JOURNAL OF
EMERGING INVESTIGATORS

Article

A novel deep learning model for visibility correction of environmental factors in autonomous vehicles

Aniket Dey[1], Abhijit Dey[2]
[1]Dougherty Valley High School, San Ramon, California
[2]DataCore Software, Fort Lauderdale, Florida

SUMMARY

Intelligent vehicles utilize a combination of video-enabled object detection and radar data to traverse safely through surrounding environments. However, since the most momentary missteps in these systems can cause devastating collisions, the margin of error in the software for these systems is small. Furthermore, extenuating weather conditions such as rain, snow, and fog exponentially increase the likelihood of accidents by reducing visibility and increasing the time for detection. In this paper, we hypothesized that a novel object detection system that improves detection accuracy and speed of detection during adverse weather conditions would outperform industry alternatives in an average comparison. To do so, the model employs multiple classical deep learning techniques in two separate sub-modules: a Visibility Correction Module (VCM) and an Object Detection Module (ODM). Firstly, the model employs image classification techniques and masking to identify environmental factors frame-by-frame within an image, and then uses a novel dimensionality reduction network to remove said effects. Next, corrected images are analyzed to classify and label objects within frames. The proposed algorithm achieved an average accuracy of 89.72%, and outperformed industry alternatives in mean accuracy and time for detection, demonstrating the validity and efficiency of utilizing dimensionality reduction to improve object detection.

INTRODUCTION

While modern transportation is trending toward smart traffic, societal acceptance and full implementation of autonomous vehicles ultimately depends on the safety guaranteed during operation of these unmanned automobiles. A critical component of assuring safety in the implementation of these devices is guaranteeing the ability to accurately detect images of pedestrians and other road conditions to make instant decisions predicated on these variables (1). Crucially, as climate change worsens on a global scale, accuracy of such detections can be impacted or degraded by the increased prevalence of environmental conditions like rain, wind, snow, haze, and other natural calamities. These issues could lead to incorrect classifications of traffic environments and surroundings, driving a control system to make incorrect decisions that have disastrous pedestrian and driver safety implications (2). Because of this, building systems that can accurately detect objects through extenuating environmental circumstances is crucial. However, one major concern that arises with incorporating adjustments to environmental conditions is the time required to detect objects. Should a model require more time, on average, to efficiently create a detection, sufficient time may not be provided to the control system to react and prevent a collision. Adding more layers to a model to account for environmental conditions risks increasing time for detection, which could reverse the benefits of safety.

With this problem in mind, this study investigated the validity of creating a novel model for object detection that can perform at high levels of accuracy even with environmental conditions that affect visibility in sensor data, all while critically maintaining a low detection time. We hypothesized that said model, which would employ multiple classical machine learning techniques to edit and recompose images, would outperform industry alternatives in accuracy and speed. The proposed algorithm consists of a Visibility Complementary Module (VCM) as well as an Object Detection Module (ODM). The VCM consists of an assessment of noise & dimensions within each frame, followed by the implementation of deep learning methods to isolate and remove these factors, or mitigate their presence within each image. The corrected image is then fed into the ODM, which primarily consists of identifying and creating bound boxes around said objects based on factors like height, width, and depth. This includes deep learning, as well as convolutional networks for image classification. The model was validated and tested with the Waymo Open Dataset for autonomous vehicle data published by Waymo, a child company of Google (3). To measure relative accuracy, we compared the performance of the model against the YOLO.V3 model (You Only Look Once), a common industry leader in the field of pedestrian detection (4). The algorithm achieved an accuracy average of 89.72% across the board, but notably outperformed YOLO.V3 in every category of environmental conditions. Furthermore, time to detect an object after an appearance was much lower on our model, taking roughly only 0.849 seconds on average, in comparison to YOLO.V3's 1.301 seconds. Splitting the task into two modules and then integrating them into one model after retraining adequately accounted for environmental conditions in real-time, and improved efficiency compared to market alternatives. This paper and its findings have the potential to improve autonomous vehicle safety in hazardous weather by utilizing high accuracy detection algorithms, possibly improving public perception surrounding these vehicles.

RESULTS

The experimental setup consisted of data set enrichment as well as model development. This required the Waymo Open Dataset to be cleaned and parsed to maintain a balance of data between each environmental factor as well as clear data. Finally, model development required software development to match the features present within the Waymo data, as well as validation and testing.

