Advances in Science and Technology
Research Journal

Advances in Science and Technology Research Journal 2022, 16(1), 294–302
https://doi.org/10.12913/22998624/144537
ISSN 2299-8624, License CC-BY 4.0

Received: 2021.11.28
Accepted: 2021.12.21
Published: 2022.01.03

Electronic Toll Collector Framework

Abdul Mujeeb[1*], Nisar Ahmed[2], Husnain Arshed[2], Farhan Ajmal Khan[3]

[1] Engineering in Computer Science, DIAG, Sapienza University of Rome, Rome, Italy
[2] Department of Computer Science, Sapienza University of Rome, Rome, Italy
[3] Department of Artificial Intelligence, Sapienza University of Rome, Rome, Italy
* Corresponding author's e-mail: mujeeb.1972811@studenti.uniroma1.it

ABSTRACT
Manual toll collection systems are obsolete due to time, fuel, and pollution issues and need to be replaced by new and better alternatives. Traditionally, governments have always employed people to collect toll, but the manual labor isn't much effective when it comes to monitoring and efficiency. We took this problem and researched out an effective solution i.e., "Electronic Toll Collector Framework" which is a framework mainly for collection and monitoring of the toll fees collected by the toll plazas in the vicinity of metropolitan cities like Lahore or Karachi. The software can generate toll tax based on vehicle type. Additionally, it can also generate daily/monthly/yearly revenue reports. The framework can serve other purposes like monitoring of vehicles (by the law enforcement agencies) and generation of analytics. It can also serve as a backbone for the government departments who are having a hard time monitoring the revenue generated by the employers. There are two operational modes of the framework (partly manual and automatic). The partly manual approach uses TensorFlow backend, and the automatic approach uses Yolov2 backend. This work will be helpful in guiding future research and practical work in this domain.

Keywords: electronic toll collector, TensorFlow, Yolov2, image classification, analytics, centralized system.

INTRODUCTION

Automatic toll collection or electronic toll collection (ETC) is not a new domain. People have been contributing their research and trying to improve the existing solutions or coming up with the new one for a decade now [1]. Most lack perfectness and wholeness. Some offer limited functionalities and others have certain caveats. The problems with most systems are addressed in Table 1. We propose a complete solution with numerous features and two operational modes i.e., partly-manual (primary mode) and automatic. The resultant framework can automate the toll collection, monitor the vehicles, and generate the analytics based on the information stored in the database. The software system is based on deep learning and computer vision [2, 3]. It recognizes the vehicle type passing through the toll plaza with the help of a camera. Vehicle detection is done using Python/OpenCV and TensorFlow/Yolov2. It recognizes whether the vehicle is from the classes that the model is trained on and would generate the toll according to the vehicle class with predefined rates set by the government. This system can also count the total number of vehicles and toll collected by the plaza/booth. The framework offers a website portal for the administrative staff to login and view/monitor the toll collection process. It offers flexibility as toll collection and monitoring is done using two primary methods.

Primarily, toll collection is done using fully automatic system but keeping in mind the unemployment ratio of the country, we have also devised a partly automatic toll collection mechanism where the employee takes a picture of the incoming vehicle (employees would be assigned more task, see below), the picture is classified by TensorFlow after which the image is named with current timestamp and the vehicle type and toll is stored in the database. All the software's use SQLite as the local database, these databases are converted into a .json format and are sent to the centralized server and on successful merge with the centralized database, the local databases are erased completely after generating a local back-up copy in .csv file format. The software starts a new data insertion operation after the erasure part. With all the information that has been stored in the database, the employee can generate analytics for the day which include pie-charts, bar graphs and heatmaps. The web portal also offers the same functionality but with all the data from all the software being used in a country/region. Some of the available solutions [4] in the market are ineffective.

Our approach is, however, better as we don't divide the toll plaza into lanes, all the booths have same software so you can go to any lane where the queue is empty, and you'll be entertained. This helps in reducing the traffic congestion. Other benefits are listed below.

Most electronic/automated toll collection systems don't offer the monitoring feature and access to a web portal for the monitoring authority. Solution proposed offers the owners of the toll booth to either select automatic toll collection or go for partly automatic toll collection. Monitoring feature of the framework can help in reducing the corruption or to track down a specific vehicle with the characteristics for example the vehicle type and the data/time when the vehicle passed through the plaza [4, 5]. A reader works akin a scanner device. It reads the information from the tags and might send the information to the database. One of the benefits of this technology is that the communication can be done without the tag being in a direct line of sight.

