

Final Year Project

Inspection AI

Inspecting defects in EV batteries

FYP Team

Abbas Mustafa

Shayan Bakht

Ali Chohan

Supervised By

Mr. Hasan Raza

Mr Amad Nadeem

FAST School of Computing

National University of Computer and Emerging Sciences

Islamabad, Pakistan

2021

Anti-Plagiarism Declaration

This is to declare that the above FYP report produced under the:

Title: _____

is the sole contribution of the author(s) and no part hereof has been reproduced on as it is basis (cut and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined.

Date: _____

Student 1

Name: _____

Signature: _____

Student 2

Name: _____

Signature: _____

Student 3

Name: _____

Signature: _____

Supervisor (Faculty)

Name: _____

Signature: _____

Authors' Declaration

This states Authors' declaration that the work presented in the report is their own, and has not been submitted/presented previously to any other institution or organization.

Table of Contents

| | |
|--|---|
| Introduction | 5 |
| Literature Review and Architecture | 6 |
| Project Vision | 6 |
| Software Requirement Specification..... | 7 |
| Features: | 7 |
| Real-time Video Processing:..... | 7 |
| 3D Model Generation: | 7 |
| Object Detection and Recognition: | 7 |
| Anomaly Detection: | 7 |
| Measurement and Dimension Analysis: | 7 |
| Annotation and Reporting: | 7 |
| Integration with Data Systems: | 7 |
| Augmented Reality (AR) Overlay:..... | 7 |
| User Authentication and Authorization: | 7 |
| User-Friendly Interface: | 7 |
| Functional Requirements: | 8 |
| Video Stream Capture:..... | 8 |
| 3D Model Reconstruction: | 8 |
| Object Detection Algorithms: | 8 |
| Anomaly Detection Algorithms: | 8 |
| Measurement and Dimension Calculation:..... | 8 |
| AR Overlay Generation:..... | 8 |
| Data Storage and Management: | 8 |
| Notification and Alert System: | 8 |
| User Authentication and Roles: | 8 |
| Integration with External Systems: | 8 |
| Quality Attributes:..... | 9 |
| Accuracy: | 9 |
| Performance:..... | 9 |
| Scalability: | 9 |
| Reliability:..... | 9 |
| Security: | 9 |
| Usability: | 9 |
| Interoperability: | 9 |
| Maintainability: | 9 |

| | |
|--|----|
| Adaptability:..... | 9 |
| Privacy:..... | 9 |
| Non-Functional Requirements:..... | 10 |
| Response Time:..... | 10 |
| Processing Speed:..... | 10 |
| Data Storage Capacity:..... | 10 |
| Security Measures:..... | 10 |
| Compatibility:..... | 10 |
| Scalability Limits:..... | 10 |
| Accuracy Metrics:..... | 10 |
| Backup and Recovery:..... | 10 |
| User Training and Support:..... | 10 |
| Regulatory Compliance:..... | 10 |
| Iteration Plan..... | 11 |
| Iteration 1:..... | 11 |
| Iteration 2:..... | 11 |
| Iteration 1..... | 12 |
| Data Collection..... | 12 |
| Model Creation for 2D image to Depth Map..... | 12 |
| Model creation for 2D image + Depth map to 3D model..... | 12 |
| Creation of Front End..... | 12 |
| Implementation of Login System..... | 12 |
| Iteration 2..... | 13 |
| Annotation of Data for detection..... | 13 |
| Model training for detection..... | 13 |
| Deployment of model with front end and 3D generation pipeline..... | 13 |
| Adding database backend to store detections along with IDs..... | 13 |
| Model Optimization to work in Realtime..... | 13 |
| Implementation Detail..... | 14 |
| 1. System Architecture:..... | 14 |
| 2. Video Stream Capture:..... | 14 |
| 3. 3D Model Generation:..... | 14 |
| 4. Object Detection:..... | 14 |
| 5. Anomaly Detection:..... | 14 |
| 6. User Interface:..... | 14 |
| 7. Data Storage and Management:..... | 14 |

| | |
|---|----|
| 8. Integration with Smart Spectacles: | 15 |
| 9. Security Measures:..... | 15 |
| 10. Performance Optimization: | 15 |
| 11. Testing and Validation: | 15 |
| 12. Documentation and User Training: | 15 |
| 13. Deployment:..... | 15 |
| 14. Continuous Improvement: | 15 |
| References..... | 16 |

Introduction

Rapidly and relentlessly, Artificial Intelligence is revolutionizing the entire world at a fast pace. Let it be Education, healthcare, or any other field, AI has left no stone unturned. AI is feared to replace the human workforce but till now it has proved to be a tool that assists humans to enhance their performance and boost their capabilities.

