Health and Safety Monitoring for the Modern Workspace

Kotla Deekshith Reddy
School of Computing and Augmented Intelligence
Arizona State University
Tempe, AZ USA
dkotla@asu.edu

Sathvika Talakanti
School of Computing and Augmented Intelligence
Arizona State University
Tempe, AZ USA
stalakan@asu.edu

Mogileeswar Reddy
School of Computing and Augmented Intelligence
Arizona State University
Tempe, AZ USA
mmorramr@asu.edu

Rohith Kumar Reddy Bathula
School of Computing and Augmented Intelligence
Arizona State University
Tempe, AZ USA
rbathul1@asu.edu

ABSTRACT

This project aims to develop a real-time monitoring system that uses computer vision to identify and warn individuals about potential health and safety risks in various scenarios. Our proposed solution uses convolutional neural networks (CNNs) with stochastic gradient descent (SGD) to integrate sub-features such as emotion detection, heart attack detection, and eye detection into a single model. We trained our models on datasets of images with normalized grayscale values and utilized pickle files for efficient data storage and manipulation. Our evaluation metrics include accuracy, F1 score, precision, and real-time evaluation using the user's webcam video feed. We have achieved an accuracy of 96.8%, 67.8%, and 71.3% respectively on the eye detection, emotion detection, and heart attack detection datasets. Our findings demonstrate the potential for our system to assist individuals in maintaining their health and well-being, preventing mishaps and injuries, and enhancing their overall quality of life.

KEYWORDS

Keras, Numpy, Pandas, TensorFlow, CNN, SGD, haar cascade classifier, face detection, emotion detection, eye detection, heart attack detection, grayscale conversion.

1. INTRODUCTION

The ever-evolving modern lifestyle exposes individuals to a variety of health and safety hazards, be it the sedentary work culture, long commutes, or the recent shift to online learning and remote work. While these changes offer flexibility and convenience, they come with their own set of challenges. Prolonged hours of sitting in front of a computer screen for work or education can lead to poor posture, eye strain, and mental fatigue, adversely affecting the individual's overall health and well-being.To tackle these issues and enhance the health and safety outcomes for people and organizations, there is a growing need for real-time monitoring systems that can detect and warn individuals about potential health risks.

To mitigate these challenges, we propose a real-time monitoring system that uses computer vision to detect and alert users to potential health and safety risks. The system analyzes the user's body posture, eye movements, and facial expressions to detect signs of discomfort, fatigue, and distraction. Based on this analysis, the system provides feedback to the user to adjust their posture or take a break and also helps to re-engage or refocus their attention.

In this paper, we describe the implementation of the proposed system using Convolution Neural Networks (CNN) with Stochastic Gradient Descent (SGD) for eye detection, emotion detection, and heart attack detection models. We trained the model with a real-time image dataset and implemented it using the user's camera, taking real-time frames for prediction. To evaluate the system's effectiveness, we compared classic metrics such as accuracy against other baseline approaches.

Furthermore, we demonstrate the real-time evaluation of the system, where our script displays the user's webcam video feed and overlays text on the video frame to show the user's body posture score, emotion score, and eye state score. Additionally, we implemented a script that outputs a sound alert when the user's score falls below a certain threshold, indicating poor posture, and displays the user's emotion. Our results demonstrate that the proposed system has the potential to improve health and safety by detecting potential health risks in real time and promoting a more engaging and positive workplace experience.

The rest of the paper is organized as follows. Section 2 describes related work and highlights the research gap that the proposed system aims to address. Section 3 provides a detailed description of the dataset used for this model. Section 4 describes the proposed system, including the system architecture and implementation details. Section 5 presents the results of our experiments and Section 6 evaluates the system's effectiveness. Finally, Section 7 concludes the paper and highlights the future work needed to further improve the proposed system.

2. RELATED WORK

Microexpression detection is one way that deep learning is used in health and safety monitoring. With a 65.83% accuracy rate, Guo et al. [2] suggested a technique for recognizing microexpressions based on local binary patterns from three orthogonal planes (LBP-TOP) and the closest neighbor method. While LBP-TOP was quick at identifying faces and other landmarks, it produced a lot of misleading findings when used to classify emotions. The CASME II dataset was used by Wang et al. [3] to identify subtle emotions with an accuracy of up to 75.30%. They amplified face emotions using Eulerian Video Magnification (EVM), using LBP-TOP features, and SVM classifiers to identify them. Because of its high accuracy rate, CNN has become the best approach for identifying emotions. In order to recognize tiny facial expressions that last only a few milliseconds, Ayyalasomayajula et al. [1] combine face detection with Eulerian Video Magnification (EVM) amplification of microexpressions. This technique is followed by classification using a separately trained CNN model. These studies show how computer vision can be used to monitor emotional and mental health in the workplace by using the ability to recognize tiny changes in human expressions.

