Department of Artificial Intelligence

DAWOOD UNIVERSITY OF ENGINEERING & TECHNOLOGY

Gulburg Campus

MACHINE LEARNING

PROJECT REPORT

5TH SEMESTER

Submitted by:

Hadia (23F-BSAI-01)
Muhammad Waqas Ali (23F-BSAI-59)
Muhammad Umar Khan (23F-BSAI-71)

Section: A – 1

Year: 3RD

Semester: 5TH

Evaluated to:

Engr. Hamza Farooqui

Department of Artificial Intelligence

A&VR OBJECT DETECTION MODEL

ABSTRACT

This project presents a real-time object detection and classification system integrated with an Augmented Reality (AR) overlay. The system is designed to detect multiple objects from a live camera stream, classify them into predefined categories, and display labels with confidence scores directly on the video feed. A pre-trained YOLO model based on the COCO dataset was fine-tuned using the Ultralytics framework and optimized for real-time inference. The trained model was exported to ONNX format and integrated into a Unity 6.0 application, enabling cross-platform deployment on both desktop (webcam-based) and Android mobile devices. The system achieves real-time performance with acceptable accuracy and frame rate, demonstrating an end-to-end ML pipeline including training, optimization, deployment, and AR visualization.

INTRODUCTION

Background & Motivation

Real-time object detection has become a core component of modern intelligent systems, including surveillance, autonomous assistance, AR-based education, and human–computer interaction. When combined with Augmented Reality, object detection systems can provide contextual and interactive information directly within the user's visual environment. This project aims to bridge machine learning and AR by creating a system that performs object detection in real time and overlays meaningful visual information on live camera input.

Project Objective

The main objective of this Complex Computing Activity (CCA) project is to design and implement an end-to-end machine learning system that integrates multiple lab concepts. The project focuses on real-time object detection, model optimization, cross-platform deployment, and AR-based visualization to demonstrate applied mastery of ML system design.

Scope of the Project

Included:

- Real-time object detection using a deep learning model
- AR overlay with labels and confidence scores
- Deployment on desktop (webcam) and Android mobile
- Model optimization and performance evaluation

Not Included:

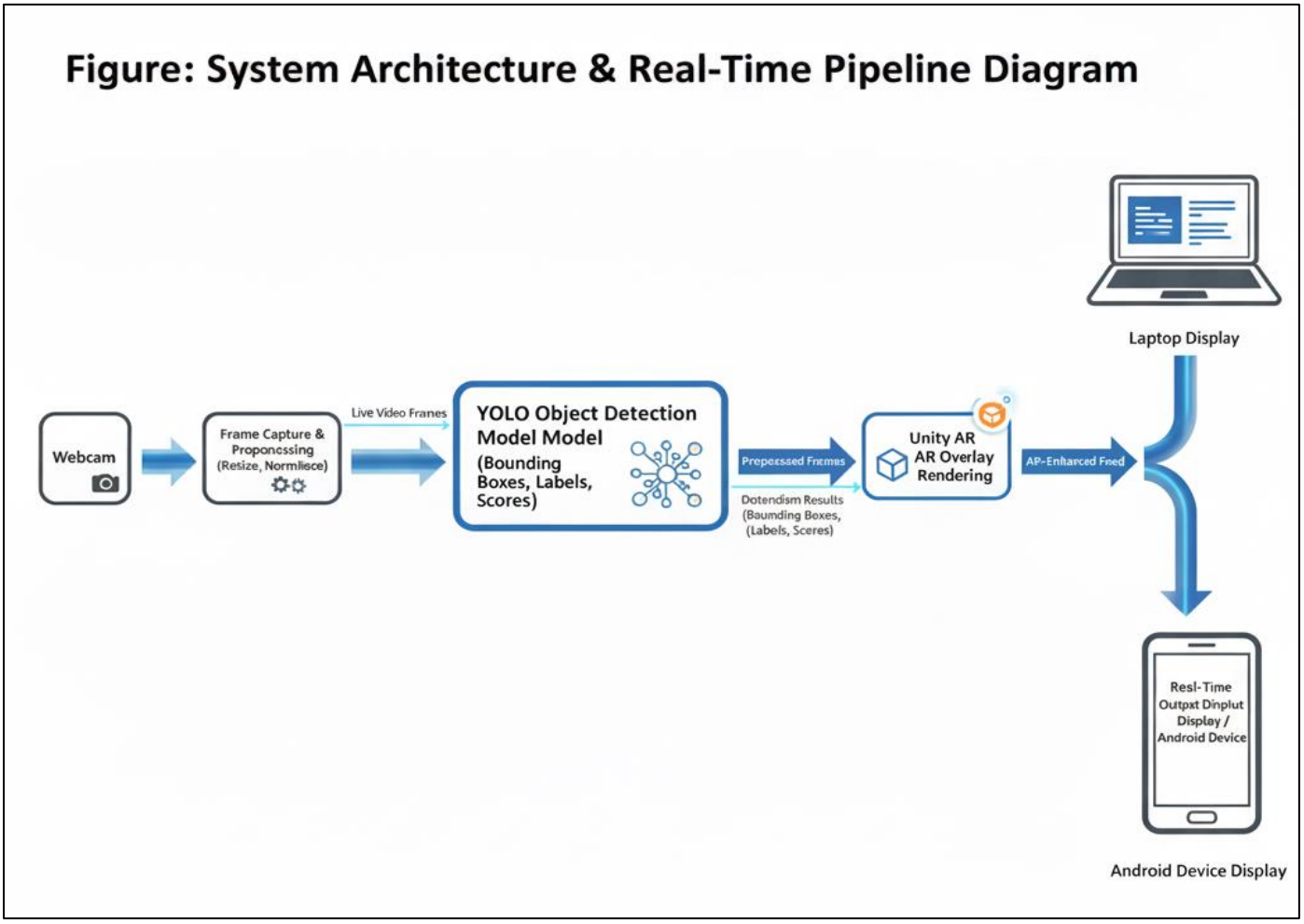
- Custom dataset collection from scratch
- Advanced 3D AR object placement
- Cloud-based inference or backend integration

