Multimedia Tools and Applications (2024) 83:36719–36737
https://doi.org/10.1007/s11042-023-15906-9

1230: SENTIENT MULTIMEDIA SYSTEMS AND VISUAL INTELLIGENCE

Car crash detection using ensemble deep learning

Vani Suthamathi Saravanarajan[1] · Rung-Ching Chen[1] · Christine Dewi[1,2] · Long-Sheng Chen[1] · Lata Ganesan[3]

Received: 22 July 2022 / Revised: 15 February 2023 / Accepted: 18 May 2023 /
Published online: 30 June 2023

Abstract
With the recent advancements in Autonomous Vehicles (AVs), two important factors that play a vital role to avoid accidents and collisions are obstacles and track detection. AVs must implement an accident detection model to detect accident vehicles and avoid running into rollover vehicles. At present many trajectories-based and sensor-based multiple-vehicle accident prediction models exist. In Taiwan, the AV Tesla sedan's failure to detect overturned vehicles shows that an efficient deep learning model is still required to detect a single-car crash by taking appropriate actions like slowing down, tracking changes, and informing the concerned authorities. This paper proposes a novel car crash detection system for various car crashes using three deep learning models, namely VGG16(feature extractor using transfer learning), RPN (region proposal network), and CNN8L (region-based detector). The CNN8L is a novel lightweight sequential convolutional neural network for region-based classification and detection. The model is trained using a customized dataset, evaluated using different metrics and compared with various state-of-the-art models. The experimental results show that the VGG16 combined with the CNN8L model performed much better when compared to other models. The proposed system accurately recognizes car accidents with an Accident Detection Rate (ADR) of 86.25% and False Alarm Rate (FAR) of 33.00%.

Keywords Car crash recognition · Object detection · Deep learning · Convolutional neural networks · VGG16 · MobileNetV2 · InceptionResNetV2 · Resnet50

1 Introduction

Motor vehicle accidents are increasing day by day, causing deaths and disabilities in a large number [44]. The main reason for the accidents is human mistakes, unavailability of timely treatment, and secondary accidents [35]. Usually, these accidents happen from vehicle-to-vehicle collisions and pedestrians, single-vehicle accidents, and unnoticed accidents on one road ahead.

Hence, it becomes crucial to develop automatic accident detection methods to reduce fatalities and injuries. Such systems can help rescue agencies and vehicles around the accident by providing timely information in order to avoid road congestion and

✉ Rung-Ching Chen
crching@cyut.edu.tw
Extended author information available on the last page of the article

secondary accidents. Road perception, surrounding object understanding, and traffic detection techniques are the fundamental function of the autonomous vehicle to determine various driving rules. Thus, many companies and researchers are involved in developing algorithms for driverless systems to drive safely by identifying the obstacles present in the trajectory.

The path planning and traffic-driving directions of autonomous vehicles are built upon image recognition technologies. These technologies help in lane detection and locating the obstacle and/or objects present in a different road segment. It is to be noted that vehicle accident and rollover detection using image recognition algorithms are reliable and easy to deploy in real-time. A highly reliable accident prediction model is developed by analyzing previously available accident data. Thus, a traffic accident prediction model based on image recognition is a hot research area among academicians and corporate companies. According to the researchers, the six major factors which are the causes for majority of crashes are driver mistakes, traffic congestion, road geometry, weather conditions, vehicle conditions, and the time of the crash [1].

Besides the high-end sensors and advanced machine learning algorithms for segmentation and object localization [10, 39, 40], the safety of the autonomous vehicle is still a public concern. Many incidents show that the automated vehicle fails to recognize the objects on the road. An example of such a case is the Tesla sedan that ran into an overturned truck in Taiwan. Road segmentation and object recognition algorithms should improve recognition parameters like rollover, vehicle collisions, and accident vehicles.

The principal contribution of this research is as follows: (1). We propose a car crash detection system to detect car crashes in various car crash scenarios, making it suitable for autonomous vehicles' navigational decision-making process. (2). We designed a novel CNN8L with fewer training parameters to detect accident vehicles, improve the actual detection rate and reduce the false alarm. (3). We evaluated different state-of-the-art models using a transfer learning-based approach for feature extraction in a customized car crash dataset.

The rest of the paper is organized as follows: In Section 2, the proposed object recognition and detection technique are discussed in detail; Section 3 offers a description of the dataset and the experimental set-up and focuses on the evaluation of the recognition and detection results of the individual modules, Section 4 discusses on the final performance of the detection system, and Section 5 describes our conclusion and future works.

2 Related works

Ijjina et al. [22] proposed a framework for dealing with traffic accidents. The accident probability is calculated based on the observed speed and the anomalous pattern. The Mask R-CNN with centroid-based tracking is used for accident detection. Presented in [4] is a real-time algorithm to predict the pedestrian path to avoid accidents. Using Bayesian inference, the model identifies and learns the features and patterns of 2D trajectory data using both global and local characteristics. Their framework performed well on noisy and sparse data extracted from dense indoor and outdoor crowd videos.

In [11], the authors have suggested two deep learning techniques, gated recurrent unit (GRU) and convolutional neural network (CNN). These methods applied the ensemble technique with a weighted average on video and audio signals from dashboard cameras and multiple classifiers on these unstructured data to analyze the model prediction

ability. But it validated and compared the model efficiency only with video or audio data from single classifiers. DeepCrash, a deep learning model for the Internet of Vehicles (IoV), proposed by Chang et al. [7], includes different components and software modules in a single platform named in-vehicle infotainment (IVI) telematics platform. The self-collision detection sensor and the front camera data are passed to a cloud-based deep learning server for pattern analysis. A cloud-based platform manages the hardware and software components in the IVI platform.

A Cooperative Vehicle Infrastructure System (CVIS) combined with machine vision proposed by Tian et al. [48] to detect small objects, utilizes dynamic weights in the loss function and Multi-Scale Feature Fusion (MSFF) for improving the performance of automatic car accident detection systems. Rahim et al. [37] proposed a traffic crash severity prediction model based on deep learning techniques with a customized f1-loss function. He has compared its performance with other machine learning models using precision and recall indicators.

Lu et al. [29] proposed a deep learning framework to extract features from urban traffic video records using residual neural networks (ResNet) and attention modules. This feature fusion-based model achieved a high detection speed on par with accuracy for constrained computing resource systems. [36] developed a Deep Learning system capable of spotting accidents that are captured on video. The proposed method assumes that visual cues that appear over time can adequately describe traffic collision events. Therefore, a visual features extraction phase composes the model architecture, after which a temporary pattern is identified. The model is trained using various public datasets. This enables the model to learn the various visual and temporal characteristics during the training phase by using custom-built convolution and recurrent layers. While using the public traffic accident datasets, a 98% accuracy is achieved. Thus, we can see that independent road structures are detected efficiently [26].

The existing car crash detection models are efficient at predicting multiple vehicle collisions but fall short in identifying single-vehicle accidents. Many autopilot failures have been the cause for crashes involving repaired and accident vehicles. In [11], the authors proposed a car crash detection system using an ensemble method based on multiple classifiers for video and audio data from dashboard cameras. Also, they validated their study using YouTube videos of auto accidents. Bakheet et al. [3] proposed temporal templates of moving objects to extract local features. A deep neural network (DNN) model is trained on the extracted features to detect abnormal vehicle behavioral patterns and predict accidents before they happen.

