# Detection of Distracted Students Using Machine Learning

CYRILL ANWAR NORAZMAN[1]

[1]Faculty of Computer Science and Information Technology,
University Malaysia Sarawak, 94300, Kota Samarahan, Sarawak
63154@siswa.unimas.my

IRWANDI HIPNI MOHAMAD HIPINY[1]

[1]Faculty of Computer Science and Information Technology,
University Malaysia Sarawak, 94300, Kota Samarahan, Sarawak
mhihipni@unimas.my



## ABSTRACT

**Recently, online examination has been implemented in huge scale as the pandemic hits. However, currently there is no automated and structured system to conduct this examination system in a systematically manner. Online examination also requires an examination invigilator to keep an eye on the candidates However, it is impossible to assign an examination invigilator in an online examination since all the candidate's location are not at one place. Therefore, this problem has led to the development of this research. In this research model, we are constructing a machine learning model that can detect students that are attempting to cheat and sleep in online examination. The term distracted was used in this research title as this research just not only detecting students that are cheat, but also detecting students that are sleep during online exam.**

Keyword: Online Examination, Education 4.0, Distracted detection, Machine Learning, Binary classification, K-Fold cross validation

## INTRODUCTION

Within the previous decade, the implementation of technology in various fields has shown significant growth and brings impactful progress to those fields. The educational field is not an exception for technology to thrive. Although, in some way, we are still stuck with the same traditional education system, there is always room for improvement where technology can be fully utilized.

An examination is a crucial part of the educational process used to assess students' knowledge and understanding of the topics they have learned at school. Examinations are considered by society as a vital milestone to achieve their respective goals in life. Good grades may reward students a stable job or admission into higher institutions after graduating. Due to the importance of having good grades, some students opt to cheat during examinations. Cheating in examinations is also called academic dishonesty and violates the rules of academic protocol. Study shows that cheating does not exist in a particular country only, but it has become a universal phenomenon in educational institutions (Sharifuddin, 2009).

Another scenario that we can usually observe during examination is that candidates easily tend to fall asleep during and complain they did not have enough time to answer all the questions. It has become one of the invigilator's responsibilities to ensure students stay awake during the exam. With the existence of this research model, time and energy of the invigilators can be definitely used productively elsewhere.

In this project, machine learning will be used to detect distracted students in an online-based examination system. The possible way to detect distracted students are by determining the position of student's head pose and eye gaze. When the student's head is posing sideway for a long time, this can be interpreted as an attempt to cheat. Meanwhile, if the student's head pose is fixated at non-relevant areas for a long time, this can be explicated as the student is trying to steal glances at their friend's answer sheet. Lastly, we can interpret that the student may have a high propensity to fall asleep when the eyes are closed. The aim of this project is to detect cheating activities in a computer-based exam that usually conducted in candidate's personal place such as at home. Besides, this system also aims to detect students that sleeps during a test. In our model, this is done by detecting the eye and head pose of the students during the examinations using a webcam camera that the candidates use such as laptop or personal computer. The laptop camera which is usually placed in front of the students will be used to observe the action of the candidates during exam. Actions like sleeping, head posing sideways, and faces posing downward will be considered as possible attempt of cheating or distracted. The term 'distracted' was used

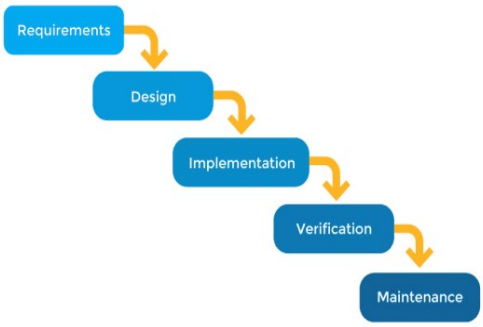
in the project title because the system will not only try to detect student's attempt to cheat during online exam, however, this proposed system also aims to detect the students that sleep during the test.

**Research questions**

The target audience for this project is intended for educational institutions that held online exams for their students. This system will become handy when it comes to an examination week. Only simple hardware is needed for this system to operate thoroughly. Any camera that can be connected to the laptop is enough for the system to work. This research may contribute to the educational field by improving the current traditional examination system to a technology-driven system where the needs of examination invigilator during exam can be eliminated.

## MATERIALS & METHODS

**Waterfall model**




(Santos, 2020)

Figure 1 : Waterfall Methodology.

