International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 12 (2019), pp. 2550-2558
© International Research Publication House. http://www.irphouse.com

Video-Based Vehicle Counting System for Urban Roads in Nigeria Using Yolo and DCF-CSR Algorithms

Aderonke A. Oni

Department of Computer and Information Science, Covenant University, Ota, Nigeria.

Nicholas Kajoh

Department of Computer and Information Science, Covenant University, Ota, Nigeria.

ORCID: 0000-0002-8625-2748

Abstract

This study improves the traffic situation on, and condition of Nigerian roads by implementing a vehicle counting system that provides accurate data for traffic control agencies and systems. After comparing different detection and tracking algorithms, You Only Look Once and Discriminative Correlation Filter with Channel and Spatial Reliability were chosen as detection and tracking algorithms respectively for the system. The system was implemented using Python programming language and OpenCV. The significances of this system include estimating traffic flow on a given road per time, predicting future traffic conditions, understanding traffic patterns and the factors that affect them, and optimizing existing manual traffic management systems.

Keywords: Vehicle counting, detection, tracking, YOLO, DCF-CSR

1. INTRODUCTION

Cities and traffic congestion have developed shoulder to shoulder since the earliest large human settlements. Traffic congestion has continued to be one of the most persistent problems facing road users and urban planners across the globe [1]. Global Traffic Volume (GTV) is estimated to double between 1990 and 2020 and again by the year 2050 [2]. This order of growth is indicative of what the future of traffic congestion would mean for people living and working in urban areas. Throughout the world in both large and small cities, traffic congestion is getting worse. The result of this traffic congestion is the increased cost of living and loss of time for other activities.

Several cities across Nigeria are experiencing severe traffic congestion as their vehicle populations are becoming increasingly large. It is imperative that the Government (Local, State and/or Federal) should develop new systems to address the challenges affecting transit in urban areas. The vehicle counting system is a very important part of any efforts to manage vehicular traffic effectively as traffic flow data is needed before important decisions can be made. Vehicle counting systems are designed to collect information (tally, and in some cases categorical data) about the movement of traffic.

Several ways to count vehicles driving along a given road exist such as using infrared detectors, inductive-loop detectors or radar detectors [3]. While these solutions work for common use cases, they are not sufficiently robust to work well in areas with poor weather conditions, electromagnetic interference and traffic jams—where feature-based detection methods are more accurate [4]. More so, these methods are very expensive to implement and maintain at scale.

In recent years, the use of video-based techniques in vehicle detection and counting has received increasing attention due to advancements in digital image processing and systems infrastructure. Video analysis techniques such as background subtraction [5] shadow elimination [6] and occlusion detection [7] have made it possible to detect moving objects, specifically vehicles on the road, with high levels of accuracy. Other techniques like the virtual loop [8] and virtual line detection [9] can be used to count the vehicles.

Traffic congestion is becoming very problematic in different parts of Nigeria, and indeed in many cities around the world. Conventional methods for traffic management are not scaling with the growth of vehicle populations in urban areas. There is an urgent need for an Intelligent Traffic System (ITS) of which the vehicle counting system is an important component. Such a vehicle counting system should be able to provide highly accurate data and work well in complex and unfavourable environments. It should also be relatively cheap to implement, integrate with existing traffic management infrastructure and easy to maintain.

In this study, the focus is on creating a vehicle counting system to be used on urban roads in Nigeria, specifically to be installed on pedestrian and overhead bridges. The goal is to study existing vehicle counting systems in Nigeria and develop an improved and robust system by exploiting state-of-the-art Computer Vision techniques and algorithms for object detection and counting.

2. LITERATURE REVIEW

Vehicle counting systems are a vital component of any transportation management system as they provide traffic flow data which is crucial in understanding and improving vehicular traffic. Road transportation encompasses vehicles, users

© International Research Publication House. http://www.irphouse.com

(pedestrians, drivers and passengers), infrastructure, traffic management and interfaces with other modes of transportation.

A vehicle counting system records data (tally, and sometimes categorical data) about the flow of traffic. Vehicle count (also referred to as traffic count) is a count of vehicular traffic which is carried out along a particular road. It can be conducted manually by individuals who count and record traffic on a tally sheet or hand-held electronic device, or automatically by installing temporary or permanent traffic counting equipment.

Vehicle counts provide primary data used to calculate Annual Average Daily Traffic (AADT), a measure used mainly in transportation planning, transportation engineering and retail location selection. AADT is the total volume of vehicle traffic of a highway or road for a year divided by the number of days in that year. Vehicle counts are also used to identify the busiest routes in a road network in order to improve them or provide alternatives. The need for a system to record traffic count arose naturally as vehicle populations in towns and cities increased, and ways were sought to optimize road transport. The earliest forms of vehicle counting involved manually counting vehicles as they drove passed a given road using tally sheets or electronic counters. This quickly became unscalable as vehicular traffic skyrocketed and road networks expanded.

