"Observations of Urban Activities with Computer Vision"

Guo Xiang **Ong**[1], Ye **Zhang**[1], Zhe **Jin**[2], Choon Meng **Seah**[2], Tat Seng **Chua**[2]

[1]*National University of Singapore, 4 Architecture Drive, Singapore 117566*
[2]*National University of Singapore, 13 Computing Drive, Singapore 117417*

E-mail of corresponding Author: akizy@nus.edu.sg

ABSTRACT

Advances in Computer Vision and Deep Learning has enabled the use of new techniques in the study of urban spaces. While there has been significant research efforts in the use of Computer Vision to study urban forms with semantic segmentation, there is a conspicuous lack of development in using Computer Vision to observe activities in urban settings. Conventional methodologies to survey and observe human activities primarily involves labor intensive manual field audits and questionnaires. This made it time consuming and expensive to scale. Given the interdependence of urban forms and human activities, there is a need to improve how activity observations are conducted to match the data quantity and quality that can be generated from Computer Vision based techniques in the mapping of urban forms. In this paper, we will present an approach that utilizes an action detection model to address the laborious nature of activity observations and demonstrate how such an implementation can allow the survey of large areas with minimal manpower.

Keywords: *action detection, object detection, activity observation, urban survey, computer vision, deep learning*

1. INTRODUCTION

The analysis of the relationship between urban forms and human activities is a primary focus of the urban design discourse. Seminal works over the last century have underscored the significance of urban forms on the legibility of a city [1], its impact on public health by facilitating physical activities [2] and its effects in promoting urban vitality [3]. All of which presents a common consensus on how urban forms and human activities are interdependent.

Recent advances in Deep Learning have revolutionized the field of Computer Vision [4]. The breakthrough implementation of a deep Convolutional Neural Networks (CNN) by Krizhevsky *et al*. in 2012 [5], drastically lowered the error rate for object detection. Their success cemented CNN as the dominant approach for recognition and detection tasks [4]. Much of the existing research is open sourced with direct R&D support from technology giants including Google, Facebook and Microsoft. This led to its adoption across diverse research disciplines outside of computer science, in particular disciplines that deal with image based problems such as medical imaging [6] and defect inspections in manufacturing [7].

Within the field of urban design and planning, researchers have harnessed these new techniques to study urban forms [8] and assess how the built environment is perceived [9]. These efforts have allowed researchers to quantify semantic information in images [10] and work with much larger datasets [11] that attempt to map the physical environment. Among these efforts, there are significant research interests applying detection models on images extracted from Google Street View. A common application involves using semantics segmentation on street view images to measure the composition of the urban environment [8; 9]. Related approaches see the utilization of image classifiers to automate the process of classifying urban streets with street imagery data [12] and object detection models to detect traffic signs [13].

Despite the consensus on the interdependence of urban forms and human activities, the overall focus of the urban design community on utilizing Computer Vision has been disproportionately skewed towards the study of urban forms. There is a conspicuous lack of development in utilizing these new tools to observe activities in public urban spaces. Existing methodology to survey human activities involves primarily manual field audits [14; 15] and questionnaires[16; 17]. The labor intensive nature of field audits makes data collection both time consuming and expensive to scale. These factors ultimately limit the overall size of the study area and the size of the corresponding dataset.

It is high time for us to re-evaluate how activity surveys can be done to match the data quantity and quality that advances in Computer Vision and Deep Learning has brought about for the study of urban forms. The same breakthroughs can potentially remedy the constraints faced by existing activity observation methodology, allowing researchers to collect larger population samples to gain a more accurate understanding of activity distribution and potentially deriving new insights.

In this paper, we will propose an approach that addresses the laborious nature of street level activity observations. We will adapt an *activity detection model* to automate the survey of human activities on the street level. We will then demonstrate how the process can be implemented through our study of the Park Connector Network (PCN) [18] in Singapore. Lastly, we will discuss the results of the deployment and its limitations.

2. ACTIVITY OBSERVATIONS – EXISTING CONSTRAINTS AND NEW OPPORTUNITIES

2.1. Existing Constraints – A dilemma between survey scope and resources

Conducting a headcount of people while recording the demographics and their interactions are crucial parts of activity observations that can enable researchers to evaluate how urban spaces are used. A typical process requires trained human surveyors to observe and record activities on site or to collect questionnaire responses via interviews [14-17]. Such a process would suffice for studies that involve a handful of selected locations and would entails modest manpower commitment. However, to obtain an accurate understanding across broader geographic areas, we need to expand beyond conducting observations at a couple of isolated locales. As we attempt to survey more locations and at different times of the day for improved accuracy, we would expect our manpower and time commitment to grow exponentially.

Researchers often face a dilemma of deciding between smaller sample sizes or to commit additional resources. Opting for smaller sample sizes could impede our ability to make inferences on the population and in turn limits the accuracy of understanding. While opting to commit more time and resources may not be possible for research groups with smaller budgets. This illustrates the issue of *scalability* and the trade-offs that are inherent in a labour intensive process.

2.2 New opportunities: Automating Observations with Computer Vision

At the time of writing, object detection and segmentation in context is a well-established and a popular research domain in Computer Vision. There are state-of-the-art detection models being released every few months[1]. Most of them are trained with and benchmarked against the COCO (Common Objects in Context) dataset [19] and within which contains thousands of instances of people captured in everyday environments. This implicitly implies that the models that performed well against the benchmark, are effective at detecting and identifying human figures within common environments.

Figure 1. Visualization of each actor's list of actions and confidence score. Note that the model can detect up to 80 different actions (details later) and only the top scoring actions are shown in the visualization. Source: authors.

A closely related domain in Computer Vision activity detection, where researchers seek to use train models to identify the activity of a person [20]. There are existing implementations of activity detection models that are already capable of differentiating common actions among multiple actors [21]. It extracts human figures as 'actors' within a scene, and generates a list of actions for each actor and the respective confidence score for each action (Figure 1).

