GIP Project, Winter 2020-2021

Efrat Israel, Akiva Zonenefeld

Supervisors: Yaron Honen, Prof. Tzipi Horovitz-Kraus, Dr. Michal Zivan, Amit Bracha

Computer Science Department, Technion - Israel Institute of Technology, Haifa

Educational Neuroimaging Center, Technion - Israel Institute of Technology, Haifa

Table of Contents

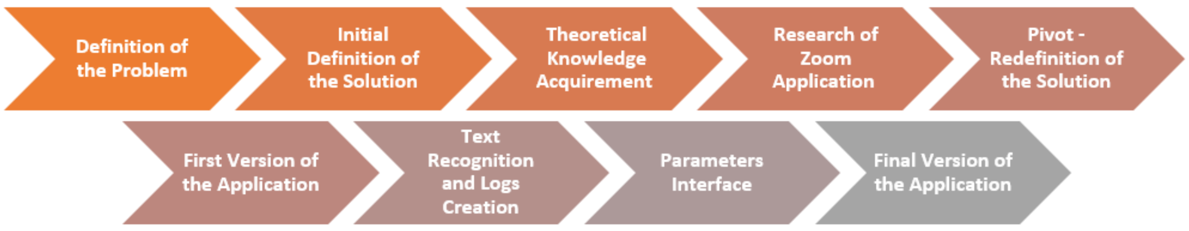
Abstract

Our project is the product of a collaboration between the Geometric Image Processing Lab (GIP) at the Computer Science department and the Educational Neuroimaging Center (ENIC) at the Technion.
The goal of the project is to create a desktop application that analyzes children's emotions during virtual classes and indicates the children that express negative emotions in real time. In addition, we provide a log with the emotions of each child during the class, including times that face was not recognized, and statistics of the class emotions as a group. The log is used for research in ENIC.
We track the children along a video using face recognition and image processing tools, and output their emotion using a classifier. We are deploying a state-of-the-art model which was developed by the GIP lab for facial expression recognition on children, RonNet.
Although the name of the project indicates that the video conference must be held in Zoom application, our desktop application can be used with any kind of video conference application as well as with video recording.
In order to track each child as an individual we use text recognition tools and match the reappearing name with the relevant adjacent face and emotion.

Background and Motivation

Distance education and virtual classes among young students are the future, and are emphasized worldwide in the last year during the era of COVID-19, where children around the world study partially or exclusively distantally. Teachers struggle with many difficulties teaching in that new method, and being able to feel their students virtually is one of their main struggles. For that reason, creating a real time application that assists the teachers in indicating which children need more attention at the moment, because of expressing negative feelings, might help the teachers to understand their students' feelings better. In addition, being able to analyze the children's feelings during the class might contribute to better understanding of the children's behaviour and needs in classes at general, And in particular during virtual classes.

Workflow


Definition of the Problem
Initial Definition of the Solution
Theoretical Knowledge Acquirement
Research of Zoom Application
Pivot - Redefinition of the Solution
First Version of the Application
Text Recognition and Logs Creation
Parameters Interface
Final Version of the Application

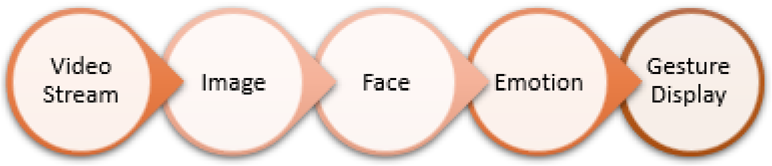
1. **Definition of the Problem**

   The problem is to create a desktop application that analyzes children's emotions during virtual classes and indicates the children that express negative emotions in real time. In addition, a log with the collected data is required.

2. **Initial Definition of the Solution**

   We defined the solution as follows:

   1. Get the video stream of each meeting participant from the Zoom meeting.
   2. Capture image from the video stream.
   3. Detect the faces of the children in the image.
   4. Detect the emotion of each child.
   5. Display the emotion gesture back on the video.



