

Article

Large-Scale Road Network Traffic Congestion Prediction Based on Recurrent High-Resolution Network

Sachin Ranjan ¹, Yeong-Chan Kim ^{1,2} , Navin Ranjan ³ , Sovit Bhandari ¹  and Hoon Kim ^{1,2,*}

¹ IoT and Big-Data Research Center, Incheon National University, Yeonsu-gu, Incheon 22012, Republic of Korea; ranjansachin1993@gmail.com (S.R.); kyc0288@inu.ac.kr (Y.-C.K.); sovit198@gmail.com (S.B.)

² Department of Electronics Engineering, Incheon National University, Yeonsu-gu, Incheon 22012, Republic of Korea

³ Department of Electrical and Computer Engineering, Rochester Institute of Technology, Rochester, NY 14623, USA; ranjannavin07@gmail.com

* Correspondence: hoon@inu.ac.kr

Abstract: Traffic congestion is a significant problem that adversely affects the economy, environment, and public health in urban areas worldwide. One promising solution is to forecast road-level congestion levels in the short-term and long-term, enabling commuters to avoid congested areas and allowing traffic agencies to take appropriate action. In this study, we propose a hybrid deep neural network algorithm based on High-Resolution Network (HRNet) and ConvLSTM decoder for 10, 30, and 60-min traffic congestion prediction. Our model utilizes the HRNet's multi-scale feature extraction capability to capture rich spatial features from a sequence of past traffic input images. The ConvLSTM module learns temporal information from each HRNet multi-scale output and aggregates all feature maps to generate accurate traffic forecasts. Our experiments demonstrate that the proposed model can efficiently and effectively learn both spatial and temporal relationships for traffic congestion and outperforms four other state-of-the-art architectures (PredNet, UNet, ConvLSTM, and Autoencoder) in terms of accuracy, precision, and recall. A case study was conducted on the dataset from Seoul, South Korea.

Keywords: traffic congestion; deep learning; big data; recurrent HRNet



Citation: Ranjan, S.; Kim, Y.-C.; Ranjan, N.; Bhandari, S.; Kim, H. Large-Scale Road Network Traffic Congestion Prediction Based on Recurrent High-Resolution Network. *Appl. Sci.* **2023**, *13*, 5512. <https://doi.org/10.3390/app13095512>

Academic Editor: Xianpeng Wang

Received: 4 April 2023
Revised: 24 April 2023
Accepted: 25 April 2023
Published: 28 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Traffic congestion is a real problem that affects urban areas around the world due to rapid urbanization and desire for private travel [1]. It occurs when demand exceeds the existing road system's capacity, causing the traffic to slow down or come to a complete halt. The consequences of traffic congestion are far-reaching and include increased commute times, energy consumption, environmental degradation, and traffic accidents [2–4]. According to a report by the Texas A&M Transportation Institute, traffic congestion costs the US economy approximately \$166 billion USD per year in lost productivity and increased fuel cost. In China, traffic congestion contributes to additional air pollutants such as PM_{2.5}, and O₃, leading to increased rates of premature mortality [5]. Furthermore, the World Health Organization's report on road traffic crashes shows that the deaths of approximately 1.3 million people around the world each year are linked to congestion [6]. Therefore, managing traffic congestion is a crucial area of study, and scholars from various disciplines are working on preventing congestion by analyzing big data using artificial intelligence [7] to improve the current scenario.

The intelligent transportation system (ITS) is an efficient and robust traffic management system that integrates a variety of advanced technologies [8,9] and has improved the traffic congestion problem. A crucial component of ITS is the predictive model, which estimates traffic patterns [10,11] and predicts short- or long-term traffic conditions [12–14] using traffic data such as traffic speed, flow volume, and congestion levels. Previous

research on traffic prediction has mainly relied on data from fixed sensors (such as road sensors, inductive loops, or traffic cameras) installed on road networks or from vehicle networks, including VANET or Floating Car systems. In [15], traffic-flow data from human-driven vehicles at road intersections was used to predict and model the road traffic flow pattern using an Artificial Neural Network. In [16], several simulation studies were conducted based on real vehicle networks and intelligent traffic communication systems to streamline the traffic density and reduce traffic congestion using wireless communication technologies such as vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-everything (V2X). However, acquiring and using such data and technologies can be costly and often requires special permission.

In recent years, public web services such as Google Traffic [17], Bing Maps [18], Seoul Transportation Operation and Information Service (TOPIS) [19], and Baidu Maps [20] have started providing road-level traffic information. These web services are publicly accessible, pre-processed, and provide traffic information as a map for almost all the cities in the world, but only a few studies have utilized this type of data. In the past, the curse of dimensionality posed a challenge, as predicting traffic patterns involves processing series of images, which could be computationally expensive. However, the significant increase in computing power in recent years has made using this type of data for traffic prediction feasible.

Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM) and Convolutional LSTM (ConvLSTM) are well-known for their capability to learn temporal relationships across sequential data and have shown promising results in various applications such as recognition [21], translation, and time-series prediction [22,23]. Convolutional Neural Networks (CNNs) are well known for their ability to extract features from image or video datasets. Many existing CNN-based algorithms are designed to go from high-resolution to low-resolution and progressively recover the original resolution using upsampling operations, which can limit their ability to capture fine-grained details and lead to suboptimal performance. In contrast, the Deep High-Resolution Network (HRNet), introduced in [24], adopts a unique approach for visual representation learning. HRNet starts with a high-resolution convolution stream and gradually adds high-to-low resolution convolution streams one by one, connecting multi-resolution streams in parallel and conducting repeated multi-resolution fusions to capture fine-grained details. This approach allows HRNet to capture more comprehensive information and achieve better performance.

Inspired by the success of HRNet and ConvLSTM neural networks in computer vision, this paper presents an approach for learning comprehensive features from sequential traffic images. Our approach is based on the HRNet backbone, followed by a ConvLSTM-based decoder that generates the traffic forecast map. The main contributions of the paper are as follows:

- We introduce a new prediction model called Recurrent High-Resolution Networks (RHRNet), which consists of HRNet as a backbone and a ConvLSTM-based decoder.
- The proposed architecture leverages the advantages of HRNet, which maintains high-resolution features along with low-resolution features throughout the network in parallel, plus a ConvLSTM-based decoder to learn the spatio-temporal relationships of all resolution feature maps from HRNet, and aggregates these features into a unified high-resolution forecast.

In this section, we describe the background and motivation of the study. The rest of the paper is structured as follows: in Section 2, we present the literature review of related work on traffic congestion. In Section 3, we describe the methodology, including the problem statement and architecture design. In Section 4, we present the data description, the metrics used for training and testing the algorithm, and comparison models. In Section 5, we provide details on the proposed model architecture and performance comparison. Finally, in Section 6, we present the conclusion of our study and discuss future directions for traffic congestion prediction research.

2. Related Work

In the past, researchers typically used a parametric approach, which was a data-driven method that emphasized creating statistical and mathematical models for examining time-series relationships within traffic data. The researchers believed that linearity and stationarity were sufficient for predicting future trends in traffic data and used techniques such as the historical average model [25] and error component model [26]. Later, the ARIMA model was introduced, but it has limitations, including a focus on mean values and an inability to predict extremes. Other models, such as seasonal ARIMA [27] and Kalman filter models [28], require large historical datasets and are sensitive to changes in traffic data.

The limitation in parametric approaches led researchers to explore nonparametric models such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Bayesian Network (BN), and Artificial Neural Network (ANN). Nonparametric models rely on training data to determine the model structure and the number of parameters. KNN searches for similar data points in the historical database to predict traffic flow while SVM models [29] use the principle of structural risk minimization and are well suited for small samples and high-dimensional nonlinear data. Various SVM-based models, including seasonal SVM, least-square SVM [30], online-SVM [31], and wavelet-SVM [32], have been developed to overcome issues such as overfitting and local minima. BN in [33,34] considers the causal relationship between random variables and can handle incomplete data through message-passing. All the above-mentioned work requires significant domain knowledge and feature engineering. However, the ANN model in [35] shows its capability to work with multi-dimensional data without any feature engineering and can introduce non-linearity in the learning process but lacks satisfactory performance due to its shallow hidden layers.

