Traffic Congestion Classification Using GAN-Based Synthetic Data Augmentation and a Novel 5-Layer Convolutional Neural Network Model

Umair Jilani 1,2,*, Muhammad Asif 1, Munaf Rashid 1, Ali Akbar Siddique 2, Syed Muhammad Umar Talha 2,3 and Muhammad Aamir 2

Abstract: Private automobiles are still a widely prevalent mode of transportation. Subsequently, traffic congestion on the roads has been more frequent and severe with the continuous rise in the numbers of cars on the road. The estimation of traffic flow, or conversely, traffic congestion identification, is of critical importance in a wide variety of applications, including intelligent transportation systems (ITS). Recently, artificial intelligence (AI) has been in the limelight for sophisticated ITS solutions. However, AI-based schemes are typically heavily dependent on the quantity and quality of data. Typical traffic data have been found to be insufficient and less efficient in AI-based ITS solutions. Advanced data cleaning and preprocessing methods offer a solution for this problem. Such techniques enable quality improvement and augmenting additional information in the traffic congestion dataset. One such efficient technique is the generative adversarial network (GAN), which has attracted much interest from the research community. This research work reports on the generation of a traffic congestion dataset with enhancement through GAN-based augmentation. The GAN-enhanced traffic congestion dataset is then used for training artificial intelligence (AI)-based models. In this research work, a five-layered convolutional neural network (CNN) deep learning model is proposed for traffic congestion classification. The performance of the proposed model is compared with that of a number of other well-known pretrained models, including ResNet-50 and DenseNet-121. Promising results present the efficacy of the proposed scheme using GAN-based data augmentation in a five-layered convolutional neural network (CNN) model for traffic congestion classification. The proposed technique can be used for urban traffic planning and maintenance managers and stakeholders for the efficient deployment of intelligent transportation system (ITS).

Keywords: intelligent transportation system (ITS); image synthesis; generative adversarial networks (GAN)

1. Introduction

The growing demand for travel and transportation has contributed to an increase in the severity of traffic problems, such as an increase in the amount of congestion and a decrease in the level of safety. This is a direct consequence of the increased demand for travel and transportation that has been seen recently. Not only does this have a significant negative impact on the growth of the economy, but it also has significant negative impacts...
Accidents, poor weather, slow-moving heavy vehicles, sudden vehicle breakdown, not obeying traffic laws, sudden jams at road intersections, and physical road conditions are only a few of the elements that may contribute to this congestion problem. The most essential and fundamental component of resolving these traffic issues is the precise evaluation of the current situation of road traffic. The readings obtained from a large number of traffic detectors are used to depict the current situation of road traffic. In the current body of research, the term “traffic state” most often refers to the flow of traffic, the density of traffic, the speed of traffic, and the journey time [2]. It is possible for administrators of traffic systems to successfully regulate transportation networks and prevent traffic jams by using accurate estimates of the current status of the traffic. Thereby, passengers are able to more effectively arrange their departure dates and travel itineraries, which allows for greater efficiency during their journeys [3].

Traffic congestion on a country’s roadways may have both direct and indirect effects on the economy of that nation as well as on the health of its inhabitants. According to a study conducted by Ali et al. [4], the opportunity cost and increased fuel consumption that result from traffic congestion cost PKR 1 million every single day. The effects of traffic congestion may be felt on an individual level as well. Loss of time, particularly during rush hour, mental stress, and the aggravation of global warming brought on by increased pollution are other key concerns brought on by traffic congestion.

The prediction of traffic congestion gives authorities the necessary time to organize the distribution of resources to ensure smooth trips for those who are traveling [5]. The subfield of computer science known as artificial intelligence (often abbreviated as AI) is now the most essential component of the field, particularly today, when big data is becoming an increasingly crucial factor. Over the course of the last half-century, there has been a substantial amount of development in the field of artificial intelligence (AI), particularly in the subfields of machine learning, data mining, computer vision, expert systems, natural language processing, robotics, and other related applications [6]. The study of artificial intelligence (AI) has been around for around half a century. Both the study domain of operations management and the study domain of information systems have devoted a significant amount of attention to the role that information technology (IT) plays in the management of operations that allow for environmentally sustainable development [7].

