Accuracy of Vehicle Counts by Public Object Recognition Model Using Existing CCTV Infrastructure

George Townsend

September 10, 2019

Abstract

We test one of the best performing publically available object detection models, YOLOv3, on it's ability to perform accurate vehicle counts on existing CCTV hardware in London. We find that the performance of the model is not sufficent to meet current industry standards for smart traffic monitoring and modelling, however the technology has been proven in proprietary environments with focused training datasets. The best performing tests show promise for the application of YOLOv3 to the task, especially given further research into the effects of camera angle on a scene and the economic cost of real time analysis.

Contents

List of Figures

List of Tables

1 Literature Review

1.1 Introduction

Road traffic tracking and prediction are vital considerations for an efficient, smoothly run transportation network. Modern sensors and communication networks allows large amounts of high resolution traffic data to be analysed in real time to create automated intelligent transportation systems. The volume of data collected by modern techniques presents an unreasonable challenge to traditional traffic engineering, necessitating cross-domain knowledge of machine learning to create next generation management systems. Effective application of machine learning (typically including linear regression models, ARMA, and MLF/RGB/ELman/SOM neural networks) is dependant on clean, organised data for training, restricting the pool of available data for models. Given the difficulty involved in identifying mistakes or inconsistencies in complex machine learning models, preemptively ensuring the accuracy of collected data is critically important for the design and development of intelligent transport systems.

1.2 Technologies

Many distinct technologies are used to collect traffic data in disparate conditions, and are primarily classified as either ‘point’ sensors or ‘probe’ sensors (roadside or attached to the vehicle respectively). The Minessota Department of Transportation [Lin et al., 2006] compiled a review of past WLT (wireless location technology) based traffic monitoring studies, and concluded that contemporary probe-based monitoring techniques, primarily NFC and GPS, produce data of moderate quality, with major potential for improvements. The more common roadside point sensors generally produce more reliable data and have been studied more thoroughly.

Pneumatic road tube counters are among the most common sensors due to their ease of deployment, with studies indicating that the absolute error of the counts average closer to 10% over a 15-minute interval, with positive and negative counting errors gradually cancelling each other out as the time interval increases. Daily count errors could be as low as 1%. Errors classifying vehicle speed and type were much greater and as such had limited application in statistical models.

[McGowen and Sanderson, 2011] Tests performed by the South Dakota Department of Transport on high traffic sections of freeway showed errors that totaled 15.8% lower than the baseline count.

Inductive loops are built into the road, and as such require foresight and greater capital investment from the operators of the road network. Count accuracy consistently lands above 95% during free flowing traffic, but notably decay to 70%-90% during congestion. Speed and classification were largely limited, especially during congestion. The technology generally has a short life expectancy and can be expensive to maintain. Various Inductive loop systems can provide modest performance improvements, at the cost of a more difficult installation. [Martin et al., 2003]

The Minnesota Department of Transport tested various implementations of active and passive infrared systems. Different systems produced vastly different results, the best performing show errors averaging below 1% with the worst at 10%. [Kranig et al., 1997]. Infrared systems are minimally affected by weather conditions, however only passive systems can provide data on lane occupancy and presence [Leduc, 2008].

Video image processing has existed for decades, but has only recently reached a level of reliability and accuracy required by distributed intelligent transport systems. Algorithmic approaches are fundamentally split into two types; detection based on the change in light level within a scene, and object identification and tracking. Under optimal daytime conditions, light level based systems generally produced more accurate vehicle counts while object detection approaches produced more accurate speed measurements. Count errors typically remained within 20% of a baseline under suitable conditions for both systems [Kastrinaki et al., 2003]. Modern research has focused on object detection approaches due to its scalability, ability to classify objects, and the ease of deployment. [Sang et al., 2018]

1.3 Video Based Vehicle Detection Methods

Vehicle detection is one of the most important applications of object detection in intelligent transport systems. Most modern models use a variant of neural network trained on a database of pre-catagorised images [Antsaklis, 1990]. Information including classification and vehicle condition can be derived from video streams, providing a substantial advantage over purely count based light level systems. While abundant information can be found on the performance of existing processes, the lower level technical details remain obscured and proprietary. The exact systems used by road network operators are not publically available and cannot be independently verified. Many public, generalised object detection models are trained on datasets containing vehicles and have been used in open source road traffic analysis programs. Further research is needed as to the performance of open, generalised models on video produced by existing infrastructure.