Other researchers have shifted their focus towards the Deep Neural Networks (DNNs) including RNN and hybrid architecture to predict the traffic information. Recurrent neural networks such as LSTM directly learn the temporal relations from the sequential data and ignore the spatial information. They have been used for various traffic-related applications such as traffic speed prediction [36], traffic flow prediction [22,37], and traffic congestion prediction [38]. In [39], the researcher uses the LSTM network to predict the road-based congestion information for few roads; the data was collected from an online source. In [37], the researcher presents the LSTM network in a predictive framework based on the correlated traffic data, i.e., the data recorded simultaneously from the different regions of the same transportation networks, which enables the incorporating of both temporal and spatial correlations in the data. In [14], the researcher presents an autoencoder-based architecture that learns temporal information to predict the short-term traffic forecast, however, this approach does not incorporate spatial relation learning. The authors unnaturally compressed the larger traffic map resolution to fit in their computing model; the compressed image was not visually intuitive, and a lot of road information was lost. The authors in [40] introduce a hybrid model by combining KNN-LSTM; the model mines spatio-temporal relationships by selecting similar neighbors in the region and accounting for temporal variability to predict the traffic flow. In [41], the author designs an advanced traffic congestion early-warning system based on an extreme learning machine combined with a modified multi-objective dragonfly optimization algorithm.

In [42], the researcher proposes a LC-RNN model, where the LC part uses look-up convolutional operations to extract the spatial information from the adjacent road and recurrent layers to learn the temporal patterns. In [43], the research presents an SCRNN model, a combination of CNN and LSTM neural networks, to extract spatial and temporal relations to predict the traffic speeds of 278 road links. In [44], the researcher proposes a PCNN model, which uses vehicle passage records from the surveillance cameras on roads, to predict the short-term traffic congestion. In [45], the authors propose a spatial-temporal complex graph convolution network (ST-CGCN) to predict traffic. The authors first generate a complex

correlation matrix for spatial and temporal feature and then feed it to a 3D convolution operator followed by LSTM. In [46], the author presents a traffic congestion prediction with bilateral alternation of a spatiality and temporality (TCP-BAST) algorithm. The algorithm first captures spatial correlation based on graph attention networks, then captures temporal correlation using masked attention networks, and finally predicts traffic congestion on multiple road sections. In [47], the author presents a city-wide traffic congestion prediction based on data from TOPIS using a hybrid neural network based on CNN, LSTM and Transposed CNN which incorporates both spatial and temporal features, in which the CNN operation learns the spatial information and LSTM learns the temporal information.

In contrast to [42–46], where the authors had to go through a complex process to incorporate spatial-temporal features of the dataset into the model, in [47], the author presents a simple and straightforward algorithm without any complex data-preprocessing for traffic congestion prediction. Building on the simplistic design presented in [47], in this study, we replace the initial CNN blocks with the HRNet architecture to learn multi-scale feature representation of the input dataset and replace the LSTM layer by Convolutional LSTM layer for learning spatial and temporal information.

3. Methodology

In this section, we first describe the problem statement for the time-series traffic congestion prediction, then describe our proposed architecture.

3.1. Problem Statement

Let $N \in \{x_1, x_2, \dots, x_n\}$ be n chronological images in a database collected at interval t . The main objective of this study is to develop a deep neural network \mathcal{F} , which is capable of utilizing p past sequential images to forecast the traffic congestion level of the road network at the prediction horizon of k . For the i^{th} time-series sample, the input to the network is $X_i = \{x_{i-p}, x_{i-(p-1)}, \dots, x_{i-1}, x_{i-0}\}$, and the corresponding label is $Y_i^k = \{x_{i+k}\} \in N$. Since we have a label for each input sample, we can use a supervised learning method to train the model. The model \mathcal{F} can be defined as in Equation (1):

$$Y_i^k = \mathcal{F}(X_i, \theta) \quad (1)$$

where θ is the model parameter.

3.2. Database

There are numerous online and real-time traffic information providers worldwide, such as Google Traffic Map [17] and Bing Map [18], which provide traffic information for nearly all the cities around the world. Baidu Map [20] provides information for all of China, and TOPIS [19] focuses on Seoul, South Korea. For our study, we obtained the congestion map from the TOPIS online web service, which provides accurate real-time congestion levels of the city's road network. Figure 1a shows the raw image of central Seoul, South Korea, captured on 20 September 2019, at 15:05, which includes road network, background, and text. The TOPIS traffic image has three congestion levels: Jam, Slow, and Free State, where the color 'Red' represents the Jam, 'Yellow' represents the Slow, and 'Green' represents the Free congestion levels. The congestion levels are classified based on the average speed of vehicles on the road segments, where a speed greater than 25 km/h is Free, between 10 km/h to 25 km/h is Slow, and lower than 10 km/h is Jam. To extract the road-level congestion information based on [10], we performed an image-masking operation. First, we calculated the RGB color boundary for each congestion level and generated a mask image. Finally, we performed a bitwise operation between the mask image and raw image to obtain the traffic image with only congestion levels, as shown in Figure 1b.

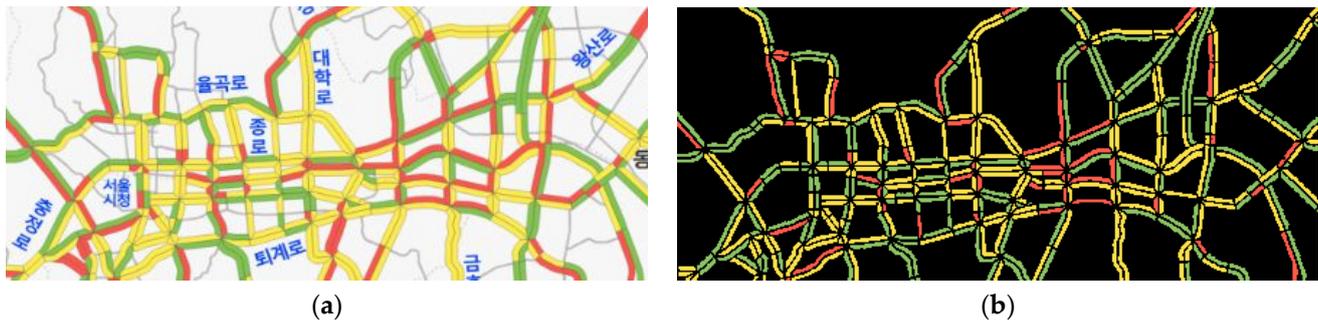


Figure 1. Road network of Seoul city with the Jam, Slow, and Free congestion levels denoted in red, yellow, and green, respectively. (a) A sample of raw image data capture from TOPIS website. (b) A sample of image after masking operation where black color represent the background.

3.3. Proposed Architecture

Most Convolutional Neural Network (CNN)-based architecture for traffic congestion prediction follows a backbone architecture that gradually reduces the spatial size of the feature maps, using convolutional and pooling operations to progressively go from high-resolution to low-resolution and then revert back to the original resolution using upsampling operations such as dilated convolution or bilinear interpolations. However, this approach results in weak, spatially imprecise results. In this paper, we propose a multi-scale deep neural network, named the Recurrent Multi-scale High Resolution Network (RHRNet), which consists of a modified HRNet [25] backbone and a convolutional LSTM layer (convLSTM)-based decoder. Our backbone is based on modified HRNet [25], which provides a multi-scale approach for spatial feature representations. It incorporates both high- and low-resolution feature maps across the network, maintaining high-resolution feature maps instead of recovering them from low-resolution feature maps. Additionally, the decoder in our model is based on convLSTM, allowing for the capture of time-series representations from the spatial features. The detail on the architecture is presented in the following sub-sections.

3.3.1. Tiny HRNet

Our modified HRNet architecture includes four stages and four resolution levels, like the original HRNet. However, unlike [25], each stage in our architecture has only two convolutional layers at each parallel level. This simplified backbone is shown in Figure 2 and is referred to as Tiny HRNet and has significantly reduced complexity compared to HRNet [25]. The backbone includes a stem, n parallel multi-resolution convolutions at n , and multi-resolution fusion. The past p chronological traffic images are first concatenated at the channel dimension, $H \times W \times F$ (here, $F = 3 \times p$) and fed into the stem, which consists of two convolution layers, each layer with strides of 2×2 , a kernel size of 3×3 , and a kernel channel of C to decrease input resolution to $\frac{H}{4} \times \frac{W}{4} \times C$.

Parallel Multi-Resolution Convolution. The output of the stem is fed to the 1st stage of the Tiny HRNet. The 1st stage has one level (L_1) with the feature resolution of output of the stem. The 2nd stage has two resolution levels (L_1 and L_2) with the feature maps resolution of $\frac{H}{4} \times \frac{W}{4} \times C$ and $\frac{H}{8} \times \frac{W}{8} \times 2C$ at L_1 and L_2 , respectively. While the resolution is decreased by one-half on both directions, the channel size of the feature maps is doubled. Similarly, the 3rd stage has three resolution levels (L_1, L_2 and L_3) with the feature maps resolution of $\frac{H}{4} \times \frac{W}{4} \times C$, $\frac{H}{8} \times \frac{W}{8} \times 2C$ and $\frac{H}{16} \times \frac{W}{16} \times 4C$ at L_1, L_2 and L_3 , respectively. Likewise, the 4th stage has four resolution levels (L_1, L_2, L_3 and L_4) with the feature maps resolution of $\frac{H}{4} \times \frac{W}{4} \times C$, $\frac{H}{8} \times \frac{W}{8} \times 2C$, $\frac{H}{16} \times \frac{W}{16} \times 4C$, and $\frac{H}{32} \times \frac{W}{32} \times 8C$ at L_1, L_2, L_3 and L_4 , respectively. All the convolutional layers in the Tiny HRNet use a 3×3 kernel size and have a stride of 1×1 , and a convolutional layer with 2×2 kernel size and 2×2 stride is used for feature map reduction.

