacher cannot keep track of an entire class at once. During this paper we will discuss the implementation of such an application using existing neural networks like FaceNet MTCNN [6] for face detection, ResNet50 [5] for facial authentication and VGG16 [7] for emotion recognition toghether with an .Net WPF application for the management of the classroom, applications, sessions and recordings.

This paper is about the integration of multiple technologies and existing AI papers to create a useful application for managing the classroom, not about building or the optimizations of specialized neural networks.

# Contents

# List of Tables

# List of Figures

# List of Algorithms

# Chapter 1

# Introduction

## 1.1 What? Why? How?

Young pupils have the tendency to be distracted or get demotivated easily while doing tasks in the classroom. Monitoring their emotions in relation to the task at hand can help us better understand when and why it happens. This allows the teachers to get insights on their behaviour and be able to restructure lessons or give personalized assistance to each pupil in order to maximize the information passed. Our approach is to have a specialized application that allows the teacher to manage pupils, allowed apps and session recordings on which children can be logged in and out using facial recognition for ease of use.

## 1.2 Paper structure and original contribution(s)

The research presented in this paper advances the design, combination and implementation of and application using particular models and technologies.

The main contribution of this report is to present an intelligent solution for solving the problem of identifying how emotions can affect a classroom's productivity.

The second contribution of this report consists of building an intuitive, easy-to-use and user friendly software application. Our aim is to build an application that requires no teacher interaction while pupils use it such that he/she can concentrate on other tasks.

The present work contains 7 bibliographical references and is structured in five chapters as follows.

The first chapter/section is a short introduction.

The second chapter/section describes the problem the paper wants to solve.

The third chapter/section talks about related work in the field.

The chapter/section 4 describes the proposed approach.

The fifth chapter/section talks about the implementation of the model used.

The sixth chapter/section presents the conclusion.

# Chapter 2

# Scientific Problem

## 2.1 Problem definition

The goal of our application is to help teachers track the pupils emotional state during the use of computers in didactic activities in order to better understand their behaviour. The need for an automated solution like this comes from the sheer amount of information and subjects. A single person would not be able to perform their didactic activities while also monitoring the current activity and facial expression of each pupil in a full classroom.

As we know, face recognition is nothing new as it started in the years 1964-1968 [4] where facial features were mapped by hand. A huge breakthrough came when in the early 2000's the process was automated using AI. Since then many advancements the field of face recognition and mapping were made using state of the art topologies.

For this task we used three neural networks: ResNet [5] and VGG [7] from the Keras [1] library in Tensorflow [2] and a FaceNet MTCNN [6] implementation.

## 2.2 FaceNet MTCNN I/O

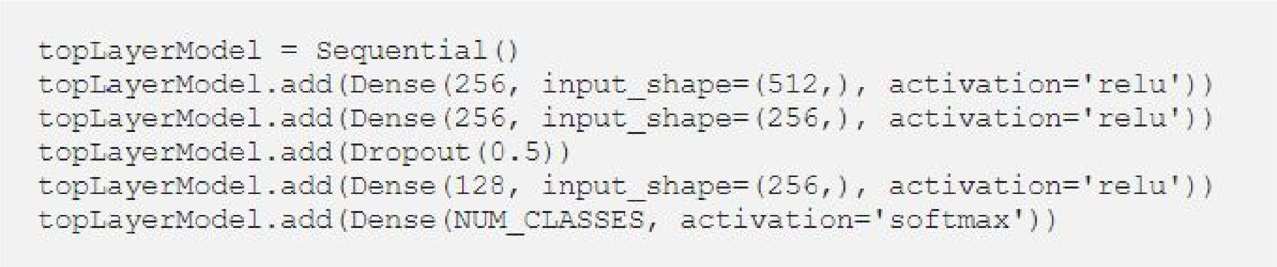
FaceNet [6] is used in order to crop the face if the pupil from a raw camera image. The input image can be any size although the performance will be impacted. The output that we are interested in is an array of bounding boxes that represent detected faces.

## 2.3 ResNet I/O

ResNet [5] is used in order to extract facial features in the cropped image such that we can identify individuals. The netural network was pre-trained on the ImageNet dataset. The input image has to be 224x224 pixels having three color channels. The output is a 2048 array of doubles that represent the facial features of the subject.

