Car Damage Analysis and Insurance Claim System

Gaurav Gunjal, Harshada Sapkal, Pooja Kulkarni, Snehalata Nandanwar

***Abstract*–** Vehicle insurance claiming is an important area in vehicle industry. In this paper, we consider the problem of damage identification and damage classification of the various parts of the car. For this, we explore various deep learning techniques through which we can achieve image processing for damage identification, like Convolutional Neural Network and Transfer Learning. Initially, we test the accuracy of both the techniques with same amount of training image dataset. Experimental results show that transfer learning works better than convolutional neural network. Then we work for various techniques viz. fetching car and its owner information from RTO which can eliminate frauds. After that, insurance cost of each part will be shown to user and all the generated data will be reflected in a pdf file.

Keywords – deep learning, convolutional neural network, transfer learning

1. INTRODUCTION

Today, there is a lot of manual human intervention in vehicle industries [1]. A general vehicle insurance claim process starts with manual inspection of damaged car by the insurance agent. After this visual inspection and validation, insurance agent gives the final insurance amount. Due to this manual inspection, a lot of time get wasted and delays can be generated in insurance claim process[3]. Also, a lot of manpower may be required depending on the pending insurance cases.

Previous version of the system was the online web application, where user could upload the image of his car, and system used to identify whether the car is damaged or not. Previous system was a simple image classifier was used which classify images in two categories i.e. damaged and undamaged[2]. This image classifier was created with the help of Convolutional Neural Network, which is a deep learning technique. But due to limited training dataset, accuracy of previous system was very less. Also, there were only two classes for classification i.e. damaged and undamaged. System could not identify which parts of the car were damaged and which parts were undamaged.

Also, there was no provision made in previous version of system which can detect frauds from the user side. User could create the false insurance case and claim the insurance money. Below is the diagram of basic overview of previous system.

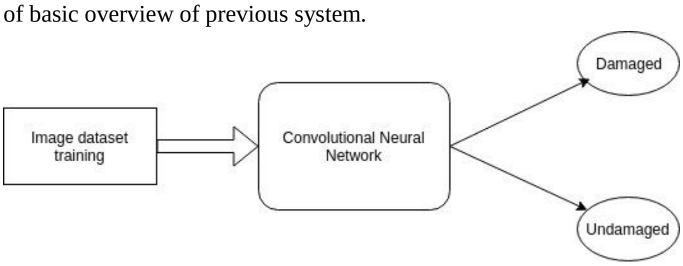

Fig 1: Previous System

2. CONVOLUTIONAL NEURAL NETWORK

Image classification is an important technique in artificial systems, and has large amount of interest over the last decades. This field has goal to classify images into different classes. Now a days, most researches have depends upon hand-crafted features, HOG[3] or SIFT[5] to describe an image in a different way. Then, learnable classifiers, like SVM, random forest, decision tree are applied to make final decision. If you give lot of images are then it becomes difficult to extract features from those. That's why deep neural network model comes to in picture. A few years ago, Hinton et al. exposed deep belief nets, in that each layer is trained layer by layer. In this type of deep learning contrastive divergence (CD)[10] is used, in that each layer is trained layer by layer. Because of deep learning, it becomes easy to represent hierarchical nature of features using many layers and corresponding weights. Whenever the input is in large amount, the deep network takes long time to train. At that time, CNN solved this problem. It improved classification performance on different kind of dataset. From above, it is noted that to train the following fully connected layers, features were used from the top layer only[7]. The Pierre et al. bridged between the lower layer's output and the classifier. It takes global shape and local details into account. Because of multi-stage features accuracy of system is improved. It uses single stage feature on multiple tasks, like pedestrian detection. In this paper, we propose alternative technique which can be applied on CNN whose weight parameter is fixed after training has been finished. The experimental results show that our approach improve performance of CNN.

