Logo

Description automatically generated

**Capstone Project Phase A**

**HighAlert – Security awareness detection system**

Supervisor: Aliza Attaly

Project Number:23-2-D-13

Students:

Yohan Leykin yohan.leykin@e.braude.ac.il

Haim Hadad [haim.hadad@e.braude.ac.il](mailto:haim.hadad@e.braude.ac.il)

**Abstract.** The focus of this academic final project is to develop a system that can detect the awareness levels of security guards, who are responsible for ensuring the safety of people and property in public places. Inattention and distraction of security guards can lead to serious security breaches, and it is crucial to monitor their awareness levels to prevent such incidents.

In this project, we propose an approach that utilizes computer vision techniques to analyse the behavior and facial expressions of security guards and predict their awareness levels. We develop an image processing procedure to recognize the security guard's face and behavior and an algorithm that extracts unique features of inattention and distraction.

Overall, this project contributes to the development of effective and efficient techniques for monitoring the awareness levels of security guards. The proposed approach has potential applications in various settings, including public safety, transportation, and commercial areas. By detecting and notifying the security guard of their inattention and distraction, we can prevent security breaches and ensure the safety of people and property in public places.

***Keywords:*** DeepLearning, FaceDetection, AwarenessDetection, ImageProcessing, Camera, Jetson.

1. **INTRODUCTION**

Security guards play a critical role in ensuring the safety of public places such as airports, malls, and public transportation systems. According to a report by the Security Industry Association (SIA)[1], 68% of security breaches occur due to human error, with the majority of these errors being related to the inattention and distraction of security guards. Incidents such as theft, vandalism, and violence can occur when security guards are unaware, leading to loss of property and sometimes injury or loss of life. In a study by the National Institute for Occupational Safety and Health (NIOSH)[2], it was found that security guards who work long hours or irregular shifts are more likely to experience fatigue and reduced alertness, leading to a higher risk of security breaches. The study also found that security guards who have received little or no training in security awareness and emergency response are more likely to be involved in security incidents. Furthermore, a survey conducted by the International Foundation for Protection Officers (IFPO)[3] found that 70% of security officers have reported being tired or fatigued at work, with 33% reporting that they have fallen asleep on the job. The survey also revealed that fatigue and sleepiness were the leading factors contributing to the inattention and distraction of security guards.

To address this issue, this academic project proposes an approach that utilizes computer vision techniques to analyze the behavior and facial expressions of security guards and predict their awareness levels. The proposed approach is based on the analysis of real data from surveillance videos collected from a public place. The proposed system can notify the security guard when their awareness levels drop below a certain threshold, preventing security breaches and ensuring the safety of people and property in public places. It can also produce a report of the security guards awareness levels over time, enabling security managers to identify patterns and take appropriate measures to improve the awareness levels of their security guards. The system utilizes image processing algorithms to track the movements of the eyes and mouth of the security guard and examine their behavior. It evaluates the fatigue levels of the security guard by monitoring the blink frequency, yawning, and snoozing. The system includes a camera, mini pc, with the majority of the project being the software running on the mini pc. These statistics highlight the need for effective monitoring and management of the awareness levels of security guards to prevent security breaches and ensure the safety of people and property in public places. The proposed approach in this academic project aims to address this issue by utilizing computer vision techniques to detect and monitor the awareness levels of security guards in real-time.

1. **RELATED WORK**

There are three fields that discuss drowsiness detection and how to overcome them,

In the physiological-based detection category, researchers have explored various methods that involve using physiological sensors such as EEG[4], Electrocardiography, and Heart Rate Variability (HRV). These sensors collect data, which is then combined with machine learning algorithms. One study describes an LSTM model that utilizes EEG data[5] and achieved a high accuracy. However, a drawback of using EEG sensors is the requirement of wearing a helmet.

Another approach in the physiological field is multivariate statistical process control, which detects abnormalities in HRV[6]. While this algorithm showed excellent accuracy, it was tested on a small dataset consisting of a limited number of individuals.

In the vehicle-based detection category, systems have been developed to monitor and analyze vehicle behavior, such as steering wheel angle and lateral distance. One study used steering wheel angles and speed movements to identify driver drowsiness during steering wheel rotations[7].

In the image processing detection field, various methods can be applied, including eye, mouth, and head tilt detection. One system identifies the face, eyes, and localizes the pupil and iris boundary using the circular Hough Transform on the extracted eye region[8]. Another study used a Convolutional Neural Network (CNN) to detect drowsiness features such as yawning and eye movements[9]. This approach achieved a high accuracy using a large dataset with different driving scenarios.  
  
our seniors made a similar version to detect drowsiness using image processing detection,  
they discuss a driver drowsiness detection system that utilizes real-time camera frames, facial landmark extraction, fatigue detection, and fall asleep detection to identify and alert drivers about potential drowsiness-related hazards during driving.

The system captures frames using a camera and extracts 68 facial landmark points using the Dlib library[10]. It then runs parallel processes for fatigue detection and fall asleep detection. Fatigue detection involves counting behaviors such as blinking, yawning, and snoozing, and notifying the driver in real-time if the count exceeds a threshold. Fall asleep detection includes analyzing eye closure duration and head movements to identify situations where the driver's attention is not on the road. In such cases, the system alerts the driver with alarms and monitor displays.

Additionally, the system employs a scoring mechanism to evaluate drowsy behavior, assigning scores to each feature based on their importance. Graphs of feature scores are displayed periodically. If the total drowsy score exceeds a predefined limit, the system notifies the driver with a voice message, prompting them to take a rest.

Overall, this driver drowsiness detection system combines facial landmark analysis, fatigue detection, and fall asleep detection to enhance driver safety by identifying and addressing potential drowsiness-related risks during driving.  
  
