
Divide and Conquer: an Accurate Machine Learning Algorithm to Process Split Videos on a Parallel Processing Infrastructure

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Every day the number of traffic cameras in cities rapidly increase and huge amount
2 of video data are generated. Parallel processing infrastructure, such as Hadoop,
3 and programming models, such as MapReduce, are being used to promptly process
4 that amount of data. The common approach for video processing by using Hadoop
5 MapReduce is to process an entire video on only one node, however, in order to
6 avoid parallelization problems, such as load imbalance, we propose to process
7 videos by splitting it into equal parts and processing each resulting chunk on a
8 different node. We used some machine learning techniques to detect and track the
9 vehicles. However, video division may produce inaccurate results. To solve this
10 problem we proposed a heuristic algorithm to avoid process a vehicle in more than
11 one chunk.

12 1 Research problem and motivation

13 According to [1], by 2020 one billion cameras will be installed throughout cities. More cameras
14 implicate a generation of huge amount of video data. Fortunately, parallel processing ecosystem
15 based on distributed environments have permitted to face computational complexity and big datasets.
16 Few intelligent traffic management systems (ITMS) approaches have been developed using parallel
17 processing infrastructure. Some of those solutions have used the Hadoop MapReduce ecosystem
18 which allows parallelization of multiple jobs by using several computers or multiple processors in
19 one computer[2].

20 In ITMS, researches have used Hadoop MapReduce to sequentially analyze multiple videos by using
21 batch processing, where each entire video file is only processed by a unique node [3][4]. In this
22 case, when the videos are reasonably long, it can arise a problem known as load imbalance. On
23 the other hand, this way of processing videos only allows to take the maximum advantage of the
24 parallel ecosystem when the dataset contains a big number of videos [5]. To avoid these problems we
25 have divided each video to process them by chunks in different nodes. Unfortunately, this approach
26 brings some problems. For example, if we divide a video, we could find a vehicle in different video
27 chunks, possibly affecting the precision of the analysis algorithm. For the case of a counting vehicles
28 system, classical algorithms would detect more vehicles in video chunks that when an unsplit video
29 is analyzed.

30 Motivated by the idea of processing chunk videos within a parallel processing ecosystem without
31 obtaining wrong results, we propose an intelligent video processing algorithm to process the chunks
32 and avoiding to take the same vehicle into account many times. In the framework we propose, it does
33 not matter the number and size of chunks in which the video is divided. Machine learning techniques
34 are used to process videos and detect vehicles. The considerations above motivate the formulation of

the following research question: How to process chunks of video using Hadoop MapReduce in order to decrease the computational time and reach accurate results?

2 Technical contribution

Our proposed algorithm takes as input datasets of videos from traffic cameras. Some of the videos we used were taken from YouTube and other videos from traffic cameras of Cali (Colombia) city. We developed a software platform that allows a user to select a street from a video, where the vehicle counting will be performed. By drawing lines, the user is able to specify the target counting area. This configuration is saved in a dataset to be used in futures videos. In the master node, a partitioning algorithm divides the video in chunks, taking into account the number of data nodes. Each chunk video is sent to a Hadoop data node, where 3 algorithms are applied: first, we detect the vehicles in the chunk by using the deep learning algorithm YOLO; then, we track each vehicle detected by using kalman filters and the hungarian algorithm; and finally, we filter the vehicles that were registered in others chunk videos. The videos are analyzed and the result is saved in a database than can be continuously accessed for probabilistic analysis in vehicular traffic tasks (see Figure 1).

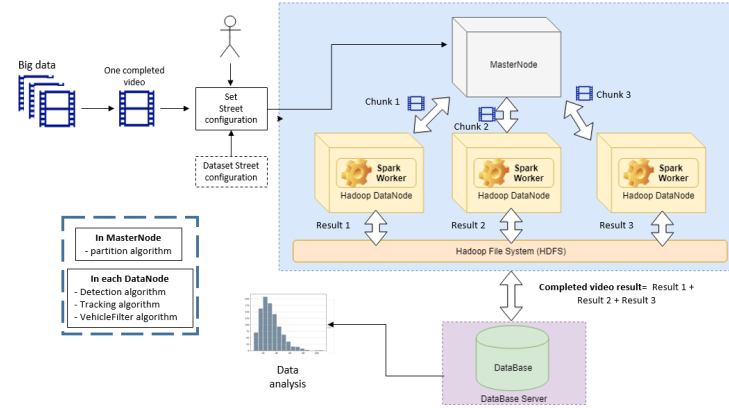


Figure 1: Framework proposed.

The filtering algorithm uses the initial street configuration (see figure 2-a) to define what vehicles to track, avoiding to follow the vehicles out of the select street and the vehicles that are part of other chunks. To identify the vehicles that are in more than one chunk we consider the direction of the vehicle and the street. Then, we examine if the vehicle was detected before and after the red line or if the vehicle only was detected after the line some few frames ago. If one of these conditions is satisfied, the vehicle is considered into the chunk, in any other case it is filtered using our heuristic algorithm. In the experiment of the figure 2, we counted the number of vehicles that pass by a specific place (red line) using two different ways: in a first approach we processed the whole video in only one Hadoop DadaNode (see figure 2-b), a second experiment consisted on dividing the video into 4 chunks (see figures 2-c to 2-f) and processing each of them on a different data node. We clearly show that the final result (i.e., the vehicle frequency count) for both experiments was the same, revealing the accuracy of our approach.

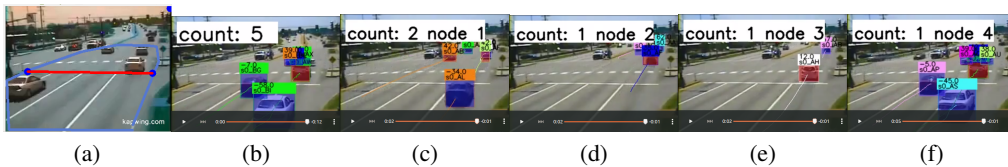


Figure 2: Experiment results. (a) Street configuration; (b) entire video processed; (c),(d),(e) and (f) video processed by chunks

61 **References**

- 62 [1] Wei, Y., Song, N., Ke, L., Chang, M. C. & Lyu, S. (2017) Street object detection/tracking for AI city traffic
63 analysis. *In 2017 IEEE SmartWorld, Ubiquitous Intelligence Computing Advanced Trusted Computed, Scalable*
64 *Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation*
65 *(SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI) (pp. 1-5). IEEE.*
- 66 [2] Triguero, I., Figueredo, G. P., Mesgarpour, M., Garibaldi, J. M. & John, R. I. (2017). Vehicle incident hot
67 spots identification: An approach for big data. *In 2017 IEEE Trustcom/BigDataSE/ICSS (pp. 901-908). IEEE.*
- 68 [3] Rathore, M. M., Son, H., Ahmad, A. & Paul, A. (2018) Real-time video processing for traffic control in
69 smart city using Hadoop ecosystem with GPUs, *Soft Computing* **22**(5): 1533-1544
- 70 [4] Natarajan, V. A., Jothilakshmi, S. & Gudivada, V. N. (2015) Scalable traffic video analytics using hadoop
71 MapReduce, *ALLDATA* p 18.
- 72 [5] Yoon, I., Yi, S., Oh, C., Jung, H. & Yi, Y. (2018) Distributed Video Decoding on Hadoop, *IEICE TRANSAC-*
73 *TIONS on Information and Systems* **101**(12):2933-2941.