Project Odysseus

Patrick O'Hara[1*], James Walsh*, Chance Haycock, Edward Thorpe-Woods, Oluwafunmilola Kesa, Andrew Wang, Mihai Ilas, James Brandreth, Oscar Giles, Neil Dhir, Theodoros Damoulas

Data-centric Engineering Programme, The Alan Turing Institute
Departments of Computer Science & Statistics, University of Warwick

*Equal contributions. [1]patrick.h.o-hara@warwick.ac.uk

SUPPORTED BY MAYOR OF LONDON

Busyness in London

Introduction

In response to the current COVID-19 pandemic, policy makers at the Greater London Authority (GLA) and Transport for London (TfL) require prompt and accurate data sources to understand how the *busyness* of the city is changing. This poster presents *Project Odysseus* at the Alan Turing Institute which integrates multiple *large-scale, heterogeneous datasets* and *develops machine learning models* to *infer mobility*, transportation and footfall activity throughout the city of London.

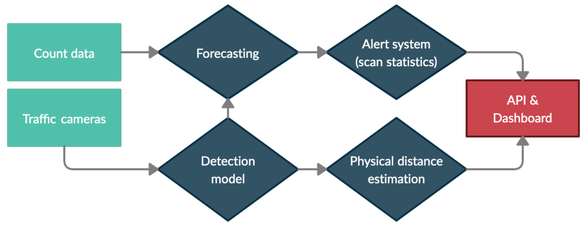


Figure: Components of Project Odysseus and how they fit together

Scalable Cloud Infrastructure

Our Azure cloud infrastructure and databases are the foundation upon which our models and analysis operate in *near real-time*. Our deployed codebase and self-described infrastructure are built with *reproducible technologies* that may automatically construct synonymous projects in other cities.

Detection Model & Physical Distance Estimation

By utilising existing city infrastructure including over 900 traffic camera feeds, we can *estimate social distancing adherence* by the general public; *count the footfall* of pedestrians walking London's streets; and *classify passing vehicles* such as buses, cars, motobikes and trucks. We develop a variant of the YOLOv4 object detection model with *transfer learning* on a variety of urban activity datasets, including additional new labels created on traffic camera samples with multiple DGX-2s. Inference upon sampled live feeds, 30Gb/day, is completed in near real-time within a Kubernetes compute cluster in the cloud.

Alert System

Building on top of our models for social distancing estimation, footfall density and vehicle classification, we describe an *early warning system* [1] that enables the monitoring of human activity levels and ultimately, the *rapid detection of spatio-temporal regions* that differ significantly from expected levels. Our approach introduces an *Expectation-Based Network Scan Statistic* which extends prior work and deploys Gaussian processes for time-series forecasting in order to quantify uncertainty.

API & Dashboard

Our outputs are queried through a REST API and complemented by a live dashboard, allowing fast and efficient dissemination of information to the policy makers who need it.

Detection Model

Camera calibration: no *a priori* camera parameters - must be assumed or estimated. Find mapping from the *image plane* $(u, v, 0)$ to the *world geometry* (X, Y, Z).

Data: 10 second videos from 911 live TfL "JamCam" cameras every 4 minutes. One day $\approx$ 220,000 files $\approx$ 20-30GB.

Classify objects by person, car, bus, motorbike, bicycle, and truck.

Compute: Tesla V100-SXM3-32GB GPU.

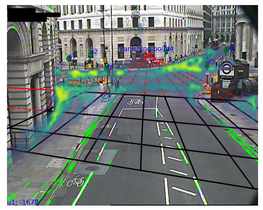
Train improved YOLOv4 detection model on 6 labels from the Coco 2017 [2] and MIO-TCD localization datasets.

Training	Validation	mAP@0.50 ↑	Precision ↑	Recall ↑	F1-score ↑
Coco	Coco	**67.55**	**0.73**	**0.70**	**0.71**
	MIO-TCD	20.39	0.38	0.49	0.43
	JamCam	41.64	0.62	0.59	0.60
MIO-TCD	Coco	14.24	0.35	0.30	0.30
	MIO-TCD	**85.80**	**0.83**	**0.90**	**0.86**
	JamCam	35.12	0.75	0.45	0.57
Joint	Coco	64.56	0.71	0.69	0.70
	MIO-TCD	**80.32**	**0.79**	**0.88**	**0.83**
	JamCam	46.53	0.76	0.57	0.65

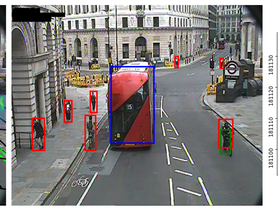
Table: Comparing models fine-tuned on the Coco 2017 dataset, MIO-TCD dataset, and joint training set using YOLOv4 architecture.

Evaluation: compute the mean Average Precision (mAP) at IOU threshold of 0.5 over the Coco 2017, MIO-TCD, joint (Coco 2017 + MIO-TCD), and *JamCam* datasets (see table).

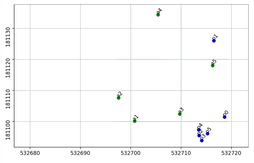
Physical Distance Estimation



(a) Estimated ground plane from camera calibration, overlaid pedestrian detection density



(b) High accuracy detections. Pedestrians, buses, bicycles in red, blue, green respectively



(c) Estimated locations of urban furniture (green) and pedestrians (blue) from (a) & (b).

Points of reference (PoR): defined by geo-tagging static urban furniture such as traffic lights. PoR exist in flat 2D projected coordinate reference system (British National Grid 27700) which is a *world plane* $(X, Y, 0)$ of the world geometry (X, Y, Z).
Pedestrian detections in Figure (b) are mapped to the *world plane* in Figure (c).
Measure distance between pedestrian detections in the world plane using coordinate reference system.