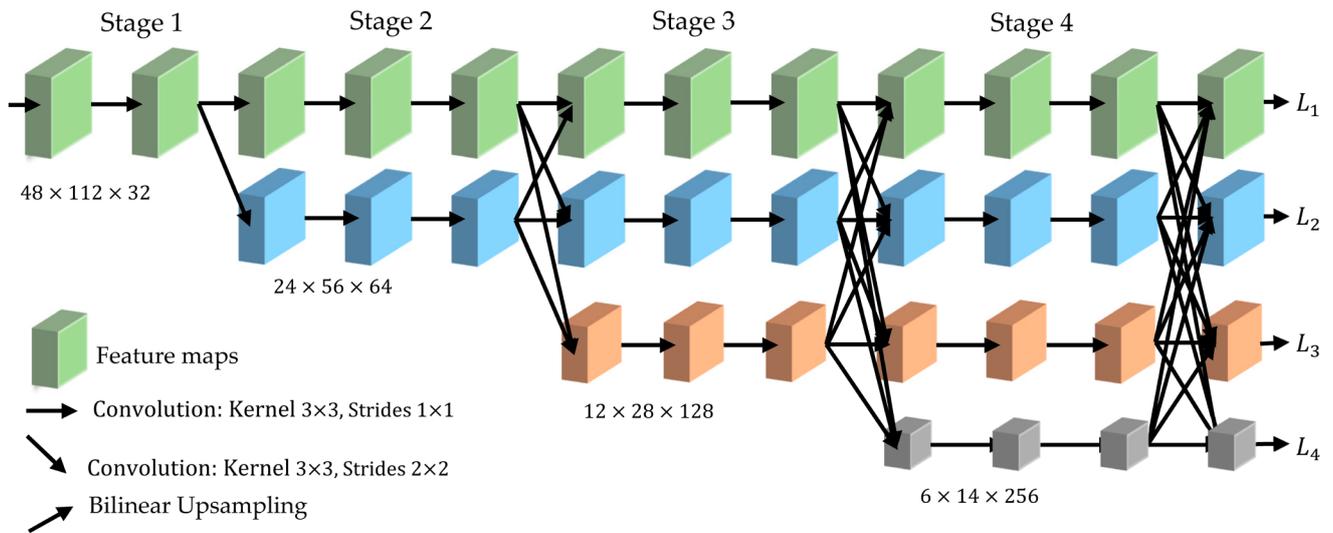


Figure 2. Backbone architecture of proposed RHRNet model. The figure shows the tiny-HRNet backbone architecture with four parallel multi-scale features map. The backbone is made up of four stages, with n th stage having ‘ n ’ level of parallel sub-networks. Each sub-network has the same feature map resolution and channel size throughout the network. The sub-networks are represented by different colors. The forward arrow represents 3×3 convolution operation, downwards arrow represents downsampling operation, and upward arrow represents upsampling process using bilinear interpolation.

Multi-Resolution Fusion. The multi-resolution fusion is employed when the network progresses from one stage to the next. The use of multi-scale feature fusion is motivated by the fact that single-scale feature maps may not be sufficient to capture all the information, particularly when objects in the image vary in size. Fusing feature maps across multiple scales allows the network to learn from fine-grained and coarser features, which leads to more spatially precise results. The feature maps of all levels in a particular stage are fused to generate the next stage. Since the resolution of the feature maps varies at different resolution levels, the first step involves aligning these feature maps. Specifically, we aim to convert the resolution of each level in the current stage to match the feature map resolution of the target level in the next stage. This alignment process involves performing convolutional downsampling to decrease the resolution of high-resolution feature maps and bilinear upsampling operations to increase the resolution of lower-resolution feature maps. Finally, we concatenate all the feature maps obtained from the previous step.

Figure 2 provides an example of multi-resolution fusion when transitioning from stage 3 to the feature map resolution level L_2 of stage 3. Stage 3 comprises three levels (L_1 , L_2 and L_3) with feature map resolutions that differ from the target level. We first transform the misaligned feature maps of stage 3. We reduce the feature map of L_1 by performing convolutional down-sampling using a 2×2 kernel size with a 2×2 stride. We increase the feature map resolutions of L_3 by performing a bilinear interpolation followed by a convolution layer with a kernel size of 1×1 and stride of 1×1 to match the feature map resolution of the target level. Finally, we concatenate all the obtained feature maps. Similar operations are performed for all other multi-resolution fusion layers.

3.3.2. Decoder Module

Our decoder module is designed to learn time-series representations for each resolution-level feature map from the 4th stage of the backbone and fuse them together to generate the accurate traffic congestion prediction. Figure 3 shows the decoder architecture. It consists of a stack of two convLSTM layers for each resolution level, followed by a multi-resolution fusion layer that generates a single output with a feature map of resolution equals to level L_1 . After fusion, the feature maps are combined with low-level features from the stem and

finally are followed by two convolution transpose layers to generate the original-resolution traffic prediction roadmap.

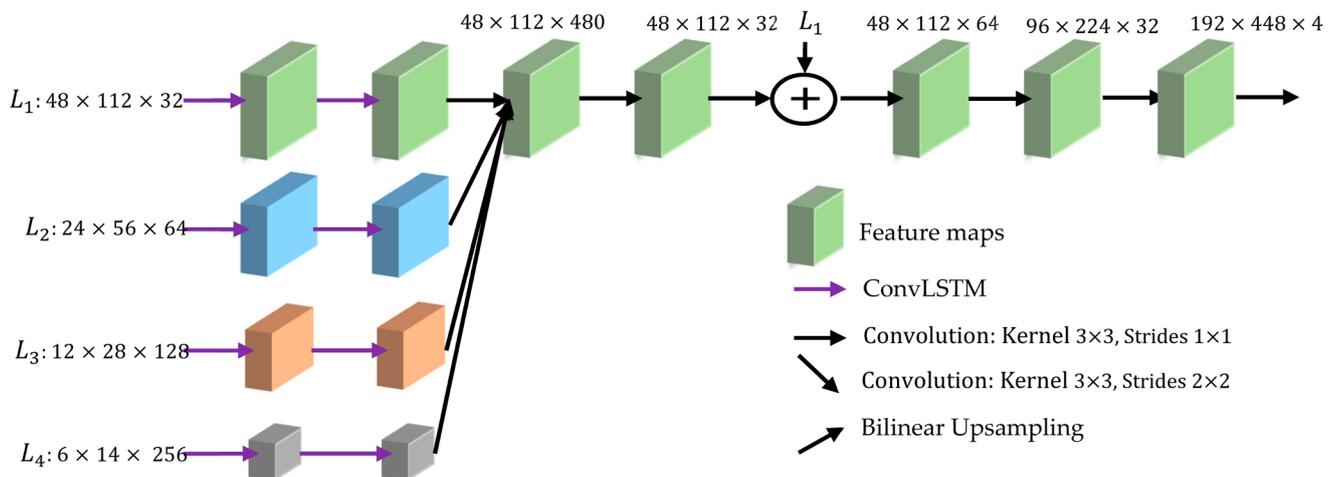


Figure 3. ConvLSTM-based decoder module. The output from all four levels of the backbone is fed to the decoder block where two ConvLSTM are performed. After that, at first, low-resolution feature maps are upsampled to match with the L_1 feature map and then concatenated to obtain 480 channels. The channel size is then reduced to 32 using 1×1 convolution operation and concatenated with L_1 feature map from end of backbone. This is followed by a 3×3 convolution and two transpose convolutions to obtain the original input image resolution with four channels, one for each class (Jam, Slow, Free, and background).

4. Experiment

4.1. Data Source

In this study, we used the road traffic congestion data in the form of a roadmap, which was obtained from our previous work [47,48], as shown in Figure 1. The data was collected by taking snapshots of traffic congestion maps from the open-source online web service, Seoul Transportation Operation and Information Service (TOPIS). This dataset includes information on the congestion levels (Jam, Slow and Free) of each road, represented by the colors red, yellow, and green, respectively. The dataset covers the traffic information of central Seoul, South Korea, with each snapshot image sized at 192×448 , covering a geographical area of $7.5 \text{ km} \times 17 \text{ km}$ (with a scale of $1 \text{ cm} = 1.3 \text{ km}$).

