Traffic Network Flow Prediction

Project-II report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Bachelor of Technology

in

Electronics and Electrical Communication Engineering

by

Cheruvu Surya Sai Ram

(19EC39008)

Under the supervision of

Prof. Manjira Sinha

&

Prof. Aneek Adhya

Department of Electronics and Electrical Communication Engineering

Indian Institute of Technology Kharagpur

Spring Semester, 2022-23

May 02, 2023

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: May 02, 2023 (Cheruvu Surya Sai Ram)

Place: Kharagpur (19EC39008)

DEPARTMENT OF ELECTRONICS AND ELECTRICAL COMMUNICATION ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Traffic Network Flow Prediction**" submitted by **Cheruvu Surya Sai Ram (19EC39008)** to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Bachelor of Technology in Electronics and Electrical Communication Engineering is a record of bona fide work carried out by him under our supervision and guidance during Spring Semester, 2022-23.

Date: May 02, 2023
Place: Kharagpur

Prof. Manjira Sinha
&
Prof. Aneek Adhya
Department of GSSST
Indian Institute of Technology Kharagpur
Kharagpur - 721302, India

Abstract

Name of the student(s): **Cheruvu Surya Sai Ram**

Roll No: **19EC39008**

Degree for which submitted: **Bachelor of Technology**

Department: **Department of Electronics and Electrical Communication Engineering**

Thesis title: **Traffic Network Flow Prediction**

Thesis supervisor: **Prof. Manjira Sinha** & **Prof. Aneek Adhya**

Month and year of thesis submission: **May 02, 2023**

You probably don't consider the traffic forecasting system when you're driving, but it may have saved you from a gridlock. But it exists and works in the background to bring you where you need to go. Traffic prediction technology is continually developing. Typically, it is created by estimating future traffic levels based on past trends and current circumstances. A road network's traffic flow is mutually interactive and dependent on one another and so utilizing analytical techniques for traffic network flow dynamics is difficult. Hence a Convolutional LSTM model is employed in this article to estimate short-term traffic network flow such that it deals with spatio - temporal prediction. But most current traffic forecasting systems rely on the concept of the same underlying distribution for both training and testing data, which restricts their practical application. We need a robust forecasting model towards domain shifts in space and time. A multi - task learning framework with U-Net has been presented for the same to outperform baseline models by a significant margin.

Acknowledgements

First, I would like to express my deepest gratitude to my project supervisor Prof. Manjira Sinha and Prof. Aneek Adhya of G.S.Sanyal School of Telecommunication for their encouragement, perspective advice, invaluable guidance and support throughout my project work. Their boundless enthusiasm, optimism and dedication to excellence always motivated me to work forward. They provided me his outstanding ideas and knowledge from time to time that helped me in solving difficult aspects of my project. Despite of their busy work schedule, they spent an ample amount of time with me and always kept motivating me to be productive and bring out some utility work. I will cherish the experience of learning and working with them forever.

I want to express my sincere gratitude to my family and close friends at IIT Kharagpur for constant encouragement and support over the course of my work.

Last but not the least, I want to express my gratitude to my parents for their unwavering support throughout the writing of this thesis.

Contents

List of Figures

Abbreviations

CNN	**C**onvolutional **N**eural **N**etwork
RNN	**R**ecurrent **N**eural **N**etwork
DCNN	**D**eep **C**onvolutional **N**eural **N**etwork
LSTM	**L**ong **S**hort **T**erm **M**emory Network
ConvLSTM	**Conv**olutional **L**ong **S**hort **T**erm **M**emory Network
MERNN	**M**odified **E**lman **R**ecurrent **N**eural **N**etwork
GA	**G**enetic **A**lgorithm
GA-MENN	**G**enetic **A**lgorithm based MERNN
FC-LSTM	**F**ully **C**onnected **LSTM** Network
SGD	**S**tochastic **G**radient **D**escent
RMSProp	**R**oot **M**ean **S**quare **Prop**agation
Adam	**Ada**ptive **M**oment Estimation
HDF5	**H**ierarchical **D**ata **F**ormat
MSE	**M**ean **S**quare **E**rror
GPU	**G**raphics **P**rocessing **U**nit
ST-GNN	**S**patial **T**emporal **G**raph **N**eural **N**etwork
ARIMA	**A**uto **R**egressive **I**ntegrated **M**oving **A**verage
SVM	**S**upport **V**ector **M**achine

Chapter 1

Introduction

The task of traffic forecasting is crucial in the context of transportation planning. Using historical information and the state of the roads at the time, it predicts future traffic volumes. The main goal of traffic forecasting is to estimate the future demand for transportation services, such as the number of vehicles on the road, the travel time, and the mode of transportation. The demand for transportation services is affected by a variety of factors, including population growth, economic activity, land use, and technological advancements. It can be used to increase travel time reliability and decrease travel time variability, which are crucial elements influencing people's decisions regarding their mode of transportation. By offering a platform for real-time monitoring and alterations to travel routes and schedules, it can also help transportation organisations manage road conditions better. In real-time traffic management systems (RTMS) and smart cities, traffic prediction has various uses. These can be utilised for route guidance, congestion avoidance, service operations planning, and incident identification and management.

Traffic forecasting typically involves the use of historical traffic data, such as traffic counts, speed measurements, and travel time data, to develop models that can predict future traffic conditions. These models may be based on statistical techniques, machine learning algorithms, or simulation models. The choice of modeling

technique depends on the available data, the complexity of the problem, and the level of accuracy required. With the advances in machine learning and simulation technologies, the accuracy and reliability of traffic forecasting are likely to improve in the future.

Statistical techniques such as time series analysis and regression analysis are commonly used for traffic forecasting. Time series analysis involves analyzing historical traffic data to identify patterns and trends in the data, which can then be used to make predictions about future traffic conditions. Regression analysis involves modeling the relationship between traffic demand and various factors that influence traffic, such as population, employment, and land use.

Support vector machines and artificial neural networks, two examples of machine learning techniques can also be used for traffic forecasting. These algorithms are capable of learning complex relationships between traffic demand and various factors, and can make predictions based on a wide range of data sources, including traffic counts, weather data, and social media activity.

Simulation models, such as microscopic traffic simulation models and agent-based models, are used to simulate traffic flows and predict future traffic conditions based on different scenarios. These models can be used to evaluate the impact of different transportation policies and infrastructure investments on traffic demand and congestion.

Traffic prediction is very challenging, mainly affected by the following complex factors - Complex spatial dependencies and Dynamic temporal dependencies. Figure 1.1 shows that while the influence of the same location on the expected position remains constant over time, the influence of different positions on the predicted position varies. The relationship between various positions in space is highly dynamic. As shown in Figure 1.1, the observed values for the same position at different periods exhibit non-linear changes, and the traffic status at the far time step sometimes has

a higher impact on the anticipated time step than at the recent time step [16]. Typically, traffic data includes periodicity, such as period, closeness and trend. Hence we develop a rigorous model initially that deals with these complex relations in a typical traffic network - the Convolutional LSTM model.

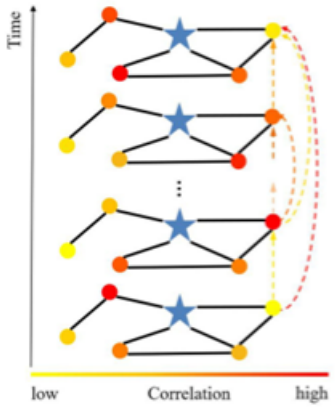

FIGURE 1.1: Complex Spatio Temporal Relations in a Traffic Network [16]

This article initially deals with short term network flow prediction as already mentioned. The dataset used is extracted from three different cities of widely distinct regions namely Berlin, Istanbul and Moscow. Later three kinds of solutions were proposed for the required purpose. The first is a very trivial and naive solution that always predicts a constant output irrespective of the history of the flow. Next a moving average model was developed that forecasts the future network flow based on knowledge of previous road networks. Last but not the least is a Convolutional LSTM model which determines the future state of the network by the inputs and past states of its local neighbors. Next we move on to predict the traffic flow conditions after a long time and to entirely unseen cities. The respective techniques for Temporal and Spatial Few Shot Transfer Learning are presented and discussed. Multi Task Learning in combination with U-Net has been used for such domain adaptation. Also the dataset that would be used would have a static high resolution map of the road network, connectivity channels as well as 8 dynamic channels.

The rest of this thesis is divided into eight chapters, each dealing with a distinct aspect of the project. Chapter 2 presents some of the previous work done in similar areas and how the work presented in this thesis is different from those. All the required theoretical and mathematical background has been presented in Chapter 3. Chapter 4 discusses the basic problem statement and the corresponding dataset used in detail and the proposed models for the future prediction of traffic flow. Chapter 5 provides the visualisation of the dataset, prediction outputs of different solutions and presents the inferences that could be drawn. Chapter 6 discusses an advanced version of the basic problem statement and the respective dataset used in detail. Some of the techniques to solve the problem are also presented. Chapter 7 presents the visualization as well as simulation results of the advanced problem statement. 8 gives a summary of the thesis and the planned work that is to be done in the future.

Chapter 2

Related Work

The research on short-term traffic flow prediction started many decades ago. All the previous work that was done on the future prediction of traffic network can be broadly divided into two categories - model driven approaches and data driven approaches. Previously the model driven approaches such as the Kalman filter model were of significance for this purpose. This is as a result of the historical traffic network flow's recurring resemblance. These methods largely presupposed that the traffic flow's dynamic fluctuations were linear. Additionally, traditional statistical learning techniques like Auto Regressive Integrated Moving Average (ARIMA) [2], wavelet transform [3] and radial basis function network [13] were extensively used for conventional traffic forecasting. The below figure shows a roadmap of the training process of an ARIMA model.

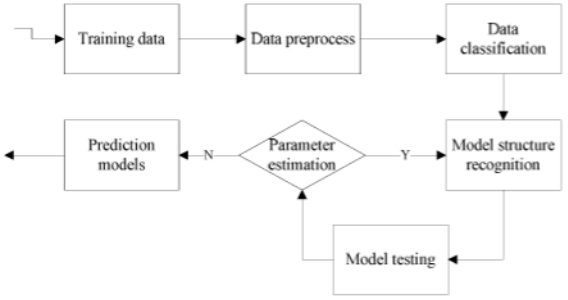

FIGURE 2.1: ARIMA Model Training Process [2]

As time progressed, the data driven approaches gained importance and at present all the future prediction is being done using such models. Some of these include Neural Networks, Radial Basis Funtion, Decision Tree Regression, Ensemble learning and many more [4]. We have done the literature review of a couple of research papers before proposing the Convolutional LSTM model. Two of them are briefly described below.

One of them discusses the DCNN prediction model. As already mentioned previously, traffic prediction consists of complex spatial and temporal dependencies. The extensive spatial-temporal features present in traffic image data samples can be captured by the DCNN model. The local correlations of traffic flow among distinct roadways can be detected by the continuous sliding of convolutional kernels in feature maps. In the interim, the pooling window is sliding on the convolutional feature map, further reserving the important traffic flow correlations, and decreasing the parameter dimension. Finally, the fully linked neurons further recreate the traffic global features using different weights and biases. A structure of the DCNN prediction model of traffic network flow is shown in Figure 2.1. So this suggested approach mainly focuses on the spatial correlation of the network flow since a CNN learns local features [17].

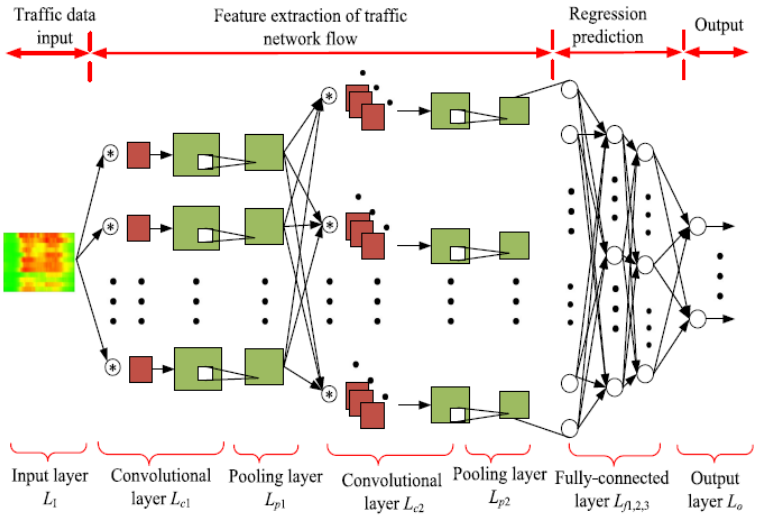

FIGURE 2.2: Deep Convolutional Neural Network Prediction Model [17]

The GA-MENN algorithm, which was covered in the second paper, was produced by combining the genetic algorithm and Elman recurrent neural network. It makes an effort to address the problem that critical phenomena may manifest in many dynamic systems later than anticipated, which can greatly impair the accuracy of analysis and results. The best answer to this issue is provided by the delays of the Elman recurrent neural network. In this study, short-term traffic flow is predicted using the evolutionary method in conjunction with a modified Elman recurrent neural network model. GA is used to optimise the MERNN's parameters. Figure 2.2 depicts the construction of this modified Elman recurrent neural network. But this has problems of its own. This approach mainly focuses on the temporal correlation of the network flow since an RNN learns timed sequences. LSTM is an advanced version of RNN [10].