3. **Theoretical Knowledge Acquirement**

   Starting this project, we dived into understanding of deep learning with an image processing approach by studying the Stanford 231n course (Convolutional Neural Networks for Visual Recognition). In addition, we researched the RonNet Model for emotion recognition in children that we used in the application. RonNet Model is a classifier that was developed by Mr. Ran Breuer during his master's work for prop Ron Kimmel [1]. This is a classic feedforward convolutional neural network. Ran's implementation consists of 3 convolutional blocks, each with a rectified linear unit (ReLU) activation and a pooling layer with 2×2 pool size. The convolutional layers have filter maps with increasing filter (neuron) count the deeper the layer is, resulting in a 64, 128 and 256 filter map sizes, respectively. Each filter supports 5×5 pixels. The convolutional blocks are followed by a fully connected layer with 512 hidden neurons. The hidden layer's output is transferred to the output layer, whose size is affected by the task at hand - eight for emotion classification. Dropout was applied

after the last convolutional layer and between the fully connected layers to reduce overfitting with probabilities of 0.25 and 0.5 respectively. A dropout probability p means that each neuron’s output is set to 0 with probability p. For training ADAM optimizer was used with a learning rate of 1e−3 and a decay rate of 1e-5.
In addition, we researched the tools for the solution implementation: OpenCV, Torch and PyAutoGui.

4. Research of Zoom Application

Zoom provides videotelephony and online chat services through a cloud-based software platform and is used for telecommuting, distance education, and social relations. Zoom runs across mobile devices, desktops, telephones, and room systems.
Zoom enables developers to build applications and integrations on top of their unified video communications platform, spanning video, voice, content sharing, and chat across desktop, mobile and workspaces.
We researched deeply the different options that are provided by Zoom. We focused on our needs for the project as described in the problem definition.
We came to the next conclusions:

- Zoom API is used by developers to create a custom managed meetings, for example creating new meetings, creating, adding and removing users, viewing reports and dashboards on various usage. It does not meet with any of our needs.

- Zoom Client SDKs are basic app development kits provided to enrich apps with video collaboration features to connect customers and communities. These kits do not give access to the raw video of each participant. In addition, annotating on the videos in order to display the child's emotion is not possible.

- Zoom Fully Customized SDKs are the extended version of the Zoom client SDKs. These kits give access to the raw video of each participant. Display annotation on the screen (but not on video) is possible but might be displayed for all users. Major downside was the need to purchase the kits but Zoom Sellings was not responsive so we were not able to purchase them.

Summary diagram:

	Client SDKs	**Fully Customized SDKs**
Image Processing	Analysis of full screen	Analysis of each video stream separately
Annotation Display on Screen	Not supported	Supported - Display for all users
Annotation Display on Video	Not supported	Not supported
UI Customization	Not supported	Supported
Addition Difficulties	Non	Need to be purchased and Zoom Sellings is unresponsive

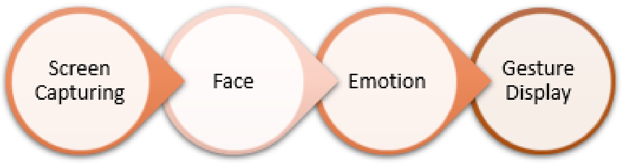
In conclusion, in order to support all needs we had to come up with a new solution to solve the problem since Zoom API and SDKs were not sufficient.

5. Pivot - Redefinition of the Solution

The new solution does not involve Zoom application. It is using sequential screen capturing of the video conference, analysing the captured images, and displaying annotations on top of the video screen using an external application.

The solution was redefined as follows:

1. Capture of the video conference screen (Main change).
2. Detect the faces of the children in the image.
3. Detect the emotion of each child.
4. Display the emotion back on the video.