User attention monitoring using deep learning is another area of research in health and safety. An additional comparative analysis of the BiLRCN model and other cutting-edge models has to be done after Ma et al.'s [4] proposal of a bidirectional long-term recurrent convolutional networks strategy for online learning engagement recognition. Eye closure ratio and eye aspect ratio were suggested by Mehta et al. [6] as a real-time driver drowsiness detection method. Using either the Naive Bayes or SVM classifier, or in the case of Random Forest, it was also shown to be 84% accurate, the Dlib package is used to detect the driver's face, and a threshold value is applied to determine drowsiness. Rafid et al.'s [5] real-time driver fatigue detection model surpassed other existing approaches by reaching real-time accuracy of 94.5%. It uses the CenterFace algorithm for facial recognition and the Haar-Classifier for eye feature extraction.

The monitoring of medical issues like heart attacks has also made use of deep learning techniques. Deep learning can be used to track ergonomic posture, as demonstrated by a method described by Newell et al. [9] that uses layered hourglass networks for human stance estimate. In order to detect unstructured human activity using RGBD photos, Sung et al. [8] suggested a method, which highlights the promise of deep learning in this regard. Convolutional neural networks were used by Rojas-Albarracn et al. [7] to construct a system that accurately detected heart attacks in color photos by leveraging references from Newell et al. [9] and Huang et al. [8]. This study shows the capability of deep learning in the early detection of medical emergencies, which can be used to avoid workplace mishaps or injuries.

In the end, there are numerous different strategies available for carrying out Health and Safety Monitoring in contemporary workplaces. The works of Rojas-Albarracn et al. [7], Rafid et al. [5], and Ayyalasomayajula et al. [1] can be used as examples to develop a unified model because they use convolutional neural networks and color image datasets to obtain high accuracy.

3. DATASET

We are using three datasets to train our model: fer2013 for emotion detection, a heart attack dataset [7] for heart attack detection, and an eye dataset for eye detection. The fer2013 dataset contains 35,887 grayscale images of faces with various emotional expressions, while the heart attack dataset contains 1,520 images manually classified into two classes - non-infarction and possible infarction - and augmented to generate 20 different images for each original picture using six transformations. The eye dataset consists of images of human eyes captured under different lighting conditions and angles, with images of open eyes and images of closed eyes.

Detection	Instances	Features
Emotion	35,887	48x48 pixels
Eye	4850	24x24 pixels
Heart attack	1520(before preprocessing)	48x48 pixels

Table 1. Datasets Description

3.1 Visualization of classes in each dataset:

Figure 1 shows the distribution of classes in the eye dataset. The dataset has 2386 instances with closed eye class and 2464 instances with open eye class.

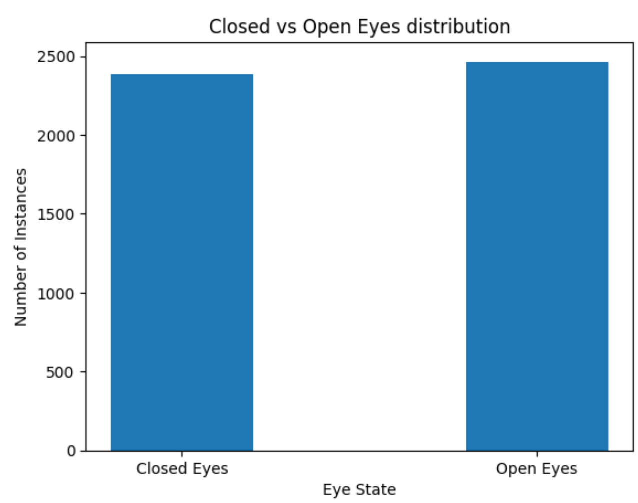

Figure 1: Eye dataset distribution

Figure 2 shows the distribution of the Emotion dataset, which comprises seven classes, namely Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. The instances for each class are as follows: Angry: 4953, Disgust: 547, Fear: 5121, Happy: 8989, Sad: 6077, Surprise: 4002, and Neutral: 6198.

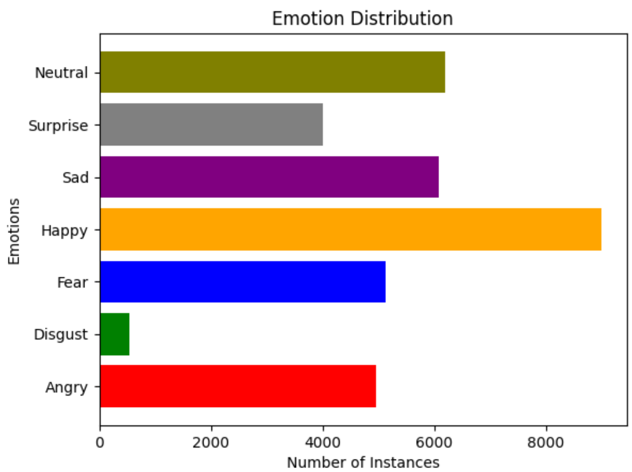

Figure 2: Emotion dataset distribution

Figure 3 illustrates the distribution of the Heart Attack dataset, which contains a total of 1520 images. The dataset is divided into two classes: Infarct and Non-Infarct. The Infarct class comprises 760 images that depict instances of heart attack, while the Non-Infarct class includes the remaining 760 images that do not show any signs of heart attack.