The first phase is a requirement in which project requirement and scope is defined by gathering information from the targeted user. This phase involves understanding what needs to design and its function and purpose of this research project. In this phase, data was gathered by conducting a questionnaire survey to a group of students to analyse and determine the current examination system's problem and needs. This survey is being explained in detail in the data analysis section.

After the requirement has been defined, the system will undergo a design phase based on the requirements that have been gathered in the previous stage. In this phase, the system architecture, flowchart, activity diagram, use case diagram will be visualized as a guideline to develop the research model. All these diagrams will be documented and explained in the system design section of this chapter. Besides, the hardware and software requirements also have been listed as the user's specification requirements to install and run this project.

The next phase is where the real development will begin as the developer will implement the proposed research in this phase. The system will be developed based on the design requirements that have been gathered in the previous phase.

After the development process finished, the testing phase take part. Since this research focuses on developing a machine learning solution, accuracy testing will be used to validate the accuracy of the machine learning model. Besides, F1, precision, and recall functions will also be calculated in this testing phase.

After the accuracy testing has been done, the machine learning model will be evaluated using three methods. The first evaluation is performed by test the machine learning model using a real-time camera. This method will detect student's action in real time and produce the accuracy value of distracted or non-distracted. The second method is using the model to predict the image from the datasets that contains distracted or non-distracted students.

Flowchart Diagram

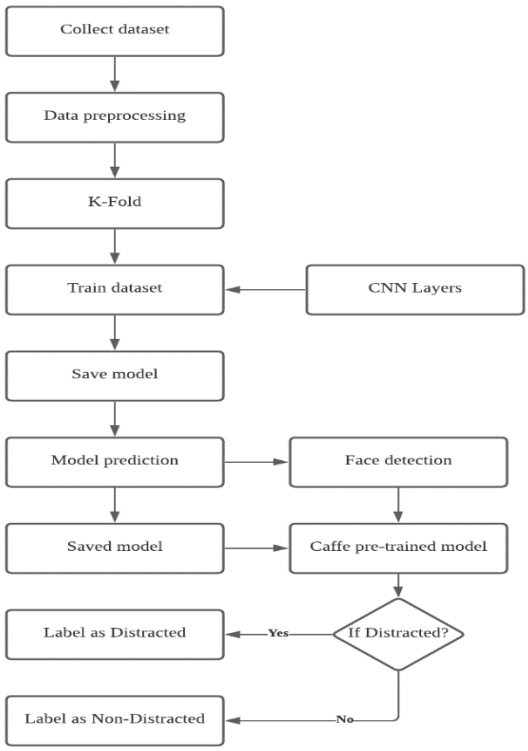



Figure 2: Flow of the proposed system

Firstly, experiment will be setup to collect dataset that contain images of distracted and non-distracted students. Then, the data will be processed to handle issues such as missing value, incorrect image labels and so on. After that, the dataset will be split into number of folds by using K-Fold machine learning concepts. Then each fold will be trained using neural network

layer. After finish training every fold, the model will be saved and ready to make prediction alongside the face detection model. These two models will work together to classify an action of students into distracted or not distracted categories.

## Distracted detection techniques

### 1) Image Dataset Acquisition

In this step, the image of various faces for each person are collected to form a dataset. In this project, we built our dataset using Keras that consist of these following criteria:

- Head pose facing sideways
- Face pointing downwards
- Head facing front with both eyes opened
- Head facing front with eyes closed

These collected images will then be processed and trained to assist the computer detecting faces with those criteria later.

### 2) Image Pre-Processing

Image or data processing is performed to normalize the face images geometrically and photometrically (Khanday, 2018). The geometrical normalization method will convert the face image into a normal face by copping the face while the photometrical method normalizes the face based on properties for example illumination grayscale, noise reduction, geometrical correction, etc. Image pre-processing aims to enhance the quality of the acquired image (Gunasekaran, 1996).

#### 3) Face Detection

This is the last and vital step in every face detection system. In this step, DNN Face Detector method was used by using OpenCV library.

***DNN Face Detection***

OpenCV is an open-source library that specializes in computer vision and image processing that is written in C++ language. It offers many face detection techniques that we can use and the first real-time face detector was Viola-Jones Algorithm. However, as time passes by, there is a lot of new face detection methods that provides better result than the Viola-Jones algorithm. Therefore, we'll be using DNN Face Detector as our face detection model. The DNN Face Detector is based on the Single Shot-Multibox Detector (SSD) that uses ResNet-10 architecture as its backbone.