2 The Scientific Research Question

Can publically available object detection models provide accurate vehicle count data for use in existing intelligent transport management systems?

3 Hypothesis

H_0: Public object detection models can be used to perform accurate vehicle counts on existing camera infrastructure.

H_A: Public object detection models cannot accurately perform vehicle counts on existing camera infrastructure.

4 Methodology

Testing the accuracy of an object detection model required comparing a large set of automated counts against a known baseline, in this case a manual count. To ensure the video data is representative of existing camera installations, we collect data from the Transport for London (Tfl) CCTV data feeds available online at http://www.tfljamcams.net/. The website returns ten second snippets of video for a given request and updates every 5 minutes. The download and sorting process is automated by a Python script to allow for efficient scaling of data sizes. Downloads were performed in batches of 50 to create distinct databases of related video with similar environmental conditions, with each database covering a span of approximately four hours in the real world. Multiple cameras each with different viewing angles and lighting conditions were used to create databases, and there are a minimum of 3 databases created per camera. In total, 6 cameras were used to create 23 databases of 50 videos each, giving a total of 1 150 video clips with a runtime of 3 hours 20min. If a camera produced unexpected results (e.g. the camera was temporarily disabled, or the angle of the camera was changed by an operator) during the four hour span of creating a database, then that database is removed and not analysed. Only databases with fixed viewing angles and 50 usable, non-duplicated videos were deemed acceptable.

While many object recognition models were considered for their accuracy and speed, the final system used OpenCV for video processing, YOLOv3-416 [Redmon and Farhadi, 2016] weights, trained on the COCO dataset for object detection, and SORT [Bewley et al., 2016] for object tracking between frames, implemented in Python 3.7.0. Our analysis was performed on a 64-bit linux machine, running a stock i7-4790 with 8GB RAM. The object detection code was branched from https://github.com/guillelopez/python-traffic-counter-with-yolo-and-sort, with minor changes to remove detection boxes from the output video and to write the count to a file instead of onto the output video. A boundary line was manually defined for each database to cover a lane of traffic in one direction and used as the basis for automatic and manual counts.

The manual counts were performed by a selection of high school students in grades 7-11.
Students viewed each video, which contained the same boundary line superimposed over the source, and manually counted each vehicle that passed over it. Students were instructed to use 'number of drivers' as a rule of thumb for vehicle counts (e.g. a semi trailer with cars on the back only counts as one vehicle passing over the line). Each student was given two databases to review, and each database was reviewed by multiple (either 2 or 3) students to allow for cross checking of their observations. In the analysis of the results, the automatic count is compared against the mode value of the set of manual counts for each video. That is, if students give a conflicting series of answers for the same source video, the most common answer from the group is used. If each answer occurs an equal number of times, the lowest value is used. (e.g. video A manual counts [5, 4, 5] $-->$ 5, video B manual counts [3, 2] $-->$ 2).

5 Results

Database	Camera	Local Start Time	Manual Count	Automatic Count	Disparity (%)
database 1	640	07:00:45	676	346	195.38
database 2	640	13:04:04	570	285	200.00
database 3	640	01:05:08	412	28	1471.43
database 4	640	08:55:08	406	374	108.56
database 5	633	07:02:11	433	45	962.22
database 7	633	11:15:05	147	68	216.18
database 12	633	09:21:13	422	57	740.35
database 14	633	00:06:05	146	0	-
database 6	623	08:56:22	152	343	44.31
database 10	623	09:14:25	625	396	157.83
database 17	623	09:04:47	151	369	40.92
database 19	623	00:50:46	413	42	983.33
database 8	622	11:16:28	147	68	36.51
database 9	622	00:22:33	541	7	7728.57
database 11	622	09:17:21	87	161	54.04
database 15	622	00:07:22	475	3	1583.33
database 13	627	09:04:16	144	292	49.32
database 16	627	00:07:33	481	22	2186.36
database 21	627	01:23:38	473	57	829.82
database 22	627	09:07:02	172	221	116.20
database 18	628	18:06:38	545	92	592.39
database 20	628	00:50:57	550	74	743.24
database 23	628	09:07:02	172	221	77.83

