Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
e-ISSN: 2788-7421 p-ISSN: 2958-0544

Detecting Abnormal Driving Behavior Using Modified DenseNet

Aisha Ayad[1]*, Matheel E. Abdulmunim[2]

[1,2] Computer Science Department, University of Technology, Baghdad, Iraq.

DOI: https://doi.org/ 10.52866/ijcsm.2023.02.03.005
Received April 2023; Accepted May 2023 ; Available online June 2023

ABSTRACT: Car accidents have serious consequences, including depletion of resources, harm to human health and well-being, and social problems. The three primary factors contributing to car accidents are driver error, external factors, and vehicle-related factors. The main objective of this paper is to address the issue of car accidents caused by driver error. To achieve this goal, a solution is proposed in the form of a modified version of the Dense model, called the 1Dimention-DenseNet, specifically designed to detect abnormal driving behavior. The model incorporates adapted dense blocks and transition layers, which enable it to accurately identify patterns indicative of abnormal driving behavior. This paper compares the performance of the 1D-DenseNet model to the original DenseNet model in detecting abnormal driving behavior in the Kaggle distracted driver behavior dataset. Results show that the 1D-DenseNet model outperforms the original DenseNet model in classification and validation accuracies, loss, and overhead. The 1D-DenseNet, after 100 epochs of training using Keras on top of TensorFlow, the 1D-DenseNet achieved a categorical cross-entropy loss of 0.19 on the validation set, with classification and validation accuracies of 99.80% and 99.96%, respectively. These findings demonstrate the effectiveness of the 1D-DenseNet model in improving the detection of abnormal driving behavior.
Keywords: Car accidents, Deep Learning, DenseNet , classification.

1. INTRODUCTION

Driving behavior is a critical factor that affects road safety. The complex and unpredictable nature of driving conditions and the driver's state of mind can lead to dangerous situations on the road, such as impaired driving, drowsy driving, and drunk driving. Common driving behaviors include accelerating, decelerating, turning, and changing lanes.

As per the World Health Organization (WHO), the annual global fatalities resulting from road accidents exceed 1.25 million. [1]. In Iraq, traffic accidents are the main death cause for people aged 15 to 29 [2]. Since 2003, the number of road accidents in Iraq had increased significantly due to the lack of traffic management techniques. In 2013, Iraq had a road accident death rate of 20.20, which is higher when compared to the global average of 17.40. Iraq ranks 113th in the world in terms of road traffic deaths, and it has the second-highest number of road traffic deaths in Eastern Mediterranean Region (EMR) as shown in Fig (1) [1].

Vehicular accidents can cost countries a significant portion of their economy, and abnormal driving behavior is a major factor in many of these accidents [1]. To detect such behavior, advanced technologies such as sensors, cameras, and machine learning algorithms can be utilized [2]. Deep learning models have shown remarkable performance in image recognition, but they require extensive amounts of data and computation time [3]. However, techniques such as transfer learning, pruning, and quantization can reduce the computational demands of DL [4]. In addition, numerous studies have investigated ways to increase effectiveness and efficiency of the deep learning in various domains such as image classification, speech recognition, and machine translation [5]. By emulating the human brain's ability to extract meaningful features, deep learning can improve representation and performance, enabling better decision-making in a broad range of the applications [6].

To address the issue of abnormal driving behavior, this paper proposes a 1D dense net model for detecting distracted driving behaviors using the State Farm Distracted Driver Data-set (SDDD) images from Kaggle. The model was trained and tested on 22,424 images split into 70% for training and 30% for testing. The primary contributions of this study include improved identification and categorization of anomalous driving behaviors using the 1D dense net and the development of a preprocessing stage to enhance efficiency.

*Corresponding author: cs.21.11@grad.uotechnology.edu.iq

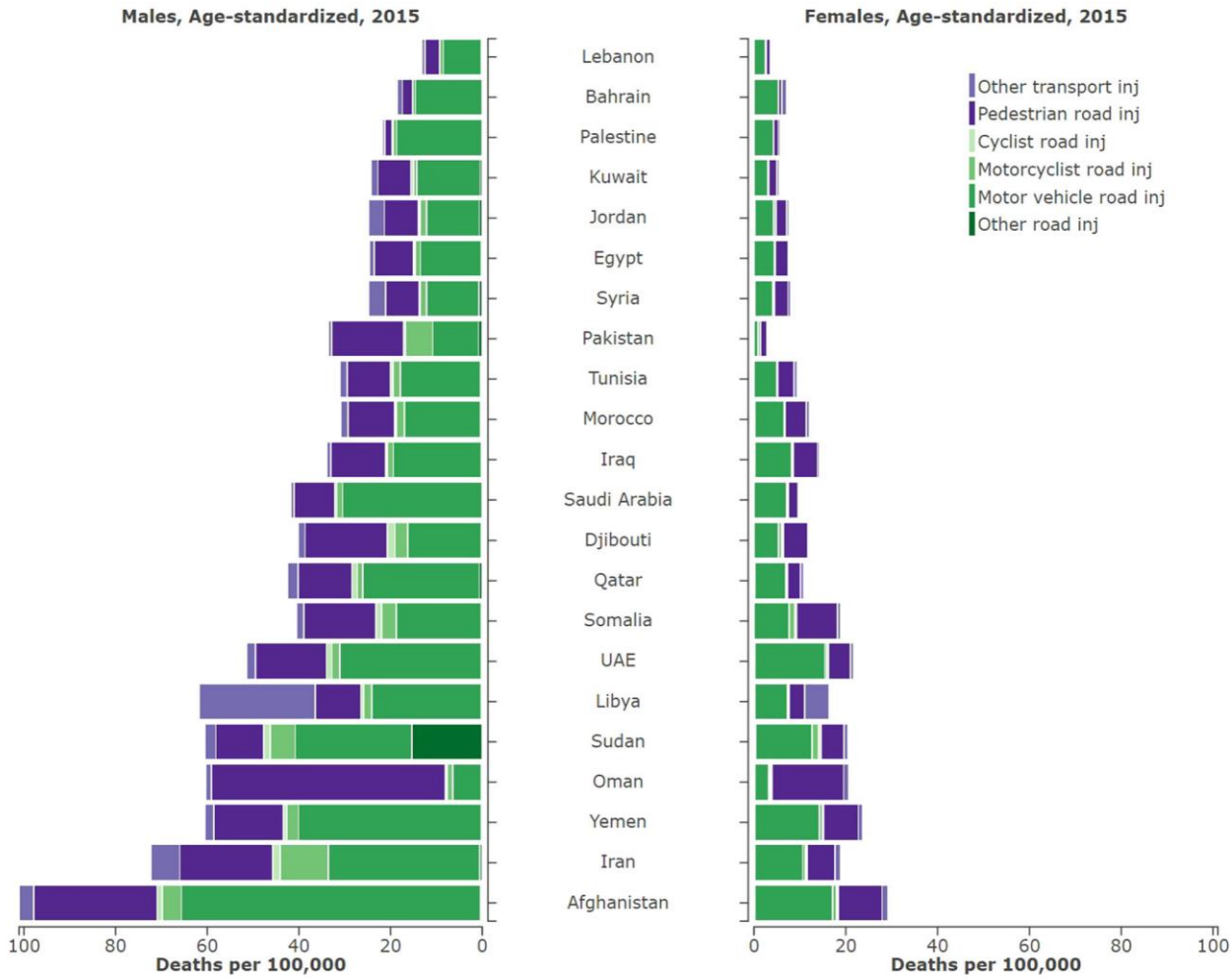

FIGURE1. Rates of death from transportation-related injuries, adjusted for age, in the 22 countries of the Eastern Mediterranean region, broken down by gender [1]

Additionally, the 2D DenseNet architecture was modified to create a more efficient 1D architecture that reduced memory and Central Possessing Unit (CPU) usage and allowed for faster training and classification speeds. By utilizing deep learning's ability for the extraction of the meaningful features and improve representation and performance, the proposed model can better identify and categorize abnormal driving behavior, ultimately leading to safer roads for everyone.