The reason we chose DNN Face Detector rather than the traditional Viola Jones-algorithm is mainly because it provides faster detection result in real-time. It is highly accurate and also easy to be implemented in any data science project. To implement, this DNN Face Detection method, there are few files that are required to be included in our structured folder. These files can be acquired from the official Caffe github repository.

### 4) Classification using Machine Learning (K-Fold)

## Binary Classification

In this project we will be focusing on Binary classification since this project aims to solve a binary classification issue. Binary classification is a classification task that have two class labels such as 0 or 1, true of false, infected or uninfected. It involves on predicting one of two classes. There are a lot of projects that can be implemented by using this classification type. For example, in Titanic dataset which are oftenly used as a beginner machine learning project, will be predicting a probabilistic output of 'Survived' or 'Not Survived'. Same as Customer Churn Prediction project in Telco company, the predicted outcome that need to be identified are 'Churn' or 'Not Churn'

Based on the example provided, the usage of binary classification is suitable to be implemented in this project since we are trying to detect the distracted student which will consists of two categorical classes which are 'Distracted' and 'Non-Distracted'. To evaluate a binary classification performance, we may provide the results by calculating precision, recall, or F1 value.

#### K-Fold Cross Validation

K-Fold is a cross-validator that divides the dataset into a number of folds. In this project context, the values of K were set to 3 which means that the dataset will be divided into 3 folds. The data will be shuffled so that each image of participant exists in each fold. Since k fold will do the splitting of data by itself, the train and validation set is combined by using concatenate functions. These concatenations were separated into two categories which are the input and target labels. Inputs were feed into the CNN layer model and labels are used to test the model prediction. K-Fold can be easily implemented by using the Scikit-Learn library. Below is the pseudocode to visualize the implementation of K-Fold.

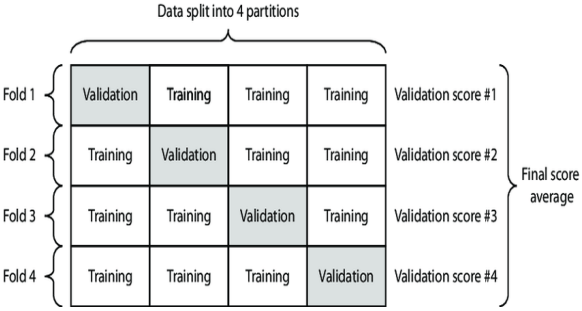



Figure 3: The concept of K-Fold cross validation

**Existing works on preventing student's academic misconduct**

In this section, we will review related papers based on researchers' efforts to prevent students from cheating by using technology.

**I. An E-exam cheating detection system**

One of the solutions to prevent academic dishonesty when conducting online-based examination is by using authentication methods that can be used to avoid students cheating. The authentication methods will use fingerprints and eye identification to differentiate each student. Based on the paper 'E-exam Cheating Detection System' written by Bawarith in 2017, The authors claimed that E-proctor is an efficient and integrated solution to tackle this problem by monitoring students while taking an online examination. E-proctor works by authenticating and detecting fingerprint and eye movement of the students. The eye tracker will use a camera from the device that students use to take exams and track their pupils' movement. Students are required to pass the ID authentication process by using a fingerprint scanner to register for the examinations. After that, users will be asked to look into the display screen to calibrate their eye position. After the users successfully pass the authentication and calibration process, the E-proctor launches the exam application while locking the computer until the exam is finished. During the examination process, the proctor will record every eye movement change (Bawarith, 2017).

**II. Intelligent Alarm based visual eye tracking**

This research paper was written by Javed and Aslam, focusing on developing a system to avoid student cheating in paper exams by detecting and tracking the student's eye mainly using the Viola-Jones algorithm. Viola-Jones algorithm used in detecting faces in this system because it produced an excellent result despite using a low resolution from a low-spec camera. This paper presented an efficient and effective algorithm for face and eye detection (Javed & Aslam, 2013). However, this system can detect potential students that may cheating from the frontal view only.