To compute the accuracy of the algorithm, we compared the model to the YOLO.v3 object detection software, which is readily used as a leading software for pedestrian and vehicle detection in autonomous vehicles (4). YOLO.v3 currently does not have an inherent feature that accounts for environmental factors, so it served as a control method. We conducted a comparison between the two algorithms on a variety of features present within the visual data, and both models were run side by side on similar data after training with the Waymo Open Dataset. After testing with 1,000 video segments, consisting of 20,000 total frames in a variety of environments and weather conditions, our model outperformed that of YOLO.v3 in two metrics: object detection accuracy as well as time for detection from the first appearance (**Table 1**). A measured value of 100% accuracy would correspond to a model being able to detect every critical object that is visible on the screen, while a near-zero value for detection time corresponds to a model that can detect objects almost instantaneously. Measurements of accuracy had limitations at times, especially when objects were in very close proximity to one another. However, because there is no purpose in distinguishing between two objects that are close to each other in real-world scenario, so long as the system can detect some type of object in that vicinity that needs to be avoided,

	Weather Classifiers			
Mean Accuracy in Object Detection (%)	Clear	Rain & Snow	Glare	Haze
Our Model	94.21%	85.22%	90.82%	88.63%
YOLO.V3	93.43%	67.31%	83.47%	84.67%
Mean Speed of Detection in Seconds (if detected at all)	Clear	Rain & Snow	Fog	Haze
Our Model	0.783	0.935	0.845	0.834
YOLO.V3	1.065	1.435	1.371	1.333

Table 1: Mean accuracy and speed of detection in object detection for proposed model versus YOLO.V3. Mean accuracy in object detection and mean speed of detection in seconds is displayed for multiple categories of different weather categories such as clear, rain & snow, glare, and haze induced conditions for both models. Accuracy and speed statistics for both models were produced by training said model on the Waymo Open Dataset.

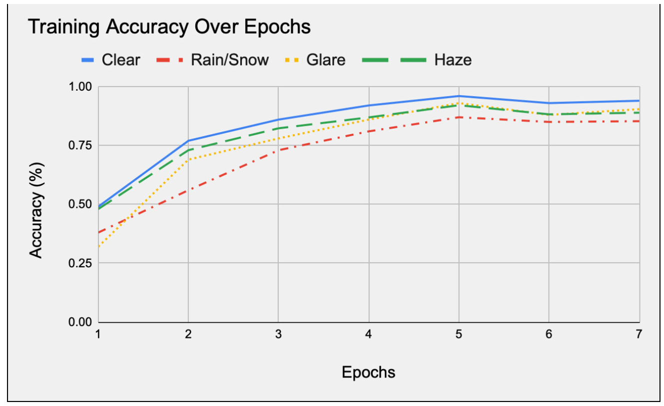

Figure 1: Training accuracy over epochs for proposed model under several weather conditions. Graph displaying accuracy over multiple epochs, or run-throughs, of data set with scale starting at one epoch and retraining up to eight epochs; different line for each type of weather condition.

this limitation to accuracy was largely discarded. Retraining of the model also meant that accuracy and time for detection improved over multiple epochs, or run-throughs of the testing data set. On average, runtime decreased through retraining, and accuracy increased in a logistical manner. Specifically, for each epoch the accuracy of the model increased substantially, yet the acceleration of this increase tended to decrease over time as well (**Figure 1**).

While accuracies for our model differed by level across the board, the average accuracy for the model was approximately 89.72%, with a standard deviation (SD) of 0.53% (**Table 1**). Notably, the model's performance was best under images with entire image factors (EIF) including haze and glare. For the validation dataset, the mean accuracy for the model was approximately 88.63% (SD=0.42%) for haze-induced conditions and 90.82% (SD=0.47%) accurate in glare-induced conditions (**Table 1**). The category of data that had the lowest accuracy in the model was determined to be on-camera factors (OCF), such as rain or snow, which had a mean accuracy of approximately 85.22% (SD=0.66%), significantly lower than other categories such as haze and glare (**Table 1**). It is important to note that while rain and snow can often contribute to EIFs, the frames influenced by rain and snow and classified as OCFs were labelled as such for having visual blots on the frame itself rather than effects on the entire frame. This specification also made the difference between a rain or snow blot being indistinguishable in this case.