In [53], the authors propose an accident detection approach based on spatiotemporal feature encoding with a multilayer neural network to cluster the video frames effectively and efficiently detect accidents from driving videos. The spatial relationships of the objects detected from these potential accident frames are then captured and encoded to determine whether these frames are accident frames. In [20], the authors suggest an ML framework based on multimodal in-car sensors for automated car accident detection using CNN and SVM classifier techniques to identify real-world driving collisions on the strategic highway research program (SHRP2) naturalistic driving study (NDS) crash data set, five different feature extraction method including ones based on feature engineering and feature learning with deep learning are assessed. In [34], the authors suggest an intelligent accident detection and rescue system that uses the Internet of Things (IoTs) and Artificial Intelligence (AI) to mimic the cognitive functions of the human mind. It uses a customized dataset created from a variety of online videos and gathers all accident-related data, such as position, pressure, gravitational force, speed, etc., and

sends it to the cloud. When the DL module notices an accident, it immediately alerts all nearby emergency services, including the hospital, police station, mechanics, etc.

An in-depth analysis of action recognition using the various methods, taxonomies, and algorithms for autonomous driving and accident detection is presented in [2]. This analysis focuses on different types of traffic video capturing, including dashcams, drone cameras, highway monitoring cameras, and stationary surveillance cameras at traffic intersections. The authors have advised future research to concentrate on scaling up accident detection systems for spontaneous detection to alert first responders about traffic accidents and provide a quick response to victims. Such model designs in AVs reduce the possibility of collision, automatic brake control, path planning and lowering the speed limit. Fully autonomous vehicles need to plan the trajectory on their own based on the obstacles and already occurred accidents in the front to avoid collision. Several existing studies based on the machine learning techniques above consider car crash prevention and identify near crash scenarios to alert the drivers. But, the car crash detection approach proposed in this paper, uses ensemble learning for feature extraction without prior training and concentrates on detection rather than prevention of accidents. Aiming at the requirement of autonomous vehicles, we have developed a car crash detection system for detecting diverse car crash including single-car crash scenarios.

3 Methodology

Our approach focuses on three different neural networks for casting car crash detection tasks: feature extraction network (VGG16), Region Proposal Network (RPN), and Region-Based detection network (CNN8L). The feature extractor part of the car crash detection system is reviewed with different transfer learning neural networks, thus allowing us to compare the performance of transfer learning networks in crash detection tasks. The RPN is another neural network that identifies the regions of interest (RoI) in an image by proposing bounding boxes, each of which has a score indicating the likelihood of the object present in the current sliding window. The Region-Based detection network is named CNN8L. It is a novel neural network architecture with a regressor and classifier for car crash classification and detection. Figure 1 illustrates our proposed car crash detection system. The flowchart of the proposed methodology is shown in Fig. 2.

3.1 Feature extractor

CNNs are built from multiple interconnected neural network layers, from which powerful low, middle, and high-level features are extracted hierarchically to create more complex networks [7, 8]. The InceptionResNetV2 network, created by GoogLeNet and available for download [12, 24], is one such example. The two most popular CNNs, ResNet [13, 19] and Inception [45] are integrated into InceptionResNetV2, where batch-normalization replaces the traditional summations process on the top layer of the network. The numerous residual block within the network increases the training complexity. To solve the training complexity problem efficiently [30] and minimize the residuals, Mahdianpari et al. applied more than 1,000 filters in the network layers.

According to its predecessor, AlexNet [28], the VGG network [43] took second place in the ILSFRC-2014 competition localization and classification track. The VGG

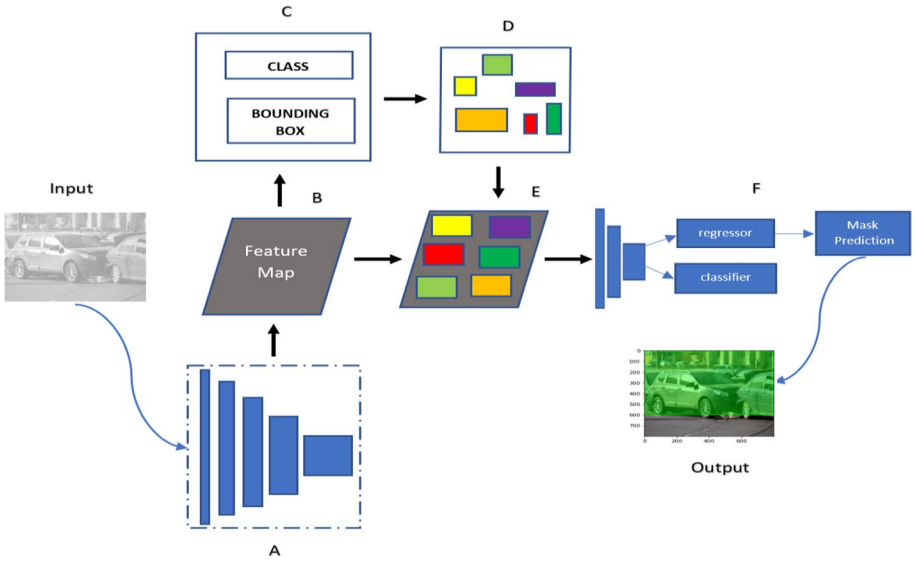


Fig. 1 Schematic representation of the Car Crash detection CNN model. A. Feature Extractor, B. Feature Map, C. Region of Interest (RoI), D. Region Proposal Network, E. RoI Pooling, and F. Region-Based Detection Network

network is distinguished by having a deep network structure and a small convolution filter of 3×3 compared to AlexNet. In the competition, the VGG-VD group introduced six deep CNNs, two of which, namely VGG16 and VGG19, were more successful than the others. The remaining four were less successful. Thirteen convolution layers and three fully-connected layers were used in VGG16 [13]. At the same time, VGG19 has sixteen convolution layers and three fully connected layers. Both networks use a 3×3 small convolution filter stack with step 1, followed by many non-linear layers in a stacked configuration. This contributes to learning more complex features by deepening the depth of the network and increasing its breadth. Due to the impressive VGG results, it has been found that tissue depth is an essential factor in achieving high classification accuracy [21].

The Mobile Net architecture can be summarized as follows; the structure is built on different abstraction layers, each of which is a component of various convolutions. These convolutions appear to be the quantized configuration to evaluate the in-depth complexity of a regular problem [18, 36]. The complexity of 1×1 can be described as a platform for the total production of abstraction layers with in-depth structures and to point through a standard, rectified linear unit (ReLU).

ResNet, the ILSVRC-2015 competition's classification task winner, has an intense network of 152 layers [47, 52]. In particular, the residual module in ResNet employs a direct path between the input and output. Each layer in the stack complies with a residual mapping instead of matching the desired underlying mapping directly [14].

