An Anti-fraud System for Car Insurance Claim Based on Visual Evidence

Assistant Prof Harshitha G .M[1], Parikshith Adiga[2], Rasik Shetty[3], Sanath Shetty[4], Veekshith Shetty[5]

[1-5]Department of Computer Science and Engineering, Alva's Institute of Engineering and Technology

---

Abstract: Automatically scene understanding using machine learning algorithms has been widely applied to different industries to reduce the cost of manual labor. Nowadays, insurance companies launch express vehicle insurance claims and settlements by allowing customers uploading pictures taken by mobile devices. This kind of insurance claim is treated as a small claim and can be processed either manually or automatically in a quick fashion. However, due to the increasing number of claims every day, systems or people are likely to be fooled by repeated claims for identical cases leading to big losses to insurance companies. Thus, an anti fraud checking before processing the claim is necessary. We create the first data set of car damage images collected from the internet and local parking lots. In addition, we proposed an approach to generate robust deep features by locating the damages accurately and efficiently in the images. The state-of-the-art real-time object detector YOLO is modified to train and discover damage regions as an important part of the pipeline. Both local and global deep features are extracted using the VGG model, which are fused later for more robust system performance. Experiments show our approach is effective in preventing fraud claims as well as meet the requirement to speed up the insurance claim prepossessing.

1. Introduction

The anti-fraud or fraud detection system is coupled with the claim history database. Each user has their own record in the database after they have a claim for the first time. The fraud claim could happen in two ways, one is that the same vehicle re-claim and another is cross-vehicle reclaim. Same vehicle reclaim happens when the user tries to make a quick insurance claim by uploading a similar image which is collected during the same case. Another fraud claim would happen when the user uses a similar image taken from another car as his or her own vehicle claim. Both formats could lead to reclaim for the same cases that actually have been issuing the settlement. Traditional insurance claim is handled manually by representatives which is necessary when the claim value is above a certain threshold.

However, when the claim number increases each day and some of the claims are only about small damages such as scratches, dent etc. Manually handling all these claims decrease the efficiency of insurance company's service and increase the cost of staffing. In this case, a system that can help to detect the fraud claims is needed when we want to make the system more automatically. The user is required to upload several images following the requirement in order to open a new claim. The anti-fraud system will then use algorithms to extract features to search in the enrolled history database. If the system reports a high score suspicious of a certain claim, humans can take into the loop and manually retrieve the history to do the loss assessment. To make the system more accurate and fast, we need a robust feature which is based on accurate locating the damages in the image. We adopt two state-of-the-art real-time detectors for damage detection and propose an approach to form robust features for the anti-fraud searching. Anti-fraud system for car insurance claim based on visual evidence. Automatically scene understanding using machine learning algorithms has been widely applied to different industries to reduce the cost of manual labor. Nowadays, insurance companies launch express vehicle insurance claim and settlement by allowing customers uploading pictures taken by mobile devices. This kind of insurance claim is treated as small claim and can be processed either manually or automatically in a quick fashion. However, due to the increasing number of claims every day, systems or people are likely to be fooled by repeated claims for identical cases leading to big losses to insurance companies. Thus, an anti-fraud check before processing the claim is necessary. The creation of the first data set of car damage images collected from the internet and local parking lots. In addition, proposing an approach to generate robust deep features by locating the damages accurately and efficiently in the images. The state-of-the-art real-time object detector YOLO is modified to train and discover damage regions as an important part of the pipeline. Both local and global deep features are extracted using the VGG model, which are fused later for more robust system performance. Experiments show the approach is effective in preventing fraud claims as well as meet the requirement to speed up the insurance claim prepossessing. The whole system consists of two parts: a real-time damage detector to provide accurate damage locations in the picture and a deep feature extractor to generate combined global and local deep features for anti-fraud matching in the claim history database. The

system is evaluated with both data sets collected on the internet from three of the biggest search engines and local public parking lots.

Fully or semi-automatically insurance claim processing could be very useful in the insurance industry when handling small but more frequent insurance claims that under a certain amount of rate. It increases the speed of claim investigation and loss assessment. Some of the companies employ 3G mobile techniques to accelerate the investigation process by freeing the customers from submitting complicated claims in order to shorten the whole claim process up to eight hours. However, fraud claims for which the payment has been claimed more than one time could potentially cause loss to insurance companies. Manually validating on a large scale of claims cannot meet the speed requirement for the express claim process anymore and reporting duplicated claim candidates automatically before the settlement is needed. However, the solution remains a challenging task due to a number of factors. In pictures or video frames captured by mobile devices, the viewpoint and illumination conditions are not taken in a controlled environment Damage cannot be localized accurately and fast enough to generate robust local features for matching.

