

Object Detection from a Vehicle using Deep Learning Network and Future Integration with Multi-Sensor Fusion Algorithm

Raja Sekhar Rao Dheekonda*

Sampad K. Panda[§]

MD. Nazmuzzaman Khan[§]

Mohammad Al-Hasan*

Sohel Anwar[§]

Indiana University Purdue University Indianapolis

[§]Department of Mechanical Engineering

*Department of Computer Science

Copyright © 2017 SAE International

ABSTRACT

Accuracy in detecting a moving object is critical to autonomous driving or advanced driver assistance systems (ADAS). By including the object classification from multiple sensor detections, the model of the object or environment can be identified more accurately. The critical parameters involved in improving the accuracy are the size and the speed of the moving object. All sensor data are to be used in defining a composite object representation so that it could be used for the class information in the core object's description. This composite data can then be used by a deep learning network for complete perception fusion in order to solve the detection and tracking of moving objects problem. Camera image data from subsequent frames along the time axis in conjunction with the speed and size of the object will further contribute in developing better recognition algorithms. In this paper, we present preliminary results using only camera images for detecting various objects using deep learning network, as a first step toward multi-sensor fusion algorithm development. The simulation experiments based on camera images show encouraging results where the proposed deep learning network based detection algorithm was able to detect various objects with certain degree of confidence. A laboratory experimental setup is being commissioned where three different types of sensors, a digital camera with 8 megapixel resolution, a LIDAR with 40m range, and ultrasonic distance transducer sensors will be used for multi-sensor fusion to identify the object in real-time.

INTRODUCTION

Advanced driver assistance system (ADAS) and autonomous driving are promoted by major OEMs and researchers as the technology of the future automobiles. Many aspects of the active safety systems in automobiles

that currently exist (e.g. collision avoidance system, lane keeping assist, blind spot detection, adaptive cruise control) can provide the basics of an autonomous vehicles' framework. However, many challenges remain with the current state of the art in ADAS technologies that hinders safe and robust implementation of the autonomous vehicles for consumers. One of the key challenges is real-time detection and identification of a moving object of various shapes and speeds under all possible environments and backgrounds.

For object detection and classification, image and video data from camera is the only information source in the existing methodologies [17-23]. However, merely image data is not able to deliver real-time and high performance—what is needed for autonomous vehicle. So some recent works are focusing on sensor fusion. For example, Chavez-Garcia et al. [16] used LIDAR, RADAR, and camera in fusion for moving object detection in the vehicle setting. However, their image detection methodologies use HOG (histogram of oriented gradients) descriptors as feature, whose performance is much poorer than the performance of recently proposed deep learning approaches.

In this ongoing work, we are using sensor fusion where LIDAR, digital camera, and ultrasonic distance transducer data are being used for multi-sensor fusion to identify objects in real-time.

BACKGROUND

A.D.A.S. AND AUTONOMOUS DRIVING

The Stanford cart, built in 1961 is considered to be the first intelligent vehicle [1]. It was built to control moon rover from Earth with only one camera. In 1987, 'Prometheus project' was proposed by Dickmanns to conduct research on autonomous cars [2]. This project created two important demonstrations. First, in 1994, two

robot cars (VaMP and VITA-2) were driven in Paris multilane highway at speeds up to 130 km/hr. Second, in 1995, S-Class Mercedes-Benz car did a round trip from Munich to Copenhagen, driving up-to 175 km/hr on the German Autobahn. But they had limitations like, using only one lane, limited obstacle detection and constant human inputs.

The Defense Advanced Research Projects Agency (DARPA) challenge was developed by U.S. government for the development of autonomous vehicle research. The challenge was to traverse a 200 km road through Mojave Desert in 10 hours. In 2004, no vehicle completed more than 5% of the course but in 2005, five vehicles were able to complete the course. TerraMax was one of the vehicles that reached the end of the DARPA Grand Challenge with vision as its primary sensing technology [3]. But these robots could not accommodate moving traffic (which was not a requirement of the challenge); besides, they used redundant sensor configurations to achieve these goals [4].