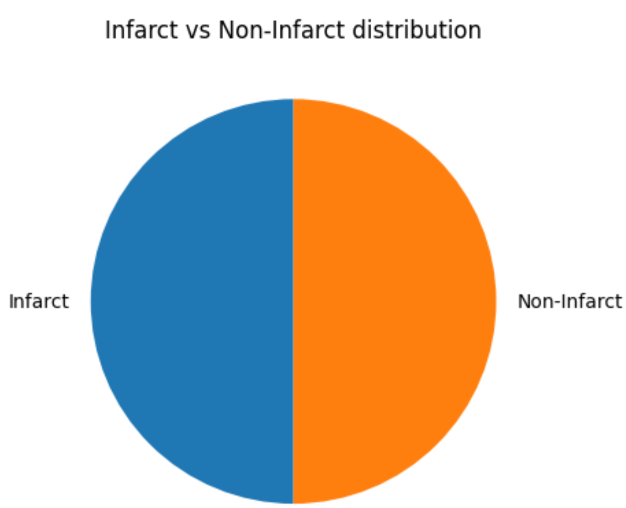

Figure 3: Heart Attack dataset distribution

To train each model, each dataset is divided into three parts: 70% for training, 15% for validation, and 15% for testing.

4. APPROACH

In the following section, we will discuss the preprocessing steps taken for the datasets and the architecture of the code. We then explain how we utilize our generated individual models to create a unified model. Finally, we will discuss how we test the model in real-time using Haar Cascade Classifier, by extracting frames from live video.

4.1 Data preprocessing

We used the Keras library's ImageDataGenerator class to apply various image augmentations to our datasets. These augmentations include random rotations, shifts, zooming, and flipping of the images. By using the flow method, we generated batches of augmented images and their corresponding labels on-the-fly. This allowed us to increase the size of our dataset and improve the model's robustness to variations in the input data, which can ultimately lead to better performance.

Emotion dataset considered for the model exhibits class imbalances, meaning that some classes have a greater number of samples than others. To mitigate this issue, we have uniformly sampled a fixed number of instances from each class but we didn't find much difference in the predictions. This approach usually helps to mitigate any inherent biases present in the data, thereby enhancing the fairness and robustness of our analysis.

4.2 Architecture

The general design of each of our CNN models starts with a number of convolutional blocks that vary in the amount of filters and activation functions they have, then a dropout layer and a flattening layer. To get at the final model, which was created to carry out the necessary classification while considering the defined size of the input images, many experiments were conducted using various layer combinations and configurations. Each convolutional block typically starts with a convolution layer to extract broad features from the input image, then moves on to a max pooling layer to minimize the number of parameters in the network. The inclusion of a dropout layer in the network's middle, following the convolution blocks, helps to prevent overfitting because of the scant amount of data. In order to send the values to the conventional neuron layers, a flattened layer is used to transform the convolutional layers' 2D architecture into a vector. Several layers of conventional neurons typically make up the network's termination, producing the output of forward propagation to a function, and the function is classification-dependent.The number of layers, filters, blocks, and activation functions used depends on the specific configuration of the model, as determined through testing and experimentation. SGD (Stochastic Gradient Descent) is also used to minimize the loss function of the neural network by iteratively adjusting the parameters (weights and biases) of the network in the direction of the steepest descent of the gradient.

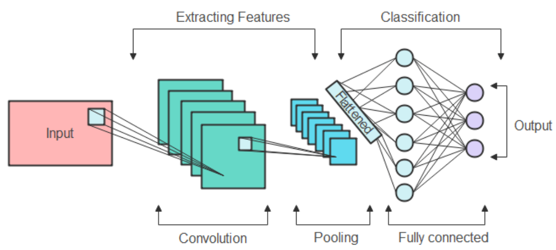

Figure 4. Building blocks of a CNN [11]

Eye Detection Model: The CNN model has a total of 3 blocks of Convolutional layers, with each block consisting of 2 Conv2D layers and an activation function (ReLU) followed by a MaxPooling2D layer with a pool size of (2,2) for down-sampling. Each Conv2D layer has a kernel size of (3,3) and a stride of 1. The first block has 32 filters, the second has 64 filters, and the third has 128 filters. Dropout layers are also present after each block with a rate of 0.25 to prevent overfitting. After the convolutional layers, the output is flattened into a 1D array using the Flatten() layer. The flattened output is then fed into a fully connected layer with 512 units, followed by an activation function (ReLU) and a dropout layer with a rate of 0.5 to further prevent overfitting.

Finally, the output is passed through another fully connected layer with the number of units equal to the number of classes 2 in the dataset. The activation function used in this output layer is sigmoid because the problem is a binary classification task. The model is trained using the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01, a momentum of 0.9, and a decay rate of 1e-6. The SGD optimizer is chosen to speed up the convergence of the model. Nesterov momentum is used to improve the convergence speed and accuracy of the model.

Heart Attack Prediction Model: The CNN model has 4 blocks of Convolutional layers, each block followed by a MaxPooling2D layer with a pool size of (2,2) for down-sampling. Each Conv2D layer has a kernel size of (3,3) and a ReLU activation function. The first Conv2D layer has 32 filters, the second has 64 filters, the third has 128 filters, and the fourth has 256 filters. There is a dropout layer after the fourth block with a rate of 0.3 to prevent overfitting. After the convolutional layers, the output is flattened into a 1D array using the Flatten() layer. The flattened output is then fed into a fully connected layer with 512 units, followed by an activation function (ReLU) and a dropout layer with a rate of 0.5 to further prevent overfitting.