New techniques and technologies to compute traffic count have been developed over the years, each with its merits and demerits. These systems and their workings will be discussed in a subsequent section. There are several vehicle counting systems available today. Some of the more popular vehicle counting techniques used in today's vehicle counting systems include manual counts, pneumatic road tubes, piezoelectric sensors, inductive loops [10], magnetic sensors [11], acoustic detectors [12], passive infrared [13], doppler and radar microwave sensors, and video vehicle detectors.

2.1 Vehicle Counting Systems in Nigeria

The Lagos Metropolitan Area Transport Authority (LAMATA) is a government agency in Lagos State which was formed in 2002 to oversee the planning of transportation, implementation of associated policies and development of public transportation infrastructure in the city of Lagos. LAMATA performs vehicle counts regularly as part of their efforts to understand traffic flow in major corridors in the city of Lagos so that they can optimize the transportation systems as well as forecast traffic in order to know how best to plan for the future [14]. They almost exclusively do manual counts.

The Federal Road Safety Corps (FRSC) is a government agency saddled with the task of making Nigerian roads safe for motorists and pedestrians. The main responsibilities of the FRSC include making roads safe for all road users, ensuring that roads are fit to be on the road, recommending transport infrastructure that can be developed to increase safety and reduce accidents, and educating motorists and the general public on road ethics, and traffic rules and regulations. To perform these duties and more, the FRSC performs routine vehicle count exercises on various roads across the country to get a sense of road usage and traffic patterns in order to determine how best to make roads safer for Nigerians. They use the pneumatic road tube vehicle counting system.

2.2. Vehicle Counting Methods

Manual counting is a simple method of counting vehicular traffic. It involves an individual using a tally sheet or an electronic hand-held counter to make records. The observers may stand beside the road or even watch a pre-recorded or live video footage of the road and count from that. Only a small sample of data is collected (often, less than a day) in this method and the results are used to extrapolate the traffic for the rest of the year or a given period.

Pneumatic road tubes have been a popular method of vehicle counting for many years. In this system, one or several rubber tubes are stretched across a road and attached at one end to a logging device. The opposite ends of the tubes are covered. When a vehicle's wheels go over the tubes, air pressure in the squeezed tubes activate the logging device which records the time the event occurred. A set of pneumatic tubes may be stretched over multiple lanes of traffic.

Figure 1: FRSC personnel using the MetroCount pneumatic road tube counter to record traffic count along the Lagos-Ibadan expressway (Federal Road Safety Corps, 2019)

The logging device can identify what direction a vehicle is moving in by recording the tube which is crossed first. This has the disadvantage that if two vehicles cross the tubes at the same time then the direction might not be accurately measured. Also, if two cars are very close together when they cross the tubes, the system may see them as one very long vehicle (e.g., a trailer).

Most vendors of this system claim that it is ninety nine percent accurate. Studies, however, show that an average fifteen-minute count has an absolute error of about ten percent. This suggests that the inaccuracy of the system is being veiled by false-positive and false-negative errors cancelling each other out [15].

Other limitations of the pneumatic road tubes used by the FRSC are:

- The system is generally quite inaccurate. It is not efficient on high volume expressways where it has been commonly used by the FRSC.
- The traffic count data needs to be physically transferred from the logging devices installed on the

roadside to a computer system as the data is not transmitted over a network.

Other methods used in vehicle counting include inductive loop, magnetic sensor, acoustic detector, passive infrared and radar and doppler microwave sensors.

The inductive loop system consists of a square of wire which is set up under the surface of a road. It makes use of the principle that a magnetic field placed close to an electrical conductor causes induction of electrical current. In its use for vehicle counting, a vehicle (which consists mainly of metal) acts as the magnetic field and the inductive loop is the electrical conductor [16]. A counting device beside the road stores the generated signals.

The magnetic sensor system works by calculating the shift in the earth's magnetic field as a vehicle passes over it. The sensor may be placed under the road's surface or inside a protective casing beside the road [11]. If two or more vehicles pass over the sensor at the same time, the magnetic detector may find it difficult differentiating them.

Acoustic Detector detects vehicles using the noise created as they drive along the road. The device is usually installed on a pole which points down towards the direction of traffic. The detector is capable of making counts for one or several lanes on a road [12].

The passive infrared sensor detects vehicles by evaluating the energy radiating from the area where the device is installed. When a vehicle goes through the area, the infrared energy radiated changes and a count is made. Road surface temperature changes brought about by the current weather condition are neglected by the device [13].