In the subject of transportation engineering, unmanned aerial vehicles (UAVs) are becoming an increasingly popular instrument for the purpose of monitoring and evaluating traffic, and many researchers have employed them for this task. This in-depth study examines the scientific contributions that have been made to the use of unmanned aerial vehicles (UAVs) in civil engineering, specifically those contributions that relate to the monitoring of traffic [8]. Numerous nations across the world are investing in the remote monitoring and management of traffic; the data gathered from road traffic will reveal various factors that contribute to traffic congestion [9]. The Texas Transportation Institute (also known as TTI) is widely regarded as among the most reputable institutions in the United States. In order to demonstrate and keep track of the traffic flow and congestion in more than 400 of cities, it is tasked with preparing an Urban Mobility Report (UMR) on a yearly basis. According to the TTI, traffic congestion is still a widespread problem that is continuing to worsen in American cities. When compared with the situation that existed in 1982, the TTI estimated that the annual cost of traffic delay for each passenger was around $1080 annually in 2018. This is a twofold increase when accounting for inflation in comparison with the scenario that existed in 1982. At the same time that the quantity of gasoline that was wasted due to traffic jams more than tripled, reaching 3.3 billion gallons a year, the amount of time that commuters lost due to traffic congestion almost quadrupled, reaching 54 h a year [10,11].

Congestion causes lengthy travel times, high costs for commuters and shipping firms, high energy and carbon emissions, air and noise pollution, commuter stress, physical inactivity, and a dampening influence on economic development; in most cases, it impacts labor expenses, working hours, and product delivery as well [12]. As a result of fast urbanization
and a rise in population, people will need a variety of means of transportation in order to fulfill their growing demand for travel. These variables are the primary contributors to the issue of congestion, which is a basic concern [13]. Authorities have been burdened with the responsibility of taking steps to mitigate the negative consequences that are generated by congestion, and managing these circumstances has become a problem [14]. Impedovo, Donato et al. attempted to categorize vehicular congestion using visual features and used traditional machine learning approaches to categorize traffic congestion, and they then compared those methods with the most recent developments in deep learning methods [15]. Congestion on roads can be more effectively managed with the assistance of an intelligent transportation system (ITS) that is based on the internet of things (IoT) [16,17]. Intelligent traffic management includes a number of significant features, one of the most important of which is the estimation and categorization of the traffic congestion status on various route segments. The identification of the traffic congestion on various road segments provides assistance to the traffic management authorities in their efforts to achieve optimal traffic control within a transportation system. Commuters may also choose the most efficient path to their destination by taking into account the level of traffic congestion on each individual road segment [18]. IoT devices can produce information that cannot be managed using available storage or any other data analytics tool, and Dhingra, Swati et al. utilized cloud computing technology that could address this issue to some extent [19]. Arguably, when transferring a large amount of data to the cloud for the purpose of storing it and processing it, there is a possibility that this could result in the cloud being overloaded, which will, in turn, cause the available bandwidth on the network to be depleted [20]. Researchers have also introduced convolution neural networks (CNNs) to classify or predict traffic congestion [21]. TrafficNet is a deep learning technique that was introduced by Wang, Ping et al. to identify and locate instances of traffic congestion using a large-scale surveillance system [22].

Accurate short-term traffic flow estimates are essential to the creation of an intelligent transportation system as well as the efficient operation of the traffic system [23]. It is essential to have an accurate and efficient short-term traffic flow in addition to congestion predictions in order to successfully implement real-time traffic light adjustment [24]. For one, the present signal light localization is set almost entirely, as opposed to being based on a fixed timeline, by the skill of the individual installation. This is the case even though there is no standard. This strategy does not match the timing of the traffic signals to the real flow of traffic, which results in the waste of employees and resources and an outcome that is not desirable. The low cost of operation and maintenance of the traffic lights is another benefit of this strategy. Utilizing a signal timing strategy that is both well-designed and well-executed in order to segment the different time- and space-dependent traffic flows may help decrease or eliminate congestion at intersections. As a consequence of this, optimizing signal timing in real time and with a high degree of precision is essential to the development of intelligent transportation systems [25]. It is feasible to ease traffic congestion more effectively, which will be to the advantage of all parties involved. Because of the high number of variables in a video that need to be examined in real time while the video is still being captured in real time, an algorithmic solution may run into dimensionality difficulties [26]. Many researchers have used long short-term memory (LSTM) for the classification of traffic congestion [27]. In most cases, deep learning algorithms are also used for the detection of traffic accidents. However, there is a possibility that the information included in the expert knowledge base for traffic accidents is not entirely correct. More research work remains to determine how to correctly extract traffic accidents from traffic flow data. An incremental learning-based CNN model was suggested by Shao, Yanli et al. based on LSTM. They included partial uncertainty characteristics in the model in order to increase the prediction accuracy [28].

