

Computer Vision for Attendance and Emotion Analysis in School Settings

Sarah Deniz¹, Dakyung Lee¹, Grace Kurian², Lourdes Altamirano³, Darren Yee³, Michael Ferra³, Blake Hament⁴, Justin Zhan⁴, Laxmi Gewali⁴, Paul Oh⁴

Abstract—This paper presents facial detection and emotion analysis software developed by and for secondary students and teachers. The goal is to provide a tool that reduces the time teachers spend taking attendance while also collecting data that improves teaching practices. Disturbing current trends regarding school shootings motivated the inclusion of emotion recognition so that teachers are able to better monitor students' emotional states over time. This will be accomplished by providing teachers with early warning notifications when a student significantly deviates in a negative way from their characteristic emotional profile. This project was designed to save teachers time, help teachers better address student mental health needs, and motivate students and teachers to learn more computer science, computer vision, and machine learning as they use and modify the code in their own classrooms. Important takeaways from initial test results are that increasing training images increases the accuracy of the recognition software, and the farther away a face is from the camera, the higher the chances are that the face will be incorrectly recognized. The software tool is available for download at <https://github.com/ferrabacus/Digital-Class>.

I. INTRODUCTION

Computer vision is the field of science that deals with the automatic extraction, analysis, and understanding of useful information from a single image or a sequence of images [1]. Computer vision would enable schools to take attendance electronically with a facial recognition program. Not only will this save teachers precious time, it will also aid in educating teachers and their students about coding and the variety of applications possible with computer vision. An emotion analysis program would give school faculty and staff more information on students and early warning of significant shifts in emotional state. Although many schools have security cameras installed, their function is rarely anything more than recording video footage. These cameras can be used to implement computer vision in school settings. Teachers and staff can then use the data collected to monitor student engagement and emotion.

In light of recent devastating events surrounding American school shootings, there is an imminent need to improve upon the current security systems in place. Since the 2012 Sandy



Fig. 1. Facial detection in the classroom is challenging but affords the opportunity to collect valuable data on class attendance and emotion over time.

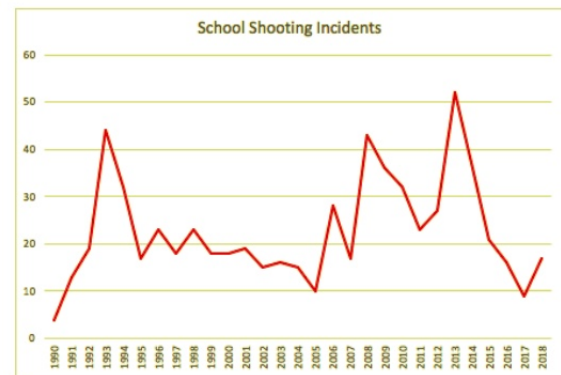


Fig. 2. School shootings from January 1990 to May 2018. Source: US Census Bureau.

Hook Elementary school shooting, there have been over 200 school shootings in the USA [2]. Fig. 2 graphically displays the number of shootings per month from January 1990 to May 2018. Many of the shooters at these sites are former or current students. Evidence from the events have shown that up to 60 percent of these shooters exhibited signs of depression or mental illness [3].

Bringing computer vision into American schools could serve as a solution to identifying the mental health state of students. Face detection, face recognition, and emotion analysis are some of the many features of computer vision and would help in addressing the issue of school security. This program will be able to send an early warning to the school's guidance counselor if a student is showing prolonged signs of sadness or anger. Through these features,

*This work was supported by the University of Nevada, Las Vegas, the National Science Foundation, and the US Department of Defense. Award Number: 170716

¹Army Educational Outreach Program UNITE 2018: denizzebra@gmail.com, demilee543@gmail.com

²University of Nevada, Las Vegas STEM 2018: graciek106@gmail.com

³Army Educational Outreach Program RET 2018: altamlf@nv.ccsd.net, yeeda@nv.ccsd.net, ferram2@nv.ccsd.net

⁴University of Nevada, Las Vegas: blakehament@gmail.com

teachers may utilize face recognition to save time spent on current attendance methods and familiarize themselves and their students with computer vision and computer learning software. With this software providing motivation and a launching point, teachers can build curriculum and investigatory projects for students on Computer Vision and Machine Learning. Facial recognition and emotion analysis can help teachers monitor the changes in their student behaviors to increase productivity and student achievement.

