Adaptive Learning Experience Through Computer Vision

Ethan Tam

Sunny Hills High School c/o 2025, CA, USA

----------***----------

Abstract - *With technology usage increasing worldwide due to rapid advancements and technological growth, students of the younger generation tend to lose focus in a short time span. Furthermore, unequal access to education is a prevalent issue seen throughout many areas of the world where poor standards of living and impoverished schools are incapable of providing students with an education that fits their needs.*

This artificial intelligence (AI) project and app utilizes computer vision, a field of AI, to create a system that combines two facial recognition models: one for emotion detection and one for tracking gaze direction. With the implementation of a logic function, the system determines a student's level of focus through analyzing data from their facial features in an uploaded video. Finally, an output indicating whether or not the student is at the appropriate level in their adaptive lesson is returned, allowing any user with a cellular device to have access to a tailored education.

1. INTRODUCTION

As advancements in AI and machine learning (ML) quickly arise, education is a major industry that has potential uses for this technology. Students of all ages are affected by the implementation of AI, whether it is a tool that teachers utilize to teach lessons and grade assignments or something students use to gather information and conduct research. Either way, AI provides numerous possibilities in completely changing the education of the next generation.

One concept that AI has the opportunity to improve is in adaptive learning. Creating an app or system that can accurately distinguish a student's focus level would provide a useful opportunity for students who might not have access to education that fits their standards. Examples of this situation might include schools that do not have enough educators to teach enough grade levels or an area that simply does not have enough schools for a growing young population. The combination of two AI models could also begin work on combining other ML algorithms that can improve education.

Through research, the goal is to create a system that utilizes two computer vision models, one for emotion detection and one for gaze tracking, that will gather data to feed into a logic function which will determine a certain level of focus, adjusting the student's next level in a module or lesson. The data used to test the system are self-generated videos labeled to one of three levels of focus and will be analyzed as to its accuracy in determining the correct level for each video.

2. WHAT IS COMPUTER VISION?

Computer vision is a field of AI that trains computers to understand, interpret, and gather information from the visual world of images and videos as a human would. This digital technology can be seen in various applications in the real world from object detection in self-driving vehicles to disease identification in medical imaging. With so many advancements in the field, however, it is still important to understand the history of the concept.

Early research on computer vision started in the late 1950s as scientists conducted biological experiments to understand how the neural system in living organisms processed visual data, developing the base of artificial visual systems [7]. Over the next two decades, computational approaches for processing visual data developed from the growth of ML and neural networks, contributing to major advances in image classification, object detection, and image segmentation [7]. Nowadays, computer vision systems are integrated in a variety of devices from within our cellular device's facial recognition systems to satellites in outer space.

Within this field of computer vision, the two individual models that were developed and conjoined, emotion detection and gaze tracking, which although both in the

same discipline, still have various differences in terms of algorithms used, previous research and findings, as well as functions and purpose which need to be understood before diving into the programming.

2.1 Convolution Neural Networks (CNNs)

Detecting human emotion and analyzing gaze direction have very different algorithms and programming required in order to solve, however they both require facial recognition before any classification occurs. To tackle the various processes that need to be applied to data with numerous factors that may affect the results, convolution neural networks (CNNs) are used to solve.

A CNN is a deep learning approach that is used to solve a wide variety of complex problems because of its ability to overcome the limitations of traditional ML approaches [5]. In ML, a CNN is a type of neural network (NN), an algorithm that many AI fields, including computer vision, utilize to recognize complex patterns. A NN consists of many neurons, an artificial value calculated by the multiplication of inputs and weights and the addition of biases. These neurons are stacked into layers of different sizes which are then organized into layers as shown in Figure 1. Most NNs consist of an input layer, various hidden layers, and a single output layer.