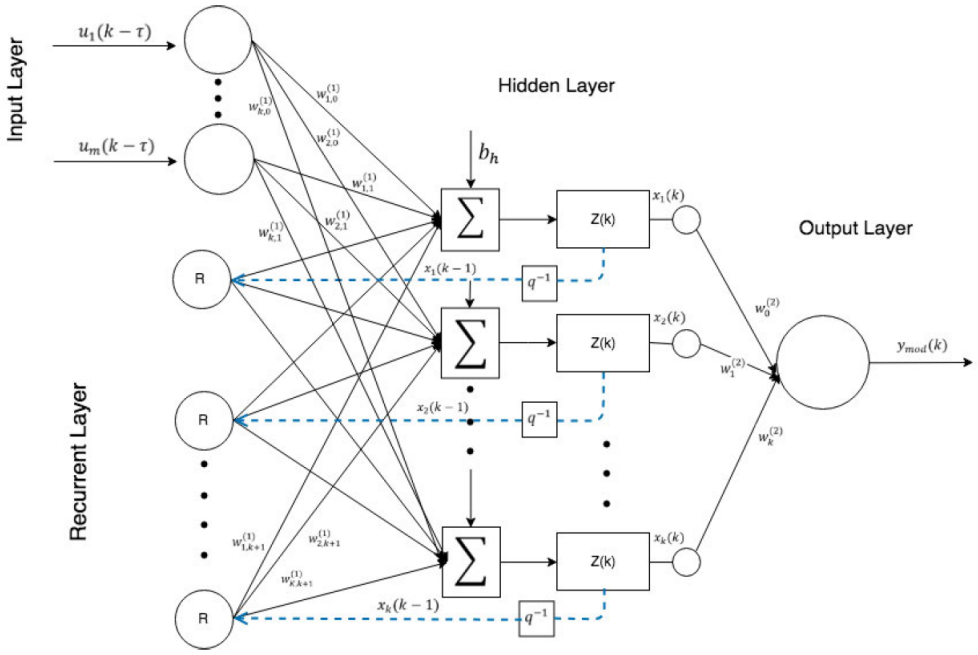

FIGURE 2.3: Modified Elman Recurrent Neural Network Structure [10]

We already saw that a general traffic network consists of both complex spatial and temporal correlations. In order to handle both of them equally well, we need to propose a model that has the functions both of a CNN and an LSTM. A Convolutional

LSTM model now comes into picture. The main focus of this report is the ConvLSTM model. We define traffic forecasting as a problem of spatio temporal sequence prediction. We extend the concept of FC-LSTM to Convolutional LSTM in order to effectively model the corresponding correlations. We can create a model for traffic forecasting by stacking ConvLSTM layers and creating an encoding-forecasting framework.

Convolutional and recurrent neural networks, on the other hand, are restricted to working on input that has an underlying Euclidean or grid-like structure, and as a result, they are unable to capture the intricate graph patterns found in transport systems like the road network [1]. Due to their capacity to recognise spatial relationships given as non-Euclidean graph structures, graph neural networks have recently demonstrated excellent performance in a range of traffic flow prediction tasks. Additionally, it has been shown that graph neural networks are better able to generalise when making predictions about as-yet-unknown cities [6] due to their increased capacity to learn properties provided by the underlying road network. The below figure shows a general pipeline of a Spatial Temporal Graph Neural Network (ST-GNN) model for traffic prediction.

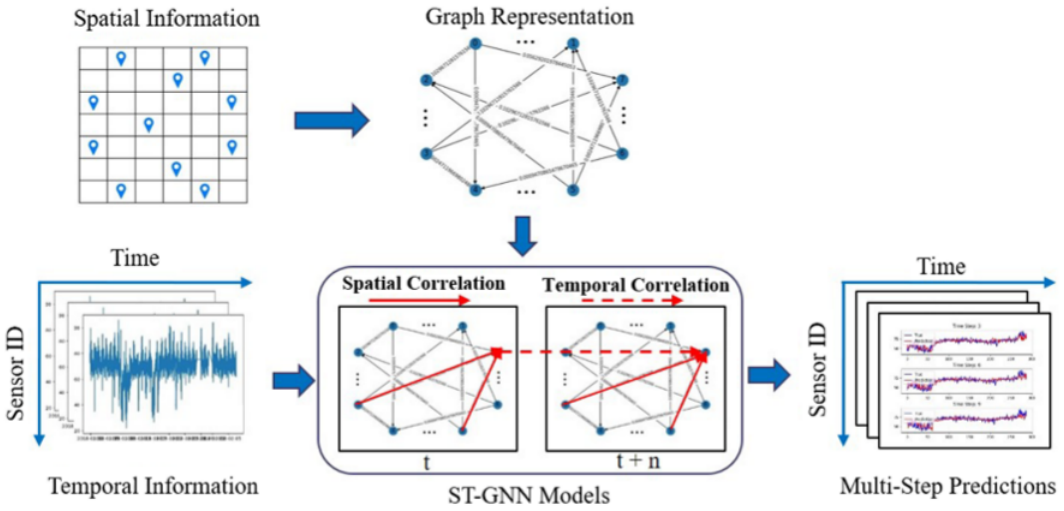

FIGURE 2.4: General pipeline of ST-GNN for traffic prediction [1]

In a variety of machine learning tasks, deep neural networks [15] have produced state-of-the-art outcomes, but they frequently assume that the training and testing sets of data come from the same distribution. This assumption might not hold true in real-world situations, necessitating domain adaptation [7]. The idea underlying domain adaptation is to acquire representations that are both performant in the target domain and domain-invariant. Three categories of deep domain adaptation techniques [11] can be distinguished.

- Techniques for domain adaptation based on discrepancies that aim to lessen the domain shift by reducing the separation between empirical source and target mapping distributions.
- Adversarial-based domain adaption techniques that use domain discriminators aim to promote domain confusion.
- Techniques to domain adaptation that rely on reconstruction of the data as a support function to guarantee feature invariance.

Multi-task learning has also been shown to be a successful method of domain adaptation [14]. In that it implicitly minimises the pairwise discrepancy between all of the tasks [18], multi-task learning can be seen as a discrepancy-based domain adaptation strategy. Multi Task Learning has been used to deal with the spatial shift in the advanced version of the problem.

Chapter 3

Theoretical Background

In this chapter, we briefly discuss the required theoretical background so that the developed solutions could be easily understood.

3.1 Convolutional Neural Networks

Convolutional neural networks, or CNNs, are a type of neural networks that excel at processing data having a grid-like design, such as an image. The convolution operation and nonlinearity are demonstrated in the equations below.

$$x_{ij}^{t} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} w_{ab} y_{(i+a)(j+b)}^{l-1}$$

$$y_{ij}^{l} = \sigma(x_{ij}^{l})$$

3.2 LSTM Network

One of the allures of RNNs is the potential for them to make connections between previous knowledge and the present task. However we have the problem of long

term dependencies. A typical RNN gives more weight to the recent information which can lead to giving less importance to any long back time input which might be significant in predicting the future predictions. LSTMs are created to prevent the long-term dependence issue. Long-term memory retention is basically their default mode of operation. The following equations and figure describe the working of an LSTM unit.

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{(t-1)} + b_{hf})$$

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{(t-1)} + b_{hi})$$

$$g_t = \tau(W_{ig}x_t + b_{ig} + W_{hg}h_{(t-1)} + b_{hg})$$

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{(t-1)} + b_{ho})$$

$$c_t = f_t \circ c_{t-1} + i_t \circ g_t$$

$$h_t = o_t \circ \tau(c_t)$$

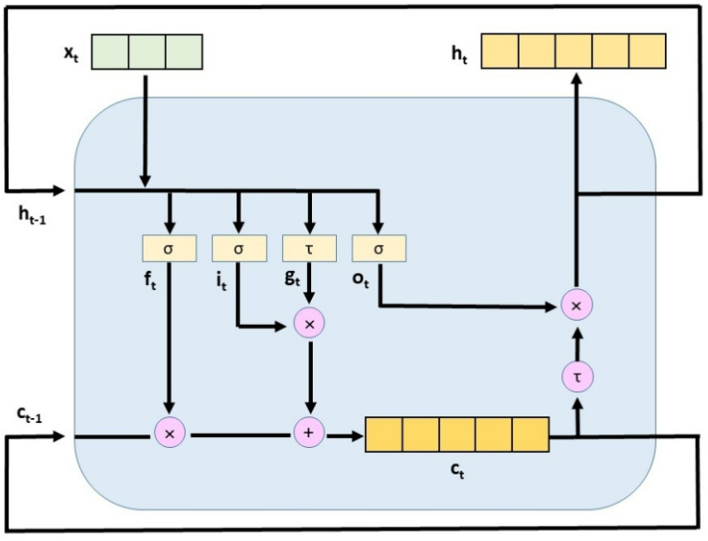

FIGURE 3.1: Working of a Long Short Term Memory Cell/unit

3.3 Convolutional LSTM

ConvLSTM, to put it simply, is an LSTM Network mixed with a CNN convolution neural network. Its input is a sequence of data from the CNN convolution neural network, which is ideally suited for images and videos, rather than just a sequence of data. ConvLSTM will capture the underlying local spatio-temporal correlations as a result of this combination. The equations and the figure below give us a clear understanding of a ConVLSTM unit.

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot C_{t-1} + b_i)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \odot C_{t-1} + b_f)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c)$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \odot C_{t-1} + b_o)$$

$$H_t = o_t \odot tanh(C_t)$$

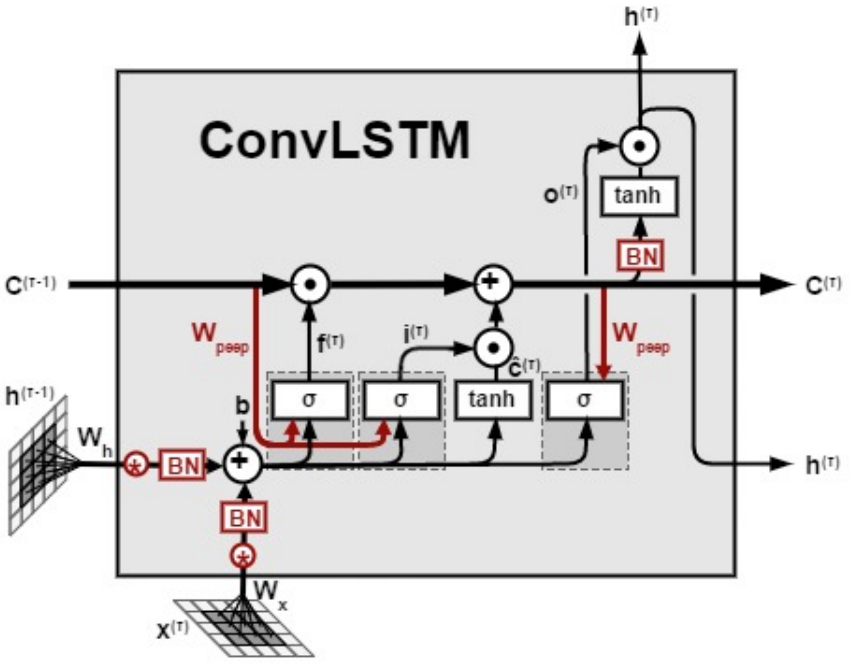

FIGURE 3.2: Working of a ConvLSTM Cell/unit

3.4 Adam Optimizer

A short-hand name for Adaptive Moment Estimation, Adam combines the worlds of SGD with momentum, as well as RMSProp. The basic equations of Adam can be summarized as follows: [5]

$$g_t = \nabla_\theta f_t(\theta_{t-1})$$

$$m_t = \beta_1.m_{t-1} + (1 - \beta_1).g_t$$

$$v_t = \beta_1.v_{t-1} + (1 - \beta_2).g_t^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_t = \theta_{t-1} - \alpha.\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

Using Adam has a number of benefits. It is simple to design, computationally effective, requires minimal memory, is suitable for issues with extremely noisy/or sparse gradients, and the hyper-parameters have an obvious interpretation and typically need some adjusting. [5] When traversing areas with small gradients, the fact that the step size of the Adam update algorithm is invariant to the gradient's magnitude is quite helpful (such as saddle points or ravines). Actually, Adam was developed to combine the advantages of Adagrad, which excels with sparse gradients, and RMSprop, which performs well in online environments. Now that we have both of them, Adam can be put to more varied uses.

3.5 U-Net Architecture

U-Net is a convolutional neural network (CNN) architecture that was originally designed for biomedical image segmentation, but has subsequently found use in a number of different industries. Its U-shaped shape, which comprises of a contracting path and an expansion path, gave rise to its name.

The input image's spatial resolution is gradually decreased as its depth is increased by a series of convolutional and max-pooling layers that make up the contracting route. This route is intended to capture the image's overall context and extract key features. The output's spatial resolution is raised while its depth is decreased by the transposed convolutional layers that make up the expanded path. This path is intended to take in the image's regional information and polish the segmentation output from the contracting path.

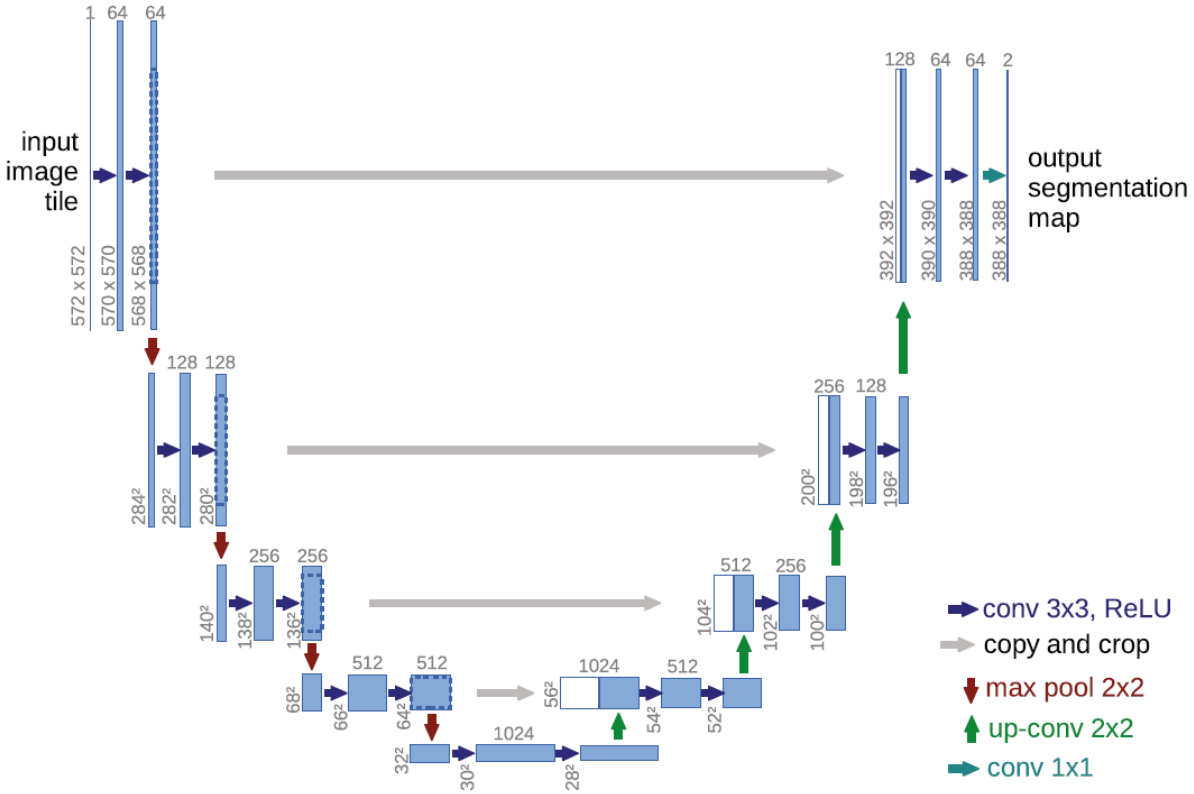

FIGURE 3.3: Sample U-Net [8]

Skip links that link the contracting and expanded channels enable the network to keep track of spatial information and enhance segmentation accuracy. The skip connections, in particular, duplicate the feature maps from the contracting path and concatenate them with the feature maps from the equivalent layer in the expansive path. This enables the network to improve the segmentation results by using both low-level and high-level features.