Finally, the output is passed through another fully connected layer with 2 units, representing the 2 classes in the binary classification task. The activation function used in this output layer is sigmoid because it's a binary classification problem. The model is trained using the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.003, a decay rate of 1e-4, a momentum of 0.9, and Nesterov momentum to improve the convergence speed and accuracy of the model.

Emotion Detection Model: The CNN model consists of a base block followed by four modules, the base block has two convolutional layers with 8 filters of size 3x3, followed by batch normalization and ReLU activation. Each module has a residual connection from the previous block, and the first convolutional layer has 16, 32, 64, and 128 filters, respectively, for modules 1 to 4. The second convolutional layer in each module also has 16, 32, 64, and 128 filters, respectively. The kernel size for all convolutional layers is 3x3, and the max pooling size is 3x3 with a stride of 2.

There are no dense layers in this architecture, and the GlobalAveragePooling2D layer is used to reduce the spatial dimensions of the output tensor and obtain a feature vector. The softmax activation function is used as the final activation function to obtain a probability distribution over the output classes.

4.3 Classifier for Unified Model

Haar Cascade Classifier is an object detection method that analyzes an object's features using rectangular regions of an image. It creates a cascade of classifiers to detect the object with increasing complexity, making it fast, low memory, and highly accurate. It's widely used for face detection and has been applied in surveillance systems, biometrics, and augmented reality. The algorithm's accuracy depends on the number of features it's trained on, but using too many features can be time-consuming. The cascade structure divides features into groups, providing them with different stages of classifiers to speed up the detection process. The Haar Cascade Classifier has had a significant impact on computer vision and is an essential tool for feature extraction in various applications.

4.4 Unified Model

Our system utilizes advanced computer vision techniques to analyze video streams in real time. We first use pre-trained Haar cascade classifiers to detect faces and eyes in the video stream and preprocess the detected faces and eyes by resizing, normalizing, and reshaping them to improve detection accuracy. We then utilize our own pre-trained individual models to classify the emotions of the detected faces and to determine whether the eyes are open or closed. Additionally, a complete data frame is used to determine whether the detected person is an infarct or not. To monitor the level of drowsiness and infarct, we have implemented a feature that counts the number of consecutive frames where a person is classified as an infarct, and either eye is predicted to be closed, and this count is displayed on the video stream. Our system also predicts the emotion of the detected face and displays it on the video stream, while providing a buzzer in any case where the infarct is detected or the eyes remain closed for more than 10 seconds. The system is built using the robust and efficient

OpenCV library, which provides us with powerful image processing and analysis tools.

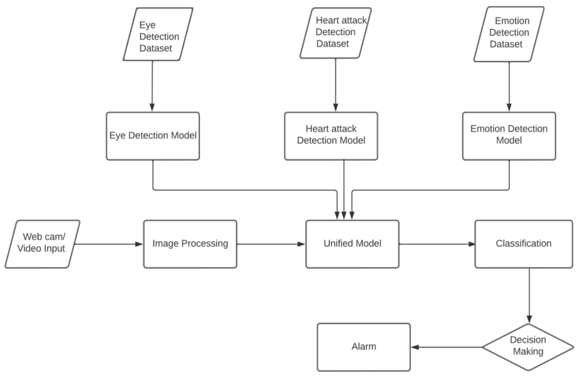

Figure 5. Proposed Architecture

5. Experimental Results

In this section, we will describe the evaluation metrics for our model. The class metrics for each individual model are presented in the following table. Additionally, the real-time evaluation results are showcased in this section.

Model	Accuracy	Precision	Recall	F1 score
Eye Detection	0.968	0.968	0.971	0.965
Emotion Detection	0.678	0.673	0.677	0.674
Heart Attack Detection	0.713	0.632	0.684	0.658

Table 2. Evaluation of Classical metrics for each model

5.1 Classic Evaluation Metrics

The following are the eye detection model's metrics.

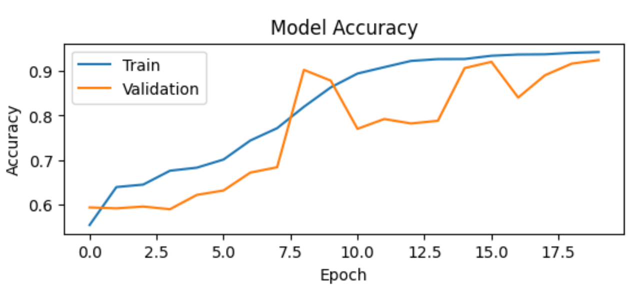

Figure 6. Eye Detection Model's Accuracy vs Epoch plot

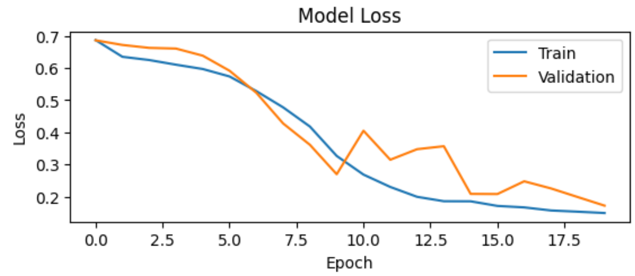

Figure 7. Eye Detection Model's Loss vs Epoch plot

The above figure 6 and 7 displays the eye model accuracy change with the number the epochs and model loss change with the number the epochs. By reaching 20 epochs the training accuracy reaches 96.6%, model loss reaches 0.0917 and validation accuracy reaches 95.9%, model loss reaches 0.1109.