The following structure has been adopted for this present manuscript: Section 2 refers to the literature analyses that were presented earlier. Section 3 provides a brief description of the dense net, while Section 4 offers an in-depth discussion of the proposed model, encompassing preprocessing, feature extraction, and classification procedures. Section 5 presents a concise summary of the dataset employed in the study, while Section 6 provides an outline of the evaluation metrics utilized. The results of the proposed model are elaborated upon in Section 7, while final remarks are provided in Section 8.

2. RELATED WORK

The aim of abnormal driving detection is to ensure the safety of the driver, passengers, and people on the road, and to minimize fatal accidents as much as possible. Numerous studies have attempted to detect abnormal driving behavior using various techniques and algorithms, and some of these investigations will be discussed.

Thakur et al. [7] developed a system that uses a CNN to estimate the driver's pose and alert them if they are distracted by any activity. They used various regularization techniques to reduce overfitting. They achieved 96.16% accuracy for 10-class classification and 99.12% for binary classification, using transfer learning and regularization. Aljasim et al. [8] introduced Ensemble-based Driver Distraction Recognition (E2DR), a scalable model that combines two or more deep learning models using stacking ensemble methods. The best E2DR variant, which included ResNet50 and VGG16 models, achieved 92% test accuracy on the State Farm Distracted Drivers dataset, using new data splitting strategies. The recall, precision, and F1 score were 0.92, 0.91 and 0.92 respectively. Furman et al. [9] proposed a

method for detecting the driver's aggressive emotion based on a CNN model. The model takes in the driver's facial images from an Near InfraRed spectroscopy(NIR) light camera and a thermal camera.

Lee et al. [10] developed a method to accurately categorize drivers' actions by combining the scores from Convolution Neural Networks (CNNs) on images captured by NIR and infrared cameras, its approach surpassed existing methods with a 99.95% accuracy rate in classifying aggressive and smooth driving behaviors.

Hu et al. [11] developed a system to classify driving behaviors into normal, intoxicated/fatigued, reckless, or phone-using. This system achieved high micro-recall and micro-precision of 92.66% and 92.69%, respectively, and demonstrated its effectiveness in detecting abnormal driving using a new dataset. Alam Noor et al. [12] developed a deep-learning model based on the DenseNet 3D backbone to detect car drifting (aggressive driving) using traffic footage, the model achieved 96.87% accuracy on the car drifting dataset. Darapaneni et al. [13] tweaked a conventional CNN achieving up to 99.5% precision in predicting unusual driving patterns from smartphone sensor data. Li et al. [14] combined CNN and Long Short-Term Memory (LSTM) to create a recognition model for anomalous driving behavior. The model achieved an accuracy of 95.22% and outperformed models constructed with CNN or LSTM alone.

Eraqi et al. [15] achieved a 90% accuracy rate using an ensemble of CNNs. Alotaibi et al. [16] trained a CNN on the State Farm Distracted Driver Detection dataset and achieved a 98.76% accuracy rate. Nasari et al. [17] proposed DistractNet, which achieved an accuracy of 99.32%. Vaegae et al. [18] developed a real-time distracted driver detection framework using Residual Network (ResNet-50), which outperformed VGG-16 with an accuracy of 87.92.

Roy et al. [19] used three deep learning models (CNN, Deep Residual Network, and Visualized Geometric Group16) to detect abnormal driving behavior from dashboard camera images. Their method achieved 87.44% accuracy, higher than the previous deep learning fusion method with 82% accuracy. Chawan et al. [20] proposed three CNN models and achieved a 91.2% accuracy rate using the State Farm distract driver data set .Table (1) illustrates benefits and limitations of previous studies.

Table 1. Comparison of previous work.

Ref	Benefits	Limitations
[7]	Using transfer learning, regularization techniques.	High computational coast.
[8]	Combined skills and strong points of various models into one. Used the stacking ensemble approaches for improving accuracy, enhancing generalization, and reducing overfitting.	Developing E2DR models from scratch has a computational complexity limitation.
[9]	The system uses Heart Rate Variability (HRV) as a non-invasive and cost-effective approach to detect when a driver is about to fall asleep. This allows for timely intervention to prevent accidents, ultimately leading to a reduction in fatigue-related accidents and fatalities.	Could lead to false alarms, requires the driver to wear a device to monitor HRV, could be unreliable in some scenarios, and could be misinterpreted by drivers.
[10]	Thermal cameras can measure the variations of the temperature in the driver's body, which cannot be checked by the naked eye, while, an NIR camera has the ability of detecting the facial feature points and measuring their variations. An intensively trained CNN is robust in different environmental and driver conditions.	Using two cameras adds complexity to the algorithm and increases the processing time required for analysis. To achieve precise and efficient detection, it is crucial to undertake extensive training of Convolutional Neural Networks (CNNs).
[12]	Enhanced safety: By detecting aggressive driving behaviors, DriftNet can help reduce the risk of accidents and improve overall safety on the roads.	Computational power of GeForce RTX 2070 GPUs, and the incomplete collection of the dataset.
[13]	Detecting driver distractions in real-time and alerting the user to prevent accidents, deploying the solution on an edge device setup on the vehicle to give an instant alarm, interacting through IoT to process data and give insights in asynchronous mode.	Data leakage resulting in lower accuracy on production data testing. Colab crashes due to the large volume and size of images. Lack of understanding of deployment of the model on 5G devices and execution in real-time.

[14]	Raw data gathered by on-board sensors are readily available in the market and easily accessible. This data can be utilized to detect and classify driving events. In terms of both lateral and longitudinal movements, pattern-matching algorithms outperform rule-based algorithms for identifying driving events.	Depending on the type of smartphone, the data collected may be limited or inaccurate. It can be difficult to accurately classify aggressive driving events due to the variability in individual driving styles. Depending on the phone's battery life, the data collected may be limited.
[15]	Utilizes a variety of image sources to provide more robust results, leverages transfer learning to reduce training time, offers an ensemble of convolutional neural networks to increase accuracy.	Potentially complex to implement and can be computationally expensive, results may be heavily dependent on the weighting of the networks.
[16]	Combining the three advanced deep learning models into one model increases efficiency. The inclusion of the ResNet, HRNN and inception modules provides a comprehensive range of features which can be used to classify data; the model includes two convolutional layers which can increase accuracy.	The complexity of the model may lead to slower processing times, the model may have a higher risk of overfitting due to the large number of parameters, training the model may be difficult and time consuming.
[17]	Achieves an overage accuracy of more than 99.32% -Classifies driver states into ten different classes most accurate model compared to other related work- significant performances in terms of training time and model size.	The proposed CNN model is not yet implemented in embedded systems to monitor driver states in real-time, raining time depends on many factors such as the number of image datasets, network architecture, and processing platform performances.
[18]	Combining two or more architectures and including face-based approaches helps to solve the misclassification problem.	Lowering the number of parameters and reducing computational time is necessary.
[19]	First attempt for the incorporation of recently suggested DenseNet into challenging abnormal driving behavior detection tasks, improvement of the width and cardinality in the WGD, sophisticated integration of the DenseNet and ResNet in AWGRD and WGRD.	Not all abnormal driving behaviors can be detected (e.g., those caused by the surrounding environment), not all mobile phone use is detected, as only important factors are considered, limited use of data and experiments to substantiate the superiority of the proposed models.
[20]	To improve accuracy, the selective cropping of specific parts of the images, such as hands and eyes, which contain informative details, can be done. By doing so, the focus can be placed on these crucial regions, enabling the model to achieve better performance in image recognition.	Since only a CPU from the Google Cloud Platform was used for the project, it is possible that the available computing resources were not sufficient to enhance the results.