The majority of the classification techniques for traffic congestion rely on human observation rather than automated procedures. When it comes to the detection of congestion, there are a number of limitations that may be worked around using an automated deep
neural network approach in order to avoid certain limits. This article presents a novel method for the classification of traffic congestion that is based on CNN as an alternative method that improves the speed of traffic control and at the same time saves a significant amount of time for commuters. The CNN method is presented as an alternative method that improves the accuracy of traffic control. This concept is being put forth as a potential replacement for the standard approach of classifying traffic congestion. In order to evaluate and contrast the outcomes that the trained CNN network produces, we use not one but two unique pre-trained classifier models, named ResNet-50 [29] and DenseNet-121 [30].

2. Database Applied for Training and Testing

When training the model, the data were taken from two different parts of the dataset: One was used to represent places with heavy traffic, and the other represented areas with little or no traffic. Within the presented collection, 750 of the images were assigned the category “non-congestion,” and the remaining 1250 were labeled as depicting traffic congestion. Sample images of congestion are shown in Figure 1a and samples with no or little traffic are shown in Figure 1b. Because it might be challenging to classify different types of traffic congestion using the standard method, the dataset must be adequate for the work in order to complete it in an effective manner. In order to train a model with the maximum possible degree of accuracy, the vast majority of classical machine learning approaches need a substantial dataset. However, if the dataset is flawed and makes absolutely no sense, the chosen machine learning approach will not be able to effectively train the model, and the results will not be as accurate as they might have been if the proper dataset had been used.

![Figure 1. (a) Samples images of congestion. (b) Samples images of no congestion or little traffic.](image)

3. Generative Adversarial Networks (GAN) for Image Synthesis

With deep learning algorithms, the selection and application of the algorithm and the availability of a database are the most critical components. Due to the limited size of the traffic database, this study places a significant emphasis on the use of data aug-
mentation. To avoid traffic congestion, an innovative approach of sequential generative adversarial networks (LSTM-GAN) was proposed that would create realistic traffic scenarios by capturing the temporal dynamics of roadways [31]. A novel model for traffic flow prediction was proposed that uses generative adversarial networks to evaluate real-time traffic datasets with appreciable improvement over the baseline method [32]. The available traffic states are often incomplete in the real world, and therefore, a novel deep learning framework is proposed for traffic state estimation for the generation of traffic road states in real time. Investigators have conducted much work to avoid traffic congestion such as novel complex-condition-controlled urban traffic estimation through generative adversarial networks (C³-GAN), proposed by Zhang et al. [33], which produces high-quality traffic estimations and outperforms state-of-the-art baseline methods. GAN-based traffic congestion estimation is also useful in many other applications, including automatic signal control, traffic engineering-based road planning and management and toll counter placement. One such innovative approach for the application of deep reinforcement learning was proposed by Gregurić, Martin et al. [34] to overcome the problem of persistent congestion in adaptive traffic signal control (ATSC).

In the proposed traffic congestion algorithm, we have utilized the GAN model introduced by Chatterjee, Subhajit et al. [35]. Images are first enhanced with a higher resolution by utilizing GAN and then augmented. The GAN model is used to produce synthetic images that resemble the original, and then data augmentation and image modification techniques are used to enhance those images. Basically, GAN utilizes a min–max game in conjunction with a value function known as \( V(D, G) \) where \( D \) is the discriminator and \( G \) is the generator. At stake are two goals that are in direct opposition to each other: maximizing the loss for the generator while minimizing the gain for the discriminator. Both the generator \( (G) \) and the discriminator \( (D) \) are working toward achieving one of two objectives: either lowering or raising the next loss function. The GAN generator is run with a random noise \( P(\text{rnv}) \) attached to the initial input data \( x \) in order to produce samples \( G(\text{rnv}) \); here, \( (\text{rnv}) \) represents the random noise variable. These samples are produced by the GAN generator. In fact, it is possible to think of the noise as the latent representation of the data that the generator is making an implicit attempt to learn. Every component of the “noise” vector can be considered a feature provided to the generator; the greater the value of any feature, the better the resulting image will be. Based on a discriminator’s evaluation of the probability, data distribution \( P_d \) could contain or not contain an instance \( x \). In order to assess whether a given instance is false or genuine, the discriminator applies a mathematical formula with the name \( D(G(\text{rnv})) \). In order to trick the discriminator, the generator attempts to produce images that are very close to being flawless. The discriminator makes an effort to enhance its performance by discriminating between actual and fraudulent samples, despite the fact that these two types of samples cannot be distinguished [35]. The discriminator has the goal of increasing the next loss function, whereas the generator has the objective of decreasing it. In the process, \( x \) is input into the GAN generator, which then creates samples \( G \) by including a random noise variable \( P(\text{rnv}) \) into the equation \( (\text{rnv}) \). The estimate of the likelihood that an actual instance \( x \) occurs across the data distribution \( P_d \) is denoted by the discriminator’s variable \( D(x) \). To accomplish its primary goal, the encoder must first acquire knowledge of the representation feature, whereas the discriminator must work to identify the discriminating feature. Because the information stored in the generator and the discriminator often coincide, Chatterjee, Subhajit et al. [35] introduced an encoder into the discriminator. Figure 2 demonstrates the synthetic augmentation performed using GAN.