3.1 OVERVIEW OF CNN ARCHITECTURE

CNNs are one of the feedforward networks. In CNNs the flow directed towards the outputs from the inputs. CNNs are biological inspired like an artificial neural networks (ANN). The visual cortex in the brain gives motivation for CNNs architecture[6].CNN architectures divided into many layers; but, generally, they consist of convolutional and pooling layers. These layers are grouped into modules. Either one or more fully connected layers follow these modules. Most of time, to form a deep model, modules are arranged on top of each other[5]. The CNN architecture for a toy image classification is shown in figure 1. The input image is directly given to the network, which followed convolutional and pooling stages. Then, representation from these operations gives to one or more fully connected layers[7]. The end fully connected output layer outputs the class label. It is the most popular base architecture found in literature. By taking care of image classification accuracy and reducing computation costs, several changes in architecture have been proposed from recent years.

We train a model using total of 3000+ car images by the Convolutional Neural Network technique. As it require to define and create each neural network layer manually, it is very costly process in terms of computational speed and time required. For 3000+ images, convolutional neural network took 18-19 hours approximately to complete its training[5]. Which was not suitable for our purpose, because in future, we might require to train with thousands of images, which will take few days to few weeks to train.

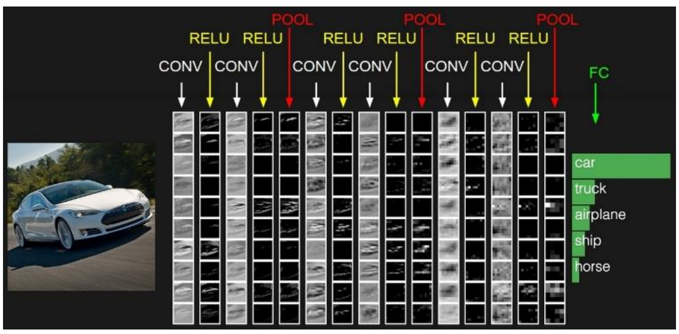

Fig 2: Convolutional Neural Network Model

3. Transfer learning

In many engineering areas like classification, regression and clustering, the machine learning[1] and data mining technologies have achieved large amount of success. Most of time, machine learning methods work well under a common assumptions like the training data and testing data are drawn from same distribution and same feature space. For different kind of classification, machine learning need to start from scratch, requiring users to collect large amount of training and testing data again[9]. This re-collection of training and testing data and re-trained models is the expensive process for the most of real world applications. If the re- collecting and reliable training and testing data step is neglect then time is reduced, and process becomes faster. For that purpose, transfer learning or knowledge transfer between task domains would be necessary. In transfer learning[10], the knowledge learned in one or more source tasks is transferred which is used to improve the learning of related target task. The development of transfer learning algorithms leads to growing an interest in machine learning community. Most machine learning algorithms are designed to address single task.

Due to the heavy cost of training neural networks with Convolutional Neural Network, we try Transfer Learning as well, to check if we get to see any improvement in training[11]. We again train a neural network model with 3000+ images, but now using transfer learning. In transfer learning, we do not require to define and create each layer manually. Instead, we just import a readymade neural network model, VGG16 by Oxford in our case, and then just modify required layers[8]. This result in better training speed and better accuracy. Where convolutional neural network took 18-19 hours to train 3000+ images, transfer learning did its job in just 2-3 hours, and with better accuracy.

3.1 Why Transfer Learning

Many deep neural networks trained on natural images show a curious occurrences in common: on the first layer they learn features almost same as Gabor filters and color blobs. Such first-layer features appear not designed only for a particular dataset or job but are general in that they are related to many datasets and tasks[11]. As finding these standard features on the first layer seems to happen without any concern about the exact cost function and natural image dataset, we call these features general. On the other hand, we call the last-layer features specific. In transfer learning we first give training to a base network on a initial dataset and task, and then we reprocess the learned features to the next target network to be trained on a final dataset and task[9]. This process will work nicely if the features are general, that is, suitable to both initial and final tasks, instead of being specific to the initial task. In reality, very few data scientists train a complete Convolutional Network from zero because it is very rare to have a dataset of such large size[4]. Instead, it is common to train a ConvNet in advance on a very huge dataset (e.g. ResNet, which contains 4.5 million images with 1000+ categories), and then use the ConvNet either as an starting point or a fixed feature extractor for the job of interest.