According to our previous research [48], the stochastic congestion maps of Central Seoul indicate a high likelihood of congestion between 09:00 to 21:00 h and a low likelihood in other hours. Additionally, all road networks experience congestion between 18:00 to 21:00 h, and forecasting alternate routes is not an effective method for traffic management during that period. Instead, congestion pricing, traffic redirect, or no-entry to heavy vehicles can be utilized. In contrast, from 09:00 to 12:00 h, the road networks are moderately congested, and the traffic forecasting method could be a more effective method. Thus, in this study, we focused on predicting traffic congestion levels during the morning peak hours from 07:00 to 12:00 on weekdays. The data was collected every 5 min from 19 September 2019 to 31 December 2019, resulting in a total of 104 days. However, for 26 days, either data was missing or only partially collected, and therefore these days were removed from the database. The training set consists of data from 19 September to 25 November, the validation set consists of data from 25 November to 30 November, and the remaining data from 1 December to 31 December are used for testing the trained model. Further details on data source, acquisition and preprocessing can be found in our previous work [47,48].

4.2. Metrics and Simulating Parameter

In order to demonstrate the effectiveness and superiority of the proposed RHRNet architecture, we compared it to baselines such as UNet as well as state-of-the-art works including convLSTM [47] Autoencoder [14,47]. The details on these architectures are presented in Section 4.3. RHRNet was trained using categorical cross-entropy loss, as shown in Equation (2), and its performance was evaluated using three metrics, outlined in Equations (3)–(5):

$$\mathcal{L}(y, y') = - \sum_i^c y_i (\log(y'_i)) \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

where y represents the ground truth label, y' represents the predicted value, c denotes the number of categories (Background, Jam, Slow and Free congestion levels), TP , TN , FP , and FN denote True Positive, True Negative, False Positive, and False Negative, respectively

All the aforementioned models were trained on a real-world traffic congestion data. We utilized the Adaptive Moment Estimation (Adam) optimizer with a learning rate of 1×10^{-4} a learning rate decay rate of 0.95, a variable moving average decay of 0.999, and a batch size of 8. The models were implemented in Python 3.7 language using the Keras deep learning library and trained on an Ubuntu 18.04.4 machine equipped with NVIDIA TITAN Xp Graphics Cards.

4.3. Model Training

The proposed architecture's training process is summarized in Algorithm 1. Input constraints for training the model are the traffic congestion dataset, the number of past image sequences, and the traffic congestion prediction horizon, as shown in Line 1. The learning rate is a hyperparameter described in Line 2. The input sequence and output label for training the model using gradient-descent backpropagation and Adam optimization algorithm are generated in lines 4 to 9. We use the 'HeUniform' distribution to initialize the parameters of the traffic congestion prediction, as shown in line 10. Next, we randomly select a mini batch of training instances S_b from the set S , as in line 12 and minimize the loss for the mini batch. The process is repeated until predefined stopping criteria are satisfied, as in lines 11 to 14. After the iteration is complete, an optimal set of parameters θ that represents the prediction model \mathcal{F} is generated, as in line 14.

4.4. Comparison Model

In our study, we evaluated the effectiveness of RHRNet by comparing it to four other models: PredNet [47], ConvLSTM [47], Autoencoder [14,47], and UNet [47,48], as well as a baseline that calculates the performance gap between the input and ground truth images. PredNet [47] is made up of 30 layers, including 12 convolutional layers, five downsampling layers, five upsampling layers, four LSTM layers, and one flatten, reshape, input, and output layers. The convolutional, downsampling, and upsampling layers all use ReLU activation with a dropout of 0.1 and batch normalization, while the LSTM layer has a 0.2 dropout. ConvLSTM has six layers with a configuration of [48, 36, 24, 24, 12, and 4], using filter size (3×3) , strides (1×1) , and zero padding. Each layer, except the last one, has ReLU activation, 0.1 dropout, and a batch normalization layer, while the last layer uses softmax activation. For the Autoencoder, we followed the literature [14] and used the configuration [512, 384, 256, and 128], with ReLU activation for each layer. We changed the last layer's sigmoid activation to softmax activation and the loss function to categorical

cross-entropy for fair comparison. For the UNet [47,48], we used the same architecture as the PredNet without the recurrent layer between the convolutional encoder and decoder, and we trained the model with the same hyper-parameters as PredNet.

Algorithm 1: Training process of RHRNet

1. **Input:** Traffic congestion map input: $\{X_{i-p:i}\} | i = p + 1, p + 2, \dots, n - 1$,
Sequence of past image: p ,
Congestion prediction horizon: k
 2. **Required:** learning rate: α
 3. **Output:** RHRNet model \mathcal{F}
 4. $\mathbf{S} \leftarrow \emptyset$
 5. **for** all available frame instances $i, (p + 1 \leq i \leq (n - 1) - k)$ **do**
 6. $X_{i-p:i} = \text{concat}\{x_{i-p}, x_{i-(p-1)}, \dots, x_{i-1}, x_{i-0}\} = X_i$
 7. $Y_i^k = \{x_{i+k}\} = Y_i$
 8. put a training instance (X_i, Y_i) into \mathbf{S}
 9. **end**
 10. initialized a model parameter θ
 11. **while** the stopping criteria is not satisfied **do**
 12. sample random mini batch of training instance S_b from \mathbf{S}
 13. find θ by minimizing the objective $\mathcal{L}(\theta)$ with S_b
 14. Output the learned RHRNet model \mathcal{F}
-

5. Result and Analysis

5.1. Model Implementation

In our previous experiment [47], we tested the PredNet model using input sequences of nine, 11, 12, and 13 traffic images to predict congestion levels for 10-, 30-, and 60-min horizons. We found that using an input sequence of 12 images yielded the best results on the validation dataset. Therefore, in this work, we fixed the past sequence to be 12 images for all experiments. To prepare the input for the network, we concatenated the 12 images at the channel dimension to create a $192 \times 448 \times 36$ input tensor, which was then passed through the stem to reduce the resolution to $48 \times 112 \times 32$. The feature maps generated by the Tiny HRNet backbone had sizes of $48 \times 112 \times 32$, $24 \times 56 \times 64$, $12 \times 28 \times 128$, and $6 \times 14 \times 256$ at resolution levels L_1, L_2, L_3 , and L_4 , respectively. The output of the decoder's convLSTM layer had the same resolution as the backbone's output, and they were fused to generate a feature map size to $48 \times 112 \times 480$. A 1×1 convolutional layer was used to reduce the number of feature maps to $48 \times 112 \times 256$, which was then concatenated with a low-level feature map from the stem to yield feature maps of $48 \times 112 \times 288$. Finally, two convolution transpose layers of 2×2 kernel size and 2×2 were added to restore the feature map to its original size with four channels, i.e., $192 \times 448 \times 4$.

5.2. Performance Comparison

5.2.1. Performance Comparison on Training Dataset

In Table 1, our proposed model's performance is compared to the other-state-of-the-art models (such as PredNet, UNet, ConvLSTM, and Autoencoder) on a training dataset. The comparison is made in terms of precision, recall, and accuracy at prediction horizons of 10, 30, and 60 min, respectively. Due to a class imbalance in background and congestion level classes and variations in the number of pixels for each road segment, evaluating the model's performance based on entire pixels or the road segments can lead to inaccurate results. Therefore, in this study, we opted to evaluate the model's performance based on a single value from each road segment. On the training dataset, for prediction horizons, beyond 1 h, the performance of the model decreases drastically; therefore, in this study, we choose to present our results up to prediction horizons of 1 h.

Table 1. Performance comparison on training dataset for the prediction horizons of 10, 30, and 60 min. The best result is marked in bold.

P.H	Precision					Recall					Accuracy				
	R.H	P.N	U.N	C.L	A.E	R.H	P.N	U.N	C.L	A.E	R.H	P.N	U.N	C.L	A.E
10	0.898	0.872	0.844	0.836	0.766	0.887	0.857	0.826	0.812	0.726	0.928	0.884	0.862	0.857	0.772
30	0.866	0.849	0.822	0.821	0.719	0.871	0.853	0.797	0.810	0.712	0.886	0.860	0.847	0.839	0.754
60	0.856	0.847	0.804	0.746	0.730	0.853	0.846	0.701	0.731	0.705	0.867	0.842	0.821	0.757	0.749

P.H: Prediction Horizons; R.H: Recurrent HRNet; P.N: PredNet; C.L: ConvLSTM; A.E: Autoencoder.