\[
\min \max \left[ V(D, G) \right] = -E_{\text{rnv} \sim P_d(x)} \left[ \log D(x) \right] + E_{\text{rnv} \sim P_d(x)} \left[ \log (1 - D(G(\text{rnv}))) \right] \tag{1}
\]

\[
L_{\text{recons}}^{\text{pixel}} = E_{I \sim D_{\text{Encoder}}(\$), \$ \sim I_{\text{off}}} \left[ \| K(q) - T(\$) \| \right] \tag{2}
\]
More common means of data augmentation include horizontal and vertical flip, which may be accomplished by employing Equations (3) and (4) in the case of horizontal flipping and Equations (5) and (6) in the case of vertical flipping. This augmentation is one of the easiest to implement, and its usefulness has been shown across a variety of datasets. After the images underwent synthetic augmentation, traditional means of augmentation were used to increase the dataset as shown in Figure 3.

\[
\begin{bmatrix}
\hat{p} \\
\hat{q} \\
1
\end{bmatrix} =
\begin{bmatrix}
-1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix} \times
\begin{bmatrix}
p \\
q \\
1
\end{bmatrix}
\]

\hat{p} = -p, \hat{q} = q \quad (3)

\[
\begin{bmatrix}
\hat{p} \\
\hat{q} \\
1
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 \\
0 & -1 & 0 \\
0 & 0 & 1
\end{bmatrix} \times
\begin{bmatrix}
p \\
q \\
1
\end{bmatrix}
\]

\hat{p} = p, \hat{q} = -q \quad (5)

The block representation of the complete procedure from experimental setup to classification is presented in the GAN images in Figure 4. The GAN is used to provide high-quality images to data augmentation, which improved the segmentation performance for the produced dataset.

In order to generate the database, two cameras were installed in Karachi, one at Sir Syed University of Engineering and Technology, Gulshan-e-Iqbal, and the other at Ziauddin University North Nazimabad, as shown in Figure 5. HikVision cameras with ultra-HD (4K) with 40 fps were used because of their long-distance viewing capacity to capture live video in afternoons and evenings. From Monday through Friday, the recording period was between two and three minutes during the peak office hours of 5:00 p.m. to 5:10 p.m. and the peak school hours of 12:30 p.m. to 12:40 p.m.

Eight videos were recorded, six of which were for congestion and two of which were not. Frame extraction was used to extract frames from available recordings. Finally, a subset of the thousands of frames in the retrieved database were chosen for further processing. Images were selected that demonstrated changes in congested states prior to providing them to the GAN for high-resolution images and data augmentation.

Before the augmentation, the total number of accessible photographs was 700 for congestion, while the number of images for non-congestion was 450. Both training and testing versions of this dataset were created. The datasets of the congested and non-congested images, both before and after the process of augmentation, are shown in Tables 1 and 2, demonstrating the distributions of the images for training and testing.
Figure 3. Conventional augmentation applied on a GAN-based synthetic augmented dataset: (a) original synthetic high-contrast image; (b) image flipped 90°; (c) image flipped 270°; (d) mirror image.

Figure 4. Architecture of the setup used with GAN along with CNN as the classifier.

Figure 5. Pictorial view of the installed camera.
Table 1. The numbers of images in the dataset before and after the process of augmentation.

<table>
<thead>
<tr>
<th>Image Class</th>
<th>Original</th>
<th>Total Images (Classical Images + GAN Augmented Images)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Congested</td>
<td>700</td>
<td>1250</td>
</tr>
<tr>
<td>Non-congested</td>
<td>450</td>
<td>750</td>
</tr>
<tr>
<td>Total</td>
<td>1150</td>
<td>2000</td>
</tr>
</tbody>
</table>

Table 2. Distribution of the image dataset for training and testing.