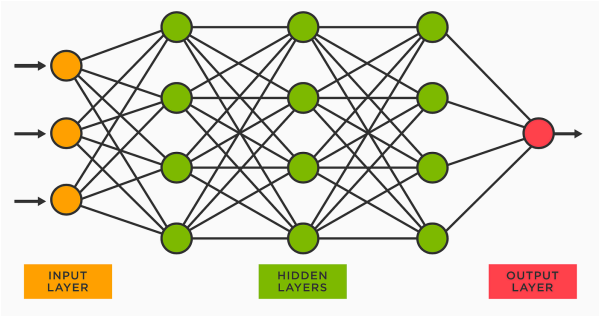

Figure 1: Schematic diagram of a simple representation of a neural network's input, hidden, and output layers [1]

The computer starts with an initial set of random weights and biases for the neurons and makes a prediction as to the correct answer for a certain task and passes through the program. It then uses a loss function that compares the prediction to the actual answer and changes the weights and biases depending on the result.

CNNs take it a step further with convolutional and pooling layers. Convolutional layers provide the foundation for CNNs, as they use kernels, links between previous layers and convolutional layers represented in numbers, to extract features that distinguish visual data from one another [3]. To perform a convolution, CNNs iterate through an image and multiply a corresponding element in a kernel or (as seen in Figure 2) convolution filter, to create an output layer that is more representative of the image features.

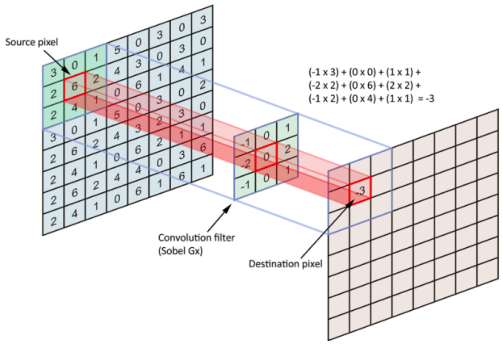

Figure 2: Visual representation of convolutional filter/kernel multiplication [4]

After the convolutional layer is applied, another unique layer called the "pooling layer" is used to reduce the spatial size of the network which enhances the computational efficiency of a system and also prevents overfitting, or the unintentional memorization of training data, leading to inaccurate performance on real data. There are various types of pooling, one of the most popular for CNNs being Max-Pooling which is used in the Tiny VGG architecture, including a 2x2 kernel that discards 75% of activations [3]. Figure 3 below provides a visual display of how pooling layers work; this one takes the average of a certain area while others may take the maximum value

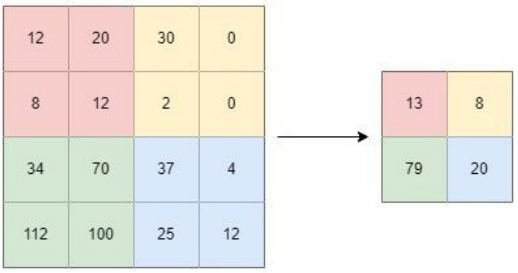

Figure 3: Example of a pooling layer, image takes average of each sector for a smaller output [8]

2.2 Emotion Detection

Detecting emotion is an ongoing topic in AI research that has proven to be challenging as there are a plethora of factors that need to be considered in the process. In order to recognize emotion through facial expression, a system must first detect where the user's face is, perform feature extraction, go through a recognition process, and then determine the label of emotion [6]. These complex processes, combined with the many factors such as background and facial features, are obstacles that programmers use convolutional neural networks (CNNs) to solve.

While research on using CNNs to predict emotion has been previously studied, a major limitation occurred in the accuracy of models because of the scarcity of labeled data. Additionally, the majority of emotion detection AI models have been tested on image data in databases like *Extended Cohn–Kanade database, Japanese Female Facial Expression Database,* and *Binghamton University 3D Facial Expression database* [2] rather than videos.

2.3 Gaze Tracking

Eye-tracking or gaze-tracking detection has also been previously studied, however the different types of models that exist range from predictive eye-tracking to optional eye-tracking. The main purpose of such methods is to assess visual attention and focus through information such as where a person is looking, how long they are looking for, and what they are looking at. When it comes to choosing which eye-tracking technology is the most ideal to use, it truly depends on the scenario, but traditional eye-tracking has proven to have the highest performance accuracy while predictive eye tracking is more flexible data-wise. [11]

As for practical usage in the educational field, eye-tracking has been employed to extract information on teacher's and student's cognitive load in different class experiences as well as capturing attention of individual learners to investigate underlying mechanisms for positive learning outcome. [10]

3.0 METHODOLOGY

The following project was programmed using Google Colaboratory and connects to the user's Google Drive to access uploaded videos to input into the system. Installed dependencies include deepface, tf_keras, open-cv python, dlib, cv2, an OpenCV library in Python that provides access to functions for image processing; NumPy, a library providing flexibility to create multidimensional array objects and computer mathematical operations; and DeepFace, a Python library used to perform facial recognition and interpret facial attributes and features.