License plate recognition based toll collection

License plate recognition-based toll collection is one of the new technologies where you use deep learning algorithms to read the characters from the number plate and charge the credit accordingly [6]. The software has to detect a license plate in an image/frame and after, it has to run deep learning algorithms to determine the characters in the image of that particular license plate [7]. This technique also has some drawbacks stated in the Table 1.

Manual toll collection

Manual tolling doesn't require any technical mechanism, assistance, or specialized knowledge. Everything being done is manual, all the tasks are performed by an employee. Vehicle arrives at the correct lane, pays the toll, and leaves with the change and toll token. This method consumes a lot of time resulting in traffic congestion, air pollution and no record keeping. The toll plaza doesn't have any information about the vehicle resulting in no data that might help scientific studies. Data is important for researchers and to generate specific analytics to determine certain things for example, how many vehicles were minicars? At what time of the day do most cars visit the tolling booth?

RFID based toll collection

This mechanism is popular in the market and is being used by different organizations from hospitals to military. RFID stands for radio frequency identification. The RF reader scans the tag on vehicles and send the obtained information to the database present on the tag. So, this technology depends on two main components namely a tag and a reader [8, 9].

A tag usually consists of an antenna and a chip that can store a unique serial number or certain other information which depends on the memory type. Antenna is responsible for transmitting the information from the chip to the reader. This technology is not yet adopted widely.

PROPOSED SOLUTION

We propose a cheap and effective approach to solve this issue with great accuracy and very little

Table 1. Advantages and disadvantages

No	Manual tolling system	RFID based tolling	License plate based tolling systems
1	Time-consuming	Uses plastic and are not reusable	Intensive computation power required
2	Causes traffic congestions	You need to register to get an RFID tag	Clean license plates are a requirement
3	You can't leave until the employee opens the barrier	A lot less information can be stored on them	Expensive camera is required to read plates even at night
4	Corruption element	They can be hacked easily	Inaccuracy is expected

human effort. The proposed solution is a computer software capable of vehicle detection using live video feed (can also work with images) from the camera installed at the toll booth. The software detects vehicle type, generates the toll against it and stores the information in the database with current timestamp. We have used two methods to accomplish the task. We have utilized the YoloV2 Algorithm and TensorFlow to achieve our desired results. Following para. Illustrates further on the methods and the components that we have used to accomplish the task. Please see the block diagram (Fig. 1) to understand the proposed solution.

The framework is divided into three main components:
- Employee end,
- Authorized user access,
- Admin end.

Employee end

Employee end consists of a physical computer system attached to a camera, having software already installed and all the hardware requirements satisfied. We provide the choice to the employer of either letting the software operate automatically or to ask an employee to do the task manually with a press of a button, the employee can also help with any queries people have. The information i.e. toll, vehicle type, timestamp is saved in a local database, online status service runs in the background and on active internet connection converts the data from SQLite database into json and sends to the website where it is stored in a central database.

Authorized user access

The database can only be accessed i.e., read by authority from anywhere in the world via website portal. Data can only be changed by the administrator. All the record from every toll booth gets stored in this centralized database from where the local governments and authority can check on a certain type of vehicle or perform analytics with the data.

Admin end

Admin end has root access to the website and the database. They can perform all types of operations, modify the website, or extend it to add certain more functionalities. They can create backups of the database and can revoke or provide access to new registered authority members.

