https://doi.org/10.1093/comjnl/bxac160

Near Real-Time Social Distance Estimation In London

James Walsh[1,*], Oluwafunmilola Kesa[2], Andrew Wang[3], Mihai Ilas[3], Patrick O'Hara[2], Oscar Giles[1], Neil Dhir[1,2], Mark Girolami[1,4] and Theodoros Damoulas[1,2,5]

[1]*The Alan Turing Institute; Departments of*
[2]*Computer Science and*
[3]*Christ's College, University of Cambridge*
[4]*Department of Engineering, University of Cambridge*
[5]*Statistics, University of Warwick*
[*]*Corresponding author: jw2250@cam.ac.uk*

To mitigate the current COVID-19 pandemic, policy makers at the Greater London Authority, the regional governance body of London, UK, are reliant upon prompt, accurate and actionable estimations of lockdown and social distancing policy adherence. Transport for London, the local transportation department, reports they implemented over 700 interventions such as greater signage and expansion of pedestrian zoning at the height of the pandemic's first wave with our platform providing key data for those decisions. Large well-defined heterogeneous compositions of pedestrian footfall and physical proximity are difficult to acquire, yet necessary to monitor city-wide activity (busyness) and consequently discern actionable policy decisions. To meet this challenge, we leverage our existing large-scale data processing urban air quality machine learning infrastructure to process over 900 camera feeds in near real-time to generate new estimates of social distancing adherence, group detection and camera stability. In this work, we describe our development and deployment of a computer vision and machine learning pipeline. It provides near immediate sampling and contextualization of activity and physical distancing on the streets of London via live traffic camera feeds. We introduce a platform for inspecting, calibrating and improving upon existing methods, describe the active deployment on real-time feeds and provide analysis over an 18 month period.

Keywords: COVID-19; Computer Vision; Real-time; Computers and Society; Machine Learning; Policy Intervention; Change Point Detection; Social Distancing

Received 4 February 2022; Revised 7 July 2022; Editorial Decision 8 August 2022
Handling editor: Aristidis Likas

1. INTRODUCTION

Before 2020, the phrase 'social distancing' had hardly any visibility to the public eye [1] as vernacular more frequently found in epidemiology textbooks and historical reports [2]. However, during the COVID-19 pandemic, physical spacing between strangers became a means of trying to curb the spread of the virus.

As the global community is actively engaged in understanding more about the effects and transmission mechanisms of COVID-19, many governments have enacted temporary restrictions targeted at reducing the proximity of the public to one another, including measures such as limiting capacity within enclosed spaces, communicating new pedestrian traffic flow and, when necessary, enacting broader controls via 'lock-downs' [3]. The monitoring of public response to these measures has come out of necessity for policy makers to better understand their adoption, plan economic recovery and eventual suspension. When social restrictions were first implemented in the UK, there were limited measures of public activity in the context of likely vectors for viral transmission. A number of private companies trading in public movement data began providing aggregate information at the request of

local government, from sources such as workplace reporting, wearable sports activity trackers and point of sale transactions [4]. It became clear there was an immediate need for additional response metrics for pedestrian activity, unmet by the aforementioned sources.

This work seeks to estimate social distance in areas of high footfall in London, UK. The goal is to gauge adherence at high spatial and temporal granularity, and most importantly provide near real-time access to policy makers. We describe a *social-distancing estimation system* using Open Government Licensed [5] traffic cameras directed towards pedestrian crossings and pavements. We include the description of our pipeline, methodology, algorithms and new accuracy results as urban object detection benchmarks.

The basis for this approach was initially built for constructing greater predictive features to improve a live air quality model of London [6]. It is known that pollutants are generated at different rates depending on driving activity [7]. Traffic camera footage is a suitable candidate for proving features on typical vehicle movement, capable of assisting the modelling of fine airborne particulates contributing to air pollution. The cloud infrastructure developed for the air quality model serves as the foundation for our social distancing estimation system.

Due to the nature of large-scale CCTV capture, there were initial substantial privacy concerns. All footage employed throughout the process is anonymized via deliberate restrictive sampling and systematically undergoes continuous review by The Alan Turing Institute's Ethical Advisory Group [8].

2. METHOD

Cameras available to the public are heterogeneous in quality and fall victim to the sporadic physical nature of London's historical streets. This scenario presents numerous challenges from a geospatial statistical and technical perspective, see Fig. 2 for an example of a post-processed still frame. Our platform predominately relies upon 912 independent traffic camera feeds, over 500 of which typically overlook an intersection or crossing with an expected pedestrian footfall. In order to mitigate potential deanonymization, all input visual data are reduced in video resolution, significantly hampering facial identification. Additionally, to place our results in an appropriate broader context, our research goal is to measure variations in social distancing and feed quality over extended periods of time.

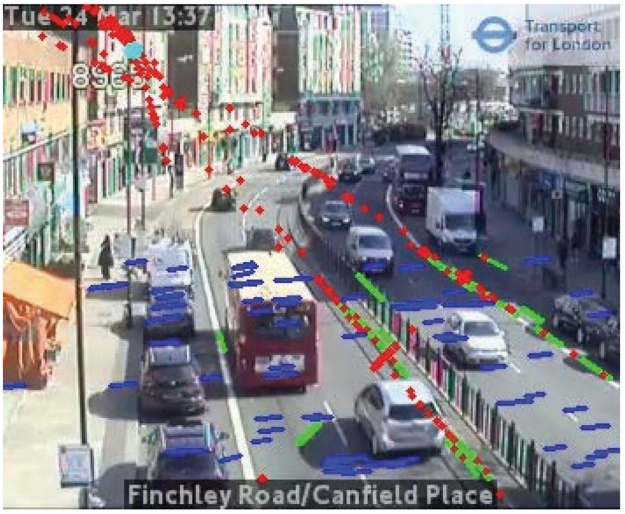
Each camera feed provides two data elements: a short video every few minutes and a restrictive set of static metadata regarding location and approximate cardinal direction. Hence, before attempting to estimate any pedestrian location, each camera requires an initial digital twin abstraction to define the *world-plane* of the visible scene stage, usually synonymous with visible road structure. A final *real-world* calibration is applied using human-labelled mappings from pixel locations within the image to physical coordinates of objects identified within the scene. These *anchors* are considered as 'ground-truth', examples include road markings, telephone booths and traffic lights. The objects selected are collectively referred to as 'urban furniture', and are of most benefit if visible from aerial or satellite photography for later calibration. As a form of *image registration*, this enables mapping from the two-dimensional video frame to an inferred unreferenced world-plane, finally to a real-world location. The process is difficult with highly variate CCTV scenes; our method learns one set of parameters for mapping a 2D scene to a 3D real-world coordinate projection and is described in Section 2.1.

Once complete, active data collection continuously ingests 10 s camera clips from the public domain. Upon successful retrieval of each video sample, they are batched for object detection. Our image processing pipeline is composed of a cluster of dynamically scaled compute resource via virtual machines operated by a container-orchestration system called Kubernetes [9]. As a batch of 500 clips are ingested at a time, we are careful to ensure our model and computational resources are sufficient to process each sample in less or equal time than they represent, i.e. 30 min of footage must complete within 30 min, or the system would perpetually slip behind real-time. We employ a tuned state-of-the-art object detection model called YOLOv4 [10, 11] for identification of pedestrians within video frames. The reasoning for this selection and the tuning process is included in Section 2.2, and active deployment as described in Section 2.5.

Results from the camera calibration and object detection stages are then stored within high-availability databases [12] near our data storage and image processing cluster. These databases permit immediate availability to public policy makers, specifically the greater London authority (GLA) and transport for London (TFL) via a reliable representational state transfer application programming interface (REST API). Additionally, high availability increases capacity for complimentary research tasks, such as simultaneously watching for spikes or irregularities via expectation-based network scan statistics [13].

A primary challenge borne out of long-term experimental processing is the unexpected consequences of relying upon cameras prone to real-world interference. Some examples identified during the developmental phase of this system are graffiti, wind progressively drifting the view direction, physical malfunction and trees sprouting leaves restricting previously clear views. In response to these detriments, we designed a camera stability change point detection process for identifying and alerting when scene dissimilarity meets a predetermined threshold, as described in Section 2.3.