Table 1: Manual and automatic count results

Database	Manual Std.	Auto Std.	t-stat	p value
database 1	3.822	3.11	9.375	<0.001
database 2	3.119	2.955	4.465	<0.001
database 3	1.954	0.828	8.112	<0.001
database 4	3.568	3.425	4.841	<0.001
database 5	3.48	1.36	13.788	<0.001
database 7	3.208	1.52	15.974	<0.001
database 12	2.889	1.058	15.881	<0.001
database 14	2.349	0.0	9.059	<0.001
database 6	3.181	2.623	7.029	<0.001
database 10	3.583	2.938	5.257	<0.001
database 17	4.038	3.187	4.681	<0.001
database 19	1.481	1.027	3.496	0.001
database 8	3.597	2.474	8.467	<0.001
database 9	2.459	0.4	9.272	<0.001
database 11	3.089	2.394	8.991	<0.001
database 15	1.725	0.237	11.576	<0.001
database 13	3.198	2.517	4.851	<0.001
database 16	1.475	0.697	6.008	<0.001
database 21	2.622	1.649	4.023	<0.001
database 22	3.158	2.942	3.796	<0.001
database 18	4.263	1.759	16.182	<0.001
database 20	2.036	1.676	3.716	<0.001
database 23	3.405	3.623	7.321	<0.001

Table 2: Standard deviation and t-test of databases

Database	Analysis Time
database 1	1:23:35
database 2	1:25:02
database 3	1:33:19
database 4	1:23:17
database 5	1:28:55
database 6	1:34:33
database 7	1:19:08
database 8	1:26:49
database 9	1:24:23
database 10	1:33:54
database 11	1:36:41
database 12	1:28:35
database 13	1:15:09
database 14	1:34:37
database 15	1:29:34
database 16	1:25:18
database 17	1:19:19
database 18	1:23:21
database 19	1:32:00
database 20	1:27:36
database 21	1:18:46
database 22	1:22:23
database 23	1:26:34

Table 3: Analysis time by database.

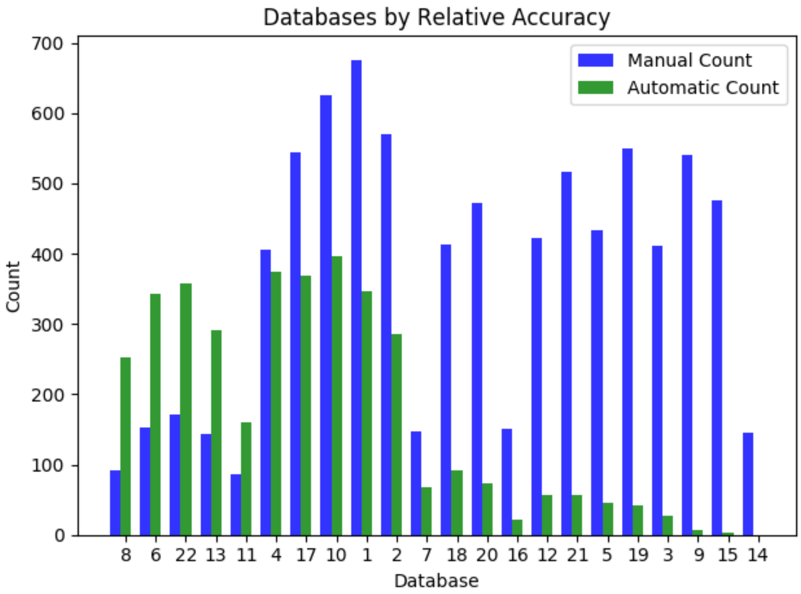

Figure 1: Automatic and manual counts from all databases, sorted by relative accuracy of automatic count.

