Configuration Manual

MSc Research Project
Programme Name

Erkan UciForename Surname
Student ID: 23210494

School of Computing
National College of Ireland

Supervisor: Sheresh Zahoor

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Erkan Uci
Student ID:	23210494
Programme:	Master's Degree in Artificial Intelligence **Year:** 2023-2024
Module:	MSc Research Practicum
Lecturer:	Sheresh Zahoor
Submission Due Date:	12/08/2024
Project Title:	Deep Learning-Based Multi-Object Group Detection and Tracking in Video Streams
Word Count:	5608 **Page Count:** 27

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:

Date: 12/08/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐ ✓
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐ ✓
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐ ✓

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Erkan Uci
Student ID: 23210494

This Configuration Guide provides the software tools and settings needed to replicate the experimental setup for a multi-object tracking system. It describes the Python libraries and settings needed for data processing, model training and implementation of tracking algorithms. In particular, it includes the modules needed for training the Faster R-CNN model and for multi-object tracking. This document is designed to enable other researchers to accurately reproduce the results of this work. It does not cover setup procedures for standard software tools.

1. Introduction

This guide provides the Python libraries and settings needed for data processing and model training. In particular, it focuses on training the Faster R-CNN model and implementing object tracking algorithms.

2. Software Tools and Libraries

Getting Started
To get started, you need to have Python and a few basic libraries installed on your system. If you don't have these tools yet, here's what you need:

- **Python: Make sure you have Python 3.8 or later installed. It's the programming language we'll be using.**
- **Environment Manager: Using Conda or Miniconda is recommended for managing your Python packages.**

Required Python Libraries
To make sure everything works properly, you need to install a few Python libraries. Open your command line interface and run the following commands:

These libraries serve the following purposes:

- **torch** and **torchvision:** For implementing the Faster R-CNN model and transformations.
- **opencv-python:** For processing and displaying video frames.
- **numpy:** For handling numerical operations.
- **pandas:** For data manipulation, if needed.

Choosing an IDE
You can use any **Python-compatible IDE** or code editor to write and run your code. Popular options include **Jupyter Notebook, PyCharm or Visual Studio Code.**

3. Model Training and Setup

Install and Configure the Model
Your system consists of two main components: ObjectDetector and VideoProcessor classes. Here is a brief summary of what each one does:

ObjectDetector Class:

- Initialization: This class loads the Faster R-CNN model and prepares it for object detection.
- Model Loading: The model is configured with a custom number of classes and loaded from a file.
- Prediction: Processes images to detect objects and filters out low-reliability predictions.

VideoProcessor Class:

Video Processing: Reads video frames, applies object detection and tracks objects across frames.
Visualization: Draws bounding boxes and labels on detected objects and displays the video in a window.

Configuration Details

In the main() function, you need to provide paths to the trained model and the video file you want to process:

```
def main():
    model_path = 'C:/Workspace/MOT17_Object_Tracking/models/faster_rcnn.pth'  # Path to your trained model file
    video_path = 'MOT17_Object_Tracking/data/MOT17-13-SDP/test.mp4'  # Path to your video file
    #video_path = 'MOT17_Object_Tracking/data/MOT17-13-SDP/traininMOT17-14-FRCNN.mp4'
```

References

Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016). Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)* (pp. 3464-3468). IEEE. https://doi.org/10.1109/ICIP.2016.7532876

Sun, S., Akhtar, N., Song, H., Mian, A., & Shah, M. (2019). Deep affinity network for multiple object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(1), 104-119. https://doi.org/10.1109/TPAMI.2019.2917935

Milan, A., Leal-Taixé, L., Reid, I., Roth, S., & Schindler, K. (2016). MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*. Retrieved from https://arxiv.org/abs/1603.00831

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

Kim, C., Li, F., & Rehg, J. M. (2018). Multi-object tracking with neural gating using bilinear LSTM. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 200-215). https://doi.org/10.1007/978-3-030-01264-9_13

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems* (pp. 91-99). Retrieved from https://arxiv.org/abs/1506.01497

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*. Retrieved from https://arxiv.org/abs/1804.02767

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In *European Conference on Computer Vision* (pp. 21-37). https://doi.org/10.1007/978-3-319-46448-0_2

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*(1-2), 83-97. https://doi.org/10.1002/nav.3800020109

Zhang, L., Li, Y., & Nevatia, R. (2008). Global data association for multi-object tracking using network flows. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-8). https://doi.org/10.1109/CVPR.2008.4587516

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3354-3361). https://doi.org/10.1109/CVPR.2012.6248074

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *European Conference on Computer Vision* (pp. 740-755). https://doi.org/10.1007/978-3-319-10602-1_48