Our proposed HRNet outperforms all other models for all three prediction horizons and all three performance metrics. Specifically, RHRNet showed an improvement of approximately 3% in precision for 10- and 30-min prediction horizons and around 1% for the 60-min prediction horizon, compared to the next-best model, PredNet. Moreover, the proposed model showed a gain of around 3% in recall for the 10-min prediction horizon and around a 1–2% gain for the 30- and 60-min prediction horizon, compared to PredNet. In term of accuracy, the proposed model demonstrated a 4% improvement for the 10-min prediction horizon and was approximately 2% better for the 30- and 60-min prediction horizons compared to the next-best model, PredNet. Based on performance metrics from Table 1, the models are ranked as follows, from high to low: proposed RHRNet, PredNet, UNet, ConvLSTM, Autoencoder.

5.2.2. Performance Comparison on Testing Dataset

Table 2 presents the hourly road-wise accuracy performance of our proposed model and four other state-of-the-art neural networks on the testing dataset from December 3 to December 10, between 08:00 to 11:00. Our proposed model outperforms other state-of-the-art models by a large margin. Specifically, for the 10-min prediction horizon, our proposed RHRNet demonstrates an average gain of approximately 4% compared to the next best model, PredNet and an average gain of 6–15% compared to other methods. For 30-min prediction horizons, our proposed model achieves an increment in accuracy of 2–13% compared to other compared models. Similarly, for the 60-min prediction horizons, our proposed model attains an average improvement of 1.5–15% compared to other state-of-the-art models.

Figure 4 presents a comparison of road-wise prediction accuracy on 3 December 2019 between 08:00 to 12:00, at 5-min intervals. The proposed RHRNet model is compared to other state-of-the-art models such as PredNet, UNet, ConvLSTM, and Autoencoder, and with the prediction baseline. Figure 4a shows the road-wise accuracy comparison at a prediction horizon of 10 min. From the line chart, we can see that the proposed RHRNet achieves higher accuracy compared to all other models for almost all the testing timeline, reaching the highest value of 95% at 11:15, followed by the PredNet. The UNet and ConvLSTM have much lower performance than RHRNet but have similar performance to one another, while the Autoencoder performance is closer to the baseline. Similarly, Figure 4b shows the road-wise accuracy comparison at a prediction horizon of 30 min. This chart shows similar trends as in Figure 4a, with RHRNet achieving better accuracy followed by PredNet. The UNet and ConvLSTM show similar performance but are lower than the proposed model, while Autoencoder performance is much closer to the baseline.

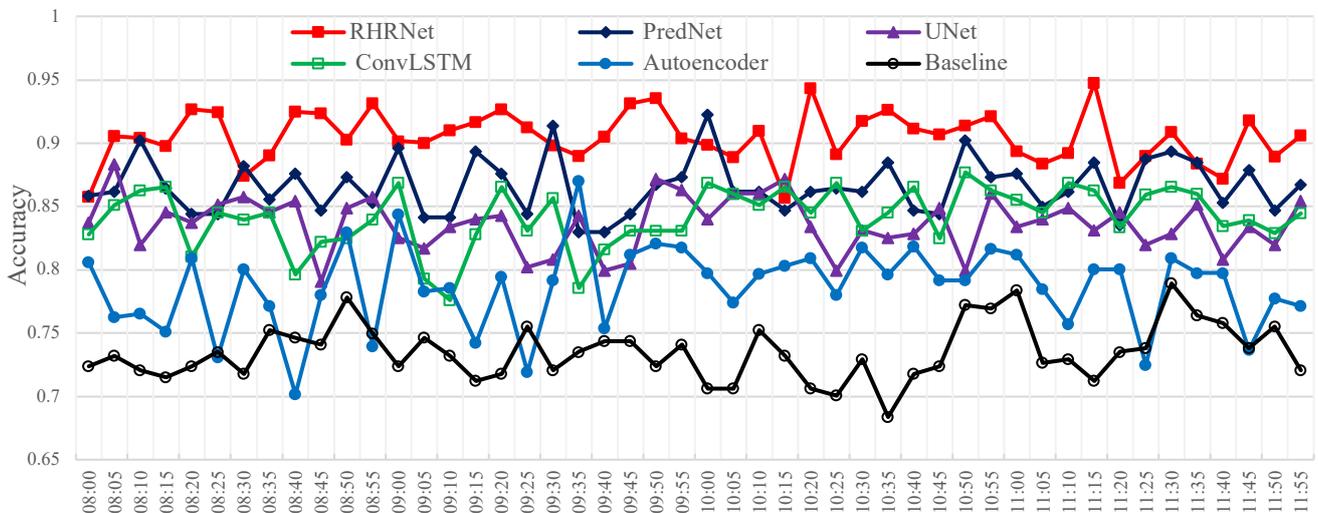
Table 2 and Figure 4 demonstrate that the RHRNet performs better in terms of accuracy than other state-of-the-art traffic congestion prediction models. However, it is uncertain whether the trained model can represent other congestion levels better. To address this, Figure 5 presents a comparison of precision and recall metrics on the testing dataset with respect to PredNet, UNet, ConvLSTM, and Auto-encoder. Quartiles Q1, Q2, and Q3 are shown in the Figure 5, with the mean value indicated by a cross, and the range of the data distribution shown by lines above and below the quartiles. Figure 5a shows the precision and recall comparison for 10-min prediction horizons. All three congestion levels, namely

Jam, Slow, and Free, demonstrate significantly higher performance in both precision and recall metrics, followed by PredNet. The prediction distribution range for the proposed RHRNet is smaller compared to all other models, with the autoencoder having the largest range. Similar results can be seen for 30-min and 60-min prediction horizons in Figure 5b,c, respectively. In both cases, the proposed RHRNet shows higher precision and recall results, followed by the next best model, PredNet.

Table 2. Road-wise accuracy comparison on testing dataset for the prediction horizons of 10, 30, and 60 min. The best result is marked in bold.