3.1 Datasets

The dataset used for this system consists of 20 self-generated and recorded videos shown below in Figure 4, acting out different emotions and gaze directions with different conditions such as lighting, distance from the camera, etc. Although any ML model ideally trains on a larger dataset, the dataset created was used as testing data as the emotion detection and gaze tracking models were pre-trained on their respective datasets. The videos were of different lengths and some consisted of exaggerated emotions and gaze to test for the sensitivity of the models. Ideally, the system would be tested on more people with a larger dataset of different background and other crucial factors that may impact the accuracy of the model.

Dataset 1 (Attentive)	watching a video, taking notes, reading text, happy/interested
Dataset 2 (Distracted)	looking at phone, looking around, falling asleep, confused/annoyed
Dataset 3 (Daytime)	neutral, mad, happy, testing gaze distance
Dataset 4 (Nighttime)	neutral, mad, happy, testing gaze distance
Dataset 5 (Nightlight)	looking at phone, neutral, mad, happy

Figure 4: Datasets created with four videos filmed within each category

3.2 Emotion Detection Model

Before initiating the emotion detection code, the Haar cascade classifier, or Viola-Jones face detection framework, is loaded [9]. The ML algorithm helps to detect objects in images, in the case of this system, the human face and its various distinct features. The function then converts each frame into grayscale then back to an RGB frame to eliminate computational complexities during image processing

Once the frame is ready for classification, the model first finds the face's region of interest (ROI) within a detected frame. If and when a face is detected, emotion detection is then able to be performed on the ROI. The model utilized includes six different possible emotions: happy, sad, angry, neutral, fear, and disgust. A series of weights is applied to the result and the dominant emotion is stored in a dictionary that keeps track of the most prominent emotion in each frame to later gather a total amount of each. Finally, the frame is outputted with the detected face outlined and labeled with the predicted emotion as seen below in Figure 5.

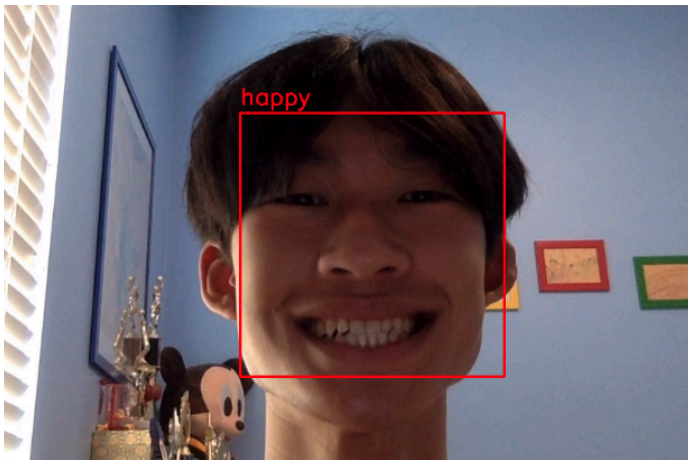

Figure 5: Frame output with "happy" emotion detected, Dataset 3 - happy

3.3 Gaze Tracking Model

The gaze tracking model implemented utilizes the GazeTracking class to apply the gaze direction inspection onto each individual frame. The concept used is simple with a pre-programmed algorithm to determine the direction (either left, right, or center) in which the user's gaze is directed. As facial recognition had already been applied before the emotion detection function, the algorithm for detecting gaze can be directly implemented. Similarly to the emotion detection function, the deleted direction is stored in a separate dictionary that keeps track of the amount of each direction gaze is detected. The frame is also further edited by overlaying a text indicating the location of the pupils and a label of the predicted gaze direction as seen in Figure 6 below while keeping the emotion detection output.