In recent days, many research groups and companies are doing extensive investigation to develop driverless cars. High-speed car control was tested by Stanford [5] in real life scenarios in Pikes Peak Rally drive. Autonomous operations in real scenarios were targeted by many others like Google [6], Braunschweig University [7], and Berlin University [8], which create new milestones in autonomous driving. But these cars use sophisticated sensors like 3D LIDAR scanner, 2D LIDAR scanner, stereo cameras and many other software resources which most of the times exceeds the cost of the car itself.

Advanced driver assistance system (ADAS) is an intelligent system to assist the driver to avoid dangerous situations and safe navigation. The job of ADAS is to first detect and process objects such as pedestrians, cars, bikes, buses, trucks in highway, countryside and urban areas, and then to make real time decisions and warn the driver if a dangerous situation (possible clash with cars or pedestrians) arises. All the intelligent cars mentioned above were not fully autonomous which means they were not capable of dealing with all the possible situations that may arise during every day driving. One of the main reasons is the lack of information needed about the environment to make decisions. One single sensor is not enough to deliver all the data needed about the environment to make safe decisions. Hence, multi-sensor fusion (combining data from different sensors) is needed where signals from different sensors can be utilized to mutually overcome their respective shortcomings.

DEEP LEARNING BASED METHODS FOR OBJECT DETECTION

For ADAS and autonomous vehicle, achieving high detection performance and near-real-time object detection on an embedded system is a key requirement. Among different classification algorithms, cascade classifier is one of the fewest algorithms that can meet this requirement. The key to its high performance is the use of

integral image, which only performs basic operations in a very low processing time. However, a crucial limitation of cascade classifier is that it works only for fixed aspect ratio. Viola and Jones [25] have overcome this limitation by designing rectangular Haar-like features and then use these features with AdaBoost learning algorithm combined with a cascade of strong classifiers. Then Leinhardt and Maydt [26] have improved the set of Haar-like features. In another work, Ahonen et al. [27] applied Local Binary Patterns (LBP) for face detection and obtained good results compared to Haar-like results. Since then, both Haar-like and LBP has been used extensively for object tracking in real time. They are also implemented in OpenCV (opencv.org), an optimized C++ library for computer vision applications.

Meanwhile, machine learning community had been working on learning features from raw images using multiple layers of non-linearity, leading to deep learning. Early example on these are Restricted Boltzmann Machines [28], Deep Belief Networks [29], and Stacked Autoencoders [30]. The main idea of these methods is to learn features in an unsupervised manner so that a pre-train model can be built, which works as a good initialization for subsequent supervised classification system. Later in 2012, Krizhevsky et al. [29] proposed a Convolutional Neural Network (CNN) based deep learning model, which brought down the error rate on a benchmark classification task (ILSVRC competition) by half, beating traditional hand-engineered approaches. Importantly, CNN can achieve this performance by an end-to-end supervised training, without the need for unsupervised pre-training. Since then, CNN and its various architectures has become the best tool for real-time object classification from image data [17-21, 32-33]. A recent survey on CNN [34] discusses different CNN architectures along with their successful applications for object recognition in different domains. In this ongoing work, we are using CNN along with sensor fusion for better object recognition in real-time for supporting ADAS and autonomous vehicles.

MULTI-SENSOR DATA FUSION METHODS

Theoretical scientific literature on data fusion started to appear in late 1960s with implementations and more algorithmic developments in 1970s and 1980s. Hall and Garga [9] have highlighted the pitfalls in data fusion that justify the active research which is still ongoing. Especially the point '*There is no such thing as a magic or golden data fusion algorithm*' implies that fusion techniques very much depend on the sensors used, the problem to be solved and even the level of pre-processing done on the sensor data [10]. Multi-sensor data fusion is the process of combining **several** observations from different sensor inputs to provide a more complete, robust and precise representation of the environment of interest [11]. The fused results from more than one sensor is believed to be better than individual sensor inputs. In the following, we review the main fusion approaches highlighting advantages and drawbacks.

Probabilistic Methods

Occupancy Grids: Probabilistic occupancy grids (POGs) are conceptually the simplest approach to implement Bayesian data fusion methods. Although simple, POGs can be applied to different problems within the perception task: e.g., mapping [12], moving object detection [13], and sensor fusion [14].

