ISSN: 0067-2904

Vehicle Accident Detection and Notification System

Duaa Hadi Nassar*, Jamal Mustafa Al-Tuwaijari
Department of Computer Sciences, College of Science, University of Diyala, Diyala, Iraq

Received: 26/2/2023 Accepted: 5/7/2023 Published: 30/7/2024

Abstract

As it is observed these days, the roads are crowded with different kinds of vehicles, and as a result, accidents have become more dangerous every day. In this paper, a vehicle accident detection and notification system is designed by using road surveillance camera data and machine learning techniques to use the recorded footage from the surveillance cameras to detect the accident status, activate an alerting sound, and send the notification message as a Google email (Gmail) to the specialist. The system model was made by combining two algorithms, convolutional neural networks (CNN) and support vector machines (SVM), to make a CNN-SVM hybrid model, which was then trained using two datasets. The first dataset contains 4814 images with sizes (28, 28) and extensions (jpg), and the second dataset contains 990 images with sizes (32, 32) and extensions (jpg). The outcomes scores of the evaluation matrices are: accuracy for the first dataset is 99.74% and loss is 1.14%, and for the second dataset, accuracy is 98.88% and loss is 3.39%. When testing it with real-world data, it achieved its objectives in 30 seconds, with accuracy reaching 100%.

Keywords: Accidents, Detection, Machine learning, hybrid model, CNN-SVM model, Notification, Vehicles

نظام كشف حوادث المركبات والأخبار عنها

دعاء هادي نصار * , جمال مصطفى التويجري
قسم علوم الحاسوب ,كلية العلوم, جامعة ديالى, ديالى, العراق

الخلاصة

كما نشهد هذه الأيام طرقنا اصبحت تزدحم بأنواع مختلفة من المركبات وكنتيجة لذلك فان الحوادث اصبحت اكثر خطورة. في هذه الورقة البحثية تم تصميم وبناء نظام لكشف حوادث المركبات و الأخبار عنها باستعمال بيانات كاميرات مراقبة الطرق وبالاعتماد على تقنيات التعلم الألي. يهدف هذا النظام الى استعمال اللقطات المسجلة من قبل كاميرات المراقبة واكتشاف الحوادث ثم يقوم بإصدار اصوات تحذيرية و إرسال رسالة يخبر بها عن الحادث، تكون على شكل بريد غوغل الالكتروني (Gmail) الى المختصين. نموذج النظام تم بناءه من خلال دمج خوارزميتين هما خوارزمية الشبكة العصبية الالتفافية (CNN) وخوارزمية دعم المتجهات (SVM) لتشكيل النموذج الهجين CNN–SVM نموذج النظام تم تدريبه باستعمال مجموعتي بيانات المجموعة الأولى تحتوي على 4814 صورة بأبعاد (28,28) وامتداد (jpg) أما الثانية تحتوي على 990 صورة بأبعاد (32,32) وامتداد (jpg). بعد مرحلة التدريب فان قيم مقاييس التقييم (Evaluations measures) مثل الدقة لمجموعة البيانات الاولى % 99.74 اما نسبة الخطأ % 1.14 بالنسبة للمجموعة الثانية الدقة % 98.88 والخطأ %

*Email: scicompms2209@uodiyala.edu.iq

3.39 عند اختبار النظام باستعمال البيانات الواقعية فانه حقق اهدافه خلال فترة زمنية قدرها 30 ثانية وبدقة اكتشاف تصل الى 100%.

1. Introduction

The population is growing gradually. Along with the growth in population, the number of vehicles is also increasing [1]. These days, people are increasingly searching for comfortable ways to fulfill their daily duties, even while doing their simple daily tasks. The trend toward owning a private vehicle for daily procedures is growing rapidly.

As additional people obtain vehicles, the traffic load is growing on the roads, initiating accidents [2]. According to the World Health Organization (WHO), each year, 1.25 million people lose their lives as a result of traffic accidents, and similarly, 20 to 50 million extra people get wounded [3]. In addition to the risks to people's lives, there are other risks that could be involved in these accidents, like the associated medical costs and the overall repairs and maintenance costs of the road infrastructure. Also as a result of accidents is the traffic gathering; a lot of people lose their time on the roads as a result of car accidents, and this gathering makes the medical and other help providers work very hard and late [4]. Manual control of traffic by traffic officers or using predefined timers has been confirmed as not being an actual solution to all the problems mentioned above caused by traffic accidents [5]. So we need to design better traffic management systems that can detect accidents and increase survival rates after an accident. By reducing the time gap between an accident happening and the first responders being transmitted, we reduce death rates so that we can save lives by giving an urgent notification to the nearest medical station [6].

With the thousands of surveillance cameras that cover our roads and footage of how vehicles behave on the roads, this provides a huge source of data that can be analyzed, examined, and used to detect accidents before they happen or at the moment they happen. With the existing internet networks that can be used as carriers for the notification signals generated by the system, all these factors act as the foundation of this system. In this thesis, the system will be accomplished by passing on the road surveillance camera images as input, and in order to efficiently manage the data generated every day and use better system engineering and artificial intelligence (AI), artificial intelligence and machine learning (ML) play an important role in predicting road system occurrences [7].

In this paper, an accident and notification system is proposed by employing advanced machine learning (ML) methods, particularly deep learning (DL), that are used for decision-making at different levels. Based on the road footage that has been recorded by the surveillance cameras, a decision must be taken by the module if there is an actual accident happening, and based on that decision, the notification message will inform the nearest medical station and other help provider stations about the accidents, as well as the sound alerting to inform other road users about the accident, as shown in Figure 1 below.