To claim an insurance, the insurance company sends an agent to verify the vehicle and to check the damaged part. Hence this includes a lot of manual work for the agent and the company. Hence, a web based user interface for insurance claims is introduced, making it safe from frauds for the insurance company.To create a model that will cut down the human involvement for verification of the damaged vehicle and will ensure the customer to apply for insurance claim from anyplace. An anti fraud system which will claim insurance only for the registered valid customer. For generating deep features by locating the damage accurately and efficiently in the images..

2. Related Works

In [1] Author David G. Lowe gives information on object recognition systems that have been developed that use a new class of local image features. The features are invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection. These features share similar properties with neurons in inferior temporal cortex that are used for object recognition in primate vision. Features are efficiently detected through a staged filtering approach that identifies stable points in scale space. Image keys are created that allow for local geometric deformations by representing blurred image gradients in multiple orientation planes and at multiple scales. The keys are used as input to a nearest neighbor indexing method that identifies candidate object matches. Final verification of each match is achieved by finding a low-residual least-squares solution for the unknown model parameters. Experimental results show that robust object recognition can be achieved in cluttered partially-occluded images with a computation time of under 2 seconds. The first step can be done in a number of ways. One approach would be to select the feature that had the highest density around it in appearance space, where the weightings on each dimension are equal. This is equivalent to selecting the feature that would maximize the likelihood of the data for our model, where it was the only part of the model and we drop terms on location and scale. This is the approach taken in the experiments. This approach only works, however, if there is a feature that is common to many of the images. Another approach would be to match some of the images in the dataset to other images in the dataset, and select the parts that tend to appear in the matchings. This approach and others are currently being investigated. The second step of the approach avoids constructing a model with loose variances, which will result in an increase in the number of significant matchings h for each image. To achieve this we utilize what the model has learned so far in order to transform the locations and scales of the features of each image into model coordinate space. With this information we can then sample from the dataset to initialize a new part for which its appearance, and the location and scale in model coordinate space have a high number of matches in the dataset. In [2] Author R. Girshick, J. Donahue, T. Darrell, and J. Malik give information on — Object detection performance, as measured on the canonical PASCAL VOC Challenge datasets, plateaued in the final years of the competition. The best performing methods were complex ensemble systems that typically combined multiple low-level image features with high-level context. In this paper, we propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 50% relative to the previous best result on VOC —achieving a mAP of 62.4%. Our approach combines two ideas: (1) one can apply high-capacity convolutional networks (CNNs) to bottom-up region proposals

in order to localize and segment objects and (2) when labeled training data are scarce, supervised pre training for an auxiliary task, followed by domain-specific fine-tuning, boosts performance significantly. Since we combine region proposals with CNNs, we call the resulting model an R-CNN or Region-based Convolutional Network. Object classification at first level of class hierarchy with highly dissimilar object classes has proven to be successful. With advances in CNN models, recently many researchers gained interest in exploring fine-grained classification, or fine grained visual recognition (FGVR), where the object classes are closer to each other in terms of visual features. This is a novel and challenging task of great significance. Furthermore, it is much more difficult to distinguish between classes at levels lower down in the hierarchy, i.e., sub-classes compared to classes at the first level. The difference across sub-classes is usually subtle and requires expert knowledge. The 16 or 19 layers VGG network has a uniform architecture and yields powerful performance. The entire network is composed of a series of small (3x3) convolutional layers with stride one. The advantage of the architecture is to increase the discriminative power of the rectified linear activation by having more layers (two or three layers as opposed to one, as in the case of 7x7 receptive field) followed by three fully-connected layers. VGG net is a computational heavyweight but provides high classification accuracy. After damage detection is generated, robust features need to be extracted to overcome lighting conditions and different angles of views. Also, both global and local features are being used to finally determine whether a claim has ever appeared in the history database as each claim requires at least one image of the image containing the close view of the damage as well as another image shows the whole view of the body of the corresponding vehicle . To extract local features, we employ the pre-trained VGG16 object recognition model as a feature extractor. Global feature consists of global deep features and color histogram. We fuse these features to obtain a more discriminative feature and conduct a 1 to N match between. The probe images and the images in the post-claim history database.