Despite the relative maturity of object detection and potentials of action detection, there has been little exploration into how these tools can be exploited to assist the observations of activities in public spaces. The rationale behind activity observations is fundamentally, to understand the usage patterns within a given location. A basic observation can simply be the collection of headcount and recording the activities of individuals observed. Given the current capabilities of action detection, automating the processing of counting and basic activity detection is well within reach. Hence, in our quest to address the scalability issue of conventional methodologies, the next logical steps will be to attempt to test the capabilities of the action detection models with the large scale observations of human activities in real world environments. In Section 3, we will first attempt to outline a generic approach that utilizes Computer Vision tools for activity observations. After which, we will demonstrate how such an approach can be implemented in Section 4.

[1] https://paperswithcode.com/task/object-detection. This website maintains a leader board of the state-of-the-art models.

3. AUTOMATING ACTIVITY OBSERVATIONS – VIDEO SURVEY AND DETECTION WITH COMPUTER VISION

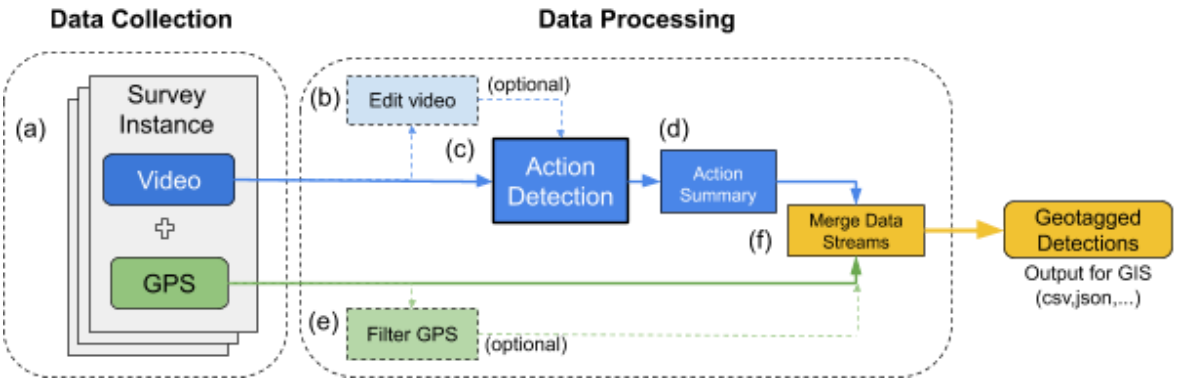

Figure 2. Process overview. Processing two streams of data in parallel before merging. Source: authors.

This section will describe our proposal to deploy activity detection models for activity survey and subsequent integration into a GIS based workflow. The approach is by no means the definitive method, but rather it should serve as a working template to highlight what a typical process might entail.

The challenge of implementing activity surveys with Computer Vision can be divided into two problem domains, *data collection* and *data processing* (Figure 2). The two domains are by design discrete and can be conceptualised to operate on two separate streams of data, the *video data stream* and the *GPS data stream*. These streams are processed separately in stages before finally being merged into a single output stream (Figure 2f).

Videos are used as the primary means of collecting activities data on the ground and a pre-trained action detection model is then used to extract the data from the video. GPS is used to determine where the videos were taken and allows us to accurately map the location of the detection results in physical space.

3.1. Data Collection

Data collection involves collecting *survey instances*, each of which should consist of a video recording of the surveyed area and a GPS log of the corresponding locations (Figure 2a). This can be easily done with a separate camera and a GPS logger when the nature of investigation primarily involves fixed and isolated observation locations.

However, when there is a need to observe a much larger stretch of area beyond the field of view of a single camera, having to set up multiple fixed observation posts may not be logistically practical. Instead, we recommend the use of a video camera mounted on a moving platform for the survey of large areas (to be demonstrated in section 4) to reduce the complexity of the observation setup. In this scenario, we would also recommend the use of a video camera that has inbuilt GPS capabilities (e.g. a GoPro Hero) to make it easier to match the video footage to the GPS data.

The exact video recording and GPS data format will vary depending on the devices available and will need to be tested during implementation. However, there are general guidelines to the collection process that will help

to ensure that the footage is well captured and usable. Firstly, the videos themselves should be taken at close to eye level as most training datasets are extracted from home videos and movies[2] and is closer to how humans perceive crowds. Each video should be a continuous clip that has not been spliced together from multiple sources to ensure consistent tracking of the actors in a scene. And finally, the videos should also be captured in good lighting conditions where the people and environment is clearly visible, which minimizes detection noise and enables better tracking.

3.2 Data Processing

The data processing stage involves separately processing the *video portion* and the *GPS portion* of each survey instance before merging them into a single output. The bulk of the processing will be on the video portion of the survey instance.

The video portion is to be processed by a Deep Learning model that was trained to recognize people and their activities within common contexts. We propose to use an *activity detection model* as it can effectively detect people in a scene and provide an interpretation of what the individual is doing (implementation details in section 4). Depending on the specifications of the selected activity detection model and the quality of the recorded video, there may be a need to first edit the raw video footages, to adjust parameters (Figure 2b) such as image brightness, image resolution or dimensions, before running it through the model (Figure 2c).

The action detection model will detect and track actors over the duration where the actors are visible in the video, and generates a list of predictions of what it thinks the actor's action is (Figure 1). This prediction can change over the duration of tracking as the overall silhouette of the actor changes in relation to the camera (Figure 3). This variability in prediction results may make it difficult to interpret and record what the actor is doing. Hence, in order to reduce the data noise caused by the variability, we can *summarize* the actor's most frequent action. A log of the actor's action history over the course of tracking is kept and the most frequent action up till the point of query is assigned as the actor's overall action (Figure 2d).

Figure 3. Detected action of tracked actors changing across video frames. Source: authors.