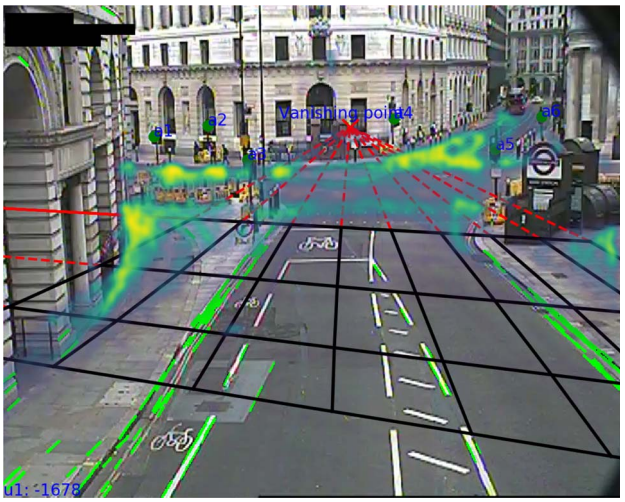
Finally, purely recognition, localization and relative distance are not enough for adequate social distancing metrics, as pedestrian activity typically includes grouping behaviour. As individuals seek to preserve physical distance with strangers while reducing the chance of disbanding their safe social group



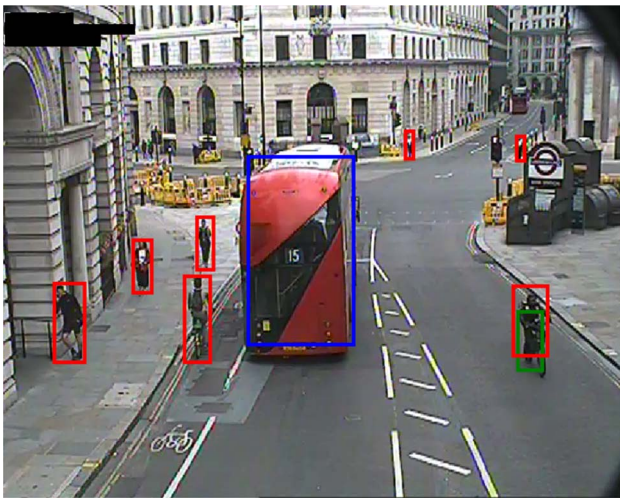
(a) Road curvature challenging condition example, leading to erroneous vanishing point.

(b) Harsh shadow parallel to vanishing lines, a beneficial scenario.

FIGURE 1. Lines detected by our feature extraction algorithm; two orthogonal sets of lines: those parallel to the foreground road (green, *road edges*) and those perpendicular (blue, *road perpendiculars*). The intersection of each set, the vanishing point (light blue) which lies on the horizon. Some challenging conditions are visible, including varying lighting and non-zero road curvature.



(a) Detected edges, generated ground plane, and overlaid pedestrian detection density in bright green, black, and viridis heat map respectively.



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

FIGURE 2. Example of our method applied to a traffic camera at Bank, London.

(or 'bubble') [14], an inclusion of group agency is considered. An algorithm for group detection operating at frame and scene granularity is presented in Section 2.4 for discussion as part of the final results.

2.1. Camera calibration

Obtaining a *world-plane mapping* of a camera scene is extensively described in the computer vision and photogrammetry literature [15, 16]. A large portion of literature requires manual calibration using known patterns to estimate a mapping from sensor data to a real world contextualization [17–19]. We aim to learn the geometric relationship between camera view and physical scene, frequently described through similarity, affine or projective transformations. A *vanishing point* of an image is the location of apparent three-dimensional convergence of parallel *vanishing lines* from a two-dimensional perspective (Fig. 2a). Estimation of these vanishing lines is a common

technique to recover some of these transformations. Our input scenes have multiple limitations: roads are usually curved or contain junctions of varying width; irregular road markings vary in quality; low video resolution of the feed; lighting conditions change frequently and are individually very short in duration.

Scene object context methods [20, 21] use the activity of multiple vehicles travelling parallel and regularly to estimate the vanishing point, which is a feasible solution for our problem if multiple samples were stitched together and vehicle movement manually corrected. Calibration methods [22–25] require clear, regular or known lines in the scene, which is not practical in the case of a large spread of physical geometry. A stratified transformation approach discussed in [26] relies upon maximum likelihood estimation (MLE), a popular method for parameter estimation of an assumed probability distribution, given some observations. This is applied over multiple extracted lines from high-quality images to build a real-world model, an issue for our low-resolution samples. Finally, [20, 23, 24, 27] extract visible road features using a derivative-based binarization operator. This is principally suitable for cameras overlooking straight and visually similar lanes, which is turn is only suitable for a portion of our input domain. Overall, we sought a more easily generalizable method considerate of our cluttered urban traffic scenes at low resolution that leverages our high sample quantity.

2.1.1. *Simplified pinhole camera model*

A mapping, $(u, v, 0) \mapsto (X, Y, Z)$, is sought from the image-plane to world geometry—for example, the transformation from pixels representing the bikes in Fig. 2 to a physical location. Without a priori truth of any parameters describing the camera properties, these properties should be estimated or assumed and categorized into two groups: *intrinsics* and *extrinsics*. Examples of intrinsics include focal length, principal point, skew and aspect ratio, whereas extrinsics include positioning and direction. After manual inspection of all cameras, we conclude the suitability of the *Simplified Pinhole Camera Model*, as fewer than 0.5% of cameras have ultra wide-angle (fisheye) lenses.

Our simplified pinhole camera model allows the transformation to be described by four parameters u_0, v_0, u_1, h, where (u_0, v_0), (u_1, v_0) are the vanishing points of two orthogonal planar directions subtending the horizon line, and h is the height of the camera above ground (Fig. 3). Parallel lines on the road and on cars, such as road edges, advanced stop lines and car and truck edges, are used to estimate this transformation (Fig. 2a). This model makes the following assumptions:

(a) Unit skew, i.e. regularly square pixel grid.
(b) Constant aspect ratio, i.e. no change in width-to-height ratio of pixels.
(c) Coincidence of principal point and image centre, i.e. no change in the center pixel from the center of the camera view.

These are commonplace and rarely estimated when lacking more detailed visual information [28], [21]. External assumptions as follows:

(d) Rectilinear lens, i.e. zero radial distortion; the image has already been pre-corrected such that perpendicular straight lines in reality are straight on the perpendicular pixel grid.
(e) Flat horizon $v_0 = v_1$, i.e. camera has zero-roll.
(f) Zero-inclined roads $Z = 0$, i.e. pedestrians do not move in a space large enough to calibrate deviation in elevation.

Where cameras fail these external assumptions, a pre-processing stage included additional information to correct radial distortion [21], inclined horizon (setting $v_1 \neq v_0$) and non-zero inclination Z [29].

2.1.2. *Edge detection*

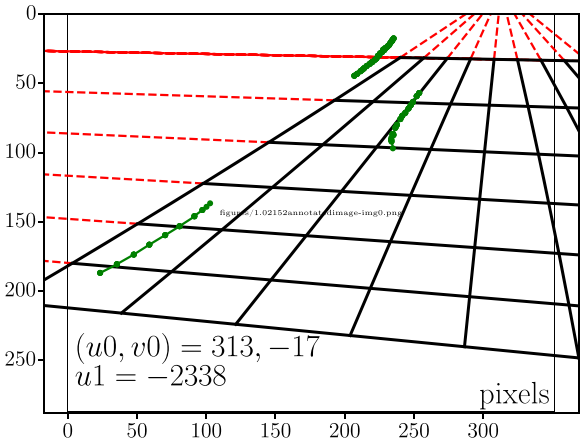
Our method for edge detection should be robust in noisy, low-resolution scenes with varying light conditions. While deep learning approaches for edge detection such as Visual Geometry Group (VGG) models [30] have seen significant advancements in the last decade [31, 32], they require hours of training on a large set of labelled edges. Note that our dataset does not have labelled edges. The value of rapid perspective mapping outweighs the time necessary to produce a suitable dataset of the scale of this task on our 912 scene samples. This problem extends to considering direct vanishing point estimation. The aforementioned deep learning approaches would require labelled data on the order of magnitude of hundreds of samples [32] for direct perspective estimation. We instead turn our attention to classical methods [33].