The U-Net design has the advantage of being able to handle less training data, which is typical for biomedical image segmentation applications. This is accomplished by adding more unpredictability to the training data and preventing overfitting using data augmentation techniques. Its effectiveness and speed are further benefits. The spatial resolution of the input image and the depth of the feature maps are both decreased when max-pooling layers are used in the contracting path, which greatly lowers the computational cost of the network. Additionally, the network can produce high-resolution segmentation results in a single forward pass due to the use of transposed convolutional layers in the expansive path.

3.6 Multi Task Learning

A single model is trained using the multi-task learning machine learning technique to carry out several tasks at once. In conventional machine learning, a different model is learned for each task, which can be time-consuming and resource-intensive. Multi-task learning reduces the complexity of the training process by learning a shared representation across tasks, thereby improving the efficiency and accuracy of the model.

The basic idea of multi-task learning is to share the parameters of a model across multiple tasks while also allowing for task-specific parameters. This is achieved by defining a joint loss function that combines the losses of all the tasks. The shared

parameters are updated based on the joint loss function, while the task-specific parameters are updated based on the loss of their respective tasks.

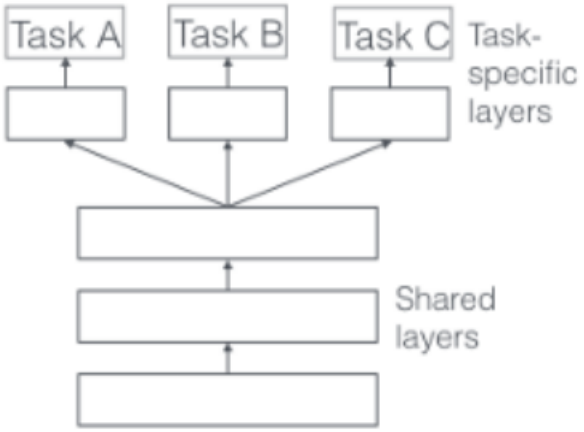

FIGURE 3.4: Parameter Sharing in Multi Task Learning [9]

By using information from related tasks, multitask learning can enhance the performance of specific tasks, which is one of its main advantages. For instance, we can leverage the shared representation that the model would learn to increase the accuracy of both tasks if we wish to predict a person's age and gender from an image. The model can find shared characteristics that are pertinent to both age and gender prediction by jointly learning both tasks.

Multi-task learning also has the advantage of reducing overfitting by regularising the shared representation. The model is pushed to learn a more generalizable representation that reflects the underlying structure of the data since it is trained simultaneously on several tasks, as opposed to simply learning task-specific features.

Chapter 4

Basic Problem Statement and Proposed Methodology

We would describe the problem statement and the dataset used in detail in this chapter. Also we would have a look at the developed three solutions for the problem, out of which the ConvLSTM is our main focus.

4.1 Problem Statement

We are provided traffic videos in the form of HDF5 files that would be described in detail in the next section. Our main task is to predict the next three photos in our traffic videos, which encode the volume, speed, and direction of observed traffic in each 100m x 100m grid within a 5min interval into an RGB pixel colour, as specified, for Berlin, Istanbul, and Moscow, for five times on a given test day. Assuming that the objective function of all to be submitted tensors (one for each day in the test set list and each city) is the mean squared error of all pixel channel colour values to pixel colour values derived from true observations, submit the 5 sequences of 3 images in each day of the test set as a multi-dimensional array (tensor) of shape

(5,3,495,436,3). By dividing these pixel colour values by 1, normalise them to be between 0 and 1.

4.2 Dataset Description

The dataset used is taken from **HERE**. Since HDF5 is both space-efficient and extensively supported by a variety of programming and analysis environments, including C/C++, Python, R, etc., data is made available in that format. The data structure is as follows:

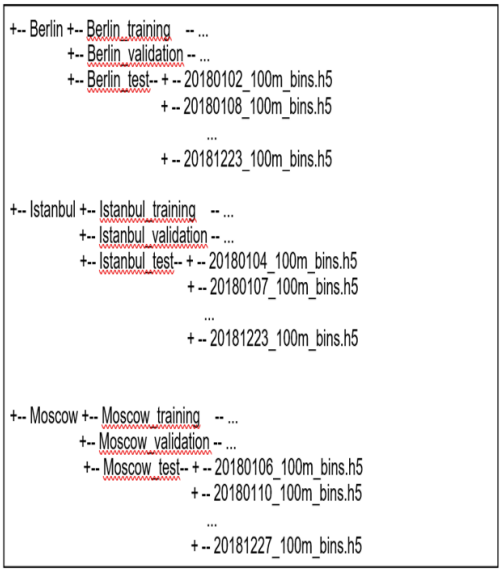

FIGURE 4.1: Structure of the Dataset used

A compressed zip file contains all the information for each specific city. When the package is opened, a top-level directory containing the name of the city and the subfolders *test, *training, and *validation is revealed. The training set consists of 285 HDF5 (*.h5) files, each of which represents a day and has the date in ISO format order in the file name. Seven *.h5 files make up the validation set, whereas 72 *.h5 files make up the test set. Each HDF5 file in the training and validation sets is formatted as an 8-bit unsigned integer tensor of the form (t, h, w, c), where t

denotes the labels for the 5-minute time bins in temporally ascending order and h, w, and c denote the height, width and data channels respectively. With the exception of using 16-bit signed int tensors and zeroing out all but five sets of 12 consecutive image frames, every file in the test set has the same format. Predicting the three frames that come after these non-zero sequences of consecutive frames is the main issue at hand. This means that you can make 15 guesses for each test day.

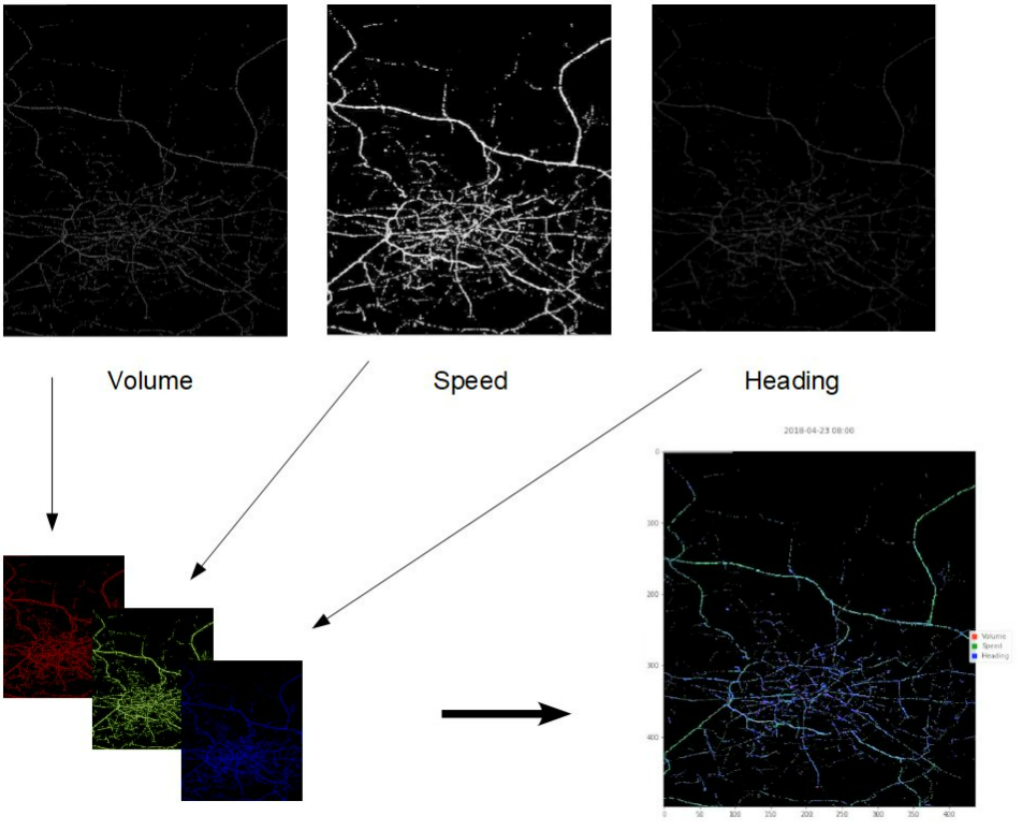

FIGURE 4.2: Colour Encoding in the Dataset

The data set was created from the trajectory of unprocessed GPS location fixes (comprising a latitude, a longitude, a time stamp, the vehicle's speed, and the direction it was travelling at the moment). The information comes from a vast fleet of probe vehicles that tracked people's movements over the course of an entire year in a number of metropolitan locations with various cultures and social structures. The dataset is aggregated using the following steps. The study area is tessellated

using uniform grid cells. All probe points were chosen that were recorded during the chosen time interval and in intersection with one of the cells. Probe points are gathered based on their spatial and temporal characteristics, such as the grid cell in which they are located and the 5-minute time window to which their time stamp corresponds. The mean traffic volume, direction, and speed are calculated using core channels. The encoded values are kept in a tensor of the form (t; h; w; c), where t is the quantity of distinct 5-minute time bins, or the number of frames, and h and w are the height and width of the frame, respectively. c denotes the number of data channels and is equal to 3 when just one channel is used for a traffic state feature.

4.3 Proposed Solutions

4.3.1 Trivial Random Constant Solution

Since we are dealing with HDF5 files for the first time, we wanted to get a feel of the dataset and how to use it in order to be able to implement the ConvLSTM model efficiently. Hence we thought of developing a very trivial solution that outputs a uniform intensity image no matter what the previous traffic flow is. As our objective is to reduce MSE, and since most of the flow is depicted in black as seen from Figure 4.1, we can always output a black image. This seems to be illogical in terms of it not considering the history of the flow, however it achieves our objective of minimizing mean square error to some extent.

4.3.2 Moving Average Solution

We first extract the relevant test cases from the dataset by identifying the 5 sets of 12 non - zero frames in every sample of the test set. This is also done in the above solution. Since Istanbul and Moscow lie in the same time zone, we have the same indices of those 5 sets in both of them whereas Berlin has its corresponding

indices in each of its test samples. Our aim is to predict the next 3 images of traffic flow i.e. a 15 min interval traffic forecasting. We presented a slightly non - trivial solution similar to a sliding window that takes the average of the road networks of the previous 3 time bins and predicts the required outcomes.

4.3.3 ConvLSTM Model

Now we move onto making an end to end trainable model using the ConvLSTM model as already mentioned previously. We developed a 2 layered ConvLSTM sequence to sequence model. Training the model divides each training sample into 48 batches of size 6, in which the first 3 are taken as the input images and the next 3 trained images are predicted. We used Adam optimizer and the loss function is mean square error. Then we tested our trained model by giving some test data input and predicting the future 15 min interval traffic network as done for the above two solutions as well. A pictorial representation of the model can be seen below.

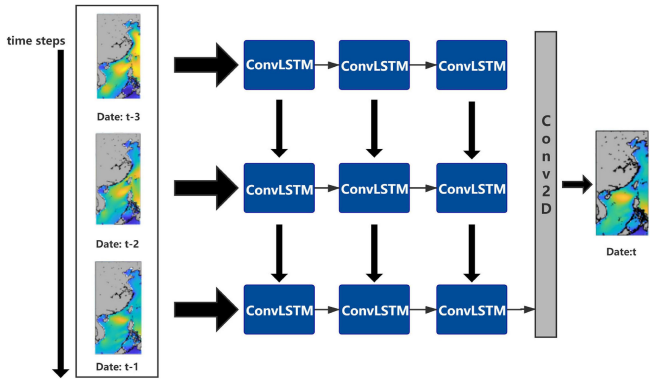

FIGURE 4.3: High Level Representation of a ConvLSTM Network

Also we can ensemble three to four models, for instance, a DCNN model, an LSTM model and a ConvLSTM model and average our predictions to improve the MSE. Also an SVM layer can be put after the flattened layer so that we get a performance boost as in Object Detection tasks.

Chapter 5

Simulation Results of the Basic Problem

In this chapter, we visualise the given dataset as images and convert them into a video to have a pictorial overview of the traffic flow on a single day. Then we show the different traffic flow prediction outputs from the three proposed solutions. We would also have a look at the summary of our ConvLSTM model and how our MSE loss decreases on increasing the number of epochs.

5.1 Visualisation of the Dataset

As stated in the dataset's description, photos are taken every day at intervals of five minutes on a grid of 100 metres by 100 metres. We would therefore have 288 photos packed in the matching HDF5 file for a single training sample. In addition, we have seen that the red channel denotes the volume or density of the traffic flow, the green channel the speed, and the blue channel the direction of the flow.

The below image shows us some of the traffic network flow snapshots in the first training sample of Berlin.

