Automatic traffic surveillance system utilizing object detection and image processing

Deval srivastava[1[0000-1111-2222-3333]] ,Saim shaikh[1[1111-2222-3333-4444]] and Priyank shah[1[0000-0003-1211-8252]]

[1] Fr. Conceicao Rodrigues College of Engineering

deval.srivastava24@gmail.com

Abstract. In our day and age where the numbers of cars on the road are rapidly increasing, thereby causing traffic. Drivers are becoming more reckless and carefree as the burden on the current human and automated system grows. Drivers and bikers who may wish to save a few minutes may break red lights and avoid wearing helmets but these small actions can have a significant impact and can result in the loss of lives. We propose a system that will intelligently use deep learning-based object detection to identify traffic offenders and provide methods to penalize them by recognizing their number plate. Our system will be able to detect traffic light violators and bikers without helmets. It has been designed in such a way that it is robust enough to work in drastic conditions and intelligent enough to reduce human dependence.

Keywords: Intelligent systems, smart cities, Deep Learning, Object detection, Image processing, automatic number plate extraction, OCR, neural networks.

1 Introduction

Every day more than thousands of traffic laws are broken and a direct consequence of that is accidents that take place which more than often result in loss of life. Drivers tend to not take the laws that are in place considerably, this belief is mounted on the fact that the systems are ill-considered and unguarded. Drivers are confident in their ability to deceive the system and avoid getting caught by the current laws. This thought when multiplied many times over depending on the population can lead to a severely dangerous driving environment. Currently, many systems and methods have been put in place to prevent the aforementioned problems. Ranging from appointing traffic police officers that perform the job of catching offenders and maintaining order on roads however, there are far more roads than traffic police, and it's only obvious to be aware of the fact that humans will not be available around the clock and also might be able to catch every offender due to being humans. In recent times a lot of thought has been put into this problem and solutions have emerged many of which are already deployed in many cities. A few of these solutions employ intelligent methods to detect lawbreakers and have been perceived with mixed popularity by the police force and the citizens. The concerns regarding the current systems have involved the facts

such as the dependence on humans to identify the offenders from the captured camera fields implying that they are not truly intelligent but rather automated. Other criticisms have been around the fact that the previously mentioned reckless drivers already have devised methods to trick the systems. Lastly, it should be mentioned that the widespread deployment of these systems has been limited due to cost and factors such as needing to add multiple camera feeds on a single road to accomplish the goals. We have developed a novel system for traffic surveillance to detect traffic offenders such as car drivers breaking red lights, bikers not wearing helmets. We achieve this by employing object detection and image processing on a single camera feed. Further, if multiple cameras have been added to observe roads which may be an intersection the system can be used to manage traffic and intelligently manage traffic as per the volume of cars on the road rather than using pre-programmed timers.

From the previous work and research done in this domain, it can be concluded that most systems in this field have employed neural networks to perform classification based tasks and up until recently detecting bikers without helmets has not been a subject of relevant research.

The paper has been organized in the following manner next we will be looking at some of the most comprehensive research that has been done in this field and is relevant to this application, Further that the algorithm and its peculiarities will be explained, after that we can take a look at the complete flow of the system and our design methodology. Post that the dataset, training, and parameters will be discussed. Towards the end, we can look at the results received by us and the conclusion.

2 Literature survey

We have conducted research in this field and can say that lots of work has been done using computer vision for traffic surveillance. Most of the work that has been done has been around vehicle tracking in traffic and traffic flow estimation. Some of the main methods of performing this task are via CNNs and object detection, Background subtraction with image processing, or using a feature-based method. Aleksandr et al[1] have used a faster R-CNN detection algorithm to detect the vehicles along with a SORT tracker[2] using these methods they can detect the traffic density, count of vehicles, and directions of vehicles for traffic flow estimation. The other method is where the moving objects can be separated from the background using background subtraction. The extracted footage will include all the moving vehicles in the camera feed. Neural networks may be trained for the classification of vehicles or not and thus implement a way to detect, track and count vehicles

For the task of traffic light violation detection, some research has been conducted, few companies have made available enterprise systems that use multiple cameras to perform detection. Katanyoo et al[3] have developed a system that is able to detect lane change violations and red light detection. Firstly they are detecting the vehicles and the traffic light whether it is red or not. and the trajectories of the vehicles, after trajectory evaluation they are able to detect the violations.