Our module will be using a large dataset of videos of people drowsiness to learn and understand how people look when they are tired and falling asleep,  
while our seniors can only detect drivers or only faces that are physically located in front of them, our system will be able to detect almost anywhere we will located our camera and in any field of work.

1. **BACKGROUND**
   1. **Supervised learning**

Supervised learning is a type of machine learning where a model learns to make predictions on new inputs by training on labeled data[11]. The goal is to learn a mapping function from the input features to the output labels. The model adjusts its internal parameters during training to minimize the difference between its predicted output and the actual output label. Supervised learning is used for tasks such as classification and regression, and is applied in various applications such as customer churn prediction, fraud detection, and image recognition.

* 1. **YOLO**

YOLO (You Only Look Once)[12] is an object detection model that divides an input image into a grid and predicts bounding boxes and class probabilities for each cell. The model is trained on labeled images and learns to detect objects of interest with high accuracy. YOLO can perform real-time detection and detect multiple objects in a single pass but can struggle with small objects and produce false positives or false negatives.

* 1. **OpenCV** Icon

     Description automatically generated

OpenCV[13], also known as Open Source Computer Vision Library, is a popular and widely used open-source software library. It is designed specifically for computer vision and machine learning applications. By leveraging its extensive collection of over 2500 optimized algorithms, OpenCV enables developers to perform a diverse range of tasks in the field of computer vision.

Some of the key functionalities offered by OpenCV include face detection, which allows for the automatic identification and localization of human faces within images or video streams. Additionally, the library provides robust algorithms for object identification, enabling the recognition and categorization of various objects in real-time.

OpenCV also excels in human action classification, allowing developers to analyze and classify different human actions captured in video footage. This functionality finds applications in areas such as video surveillance, motion analysis, and gesture recognition.

For camera tracking and motion analysis, OpenCV offers sophisticated algorithms that can track camera movements and moving objects within a video sequence. These features are instrumental in applications such as video stabilization, object tracking, and motion-based object detection.

The library also facilitates 3D modeling by providing algorithms to extract detailed 3D models of objects from images or reconstructing 3D scenes using stereo cameras. This capability is beneficial in fields such as virtual reality, robotics, and autonomous navigation.

Image stitching is another powerful feature of OpenCV, allowing developers to seamlessly combine multiple images into a high-resolution composite image. This technique is useful for creating panoramic images or generating a complete view of a large scene from a set of overlapping images.

OpenCV's image database search capabilities enable efficient retrieval of similar images from a database based on visual similarity. This functionality is valuable in applications such as content-based image retrieval and image indexing.

The library also includes algorithms for specific tasks like red-eye removal, eye movement tracking, and scenery recognition. These features contribute to enhancing image quality, eye-tracking applications, and scene understanding, respectively.

Finally, OpenCV supports augmented reality by enabling the placement of virtual markers or objects on real-world scenes, facilitating the overlay of digital content onto live video feeds.

Overall, OpenCV's broad range of functionalities and its commitment to being an open-source platform have made it an essential tool in the field of computer vision, fostering innovation and driving the integration of machine perception in commercial products.

* 1. **Nvidia Jetson**

The NVIDIA Jetson Nano[14] is a small, low-power embedded computing device designed for AI and machine learning applications. It's powered by an ARM processor and NVIDIA's Maxwell GPU architecture, which provides high-performance computing for edge devices and embedded systems. The Jetson Nano is pre-installed with software, including the NVIDIA JetPack SDK, which allows developers to get started quickly and easily with building and training AI models, running inference, and deploying applications to the device. It has a range of connectivity options and is suitable for a wide range of applications such as robotics, drones, smart cameras, and IoT devices.

* 1. **Unawareness score**

We are going to count a score that will represent the unawareness level of the guard.

Score >= 15 – yellow alert.

Score >= 30 – red alert.

Scoring system:

**Semi unaware:**

blinking:

0 < blinking per minute <= 20 – 1 points

20 < blinking per minute <= 40 – 4 points

40 < blinking per minute <= 60 – 5 points

yawn:

1 <= yawns per minute=< 2 – 3 points

yawns per minute > 2 – 6 points

body:

head rests on one arm – 3 points

head rests on two arms – 6 points

sloppy body position – 6 points

**Totally unaware:**

Closed eyes:

blinking per minute = 0 = eyes closed – 15

Falling head:

No face detect - 15

1. **EXPECTED RESULTS**

We expect that the security guard awareness detection system developed in this project will improve the safety and effectiveness of security guards. Specifically, we expect the system to achieve high accuracy, real-time alerts, adaptability to different settings, and generalization to different scenarios. The expected results will be validated through experiments and evaluations on real-world datasets, and compared against existing state-of-the-art methods.

1. **ENGINEERING PROCESS**

Initially, we conducted research on the causes of security incidents and discovered that security guard errors were the primary cause. Determined to address this issue, we embarked on a two-step process. Our first step was to explore techniques for identifying unawareness, which other projects in this field had primarily focused on detecting whether the guard was asleep. Instead, we aimed to develop a system that could detect unawareness and alert the guard before any incident occurred. In the second step, we considered several options for detecting unawareness and conducted experiments to determine the most effective method. Ultimately, we chose to focus on behavioral detection, including measuring the frequency of blinks per minute, monitoring body language, and detecting yawning, using the YOLO model, which can detect objects and various situations in an image. With some experience in using the YOLO model to detect objects from the COCO dataset, we proceeded to design the system architecture.

