Project Proposal: Smart Eye

Project team

Team leader	21P0197	Omar Alaa Elnahass	21P0197@eng.asu.edu.eg
Team member	21P0012	Abdelrahman Mohamed Barakat	21P0012@eng.asu.edu.eg
Team member	2101544	Maeen Mohamed Sayed	2101544@eng.asu.edu.eg
Team member	21P0284	Tsneam Ahmed Eliwa	21P0284@eng.asu.edu.eg
Team member	21P0279	Salma Mohamed Sayed	21P0279@eng.asu.edu.eg

Background

The project "SmartEye" is undertaken to address the growing demand for effective real-time object detection technology. This need arises from various sectors such as security, surveillance, and automation, where timely identification and tracking of objects are crucial for decision-making and operational efficiency. By leveraging advancements in artificial intelligence and computer vision, SmartEye aims to provide a solution that enhances situational awareness through live camera feeds. The opportunity lies in creating a user-friendly web application that enables users to analyze live camera feeds, identify objects in real-time, and respond swiftly to potential threats or events. Additionally, the project aims to contribute to the advancement of object detection technology by showcasing expertise in AI and web development, ultimately providing a valuable tool for enhancing safety, security, and productivity in various domains.

Objectives

1. Develop and deploy a deep learning model for real-time object detection, achieving an accuracy of at least 85% on the validation dataset.
2. Implement a user-friendly web interface that allows users to access the live camera feed, view real-time object detections, and keep track of the number of objects detected.
3. Optimize the application's performance to ensure smooth real-time processing of the camera feed, minimizing latency and providing a responsive user experience.
4. Integrate features such as multi-object detection to enhance the application's functionality and utility in diverse scenarios.
5. Document the development process, including model architecture, training methodology, and software implementation, to facilitate knowledge sharing and future enhancements.

Scope

The end result of the SmartEye project would be a functional AI model capable of real-time object detection when connected to a camera. Phases of work that might be undertaken:

1. Research and Planning: This initial phase involves researching existing object detection models, understanding their strengths and weaknesses, and planning the architecture and implementation strategy for SmartEye.

The "You Only Look Once: Unified, Real-Time Object Detection" paper by Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, introduces the YOLO model, which revolutionizes real-time object detection by simultaneously predicting bounding boxes and class probabilities for objects in an image. The paper provides detailed insights into the architecture, training procedure, and optimization techniques used in YOLO, enabling developers to enhance the performance of their models and contribute to the broader community of researchers and developers in the field of computer vision. (https://pjreddie.com/darknet/yolo/)

Project Proposal: Smart Eye

2. Data Collection and Annotation: To train an effective AI model, a large dataset of images containing various objects of interest needs to be collected. Additionally, these datasets would need to be annotated, i.e., labeled with bounding boxes indicating the location and type of objects present. When it comes to data collection and annotation for training AI models, **Roboflow** is a powerful tool that can streamline the process. Gather a diverse set of images relevant to your problem domain, and then use Roboflow Annotate for ultra-fast labeling and real-time teamwork. Roboflow's features include automated labeling, collaboration tools, security measures, and various export formats.

3. Model Development: In this phase, the AI model architecture would be developed, based on the chosen approach (e.g., convolutional neural networks like YOLO). This involves designing the network layers, optimizing hyperparameters, and fine-tuning the model to achieve the desired performance metrics.

4. Training: Using the annotated dataset, the model would undergo training to learn how to accurately detect objects in images or video frames. This process involves feeding the data through the model, adjusting the model's parameters based on the error (loss) between predicted and ground truth annotations, and iteratively improving performance.

5. Testing and Evaluation: After training, the model needs to be thoroughly tested on a separate dataset to evaluate its performance. Metrics such as precision, recall, and F1 score would be used to assess how well the model detects objects of interest while minimizing false positives and false negatives.

6. Optimization and Fine-Tuning: Based on the evaluation results, the model may require further optimization and fine-tuning to address any performance issues or specific requirements. This could involve adjusting parameters, augmenting the training data, or refining the model architecture.

