Improving Traffic Safety Through Video Analysis in Jakarta, Indonesia

João Caldeira[*]
Department of Physics
University of Chicago
jcaldeira@uchicago.edu

Alex Fout[*]
Statistics
Colorado State University
alex.fout@colostate.edu

Aniket Kesari[*]
Jurisprudence & Social Policy
University of California, Berkeley
akesari@berkeley.edu

Raesetje Sefala[*]
Machine Learning
University of the Witwatersrand
raesetje.sefala@students.wits.ac.za

Joseph Walsh
Center for Data Science and Public Policy
University of Chicago

Katy Dupre
Center for Data Science and Public Policy
University of Chicago

Muhammad Rizal Khaefi
Pulse Lab Jakarta
muhammad.khaefi@un.or.id

Setiaji
Jakarta Smart City
setiaji@jakarta.go.id

George Hodge
Pulse Lab Jakarta
george.hodge@un.or.id

Zakiya Aryana Pramestri
Pulse Lab Jakarta

Muhammad Adib Imtiyazi
Jakarta Smart City

Abstract

This project presents the results of a partnership with Jakarta Smart City (JSC) and United Nations Global Pulse Jakarta (PLJ) to create a video analysis pipeline for the purpose of improving traffic safety in Jakarta. The pipeline transforms raw traffic video footage into databases. By analyzing these patterns, the city of Jakarta will better understand how human behavior and built infrastructure contribute to traffic challenges and safety risks. The results of this work should also be broadly applicable to smart city initiatives around the globe as they improve urban planning and sustainability.

1 Introduction

The World Health Organization's *Global status report on road safety 2015* estimates that over 1.2 million people die each year in traffic accidents [1]. Nearly 2000 such fatalities occur annually in the city of Jakarta, Indonesia. Many of these deaths are preventable through effective city planning. Jakarta has experienced rapid population growth over the last 50 years, from roughly two million people in 1970 to more than 10 million today. With this growth comes a rise in vehicle ownership and congestion, leading to an increase in the number of traffic incidents.

Global efforts to tackle this problem abound. The United Nations Sustainable Development Goals (SDG) specifically address the need for effective transit in SDG 11: "Make cities inclusive, safe,

[*] J. Caldeira, A. Fout, A. Kesari, and R. Sefala contributed equally to this work.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

resilient and sustainable." In articulating this goal, the UN specifically notes that 95% of urban expansion in the next decades will take place in the developing world. One of SDG 11's targets is to, "By 2030, provide access to safe, affordable, accessible and sustainable transport systems for all, improving road safety, notably by expanding public transport, with special attention to the needs of those in vulnerable situations, women, children, persons with disabilities and older persons" [2].

One of the core problems with using machine learning and other data-driven techniques in traffic safety analysis is that it is difficult to collect high-quality data. In partnership with Jakarta Smart City (JSC)[2] and Pulse Lab Jakarta (PLJ)[3], a team of fellows at the Data Science for Social Good (DSSG) fellowship at the University of Chicago[4] was formed to tackle this problem. We developed a video analysis pipeline that furnishes JSC and PLJ with the ability to generate rich databases that contain massive amounts of information about traffic behaviors.

We want this project to provide a template for others who hope to successfully deploy machine learning and data driven systems in the developing world. Through intense cooperation between the fellowship team in Chicago and the project partners in Jakarta, we gleaned insights into how to effectively build a system that is likely to be used by a partner in the developing world. Specifically, we became attuned to the need for mapping technical solutions to social problems that are articulated by people working in the field, understanding cultural context and awareness, and creating a feasible deployment strategy. These lessons should be invaluable to the many researchers and data scientists who wish to partner with NGOs, governments, and other entities that are working to use machine learning in the developing world.

2 Methodology

2.1 Data

JSC provided approximately 700GB of 1024 by 768 pixel video footage taken from seven locations across Jakarta, chosen to represent varying geography, infrastructure, and traffic behavior. Starting from these videos, we were tasked with generating quantitative data that could be used for more standard traffic analysis. In order to evaluate our results, we needed to obtain annotated videos. This was done by hand-labeling vehicles and pedestrians in a sample of our videos using the *Computer Vision Annotation Tool* (CVAT) [3].

2.2 Translating Data to Methods

Before translating raw video into structured data, extensive work had to be done so that all partners had a common vision of the policy interventions that the Jakarta authorities hoped to deploy given better traffic information. We established that in the medium term, Jakarta was interested in learning the best places that it can place "traffic stewards" and build traffic lights. In the long term, it is interested in learning where bigger infrastructure projects may be most successful. In addition to these specific interventions, we also set out to define the scope of problematic traffic behaviors that the city hopes to curtail. In this case, we are concerned with a few specific behaviors, including vehicles driving against traffic, motorcycles and scooters driving on pedestrian surfaces like sidewalks, and illegal stopping or parking. Once we understood the most dangerous driving behaviors, and the policy levers available, we were able to think about how to map social policy problems to technical solutions. This map informed the specific data that we generated. We detail our particular choice of computer vision methods in Section 3.2.

