Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 3): pp. 4184-4192
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.03.432

Iot Based Car Detection System With Machine Learning

Akansha Gupta Department of Computer Science & Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand India, 248002 akanshagupta.cse@geu.ac.in

Poonam Verma Department of Computer Application Graphic Era Hill University, Dehradun, Uttarakhand India, 248002 pverma@gehu.ac.in

ABSTRACT

AI, in the form of neural network models supervised learning, can be helpful in finding answers to difficult questions. A mask geographic deep convolution networks (Mask RCNN) and a transfer-learning-based vehicular detection method are used to swiftly deal with accident compensation issues. The Organization for Safe Global Motor Traffic estimates that between 20 and 50 million individuals are wounded or 1.3 million are killed on the world's roads every year. Every year, traffic accidents cost around 2% of the GDP (Gross Domestic Product) of every country in the world. Systematically detecting accidents is the goal of smart incident detection technologies. The Internet of Things (IoT) facilitates the instantaneous transmission of data from a wide variety of interconnected devices, including sensors, to consideration was given such as first responders and law enforcement in the event of an accident. In order to do their job successfully the majority of the time and limit casualties in the case of a failure, smart hazard detection techniques must strike a balance between mechanization and autonomous and human surveillance and response. A system like this ought to be able to make up for human error. The damaged area of a vehicle is located using techniques, and the seriousness of the damage is estimated. Transfer learning, which takes advantage of pre-trained models for a different type of object recognition problem, has yielded very promising results. Results from our projections suggest that transfer training and L2 regularization can improve performance over fine-tuning in some situations.

1. INTRODUCTION

The insurance sector has put money into research and development of new technologies, including machine learning (AI) [1]. In a world where the number of automobile accidents is on the rise, insurance firms are losing millions of dollars predominantly due to "claims leakage." Replicating the underwriting process in the financial sector is only one of the many problems that can be solved with the help of artificial intelligence [2]. For this reason, insurers have been searching for ways to expedite the evaluation of damage and settlement of claims. Employing deep learning to the analysis of vehicle damage, however, presents particularly difficult obstacles that must be addressed in the construction of cutting-edge applications. In order to train a network, transfer learning necessitates a big dataset and much more computational time than other methods, yet it is a highly effective method for handling complicated things. This study concentrates on two difficulties in developing an effective model for realising a deep learning methodology for automobile final inspection: I the availability of automotive damage information for training, and (ii) the need to minimise computational effort.

Due to the niche nature of automotive final inspection, there are few publicly available databases of labelled photographs of car damage. The hardest part of creating a model is doing it with a limited amount of data. When the available data is too limited to build a CNN model, the classification technique becomes very challenging, as shown by the work of [3]. This issue can be resolved by the use of data preprocessing by actively gathering and labelling information found on the Internet [2, 4, 5, 6]. The most difficult task is also to shorten the time it takes to prepare a model. Conventional CNN models require numerous forward and reverse iteration to accomplish image supervised classification and determine the optimal weights again for network, both of which can take a significant amount of time. Using GPUs to execute this task could take several days and weeks. However, thanks to pre-trained Convolutional networks, which have already been taught on large test sets like that of an ImageNet database [7], the time required to train the network can be drastically reduced. With domain adaptation, we can easily extract their weights and employ their structures for additional specific tasks.

In addition, pre-trained Convolutional networks can be utilised in the feature extraction and tuning processes. As a result of the intensity variety, nevertheless, their frames are notoriously difficult to grasp. By figuring out how to zero in on the effect of tweaking a few hyper-parameters, Jeffrey [8] was able to make significant progress toward improving the model. Whenever the validating performances converges to the proper predicted values, the best learning measurements are obtained using the train a model with k-epochs and assessing its effectiveness learning technique. To combat the generalization error issue and the impact of hyper-parameters, regularisation can be fine-tuned in this way [9]. Since 2014, when large amounts of data and computational power were made available through transfer approaches [8, 10, 11, 12] and the knowledge of how to adjust the VGG model by means of batch normalisation [13], deep learning has performed exceptionally well in picture classification. Using a transfer learning framework, [14] provides a thorough overview of deep learning. When it comes to localisation and identification, but not the calculation of strength level, an end-to-end approach with transferable learning on Cnns on an Image dataset was suggested in [15]. Similarly, in studies [2, 5, 16], researchers trained a Network model employing backpropagation algorithm and supervised learning, evaluating its performance on an Imagenet with an emphasis on fault diagnostic efficiency by contrasting its fine-tuned results to those of a previously trained CNN model. To enhance their damage identification capabilities, Mahavir Dwivedi [4] used the YOLO image classifier [17] to training and recognize the damages region. When it's all said and done, most articles only concern themselves with studying CNN models for damage detection via transfer learning.