Mean time for detection for the model was approximately 0.849 seconds with a standard deviation of 0.00521 seconds (**Table 1**). The variations in performance largely followed the same trend as mean accuracy, with EIF-induced data producing the lowest time for detection and OCF-induced data producing the highest. In a comparison to the industry standard, our model outperformed current market alternatives like the YOLO.V3 detection systems. On average, our model is 9.73% more accurate across the board compared to YOLO.V3, and is up to 19% more accurate at its best. The model also takes, on average, 0.51 seconds less to identify an object than YOLO.V3.

DISCUSSION

The model proposed in this study maintains an accuracy of 89.72% while taking on average 0.849 seconds to detect objects, outpacing industry models. However, the model tends to underperform in OCF-induced data in both mean accuracy and mean speed of detection, hindering overall efficiency. Limitations and inefficiencies within the study can largely be attributed to the selection of data that the model was primarily tested upon. Most notably, the gap in accuracy and runtime for data with noticeable OCFs was likely because the recomposition network received less training with OCF images due to limited data within the Waymo Open Dataset. Furthermore, by splitting the model into two modules, although integrated together, the runtime is nonetheless affected as data needs to be processed separately by both modules. Another inefficiency that may positively influence accuracy is the inherency of false detection. In a case scenario in which two objects are extremely close to one another, such as two pedestrians or a pedestrian riding a bicycle, the model could classify multiple objects as one, creating an inverse trade-off between detection rate and accuracy.

For future experimentation, we plan to incorporate two main improvements to improve model accuracy and lead to a faster run time. We plan to train the model with more diverse data including rain and dust spots to improve the accuracy of our model in conditions with OCFs. As our current dataset has limited data with OCF-induced frames, future training and testing could involve using alternate datasets, such as Berkeley Deep Drive 100K which has similar properties to Waymo Open (5). We also plan to improve the ODM by incorporating three-dimensional topographical data. This would involve the usage of light detection and ranging (LIDAR) Data, which is included within the Waymo Open Dataset. LIDAR imaging creates topographical maps of a vehicle's surroundings, thus using corresponding LIDAR data to our data would be crucial in developing a more accurate model that can process multiple dimensions of data (6).

The results produced by the model demonstrate that environmental conditions can be properly accounted for even with a low runtime for detection systems. By making the detection of critical subjects like pedestrians and other vehicles more accurate even when there are strenuous weather conditions, the model created and tested in this study has the potential to revolutionize the safety of autonomous vehicles, as well as the accuracy of their object detection systems.

MATERIALS AND METHODS

To train and test the model, 50,000 video segments from the Waymo Open dataset were used (3). Each segment within this dataset consists of roughly 20 seconds of video recorded by an autonomous vehicle, as well as LIDAR data which was not used in creation of the model. The dataset was split into training & testing data, which was 70% training and 30% testing. With training data, the model was fed data based off predetermined weather categories such as snow, rain, and haze. Furthermore, each 20 second video segment was broken into roughly 500 frames for analysis using OpenCV, a platform for image and video processing (7). OpenCV was also used for grayscale application and masking of frames. Model development followed a cycle that consisted of both the VCM and ODM, with numerous steps for dataset enrichment and software development in the process (**Figure 3**).

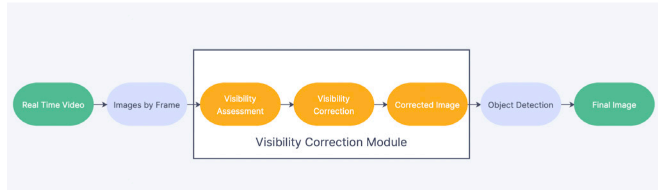

Figure 3: An algorithm flowchart of module functionality and structure. The two modules within the model are interconnected and pass frames from real-time video to one another after performing their functionality.

Image frames were first fed into the VCM, which conducted a visibility assessment, followed by dimensionality reduction or feature selection to remove environmental factors. First, to assess the visibility within each frame the model computes the standard deviation of the pixels within the image, comparing the assigned value of each pixel's contrast to the mean value of the frame on a numerical scale as specified by the OpenCV software. A threshold for standard deviation was determined for accurate classification of images as having an EIF such as haze or fog. To do this, multiple standard deviations were used, and relative performance was computed based on a comparison between the number of factors detected and the actual number of factors within the image. After delineation between several thresholds ranging from 60-110, the most optimal threshold was computed to be $S_x = 95$, so that any standard deviation above this threshold was classified as possessing an EIF.