3 Results

3.1 Pipeline

Our pipeline was created with a "streaming" approach, which breaks a video into individual frames at the beginning of the pipeline, then passes these frames through a system of workers and queues.

[2] `https://smartcity.jakarta.go.id/`

[3] `http://pulselabjakarta.org/`

[4] `https://dssg.uchicago.edu/`

Each worker is given a particular "task" (e.g. object detection) that it performs on each frame. It then sends that frame to the next queue, where the frame waits until the next worker is ready. Frame order is preserved, and at the end, a worker puts frames back together to output the original video, any new annotations or analysis, and quantitative information that can be loaded into a database.

The pipeline is modular, which is a key feature that allows a user to optimize its performance. This also avoids loading large uncompressed videos into memory as implied by a batched approach, permits simultaneous execution of multiple tasks, and permits load balancing by adding or removing workers from tasks as necessary. There are some limitations to this decision, namely that GPU computations utilized by many machine learning algorithms are optimized for batch computation, and workers cannot use yet-unseen frames when performing a task, which limits the exploitation of temporal dependence between frames. We note that both of these concerns can be addressed through use of appropriate buffering in the workers.

3.2 Detection, Motion, and Segmentation

We developed several modules that make up the pipeline, and are directly related to Jakarta's specific policy requirements. Detection and classification are necessary components as they determine what objects in a frame are motor vehicles. The results of detection and classification provide the foundation for detecting specific traffic behaviors. We use YOLOv3 trained on the COCO dataset [4].

Motion estimation is similarly important because it helps determine when a vehicle is traveling the wrong way. We used optical flow as it allowed us to extract the information about whether an object was moving and in what direction. Using Lucas-Kanade optical flow algorithm [5] in conjunction with Shi-Tomasi corner detection [6], we were able to calculate the direction of movement for every detected vehicle in a frame. We used the existing implementation available in OpenCV [7].

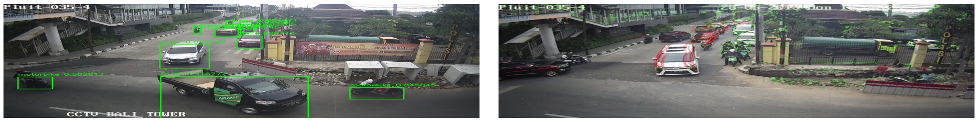


Figure 1: On the left, detection/classification with YOLOv3. On the right, motion detection with Lucas-Kanade Optical Flow.

Finally, we needed to classify the different regions of the image into different classes, such as road or sidewalk, in order to determine whether motor vehicles were moving in an illegal way. For this task, one can use semantic segmentation, which classifies each pixel as belonging to one of several different classes. We used a pretrained version of the WideResNet-38 model described in [8]. An example result can be seen in Figure 2.

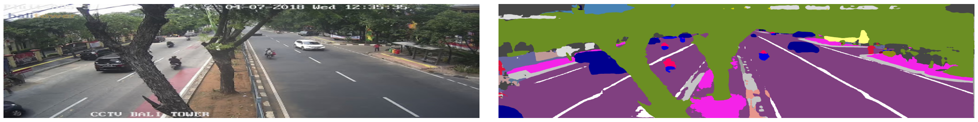


Figure 2: A scene pre- and post-segmentation.

Combining these methods, we can answer questions such as, "Is this vehicle traveling on the wrong side of the road?" or "Is this motorcycle illegally parked on a sidewalk?" Figure 3 shows one example of this. In this case, our system flagged four instances of a car moving in the wrong lane within a three-day span. In fact, three of these instances occurred in the same 2-hour period. One can imagine the utility such a system could provide, as an analyst can quickly identify that this intersection sees problematic behavior at particular days and times. This insight can then be used to inform interventions such as building a traffic light or median, or deploying a traffic steward at a busy time of day.