The feature extractor module in the detection system extracts high-level features from the image and transfers them into feature maps. Many of the object detection models use transfer learning techniques to extract object. Transfer learning (TL) is a fundamental machine learning method that applies knowledge learned from a source domain to a target domain that is different but related. It can produce excellent results while

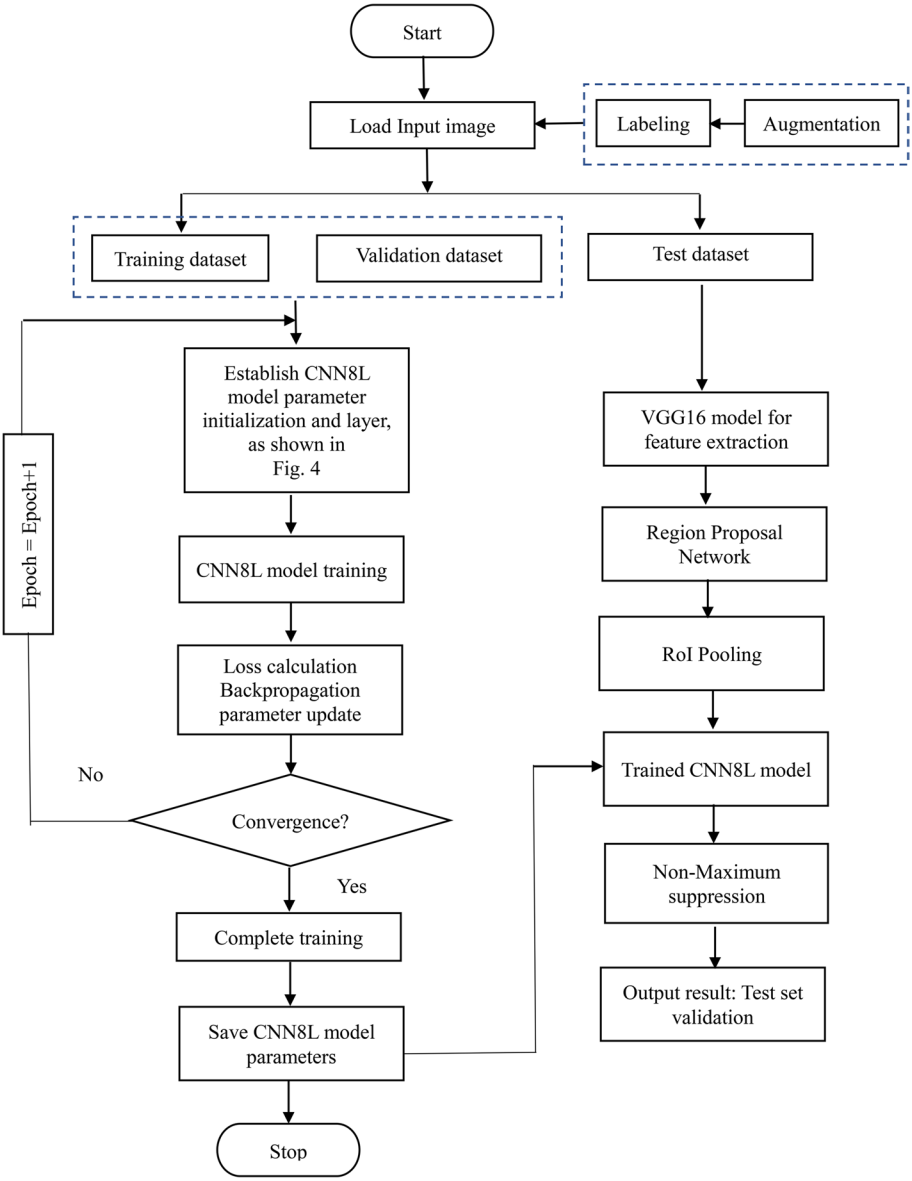


Fig. 2 Flowchart of the proposed methodology

also encouraging the development of areas that are challenging to progress due to a lack of training data.

To illustrate the advantages of using TL as the feature extractor, we compare the performance of more relevant models, including InceptionResNetV2 [15], VGG 16, Mobile Net V2 [9], and Resnet 50. In our proposed transfer learning model, all the convolutional layers, excluding the top fully connected layer of the pre-trained model, are used as a backbone in the feature extraction block.

3.2 Region proposal network (RPN)

The RPN takes in any input size from the feature extractor, the base layer of the transfer learning models, and proposes rectangular bounding boxes and the likelihood objective score for each bounding box. This model requires anchors, the boundary of proposals of different scales and ratios. Three scales and three ratios are used, yielding nine anchor boxes at each sliding position of the image. The number of anchors depends on the image's width(W) and height(h), given as 9xWxH. The scale factors 8,6 and 32 are used. The ratios used for calculations are 0.5, 1, and 2. anchor generation and image scaling are explained in Algorithm 1.

The RPN uses multiple convolutional blocks to extract a feature vector of size 50×50x512 in the lower dimension and receives the input feature vector from the feature extractor block. This output is passed to two fully connected layers for performing two tasks: an offset regression (a box regression layer) and an objective score calculation (a box classification layer). The offset regression layer outputs the bounding boxes, and the box

Algorithm 1: Anchor box generation and image scaling

```
Input:  w, h, w_fm, h_fm, a_r_ls = [0.5, 1, 2],   a_s_ls = [8,16,32]
        w ← width of the input image
        h ← height of the input image
        w_fm ← width of the feature map
        h_fm ← height of the feature map
        a_r_ls ← anchor ratio list
        a_s_ls ← anchor scale list
Begin
    x_stride = int(w / w_fm)   #stride for sliding window
    y_stride = int(h / h_fm)
    x_center = range(8, w, x_stride)   # center (xy coordinate) of anchor location on image
    y_center = range(8, h, y_stride)
    c_ls ← {(x_center,y_center)}   #center list
    al = []
      anchor_list ← { }
      count = 0
      for center in c_ls do
          center_x, center_y = center[0], center[1] #x, y coordinates of the anchor centers
          for ratio in anchor_ratio_list do
              for scale in anchor_scale_list do
                  a_h = pow(pow(scale, 2)/ ratio, 0.5) # compute height and width and
                  a_w = a_h * ratio                    #scale them by constant factor
                  a_xmin = center_x - 0.5 * a_w        #x, y coordinates of the anchor
                  a_ymin = center_y - 0.5 * a_h
                  a_xmax = center_x + 0.5 * a_w
                  a_ymax = center_y + 0.5 * a_h
                  al.append([center_x, center_y, w, h])
                  anchor_list[count] = [a_xmin, a_ymin, a_xmax, a_ymax]
                  count = count+1
              end
          end
      end
      return anchor_list,al
End
```

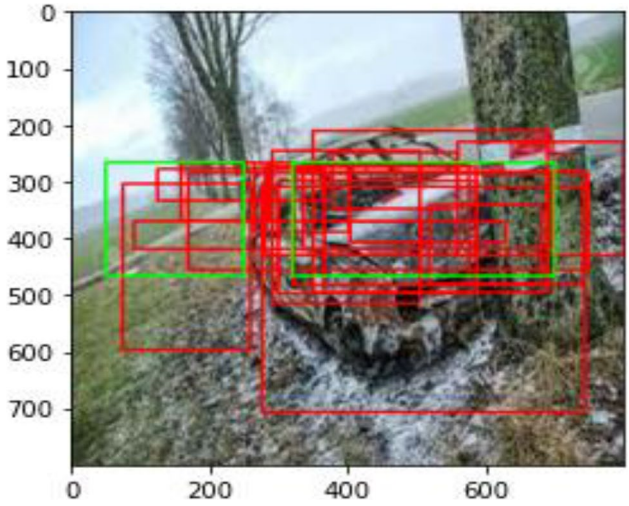


Fig. 3 Top 20 anchor boxes proposed by the RPN network

classification layer outputs objective scores to determine the probability of the object present within the bounded area. The top objective score anchors are selected as RoI.