3. DenseNet OVERVIEW

Yann LeCun developed CNNs in the late 1980s, which have since been widely used [21]. In 1998, LeNet [22] was proposed as the first deep learning model. In 2012, AlexNet [23], [24] was introduced, which outperformed the 2nd place finisher in ILSVRC by 10.9%. In 2015, ResNet was introduced to solve the issue of degradation of performance in deep models, and in 2017, DenseNet [25] was introduced to address issues with traditional CNNs. DenseNet uses batch normalization, skip connections, and transition layers to improve efficiency. Equation (1) can summarize its architecture:

$$x_l = \mathrm{H}_l([x_0, x_1, \ldots, x_l - 1]) \quad (1)$$

In a DenseNet, each layer in a dense block takes as input the concatenated outputs of all previous layers, denoted as $[x_0, x_1, \ldots, x_l - 1]$. These inputs are passed through a nonlinear transformation module, H_l, and the output is concatenated with the previous inputs to create the input for the next layer. Compared to a traditional structure, this design results in an L-layer model with more connections: specifically, L(L+1)/2 connections as shown in Fig (2). This increase in connections facilitates more efficient learning by strengthening the information flow between the layers [26] .

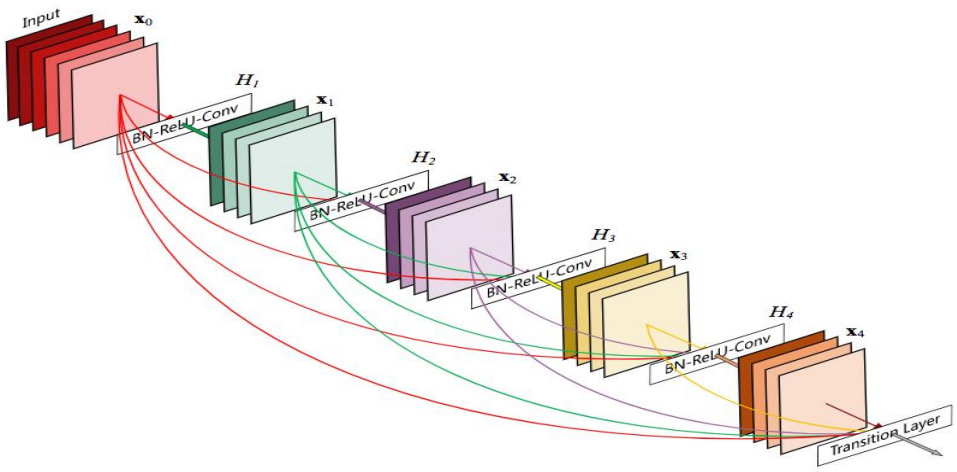

FIGURE 2. DenseNet Arichtecure [23].

4. THE PROPOSED MODEL

This paper presents a modified DenseNet, a new model for detecting abnormal driving behaviours, which consists of three stages, as shown in Fig (3):

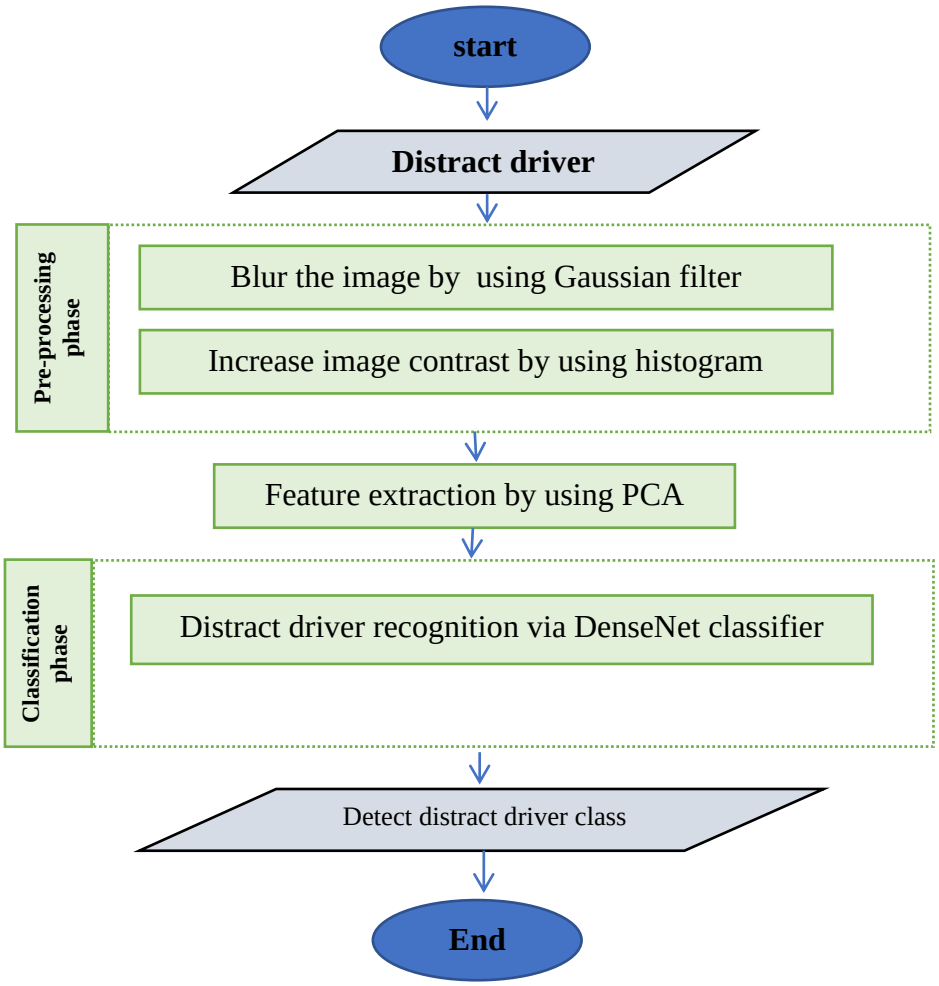

FIGURE 3. General structure of the proposed model.

[1] Pre- prossing phase: In this step, the initial input image is processed to ensure its suitability for further processing in the subsequent stage. To achieve this, the following operations were performed in this particular model for preparing the input image:

1. Convert the color image (consisting of three channels, i.e., red, green, and blue) into a grayscale image (consisting of one channel) by using Equation (2) [27]:

$$\text{Grayscale (i,j)} = (0.2989*r) + (0.5878*g) + (0.1140*b) \quad (2)$$

2. To improve the quality of the images for better processing, a Gaussian blur filter was applied to smooth the images. The specific Eequation used for this purpose is as follows(3)[28]:

$$Gb(i,j) = \frac{1}{2\Pi\sigma^2} e^{\frac{-i^2+j^2}{2\sigma^2}} \quad (3)$$

Where: i and j are coordinates in the x and y directions , σ controls the spread of the function, and $2\Pi\sigma^2$ is a normalization factor.