Kalman Filter (KF): KF features make it suited to deal with multi-sensor estimation and data fusion problems [11]. First, its explicit description of processes and observations allows a wide variety of different sensor models to be incorporated within the basic algorithm. Second, the consistent use of statistical measures of uncertainty makes it possible to quantitatively evaluate the role each sensor plays in overall system performance.

Monte Carlo (MC) Methods: MC methods are well suited for problems where state transition models and observation models are highly non-linear [11]. The reason for this is that sample-based methods can represent very general probability densities. In particular, multi-modal or multiple hypothesis density functions are well handled by Monte Carlo techniques [12].

The main limitations of probabilistic methods for

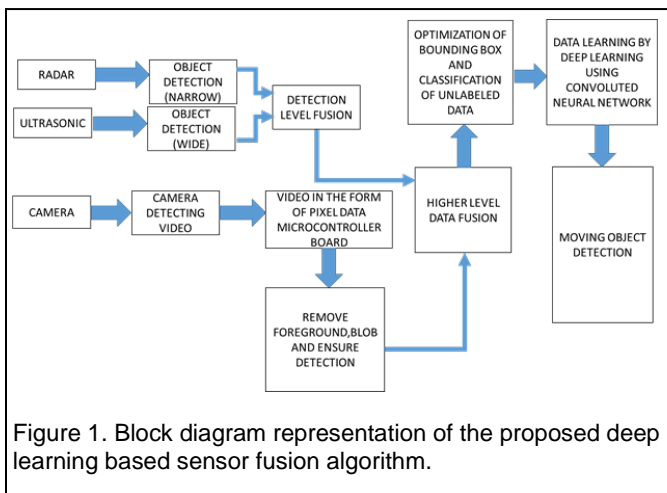


Figure 1. Block diagram representation of the proposed deep learning based sensor fusion algorithm.

information fusion are [11] complexity (need large number of probabilities), inconsistency (difficult to specify consistent set of beliefs in terms of probability) and model precision (precise probabilities about almost unknown events).

Non-Probabilistic Methods

Interval Calculus (IC): In this method, uncertainty is represented by bound values. One major advantage compared to probabilistic method is that IC provides better measures of uncertainties in absence of probability information but the errors of sensor data are bounded to a certain value. However, IC are not generally used in data fusion because of the difficulty to get results that converge to a desired value; and the difficulty to encode dependencies between variables which are at the core of many data fusion problems [11].

Fuzzy Logic: Fuzzy logic is a popular method in control and sensor fusion to represent uncertainty where reasoning is based on degrees of truth rather than absolute value. But this method becomes more complex with the increase of sensor inputs. Also, validation of this method needs extensive testing where safety is an important factor [15].

Evidence Theory (ET): The advantage of ET is its ability to represent incomplete evidence, total ignorance and the lack of a need for a priori probabilities [11]. In the field of intelligent vehicle perception there is a variety of imperfect information: uncertain or imprecise. For example, objects are missing (occlusions), sensor cannot measure all relevant attributes of the object (hardware limitations), and when an observation is ambiguous (partial object detection). But with higher number of hypotheses ET becomes less computational tractable.

DEEP LEARNING BASED METHODOLOGY

It is observed that the accuracy in detecting and localizing a moving object still lacks robustness when environmental and surroundings change. In this work, as a first step toward sensor fusion based deep learning, we utilize the deep learning technique using only imaging data to detect a moving object under varying environment and surroundings. The deep learning algorithm allows us to learn the environmental conditions and background with higher degree of accuracy which in turn helps improve the accuracy of detection, localization, and classification of the object in the foreground.

The objective is to use a relatively inexpensive camera with 8 megapixel resolution and a 77 GHz radar. This millimeter wave radar technology offers simultaneous long- and mid-range functionality and allows one radar to be used for multiple systems. The radar's transmitter emits signals reflected from objects ahead, at the side and to the rear of the vehicle and are captured by multiple receivers. The radar system can detect and track objects in the frequency domain.

Figure 1 delineates the proposed multi-sensor based fusion and deep learning methodology aimed for this research. Here the radar and ultrasonic sensor data are to be used to detect the distance of the object which would be fed into the first level of sensor fusion to improve the robustness of detection for wide range of distance and sensor horizon. Camera images are converted into pixel level data which are processed to remove foreground. This image data are segmented and then all segmented data are structured in the form of a tensor. The tensor data is then fused with the distance detection data. This composite tensor is then further processed to optimize bounding box. At this point, the data is ready to be using the deep learning algorithm via CNN.