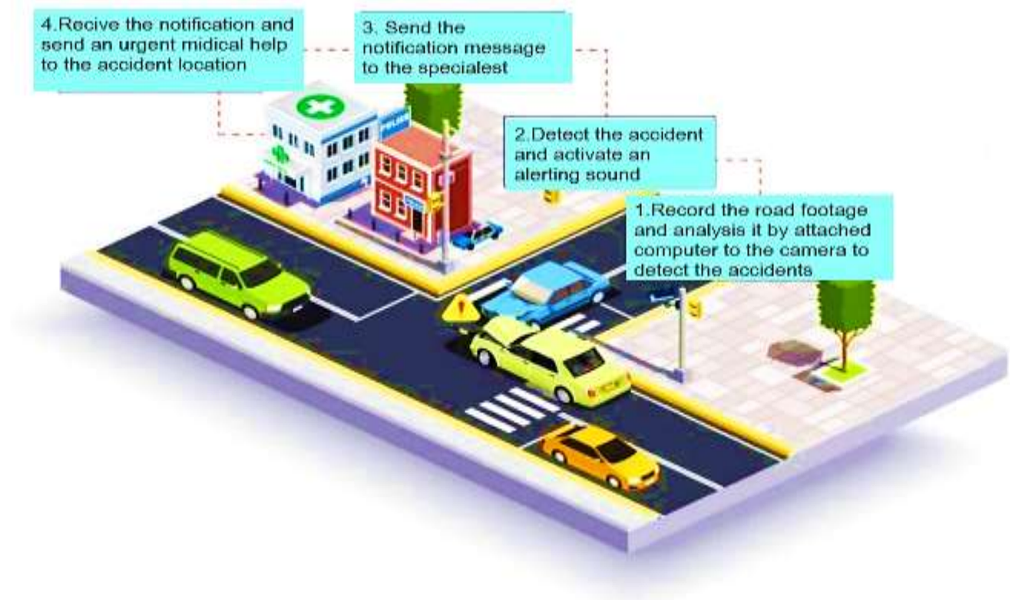

Figure 1: Accident and notification system mechanism

The rest of this paper is organized as follows: Section 2 presents an overview of related work. Section 3 describes briefly the following methodology: Section 4 shows the proposed system architecture. Section 5 presents the experimental results of the proposed system. Section 6 reports the final conclusions. Finally, Section 7 presents suggestions for future work.

2. Related works

Vehicle accident detection and notification: many systems have been proposed whose main purpose is to accomplish this goal. Lots of methods, techniques, and algorithms are used to apply these systems, and the below articles describe some of them.

In 2018, Zhenhua Zhang et al. [8] employed DL to detect accidents from social media data. First, they systematically examined the 3 million tweets for accidents. Their results showed that matching marks can retain the connotation rules essential to accident-related tweets and increase the accuracy of accident detection. Second, two DL methods, Deep Belief Network (DBN) and Long Short-Term Memory (LSTM), are studied and applied to extracted marks. Results showed that DBN can achieve an overall accuracy of 85% with about 44 individual mark features and 17 paired token features. The classification results from DBN outperform those of support vector machines (SVMs) and supervised latent Dirichlet allocation (sLDA).

In 2019, Daxin Tian et al. [9], based on cooperative vehicle infrastructure systems (CVIS) and machine vision, suggested an automated system for detecting automobile accidents. The accuracy of accident detection based on intelligent roadside equipment in CVIS is first improved by a recognized original picture dataset called CAD-CVIS. Second, using deep learning methods and the CAD-CVIS accident detection framework, they created the YOLO-CA deep neural network model. The results demonstrate that their suggested approach can do so in 0.0461 seconds (21.6 FPS) with an average precision (AP) of 90.02%.

In 2020, Zhenbo Lu et al. [10] created a feature fusion-based deep learning framework for the video-based identification of urban traffic crashes with the goal of balancing detection speed and accuracy while using minimal computer resources. In this framework, it was suggested to use a residual neural network (ResNet) along with attention modules to extract crash-related appearance features from urban traffic videos. These features were then fed to the spatiotemporal feature fusion model Conv-LSTM (Convolutional Long Short-Term Memory). A collection of video clips containing 342 non-crash occurrences and 330 crash events were used to train the proposed model. The suggested model obtained a detection speed of FPS > 30 with an accuracy of 87.78%.

In 2021, Jae Gyeong Choi et al. [11] used DL methods, gated recurrent units (GRU), and convolutional neural networks (CNN) to progress a car accident detection system. A biased average ensemble is used as an ensemble technique. The projected accident detection system, which is built on several classifiers that use both video and audio data from instrument panel cameras, is genuine using an evaluation with single classifiers that use video data or audio data only. Accidental YouTube fasteners are used to validate this research. The results show that the projected accident detection system achieves considerably better results than single classifiers, and it fulfills ROCAUC by 98.60%.

In 2022, Vedika Sadavarte et al. [12] aim to provide an automatic accident detection system with alert generation to offer appropriate assistance in critical ways by using the CNN TensorFlow object detection API. To train the system, 120 video frames have been collected in various situations. The results show that the proposed system can detect an accident and send an alert with a mean absolute percentage error of less than 20%.

In 2022, Hawzhin Hozhabr Pour et al. [13] proposed an ML structure for automatic accident detection built on multimodal sensors. There are five different ways to extract features. Counting techniques based on feature production and feature learning with DL are used to figure out how many crashes were in the SHRP2 NDS crash data set. The main explanations are as follows: (1) CNN features with a SVM classifier gain very hopeful results. (2) Feature engineering and feature learning approaches were finding dissimilar best-performing features. Therefore, experiments show that these two feature sets can be proficiently combined. (3) Unsupervised feature extraction unusually achieves a projecting performance score; they achieved 85.72% accuracy for CNN and 84.9% for SVM.

3. Methodology

3.1 Deep learning libraries and frameworks

Two main frameworks and libraries are used to build the system's core model: the Keras library and TensorFlow [14] framework version 2.0 as the backend to Keras.

3.2 Datasets

In this system, and to get a more accurate result, two sets of data have been used to train the model: the first one is under the name "Accident Images Analysis Dataset," downloaded from the GitHub website [15], with a size of 41.3 MB and 4814 images with the extension "jpg." The diminutions of the image are (28, 28). The second dataset is downloaded from the Kaggle website [16] under the name "Accident detection from CCTV footage," has a size of 251 MB, and contains 990 images all with the extension "jpg." It’s a set of video frames recorded by the road surveillance cameras. Each image in this dataset has different dimensions, which makes it difficult to work on it directly, so these images have been resized to equal dimensions for all dataset images, which are (32, 32).

