**VIGILANT EYE**

**A Computer-Vision Based Security System**

Software Requirement Specifications

Bachelor of Science in Software Engineering

By

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Name** | **Registration #** | **Mobile #** | **E-Mail** |
| 1 | Muhammad Hashim Aqeel | BSSE-CSS20-7019 | 03099449779 | mhashim0224@gmail.com |
| 2 | Faraz Mehmood | ???? | 03034450790 | faraz.m4765@gmail.com |

**Supervised by Co-Supervised by**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Teacher Name Teacher Name

****

Department of Software Engineering

Lahore Garrison University

Lahore

**Table of Contents**

[**1. Introduction** 4](#_Toc181291203)

[1.1 Purpose 4](#_Toc181291204)

[1.1.1 High Detection Accuracy 4](#_Toc181291205)

[Achieve a high level of accuracy in detecting anomalies to minimize false positives and false negatives, ensuring reliable security. 4](#_Toc181291206)

[1.1.2 Model Flexibility 4](#_Toc181291207)

[Allow users to select from ResNet, MobileNet or Inception (GoogLeNet) models based on specific accuracy requirements and performance preferences. 4](#_Toc181291208)

[1.1.3 Real-Time Monitoring 4](#_Toc181291209)

[Provide real-time analysis and detection to alert users immediately upon identifying any unusual activity. 4](#_Toc181291210)

[1.1.4 User-Friendly Interface 4](#_Toc181291211)

[1.1.5 Scalability and Adaptability 4](#_Toc181291212)

[**2. Product Scope** 5](#_Toc181291213)

[2.1 AI Model Selection 5](#_Toc181291215)

[2.2 Real-Time Anomaly Detection 5](#_Toc181291217)

[2.3 Accuracy Comparison 5](#_Toc181291219)

[2.4 Intuitive User Interface 5](#_Toc181291220)

[2.5 Data Input Compatibility 5](#_Toc181291221)

[2.6 Scalable Architecture 5](#_Toc181291222)

[2.7 High Performance 5](#_Toc181291223)

[**3. Objectives and Goals** 6](#_Toc181291224)

[3.1 Precision in Anomaly Detection 6](#_Toc181291225)

[3.2 Accessibility and User-Friendly Interface 6](#_Toc181291227)

[3.3 Integration of Advanced Machine Learning Algorithms 6](#_Toc181291228)

[3.4 Empowerment through Data-Driven Insights 6](#_Toc181291229)

[3.5 Utility for Security Professionals 7](#_Toc181291231)

[**4. Product Perspective** 7](#_Toc181291233)

[**5. Product Functions** 7](#_Toc181291235)

[5.1 User Authentication 7](#_Toc181291227)

[5.2 Input Interface 8](#_Toc181291225)

[5.3 Machine Learing Models 8](#_Toc181291228)

[5.4 Anomaly Detection Analysis 8](#_Toc181291229)

[5.5 User Management 8](#_Toc181291231)

[5.6 Performance Optimization 8](#_Toc181291229)

[5.7 Clear System Communication 8](#_Toc181291231)

[5.8 Usability Enhancement 9](#_Toc181291229)

[5.9 Continous Improvement Mechanism 9](#_Toc181291231)

[**6. Operating Environment** 9](#_Toc181291235)

[**7. Software Dependencies** 9](#_Toc181291235)

[**8. Functional Requirements** 10](#_Toc181291235)

[**9. Non-Functional Requirements** 11](#_Toc181291235)

[**References** 13](#_Toc181291236)

# **1. Introduction**

## 1.1 Purpose

The purpose of the **Vigilant Eye** anomaly detection system is to provide a reliable and accessible security solution for detecting anomalies in various environments. This system leverages advanced machine learning and artificial intelligence models, including **ResNet**, **MobileNet**, and **Inception (GoogLeNet)**, to analyze diverse patterns and detect unusual activities. The primary goals are to ensure high detection accuracy, offer flexibility for users to choose the most suitable AI model based on accuracy requirements, and deliver rapid, reliable security insights to safeguard against potential threats. The primary goals are:

## 1.1.1 High Detection Accuracy

## Achieve a high level of accuracy in detecting anomalies to minimize false positives and false negatives, ensuring reliable security.

## 1.1.2 Model Flexibility

## Allow users to select from ResNet, MobileNet or Inception (GoogLeNet) models based on specific accuracy requirements and performance preferences.