In this work, we employed the VGG model that had already been trained [11] for both VGG16 and VGG19. With respective error rates of 7.32 and 25.32 percentage points, VGG16 and VGG19 took first and second place, respectively, in the features extraction and text categorization in the 2014 ILSVRC competition. Fully connected layers with a narrow input patch (3x3), a max-pooling frame of 2x2, and three fully-connected layered are their distinguishing features. Their completely connected layer topologies are identical to AlexNet's [10]. Even though their final pooling layer's outputs extracted features have the same height, they can span a wider region of interest than AlexNet. Their sample weights can be loaded and used without cost for our purposes. On top of that, using them saves us the trouble of adding annotation spaces to our freshly collected data. With the help of selenium, we physically gather photographs of damaged and intact cars from throughout the Internet, and then we use ensemble learning to make the datasets more comprehensive. However according Libertymutual.com, we need to classify the extent of the harm to each component as either slight, medium, or serious. [18]. We then use our algorithms with learning algorithms and L2 regularisation to shorten training times and alleviate the generalization error issue. There is no need to worry about selecting features explicitly while using L2 regularisation [19]. Afterwards, we use fine-tuning to tweak a few high energy in our algorithms so that we can evaluate their performance alongside that of domain adaptation and L2 regularisation to determine which is best for our system. Simply put, our system's damage identification, localization, and severity showings are higher when we use transfer learning and L2 regularisation rather than fine-tuning.

The rest of this paper is organized as follows. In section 2, we describe our proposed methods. The experimental results are reported and discussed in section 3, and finally, the conclusion and future work are presented in section 4.

2. PROPOSED SYSTEM

In this article, we discuss the idea of using technology to automatically detect accidents and keep tabs on a patient's vital signs inside an ambulance. Through a straightforward embedded-vehicle configuration, we can reliably identify accidents. If an incident happens, the vehicle's accident detection system will promptly notify emergency services and relay the precise location of the collision. The ambulance would be dispatched as soon as the directions to the scene of the accident are received. This type is highly adaptable and can be used in a variety of settings. Assessing the intensity of the accident, as shown in Fig. 1, provides insight into the extent of the damage. When an accident is classified as "minor," it is usually because of anything relatively trivial, such a small hole or a simple rupture. In addition, if the rate of significant accidents in a given region is notably greater than average, you might assume that the incident was particularly devastating. The accident rate data is sent to remote servers.