1. **PROJECT DESIGN**

**6.1. Project Main flow**

This section describes how our system should work. During all security guard shifts, there is a camera that captures frames in real-time and makes calculations on the captured frames in the background using the YOLO model for object detection. The process contains the following steps:

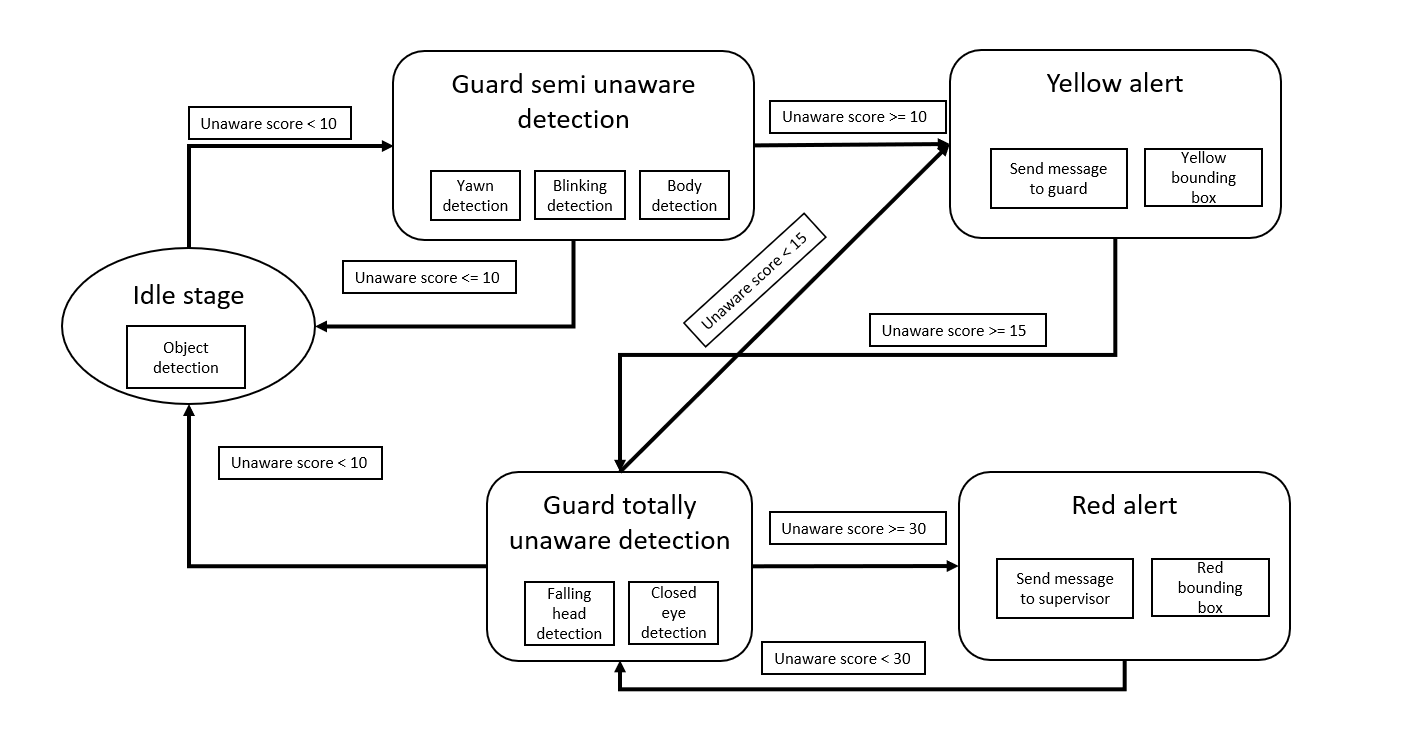
The system will capture one frame from the camera feed each time.

The YOLO model will be used to detect objects of interest, such as blinking, yawning, body position, awareness, and unawareness.

If any suspicious objects are detected, the system will take appropriate action, notifying the security guard, or initiating further investigation.

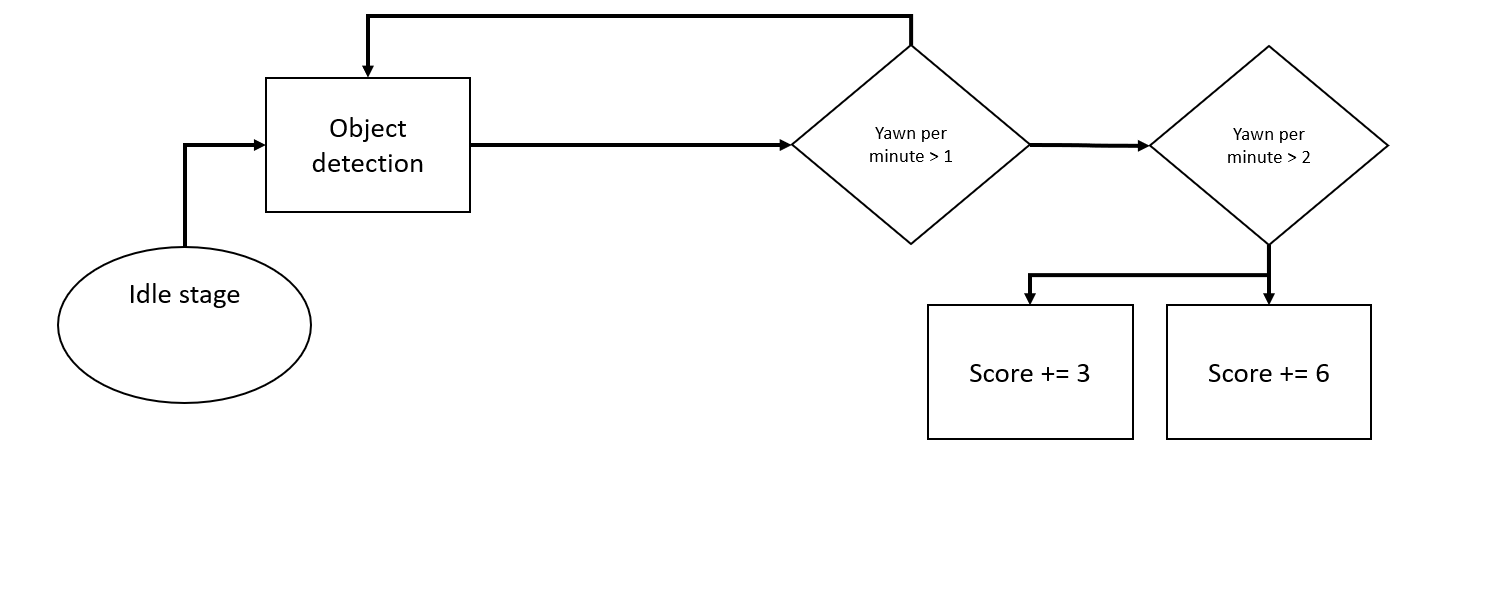
If any concerning behavior is identified, the system will notify the security guard and display relevant information on the monitor or control panel.