3.3 The Systems Core Model

The system is constructed by merging two algorithms into one model called the hybrid model. The algorithms are convolutional neural networks (CNN) and support vector machines (SVM). The model architecture consists of four building blocks. Each block contains a number of layers of CNN algorithms. In the fourth block, the SVM algorithm was built as one of the CNN algorithm layers, which is the classifier layer where the final classes of the data are predicted.

3.3.1 CNN-SVM hybrid model

A CNN-SVM hybrid model is planned to combine both CNN and SVM benefits. The overall architecture of CNN with layers has been established. The configuration of the collective CNN-SVM model is going to be done by swapping the latest layer of CNN with the SVM model; then, the production of the fully connected layer of CNN is heading to SVM to make better enhancements to the classification. The main reason for combining the benefits of CNN and SVM is that the CNN model has always been the best because it is easy to add hidden layers that help with extracting features and improving accuracy and performance as a whole. However, SVM, with its exceptional appearance in extracting features, also has high effectiveness and speed among other algorithms [17]. After CNN's training, the fully connected layers are swapped by the SVM classifier to execute classification. SVM takes the outputs from CNN's hidden layer as a feature vector for the training phase. The classification phase is achieved by SVM on the test set using the mechanically extracted features. The process of the CNN-SVM combined model can be summarized as follows: After the input data is fed to the CNN classifier for training, a feature vector can be automatically extracted. The fully connected layers of CNN are swapped by the SVM classifier, which will be trained using the automatically extracted feature vectors. In the test phase, a given input map is fed to CNN to get a test feature vector. The classification is completed by the SVM classifier using the given test feature vector [18]. The structure of the CNN-SVM hybrid model is shown in Figure 2 below.

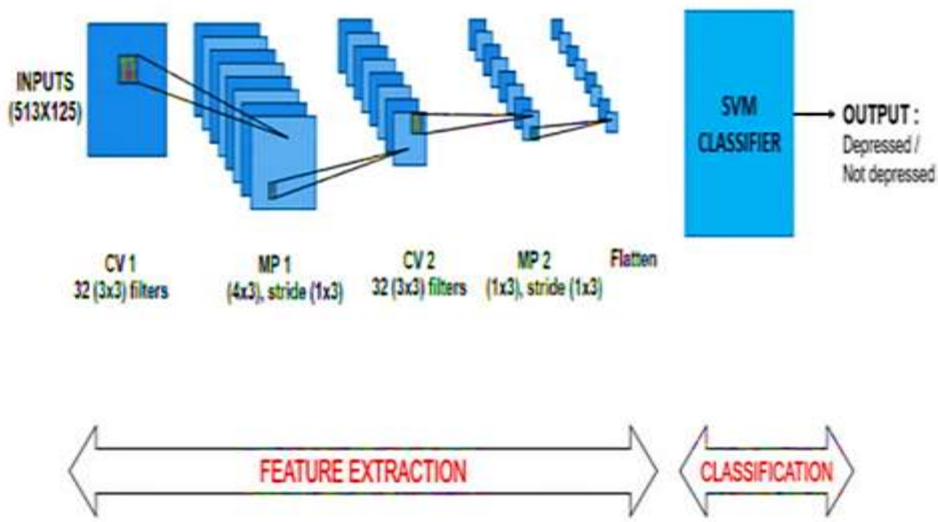

Figure 2: Architecture of the CNN-SVM hybrid model [18].

3.3.1.1 Convolutional neural network (CNN) algorithm Architecture

A convolutional neural network (CNN) is a multilayer neural network, or DL architecture, stimulated by the visual system of existing creatures [19]. CNNs have been exposed to their capable presentation on numerous actual applications. It has been known that the performance of CNNs is extremely dependent on their architectures, such as how many building-block layers are used, how the used building-block layers are collected, and how the parameters related to the used building-block layers are identified [20]. The CNN algorithm in the CNN-SVM model is constructed by four building blocks, each of which contains a number of layers and activation functions, as shown in Table 1 below.

Table 1: CNN Building Block Architecture Built in the Proposed System

Blocks	Number of layers	Type of layer	Is activation function added	Number	Type
Block-1	4	1.Convelution 2.Convelution 3. Max-Pooling 4. Dropout	yes	2	Relu
Block-2	4	1.Convelution 2.Convelution 3. Max-Pooling 4. Dropout	yes	2	Relu
Block-3	3	1. Convolution 2. Max-Pooling 3. Flatten	yes	1	Relu
Block-4	3	1. Dense 2. Dropout 3. SVM	yes	2	Relu Sigmoid

There are three frequently used activation functions: sigmoid, tanh, and ReLU. The function outputs for sigmoid and tanh converge to a single constant such as 0, 1, or -1, resulting in disappearing gradients if the absolute values of initial weights are too large. Going through ReLU activation, negative values are gone, while large positive values result in exploding gradients. If the absolute values of initial weights are too small, some simple networks work. Still, as the network goes deeper, the weights lie in a very small range of values, so that gradients become nearly machined and learning becomes problematic. The non-linearity performance of CNN's layers allows the CNN model to learn more complex things and manage to map inputs to outputs nonlinearly. The important feature of an activation function is that it should be differentiable in order to enable error-back propagation to train the model [21]. The most commonly used activation function in our time is the ReLU function. ReLU is commonly used over tanh and sigmoid, and the SoftMax function is a smooth ReLU function [22]. Equations of sigmoid, relu, and tanh activation functions are shown below.

$$f(x) = \frac{1}{1+\exp(-x)} \tag{1}$$

$$f(x) = \max(0, x) \tag{2}$$

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3}$$

3.3.1.2 Support vector machine (SVM) algorithm

SVM is a sparse technique that requires all the training data to exist in memory through the training stage, when the parameters of the SVM model are learned. Though once the model parameters are acknowledged, SVM will be subject to only a subsection of these training cases, called support vectors, for upcoming predictions. SVM is also a kernel technique. SVM uses the kernel to plot the data in a higher space before describing the machine learning task as a curving optimization problem in which targets are found systematically instead of heuristically, as with other machine learning techniques. Often, real-world data are non-linearly divisible in the original input space. In other arguments, cases that have different labels share the input space in a way that stops a linear hyperplane from properly unraveling the different classes tangled in this classification task [23]. The main equations of this algorithm and the parameters that are constructed by its equations are listed below.