In [3] Author Duy Thanh Nguyen and Tuan Nghia Nguyen give information about Implementation of YOLO CNN for Object Detection. To reduce expensive external memory accesses, the resolution of number representation is reduced. They aggressively quantize the weight and the activation to a single bit. The multiplier-accumulator (MAC) operation is replaced with a low-cost pop-count computation, and the comparator in a max-pooling layer is implemented by an OR gate. Liang et al. report the need for the floating-point number for batch normalization to avoid severe degradation of the accuracy. This shows an example that the performance of the binary network is very poor for a challenging data set, such as ImageNet. For the implementation of YOLO, several FPGA designs have been proposed. Tincy YOLO, presented in, uses an extended version of the design to off-load 12 hidden layers to programmable logics in Zynq Ultrascale+ FPGA. The hardware accelerator processes these hidden layers one by one. Moreover, the first and last layers are run on software causing a low frame rate. Lightweight YOLO-v2 is proposed to combine a binary network with support vector machine (SVM) regression. The authors design a shared streaming binary convolutional circuit in which each layer is processed sequentially. Although these previous designs succeed in the speed up by reducing the complexity of the algorithm, they do not consider the reduction of external memory accesses. To avoid the frequent off-chip access for intermediate data or large interlayer double buffers caused by unoptimized data path in previous works, this paper proposes an efficient Tera-OPS streaming architecture design. YOLOv2 network is used for evaluating the performance of the proposed FPGA design in terms of both hardware performance and detection accuracy. The network is retrained and quantized using 1-bit weight and flexible low-bit activation. The convolutional layer is used to extract higher features from the input image. The convolutional computation in the input image, comprising N channels, is convolved with an M number of N-channel filters to produce an M-channel output image. Each kernel has a size of K × K. Algorithm 1 elaborates the convolutional operation in detail. For simplicity, the stride is assumed to be 1, and the bias is assumed to be 0. As a result, the output image has the same size as the input image. In batch mode, the buffer and convolution kernel increase linearly as the batch size increases. The next layer can start only after the entire output of the previous layer is computed. On the other hand, the proposed scheme requires smaller buffers and causes a smaller delay between the layers. The weight prefetching can be used to hide the latency of the weight read. It is also noteworthy that the proposed scheme does not incur hardware resource overhead in the batch mode. Therefore, the proposed line-based weight. reuse scheme outperforms the other schemes in terms of both hardware cost and performance. In [4] Author Zhimin Mo and Liding Chen give information about Yolo. YOLO is an object detection algorithm based on convolutional neural networks. Unlike the method of FasterRCNN, which generates proposal regions and classifies objects in the regions, it is a regression-based detection algorithm that obtains the position and probability of an object directly from all pixels of the entire picture. It can

realize end-to-end training and testing by using only one convolutional neural network, which is different from Faster-RCNN that uses at least two convolutional neural networks. Therefore, YOLO can greatly reduce training and testing time so that it can detect objects in real time. YOLO first divides the input image into S*S grids cell. Each grid cell predicts B bounding boxes that contain five predicted values, namely x, y, w, h and confidence. The coordinates x, y is the relative value of the center of the bounding box and w, h represents the ratio of the width and height of the bounding box to the width and height of the entire image. YOLO predicts the coordinates of the bounding box, confidence and the classes of the grid separately. It is hard to achieve the balance of the three aspects with simple sum squared error loss. YOLO adopts a weighted sum-squared error loss to solve this issue. Comparing the machine vision algorithm with deep learning, it can see that the accuracy of the two types of algorithms is not much different. However, the machine vision algorithm spends much more time than the deep learning algorithm in the detection, which will seriously affect the effectiveness of the production line.In addition, the variability of the production environment can affect the accuracy of machine vision recognition algorithms. It fully demonstrates the superiority of deep learning algorithms in the identification of solder joints in automotive door panels. In terms of MAP, YOLOv3 and Faster-RCNN are almost the same. Both of them have poor recognition of rectangle solder joints, suggesting that the characteristics of rectangular solder joints are not obvious. It is hard for them to learn the characteristics of rectangular solder joints well. When collecting the data set, several people participate in the label work. Owing to the small solder joints, the difference of the truth bounding boxes is a little large, resulting in a decrease in the recognition accuracy. But the classification and positioning can still achieve a high accuracy, which can satisfy the quality requirements of the production. As for detection speed, YOLOv3 is significantly faster than Faster-RCNN, and the detection time is about 4 times that of Faster-RCNN. Owing to the limitations of the experimental environment, the average detection time of YOLOv3 for solder joint identification is 0.177s/sheet. But it can still satisfy the production requirements of automotive door panel welding production lines and can be used for real-time detection.

