Grenze International Journal of Engineering and Technology, Jan Issue

Intelligent Video Analytics for Better Planning of Smart Cities using Edge Computing

Savita Shetty[1], Raghav Maheshwari[2], Rithika Mehta[3] and Shirsh Vardhan Kashyap[4]
[1-4]Ramaiah Institute Of Technology , Visvesvaraya Technological University, Belagavi Bangalore, India
Email: savita_ks1@msrit.edu, raghav.ddps2@gmail.com, mehtarithika@gmail.com, shirshvardhan@gmail.com

***Abstract*—Recent years have shown how machine learning and deep learning techniques can be used to radically transform smart cities. However, current cloud-centric processing approaches present latency, privacy, and bandwidth-related challenges. Edge computing is a solution to the above challenges where the video feeds captured can be analyzed right at the source of generation of the feed while deeper insights and analytics can still be performed on the cloud. The edge computing approach to video analytics also enables us to take action to certain situations in real-time which is not possible in cloud-centric approaches.**
In this paper, we propose certain methods for better planning of smart cities using edge computing. The data points generated for the proposed methods are based on 2 use cases - traffic counting using Nvidia DetectNet and pedestrian face mask detection using MobileNetV2. We also discuss how video frame resizing can be used for enhanced processing on the edge device.

***Index Terms*— Edge Computing, Machine Learning, Smart Cities.**

I. INTRODUCTION

According to Wikipedia, “A **smart city** is an urban area that uses different types of electronic methods and sensors to collect data. Insights gained from that data are used to manage assets, resources, and services efficiently; in return, that data is used to improve the operations across the city [1].
Smart cities usually have a lot of traffic camera deployments throughout the city. The footage from these traffic cameras is used to gather insights for better planning of smart cities. However, there are 3 major issues to video analytics performed at the centralized server. They are Network Bandwidth, latency in real-time situations, and citizen data privacy [2]. Processing the videos at the centralized server requires huge bandwidth to transfer the feeds to the cloud and is in fact not real-time which limits the benefits of the analysis. Moreover, In today’s connected world, people are more and more concerned about their privacy and do not prefer the idea of centralized analysis.
Edge computing is proposed as an alternative to traditional cloud-based processing which helps in resolving the above challenges. **Edge Computing** is a new distributed Cloud Computing paradigm that describes a decentralized way of computing and storage at the topological edge of the network, i.e. in physical proximity to the data sourcing device. In contrast to the Cloud Computing paradigm, where the collected data is fully transmitted to a central server before being analyzed and used, Edge Computing allows to do these tasks on devices or close to it [4].

Grenze ID: 01.GIJET.8.1.15

Smart cities are already equipped with CCTV Cameras. All they need to upgrade is to attach an edge device that can perform video analysis right at the source of generation of data.
In this paper, we present 2 use cases that can be used for better planning of smart cities

1. Traffic count includes Vehicle counting, Crowd counting, etc.
2. Pedestrian Face Mask detection

Both the above data points can be used for better planning of smart cities which is described in the paper.

II. Background

A. Traffic Regulation Scenario

Nowadays, smart cities have become large information collection centers [22], where raw data like traffic patterns, environmental patterns, etc. is being gathered from a large number of sensors deployed throughout the city's infrastructure. Existing efforts in traffic video surveillance are processed and analyzed by a human operator manually watching data feeds and selecting the clips where an event occurs. However, the effectiveness of the operator is maximum only for a few minutes, and the chances of making errors increase when the focus is lost.
With the advancement in the field of machine and deep learning, Intelligent Video Analytics (IVS) systems[23] are proposed which focus on analyzing the traffic video feed using machine learning without any human efforts. However, This analysis uses a centralized architecture where all the feed from the cameras is fed to the cloud and then processed. This type of analysis is time-consuming and takes an enormous amount of memory space on the cloud as the data being transferred is in terabytes. Moreover, the post-analysis is not very useful in situations where real-time action is required, for example - Accident detection near traffic signals.

B. Edge Computing

Edge computing is a new paradigm that allows computing right at the source where data is generated. It is a decentralized way of computing that is extremely helpful in use cases that require just-in-time analysis. The difference between traditional cloud processing and edge processing is that, in the case of centralized cloud-based processing, all the data collected is sent to centralized storage before analysis, whereas edge computing allows the analysis somewhere close to the source of generation of data.
The usage of edge computing is crucial for the smart cities of the future. Various edge implementations like Fog computing, cloudlet, and mobile edge computing are used to bring the exciting features of the cloud-based approach to the edge.