The doppler microwave detector works by relaying a constant signal of low-energy microwave radiation at a target area and then analyzing the signal being reflected. The device records shifts in the frequency of microwaves. This happens when the wave source and a vehicle are moving relative to one another. This makes it possible for the device to detect moving vehicles. Radar is used to detect objects at a distance and evaluate their position and how fast they are moving. This is applied in vehicle detection by directing high-frequency radio waves on the road to determine the time delay of the signal returned thus evaluating the distance of the detected vehicle from the sensor.

2.2 The Video-based Counting System and its Component Parts

Video-based vehicle counting systems are relatively new compared to their non-video-based counterparts. This is largely due to advancements in image processing and systems infrastructure over the past two decades.

There's a lot of literature on techniques and algorithms for video-based vehicle counting. These techniques majorly consist of detection, tracking and [17]. For a video-based system to work, it has to first detect vehicles driving on the road. Next, it has to track their movement until they leave the frame of the video capture stream or device. This is necessary to avoid double counting vehicles. The last step is counting, which has to happen before a vehicle leaves the frame of the video.

Detection

Object detection is an aspect of computer vision and image processing concerned with identifying instances of objects of a certain class, like vehicles or people, in electronic images and videos. Popular areas of interest in object detection include pedestrian detection and face detection. Object detection can be applied in solving hard problems in areas like image search and video surveillance. It is used widely in computer vision tasks including face detection, face recognition, and object tracking.

All object types have special attributes that help in classifying them. For instance, all faces are round. Object detection algorithms use these special attributes to identify objects in images and videos.

There are generally two techniques used in object detection: machine learning-based techniques and deep learning-based techniques. Machine learning algorithms for object detection include Histogram of oriented gradients (HOG) features [18], the Viola-Jones object detection framework (based on Haar features) [19] and Scale-invariant feature transform (SIFT) [20]. Deep learning algorithms for object detection include You Only Look Once (YOLO) [20], Region-based Convolutional Neural Networks (Faster R-CNN, Fast R-CNN and R-CNN) [22] and Single Shot MultiBox Detector (SSD) [8].

Figure 2 shows the detections made by a YOLO model trained on the popular COCO common objects dataset.

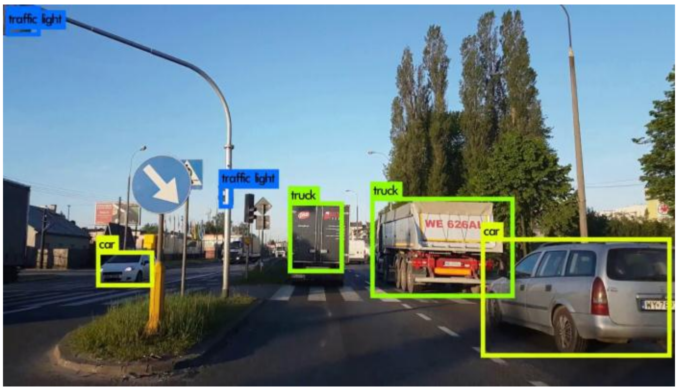

Figure 2: Object detection using YOLO

Tracking

Tracking is the process of following the path or movements of an object with the purpose of finding it or observing its course using a camera [23]. The uses of video tracking include augmented reality, surveillance and security, video compression and communication, video editing, human-computer interaction, traffic control and medical imaging. Video tracking can be very time-consuming because of the volume of data available in video.

The goal of tracking is to associate target objects in sequential frames of a video. This association can be very hard to accomplish when the objects are moving fast in relation to the frame rate of the video. Things get even more complicated when tracked objects change their orientation over time. In this scenario, video tracking systems normally use a motion model which details how the image of the target might look for several possible orientations of the object.

To track objects, an algorithm inspects consecutive video frames and returns the movement of targets between the frames. Several algorithms used for object tracking exist, each with its merits and demerits. It is important to put into consideration the use case before choosing an algorithm to utilize. The main parts of a video tracking system are target representation, target localization, filtering and data association.

Counting

Counting is the final step which involves determining the number of vehicles that have passed at any given time. Vehicle counts may be recorded on the counting device or transmitted to a remote location over the internet. A real-time system requires the latter. This means detection, tracking, counting and transmission has to happen in a matter of milliseconds to a few seconds

3. METHODOLOGY

This study developed a vehicle counting system based on YOLO detection algorithm developed by [21]**.** YOLO is a state-of-the-art deep-learning-based object detection system built for real-time applications. Unlike other detectors that use localizers or classifiers for detection purposes, YOLO applies a single neural network to a whole image. It works by dividing the image into regions and predicting object bounding boxes and probabilities for each region. The bounding boxes are then weighted by the calculated probabilities. This technique gives the YOLO algorithm an edge over other object detection algorithms in terms of speed. It has been benchmarked to be 1000 times faster than the R-CNN model and 100 times faster than the Fast R-CNN model [24].