The advantages of the new method are standing bold:

1. Ease of Implementation - The application is modular and can be used with every video conferencing platform as well as on recordings because it does not depend on the platform.
2. Ease of Use - The teachers run the application as a background program and can use their preferred video conference platform and be assisted by our application simultaneously.
3. Displaying annotation on top of the video is possible so the application annotation appears in a more intuitive way.
4. The teacher can use the most updated version of the video platform and is not limited to an application that is based on a specific version of the platform.

The main disadvantage is that the analysis is of the entire screen and not of each video stream separately. It requires additional analysis in order to track the emotions of each child during the session. We proposed using text recognition in images and matching it with the detected face and emotion, by means of position on the screen.

6. First Version of the Application

Our mid-meeting milestone was a demo application that recognizes two children's faces, recognizes their emotions and displays relevant annotations on the video screen.

7. Text Recognition and Logs Creation

In order to track the children's emotions during the session, and overcome the fact that the analysis is of the whole screen, a method of grouping the faces of each child has to be done. Face tracking algorithm was considered, but the need of using it on up to sixteen faces in real time made it irrelevant because of computation time.
We decided to use text recognition from images. We matched the text that was recognized from the screen capture and matched it with the detected face and emotion, by means of position on the screen. We used Tesseract which is an optical character recognition engine and is the most popular and qualitative OCR-library. OCR uses artificial intelligence for text search and its recognition on images. Tesseract is finding templates in pixels, letters, words and sentences. It uses a two-step approach that calls adaptive recognition. It requires one data stage for character recognition, then the second stage to fulfil any letters, it wasn’t insured in, by letters that can match the word or sentence context.
Each recognized face and emotion is kept with the child name and the timestamp in the session that it was recognized at.
After the face and emotion were matched with the name, we expect that the face will be recognized again if the name is detected once again. In that way we can provide information about each child that was recognized with emotion during the session.
In order to get more accurate names recognition by Tesseract, we used Difflib, an algorithm that finds the closest matching strings in a list of strings. We matched the names that were found by Tesseract to the names that were already found in previous iteration or that were provided in the participants names list (see explanation in the User Manual section). In that way the process of mapping each name to a single child is more efficient.
We held the data in two logs, one that summarizes the emotions that were detected in the whole session, and the other one that contains the data about each student as an individual.

To mention that attempts to track children that their faces were not recognized although they were supposed to be (and by that to indicate that the child was not concentrated during class) did not come through since the tracking is based on the text recognition, and its results were not sufficient for that method.

8. Parameters Interface

We added interface that enables the teacher to define different parameters regarding the emotions that are detected:

1. Emotions Threshold - the minimal confidence, between 0 to 1, to show the recognized emotions on screen.
2. Maximal faces to draw - maximal number of emotions that are drawn on screen simultaneously.
3. Gesture on screen time - for how long the gesture will be displayed on screen.

9. Final Version of the Application

We extended the application capabilities to handle up to sixteen faces. Main limitation is the resolution of the faces when more than sixteen faces are displayed on screen. In addition, we created logs and graphs that show each student's emotion during the class, and a graph that summarizes all emotions that were detected in the session, as a function of time.

Description of the Algorithm

1. **Capture the Video Screen**
2. **Analyze the Captured Image:**

   2.1. Find Video Frames in the Image

   (We know that the contour of the video frames pixels are black. Therefore, we converted the rest of the image pixels to white, to receive mainly the video frames contours. Then we used CV's Canny, which finds rectangles in the image. We kept only the rectangles in size that are possibly the video frames.)