On the front of helmet detection, there has been some research. Most methods employ CNN for this task. C Vishnu et al [4] have used background subtraction to detect moving objects, they have improved the accuracy of background subtraction using gaussian mixture modeling. After that, they have trained two convolutional neural networks one to classify between bikers and nonbikers and the second to classify between helmet wearing bikers and non helmet this process may seem incessant and arduous; however, the nature of image processing is rapid and much faster than a neural network and easier on the memory bikers.

3 Proposed method

Firstly we can discuss the overall flow and how the core components work together in this high-level view. The camera is set up on the traffic signals and starts capturing frames from the videos. The frames are preprocessed and image standardization is applied before they are fed to multiple object detection models. We have trained three object detection models to detect cars and bikes and helmets, these models are active all the time and are receiving frames. The Features of the system can be broken down into two sections, those which are active during the red light and other features active all the time. When the timed red light goes on we detect the white zebra crossing lines using thresholding and color masking if the lines are not present on the road then a virtual line will have to add to the that are present on roads on intersections, and is compared to the positions of cars predicted using object detection if detected to be ahead of the zebra lines, we will process the image of the car using our custom number plate extraction and recognition algorithm. To localize the number plate we use image processing operations such as edge detection and contour filtering, some post-processing operations such as affine transformation and deskewing are applied to ease optical character extraction. Further, the number plate‘s text can be extracted using OCR. Once the text is extracted the image of the car and the number plate information is pushed to a database for logging offenders.
When the red light is not on we detect bikers using object detection and then detect if the biker is wearing a helmet if not then again the number plate is extracted and the image of the biker and the number plate information is logged in the database.
Finally, if multiple cameras are installed on roads in intersection cars can be counted using the previously mentioned object detection model and in situations when the volume of cars is uneven on the roads of the intersection we can modulate the red light to save time for the drivers and give them less of an incentive to break red lights.

4 Object detection

The deep learning algorithm that powers the crux of our system is known as an object detection algorithm, these classes of algorithms do a combination of tasks where they

have to first classify the image as containing the object or not and then define the location of the object. Tremendous research has been performed in this field and algorithms have been developed that either uses a region proposal network that defines regions that may contain the object then on each of the proposed regions use regression to find the exact coordinates of the box or may utilize a single-shot approach where algorithm does both the region proposal part and regression using the same neural network. In our use case, we have employed a Retinanet based model. Retinanet[5] is an object detection model created by the engineers at Facebook AI research. It uses focal loss to overcome the problem of extreme class imbalance between the foreground and background which is prevalent in one-stage detectors. One-stage detectors such as YOLO[6] although provide much higher speed and performance but lose out on accuracy compared to two-stage detectors such as Faster R-CNN[7]. Focal loss is used to reduce the weightage assigned to samples that have been classified correctly known as 'easy' samples. Focal loss emphasizes the training of a set of hard samples. and prevents the number of easy negative samples to burden the detector during training.

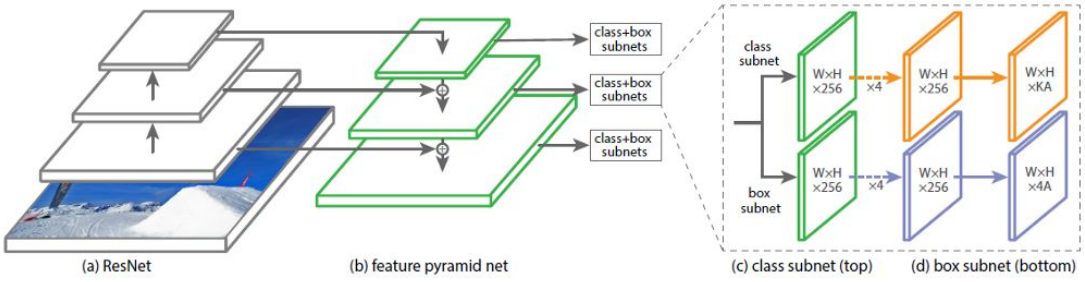

Fig. 1. RetinaNet architecture

The architecture of Retinanet utilizes ResNet[8] for feature extraction from the input images, A Feature pyramid network or an FPN[9] is used along with a Resnet model to generate a multi-scale feature pyramid just from one single resolution image. In object detection detecting objects in varying scales is challenging. A feature pyramid network creates multi-scale feature maps with quality information which is then used for prediction and detecting objects. FPN is fast to compute and is also not heavy on the memory during inference hence is very useful. Retinanet uses two subnets, which are used for classification and bounding box regression. In our use case, we have used a Retinanet with a backbone of Resnet 50 with an FPN for feature extraction. The model has been trained with the focal loss on the coco dataset with our additions.