	precision	recall	f1-score	support
0	0.97	0.98	0.97	248
1	0.98	0.97	0.97	248
accuracy			0.97	496
macro avg	0.97	0.97	0.97	496
weighted avg	0.97	0.97	0.97	496

Figure 8. Classification Report for Eye Detection Model

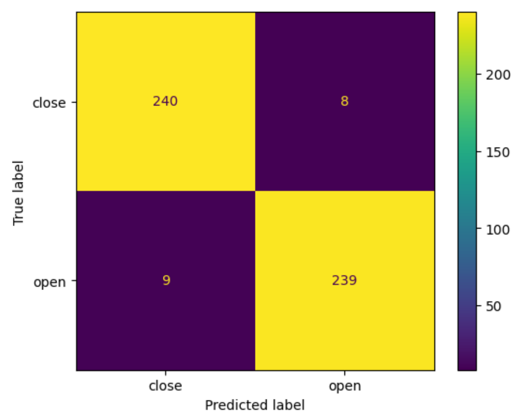

Figure 9. Confusion Matrix for Eye Detection model

The eye model's ability to accurately predict open and close eye classes is reflected in the small number of misclassifications, which indicates that the model is effective in distinguishing between the two classes. Overall, the confusion matrix suggests that the eye model performs well, and can be relied upon for accurate predictions in real-world applications.

The following are the emotion detection model's metrics.

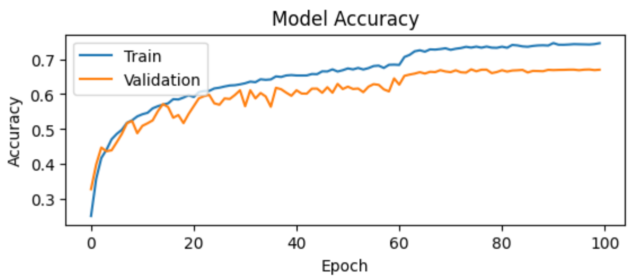

Figure 10. Emotion Detection model's Accuracy vs Epoch plot

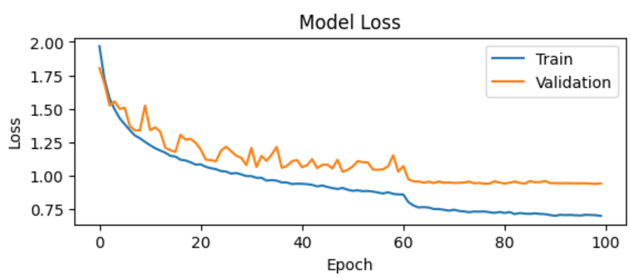

Figure 11. Emotion Detection model's Loss vs Epoch plot

The above figure 10 and 11 displays the emotion model accuracy change with the number the epochs and model loss change with the number the epochs. By reaching 100 epochs the training accuracy reaches 75.2%, model loss reaches 0.6974 and validation accuracy reaches 68.1%, model loss reaches 0.8961.

	precision	recall	f1-score	support
0	0.60	0.61	0.60	807
1	0.92	0.96	0.94	791
2	0.52	0.46	0.49	803
3	0.82	0.80	0.81	791
4	0.54	0.47	0.50	782
5	0.74	0.74	0.74	802
6	0.55	0.68	0.61	827
accuracy			0.67	5603
macro avg	0.67	0.67	0.67	5603
weighted avg	0.67	0.67	0.67	5603

Figure 12. Classification Report for Emotion Detection

From figure 12, we can see the overall emotion detection model's accuracy, precision, recall and f1-score along with each individual class as well.

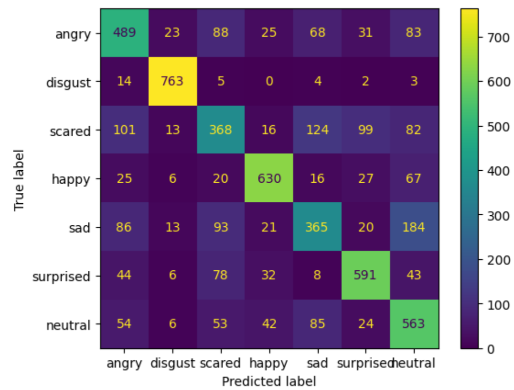

Figure 13. Confusion Matrix for Emotion Detection model

Looking at the diagonal elements of the matrix, we can see that the highest numbers are along the diagonal, indicating that the model is performing relatively well in correctly predicting the emotions.However, there are also a number of off-diagonal elements that indicate the model is making incorrect predictions. The model confuses "angry" with "scared", "sad", and "neutral", and "surprised" with "scared" and "sad".The model has decent performance in predicting some emotions correctly, but there is scope for improvement in correctly predicting other emotions.

The following are the heart attack detection model metrics.