   2.2. For Each Frame:

   2.2.1. Detect Face In Frame (Using CV Face Detection)

   2.2.2. Recognize Name From Image And Match with the Face (Using Tesseract-OCR - the name is searched at the lower part of the video frame, where it is supposed to be found, and match it with known list of name, from previous iteration or from a provided list)

   2.2.3. Detect Emotion Of Face (Using RonNet)

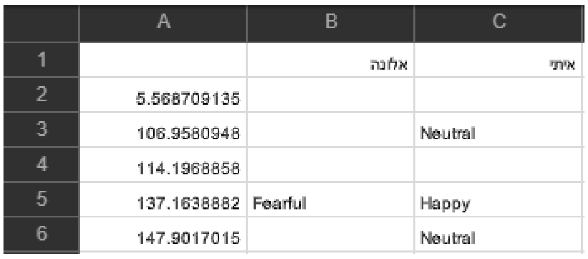
   2.2.4. Update Log with Relevant Data (time, name and emotion if were detected)

Results

After closing the application, it produces two differents logs and several graphs:

1. **'detailed log.csv'** holds a table that for each timestamp indicates which children exhibit which emotion (if the child was not found or the emotion was not conclusive to be displayed, the cell will remain empty).

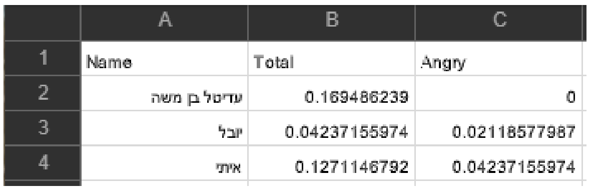
   Example:



	A	B	C
1		אלונה	איתי
2	5.568709135		
3	106.9580948		Neutral
4	114.1968858		
5	137.1638882	Fearful	Happy
6	147.9017015		Neutral

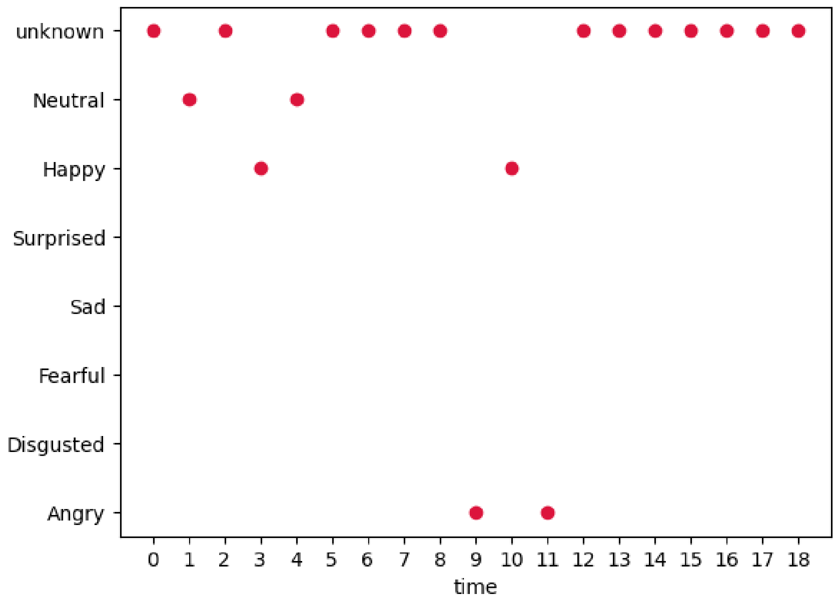
2. **'summary log.csv'** summarizes the amount of time that each emotion appeared for each child, and how long (conclusively) the emotion was displayed during the whole session.

   Example:



	A	B	C
1	Name	Total	Angry
2	עדיטל בן משה	0.169486239	0
3	יובל	0.04237155974	0.02118577987
4	איתי	0.1271146792	0.04237155974

3. Graph of emotion per timestamp for each child (The graph file is named after the name of the child).
   Example:



Conclusions

We created an application that assists both teachers and researchers with analyzing children's emotions during virtual classes. Teachers will most likely use the application in real time, while researchers will probably use it offline.
The application provides data about the emotions of the children that are participating in virtual class.
As the application is based on convolutional neural network which is not 100% accurate, sometimes the emotion that is displayed is not matching with the emotion recognized by human in video (confusion matrix can be found in previous work [4]).
During work we had to make some pivots: Instead of analyzing separate video streams, we had to analyze the video as an image. We dived into the world of image processing and data matching. Being open to changing our first solution helped us in solving the problem, and coming with a solution that hits the mark.