The GPS portion of the survey instances should not require additional processing. However, readers should note that GPS devices need time to acquire the GPS signals and calibrate its location. Some devices may start logging during this phase resulting in incorrect location logs at the beginning. Hence, outputs from GPS

[2] https://gist.github.com/jin-zhe/3a6054e99162bc9277940867f942bba2. This gist provides an overview of recent action detection datasets and their detection classes.

devices may need to be filtered (Figure 2e) to ensure that only valid GPS logs that describe the location where the video is taken remains.

Once the GPS data and the action detection data are processed, they are merged into a single output stream that consist of a list of geotagged results (Figure 2f). The output should ideally be formatted in common formats such as CSV or JSON, which is compatible with most GIS and mapping applications, for downstream analysis in a more conventional GIS workflow.

4.0 DEMONSTRATION

Figure 4. Park Connector Networks in Singapore. Source: National Parks Board.

This section will demonstrate the implementation of our proposal, which we had outlined in Section 3, in the survey of the Park Connector Network (PCN) (Figure 4). The PCN is a system of greenways strategically planned to link parks and open spaces across the entire Singapore [18]. The environmental features of these thoroughfares vary along a spectrum. At one end of the spectrum we have predominantly built-environments with hard edges, while at the other end are natural environments with lush greenery (Figure 5).

Figure 5. (Left) Landscaped thoroughfare with predominantly built-up features. (Right) Lush greenery in a park. Source: authors.

The surveys were also conducted at various times of the day under different environmental lighting conditions. Hence, the scope of the investigation gave us an opportunity to test the effectiveness of video surveys with Computer Vision tools under different landscape, lighting and crowd conditions.

4.1. Data Collection - Mobile video survey with geolocation

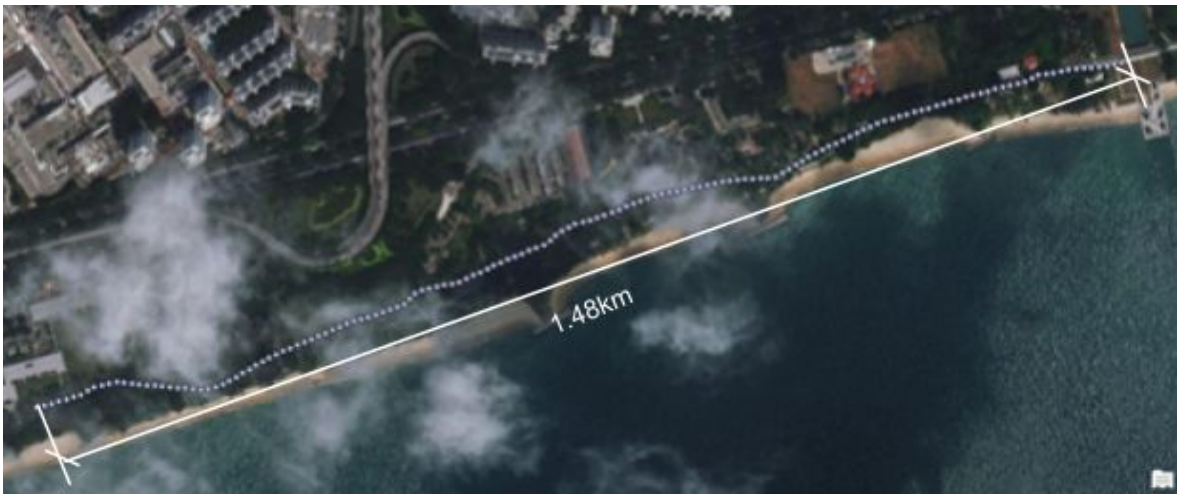

Figure 6. A typical survey segment of approximately 1.48km. Source: authors.

In our implementation, we chose to use a GoPro Camera as the primary means to collect survey instances. Video footages recorded with the GoPro are embedded with GPS data which can then be extracted to accurately pinpoint the location that corresponds to the time signature within the video.

Due to the large distances that need to be covered (Figure 6), we had mounted the GoPro camera on an electric scooter, which provides a relatively stable recording platform and increases the range that a single surveyor can cover. The GoPro camera was mounted on the handle bars at approximately 1.5m to position it closer to the eye level.

Each survey instance consists of a continuous video clip taken with the scooter making a single non-overlapping pass along the survey segment, in either direction of the path. Multiple survey instances are collected for each segment to reduce variability by making multiple scooter passes. While repeated passes may sound labor intensive, the process can be performed with relative ease and efficiency. In our approach, data processing is decoupled from data collection and occurs offsite (Figure 2). This means that the surveyor only has to spend time onsite to capture the required videos and do not need to be concerned with identifying what was observed. To put things in perspective, a single scooter pass along the 1.48km segment shown in figure 6 takes approximately 12 minutes. Within a span of an hour, a surveyor can complete 5 survey instances of the above segment by travelling to and fro between the start and end points.

4.2. Data Processing – Detection with action detection model

Recall that in our approach, each survey instance consists of the video data stream and the GPS data stream. As the GPS data are embedded within the GoPro videos for, we would first need to extract them such that they can be processed.

We had used a set of custom scripts and guides[3] from the GoPro forums to extract the GPS telemetry into a separate GPX file. The exact method that we had used no longer works at the time of writing due to new software releases and outdated dependencies that were no longer maintained. Readers who intend to use the GoPro cameras for similar purposes should refer to the official GoPro website and forums for the latest updates and documentation. Once the GPX file is extracted, we can then process the video file and the GPX file as separate data streams in parallel, before merging the processed data into a single output stream towards the end.