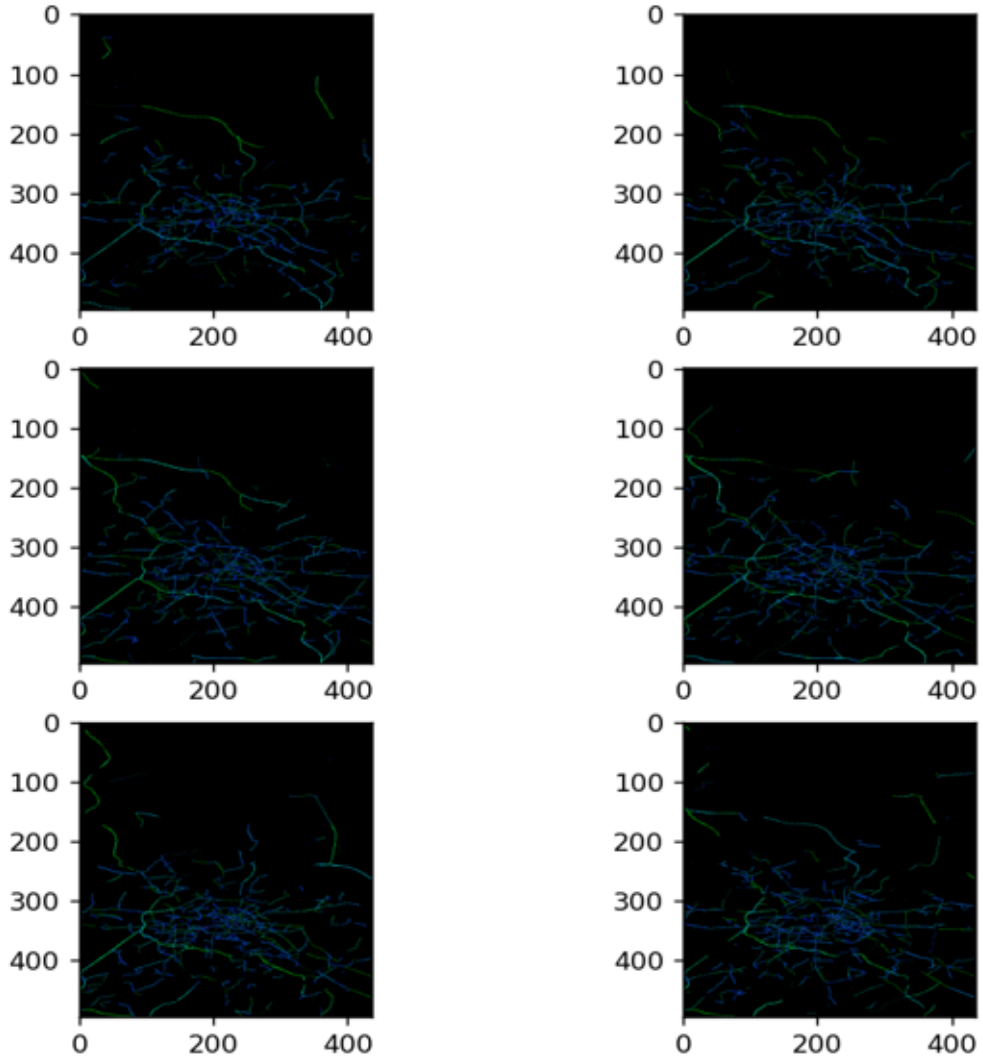

FIGURE 5.1: Traffic Network Flow Snapshots in a day of Berlin

All the 288 images of a particular training sample can be taken and merged into a video which is made available **HERE**.

5.2 Predictions of Random Constant Solution

Here are the predictions of our random constant solution. We chose the random constant to be zero since from the above figure, it is very clear that there is no

traffic flow in many of the areas. As expected, we get black images on proposing this solution. This outputs black images for any test sample.

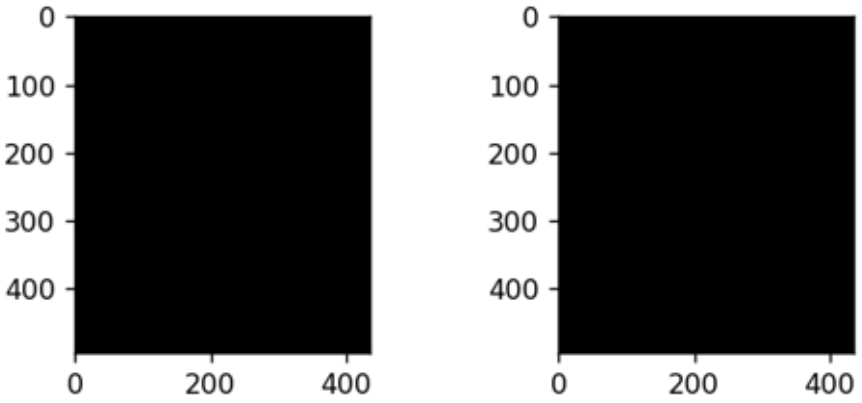

FIGURE 5.2: Predictions of Solution 1

5.3 Predictions of Moving Average Solution

The prediction outputs of the moving average solution are shown below. We have shown the 3 consecutive predictions of the first set of 12 non - consecutive time frames in a test sample of Berlin.

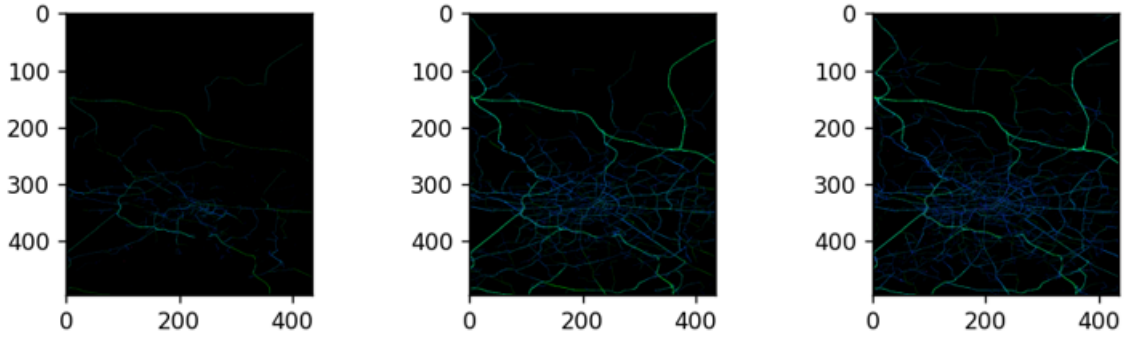

FIGURE 5.3: Predictions of Solution 2

It is very clear that is much better than the trivial solution. However it just takes a linear combination of the previous traffic flow conditions which is not really justifiable. Hence we go on to making a ConvLSTM model.

5.4 Predictions of ConvLSTM Model

5.4.1 Model Architecture

The figure below shows us the Convolutional LSTM architecture developed for the problem at hand. The developed model is a 2 layered Convolutional LSTM that has an encoder - decoder architecture.

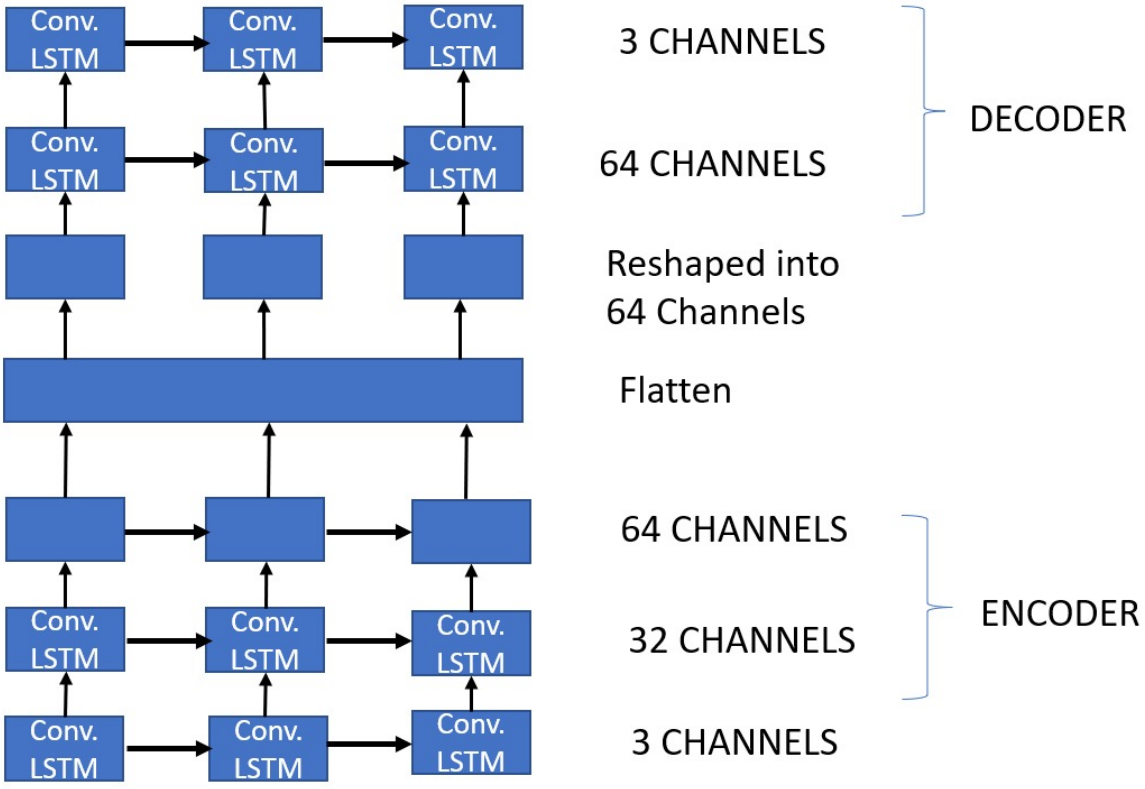

FIGURE 5.4: Architecture of the developed ConvLSTM Model

5.4.2 Model Summary

Below is the summary of the developed ConvLSTM model indicating the total number of trainable parameters and the output shape in each of the layers/blocks. The total number of trainable parameters is about 3 million.

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 prev_frames (InputLayer)    [(None, 3, 3, 495, 436)]  0

 convlstm_0 (ConvLSTM2D)     (None, 3, 32, 495, 436)   219648

 convlstm_1 (ConvLSTM2D)     [(None, 64, 495, 436),    1204480
                              (None, 64, 495, 436),
                              (None, 64, 495, 436)]

 flatten (Flatten)           (None, 13812480)          0

 repeat_vector (RepeatVector  (None, 3, 13812480)      0
 )