The system will continue to monitor the camera feed, capturing and analyzing frames in real-time, to ensure the security of the premises.

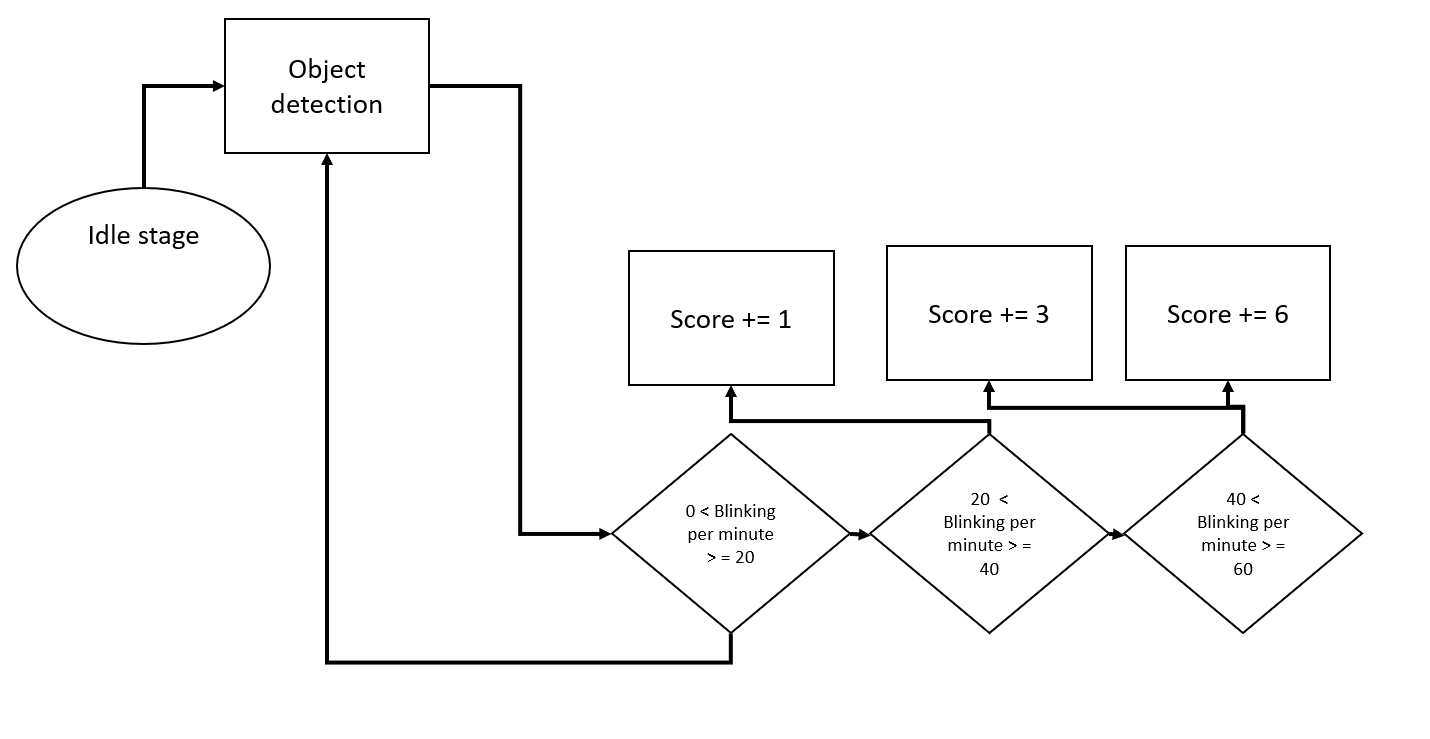
****

**6.2 Semi Unaware Detection**

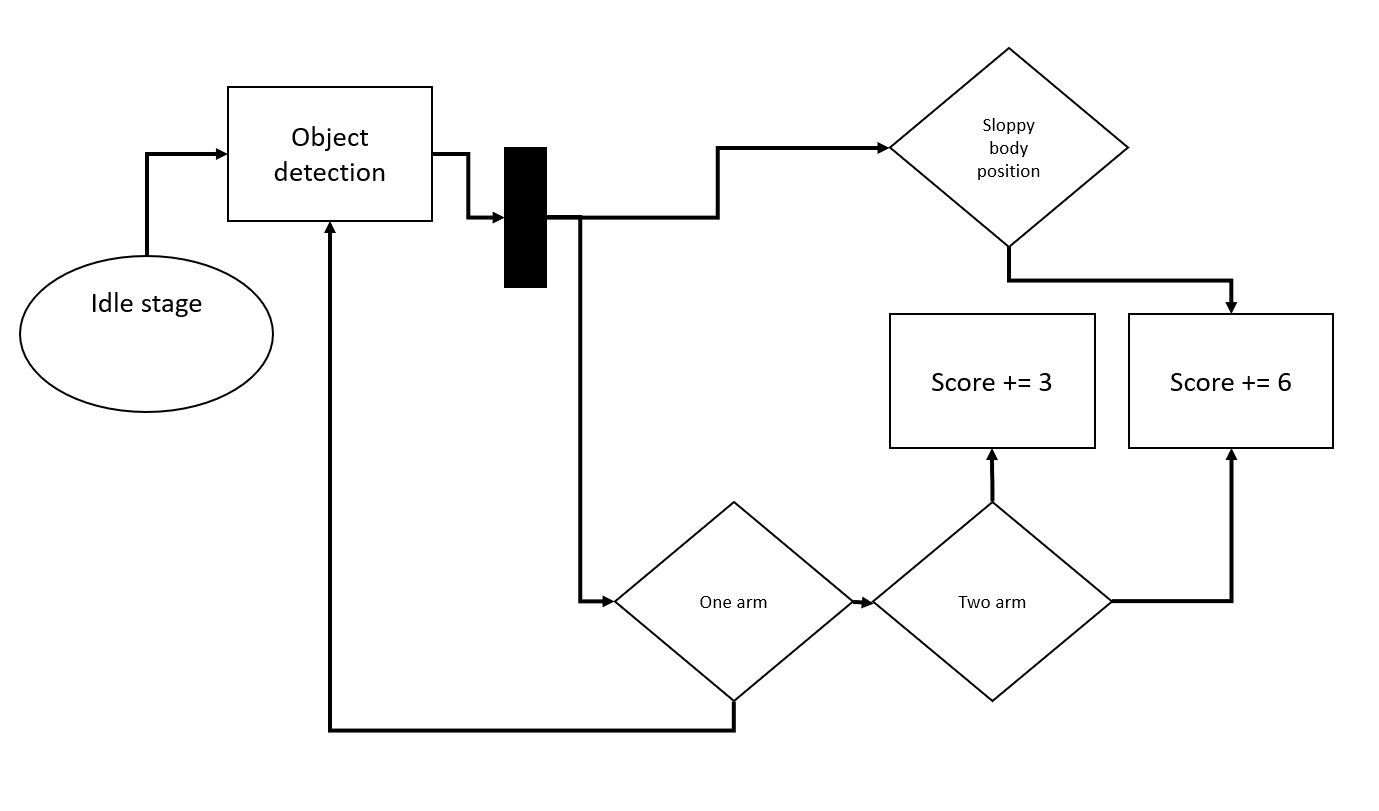
**6.2.1 Yawn Detection**

****

**6.2.2 Blinking Detection**

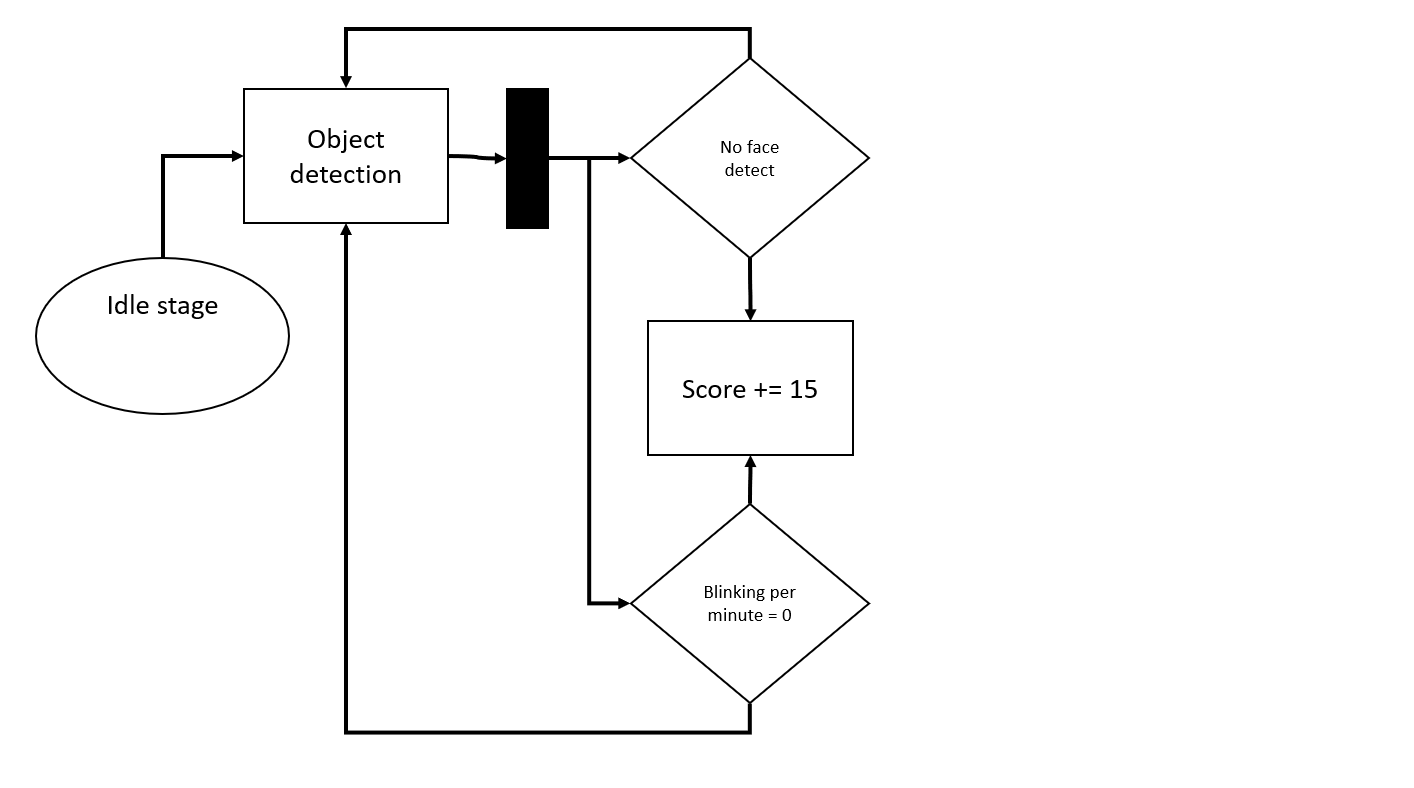
****

**6.2.3 Body Position Detection**

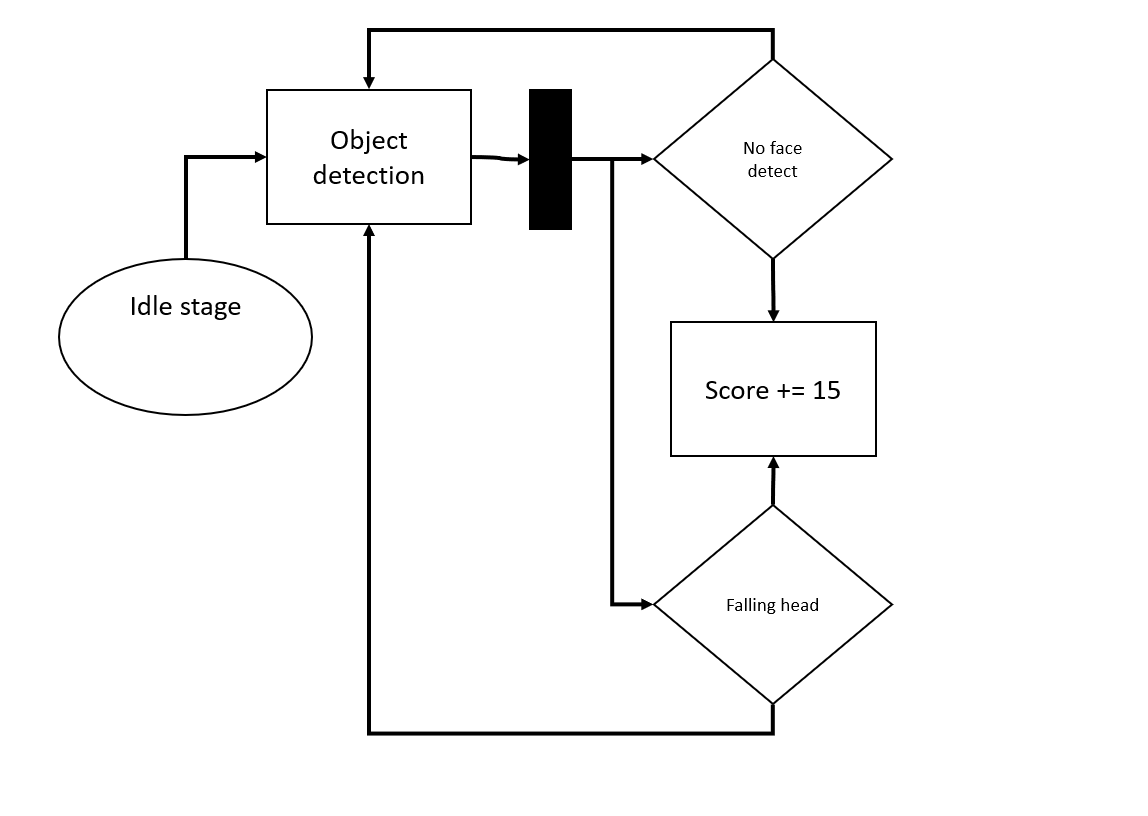
****

**6.3 Totaly Unaware Detection**

**6.3.1 Close Eye Detection**

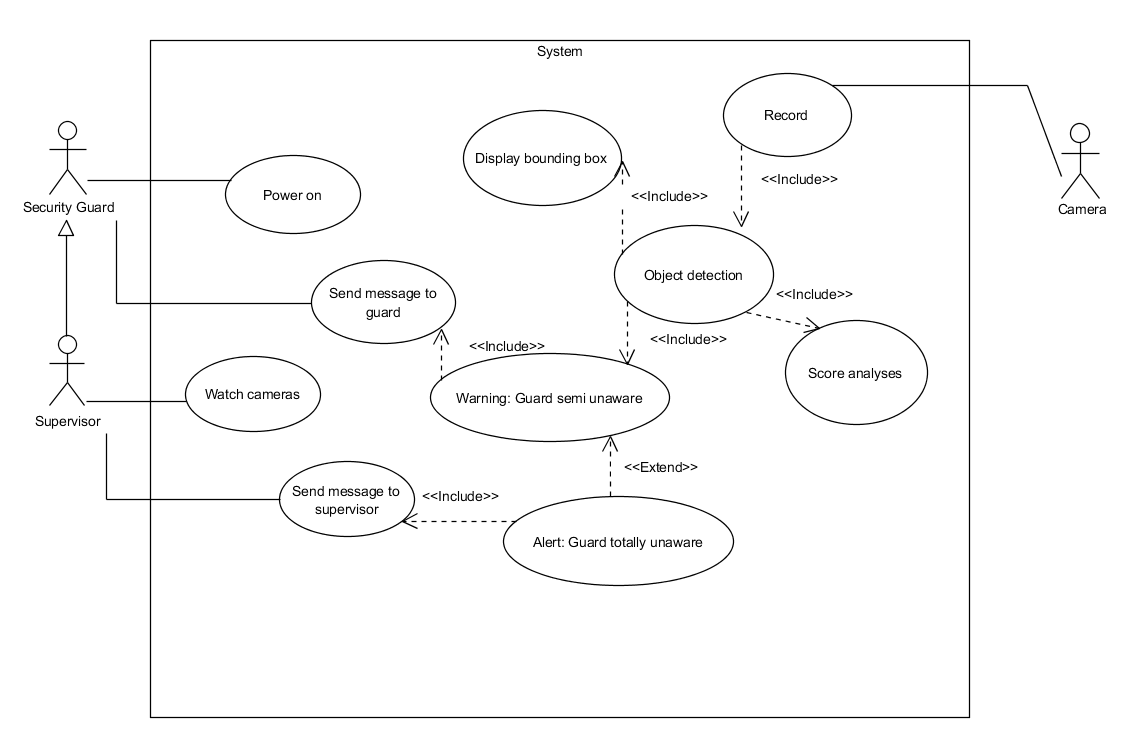
****

**6.3.2 Falling Head Detection**

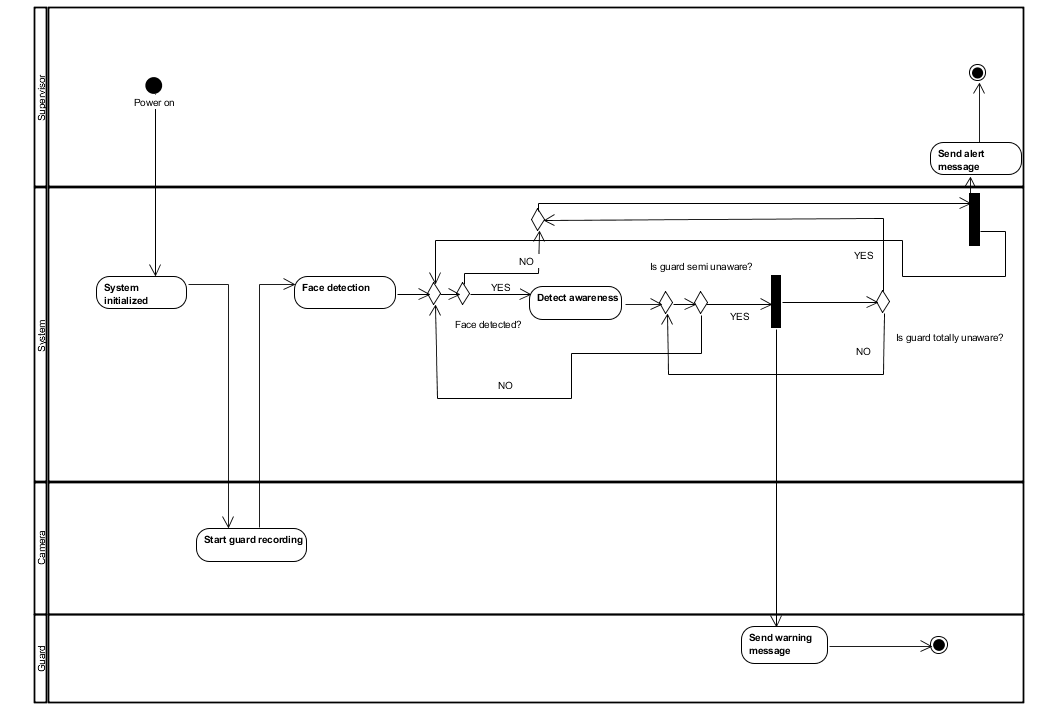


**6.4 Project Diagrams**

**6.4.1 Use Case Diagram**



**6.4.2 Activity Diagram**



1. **Verification plan**

**Test 1: Camera Test**

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

**| Connect camera to system |**

**| Launch program for camera footage display |**

**| Verify camera footage is displayed correctly |**

**| Troubleshoot if camera footage is incorrect |**

**Test 2: Face Recognition Test**

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

**| Turn on face recognition module in system |**

**| Place subject in front of camera |**

**| System recognizes face and alerts operator |**

**| Troubleshoot if system fails to recognize face |**

**Test 3: Blink Detection Test**

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

**| Activate blink detection module in system |**

**| Ask subject to sit in front of camera |**

**| System detects when subject blinks |**

**| Troubleshoot if system fails to detect blinks |**

**Test 4: Yawning Detection Test**

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

**| Activate yawning detection module in system |**

**| Ask subject to sit in front of camera |**

**| System detects when subject yawns |**

**| Troubleshoot if system fails to detect yawns |**

**Test 5: Sloppy Detection Test**

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