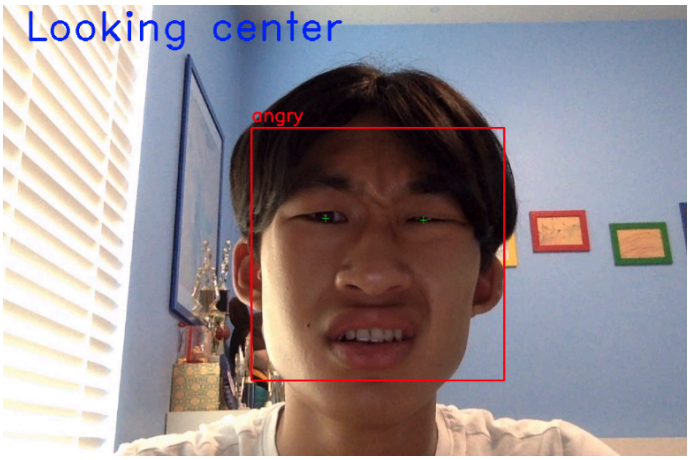

Figure 6: Frame output with "angry" emotion detected, center gaze direction, and pupil location highlighted, Dataset 2 - confused/annoyed

3.3 Logic Function Assessing Focus

To complete the system, a method needs to be implemented to determine the student's overall level of focus based on their dominant emotion and gaze direction detected. In order to program such a logic function, the level of focus based on emotion must first be analyzed. In this system, if the dominant emotion is happy or neutral and is at least a scale of 1.1 times greater than the second dominant emotion, then the emotion is considered focused. Next, the gaze is programmed to be focused if the number of center directions counted is greater than left and right directions. Finally, the overall level of focus is

determined based on the focus from the two functions as seen in Figure 7.

emotion is focused AND gaze is focused	High focus → move on to the next higher level
emotion is focused OR gaze is focused	Somewhat attentive → remain on current level
emotion is NOT focused AND gaze is NOT focused	Distracted → move down to the next lower level

Figure 7: Logic function determining overall level of focus and next action in system/app

4.0 RESULTS

Overall, the system achieved an accuracy of 85% on determining the correct level of focus on the dataset of 20 videos with 17 out of the 20 correctly predicted. The emotion detection model was highly accurate in predicting the correct dominant emotion, and the detection rate of when a face was present was also accurate after adjusting hyperparameters. One limitation to the model, however, is the dependence on the user's resting face. Especially for the intended use of this system, a student's facial expressions during a lesson or module will not likely be exaggerated as in Figure 8 below. Therefore, the emotion detection model may detect anger or sadness when the student is in reality, just focused on their work.

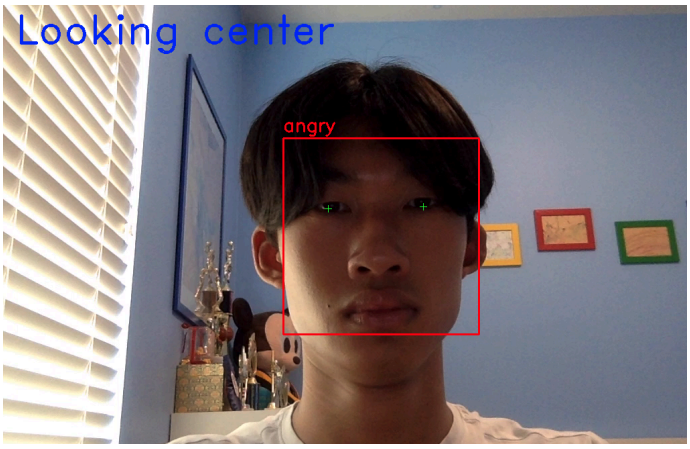

Figure 8: Incorrect detection of emotion when focused, Dataset 1 - watching video

The gaze tracking model was accurate as well in predicting the gaze direction, however the detection rate of a face being present was low compared to the emotion detection model. The gaze tracking accuracy proved to be highly dependent on external factors such as lighting and distance of the face from the camera. The variability at similar distances points to a very sensitive architecture. Furthermore, the algorithm appeared to track the pupils in relation to the rest of the eye socket rather than the direction the eyes are looking in relation to the camera and screen as seen below in Figure 9.