The methodology deals with object tracking and classification in order to detect different obstacles

traversing at different speeds and designing ways to implement learning the process of detection.

The following steps are proposed to be included in the overall algorithm.

1. Moving object detection by tracking and classification
 - Classification of unlabeled data obtained from the camera
 - Calculation of distance from the object indicated and estimating the error found from the sensor data
 - Determination of the size of the object from the sensor by a prior knowledge of the specified class.
2. Sensor Fusion
 - The combination of sensor data obtained at the detection level can be used to determine different attributes of an object.
 - Different probabilistic methods are adapted to sample the data obtained from sensors to find a composite representation of the best hypotheses in a sliding window of time.
 - The fusion strategy is dynamic to take care of the transient changes with the object description.
3. Deep learning technique to learn data continuously tracked.
 - The deep learning method involves learning of data from different levels of information collected from the previous output. It uses complex machine vision approach to train the data from different cascade layers.

The principal objective here is to improve the detection by taking into account the evidence distribution over different propositions of class by using sensor fusion added by learning the model obtained at different layers to reduce false and misdetections observed.

As a first step toward full sensor fusion, this paper focuses on the real-time detection of objects in continuous images (or video) captured with the dashboard camera of a moving vehicle. For object detection, deep learning based machine learning methodologies is used. The training of the detection model was performed offline, but the detection can be performed in real-time on real-life camera data mounted on vehicle dashboard. The overall research objective is multiple sensor fusion for augmenting and enhancing the existing machine learning methodologies by aggregating information from multiple sensors. The following sensors are considered: RADAR based distance sensors, ultrasonic distance transducer, and image sensor or camera. For detection task, a learning model which can detect the following six objects: roads (along with lane dividing marks), pedestrians, bikes, cars is developed.

To achieve the above objective, the following steps are needed: (i) Train a deep-layered convolutional Neural Network (CNN) which can detect the above four objects from image data. (ii) evaluate real-life detection performance of the model trained in (i) using camera data that we will collect with our own instrumentation; (iii) detect

distance of obstacles using a 120m RADAR (iv) detect short range objects using ultrasonic transducer (v) perform a lower level sensor fusion with data from mm wave radar ultrasonic distance sensor in order to generate a composite data capturing short and long range object data with varying degree of horizon; (vi) feed the composite distance data in real-time to CNN model for improving object detection performance.

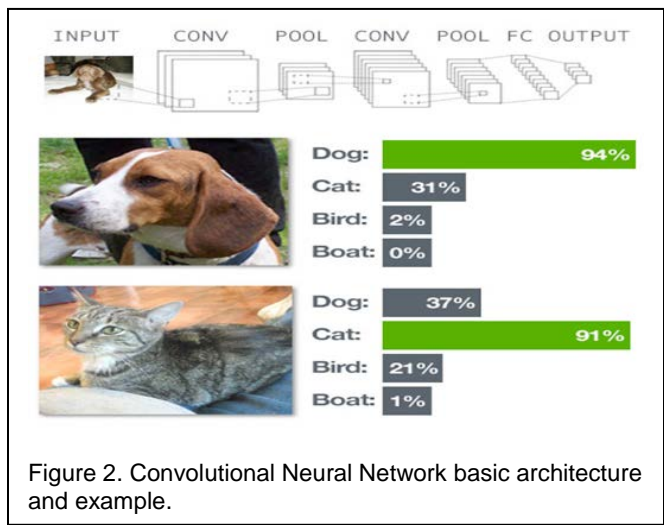
PRELIMINARY RESULTS

Convolutional Neural Networks (CNNs) has been applied successfully for deep learning algorithms for classifying two and three dimensional images, however, their application in ADAS and autonomous vehicle is still in infancy. The most recent work [35] which is somewhat concrete is an empirical evaluation of deep learning on real life driving data at real-time for two tasks: lane detection and vehicle detection. The conclusion of this work is that CNN can provide acceptable solution for the above tasks by running at frame rate which is required for a real-time system. In another recent work, authors use deep learning for detecting pedestrians [36], however, authors comments that although the performance of deep network is better than cascade algorithms for detecting complex patterns, it is slow for a real-time pedestrian detection. None of the existing works consider sensor fusion along with deep neural network, which is the focus of our work.