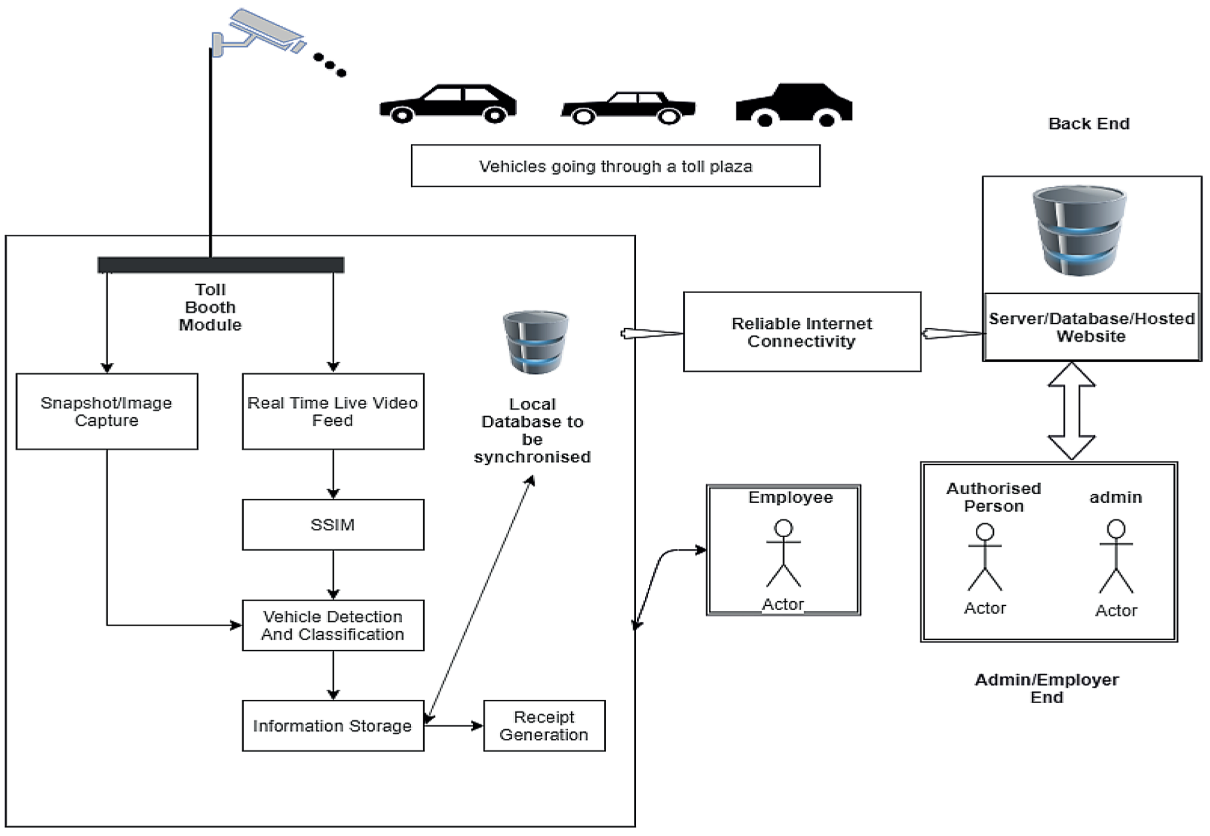
Vehicles going through a toll plaza
Back End
Toll Booth Module
Snapshot/Image Capture
Real Time Live Video Feed
SSIM
Vehicle Detection And Classification
Information Storage
Receipt Generation
Local Database to be synchronised
Reliable Internet Connectivity
Server/Database/Hosted Website
Employee
Actor
Authorised Person
admin
Actor
Actor
Admin/Employer End

Figure 1. Block diagram

METHOD

Dataset description

We studied and picked 2,850 images that were distributed into five classes. We have used TensorFlow image classification algorithm which requires images to be in jpg format. Each class contains a particular type of vehicle images and we attempt to predict the correct vehicle type and generate the toll against that vehicle. All five classes have different toll rates. Training images are unmodified i.e., no thresholder or gray-scaled images. Most of the images are extracted from the web using web scraping scripts and a considerable amount of them were downloaded manually. We also took some pictures ourselves. Our goal was to get a clean dataset with no anomaly, so we monitored all the images downloads and checked them before training. The model was trained on a typical classic laptop on a CPU and took about an hour to train. Figure 2 shows the total number of classes with the type images the classes contain. HTV class contains images of construction vehicles and garbage trucks.

Accuracy measurement

For getting to know how our approach will perform and accuracy of the training model, we ran our experiment on sets like 90–10 (90% training images, 10% testing images), 60–40 (60% training images, 40% testing images). Figures 3 and 4 show the accuracy and cross-entropy curve produced by tensor board respectively [15]. The confusion matrix illustrates the accuracy this system offers (Table 2). The training and validation curves are shown in Figure 5. Activation curve (train and validation) is shown in Figure 6.

APPROACH

To solve the problem in discussion, we selected and downloaded images of different types of vehicles from the web for training purposes. We had to take some images ourselves too. We evaluated our model's accuracy by using tensor board which is bundled with TensorFlow. The inception model [10] from Google was trained on 2,850 images. The accuracy comes out to be very decent even though the number of images for most classes were less than a typical standard i.e., 1,000 (Classes in Alex Net [11] are trained on less than 1,000 images). Architecture of Inception v3 is shown in Figure 7 [12].