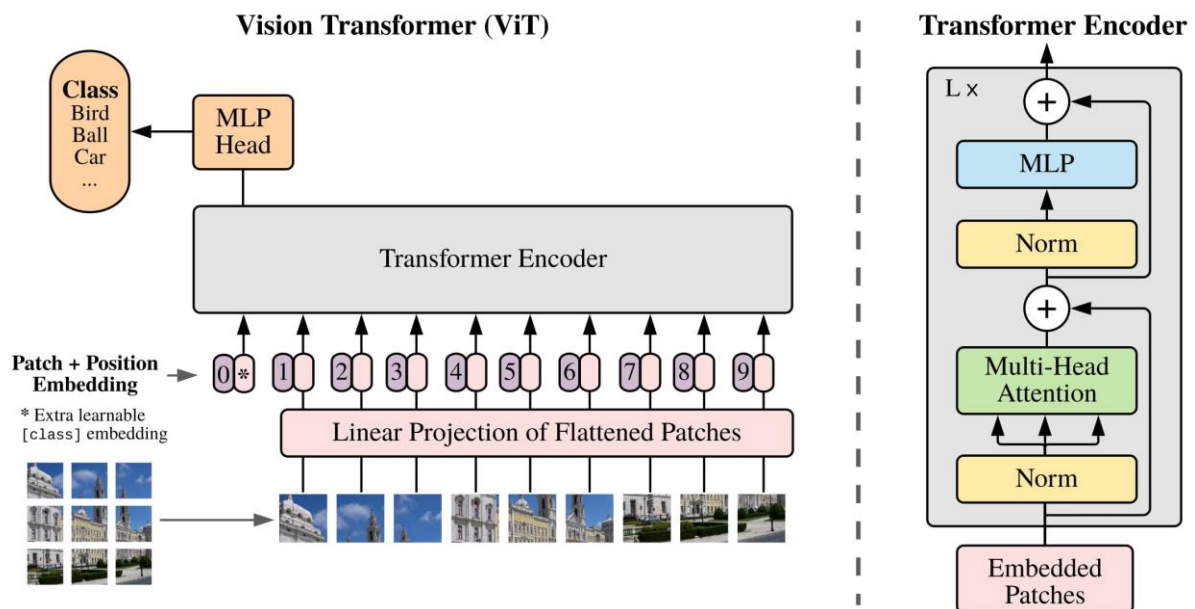
Like all other fields/sectors, AI is contributing a lot in the automotive industry. Let it be the manufacturing side or consumer side applications, AI has played a vital role. Our project,

“Inspect AI”, in collaboration with VisionRD will assist the manufacturing sector of the industry to inspect EV Batteries, using a physical visual apparatus/device.

This project mainly focuses on assisting a worker in inspecting the EV battery using a physical device such as glasses. The worker would give a live video feed from all 6 angles of the battery as the input, and the defects would be highlighted. With this product, the time consumption would be reduced and the inspection would be done in a more efficient way.

Literature Review and Architecture

One of the architectures we studied for this project is Vision Transformer Architecture. This uses self-attention mechanism to process images.



Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).

Project Vision

To revolutionize the Inspection method of the automotive industry for EV Batteries by developing such a system that would highlight the defects using a live video feed.

The main focus of the project is detecting defects in EV batteries according to industry standards. For this, a worker inspecting the batteries, will be using a device or apparatus like glasses. The device would send a live video feed which would be used to detect any defect in the battery, in case of a detected defect, the defected part would be highlighted.

Software Requirement Specification

Features:

Real-time Video Processing:

The system should be able to process video streams from smart spectacles in real time.

3D Model Generation:

Convert the captured video frames into a 3D model of the inspected object.

Provide a user-friendly interface to visualize and manipulate the 3D model.

Object Detection and Recognition:

Detect and recognize objects within the captured video.

Highlight and label recognized objects in both the video stream and the 3D model.

Anomaly Detection:

Identify and flag anomalies or defects on the inspected object based on predefined criteria.

Provide notifications and alerts when anomalies are detected.

Measurement and Dimension Analysis:

Allow users to measure dimensions of objects within the 3D model.

Provide accurate measurements for inspection and analysis.

Annotation and Reporting:

Enable users to annotate detected anomalies and add notes for further analysis.

Generate detailed inspection reports with annotated images and 3D models.

Integration with Data Systems:

Provide integration with data storage and management systems to save inspection data and reports.

Enable seamless data sharing and retrieval for future reference.

Augmented Reality (AR) Overlay:

Overlay relevant information, such as object specifications or inspection guidelines, onto the user's view through the smart spectacles.