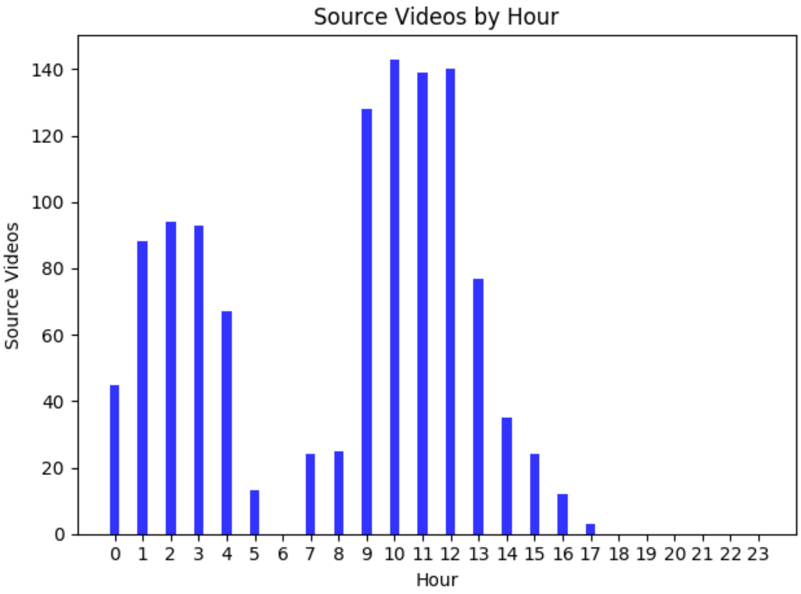

Figure 2: Distribution of the local creation time of source videos.

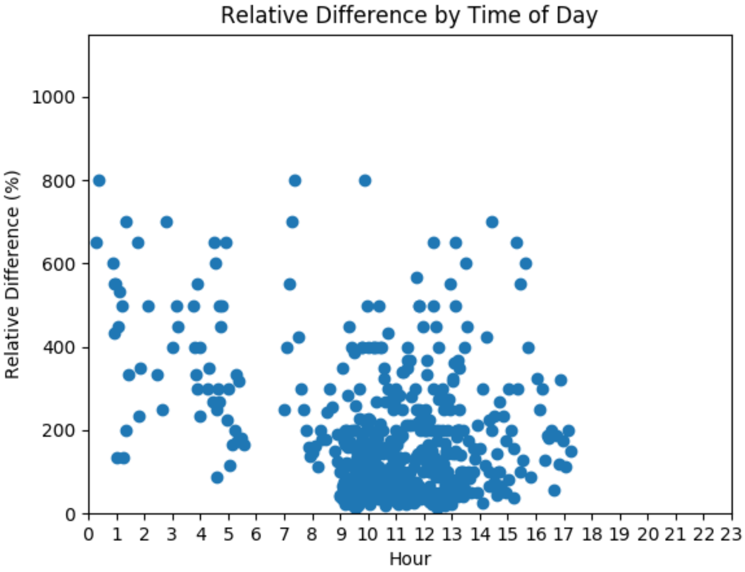

Figure 3: Relative accuracy of each video analysed by creation time.

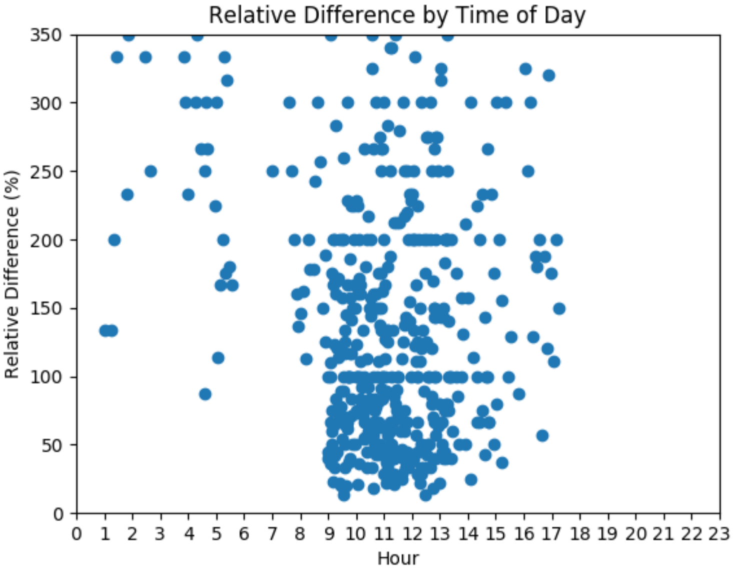

Figure 4: Small scale view of relative accuracy over creation time.