Developed in 1986, the Canny Edge Detector has seen wide adoption for its adept ability to find edges under the edge detection goals of low error rates and minimized false edges in noisy-scenarios, suitable for our low-resolution highly light-variate input scenes, see Figs 1 and 2a.

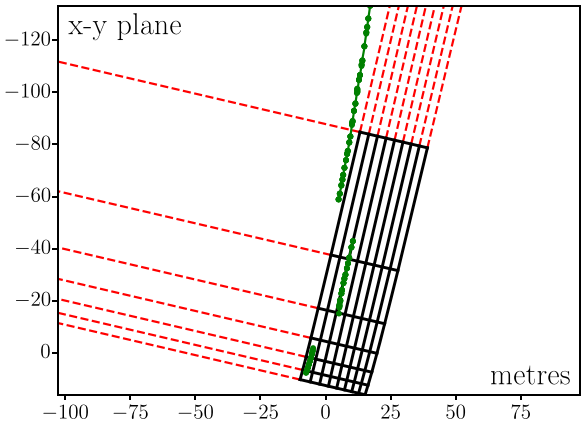
The method relies upon *Gaussian filters* to first smooth potential noise and then applies four filters to find *intensity gradients* with reference to gradient angle direction. Edge-thinning is subsequently applied via *magnitude thresholding*, but this is not enough to remove spurious variations in colour and noise. A second *double threshold* is applied using the surviving edge gradients, this time utilizing high and low empirically determined from the whole edge set. Finally, some final weak edges remain. A process called *hysteresis* is applied via blob analysis to determine survival based on proximity to neighbouring strong pixels.

2.1.3. *Parameter estimation*

In order to learn our simplified pinhole camera model detector parameters (u_0, v_0, u_1, h), scenes with light vehicle traffic are selected and the edge detector applied per frame to find sets of *road edges* and *road perpendiculars* as shown in Fig. 2. In



(b) Intermediate perspective mapping from image plane.



(d) Intermediate world plane from perspective mapping.

FIGURE 3. Demonstration of perspective mapping of camera calibration from image to world plane before estimated registration to British National Grid. Rays (black, solid) are drawn as grid lines and extended (red, dashed) to the estimated vanishing points (u_0, v_0) and (u_1, v_0). After mapping onto world coordinates, for example, vehicle trajectories (green, dotted) are also mapped by this transformation.

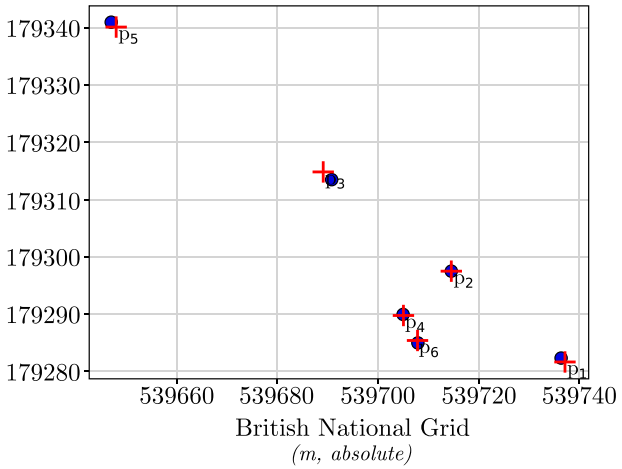


FIGURE 4. Estimated locations of urban furniture (green) and transformed ground-truth scene anchors (blue) on British National Grid.

order to learn a set of vanishing lines from these edges, the Hough transform matches collinear edge segments into linked lines which are then filtered by gradient [20] and dimensions. The vanishing point is then simply estimated as the highest frequency of the pairwise line intersections. This is chosen over more computationally expensive aforementioned MLE methods [34], where the vanishing point error is optimized using least squares [28], [27] or Levenberg–Marquardt [20], [26]. This procedure is repeated across different contrast factors to provide a robust line detector in challenging lighting conditions (as British weather traditionally exhibits). These u_0, u_1, v_0 values are averaged over all frames to extract a final estimate.

Finally, the camera height h must be manually estimated. One method is by transforming an object of known dimensions. For example, using frequently appearing London buses of fixed 4.95m height, the calculated height averages $h = 9.6m$ with 10% average deviation across seven randomly picked cameras. Other ways to obtain the scale h include using car length averages [29] or known lane spacing [25, 27]. Given few known consistent standardized urban furniture upright heights, the London Bus method is appropriate. With each parameter estimated is it possible to define a world-plane.

2.1.4. *Real-world reference*

The world-plane projection (Fig. 3) is as yet unreferenced to the real-world; a Euclidean Norm may be applied but not uniformly across all cameras. We employ points of reference via *geotagged* static urban furniture, such as traffic lights or road markings to map this intermediate world-plane to a real-world representation,

$$\begin{bmatrix} x' + e_x \\ y' + e_y \\ 0 \end{bmatrix} = \begin{bmatrix} k_1 \cos(\theta) & k_3 \sin(\theta) & t_x \\ -k_2 \sin(\theta) & k_4 \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \end{bmatrix}$$

We select an appropriate flat 2D projected coordinate reference system, British National Grid (OS 27700). We then employ a second transformation between these two 2D Cartesian frames of reference, represented above with scale and shear factors, k, angle of rotation, θ, translations, t, and error terms, e. The estimated real-world representation is the result of optimization of the sum of squares error between transformed image-coordinates of the urban furniture and the world-plane image registration.

2.2. Pedestrian detection

2.2.1. *Camera dataset*

Footage is sourced openly via Transport for London sourced from the Open Roads initiative, known as JamCams [35]. A day of collection constitutes approximately 220 000 individual files of a total of 20–30GB, deleted upon processing in accordance with our data retention policy. The nature of monitoring public spaces means we cannot a priori request consent. The reductive resolution of footage collected from this source inhibits any capacity to personally identify an individual. Thus only their humanoid likeness is utilized for detection.

2.2.2. *Object detector*

In order to detect entities quickly enough to assist policy makers, we evaluate object detection models such as SSD [36] and YOLO v3 [37] to balance speed against accuracy. These are typically determined by architecture, model depth, input sizes, classification cardinality and execution environment. You Only Look Once (YOLO) [38] is a one-stage anchor-based object detector which is both fast and accurate. YOLOv3 achieves an accuracy of 57.9 AP_{50} in 51ms [37]. Recently, a faster version named YOLOv4 [10] was released with a state-of-the-art accuracy than these alternative object detectors. Notably, YOLOv4 can be trained and used on conventional GPUs which allow for faster experimentation and fine-tuning on custom datasets. YOLOv4 improves performance and speed by 10% and 12%, respectively [10].

We employ both YOLOv3 and v4 in our experiments. Both were pre-trained on Coco [39] dataset, a large-scale repository of objects belonging to 80 class labels. Due to our objective, the classes of interest are limited to six labels: person, car, bus, motorbike, bicycle and truck. We fine-tuned the model on six labels using joint datasets from COCO, MIO-TCD [40] and a training set of custom manually labelled JamCam-specific set. A validation set was also partitioned from the manually labelled dataset for model evaluation. Results in the evaluation section documents the success of this fine-tuning to traffic camera footage.

2.3. Scene stability and camera drift

2.3.1. *Similarity indices*

Due to the extended duration of this project, it is necessary to include an evaluation of physical change in scene perspective or visible feed quality. Examples exhibited over time include intended adjustments made via motor-driven camera equipment, strong weather laterally progressively shifting direction, detachment from mounting hardware and when local vegetation sprouts to inhibit visibility of the original scene. To mitigate and detect these issues, we construct a variation metric using past frame information to detect variations in the captured scene. Direct application of pixel-for-pixel Mean Square Error (MSE) is not suitable under the change of lighting conditions, and is too sensitive to minute pixel differences. The Structural Similarity Index Measure (SSIM) has been shown [41] to aptly measure image distance via a kernel comparison approach.

$$\mathrm{SSIM}(x,y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)},$$

where $\mu_x, \mu_y, \sigma_x^2, \sigma_y^2, \sigma_{xy}$ are x and y window average, variance, and covariance, respectively; $c_{1,2}$ represent a weighted dynamic pixel range.