User Authentication and Authorization:

Implement secure user authentication to ensure only authorized personnel can access the system.

Assign different permission levels for various user roles.

User-Friendly Interface:

Design an intuitive and user-friendly interface for controlling the smart spectacles and interacting with inspection results.

Functional Requirements:

Video Stream Capture:

Capture high-quality video streams from the smart spectacles' cameras.

Ensure synchronization between the video streams and the 3D model generation.

3D Model Reconstruction:

Process captured video frames to reconstruct an accurate 3D model of the inspected object.

Utilize computer vision and depth-sensing techniques for precise reconstruction.

Object Detection Algorithms:

Implement advanced object detection algorithms to identify and locate objects within the video stream.

Utilize deep learning models for object recognition.

Anomaly Detection Algorithms:

Develop algorithms to detect anomalies based on predefined patterns or criteria.

Ensure high accuracy and low false-positive rates.

Measurement and Dimension Calculation:

Implement algorithms to calculate accurate measurements and dimensions from the 3D model.

AR Overlay Generation:

Generate AR overlays with relevant information and annotations to enhance the user's view.

Data Storage and Management:

Provide a reliable database to store inspection data, 3D models, annotations, and reports.

Ensure data security and integrity.

Notification and Alert System:

Implement a notification system to alert users in real time when anomalies are detected.

User Authentication and Roles:

Develop a secure authentication mechanism with role-based access control.

Administer user roles such as inspectors, supervisors, and administrators.

Integration with External Systems:

Provide APIs or integration points for connecting the inspection AI system with external data and reporting systems.

Quality Attributes:

Accuracy:

The system should exhibit a high level of accuracy in object detection, anomaly recognition, and measurement calculations to ensure reliable inspection results.

Performance:

The system should process video streams and generate 3D models in real time with minimal latency to support efficient and effective inspections.

Scalability:

The system should be designed to handle an increasing number of concurrent users and video streams without compromising performance.

Reliability:

The system should be highly reliable, minimizing downtime and ensuring consistent availability for inspection tasks.

Security:

Data transmission, storage, and access should be secured using encryption and proper authentication mechanisms to protect sensitive inspection data.

Usability:

The user interface should be intuitive and user-friendly, allowing inspectors to easily control the smart spectacles and interact with inspection results.

Interoperability:

The system should be designed to integrate with various data storage systems, APIs, and external reporting tools commonly used in the inspection industry.

Maintainability:

The system's architecture and codebase should be well-organized and documented to facilitate future maintenance and updates.

Adaptability:

The system should be adaptable to different types of objects and inspection environments, accommodating a variety of inspection scenarios.

Privacy:

The system should adhere to privacy regulations and guidelines, ensuring that captured video and inspection data are handled with appropriate consent and safeguards.

Non-Functional Requirements:

Response Time:

The system should respond to user interactions and anomaly detections within a predefined acceptable time frame.

Processing Speed:

Video processing and 3D model generation should be completed within a reasonable time to provide real-time insights.

Data Storage Capacity:

The system should be able to handle a large volume of inspection data, including 3D models, video streams, and reports.

Security Measures:

Ensure data encryption during transmission and storage, implement secure authentication and authorization mechanisms, and conduct regular security audits.

Compatibility:

The system should be compatible with a variety of smart spectacle models, operating systems, and hardware configurations.

Scalability Limits:

Define the maximum number of concurrent users, video streams, and objects that the system can handle while maintaining performance.

Accuracy Metrics:

Define accuracy metrics for object detection, anomaly recognition, and measurement calculations, and ensure that the system meets or exceeds these metrics.

Backup and Recovery:

Implement regular data backups and establish a disaster recovery plan to minimize data loss in case of system failures.

User Training and Support:

Provide comprehensive user training materials and support resources to help inspectors effectively use the system.

Regulatory Compliance:

Ensure that the system complies with industry-specific regulations and standards related to inspections and data handling.

Iteration Plan

Iteration 1:

1. Data Collection
2. Model Creation for 2D image to Depth Map
3. Model creation for 2D image + Depth map to 3D model
4. Creation of Front End
5. Implementation of Login System

Iteration 2:

1. Annotation of Data for detection
2. Model training for detection
3. Deployment of model with front end and 3D generation pipeline
4. Adding database backend to store detections along with IDs
5. Model Optimization to work in Realtime

[illegible]

Iteration 1

Data Collection

1. Collection of 2D images (around six images per object from multiple angles) along with their 3D models for training a 2D to 3D model via scrapping from the internet and using prebuilt conversion models to convert task specific images and objects.
2. Generation of depth maps for collected images