4.2.1 Video – Running the action detection model and summarizing overall actions

There were multiple research groups that were independently developing state-of-the-art models for activity detection back in late 2018 when we first started researching means to use Computer Vision to perform activity observations. Our objective was simply to find an open source implementation with a pre-trained model that we could adapt for our experiments, as we did not have the resources nor was it our research focus to develop cutting edge models. One of the open sourced action detection models available at the time was Ulutan and team's implementation of a real-time action detection model [21]. We had tested Ulutan's implementation action detection model and found that with some modifications, it can reliably detect the actions of the actors within our video surveys under most circumstances (limitations to be discussed in section 5).

Ulutan's implementation was initially optimised for real-time detection with the object detection model SSD MobileNetV2 [22] in their pipeline. We found that by substituting SSD MobileNetV2 with another object detection architecture Faster RCNN [23], we can achieve more accurate detection results at the expense of speed.

We also only had access to a single GTX 1080 Ti GPU in our hardware setup. During our tests, the GPU ran out of memory and crashed when processing scenes with more than 14 actors at a time. We found that by limiting the number frames held in the GPU from Ulutan's original implementation of 32 frames to 16 frames, we can increase the number of actors that we can track to 26 without crashing the GPU. Readers should note that the limits may vary with different hardware and code implementation. In general, implementations that make use of multiple GPUs in parallel processes should not have the same memory limitations.

[3] https://github.com/jin-zhe/gopro/tree/a7e563a65dc934515a88a5f2408db674b92a58fc

Figure 7. The actions detected for a typical scene. The tally of the above scene is “stand”=5, “walk”=3, “ride”=2, “sit”=1, “run/jog”=0. Any other actions that are not part of the 5 categories are not recorded. Source: authors.

Aside from the above modifications, additional code[4] was written to contextualize the detections to our survey environment and to perform the action summary (Figure 2d). The model which we had used was trained on the AVA dataset [24] and within which there are 5 action labels that are relevant to activity observations in our survey area. These 5 action labels, “walk”, “stand”, “run/jog”, “sit” and “ride” are also the most commonly detected actions within our survey. Any other actions that the model detects were unlikely to be found in our survey area and are likely to be misdetections. These were filtered and were not included in the tally.

The most frequently detected action for each actor was assigned as the ‘summarized’ action for the actor (refer to section 3.2). As the video progresses, a tally of all the actor’s action was generated (Figure 7) at every 8 video frames. This was based on Ulutan’s original implementation where the action detection was run every 8 frames. These tallies for the entire video are then logged in a list (Figure 8), which pairs the detection tallies with the corresponding video frame in the video.

Frame no.	run/jog	sit	stand	walk	ride
5580	0	2	1	5	0
5588	0	2	1	5	0
5596	0	2	2	6	0
5604	0	3	2	5	0
5612	0	2	2	4	0
5620	0	1	5	5	0
5628	0	1	2	5	0
5636	0	1	1	4	0

Figure 8. A snippet of the tally list. Each row is the tally of the actions detected in a scene. The first column records the video frame where the tally is generated. Notice that this tally is generated every 8 video frames. Source: authors.

In summary, for each video that was processed by the action detection model, a list of action tallies was generated for the entire length of the video clip. The tallies corresponds to a specific video frame within the video. At this stage, the tallies were not yet tagged with a geographic location. The following steps will resolve this by matching the GPS data within the GPX file with the action tally list.

[4] Github repository for our project source code. https://github.com/gxite/pcn-acam

4.2.2 Matching the video data stream with the GPS data stream

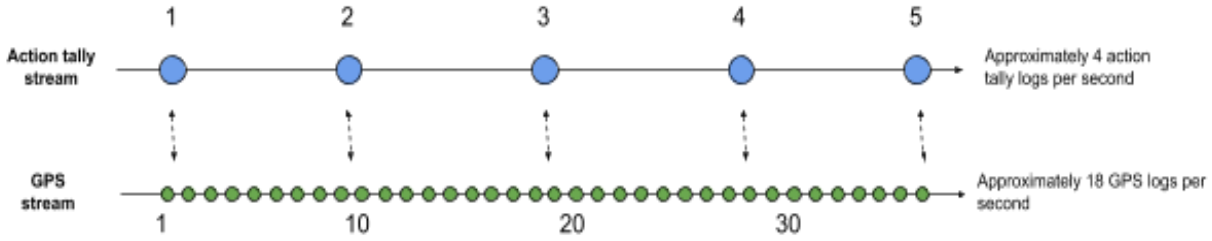

Figure 9. The action tally data and the GPS data streams at different frequencies. Source: authors.

Recall that each of the GoPro videos consist of both the actual video clip and embedded GPS telemetry (Section 4.1). We had extracted the embedded GPS data in a separate GPX file that consisted of a list of GPS coordinates at the start of Section 4.2 and in section 4.2.1 we had processed the video clip to obtain an action tally list (Figure 8). These two lists of data can be visualized as two indexed data streams (Figure 9). They had to be matched such that each action tally is appended with the correct GPS location, which would give us a single list of geotagged action tallies.

The GoPro video was captured at approximately 30 frames/second and at every 8 frames the action tally was logged. This results in an action tally stream that has approximately 4 logs/second. The GPS stream extracted from the GoPro has approximately 18 logs/second. The difference in the log frequency meant that we cannot get the corresponding GPS location of each action tally by directly associating the index between 2 streams (i.e. we cannot map index 2 in the action tally stream to index 2 of the GPS stream). Instead, the corresponding index can be obtained by multiplying the index of the action tally by the ratio *<Total no. of GPS logs>/<Total no. of action tally logs>*. With this, we can then append the corresponding GPS coordinate to each action tally, producing a single geotagged output list as shown in Figure 10, where the frame number field is now replaced with longitude and latitude.

Latitude	Longitude	run/jog	sit	stand	walk	ride
1.28627	103.8599	0	0	1	5	0
1.286267	103.8599	0	0	5	5	0
1.286265	103.8599	0	0	3	5	0
1.286262	103.8599	0	1	4	4	1
1.28626	103.8599	0	2	6	5	1
1.286257	103.8599	0	4	5	5	2
1.286256	103.8599	0	6	6	5	2
1.286255	103.8599	0	10	3	5	0

Figure 10. A snippet of the action tally list with the GPS coordinates appended. Source: authors.