2.3.2. *Application to scene imagery*

Numerous *scene reference* periods were selected to measure SSIM for each camera: first known scene frame versus first hourly frame, 1-week historical offset to first hourly frame and mean of non-erroneous initial frames over 7 days from initial data acquisition versus first daily at noon. It became clear that the final measure is most appropriate both for noise reduction and computational efficiency. Over time, this generates a univariate time series measuring scene variation. Sustained linear drift is less likely to negatively effect our pedestrian location estimation; however, it can be detected once the threshold is met. More importantly, a single major movement must be detected for subsequent alerting during the live experiment operation. This option however does deviate from other automated tasks, requiring the 7-day period to be adjusted to the new scene reference upon human intervention.

2.3.3. *Change point detection*

Detection of abrupt shifts in frame similarity over time is a task suitable to the unsupervised learning problem of *change point detection*, the study of algorithms designed to find underlying change in time series [42]. An offline solution is still suitable, providing our large number of input feeds and necessity for appending daily measures. Upon evaluation, we determined Pruned Exact Linear Time (PELT) [43] under a standard RBF kernel could accurately partition camera scene changes. Under this measure, cameras of high variability are also excluded from later analysis.

2.4. Group detection

In order to better describe social distancing efforts, we implement a group detection process (algorithm 1) and define seven metrics to describe a scene over time (Table 1). Selecting groups from pedestrian detection locations are calculated by generating the Delaunay triangulation in the British National Grid (BNG) projection for pedestrians within individual frames per scene sample.

Algorithm 1: Group proximity frame tracking

Input: Scene of localised detections, S_L
Parameters: Confidence threshold, T_c
Distance threshold, T_d
Output: Total detected groups, G_n
Max groups per-frame, $G_{\max(n)}$
Min distance between groups, $G_{\min(d)}$
Mean distance between groups, $G_{\overline{d}}$
Mean internal group size, $I_{\overline{n}}$
Mean internal group distance, $I_{\overline{d}}$

$\widetilde{S_L} \leftarrow S_{L,\text{conf}} \geq T_c$; ▷ threshold confidence
$I_l, I_d \leftarrow$ empty;
foreach $f_L \in \widetilde{S_L}$; ▷ locations per frame **do**
  $C \leftarrow$ empty;
  if $|f_L| \leq 2$ **then**
    append(I_l, $|f_L|$);
    if $|f_L| = 2$ **then**
      append(I_d, Euclidean(f_{L_x}, f_{L_y}));
      append(C, Mean(f_{L_x}, f_{L_y}));
    end
  else
    $E \leftarrow$ DelaunayEdges(f_L);
    $D \leftarrow$ Euclidean(e_{v_x}, e_{v_y}) $\forall e \in E$;
    $\widetilde{E} \leftarrow E_d \leq T_d \forall d \in D$; ▷ threshold d.
    $\boldsymbol{A} \leftarrow$ BuildCoordinateMatrix($\widetilde{E}$);
    foreach $c \in$ ConnectedComponents($\boldsymbol{A}$) **do**
      append(I_l, $|c|$);
      append(I_d, Mean(D_c));
      append(C, Mean(e_{c_v});
    end
    append($\widetilde{S_C}$, C); ▷ group centres
  end
end
foreach $f_C \in \widetilde{S_C}$; ▷ groups per frame **do**
  $E \leftarrow$ DelaunayEdges(f_C);
  $L \leftarrow$ Euclidean(e_{v_x}, e_{v_y}) $\forall e \in E$;
  append(f_d, Mean(L))
end
$G_n \leftarrow |I_d|$;
$G_{\max(n)} \leftarrow \max(\max(g)\ \forall g \in f_C)$;
$G_{\min(d)} \leftarrow \min(f_d)$;
$G_{\overline{d}} \leftarrow$ Mean(f_d);
$I_{\overline{n}} \leftarrow$ Mean(I_n);
$I_{\overline{d}} \leftarrow$ Mean(I_d);

Each metric is calculated depending on two constants: detection confidence, T_c, the threshold required to include a detection, and a distance threshold, T_d, the maximum group diameter distance (metres). This task is conducted per frame, producing frame-level results: total number of groups, G_n; number of people within a group I_l; mean distance between individuals within group, I_d; and group centres in BNG projection, C. Intermediate per-frame groups are determined by refactoring within threshold detection locations into a coordinate matrix from the Delaunay graph. Individual groups are then classified as connected components per [44]. Upon completion, each set of groups detected per frame supports an additional Delaunay triangulation, permitting final calculation of scene-level metrics: maximum number of groups per frame, G_n; minimum distance between groups; $G_{\max(n)}$; and mean distance between groups, $G_{\overline{d}}$.

We fix the detection confidence threshold to 0.7, meaning inferred detection certainties from object detection as pedestrian below 70% are excluded. A maximum interest area is defined as 6 m between any two individuals. In practice, this task is expensive and distributed among many processing nodes using Python Dask parallelization.

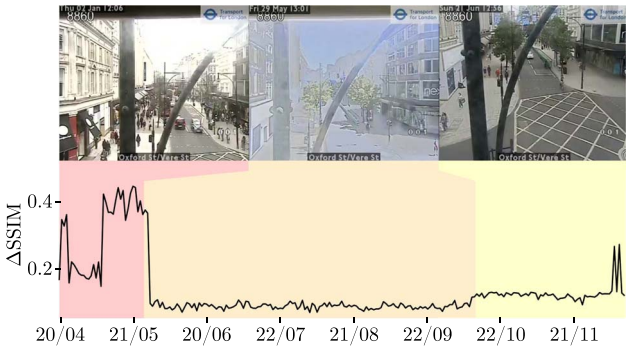


FIGURE 5. Change points detecting scene stability of Oxford St/Vere St. Each colour represents a detected change, example frames included and coloured, respectively, to indicate variation in camera quality and direction.

2.5. Deployment

Deployment provisioning is controlled declaratively by Terraform, containing each component of the processing pipeline (Fig. 6). Kubernetes manages two compute clusters: A GPU accelerated horizontally scaled video processing pool, and a stability-focused horizontally scaled burstable CPU pool executing scheduled tasks and hosting API access points for direct data acquisition and service for control centre output.

3. EVALUATION

3.1. Camera calibration

3.1.1. *World-plane estimation*

Uncertainty in the estimation of the vanishing line and extrinsic camera height arises due to imperfect camera effects eliminated in the assumptions and inaccurate automatic line extraction. The estimated errors in mapped world position, dX, dY, are

TABLE 1. Group metrics calculated over a given sample of detection results.

Metric	Definition
Individuals	Total number of unique pedestrians
No. groups	Total number of unique groups
No. groups max.	Maximum number of exhibited groups in a given location.
Outer group min. distance	Minimum distance exhibited between two groups in a given location.
Outer group mean distance	Mean distance of all exhibited groups in a given location.
Inner group mean size	Mean population within a group.
Inner group mean. distance	Mean of distance exhibited within a group.

evaluated for three randomly selected cameras manually calibrated beforehand using the total differential over all estimated parameters $p_i \in \{u_0, v_0, u_1, h\}$ assuming that the vehicle tracking u, v are accurate at the point of evaluation. The average relative uncertainty in position mapping due to parameter estimation $|\frac{dX}{X}|$ is calculated to be 17.7%, $\sigma = 7.9\%$.

3.1.2. *Calibration to real-world reference*

Of 912 total cameras, 504 were selected for analysis within the Boroughs of Inner London with non-pedestrian motorway scenes predominately excluded. For this training subset, 3298 manually labelled urban furniture anchors were employed for frame real-world calibration. During labelling, care was taken to maximize spacial coverage in each dimension, i.e. anchors were sparsely labelled to include the width and depth of the field of view. There are an average of 5.53 labels, $|F_s|$, per scene s, with a minimum of 4, $|F_s| \geq 4$, where few urban furniture could be identified. Given that we are interested in the distance between individuals, the most appropriate error would be distance between known real-world locations and their pixel coordinates post transformation. The value of this comparison stems from the interest of comparing two individuals or groups within the scene. All possible lengths between all anchors, N, were calculated before optimization. The error function was the mean squared error between these and the learned transformation results, M,

$$\varepsilon = \text{MSE}\left(N, \sum_{i}^{M}\sum_{j\neq i}^{M}\sqrt{i_{x,y}^2 - j_{x,y}^2}\right).$$

This tests the complete calibration pipeline, from pixel coordinates in the image plane, to relative locations in the world plane, and finally to the real-world distances between points. The median optimization error was 0.8210 in BNS, meaning our model is able to locate an object in the image within 82.10 cm. The distribution of this error is shown in Fig. 7.