**| Activate sloppy detection module in system |**

**| Ask subject to sit in front of camera |**

**| System detects when subject is sloppy |**

**| Troubleshoot if system fails to detect sloppy positions |**

**Test 6: Full Sleep Detection Test**

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

**| Activate full sleep detection module in system |**

**| Ask subject to sit in front of camera |**

**| System detects when subject appears to be asleep or dozing off |**

**| Troubleshoot if system fails to detect full sleep |**

**Test 7: System Calculation Test**

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

**| Compile data from all previous tests |**

**| Run system's scoring algorithm to calculate final score |**

**| Verify system accurately calculates points for each test |**

**Test 8: Blink Count Test**

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

**| Activate blink counting module in system |**

**| Ask subject to sit in front of camera |**

**| System counts number of blinks made by subject |**

**| Calculate points based on system's scoring algorithm |**

**Test 9: Yawn Count Test**

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

**| Activate yawning counting module in system |**

**| Ask subject to sit in front of camera |**

**| System counts number of yawns made by subject |**

**| Calculate points based on system's scoring algorithm |**

**Test 10: Sloppy Count Test**

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

**| Activate sloppy counting module in system |**

**| Ask subject to sit in front of camera |**

**| System counts number of times subject assumes sloppy position |**

**| Calculate points based on system's scoring algorithm |**

**TEST 1: CAMERA TEST**

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