 reshape (Reshape)           (None, 3, 64, 495, 436)   0

 convlstm_2 (ConvLSTM2D)     (None, 3, 64, 495, 436)   1605888

 convlstm_3 (ConvLSTM2D)     (None, 3, 3, 495, 436)    39408

=================================================================
Total params: 3,069,424
Trainable params: 3,069,424
Non-trainable params: 0
```

FIGURE 5.5: Model Summary

The above summary is explained in detail as follows.

Number of parameters in a Convolutional Neural Network

$$= (k * k * m + 1) * n$$

Number of parameters in a standard LSTM Model

$$= 4(n * m + n^2 + n)$$

Number of parameters in a Convolutional LSTM Model

$$= 4 * n * (k^2 * (m + n) + 1)$$

So for the first layer we have k=7, m=3 and n=32. Therefore, the total number of parameters are

$$4 * 32 * (7^2 * (3 + 32) + 1) = 219648$$

. This is the overview of the final developed model.

5.4.3 Variation of MSE with Epochs

Epoch Number	**Mean Square Error**
0	0.01170
1	0.00987
2	0.00880
3	0.00821
4	0.00795
5	0.00785
6	0.00775
7	0.00770
8	0.00767
9	0.00764
10	0.00762
11	0.00759
12	0.00758
13	0.00757
14	0.00755

TABLE 5.1: Variation of Loss Function w.r.t Epoch Number

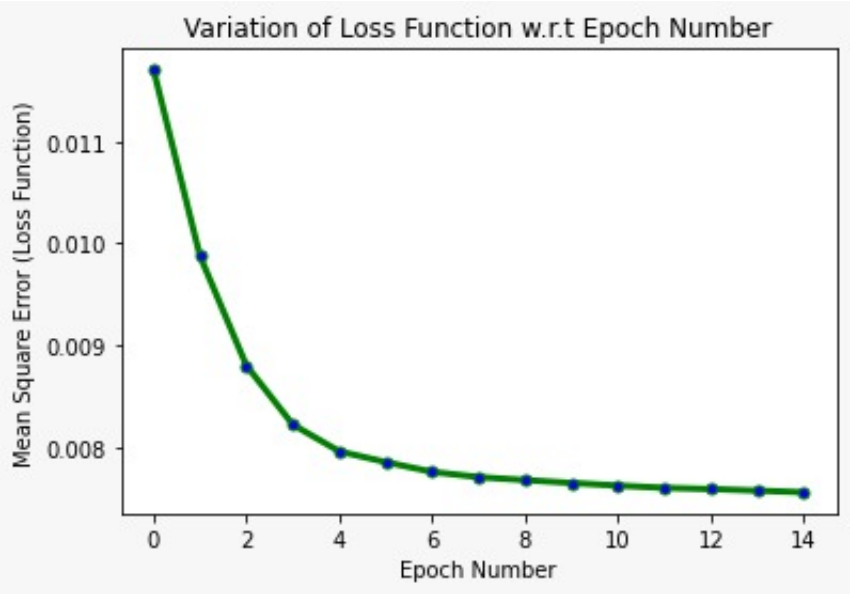

FIGURE 5.6: MSE Loss vs. Epoch Number

The table and the figure above show us the convergence of error as the number of iterations increase. The shown data are associated with the first training sample of Berlin. It can be observed that after 10 epochs, the error remains almost constant. Now we can look at the predicted outputs when we use the trained ConvLSTM model on a test sample.

5.4.4 Outputs

The prediction outputs of the ConvLSTM network are shown below. We have shown 3 non - consecutive predictions of a test sample of Berlin out of all 15. However the outputs that are shown correspond to the model that was obtained when we trained to our best capability using our resources. The intensity values are scaled for better visualisation. Also the GPU of Google Colab has been used and it comes with limitations.

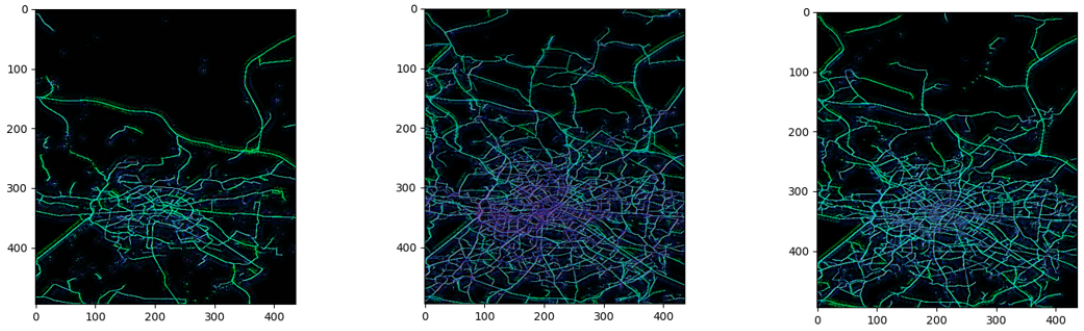

FIGURE 5.7: Predictions of Solution 3

On complete training of the developed model, we could be able to predict short term traffic flow quite accurately.

5.5 Error Analysis - Comparison of Outputs

In this section, we see which of the solutions is the best out of the proposed three solutions by looking average pixel error of the proposed three models.

Solution Type	**Average Pixel Error**
Constant Zero	111.530
Moving Average	62.927
Convolutional LSTM	40.366

TABLE 5.2: Comparsion of Outputs of different solutions

Hence we can say that the constant zero solution has very high error as expected and is of no use. The moving average solution is better than the constant zero solution but not very optimal since it is just a linear combination of the previous inputs. The developed ConvLSTM model has the least error of all three and can be further decreased on training more and more samples.

Chapter 6

Advanced Problem Statement and Proposed Methodology

In this chapter, we would explore a variant of the previous problem statement. We mainly focus on the issues of Temporal and Spatial Few Shot Transfer Learning. The problem statement, the dataset used as well as the proposed methodologies would be described in detail in this chapter.

6.1 Problem Statement

Now we would like to extend the previous problem statement. Exploring models that can adapt to domain shifts through time and space is of great interest to us. In particular, dynamic traffic data is offered for 4 separate cities in an improved format of 8 channels as opposed to simply 3. The following section has more information about the subject. The first half of this data will be from 2019, prior to the COVID epidemic, and the second half will be from 2020, when the pandemic began to have an impact on every aspect of our life. Static data on road shape are also given apart from the dynamic data.

A pictorial representation as shown below of the problem statement would help to get an overview of the task we are about to perform.

FIGURE 6.1: Pictorial Representation of the Problem Statement

Our first task is to handle the COVID-19 related temporal domain shift in traffic. We have the whole data for the four locations mentioned above, as well as pre-COVID data for four more cities and 100 1-hour time intervals in 2020 after COVID struck. The next task is to forecast the dynamic traffic conditions for each of the extra 4 cities 5, 10, 15, 30, 45, and 60 minutes in advance after each of the 100 time slots.

Next we proceed to estimate dynamic traffic conditions for two additional, previously unknown cities. Similar to the last example, traffic must be predicted 5, 10, 15, 30, 45, and 60 minutes in advance of 100 assigned one-hour time periods. However, no more traffic information will be given for these cities. Furthermore, 50 of the 100 1 hour time slots for each city will be drawn from the time before COVID and 50 from the time after COVID, without specifying which. The issue can be briefly visualized as follows.

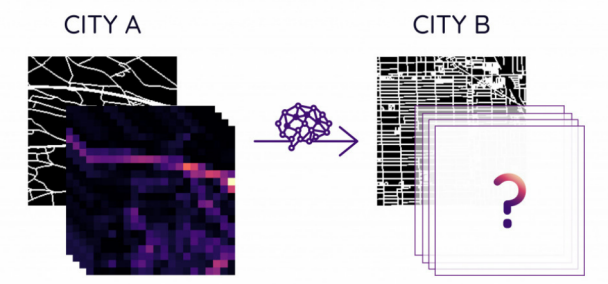

FIGURE 6.2: Temporal and Spatial Few Shot Transfer Learning

6.2 Dataset Description

6.2.1 Data Collection

The information is collected from industrial-scale trajectories of raw GPS location fixes, which include a time stamp, latitude, longitude, as well as the vehicle's speed and direction at the moment. The information comes from a sizable fleet of probe cars and is compiled and made accessible by HERE Technologies. The data spans the years 2019 and 2020, giving the possibility to track the COVID pandemic's effects across several cities. To support both the temporal and spatial transfer learning tasks, datasets from 10 cities are provided in various arrangements.

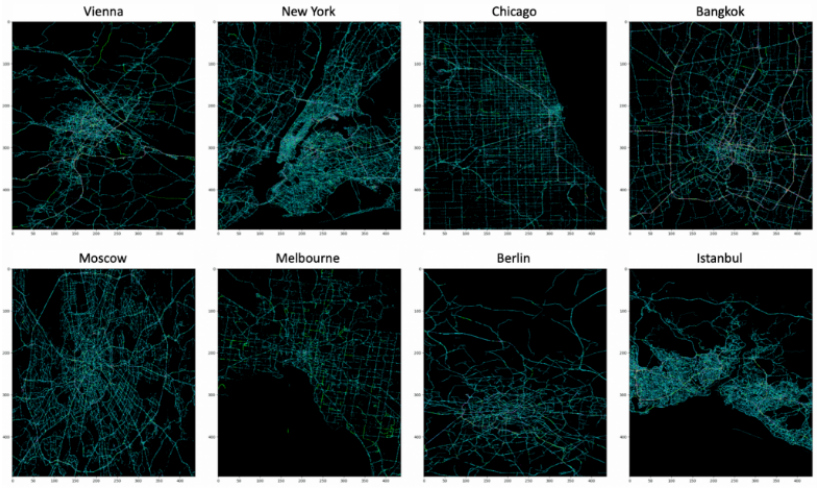

FIGURE 6.3: Sample Traffic Snapshots in Different Cities

The picture above displays representative traffic drawings from eight distinct cities, showcasing the variety of places covered with variations in the layout of the fleet, traffic patterns, and other cultural aspects. Each snapshot covers an urban region of about 50 km by 50 km, and our data consequently offers thorough coverage of complicated cities. The amount of traffic in a 100 m by 100 m region at the moment of the snapshot and for a 5-minute time window is shown by pixel brightness. The sum over several channels and time windows is displayed.

6.2.2 Structure of the Dataset

Two features are calculated for each heading direction quadrant of North-East (heading 0°-90°), South-East (heading 90°-180°), South-West (heading 180°-270°), and North-West (heading 270°-0°) using the GPS data that have been aggregated into an 8-channel encoding. One of them is Volume, which represents the total number of probe points from all HERE sources that have been normalised, discretized, and capped at both the upper and lower limits. This number ranges from 0 to 255. The other is the average speed determined from the gathered probe sites is the mean speed. The values are capped at a maximum level, discretized to 1 to 255, then rounded to the nearest integer by linearly scaling the capping speed to 255. Value 0 denotes that no probes were gathered. An illustration of the eight dynamic probe data channels—two for each heading quadrant—can be seen in the following picture.

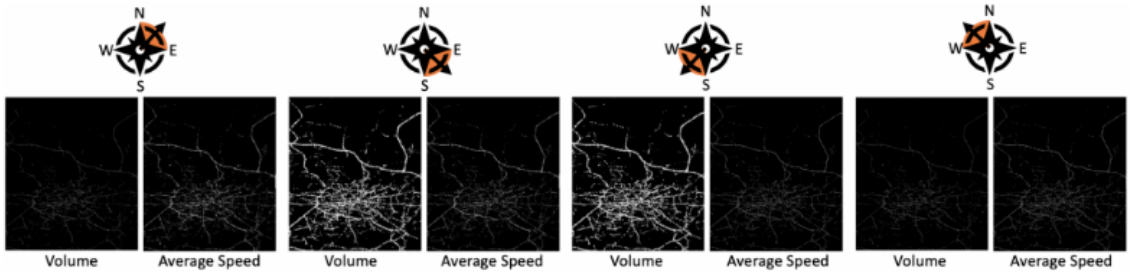

FIGURE 6.4: Dynamic Channels

To encode contextual information about a cell, additional static channels are offered. These static channels include details on the complexity, interconnectedness, and road topology of a particular cell. For these channels, all that is needed is a spatial resolution because the road geometry features at this aggregate level are often quite static over time in most cities. There are additionally given connectivity channels that encode the road geometry connections to the neighbouring cells and their general relative throughput. To facilitate the learning of universal, transferable rules that solely depend on standardised, city-independent road aspects, such non-local connection information is essential for the spatial transfer learning component.

Two static files are provided for each city.

- ⟨CITY NAME⟩_static.h5 which has a tensor of (9,495,436). The city map is shown in grayscale on the first channel in the same resolution as the dynamic data. The remaining 8 layers represent a binary encoding of the neighbour cell's connectivity to the North, North East, East, South East, South, South West, West, and North West, respectively.