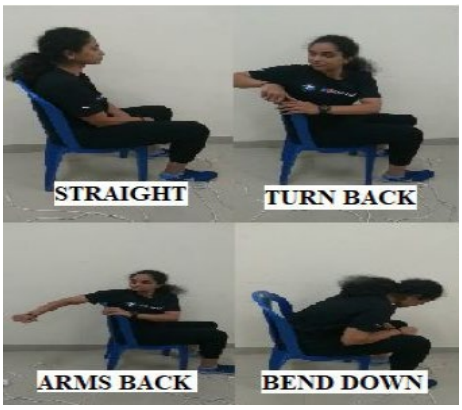
The result from the experiment made by the author shows that Viola-Jones human detection algorithm can detect the eye movement up to 78% successfully. In comparison, the proposed algorithm shows the best results up to 93% to detect the eye movement. The proposed algorithm has been tested using a dataset of images with different light resolutions camera and under different light intensity. The author concluded that proposed algorithm produced the best result under all conditions and circumstances. However, this system's limitation is that it can only detect up to 5 more students only at the same time. Besides, the system cannot detect students who cheat from the side view since it focuses on the student's images' frontal view.

**III. Automated cheating detection in exams using posture and emotion analysis**

In this research paper, the authors proposed a model to detect students from cheating by detecting abnormal or cheating activities in an examination hall using CCTV footage (Nishchal et al., 2020). These irregular activities can be done by detecting the student's body posture. Actions like head position, turning back, bending etc., are detected, which are considered a possible cheating attempt.

This model is the improved version from the previously reviewed research systems which can only detect students cheat from frontal view only. This author improvises the previous model by furthering their studies to detect students cheating that focuses on students' lateral view in the examination hall. Hence, it is required to place the camera at place such that the camera can clearly see the lateral view of students. This model comprises 5 methodology parts: Posture detection using OpenPose, AlexNet model to detect the type of cheating, emotion analysis, and face recognition and E-report generation

This model system predicted if a person is turning back or sitting straight with the position of eyes and arms. If students' posture is seen to be not sitting straight, the system will consider as an attempt of cheating. The diagram below illustrates how the system detects students' body posture using points from OpenPose. The diagram below illustrates the sample dataset classes that were used to train and develop this system




(Nishchal et al.,2020)

Figure 4: The sample dataset used in this system

The system also uses sentiment analysis in the system. The author found that facial expressions play a vital role in human interactions. Since human interactions are not allowed among students during exams, this reviewed model includes the system's sentiment analysis code.

## RESEARCH IMPLEMENTATION

### Experiment Setup

Before performing the initial implementation for this research project, an experiment was conducted for data acquisition. Since data was very crucial in any machine learning project, the data gathered must be acquired in a neat way to avoid missing data and other issues during coding process. To conduct the experiment, a laptop camera was used to capture videos and it is situated in front of the exam candidate since this project focuses on detecting students that were distracted in a Computer-Based Test.

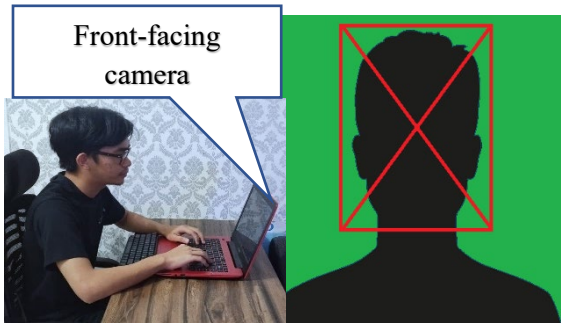



Figure 5: Experiment setup and face Region-Of-Interest (ROI) in frontal view.

#### 1) Data Collection

In the data collection process, 6 videos of distracted students and another 6 videos for non-distracted were recorded. These videos contain an action or behaviours of students based on their categories.

The action of distracted students recorded are include student's head is posing sideways, closed eye and the pose of students that their face is pointing downwards. For the non-distracted videos, it contains an action of students that are facing towards the laptop.

Each video was recorded for 10 seconds long. There are 4 participants that involved in the data collection process to enable the model to learn the variety of features in the images. In the distracted videos, some of the features are being recorded in the video such as:

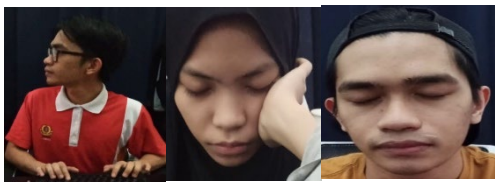


Figure 6: Shows data sample for Distracted students

While, for the non-distracted videos, the candidate is focusing on the camera while typing his keyboard. The head pose of the students points toward the camera to visualize that the student is not distracted. The eye of the student is wide open to distinguish that the student is not sleeping.