**| Connect camera to system |**

**| Launch program for camera footage display |**

**| Verify camera footage is displayed correctly |**

**| Troubleshoot if camera footage is incorrect |**

**TEST 2: FACE RECOGNITION TEST**

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

**| Turn on face recognition module in system |**

**| Place subject in front of camera |**

**| System recognizes face and alerts operator |**

**| Troubleshoot if system fails to recognize face |**

**TEST 3: BLINK DETECTION TEST**

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

**| Activate blink detection module in system |**

**| Ask subject to sit in front of camera |**

**| System detects when subject blinks |**

**| Troubleshoot if system fails to detect blinks |**

**TEST 4: YAWNING DETECTION TEST**

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

**| Activate yawning detection module in system |**

**| Ask subject to sit in front of camera |**

**| System detects when subject yawns |**

**| Troubleshoot if system fails to detect yawns |**

**TEST 5: SLOPPY DETECTION TEST**

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

**| Activate sloppy detection module in system |**

**| Ask subject to sit in front of camera |**

**| System detects when subject is sloppy |**

**| Troubleshoot if system fails to detect sloppy positions |**

**TEST 6: FULL SLEEP DETECTION TEST**

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

**| Activate full sleep detection module in system |**

**| Ask subject to sit in front of camera |**

**| System detects when subject appears to be asleep or dozing off |**

**| Troubleshoot if system fails to detect full sleep |**

**TEST 7: SYSTEM CALCULATION TEST**

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

**| Compile data from all previous tests |**

**| Run system's scoring algorithm to calculate final score |**

**| Verify system accurately calculates points for each test |**

**TEST 8: BLINK COUNT TEST**

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