3. To enhance the contrast of the images and improve their visual quality, histogram equalization was employed. The corresponding equation for this operation is given as Equation (4):

$$\text{Hist(v)} = \text{cut}\left(\frac{(cdf_{(v)} - cdf_{min}}{(m*n)-1}\right) * (L-1) \quad (4)$$

Where: **Cdf** represents the cumulative distribution function, **(m, n)** represents the image's number of pixels, and L represents the number of gray levels used (256).

4. After resizing and cropping the preprocessed images to a uniform size, their size was reduced, and they were subsequently fed into the feature extraction step for further processing.

[2] Feature extraction: This model employs Principal Component Analysis (PCA) [29] as one of its techniques for feature extraction. The process of feature extraction is of utmost importance in the fields of image classification and pattern recognition, as it facilitates the elimination of superfluous and insignificant data from an image. This results in improved classifier performance by reducing interference and enhancing the accuracy of predictions. PCA is a multivariate data analysis technique that transforms a set of potentially correlated variables into a smaller number of uncorrelated ones, which are referred to as principal components. This method is commonly used for reducing the dimensionality of large data sets and is rooted in linear algebra. PCA [30] can be applied for various tasks, including denoising signals, blind source separation, and data compression. It involves solving an eigenvalue-eigenvector problem for a positive-semi-definite symmetric matrix and is a popular statistical tool utilized in virtually every statistical software package [31].

Data is transformed linearly into a lower-dimension subspace when PCA is applied. This is accomplished by increasing the data's variance, which results in a vector of orthogonal, uncorrelated basis groups. PCA is performed on a dataset comprising of d observations, each being n-dimensional and represented by z_1, z_2, ..., x_i € Run. The following procedures are utilized to carry out PCA [32].

1. The m-dimension mean vector *MV* is computed by Equation (5):

$$MV = \frac{1}{d} \sum_{i=1}^{d} z_i \quad (5)$$

2. The covariance matrix *CM* for the observations is calculated by Equation (6):

$$CM = \frac{1}{d} \sum_{i=1}^{d} (z_i - MV)\,(x_i - MV)^d \quad (6)$$

3.Calculate the eigenvalue and eigenvector by Using the Equation (7) below:

$$V^{-1}CV = D \quad (7)$$

Where :V is a matrix whose columns are the eigenvectors of C, $V^{-}$is the inverse of V,C is the original matrix that we want to diagonalizable, D is a diagonal matrix whose diagonal entries are the eigenvalues of C.

4. By performing an inner product on the data matrix, the resulting eigen-vectors can be sorted in a descending order based on their corresponding eigen-values in D, data is projected on these eigen-vector directions for the purpose of locating feature vector .As in the following Equation(8):

(8)

Projected data = $[V^T (X - x)^T]^T$

where : V is of $n \times n$ dimension, and every row of it is an eigen-vector. The features can may be obtained.

[3] 1D-DenseNet Classification

The 1D-DenseNet model uses densely connected CNN blocks with compact interconnections and transition layers to effectively reuse acquired features, decrease the necessary parameters for model training, and enhance performance while balancing model capacity and computational efficiency. The architecture generates new feature maps with a growth rate of 12, captures features at different abstraction levels, and maintains spatial resolution throughout the network. Fig (4) represents a proposed one-dimensional DenseNet architecture with a growth rate(k) of 12, four dense blocks, each repeated 12 times, and three transition layers.

a. Input Layer : The Input Layer has a shape of (None, 400, 1) and stores the grayscale image's raw pixel values with a resolution of 340*480 pixels. The images are proportionally reduced in size into 20*20-pixels to expedite training.

b. b. First convolution : The input data is processed with a Convolution-1D (Conv1D) layer of 24 filters, kernel size 3, and "same" padding. L2 regularization is used with a parameter of 0.001 to prevent large weights.

L2 regularization computes the L2 norm of the weights (the sum of squares) as given in Equation (9):

$$\textbf{Regularization}(w) = ||w||_2^2 = \sum_{i=1}^{n} w_i^2 \quad (9)$$

The regularization encourages small weights, thus helping to reduce overfitting [33]. The first convolutional layer is initialized with weights from a random Gaussian distribution to encourage the exploration of different weight configurations during training. This is represented by the normal distribution formula in Equation (10):

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{1}{2}x^2} \quad (10)$$

whereμ is the mean of the distribution， andσ is the standard deviation of the distribution.

c. Dense Block Layer (DB): A dense block is a block of interconnected layers in which each layer is connected to all the other layers. In 1D densenet, there are four Dense Blocks with Batch Normalization (BN) and ReLU activation layers. ReLU helps avoid the vanishing gradient problem and produces a sparser representation than other activation functions. A 1D convolutional layer is added, followed by a spatial dropout layer. The output of the convolutional layer is concatenated with the original input and used as input for the next layer.

d. Transition Layer (TL): The 1D-denseblock in the model includes three transition layers with batch normalization, 1x1 convolution, spatial dropout, and average pooling (Avg-pool) with a kernel size of 2. The pooling layer reduces spatial dimensions, parameters, processing, and optimizes memory. The transition layer is inserted between dense blocks, reducing channels and spatial size to decrease complexity, the function applies BN, ReLU, 1D-convolution, 1D-spatial dropout, and pooling, returning the average.

g. Fully Cconnected Layer (FCL): In the modified model, inputs are normalized with batch normalization, followed by ReLU activation and average pooling to reduce dimensions, the flattened data is passed through a dense layer with softmax activation and kernel regularizer. The model training process aims to minimize the loss by penalizing the difference between the predicted and actual probabilities. Equation (11) offers the mathematical representation of the cross-entropy loss [34]:

$$\textbf{Lcls} = -\sum_{c=1}^{10} t_c . \log(p_c) \quad (11)$$

Where:

'c' represent the number of classes (10 in this case), 't' represent the target indicator (0 or 1), 'log' is the natural logarithm function, and 'p' is the predicted probability. Algorithm (1) describe (1D-densenet) classifier **.** And Table (2) describe details of 1D-DenseNet.

Algorithm1: (1D- DenseNet model)

Begin
Input: Vector of distract driver feature.
Output: Classification for distract driver image**.**
Step 1: Split data into Training and Validation sets.
Step 2: Initialize Models and Hyperparameters.
Step 3: Loop over Hyperparameters (p) and Models (m):
Loop over Epochs (e):
For each batch in Training Data:
1. Forward Propagation.
2. Compute Loss.
3. Backward Propagation.
4. Update Weights.

Step4: Check Model Performance on Validation Set and Update:
If (Validation Loss improved):
update Best Model, Best Param.
Step5: Evaluate Model Performance on Testing Set using Classification Metrics:
If (Testing Loss is satisfactory):
Use the Model for Inference.
Else:
Adjust the Hyperparameters and Repeat the Process.
Step6: Final Model = Train Model with Best Model and BestParams using all Training Data.

End.

TABLE 2. Modified DenseNet layers details.