$$H_{0=W\,.\,X+b=0} \tag{4}$$

$$H_{1=W\,.\,X+b=+1} \tag{5}$$

$$H_{2=W.\,X+b=-1} \tag{6}$$

where H1 and H2 are the hyperplanes and H0 is the median plane, w is the weight of x, and is the feature of the equation defining the decision surface separating the classes.

$$w^T x + b = 0 \tag{7}$$

where w is a weight vector, x is the input vector, and b is the bias. The SVM algorithm attempts to discover the finest unraveling plane, i.e., the one utmost from both classes +1 and -1. So, SVMs concurrently maximize the distance between two parallel supporting planes for each class and minimize the errors using the equations below [24].

$$margin = 2/\|w\| \tag{8}$$

3.4 Measures of system evaluation

The performance of the projected technique is evaluated by matching the trained models with diverse measures. The excellence of the learning algorithms is mostly evaluated by studying how well they perform on test data. [25]. *Accuracy* is the ability to distinguish between the different classes correctly. To estimate the accuracy of a network, the quantity of true positive and true negative among all samples is computed, as given by Equation (9).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

where TP refers to the true positives, which are the number of cases that are positive and are properly recognized, and FN represents the false negatives, which are the number of positive cases that are misclassified as negative. TN is the number of true negatives, or negative cases that are negative and classified as negative, and FP is the number of false positives, defined by the negative cases that are incorrectly classified as positive [26]. Loss function Loss is the difference between the ground truth of the inputted image and the output image provided by the CNN, which means it's the difference between the dataset before training, which is the ground truth, and the predicted class after the training [27]. The loss function, which is called an estimation function, estimates the stability of the network's output estimates by assigning ground-truth labels and forward propagation [27]. The equation (10) below shows how to calculate the loss.

$$Loss(\theta, I, O) = \sum_{i=1}^{w_o} \sum_{j=1}^{h_o} |f(\theta, I) - O| \tag{10}$$

where θ are the CNN parameters, f is the transformation function learned by the CNN, and w_o and h_o are the width and height of the ground-truth image O.

4. Vehicles Accidents Detection and Notification system Architecture

As the name implies, the main purpose of this system is to detect accidents and generate a notification signal for the nearest medical assistance station. To fulfill this purpose, it's required that several preprocessing steps be applied to the selected dataset and the chosen classifier to predict the final class, and depending on this class, the notification will be generated or not. These entire steps are shown in Figure 3 below.