In [5] Author Jeff Donahue and Jitendra Malik give information about recognizing objects and localizing them in images is one of the most fundamental and challenging problems in computer vision. There has been significant progress on this problem over the last decade due largely to the use of low-level image features, such as SIFT and HOG, in sophisticated machine learning frameworks. But if we look at performance on the canonical visual recognition task, PASCAL VOC object detection, it is generally acknowledged that progress slowed from 2010 onward, with small gains obtained by building ensemble systems and employing minor variants of successful methods. SIFT and HOG are semi-local orientation histograms, a representation we could associate roughly with complex cells in V1, the first cortical area in the primate visual pathway. But we also know that recognition occurs several stages downstream, which suggests that there might be hierarchical, multi-stage processes for computing features that are even more informative for visual recognition. The description of an object detection and segmentation system that uses multi-layer convolutional networks to compute highly discriminative, yet invariant, features. We use these features to classify image regions, which can then be output as detected bounding boxes or pixel-level segmentation masks. On the PASCAL detection benchmark, our system achieves a relative improvement of more than 50% mean average precision compared to the best methods based on low-level image features. Our approach also scales well with the number of object categories, which is a longstanding challenge for existing methods. Unlike image classification, detection requires localizing (likely many) objects within an image. One approach is to frame detection as a regression problem. This formulation can work well for localizing a single object, but detecting multiple objects requires complex workarounds or an ad hoc assumption about the number of objects per image. An alternative is to build a sliding-window detector. CNNs have been used in this way for at least two decades, typically on constrained object categories, such as faces, hands, and pedestrians. This approach is attractive in terms of computational efficiency, however its straightforward application requires all objects to share a common aspect ratio. The aspect ratio problem can be addressed with mixture models. A second challenge faced in detection is that labeled data are scarce and the amount currently available is insufficient for training large CNNs from random initializations.. The conventional solution to this problem is to use unsupervised pre-training, followed by supervised fine-tuning. The second principle contribution of this paper is to show that supervised pre-training on a large auxiliary dataset (ILSVRC), followed by domain-specific fine-tuning on a small dataset (PASCAL), is an effective paradigm for learning high-capacity CNNs when data are scarce. In our experiments, fine-tuning for detection can improve mAP by as much as 8 percentage points. After fine-tuning, our system achieves a mAP of 63% on VOC 2010 compared to 33% for the highly-tuned,