Scan statistics

Input: A time-series sampled at every hour up to the present time T at a set of N_s fixed spatial locations $\mathcal{X} = \{x_i \in \mathbb{R}^2\}_{i=1}^{N_s}$. The fixed spatial locations lie on a network $\mathcal{G} = (V, E)$.
Goal: Identify spatio-temporal clusters. Compare expected counts $b_i^t \in \mathbb{Z}_+$ with true counts $c_i^t \in \mathbb{Z}_+$.
Method: Scan over the most recent W hours of data. Assign scores to each search region. Report on significant increases/decreases in activity.

- *Generate baselines* b_i^t by forecasting expected count data which represents typical behaviour (null hypothesis). Compare b_i^t against actual count data to locate emerging regions of busyness. We compare two forecasting methods: Holt-Winters (HW) and Gaussian Processes (GP).
- *Define search grid* on the search domain (the road-network $\mathcal{G}$). Search regions are defined by a lower and upper bound on path length in the network.
- *Define a metric* which identifies space-time regions in which activity is quieter than expected.

Evaluation: Test data is semi-synthetic, hourly vehicle counts. Measure spatial precision and recall.

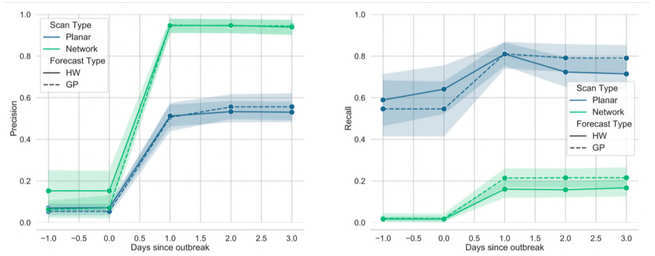


Figure: Average time to detect a simulated surge. Spatial performance. The number of days since an outbreak cannot be measured, but this simulation helps us understand how promptly each scan can detect significant increases in activity.

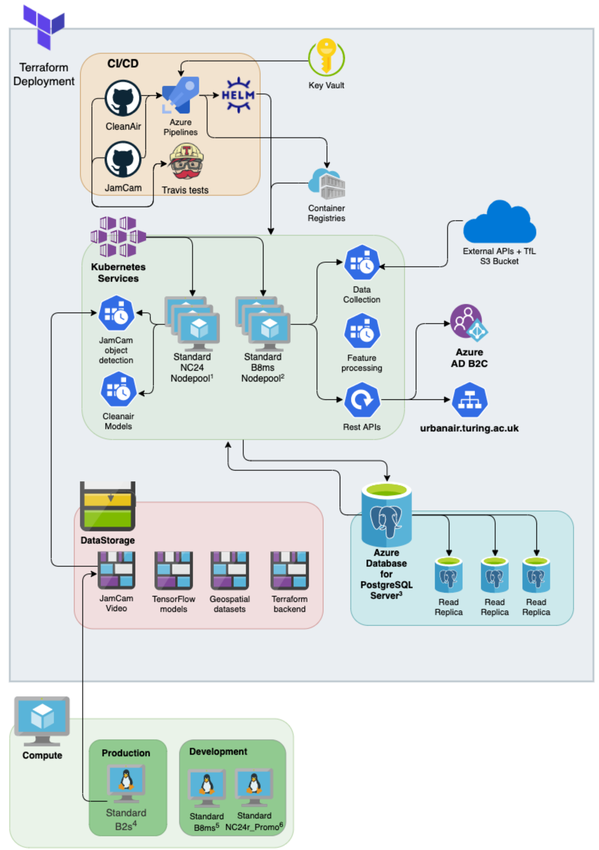
API & Dashboard

Our API disseminates the outputs of our models, scan statistics and physical distance estimation to our partners at GLA and TfL.

The *live dashboard* (see right) displays outputs from the detection model and physical distance estimation by querying the API:

- Each camera is represented by a point. The colour intensity indicates number of counts or average physical distance. These may be subset by detection class and date. For comparison purposes the colour scale may be locked to compare activity levels.
- Activity levels described as histogram profiles for immediate comparison with historical expectations
- A live view from the camera is shown alongside the calibration results and optional ground plane estimate
- Aggregated hourly activity is finally displayed for all classes for comparison between transportation types

Cloud architecture



Deployment provisioning is controlled declaratively by Terraform. Our cloud infrastructure is deployed on Microsoft's Azure platform. Most tools are open source or have equivalents on other platforms:

Storage: Blob containers with independent collection servers for continuous ingestion.

Compute clusters managed by Kubernetes:

- *GPU accelerated* video processing pool. Horizontally scales 1-4 nodes.
- *Stability focused* burstable CPU pool executing scheduled tasks and hosting API access points.

Database stores results. Duplicated into read replicas for redundancy and external API access.

Staging servers: develop under same environment as the deployed images operating on Kubernetes.

Acknowledgements

Funded by Lloyd's Register Foundation programme on Data Centric Engineering and Warwick Impact Fund. Further supported by the Greater London Authority, Transport for London, Microsoft, Department of Engineering at University of Cambridge and the Science and Technology Facilities Council.

References

C. Haycock, E. Thorpe-Woods, J. Walsh, P. O'Hara, O. Giles, N. Dhir, and T. Damoulas.
An expectation-based network scan statistic for a covid-19 early warning system.
In *Workshop on Machine Learning in Public Health, NeurIPS'20.*, 2020.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick.
Microsoft COCO: Common Objects in Context.
In *Computer Vision – ECCV*, Lecture Notes in Computer Science, pages 740–755. Springer, 2014.