III. Related Work

In this section, we mainly focus on the research works in the area of traffic surveillance in Edge Computing.
In [3], the author highlights the importance of edge computing for the smart cities of the future. This survey evaluates, in-depth, the need for the edge computing paradigm and also highlights the challenges in the deployment of edge-based solutions. In [4], the authors present a detailed survey of current initiatives in the domain of edge computing and also present a roadmap for sustainable edge computing development. The roadmap discusses how edge computing concerns like security issues, real-time capabilities affect the social, ecological, and economical dimensions of sustainability.
In [5], we can see how a hybrid of cloud computing and edge computing is used for cloud monitoring applications. The approach used was ResNet50 and the investigation showed how the performance of deep learning models for crowd counting is highly dependent on the video size. In [2], The article presents a detailed overview as to how deep learning can be used for various applications at the edge of the network. It also discusses applying deep learning techniques at the edge to smart cities.
In [6], Deep learning techniques are discussed with reference to large-scale IoT deployments. This paper focuses on distributing the model pipeline across edge and cloud resources. In this approach, the basic processing is focused towards the edge while the compute-intensive tasks are performed on the cloud. Results from this paper suggest that the efficiency of the system increases if a hybrid architecture (edge, in-transit, and cloud) is used as compared to a centralized cloud-based architecture.
In [24], a similar distributed deep learning training model is discussed which focuses on Distributed Video Surveillance using deep learning techniques. This system is deployed in an edge computing environment.
In [7], A similar paradigm to edge computing is fog computing and its pipeline design is discussed. It addresses the challenges in the traditional traffic data storage and analysis and suggests better solutions to the same.

In [8], Microsoft research presents a killer application for edge computing with three major aspects. It presents an intelligent hybrid architecture that is used for load sharing on the cloud and the edge, highlights resource accuracy tradeoff while dealing with edge applications, and also presents an interactive query mechanism on the video frames for after the fact analysis. In [9] A smart visual sensor is developed for a research project in Liverpool. The aim of the project was to design an edge computing infrastructure that can be used for tracking multi-modal traffic in real-time while safeguarding citizens' privacy at the same time.

IV. System Description

The proposed system uses an Nvidia Jetson[18] with a Raspberry Pi Camera. The Pi camera is analogous to the CCTV cameras in Smart Cities.

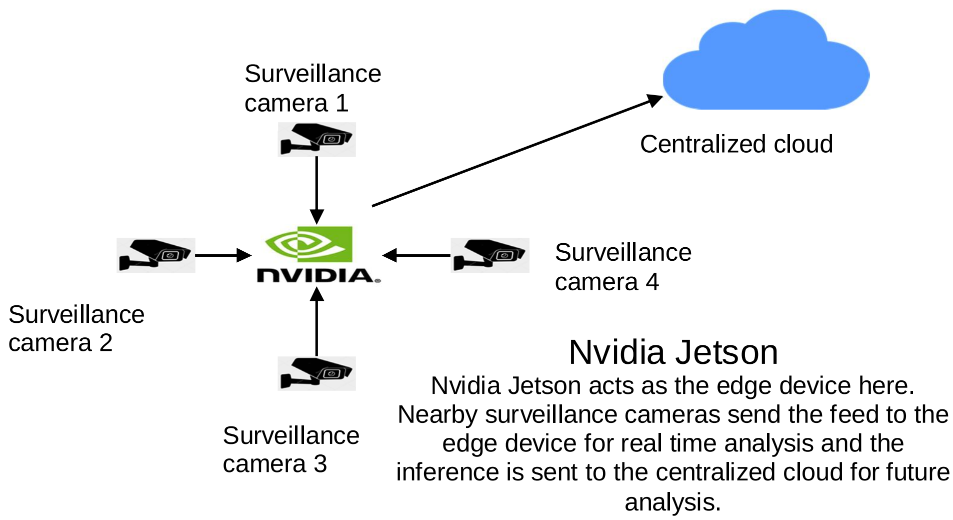

Fig 1. The basic architecture of the proposed system

The frames in the video stream captured by the surveillance cameras are processed by the Edge device. The Edge devices perform two types of processing on the captured frames. i) It does the traffic analysis on the frame and sends the analysis of the vehicle count and the crowd count to the cloud. ii) If any faces are detected in the frame, the face mask model tries to identify whether the pedestrian is wearing a mask or not and whether social distancing norms are followed or not.

The data points are captured and sent to the cloud to be used for better planning of smart cities. The various software used in the design are as follows:

OpenCV: This is a library that has several built-in functions for performing computer vision and machine learning tasks [10].

Keras MobileNetV2: Keras is an open-source neural network (NN) library that provides a high-level wrapper to Tensorflow [11]. MobileNetV2[19] is a lightweight convolutional Neural Network that reduces the inference time on mobile and embedded devices.

Nvidia DetectNet: It is a Deep Neural Network that is used for Object Detection. In a detect net, A fully convolutional neural network performs feature extraction and prediction of object classes and bounding boxes per grid square. The Loss functions simultaneously measure the error in the two tasks of predicting the object coverage and object bounding box corners per grid square [12].