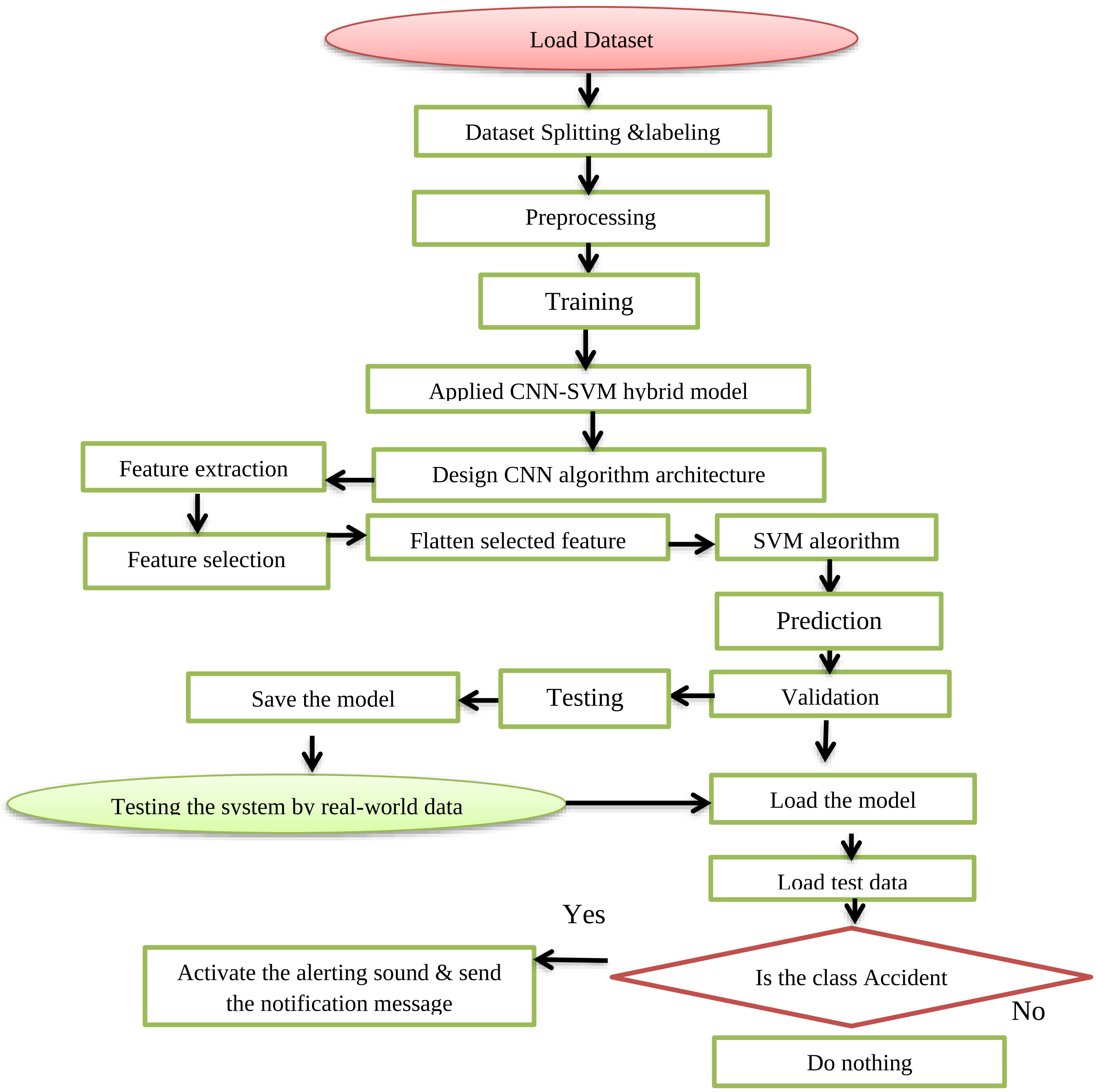

Figure 3: Vehicle Accident Detection and Notification System Architecture

1. Load dataset: The dataset loading is described in section 3.2

2. Dataset splitting and labeling: According to each image's content, it will be split into two classes, with class-id labeled by (0) and class name "No-Accident" for the images that don't contain an accident scene and class-id labeled by (1) and class name "Accident" for the images that contain an accident scene. All this information is organized as a CSV file that is used later by the classifier to predict the classes. After labeling the datasets, each of the previous datasets was divided into three sets (training set, testing set, and validation set) according to the splitting ratio of 70% for training, 30% for testing, and 15% for validation.

3. Preprocessing: After labeling and splitting the dataset, images must go through several stages of preprocessing before they are ready to be fed to the model as input. These preprocessing stages will fix the problems in the images and make them appropriate for more accurate results. The preprocessing stages were resizing the images, conversion to grayscale, histogram equalization, normalization, and image data generation (augmentation).

4. Training the model: After describing the design detail of each block and listing the layers and the activation functions that belong to them in Table 1, the layers' mechanism as well as the activation function, the inputs, outputs, and parameters for each layer All this information is explained in the number of points listed below.

a. In bkock-1, the first convolution layer is the input layer, where the training set, after passing through all the preprocessing stage steps, will be input to the CNN model through the convolutional layer. In addition to the data, there are other parameters that must be defined at this layer, which will be the same for all convolutional layers in the model. These parameters are: a. Padding, which is the number of pixels added to an image when it is convolved with the kernel, is defined in this model as (padding = same), which means that the output image size will be the same as the input. b. Stride is the number of pixels that are shifted every time the filter moves along the image array, so if stride = 2, the filter will move by two pixels every time. In addition to these parameters, there is the activation function because the classification type is binary, which means there are only two classes, and according to that, the used activation function is rectified linear (Relu), which is a linear function where the output will appear directly if it's positive or it will output zero.

b. Feature extraction is the output of convolution layers and the most important operation in classification because the classifier depends on these features to recognize the patterns and objects in the images. The features are extracted by performing the convolution process between the filters and the image array. The resulted array is called a feature map, which contains the features of this image, which could be an array containing the pixels' values in the configuration of the object's outline or edges. The resulted feature depends on the type of filters used (edge filters, mean, median, and so on). After the first extraction, this feature will pass through another convolution layer for another round of extraction. This will reduce the number of features and give more accurate features.

c. In pooling layers where Max-pooling has been used, there is one parameter, which is pool-size, which determines the number of pixels from the feature map to apply the pooling operation to. It is applied with the stride, which means if pool-size = (2, 2), this means stride = 2 and pool-size = 2, which means two pixels will be selected every time, and by using the max-pooling operation, the pixel with the larger intensity will be taken. The purpose of this layer is to reduce the size of features that have been extracted before by selecting the important ones according to pooling layer type.

d. Feature selection It's the next most important operation in the classification after feature extraction, as explained before. The pooling layer is the one that performs the feature selection by applying one of its feature section techniques, like max-pooling or average pooling. The feature selection is important because: a. it reduces the number of features; b. it speeds the model training; c. it reduces the time and memory required for training; d. it increases the accuracy score.

e. In the dropout layer, where the model will do organized training with a high accuracy score and reduce the loss by avoiding overfitting, it will ignore the result of Chapter Three Vehicle Accident Detection and Notification System 64 and a set of nodes based on the dropout rate, which is the parameter used in this layer. For example, if the dropout rate is 0.2, this means that 20% of the nodes will be ignored.

f. In the flatten layer, where there is no parameter in this layer, the extract feature before, which is a 2D array, is going to be converted to a 1D array to be ready to be interconnected with the fully connected layer (dense).

g. In the dense layer, which is the fully connected layer, this layer is going to connect the 1D array items that resulted from the flatten array to be inputted to the classifier. The parameters in this layer are the units, which are the number of nodes or neurons, as well as the activation function Relu.

h. Support vector machine (SVM) It's the last layer in the CNN-SVM hybrid model where the data is going to be classified and the prediction of the class made. The CNN-SVM hybrid model will not be treated as a separate model; it will be built in as one of the CNN algorithm layers, so it will use the same parameters that have been used in other layers as well as the activation function, and it will train along with CNN as one of its layers. As observed during the training process, it achieved the best among them, especially in training time.

5. Training, Validation and Testing Experimental Results

The training and validation operations happen concurrently at the same training iteration (epochs = 50), so their evaluation results are produced at the same time, and after that, testing results are produced below the experimental and evolutionary results for the first and second datasets. Each image in both datasets is augmented with five additional images.

Splitting the dataset according to the ratio of 70% of the dataset for training, 30% for testing, and 15% for validation, as shown in Table 2 below,

Table 2: Splitting of the first dataset according to 70% and 30% ratios and augmentation

Sets	Number of images	Augmentation
First dataset	4814	24070
Training set	3477	17383
Testing set	723	3615
Validation set	614	3070

According to the splitting ratio mentioned above, the number of images that belong to the training set is 3477. This set is divided into two sets of images: the first contains 1742, which is the image labeled "No-Accident Class (0)," and the second contains 1753, labeled "Accident Class (1)." This dataset is also divided into three sets according to the splitting ratio of 70% for training and 30% for testing and validation, as shown in Table 3 below.

Table 3: Splitting of the second dataset according to the 70% and 30% ratios and augmentation

Sets	Number of images	Augmentation
second dataset	990	4950
Training set	714	3570
Testing set	149	745
Validation set	126	630

The number of images that belong to the training set is 714. This set is divided into two sets of images. The first contains 379 images, which are labeled "No-Accident Class (0)," and the second contains 335 images labeled "Accident Class (1)." Table 4 below shows the numerical experimental results, and its illustrator results are shown in Figures 4 and 5.