The RPN model is trained to make predictions for each anchor box. It predicts the Intersection of Union (IoU) between the anchor box and ground truth and the class score for the object inside the box. The best anchor box is selected based on the IoU and assigned to ground reality with the class label = 1. If no such match is found, the anchor box is given class label = 0. This helps separate the image background from the foreground. The training loss function is a weighted sum of the regression and classification loss. The smooth L1 loss is used as regression loss, and binary cross-entropy is used as classification loss. Figure 3 illustrates the top 20 anchor boxes proposed by the RPN network. The red color locates the predicted ROIs, and the green color discovers the ground truths of the object.

3.3 Region-based detector

We propose a CNN8L neural network to classify the RoIs proposed by the RPN. CNN8L consists of 4 convolutional and 4 pooling layers summing up to 8 layers, so it is named CNN8L. We replaced the fully connected layer of RPN and FRPN with CNN8L since this is a detector with prior knowledge about car crash and trained independently without the VGG and RPN. This helps integrate multiple functionalities like the alarm, speed alert and brake control without the need for training the whole system again. A Region-Based Detector, CNN8L structure consists of a feature extractor block, image classifier block, and bounding box regressor block, as shown in Fig. 4. The feature extractor block integrates the low-level and high-level features and provides comprehensive feature information by combining four different convolution layers. The classifier block classifies the image, and the bounding box regressor block locates the image. The feature extractor block is the base layer for the proposed CNN8L model with four convolution layers and three max pool layers. The convolution layers bl_1, bl_3, and bl_5 are connected to the max pool layers bl_2 of size 3×3, bl_4 of size 3×3, and bl_6 of size 2×2, respectively. The last convolution layer bl_7 receives input from the max pool layer bl_6 and gives the output from bl_7

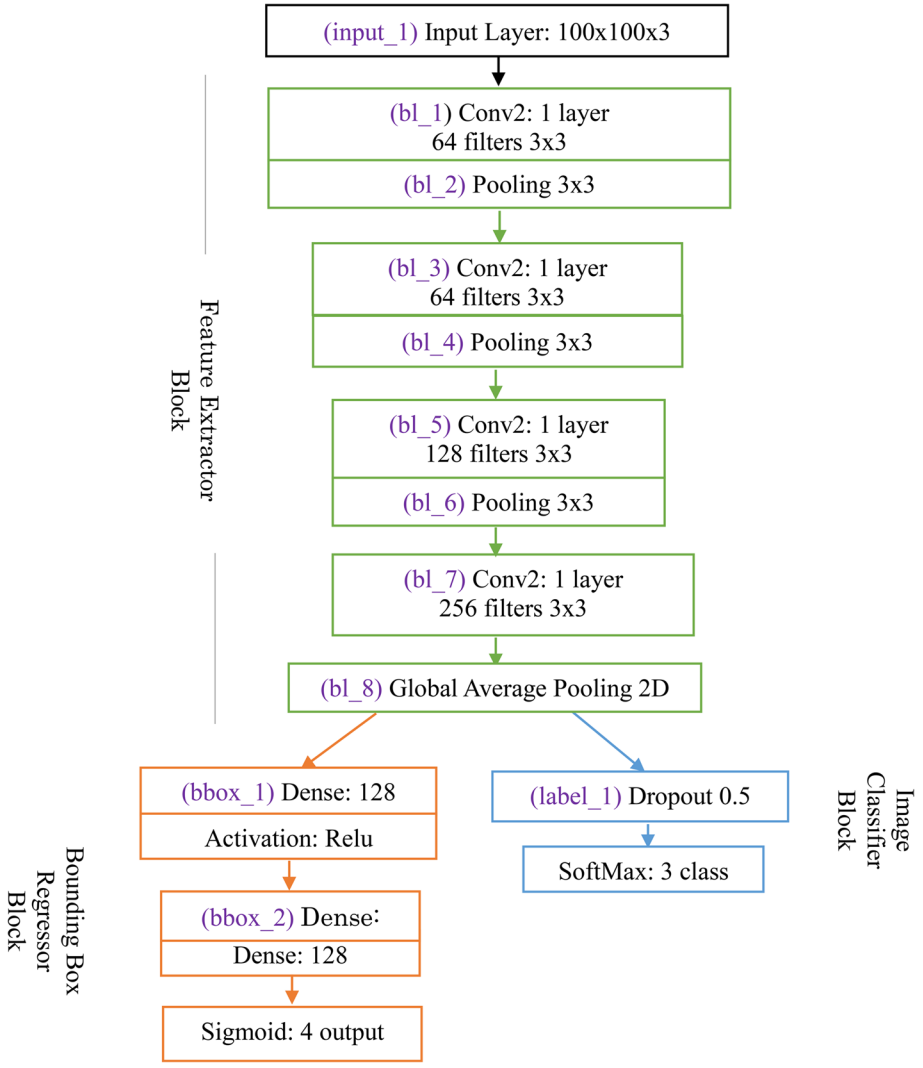


Fig. 4 CNN8L Network Architecture

to the global average pooling layer bl_8. The global average pool layer ends the feature extractor block and converts each feature map from bl_7 into one value. The extracted feature map size from the convolution layers (bl_1, bl_2,….bl_8) varies with the width and the height as shown in the formula given below—Eq. (1) and Eq. (2).

$$bl_w = (w - f_w + 2p)/(s_w) + 1 \tag{1}$$

$$bl_h = (h - f_h + 2p)/(s_h) + 1 \tag{2}$$

where pooling p defines the input sample's border, the input layer of this model is images with Width w, Height h, and three channels RGB. The feature map size and the required

parameters for the convolution layers bl with a filter size of N, kernel size of width f_w, and height f_h define the field of view of convolution. The stride size of width s_w and height s_h determines the step size of the kernel given in Eq. (3) and Eq. (4).

The final output size of

$$bl = (N, bl_w, bl_h) \tag{3}$$

Parameters needed for

$$bl = N \times bl_w \times bl_h \tag{4}$$

3.3.1 Image classifier block

The image classifier block is the classification layer for the proposed CNN8L model with a dropout layer and SoftMax activation function. The model reduces the features overfitted in the label_1 dropout layer by dropping 30% of the features. The dropout layer is connected to the three-output dense layer with the SoftMax activation function to obtain the probability distribution for the various classes [51]. The loss function of the Image Classifier Block is the label loss l_l as given in Eq. (5).

$$Label\ loss\ l_l = lw_l \times l_l \tag{5}$$

where lw_l is defined as the label weight loss.