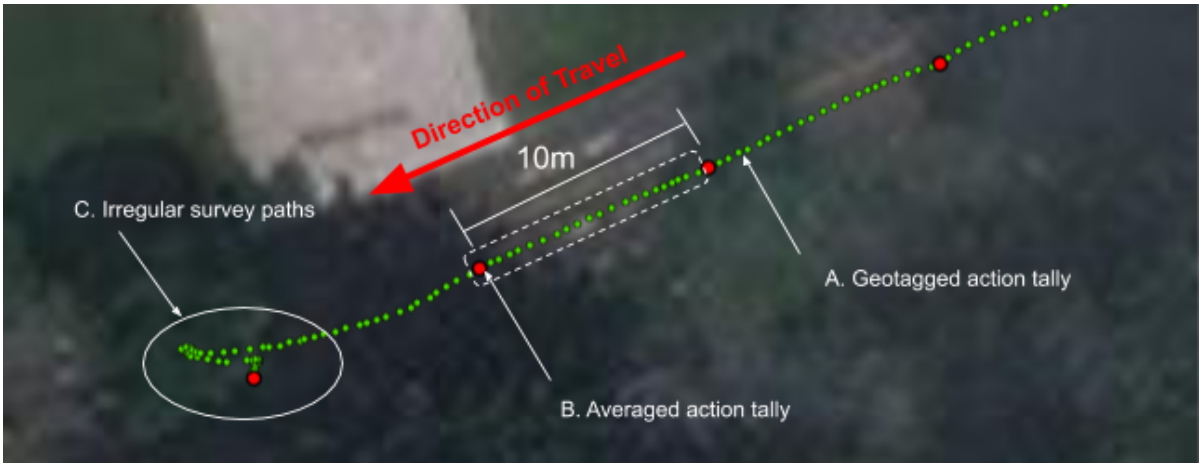

Figure 11. Visualization of the location of the action tallies in GIS. (A) Geotagged action tally. (B) Averaged action tally. (C) Irregular survey paths. Source: authors.

The geotagged action tally at the current stage of the process could already be directly plotted within a GIS application (Figure 11a). However, there were irregularities and inconsistencies that needed to be addressed. The survey scooter may take irregular paths (Figure 11c) or dwell at a location for too long (inconsistent travel speeds). This would result in excess and repeated activity logs in a localized area that could create a false signal that suggests a large gathering of people.

This can be addressed by averaging the geotagged action tallies and log them at fixed intervals (Figure 11b). We selected an interval of 10 metres as that is the approximate distance where the actors captured by the camera is registered by the action detection model. The averaging was done in the direction of travel. Each interval point (Figure 11b) holds the average value of all trailing action tallies (Figure 11a) within a 10m range. The camera captures activities that were approximately 10m in front of the lens while the GPS records the position of the camera. This property results in an offset between where the detections are plotted and its *actual* location onsite. Hence, by averaging in the direction of travel, we can compensate for and reduce the detection offset. At the end of the process, we will end up with the final list of *the average action tallies* at fixed intervals *of 10m* (Figure 12).

Latitude	Longitude	run/jog	sit	stand	walk	ride
1.286288	103.8603	0	0.011905	0.107143	0.035714	0
1.286333	103.8602	0	0.117647	0.588235	0.705882	0
1.286363	103.8601	0	0.8125	1.0	1.125	0
1.286387	103.86	0	0.85	3.0	1.05	0
1.286355	103.8599	0	0.166667	3.555556	2.444444	0
1.286267	103.8599	0	5.315789	4.842105	5.0	0.421053

Figure 12. A snippet of the averaged action tally list. Notice that fields now consist of floats instead of integers. Source: authors.