Table 4: The results of the CNN-SVM model during the training, validation, and testing stages

CNN-SVM hybrid model						
Datasets	**Training Accuracy**	**Training loss**	**Validation Accuracy**	**Validation loss**	**Testing accuracy**	**Testing total loss**
First dataset	99.74%	1.14%	93.16%	28.69%	90.73%	43.07%
Second dataset	98.88%	3.39%	96.03%	12.05%	93.28%	33.46%

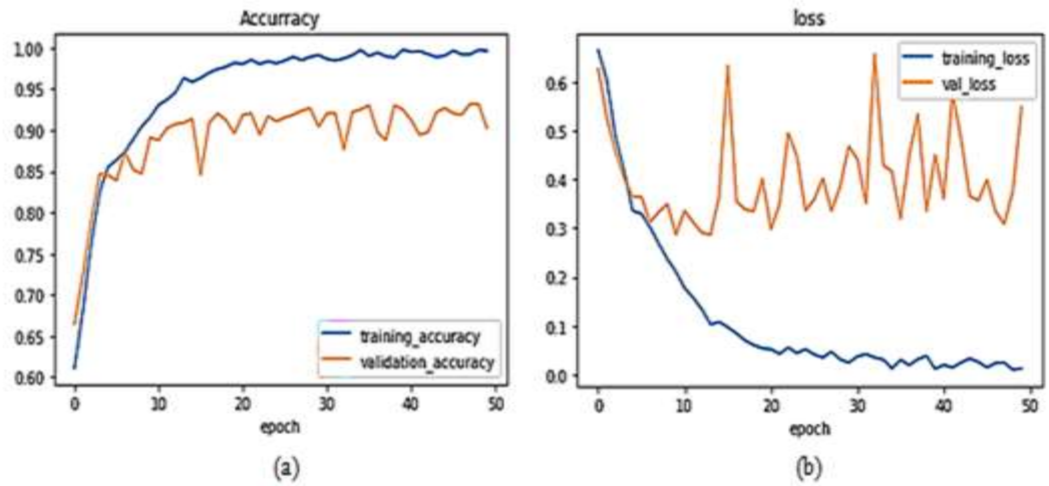

Figure 4: The illustration figures of (a) the accuracy scores and (b) the loss scores of the first dataset

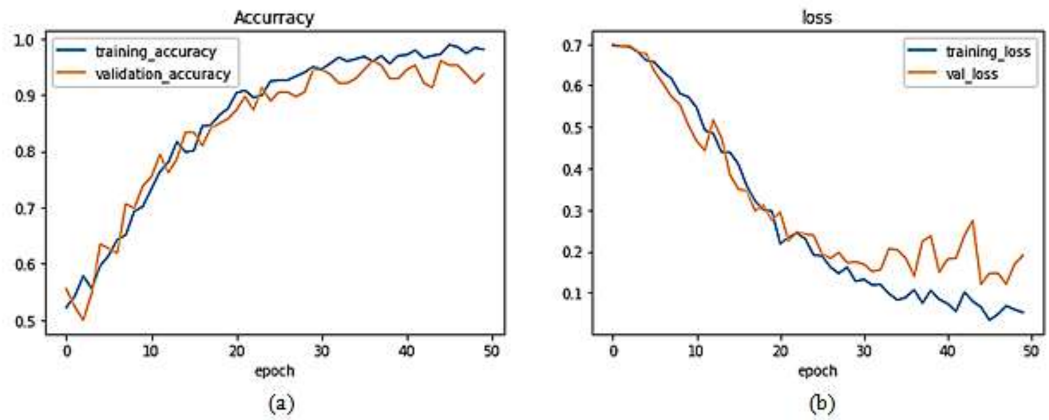

Figure 5: The illustration figures of (a) the accuracy scores and (b) the loss scores of the second dataset

5.1 Testing the system by real-world data

After testing the system by using the same datasets used during training and validation, a set of real-world data like videos and images of different sizes will be used to test the efficiency of the system at accomplishing the purpose and goal that it has been designed for, which is to detect accidents happening on the roads. During the testing operation, the model that has been trained previously and stored under the name and extension "Accident-detection.model" will be imported and used to classify the images and videos that have been captured from the roads based on the information that was extracted during the training stage, as shown in Figure 6 below.

Figure 6: Testing results based on real-word data from different distances and angles

5.2 The Form of Notification Signal

The system will generate two kinds of notifications: the first is a sound alert notification, and the second is the last stage of the system's work, where the system is going to use the sending mail transfer protocol (SMTP) by embedding the server host address, which is smtp.gmail.com, and the port number (587), in the code of the system. The sending of the message depends on the status of the entered data; if it was an accident, then the system is going to send the notification message; otherwise, it will not be sent. The message will be delivered as a Google Mail message (Gmail), and the Gmail address will be accidentnotification@gmail.com. This address was created especially for the system. The message content can be any statement that can inform the recipient about the accident; it could be something like "accident happening at the sender location." This means that the form of the notification signal will be a text message, as shown in Figure 7 below.