Table 2. Confusion matrix

Object	Cars	Buses	Trucks	HTV	Oil tankers
Cars	1200	0	0	0	0
Buses	0	1200	0	0	0
Trucks	0	2	1198	0	0
HTV	0	2	6	1188	4
Oil tankers	0	0	1	3	1196

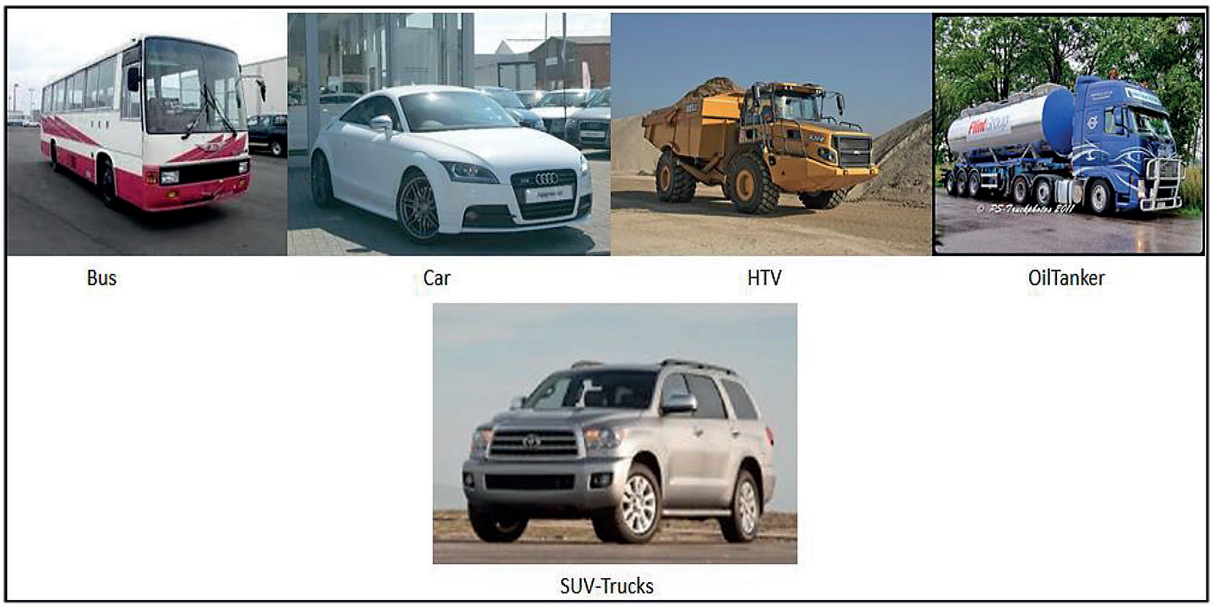

Figure 2. Classes distribution

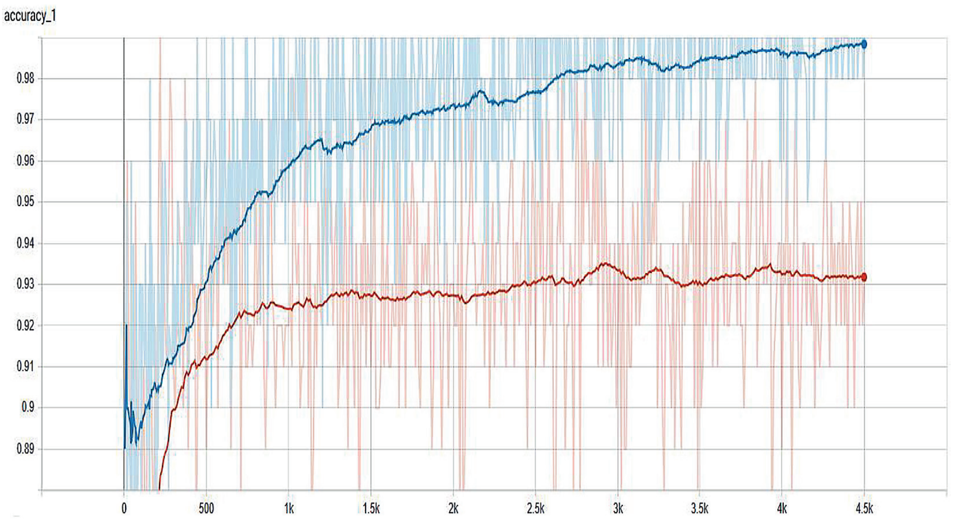

Figure 3. Systems accuracy curve

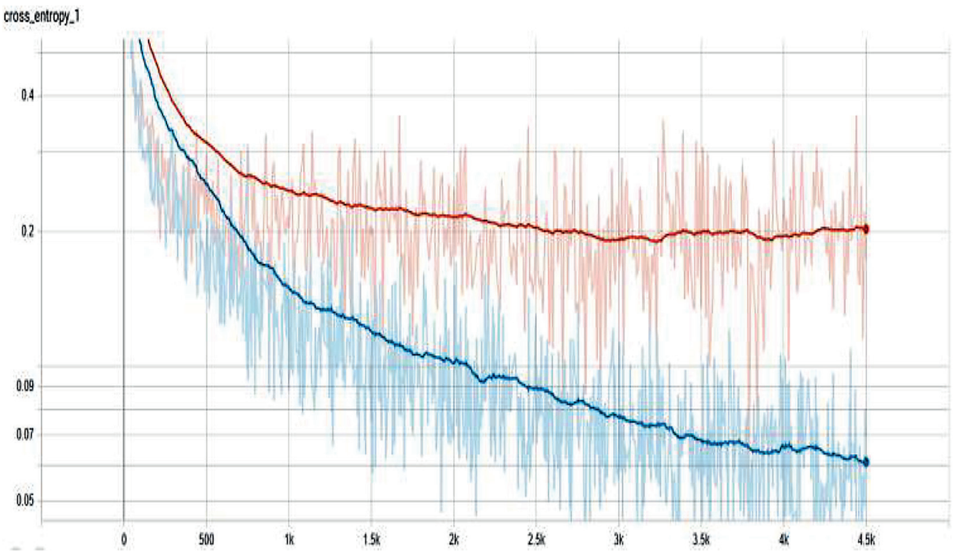

Figure 4. Cross entropy curve

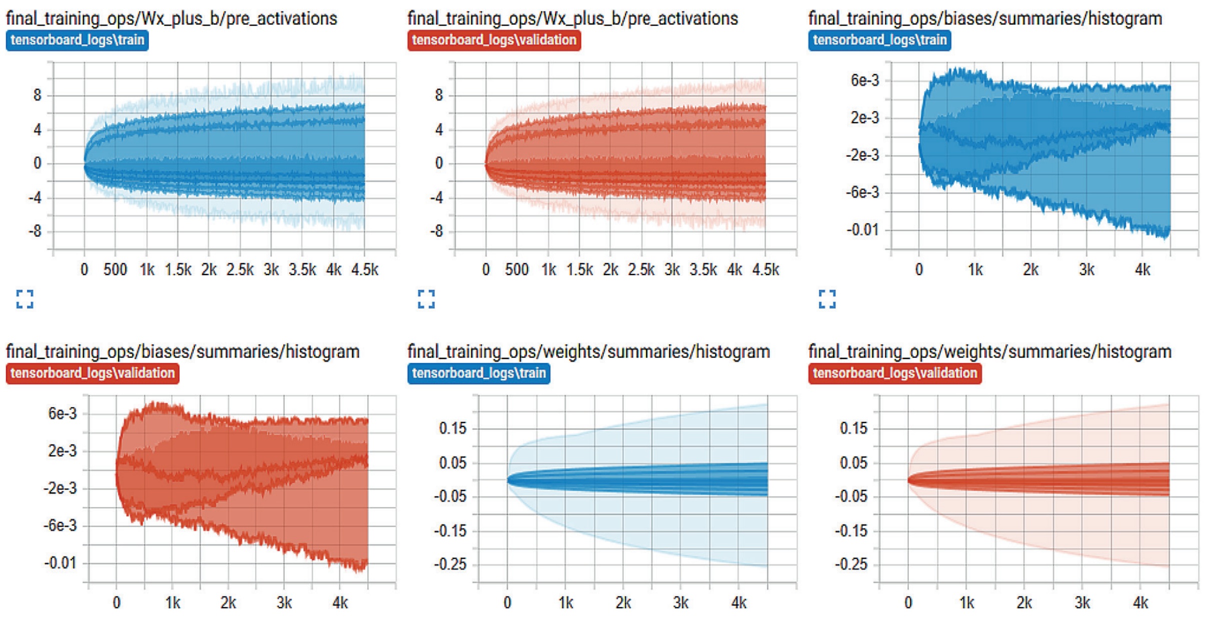

Figure 5. Pre-activation curve for train/validation

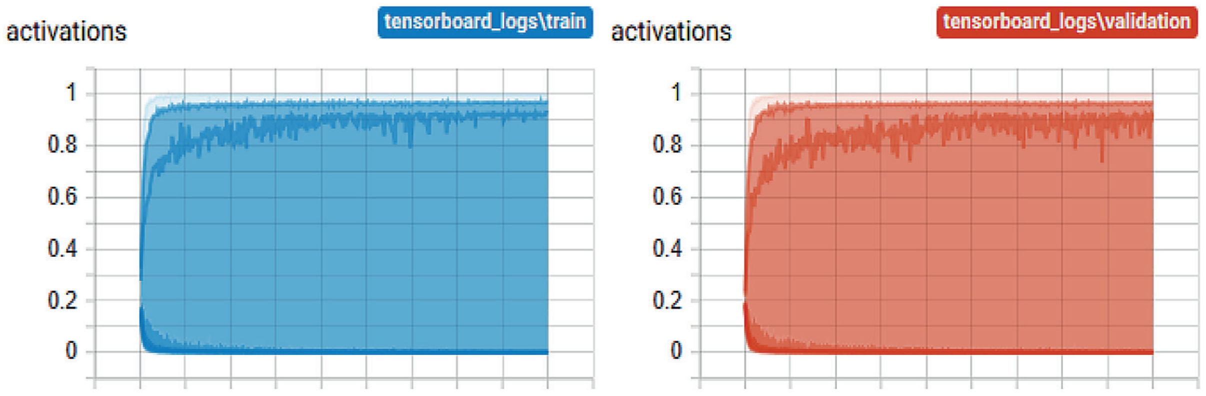

Figure 6. Activations curve for train/validation

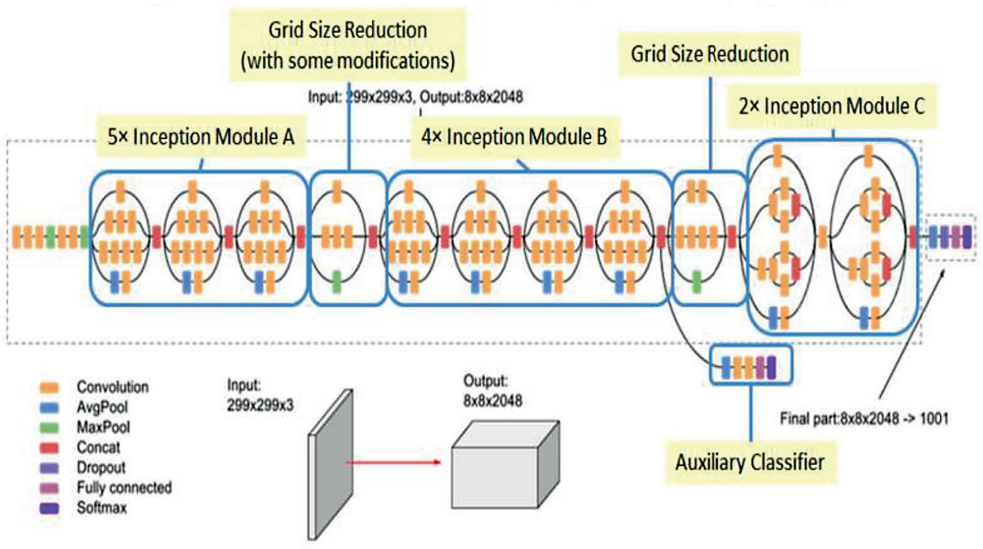

Figure 7. Architecture of Inception v3

Initial approach was to automatically do toll collection, generation and monitoring but keeping in account the contemporary situation of Pakistan and the unemployment issues the nation's been dealing with, we thought of providing a choice to the employer to either use the software for automatic toll collection or the manual (partly).

The automatic toll collection mechanism works by getting live video feed from the connected camera and detecting the vehicle type from the frames. The camera points at a particular region of interest (ROI) that is selected manually while installing the camera. When the vehicle approaches inside the ROI, we utilize the structural similarity index method and detect the vehicle type using YoloV2 trained on custom dataset. On type detection, we save the required unique information in the local database.

Structural similarity index method

As in video signal, there is strong correlation among the frames belong to same scene. In this case, the detection, recognition and pronouncing the same objects is not informative/useful. It is better to check the similarity between two successive frames in displaying order. If similarity index is less than certain threshold T then process current frame otherwise skip this frame i.e., there is no need to process. This skipping not only save the computation power but also help to build real-time system.