The tracking module in the vehicle counting system uses Discriminative Correlation Filter with Channel and Spatial Reliability (DCF-CSR) tracking algorithm. DCF-CSR introduces channel and spatial reliability concepts to discriminative correlation filters (DCF) tracking to produce a better tracking algorithm. The spatial reliability map adjusts the filter support to the area of an object suitable for tracking. This makes it possible to enlarge the search region as well as improves tracking of objects that are not rectangular in shape. Reliability scores reflect the quality of the channels of the learned filters and are used as feature weighting coefficients in the localization process [25].

The vehicle counting system was developed using Visual Studio Code (also called VSCode), the Python interpreter and the OpenCV library. VSCode is a free and open-source cross-platform text editor developed by Microsoft. It has many great features including syntax highlighting, intelligent code completion, code snippets and refactoring, Git (a version control system) and debugging. It supports Python development via officially and third-party extensions.

Python is a general-purpose programming language which was developed by Guido van Rossum in 1991. It is a high-level interpreted language that prioritizes code readability. The version of the Python interpreter used in the development of this project is version 3.6.

OpenCV is an open-source library developed for real-time Computer Vision applications. It was originally created by Intel and has found application in many areas including facial and gesture recognition, motion understanding, object identification, motion tracking and augmented reality.

The functional requirements of the vehicle counting system are:

- The system shall be able to process video in common video formats including AVI (Audio Video Interleave), FLV (Flash Video Format), WMV (Windows Media Video), MOV (Apple QuickTime Movie) and MP4 (Moving Pictures Expert Group 4).
- The system shall be able to accept video input from a video camera or a file (pre-recorded video).
- The system shall be able to project the detection and tracking process visually by drawing bounding boxes around vehicles as well as labelling them appropriately.
- The system shall be able to detect and track different types of vehicles including cars, buses, trucks, motorcycles and bicycles.
- The system shall be able to count vehicles using a counting line placed around the point where the vehicles exit the video frame.
- The system shall be able to record, update and display the vehicle count on a screen/window.

Non-functional requirements are the quality attributes of a system, that is, the criteria which can be used to evaluate the operation of a system, as opposed to particular behaviours of the system. The plans for the non-functional requirements of a system considered in the system development are:

- **Auditability:** The system should be easily auditable to validate the results it produces and verify its accuracy.
- **Capacity:** The system should detect and track multiple vehicles at a time accurately.
- **Cost:** The system should be cost effective to scale across a town or city.
- **Data retention:** The system should persist captured data and metadata so that it can be retrieved at a later time and/or transferred to other systems.
- **Extensibility:** It should be easy to add more functionality to the system and use it for other

purposes including, but not limited to vehicle classification, plate number identification and traffic violation detection.

- **Integrability:** The system should be compatible with other transport management systems by using standard data exchange protocols so that it can integrated into existing transport infrastructure.
- **Maintainability:** The system should be easily repairable and updatable in order to minimize its Mean Time To Repair (MTTR).
- **Performance:** The system should be near real-time so that it can provide value quickly for other systems and/or individuals/organizations who need to make decisions based on the data supplied.

3.1 System Architecture

System architecture is a conceptual model that describes the structure and behaviour of a software system [26]. It comprises components and subsystems that collaborate to make the overall system work. The vehicle counting system is made up of loosely coupled components which are integrated together. It consists majorly of a detector, a tracker and a counter.

- **Detector:** The detector identifies vehicles in a given frame or image of a traffic scene. It is able to detect different vehicle types including cars, buses, trucks, motorbikes and bicycles and returns a list of bounding boxes for all the objects detected. A bounding box consists of x-y coordinate of an object as well as its width and height.
- **Tracker:** The tracker receives a set of bounding boxes which it identifies a regions of interest (ROI) to track. It create creates blobs from each bounding box and starts tracking them across frames of a video. A blob is an object or (data structure) which contains information about a vehicle (in this case, a vehicle). Information about a vehicle including its bounding box, centroid (center position or coordinates), area, tracker and count status is stored in the blob.
- **Counter:** The counter monitors blobs to see which ones have crossed a predetermined “counting” line. The counting line is drawn at a close to where the vehicles exit the frame. Once a blob crosses the counting line, it is counted and its count status is updated so that the system does not double count it.

3.2 System Design

A class diagram depicting the relationship between objects and classes in in the vehicle counting system is shown in Figure 3. A class diagram provides a target system summary by defining the objects and classes in the system and their relationships.