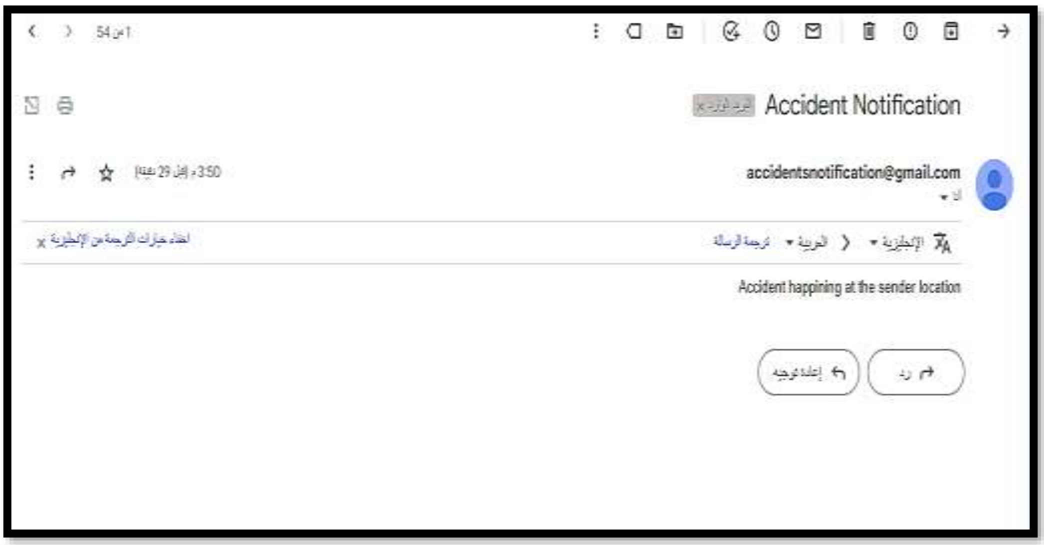

Figure 7: The received notification message form

6. Discussion

After presenting the important aspects of the system's design and implementation results, we would like to highlight the system's strengths. While testing the system with real-world data, it proved its efficiency by fulfilling its aim by discovering the accidents early and reducing them by releasing the alerting sound to inform the other user of the road. In addition to that, it achieved better results compared to the related works, as shown in the table below. Regardless of these strengths, the system has some weak points, which are that it needs large memory space because it consumes 10% of the device memory and that it totally depends on some hardware equipment like cameras, microphones, and internet connections. Without these pieces of equipment, or at least one of them, the system will not work properly. Table 5 below shows the comparison points between the proposed system and the related works.

Table 5: Comparison between the designed system and related work systems

Ref	Authors, Year	Objective	Methods & Models	Data	Results
[8]	Zhenhua Zhang,et al,2018	Detecting the traffic accident from social media data	1. DBN 2. LSTM 3. SVMs 4. sLDA	3million tweet contents	Overall accuracy 85%
[9]	DAXIN TIAN,et al,2019	Automatic car accident detection based on (CVIS) and machine vision	1.YOLOCA 2. MSFF	CAD-CVIS	Detecting time 0.0461 seconds (21.6FP) and 90.02% AP
[10]	Zhenbo Lu,et al,2020	Detecting traffic crashes and balancing detection speed and accuracy with limited computational resources	1.ResNet 2.Conv-LSTM	video clips	87.78% accuracy
[11]	Jae Gyeong Choi,et al,2021	Develop a car accident detection system	1.ensemble technique 2.CNN 3.GRU	Car accident YouTube clips	98.60% accuracy
[12]	Vedika Sadavarte,et al,2022	Direct a self-operating accident detection system with alert generation to offer timely assistance in critical situations	CNN	Surveillance cameras data	Loss less than 20%
[13]	Hawzhin Hozhabr Pour,et al,2022	Automated car accident detection based on multimodal in-car sensors.	1. CNN 2. SVM	(NDS) crash dataset	CNN=85.7% SVM=849% accuracy
The proposed system		**Accidents detection and generate notification signals**	**CNN-SVM hybrid model**	**1.First dataset 4814 images 2.second dataset 990 images**	**99.74% first dataset accuracy 98.88% second dataset accuracy**

7. Conclusions and Suggestions for Future Works

This section presents briefly the main ideas and objectives that the system has been built for, which are detecting road traffic accidents and sending a notification message to the specialists to deal with the issues that arise from the accidents, and then summarizes the outcomes that have resulted from this system. The system was built based on machine learning techniques and methods, especially deep learning methods according to the supervised deep learning approach. The model that has been employed to classify the datasets is the CNN-SVM hybrid model, where two algorithms work together as one model by building the SVM algorithm inside the

CNN algorithm as one of its layers. Two datasets have been used to train the model; each dataset has a specific number of images. The first dataset has 4814 images, and the image dimensions are (28, 28), and the second dataset has 990 images, and the image dimensions are (32, 32). The datasets are divided into three sets: training, testing, and validations, according to the splitting ratio. 70% for training and 30% for testing and validation, and then the images go through many preprocessing procedures like resizing, conversion to gray-scale, histogram equalization, normalization, and image data generation before they are interred as the first building blocks of the model for training. The trained model will be saved under the name and extension "accident detection model." The results of the training stage are 99.74% of training accuracy, 93.16% of validation accuracy, and 90.73% of testing accuracy by standard dataset for the first dataset, and 98.88% of training accuracy, 96.03% for validation, and 93.28% for the second dataset. The system loaded the previously trained copy of the model and tested it against real-world data, videos, and images. The system predicted the accident's status with the best accuracy score of 100%, activated the alerting sound, and sent the notification message during the first 30 seconds after the accident happened.

Actually, this system has achieved its goal with good results, but it needs to be continually developed to keep up with the rapid advancements in the world and continue to present its services effectively. Some suggestions of how this system can be developed in the future include designing an accident detection system that can detect the accident locations and severity of injuries in addition to the accident detection and notification messages. Using the latest deep learning techniques while designing the system, like the TensorFlow 2.0 API and the pre-trained object detection models, impeded the system with vehicles' main systems to notify the accident automatically and not with road surveillance cameras only.

References

[1] R. G, T. Roy, V. N. V, S. Shaju, and A. R. Paul, "Semantic Video Mining for Accident Detection," *Int. J. Innov. Sci. Res. Technol.*, vol. 5, no. 6, pp. 670–678, 2020, doi: 10.38124/ijisrt20jun432.

[2] M. Naseer and J. Naz, "AUTOMATED ACCIDENTS ON ROAD ANALYSIS: AN OVERVIEW OF STATE OF THE INSIGHTS," *Azerbaijan J. High Perform. Comput.*, vol. 4, no. 2, pp. 242–262, 2021, doi: 10.32010/26166127.2021.4.2.242.262.

[3] V. E. Machaca Arceda and E. Laura Riveros, "Fast car crash detection in video," in *Proceedings - 2018 44th Latin American Computing Conference, CLEI 2018*, 2018, pp. 632–637. doi: 10.1109/CLEI.2018.00081.