Layers	Filter (Size/number)	Input shape	output shape	Parameter number
Input layer	/	(None, 400, 1)	(None, 400, 1)	0
Convolution	(3*1Conv) \24	(None, 400, 1)	(None, 400, 24)	72
1D-1^{st}DenseBlock	$\binom{1}{3*1}Conv$/12	(None, 400, 24)	(None,400,16)	7776
1D -1^{st}transition layer	$\binom{1}{2*1}Conv$/1	(None,400,168)	(None, 200, 84)	0
1D -2^{nd}DenseBlock	$\binom{1}{3*1}Conv$/12	(None, 200, 84)	(None, 200,228)	8856
1D - 2^{nd} transition layer	$\binom{1}{2*1}Conv$/1	(None, 200, 228)	(None, 100,114)	0
1D -3^{rd}DenseBlock	$\binom{1}{3*1}Conv$/12	(None, 100, 114)	(None, 100,258)	9396
1D - 3^{rd} transition layer	$\binom{1*1Conv}{2*1AvgPool}$/1	(None, 100, 258)	(None, 50, 129)	0
1D -4^{th}DenseBlock	$\binom{1}{3*1}Conv$/12	(None, 50, 129)	(None, 50, 273)	9396
1D -Global Average pool	(3*1AvgPool)/273	(None, 50, 129)	(None, 25, 273)	0
Flatten layer	/	(None, 25, 273)	(None, 6825)	0
1D- FCL with SoftMax	/	(None, 6825)	(None, 10)	68260

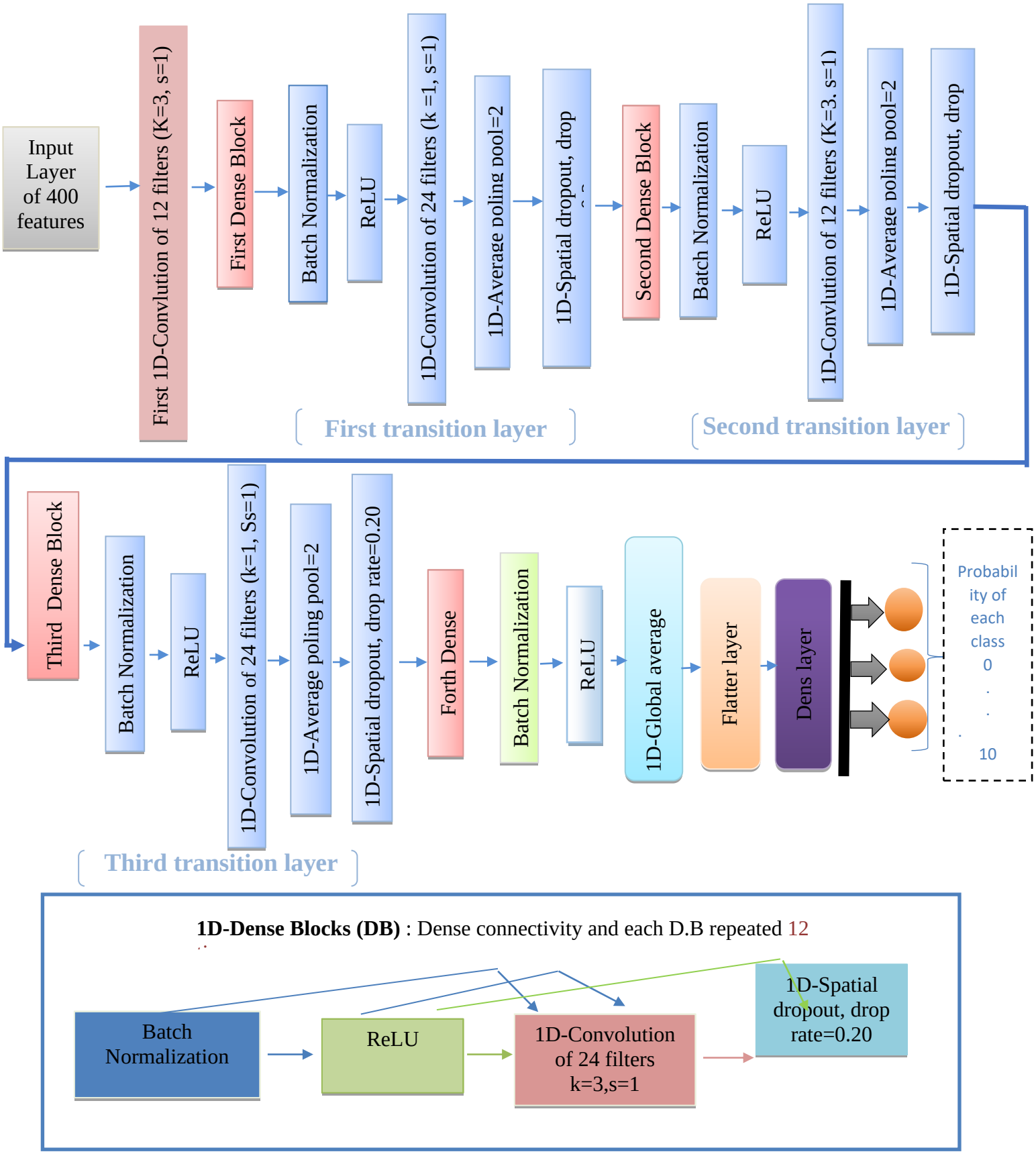

FIGURE 4. block diagram of proposed 1D-DenseNet (k=12).

5. DATASET DESCRIPTION

State Farm launched a Kaggle competition in April 2016 to visualize the dangers of distracted driving, providing the State Farm Distracted Driver Data-set (SDDD) [36] for model training. The dataset consists of 22,424 images, with 12,223 for training and 5,239 for testing. The images depict drivers engaging in various activities, including safe driving, using a phone, texting, drinking, operating the radio, and doing hair and makeup. More details about the dataset are presented in Table (3).

TABLE 3. Original SDDD summary information.

Number of classes	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
Driver behavior	Safe driving	Texting-right hand	Right hand-taking on the phone	Texting-left hand	Left hand-Taking on the phone	Operating the radio	Drinking	Reaching behind	Makeup and hair	Talking to passenger
No. of images	2489	2267	2317	2346	2326	2312	2325	2002	1911	2129

FIGURE 5. samples of the state farm distract driver dataset.

6. EVALUATION

Multiple metrics such as accuracy, F-score, recall, and precision were used to evaluate the distracted driver classifier's performance. These metrics helped minimize false positives and negatives and provided a comprehensive understanding of the model's performance, aiding in decisions to improve road safety [37]-[39]:

A. Accuracy is the ratio of correct predictions to the total number of predictions, as given in Equation (12).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

B. F1 score: Is a measure of a test's accuracy and is calculated using the following Equation (13):

$$F1 = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall}) \quad (13)$$

C. Precision. It is a measure of the model's ability to correctly identify positive results and minimize false positives. Class precision is quantified as follows in Equation (14).

$$Precision = \frac{TP}{TP+FP} \quad (14)$$

D.Recall is the recall is the degree to which a classifier can recognize documents belonging to a particular class. It is possible to count the recall of class by using Equation (15):

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

7.RESULT and DSCUSSION

The 1D-DenseNet was adapted and assessed using a GPU and an Intel Core I7-4770 CPU @ 3.40 GHz with 8GB of RAM from a personal computer. The model was optimized using Adam and a learning rate of 0.001.Table (4) presents the hyperparameters of 1D-DenseNet and DenseNet.

Table 4. 1D-DenseNet and original DenseNet hyperparameters used.