A. Educational Tool

Not only can this program be applied to the school's security system, it can also be used to expand students' and teachers' knowledge about coding. The interactive Scratch website that was created by MIT is a tool schools use to teach their students the basics of programming [4]. Similarly, this program can help educate students about face detection, recognition, and emotion analysis by providing a launching point for project-based learning. This program incorporates computer vision and machine learning, and students can explore other possible applications of these topics. Many schools already have computer science classes, and students can learn how this program works as they learn to code. Seeing a real-life application of code can inspire students to make their own modifications and versions of the project. School attendance and security are relevant topics for students, and the authors hope students will be motivated to further develop and customize the code as an open-source community. The code is available at <https://github.com/ferrabacus/Digital-Class>.

B. Literature Review

Python was selected as the primary software language for this projects as it is very human-readable and accessible to beginner and intermediate coders. The most important dependency for this program is Open Source Computer Vision Library (OpenCV) which is commonly utilized for facial recognition purposes [5]-[9]. OpenCV was used to implement proven facial detection and recognition methods like Haar features and Local Binary Pattern Histogram (LBPH) classification. Haar features are digital image features used in object recognition and face detection. First, the image taken from the video feed is made grayscale. Then, the program scans the face to compare the shadows and highlights on the image to the Haar features. Haar features are vital for the program to detect faces and an essential step in this facial recognition pipeline [10]-[11]. LBPH is a method used for facial recognition, along with EigenFaces and FisherFaces. LBPH is well-adapted for feature extraction because it examines the texture and structure of an image in small, local neighborhoods of pixels. The main focus of LBPH is on the details rather than the big picture, defining image features in relative terms that make the algorithm less sensitive to changes in lighting in training and test images [12]-[14]. This makes the program's ability to recognize a person more robust in varying lighting conditions.

Finally, in order to analyze student emotion, Microsoft Azure Face was used. Microsoft Azure is a growing collection of cloud services for building, deploying, and managing intelligent applications through a global network of data centers [15]. Although it has many different fields and applications, this project only utilized the emotion analysis feature of the FACE application programming interface (API). It is this team's hope that this software tool will be further developed to include proprietary ML algorithms for emotion recognition that will eventually replace the Azure Face API query. In the meantime, the Azure Face API accomplishes our emotion analysis.

In order to keep the code accessible to high school teachers and students who may be beginner or intermediate coders, the program balances the trade-off between implementing state-of-the-art algorithms and maintaining a software architecture that is friendly to users of all levels of coding experience. The OpenCV library was used because it offers many beginner-friendly tools and ways to incorporate proven computer vision algorithms. In future stages of development, more low-level control over classifiers and other machine learning algorithms used in facial detection and recognition will be introduced. OpenCV allows for students to get started using stock machine learning algorithms right away, while also introducing them to enough of the image processing and program architecture that they will be able to transition to lower-level control of the algorithms later on. This will be accomplished with the incorporation of popular machine learning libraries like TensorFlow and Keras.

II. METHODOLOGY

In order to properly be able to execute the project, we will need the code for face detection, face recognition and emotion analysis, as well as a computer to run the program on. This program has been successfully tested on an HP Envy x360 with Windows x64 i7 with 8GB memory as well as a MacBook Pro with iOS High Sierra i7 with 16GB memory. Python is the primary coding language, Microsoft Visual Studio Code and Atom Editor were used as the integrated development environments (IDEs).

