Traffic Nowcasting using Deep Learning

Kaustubh Kasture[1], Kinjalk Kamble[2], Vanshika Pandey[3], Prof. (Jaya) Dr. Ayesha Butalia[4]
[1], [2], [3]Student of department of IT Engineering, [4]Professor of department of IT Engineering
MIT-Art Design and Technology University,
Pune, Maharashtra, India

Abstract:- Nowcasting is the prediction of the present and the very near future of an indicator. Traffic Nowcasting is the prediction of various traffic factors occurring in the near future. This paper describes our approach, to predict short term city-wide high resolution traffic states with the static and dynamic information provided. We achieve this by utilizing the U-Net architecture to build a deep CNN model; test it on different cities to evaluate accuracies; and average the results at the end. With this, the model is better trained and will return more accurate, generalized results for different cities. The models are trained on traffic datasets provided by Traffic4Cast 2021 challenge. Thus, the aim of this project is to build a system for predicting and calculating traffic flow volume, heading and speed on a high-resolution whole city map by predicting the traffic states in the near future. The system could be utilized in a Traffic Prediction System and its scope entirely lies within traffic prediction.

Keywords:- *Traffic Nowcasting, CNN, APU-Net, Deep Learning.*

I. INTRODUCTION

As the rate of urbanization increases in the current era, adoption of vehicles is increasing in most countries. This inadvertently causes an increase in traffic. Various methods ranging from traffic signals to traffic monitoring systems are used to evaluate and keep check on traffic. We have developed this system as an addition to current traffic monitoring systems, to aid in prediction and evaluation of traffic in selected cities. The aim of this project is to predict future traffic flow volume, heading and speed on a high-resolution whole city map in which the dataset contains 100 billion probe points covering many cities throughout a year. The basic architecture used is U-Net; various alterations are done in each block of U-Net where various permutations and combinations of layers are used to construct different models to train and test on different cities and their combinations. We predict a future traffic map having traffic volume and speed information per each pixel, representing one hundred square meter local area. Prediction is done on each pixel of the high-resolution map to construct a future traffic map. U-Net produces an output prediction having an image dimension equal to the input image, and is a popular and effective method for problem settings of this kind. Given eight worldwide cities as train data, we train various U-Net models which differ in model structure or train set composition. Then we average predictions from all single models to present a single error metric for the problem statement.

II. LITERATURE SURVEY

A. U-Net : Convolutional Networks for Biomedical Image Segmentation

In this paper, a deep learning network with an architecture that consists of a contracting path to capture context and a symmetric expanding path to precise localization is devised. The paper demonstrates that U-net can be used where train images are few, while also outperforming other state of the art networks. We have used this U-net architecture for our traffic prediction purposes, adding and testing different types of layers to get better accuracies.

B. Traffic map prediction using UNet based deep convolutional neural network

This paper describes a U-Net based deep convolutional neural network approach on the Traffic4cast challenge 2019. This approach concentrates on evaluating data of individual cities through different structures of U-net model blocks and achieves lowest MSE in Traffic4cast 2019 challenge and achieves first place. We have used the above approach as it has won the Traffic4cast 2019 challenge and performed adequately in Traffic4cast 2020 challenge. This approach lets us build models by configuring the layers and epochs to get the same accuracy for less train time.

C. Fully Convolutional Networks for Semantic Segmentation

Explains how convolutional neural networks can learn dense predictions for per-pixel tasks like semantic segmentation and introduction of skip connections in fully convolutional networks. Above paper is used by us for reference for tuning our convolutional layers in our model and referred to its optimisation recommendations such as batch size and learning rate to get better results. The authors had used an Nvidia Tesla K40c for training their models but we have used an Nvidia 3070 mobile GPU making it easier for us to train our model on a bigger dataset with a larger batch size.

D. *A disciplined approach to neural network hyper-parameters : Part 1 - learning, rate, batch size, momentum and weight decay*

This paper proposes several efficient ways to optimize neural networks through hyper-parameters that aids in reducing training time and improves performance of neural networks. It examines training validation and test loss function to evaluate underfitting and overfitting and suggests guidelines to find optimal balance point.

III. METHODOLOGY

A. *Basic Task*

Traffic data provided by Traffic4cast has dimensions of 495x436 i.e., height x width. The core data includes dynamic features, encoding traffic volume, direction, and incidents, and static features, describing the road junctions and points of interest, such as food and drink, shopping, parking, transport, etc. Each pixel in image provided represents a one hundred square meter area. The dataset of a city captured has images which represent a time interval of 5 mins. Given a 12-timeframe traffic map image which represents one-hour long traffic map of cities, our goal is to predict next hour i.e., 6th timeframe, and evaluate the accuracies over different types of models trained.