HOG-based deformable part model (DPM). Our original motivation for using regions was born out of a pragmatic research methodology: move from image classification to object detection as simply as possible. Since then, this design choice has proved valuable because RCNNs are straightforward to implement and train (compared to sliding-window CNNs) and it provides a unified solution to object detection and segmentation. In [6] Author Abhishek Verma and Yu Liu give information on object classification at the first level of class hierarchy with highly dissimilar object classes has proven to be successful. With advances in CNN models, recently many researchers gained interest in exploring fine-grained classification, or fine grained visual recognition (FGVR), where the object classes are closer to each other in terms of visual features. This is a novel and challenging task of great significance. Furthermore, it is much more difficult to distinguish between classes at levels lower down in the hierarchy, i.e., sub-classes compared to classes at the first level. The difference across sub-classes is usually subtle and requires expert knowledge. Fine-grained classification is in-depth computer recognition and it is challenging for two reasons. First comes from the task itself that the classes are similar and their differences are subtle, thus it is necessary to extract highly distinctive features to achieve good sub-class categorization. Typically, there are two sub-problems that need to be addressed: localization and feature representation of the distinctive parts. Second reason is it is known that not having enough training data leads to overfitting. Very large scale fine grained datasets are not publicly available, which could match the size of non-fine grained datasets such as MS Coco and ImageNet that are extremely large in terms of total number of images and classes. Relay backpropagation addresses the issues related to regular backpropagation in deep networks. As the network goes deep, parameter size and model complexity grows, that poses a great challenge for optimization. We encounter vanishing and exploding gradients phenomenon as the gradients might be prone to either being very large or too small as error is propagated across many layers. This could lead to degradation in the network. Such degradation amplifies as the network becomes deeper. The key idea of relay backpropagation is to encourage the propagation of error information in a manner that it goes back up to a certain layer. The introduction in one or more interim output modules (including loss layer) to selectively manage the back propagation of error in intermediate segments of the network, such segments comprise of fewer layers and are able to reduce the issue of degradation by minimizing the ensembles of losses.In [7] Author Cesar G gives information about CNNs that have been used in different areas; they are implemented for the recognition of gestures made by hand to differentiate between the open or closed hand gestures. Also, some studies found the applications aimed at detecting lanes in traffic zones and the detection and identification of traffic signals on road surfaces, where up to 85.58% accuracy is obtained in this task. Another of its applications is presented focused on the classification of tumors in digital mammography images and CNN's are applied in mobile robots in disaster areas for the identification of their surroundings. R-CNN is used only to detect the vehicle and have its location in the image, which is why the output categories in the network correspond to the car and the background. In the operation of each of the layers of a CNN is shown, where deconvnet techniques are implemented to visualize the behavior in the layers. In the training options of the R-CNN, a Learning Rate of 0.0001 is set, the batch size is set to 32, in this way there are 10 iterations per epoch, where each batch corresponds to 10% of the total size of the training database. Based on training tests, 200 epochs are established, being sufficient to obtain adequate learning in the network, obtaining high accuracy in the recognition and location of vehicles. This demonstrates the ability of CNN's to face different types of problems in which the detection and classification of objects or environments in images is required. For the detection of specific objects and their subsequent location in an image, techniques based on CNN have been developed, such as the R-CNN presented. They have been used on applications such as, for example, the detection of everyday and sports actions in people, where an accuracy between 82.3 and 94.2% in the classification of each action in their respective category is obtained, and the extraction of traffic signals in order to subsequently perform a processing to highlight their characteristics and to have a database with which to train the CNN. This seeks to evaluate the accuracy of CNN for the task of detecting scratches in cars, which may have applications in the automotive industry or video surveillance of vehicles. In the first instance, implementing an R-CNN, the region of interest (RoI) is detected, in this case, it is the car. Then the vehicle is extracted from the image and later the region is divided into multiple sections to be entered into a CNN, which detects which sections of the vehicle have scratches and which do not. The R-CNN is used only to detect the vehicle and have its location in the image, which is why the output categories in the network correspond to the car and the background. In the operation of each of the layers of a CNN is shown, where deconvnet techniques are implemented to visualize the behavior in the layers. In the training options of the R-CNN, a Learning Rate of 0.0001 is set, the batch size is set to 32, in this way there are 10 iterations per epoch, where each batch corresponds to 10% of the total size of the training database. Based on training tests, 200 epochs are

established, being sufficient to obtain adequate learning in the network, obtaining high accuracy in the recognition and location of vehicles. The architecture designed for CNN is shown, where a size of 200x200 pixel with three channels (RGB) is set as input, this input size is selected based on the dimensions of each of the vehicle sections. The architecture consists of 4 convolutional layers, 3 of Max Pooling and 3 Fully-Connected, for the normalization of the values obtained from the membership functions, just like the RCNN, a Softmax type function is implemented. An important part in the design of architectures is the generalization of the characteristics, i.e. CNN does not memorize the images to avoid over-training in the network, why a Dropout layer is used in the first two Fully-Connected at with a 50% of disconnection, in this way, half of the neurons are disconnected and a better generalization of the CNN is performed. In the training options of CNN, the Learning Rate is set to 0.0001. The size of the batch is set to 11, to reduce the load on the GPU and have better results in the optimization of the stochastic gradient descent as mentioned where it is advisable to keep values between 2 to 32 for the sizes of the batch. Based on training tests, 150 epochs are set, being enough to obtain results with a high accuracy percentage.

3. Dataset

We introduced a car damage dataset collected from one of three biggest searching engines: Google Bing and Baidu using keywords like 'scratch','dent'. This contains 1790 im-ages with manual annotation. In addition, we collect vehicle images with small damage in public parking lot without in-cluding the identification information of the vehicles. These data set has 92 cars each with four or five images and are also manually annotated with bounding boxes tightly bounds the damage regions. In order to mimic customer be-haviors, we use mobile devices to capture the most convinc-ing evidence for a successful insurance claim. In each case, we capture images of a whole car body and local scratch images from the same vehicle. Each of the cases contains two images captured the same damage regions which rep-resent the first claim and the other represents the repeated claim that treated as a fraud claim. Example images are shown in Figure 1 and Figure 2 with random samples from the dataset. The whole dataset is used for two purposes: one for damage detection training and evaluation, another for studying the whole system which contains both damage localization and fraud claim detection.

