**COS40007 Artificial Intelligence for Engineering**

**Portfolio Assessment-6: Graffiti Detection Project Report**

**Name: Tahmidul Haque Chowdhury  
Student Number: 103541308   
Studio Class: Studio 1 (BA 405)**

**Introduction**

The goal of this task is to use deep learning techniques, especially the YOLOv5 model, to create an efficient system that recognizes graffiti in videos. In order to recognize graffiti artworks in a variety of locations and help with automated monitoring and analysis, the primary focus is on real-time object detection.

**Project Structure**

The project is organized into several key components:

1. **Dataset Preparation**:

For model training, a set of graffiti pictures and csv files were collected. A custom Python script was used to convert CSV files to YOLO format in order to make annotations. For each image, the script produced.txt files with normalized bounding box coordinates, which were then utilized for training.

1. **Model Training**:

The initial idea was to use the provided dataset to train a YOLOv5 model. However, this section was left out of the final submission because of limited resources and repeated crashes of the system that arose during the training process.

1. **Video Detection Code**:

A pre-trained YOLOv5 model was used to detect graffiti in the given video file. In order to identify and point out any graffiti seen in the video; the code was implemented.

**Realtime Graffiti Detection Code**

The following code snippet demonstrates the process of performing graffiti detection in a video using the YOLOv5 model:

import cv2

import torch

def detect\_graffiti\_in\_video(video\_path):

    # Load the pre-trained YOLOv5 small model

    model = torch.hub.load('ultralytics/yolov5', 'yolov5s')

    # Open the video file

    cap = cv2.VideoCapture(video\_path)

    if not cap.isOpened():

        print("Error: Could not open video.")

        return

    frame\_count = 0

    while cap.isOpened():

        ret, frame = cap.read()

        if not ret:

            break

        # Process every 5th frame to reduce computational load

        if frame\_count % 5 == 0:

            results = model(frame)  # Perform inference

            # Print the number of detections

            print(f"Detections: {results.xyxy[0].shape[0]}")

            # Iterate over detected items

            for det in results.xyxy[0]:

                x1, y1, x2, y2, conf, cls = det

                print(f"Detected class {int(cls)} with confidence {conf:.2f}")  # Print detected class and confidence

                cv2.rectangle(frame, (int(x1), int(y1)), (int(x2), int(y2)), (255, 0, 0), 2)

                cv2.putText(frame, f'Confidence: {conf:.2f}', (int(x1), int(y1)-10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (255, 0, 0), 2)

        frame\_count += 1

        cv2.imshow("Graffiti Detection", frame)  # Display the frame with detections

        # Exit the loop if 'q' is pressed

        if cv2.waitKey(1) & 0xFF == ord('q'):

            break

    cap.release()  # Release the video capture object

    cv2.destroyAllWindows()  # Close all OpenCV windows

if \_\_name\_\_ == "\_\_main\_\_":

    video\_path = r'D:\Uni\AI for Engineering\Week 6\sample video.mp4'  # Change to your video path

    detect\_graffiti\_in\_video(video\_path)

**Key Features of the Video detection Code:**

* **Model Loading**: The code uses pre-existing information for graffiti recognition by loading a pre-trained YOLOv5 small model (yolov5s).
* **Real-Time Video Processing**: The approach ensures an acceptable detection frequency while optimizing efficiency by processing every fifth frame.
* **Visualization**: Bounding boxes are used to emphasize detected graffiti, and each detection's confidence value is shown.

**Challenges Encountered**

* **Training Issues**: Due to local limitations on resources, attempt to develop the model on the provided dataset were disrupted, and as a result, the training part was left out of the submission.
* **Detection Performance**: Although the pre-trained algorithm does quite well, the level of detail and clarity of the graffiti in the video can affect how effective it is.

**Conclusion**

This study demonstrates how deep learning can be used to identify graffiti. The above code shows that YOLOv5 can recognize graffiti in video data, even if the training phase was not finished. In order to increase the system's efficiency, future research should address the training problems and look at further improvements.