<table>
<thead>
<tr>
<th>Image Class</th>
<th>Training</th>
<th>Testing</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Congested</td>
<td>1062</td>
<td>188</td>
<td>1250</td>
</tr>
<tr>
<td>Non-congested</td>
<td>637</td>
<td>113</td>
<td>750</td>
</tr>
<tr>
<td>Total</td>
<td>1699</td>
<td>301</td>
<td>2000</td>
</tr>
</tbody>
</table>

4. Pre-Trained Classifiers: A Brief Overview

In this paper, a CNN-based five-layered deep learning model is proposed, and its performance is compared with that of a number of other well-known pre-trained models, including ResNet-50 and DenseNet-121.

4.1. ResNet-50

An input image collection with a resolution of $224 \times 224 \times 3$ pixels is required for the ResNet-50 architecture. The first layer of convolution consists of 64 filters and has a kernel size of $7 \times 7$, which produces a matrix that is $112 \times 112$. Even though they are included in the coded model, the pooling layers are not included in the model of ResNet-50 that can be seen in Figure 6. In the subsequent convolution, there will be a $1 \times 1$ matrix of 64 kernels, then there will be a $3 \times 3$ matrix of 64 kernels, and finally there will be a $1 \times 1$ matrix of 256 kernels. A total of nine convolutional layers were created by combining the results of these three layers into a batch that was then repeated three times. After that, we could see a kernel consisting of $1 \times 1 \times 128$, then a kernel consisting of $3 \times 3 \times 128$, and lastly a kernel consisting of $1 \times 1 \times 512$. This procedure was repeated a total of four times, giving us a total of twelve layers to work with at this point. Following that was a kernel with the dimensions of $1 \times 1 \times 256$, which was followed by 2 more kernels with the dimensions of $3 \times 3 \times 256$ and $1 \times 1 \times 1024$, all of which were repeated 6 times for a total of 18 layers. Then came kernels with sizes of $1 \times 1 \times 512$, followed by 3 more kernels with sizes of $3 \times 3 \times 512$, and $1 \times 1 \times 2048$. This process was repeated three times, resulting in a total of nine layers for this batch. After that, we carried out an average pool, and at the end of the process, we obtained a completely connected layer that consisted of 1000 nodes and a SoftMax function that gave an output of 1 layer [36,37].

Figure 6. ResNet-50 layered architecture from [29].
4.2. DenseNet-121

DenseNet, which stands for dense convolution network, is a solution to the issue of vanishing gradients that may arise in CNNs owing to their deep structure. DenseNet solves this issue by changing a CNN’s design to include direct connections between each layer. DenseNet-121’s architecture consists of dense blocks with varying numbers of transition layers; within the same dense block, weights are distributed among many inputs from which early-stage features are retrieved. Each dense block undergoes convolution with a kernel size of $3 \times 3$. DenseNet’s dimensions are constant, but each dense block has distinct filters [38,39]. A block representation of DenseNet-121 is shown in Figure 7.

Figure 7. DenseNet-121 layered architecture where yellow block represents Dense Block; Blue Blocks represents Transition Block, from [30].

5. Proposed CNN Model for Traffic Congestion and Non-Congestion Classification

The use of images taken from the street to determine whether or not there is traffic congestion provides a practical illustration of the many possible results that may be reached using a deep learning algorithm. By itself, the model is able to learn all of the essential characteristics. As a programming optimization, gradient descent boosts neurons or nodes until it finds the function of local minimum. The execution of the proposed 5-layered model is carried out on an Intel 7700 processor from the 7th-generation Core-i7, which has 16 gigabytes (GB) of DDR4 random access memory (RAM) and 2400 busses. NVIDIA GeForce GTX 1080 with 8 GB of RAM make up the system’s graphical processing unit (GPU).