General principle of a traditional CNN algorithm is given below. CNN is a multi-layer perceptron which alternates between the convolutional and pooling layers for each input channel. It can be controlled by varying their depths and breadth, compared with similar sized layers CNNs have fewer connections and parameters to train. Convolutional layers consist of small learnable filters that convolve or slide over the input image. In the forward process we slide each filter across the width and height of the input image and calculate the dot product at a position. Pooling layer helps us dimensionality reduction of the input image which helps in computational complexity and also helps in reducing the over fitting. It also has fully connected layer which connects to the entire input volume as in ordinary Neural Network. Fig. 2 illustrates an example of CNN based object detection.

In this study, we used CNN for the detection of bike, car, pedestrian, and roads by using Google TensorFlow, a popular open source library for deep learning. The library supports multiple CPUs or GPUs along with different programming languages like Python, C++ etc. It contains implementation of CNN algorithm, which is highly generic; the learning model is represented as data flow graphs where nodes are called operations and each operation takes input as tensor and outputs as a tensor. User can set parameters for network architecture, and also parameters such as learning rate, batch size and number of iterations etc. for balancing under-fitting and over-fitting of the classifier. It also supports visualization of the graph in tensor board where we can see the variance between

the weights, based on that we can tune the initialization parameters.



Another key feature in TensorFlow is its support for pre-training through Inception, Google’s open source image classifier which is trained on 1.2 million images with 1000 different class labels over 2 weeks with 8 GPUs. Pre-training works like a transfer learning paradigm, such that the patterns learnt from Inception data can be transfers to an application that has its own set of training data instances. The retraining process leverages existing parameters that have learnt as part of Inception classification and hence using this saves substantial time in training. Besides, it helps in terms of training data need, an highly accuracy classifier can be built with less training data by exploiting the transfer learning paradigm.

SIMULATION EXPERIMENT SCENARIOS

In our setup with TensorFlow, we have trained the classifier in a single CPU with 4 core machine, having 4 classes such as bike, cars, pedestrian and road image datasets with retraining classifier. The input format of the input image is JPEG RGB colored. Below are the statistics of the datasets. It is to be noted that we have used only a few thousand images for training, yet we obtained a very satisfactory classification performance due to the transfer learning features of TensorFlow.

Table 1. Image datasets used for training the classifier

| Dataset | No. of images | Source |
|------------|---------------|---|
| Bike | 365 | http://www.emt.tugraz.at/~pinz/data/GRAZ_02/ |
| Cars | 8144 | http://ai.stanford.edu/~jkrause/cars/car_dataset.html |
| Pedestrian | 6555 | http://www.vision.caltech.edu/~dhall/projects/CRP/ |
| Road | 289 | https://www.cs.toronto.edu/~vmnih/data/ |

RESULTS

Once training is completed, we have tested our trained classifier on a set of hundreds images downloaded from Google and performed prediction. For all four tasks, all four classes of objects were detected with very high accuracy, with the car detection receiving the highest accuracy of up to 98%. It is to be noted that TensorFlow provides the probability of an image belonging to each of the four classes. Fig. 3 through Fig. 6 illustrate the probability of main object matching one of the four classes with the class name that was assigned the highest probability by the proposed algorithm.