A. Code Overview

The first step in the program is to receive either video footage or an image from the user. If a video is received, the program separates the video into individual frames to be processed as images. Afterwards, the program uses Haar features to detect the faces in the photo by cascading. Rectangular coordinates are specified around the corners of the face detected in the image. A rectangle is drawn around any faces that are detected. Next, the program cross-references the face in the image with all the faces in the training set. If there is a match, then the corresponding name will be displayed above of the rectangle drawn previously. If the confidence is greater than 200, and no suitable match is found, the message "ERROR: UNKNOWN FACE" is displayed on top of the rectangle in place of a name. Fig. 3. shows how the program displays the matched person's name with the



Fig. 3. After face detection, facial recognition is performed using LBPH classification, and an error score is computed.

```

“emotion”; {
  “anger”: 0.0,
  “contempt”: 0.003,
  “disgust”: 0.0,
  “fear”: 0.0,
  “happiness”: 0.153,
  “neutral”: 0.844,
  “sadness”: 0.0,
  “surprise”: 0.0,
},

```

Fig. 4. Emotion analysis outputs probability distribution for 8 emotions.

confidence score next to it. Though the official OpenCV term for this value is confidence score, it is misleading because a lower score implies higher accuracy. The confidence score is calculated by taking the difference or root mean squared distance between the classification model and an observed feature. Thus, if it is a close match, the score will be low, but if they deviate significantly, the score will be high. This paper interchangeably refers to the confidence score as an error score.

Once the face is matched to a name, the data collection process is initiated. If the program sees that person’s face, it marks them as “Present.” Otherwise, it will mark them as “Absent.” Furthermore, the program saves the data about the person’s overall emotions in the image. There are eight emotions the program can recognize: anger, contempt, disgust, fear, happiness, neutral, sadness, and surprise. Although all eight emotions are recognized, the program will only track the data for happiness, sadness, anger, and neutrality. Fig. 3 shows a sample image and Fig. 4 displays the output after the program has analyzed the emotions of the face shown in Fig. 3. After the data has been recorded in a CSV file, the program finds the average, and then the program will send the data via email to the addresses provided.

Notifications will be sent via email weekly to teachers and parents regarding the students’ presence and emotional state. Another notification will be sent to the school’s guidance counselor if the student is especially sad or mad for more than 4 consecutive days. Algorithm 1 is the pseudo code for the criteria for the notifications to be sent to the guidance

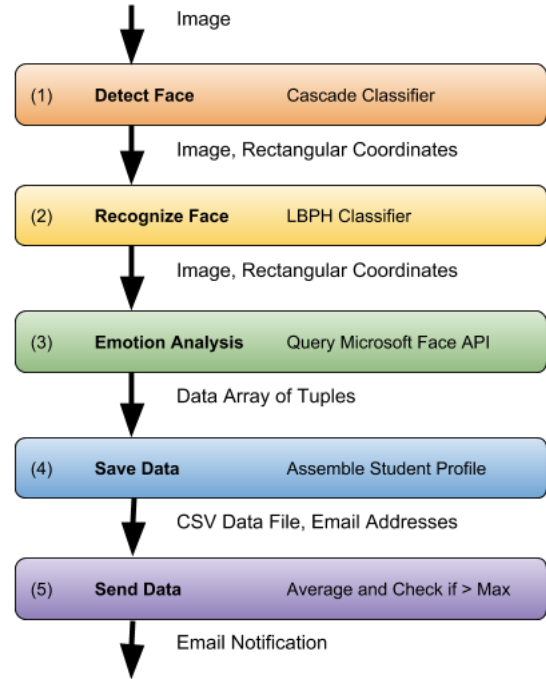


Fig. 5. A visual representation of the program modules with input and output at each level.

counselor as well as the averaging. Below is the standard equation for averaging data used for emotion analysis and profiles.