The vehicle counting uses different data structure to store and process data about vehicles during a counting session. These include lists, dictionaries and classes (such as the vehicle Blob). The data structures are discussed below:

List: A list is an abstract data structure which is used to store a sequence of values in a specific order where a value may appear multiple times. They are “containers” that can be used to store other data types or structures. They are used extensively in the vehicle counting system application to group and manipulate other data structures.

Ordered Dictionary: The ordered dictionary is a data structure which stores data in key-value pairs where each key is unique and maps to a linked value just like a regular dictionary but also

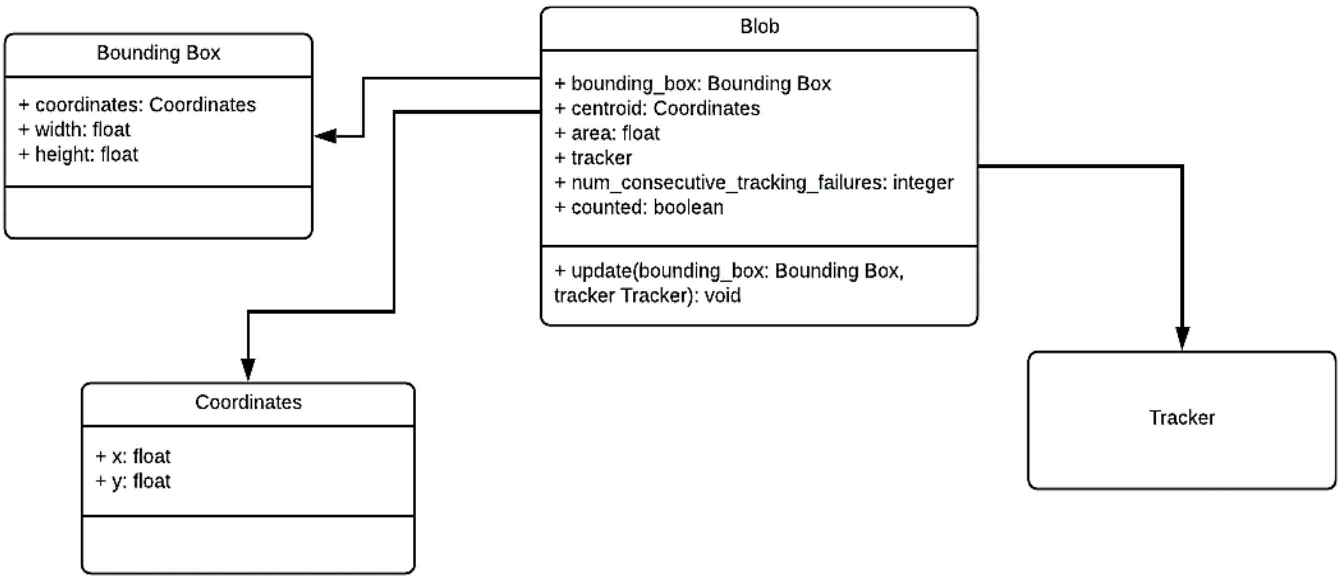

Figure 3: Class diagram representing the data structures in the vehicle counting system

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 12 (2019), pp. 2550-2558
© International Research Publication House. http://www.irphouse.com

stores those pairs in the specific order in which they were added like a list or array. It is used in the vehicle counting application to track Blobs (vehicle objects) with a unique id or key. Table 1 describes the data fields of the ordered dictionary.

Table 1: Ordered dictionary data structure

Field	Description	Data Type	Null
key	Unique identifier of a value.	String	No
value		Any	Yes
index	Uses zero-based integer indexing to order the items in the dictionary.	Integer	No

Coordinates: Coordinates are a set of values that denote a specific point. The vehicle counting application uses a two-point coordinate system to identify the position of objects in a video. This is because, a video is essentially a 2D plane. Table 2 describes the data fields of a Coordinates object.

Table 2: Coordinates data structure

Field	Description	Data Type	Null
x	x-axis.	Float	No
y	y-axis.	Float	No

Bounding Box: A bounding box is a rectangle that encloses an object. The return value of the detector module in the vehicle counting system is a list of bounding boxes. A bounding box itself is a tuple or list of the x and y coordinates of a detection and its width and height. These values can be used to draw a box enclosing an object. Table 3 describes the data fields of a bounding box.

Table 3: Bounding box data structure

Field	Description	Data Type	Null
x	x-axis.	Float	No
y	y-axis.	Float	No
width	The width of an object.	Float	No
height	The height of an object.	Float	No

Blob: A blob is a class-based data structure which stores all the data about a vehicle including its bounding box, centroid and tracker. It is the main data structure in the vehicle counting software application. Table 4 describes the data fields of a Blob object.