6 Discussion

As seen in table (1), only a single database's automatic count fell within 10%, two within 20%, and three within 30% of the baseline count. The vast majority (18/23) of databases had automatic counts with greater than 50% disparity from the baseline. Figure (1) shows the majority (18/23) of databases resulted in the YOLOv3 model (often substantially) undercounting, with the rest of the results showing overcounts. These results show that YOLOv3 cannot be immediately generalised to vehicle traffic counts on existing infrastructure, forcing us to reject the null hypothesis and accept the alternative. Many results are accurate to within reasonable margins of error, allowing us to determine the conditions that most benefit accurate results.

Figure (2) shows the distribution of creation time for each of the 1150 source videos. It highlights that the majority of the samples were taken in daylight hours. This means that the overall rejection of the null hypothesis cannot be solely attributed to poor lighting conditions. Figure (3) plots the relative accuracy of each source video's automatic count against the time of day (implying lighting conditions). The plot does not include tests where the object recognition model failed to detect a single vehicle. The largest clustering appears between 0900 and 1400 hours, between 0% and 200% relative accuracy. This clustering, expanded on figure (4), contains the majority of points on the plot and is much more accurate than the average accuracy of each database, suggesting that outliers could be an element of the extremely large disparities observed. This is supported by the T-test results in table (2), showing that the precision of the automated counts differs substantially from the precision of the baseline counts. A smaller number of outliers appear above the large cluster, with dots becoming less dense as relative difference increases. Dots appear much less frequently in the hours 1400-1800 than the hours preceding, but this is explained by figure (2) and the lower number of source videos for this time period. The final section is between the hours 0000-0600, containing a relatively even spread between 50% and 800% disparity. The lower frequency of dots in this area cannot be explained by fewer source videos, but instead by many videos being excluded from the dataset for not meeting the ¿ 1 automatic count requirement.

An important observation to note is the horizontal lines that appear in figures (3) and (4)- these lines represent common fraction results of the relative disparity calculation, e.g. a video with a manual count of 8 and an automatic count of 2 produces a relative disparity of (8/ 2) * 100 = 400%. Smaller numbers of cars in a video produce less precise results that are more likely to fall along these lines. This also illustrates how relative accuracy can appear to be very imprecise for videos with small numbers of vehicles, and the reason why averages of databases must be taken for conclusions to be drawn.

Four examples of the angle of cameras relative to passing cars are given in figure (5), highlighting the range of camera angles observed throughout the databases. As the databases were not normalised for daylight hours, we cannot simply compare the results between cameras to determine which is the best angle of observation. This would require new databases be created with plans to ensure an even number of day/night videos for all cameras. Our detection process was powered by YOLO and the COCO dataset, which contains training images mostly from street level- meaning it is likely that camera angles closer to horizontal ground level will perform better than any alternatives. Further testing is required to prove this for individiual object detection models with different structures.

The runtime of each database analysis is shown in table (3). A database of 50x10sec clips (approx. 8 minutes runtime) takes on average 1hr 25mins to complete analysis on the host system, more than 10x slower than realtime. While this is objectively bad performance, it is unlikely to represent real world conditions, and is further mitigated by the uncontrolled environment in which the study was run- other, unrecorded processes could have taken CPU time during analysis. Existing object detection models used today are built and optimized for GPUs, presenting a far more cost effective compute solution, that cannot be compared with the consumer CPU used in our analysis. Source video could also be reduced in fps or resolution to increase model speed.