4.3 Overall Results

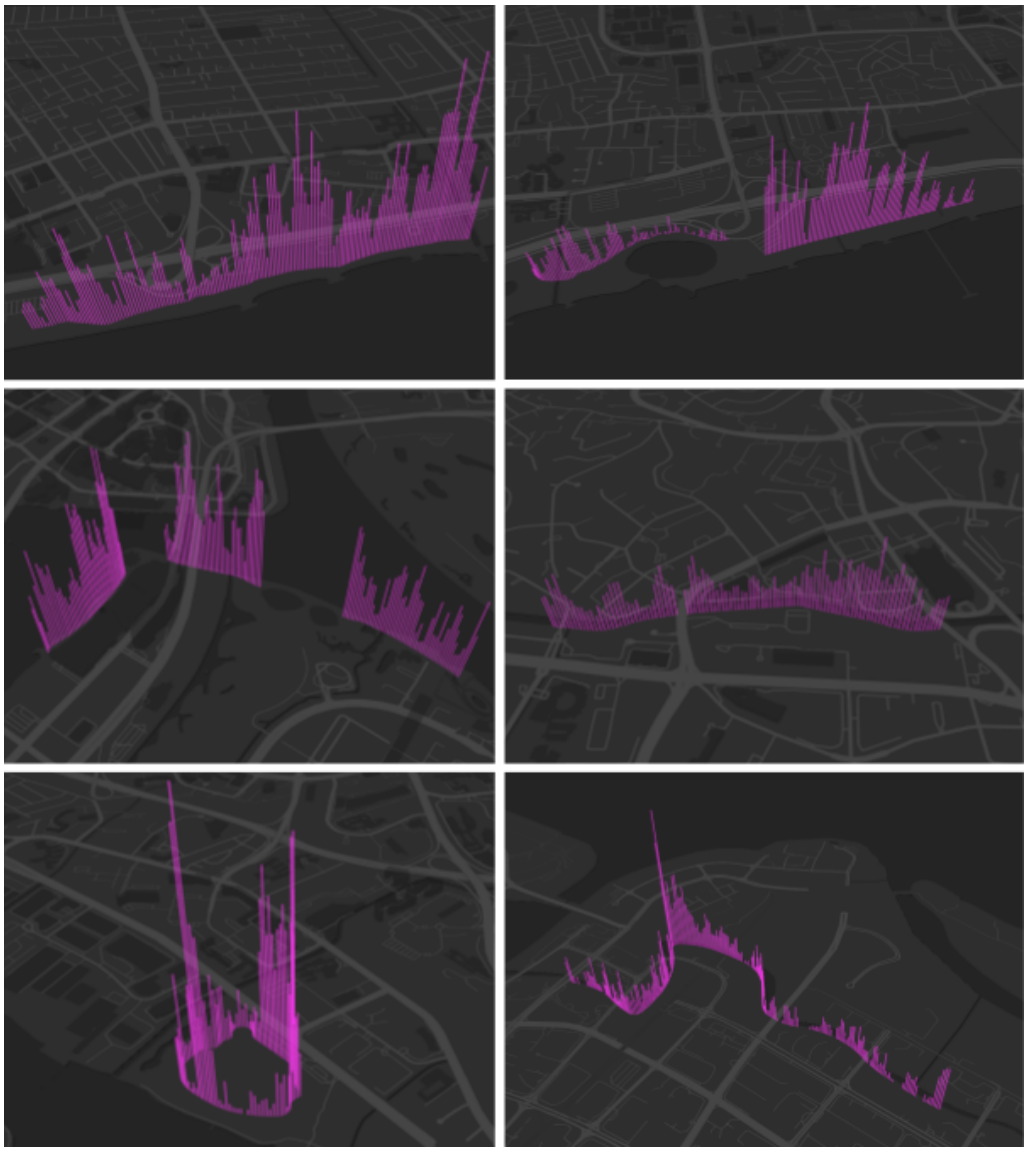

Figure 13. Visualization of the distribution of people across different survey segments. The length of each vertical line represents the number of people detected at an interval point. Source: authors.

Our proposed process generally worked well in detecting and tracking multiple actors in our survey environment (Figure 7). It was capable of correctly assigning action labels to each actor with some limitations (to be discussed in section 5). The activity observations were also simple and efficient to execute. This made it possible for us to collect observations quickly and feasible to work with a large number of observation points. In trying to visualize the distribution of people observed across the survey segments, the density of the available observation points made it intuitive to observe any fluctuations in the distribution (Figure 13). The general ease of data collection also made it feasible to observe survey segments over multiple times within a day. Figure 14 shows the cumulative frequency of people detected at the interval points along a survey segment, over 6 consecutive time intervals.

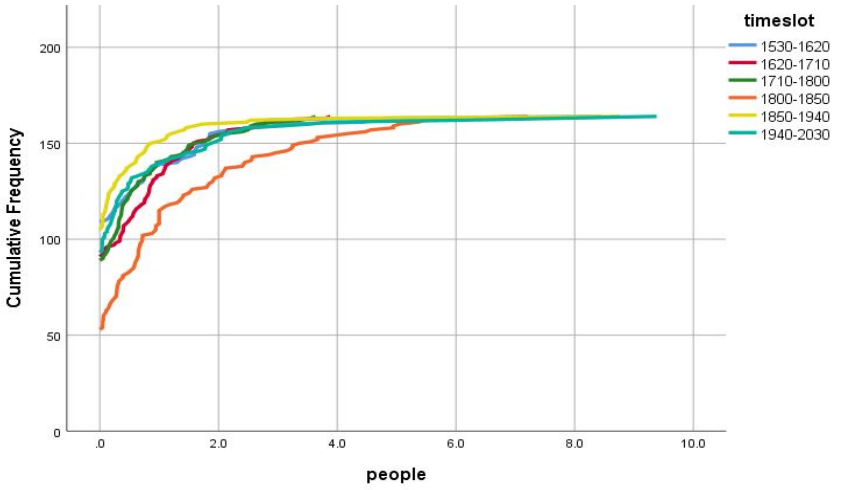

Figure 14. The cumulative frequency of people along a survey segment over 6 consecutive time intervals. Source: authors.

The difference between the gradients of the curves allow us to see how the overall distribution of people changes across time. A curve belonging to the 1850-1940 timeslot can be interpreted as follows. The steep gradient shows that the detections were predominantly small clusters of fewer than 2 people. The long tail suggests that there are likely hotspots where we were observing clusters of between 6 to 8 people. The relatively high y-axis intercept indicates that more than half of the survey points had 0 detection and could indicate sparse distribution of people across the segment. Comparatively, a curve with a gentler gradient at the 1800-1850 timeslot and a lower y-axis intercept indicates that there were larger clusters of people observed that are between 0 and 6 people and fewer 0 detections. This suggests the presence of more activity and that people distributed more evenly across the segment.