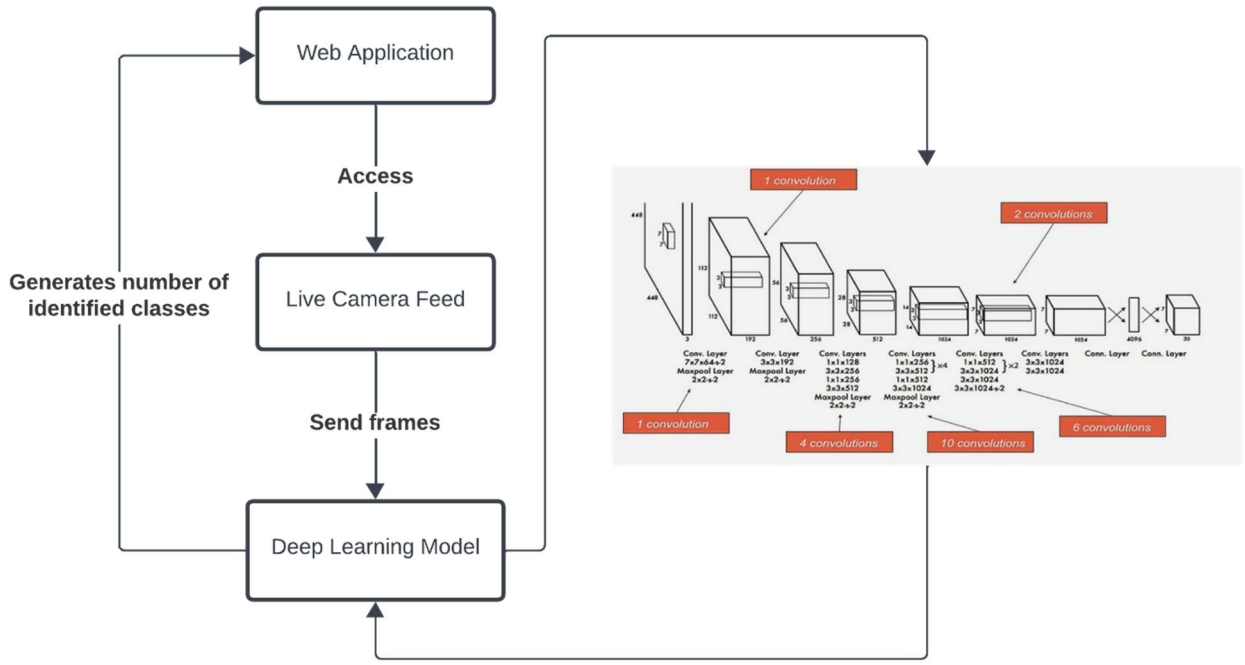
7. Integration and Deployment: Once the model meets the desired performance criteria, it can be integrated into the SmartEye system, which includes developing software to interface with cameras, process video streams, and display detection results. Deployment involves testing the integrated system in real-world scenarios and ensuring it functions reliably and accurately.

8. Monitoring and Maintenance: After deployment, ongoing monitoring and maintenance are essential to address any issues that arise, such as changes in the environment, new objects to detect, or updates to the underlying AI model.

To Sum Up, by completing these phases, the SmartEye project aims to deliver an effective AI model for real-time object detection, enhancing the capabilities of connected cameras for various applications such as security, surveillance, or automation.

Project Proposal: Smart Eye

System overview



To visually represent the SmartEye project's system overview and integration of different phases of work, a diagrammatic representation can be created. Here's a description of the diagram:

1. Components:

- **Live Camera Feed:** This represents the input source, which is the live feed from the camera.
- **Deep Learning Model:** The trained deep learning model responsible for object detection.
- **Web Application:** The interface through which users interact with the system.
- **Object Detection Output:** The identified objects highlighted and labeled on the live camera feed.
- **User Interface:** The user-facing part of the web application where the live camera feed and object detection output are displayed.