3.3.2 Bounding box regressor block

The bounding box regression block is the object detection layer for the proposed CNN8L model with three dense layers. The first two dense layers, bbox_1 and bbox_2, activate using the ReLu activation function and give the output of feature sizes of 128 and 64, respectively. The final dense layer gives four outputs for locating the object in the image. The regression block's final layer is activated using the sigmoid activation function. The loss function of the bounding box regressor block is bb_l as given in Eq. (6), and the overall model loss is shown in Eq. (7).

The bounding box loss

$$bb_l = bbw_l \times bb_l \tag{6}$$

where bbw_l is the bounding box weight loss.

The overall model loss

$$L = l_l + bb_l \tag{7}$$

CNN8L detected the same classes with overlapping areas often. So, a computer vision technique called non-maximum suppression is applied to the detected bounding boxes to repeatedly choose the entity with the highest probability, output that as the prediction, and then remove the redundant bounding boxes having an IoU greater than (Threshold)Th = 0.4 compared to the other bounding boxes of the same class. Algorithm 2 explains the Non-Maximum Supression(NMS) technique.

Algorithm 2: Non-Maximum Suppression

```
Input: roi = {roi1, roi2,….roiN}, cls = {cls1, cls2,….clsN}, scr={scr1,scr2,…..scrN}, Th = th
        roi ← List of bounding boxes
        cls ← Corresponding classes
        scr  ← Corresponding detection confidence scores
        Th ←Threshold to remove the overlapping rois
Begin
    B  ← 0
    while roi ≠ 0 do
        m ← argmax scr
        M ← roim
        B ← B ∪ M, roi← roi−M
        for roii in roi do
            if IoU(M,roii) ≥ Th then      #IoU defined in Equation (9) where bbg = M, bbp= roii
                roi← roi − roii;   scr←scr – scri;   cls←cls – clsi;    # delete the overlapping rois
            end
        end
    end
    return B, scr, cls
end
```

3.4 Model validation

Our experiment utilized different parameters [25, 42] to measure the efficiency and performance of the models. In computer vision, the research community has converged on using the accuracy metrics, average Precision, and mean average Precision to measure the objection recognition and detection model. The label is classified based on the probability scores. If the score of the predicted class is greater than or equal to 0.5 and equal to annotated class, it is considered True Positive (T_P). If the score of the predicted class is greater than or equal to 0.5 and not equal to the annotated class, it is considered False Positive (F_P). If the score of the predicted class is lesser than 0.5 and similar to the annotated class, it is considered False Negative (F_N). If the score of the predicted class is less than 0.5 and not equal to the annotated class, it is considered True Negative (T_N). The accuracy of the classification model is defined in Eq. (8).

$$Accuracy = \frac{T_p + T_N}{T_P + F_P + T_N + F_N} \tag{8}$$

The object detection measurement is based on two important metrics, Intersection Over Union (IoU) and class label. The object in the image is located using the rectangle bounding box(bb). Using this representation, the IoU is defined as given in Eq. (9) [24].

$$\text{IoU} = \frac{Area\ of\ Intersection\ of\ bb_g Over\ bb_p}{Area\ of\ Union\ of\ bb_g\ and\ bb_p} \tag{9}$$

where bb_g is the ground truth bounding box, and bb_p is the Predicted bounding box.

The Precision (p_{class}) and recall (r_{class}) is calculated for the IoU threshold greater than or equal to 0.5 for a class is shown in Eq. (10) and Eq. (11) [41], T_P lesser than the 0.5 IoU threshold is considered F_P.

$$p_{class} = \frac{T_P}{Total\ Number\ of\ Annotation} \tag{10}$$

$$r_{class} = \frac{T_P}{T_P + F_P} \tag{11}$$

The Average Precision of the class (AP_{class}) is the weighted sum of Precision for the class and each IoU threshold, where the weight is the difference between the current and next recall. AP_{class} is defined in Eq. (12).

$$AP_{class} = \sum_{t=0}^{t=th-1} [r_{class}(t) - r_{class}(t+1)] p_{class}(t) \tag{12}$$

Further, th is the total number of IoU thresholds. $r_{class}(th) = 0$ and $p_{class}(th) = 1$. The Mean Average Precision(mAP) is given in Eq. (13).

$$mAP = \frac{1}{n} \sum_{class=1}^{n} \mathrm{AP}_{class} \tag{13}$$

where n is the number of classes.

The car crash accident detection framework performance is based on two major parameters that have been used in most of the comparative studies for accident detection models [17]. The two parameters are Accident Detection Rate (ADR) and False Alarm Rate (FAR), as given in Eqs. (14) and (15) [6, 16, 46].

$$ADR = \frac{Correctly\ detected\ accidents}{Total\ accidents} \times 100 \tag{14}$$

$$FAR = \frac{Falsely\ identified\ classes}{a\ total\ number\ of\ classes} \times 100 \tag{15}$$

4 Experiment results and discussion

4.1 Dataset

The model used in this paper classifies three different categories background, car, and car crash, as shown in Fig. 4, where the three categories are represented as 0 for background, 1 for car, and 2 for a car crash. In the case of Autonomous vehicles, locating the boundary of each object is critical for path navigation. The car and car crash classes easily overlap without having high confidence in the other class. Thus, adding a background class helps locate the objects clearly without embedding the image background in the detection process. We used different datasets for each category. Our research work collected background images from the Pothole dataset compiled by the Electrical and Electronic Department, Stellenbosch University, 2015 [31, 32], car images from Stanford Cars Dataset [27], and car crash images from Unsplash [50], and iStock [23] photo. Along with this, we used the car crash and car images from [33]. The car crash dataset is released at: https://github.com/vanisaravanarajan/Car-Crash-Dataset.git.