- ⟨CITY NAME⟩_static_map_high_res.h5 which contains a (4950,4360) tensor of high-resolution gray-scale map with pixels that are roughly 10m x 10m in size and can be easily mapped by a factor of 10. In actuality, the first channel's lower quality map is a downsampled reproduction of this high resolution map.

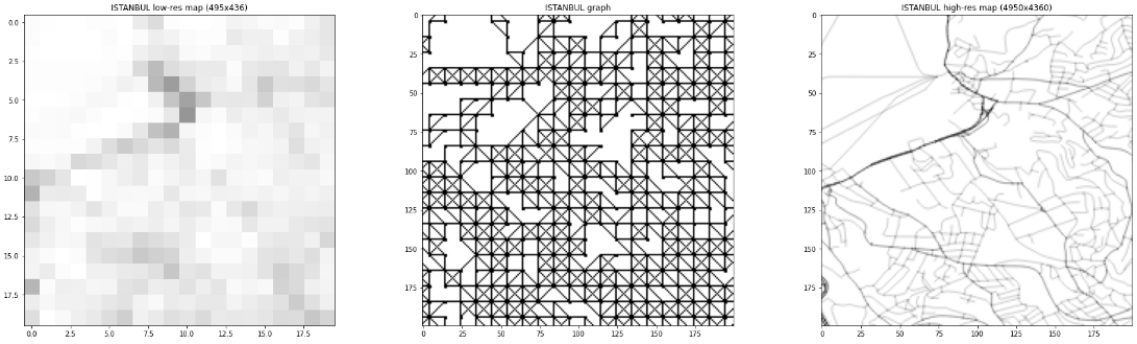

FIGURE 6.5: Road Map of Istanbul

An excerpt of the Istanbul map is seen in the image above. The low resolution map is on the left, the high resolution map is on the right, and in the middle is a representation of the graph made up of the 8 connection layers, which are nothing more than the 8 static channels. Since the reversed edge is always present, the graph is essentially undirected.

The test dataset is as follows.

- Any file ⟨CITY NAME⟩_test_[temporal/spatiotemporal].h5 in the testing set contains a tensor of size $(100, 12, 495, 436, 8)$ to create predictions on that day.

The number 12 denotes that we present 12 consecutive photographs of our time bins of 5 minutes, spanning a total of 1 hour. It is our responsibility to make predictions for the next 5, 10, 15, 30, 45, and 60 minutes. A tensor of dimension $(100, 6, 495, 436, 8)$ representing the six time forecasts must be the output.

- In addition, ⟨CITY NAME⟩_test_additional_[temporal/spatiotemporal].h5 contains a (100,2) tensor, with the first channel denoting the day of the week (0 = Monday,..., 6 = Sunday) and the second channel the test slot's local time (0,...240).

6.2.3 Analysis of the Temporal Shift

Let's examine the temporal change in the data for certain cities from pre-COVID to post-COVID. We plot the total volume over all four heading directions for all 288 5-minute bins in a day. For all cities, there is a noticeable change in the relative quantities. The sums do not represent the absolute volume because volumes in each pixel are capped and normalised during data preparation. Local time is displayed on the X-axis.

The Istanbul curve is shown below. Even if there has been a noticeable decrease in volume, the daily pattern of traffic volume evolution is otherwise relatively consistent. In Turkey in 2020, only certain age groups and weekends were subject to curfews.

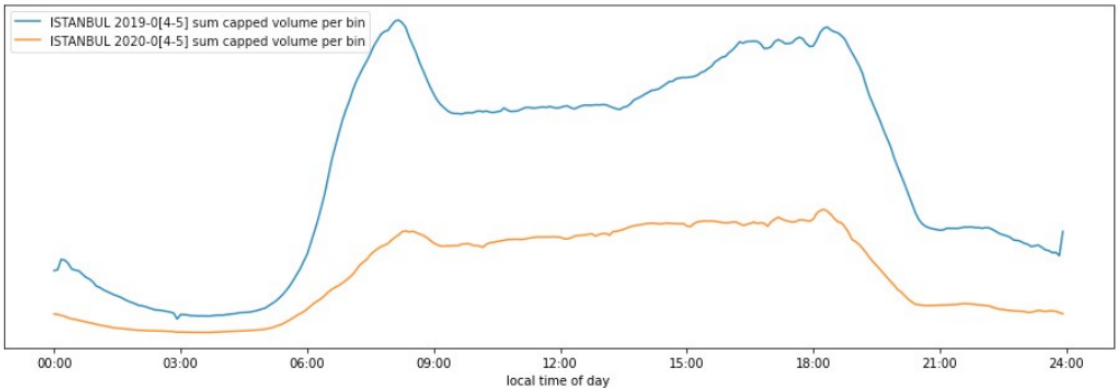

FIGURE 6.6: Temporal Shift in Istanbul

The plot of Berlin is especially intriguing since while German initiatives appear to have significantly reduced morning rush hour congestion, the afternoon peak persisted at nearly pre-pandemic levels.

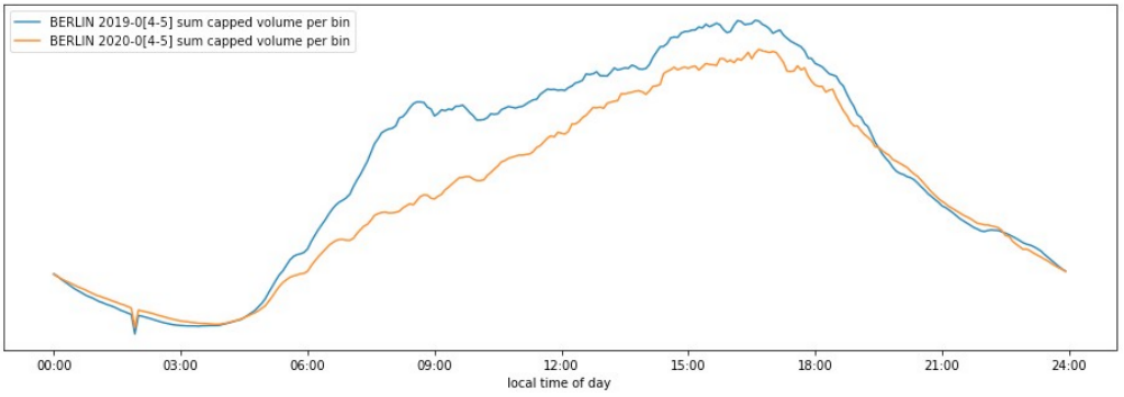

FIGURE 6.7: Temporal Shift in Berlin

6.2.4 Analysis of the Spatial Data Properties

Let's now examine the spatial data attributes of the cities under consideration. We notice a large disparity in the road network's density, which is obviously related to the population density. The GPS data used to create the probe data encodings include extra features and biases. In addition to this municipal bias, the frequency

and stability of the recordings vary depending on the city district. Berlin is used as an example to show this.

We choose three examples from Berlin. 1 is a highway on the periphery, 2 is a highway at the major ring road and 3 is a boulevard in the in the heart of Berlin containing both business and homes. The graph below displays the histograms of the normalised volumes (4 channels) and speeds (4 channels) that were captured during a 5-minute period at 10:00 am. The volumes show the variations between less-frequented city centre areas (2 and 3) and outlying places (1). Due to a greater speed restriction and fewer traffic, the highway in 1 is believed to have higher speeds. Due to the reduced speed restriction and heavier traffic on the route in 2, daytime speeds are lower. The expectedly slow speeds on boulevard(3) are also present.

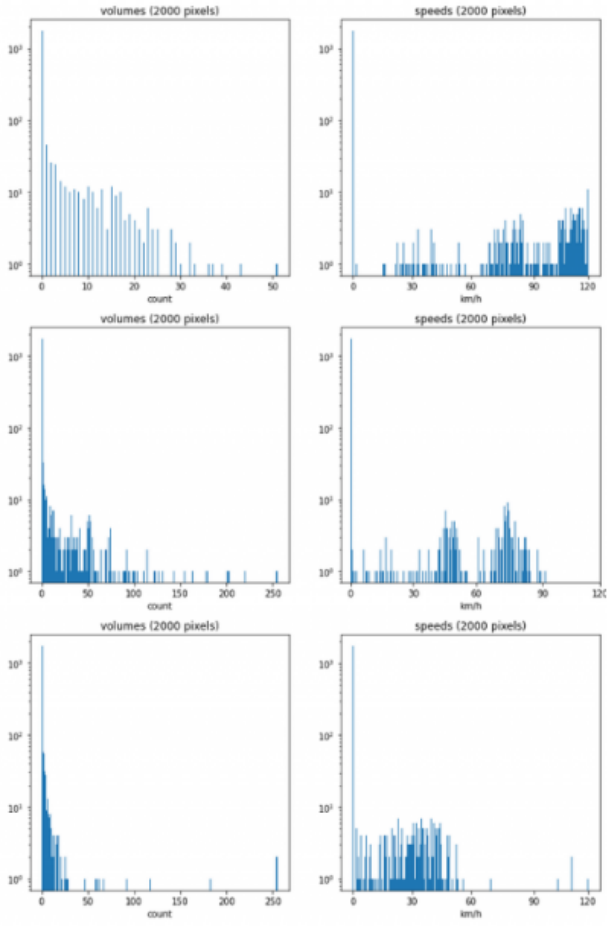

FIGURE 6.8: Histograms of Volume and Speed in 3 areas of Berlin

The following graphs are produced when we concentrate on a single cell in the centre of the aforementioned regions and examine the volume and speed throughout the course of a day, say Wednesday. Red lines represent the speeds, and blue bars represent the volumes. With growing traffic throughout morning and evening rush hours but no congestion because we are on the periphery, the first plot for area 1 shows the typical daily trend. The second figure for region 2 shows the volume evolution during the day, with slower speeds during the day, particularly in the morning and afternoon, and a congestion between 6 pm and 7 pm. The city boulevard is depicted in the third plot for area 3 with very flaky data because speeds are often low and volumes are spikey because of things like cars stopping.

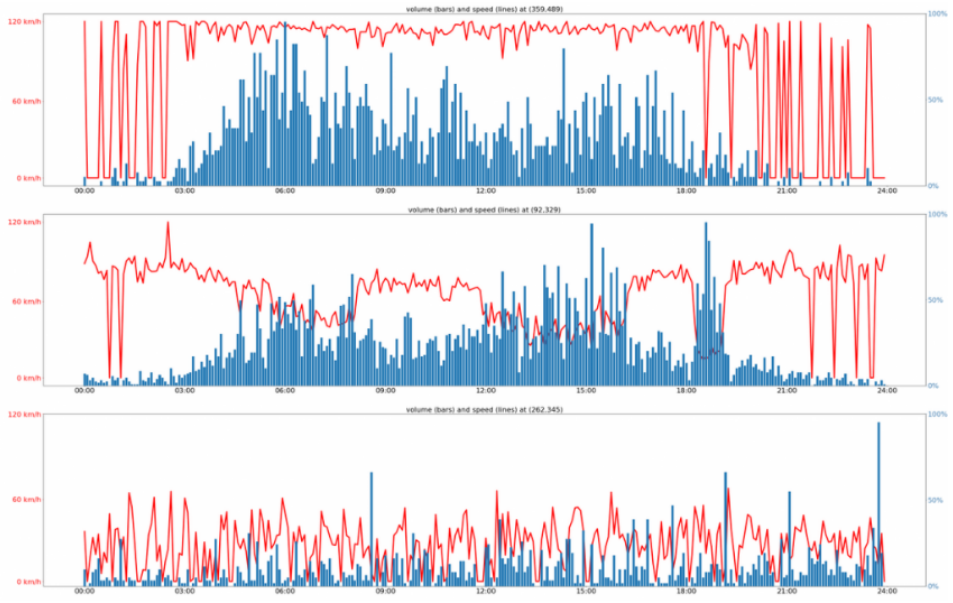

FIGURE 6.9: Volume and Speed plots on a typical Wednesday

The next charts, in comparison, display the identical pixels' speeds and volumes for a Sunday. All three plots show a clear reduction in volume in the early morning hours, while the second plot for area 2 also shows high traffic on the ring road in the late morning.

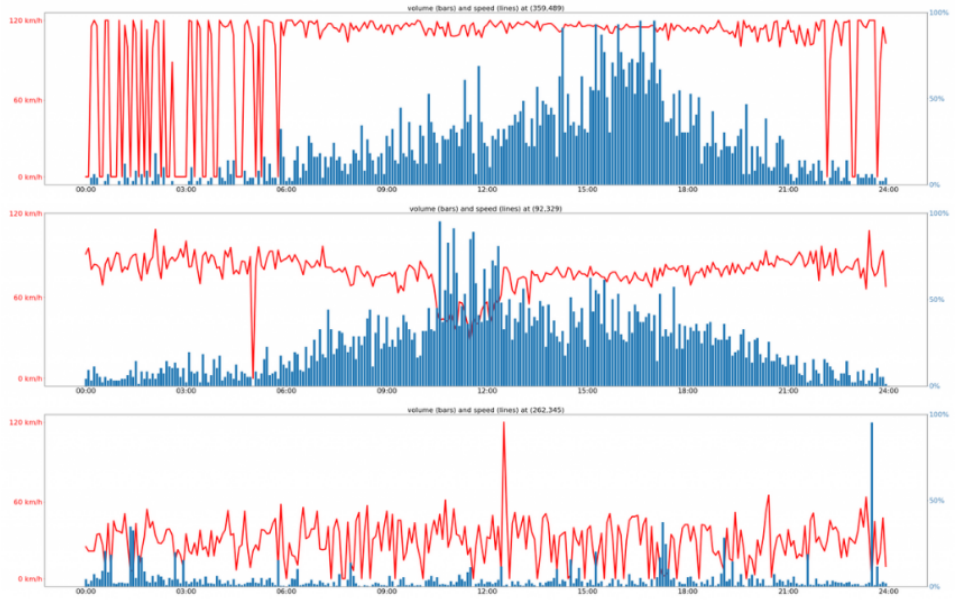

FIGURE 6.10: Volume and Speed plots on a typical Sunday

As a result, the input data might be described as sparse. Models must be able to handle this sparsity in order to accurately anticipate the traffic volume.

6.3 Proposed Methodology

6.3.1 Tackling Temporal Shift

Due to U-Net's proven superior performance in traffic forecasting, we use it as our model to handle temporal change. U-Net3.3 has been widely used in a number of dense prediction applications at the pixel level.

The encoder is made up of a series of K blocks, each of which is made up of two 3 $\times$ 3 convolutional layers, followed by a rectified linear activation unit and a group normalisation layer with group size set to 8 [12], followed by a 2 $\times$ 2 max-pooling layer with stride 2. After each downsampling process, the number of filters in the convolutional layers is doubled. Additionally, the decoder consists of a series of

K blocks, each of which is made up of two 3 × 3 convolutional layers and a 2 × 2 upsampling layer that performs a transposed convolution operation and halves the number of filters. After linking the encoder and decoder, we apply two 3 × 3 convolutional layers, and as the final layer, we apply a 1 × 1 convolutional layer to produce predictions for future traffic flow.

The input to the traffic forecasting model is a 12 × 495 × 436 × 8 tensor, where 495 by 436 is the spatial resolution of the city heatmap, and each pixel's channel values represent the observed traffic in each 100m by 100m grid within a 5 minute interval. The volume and speed in the four headings (NE, SE, SW, and NW) are represented by the eight channels of each heatmap, where the values are discretized and normalised to an integer number between 0 and 255. A stack of 12 heatmaps at 5 minute intervals, spanning a total of 1 hour, is used as the input to the traffic forecasting model.

We combine the 12 heatmaps across the channel dimension to produce a tensor of the dimensions 495 × 436 × 96. Additionally, we concatenate the input with the static information, which has the dimensions 495 × 436 × 9, where the nine channels of static information encode the road network's density and its connections to its eight surrounding cells. U-Net produces a tensor with the dimensions 495 × 436 × 48 from an input tensor with the dimensions 495 × 436 × 105. The output of U-Net is then reconfigured into 6 × 495 × 436 × 8, where the 6 projected heatmaps represent the predicted traffic states for 5, 10, 15, 30, 45, and 60 minutes in the future, respectively.

6.3.2 Tackling Spatial Shift

For traffic forecasting, a multi-task learning architecture is provided to handle the spatial shift from one city to another. In order to train the model to jointly forecast the future traffic states for various cities, we randomly choose data from all the

cities that are accessible during training rather than feeding the model with data from a single city. Assume there are M cities and N_i training examples in the i^{th} city. The multi-task learning framework's primary goal is to simultaneously reduce the pixel-wise squared disparity between the ground facts and forecasted traffic map movies across all cities. The goal function would look like

$$Loss = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{N_i}\sum_{j=1}^{N_i}\frac{1}{495\times 436\times 8}\sum_{h=1}^{495}\sum_{w=1}^{436}\sum_{k=1}^{48}(X(i,j,h,w,k)-\hat{X}(i,j,h,w,k))^2$$