3.2 Transfer Learning Scenarios

Depending on both the size of the new dataset and the thing that's almost the same as another thing of the new dataset to the original dataset, the approach for using move from one place to another learning will be different[6]. .Keeping in mind that ConvNet features are more plain and common thing/not a brand-name drug in the early layers and more original-dataset specific in the later layers, here are some common rules of thumb for traveling safely through the four major pictures/situations :

1. The target dataset is small and almost the same as the base Training dataset. Since the target dataset is small, it is not a good idea to finetune the ConvNet due to the risk of overfitting[2]. Since the target data is just like the base data, we expect higher-level features in the ConvNet to be clearly connected with or related to this dataset also. Hence, we :
◦ Remove the fully connected layers near the end of the pretrained base ConvNet
◦ Add a new fully connected layer that matches the number of classes in the target dataset
◦ Randomize the weights of the new fully connected layer and freeze all the weights from the pre-trained network
◦ Train the network to update the weights of the new fully connected layers[4]

2. The target dataset is large and almost the same as the base training dataset. Since the target dataset is large, we have more confidence that we won't overfit if we try to fine-tune through the full network. Therefore, we:
◦ Remove the last fully connected layer and replace with the layer matching the number of classes in the target dataset
◦ Randomly initialize the weights in the new fully connected layer
◦ Initialize the rest of the weights using the pre-trained weights, i.e., unfreeze the layers of the pre-trained network
◦ Retrain the entire neural network[5]

3. The target dataset is small and different from the base training dataset. Since the data is small, overfitting is a concern. Because of this, we train only the linear layers. But as the target dataset is very different from the base dataset, the higher level features in the ConvNet would not be of any relevance to the target dataset. So, the new network will only use the lower level features of the base ConvNet. To put into use this big plan/layout/dishonest plan, we:
◦ Remove most of the pre-trained layers near the beginning of the ConvNet
◦ Add to the remaining pre-trained layers new fully connected layers that match the number of classes in the new dataset
◦ Randomize the weights of the new fully connected layers and freeze all the weights from the pre-trained network
◦ Train the network to update the weights of the new fully connected layers[6]

4. The target dataset is large and different from the base training dataset. As the target dataset is large and different from the base dataset, we can train the ConvNet from scratch. However, in practice, it is helpful to initialize the weights from the pre-trained network and fine-tune them as it might make the training faster. In this condition, the putting into use is the same as in case 3[7].

4. Proposed System

In this research paper, we propose a system based on deep learning for the purpose of car damage identification and classification. We consider 10 car parts which are more prone to damage – door, hood, bumper, doorglass, windshield, mirror, roof, headlamp, taillamp and grille. There is no publicly available dataset of these parts for damage classification. So, we created our own dataset by downloading images from various online sources. We sanitize these images according to the classes we want for classification.

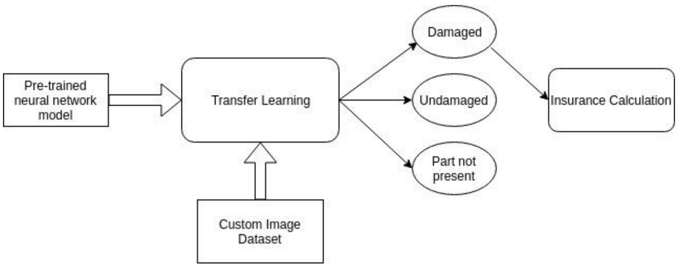

Fig 3: Proposed System

We create deep learning neural network models using these various datasets. We use two deep learning techniques i.e convolutional neural network and transfer learning side by side. This will allow us to see the comparison between both the algorithms in We came to know that transfer learning performs better than traditional convolutional neural network, in both the speed of training and required computational power. We observed that transfer learning gave better accuracy than convolutional neural network[3].

We are also working on fraud detection. In our system, we use vehicle registration number to fetch the owner and car information from RTO. Currently, we use Madhya Pradesh RTO website to fetch the vehicle details, as Madhya Pradesh RTO website doesn't have any Captcha, so we can programmatically fetch the owner and vehicle details just using vehicle registration number. There some paid APIs available which are also able to fetch all these details about owner and vehicle. Matching these details with any government identity of the owner can lead us to whether it is a fraud or not.