Hyperparameter	Value
Optimizer	Adam
Learning rate	0.001
Epochs	100
Batch size	64

The proposed model and original DenseNet used 20 x 20-pixel images as inputs and were trained for 10 epochs. 1D-DenseNet achieved a validation accuracy of 98.78% and a validation loss of 0.1880, this was better than the 2D-DenseNet model, which had a lower validation accuracy of 90% and a higher loss of 1.18. Moreover, the proposed model was faster than the 2D-DenseNet model. Figs. (6) and (7) illustrate the training process of the 1D-DenseNet and 2D-DenseNet models for 10 epochs, respectively.

```
2023-05-15 07:17:51.778329: I T:\src\github\tensorflow\tensorflow\core\platform\cpu_feature_guard.cc:140] Your CPU supports instructions
that this TensorFlow binary was not compiled to use: AVX2
12223/12223 [==============================] -  750s 61ms/step - loss: 5.7358 - acc: 0.5034 - val_loss: 4.9862 - val_acc: 0.5851
Epoch 2/10
12223/12223 [==============================] - 750s 61ms/step - loss: 4.4871 - acc: 0.6074 - val_loss: 4.1299 - val_acc: 0.5911
Epoch 3/10
12223/12223 [==============================] - 750s 61ms/step - loss: 3.3599 - acc: 0.7703 - val_loss: 3.1477 - val_acc: 0.7585
Epoch 4/10
12223/12223 [==============================] - 750s 61ms/step - loss: 2.4563 - acc: 0.8234 - val_loss: 2.4937 - val_acc: 0.6480
Epoch 5/10
12223/12223 [==============================] - 750s 61ms/step - loss: 1.8299 - acc: 0.8200 - val_loss: 1.7819 - val_acc: 0.8007
Epoch 6/10
12223/12223 [==============================] - 826s 68msstep - loss: 1.3005 - acc: 0.9579 - val_loss: 1.3386 - val_acc: 0.8784
Epoch 7/10
12223/12223 [==============================] - 750s 61ms/step - loss: 0.3141 - acc: 0.9634 - val_loss: 0.5320 - val_acc: 0.9754
Epoch 8/10
12223/12223 [==============================] - 816s 67ms/step - loss: 0.1919 - acc: 0.9732 - val_loss: 0.1819 - val_acc: 0.9872
Epoch 9/10
12223/12223 [==============================] - 816s 67ms/step - loss: 0.1819 - acc: 0.9850 - val_loss: 0.1880 - val_acc: 0.9878
Epoch 10/10
12223/12223 [==============================]  - 816s 67ms/step - loss: 0.1761 - acc: 0.9857 - val_loss: 0.1880 - val_acc: 0.9878
```

FIGURE 6. 1D-DenseNet training process for 10 epochs.

```
Train on 28520 samples, validate on 5239 samples
Epoch 1/10
2023-05-16 08:12:51.778329: I T:\src\github\tensorflow\tensorflow\core\platform\cpu_feature_guard.cc:140] Your CPU supports instructions
that this TensorFlow binary was not compiled to use: AVX2
12223/12223 [==============================] -   3814s 20s/step - loss: 8.6135 - acc: 0.1370 - val_loss: 6.5365 - val_acc: 0.1473
Epoch 2/100
191/191 [==============================] - 4292s 282s/step - loss: 4.6102 - acc: 0.3798 - val_loss: 3.9522 - val_acc: 0.2627
Epoch 3/100
191/191 [==============================] - 65974s 345s/step - loss: 2.7046 - acc: 0.6337 - val_loss: 2.8144 - val_acc: 0.4698
Epoch 4/100
191/191 [==============================] - 5360s 18s/step - loss: 1.8527 - acc: 0.7650 - val_loss: 1.6683 - val_acc: 0.7575
Epoch 5/100
191/191 [==============================] - 65974s 345ms/step - loss: 1.4037 - acc: 0.8405 - val_loss: 1.2795 - val_acc: 0.8655
Epoch 6/100
191/191 [==============================] - 8426s 345s/step - loss: 1.3005 - acc: 0.8579 - val_loss: 1.3386 - val_acc: 0.8484
Epoch 7/10
191/191 [==============================] - 7750s 616s/step - loss: 1.3141 - acc: 0.8834 - val_loss: 1.5320 - val_acc: 0.8754
Epoch 8/10
191/191 [==============================] - 8176s 677s/step - loss: 1.1919 - acc: 0.8732 - val_loss: 1.1719 - val_acc: 0.8872
Epoch 9/10
191/191 [==============================] - 8196s 678s/step - loss: 1.1719 - acc: 0.9050 - val_loss: 0.1880 - val_acc: 0.9178
Epoch 10/10
191/191 [==============================]  - 8196s 687s/step - loss: 1.1761 - acc: 0.9157 - val_loss: 1.1880 - val_acc: 0.9078
```

FIGURE 7. 2D-DenseNet training process for 10 epochs.

After an additional 100 epochs of training, an accuracy of 99.80% was achieved. Fig (8) displays a scatter plot of the 2D-DenseNet model's accuracy with the number of epochs, demonstrating that the model's performance improved as more epochs were used. Table (5) contrasts the 100-epoch training results of the 1D-DenseNet and 2D-DenseNet models.

TABLE 5 .1D-densnet and 2D DenseNet compression.

Model name	Epoch	Computation	No. of parameter	Accuracy
1D-Densnet	100	860 s (per step), 1 day for 100 epochs	440,626	99.80%
DenseNet	100	14651s (per step), 4 days for 100 epochs	907,042	98.26%

As shown in Table (5), 1D-DenseNet has a higher validation accuracy than 2D-DenseNet by 1.54%, which indicates that 1D-DenseNet is a better model for the human activity recognition task. Moreover, 1D-DenseNet is much faster than 2D-DenseNet, as it takes 13791 seconds or about 3.8 hours less to complete one step.

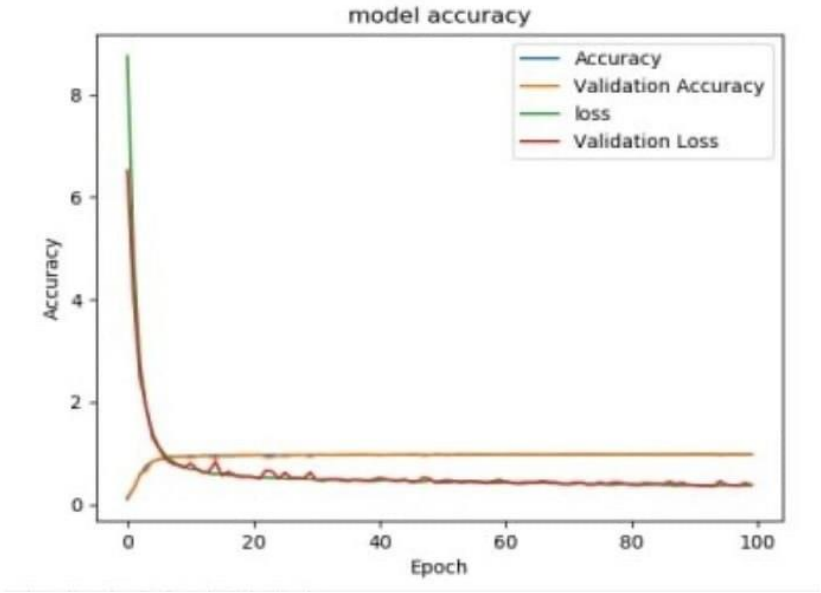

FIGURE 8. 2D-DenseNet model accuracy after 100 epochs.