Model Creation for 2D image to Depth Map

Creating an architecture based on GANs to convert given images into a depth map to specify the third dimension in an image and later use the said depth map for geometric measurements to help creation of 3D model

Model creation for 2D image + Depth map to 3D model

Using the previous model and preprocessing create a pipeline to feed all six images (plus depth maps) to train another GANs based model to convert 2D images into a 3D model (.obj file)

Creation of Front End

Using UI UX and web dev to create a user-friendly design for a web app (based on Flask or Fast API) that integrates previously created models. That provides livestream as well as detections on the livestream and creates a log file for found anomalies

Implementation of Login System

Creation of a login system integrated into the front end along with a database (firebase or AWS/ google big cloud) to stop unwanted access to the application and provide hierarchy-based feature control

| | Week | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Module | Data Collection | | | | | | | | |
| | Model Creation for 2D image to Depth Map | | | | | | | | |
| | Model creation for 2D image + Depth map to 3D model | | | | | | | | |
| | Creation of Front End | | | | | | | | |
| | Implementation of Login System | | | | | | | | |

Iteration 2

Annotation of Data for detection

Annotate the data received by creating bounding boxes around all anomalies

Model training for detection

Finetuning a prebuilt model (yolov7) to detect the anomalies

Deployment of model with front end and 3D generation pipeline

Integrating the complete pipeline into the frontend using web frameworks such as fast api and Django.

Adding database backend to store detections along with IDs

Creating a database (mysql) to store the model, predictions and ids

Model Optimization to work in Realtime

Optimizing the models, and configuring the pipeline to work with minimum resources

| | Week | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|
| Module | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| | Annotation of Data for detection | | | | | | | | |
| | Model training for detection | | | | | | | | |
| | Deployment of model with front end and 3D generation pipeline | | | | | | | | |
| | Adding database backend to store detections along with IDs | | | | | | | | |
| | Model Optimization to work in Realtime | | | | | | | | |

Implementation Detail

1. System Architecture:

Design a modular architecture that separates components for video capture, video processing, object detection, 3D model generation, user interface, and data management. Consider a microservices or service-oriented architecture to allow scalability and maintainability.

2. Video Stream Capture:

Develop drivers or APIs to interface with the smart spectacle's cameras and sensors for real-time video stream capture.

Utilize libraries like OpenCV to access and manage video feeds.

3. 3D Model Generation:

Implement computer vision algorithms to process the captured video frames and reconstruct a 3D model of the inspected object.

Utilize depth sensing techniques (e.g., stereo vision, structured light) to enhance the accuracy of the 3D reconstruction.

4. Object Detection:

Choose and implement state-of-the-art object detection models (e.g., YOLO, Faster R-CNN) to identify and localize objects in the video streams.

Fine-tune or train the models on relevant datasets for the specific objects and anomalies you intend to detect.

5. Anomaly Detection:

Develop algorithms to analyze the 3D model and detected objects to identify anomalies or defects based on predefined criteria.

Define threshold values and anomaly patterns for accurate detection.

6. User Interface:

Design an intuitive user interface that displays the live video feed, 3D model, detected objects, and annotations.

Implement AR overlays to display additional information and annotations in real time.

7. Data Storage and Management:

Choose an appropriate database system to store inspection data, 3D models, annotations, and reports.

Implement APIs for data CRUD operations and integrate data storage with the rest of the system.

8. Integration with Smart Spectacles:

Develop software that communicates with the smart spectacles, allowing users to control camera settings, start/stop video streams, and trigger inspections.
Utilize hardware-specific SDKs or APIs provided by the smart spectacle manufacturer.

9. Security Measures:

Implement encryption mechanisms for data transmission and storage to ensure the security of sensitive inspection data.
Integrate user authentication and authorization to control access to the system's features.

10. Performance Optimization:

- Profile and optimize the system to achieve real-time processing and low-latency response times.
- Utilize hardware acceleration (GPU) for computationally intensive tasks like object detection and 3D reconstruction.

11. Testing and Validation:

- Conduct thorough testing of each component, including unit tests, integration tests, and end-to-end tests.
- Collaborate with domain experts to validate the accuracy and effectiveness of the detection and measurement algorithms.

12. Documentation and User Training:

- Create comprehensive documentation for installation, configuration, and usage of the system.
- Develop user training materials to ensure inspectors can effectively use the smart spectacles and the AI system.

13. Deployment:

- Deploy the system on appropriate hardware and cloud infrastructure, considering factors like scalability, availability, and data privacy.

14. Continuous Improvement:

- Implement mechanisms for collecting user feedback and monitoring system performance.
- Continuously update and improve the system based on user needs and technological advancements.

References