Algorithm 1 Algorithm for Guidance Counselor Notification:

Input: Array of 8 Emotion Averages

Output: Guidance Counselor Notification

Calculate 8 emotion averages and store in $emoAves$
 $i = 0$

for avg in $emoAves$ **do**

if ($avg > threshold$ **and** ($i = index_{anger}$ **or** $i = index_{sad}$)) **then**

 send notification to Guidance Counselor

end if

$i++$

end for

return

$$A = 1/n * (x_1 + x_2 + x_3 + \dots + x_n) \quad (1)$$

Fig. 5 displays a flowchart that explains the modules and methods of the program, as well as the input and output to each of module.

B. Testing

The changes in the error/confidence score were measured to test how different variables affect the program’s success rate. This score is a measurement of how closely an image

TABLE I
ERROR SCORE AND TRAINING DATA-SET SIZE

Number of Training Images	Avg Score For Correct Face Recognition	Avg Score For Incorrect Face Recognition
20	77.5	71.25
180	40.0	43.0

TABLE II
ERROR SCORE AND DISTANCE FROM CAMERA

Distance from Camera	20 Image Data-set	180 Image Data-set
60	77.5	40.0
120	115.0	61.8
180	135.0	81.50

matches the program’s classifier. A lower error/confidence score means that the program is more closely identifying the face in the image with a face from training data. By changing the number of training photos and the distance from the camera, the best possible conditions will be determined through various trials.

C. Data

The data in Table 1 is the average error/confidence score for the correct prediction when the face was 60 cm from the camera. Average correct error/confidence score tells us the average confidence score of when the code guessed the person correctly. The average incorrect error/confidence score tells us the average confidence score of when the code predicted or guessed the wrong person.

Table 2 compares the average error/confidence score of the program when the distance from the camera was altered. It also compares how the number of images in the training set affects the error score. A higher error score (or confidence score in OpenCV literature) means that the program is less sure of whose face is shown in the image. Having more training photos leads to lower error scores, which goes to show that an increase in the number of training photos is directly related to an increase in the software’s accuracy.

The Yale Face Database is a database of faces that exhibit certain emotions, as well as accessories such as glasses. We found the accuracy of our program, by analyzing the emotions of the faces in the Yale Face Database. The results, which are shown in Table 3, are nearly perfect. For Subject 2, it may have been difficult for the program to discern the subject’s sadness due to his mustache. Instead of saying that he was sad, the program said he was 90 percent neutral and

TABLE III
AGREEMENT (%) WITH YALE FACE DATABASE GROUND TRUTH

Trial	Happy	Sad	Neutral	Glasses
Trial 1	100	99	100	98
Trial 2	100	0	94	100

8 percent happy.

III. DISCUSSION

The data collected from testing demonstrates that the program will be more accurate by increasing the number of training images. Furthermore, when there are more pixels in an image, it takes the program longer to train. The testing has also demonstrated that the program will operate consistently despite change in distance when utilizing 180 images per person in the training data. This is especially beneficial if a school decides to implement the program while utilizing cameras located in hallways, offices, or larger classrooms.

Emotion recognition testing was also conducted by utilizing the Microsoft emotion API code and the Yale Face Database. Two subjects from the Yale Face Database were selected and each subject had four images consisting of happy, sad, neutral, and glasses categories. The confidence score of what each person appeared to be feeling was established by feeding these images to the program. According to the results, most of the subject’s categories were matched, however, the emotion of sadness was not recognized due to one of the subjects having a mustache. Due to the majority of the images being matched with the correct emotions, schools can expect that the majority of their students will have accurate data in their student emotion profiles. Data in Table III shows percent match with dataset ground truths.

A. Possible Issues

Distance from the camera may potentially have an effect on the level of confidence the software has when identifying a person. Failure to identify a person correctly could lead to potential issues such as incorrectly marking a student absent or matching the emotional state of a student to the incorrect student profile. Other potential problems the software may experience is not having enough training images to make accurate matches if a student makes changes to their appearance. For example, wearing glasses, makeup, or even changing hairstyles would have an impact in the confidence score when identifying a student.