Figure 1: Example images from images collected from the internet

4. The anti-fraud system for car insurance claim

The anti-fraud or fraud detection system is coupled with the claim history database. Each user has their own record in the database after they have a claim for the first time. The fraud claim could happen in two ways: the same-vehicle re-claim or cross-vehicle reclaim. Same-vehicle reclaim hap-pens when the user tries to make a quick insurance claim by uploading a similar image which is collected during the same case. Another fraud claim would happen when the user uses a similar image taken from another car as his or her own vehicle claim. Both formats could lead to reclaim for the same cases actually has been issuing the settlement. Traditional insurance claim is handled manually by repre-sentatives which is necessary when the claim value is above a certain threshold. However, when the claim number in-creasing each day and some of the claims are only about small damages such as scratches dent etc. Manually handle all these claims decrease the efficiency of insurance com-panies' service and increase the cost of staffing. In this case, a system that can help to detect the fraud claims is needed when we want to make the system more automati-cally. The user is required to upload several images following the requirement in order to open a new claim. The anti-fraud system will then using algo-rithms to extract features to search in the enrolled history database. If the system reports a high score of suspicious of a certain claim, humans can take into the loop and manu-ally retrieve the history to do the loss assessment. To make the system more accurate and fast, we need a robust fea-ture

which based on accurate locating the damages in the image. We adopt two state-of-the-art real-time detectors for damage detection and propose an approach to form robust features for the anti-fraud searching. We will introduce each part in the following sections.

5. Methodology

Damage detection

In most of the cases, users are required to upload few images of the close view of the damages on the vehicle and another picture with a farther view that contains the whole body of the car as well as the plate which shows the model and identity of the car. The first step to obtaining a robust feature descriptor for fraud detection is the damage detec-tion. This task is challenging since it is not a traditional object detection problem and the damages would have dif-ferent forms. We consider the damage mainly three types due to analyzing our dataset: scratch, dent, and crack. Localization of a damage in an image uploaded by users is a critical part of our system as the system would obtain more stable and discriminative features when accurate de-tection happens. In addition, to meet our real-time require-ment without sacrifice detection accuracy, we employed YOLO, which is a state-of-the-art lightweight real-time de-tector to be part of the system. We proposed our approacht o train and evaluate on the new dataset we collect.

YOLO Detector

YOLO is a fast, accurate object detection framework. Its base network runs at 45 frames per second with no batch processing on a Titan X GPU and a fast version runs at more than 150 fps which allows us to process streaming video in real-time with less than 25 milliseconds of latency. It is extremely useful to assist needs in auto-drive, real-time responsive services. YOLO frame object detection as a sin-gle regression problem. Image pixels are mapped to bound-ing box coordinates and class probabilities straightly. As damages like scratch or dent might have a different shape, YOLO overcome this by learning features through regres-sion of four coordinates.YOLO, with its standard model, achieve 45FPS with mAP of 63.4 on VOC2007. All these performances meet the requirement of the efficient needs of an express anti-fraud system for insurance claims.it consists of global deep features and color histogram. We fuse these feature to obtain a more discriminative feature and conduct a 1 to N match between the probe images and the images in the post-claim history database.

SYSTEM WORKFLOW

The proposed system is implemented using the deep learning approach to detect the damage to the vehicle. Object detection tool (YOLO) is used for detecting the damaged part of the vehicle. By using Python In-Built libraries like OpenCV and PIL, we can find Root mean Square difference.

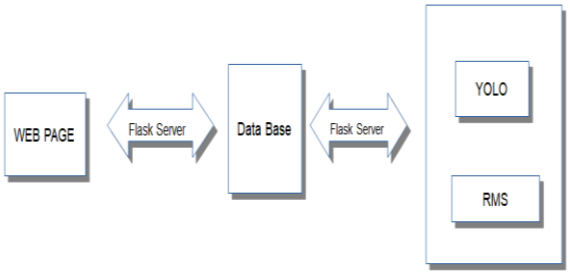

Figure 2: System Workflow

The best performing methods were complex ensemble systems that typically combined multiple low-level image features with high-level context. We propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 50% relative to the previous best result on VOC —achieving a MAP of 62.4%. Our approach combines 3 ideas: One can apply high-capacity convolutional networks (CNNs) to bottom-up region proposals in order to localize and segment

objects When labelled training data are scarce, supervised pre training for an auxiliary task, followed by domain specific fine-tuning, boosts performance significantly.