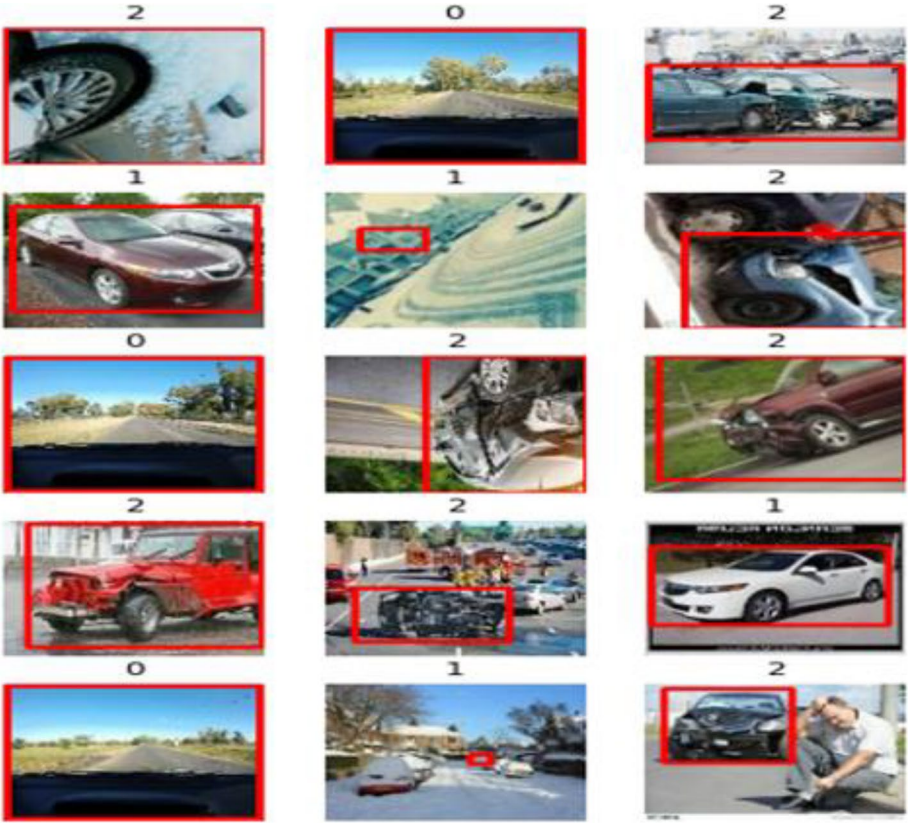


Fig. 5 The labeled images

Our experiment applied four types of augmentation to the collected images RandomCrop, Rotate, HorizontalFlip, and VerticalFlip.

Moreover, our work used the LabelImg as a tool [49] to create an axis-aligned rectangular at the end of the day and a bounding box around the object for labeling the newly generated Accident detection dataset. We constructed a car accident detection dataset, and the sample images are shown in Fig. 5.

4.2 Data preprocessing

We applied five preprocessing strategies to the dataset. Firstly, all the images are resized to 100×100 pixels. Secondly, we checked the image type and file type-image format. Thirdly,

Table 1 Dataset Information

Class	Background (Class 0)	Car (Class 1)	Car crash (Class 2)
Training	162	308	1230
Validation	21	86	160
Testing	14	150	438

Table 2 Hyper-Parameter Settings

Parameters	Values
epoch	200
Batch size	32
learning rate	0.001
momentum	0.9
lw_l	1.0
bbw_l	1.0

check for missing annotations. Fourthly, labeled the background category as '0', the car category as '1', and the car crash category as '2'. Lastly, divide the dataset into three sets such that 1700, 267, and 607 for training, validation, and testing, respectively. The categories of car accident detection datasets are shown in Table 1.

4.3 Experiment

CNN 8L and Transfer Learning models are evaluated on the car accident detection dataset. The processor used to conduct these experiments is 11th Gen Intel(R) Core (TM) i7-11800H @ 2.30 GHz, GPU of NVIDIA GeForce RTX 3050 Ti, Python 3.8, TensorFlow 2.6.2, CUDA 11.2, and cuDNN 8.1. The models are compiled using an Adam optimizer with two losses Sparse Categorical Cross Entropy and Mean Squared Error (MSE). We tested the model with various epochs (50,100, 150 and 200), and batch sizes (8, 16 and 32). The model validation stabilized at 200 epochs and the batch size = 32. The hyper-parameters are listed in Table 2. In the proposed methodology, we applied a sparse categorical loss function for the image classification and MSE(Mean Squared Error) for bounding box regressor with Adam optimization. The model learns from the loss function and optimizes the parameters.

4.4 Results and discussion

We performed an ablation study to analyze the system mAP as per Eq. 13 by combining the state-of-the-art models as feature extractors with the CNN8L model. The VGG16 and CNN8L combination performed the best, as shown in Table 3.

Table 4 lists the ablation study performed to identify the number of layers for a CNN model as a Region-Based detector. We tested the CNN model accuracy by varying the

Table 3 Ablation Study

Model	AP Class 0 (%)	AP Class 1 (%)	AP Class 2 (%)	mAP (%)
InceptionResNetV2 + CNN8L	9.5	25.6	28.0	21.0
MobileNetV2 + CNN8L	7.1	11.9	41.6	20.2
Resnet50 + CNN8L	64.0	36.0	20.0	40.1
VGG16 + CNN 8L	86.0	85.0	89.0	86.6

Table 4 Ablation study by varying the layers

Layers	Accuracy (%)
5(3conv + 2pool)	59.0
7(4conv + 3pool)	62.0
8(4conv + 4pool)	89.5
16(8conv + 8pool)	90

number of layers. Considering the model with optimum layers in terms of accuracy and the number of parameters, we selected CNN with 4 conv(convolutions) and 4 pool(pooling) summing to 8 layers as the Region-Based detector in the name of CNN8L, even though CNN with 8 conv and 8 pool showed 0.5% slightly higher accuracy than the CNN8L.

We compared the processing time and the number of parameters of CNN8L with the other state-of-the-art models, as shown in Table 5. CNN8L is the best with the lowest parameters and processing time. The CNN 8L model has 36 times fewer parameters and is 6 times faster than the second best VGG16 model. In autonomous vehicles, both model speed and accuracy are equally important.

A real-time accident detection system should identify the accidents from a wide variety of accident data though those patterns did not appear during the model training process. Such models are preferred for practical applications, as they can generalize well and detect patterns in any condition. Hence, to study the performance of the detection system, we tested a wide range of crash scenarios. The system showcased excellent generalization properties by identifying many previously unseen car crashes, as shown in Fig. 6. The determined car crash and car classes are masked with green and blue colors. The detection confidence score threshold is more significant than 0.7 for identifying and locating classes in the tested images.

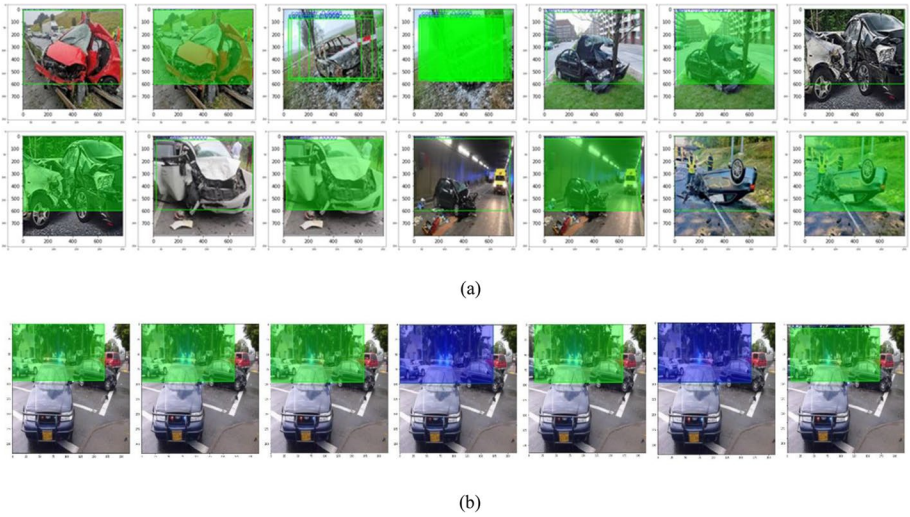
In some cases, both the car and car crash had a good confidence score and overlapping, as shown in Fig. 6(b). We selected 125 car crash frames and 75 normal frames without car crash for testing. Out of 125 frames, 105 frames are correctly detected for car crash, and the other 15 frames are detected as cars. A total of 225 classes were identified in the 200 frames selected for testing, as some of the frames had multiclass. The final detection output showed 75 false alarms out of 225 classes. The ADR and FAR of our framework are listed in the different framework comparisons in Table 6.