For the 5-layered CNN model that has been presented, a grayscale image with a resolution of $224 \times 224$ pixels is needed as input. The image that was captured from the live video feed of the traffic congestion was then altered so that it could fulfill the appropriate model criteria for both training and testing. Table 3 illustrates the hierarchy of the 5-layered CNN model that was presented, with the input picture having dimensions of $224 \times 224 \times 1$. The model comprises a total of five convolutional layers and three pooling layers. The information shown in Table 3 makes it abundantly evident that every convolutional layer has its unique set of trainable parameters. For example, convolutional layer 1 comprises...
800 trainable parameters. The model that has been proposed has a total of 124,450 trainable parameters, whereas Resnet-50 and DenseNet-121 have 23 million and 7.2 million trainable parameters, respectively. The model proposed provides better accuracy with far fewer trainable parameters compared with the parameter of the pretrained classifiers. Figure 8 is a block representation of the model that has been presented. It demonstrates the input and output resolution of each layer in addition to the corresponding filters that were used to train the model. The rectified linear unit (ReLU) was employed as an activation function for each convolutional layer, but the sigmoid was only used for the fully connected layer’s activation function. The first two convolutional layers each have a stride of 2, whereas the latter three convolutional layers each have a stride of 1.

**Figure 8.** Block model of the proposed 5-layered CNN classifier to predict traffic congestion.
Table 3. Tabular representation of the 5-layered traffic congestion classification model.

<table>
<thead>
<tr>
<th>Layers</th>
<th>Nos of Filters</th>
<th>Filter Size</th>
<th>Activation Function</th>
<th>Padding</th>
<th>Stride</th>
<th>Feature Map Size</th>
<th>Trainable Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>224 x 224 x 1</td>
<td>-</td>
</tr>
<tr>
<td>Conv2D (1)</td>
<td>16</td>
<td>7 x 7</td>
<td>ReLU</td>
<td>-</td>
<td>2</td>
<td>109 x 109 x 16</td>
<td>800</td>
</tr>
<tr>
<td>Max-Pool (1)</td>
<td>-</td>
<td>3 x 3</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>107 x 107 x 16</td>
<td>-</td>
</tr>
<tr>
<td>Dropout</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>108 x 108 x 16</td>
<td>-</td>
</tr>
<tr>
<td>Conv2D (2)</td>
<td>32</td>
<td>5 x 5</td>
<td>ReLU</td>
<td>-</td>
<td>2</td>
<td>54 x 54 x 32</td>
<td>12,832</td>
</tr>
<tr>
<td>Dropout</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>54 x 54 x 32</td>
<td>-</td>
</tr>
<tr>
<td>Conv2D (3)</td>
<td>32</td>
<td>3 x 3</td>
<td>ReLU</td>
<td>-</td>
<td>1</td>
<td>54 x 54 x 32</td>
<td>9248</td>
</tr>
<tr>
<td>Max-Pool (2)</td>
<td>-</td>
<td>3 x 3</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>18 x 18 x 32</td>
<td>-</td>
</tr>
<tr>
<td>Dropout</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>18 x 18 x 32</td>
<td>-</td>
</tr>
<tr>
<td>Conv2D (4)</td>
<td>64</td>
<td>3 x 3</td>
<td>ReLU</td>
<td>-</td>
<td>1</td>
<td>18 x 18 x 64</td>
<td>18,496</td>
</tr>
<tr>
<td>Conv2D (5)</td>
<td>128</td>
<td>3 x 3</td>
<td>ReLU</td>
<td>-</td>
<td>1</td>
<td>18 x 18 x 128</td>
<td>73,856</td>
</tr>
<tr>
<td>Max-Pool (3)</td>
<td>-</td>
<td>3 x 3</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>6 x 6 x 128</td>
<td>-</td>
</tr>
<tr>
<td>Dropout</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>6 x 6 x 128</td>
<td>-</td>
</tr>
<tr>
<td>Flatten</td>
<td>-</td>
<td>-</td>
<td>Sigmoid</td>
<td>-</td>
<td>-</td>
<td>4608</td>
<td>-</td>
</tr>
<tr>
<td>FCL (2)</td>
<td>-</td>
<td>-</td>
<td>Sigmoid</td>
<td>-</td>
<td>2 x 1</td>
<td>2 x 1</td>
<td>9218</td>
</tr>
<tr>
<td>Total</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>124,450</td>
</tr>
</tbody>
</table>

FCL: Fully Connected Layer; Conv2D: 2D-Convolutional layer; ReLU: Rectified Linear Unit.