## 2.4 VGG I/O

VGG [7] is used in order to recognize the emotion of the pupil. In order to do this, the VGG16 architecture is used without the top layer leaving a 512 double array base output. From this, a model is appended to map the 512 output to a 7 class output (Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral). Then the model is trained on the Fer2013 image set.



```
topLayerModel = Sequential()
topLayerModel.add(Dense(256, input_shape=(512,), activation='relu'))
topLayerModel.add(Dense(256, input_shape=(256,), activation='relu'))
topLayerModel.add(Dropout(0.5))
topLayerModel.add(Dense(128, input_shape=(256,), activation='relu'))
topLayerModel.add(Dense(NUM_CLASSES, activation='softmax'))
```

The input of the model will is 43x43 pixels with tree color channels. The issue is that the Fer2013 image set is grayscale and because of this the cropped image of the face has to be converted to grayscale then used as an input of all three channels.

# Chapter 3

# State of art/Related work

One example of a paper on this subject is "Deep Learning based Student Emotion Recognition from Facial Expressions in Classrooms" [3]. It proposes a system where two high resolution cameras are positioned in front of a classroom and record all emotions in order for teachers to improve their lectures such that students pay attention. The main difference in between the system described by the cited paper and the one implemented here is that theirs take a group approach as an emotion metric and it's concentrated on the emotion as a group where our approach is centered around the individual.

# Chapter 4

# Proposed approach

In order to create such a application we need to solve 4 sub-problems:

- detect and crop the the face of the subject from the image
- create a form of identification for the given face
- identify the emotion expressed by the subject
- create a application to serve as an interface for the pupils and the teacher

After solving these problems using the neural networks described in chapter 2 and using the .Net framework we can implement the application described in the following sections.