B. *Input Preprocessing*

Input is composed of dynamic input data which changes over time, and static input data which is constant.

Dynamic input data is a tensor whose shape is (12, 495, 436, 8). In this tensor the first column represents the time bin, second represents the height of image, third represents width of the image and the last column represents the feature channel. Here the eight feature channels represent traffic volume and speed of four headings: northeast, northwest, southeast and southwest. Each channel is normalized and discretized to range from minimum 0 to maximum 255.

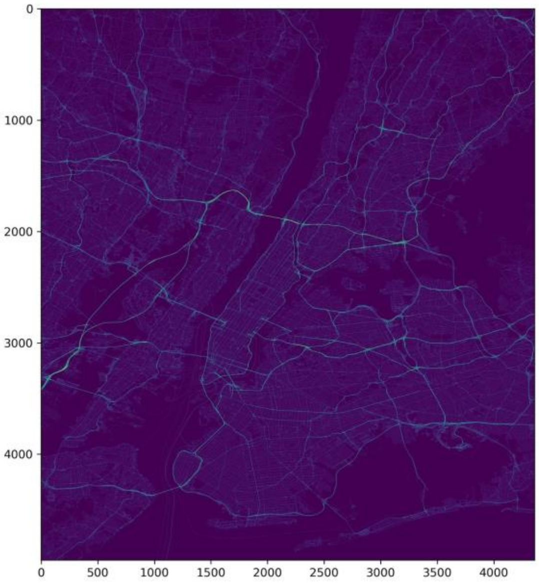

Fig 1:- Dynamic data of New-York

Our approach is similar to the previous approaches of Traffic4cast implementations, where we combine the time bin column with the feature channel, so the input data is transformed into a tensor shape (495, 436, 96).Static input data is tensor whose shape is (495, 436, 9). Static input data represents properties of the road maps, which remain consistent to time change. After appending this static input data to dynamic input data, we have input tensor shaped (496,436,105).

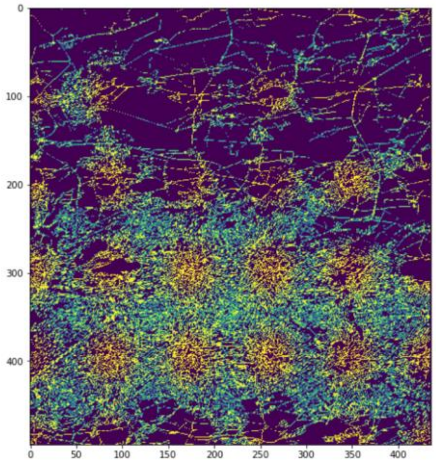

Fig 2:- Static input data of Berlin

C. *Output Sizes*

Prediction output has dimensions of (6, 495, 436, 8). First column represents the six-time bins i.e., future 5, 10, 15, 30, 45, 60 mins and the last column represents traffic volume and speed for four headings.

IV. MODEL ARCHITECTURE

We construct a model based on U-Net using Pytorch, which has an encoding and decoding phase with skip connections. Encoding phase, also known as the contraction path, is used to capture the context of the image. The encoder phase is a stack of convolutional and max pooling layers. The decoding phase also known as expanding path is used to achieve precise localization using transposed convolution. The decoder phase consists of deconvolutional layers but does not contain dense layers due to which it can accept an image of any size. Thus, making it a fully convolutional network.

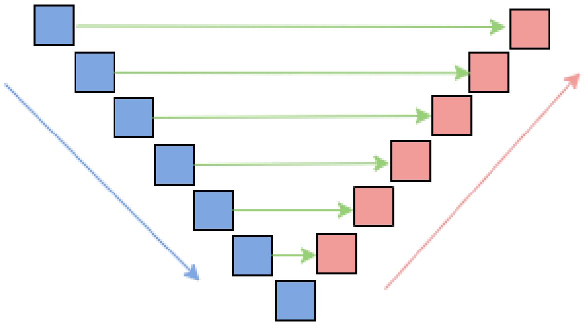

Fig 3:- Blue boxes are dense convolution layers and pink boxes are deconvolution layers. Green arrows are skip connections.

Loss function used for Traffic4cast challenge is Mean Squared Error (MSE), so we stick to it as an accuracy measure, and use Adam optimizer with learning rate of 3e-4. Further optimization is done according to the needs of specific models. Experimentation with various model structures for our approach is as follows.