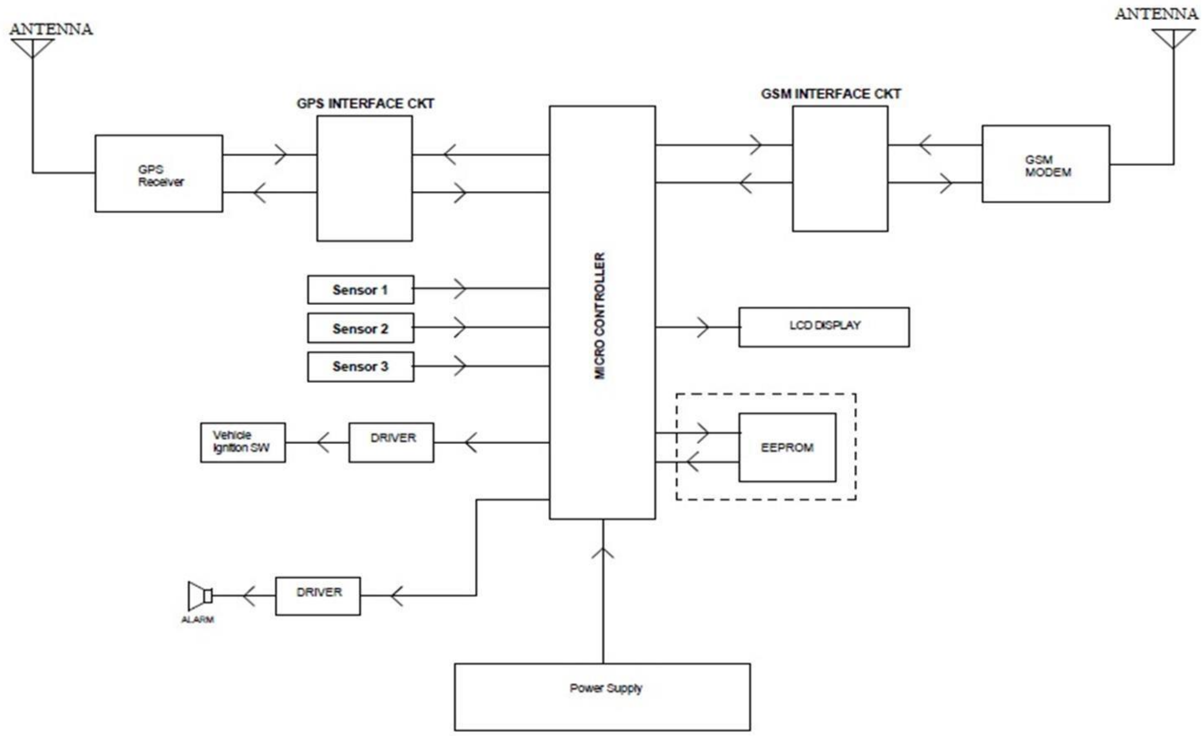

Figure 1: System Architecture

2.1 Dataset Description

Relying on the need for labelling data, the process of data collection can be fairly lengthy. As an alternative, we employed pre-trained VGG, which only needs the raw photos as input, thus we didn't have to cleanse the data. Since it is computationally costly to learn VGG for a little while, we cannot utilize cross-validation to assess our predictions. As a result, we randomly divided the dataset into two parts: one for training (80% of the total) and one for validation (20%). Due to the time-consuming nature of building and training with an ensemble of models, we instead chose to randomly place our train and validate sets. A variety of training splits were tried, but this one ended up being the most effective. In the end, the training and testing sets were divided up similarly. We used 1170 images of wrecked cars to construct our three datasets. Having the three datasets was crucial to our classification technique. Therefore, we individually gathered photographs of both damaged and intact cars from the web utilizing selenium.

2.2 Defining Damage Level

It was necessary to categorise all the injuries into mild, medium, and severe categories. There are three tiers of damage that can be found in a car [18]. Auto body damage is minimal if it only scratches the headlight or dents the hood. Gouges in the car's bonnet, fenders, or doors are rather sizable and constitute minor damage. Extreme damage, including shattered axles, twisted frameworks, and punctured airbags.

2.3 Data Augmentation

The quantity and reliability of input vectors for deep learning architectures can both be improved with the help of extracted features. In addition, the enriched data will correlate to a wider variety of available data points, which will result in a smaller gap between its training dataset and the validation set, in addition to the test images. Connor Shorten [6] carried out a study on extracted features, which is a response to the subset of data of incomplete data. This was done so that the implementation of their algorithms could be improved, and restricted datasets could be expanded so that they could take advantage of the capabilities of big data.

We utilize data enrichment to intentionally enlarge and alter our tiny datasets in order to improve their accuracy and reduce their sensitivity to the overfitting problems while they are being trained. This is necessary because there are not enough datasets including car damage for training purposes. As a result, in order to differentiate the data that was generated, we give it some random rotating, zooming, dimension shifting, and flipping remodeling plans.