6. Results

In order to verify the accuracy of the results that the trained model produced, four distinct performance parameters were applied. Accuracy, sensitivity/recall, precision, and the F1 score are the criteria that were used to evaluate the performance of the model. When comparing models, the performance was evaluated here using a traffic congestion dataset; the model’s performance might change depending on different applications or datasets that are unique to the application. The performance of pre-trained classifiers and the proposed model utilized for testing and training with the help of a confusion matrix are presented in Tables 4 and 5; these contain an analysis of the performance of the proposed five-layered CNN model based on four performance parameters. The proposed model for the classification of traffic congestion fared better than the pre-trained classifiers (ResNet-50 and DenseNet-121) because it provides the best accuracy. The suggested model was found to have an accuracy of 98.63%, while the accuracies of ResNet-50 and DenseNet-121 were 90.59% and 93.15%, respectively. Equations (7)–(10) demonstrate the parameters to measure the performance of the proposed model.

\[
\text{Accuracy} = \frac{(TP + TN)}{(TP + FN) + (FP + TN)} \quad (7)
\]

\[
\text{Sensitivity/Recall} = \frac{(TP)}{(TP + FN)} \quad (8)
\]

\[
\text{Precision} = \frac{(TN)}{(FP + TN)} \quad (9)
\]

\[
\text{F1 Score} = \frac{(2 \times TP)}{(2 \times TP + FN + FP)} \quad (10)
\]

Table 4 presents the results of using a confusion matrix to evaluate the performance of the pre-trained classifiers, which are then put to use for testing and training purposes. It has a projected class in addition to the actual class for both the congestion and non-congestion scenarios.
Table 4. Acquired Confusion Matrix of ResNet-50, DenseNet-121, and the Proposed Five-layered CNN Model.

<table>
<thead>
<tr>
<th></th>
<th>True Class</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ResNet-50</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Predicted Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>118</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>NC</td>
<td>14</td>
<td>113</td>
<td></td>
</tr>
<tr>
<td><strong>DenseNet-121</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Predicted Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>118</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>NC</td>
<td>11</td>
<td>113</td>
<td></td>
</tr>
<tr>
<td><strong>5-Layered CNN Model (Proposed)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Predicted Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>118</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>NC</td>
<td>3</td>
<td>113</td>
<td></td>
</tr>
</tbody>
</table>

C: Congestion; NC: Non-Congestion; CNN: Convolutional Neural Network.

Table 5 depicts the performance parameters of the proposed model, the pretrained models, and the state-of-the-art model proposed by different researchers. For the data that were used to detect traffic congestion, three different models were evaluated. For the purposes of training and testing, images are chosen at random from the database. The ResNet-50 model has a prediction accuracy of 90.59%, and its training time on GPU is around 2.6 h. When using the default batch size, DenseNet-121 took approximately 3 h to train and achieved an accuracy of 93.15%. The proposed 5-layer CNN architecture, which works on synthetic and enhanced images that were created by GAN for 30 epochs, reached a maximum accuracy of 98.63%, making it the most accurate solution we tested. Its performance was enhanced due to the augmented dataset used to increase the quantity of the images using GAN. Because of the high resolution of the images, this accuracy is eight times better than that of ResNet-50 and 5% better than that of DenseNet-121. Jingqiu Guo et al. [26] used a CNN–RNN model to predict traffic congestion and attained accuracy of 79.89%. Jason Kurniawan et al. [40] implemented a CNN-based deep earning model that learned using CCTV images and attained accuracy of 89.50%. We also implemented a CNN-based model [40] on our GAN-implemented dataset, and we achieved accuracy of 92.005%. Hua Cui et al. [21] use two different pre-trained classifiers to predict traffic congestion, AlexNet–CNN and GoogleNet–CNN, and succeeded in achieving 98.2% and 97.6% accuracy, respectively. We utilized our GAN-based dataset on AlexNet–CNN [21] and achieved accuracy of 97.34%. Figure 9 shows a graphical illustration of the loss and accuracy experienced throughout training and testing. The accuracy of the proposed model reaches 98.63%, as can be seen from this graphical depiction of the data.

Table 5. Performance parameters of the proposed model in comparison with pre-trained classifiers.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy (%)</th>
<th>Recall (%)</th>
<th>Precision (%)</th>
<th>f1 Score (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50</td>
<td>90.59</td>
<td>89.40</td>
<td>91.86</td>
<td>90.77</td>
</tr>
<tr>
<td>DenseNet-121</td>
<td>93.15</td>
<td>91.5</td>
<td>94.96</td>
<td>93.30</td>
</tr>
<tr>
<td>CNN-RNN [26]</td>
<td>79.86</td>
<td>30.06</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>CNN [35]</td>
<td>89.50</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>CNN [40] (on GAN applied dataset)</td>
<td>92.005</td>
<td>93.66</td>
<td>90.39</td>
<td>92.009</td>
</tr>
<tr>
<td>AlexNet-CNN [21]</td>
<td>98.2</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>AlexNet-CNN [21] (on GAN applied dataset)</td>
<td>97.34</td>
<td>96.24</td>
<td>96.68</td>
<td>97.18</td>
</tr>
<tr>
<td>GoogleNet-CNN [21]</td>
<td>97.6</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>BLSTM-CNN [36]</td>
<td>98%</td>
<td>94%</td>
<td>96%</td>
<td>–</td>
</tr>
<tr>
<td>5-layered CNN Model **</td>
<td>98.63</td>
<td>97.73</td>
<td>99.20</td>
<td>98.33</td>
</tr>
</tbody>
</table>