where $X(i,j,h,w,k)$ and $\hat{X}(i,j,h,w,k)$ indicate, respectively, the predicted and actual values of the pixels in the k^{th} channel at location (h,w) in the j^{th} training instance of the i^{th} city. A sliding window over the daily traffic map data is used to obtain the training cases. The test slot is scheduled to begin no later than 8 p.m. Therefore, we also set the sliding window's earliest possible start time to 8 p.m. We have 12 heatmaps, each of which corresponds to a 5-minute time bin, for each hour, which translates to $12 \times (12+8)+1 = 241$ training instances per city every day.

The multi task learning framework is shown below.

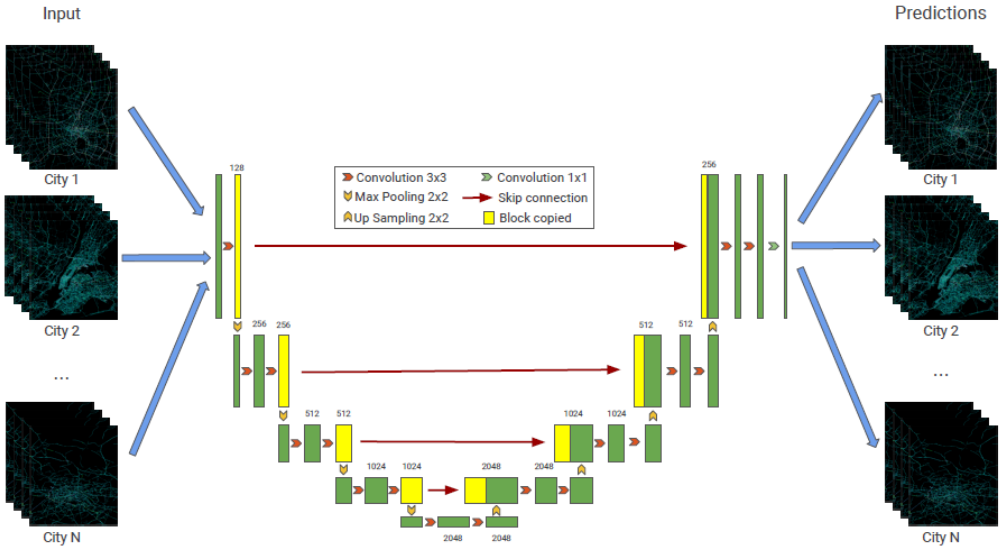

FIGURE 6.11: Multi Task Learning Framework

Motivation behind using multi-task learning is the fact that it enables the model to investigate and utilise the shared information in traffic forecasts for several cities. The multi-task learning approach would also encourage the model to capture the diverse graph structures inside the road networks of different cities when we concatenate the 9-channel static information to the input. We also note that if we train the model using only data from a single city due to the dearth of data, the machine soon memorises the training data and begins overfitting. Multi-task learning has a better performance than the baseline method since it may be seen as an implicit technique for data augmentation and regularisation[9]. Additionally, the multi-task learning framework forces the model to learn representations independent of cities, greatly enhancing data efficiency and minimising overfitting. Also because only one model needs to be trained for all cities, the multi-task learning approach is very effective in terms of training time.

Chapter 7

Simulation Results of the Advanced Problem

In this chapter, we visualise the dataset as images. We would look at both the high and low resolution road networks of cities, the static connectivity channels that make the road graph and also the dynamic channels. Then we show the different traffic flow prediction outputs from the proposed solutions. We would also have a look at the summary of the models used.

7.1 Visualisation of Road Maps

As already mentioned in the previous chapter, we have two static files for each city - one of high resolution and the other of low resolution. The high-resolution map is a gray-scale map where each pixel corresponds to approximately 10m x 10m. The low resolution map is the first channel of the 9 static channels and is a downsampled version of the former one by a factor of 10.

Below are the high and low resolution road maps of Bangkok provided as part of the dataset.

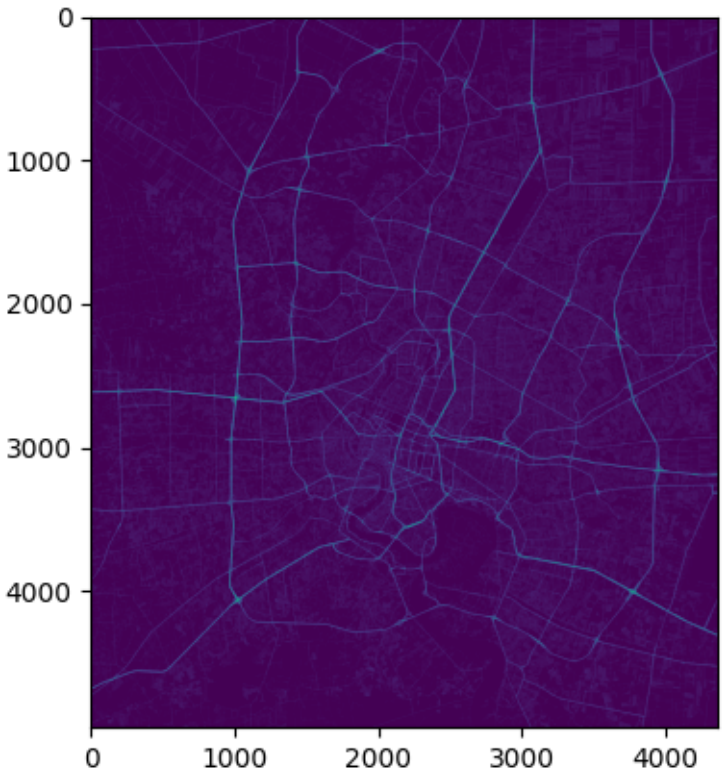

FIGURE 7.1: High Resolution Road Map of Bangkok

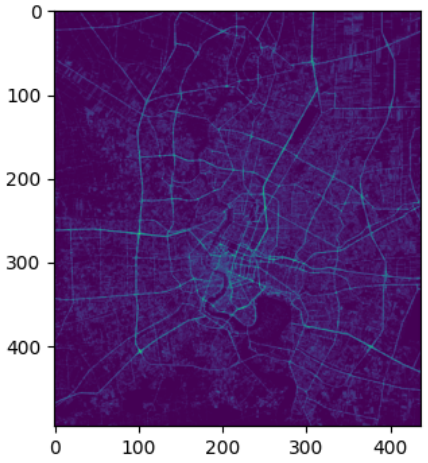

FIGURE 7.2: Low Resolution Road Map of Bangkok

7.2 Visualisation of Static Connectivity Channels

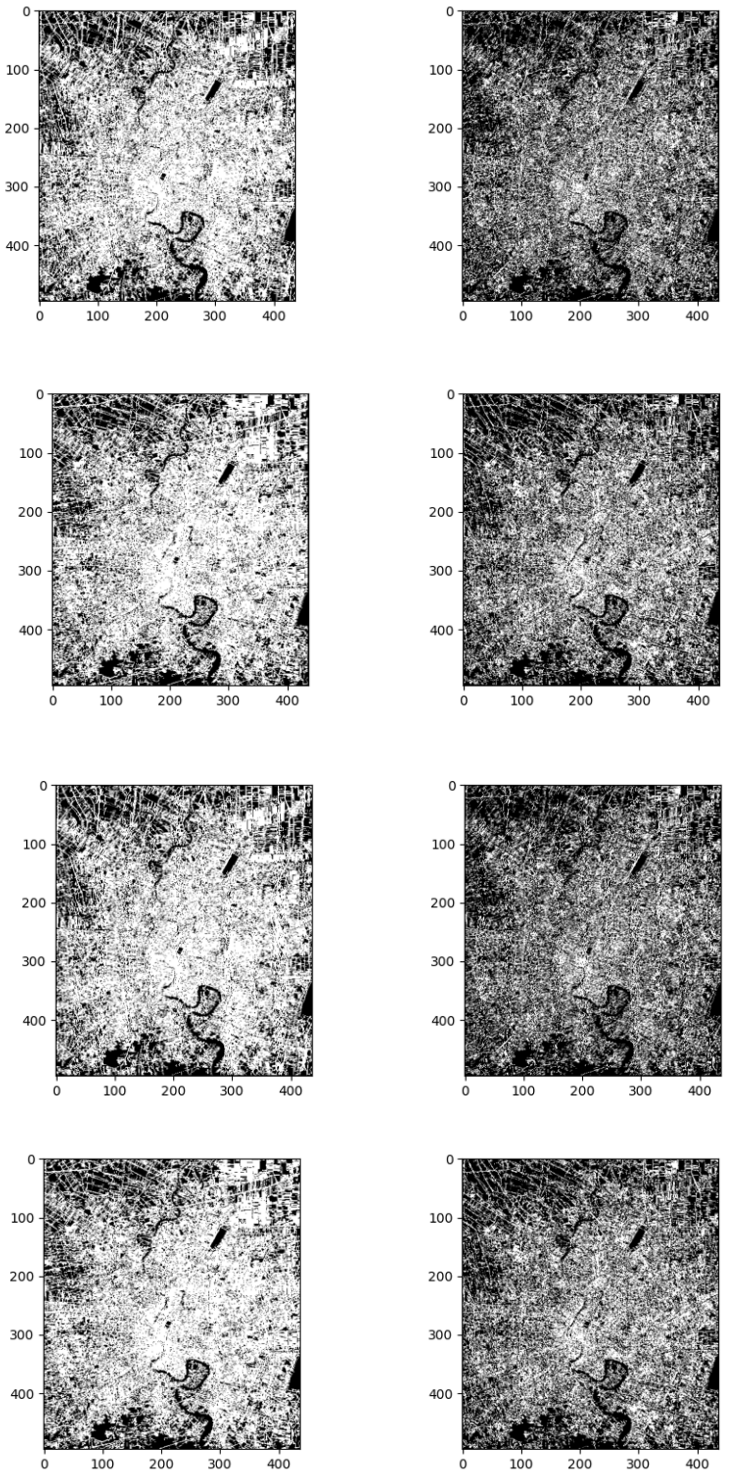

FIGURE 7.3: Static Connectivity Channels of Bangkok

The 8 connectivity channels which encode the road geometry connections to the neighboring cells are visualised above for Bangkok. They are a binary encoding that indicates whether a cell is related to its neighbours to the north, north east, east, south, south west, west, and north west, respectively.

7.3 Visualisation of Dynamic Channels

The figure below shows an example of 4 of the dynamic probe data channels for average speed, 1 for each heading quadrant i.e. North East, South East, South West and North West respectively.

The channels shown below correspond to a particular sample in the dataset associated with Bangkok. We would have similar data channels for volume as well.

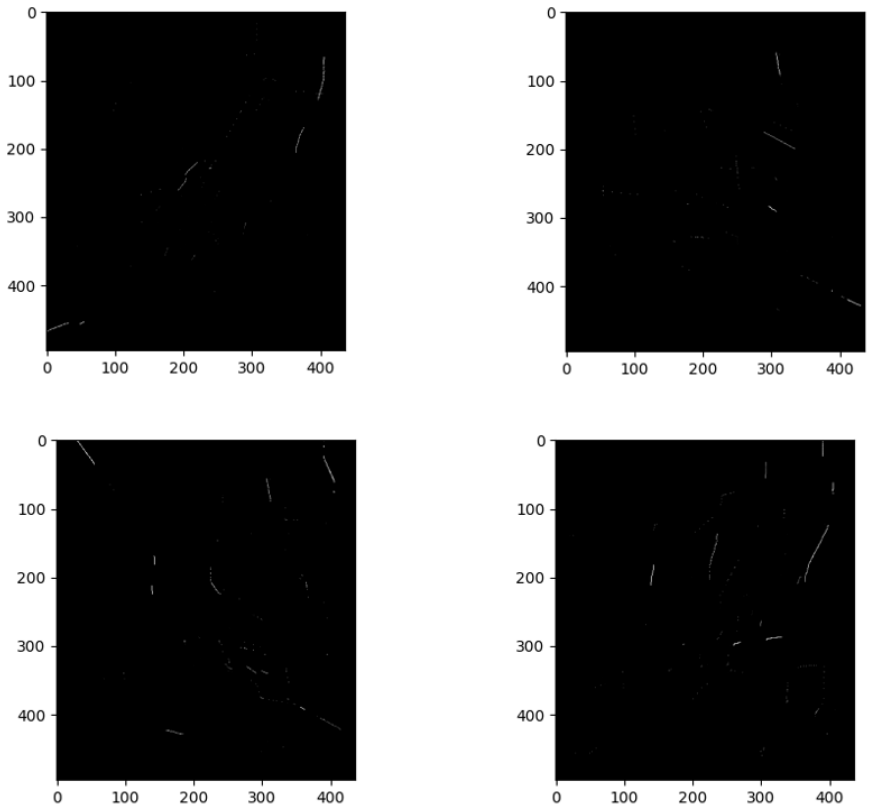

FIGURE 7.4: Sample Dynamic Channels of Average Speed in Bangkok

7.4 Predictions associated with Temporal Shift

A model based on U-Net is used to capture the temporal shift in cities. The summary of the model that is used is as follows. It indicates the total number of trainable parameters and the output shape in each of the layers/blocks. The total number of trainable parameters is about 124 million. -1 refers to batch size.

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 128, 496, 448]	121,088
GroupNorm-2	[-1, 128, 496, 448]	256
ReLU-3	[-1, 128, 496, 448]	0
Conv2d-4	[-1, 128, 496, 448]	147,584
GroupNorm-5	[-1, 128, 496, 448]	256
ReLU-6	[-1, 128, 496, 448]	0
UNetConvBlock-7	[-1, 128, 496, 448]	0
Conv2d-8	[-1, 256, 248, 224]	295,168
GroupNorm-9	[-1, 256, 248, 224]	512
ReLU-10	[-1, 256, 248, 224]	0
Conv2d-11	[-1, 256, 248, 224]	590,080
GroupNorm-12	[-1, 256, 248, 224]	512
ReLU-13	[-1, 256, 248, 224]	0
UNetConvBlock-14	[-1, 256, 248, 224]	0
Conv2d-15	[-1, 512, 124, 112]	1,180,160
GroupNorm-16	[-1, 512, 124, 112]	1,024
ReLU-17	[-1, 512, 124, 112]	0
Conv2d-18	[-1, 512, 124, 112]	2,359,808
GroupNorm-19	[-1, 512, 124, 112]	1,024
ReLU-20	[-1, 512, 124, 112]	0
UNetConvBlock-21	[-1, 512, 124, 112]	0
Conv2d-22	[-1, 1024, 62, 56]	4,719,616
GroupNorm-23	[-1, 1024, 62, 56]	2,048
ReLU-24	[-1, 1024, 62, 56]	0
Conv2d-25	[-1, 1024, 62, 56]	9,438,208
GroupNorm-26	[-1, 1024, 62, 56]	2,048
ReLU-27	[-1, 1024, 62, 56]	0
UNetConvBlock-28	[-1, 1024, 62, 56]	0
Conv2d-29	[-1, 2048, 31, 28]	18,876,416
GroupNorm-30	[-1, 2048, 31, 28]	4,096
ReLU-31	[-1, 2048, 31, 28]	0
Conv2d-32	[-1, 2048, 31, 28]	37,750,784
GroupNorm-33	[-1, 2048, 31, 28]	4,096
ReLU-34	[-1, 2048, 31, 28]	0
UNetConvBlock-35	[-1, 2048, 31, 28]	0
ConvTranspose2d-36	[-1, 1024, 62, 56]	8,389,632
Conv2d-37	[-1, 1024, 62, 56]	18,875,392
GroupNorm-38	[-1, 1024, 62, 56]	2,048
ReLU-39	[-1, 1024, 62, 56]	0
Conv2d-40	[-1, 1024, 62, 56]	9,438,208
GroupNorm-41	[-1, 1024, 62, 56]	2,048
ReLU-42	[-1, 1024, 62, 56]	0
UNetConvBlock-43	[-1, 1024, 62, 56]	0
UNetUpBlock-44	[-1, 1024, 62, 56]	0
ConvTranspose2d-45	[-1, 512, 124, 112]	2,097,664
Conv2d-46	[-1, 512, 124, 112]	4,719,104
GroupNorm-47	[-1, 512, 124, 112]	1,024
ReLU-48	[-1, 512, 124, 112]	0
Conv2d-49	[-1, 512, 124, 112]	2,359,808
GroupNorm-50	[-1, 512, 124, 112]	1,024
ReLU-51	[-1, 512, 124, 112]	0
UNetConvBlock-52	[-1, 512, 124, 112]	0
UNetUpBlock-53	[-1, 512, 124, 112]	0
ConvTranspose2d-54	[-1, 256, 248, 224]	524,544
Conv2d-55	[-1, 256, 248, 224]	1,179,904
GroupNorm-56	[-1, 256, 248, 224]	512
ReLU-57	[-1, 256, 248, 224]	0
Conv2d-58	[-1, 256, 248, 224]	590,080
GroupNorm-59	[-1, 256, 248, 224]	512
ReLU-60	[-1, 256, 248, 224]	0
UNetConvBlock-61	[-1, 256, 248, 224]	0
UNetUpBlock-62	[-1, 256, 248, 224]	0
ConvTranspose2d-63	[-1, 128, 496, 448]	131,200
Conv2d-64	[-1, 128, 496, 448]	295,040
GroupNorm-65	[-1, 128, 496, 448]	256
ReLU-66	[-1, 128, 496, 448]	0
Conv2d-67	[-1, 128, 496, 448]	147,584
GroupNorm-68	[-1, 128, 496, 448]	256
ReLU-69	[-1, 128, 496, 448]	0
UNetConvBlock-70	[-1, 128, 496, 448]	0
UNetUpBlock-71	[-1, 128, 496, 448]	0
Conv2d-72	[-1, 48, 496, 448]	6,192

Total params: 124,256,816
Trainable params: 124,256,816
Non-trainable params: 0

FIGURE 7.5: Model Summary

Number of parameters in a Convolutional Neural Network

$$= (k * k * m + 1) * n$$

where m and n are the number of input and output channels respectively. So for the first layer we have k=3, $m = 12 * 8 + 9 = 105$ and n=128. Therefore, the total number of parameters are

$$(3 * 3 * 105 + 1) * 128 = 121088$$

Similar calculations can be done for all other layers too. This is the overview of the developed model.

The prediction outputs are shown for the city Chicago using the model whose summary is provided above. The 4 dynamic data channels for average speed after 5 minutes for one of the test slots are shown below. Similarly, we would have dynamic channels for volume also.

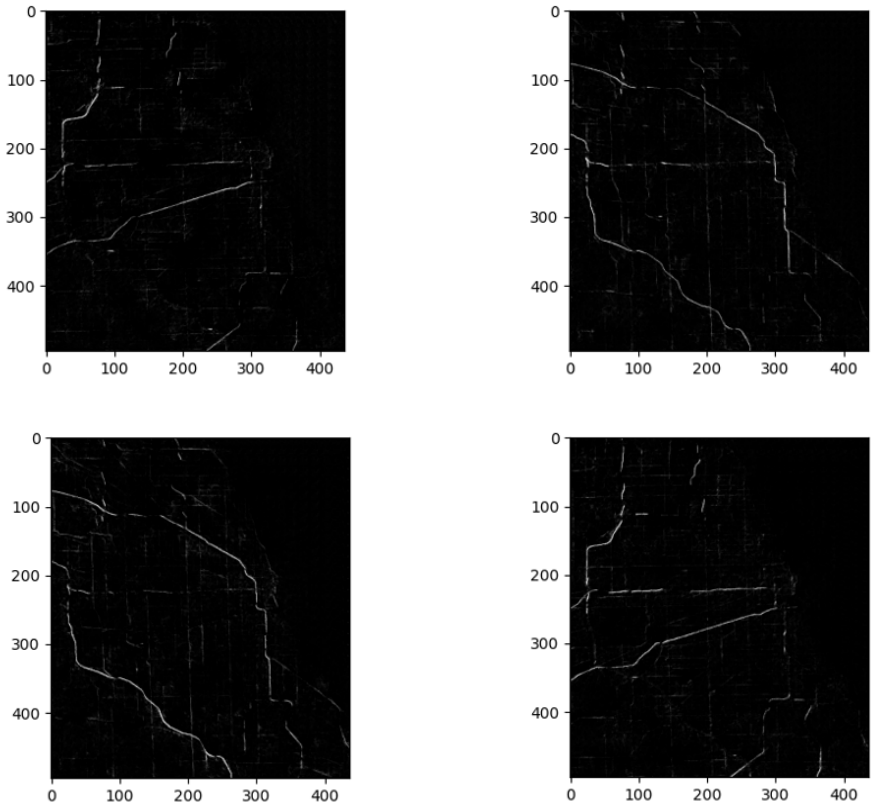

FIGURE 7.6: Predicted Dynamic Channels of Average Speed in Chicago

7.5 Predictions associated with Spatial Shift

As already mentioned, to tackle spatial shift, a multi-task learning framework is developed using U-Net as the base model. The summary of the model that is used is as follows. It indicates the total number of trainable parameters and the output

shape in each of the layers/blocks. The total number of trainable parameters is about 2.5 million. -1 refers to batch size. The parameter calculation is same as previously done.

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 128, 496, 448]	121,088
GroupNorm-2	[-1, 128, 496, 448]	256
ELU-3	[-1, 128, 496, 448]	0
UNetConvLayer-4	[-1, 128, 496, 448]	0
Conv2d-5	[-1, 128, 496, 448]	268,544
GroupNorm-6	[-1, 128, 496, 448]	256
ELU-7	[-1, 128, 496, 448]	0
UNetConvLayer-8	[-1, 128, 496, 448]	0
UNetConvBlock-9	[-1, 128, 496, 448]	0
Conv2d-10	[-1, 256, 248, 224]	295,168
GroupNorm-11	[-1, 256, 248, 224]	512
ELU-12	[-1, 256, 248, 224]	0
UNetConvLayer-13	[-1, 256, 248, 224]	0
Conv2d-14	[-1, 256, 248, 224]	884,992