2.4 Pre-trained VGG

Networked data for neural network models can be expanded and improved through data preprocessing. Additionally, the disparity between training dataset and the testing accuracy, and also the testing set, will be reduced because the enhanced data will correlate to a much more comprehensive range of available pieces of data. To better the performance of their algorithms and to increase limited databases so that they can profit from the capabilities of big data, Connor Shorter [6] researched backpropagation algorithm, a response to the large problem of limited evidence.

Due to the scarcity of available training datasets unaffected by vehicular accidents, we enhance our limited data sets with new information in order to boost their capabilities and reduce their susceptibility to overfitting. Consequently, we employ it to randomly rotate, zoom, dimension transfer, and reverse remodeling designs to differentiate the collected information.

2.5 Transfer Learning

It is common practice to use computer vision to build a Classification algorithm using data from a fixed subspace with the same distributions. There is a common belief that the feature regions of learning, verification, and test data should all match the distributions of the task at hand. On the other side, this view may well not apply, and hence modeling has to be constructed from the beginning if characteristics and dispersion vary, how hard procedure to collect associated training examples and create it again. A possible answer to this problem is necessary trainings [12]. So, transfer learning can be useful for enhancing productivity on target activities when input and output data are comparable.

Because of the intensive training durations and the generalization error issue with our little datasets, we turned to enhance the learning experience, which we deployed to pre-trained VGG networks to address the aforementioned three tasks: classifying, prediction, and grouping. Most importantly, we didn't have to start from scratch when training neural networks; instead, we could simply adapt an already existing service to perform the desired tasks. If we used the weights of previously trained VGG networks, we might drastically cut down on training times. Even though the dataset was too tiny to train a Network model, it showed great progress in resolving classifying difficulties [3].

2.6 Influence of Hyper-parameters

Theoretically adjusting the hyper-parameters in light of diagnoses can improve performance on a subset of tasks. Good requirements of the learning algorithm (which aids in controlling the dynamic changes of our cable network weights), mini - batch (which signifies the number of training instances perpetuated and through system), as well as the extent of data advancement in our system is determined by keeping a record of the damage as well as measurement features throughout our training data. If you want a steadier knowledge acquisition at the expense of more cost, slowing down the instructional rate and increasing the batch size are two complementary strategies. According to other relevant works, fine-tuning characteristics yields excellent outcomes. Accordingly, we tweaked our networks' hyper-parameters by fine-tuning the last levels of which was before VGG models.

3. EXPERIMENT ANALYSIS AND DISCUSSION

The challenges in the studies include I recognizing cars or not, (ii) determining whether or not each pixel in an image of a vehicle is destroyed, (iii) pinpointing the exact place of the damages (in front, the back, or the sides), and (iv) rating the degree of the destruction. In order to tackle these two problems and complete the aforementioned objectives, we conducted the tests depicted in Figure 1 using the techniques we presented, and we will now go through the findings.

Table1: Performance Analysis of car damage assessment

Pre-trained VGG	Resultant of damage finding			Resultant of damage location			Resultant of damage severity		
	Precision	Recall	F1-score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
VGG-16	0.95	0.95	0.95	0.72	0.72	0.72	0.62	0.58	0.54
VGG-19	0.92	0.92	0.92	0.72	0.68	0.68	0.6	0.55	0.52

Table 2 Accuracy of car damage assessment

Pre-trained VGG	Resultant of damage finding			Resultant of damage spot			Resultant of damage Severity		
	Without L2	With L2	Fine-tuning	Without L2	With L2	Fine-tuning	Without L2	With L2	Fine-tuning
VGG16	0.9457	0.9457	0.9284	0.7031	0.744	0.7343	0.5339	0.5481	0.5269
VGG19	0.9457	0.9523	0.9087	0.704	0.7649	0.732	0.5732	0.579	0.5615