3.1.3. *Validation*

There does not exist a ground-truth dataset containing relative distances between people in the traffic camera frames around London. To validate this approach, we remove ground-truth anchors enforcing a reliance on fewer manually calibrated examples. For every scene s with a set F_s of urban furniture anchors, we remove exactly one anchor a_s from F_s randomly with uniform and independent probability. We train our model only using anchors in $F_s - \{a_s\}$. The validation test set contains all the out-of-sample removed anchors a_s for each scene s.

The approach led to a mean relative distance error 83.43 cm, a discrepancy of 1.33 cm, with distribution displayed in red, Fig. 7. This indicates that the training procedure marginally benefits from more labels and is resistant to changes in the input training data.

3.2. Object detection

As pre-processing steps, we subset six labels from the Coco 2017 and MIO-TCD localization dataset. Unlike the Coco dataset, MIO-TCD localization dataset contains 11 labels with additional categories such as motorized vehicle, non-motorized vehicle, pickup truck, single unit truck and work van, not found in the Coco 2017 dataset. For comparison, we collapse the different categories of trucks as *truck* and remove labels regarding vehicle motorization. We produce a new collection of manually labelled entities specifically on frames from traffic cameras, using CVAT [45]. The dataset contains 1142 frames and 11 497 bounding boxes as shown in Table 2. For evaluation/validation, we 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.

We fine-tune a pre-trained YOLOv4 weights file on six labels from different training datasets using a batch size of 16, subdivisions of 4, image size of 416 and at least 7000 iterations on a Tesla V100-SXM3-32GB GPU. We train three different models on (1) Coco 2017 training data (2) MIO-TCD training data and (3) Joint data containing random shuffle of Coco 2017 and MIO-TCD training data. Table 2 shows the number of training data by labels. The validation data contain Coco 2017 validation data, MIO-TCD validation data and Jamcam data.

The performances of the three models are shown in Table 3. On the Coco 2017 validation data, the model achieves a mean Average Precision (mAP@0.50) of 67.55%. However, the model trained on Coco 2017 dataset perform poorly on

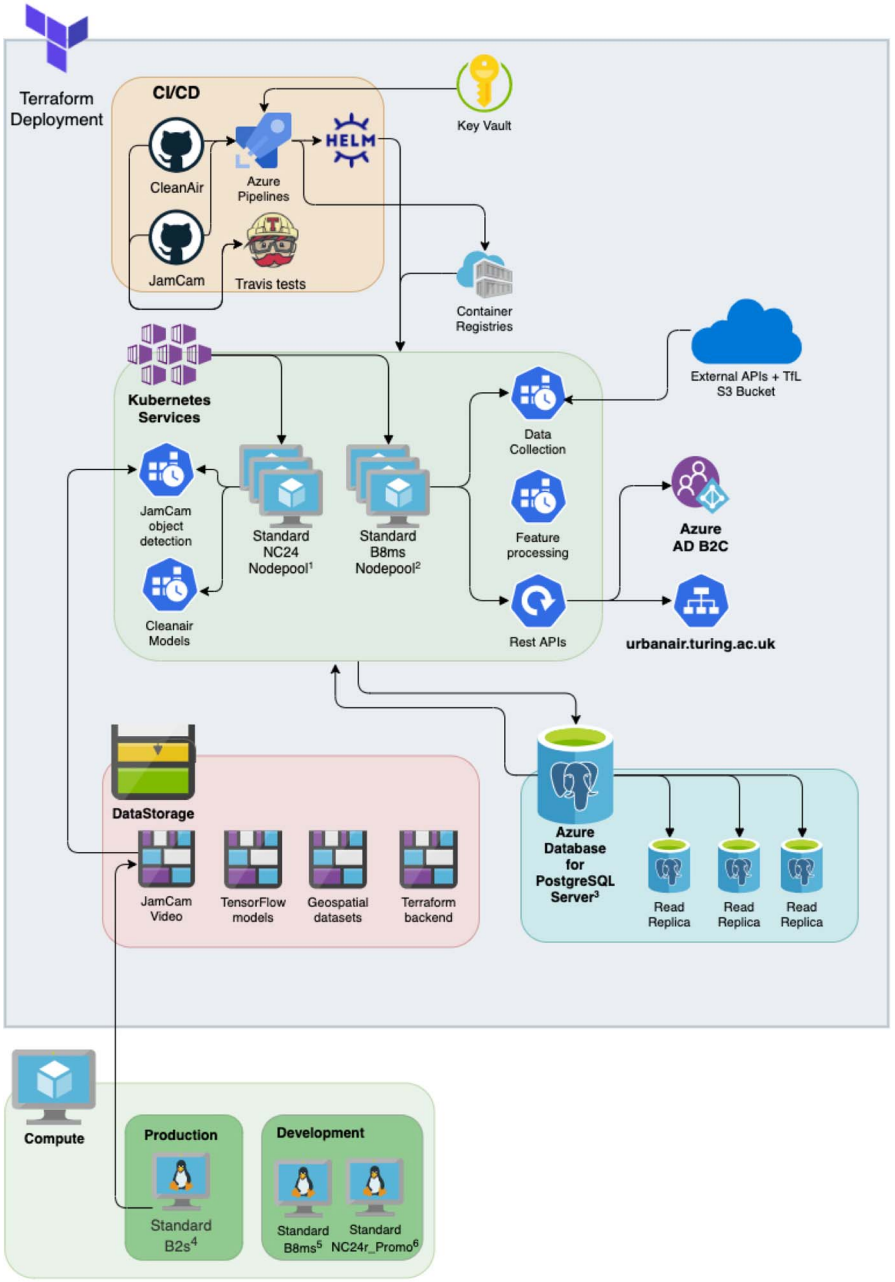


FIGURE 6. Development operations and platform architecture as deployed to Azure cloud services.

MIO-TCD localization validation dataset with an mAP of 20.39%. Likewise, the performance of the model trained on MIO-TCD dataset reduces greatly from 85.80% to 14.24% when Coco 2017 dataset is used as the validation dataset. This behaviour might be as a result of the differences in the resolutions and weather conditions in the two datasets. Performing a joint training creates a balance between the two datasets and increases the model's performance on the independent Jamcam dataset.

4. ANALYSIS

As of 23 September 2021, in the 18-month period the data collection pipeline has processed 19.31 terabytes of footage, for a total of 23 839 346 160 samples of all detection types, spanning all calibrated camera scenes. Of these, 9453 327 651 were rejected either due to irrelevant camera positioning, or out of caution when recorded during a period of high camera variability.

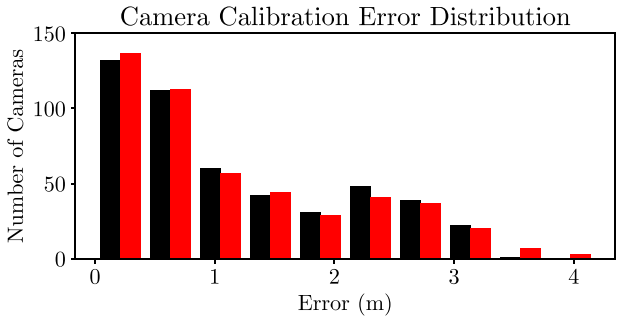


FIGURE 7. MSE distances of all possible distances from ground-truth anchors and transformation results. Full data are displayed in black, dropout validation results in red.

TABLE 2. Training and validation samples per dataset.