SYSTEM ARCHITECTURE & PIPELINE

Overall Architecture

The system follows a modular pipeline where live camera frames are captured, processed by a trained ML model, and visualized using an AR overlay. The architecture ensures low latency and real-time responsiveness.

Pipeline Diagram



TECHNOLOGY STACK

Programming Language: Python, C#

ML Framework: Ultralytics YOLO (PyTorch-based)

Computer Vision: OpenCV

Model Format: ONNX

AR & Deployment: Unity Hub 6.0

Target Platforms: Desktop (Webcam), Android (APK)

DATASET DESCRIPTION & PREPROCESSING

Dataset Selection

The COCO (Common Objects in Context) dataset was used for training. It contains a wide variety of real-world objects, making it suitable for general-purpose object detection tasks.

Number of Classes: 80

Dataset Type: Large-scale object detection dataset

Data Annotation

The dataset uses standard COCO annotation format, which includes bounding boxes, class labels, and metadata. Since a pre-annotated dataset was used, no manual labeling was required.

Preprocessing Techniques

1. Image resizing to model input size
2. Normalization of pixel values
3. Data augmentation techniques:
4. Random rotation
5. Horizontal flipping
6. Brightness and contrast variation

Integrated Lab Concepts

Lab Concept	Implementation in Project
Image Preprocessing	Resizing, normalization, augmentation
CNN / Transfer Learning	YOLO pre-trained model fine-tuning
Model Evaluation	Accuracy, confidence score analysis
Real-Time Pipeline	Live webcam and mobile camera stream
Optimization	ONNX export for faster inference
Visualization	AR overlay with labels and bounding boxes

MODEL DESIGN & TRAINING

Model Architecture

The project uses a YOLO-based convolutional neural network capable of performing real-time object detection. The model consists of convolutional layers for feature extraction followed by detection heads for bounding box regression and classification.

Training Approach

A pre-trained YOLO model was imported and fine-tuned using the Ultralytics library. Transfer learning was applied to leverage existing learned features from the COCO dataset.

Hyperparameters

Learning Rate: Default Ultralytics setting

Batch Size: Configured based on hardware

Epochs: Limited training epochs for demonstration

Optimizer: Adam / SGD (default)

EVALUATION & PERFORMANCE METRICS

Classification Metrics

- Detection accuracy (qualitative)
- Confidence score analysis
- Bounding box correctness

Real-Time Performance

- Frames Per Second (FPS) measured during live inference
- Low-latency detection suitable for real-time use

AR OVERLAY IMPLEMENTATION

AR Design

- Bounding boxes around detected objects
- Class labels displayed above objects
- Confidence scores shown in real time

Real-Time Integration

Unity accesses the webcam and mobile camera feed. The ONNX model is loaded within Unity, and inference results are synchronized with the video stream.

Optimization Techniques

- ONNX model export for faster inference
- Reduced input resolution for mobile devices
- Efficient frame handling in Unity

CHALLENGES & LIMITATIONS

Technical Challenges

- Managing real-time latency
- Mobile hardware constraints
- Unity–ML integration complexity

Limitations

- Performance affected by lighting conditions
- COCO dataset bias toward common objects
- Limited customization of AR features

SETUP INSTRUCTIONS

Environment Setup

```
conda create -n cca_ml python=3.9
pip install ultralytics opencv-python
```

RESULTS & DISCUSSION

The system successfully detects and classifies objects in real time on both desktops and mobile platforms. The AR overlay provides intuitive visualization, making the system suitable for educational and assistive applications.

CONCLUSION & FUTURE WORK

This project demonstrates a complete ML pipeline from model training to real-time AR deployment. It integrates multiple concepts from the CCA labs and showcases practical application of machine learning in real-world systems.

FUTURE WORK:

- Advanced AR features (3D overlays)
- Edge-device optimization
- Expanded dataset and custom classes

REFERENCES

- Ultralytics YOLO Documentation
- COCO Dataset
- Unity Official Documentation
- OpenCV Documentation