## 1.1.3 Real-Time Monitoring

# Provide real-time analysis and detection to alert users immediately upon identifying any unusual activity.

## 1.1.4 User-Friendly Interface

Design an intuitive interface that makes it easy for users to set preferences, select models, and view anomaly detection results.

## 1.1.5 Scalability and Adaptability

Enable the system to scale for larger datasets and adapt to different types of environments, ensuring versatility across various security applications.

# **2. Product Scope**

## The scope of Vigilant Eye is to deliver a high-accuracy, AI-driven anomaly detection solution that can identify unusual activities across diverse environments. Its primary functionalities include real-time anomaly detection, model flexibility (ResNet, MobileNet, Inception) and an intuitive interface for selecting models based on desired accuracy. The Vigilant Eye encompasses the following features and functionalities:

## 2.1 AI Model Selection

## Users can choose between ResNet, MobileNet, and Inception (GoogLeNet) models to optimize anomaly detection accuracy according to their specific needs.

## 2.2 Real-Time Anomaly Detection

## Analyzes incoming data (e.g., video feeds, sensor data) in real time to identify any unusual or suspicious patterns.

## 2.3 Accuracy Comparison

Compares the accuracy of each AI model, allowing users to make informed choices based on model performance

## 2.4 Intuitive User Interface

Provides a simple, user-friendly interface to facilitate model selection, data input configuration, and viewing of anomaly detection results.

## 2.5 Data Input Compatibility

Supports integration with various data sources like video surveillance feeds, IoT sensors, and system logs for adaptable, context-specific anomaly detection.

## 2.6 Scalable Architecture

Designed to handle different data volumes, allowing Vigilant Eye to scale for larger datasets or high-traffic environments.

## High Performance

Optimized for rapid processing to ensure timely detection, even under high data influx, making Vigilant Eye suitable for mission-critical applications that demand real-time responsiveness.

# **3. Objectives and Goals**

The Vigilant Eye Anomaly Detection System is designed with clear objectives and goals to address the need for accurate and efficient anomaly identification in various security environments. The system's objectives focus on providing a reliable, user-friendly, and advanced technological solution suitable for diverse applications. The overarching goals are as follows:

## 3.1 Precision in Anomaly Detection

## Objective: Develop a high-precision model that accurately detects unusual patterns or anomalies within monitored data. Goal: Surpass traditional methods by achieving superior detection accuracy, ensuring consistent, reliable anomaly identification across varied environments.

## 3.2 Accessibility and User-Friendly Interface

**Objective**: Create an intuitive user interface that simplifies the selection and usage of AI models.  
**Goal**: Enable users, regardless of technical proficiency, to effortlessly interact with the system and view detection results in an accessible format.

## 3.3 Integration of Advanced Machine Learning Algorithms

**Objective**: Leverage advanced AI models—ResNet, MobileNet, and Inception (GoogLeNet**)**—to provide robust anomaly detection capabilities.  
**Goal**: Highlight ResNet as the primary model due to its high accuracy, while allowing users flexibility in selecting the best model for their needs.

## 3.4 Empowerment through Data-Driven Insights

# **Objective:** Empower users with clear, data-driven insights into anomaly patterns and occurrences. **Goal:** Provide visualizations and detailed results to aid users in understanding the context and significance of detected anomalies.

## 3.5 Utility for Security Professionals

# **Objective:** Offer a practical, reliable tool for security professionals to identify potential threats or unusual activities in real time. **Goal:** Enhance the accuracy and efficiency of threat detection for professionals by providing an advanced anomaly detection solution.

**3.6 Privacy and Security**

**Objective**: Ensure robust security with user authentication to protect access and data integrity.  
**Goal**: Safeguard user data through encryption and secure handling practices, maintaining privacy and confidentiality.

**3.7 Usability and Scalability**

**Objective**: Design the system for optimal usability and scalability to accommodate various data sources and usage volumes.  
**Goal**: Ensure the system is responsive and capable of handling diverse, high-volume datasets for various security applications.

# **4. Product Perspective**

## The *Vigilant Eye* Anomaly Detection System aligns with advancements in security and monitoring technology. Designed to be user-friendly, it offers a straightforward experience for users seeking efficient and accurate anomaly detection. Leveraging advanced AI models (ResNet, MobileNet, and Inception), the system stays current with the latest trends in machine learning for security applications. *Vigilant Eye* does not replace existing security infrastructure but complements it, providing an additional layer of proactive monitoring and detection. Its scalable design ensures adaptability to growing data needs and evolving security landscapes, making it a reliable tool for both individuals and security professionals aiming to enhance monitoring effectiveness.