Dataset	Person	Bike	Car	M.bike	Bus	Truck
Train						
Coco 2017	262 465	7113	43 867	8725	6069	9973
MIO-TCD	5760	1758	186 767	1484	8443	54 340
Validation						
Coco 2017	11 004	316	1932	371	285	415
MIO-TCD	1368	502	46 730	353	2155	13 694
Jamcam	1233	106	7867	106	203	1982

We define 'Inner Bouroughs' as the Statutory Inner London according to the London Government Act of 1963. The results for this section are generated for a period beginning from our collection date 23 March 2020.

4.1. Scene stability and camera drift

Change in SSIM between the *reference scene* in the active feed, ΔSSIM, is highly relevant to determining sample suitability. Instances of multiple change points promote manual intervention or total rejection; Fig. 5 is an instance of changing stability, whereby the original perspective does not include both pedestrian crossings; then the sensor becomes damaged or over exposed for over 4 months, only to then be positioned differently in October 2020.

Variance in camera positioning was noticeably larger in areas of high pedestrian activity, indicative of the active role Transport for London (TfL) has taken in monitoring pedestrian and vehicle traffic. As demonstrated by comparing the standard deviation of ΔSSIM between inner and non-inner boroughs over the aforementioned time frame results, $\sigma_{\text{inner}} = 0.0636$, $\sigma_{\text{outer}} = 0.0053$.

4.2. Macro policy intervention

Macro interventions within London are defined as either applicable national requirements determined by the central government or city-wide policies set forth by the Mayor's Office. For this example, inner boroughs are selected for their high camera availability for a 12-month subset of these data. Applying group detection per borough provides profiles visible in Fig. 8. A timeline of intervention events [46] are documented in Fig. A.1. Each profile is smoothed under simple local regression [47], taking 5% closest points to (x_i,y_i), estimating y_i under standard weighted linear regression.

There are two predominant results generalized across the city. First visible is a substantial increase in frequency of pedestrian activity and reduction of social distancing during the 'Eat out to help out' scheme between 3 and 30 August 2020. Additionally, activity during the second lockdown plummeted, while social distancing steadily increased. During the reduced restriction periods between these events, a spike in social distancing is seen in most boroughs. This result may be indicative of successful public information campaigns and a willingness to maintain safe distancing conditions, however such a statement is extremely difficult to casually prove.

During the Christmas period the initial social restriction measures rapidly stem the quantity of individuals and groups in all boroughs. After the holiday, activity and distances rapidly grow until restrictions are relaxed leading to plummeting social distancing in almost all boroughs.

4.3. Micro policy intervention

Micro interventions are limited within our dataset, as many COVID protocols cannot be captured on traffic cameras. There exist a number of smaller interventions in the form of pavement extensions. These expansions in pedestrian space include road reclamation in specific areas of high volume, such as near restaurants and public transport stations. For our analysis, we selected three stable scenes from distinct boroughs and filtered our social distance metrics before and after the intervention for comparison.

Locations *Borough High St/Southward St*, *East Road/Vestry St* and *A10 North of Tyssen Rd* (Fig. 9) have 50.2, 43.0 and 153.1 m^2 of pavement area within our digital twin scene. Post-expansion, each gained 40.2, 20.5 and 100.3 m^2 of additional walking space. Before intervention these had mean estimated social distances of 1.33, 1.21, 0.73 m, or as a ratio to area, 0.027, 0.028, 0.005 m/m^2, respectively. Post barricade installation, estimated mean distance rose to 1.50, 1.25, 0.93 m. This is indicative of a clear usage of this space, increased physical distancing and effective policy intervention.

5. CONCLUSION

This work contributes a new social distancing monitoring platform, improves upon the accuracy of the state-of-the-art detection model for an urban domain, introduces a new camera perspective estimation method, provides physical spacing

TABLE 3. Comparing models fine-tuned on the Coco 2017 dataset, MIO-TCD dataset and joint training set using YOLOv4 architecture.

Train	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

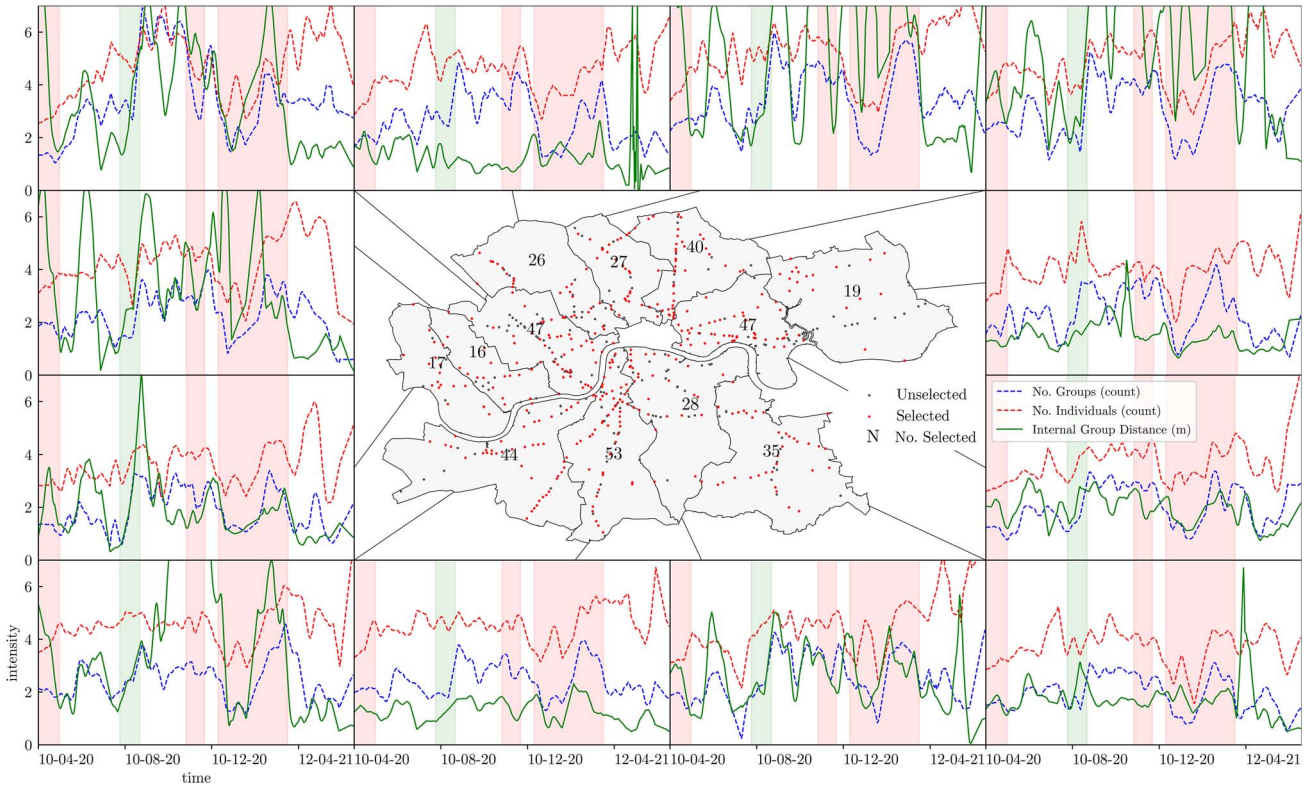


FIGURE 8. Number of individuals, I_n, mean inner group physical distance, $I_{\overline{d}}$, *outer* group social distance, $G_{\overline{d}}$, by inner borough. 'Lockdowns' and 'Eat out to help out' are represented by red and yellow, respectively. Points are representative of camera locations, *selected* and *unselected* in blue and grey, respectively.

metrics in a viable historical context and demonstrates how multiple machine learning techniques may benefit public health. According to the Greater London Authority, this tool enabled them to intervene quickly and identify where street spacing interventions were required. These interventions included moving bus stops, widening pavements and closing parking bays to create space, which enabled social distancing. 'TfL says that it implemented over 700 such interventions at the height of the pandemic's first wave, and that the Turing's tool provided key data for those decisions [48, 49].'

