Proceedings of the 2nd International Conference on Civil Engineering Fundamentals and Applications (ICCEFA'21)
~~Seoul, South Korea~~ Virtual Conference - November 21 - 23, 2021
Paper No. 126
DOI: 10.11159/iccefa21.126

Application of Tiny YOLOv2 Car Detection for Traffic Flow Quantification during the COVID Health Pandemic: Preliminary Results

Linnette M. Martinez[1], Joseph P. Salisbury[2], and Nicholas V. Scott[2]
[1]New York University, Tandon School of Engineering, Polytechnic Institute
6 MetroTech Center, Brooklyn, NY, 11201, USA
lm4432@nyu.edu; jsalisbury@riversideresearch.org; nscott@riversideresearch.org
[2]Riverside Research, Open Innovation Center
2640 Hibiscus Way, Beavercreek, Ohio, 45431, USA

Extended Abstract

Quarantine procedures enacted at the state and local levels due to the COVID health pandemic have contributed to significant modulation of vehicular traffic flow impacting such issues as road maintenance, vehicular density, and insurance costs. With no apparent cessation of these changes at the present time, there is a need for fast, robust, and efficient ways to quantify the flux of vehicles, in particular cars, on the road which allows for insight into how to address the placement of limited resources for important traffic engineering issues. To investigate whether existing neural network models can be used to rapidly quantify vehicular presence, the Tiny YOLOv2 object detection model [1,2] was applied to traffic imagery acquired from a semi-busy road using off-the-shelf panchromatic camera technology. If successful, the Tiny YOLOv2 model, a nine-layer convolutional neural network [3] trained on the Pascal VOC dataset [4] to classify twenty different object types, would enable vehicular detection and quantification from images or video in real-time.

A Brinno TLC300PRO RGB camera, mounted on a small tripod within an apartment directed outside a window, was configured to capture AVI files of traffic flow on a road in Beavercreek, Ohio, USA from April 21, 2020 to May 12, 2020. Images sampled at a frequency of 2 images per minute were extracted over a 9-hour interval on the first day from 10 am to 7 pm Eastern Standard Time. To perform car detection and quantification, images were imported into the gaming engine Unity (2019.4.13f1), where Tiny YOLOv2 was implemented to detect objects in images using the Barracuda neural network interface library [5,6]. An initial set of car images taken from a close distance was used to verify that the Tiny YOLOv2 classifier was functioning as intended. After achieving successful verification, the acquired dataset was analyzed using an automated process that detected and counted the number of images per hour that contained at least one car. As the dataset included images that were taken from a substantial distance away (approximately 150 meters from the road) and included parked cars that were not pertinent to the research objective, several image pre-processing steps were necessary to improve both sensitivity and specificity.

To improve car detection sensitivity, image contrast, hue, and saturation were modulated to improve the appearance (sharpness) of cars in the dataset. This was initially accomplished using an independent software package, with a subset of images being pre-processed and then imported into Unity for object detection with the Tiny YOLOv2 model. After this approach was shown to be successful in improving car detection sensitivity, a series of pre-processing steps was implemented directly within Unity using its post-processing capabilities [7]. To improve car detection specificity and limit car detection to traffic flux on the road, a uniform mask was applied to the image set to allow detection to occur only on the road. Image chips, which spanned the length of the road, were then fed into the Tiny YOLOv2 model car detector in 1-hour intervals over the 9-hour time span providing a probabilistic estimate of traffic flux.

A histogram of the number of cars detected per hour over the 9-hour time interval based on the car detection algorithm depicts small local maxima at two specific time intervals. These time intervals were associated with lunch time road inundation and the evening rush-hour time interval. This machine learning derived multimodal histogram structure has a similar maxima-minima structure to the histogram results obtained by counting the number of images containing at least one vehicle per hour via visual inspection. This

result suggests that the method of pre-processing of images and the use of the Tiny YOLOv2 car detector may be more robust than the use of eigenface decompositions [8, 9, 10].

While automated pre-processing was successful in improving the sensitivity and specificity of the approach, there remained several sources of error. Changes in the ambient light captured during data acquisition appeared to negatively impact the model's ability to detect cars. Based on preliminary results obtained using the external image processing tool, it is conjectured that further automation of the image processing steps within Unity before the application of the Tiny YOLOv2 car detector may further improve the sensitivity of the approach. Continued research is aimed at exploiting the overall method to estimate vehicular image density for other days and times throughout the full period of data acquisition. This would allow understanding of how the probability density function associated with vehicular density changes over a dynamical period where governmental rules seriously affected the mean number of vehicles on the road.

References

[1] J. C. Redmon, and A. Farhadi. (2017). YOLO9000: better, faster, stronger. Presented at Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. [Online], Available: https://arxiv.org/abs/1612.08242

[2] Tiny Yolo v2 – Github, (2020), MIT, Accessed: June 1, 2021. [Online], Available: https://github.com/onnx/models/tree/master/vision/object_detection_segmentation/tiny-yolov2

[3] U. Michelucci. Advanced Applied Deep Learning: Convolutional Neural Networks and Object Detection. California, USA: Apress, 2019.

[4] M. Everingham, L.V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol 88, pp. 303-338, 2010.

[5] Keijiro/TinyYOLOv2Barracuda: Tiny Yolov2 on Unity Barracuda. (2021). Github. Accessed: June 1, 2021. [Online]. Available: https:/github.com/keijiro/TinyYOLOv2Barracuda

[6] *Introduction to Barracuda: Barracuda: 1.0.4*. (2020). Accessed: June 1, 2021. [Online], Available: https://docs.unity3d.com/Packages/com.unity.barracuda@1.0/manual/index.html

[7] *Unity User Manual Documentation 2020.3*. (2020). Accessed: June 1, 2021. [Online], Available: https://docs.unity3d.com/Manual/PostProcessingOverview.html

[8] M. Turk, and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

[9] J. A. Lee, and M. Verleysen, *Nonlinear Dimensionality Reduction*. New York, USA: Springer, 2007.

[10] W. L. Martinez, and A. R. Martinez, *Computational Statistics Handbook with Matlab*. London, UK: Chapman and Hall/CRC, 2001.