Since we combine region proposals with CNNs, we call the resulting model an R-CNN or Region-based Convolutional Network. The proposed system uses a VGG model which is a convolutional neural network model. VGG model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep ConvolutionalNetworksforLarge-Scale. ImageRecognition". The model achieves 92.7% top-5 test accuracy in ImageNet which is a dataset of over 14 million images belonging to 1000 classes. The input to the cov1

layer is of fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional layers, where the filters were used with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations, it also utilizes 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3×3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2×2 pixel window, with stride 2.

Three Fully-Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks. All hidden layers are equipped with the rectification (ReLU) non-linearity. It is also noted that none of the networks (except for one) contain Local Response Normalisation (LRN), such normalization does not improve the performance on the ILSVRC dataset, but leads to increased memory consumption and computation time. After the stack of convolution and max-pooling layer, we got a (7, 7, 512) feature map. We flatten this output to make it a (1, 25088) feature vector. After this there are 3 fully connected layer, the first layer takes input from the last feature vector and outputs a (1, 4096) vector, second layer also outputs a vector of size (1, 4096) but the third layer output a 1000 channels for 1000 classes of ILSVRC challenge, then after the output of 3rd fully connected layer is passed to softmax layer in order to normalize the classification vector. After the output of classification vector top-5 categories for evaluation. All the hidden layers use ReLU as its activation function. ReLU is more computationally efficient because it results in faster learning and it also decreases the likelihood of vanishing gradient problems.

PRE-PROCESSING

1. A R-CNN model requires the user to annotate the images and identify the region of damage.
2. The dataset has to be annotated for the different types of damages to train the model for detection.The annotation is used to train the model for future detection and analysis.

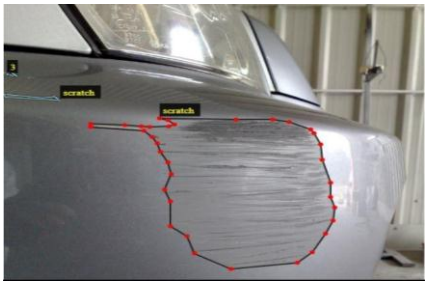

Figure 3: ANNOTATION SAMPLES

FEATURE EXTRACTION AND CLASSIFICATION

Feature extraction gets more intricate as we go through layers and hence 3x3 filter layer is used to further extract features before classification. Inspired by VGG's high performing architecture, a series of 3x3 convolutional layers and followed by

three fully connected layers. Both global and local features are being used to mine whether a claim has ever appeared in the history database as each claim requires at least one image of the image containing the close view of the damage as well as another image shows the whole view of the body of the corresponding vehicle. To extract local features, employ the pre-trained VGG16 object recognition model as a feature extractor. Global feature consists of global deep features and color histogram. The whole system consists of two parts: a real-time damage detector to provide accurate damage locations in the picture and a deep feature extractor to generate combined global and local deep features for anti-fraud matching in the claim history database.

PROCESSING THE INPUT

All the attributes are maintained in binary format. Whereas, the attributes of the premium dataset attributes are in different formats. It affects their integrity. So they are needed to transform in the unified format. Low level or primitive data are replaced by higher level concepts using generalization. Classification algorithms make use of a set of "features\attributes" or "predictor variables" for making predictions. The classification algorithms are driven by "class variables" that can be either "categorical" or "binary". "Classification" techniques are used in cases where the class label is categorical. The streaming architecture with pipelining enables the ability to run computation in batch mode to fully utilize the pipeline processing. Each layer is delayed by 1-to-4 input lines from the previous layer. Because the network is very deep, the delay from the first layer to the last layer becomes large. If a single frame is processed at a time, the first layers are underutilized for a definite period. The idea of batch processing is that, while the layers at the end of the networks run the first image, the first layers, which finish running the first image, can start running for the second image to utilize the idle period. The batch processing mode increases the throughput significantly.

6. RESULT ANALYSIS

Below figure shows a set of damaged car samples that are tested to validate the accuracy of the results that are generated.