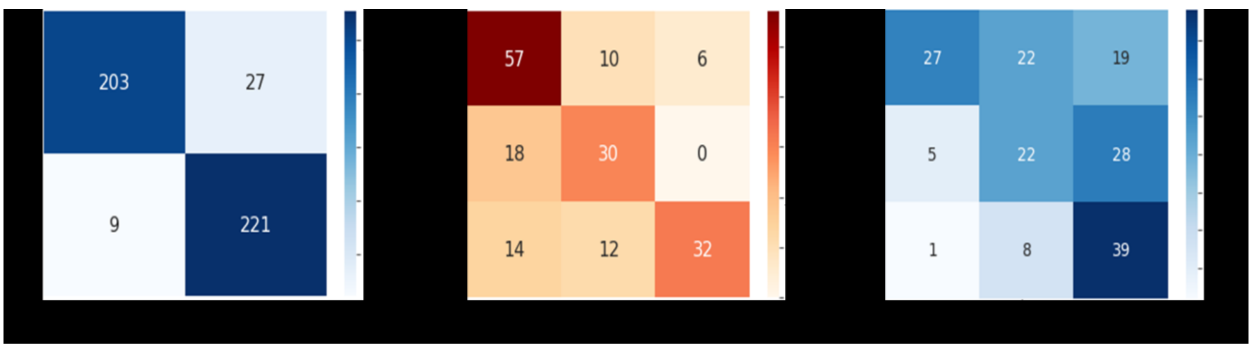

Figure 2: Confusion matrices for car damage assessment of VGG16

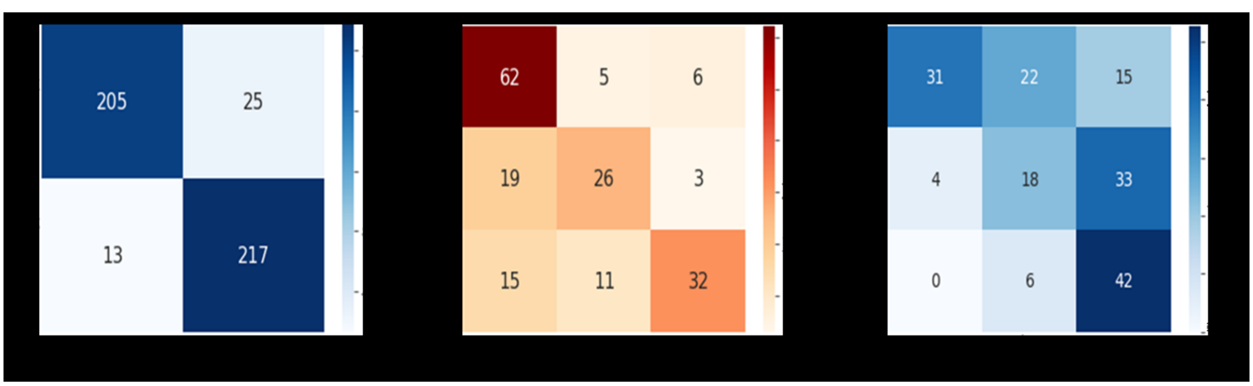

Figure 3: Confusion matrices for car damage assessment of VGG19

4. CONCLUSION

In order for intelligent accident detection systems to be able to do their jobs correctly the majority of the time and reduce the number of casualties that occur in the event that the system fails to operate normally, the framework must strike a balance between human surveillance and interference and mechanization and independence. These systems are designed to make up for human carelessness. The most effective accident detection methods are ones that are straightforward to set up and require a minimum amount of specialized hardware. The most significant barrier to the installation and widespread application of such systems is the necessity of expensive hardware, which may prevent such systems from being commercially feasible. In order to successfully apply this system, the communication links between private firms and humanitarian groups need to be of an exceptionally high caliber. This system has the potential to save countless lives if it is put into place by governments and commercial firms working together.