Future Work

1. Improve response time of the application.
2. Extend emotions and behaviours that are detected.
3. Consider implementing the initial solution since the last update in Zoom SDKs provides access to the raw video for all developers.
4. Make adaptations to other video conference platforms, for example Teams and Hangouts.
5. Extend application for analyzing adults emotions during video sessions.

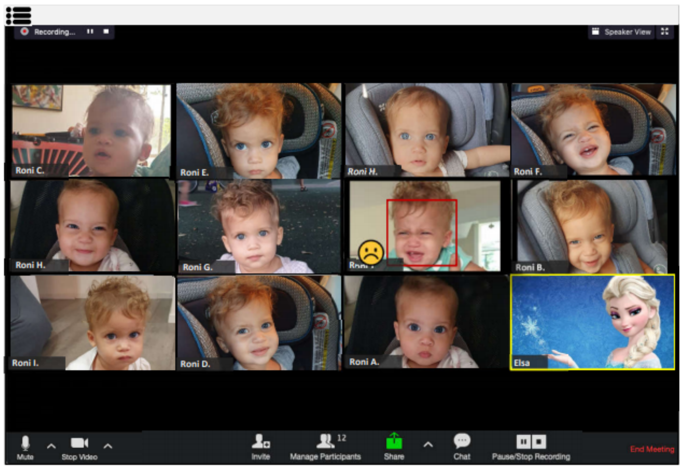
ZoomEmotion User Manual

1. **What does this application do?**

   This application meant to help teachers to recognize children with negative emotions during virtual classes (e.g. via Zoom).

   The application records your screen, processes the children's faces from its video and shows gestures on the children's faces when they express negative emotions.

   Here is an Example:

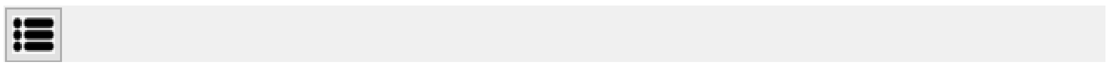


2. **Installing the Application**

   2.1. Extract the ZIP file 'ZoomEmotion.zip' into a place of your choice.

   2.2. Inside the directory (ZoomEmotion) that you extracted from the zip, you will find a file named 'ZoomEmotion.exe'. Double click it to start the app.

   It can take several seconds for the app to start. You will know it did when the menu bar (shown below) will appear on the top of your screen:



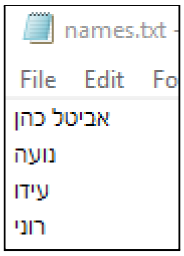
3. **Optional Configuration Setting Before Starting The Application**

   3.1. In case you have two (or more) screens, you can set the screen you want the app to work with. To do that, enter Settings/System/Display, press the screen you want and mark the box: "Make this my main display" (the last setting in the window).

   3.2. In order to help the application to track the names of the children, you can create a file named 'names.txt' in the directory where 'ZoomEmotion.exe' is located.

   The file should contain the names of the teacher and the children that participate in the session. Each name should be written in a different line. Adding this file will significantly help in getting an understandable log with the data that was collected during the session.

   Example of 'names.txt' file:


names.txt

File Edit Fo

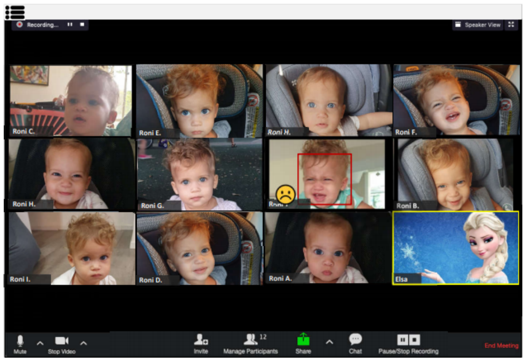
אביטל כהן

נועה

עידן

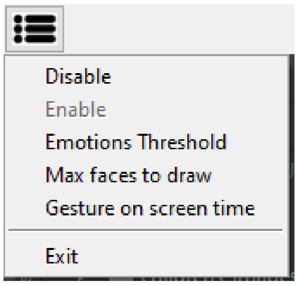
רוני

4. **Recordings Requirements**
   In order to use the application offline, the recording that is used must have black outline around the video frames, as seen in the example:



   When using Zoom's built-in recording of the participants gallery, the frames' outlin disappear. Therefore, you must use an external screen recording application so the outlines will be displayed in the recording as seen in real time.

5. **Configuration**
   5.1. There are several configurations you can change while the application is running.
   The configurations are located under the menu button:



   5.1.1. 'Enable' and 'Disable' are used to pause/resume the application run.
   5.1.2. 'Emotion Threshold' sets the threshold of certainty about whether or not the emotion was exhibited by the child (the value varies from 0 to 1).
   5.1.3. 'Max faces to draw' defines the max number of gestures that will be shown on screen at the same time.
   5.1.4. 'Gesture on screen time' defines how long the gesture will be displayed on the screen (It's not precise).
   5.1.5. 'Exit' closes the application.

   5.2. Default values:
   Emotion Threshold: 0.8
   Max faces to draw: 4
   Gesture on screen time: 5 sec

6. **Logs**

6.1. After closing the application (with the Exit button), the application will produce two differents logs and several graphs (they will be found under the directory 'logs'):

6.1.1. 'detailed log.csv' holds a table that for each timestamp indicates which children exhibit which emotion (if the child was not found or the emotion was not conclusive to be displayed, the cell will remain empty).

6.1.2. 'summary log.csv' summarizes the amount of time that each emotion appeared for each child, and how long (conclusively) the emotion was displayed during the whole session.

6.1.3. Graph of emotion per timestamp for each child (The graph file is named after the name of the child).

7. Notes

7.1. The application logic is relatively heavy. Therefore, there could be some screen freezing and slow response when pressing on buttons.

7.2. The app detects the names of the children and collects the data about the children's emotions according to their names. Hence, for best logging results, it is best if the children frames will be stable in place and size during the session.

7.3. Remember, the application records the screen directly. Ergo any window that covers the children frames will prevent the application from processing the hidden children emotions.

7.4. The logs are deleted from one run to another. Remember to save them in different directory if you need them. In addition, remember to close the logs files before starting a new run of the application.

8. Attachments

8.1. Emotions Bar:

☹️ - Sad

🤮 - Disgusted

😨 - Fearful

😮 - Surprised

😠 - Angry

8.2. [ZoomEmotion User Manual Slides]

References

[1] A Deep Learning Perspective on the Origin of Facial Expression, Ran Breuer, Ron Kimmel - Department of Computer Science Technion - Israel Institute of Technology Technion City, Haifa, Israel

[2] Langner, O., Dotsch, R., Bijlstra, G., Wigboldus, D.H.J., Hawk, S.T., & van Knippenberg, A. (2010). Presentation and validation of the Radboud Faces Database. Cognition & Emotion, 24(8), 1377— 1388. DOI: 10.1080/02699930903485076

[3] Crowdsourcing facial expressions for affective-interaction, Gonçalo Tavares, André Mourão and João Magalhães, Computer Vision and Image Understanding, Volume 147, June 2016, 102-113

[4] Facial Expression Recognition in Children GIP Project, Spring 2019, Dana Goghberg, Sapphire Elimelech, Moran Hait. Supervisors: Ron Kimmel, Tzipi Horowitz Kraus, Michal Zivan, Gary Mataev - Department of Computer Science Technion - Israel Institute of Technology Technion City, Haifa, Israel, Educational Neuroimaging Center, Israel Institute of Technology Technion City, Haifa, Israel.