[4] M. Abdou *et al.*, "End-to-end crash avoidance deep IoT-based solution," *Proc. Int. Conf. Microelectron. ICM*, vol. 2019-Decem, pp. 103–107, 2019, doi: 10.1109/ICM48031.2019.9021613.

[5] A. Yadav, V. More, N. Shinde, M. Nerurkar, and N. Sakhare, "Adaptive Traffic Management System Using IoT and Machine Learning," *Int. J. Sci. Res. Sci. Eng. Technol.*, pp. 216–229, 2019, doi: 10.32628/ijsrset196146.

[6] P. Control and N. Programming, "Visvesvaraya Technological University , Belgaum Visvesvaraya Technological University , Belgaum," *Digit. Signal Process.*, no. 1, pp. 1–6, 2009.

[7] S. Olugbade, S. Ojo, A. L. Imoize, J. Isabona, and M. O. Alaba, "A Review of Artificial Intelligence and Machine Learning for Incident Detectors in Road Transport Systems," *Math. Comput. Appl.*, vol. 27, no. 5, p. 77, 2022, doi: 10.3390/mca27050077.

[8] Z. Zhang, Q. He, J. Gao, and M. Ni, "A deep learning approach for detecting traffic accidents from social media data," *Transp. Res. Part C Emerg. Technol.*, vol. 86, pp. 580–596, 2018, doi 10.1016/j.trc.2017.11.027.

[9] D. Tian, C. Zhang, X. Duan, and X. Wang, "An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems," *IEEE Access*, vol. 7, pp. 127453–127463, 2019, doi: 10.1109/ACCESS.2019.2939532.

[10] Z. Lu, W. Zhou, S. Zhang, and C. Wang, "A New Video-Based Crash Detection Method: Balancing Speed and Accuracy Using a Feature Fusion Deep Learning Framework," *J. Adv. Transp.*, vol. 2020, 2020, doi: 10.1155/2020/8848874.

[11] J. G. Choi, C. W. Kong, G. Kim, and S. Lim, "Car crash detection using ensemble deep learning and multimodal data from dashboard cameras," *Expert Syst. Appl.*, vol. 183, p. 115400, 2021, doi: 10.1016/j.eswa.2021.115400.
[12] P. U. Ghorpade and A. Nema, "Accident Detection and Alert Generation Using Raspberry Pi," *SSRN Electron. J.*, vol. 24, no. 4, pp. 288–297, 2022, doi: 10.2139/ssrn.4184576.
[13] H. H. Pour *et al.*, "A Machine Learning Framework for Automated Accident Detection Based on Multimodal Sensors in Cars," *Sensors*, vol. 22, no. 10, pp. 1–21, 2022, doi: 10.3390/s22103634.
[14] TensorFlow Organization: https: //www. TensorFlow. o r g/install/pip
[15] GitHub:https://github.com/mghatee/Accident-Images-Analysis-Dataset/blob/master/Accident%20Images%20Analysis%20Dataset.rar
[16] Kaggle: https://www.kaggle.com/datasets/ckay16/accident-detection-from-cctv-footage
[17] M. O. Khairandish, M. Sharma, V. Jain, J. M. Chatterjee, and N. Z. Jhanjhi, "A Hybrid CNN-SVM Threshold Segmentation Approach for Tumor Detection and Classification of MRI Brain Images," *IRBM*, vol. 43, no. 4, pp. 290–299, 2022, doi: 10.1016/j.irbm.2021.06.003.
[18] A. Saidi, S. Ben Othman, and S. Ben Saoud, "Hybrid CNN-SVM classifier for efficient depression detection system," *Proc. Int. Conf. Adv. Syst. Emergent Technol. IC_ASET 2020*, pp. 229–234, 2020, doi: 10.1109/IC_ASET49463.2020.9318302.
[19] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, "Fundamental concepts of convolutional neural network," in *Intelligent Systems Reference Library*, vol. 172, no. January, 2019, pp. 519–567. doi: 10.1007/978-3-030-32644-9_36.
[20] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Completely Automated CNN Architecture Design Based on Blocks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 4, pp. 1242–1254, 2020, doi: 10.1109/TNNLS.2019.2919608.
[21] H. Lee and J. Song, "Introduction to convolutional neural network using Keras; An understanding from a statistician," *Commun. Stat. Appl. Methods*, vol. 26, no. 6, pp. 591–610, 2019, doi: 10.29220/CSAM.2019.26.6.591.
[22] C. Zhang *et al.*, "A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 140, pp. 133–144, 2018, doi: 10.1016/j.isprsjprs.2017.07.014.
[23] M. Awad and R. Khanna, "Efficient learning machines: Theories, concepts, and applications for engineers and system designers," *Effic. Learn. Mach. Theor. Concepts, Appl. Eng. Syst. Des.*, no. ResearchGate, pp. 1–248, 2015, doi: 10.1007/978-1-4302-5990-9.
[24] T. N. Do, V. H. Nguyen, and F. Poulet, "Speed up SVM algorithm for massive classification tasks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5139 LNAI, pp. 147–157, 2008, doi: 10.1007/978-3-540-88192-6_15.
[25] V. Maeda-Gutiérrez *et al.*, "Comparison of convolutional neural network architectures for classification of tomato plant diseases," *Appl. Sci.*, vol. 10, no. 4, 2020, doi: 10.3390/app10041245.
[26] D. C. Malo, M. M. Rahman, J. Mahbub, and M. M. Khan, "Skin Cancer Detection using Convolutional Neural Network," *2022 IEEE 12th Annu. Comput. Commun. Work. Conf. CCWC 2022*, pp. 169–176, 2022, doi: 10.1109/CCWC54503.2022.9720751.
[27] M. I. Georgescu, R. T. Ionescu, and N. Verga, "Convolutional Neural Networks with Intermediate Loss for 3D Super-Resolution of CT and MRI Scans," *IEEE Access*, vol. 8, pp. 49112–49124, 2020, doi: 10.1109/ACCESS.2020.2980266.