4 Evaluation

For detection, we measure precision and recall. In this case, recall is the proportion of objects of interest which are correctly identified as objects, regardless of the predicted class. Precision is the

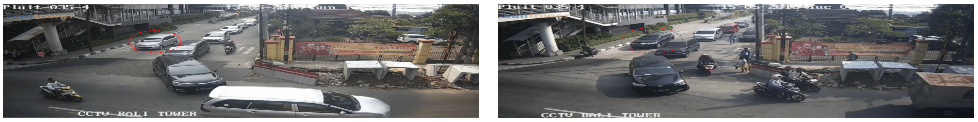
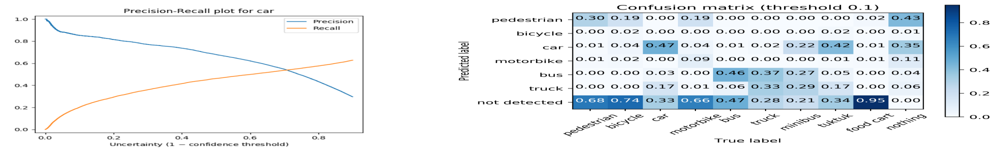


Figure 3: Examples of driving on the wrong side of the road found by our pipeline.

proportion of detections which are true objects of interest. We go through all boxes predicted by our model in decreasing order of confidence. The model's boxes typically will not align perfectly with human drawn boxes. Therefore, we used an "Intersection Over Union" (IOU) approach to determine whether two boxes were the same. If the IOU between the predicted and a true box are above our chosen threshold, we take those boxes to refer to the same object. Then we check if the predicted class is the same as the true class.

We point out that some class confusion may be immaterial to Jakarta's ultimate intervention decisions. For example, consider an intersection where illegal left turns can pose a risk to opposing traffic. City planners would benefit from knowing whether large, heavy vehicles are making such illegal left turns, but it may be less important to distinguish between buses and trucks, as both pose comparable risks. However, confusing a motorbike for a pedestrian may contribute to a misconceived understanding of ground truth. In our own evaluation, we noticed that tuk-tuks and mini-buses (which are Jakarta-specific and not in the YOLOv3 training set) were generally correctly characterized as something close to a car, but motorcycles and bicycles were frequently confused. Jakarta will experiment with more context-specific models in the near future.



	pedestrian	bicycle	car	motorbike	bus	truck	minibus	tuktuk	food cart	nothing
pedestrian	0.30	0.19	0.00	0.19	0.00	0.00	0.00	0.00	0.02	0.43
bicycle	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.01
car	0.01	0.04	0.47	0.04	0.01	0.02	0.22	0.42	0.01	0.35
motorbike	0.01	0.02	0.00	0.09	0.00	0.00	0.00	0.01	0.01	0.11
bus	0.00	0.00	0.03	0.00	0.46	0.37	0.27	0.05	0.00	0.04
truck	0.00	0.00	0.17	0.01	0.06	0.33	0.29	0.17	0.00	0.06
not detected	0.68	0.74	0.33	0.66	0.47	0.28	0.21	0.34	0.95	0.00

(a) Example Precision-Recall plot for car detection and classification throughout all annotated videos, with IOU threshold set to 0.25. Note that with a confidence threshold of 0.5, we label 50% of the cars in our labeled videos correctly, with 70% precision.

(b) Confusion matrix, normalized so columns sum to 1. The objectness threshold is 0.4, while the label threshold is 0.1. One important finding is that while our algorithm does not have labels specific to Jakarta such as tuk-tuks and minibuses, we detect them roughly as well as cars or trucks, labeling them as cars, buses or trucks.

Figure 4: Evaluation of object detection and classification.

We also evaluated motion detection. For optimal settings, the average angle between detected and true motion is 11.0^{o}. We can also use motion detection to effectively find vehicles moving in the wrong direction in a particular road, as we show in Figure 3.

5 Conclusion and Next Steps

Starting in 2017, JSC has been building a big data infrastructure, and is looking to integrate the pipeline into its existing systems. The first phase will identify several roads in Jakarta that represent two categories: problematic and safe. After classifying roads, JSC will deploy the system on a sample of the CCTVs that monitor problematic roads, and then record the output (e.g. the number of detected cars, motorcycles, etc.). JSC will then validate and tune the classification, motion, and segmentation models. JSC, together with Jakarta Transport Authority, will then gather and interpret results, and then formulate and implement interventions on problematic roads.

In the second phase, JSC will connect all of its CCTVs to the system. Once the models output information correctly and seamlessly, JSC will build information systems that support reports and/or a dashboard that will help various agencies in the Government of Jakarta understand model outputs, and hopefully improve decision making.

We hope our work illuminates the promise of using data to improve urban life around the globe. The code developed for this project is available on GitHub and we hope it proves valuable to anyone who wishes to develop or deploy a similar system and methods.

References

[1] World Health Organization. Global status report on road safety 2015. `https://www.who.int/violence_injury_prevention/road_safety_status/2015/en/`, 2015. Accessed: 2018-10-02.

[2] United Nations. Goal 11: Make Cities Inclusive, Safe, Resilient, and Sustainable, 2016.

[3] OpenCV. Computer vision annotation tool (CVAT). `https://github.com/opencv/cvat`, 2018. Accessed: 2018-11-20.

[4] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[5] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 1981 DARPA Image Understanding Workshop*, pages 121–130, 1981.

[6] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[7] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[8] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated BatchNorm for memory-optimized training of DNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.