**| Activate blink counting module in system |**

**| Ask subject to sit in front of camera |**

**| System counts number of blinks made by subject |**

**| Calculate points based on system's scoring algorithm |**

**TEST 9: YAWN COUNT TEST**

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

**| Activate yawning counting module in system |**

**| Ask subject to sit in front of camera |**

**| System counts number of yawns made by subject |**

**| Calculate points based on system's scoring algorithm |**

**TEST 10: SLOPPY COUNT TEST**

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

**| Activate sloppy counting module in system |**

**| Ask subject to sit in front of camera |**

**| System counts number of times subject assumes sloppy position |**

**| Calculate points based on system's scoring algorithm |**

**[1]Security Industry Association (SIA). (2018). Security megatrends: The top 10 trends shaping the security industry. Retrieved from** [**https://www.securityindustry.org/education-resources/security-megatrends/**](https://www.securityindustry.org/education-resources/security-megatrends/)

**[2]National Institute for Occupational Safety and Health (NIOSH). (2013). Police and security officer fatigue: Current trends and issues. Retrieved from** [**https://www.cdc.gov/niosh/docs/2013-138/pdfs/2013-138.pdf**](https://www.cdc.gov/niosh/docs/2013-138/pdfs/2013-138.pdf)

**[3]International Foundation for Protection Officers (IFPO). (2018). Sleep and fatigue: The impact on security officers. Retrieved from** [**https://www.ifpo.org/resource-links/industry-news/sleep-and-fatigue-the-impact-on-security-officers/**](https://www.ifpo.org/resource-links/industry-news/sleep-and-fatigue-the-impact-on-security-officers/)

# [4]A novel Capsule Neural Network based model for drowsiness detection using electroencephalography