2. Integration:

- The live camera feed is continuously streamed into the deep learning model.
- The deep learning model processes the live feed, performing real-time object detection.
- Identified objects are visually highlighted and labeled on the live feed.
- The processed feed with object detection output is then displayed in real-time on the user interface of the web application.
- Users interact with the web application to view the live camera feed and the detected objects.

3. Phases of Work Integration:

1. **Research and Planning:** During this phase, the architecture of the system, including the integration of the deep learning model with the web application, is planned.
2. **Data Collection and Model Training:** The deep learning model is trained on a comprehensive dataset of various object categories, ensuring robust object detection capabilities.
3. **Model Development and Testing:** The developed model undergoes testing to ensure its effectiveness in real-time object detection.
4. **Web Application Development:** The web application is developed with a user-friendly interface, enabling users to view the live camera feed and object detection output seamlessly.
5. **Integration and Deployment:** The trained model is integrated into the web application, and the system is deployed for real-world usage.

6. **Monitoring and Maintenance:** Ongoing monitoring and maintenance ensure the system's reliability and performance over time.

This diagrammatic representation illustrates how the different phases of work, from research and planning to deployment and maintenance, are integrated to achieve the goal of developing the SmartEye system for real-time object detection through a live camera feed.

Timeframe for deliverables

	Description of Work	Start and End Dates
Phase One	Project Conceptualization and Ideation	24-02-2024 – 25-02-2024
Phase Two	AI Model Classes Selection	09-03-2024 – 10-03-2024
Phase Three	Dataset collecting and Model Training	10-03-2024 – 24-03-2024
Phase Four	Web Application Development and Dataset Refinement	18-04-2024 – 19-04-2024
Phase Five	Project Proposal Preparation	28-04-2024 – 29-04-2024

Monitoring and Evaluation

1. Objectives and Key Performance Indicators (KPIs)

Objective 1: Implement Real-Time Object Detection

- KPI 1: Achieve swift processing from feed capture to display to maintain seamless operation.
- KPI 2: Maintain high accuracy in object detection to ensure reliable results.

Objective 2: Ensure System Stability and Performance

- KPI 1: Guarantee continuous system availability during operational periods.
- KPI 2: Optimize feed loading and processing times for enhanced responsiveness.

Objective 3: Enhance User Experience and Satisfaction

- KPI 1: Continuously improve system features and functionalities to enhance user satisfaction.
- KPI 2: Implement personalized recommendations and tailored experiences to meet individual user needs and preferences.

2. Evaluation Methodology

1. Model Accuracy: The primary objective of the SmartEye project is to develop a deep learning model capable of accurately identifying and labeling objects in a live camera feed. The performance of this model will be evaluated using standard metrics such as precision, recall, and F1-score.

2. Real-Time Performance: A key feature of the SmartEye application is its ability to process and analyze live camera feeds in real-time. The performance of the application in this regard will be measured in frames per second (FPS).

3. System Stability: The stability of the SmartEye application under different conditions and loads is crucial for its successful implementation. System crashes or significant slowdowns will be tracked as indicators of system stability.

Upon completion of the project, a comprehensive evaluation will be conducted to assess the overall success of the project. This evaluation will include a comparison of the initial project goals with the actual outcomes, an analysis of any challenges or issues encountered during the project, and a review of the project's impact and benefits for the broader community.

Project Proposal: Smart Eye

3. Reporting

- Proposal Report: A project proposal outlines the purpose and scope of a project. It ensures that what the project will include is agreed on by clearly defining the project's objectives, methodology, and potential risks, the proposal sets expectations.
- Final Evaluation Report: Prepare a final evaluation report summarizing outcomes, lessons learned, and recommendations for future projects.

Approval Signatures

Supervisor	Teaching Assistance
Associate Professor / Walaa H. Elashmawi	Eng. Dina Eng. Ahmed