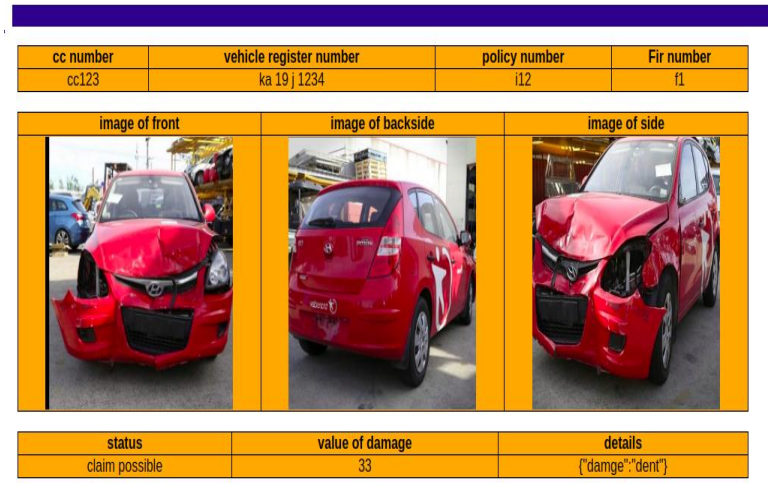

Figure 4: TEST SAMPLE FOR DAMAGED CAR

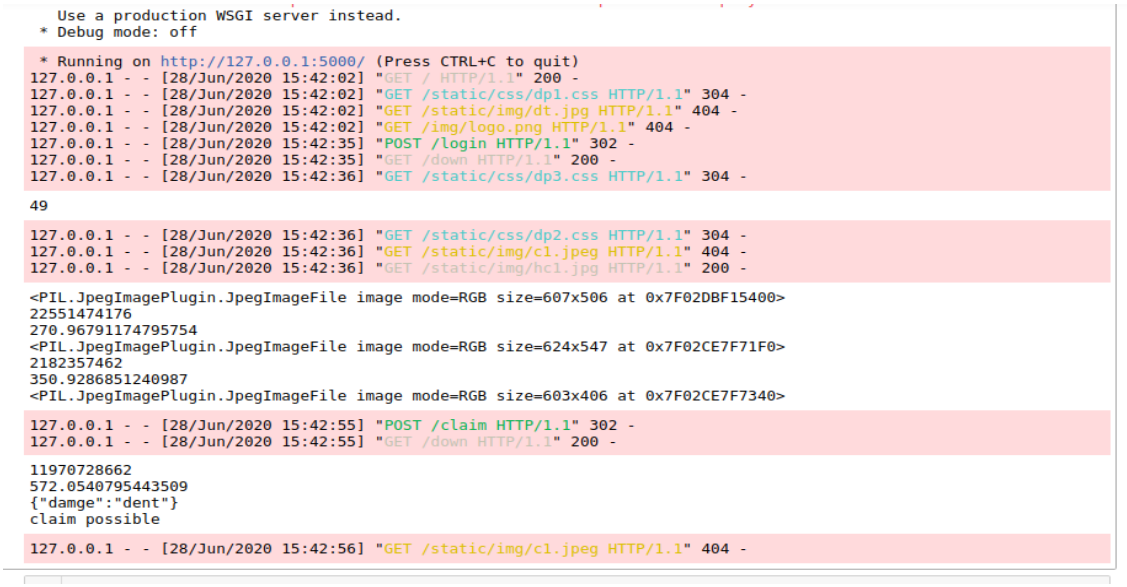

Figure 5: RESULT FOR DAMAGED CAR

Above figure shows to verify the system result obtained the test samples are manually opened and judged.

7. Conclusion and discussion

In this paper, we introduce a useful data set of car dam-age which is the first data set for car damage detection and matching. We proposed a practical system pipeline to de-tect the fraud claim before issuing the settlement in order to reduce loss for the insurance company. We research on different deep architectures for better damage detection result and analyze different feature fusions and its impact on the final matching performance.

8. References

1. Object Recognition from Local Scale-Invariant Features, David G. Lowe, Computer Science Department, University of British Columbia.

2. Region-based Convolutional Networks for Accurate Object Detection and Segmentation, Ross Girshick, Jeff Donahue,10.1109/TPAMI.2015.

3. A High-Throughput and Power-Efficient FPGA Implementation of YOLO CNN for Object Detection, Duy Thanh Nguyen, Tuan Nghia Nguyen, Hyun Kim, 1063-8210 © 2019 IEEE.

4. Identification and Detection of Automotive Door Panel Solder Joints based on YOLO, Zhimin Mo, Liding Chen, Wenjing You.978-1-7281-0106-4/19/$31.00 #2019IEEE

5. Hybrid Deep Learning Ensemble Model for Improved LargeScale Car Recognition, Abhishek Verma and Yu Liu,

6. Predicting Fraudulent Claims in Automobile Insurance, G.KowshalyaICICCT 2018,IEEE Xplore