5 Dataset and Training

For the training of object detection algorithms, we have utilized various datasets. To reiterate on the models we had trained, we have a model to detect cars and bikes and

we have a model to detect helmets. We trained an object detection model on the coco dataset[10] which is an open-source object detection dataset that contains annotated images of common real-life objects as implied in the name as COCO (common objects in context). The coco dataset contains over 81 classes and the dataset has over 200k annotated images, including classes ‘car’ and ‘motorcycle’ to fulfill our needs. In our testing, the model missed out on some Indian vehicles which may not have been covered in the dataset adequately, so the model was fine-tuned on a set of annotated images containing Indian vehicles such as autorickshaws, trucks, lorries, and scooters. We took the images ourselves and annotated them. The second model we have used is to detect helmets on bikers, at the time of project development we were not able to find any readily available open-source helmet datasets so we decided to go ahead with the creation of our own helmet dataset. We recorded over 500 images of helmets with over 800 object instances. This dataset included various kinds of helmets including full coverage helmets to hard hat type helmets and in various colors. Once the collection of images and compilation of images was completed we went ahead to the training of the models.
we have used a transfer learning approach to train the models. Firstly the model is trained on a huge image dataset containing a myriad of categories and lots of images. This allows the model to pick up on some common image features and develop a base knowledge that can be carried over to any specific detection task. In our case, the model was trained on the coco dataset specifically and then trained on our self-made car and bike dataset and also the helmet dataset. We added augmentation to the images to improve their real-world capability and robustness, random zoom, crop, and flip were added to the images before training. Next, we trained our models with an Adam optimizer and a batch size of 8 on a Cuda enabled GPU for acceleration. Once our loss had settled we observed that we have received an accuracy of 93%, 94.6% on our test data splits of the car/bike dataset and helmet dataset respectively.

6 License plate Extraction

One of the major tasks in our system was to perform license plate extraction, this task entails that once we have cropped the picture of the car from the camera feed using our previously mentioned object detection model, Firstly localize the license plate and then from that license plate extract all characters and numbers which can be used to identify and log the car. When working on license plate localization we tested various methods such as haar cascades and neural networks but decided that it will not be computationally economical to use a neural network for this task and we reserved deep neural networks for tasks where it was absolutely necessary to use them and haar cascades did not work perform well on real-world cars and footage. so we decided to build a custom image processing pipeline to extract the location of the license plate. We firstly perform some operations such as grayscale conversion, then CLAHE(contrast limited adaptive histogram equalization)[11], median filters to decrease noise. then a rectangular kernel of (1,7) is used for a morphological opening image dilation operation and then one more image is created by applying a

morphological closing operation on the grayscale image with a kernel of size (7,1). Now, here we will make the difference between the two images closing and opening once that is done we will apply a closing operation with the kernel (3,3). Once the image dilation operations are complete we perform otsu thresholding. on the thresholded image, we can search for contours, But still, we are able to find far too many contours and some filtering will be necessary to do that we apply filters on the length, height and the area of the contour as these values should align with the potential license plate. The filter values were calculated with lots of trial and testing. After filtering the plates with limits on all mentioned criteria we get a set of potential license plate candidates. On this set of license plates, we look for potential characters in the license plate. To do that we perform edge detection, morphological opening, and closing operations to remove small noise, connected component analysis to remove small areas of noise then we search for contours and perform checks on solidity and aspect ratio. These checks allow us to see whether the potential license plates had any characters in them as a license plate would. if a certain number of characters were detected in the candidate license plate we can select it as the detected license plate. In our testing and development, we were able to set values of parameters and perform enough checks such that we only used to get the correct license plate after the whole process. Now, this process may seem incessant and arduous however by the nature of image processing it was rapid and much faster than a neural network and easier on the memory too. Once the image of the license plate was captured we performed a deskew test on it to check if the license plate is deskewed and if that is the case we performed affine transformation to fix that since deskewing may assist the next step which is OCR.