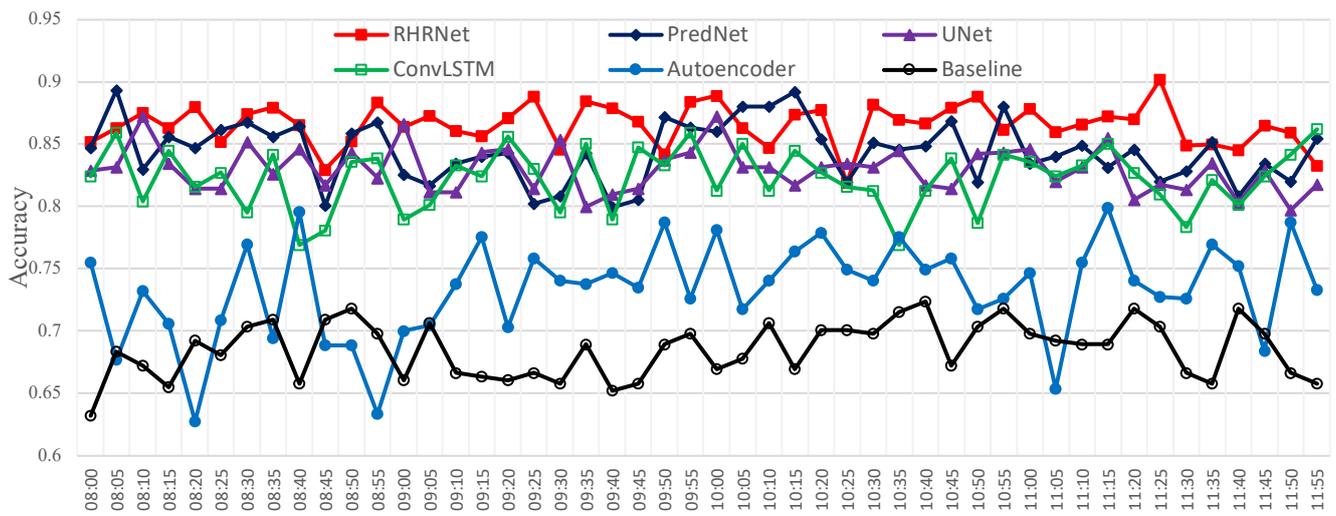
Date	P.H Time	10 Minutes					30 Minutes					60 Minutes				
		R.H	P.N	U.N	C.L	A.E	R.H	P.N	U.N	C.L	A.E	R.H	P.N	U.N	C.L	A.E
12-03	08:00	0.8784	0.8735	0.8336	0.8359	0.7709	0.8653	0.8541	0.8463	0.8296	0.7063	0.8522	0.8293	0.7770	0.7302	0.6978
	09:00	0.9142	0.8726	0.8496	0.8236	0.7845	0.8220	0.8294	0.8180	0.8357	0.7292	0.8150	0.8293	0.7930	0.7118	0.7263
	10:00	0.8732	0.8793	0.8231	0.8554	0.7936	0.8688	0.8500	0.8210	0.8286	0.7514	0.8529	0.8489	0.8222	0.7341	0.6954
	11:00	0.9261	0.8784	0.8344	0.8490	0.7748	0.8439	0.8347	0.8120	0.8361	0.7164	0.8475	0.8375	0.8160	0.7447	0.6882
12-04	08:00	0.8888	0.8716	0.8279	0.8264	0.7605	0.8550	0.8340	0.8369	0.8250	0.7232	0.8386	0.8407	0.8281	0.7638	0.7041
	09:00	0.8952	0.8743	0.8430	0.8417	0.7466	0.8667	0.8243	0.8284	0.8212	0.7338	0.8863	0.8400	0.8083	0.7425	0.6935
	10:00	0.9176	0.8639	0.8120	0.8412	0.7699	0.9070	0.8437	0.8265	0.8296	0.7227	0.8651	0.8400	0.8225	0.7512	0.6908
	11:00	0.9112	0.8716	0.8763	0.8427	0.7630	0.8321	0.8383	0.8441	0.8282	0.7152	0.8435	0.8317	0.7881	0.7408	0.7162
12-05	08:00	0.9094	0.8714	0.8461	0.8320	0.6998	0.8263	0.8301	0.8246	0.8344	0.7355	0.8384	0.8194	0.8474	0.7239	0.6908
	09:00	0.9083	0.8473	0.8339	0.8536	0.7942	0.8500	0.8262	0.8364	0.8265	0.7329	0.8464	0.8358	0.7898	0.7348	0.6882
	10:00	0.9572	0.8677	0.8432	0.8398	0.7884	0.8669	0.8420	0.8442	0.8217	0.7203	0.8232	0.8230	0.7943	0.7415	0.6949
	11:00	0.8909	0.8645	0.8023	0.8349	0.7527	0.8534	0.8335	0.8213	0.8187	0.7237	0.8430	0.8354	0.8089	0.7215	0.7068
12-06	08:00	0.8992	0.8576	0.8164	0.8238	0.7976	0.8413	0.8262	0.8173	0.8105	0.7208	0.8232	0.8335	0.8207	0.7430	0.6995
	09:00	0.8863	0.8662	0.8713	0.8373	0.7692	0.8375	0.8369	0.8281	0.8190	0.7338	0.8277	0.8402	0.8029	0.7075	0.6459
	10:00	0.8863	0.8762	0.8325	0.8373	0.7104	0.8575	0.8371	0.8237	0.8311	0.7324	0.8597	0.8303	0.8341	0.7568	0.6560
	11:00	0.9197	0.8788	0.8180	0.8419	0.6973	0.8671	0.8566	0.8432	0.8344	0.6886	0.8442	0.8358	0.8271	0.7447	0.7089
12-07	08:00	0.9095	0.8703	0.8279	0.8456	0.7213	0.8608	0.8226	0.8231	0.8190	0.6664	0.8536	0.8286	0.7982	0.7300	0.6989
	09:00	0.8863	0.8628	0.8535	0.8354	0.7693	0.8712	0.8313	0.8387	0.8335	0.6635	0.8398	0.8337	0.7782	0.7275	0.7075
	10:00	0.8774	0.8757	0.8487	0.8284	0.6964	0.8725	0.8386	0.8461	0.8361	0.7309	0.8688	0.8378	0.7805	0.7522	0.6838
	11:00	0.9123	0.8885	0.8313	0.8335	0.8078	0.8975	0.8282	0.8042	0.8139	0.7157	0.8532	0.8390	0.7865	0.7290	0.6839
12-08	08:00	0.9226	0.8631	0.8496	0.8320	0.7830	0.8616	0.8347	0.8352	0.8185	0.7329	0.8411	0.8356	0.8088	0.7063	0.7065
	09:00	0.9376	0.8521	0.8273	0.8300	0.7627	0.8866	0.8347	0.8322	0.8185	0.7329	0.8509	0.8457	0.8198	0.7389	0.7048
	10:00	0.8842	0.8587	0.8502	0.8698	0.7764	0.8437	0.8323	0.8295	0.8127	0.7200	0.8292	0.8346	0.7846	0.7135	0.6986
	11:00	0.8637	0.8535	0.8673	0.8608	0.7473	0.8375	0.8301	0.8296	0.8306	0.7167	0.8225	0.8421	0.8125	0.7140	0.7188
12-09	08:00	0.8751	0.8616	0.8469	0.8260	0.7022	0.8699	0.8484	0.8388	0.8014	0.7278	0.8553	0.8371	0.8241	0.7539	0.6978
	09:00	0.8903	0.8708	0.8399	0.8341	0.7546	0.8464	0.8279	0.8146	0.8397	0.7312	0.8339	0.8404	0.7763	0.7471	0.7162
	10:00	0.8999	0.8711	0.8279	0.8494	0.7300	0.8555	0.8221	0.8234	0.8006	0.7442	0.8612	0.8346	0.8390	0.7331	0.7101
	11:00	0.8842	0.8793	0.8543	0.8242	0.7827	0.8661	0.8303	0.8259	0.8282	0.7220	0.8523	0.8349	0.8089	0.7394	0.6989
12-10	08:00	0.9176	0.8676	0.8541	0.8441	0.8029	0.8574	0.8547	0.8466	0.8270	0.7396	0.8442	0.8385	0.8215	0.7065	0.7152
	09:00	0.8991	0.8560	0.8295	0.8313	0.7768	0.8322	0.8344	0.8417	0.8352	0.7150	0.8218	0.8385	0.7930	0.7176	0.6915
	10:00	0.8958	0.8588	0.8332	0.8455	0.7587	0.8402	0.8422	0.8397	0.8423	0.7478	0.8412	0.8363	0.8125	0.7256	0.6903
	11:00	0.9012	0.8522	0.8480	0.8675	0.7700	0.8784	0.8378	0.8356	0.8405	0.7338	0.8651	0.8274	0.8217	0.7430	0.7060
Average		0.9004	0.8674	0.8390	0.8398	0.7599	0.8572	0.8359	0.8304	0.8259	0.7227	0.8448	0.8355	0.8075	0.7335	0.6979

P.H: Prediction Horizons; R.H: Recurrent HRNet; P.N: PredNet; C.L: ConvLSTM; A.E: Autoencoder.



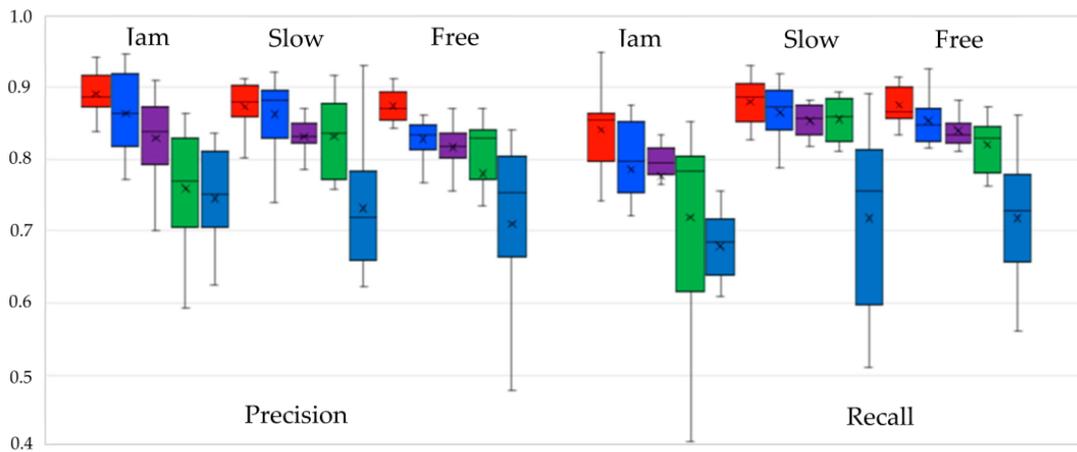
(a)

Figure 4. Cont.

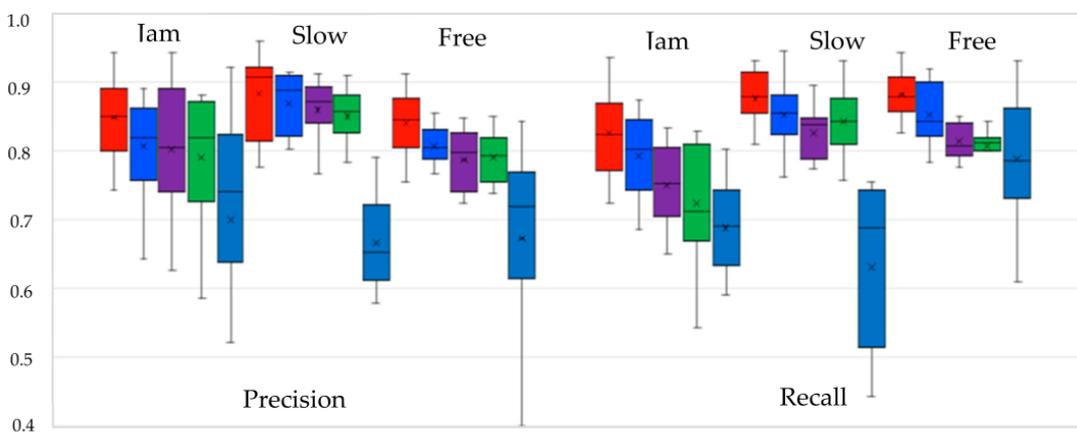


(b)

Figure 4. Road-wise accuracy for traffic congestion prediction on 3 December 2019. (a) Prediction accuracy for prediction horizon of 10 min. (b) Prediction accuracy for prediction horizon of 30 min.