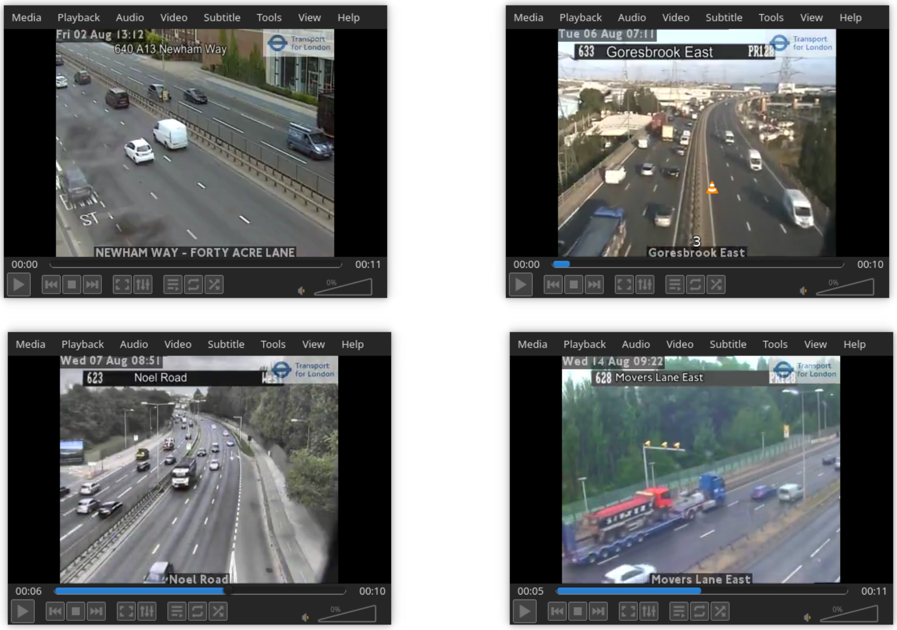

Figure 5: A selection of camera views highlighting different viewing angles.

7 Conclusion

The results of this experiment reject the null hypothesis, and accept the alternative-current public object detection models cannot accurately perform vehicles counts using existing camera infrastructure. While a small percentage of databases tested returned results accurate to within an accepted margin of error, most databases did not. Databases created during night hours performed much worse than those created during daylight hours. While we suspect that the angle of the camera relative to vehicles on the road will affect model performance, our databases were not organised in a way that these results could be easily identified. A future study could control for all other variables in order to quantify the effect of camera angle on performance. Further study is also needed to compare and contrast the cost and speed of various compute services e.g. Google Cloud, Microsoft Azure, self hosted etc. Accurate vehicle tracking using public and open object detection models on existing infrastructure is currently difficult to achieve unless environmental conditions and camera placement are ideal, however there exists huge untapped potential as evident by better performing proprietary models in use today.

References

[Antsaklis, 1990] Antsaklis, P. J. (1990). Neural networks for control systems. *IEEE Transactions on Neural Networks*, 1(2):242–244.

[Bewley et al., 2016] Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016). Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468.

[Kastrinaki et al., 2003] Kastrinaki, V., Zervakis, M., and Kalaitzakis, K. (2003). A survey of video processing techniques for traffic applications. *Image and Vision Computing*, 21:359–381.

[Kranig et al., 1997] Kranig, J., Minge, E., and Jones, C. (1997). Field test of monitoring of urban vehicle operations using non-instrusive technologies. Technical report.

[Leduc, 2008] Leduc, G. (2008). Road traffic data: Collection methods and applications.

[Lin et al., 2006] Lin, S.-P., Chen, Y.-H., and Wu, B.-F. (2006). A real-time multiple-vehicle detection and tracking system with prior occlusion detection and resolution, and prior queue detection and resolution. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 828–831. IEEE.

[Martin et al., 2003] Martin, P. T., Feng, Y., Wang, X., et al. (2003). Detector technology evaluation. Technical report, Mountain-Plains Consortium Fargo, ND.

[McGowen and Sanderson, 2011] McGowen, P. T. and Sanderson, M. (2011). Accuracy of pneumatic road tube counters.

[Redmon and Farhadi, 2016] Redmon, J. and Farhadi, A. (2016). Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*.

[Sang et al., 2018] Sang, J., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q., and Cai, B. (2018). An improved yolov2 for vehicle detection. *Sensors*, 18:4272.