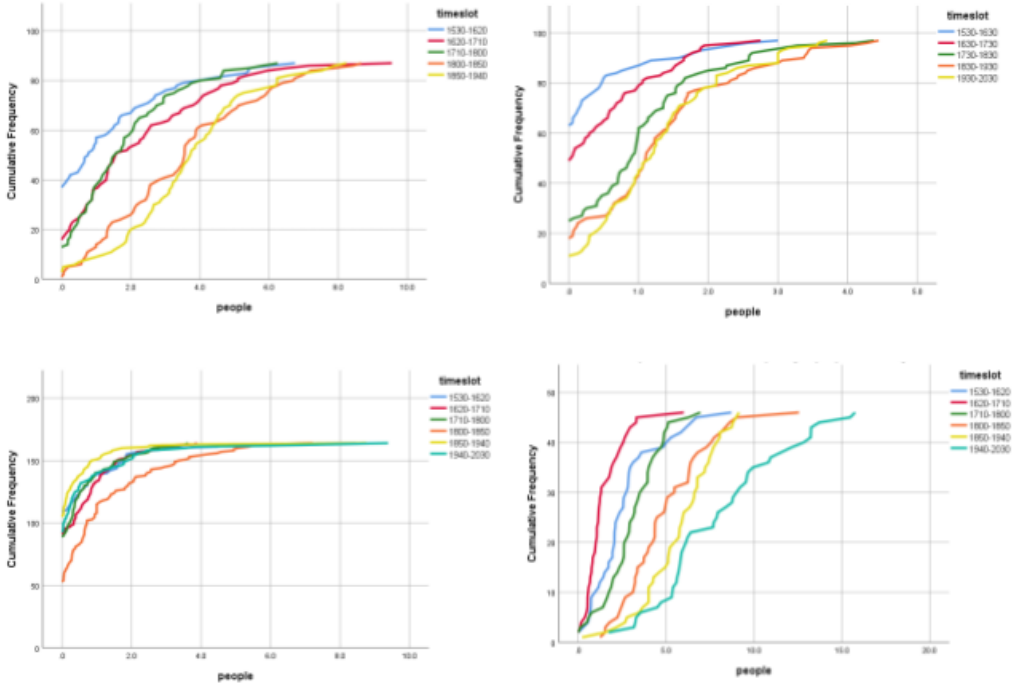

Figure 15. The cumulative frequency of people along 4 different survey segments over consecutive timeslots. Source: authors.

We can also compare the cumulative frequencies of different survey segments (Figure 15) to observe any commonalities in usage patterns. This is useful in profiling the usage characteristics of urban spaces which can help to inform its planning and design.

5. DISCUSSION

Our demonstration has shown that video survey and detection with an action detection model is a viable process to collect data that describes the geographic distribution of people with a relatively high degree of resolution (Figure 13). The use of our proposed process also reduces the amount of time that a surveyor needs to spend on site. Conventional activity observations are time consuming in that a human observer needs to observe and process what they have observed on site. Our proposal decouples data collection and processing (Figure 2). This allows us to relegate the tedious work of processing the observations offsite via an automated pipeline while enabling the surveyor to focus solely on data collection.

In addition, the simplicity of the data collection process reduces the level training that is needed to be provided to the surveyor. The surveyor does not need to be trained to interview participants or to use any questionnaire or observation templates. This further reduces the margin of human error and the overall workload. However, despite the promising results there are imitations to our proposed process.

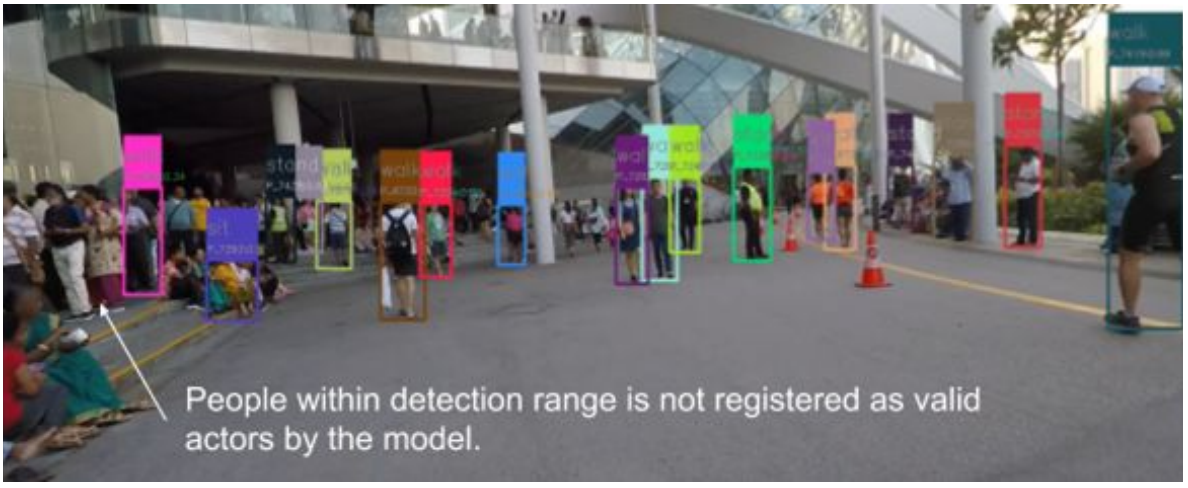

Figure 16. Detection visualization of large crowds. Source: authors.

Large crowds pose a problem for the detention model. Since the camera was positioned close to eye level, the constant movements and shuffling of people within crowds made it difficult for the model to track individual actors. The same person in a scene may be treated as a new unique actor in the scene and counted twice in the code. This generates detection noise within the dataset. Human silhouettes can also be obscured by other people in the crowd such that the model does not recognise it as a valid actor (Figure 16). While this is an inherent issue of the model, it may not be a significant problem from the urban design perspective.

Although we are unable to obtain the exact number of people in the scene, the detection results still serve as a useful proxy to describe the relative number of people in a location. It is also a closer representation of how individuals perceive the environment at eye level, where the perception of crowdedness is tied to the overall dispersion of people and not the exact number of people in a space.

Figure 17. Incorrect actions assigned to actors. Source: authors.

The overall effectiveness of the action detection model in correctly recognizing an action is mixed. The model was consistent in correctly identifying the categories "sit", "stand" and "walk". However, it had trouble consistently identifying "run/jog" and "ride". The model has a tendency to incorrectly classify these detections under one of the other 3 categories (Figure 17). "run/jog" tends to be classified as "walk" and "ride" tends to be classified as "sit". Thus, we cannot reliably produce an accurate tally of the 5 action labels (refer to 4.2.1).