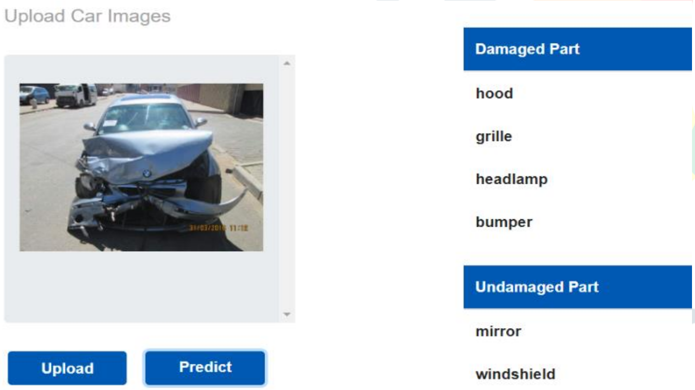

Fig 4: Damage prediction output

5. DATASET DESCRIPTION

Since there is not any readymade dataset which is available to public, we download all the available car damaged images from various online sources like Google Images, Bing Image Search etc. and sanitize these images according to the classes in which we want to classify the images i.e. damaged, undamaged and part not present. We collected approx. 3 to 4 thousand images and sanitized them in those three classes[8].

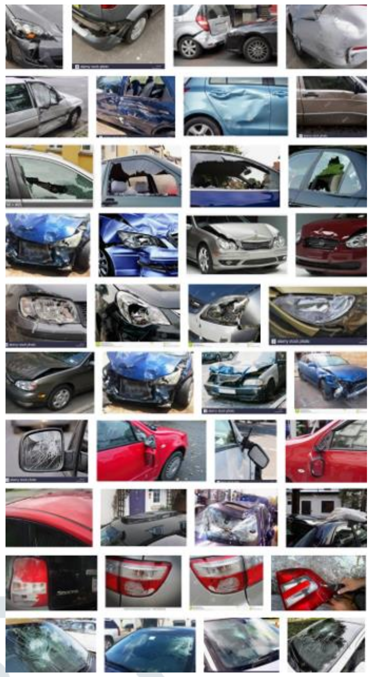

Fig 5: Dataset sample for each part

We create 10 different neural network model of 10 different car parts viz. door, hood, bumper, door-glass, windshield, mirror, roof, headlamp, taillamp and grille, by giving training with our collected dataset[9]. We tried two deep learning techniques for creating neural network models i.e. Convolutional Neural Network and Transfer Learning.

6. Fraud Detection

Frauds are the biggest concern in any insurance industry. This type of illegal dishonesty involves someone lying to an insurance company about a claim involving their personal or commercial motor vehicle. It can involve giving out sneaky and false information or providing false paperwork that proves the insurance claim[11]. Most automotive insurance illegal frauds arrests involve:

- staged car crashes and false claims of injury
- false reports of stolen vehicles
- false claims that the crash happened after a policy or coverage was bought.
- false claims for damage that already happened before.
- claimants who hid that a person kept out from coverage by their policy was driving at the time of the crash.

Insurance fraud has been in existence since the beginning of insurance as a big business[1]. Insurance crimes range in seriousness level, from stating that something is much bigger, worse, etc., than it really is, claims in a carefully-planned way causing accidents or damages. Insurance frauds causes a significant problem and the government is making efforts to discourage such activities. Insurance fraud cases claims cost the insurance industry a huge sum every year. About 90 percent of auto insurance illegal cases is the result of claims padding, which means to add damages, injuries and fake passengers to false insurance claims[3]. The other 10 percent of insurance fraud cases comes from organized car crash staging. Innocent victims like private car drivers, truck drivers, etc. are targeted by organized vehicle crash rings. These rings make an collision happen by setting up innocent people for a rear-end crash. Reporting that your vehicle has been stolen by thieves when you hid it in the jungle is a good example of false insurance claim. Even if one never files an insurance claim, providing false information on the insurance application for insurance is still accountable for fraud cases. Using fake and morphed documents for claiming is also a fraud case[9].