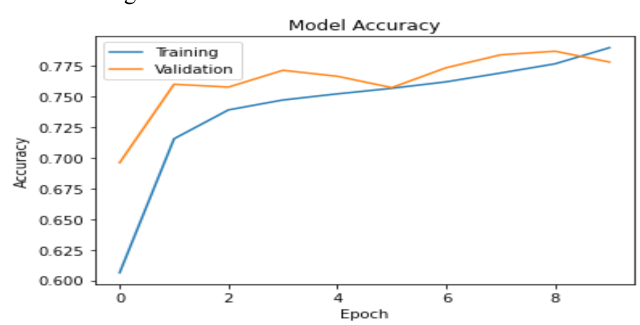

Figure 14. Heart Attack Detection Model's Accuracy vs Epoch Plot

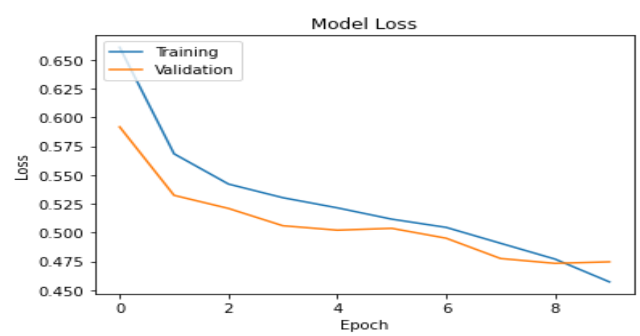

Figure 15. Heart Attack Detection Model's Loss vs Epoch Plot

The above figure 14 and 15 displays the emotion model accuracy change with the number the epochs and model loss change with the number the epochs. By reaching 100 epochs the training accuracy reaches 79.3%, model loss reaches 0.4696 and validation accuracy reaches 78.9%, model loss reaches 0.4830.

```
              precision    recall  f1-score   support

           0       0.25      0.15      0.19       790
           1       0.75      0.85      0.80      2394

    accuracy                           0.68      3184
   macro avg       0.50      0.50      0.49      3184
weighted avg       0.63      0.68      0.65      3184
```

Figure 16. Classification Report for Heart Attack Detection Model

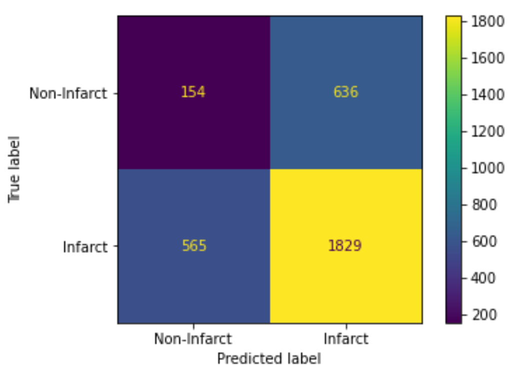

Figure 17. Confusion Matrix for Heart Attack Detection Model

In this case of Infarct detection, the model appears to have more success predicting the "Infarct" class, with a higher number of true positive predictions than false positive predictions. However, the model struggles with predicting the "Non-Infarct" class, with a higher number of false positive predictions than true positive predictions. Further analysis and refinement of the model is necessary to improve its performance for both classes.

Model's Accuracies:

Model	Train	Val	Test	EM
Eye Detection	96.6%	95.9%	96.8%	97.0%
Emotion Detection	75.2%	68.1%	67.8%	66.4%
Infarct Detection	79.3%	78.9%	71.3%	91.8%

Table 3. Model's accuracies compared with Existing Model (EM)

5.2 Real Time Prediction

The unified model is designed to analyze live video data frames and track specific movements and actions of the user. The application has different scores for different actions that are detected. The eye score is incremented when the application detects that the user has closed both their left and right eyes, and alerts the user when the score crosses 15. The face score is incremented when the application cannot detect a face in the data frame, and alerts the user when the score crosses 20. The emotion score is incremented when the application detects that the user is scared, which is determined by analyzing the face in the data frame, and alerts the user when the score crosses 3. The body score is incremented when the user places their hand on their chest in each data frame. The unified model can classify input data frames as infarct and will check for extra constraints such as the user falling out of the camera focus or closing both eyes, and alerts the user accordingly.

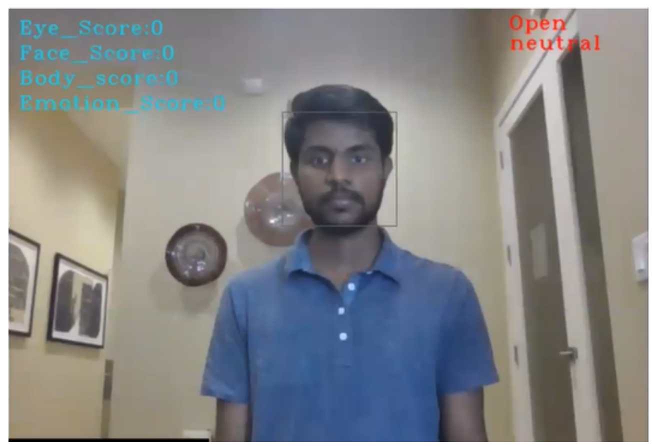

Figure 18. Open eye and neutral emotion

We can see that the emotion is neutral and eye state is open and no score is incrementing.