signals

# <https://www.sciencedirect.com/science/article/pii/S0957417422004031> [5] [Frontiers | A One-Dimensional CNN-LSTM Model for Epileptic Seizure Recognition Using EEG Signal Analysis (frontiersin.org)](https://www.frontiersin.org/articles/10.3389/fnins.2020.578126/full) <https://www.frontiersin.org/articles/10.3389/fnins.2020.578126/full> [6] [Abnormal Heart Rate Variability as a Manifestation of Autonomic Dysfunction in Hemispheric Brain Infarction | Stroke (ahajournals.org)](https://www.ahajournals.org/doi/full/10.1161/01.STR.27.11.2059) <https://www.ahajournals.org/doi/full/10.1161/01.STR.27.11.2059> [7] [Drowsiness monitoring based on steering wheel status - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S1361920917306582) <https://www.sciencedirect.com/science/article/pii/S1361920917306582> [8] [Iris localization using rough entropy and CSA: A soft computing approach - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S1568494618301066) <https://www.sciencedirect.com/science/article/pii/S1568494618301066> [9] [Sensors | Free Full-Text | Convolutional Neural Network for Drowsiness Detection Using EEG Signals (mdpi.com)](https://www.mdpi.com/1424-8220/21/5/1734) <https://www.mdpi.com/1424-8220/21/5/1734> [10] [dlib C++ Library](http://dlib.net/) http://dlib.net/ [11] [Supervised Learning | SpringerLink](https://link.springer.com/chapter/10.1007/978-3-540-75171-7_2) <https://link.springer.com/chapter/10.1007/978-3-540-75171-7_2> [12] [[1506.02640] You Only Look Once: Unified, Real-Time Object Detection (arxiv.org)](https://arxiv.org/abs/1506.02640) <https://arxiv.org/abs/1506.02640> [13] [OpenCV - Open Computer Vision Library](https://opencv.org/) https://opencv.org/ [14] [Introducing the NVIDIA Jetson Nano - Hackster.io](https://www.hackster.io/news/introducing-the-nvidia-jetson-nano-aaa9738ef3ff) https://www.hackster.io/news/introducing-the-nvidia-jetson-nano-aaa9738ef3ff