# **5. Product Functions**

#### **5.1 User Authentication**

* **Function**: Allows users to log in securely.
* **Purpose**: Ensures personalized access to the system, maintaining privacy and data security.

#### **5.2 Input Interface**

* **Function**: Provides a user-friendly interface for users to input data sources, such as video feeds and sensor data.
* **Purpose**: Enables users to easily enter relevant data for personalized anomaly detection.

#### **5.3 Machine Learning Models**

* **Function**: Integrates ResNet, MobileNet, and Inception (GoogLeNet) for anomaly detection.
* **Purpose**: Utilizes advanced algorithms to analyze diverse data inputs and generate accurate anomaly detection results.

#### **5.4 Anomaly Detection Analysis**

* **Function**: Analyzes incoming data in real-time to detect anomalies.
* **Purpose**: Provides timely insights by identifying unusual patterns, helping users recognize potential security threats.

#### **5.5 User Management**

* **Function**: Manages user accounts and model preferences securely.
* **Purpose**: Safeguards user information and ensures controlled access to personalized anomaly detection results.

#### **5.6 Performance Optimization**

* **Function**: Ensures efficient response times to user inputs, maintaining a seamless and responsive user experience.
* **Purpose**: Enhances user satisfaction by providing timely access to anomaly detection results.

#### **5.7 Clear System Communication**

* **Function**: Clearly communicates that Vigilant Eye is a tool for anomaly detection and not a standalone security solution.
* **Purpose**: Promotes responsible use by setting accurate expectations for the system’s role.

#### **5.8 Usability Enhancement**

* **Function**: Designs the system for optimal usability, making it accessible to users with varying levels of technical expertise.
* **Purpose**: Ensures a user-friendly experience, encouraging widespread adoption across various industries.

#### **5.9 Continuous Improvement Mechanism**

* **Function**: Facilitates updates based on user feedback, AI advancements, and security trends.
* **Purpose**: Ensures that Vigilant Eye remains relevant and effective, adapting to new security challenges over time.

**6. Operating Environment**

The *Vigilant Eye* Anomaly Detection System will operate in the following environment:

* **Hardware Platform**: The software will be designed to run on standard desktop computers and laptops with modern specifications, ensuring compatibility with a wide range of hardware configurations.
* **Operating System**: The software will be compatible with multiple operating systems, including:
  + Windows
  + MacOS
  + Linux (optional, depending on user demand)

This multi-platform support allows for flexibility in deploy

**7. Software Dependencies**

* **Python Programming Language**: Used for creating, implementing, training, and deploying machine learning models.
* **Jupyter Notebook**: The programming interface used for developing and running the code, providing an interactive environment for data analysis and model development.
* **Python Programming Libraries**: The following libraries will be utilized for implementing, evaluating, and deploying machine learning models:
  + **Scikit-learn**: For machine learning algorithms and model evaluation.
  + **Seaborn**: For statistical data visualization.
  + **Matplotlib**: For creating static, animated, and interactive visualizations in Python.
  + **Pickle**: For serializing and de-serializing Python objects, such as machine learning models.
  + **Streamlit**: For building and deploying interactive web applications.
  + **Pandas**: For data manipulation and analysis.
  + **NumPy**: For numerical computations and array operations.

**8. Functional Requirements**

The functional requirements collectively define the capabilities and behaviors of the *Vigilant Eye* Anomaly Detection System, ensuring its effectiveness in detecting anomalies and supporting security management. The following are the functional requirements of the system:

**8.1 User Authentication:**

* **Requirement**: The system shall provide secure user authentication, allowing users to log in.
* **Purpose**: Ensure personalized access to anomaly detection features while maintaining user privacy.

**8.2 Input Interface:**

* **Requirement**: The system shall feature a user-friendly interface for users to input data sources, such as video feeds and sensor data.
* **Purpose**: Facilitate easy and intuitive entry of relevant data for personalized anomaly detection.

**8.3 Machine Learning Models:**

* **Requirement**: The system shall integrate ResNet, MobileNet, and Inception (GoogLeNet) for anomaly detection.
* **Purpose**: Utilize advanced algorithms to analyze diverse data inputs and enhance the accuracy of anomaly detection.

**8.4 Anomaly Detection:**

* **Requirement**: The system shall analyze incoming data in real-time to detect anomalies.
* **Purpose**: Provide timely insights by identifying unusual patterns, helping users recognize potential security threats.