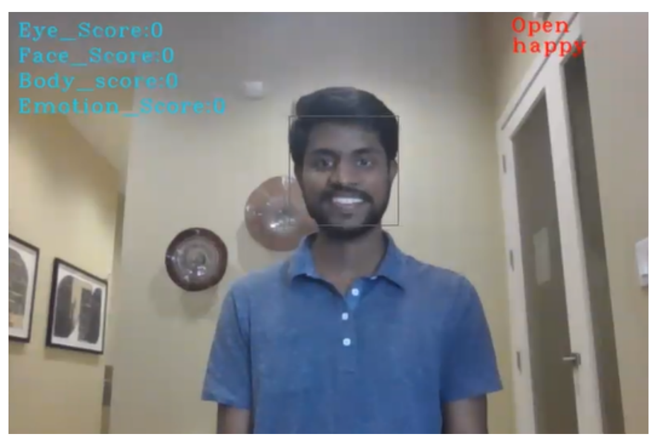

Figure 19. Open eye and happy emotion

We can see that the emotion is happy and eye state is open and no score is incrementing.

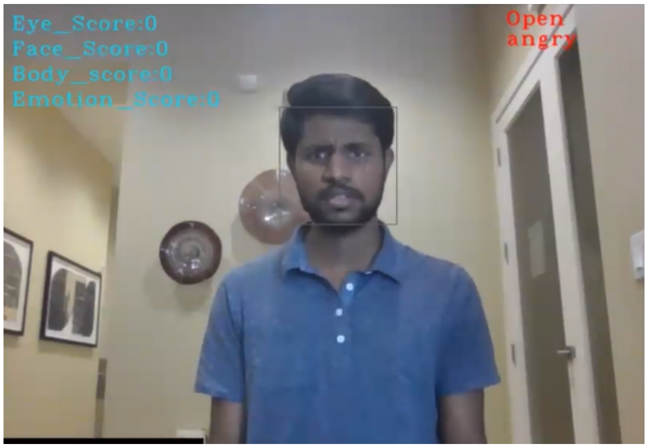

Figure 20. Open eye and angry emotion
We can see that the emotion is angry and eye state is open and no score is incrementing.

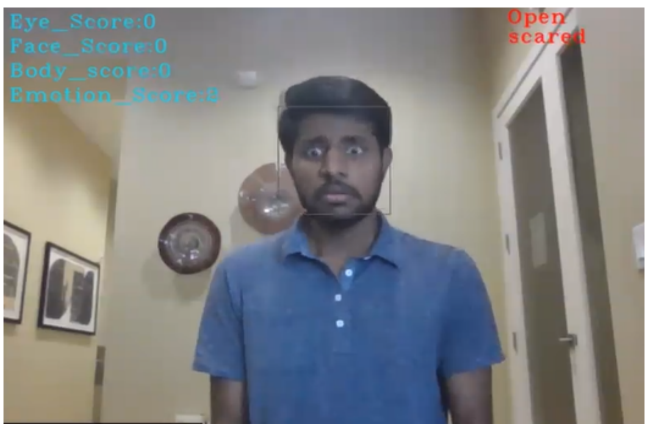

Figure 21. Open eye and scared emotion
We can see that the emotion is scared and the eye state is open and emotion score is incrementing.

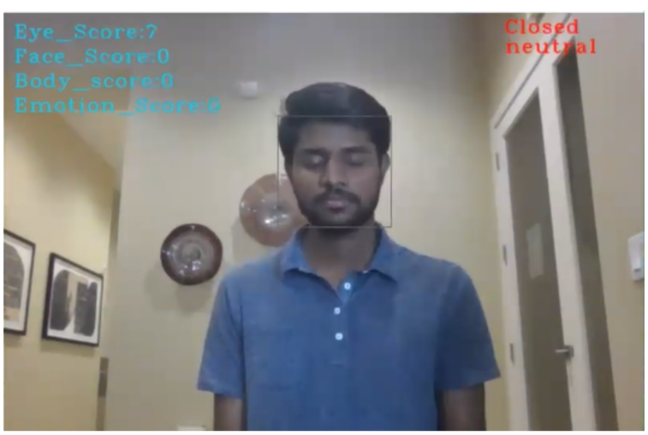

Figure 22. Closed eye and neutral emotion
We can see that the emotion is neutral and eye state is closed and eye score is incrementing.

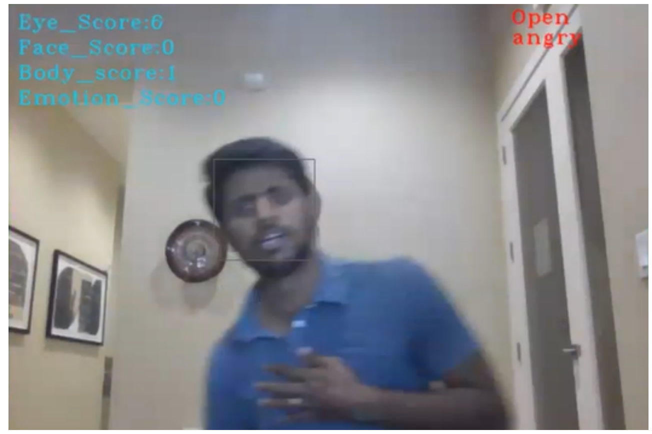

Figure 23. Open eye and Infarct detection
We can see that the eye state is open and body score is increasing.