D. Model A

Each down sampling convolutional block consists of two convolutional layers densely connected with each other in a feed forward method. Between each down sampling block, average pooling is used to reduce image size in half. Since it uses an average pooling layer, we refer to this model as APU-net.

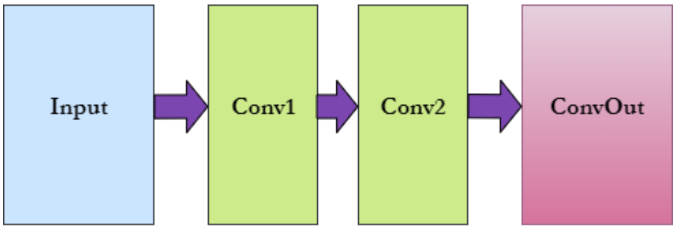

Fig 4:- Down-block structure of Model A: Inside the down-block of Model A, two convolutional layers are densely connected with each other.

E. Model B

Structure is similar to Model A, but a different scale factor is used instead of average pooling. In Model A, image size is halved so it is setting scale factor as 0.5. In Model B, we set it to 0.7, so image size is reduced gradually compared to Model A while going through each down sampling block.

F. Model C

Structure is similar to Model A, but a max pooling layer is added in the middle of the two convolutional layers. Model A did not consist of a max pooling layer.

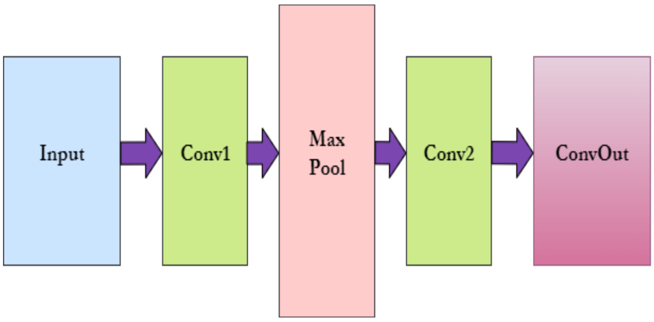

Fig 5:- Down-block structure of Model C: Two convolutional layers and one max pooling layer are densely connected with each other.

We use these different structures of blocks to train and test prediction on individual cities, which include Berlin, Chicago, Istanbul, Melbourne, Moscow, New York, Bangkok, Antwerp, Barcelona and Vienna. Four models are trained with type A structure on 4 individual cities each to induce specificity on each model and later, to compare the results with the generalized models and check their accuracies. We then train one model with the Type B structure to generalize the model, observe its workings when trained on different datasets together and compare results with the specificity-oriented models. We finally train one Type C structure model on all available datasets and try to predict results on a previous, unseen dataset.

Model	Description
Model 1	Model A used Model is trained on same city which needs to be predicted City-Berlin
Model 2	Model A used Model is trained on same city which needs to be predicted City-Chicago
Model 3	Model A used Model is trained on same city which needs to be predicted City-Istanbul
Model 4	Model A used Model is trained on same city which needs to be predicted City-Melbourne
Model 5	Model B used Model is trained on 4 cities combined. Prediction is one of the 4 cities
Model 6	Model C is used Model is trained on 8 cities combined Prediction city is outside the train set

Table 1:- Task Base Models

V. RESULTS AND EXPERIMENTATION

All the models' performance is measured by mean squared error ranging from 69.13 to 85.18 according to the test set evaluation. Our MSE is close to the mean MSE of Traffic4cast 2021 data, this is better, considering we have trained the model using smaller mini-batches and with less epochs compared to other models presented in Traffic4cast 2021 leaderboard. Training from a train set having other cities combined with the one to be predicted results in slightly higher MSE compared to MSE with training from only target city's traffic data but results in greater generalization.

ISSN No:-2456-2165

Model	MSE
Model 1	69.13
Model 2	69.74
Model 3	69.38
Model 4	69.82
Model 5	72.41
Model 6	85.18
Average MSE	72.61

Table 2:- Task evaluation results

Below table shows the shape of each block in our U-net model where input has 496 height and 436 width and downsizes it to image with height and width 4, and then it upsizes height to original inputs which is prediction of the model.