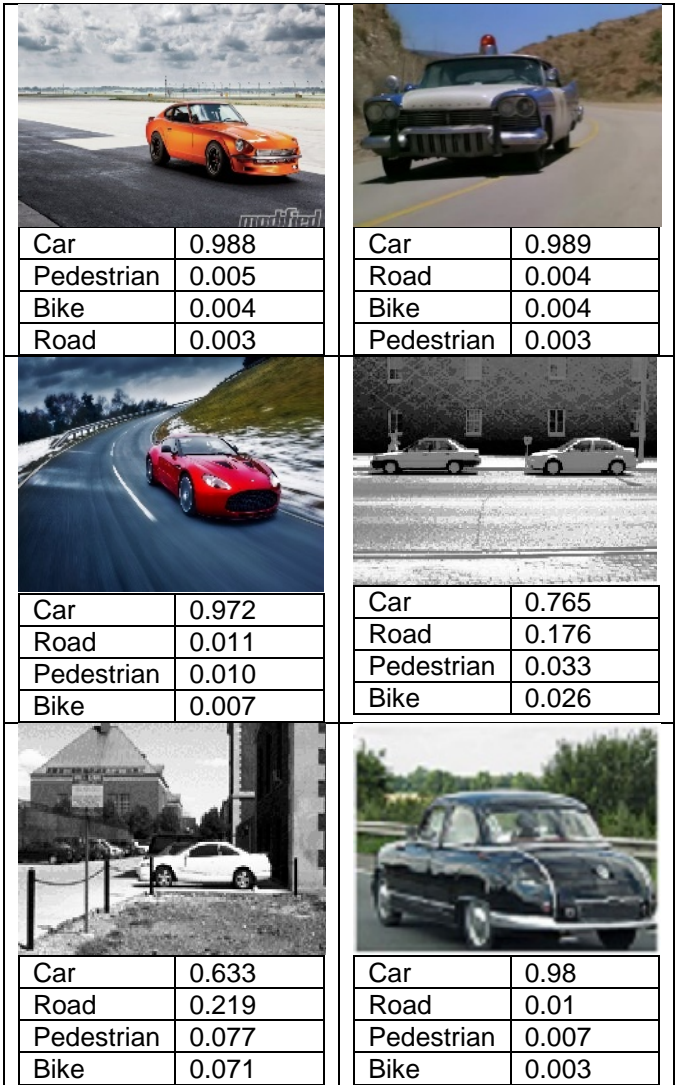


Figure 3. Accuracy of detecting a car with probability of matching listed under each image.

Fig. 3 shows six examples of test images labeled by “car” along with the classification output obtained by our model. The classification output is actually four probability values, each being the probability of the test object to belong to each of the four classes (Car, Road, Pedestrian, and Bike), where total sum of the probabilities should be 1. As we can see, the colored images provide higher level of confidence in detecting a car due to the better contrast of the foreground with the background. The least confidence

result is the fifth black-and-white image in which the white car is confused with apparently white road. Nevertheless for all the examples the top priority result is car by large margin.


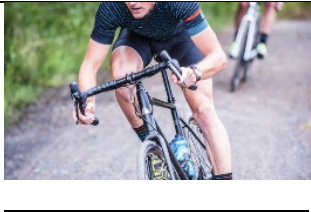
| | |
|---|---|
|  |  |
| Bike | 0.939 |
| Road | 0.033 |
| Car | 0.019 |
| Pedestrian | 0.009 |
| Bike | 0.99 |
| Car | 0.005 |
| Pedestrian | 0.005 |
| Road | 0.000 |

Figure 4: Accuracy of detecting a bike with probability of matching listed under each image.

Fig. 4 shows top two examples of classifying bicycle images. The examples show higher level of certainty in detecting a bicycle both with full view and also with partial view as shown on the image on the right.



| | |
|---|---|
|  |  |
| Pedestrian: 0.53 | Pedestrian: 0.65 |
|  |  |
| Pedestrian: 0.47 | Pedestrian: 0.6 |
|  |  |
| Pedestrian: 0.46 | Pedestrian: 0.45 |

Figure 5: Accuracy of detecting pedestrians with best probability of matching this class listed under each image.

Fig. 5 illustrates the probabilities of matching the image with the pedestrian class under various background and number of pedestrians. Overall detection accuracy is still very good since sum of probabilities for all four classes should be unity.





| | |
|--|---|
|  |  |
| Road | 0.939 |
| Pedestrian | 0.039 |
| Bike | 0.019 |
| Cars | 0.003 |
| Road | 0.953 |
| Pedestrian | 0.036 |
| Bike | 0.007 |
| Cars | 0.004 |
|  |  |
| Road | 0.612 |
| Bike | 0.331 |
| Pedestrian | 0.051 |
| Car | 0.006 |
| Road | 0.892 |
| Car | 0.057 |
| Bike | 0.041 |
| Pedestrian | 0.010 |

Figure 6: Accuracy of detecting road with lane markings with probability of matching listed under each image.