In our testing we had developed a methodology to detect and extract individual characters from the license plate and then further that trained a deep neural network to classify the characters for the task of character recognition, our neural network had been trained on a dataset of various fonts on which we had performed augmentation by adding noise and adding handwritten characters. However, in our testing and development, we encountered the issue that the character extraction from the plate was not accurate. This was due to poor and complicated fonts in Indian number plates, excessive noise, and unnecessary information being present on the plate, due to the lack of laws on standardization of license plates. Few characters were being missed to resolve this issue. We decided to test OCR models that utilize an lstm and a CNN[12] to detect characters present in the string. In our testing we went ahead with the Tesseract OCR solution as we found out much work has been put into this field by the research community and there is not much point to reinvent the wheel.

7 Results and discussion

Our model was trained for over 12 hours on a Cuda enabled GPU and we received an accuracy of 96.6% on the validation test set. The performance on the test set and the validation set was considered satisfactory but to actually test the system we recorded real traffic footage from various roads in the city of Mumbai, India. Further, We provided the videos as input to our system and recorded the results. We also took note

of the offenses manually and then compared them with the results from the model. From approximately 2 hours of footage, the model was able to detect the offense of red light violation with an accuracy of 92.6% and the offense of bikers without helmets with an accuracy of 94.2% %. Other than that we also calculated the accuracy of the license plate extraction which turned out to be 91.7 %. On multiple re-runs, we observed consistent numbers. Model performance was satisfactory in real-world testing and with more data collection and use of state-of-the-art models we could push this number higher. On the front of the license plate extraction, we saw some unprecedented edge cases reduced the accuracy to some extent with more diligent research on image processing techniques, improvements could be made.

8 Conclusion and future scope

On successful implementation, the system provides a robust and efficient method to detect red light offenders and bikers without helmets thus in a way enforcing laws on these offenders and persuading them against breaking laws again. Which in the long term will improve the quality of life and accidents which cause deaths. These systems will enable a better experience for everyone on the road whether it may be pedestrians, drivers, or bikers. Further after testing, we can conclude that our system performed with really good accuracy in our testing results of which have been discussed in the section before. In real-world testing, some unprecedented edge cases hurt performance slightly but they still met our initial goals.

Since we have already created the architecture to detect vehicles and bikers in the future features such as car counting and vehicle density can be calculated on the roads. Lots of data can be collected along with traffic densities at different times. This data can be used for analysis and traffic patterns, which can be used to optimize roads but also a dynamic traffic lights system can be used to reduce unnecessary waiting and overall improve the current static traffic lights.

9 References

1. Fedorov, A., Nikolskaia, K., Ivanov, S. et al. Traffic flow estimation with data from a video surveillance camera. J Big Data 6, 73 (2019). https://doi.org/10.1186/s40537-019-0234-z
2. A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "Simple online and realtime tracking," *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, 2016, pp. 3464-3468, doi: 10.1109/ICIP.2016.7533003.
3. K. Klubsuwan, W. Koodtalang and S. Mungsing, "Traffic Violation Detection Using Multiple Trajectories Evaluation of Vehicles," *2013 4th International Conference on Intelligent Systems, Modelling and Simulation*, Bangkok, 2013, pp. 220-224, doi: 10.1109/ISMS.2013.143.
4. C. Vishnu, D. Singh, C. K. Mohan and S. Babu, "Detection of motorcyclists without helmet in videos using convolutional neural network," *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, 2017, pp. 3036-3041, doi: 10.1109/IJCNN.2017.7966233.

5. T. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 2017, pp. 2999-3007, doi: 10.1109/ICCV.2017.324.
6. J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
7. R. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.
8. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
9. T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 936-944, doi: 10.1109/CVPR.2017.106.
10. J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.
11. G. Yadav, S. Maheshwari and A. Agarwal, "Contrast limited adaptive histogram equalization based enhancement for real time video system," 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, 2014, pp. 2392-2397, doi: 10.1109/ICACCI.2014.6968381.
12. B. Shi, X. Bai and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 11, pp. 2298-2304, 1 Nov. 2017, doi: 10.1109/TPAMI.2016.2646371.