Our unified model is currently capable of properly classifying emotions and eye state. Heart attack detection needs to be improved.

6. DISCUSSION

In our study, we achieved accuracies of 96.8% for eye detection, 67.8% for emotion detection, and 71.3% for heart attack detection. We compared our results on the FER2013 dataset with Mollahosseini et al. [10] who developed a neural network with two convolutional layers, followed by max pooling and four Inception layers. They achieved an accuracy of 66%, whereas we achieved an accuracy of 68% using the same dataset. Overall, it seems that the model is doing a satisfactory job of predicting some of the emotions correctly, but there is certainly room for improvement in correctly predicting the emotions for some of the other classes such as angry and surprised.

For heart attack detection, we used the dataset provided by Rojas-Albarracín et al. [7] and achieved an accuracy of 79.8%. However, this is currently lower than the accuracy achieved by Rojas-Albarracín et al. [7]. We speculate that this difference may be due to our limited computational resources and have taken 48*48 pixel instead of 256*256 by which made us compromise with the number of layers to be used. To address this, we plan to increase the number of epochs during training in order to match their performance. Our eye detection model performed comparably to Rafid et al. [5] and all other existing models to our knowledge.

Our study found that in modern workplaces, there are several individual models available for performing Health and Safety Monitoring. Our unified model performed well in detecting emotion and eye state, but there were challenges in detecting heart attacks, which need to be concentrated in the future. We further assessed the effectiveness of the system by comparing classic metrics such as accuracy.

7. CONCLUSION AND FUTURE WORK

The proposed real-time monitoring system that uses computer vision to detect and warn individuals about potential health and safety risks has shown promising results in detecting eye movements and emotions accurately with high precision and recall values. However, the heart attack detection model needs further improvement.

In future, there is a need for more extensive and diverse datasets to train the models and improve their performance. Additionally, incorporating other health and safety features such as detecting muscle strain or monitoring vital signs can further enhance the system's capabilities.

8. REFERENCES

[1] Satya Chandrashekhar Ayyalasomayajula, Bogdan Ionescu, and Dan Ionescu. 2021. A CNN Approach to Micro-Expressions Detection. *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)* (2021). DOI:https://doi.org/10.1109/saci51354.2021.9465542

[2] Yanjun Guo, Yantao Tian, Xu Gao, and Xuange Zhang. 2014. Micro-expression recognition based on local binary patterns from three orthogonal planes and nearest neighbor method. *2014 International Joint Conference on Neural Networks (IJCNN)* (2014). DOI:https://doi.org/10.1109/ijcnn.2014.6889620

[3] Yandan Wang, John See, Yee-Hui Oh, Raphael C.-W. Phan, Yogachandran Rahulamathavan, Huo-Chong Ling, Su-Wei Tan, and Xujie Li. 2016. Effective recognition of facial micro-expressions with video motion magnification. *Multimedia Tools and Applications* 76, 20 (2016), 21665-21690. DOI:https://doi.org/10.1007/s11042-016-4079-6

[4] Yujian Ma, Yantao Wei, Yafei Shi, Xiuhan Li, Yi Tian, and Zhongjin Zhao. 2022. Online Learning Engagement Recognition Using Bidirectional Long-Term Recurrent Convolutional Networks. *Sustainability* 15, 1 (2022), 198. DOI:https://doi.org/10.3390/su15010198

[5] Anis-Ul-Islam Rafid, Atiqul Islam Chowdhury, Amit Raha Niloy, and Nusrat Sharmin. 2021. A Deep Learning Based Approach for Real-time Driver Drowsiness Detection. *2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)* (2021). DOI:https://doi.org/10.1109/iceeict53905.2021.9667944

[6] Sukrit Mehta, Sharad Dadhich, Sahil Gumber, and Arpita Jadhav Bhatt. 2019. Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio. *SSRN Electronic Journal* (2019). DOI:https://doi.org/10.2139/ssrn.3356401

[7] Gabriel Rojas-Albarracín, Miguel Ángel Chaves, Antonio Fernández-Caballero, and María T. López. 2019. Heart Attack Detection in Colour Images Using Convolutional Neural Networks. *Applied Sciences* 9, 23 (2019), 5065. DOI:https://doi.org/10.3390/app9235065

[8] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. 2012. Unstructured human activity detection from RGBD images. *2012 IEEE International Conference on Robotics and Automation*(2012).DOI:https://doi.org/10.1109/icra.2012.6224591

[9] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked Hourglass Networks for Human Pose Estimation. *Computer Vision – ECCV 2016* (2016), 483-499. DOI:https://doi.org/10.1007/978-3-319-46484-8_29

[10] Ali Mollahosseini, David Chan, and Mohammad H. Mahoor. 2016. Going deeper in facial expression recognition using deep neural networks. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2016). DOI:https://doi.org/10.1109/wacv.2016.7477450

[11] Free Editable Neural Network Examples & Templates | EdrawMax. Retrieved May 3, 2023 from DOI:https://www.edrawsoft.com/neural-network-examples.html