The application of a preprocessing stage had a significant positive impact on the accuracy of the proposed model. By utilizing techniques such as converting images to grayscale, applying Gaussian blur, histogram equalization, and image resizing, the accuracy of the model improved from 92.22% to an impressive 99.90%. These preprocessing steps enhanced the quality and suitability of the images, allowing the model to better capture important features related to distracted driving. Fig (9) shows a sample of distracting driver images after applying pre-processing steps.

Figure 9. Samples of distracted driver images after applying pre-processing steps. (a): The original image. (b) :The image after converting to grayscale. C: The image after applying a Gaussian filter. (d): The image after applying histogram equalization.

When compared to previous deep learning models, the suggested 1D-DenseNet model performed better at detecting anomalous driving behavior. A comparison of the precision levels displayed by several models when used with the SDDD dataset can be seen in Table (6). The 1D-DenseNet architecture outperformed rival models using unique convolutional neural networks, including VGG, ResNet, and Inception, among others, with a significant accuracy of 99.80%. Comparing the 1D-DenseNet design to other models reveals several advantages. The ability to

solve the vanishing-gradient problem, improve feature propagation, allow feature reuse, and reduce the number of parameters are some of these benefits.

TABLE 6. stat-of-art accuracy compression.

Study	Model Used	Dataset	Accuracy
[15]	Ensemble of CNNs	SDDD	90%
[16]	CNN	SDDD	98.76%
[18]	ResNet-50 and VGG-16	SDDD	87.92%
[19]	CNN, Visualized Geometric Group16, Deep Residual Network	SDDD	82%
[20]	Small CNN, VGG16, VGG19, Inception	SDDD	91.2%
Original 2D DenseNet	2D- DenseNet	SDDD	98.26
Proposed model	1D-DenseNet	SDDD	99.80

The results of different models' training and validation losses are compared in Table (7), which is based on the SDDD dataset. The loss is a statistic that determines how closely the model corresponds to the data.

TABLE 7. Compression of training and validation loss.

Model name	RestNet50[1]	DL [20]	2D-DenseNet	Proposed DenseNet
Training loss	0.7	0.3084	0.3850	0.1924
Validation loss	0.70	2.379	0.3791	0.1823

Among the four models, the proposed DenseNet model shows the best performance and generalization ability, with a training loss of 0.19 and a validation loss of 0.18. Precision, recall and F-score metrics evaluate how well a classifier works. They are helpful for imbalanced problems, where accuracy is not trustworthy. Table (8) presents a contrast of 1D-DenseNet with three other DL models.

TABLE 8. Precision, f-score, and recall compression.

Model name	CNN [7]	E2DR [8]	DL [11]	1D-DenseNet
Precision	97%	92%	92.69%	**99.90%**
F-score	/	92%	/	**99%**
Recall	98%	91%	92.66%	**99%**

The 1D DenseNet model outperforms the other DL methods in terms of precision, recall, and F-score. With precision at 99.88%, recall at 99.06%, and an F-score of 99.08%, the 1D DenseNet model proves to be highly accurate for detecting distracted drivers. By using the SDDD data set, the 1D DenseNet model can accurately classify distracted drivers into one of the ten categories, such as looking at their phone, eating, drinking, or talking to a passenger. The precision, recall, and F-score are measures that assess how well the model can separate distracted and non-distracted drivers, and not overlook any distracted drivers. The model is better when these measures are higher. The 1D DenseNet model has very high values for these measures, which means it is very reliable and effective for detecting distracted drivers. Fig (10) shows how well the model can detect and classify distracted driver behavior for different activities.

Figure 10. 1D-DenseNet precision, Recall and F-score after 100 epochs.

7. CONCLUSION

The M1D-DenseNet model was designed with adapted dense blocks and transition layers to accurately identify specific patterns in driving data and detect abnormal driving behaviors. To improve the quality of input data, the model uses several preprocessing techniques, including Gaussian filtering, histogram equalization, and image resizing. The model also utilizes PCA to reduce parameters, improve efficiency, and extract essential features. These modifications have led to the M1D-DenseNet model achieving high accuracy in detecting abnormal driving behavior, while requiring fewer parameters, leading to faster training and improved performance, and reduced CPU overhead. The model has potential for real-time implementation in the future.

Funding

None

ACKNOWLEDGEMENT

None

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] I. Khalil, C. El Bcheraoui, R. Charara, M. Moradi-Lakeh, A. Afshin, N.J. Kassebaum, M. Collison, F. Daoud, A. Chew, K.J. Krohn, “Transport injuries and deaths in the Eastern Mediterranean Region: findings from the GlobalBurden of Disease 2015 Study,” International Journal of Public Health, vol. 63, no. S1, pp. 187–198, Aug. 2017, Doi: 10.1007/s00038-017-0987-0.

[2] A. H. Albayati and I. M. Lateef, “Characteristics of Traffic Accidents in Baghdad,” Civil Engineering Journal, vol. 5, no. 4, pp. 940–949, Apr. 2019, Doi: 10.28991/cej-2019-03091301.

[3] B. W. Abdullah; H. M. Ahmed. "A Convolutional Neural Network for Detecting COVID-19 from Chest X-ray Images". Iraqi Journal of Computers, Communications, Control and Systems Engineering, 22, 3, 2022, 1-14. Doi: https://doi.org/10.33103/uot.ijccce.22.3.1

[4] M. Ghazi; M. Abdulmunim. "Convolutional Recurrent Neural Networks for Text Lecture Summarization". Iraqi Journal of Computers, Communications, Control and Systems Engineering, 22, 2, 2022, 27-39. Doi: https://doi.org/10.33103/uot.ijccce.22.2.3.

[5] T. Obaida; N. F. Hassan; A. S. Jamil. "Comparative of Viola-Jones and YOLO v3 for Face Detection in Real time". Iraqi Journal of Computers, Communications, Control and Systems Engineering, 22, 2, 2022, 63-72. Doi: https://doi.org/10.33103/uot.ijccce.22.2.6 .

[6] H. M. Fadhil, M. N. Abdullah, and M. I. Younis, "A Framework for Predicting Airfare Prices Using Machine Learning," Iraqi Journal of Computers, Communications, Control and Systems Engineering, vol. 22, no. 3, 2022, Doi: 10.33103/uot.ijccce.22.3.8

[7] S. Thakur, B. Baheti, S. Gajre, and S. Talbar, "Pose Estimation for Distracted Driver Detection Using Deep Convolutional Neural Networks," Computer Vision Applications, pp. 102–114, 2019, doi: 10.1007/978-981-15-1387-9_9.

[8] M. Aljasim and R. Kashef, "E2DR: A Deep Learning Ensemble-Based Driver Distraction Detection with Recommendations Model," Sensors, vol. 22, no. 5, p. 1858, Feb. 2022, doi: 10.3390/s22051858.

[9] G. D. Furman, A. Baharav, C. Cahan, and S. Akselrod, "Early detection of falling asleep at the wheel: A Heart Rate Variability approach," 2008 Computers in Cardiology, Sep. 2008, doi: 10.1109/cic.2008.4749240.

[10] K. Lee, H. Yoon, J. Song, and K. Park, "Convolutional Neural Network-Based Classification of Driver's Emotion during Aggressive and Smooth Driving Using Multi-Modal Camera Sensors," Sensors, vol. 18, no. 4, p. 957, Mar. 2018, doi: 10.3390/s18040957.