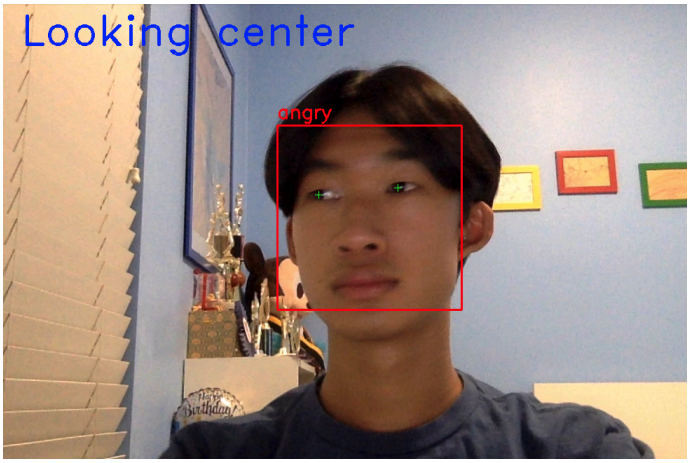

Figure 9: Incorrect detection of gaze direction, Dataset 4 - testing gaze distance

Despite the discussed limitations, which can be overcome by a finer model tuning, the overall architecture presented in this paper proved to be promising. This project also proves that such technology can be applied to the field of education and that the potential for AI uses in education is high. Future improvements to the system include analyzing live video feed instead of uploaded videos which would increase efficiency but require more resources.

ACKNOWLEDGEMENTS

I would like to thank the Github user "manish-9245" for their code repository, *Facial-Emotion-Recognition-using-OpenCV-and-Deepface*, and the Github user "MrMalik" for their code repository, *Gaze Tracking*.

I would also like to thank my mentor, Guido Andreassi, who made this work possible. His guidance and advice throughout the programming of models to conducting and analyzing research carried me through the stages of my project.

REFERENCES

[1] Amzhao. (2023, February 5). *Quantum Neural Networks*. MIT 6.s089–Intro to Quantum Computing. https://medium.com/mit-6-s089-intro-to-quantum-computing/quantum-neural-networks-7b5bc469d984

[2] Boesch, G. (2021, September 26). *AI Emotion and Sentiment ANalysis With Computer Vision in 2022*. Visio.ai. https://visio.ai/deep-learning/visual-emotion-ai-recognition/

[3] *CNN Explainer*. (n.d.). Poloclub.github.io. https://poloclub.github.io/cnn-explainer/

[4] Costa, V. C. (2019, April 17). *Understanding the Structure of a CNN - Vinicius C. Costa - Medium*. Medium; Medium. https://viniciuscantocosta.medium.com/understanding-the-structure-of-a-cnn-b220148e2ac4

[5] Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual Understanding of Convolutional Neural Network - A Deep Learning Approach. *Procedia Computer Science, 132,* 670-688.

[6] Muhamad, M., Widjaja, K. G., Adinata, M.F., & Anggreainy, M.S. (2021). Recognizing Human Emotion Using Computer Vision. *2021 2nd International Conference on Artificial Intelligence and Data Science (AiDAS).*

[7] Muiruri, D. (2023). *History of Computing - Computer Vision*.

[8] Nanos, G., Aibin, M., (2024). Neural Networks: Pooling Layers. *Baeldung CS*.

[9] OpenCV. (n.d.). *OpenCV; Cascade Classifier*. Docs.opencv.org. https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

[10] Sharma, K., Giannakos, M., & Dillenbourg, P. (2020). Eye-tracking and artificial intelligence to enhance motivation and learning. *Smart Learning Environments, 7*(1).

[11] Slivka, M. (2020, November 20). *Predictive Eye Tracking vs Regular Eye Tracking | Attention Insight*. https://attentioninsight.com/eye-tracking-vs-predictive-eye-tracking/