Here are a few protective measures insurance companies are taking to spot insurance frauds activity and make those responsible pay.

- Doubtful Loss Indicators

According to the NICB (National Insurance Crime Bureau), there are certain things that are usually listed within an insurance claim that could possibly raise some suspicion. People submitting the fake insurance claims often don't think they will get caught by insurance company. If they make the poker face after submitting a complex claim or submit the handwritten receipts for car damage repair, the agent might research a little further.

- Claims History

Have you claimed multiple insurances in the past? You can't hold responsible agents for being doubtful. All people are capable of faking accidents or claiming the car was stolen. Each time you record a loss, it goes on your record list. Insurance companies are now capable of using neural network patterns, which could raise doubt.

- Use Private Examiner

It sounds like some fictional or movie scene. But just take a minute, and think about the billions of dollars companies lose due to false insurance cases. It makes sense to zoom in on the criminal who cheats insurance companies. Say, you just recorded an insurance claim for a heavy injury after your vehicle crash and are supposed to be wear crutches[8]. A private examiner will make sure you are in fact hurt by keeping an eye on your activities, speaking to neighbors, friends or relatives, or by digging up previous criminal records. Today it's lot easier with the help of social media like Facebook and Instagram, where actual information of claimant can be retrieved.

- Fake Insurance Companies

This is a very serious issue for both the car insurance company and the customer. Fake insurance policies are not that rare these days. To avoid these types of frauds, customer must make sure to verify the genuine car insurance company. Customer should do a little research, navigate to their website, verify the company by talking to some of their clients and read their reviews.

7. Conclusion

We have trained and tested more than 3000 images on both the approaches i.e. Convolutional Neural Network and Transfer Learning. Using CNN, trained model gave accuracy approximately 79%, where model trained with transfer Learning approach gave accuracy approximately 93%. Accuracy has improved drastically in Transfer Learning approach. Also, the number of trainable layers in the transfer learning model is low as compared to our convolutional neural network model. Apart from this, the CNN scratch model took around 18 hours to train on CPU, while the transfer model took less than 6 hours to train the model on the same dataset. We can conclude that the transfer learning technique is not only performance efficient, but also is computationally efficient. We have also implemented fraud detection in our system through which we can confirm whether the original owner of vehicle is claiming the insurance or not.

8. References

1. Waseem Rawat , Zenghui Wang, "Deep Convolutional Neural Networks ffor Image Classification," Neural Computation, Volume 29, Issue 9 ,September 2017 ,p.2352-2449
2. Manuel Lagunas, Elena Garces , "Transfer Learning for Illustration Classification", W Rawat, Z Wang – Neural Computation, 2017.
3. Agostinelli, F., Hoffman, M., Sadowski, P., & Baldi, "Learning activation functions to improve deep neural networks" Submitted on 21 Dec 2014 (v1), last revised 21 Apr 2015.
4. Bachman, "An architecture for deep, hierarchical generative models", Philip Bachman submitted on 8 Dec 2016.
5. Belkin, M., Niyogi, P., & Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples". Journal of Machine Learning Research on 12 Jan 2006 .
6. Bulo, S., & Kontschieder, "Neural decision forests for semantic image labelling" – Procedings of IEEE 2014.
7. Cheng, Z., Soudry, D., Mao, Z., & Lan, " Training binary multilayer neural networks for image classification using expectation back propagation"(2015).
8. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... Wang, "Recent advances in convolutional neural networks"(2015)
9. HE K., ZHANG X., REN S., SUN J."Deep residual learning for image recognition"
10. OQUAB M., BOTTOU L., LAPTEV I., SIVIC J. "Learning and transferring mid-level image representations using convolutional neural networks". In Computer Vision and Pattern Recognition IEEE 2014.
11. LIN Y., LV F., ZHU S., YANG M., COUR T., YU K." Largescale image classification: Fast feature extraction and svm training Neural Computation, Volume 29, Issue 9 , September 2017 .