(a)



(b)

Figure 5. Cont.

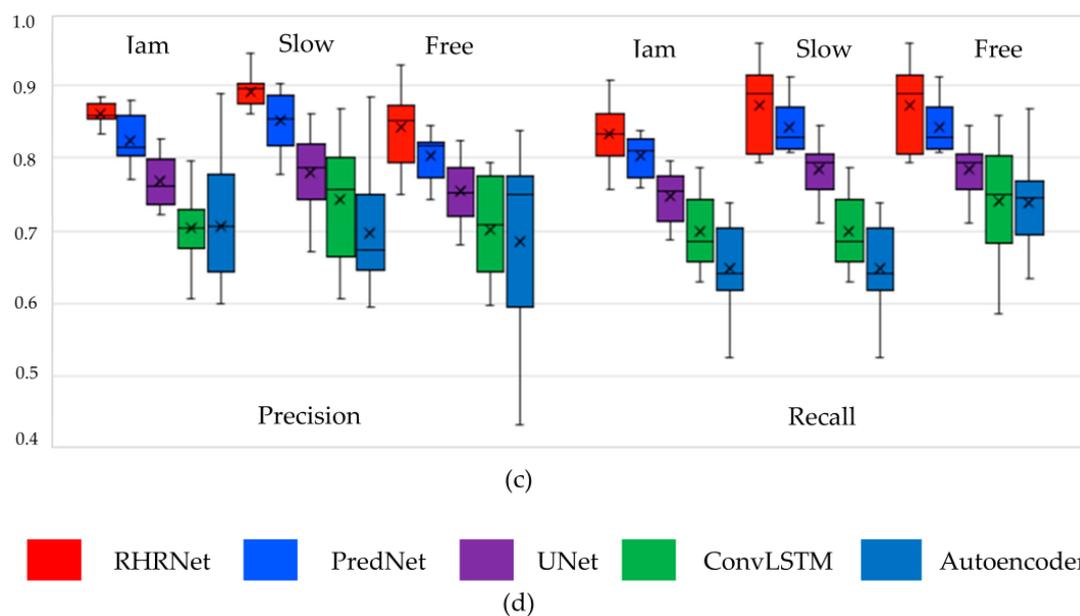


Figure 5. Road-wise recall and precision comparison for traffic congestion prediction on 3 December 2019. (a) Prediction precision and recall for prediction horizon of 10 min. (b) Prediction precision and recall for prediction horizon of 30 min. (c) Prediction precision and recall for prediction horizon of 60 min. (d) Color code for traffic congestion prediction models.

6. Conclusions

Traffic congestion is a critical issue that affects various areas, including transportation mobility, physical and mental health, economic growth, and the environment. Therefore, predicting traffic congestion is an important area of research that has the potential to alleviate congestion-related issues. Traffic authorities can utilize traffic congestion predictions to improve transportation efficiency and safety by providing users with up-to-date information on roads and adjusting road infrastructures to redirect traffic and prevent congestion. Similarly, commuters, logistic companies, and emergency response teams can leverage traffic forecasts and road conditions to schedule their routes through non-congested regions.

In this study, we provide a comprehensive review of various techniques for predicting traffic congestion, including statistical models, machine learning, and hybrid approaches. Additionally, we propose a deep neural network architecture, RHRNet, which uses image data from the TOPIS website to predict city-wide traffic congestion. The proposed RHRNet model utilizes a tiny-HRNet backbone that leverages a multi-scale feature extraction technique to learn better spatial feature representations, and a ConvLSTM-based decoder that learns time-series feature representation and aggregates all the multi-scale feature maps to obtain accurate prediction results. Moreover, we utilize an inexpensive data collection method by capturing images from the open-source TOPIS website for the central Seoul region. We conducted experiments on several state-of-the-art (SOTA) deep learning architectures, including PredNet, UNet, ConvLSTM, and Autoencoder, and compared their performance with our proposed RHRNet. Our investigation shows that the RHRNet model outperforms the others in terms of accuracy, precision, and recall for all three prediction horizons (10, 30, and 60 min) compared to all others. Specifically, RHRNet achieved an approximately 4%, 2.5%, and 1.5% improvement in accuracy over the next-best model PredNet, for the 10-, 30-, and 60-min prediction horizons, respectively. Similarly, RHRNet outperforms other SOTA models by a large margin in terms of precision and recall for all three congestion categories (Jam, Slow, and Free) across all prediction horizons. The superior results obtained by the proposed RHRNet demonstrate that by incorporating multi-scale information for feature extraction and employing recurrent neural network-based decoder the model can effectively learn from historical traffic patterns and make accurate predictions.

Although the findings of this study are promising, there is still room for enhancing the computational efficiency of the model. In the current dataset, a significant portion comprises background information, and computational resources are being wasted on learning them. In the future, we intend to eliminate the background information from the dataset to improve the efficiency of the model. Furthermore, incorporating external factors such as weather information could enhance the prediction accuracy of the model. Additionally, incorporating a Vision Transformer as a backbone, given its recent success in computer vision tasks, may potentially enhance the performance of the model in future studies.

Author Contributions: Conceptualization, S.R. and H.K.; methodology, S.R.; software, S.R., Y.-C.K. and N.R.; validation, S.R., S.B. and H.K.; formal analysis, S.R.; investigation, S.R.; resources, H.K.; data curation, S.R.; writing—original draft preparation, S.R., Y.-C.K. and N.R.; writing—review and editing, S.R., Y.-C.K., S.B., N.R. and H.K.; visualization, S.R.; supervision, H.K.; project administration, H.K.; funding acquisition, H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All the data used for the traffic congestion analysis was capture from the open-source traffic web service. They provide accurate traffic information about the city. The readers can access the data by capturing an image from the web service. The links to these web service is mentioned in references [17–20].

Acknowledgments: This work was supported by Incheon National University Research Grant in 2018.

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

References

1. Onyeneke, C.; Eguzouwa, C.; Mutabazi, C. Modeling the Effects of Traffic Congestion on Economic Activities—Accidents, Fatalities and Casualties. *Biomed. Stat. Inform.* **2018**, *3*, 7–14. [[CrossRef](#)]
2. Wang, C.; Quddus, M.A.; Ison, S.G. Impact of traffic congestion on road accidents: A spatial analysis of the M25 motorway in England. *Accid. Anal. Prev.* **2009**, *41*, 798–808. [[CrossRef](#)]
3. Hao, P.; Wang, C.; Wu, G.; Boriboonsomsin, K.; Barth, M. Evaluating the environmental impact of traffic congestion based on sparse mobile crowd-sourced data. In Proceedings of the Fifth IEEE Conference on Technologies for Sustainability (SusTech 2017), Phoenix, AZ, USA, 12–14 November 2017; pp. 1–6. [[CrossRef](#)]
4. Ye, S. Research on Urban Road Traffic Congestion Charging Based on Sustainable Development. *Phys. Procedia* **2012**, *24*, 1567–1572. [[CrossRef](#)]
5. Wang, P.; Zhang, R.; Sun, S.; Gao, M.; Zheng, B.; Zhang, D.; Zhang, Y.; Carmichael, G.R.; Zhang, H. Aggravated air pollution and health burden due to traffic congestion in urban China. *Atmos. Meas. Tech.* **2023**, *23*, 2983–2996. [[CrossRef](#)]
6. World Health Organization. *Global Status Report on Road Safety 2018*; World Health Organization: Geneva, Switzerland, 2018.
7. Olayode, I.O.; Tartibu, L.K.; Okwu, M.O. Prediction and modeling of traffic flow of human-driven vehicles at a signalized road intersection using artificial neural network model: A South African road transportation system scenario. *Transp. Eng.* **2021**, *6*, 100095. [[CrossRef](#)]
8. Xiao, J.; Xiao, Z.; Wang, D.; Bai, J.; Havyarimana, V.; Zeng, F. Short-term traffic volume prediction by ensemble learning in concept drifting environments. *Knowl.-Based Syst.* **2019**, *164*, 213–225. [[CrossRef](#)]
9. Rempe, F.; Huber, G.; Bogenberger, K. Spatio-temporal congestion patterns in urban traffic networks. In Proceedings of the International Symposium on Enhancing Highway Performance (ISEHP), Berlin, Germany, 14–16 June 2016; pp. 513–524. [[CrossRef](#)]
10. Xu, L.; Yue, Y.; Li, Q. Identifying Urban Traffic Congestion Pattern from Historical Floating Car Data. *Procedia Soc. Behav. Sci.* **2013**, *96*, 2084–2095. [[CrossRef](#)]
11. Park, J.; Li, D.; Murphey, Y.L.; Kristinsson, J.; McGee, R.; Kuang, M.; Phillips, T. Real time vehicle speed prediction using a neural network traffic model. In Proceedings of the 2011 International Joint Conference on Neural Networks, IJCNN 2011, San Jose, CA, USA, 31 July–5 August 2011; pp. 2991–2996. [[CrossRef](#)]
12. Chang, H.; Lee, Y.; Yoon, B.; Baek, S. Dynamic near-term traffic flow prediction: System-oriented approach based on past experiences. *IET Intell. Transp. Syst.* **2012**, *6*, 292. [[CrossRef](#)]