Figure 7: Shows data sample for Not Distracted students

#### 2) Data Preparation/Cleaning

This process is done to improve data quality by removing unnecessary features in the image dataset.

Initially, 12 videos that were recorded was converted into jpg image frame by frame. That means that every duration of each 10 seconds video will produce 600 images. These processes were performed with the help of OpenCV library to capture video frame and OS library to access the folder and file in the PC.

After that, the folder structure is setup by locating the dataset into two categories subfolders with name 'Distracted' and 'Non-Distracted'. These two subdirectories folder will make this model a binary classification task.

Since the dataset is captured in a video form. The image that has been converted contains some unnecessary and noisy data that may disturb the prediction on the model later. Hence, all these unnecessary images were removed or cropped before data processing. Finally, after performing data cleaning, the dataset will contain two folders which categorizes into 'Distracted' and 'Non-Distracted' subfolders. For the distracted folder, it contains 461 images while Not Distracted folder contains 393 images.

#### 3) K-Fold Cross Validation

K-Fold is a cross-validator that divides the dataset into a number of folds. In this project context, the values of K were set to 3 which means that the dataset will be divided into 3 folds. The data will be shuffled so that each image of participant exists in each fold. Since k fold will do the splitting of data by itself, the train and validation set is combined by using concatenate functions. These concatenations were separated into two categories which are the input and target labels. Inputs were feed into the CNN layer model and labels are used to test the model prediction. K-Fold can be easily implemented by using the Scikit-Learn library. Below is the pseudocode to visualize the implementation of K-Fold.

```
# Define K-Fold cross validation
kfold = KFold(number of folds = 3, shuffle data = True)
```

Figure 8: Pseudocode to define K-Fold Cross Validation.

#### 4) Train fold in CNN Layer

Due to limited computational power to train the model, the epoch was set to 20 as the result show the best in this range. It took for about an hour to wait for the training dataset to pass through all deep learning layers. The batch size is set to 10 to take less time to train all the data. Epochs is how many times the training dataset will pass through the layers while batch size means that the data is being divided evenly into multiple batch to go through the layer one by one. By reducing the number of epochs and increasing the batch size for the neural network layer, we can significantly reduce the computational time and power for the machine to train the dataset. However, the cons of doing so will affect the accuracy of the model. Finally, after the neural network layer has been created, each fold will go through these layers and will be Iterated until all 3 folds are finished. The accuracy, loss and validation loss and validation accuracy for each layer will be calculated during training. After all 3 fold has finished training, the average of the accuracy and loss is calculated based on the 3 folds.

The model architecture of the neural network layer can be visualized in the figure below.

```
#Initialize K-Fold cross validation
number of folds = 3
shuffle data = True

#loop to train all folds
for k=1
    Sequential layer
    Conv2D layer, activation = 'relu'
    MaxPooling2D layer
    Conv2D layer, activation = 'relu'
    MaxPooling2D layer
    Flatten layer
    Dense layer, activation = 'relu'
    Dropout layer
    Dense layer, activation = 'softmax'

    calculate accuracy and loss for each fold

#train next fold
k += 1
```

Figure 9: Pseudocode to train folds in neural network

## RESULT & DISCUSSION

### Average score for folds

After training phase has finished, all accuracy and loss for each fold are being recorded. Loss is a value that indicates how bad the model prediction is while accuracy is a metrics to determine on how good the model performs. The accuracy for Fold 1 is 97% and the loss is 0.11 while for the accuracy of Fold 2 is 98% and the loss is 0.25 and the last fold's accuracy gained approximately about 97% and the loss is 0.18. From the recorded accuracy and loss for each fold, their average can be calculated and produce a result of 97% accuracy and 0.18 for the loss value. Basically, model with high accuracy and low loss indicate that the model is good. The figure below shows the recorded accuracy and loss for each fold and their average.

Table 1: Score for each fold and their average score

| Score per fold | | |
|---|---|---|
| No. of Folds | Loss | Accuracy (%) |
| Fold 1 | 0.10795241594314575 | 97.54385948181152 |
| Fold 2 | 0.2523539066314697 | 98.24561476707458 |
| Fold 3 | 0.19645299017429352 | 97.88732528686523 |
| Average scores | 0.18558643758296967 | 97.89226651191711 |

### F1, Precision, Recall Value

After calculating the accuracy of the model, we also can process the f1, precision, and recall values. These three metrics can be calculated from the data of True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN).