**: Proposed model
Figure 9. (Left) Training and testing loss. (Right) Training and testing accuracy.

Figure 10 illustrates the classification output of the five-layer CNN model for both congested and non-congested traffic conditions. A random selection of sixteen images is made from the testing database, and from those sixteen images, the precise output is predicted. Of those 16 images, 14 are classed as congested road segments, while the remaining 2 are classified as non-congested road segments.

Table 6 depicts the accuracies acquired by the proposed GAN-augmented dataset with the five-layered CNN model applied and the original dataset without using GAN augmentation. It can be seen that the accuracy of the proposed model for the GAN-augmented dataset is 98.63%, which is better than the accuracy achieved by the model that did not use the GAN-augmented dataset.

Figure 10. Images showing the trained model’s predictions for the congestion and non-congestion classes.
Table 6. Accuracies of original dataset vs. the proposed GAN-augmented dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Dataset</td>
<td>90.18%</td>
</tr>
<tr>
<td>GAN augmented Dataset (proposed)</td>
<td>98.63%</td>
</tr>
</tbody>
</table>

7. Conclusions

This study aims to solve the classification issue between congested and uncongested traffic states, particularly in developing nations like Pakistan. The lack of traffic datasets is a major barrier in this case. In order to create the dataset, traffic images from video recordings captured in Karachi, Pakistan, were extracted and augmented to create a larger number of images. This attempt was made to solve this issue by utilizing a GAN’s superior deep learning to produce images from the provided dataset. The use of GAN significantly improved the image quality as there had been blurry images in the obtained dataset. In order to validate the dataset, a five-layered CNN model was proposed that was used to distinguish congested from non-congested images. DenseNet-121, ResNet-50, and the proposed five-layered CNN model were compared, and CNN truly demonstrated better performance for the prediction of congested and non-congested traffic; it achieved accuracy of 98.63%, in contrast to the accuracies of 90.59% and 93.15% for ResNet-50 and DenseNet-121, respectively. Further, the proposed model also produced superior accuracy in comparison with other state-of-the-art models based on CNN (CNN–RNN, CNN, AlexNet–CNN, GoogleNet–CNN, and BLSTM–CNN). The proposed model is simple to use and will be beneficial for all practitioners involved with traffic management agencies across the world in classifying urban traffic as congested or non-congested.

Author Contributions: Conceptualization, U.J.; Methodology, U.J.; Writing—original draft U.J.; Investigation, U.J.; Supervision, M.A. (Muhammad Asif); Project administration, M.A. (Muhammad Asif); Writing—Review & Editing, M.R.; Visualization, M.R.; Formal analysis, A.A.S.; Investigation, A.A.S.; Validation, S.M.U.T.; Writing—Review & Editing, M.A. (Muhammad Aamir). All authors have read and agreed to the published version of the manuscript.

Funding: This research work is funded by the Data Acquisition Processing and Predictive Analytics Lab, National Center in Big Data and Cloud Computing (NCBC) Lab, Karachi, Pakistan from HEC Pakistan, DP-2208-0003.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study will be available on request. The data are not publicly available due to privacy reasons.

Acknowledgments: We would like to acknowledge the NCBC Lab for their support in installing and using their resources to develop the dataset.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lu, J.; Li, B.; Li, H.; Al-Barakani, A. Expansion of city scale, traffic modes, traffic congestion, and air pollution. Cities 2021, 108, 102974. [CrossRef]
5. Bernas, M.; Placzek, B.; Korski, W.; Loska, P.; Smyla, J.; Szymala, P. A survey and comparison of low-cost sensing technologies for road traffic monitoring. Sensors 2018, 18, 3243. [CrossRef]


34. Gregurić, M.; Vujić, M.; Alexopoulos, C.; Miletić, M. Application of deep reinforcement learning in traffic signal control: An overview and impact of open traffic data. *Appl. Sci.* 2020, 10, 4011. [CrossRef]