GroupNorm-15	[-1, 256, 248, 224]	512
ELU-16	[-1, 256, 248, 224]	0
UNetConvLayer-17	[-1, 256, 248, 224]	0
UNetConvBlock-18	[-1, 256, 248, 224]	0
ConvTranspose2d-19	[-1, 128, 496, 448]	131,200
Conv2d-20	[-1, 128, 496, 448]	295,040
GroupNorm-21	[-1, 128, 496, 448]	256
ELU-22	[-1, 128, 496, 448]	0
UNetConvLayer-23	[-1, 128, 496, 448]	0
Conv2d-24	[-1, 128, 496, 448]	442,496
GroupNorm-25	[-1, 128, 496, 448]	256
ELU-26	[-1, 128, 496, 448]	0
UNetConvLayer-27	[-1, 128, 496, 448]	0
UNetConvBlock-28	[-1, 128, 496, 448]	0
UNetUpBlock-29	[-1, 128, 496, 448]	0
Conv2d-30	[-1, 48, 496, 448]	6,192

Total params: 2,446,768
Trainable params: 2,446,768
Non-trainable params: 0

FIGURE 7.7: Model Summary

The prediction outputs are shown for the city NewYork using the model whose summary is provided above. The 4 dynamic data channels for average speed after 5 minutes for one of the test slots are shown below. Similarly, we would have dynamic channels for volume also.

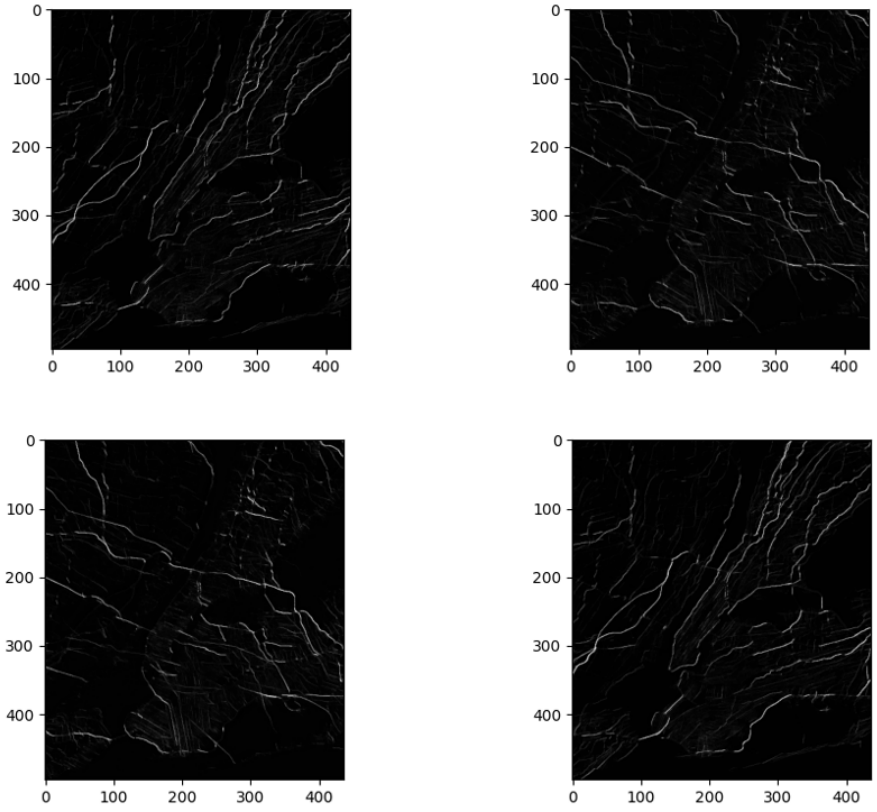

FIGURE 7.8: Predicted Dynamic Channels of Average Speed in NewYork

7.6 Error Analysis - Comparison of Outputs

In this section, we look at the average pixel error of the proposed models.

Shift	**Average Pixel Error**
Temporal Shift	20.42
Spatial Shift	54.57

TABLE 7.1: Comparsion of Outputs of Temporal and Spatial Shifts

Hence we can say that in temporal shift, we obtained better results than in spatial shift which is quite intuitive too. But we should also note that Multi-Task learning has done a pretty decent job in approximation of the pixel values given that the data is very sparse.

Chapter 8

Summary and Future Work

In this chapter, we present a summary of the system described in this project. We also discuss scope for further improvements and optimization that can be done in the current work along with the additional work that we propose in the future.

8.1 Summary

In this thesis, initially we proposed an optimal solution for short term traffic network flow prediction using the Convolutional LSTM model. We have also seen how Mean Square Error (MSE) converges with increase in epoch number. This can be shown to be better than both the standalone DCNN and LSTM models that capture only either the spatial or temporal correlations of the network. A video of the traffic flow network on separate days are also shown. We also proposed two trivial solutions - the constant zero and the moving average in order to get a feel of the dataset. As expected, the performance of both of them is quite poor. Adam's optimizer is used to train the sequence to sequence model. The GPU made available by Google Colab is used to train the model.

Finally we observe that the ConvLSTM model takes care of both the complex space and time correlations in any general traffic road network. This model can be fine tuned to make it more accurate and could be deployed for real time applications with the approval of some experts. Importance of future flow prediction of traffic is increasing day by day because of the increase in population and the necessity of social well being of people which further leads to exponential increase in traffic flow.

Then we proceed to capture the traffic flow conditions of cities through a U-Net architecture in order to tackle the problem of temporal shift. The input to the U-Net is provided by concatenating the static connectivity channels with the dynamic channels so that the road conditions would be captured. This needs to be done since the input data has inherent sparsity. Later we developed a multi-task learning framework to deal with spatio-temporal shift for predicting for entirely unseen cities. We have visualised the road networks, static and dynamic channels for training data and predictions as well.

In conclusion, U-Net with multi-task learning is a powerful technique for improving the efficiency and accuracy of machine learning models in the field of image segmentation or structured prediction. By leveraging the shared representation learned across multiple tasks, it can improve the accuracy of each task and mitigate overfitting by regularizing the shared representation. This approach has been successfully applied to a wide range of imaging applications and hence plays an increasingly important role in traffic network flow prediction.

8.2 Future Scope

The dataset we dealt with encoded only the speed, direction and volume of the road networks. But in general traffic flow depends on many other factors such as weekends, holidays, climate conditions and many more. So including all these

features in our dataset or any other similar techniques can help traffic forecasting to be quite accurate.

An end to end training model is developed here for the basic problem. Since the dataset is too large, we were able to train to our level best because of the limitations of Google Colab. To overcome this problem, parallel training approaches can be used.

Also during the development of this model, we had a very large training set. But this may not be the case always. We need to predict the traffic network flow as accurately as possible even when we have sparse data. This is one of the challenging problems in road network prediction. The problem of sparse data set can be handled by using simple interpolation techniques or synthetic sampling strategies using the SMOTEBOOST algorithms in case the samples are mostly non - overlapping. Techniques like Graph Signal Processing can be used.

For the advanced problem statement, We may compare the outcomes of our generated models with those of cutting-edge spatio-temporal learning models and more complex U-Net architecture. We may also test out a multi-task learning strategy that optimises for each city.

Bibliography

[1] Bui, K.-H. N., Cho, J., and Yi, H. (2022). Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Applied Intelligence*, 52(3):2763–2774.

[2] Dong, H., Jia, L., Sun, X., Li, C., and Qin, Y. (2009). Road traffic flow prediction with a time-oriented arima model. In *2009 Fifth International Joint Conference on INC, IMS and IDC*, pages 1649–1652. IEEE.

[3] Huang, D.-R., Song, J., Wang, D.-C., Cao, J.-Q., and Li, W. (2006). Forecasting model of traffic flow based on arma and wavelet transform. *Jisuanji Gongcheng yu Yingyong(Computer Engineering and Applications)*, 42(36):191–194.

[4] Kashyap, A. A., Raviraj, S., Devarakonda, A., Nayak K, S. R., KV, S., and Bhat, S. J. (2022). Traffic flow prediction models–a review of deep learning techniques. *Cogent Engineering*, 9(1):2010510.

[5] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[6] Martin, H., Bucher, D., Hong, Y., Buffat, R., Rupprecht, C., and Raubal, M. (2020). Graph-resnets for short-term traffic forecasts in almost unknown cities. In *NeurIPS 2019 Competition and Demonstration Track*, pages 153–163. PMLR.

[7] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

[8] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer.

[9] Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

[10] Sadeghi-Niaraki, A., Mirshafiei, P., Shakeri, M., and Choi, S.-M. (2020). Short-term traffic flow prediction using the modified elman recurrent neural network optimized through a genetic algorithm. *IEEE Access*, 8:217526–217540.

[11] Wang, M. and Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153.

[12] Wu, Y. and He, K. (2018). Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19.

[13] Yang, W., Yang, D., Zhao, Y., and Gong, J. (2010). Traffic flow prediction based on wavelet transform and radial basis function network. In *2010 International Conference on Logistics Systems and Intelligent Management (ICLSIM)*, volume 2, pages 969–972. IEEE.

[14] Yang, Y. and Hospedales, T. M. (2014). A unified perspective on multi-domain and multi-task learning. *arXiv preprint arXiv:1412.7489*.

[15] Yi, H., Jung, H., and Bae, S. (2017). Deep neural networks for traffic flow prediction. In *2017 IEEE international conference on big data and smart computing (BigComp)*, pages 328–331. IEEE.

[16] Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., and Yin, B. (2021). Deep learning on traffic prediction: Methods, analysis and future directions. *IEEE Transactions on Intelligent Transportation Systems*.

[17] Zhang, Y., Zhou, Y., Lu, H., and Fujita, H. (2020). Traffic network flow prediction using parallel training for deep convolutional neural networks on spark cloud. *IEEE Transactions on Industrial Informatics*, 16(12):7369–7380.

[18] Zhou, F., Chaib-draa, B., and Wang, B. (2021). Multi-task learning by leveraging the semantic information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11088–11096.