Table 2: The result report

| Classification Report | | | | |
|---|---|---|---|---|
| | Precision | Recall | F1-score | Support |
| Distracted | 0.96 | 0.98 | 0.97 | 45 |
| Not_Distracted | 0.97 | 0.95 | 0.96 | 41 |
| accuracy | - | - | 0.97 | 86 |
| Macro avg | 0.97 | 0.96 | 0.96 | 86 |
| Weighted avg | 0.97 | 0.97 | 0.97 | 86 |

Precision value describes on how precise the model is by the predicting how many of the predicted positive are actual positive (Shung, 2018). The precision gained for Distracted produce an output of 96% while Not_distracted produce 97% result.

On the other hand, Recall metrics calculates how many of the Actual positives that our model captures through labelling it as Positive (True Positive). Recall metric should be the decision maker when selecting the best model. The Recall value obtained from this model is 98% for Distracted class while 95% for Not Distracted class.

F1 function is the combination of precision and recall value. It is important when we want to find balance between Precision and Recall. The value of F1-Score obtained from our model is 97% for Distracted and 96% for Not Distracted class. The difference between f1 and accuracy metrics is that . Hence, f1 value provides better and more reliable measure than accuracy metrics when there is an uneven class distribution (Shuang, 2018). The figure below shows the mathematical formula to compute the f1-score.

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

Figure 10: F1-Score mathematical equation

**Accuracy graph**

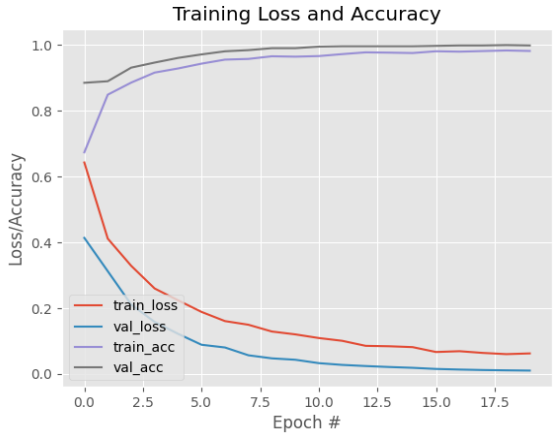



Figure 11: Plot graph for accuracy and loss functions

Based on the graph shown in Figure __ We can observe that the train loss and validation loss have been reduced as the number of epochs increases, which is a good sign for model prediction. Also, the train and validation accuracy are increasing as the number of epochs increases. Hence, we can evaluate that the model is neither underfit nor overfit.

**Test real-time camera prediction**

Finally, the model is being used in a real-time camera. The laptop camera is being used with the help of OpenCV library and Python script to access the webcam. The bounding rectangle box will appear when the face is detected, and the percentage of accuracy will appear on the top of bounding box to show the percentage level of distracted students during the live-stream video. Besides, the label will also appear to distinguish whether the students fall into 'Distracted' or 'Not Distracted' categories. The figure below shows how the model performs its prediction in real-time camera.

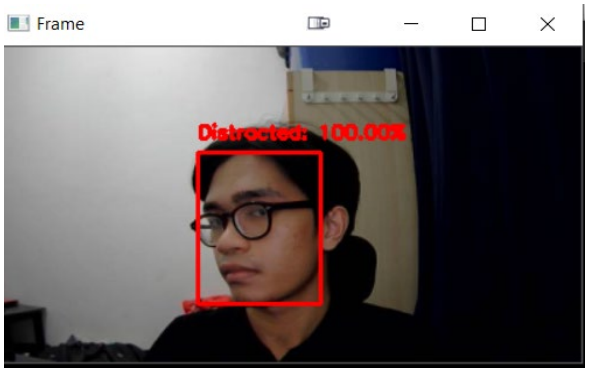



Figure 12: Real-time prediction of distracted student that head posing sideways.

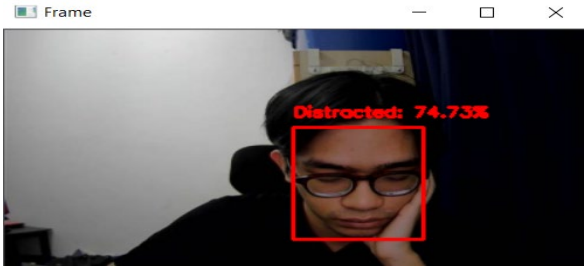



Figure 13: Real-time prediction of distracted students that the face is pointing downward.