13. Ma, X.; Dai, Z.; He, Z.; Ma, J.; Wang, Y.; Wang, Y. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors* **2017**, *17*, 818. [CrossRef]
14. Zhang, S.; Yao, Y.; Hu, J.; Zhao, Y.; Li, S.; Hu, J. Deep Autoencoder Neural Networks for Short-Term Traffic Congestion Prediction of Transportation Networks. *Sensors* **2019**, *19*, 2229. [CrossRef]
15. Akhtar, M.; Moridpour, S. A Review of Traffic Congestion Prediction Using Artificial Intelligence. *J. Adv. Transp.* **2021**, *2021*, 8878011. [CrossRef]
16. Zadobrischi, E.; Cosovanu, L.-M.; Dimian, M. Traffic Flow Density Model and Dynamic Traffic Congestion Model Simulation Based on Practice Case with Vehicle Network and System Traffic Intelligent Communication. *Symmetry* **2020**, *12*, 1172. [CrossRef]
17. Google Maps. Available online: <https://www.google.com/maps/place/Delhi,+India/@28.6471948,76.9531797,11z/data=!3m1!4e1!4m5!3m4!1s0x390cfd5b347eb62d:0x37205b715389640!8m2!3d28.7040592!4d77.1024902> (accessed on 4 May 2019).
18. Bing. Bing Maps. Available online: <https://www.bing.com/maps/traffic> (accessed on 5 May 2019).
19. Seoul TOPIS. Seoul Transport Operation & Information Service Center. Available online: <https://topis.seoul.go.kr/prdc/openPrdcMap.do> (accessed on 5 May 2019).
20. Baidu. Baidu Maps. Available online: <https://map.baidu.com/13036895.494262943,4748316.384998233,11.52z/maplayer%3Dtrafficrealtime> (accessed on 10 May 2019).
21. Graves, A.; Mohamed, A.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 6645–6649. [CrossRef]
22. Fu, R.; Zhang, Z.; Li, L. Using LSTM and GRU neural network methods for traffic flow prediction. In Proceedings of the 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC 2016), Wuhan, China, 11–13 November 2016; pp. 324–328. [CrossRef]
23. Wei, W.; Wu, H.; Ma, H. An AutoEncoder and LSTM-Based Traffic Flow Prediction Method. *Sensors* **2019**, *19*, 2946. [CrossRef]
24. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3349–3364. [CrossRef]
25. Smith, B.L.; Demetsky, M.J. Traffic Flow Forecasting: Comparison of Modeling Approaches. *J. Transp. Eng.* **1997**, *123*, 261–266. [CrossRef]
26. Tan, M.-C.; Wong, S.C.; Xu, J.-M.; Guan, Z.-R.; Zhang, P. An Aggregation Approach to Short-Term Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 60–69. [CrossRef]
27. Kumar, S.V.; Vanajakshi, L. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *Eur. Transp. Res. Rev.* **2015**, *7*, 21. [CrossRef]
28. Guo, J.; Huang, W.; Williams, B.M. Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 50–64. [CrossRef]
29. Castro-Neto, M.; Jeong, Y.; Jeong, M.K.; Han, L.D. AADT prediction using support vector regression with data-dependent parameters. *Expert Syst. Appl.* **2009**, *36*, 2979–2986. [CrossRef]
30. Su, H.; Zhang, L.; Yu, S. Short-term traffic flow prediction based on incremental support vector regression. In Proceedings of the Third International Conference on Natural Computation (ICNC 2007), Haikou, China, 24–27 August 2007; pp. 640–645. [CrossRef]
31. Wang, X.; Zhang, N.; Zhang, Y.; Shi, Z. Forecasting of Short-Term Metro Ridership with Support Vector Machine Online Model. *J. Adv. Transp.* **2018**, *2018*, 3189238. [CrossRef]
32. Sun, Y.; Leng, B.; Guan, W. A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system. *Neurocomputing* **2015**, *166*, 109–121. [CrossRef]
33. Sun, S.; Zhang, C.; Yu, G. A Bayesian Network Approach to Traffic Flow Forecasting. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 124–132. [CrossRef]
34. Hoong, P.K.; Chien, O.K.; Tan, I.; Ting, C.-Y. Road traffic prediction using Bayesian networks. In Proceedings of the IET International Conference on Wireless Communications and Applications (ICWCA 2012), Kuala Lumpur, Malaysia, 8–10 October 2012; pp. 1–5. [CrossRef]
35. Sharma, B.; Kumar, S.; Tiwari, P.; Yadav, P.; Nezhurina, M.I. ANN based short-term traffic flow forecasting in undivided two lane highway. *J. Big Data* **2018**, *5*, 48. [CrossRef]
36. Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 187–197. [CrossRef]
37. Tian, Y.; Zhang, K.; Li, J.; Lin, X.; Yang, B. LSTM-based traffic flow prediction with missing data. *Neurocomputing* **2018**, *318*, 297–305. [CrossRef]
38. Chen, Y.-Y.; Lv, Y.; Li, Z.; Wang, F.-Y. Long short-term memory model for traffic congestion prediction with online open data. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC 2016), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 132–137.
39. Afrin, T.; Yodo, N. A Long Short-Term Memory-based correlated traffic data prediction framework. *Knowl.-Based Syst.* **2022**, *237*, 107755. [CrossRef]
40. Luo, X.; Li, D.; Yang, Y.; Zhang, S. Spatiotemporal Traffic Flow Prediction with KNN and LSTM. *J. Adv. Transp.* **2019**, *2019*, 4145353. [CrossRef]

41. Jiang, P.; Liu, Z.; Zhang, L.; Wang, J. Advanced traffic congestion early warning system based on traffic flow forecasting and extenics evaluation. *Appl. Soft Comput.* **2022**, *118*, 108544. [[CrossRef](#)]
42. Lv, Z.; Xu, J.; Zheng, K.; Yin, H.; Zhao, P.; Zhou, X. LC-RNN: A deep learning model for traffic speed prediction. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 3470–3476. [[CrossRef](#)]
43. Yu, H.; Wu, Z.; Wang, S.; Wang, Y.; Ma, X. Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks. *Sensors* **2017**, *17*, 1501. [[CrossRef](#)]
44. Chen, M.; Yu, X.; Liu, Y. PCNN: Deep Convolutional Networks for Short-Term Traffic Congestion Prediction. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 3550–3559. [[CrossRef](#)]
45. Bao, Y.; Huang, J.; Shen, Q.; Cao, Y.; Ding, W.; Shi, Z.; Shi, Q. Spatial–Temporal Complex Graph Convolution Network for Traffic Flow Prediction. *Eng. Appl. Artif. Intell.* **2023**, *121*, 106044. [[CrossRef](#)]
46. Zhang, W.; Yan, S.; Li, J. TCP-BAST: A novel approach to traffic congestion prediction with bilateral alternation on spatiality and temporality. *Inf. Sci.* **2022**, *608*, 718–733. [[CrossRef](#)]
47. Ranjan, N.; Bhandari, S.; Zhao, H.P.; Kim, H.; Khan, P. City-Wide Traffic Congestion Prediction Based on CNN, LSTM and Transpose CNN. *IEEE Access* **2020**, *8*, 81606–81620. [[CrossRef](#)]
48. Ranjan, N.; Bhandari, S.; Khan, P.; Hong, Y.-S.; Kim, H. Large-Scale Road Network Congestion Pattern Analysis and Prediction Using Deep Convolutional Autoencoder. *Sustainability* **2021**, *13*, 5108. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.