5 Conclusion

A novel system for detecting car crash with a new deep-learning convolutional neural network model named CNN8L is proposed. Due to the higher number of required parameters, the training is computationally expensive for the widely accepted popular

Table 5 List of Model Parameters and Processing Time

Model	Parameters	Processing Time (seconds)
InceptionResNetV2	54,546,599	1205.035
MobileNetV2	2,434,311	335.4925
Resnet50	23,864,647	825.99
VGG16	14,790,407	686.73
CNN8L	449,927	101.67



(a)

(b)

Fig. 6 Masked Output of Car Crash Detection System from the (a) images and (b) video

models like RCNN, Faster CNN and Mask RCNN. The improvement of our proposed car crash detection system is due to faster training, easier to converge and reduced overfitting problems. The CNN8L is trained with smaller filter sizes and fewer filter numbers, with 449,927 parameters, and 101.67 s of processing time. The car crash detection system takes advantage of ensemble learning techniques of neural networks. We also proposed transfer learning techniques with the state of art models VGG16, InceptionResNetV2, MobileNetV2, and Resnet50 for object recognition and detection on the customized car accident dataset. The sequential convolutional neural networks CNN8L and VGG16 performed well. Overall system integrated with VGG16 for feature extraction, RPN for region proposal, and CNN8L for region-based detection recognized the accidents accurately with an 86.25% Accident Detection Rate and 33.00% False Alarm Rate.

Due to the extreme lack of data for AVs, our study's first limitation is the use of a single data set, which restricts the generalizability of our findings. The second limitation is the challenge to compare our results to those of the existing works. This is due to different accident definitions and input data types. Future work in our domain will extend the CNN8L model to research areas, such as road scene segmentation, navigational planning, audiovisual alarm generation, the suggestion of speed limit, and automatic brake control to lower the speed and also to reduce the possibility of a collision for AVs.

Table 6 Accident Detection framework comparison

Approach	ADR (%)	FAR (%)
Sahrawat et al. [38]	71.33	59.23
Chan et al. [5]	80.00	38.45
Chand et al. [6]	79.05	34.44
Our Framework (Section 2)	86.25	33.00

Acknowledgements This paper is supported by the Ministry of Science and Technology, Taiwan. The Nos is MOST 111-2221-E-324 -020, Taiwan.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflicts of interest The authors declared that they have no conflicts of interest in this work.

References

1. Abdulhafedh A (2017) Road Crash Prediction Models: Different Statistical Modeling Approaches. J Transp Technol 07(02):190–205. https://doi.org/10.4236/jtts.2017.72014
2. Adewopo V, Elsayed N, ElSayed Z, Ozer M, Abdelgawad A, Bayoumi M (2022) Review on action recognition for accident detection in smart city transportation systems, arXiv preprint arXiv:2208.09588
3. Bakheet S, Al-Hamadi A (2022) A deep neural framework for real-time vehicular accident detection based on motion temporal templates. Heliyon 8(11):e11397. https://doi.org/10.1016/j.heliyon.2022.e11397
4. Bera A, Kim S, Randhavane T, Pratapa S, Manocha D (2016) GLMP- real-time pedestrian path prediction using global and local movement patterns. Proc. - IEEE Int. Conf. Robot. Autom., pp 5528–5535. https://doi.org/10.1109/ICRA.2016.7487768
5. Chan FH, Chen YT, Xiang Y, Sun M (2017) Anticipating accidents in dashcam videos, Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 10114 LNCS. https://doi.org/10.1007/978-3-319-54190-7_9
6. Chand D, Gupta S, Kavati I (2020) Computer vision based accident detection for autonomous vehicles. In: 2020 IEEE 17th India Council International Conference, INDICON 2020. https://doi.org/10.1109/INDICON49873.2020.9342226
7. Chang WJ, Chen LB, Su KY (2019) DeepCrash: a deep learning-based internet of vehicles system for head-on and single-vehicle accident detection with emergency notification. IEEE Access 7. https://doi.org/10.1109/ACCESS.2019.2946468.
8. Chen RC, Dewi C, Huang SW, Caraka RE (2020) Selecting critical features for data classification based on machine learning methods. J Big Data 7(52):1–26. https://doi.org/10.1186/s40537-020-00327-4
9. Chen W, Gao L, Li X, Shen W (2022) Lightweight convolutional neural network with knowledge distillation for cervical cells classification. Biomed Signal Process Control 71. https://doi.org/10.1016/j.bspc.2021.103177
10. Chen RC, Saravanarajan VS, Chen LS, Yu H (2022) Road segmentation and environment labeling for autonomous vehicles. Appl Sci 12(14):7191. https://doi.org/10.3390/app12147191
11. Choi JG, Kong CW, Kim G, Lim S (2021) Car crash detection using ensemble deep learning and multimodal data from dashboard cameras. Exp Syst Appl 183:115400. https://doi.org/10.1016/j.eswa.2021.115400
12. Dewi C, Chen RC (2022) Combination of resnet and spatial pyramid pooling for musical instrument identification. Cybern Inf Technol 22(1):104
13. Dewi C, Chen RC, Jiang X, Yu H (2022) Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks. PeerJ Comput Sci 8:943. https://doi.org/10.7717/peerj-cs.943
14. Dewi C, Chen RC, Jiang X, Yu H (2022) Deep convolutional neural network for enhancing traffic sign recognition developed on Yolo V4. Multimed Tools Appl. https://doi.org/10.1007/s11042-022-12962-5
15. Ekundayo O, Murphy L, Pathak P, Stynes P (2022) An on-device deep learning framework to encourage the recycling of waste. Lecture Notes in Networks and Systems 296:405–417. https://doi.org/10.1007/978-3-030-82199-9_26
16. M Ferguson R Ak YTT Lee KH Law (2018) Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning Smart Sustain. Manuf. Syst. 2 1 https://doi.org/10.1520/SSMS20180033
17. Ferguson M, Ak R, Lee YTT, Law KH (2017) Automatic localization of casting defects with convolutional neural networks. In: Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017. https://doi.org/10.1109/BigData.2017.8258115
18. Gavai NR, Jakhade YA, Tribhuvan SA, Bhattad R (2018) MobileNets for flower classification using TensorFlow. In: 2017 International Conference on Big Data, IoT and Data Science, BID 2017. https://doi.org/10.1109/BID.2017.8336590