To remove any potential EIFs as well as OCFs such as rain drops or dust spots on camera lenses, a combination of dimensionality reduction and feature selection was utilized. For frames with noticeable OCFs, the following process was followed. First, the model applies a grayscale filter to the entire frame with the OCF. Then, utilizing a convolutional neural network, regions of interest (ROIs) that could be potential OCFs were identified and boxed within each frame. These ROIs were detected as havingsignificantly higher pixel values than the average pixel value of the image. Finally, the model mitigates the effects of the OCFs by utilizing a medium blur as well as a pre-trained image recomposition model to reconstruct video accurately without hampering visibility further. If a frame is classified to contain EIFs, the

Figure 2: Comparison of frames for ODM. Frame from real-time video before (left) versus after passing through the ODM (right). Images corrected by ODM are passed into a Single Shot Multi-box detector to create and classify bounded boxes surrounding said images.

model passes the frame through a corresponding dark or light channel for contrast adjustment, which reduces the effects of the EIF. This process is repeated until the standard deviation of pixels is approximately 85. Corrected frames are processed into the ODM where critical subjects are identified and classified, and the model then creates bounded boxes around said subjects. Subjects can be classified into one of eighty labels, which included “person,” “bicycle,” and more.

After image processing has been completed, the model passes frames through a Single Shot Multibox Detector (SSD). This model utilizes a single deep neural network to extract features identified within an image using an arbitrary backbone. Then, the image is passed through a matching phase to train associatinganchor boxes to bounding boxes of each ground truth object within an image. Bounding boxes are calculated at multiple resolutions, which are then reduced with extra feature layers. The output of six levels of resolution is concatenated, and non-critical bounded boxes are filtered out using non-maximum suppression. After generating bounded boxes with the SSD, a convolutional neural network is used to identify and process each image contained within each bounding box. The model then chooses from one of eighty distinct labels to classify the image for a vehicle’s control system to process. Boxes and their labels are then displayed in real-time in each frame, which enables a control system to make decisions based on these inputs (**Figure 2**).

To compare the relative performance of the algorithm to other models, two indicators of performance were measured for both our model and YOLO.V3: detection accuracy and speed. Detection accuracy was measured by comparing the number of objects detected during certain time intervals to the actual number of objects present within the frames, the values of which were procured in the Waymo Open Dataset. This was a modified version of the Mean Squared Error, a statistic estimator in which points are compared in proximity to values on a fitted line. Fitted values were produced by the Waymo Open Dataset, while the values computed by the model and YOLO.V3 were plotted in comparison. This process yielded a mean accuracy for each 20-second interval as well as a standard deviation. Similarly, the speed of detection was measured by comparing the time it took to detect an object after its appearance on screen versus how long the object was visible within the frame.

Received: April 17, 2022
Accepted: June 25, 2021
Published: October 31, 2022

REFERENCES

1. Balasubramaniam, Abhishek, and Sudeep Pasricha. "Object Detection in Autonomous Vehicles: Status and Open Challenges." *arXiv preprint* arXiv:2201.07706 (2022).
2. Hasirlioglu, Sinan, and Andreas Riener. “Challenges in Object Detection under Rainy Weather Conditions.” *Lecture Notes of the Institute for Computer Sciences*, Social Informatics and Telecommunications Engineering, 22 Feb. 2019, pp. 53–65., doi.org/10.1007/978-3-030-14757-0_5.
3. Sun, Pei, et al. "Scalability in perception for autonomous driving: Waymo open dataset." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.
4. Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint* arXiv:1804.02767 (2018).
5. Yu, Fisher, et al. "Bdd100k: A diverse driving dataset for heterogeneous multitask learning." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.
6. Zermas, Dimitris, et al. “Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications.” *2017 IEEE International Conference on Robotics and Automation (ICRA)* (2017): 5067-5073.
7. Dardagan, Nađa, et al. "Multiple object trackers in OpenCV: a benchmark." *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*. IEEE, 2021.