Combined with large-scale inexpensive consumer-distributed computing infrastructure, we provide an option for policy makers to receive a near real-time perspective of their impact via an online interface, Fig. 10. Ongoing directions for this project include validating our early warning detection system, improving the digital twin overall accuracy, providing more

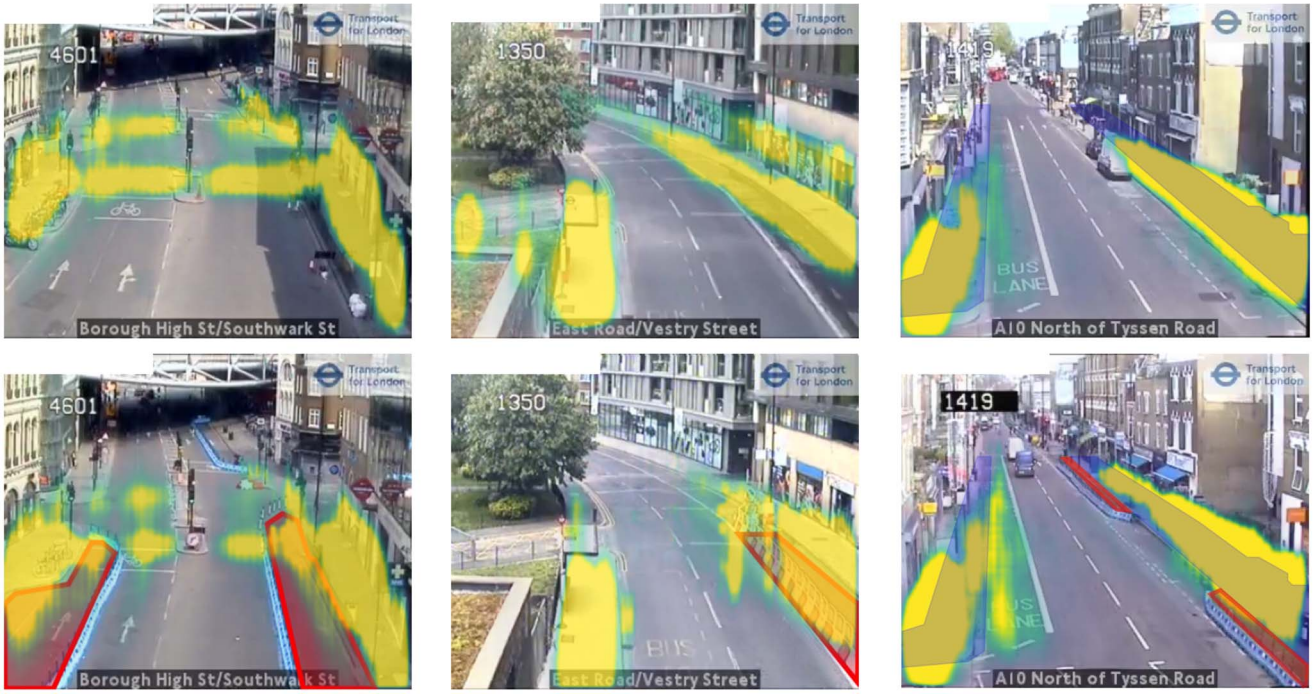


FIGURE 9. Before (top) and after (bottom) of three locations of pavement expansion interventions. Heat map of pedestrian footfall within calibrated pavement and extension (red) areas pre- and post-bollard placement.

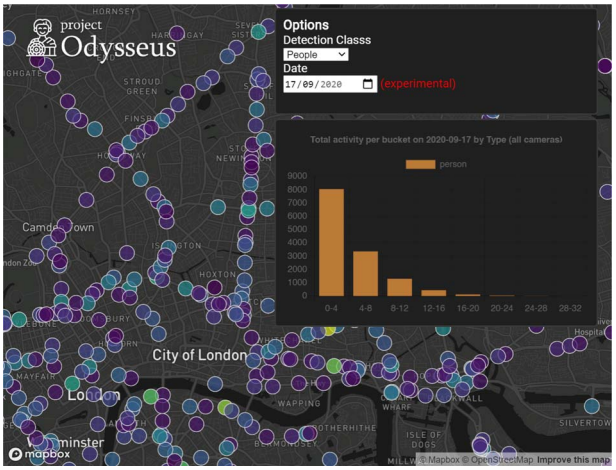


FIGURE 10. Control centre output, real-time interactive pan-London activity metrics as a web application convenient to stakeholders.

'human-in-the-loop' recommendations with high ease of use for policy makers and continuing to provide transparent and interrogatable examples of machine learning applications.

ACKNOWLEDGEMENTS

Funded by Lloyd's Register Foundation programme on Data Centric Engineering and Warwick Impact Fund via the EPSRC Impact Acceleration Account. 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. Camera location and footage are publicly available, the calibration data and a year of daily samples of CCTV footage are available in Zenodo, at https://zenodo.org/record/6472731. We would like to thank Sam Blakeman and James Brandreth for their help on multiple aspects of this work.

REFERENCES

[1] GoogleTrends (2021). Social Distancing Keyword Trend. https://trends.google.com/trends/explore?date=2020-01-01%202021-09-13&q=social%20distancing. [Online; accessed 11-Sept-2021].

[2] GoogleBooks (2021). *Social Distancing Keyword in Google Books*. https://www.google.com/search?q=social+distancing&source=lnt&tbs=cdr%3A1%2Ccd_min%3A0%2F0%2F0000%2Ccd_max%3A1%2F1%2F2020&tbm=bks. [Online; accessed 11-Sept-2021].

[3] May, T. (2020) Lockdown-type measures look effective against covid-19. *BMJ*, 370, 1. Publisher: British Medical Journal Publishing Group Section: Editorial.

[4] Authority, T. G. L. (2021). *Coronavirus (COVID-19) Mobility Report*. https://data.london.gov.uk/dataset/coronavirus-covid-19-mobility-report. [Online; accessed 16-Oct-2021].

[5] HM Government (2020). *Open Government Licence*. http://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/. [Online; accessed 11-Sept-2021].

[6] Hamelijnck, O., Damoulas, T., Wang, K. and Girolami, M. (2019) Multi-resolution multi-task gaussian processes. In Wallach, H, Larochelle, H, Beygelzimer, A, d' Alché-Buc F, Fox, E and Garnett, R (eds.), *Advances in Neural Information Processing Systems*. 32, 1–12. https://proceedings.neurips.cc/paper/2019/file/0118a063b4aae95277f0bc1752c75abf-Paper.pdf.

[7] Bigazzi, A.Y. and Figliozzi, M.A. (2012) Congestion and emissions mitigation: A comparison of capacity, demand, and vehicle based strategies. *Transportation Research Part D: Transport and Environment*, 17, 538–547.

[8] The Alan Turing Institute (2021). *Ethics Advisory Group*. https://www.turing.ac.uk/research/data-ethics/ethics-advisory-group. [Online; accessed 16-Oct-2021].

[9] Cloud Native Computing Foundation (2022). *Kubernetes*. https://kubernetes.io/. [Online; accessed 04-Feb-2022].

[10] Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y.M. (2020) Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

[11] Redmon, J. C., Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2022). YoloV4 implementation. https://github.com/AlexeyAB/darknet. [Online; accessed 04-Feb-2022].

[12] The PostgreSQL Global Development Group (2022). PostgreSQL. https://www.postgresql.org/. [Online; accessed 04-Feb-2022].

[13] Haycock, C., Thorpe-Woods, E., Walsh, J., O'Hara, P., Giles, O., Dhir, N. and Damoulas, T. (2020) An expectation-based network scan statistic for a covid-19 early warning system. In *Machine Learning in Public Health workshop, Neural Information Processing Systems*. Anywhere, Earth, 11 December.

[14] Guidance, B. G. and Support (2021). *UK Government Department of Health and Social Care guided "Support Bubble"*. https://gov.uk/guidance/making-a-support-bubble-with-another-household. [Online; accessed 16-Oct-2021].

[15] Dewitt, B. and Wolf, P.R. (2000) *Elements of Photogrammetry (with Applications in GIS)*. McGraw Hill, New York, NY.

[16] Zitová, B. and Flusser, J. (2003) Image registration methods: a survey. *Image and Vision Computing*, 21, 977–1000.

[17] Caprile, B. and Torre, V. (1990) Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4, 127–139.

[18] Faugeras, O. (1993) *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA.