The exact reason behind why the action detection model did poorly in detecting "run/jog" and "ride" remains to be determined. Preliminary observations suggest that it may be attributed to the asymmetrical distribution of training examples[5] within the AVA dataset[6] where training examples for "stand", "sit" and "walk" far exceeds that for "ride" and "run/jog". Future implementation with more comprehensive datasets could potentially improve the model's performance. A direction for future research could involve training the model to recognise activities with contextually appropriate data collected from the onsite surveys.

Active	Static
walk	stand
ride	sit
run/jog	

Figure 18. Grouping the action labels under the broader categories of "Active" and "Static".

While we cannot yet remedy the accuracy of the action detection model, we can still utilize the results in urban analysis by semantically grouping the 5 action labels into 2 broader categories "Active" and "Static" (Figure 18). "Active" can be used as a proxy for counting people in transit while "Static" can be used to count people staying in place. Evaluating the results through these 2 categories still offer valuable insights into understanding into crowd behaviours and use at public spaces.

[5] It is well understood that the model's performance is dependent on the dataset that was assigned for its training.
[6] The AVA dataset. https://research.google.com/ava/explore.html.

6. Conclusion

In summary, we have demonstrated that video surveys with Computer Vision can be a viable means for activity observations. The process can be implemented with an action detection model and generate sample headcounts with some degree of activity recognition. While the action detection model has not yet achieved the level of detail and accuracy of a human surveyor, it has drastically reduced the labor needed to survey large areas, by enabling the processing of large amounts of data through automation. Its current limitation in detection accuracy will be resolved in time. As it is, given the amount of useful data that can already be collected, we believe that the process merits further study and refinement. Future research can build on this proof of concept and continue to develop more general ways to explore the possibilities of using Computer Vision in activity observations.

7. References

[1] Lynch, K. (1960). *The image of the City*: The MIT Press.

[2] Frank, L. D., & Engelke, P. O. (2001). The Built Environment and Human Activity Patterns: Exploring the Impacts of Urban Form on Public Health. *Journal of Planning Literature*.

[3] Montgomery, J. (1995). Editorial Urban Vitality and the Culture of Cities. *Planning Practice & Research*.

[4] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature, 521*(7553), 436-444.

[5] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *Imagenet classification with deep convolutional neural networks.* Paper presented at the Advances in neural information processing systems.

[6] Saba, L., Biswas, M., Kuppili, V., Godia, E. C., Suri, H. S., Edla, D. R., . . . Kitas, G. D. (2019). The present and future of deep learning in radiology. *European Journal of Radiology, Volume 114*.

[7] Staar, B., Lütjen, M., & Freitag, M. (2019). Anomaly detection with convolutional neural networks for industrial surface inspection. *Procedia CIRP, Volume 79*.

[8] Zhang, F., Zhou, B., Liu, L., Liu, Y., Fung, H. H., Lin, H., & Ratti, C. (2018). Measuring human perceptions of a large-scale urban region using machine learning. *Landscape and Urban Planning, Volume 180*.

[9] Wang, R., Liu, Y., Lu, Y., Zhang, J., Liu, P., Yao, Y., & Grekousis, G. (2019). Perceptions of built environment and health outcomes for older Chinese in Beijing: A big data approach with street view images and deep learning technique. *Computers, Environment and Urban Systems, Volume 78*.

[10] Li, X., & Ratti, C. (2018). Mapping the spatial distribution of shade provision of street trees in Boston. *Urban Forestry & Urban Greening*.

[11] Hidalgo, A. D. N. a. P. R. A. (2016). Deep Learning the City: Quantifying Urban Perception at a Global Scale. *European Conference on Computer Vision*.

[12] Alhasoun, F., & González, M. (2019). *Urban Street Contexts Classification Using Convolutional Neural Networks and Streets Imagery.* Paper presented at the 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA).

[13] Wu, Z., & Zhou, X. (2018). *Detecting Street Signs in Cities Based on Object Recognition with Machine Leaning and GIS Spatial Analysis.* Paper presented at the Proceedings of the 1st ACM SIGSPATIAL Workshop on Advances on Resilient and Intelligent Cities.

[14] Gehl, J., & Svarre, B. (2013). *How to Study Public Life: Methods in Urban Design*: Island Press/Center for Resource Economics.

[15] Schneider, R. J. (2013). Measuring transportation at a human scale: An intercept survey approach to capture pedestrian activity. *Journal of Transport and Land Use. Volume 6, No.3*.

[16] Craig, C. L., Marshall, A. L., SJÖSTRÖM, M., Bauman, A. E., Booth, M. L., Ainsworth, B. E., . . . Oja, P. (2003). International Physical Activity Questionnaire: 12-Country Reliability and Validity. *Medicine & Science in Sports & Exercise*.

[17] Eleonora Papadimitriou, S. L., George Yannis. (2017). Human factors of pedestrian walking and crossing behaviour. *Transportation Research Procedia*.

[18] Tan, K. W. (2006). A greenway network for singapore. *Landscape and Urban Planning. Volume 76. Issues 1–4*.

[19] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., . . . Zitnick, C. L. (2014). *Microsoft coco: Common objects in context.* Paper presented at the European Conference on Computer Vision.

[20] Carreira, J., & Zisserman, A. (2017). *Quo vadis, action recognition? a new model and the kinetics dataset.* Paper presented at the proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[21] Ulutan, O., Rallapalli, S., Srivatsa, M., Torres, C., & Manjunath, B. (2020). *Actor conditioned attention maps for video action detection.* Paper presented at the The IEEE Winter Conference on Applications of Computer Vision.

[22] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). *Mobilenetv2: Inverted residuals and linear bottlenecks.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[23] Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster r-cnn: Towards real-time object detection with region proposal networks.* Paper presented at the Advances in neural information processing systems.

[24] Gu, C., Sun, C., Ross, D. A., Vondrick, C., Pantofaru, C., Li, Y., . . . Sukthankar, R. (2018). *Ava: A video dataset of spatio-temporally localized atomic visual actions.* Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.