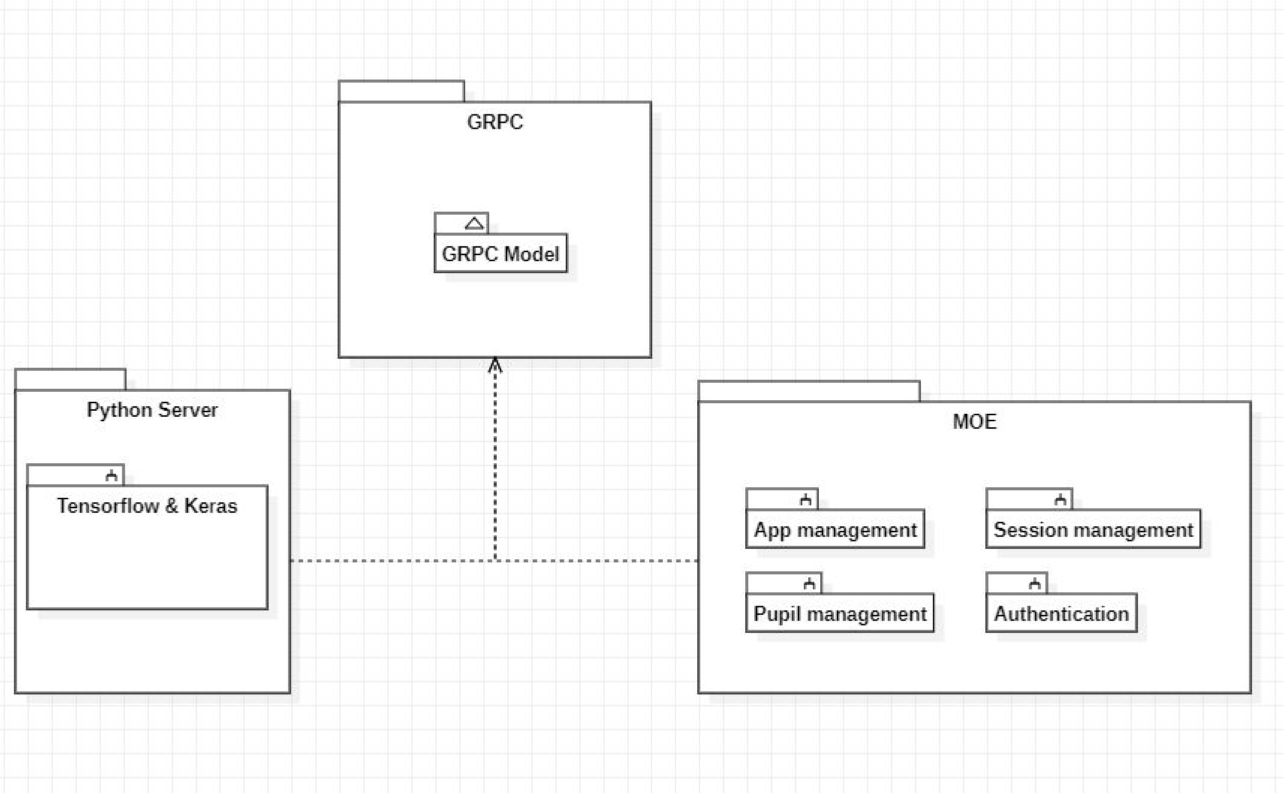
## 4.1 Overall architecture

The application is split into 2 main parts and a communication channel. This was designed this way because of the nature of the machine learning libraries as they run on the Python platform.

The communication channel is represented by the GRPC library created by Google that uses protocol buffers in order to transmit data between services. Because of this we can make remote procedure calls between Python and C# with little effort so we can transfer the necessary data with a low overhead.

The first part is the Python server which is responsible for the image processing. It handles image capture, face cropping, emotion detection and face features extraction. This data is bundled and sent through the GRPC library to the main application to be used.

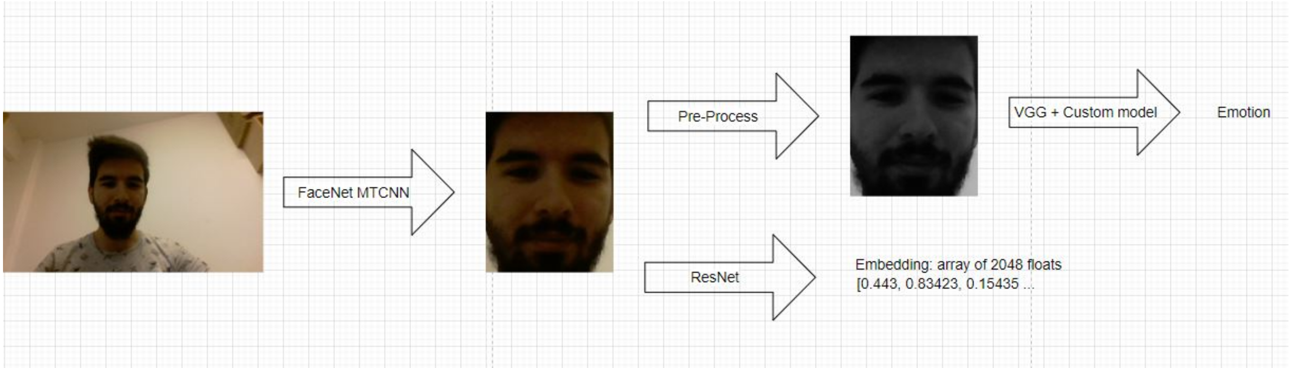
The second part is the .Net application called "MOE" which is responsible for managing applications, pupils, sessions and authentication. This application is where all business logic and user interaction layer resides.



## 4.2 Python server

The Python server serves as the visual processing hub. Using CV2 it captures live video frames and processes them using Tensorflow and Keras in order to extract the required information. The captured image is 1280x720 pixels and is fed to the FaceNet MTCNN model which in turn returns a bounding box if a face is found. Then we cut the face from the original frame and using ResNet we calculate the face embedding whitch we use as a form of identification. From the original frame we also pre-process a grayscale image that we feed to the VGG composed model such that we can extract the emotion.

The final process is that we pack the cropped image, the emotion and face embedding in a message and send it to the .Net application using GRPC to be used.