Table 4: Blob data structure

Field	Description	Data Type	Null
bounding_box	The bounding box of a vehicle object.	Bounding Box	No
centroid	The center point of a bounding box.	Coordinates	No
area	The area of a bounding box.	Float	No
tracker	A tracker object.	Tracker	Yes
num_consecutive_tracking_failures	Stores the number of consecutive frames in which a specific object has not been detected.	Integer	No
counted	Stores the count status of a vehicle object.	Boolean	No

4. RESULT

The vehicle counting application is made of three independent modules which collaborate to make the system work. They are the detector, the tracker and the counter.

Detector

The detector works by identifying the vehicles (such as cars or buses) in a given frame of video and returning a list of bounding boxes around the detected objects to the tracker for further processing. The detector module uses a YOLO (You Only Look Once) version 3 model to carry out the detection (Redmon et al., 2015). Figure 6 shows the detections made by the YOLO model on a sample traffic scene.

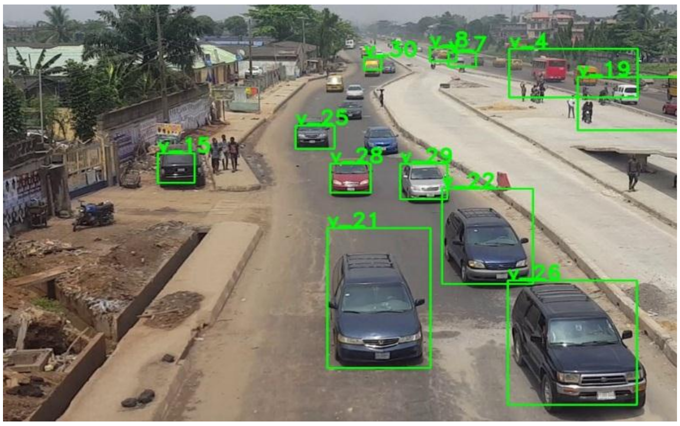

Figure 6: Vehicle detection with YOLO

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 12 (2019), pp. 2550-2558
© International Research Publication House. http://www.irphouse.com

Tracker

The tracker takes in a certain portion of a frame called a Region of Interest (ROI) as input and "monitors" the pixels in that region to note where they shift to in subsequent frames. It returns a bounding box for every frame. This bounding box is a prediction of where an object is in a current frame. Each vehicle has its own tracker instance. Figure 7 shows a sample traffic scene of vehicles tracked using DCF-CSR. Each vehicle is identified by a unique id e.g. v_25.

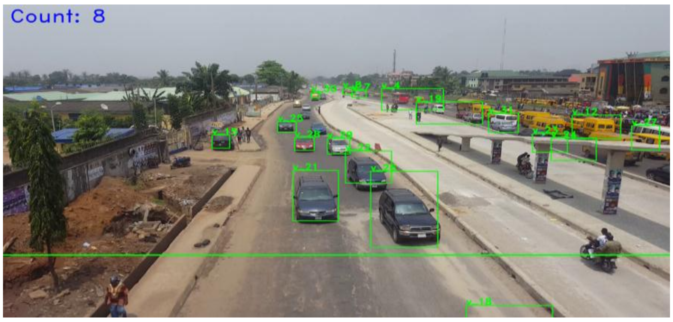

Figure 7: Tracking multiple vehicles with the DCF-CSR

Counter

The counter module counts vehicles by drawing a horizontal counting line across a video. Any vehicle that crosses a predetermined side of the line to the other is counted and metadata about the vehicle and count are logged to a file. This data can also be transmitted for further processing. The count is displayed at the top of the screen so that the counting can be observed visually. Figure 8a shows a counting line drawn across the frame of a traffic scene. It is best to place the line around the area where the vehicles exit the screen so that the detector and tracker have enough time to capture the vehicles. Figure 8b shows an indicator which displays the vehicle count on the top left of the screen.