Considering these issues, a possible solution might be to continue and add training images every time the software is being utilized. The program has demonstrated an increase in the confidence interval in identifying a person when the number of images a person has in their training set is expanded. Also, additional cameras in the classroom could potentially assist in identifying a person due to providing the software with different angles of a face.

Privacy is of special concern, especially in school environments. Currently the Family Educational Rights and Privacy Act (FERPA) protects the privacy of students and their records. Under FERPA, security video feed is allowed to be shared with parents, staff, and law enforcement without any violations [22]. However, the presented program is not recording video feed from the cameras. It creates a local database monitoring the students’ attendance and emotional states. The video feed used to analyze the student will not be available for anyone as it will not be saved. This data

is compiled into an array which is then transferred to a CSV file. This is how the student profiles are assembled. Notifications to parents and teachers can be modified to achieve desired frequency and notification content without ever requiring transmission of actual image frames.

IV. CONCLUSION

There have been other implementations of computer vision attendance in school settings, but this software tool is unique in that it can do face detection, face recognition, and emotion analysis – all at levels accessible to beginner and intermediate coders [16]-[18]. Although there are many potential applications of this program, the focus is in a school setting. On average, a teacher takes about 4.53 minutes to complete attendance and if it is assumed that a school has 180 school days with seven fifty minute classes, then each teacher spends 5,707 minutes yearly on attendance alone. In addition to being a viable time-saver, this software can be used to collect valuable data on student engagement and emotions [19]-[21]. This program can utilize the cameras in schools if they're already installed, and if not, they need to be purchased and installed.

Another possible future application of this program is to increase the security levels in schools. If an unknown face enters the building, then the facility's administration will receive a notification stating that there is an intruder, along with a photo of the person's face. Without much modification, the code can be used as a visitor log in addition to an attendance and emotion analysis system.

This software tool was designed by and for high school students and educators, with the help of university researchers. It is the authors' hope that other students, teachers, and researchers will join in further developing this open-source tool. As students download, implement, modify, and extend the code presented, they will develop knowledge and skills in computer vision, machine learning, and python programming while increasing the efficiency, quality, and safety of their own classrooms.

ACKNOWLEDGMENT

The researchers would like to extend special thanks to Dr. Justin Zhan, Dr. Paul Oh, Blake Hament, the Army Education Outreach Program (AEOP), the National Science Foundation, University of Nevada, Las Vegas, and Research Experiences for Teachers (RET). RET provides secondary students and teachers the opportunity to work with graduate students and professors on university-level research projects, and this publication is a direct outcome of that collaboration.