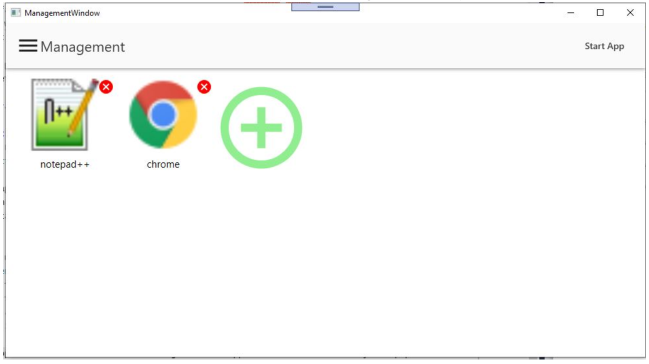
## 4.3 .Net application

The .Net application serves as the 'brains' of the operation. It manages all aspects starting from database connection to business logic and user interaction. The application is split into two parts: the managing side for the teacher and the desktop side for the pupils.

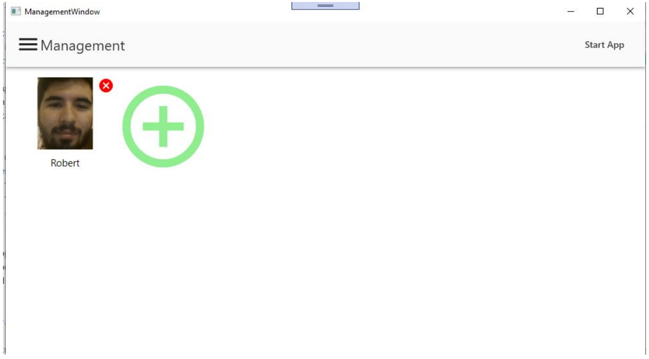
The managing side of the application offers 3 services

- the management of registered applications
- the management of registered pupils
- a history of recorded sessions.

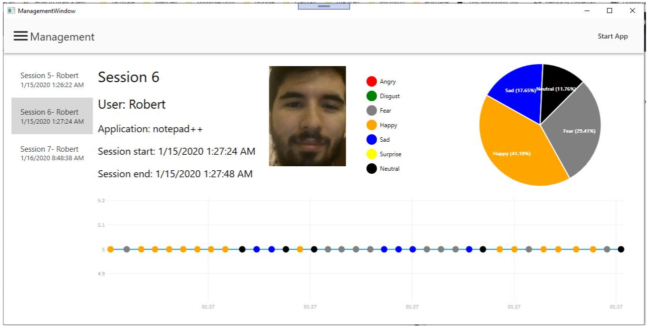
On the "Apps" screen the teacher can configure which applications can be used by the pupils in the class. Applications can be added or removed using the "+" or "X" icons on the screen. Only applications previously configured by the teacher will be available to the students.



On the "Students" screen the teacher can manage the pupils that can access the system. Students can be added or removed using the "+" or "X" icons on the screen. Only students previously added by the teacher will be available to login into the application.



On the "Sessions" screen the teacher and review previously recorded sessions. A session starts recording when a pupil opens an app and registers all emotions that he/she displays. The sessions ends when the pupil closes the application or logs out. On this screen the teacher can see when the session was recorded, which app was used an by which student. There are also two charts: one representing a timeline of the emotions displayed and one pie chart representing the percentage of time a certain emotion was displayed. All emotions are color coded and visible as a legend.


ManagementWindow

Management

Session 5- Robert 1/15/2020 1:26:22 AM

Session 6- Robert 1/15/2020 1:27:24 AM

Session 7- Robert 1/16/2020 8:48:38 AM

Session 6

User: Robert

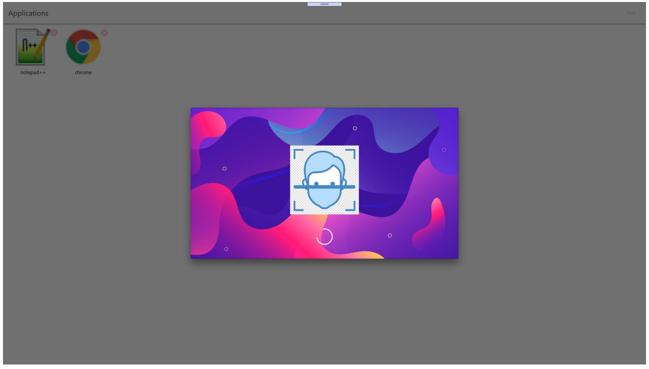
Application: notepad++

Session start: 1/15/2020 1:27:24 AM

Session end: 1/15/2020 1:27:48 AM

The desktop side of the application for the pupil has two important systems at play.