Block	Output Shape
Dense Block No-1	(495, 436, 64)
Average Pooling Layer	(248, 218, 64)
Dense Block No.-2	(248, 218, 96)
Average Pooling Layer	(124, 109, 96)
Dense Block No.-3	(124, 109, 128)
…	
Dense Block No.-7	(8, 7, 128)
Average Pooling Layer	(4, 4, 128)
Dense Block No-8	(4, 4, 128)
Convolution Layer	(4, 4, 128)
Deconvolution Block No-1	(8, 7, 128)
Deconvolution Block No.-2	(16, 14, 128)
…	
Deconvolution Block No-7	(495, 436, 128)
Convolution Layer	(495, 436, 9)

Table 3:- Output Shape per block

VI. CONCLUSION AND FUTURE SCOPE

We have created a deep learning model which is designed to help in traffic monitoring systems. We have utilized U-Net based deep convolutional neural networks to predict future traffic flow volume, speed and direction of different cities.

Our APU-net based approach effectively predicts high resolution future traffic maps when the model is trained on the same city data on which it is to be predicted on. When we use different scaling factors in our model and train it on data of different cities and predict in one of them then error rate increases slightly, but this model is able to generalize well and the number of cities which can be predicted on increases. When trained on a wide number of cities with a max pooling layer, our model is able to predict future traffic maps of cities outside the test dataset, however MSE increases.

APU-net based models trained on a single city can be deployed along with traffic monitoring systems to get quick real-world predictions of traffic conditions for that particular city. However generalized models which predict on different cities inside and outside the dataset need to be trained more on larger numbers of cities. Other possible solutions could be to train models on a dataset which have similar city styles and structures and test on cities contained in the dataset. Real world application of these methods can contribute to building accurate traffic forecast systems or highly effective navigational systems.

ACKNOWLEDGMENT

We would like to thank **Prof. Palash Sontakke**, Project Coordinator, Department of Information Technology, for extending all support during the execution of the project work.

We sincerely thank **Prof. Dr. Rekha Sugandhi**, Head, Department of Information Technology, MIT School of Engineering, MIT-ADT University, Pune, for providing the necessary facilities in completing the project.

We are grateful to **Prof. Dr. Kishore Ravande**, Principal, MIT School of Engineering, MIT-ADT University, Pune, for providing the facilities to carry out our project work.

We also thank all the faculty members in the Department for their support and advice.

REFERENCES

[1]. Traffic4cast– Traffic Map Movie Forecasting 2021 https://www.iarai.ac.at/traffic4cast/2021-competition//

[2]. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation" International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[3]. Jonathan Long, Evan Shelhamer, Trevor Darrell, “Fully Convolutional Networks for Semantic Segmentation” arXiv:1411.4038

[4]. Leslie N Smith. “A disciplined approach to neural network hyper-parameters: Part 1 - learning, rate, batch size, momentum and weight decay.” arXix:1803.09820.

[5]. Huang, Gao, et al. "Densely connected convolutional networks." in Proceedings of the *IEEE conference on computer vision and pattern recognition.* 2017.

[6]. A. Butalia, M. L Dhore, “Rough Sets as a Framework for Data Mining,” International Conference IMECS (IAENG), Hongkong, March 22-23, 2007

[7]. Paszke, Gross, Massa, Lerer, et al. "PyTorch: An Imperative Style High Performance Deep Learning Library." arXiv:1912.01703.

[8]. A. Butalia, M. L Dhore, G. Tewani, “Application of Rough Sets in the field of data mining,” in Proceedings of IEEE *International Conference on Emerging Trends in Engineering and Technology (ICETET ’08)*, pp. 498-503, July 16-18 2008.

[9]. A. Butalia, V. Suryawanshi “Distributed and Parallel Systems,” in Proceedings of *National Conference on Emerging Technologies and Applications ETA-2005,* Computer Science Department of Saurashtra University, Rajkot jointly with Amoghsiddhi Education Society, Sangli, October 1-2, 2005.

[10]. Kingma, D. P., & Ba, J. “Adam: A method for stochastic optimization.”
arXiv:1412.6980. 2014.

[11]. Li, Xiaomeng, et al. "H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes." IEEE transactions on medical imaging 37.12 (2018): 2663-2674.

[12]. Zhou, Yongjin, et al. "D-UNet: a dimension-fusion U shape network for chronic stroke lesion segmentation." IEEE/ACM transactions on computational biology and bioinformatics (2019).

[13]. Qamar, Saqib, et al. "A variant form of 3D-UNet for infant brain segmentation." Future Generation Computer Systems 108 (2020): 613-623.

[14]. Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." In Proceedings of the *IEEE conference on computer vision and pattern recognition*, pp. 3431-3440. 2015