[19] Tsai, R. (1987) A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3, 323–344.

[20] Schoepflin, T.N. and Dailey, D.J. (2003) Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. *IEEE Transactions on Intelligent Transportation Systems*, 4, 90–98.

[21] Dubska, M., Herout, A. and Sochor, J. (2014) Automatic Camera Calibration for Traffic Understanding. In Valstar, Michel, French, Andrew and Pridmore, Tony (eds.), *Proceedings of the British Machine Vision Conference*, pp. 42.1–42.12. BMVA Press, Nottingham. https://doi.org/10.5244/C.28.42.

[22] Lai, A.H.S. and Yung, N.H.C. (2000) Lane detection by orientation and length discrimination. *IEEE Trans. Syst., Man. Cybern. B*, 30, 539–548.

[23] Song, K.-T. and Tai, J.-C. (2006) Dynamic Calibration of Pan-Tilt-Zoom Cameras for Traffic Monitoring. *IEEE Trans. Syst. Man Cybern. B Cybern.*, 36, 1091–1103.

[24] Dong, R., Li, B., and Chen, Q.-M. (2009) An Automatic Calibration Method for PTZ Camera in Expressway Monitoring System. In Kang Ryoung Park, Sangyoun Lee and Euntai Kim (eds.), *2009 WRI World Congress on Computer Science and Information Engineering*, March, pp. 636–640. MDPI, Basel, Switzerland.

[25] Fung, G.S.K. (2003) Camera calibration from road lane markings. *Optical Engineering*, 42, 2967.

[26] Liebowitz, D. and Zisserman, A. (1998) Metric rectification for perspective images of planes. *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*, June, pp. 482–488. ISSN: 1063-6919. IEEE, NY, USA.

[27] Cathey, F. and Dailey, D. (2005) A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005*, pp. 777–782. IEEE, NY, USA.

[28] Criminisi, A. (2001) *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. Springer, London, London, UK.

[29] Schoepflin, T.N., Dailey, D.J. *et al.* (2003) *Algorithms for estimating mean vehicle speed using uncalibrated traffic management cameras* Technical report. Washington (State). Dept. of Transportation, Olympia, WA, US.

[30] Simonyan, K. and Zisserman, A. (2015) Very deep convolutional networks for large-scale image recognition. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*.

[31] Xie, S. and Tu, Z. (2015) Holistically-nested edge detection. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1395–1403. IEEE, NY, USA.

[32] Liu, Y., Cheng, M.-M., Hu, X., Wang, K. and Bai, X. (2017) Richer convolutional features for edge detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5872–5881. IEEE, NY, USA.

[33] Shapiro, L. and Stockman, G. (2001) *Computer Vision*. Prentice House London, London, UK.

[34] Zhang, Z., Tan, T., Huang, K. and Wang, Y. (2013) Practical Camera Calibration From Moving Objects for Traffic Scene Surveillance. In *IEEE Transactions on Circuits and Systems for Video Technology*Conference Name: IEEE Transactions on Circuits and Systems for Video Technology (Vol. 23), pp. 518–533. IEEE, NY, USA.

[35] Transport for London (2020). Our open data. https://www.tfl.gov.uk/info-for/open-data-users/our-open-data. [Online; accessed 11-Sept-2021].

[36] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A.C. (2016) Ssd: Single shot multibox detector. In Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) *Computer Vision – ECCV 2016, Cham*, pp. 21–37. Springer International Publishing, Cham, Switzerland.

[37] Redmon, J. and Farhadi, A. (2018) Yolov3: An incremental improvement. *ArXiv*, **abs/1804.02767**.

[38] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You only look once: Unified, real-time object detection. In *2016*

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788. IEEE, NY, USA.

[39] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L. (2014) Microsoft COCO: Common Objects in Context. In Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) *Computer Vision - ECCV 2014, Cham Lecture Notes in Computer Science*, pp. 740–755. Springer International Publishing, Cham, Switzerland.

[40] Luo, Z., Branchaud-Charron, F., Lemaire, C., Konrad, J., Li, S., Mishra, A., Achkar, A., Eichel, J. and Jodoin, P.-M. (2018) MIO-TCD: A New Benchmark Dataset for Vehicle Classification and Localization. In *IEEE Transactions on Image Processing* (Vol. 27), pp. 5129–5141. Conference Name: IEEE Transactions on Image Processing, NY, USA.

[41] Horé, A. and Ziou, D. (2010) Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pp. 2366–2369. IEEE, NY, USA.

[42] van den Burg, G.J.J. and Williams, C.K.I. (2020) An evaluation of change point detection algorithms. *ArXiv*, **abs/2003.06222**.

[43] Killick, R., Fearnhead, P. and Eckley, I.A. (2012) Optimal detection of changepoints with a linear computational cost. *J. Am. Stat. Assoc.*, 107, 1590–1598.

[44] Pearce, D.J. (2005) *An improved algorithm for finding the strongly connected components of a directed graph*. Victoria University, Wellington, NZ.

[45] Sekachev, B., Manovich, N., Zhiltsov, M., Zhavoronkov, A., Kalinin, D., Hoff, B., Tosmanov Kruchinin, D., Zankevich, A., Dmitriysidnev Markelov, M., Johannes222 Chenuet, M., a andre, telenachos, Melnikov, A., Kim, J., Ilouz, L., Glazov, N., Priya4607, Tehrani, R., Jeong, S., Skubriev, V., Yonekura, S., vugia truong, zliang7, lizhming, and Truong, T. (2020). opencv/cvat: v1.1.0.

[46] Authority, T. G. L. (2021). *GLA COVID-19 Restrictions Time-series*. https://data.london.gov.uk/dataset/covid-19-restrictions-timeseries. [Online; accessed 16-Oct-2021].

[47] Cleveland, W.S. (1979) Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.*, 74, 829–836.

[48] Lloyd, J. (2021). *Helping London Navigate Lockdown Safely*. https://www.turing.ac.uk/research/impact-stories/helping-london-navigate-lockdown-safely. [Online; accessed 16-Oct-2021].

[49] Vryzakis, A., Snaith, B., and D'Addario, J. (2021). Project Odysseus - using existing infrastructure to tackle new problems. https://theodi.org/article/project-odysseus-using-existing-infrastructure/. [Online; accessed 16-Nov-2021].

APPENDIX A. LOCKDOWN PERIODS

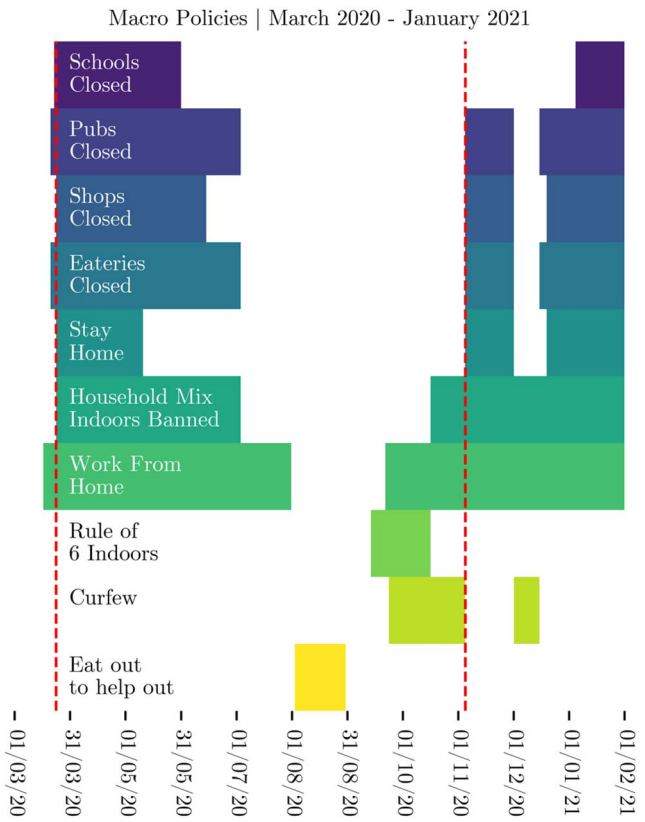


FIGURE A.1. City-wide policy interventions. Red lines indicate lockdown start dates.