19. K He X Zhang S Ren J Sun (2016) Deep residual learning for image recognition Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 770 778 https://doi.org/10.1109/CVPR.2016.90
20. Hozhabr Pour H, Li F, Wegmeth L, Trense C, Doniec R, Grzegorzek M, Wismüller R (2022) A machine learning framework for automated accident detection based on multimodal sensors in cars. Sensors 22(10):3634. https://doi.org/10.3390/s22103634
21. Hu F, Xia GS, Hu J, Zhang L (2015) Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sens 7(11):14680–14707. https://doi.org/10.3390/rs71114680
22. Ijjina EP, Chand D, Gupta S, Goutham K (2019) Computer vision-based accident detection in traffic surveillance. In: 2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019. https://doi.org/10.1109/ICCCNT45670.2019.8944469
23. iStock. (n.d.). iStock Photo. Retrieved February 21, 2022, from https://www.istockphoto.com/
24. K Jiang J Zhang H Wu A Wang Y Iwahori (2020) A novel digital modulation recognition algorithm based on deep convolutional neural network. Appl Sci 10(3). https://doi.org/10.3390/app10031166
25. Khairi MHH et al (2021) Detection and classification of conflict flows in SDN using machine learning algorithms. IEEE Access 9:76024–76037. https://doi.org/10.1109/ACCESS.2021.3081629
26. MMR Komol MM Hasan M Elhenawy S Yasmin M, Masoud A, Rakotonirainy (2021) Crash severity analysis of vulnerable road users using machine learning. PLoS One 16(8). https://doi.org/10.1371/journal.pone.0255828
27. Krause J, Stark M, Deng J, Fei-Fei L (2013) 4th IEEE Workshop on 3D Representation and Recognition, at ICCV 2013 (3dRR-13). Sydney, Australia. Dec. 8, 2013
28. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60(6):84–90. https://doi.org/10.1145/3065386
29. Lu Z, Zhou W, Zhang S, Wang C (2020) A new video-based crash detection method: balancing speed and accuracy using a feature fusion deep learning framework. Journal of Advanced Transportation 2020. https://doi.org/10.1155/2020/8848874
30. Mahdianpari M, Salehi B, Rezaee M, et al. (2018) Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. Remote Sens 10(7). https://doi.org/10.3390/rs10071119
31. Nienaber S, Booysen MJ, Kroon RS (2015) Detecting potholes using simple image processing techniques and real-world footage, SATC, July 2015, Pretoria, South Africa
32. Nienaber S, Kroon RS, Booysen MJ (2015) A comparison of low-cost monocular vision techniques for pothole distance estimation, IEEE CIVTS, December 2015, Cape Town, South Africa
33. Pashaei A, Ghatee M, Sajedi H (2020) Convolution neural network joint with mixture of extreme learning machines for feature extraction and classification of accident images. J Real-Time Image Proc 17:1051–1066. https://doi.org/10.1007/s11554-019-00852-3
34. Pathik N, Gupta RK, Sahu Y, Sharma A, Masud M, Baz M (2022) AI enabled accident detection and alert system using iot and deep learning for smart cities. Sustainability 24(14):7701. https://doi.org/10.3390/su14137701
35. MS Pillai M Chaudhary Khari RG Crespo (2021) Realtime image enhancement for an automatic automobile accident detection through CCTV using deep learning. Soft Comput 25(18). https://doi.org/10.1007/s00500-021-05576-w
36. Rabano SL, Cabatuan MK, Sybingco E, et al. (2018) Common garbage classification using mobilenet. In: 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication, and Control, Environment and Management, HNICEM 2018. https://doi.org/10.1109/HNICEM.2018.8666300
37. Rahim MA, Hassan HM (2021) A deep learning based traffic crash severity prediction framework. Accident Analysis and Prevention 154 https://doi.org/10.1016/j.aap.2021.106090
38. Sahrawat D, Anand S, and Kaul SK (2019) Improving road safety through accident detection and prediction in dashcam videos
39. Saravanarajan VS, Chen RC, Chen LS (2021) LiDAR Point Cloud Data Processing in Autonomous Vehicles, In2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT) 2021 1–5 IEEE
40. Saravanarajan VS, Chen RC, Chen LS (2022), Geometric feature learning network for detecting the objects in urban streets, Available at SSRN 4218483
41. Saravanarajan VS, Chen RC, Dewi C (2020) Chen LS (2020) Solving Unbounded Knapsack Problem Using Evolutionary Algorithms with Bound Constrained Strategy. Int J Appl Sci Eng 18(1):1–12
42. Saravanarajan VS, Chen RC, Dewi C, and Chen LS (2020) Montecarlo Approach for Solving Unbound Knapsack Problem. In: The 7th Multidisciplinary International Social Networks Conference (MISNC 2020), Taiwan, pp 1–10

43. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations (ICLR2015)
44. Singh S (2015) Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey, Natl. Highw. Traffic Saf. Adm., no. February 1–2
45. Szegedy C, Liu W, Jia Y et al (2014) GoogLeNet Going Deeper with Convolutions. ArXiv Preprint ArXiv:1409.4842 53 1–9 https://doi.org/10.1109/ICCV.2011.6126456
46. Tai S, Dewi C, Chen R, Liu Y, Jiang X, Yu H (2020) Deep Learning for Traffic Sign Recognition Based on Spatial Pyramid Pooling with Scale Analysis. Appl Sci 10(19):6997. https://doi.org/10.3390/app10196997
47. Thinh NH, Hoang Tung T, Ha LV (2020) Depth-aware salient object segmentation. VNU J Sci Comput Sci Commun Eng 36(2). https://doi.org/10.25073/2588-1086/vnucsce.217
48. Tian D, Zhang C, Duan X, Wang X (2019) An automatic car accident detection method based on cooperative vehicle infrastructure systems. IEEE Access 7:127453–127463. https://doi.org/10.1109/ACCESS.2019.2939532
49. Tzutalin, Labellmg, 2015
50. Unsplash. (n.d.). Car crash. Retrieved February 20, 2022, from https://unsplash.com/s/photos/car-crash
51. Wu J, Du J, Wang F, Yang C, Jiang X et al (2022) A multimodal attention fusion network with a dynamic vocabulary for TextVQA. Pattern Recognit 122. https://doi.org/10.1016/j.patcog.2021.108214
52. Yang EH, Amer H, Jiang Y (2021) Compression helps deep learning in image classification. Entropy 23(7). https://doi.org/10.3390/e23070881
53. Zhou Z, Dong X, Li Z, Yu K, Ding C, Yang Y (2022) Spatio-Temporal Feature Encoding for Traffic Accident Detection in VANET Environment. IEEE Trans Intell Transp Syst 23(10):19772–19781. https://doi.org/10.1109/TITS.2022.3147826

Authors and Affiliations

Vani Suthamathi Saravanarajan[1] · Rung-Ching Chen[1] · Christine Dewi[1,2] · Long-Sheng Chen[1] · Lata Ganesan[3]

Vani Suthamathi Saravanarajan
s10814909@gm.cyut.edu.tw

Christine Dewi
christine.dewi13@gmail.com

Long-Sheng Chen
lschen@cyut.edu.tw

Lata Ganesan
lataganesan@gmail.com

[1] Department of Information Management, Chaoyang University of Technology, Taichung, Taiwan

[2] Faculty of Information Technology, Satya Wacana Christian University, Central Java, Indonesia

[3] Senior Data Scientist and Software Consultant, 441 Vanderveer Road, Raritan, NJ 08869, USA