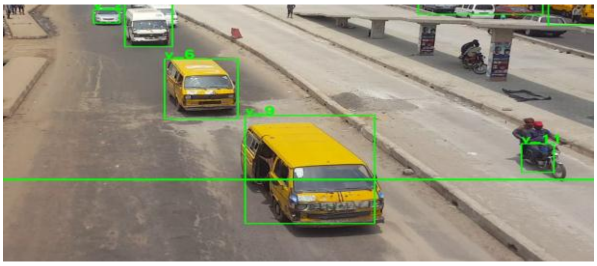

Figure 8a: The counting line

Figure 8b: The indicator showing the number of vehicles counted so far

5 SUMMARY

The vehicle counting application in this study is a prototype designed to run on a computer system (PC) via a Command Line Interface (CLI). It was design to read in real time video file of traffic scenes, detects, tracks and counts the vehicles as they move, projects a visualization of the whole process on a screen and logs the count data, as well as other metadata to a file. Ideally, the system would get its input from a video camera mounted on by a road and transmit the vehicle count to other systems for further processing or store it for analysis at a later time. DCF-CSR was chosen for its accuracy. It can track fast-moving and partially or fully occluded objects. The DCF-CSR tracker for the vehicle counting application runs over 5 consecutive frames, after which the YOLO detector runs, in order to identify new objects and update the trackers of older ones. This cycle continues until the video stream ends.

In the course of this study, several detection algorithms were experimented with. The YOLO deep learning algorithm was the detector of choice for the system as it produced the most accurate results while being relatively fast. The other detection algorithms tried to explain why YOLO was a better tradeoff are Gaussian Mixture-based Background/Foreground Segmentation Algorithm and Haar Cascade.

As with the detector module, several tracking algorithms were tried and it was determined that DCF-CSR produced the best results. The algorithms experimented with in the development of the vehicle counting system include Centroid Tracking Algorithm, Camshift Algorithm and Kernelized Correlation Filters Algorithm.

6. RECOMMENDATIONS

In order for the system to operate effectively, it could be set up in the following manner:

1. A camera should be mounted on a pole or pedestrian bridge to face incoming traffic. The view should be wide enough so that vehicles are detected on time.
2. The camera should be connected (wired or wirelessly) to stream the video captured to a computer or server where the vehicle counting software will run. This computer or server may or may not be in the same physical location of the camera system.
3. The computer/server could be fitted with a Graphics Processing Unit (GPU) to increase the performance of the system.
4. The system will project the video stream, relevant visualizations and metadata to a monitor which may be operated by a transport management personnel. It will also log the count data and relay it to other systems.

7. CONCLUSION

This study explores as aspect of building a robust transportation management system that can handle the challenges that plague urban transportation in Nigeria and indeed other countries of the world. It lays a good foundation for fixing a key component crucial in the development of a robust solution: the vehicle counting system. The vehicle counting system is an invaluable tool in transportation management agencies and systems. It provides information about the flow of vehicular traffic along a given road which is useful in optimizing transportation and planning for the future, so that people and goods can be conveyed from one place to the other faster and resources can be used efficiently. During the implementation of this system, several detection algorithms were experimented with. The YOLO deep learning algorithm was the detector of choice for the vehicle counting system as it produced the most accurate results while being relatively fast.

While existing vehicle counting systems such as the manual counting done and the pneumatic road tubes used by FRSC provide a manageable solution, their limitations make handling the traffic challenges of today very difficult. This study proposes a video-based vehicle counting solution which solves some of the limitations of existing systems such as increased accuracy, flexibility and scalability. There are improvements to be made to make the system faster, more accurate and cheaper to operate. As new computer vision techniques and algorithms are developed, the video-based vehicle counting system would improve accordingly and eventually become the standard for collecting and processing traffic flow data.

ACKNOWLEDGEMENT:

This work was supported by Covenant University. Aderonke A. Oni and Nicholas Kajoh thank Covenant University for their **Sponsor and financial support towards the success of this publication**.

REFERENCE

[1] Adewumi, A., Odunjo, V., & Misra, S. (2015, December). Developing a mobile application for taxi Service Company in Nigeria. In 2015 International Conference on Computing, Communication and Security (ICCCS) (pp. 1-5). IEEE.

[2] Engwicht, D. (1992). *Towards an Eco-city: Calming the Traffic*. Envirobook.

[3] Qi, L. (2008). Research on Intelligent Transportation System Technologies and Applications. *2008 Workshop on Power Electronics and Intelligent Transportation System*, 529–531. https://doi.org/10.1109/PEITS.2008.124

[4] Robert, K. (2009). Video-based traffic monitoring at day and night vehicle features detection tracking. *2009 12th International IEEE Conference on Intelligent Transportation Systems*, 1–6. https://doi.org/10.1109/ITSC.2009.5309837

[5] Benezeth, Y., Jodoin, P.-M., Emile, B., Laurent, H., & Rosenberger, C. (2010). Comparative study of background subtraction algorithms. *J. Electronic Imaging*, *19*, 033003. https://doi.org/10.1117/1.3456695

[6] Chia-Jung, C., Wen-Fong, H., Jun-Wei, H., & Yung-Sheng, C. (2002). Shadow elimination for effective moving object detection with Gaussian models. *Object Recognition Supported by User Interaction for Service Robots*, *2*, 540–543 vol.2. https://doi.org/10.1109/ICPR.2002.1048359

[7] Chandel, H., & Vatta, S. (2015). Occlusion detection and handling: a review. *International Journal of Computer Applications*, 120(10).