**8.5 Model Selection:**

* **Requirement**: The system shall allow users to select from ResNet, MobileNet, and Inception (GoogLeNet) based on their specific needs and preferences.
* **Purpose**: Provide users with flexibility in choosing the best model for their anomaly detection requirements.

**8.6 Performance Metrics:**

* **Requirement**: The system shall provide performance metrics (such as accuracy, precision, recall, etc.) for each selected model.
* **Purpose**: Enable users to make informed decisions based on the effectiveness of the different models.

**9 Non-Functional Requirements**

These non-functional requirements provide the framework for the *Vigilant Eye* Anomaly Detection System's performance, security, usability, scalability, ethical considerations, compatibility, reliability, data integrity, interoperability, adaptability, compliance, and documentation. They ensure that the system meets the highest standards in terms of both functionality and user experience.

**9.1 Performance:**

* **Requirement**: The system shall respond to user inputs within seconds.
* **Purpose**: Enhance user satisfaction by providing a responsive and efficient experience.

**9.2 Usability:**

* **Requirement**: The user interface shall be intuitive and accessible to users.
* **Purpose**: Facilitate widespread adoption by ensuring a user-friendly experience.

**9.3 Scalability:**

* **Requirement**: The system shall be capable of handling data efficiently, accommodating increasing data loads.
* **Purpose**: Ensure adaptability to varying data volumes for scalability.

**9.4 Compatibility:**

* **Requirement**: The system shall be compatible with major web browsers and devices.
* **Purpose**: Ensure accessibility across different platforms.

**9.5 Reliability:**

* **Requirement**: The system shall have an uptime of at least 99%.
* **Purpose**: Ensure continuous availability for users.

**9.6 Data Integrity:**

* **Requirement**: The system shall validate and maintain the integrity of user-inputted data.
* **Purpose**: Ensure the accuracy and reliability of anomaly detection results.

**9.7 Adaptability:**

* **Requirement**: The system shall allow for future enhancements and updates.
* **Purpose**: Ensure continuous improvement and adaptability to changing technological standards.

**9.8 Robustness:**

* **Requirement**: The system shall gracefully handle unexpected inputs or errors, providing meaningful feedback to users.
* **Purpose**: Enhance the system's resilience in environments where unexpected situations may arise.

**9.9 Maintainability:**

* **Requirement**: The system shall be designed with well-documented code for ease of maintenance by IT professionals.
* **Purpose**: Facilitate efficient upkeep and troubleshooting to minimize downtime and support continuous use.

# **References**

# Vigilant Eye Project Documentation: Internal documentation detailing the development and specifications of the *Vigilant Eye* Anomaly Detection System.

# **ResNet, MobileNet, and Inception (GoogLeNet):**

# He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [Link to paper](https://arxiv.org/abs/1512.03385)

# Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., & et al. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. arXiv preprint arXiv:1704.04861. [Link to paper](https://arxiv.org/abs/1704.04861)

# Szegedy, C., Ioffe, S., Vanhoucke, V., & Alekh, A. (2015). *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. arXiv preprint arXiv:1602.07261. [Link to paper](https://arxiv.org/abs/1602.07261)

# **Machine Learning Libraries:**

# Pedregosa, F., Varoquaux, G., Gramfort, A., & et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825-2830.

# McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. In Proceedings of the 9th Python in Science Conference.

# Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 9(3), 90-95.

# Waskom, M. (2021). *seaborn: statistical data visualization*. Journal of Open Source Software, 6(60), 3021.

# **Web Development:**

# Flanagan, D. (2011). *JavaScript: The Definitive Guide*. O'Reilly Media. Link to book

# W3C. (2021). *Web Content Accessibility Guidelines (WCAG) 2.1*. [W3C Recommendation](https://www.w3.org/TR/WCAG21/)

# **Software Engineering Principles:**

# Sommerville, I. (2011). *Software Engineering* (9th ed.). Addison-Wesley. Link to book

# Pohl, K. (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques*. Springer. [Link to book](https://www.springer.com/gp/book/9783642043808)

# **Ethics in Technology:**

# Binns, R. (2018). *Fairness in Machine Learning: Lessons from Political Philosophy*. In Proceedings of the 2018 Conference on Fairness, Accountability, and Transparency. Link to paper

# Dastin, J. (2018). *Amazon Scraps Secret AI Recruiting Tool That Showed Bias Against Women*. Reuters. Link to article