The first one is the authentication manager and login dialog. It uses the information provided by the python server to compare the recorded embedding to the ones registered by the teacher. If the same pupil is visible for a pre-configured amount of time, he/she will be logged in and the dialog will disappear. When the pupil leaves the camera's view, the system will countdown a pre-configured ammout of time before logging out the current user and showing the login dialog again.



The second one is the session manager which is started when the pupil selects one of the available applications. This will start a session recording which takes the current user and the selected application and stores time and emotion data pairs in the database. When the pupil closes the application the recording is stopped

# Chapter 5

# Application (numerical validation)

In this chapter we discuss discuss the model appended to VGG16 in order to detect the emotion of the user.

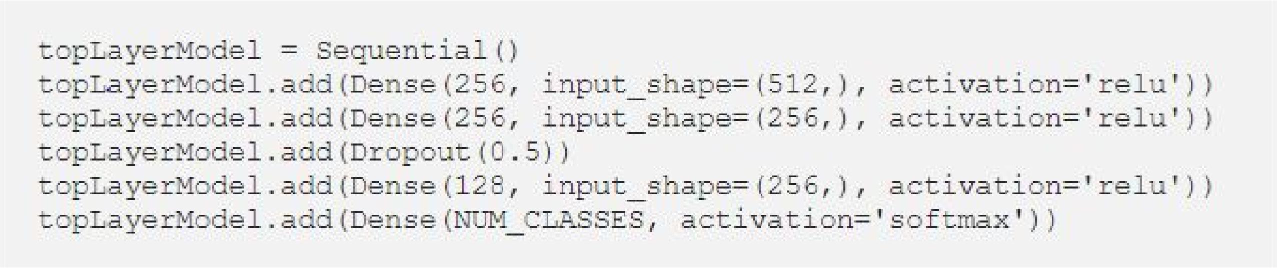
## 5.1 Methodology

- As our emotion classification is essentially a classification problem, for evaluation we use precision and validation loss.

- The aim is to get better insights on the emotions pupils express during classes and test it's performance on real world data.

## 5.2 Data

As described earlier, we performed transfer-learning on the VGG16 architecture trained on the "ImageNet" set by combining it with our custom model and retraining on the "Fer2013" data set. The model attached to the VGG16 architecture is described in the image below.



```
topLayerModel = Sequential()
topLayerModel.add(Dense(256, input_shape=(512,), activation='relu'))
topLayerModel.add(Dense(256, input_shape=(256,), activation='relu'))
topLayerModel.add(Dropout(0.5))
topLayerModel.add(Dense(128, input_shape=(256,), activation='relu'))
topLayerModel.add(Dense(NUM_CLASSES, activation='softmax'))
```

The VGG16 architecture was expecting a 3 color channel image so we had to use the grayscale image for each of the channels in order to train the network on the "Fer2013" data set.

## 5.3 Results

Using the model described above we managed to achieve the following accuracy:

- training set (Model has seen these): 99.8

- test/validation set (New inputs to model): 43.7

## 5.4 Discussion

Even if we did not reach state of the art results for emotion classification we believe that the application has real world application in education and research. Using custom models and better data sets could improve the results by a considerable margin.

# Chapter 6

# Conclusion and future work

As we stated in the beginning our goal was not to improve or implement advanced neural networks but to show how important the integration of such technologies could prove in a real life scenario. The use of AI in day to day life is still new even though we came a long way in such a short time. Using tools like this can help us improve processes that were done only with human interaction until now.

# Bibliography

[1] Keras. https://keras.io/.

[2] Tensorflow. https://www.tensorflow.org/.

[3] Vibhakar Mansotra Archana Sharma. Student emotion recognition system (sers) for e-learning improvement based on learner concentration metric. *International Journal of Engineering and Advanced Technology (IJEAT)*, Volume-8 Issue-6, 2019.

[4] Woodrow Bledsoe. Semiautomatic facial recognition. *Technical Report SRI Project*, 6693, 1968.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[6] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. cite arxiv:1409.1556.