[8] Lai, A. H. S., & Yung, N. H. C. (2000). Vehicle-type identification through automated virtual loop assignment and block-based direction-biased motion estimation. *IEEE Transactions on Intelligent Transportation Systems*, *1*(2), 86–97. https://doi.org/10.1109/6979.880965

[9] Tseng, B. L., & Smith, and J. R. (2002). Real-time video surveillance for traffic monitoring using virtual line analysis. *Proceedings. IEEE International Conference on Multimedia and Expo*, *2*, 541–544 vol.2. https://doi.org/10.1109/ICME.2002.1035671

[10] Gajda, J., Sroka, R., Stencel, M., Wajda, A., & Zeglen, T. (2001). A vehicle classification based on inductive loop detectors. *IMTC 2001. Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics (Cat. No.01CH 37188)*, *1*, 460–464 vol.1. https://doi.org/10.1109/IMTC.2001.928860

[11] Taghvaeeyan, S., & Rajamani, R. (2014). Portable Roadside Sensors for Vehicle Counting, Classification, and Speed Measurement. *IEEE Transactions on Intelligent Transportation Systems*, *15*(1), 73–83. https://doi.org/10.1109/TITS.2013.2273876

[12] Barbagli, B., Magrini, I., Manes, G., Manes, A., Langer, G., & Bacchi, M. (2010). A Distributed Sensor Network for Real-Time Acoustic Traffic Monitoring and Early Queue Detection. *2010 Fourth International Conference on Sensor Technologies and Applications*, 173–178. https://doi.org/10.1109/SENSORCOMM.2010.102

[13] Oudat, E., Mousa, M., & Claudel, C. (2015). Vehicle Detection and Classification Using Passive Infrared Sensing. *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems*, 443–444. https://doi.org/10.1109/MASS.2015.62

[14] Atomode, T. I. (2017). Analysis of Freight Vehicles' Proportion and Composition in Urban Traffic of Lagos State, Nigeria. *T. I.*, *11*(4), 15.

[15] Nordback, K., Kothuri, S., Phillips, T., Gorecki, C., & Figliozzi, M. (2016). Accuracy of Bicycle Counting with Pneumatic Tubes in Oregon. *Transportation Research Record*. https://doi.org/10.3141/2593-02

[16] Ali, S. S. M., George, B., Vanajakshi, L., Jayashankar, V., & Kumar, V. J. (2011). A multiple loop vehicle detection system for heterogeneous and lane-less traffic. *2011 IEEE International Instrumentation and*

Measurement Technology Conference, 1–5. https://doi.org/10.1109/IMTC.2011.5944278

[17] Bas, E., Tekalp, A. M., & Salman, F. S. (2007). Automatic Vehicle Counting from Video for Traffic Flow Analysis. *2007 IEEE Intelligent Vehicles Symposium*, 392–397. https://doi.org/10.1109/IVS.2007.4290146

[18] Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, *1*, 886–893 vol. 1. https://doi.org/10.1109/CVPR.2005.177

[19] Viola, P., & Jones, M. J. (2004). Robust Real-Time Face Detection. *Int. J. Comput. Vision*, *57*(2), 137–154. https://doi.org/10.1023/B:VISI.0000013087.49260.fb

[20] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, *60*(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[21] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection. *ArXiv:1506.02640 [Cs]*. Retrieved from http://arxiv.org/abs/1506.02640

[22] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2016). Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(1), 142–158. https://doi.org/10.1109/TPAMI.2015.2437384

[23] Xiang, Y., Alahi, A., & Savarese, S. (2015). Learning to Track: Online Multi-object Tracking by Decision Making. *2015 IEEE International Conference on Computer Vision (ICCV)*, 4705–4713. https://doi.org/10.1109/ICCV.2015.534

[24] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *ArXiv:1804.02767 [Cs]*. Retrieved from http://arxiv.org/abs/1804.02767

[25] Lukežič, A., Vojíř, T., Čehovin, L., Matas, J., & Kristan, M. (2018). Discriminative Correlation Filter with Channel and Spatial Reliability. *International Journal of Computer Vision*, *126*(7), 671–688. https://doi.org/10.1007/s11263-017-1061-3

[26] Osebor, I., Misra, S., Omoregbe, N., Adewumi, A., & Fernandez-Sanz, L. (2017). Experimental simulation-based performance evaluation of an SMS-based emergency geolocation notification system. Journal of healthcare engineering, 2017.