[11] J. Hu, X. Zhang, and S. Maybank, "Abnormal Driving Detection with Normalized Driving Behavior Data: A Deep Learning Approach," IEEE Transactions on Vehicular Technology, vol. 69, no. 7, pp. 6943–6951, Jul. 2020, Doi: 10.1109/tvt.2020.2993247.

[12] A. Noor, B. Benjdira, A. Ammar, and A. Koubaa, "DriftNet: Aggressive Driving Behaviour Detection using 3D Convolutional Neural Networks," 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH), Nov. 2020, Doi: 10.1109/smart-tech49988.2020.00056.

[13] B. Baheti, S. Gajre, and S. Talbar, "Detection of Distracted Driver Using Convolutional Neural Network," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Jun. 2018, Doi: 10.1109/cvprw.2018.00150.

[14] H. Li, J. Han, S. Li, H. Wang, H. Xiang, and X. Wang, "Abnormal Driving Behavior Recognition Method Based on Smart Phone Sensor and CNN-LSTM," International Journal of Science and Engineering Applications, vol. 11, no. 1, pp. 1–8, Jan. 2022, doi: 10.7753/ijsea1101.1001.

[15] H. M. Eraqi, Y. Abouelnaga, M. H. Saad, and M. N. Moustafa, "Driver Distraction Identification with an Ensemble of Convolutional Neural Networks," Journal of Advanced Transportation, vol. 2019, pp. 1–12, Feb. 2019, Doi: 10.1155/2019/4125865.

[16] M. Alotaibi and B. Alotaibi, "Distracted driver classification using deep learning," Signal, Image and Video Processing, vol. 14, no. 3, pp. 617–624, Nov. 2019, doi: 10.1007/s11760-019-01589-z.

[17] I. Nasri, M. Karrouchi, H. Snoussi, K. Kassmi, and A. Messaoudi, "DistractNet: a deep convolutional neural network architecture for distracted driver classification," IAES International Journal of Artificial Intelligence (IJ-AI), vol. 11, no. 2, p. 494, Jun. 2022, doi: 10.11591/ijai.v11.i2.pp494-503.

[18] N. K. Vaegae, K. K. Pulluri, K. Bagadi, and O. O. Oyerinde, "Design of an Efficient Distracted Driver Detection System: Deep Learning Approaches," IEEE Access, vol. 10, pp. 116087–116097, 2022, doi: 10.1109/access.2022.3218711.

[19] P. Roy and O. Ancy, "Abnormal driving behavior detection via deep learning approaches," International Journal of Advance Research, 2020. www.IJARIIT.com.

[20] P. M. Chawan, S. Satardekar, D. Shah, R. Badugu, , & A. Pawar, " Distracted Driver Detection and Classification Cloud based Automated Attendance Monitoring System using RFID and IOT View project Decision Tree View project Distracted Driver Detection and Classification". Journal of Engineering Research and Application Www.Ijera.Com, 8, 60–64. (2018).https://doi.org/10.9790/9622-0804036064 .

[21] O. G. Basubeit, D. N. T. How, Y. C. Hou, and K. S. M. Sahari, "Distracted driver detection with deep convolutional neural network." Int. J. Recent Technol. Eng. 8.4 (2019): 6159-6163. . https://doi.org/10.35940/ijrte.D5131.118419

[22] T. Zhou, X. Ye, H. Lu, X. Zheng, S. Qiu, and Y. Liu, "Dense Convolutional Network and Its Application in Medical Image Analysis," BioMed Research International, vol. 2022, pp. 1–22, Apr. 2022, doi: 10.1155/2022/2384830.

[23] Md. U. Hossain, Md. A. Rahman, Md. M. Islam, A. Akhter, Md. A. Uddin, and B. K. Paul, "Automatic driver distraction detection using deep convolutional neural networks," Intelligent Systems with Applications, vol. 14, p. 200075, May 2022, doi: 10.1016/j.iswa.2022.200075.

[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017, doi: 10.1109/cvpr.2017.243.

[25] Y. Zhu and S. Newsam, “DenseNet for dense flow,” 2017 IEEE International Conference on Image Processing (ICIP), Sep. 2017, doi: 10.1109/icip.2017.8296389.
[26] L. Wenqi, and K. Zeng. "SparseNet: A sparse DenseNet for image classification." arXiv preprint arXiv:1804.05340 (2018). https://doi.org/10.48550/arXiv.1804.05340.
[27] Umesh. D. Dixit and M. S. Shirdhonkar, “Preprocessing Framework for Document Image Analysis,” International Journal of Advanced Networking and Applications, vol. 10, no. 4, pp. 3911–3918, 2019, doi: 10.35444/ijana.2019.10042.
[28] E. S. Gedraite and M. Hadad, "Investigation on the effect of a gaussian blur in image filtering and segmentation," in ELMAR Proceedings, 2011.
[29] S. Mishra, U. Sarkar, S. Taraphder, S. Datta, D. Swain, & R. Saikhom, "Multivariate Statistical Data Analysis- Principal Component Analysis (PCA)". International Journal of Livestock Research, 7(5), 60-78. (2017). http://dx.doi.org/10.5455/ijlr.20170415115235
[30] M. S. Reza and J. Ma, “ICA and PCA integrated feature extraction for classification,” 2016 IEEE 13th International Conference on Signal Processing (ICSP), Nov. 2016, doi: 10.1109/icsp.2016.7877996.
[31] J. A. López del Val, J. P. Alonso Pérez de Agreda, Principal components analysis. Atencion primaria / Sociedad Española de Medicina de Familia y Comunitaria. 12, 333–338 (1993). https://doi.org/10.5455/ijlr.20170415115235.
[32] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, “Efficient object localization using Convolutional Networks,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2015, doi: 10.1109/cvpr.2015.7298664.
[33] F. Schilling, ‘The Effect of Batch Normalization on Deep Convolutional Neural Networks,’ Dissertation, 2016.
[34] D.-L. Nguyen, M. Dwisnanto Putro, X.-T. Vo and K.-H. Jo, "Light-weight convolutional neural network for distracted driver classification", Proc. 47th Annu. Conf. IEEE Ind. Electron. Soc. (IECON), pp. 1-6, Oct. 2021 . DOI: 10.1109/IECON48115.2021.9589212.
[35] H. M and S. M.N, “A Review on Evaluation Metrics for Data Classification Evaluations,” International Journal of Data Mining & Knowledge Management Process, vol. 5, no. 2, pp. 01–11, Mar. 2015, doi: 10.5121/ijdkp.2015.5201.
[36] State Farm Distracted Driver Detection. Available online: https://www.kaggle.com/c/state-farm-distracted-driver-detection (accessed on 25 Feb 2023).
[37] M. A. Ahmed, “A Classification of Al-hur Arabic Poetry and Classical Arabic Poetry by Using Support Vector Machine, Naïve Bayes, and Linear Support Vector Classification,” Iraqi Journal for Computer Science and Mathematics, pp. 128–137, Jul. 2022, doi: 10.52866/ijcsm.2022.02.01.014.
[38] A. H. Ali, Z. F. Hussain, and S. N. Abd, “Big Data Classification Efficiency Based on Linear Discriminant Analysis,” Iraqi Journal for Computer Science and Mathematics, pp. 7–12, Jan. 2020, doi: 10.52866/ijcsm.2019.01.01.001.
[39] N. H. Ali, M. E. Abdulmunem, and A. E. Ali, “Learning Evolution: A Survey,” Iraqi Journal of Science, pp. 4978–4987, Dec. 2021, doi: 10.24996/ijs.2021.62.12.34.