Reference

1. R. Girshick, ''Fast R-CNN,'' in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015. pp. 1440–1448.
2. S. Ren, K. He, R. Girshick, and J. Sun, ''Faster R-CNN: Towards realtime object detection with region proposal networks,'' IEEE Trans.Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/tpami.2016.2577031.
3. W. Liu, D. Anguelov, and D. Erhan, ''SSD: Single shot multibox detector,'' in Proc. IEEE Eur. Conf. Comput. Vision, Jun. 2016, pp. 21–37.
4. K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR), Jun. 2016, pp. 770–778.
5. K. He, G. Gkioxari, P. Dollar, and R. Girshick, ''Mask RCNN,'' in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2980–2988.
6. N. Kumar and R. Verma, ''A multi-organ nucleus segmentation challenge,'' IEEE Trans. Med. Imag., vol. 11, no. 1, pp. 34–39, Oct. 2019,doi: 10.1109/TMI.2019.2947628.
7. P. Pinheiro and R. Collobert, ''Learning to segment object candidates,'' in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 1990–1998.
8. W. Tang, H.-L. Liu, L. Chen, K. C. Tan, and Y.-M. Cheung, ''Fast hypervolume approximation scheme based on a segmentation strategy,''Inf. Sci., vol. 509, pp. 320–342, Jan. 2020, doi: 10.1016/j.ins.2019.02.054
9. Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, ''Fully convolutional instanceaware semantic segmentation,'' in Proc. IEEE Conf. Comput. Vis.Pattern Recognit. (CVPR), Jul. 2017, pp. 4438–4446

10. X. Rong, C. Yi, and Y. Tian, ‘‘Unambiguous scene text segmentation with referring expression comprehension,’’ IEEE Trans. Image Process., vol. 29, pp. 591–601, Jul. 2019, doi: 10.1109/tip.2019.2930176.
11. Y. L. Qiao, M. Truman, and S. Sukkarieh, ‘‘Cattle segmentation and contour extraction based on mask R-CNN for precision livestock farming,’’ Comput. Electron. Agricult., vol. 165, Oct. 2019, Art. no. 104958, doi: 10. 1016/j.compag.2019.104958
12. C. Shuhong, Z. Shijun, and Z. Dianfan, ‘‘Water quality monitoring method based on feedback self-correcting dense connected convolution network,’’ Neurocomputing, vol. 349, pp. 301–313, Jul. 2019, doi: 10.1016/j.neucom.2019.03.023
13. Nilaiswariya, R., J. Manikandan, and P. Hemalatha. "Improving Scalability And Security Medical Dataset Using Recurrent Neural Network And Blockchain Technology." In 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), pp. 1-6. IEEE, 2021.
14. Sriram, S., J. Manikandan, P. Hemalatha, and G. Leema Roselin. "A Chatbot Mobile Quarantine App for Stress Relief." In 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), pp. 1-5. IEEE, 2021.
15. Leema, Roselin G., K. Kiruba, P. Hemalatha, J. Manikandan, M. Madhin, and Raj S. Mohan. "Revolutionizing Secure Commercialization In Agriculture Using Blockchain Technology." In 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), pp. 1-6. IEEE, 2021.
16. Hemalatha, P., J. Manikandan, G. Leemarosilin, and P. Kanimozhi. "SEA FOOD PROCESSING USING INTERNET OF THINGS AND CLOUD TECHNOLOGIES." PalArch's Journal of Archaeology of Egypt/Egyptology 17, no. 9 (2020): 5877-5885.
17. V. R, R. S, V. K and S. P, "Integrated Communal Attentive & Warning System via Cellular Systems," 2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC), 2022, pp. 370-375, doi: 10.1109/ICESIC53714.2022.9783481.
18. V. K, K. Jayashree, V. R and B. Rajendiran, "Forecasting Methods and Computational Complexity for the Sport Result Prediction," 2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC), 2022, pp. 364-369, doi: 10.1109/ICESIC53714.2022.9783514.
19. K. Vijay, R. Vijayakumar, P. Sivaranjani, and R. Logeshwari, “Scratch detection in cars using mask region convolution neural networks,” Adv. Parallel Comput., vol. 37, pp. 575–581, 2020.
20. R. Vijayakumar, K. Vijay, P. Sivaranjani, and V. Priya, “Detection of network attacks based on multiprocessing and trace back methods,” Adv. Parallel Comput., vol. 38, pp. 608–613, 2021.