In this work structural similarity index method (SSIM) is used to measure the similarity between two successive frames [13]. It is used to take the decision if either the current frame should be processed or not. The following expression is used to measure SSIM index between two frames I and J of dimension M×N.

where: M represents mean of frame I and J, respectively;
N indicates variance of frame I and J, respectively;
M×N is the covariance of frame I and J;
$S_1 = (C_1 \cdot L)^2$ and $S_2 = (C_2 \cdot L)^2$ are used to stabilize the division with denominator;
L indicates the dynamic range of pixel-values which is 255 in this case.

$$C_1 = 0.01 \text{ and } C_2 = 0.03$$

The SSIM index threshold T is adjusted to 0.7 after exhaustive simulations on various video streams:

```
If SSIM > T
    skip the frame
Else
    process the frame.
```

The software tries to connect to the internet after 10 PM every day. It keeps trying to connect until successful. This is achieved by running a background script that only checks internet availability. When the connection is established between the remote server and the local software, it converts the database into .json file that is sent to the centralized server and merged with the central database and can be accessed by everyone who has access to the website portal. The reason for sending local database to the remote server every day is for the monitoring and analytics purposes. On receiving the .json file from the remote system, the server runs a python script that takes the data in. json format and converts it into MySQL queries. Queries are executed and data is stored/appended with the other data stored in the table of MySQL.

Website's frontend is made with bootstrap frontend framework and the back end which was made with Django web framework of python can be downloaded and tested to reproduce the system by clicking on the public link in the supplementary resources section.

In the backend, we have multiple Django models that are being used to store the data in the database. The first model, Tolls, contains all the information related to toll collection i.e., vehicle type, toll generated and timestamp. The second model i.e., Vehicles, contains information on the vehicles passing i.e., vehicle image, vehicle type, toll generated and the timestamp. Manual toll collection is not completely manual rather it allows the employee to handle the part of capturing the image of the oncoming vehicle, handing over the toll to the vehicle driver and getting the money (toll receipt generation is done automatically based on vehicle type detected by our system). With this approach, you will only be able to get three features:

1) Auto vehicles type based toll generation.
2) Monitoring and software-based analytics.
3) Access from web portal for monitoring and analytics purposes.

Even our partly manual toll collection mechanism proves the technical feasibility of the ETC framework. Enlisted below are the frameworks and algorithms that were utilized in the project:

- Choice of image classification system
  - TensorFlow
- Choice of live(real-time) detection algorithm
  - YoloV2
- Choice of training mechanism
  - transfer learning [14] – Inception v3
- Choice of dataset type
  - RGB images (.jpg)
- Choice of training-testing set distributions
  - training: 90%, test: 10%
  - training: 60%, test: 40%
- Choice of remote database
  - Sqlite3
- Choice of GUI framework
  - PyQt5
- Choice of central database
  - MySQL
- Choice of analytics library
  - Python Bokeh 1.2.0
- Choice of website portal backend
  - Django web framework 2.2.1
- Choice of image library for working with images and camera
  - OpenCV 3.4.4

The framework utilizes all the above libraries and algorithms to work properly. The employee-end is easy to navigate and use. The current software for the employee-end has an interactive UI but if you think about it logically and technically, one shouldn't need a GUI to start the software which is an auto toll collection system. Either it should be a back-end service that should just run-on startup or it should be an embedded program in a computer for example, Raspberry Pi. As a matter of fact, the production form of ETC

Table 3. Features comparison

No. of features	Manual toll collection	RFID based method	License plate based method	ETC framework
1	Collects toll	Collects toll	Collects toll but recognition at night is hard	Collects toll correctly based on type of vehicle
2	–	Instant, quick	Quick but might not be accurate	Quick and accurate
3	–	No monitoring	No monitoring	Offers monitoring
4	Different lanes for different vehicles	Different lanes for different type of vehicles	Separate lane	No separate lane, pay the toll tax at the booth where there is no queue
5	–	Might generate analytics	Might generate analytics	Generates all types (daily, monthly, yearly, hourly) of analytics on both the employee-end and website portal. All your questions like, when's the traffic bad? When there are less people on the road? On what dates people travel more often, etc.

Framework's software doesn't need any GUI. Primary reason for the GUI is that the software was made for presentation purposes, so it was decided to have a GUI furthermore, GUI is not an overkill either as we decided to integrate two operational mechanisms i.e., auto, and manual mechanisms for the employer to decide from. Following image shows the GUI of the Employee-End. Which is coded in one of the most popular GUI frameworks of python namely PyQt5. It offers a designer tool, a lot of widgets and an easy way to design/style the graphical interfaces. It also has a lot of active community and support.