REFERENCES

- [1] "About the BMVA," What is computer vision?[Online]. Available: <http://www.bmva.org/visionoverview>. [Accessed: 02-Jul-2018].
- [2] A. E. Hurley-Hanson and C. M. Giannantonio, "The Sandy Hook Elementary School shootings," *Extreme Leadership*, pp. 224–236, Feb. 2018.
- [3] J. M. Metzl and K. T. MacLeish, *Advances in pediatrics.*, Feb-2015. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4318286/>. [Accessed: 02-Jul-2018].
- [4] "Scratch - About," Scratch - Imagine, Program, Share. [Online]. Available: <https://scratch.mit.edu/about>. [Accessed: 03-Jul-2018].
- [5] "Face Detection using OpenCV and Python.," SuperDataScience - Big Data — Analytics Careers — Mentors — Success, 14-Jul-2017. [Online]. Available: <https://www.superdatascience.com/opencv-face-detection/>. [Accessed: 02-Jul-2018].
- [6] "Gender Classification with OpenCV," OpenCV: Image Thresholding. [Online]. Available: https://docs.opencv.org/2.4/modules/contrib/doc/facerec/tutorial/facerec_gender_classification.html#fisherfaces-for-gender-classification. [Accessed: 02-Jul-2018].
- [7] "OpenCV: How to Use Background Subtraction Methods," OpenCV: Image Thresholding. [Online]. Available: https://docs.opencv.org/master/d1/dc5/tutorial_background_subtraction.html. [Accessed: 02-Jul-2018].
- [8] "OpenCV: Object Detection (objdetect module)," OpenCV: Image Thresholding. [Online]. Available: https://docs.opencv.org/master/d2/d64/tutorial_table_of_content_objdetect.html. [Accessed: 02-Jul-2018].
- [9] "OpenCV: Similarity check (PNSR and SSIM) on the GPU," OpenCV: Image Thresholding. [Online]. Available: https://docs.opencv.org/master/dd/d3d/tutorial_gpu_basics_similarity.html. [Accessed: 02-Jul-2018].
- [10] "Training Haar-cascade," Gagan, 28-Jun-2018. [Online]. Available: <https://singhaganpreet.wordpress.com/2012/10/14/training-haar-cascade/>. [Accessed: 03-Jul-2018].
- [11] M.-T. Pham and T.-J. Cham, "Fast training and selection of Haar features using statistics in boosting-based face detection," 2007 IEEE 11th International Conference on Computer Vision, 2007.
- [12] "Enhanced Human Face Recognition Using LBPH Descriptor, Multi-KNN, and Back-Propagation Neural Network - IEEE Journals & Magazine," Design and implementation of autonomous vehicle valet parking system - IEEE Conference Publication, 10-Apr-2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8334532/>. [Accessed: 03-Jul-2018].
- [13] "Identification system from motion pictures: LBPH application - IEEE Conference Publication," Design and implementation of autonomous vehicle valet parking system - IEEE Conference Publication, 02-Nov-2017. [Online]. Available: <https://ieeexplore.ieee.org/document/8093546/>. [Accessed: 04-Jul-2018].
- [14] H. Abdulsamet and T. Olcay, "Identification system from motion pictures: LBPH application," 2017 International Conference on Computer S
- [15] Real Python, "Face Recognition with Python, in Under 25 Lines of Code – Real Python," Real Python, 11-Jun-2018. [Online]. Available: <https://realpython.com/face-recognition-with-python/#opencv>. [Accessed: 02-Jul-2018].
- [16] K. Putha, R. Hartanto, and R. Hidayat, "A review paper on attendance marking system based on face recognition," 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2017.
- [17] A. Sarrafzadeh, H. Hosseini, C. Fan, and S. Overmyer, "Facial expression analysis for estimating learners emotional state in intelligent tutoring systems," Proceedings 3rd IEEE International Conference on Advanced Technologies, 2003.
- [18] J. Whitehill, Z. Serpell, Y.-C. Lin, A. Foster, and J. R. Movellan, "The Faces of Engagement: Automatic Recognition of Student Engagement from Facial Expressions," *IEEE Transactions on Affective Computing*, vol. 5, no. 1, pp. 86–98, Jan. 2014.
- [19] S. Sharma and W. Chen, "Multi-user VR Classroom with 3D Interaction and Real-Time Motion Detection," 2014 International Conference on Computational Science and Computational Intelligence, 2014.
- [20] J. Whitehill, M. Bartlett, and J. Movellan, "Automatic facial expression recognition for intelligent tutoring systems," 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008.cience and Engineering (UBMK), 2017.
- [21] "Emotion API - Emotion Detector — Microsoft Azure." Service Level Agreements Summary — Microsoft Azure, azure.microsoft.com/en-us/services/cognitive-services/emotion/.
- [22] "Balancing Student Privacy and School Safety: A Guide to the Family Educational Rights and Privacy Act for Elementary and Secondary Schools - PFCO," Home, 30-Oct-2007. [Online]. Available: <https://www2.ed.gov/policy/gen/guid/fpco/brochures/elsec.html>. [Accessed: 02-Jul-2018].