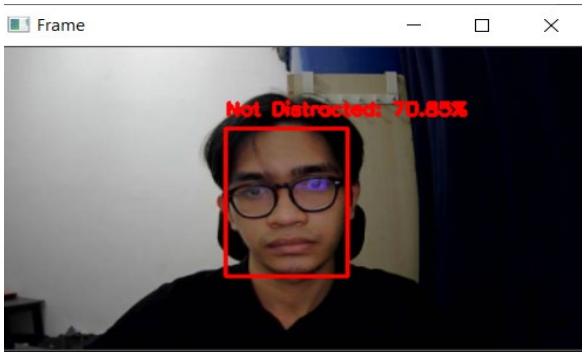



Figure 14: Real-time prediction of not distracted students that focuses on the laptop screen.

## LIMITATIONS

There are several limitations that were identified during the research development of this system.

a) The proposed system did not detect the blinking of eye well enough hence making it difficult to detect students that are sleeping during the computer-based examination. It is advisable to use pre-trained Haar-Cascade by using haarcascade_eye.xml file when it comes to detecting the blinking of an eye.
b) The detection is very sensitive even to a slight movement of head position. It will label an action as Distracted if the person tilts their head slightly. Therefore, the frequency of detecting the distracted student will be high and quite often.
c) Low computational power to run advanced deep learning model.
d) Only a few datasets used, can gain more accurate results when using huge datasets

## FUTURE WORKS

Due to its limitation, this research model can be improved in certain criteria. The potential future works can be done to equip this research with advanced functions and performance. Some suggestions on improving this research works are as follows:

a) Can implement this research into real-world
b) Can be deploy in desktop or web application based
c) Can be integrated in an online-learning platform during examination

d) Can detect students that opening chat applications or internet browser
e) Can add and detect other behavior that is possible to determine cheating attempt

These criteria can be the guideline for future researchers or developers to lengthen this research work until it can be readily perform in a large-scale system.

## CONCLUSION

By the end of this research development, we can conclude that the objectives have been achieved but there is a few limitations that can be improved in the future. Therefore, there are room for improvement for this current research model and these criteria were mentioned so that it can be the baseline for future researchers and developers to lengthen these studies.

## ACKNOWLEDGEMENT

Sincere gratitude is dedicated to my supervisor, Dr. Irwandi Hipni bin Mohamad Hipiny who gave me the golden opportunity to do this interesting project. Also, many thanks to Dr. Chai Soo See, for giving constructive feedbacks and comments on my Final Year Project. I extended my gratitude to all participants and individuals that involve in this study.

## REFERENCE

Bawarith, R., Basuhail, A., Fattouh, A., & Gamalel-Din, S. (2017). E-exam cheating detection system. *Int. J. Adv. Comput. Sci. Appl*, *8*, 176-181

Gunasekaran, S. (1996) Computer vision technology for food quality assurance. *Trends in Food Science & Technology*, *7*(8), 245-256.

Javed, A., & Aslam, Z. (2013). An intelligent alarm based visual eye tracking algorithm for cheating free examination system. *International Journal of Intelligent Systems and Applications*, *5*(10), 86.

Khanday, A. M. U. D., Amin, A., Manzoor, I., & Bashir, R. (2018). Face Recognition Techniques: A Critical Review

Nishchal, J., Reddy, S., & Navya, P. N. (2020, July). Automated Cheating Detection in Exams using Posture and Emotion Analysis. In *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)* (pp. 1-6). IEEE

Rowley, H. A., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions on pattern analysis and machine intelligence*, *20*(1), 23-38.

Santos, J., 2021. *3 Best Tools For Waterfall Project Management*. [online] Project-Management.com. Retrieved from https://project-management.com/3-best-tools-for-waterfall-project-management

Sharifuddin, S. A & Holmes, R. J. (2009). Cheating In Examinations: A Study Of Academic Dishonesty In A Malaysian College.

Shung, K. P. (2018). Accuracy, Preccision, Recall or F1?. Retrieved from Accuracy, Precision, Recall or F1? | by Koo Ping Shung | Towards Data Science