Billing/payment can either be done by pay by plate or by registration on the portal. We, however, didn't implement either the pay by plate or portal registration. As the study and work is done in Pakistan, we used online wallets for payments namely Jazz Cash and Easy Paisa. Mobile payments are timeless. Features of the framework are listed in Table 3.

CONCLUSIONS

Deep learning approach to toll collection is an effective, elegant, and efficient approach to solve this issue. The decentralized access of database from any part of the world by web portal is a new addition to the ecosystem of toll collection. The methods of toll collection that we discussed earlier, don't offer the features that our framework offers. You can detect an oncoming vehicle, its type and generate the toll against it.

The prediction comes out to be accurate 98% of the times and by training it on more images, accuracy can improve further. Additionally, this framework offers two operational modes that are elaborated above and offers flexibility to the employer or organization that might use the framework for toll collection purposes. Moreover, you have the data of all the tolls around the country, which apparently, is the new oil and can be utilized for further machine learning related work.

REFERENCES

1. Myung J., Kim D.K., Kho S.Y., Park C.H. Travel Time Prediction Using k Nearest Neighbor Method with Combined Data from Vehicle Detector System and Automatic Toll Collection System. Transportation Research Record. 2011; 2256(1): 51–59. https://doi.org/10.3141/2256-07
2. Ahmed N., Khan F.A., Ullah Z., Ahmed H., Shahzad T., Ali N. Face Recognition Comparative Analysis Using Different Machine Learning Approaches. Advances in Science and Technology Research Journal. 2021; 15(1): 265–272. DOI: 10.12913/22998624/132611.
3. Huang M.C., Yen S.H. A real-time and color-based computer vision for traffic monitoring system, 2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763). 2004; 3: 2119–2122. DOI: 10.1109/ICME.2004.1394685.
4. Redmon J., Farhadi A. 2016. YOLO9000: Better, Faster, Stronger. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, 6517–6525.
5. Atif K., Yakzan E., Adnan, Maaruf A. Radio Frequency Identification (RFID) Based Toll Collection System. 2011; 103–107. DOI: 10.1109/CICSyN.2011.33
6. Davies P., Emmott N., Ayland N. License plate recognition technology for toll violation enforcement. IEE Colloquium on Image Analysis for Transport

Applications. 1990; 7(1–7), 5.

7. Soomro, Shoaib, Javed A., Memon M., Memon A., Fahad. Vehicle Number Recognition System for automatic toll tax collection. 2012 International Conference on Robotics and Artificial Intelligence, ICRAI 2012, 125–129. DOI: 10.1109/ICRAI.2012.6413377
8. Khan A.A., Yakzan A.I.E., Ali M. Radio Frequency Identification (RFID) Based Toll Collection System. Third International Conference on Computational Intelligence, Communication Systems and Networks 2011, 103–107. DOI: 10.1109/CICSyN.2011.33
9. Ahmed S., Tan T.M., Mondol A.M., Alam Z., Nawal N., Uddin J. Automated toll collection system based on rfid sensor. In 2019 International Carnahan Conference on Security Technology (ICCST) IEEE 2019, 1–3.
10. Coy H., Hsieh K., Wu W., Nagarajan M. B., Young J. R., Douek M.L., Raman S.S. Deep learning and radiomics: the utility of Google TensorFlow™ Inception in classifying clear cell renal cell carcinoma and oncocytoma on multiphasic CT. Abdominal Radiology. 2019; 44(6): 2009–2020.
11. Lu S., Lu Z., Zhang Y.D. Pathological brain detection based on AlexNet and transfer learning. Journal of computational science. 2019; 30: 41–47.
12. Xia X., Xu C., Nan B. Inception-v3 for flower classification. In 2017 2nd International Conference on Image, Vision and Computing (ICIVC). IEEE 2017, 783–787.
13. Zhang L., Zhang L., Mou X., Zhang D. FSIM: A feature similarity index for image quality assessment. IEEE transactions on Image Processing. 2011; 20(8): 2378–2386.
14. Torrey L., Shavlik J. Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI global 2010, 242–264.
15. Luus F., Khan N., Akhalwaya I. Active learning with tensorboard projector. arXiv preprint. 2019. arXiv: 1901.00675.