Fig. 6 shows four examples of classifying the road images. The probabilities of matching the image of road with that class are very high. The lower probability in the third image may be due to the presence of bright colored leaves on the road (confusing with a bike).

Table 2. Confusion Matrix for testing on Real time data

| | | Predicted Values | | | | Total | Accuracy (%) |
|-------------|------|------------------|------|------|------|-------|--------------|
| | | Car | Road | Bike | Ped. | | |
| True Values | Car | 20 | 2 | 0 | 0 | 22 | 90.9 |
| | Road | 2 | 10 | 0 | 1 | 13 | 76.9 |
| | Bike | 0 | 0 | 16 | 0 | 16 | 100 |
| | Ped. | 0 | 1 | 5 | 12 | 18 | 66.7 |
| Total | | 22 | 13 | 21 | 13 | 69 | 84 |

Table 2 represents the confusion matrix for the total 69 instances. It is observed from this matrix that object detection accuracy are excellent for car and bike classes while those for road and pedestrians are very good. The averaged accuracy of detection is shown in the bottom right corner of the table (84%).

CONCLUSION

Deep learning based (CNN) object detection is investigated in this paper. While the main goal is to improve the robustness and accuracy of moving object detection using multi-sensor fusion and deep learning, this research investigation provides the preliminary results of object detection using deep learning and only

image sensor data, as a first step toward multi-sensor based fusion with deep learning technique. The detection accuracy and robustness for all four different objects have been analyzed and the results look promising. The current detection algorithm will be extended to include sensor data from LIDAR/RADAR and ultrasonic distance sensor at the Mechatronics Research Lab at IUPUI.

REFERENCES

1. Adams, J.L., Remote Control with Long Transmission Delays. Stanford University, 1961.
2. Dickmanns, E.D., Dynamic vision for perception and control of motion. 2007: Springer Science & Business Media.
3. Braid, D., A. Broggi, and G. Schmiedel. The TerraMax autonomous vehicle concludes the 2005 DARPA grand challenge. in 2006 IEEE Intelligent Vehicles Symposium. 2006. IEEE.
4. Montemerlo, M., et al. Winning the DARPA Grand Challenge with an AI robot. in Proceedings of the national conference on artificial intelligence. 2006. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.
5. Funke, J., et al. Up to the limits: Autonomous Audi TTS. in Intelligent Vehicles Symposium (IV), 2012 IEEE. 2012. IEEE.
6. Markoff, J., Google cars drive themselves, in traffic. The New York Times, 2010. 10(A1): p. 9.
7. Rauskolb, F.W., et al., Caroline: An autonomously driving vehicle for urban environments. Journal of Field Robotics, 2008. 25(9): p. 674-724.
8. Berlin, T., Spirit of berlin: An autonomous car for the DARPA urban challenge hardware and software architecture. retrieved Jan, 2007. 5: p. 2010.
9. Hall, D.L. and A.K. Garga. Pitfalls in data fusion (and how to avoid them). in Proceedings of the Second International Conference on Information Fusion (Fusion'99). 1999.
10. Baig, Q., Multisensor data fusion for detection and tracking of moving objects from a dynamic autonomous vehicle. 2012, Ph. D. dissertation, University of Grenoble1.
11. Chavez-Garcia, R.O., Multiple Sensor Fusion for Detection, Classification and Tracking of Moving Objects in Driving Environments. 2014, Université de Grenoble.
12. Vu, T.-D., Vehicle perception: Localization, mapping with detection, classification and tracking of moving objects. 2009, Institut National Polytechnique de Grenoble-INPG.
13. Baig, Q., et al. Fusion between laser and stereo vision data for moving objects tracking in intersection like scenario. in Intelligent Vehicles Symposium (IV), 2011 IEEE. 2011. IEEE.
14. Grabe, B., T. Ike, and M. Hoetter. Evaluation method of grid based representation from sensor data. in Intelligent Vehicles Symposium, 2009 IEEE. 2009. IEEE.
15. Subramanian, V., T. Burks, and W. Dixon, Sensor fusion using fuzzy logic enhanced kalman filter for autonomous vehicle guidance in citrus groves. Transactions of the ASABE, 2009. 52(5): p. 1411-1422.
16. Chavez-Garcia, R.O. and O. Aycard, Multiple sensor fusion and classification for moving object detection and tracking. IEEE Transactions on Intelligent Transportation Systems, 2016. 17(2): p. 525-534.
17. Andrej karpthy, George Toderici, et al. Large-Scale Video Classification with Convolutional Neural Networks in CVPR '14 Proceedings, 2014 IEEE Conference on Computer Vision and Pattern Recognition
18. S.Lawrence, et al. Face recognition: a convolutional neural-network approach in IEEE Transactions on Neural Networks, 2002
19. Karen Simonyan, Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, at ICLR 2015
20. Dumitru Erhan, et al. Scalable Object Detection using Deep Neural Networks, at The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 2147-2154
21. Dan Ciregan, et al. Multi-column deep neural networks for image classification, at The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012
22. Alex Krizhevsky, et al. ImageNet Classification with Deep Convolutional Neural Networks, at Neural Information Processing Systems (NIPS), 2012
23. Richard Socher, et al. Convolutional-Recursive Deep Learning for 3D Object Classification, at Neural Information Processing Systems (NIPS), 2012
24. Pierre Sermanet, et al. Pedestrian Detection with Unsupervised Multi-stage Feature Learning, at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012
25. P. Viola and M. J. Jones, "Robust real-time face detection," International Journal of Computer Vision, vol. 57, no. 2, pp. 137-154, 2004.
26. R. Leinhardt and J. Maydt, "An extended set of haar-like features for rapid object detection," in Proceedings of the International Conference on Image Processing (ICIP '02), vol. 1, pp. I-900-I-903, IEEE, Rochester, NY, USA, September 2002.
27. T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: application to face recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037-2041, 2006.
28. Hinton, G. E., Training products of experts by minimizing contrastive divergence. Neural computation, 14(8):1771-1800, 2002.
29. Hinton, G. E., Osindero, S., and Teh, Y.-W., A fast learning algorithm for deep belief nets. Neural computation, 18(7):1527-1554, 2006
30. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. The Journal of Machine Learning Research, 11:3371-3408.

31. Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.
32. Richard Socher, et al. Convolutional-Recursive Deep Learning for 3D Object Classification, at *Neural Information Processing Systems (NIPS)*, 2012
33. Pierre Sermanet, et al. Pedestrian Detection with Unsupervised Multi-stage Feature Learning, at the *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012
34. Srinivas Suraj, Sarvadevabhatla Ravi Kiran, Mopuri Konda Reddy, Prabhu Nikita, Kruthiventi Srinivas S. S., Babu R. Venkatesh (2016), *A Taxonomy of Deep Convolutional Neural Nets for Computer Vision*, *Frontiers in Robotics and AI*, Volume 2.1.
35. B. Huval, et al. An Empirical Evaluation of Deep Learning on Highway Driving, *ARxiv.1504.01716v3*, 2015
36. A. Angelova, A. Krizhevsky, V. Vanhoucke, A. Ogale, D. Ferguson. Real-Time Pedestrian Detection With Deep Network Cascades, *Proc. of BMVC*, 2015

CONTACT

Sohel Anwar is the corresponding author of this paper. He is currently an Associate Professor and Graduate Chair in the Department of Mechanical Engineering at Purdue School of Engineering and Technology, IUPUI, Indianapolis, IN 46202. He received his Ph.D. in Mechanical Engineering in 1995 from the University of Arizona, Tucson, AZ. Dr. Anwar worked for Caterpillar, Inc. from 1995 to 1999 where he was involved in developing novel X-by-wire control algorithms for wheel loaders. In 1999, he joined Ford Motor Company / Visteon Corporation where he focused on developing advanced fault tolerant control algorithms for drive by wire systems. Dr. Anwar holds 14 US patents and several foreign patents. He has published over 110 papers in refereed journals and conference proceedings. His research interests include battery modeling and diagnostics, autonomous driving, sensor development among others. He may be reached at: soanwar@iupui.edu. Web: <http://et.engr.iupui.edu/~soanwar/>.