Working flow of the proposed system: The proposed system employs two models, one is to count the vehicles, crowd, and the other is to detect whether pedestrians are wearing face masks or not. The first model uses Nvidia Detectnet for object detection and the second uses a pre-trained facenet to detect faces in the frame and then a MobileNetV2[19] to identify whether the pedestrian is wearing a mask or not. The inference from the frames, i.e. Traffic counts per hour and People wearing or not wearing masks are sent to the cloud. Now, over a period of time, this historical data is used for better and efficient police personnel allocation, efficient road planning based on the traffic data, etc.

V. IMPLEMENTATION

A. Traffic counting Algorithm

Nvidia Detectnet: The Training and Validation specifications for the DetectNet architecture are described as follows [12].

Training Specification [12]

i. The data layers take the images for training and a transformer layer is used for Image Data Augmentation[28].
ii. Feature extraction is performed and object classes are predicted using a Fully convolutional network.
iii. Loss functions are used to measure the error in the task of object detection and object bounding.

Validation Specification [12]:

i. During validation, a clustering function is used to produce a set of bounding boxes.
ii. Mean Average Precision is used as the metric to measure the model's performance with respect to the validation dataset.

Process Steps

i. The frame is passed through the Nvidia DetectNet architecture and the traffic is classified in the frame.
ii. The inference generated at the edge is sent to the cloud in a Time Series Database[26] over regular intervals of time.

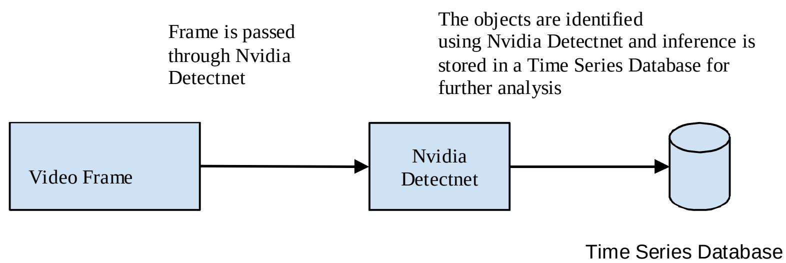

Fig 2. Traffic Counting Algorithm

B. Mask Detection Algorithm

Dataset[27]

The mask detection model is trained with around 4000 images in which 2000 images are classified as "with_mask' and the remaining 2000 images are classified as "without_mask'. Data Augmentation[28] is applied to the training set before training the model to generate more data for training.

Mask Detection Algorithm Training Steps

Load the images from the dataset using the *load_img()* from the Keras API. The images are resized to a shape of 224 X 224.
i. The images are preprocessed using the *preprocess_input* function.
ii. The labels are converted to categorical variables using the *to_categorical* Keras API.
iii. Image Augmentation is performed using the ImageDataGenerator from Keras [11].
iv. The dataset is partitioned into training and testing using the *train_test_split* function.
v. Instantiate the MobileNetV2 model with the weights from the ImageNet Dataset[20]. Transfer learning [17] is used here to use knowledge from the ImageNet Dataset for custom mask detection models.
vi. A Max Pooling Layer and a Fully connected layer are used on top of this base model.
vii. The fully connected layer is connected to a 2-output layer which is responsible for the with/without_mask classification.
viii. Adam Optimizer[25] is used to compile the model and binary cross-entropy is used as the loss function.
ix. The trained model is now used for real-time inference on the edge device.

Process Steps

i. The captured video frame is passed through the FaceNet[13] model to detect faces in the frame.
ii. Each of the detected faces is then passed through the MobileNet V2 mask detection model.

iii. The inference generated at the edge is sent to the cloud in a Time Series Database[26] over regular intervals of time.

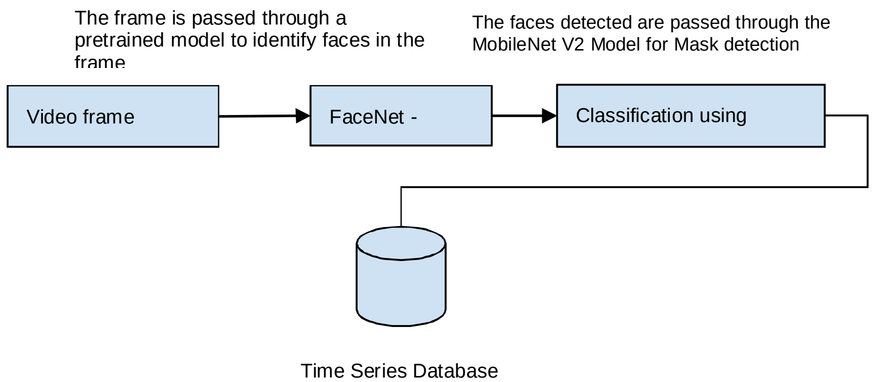

Fig 3. Mask detection algorithm

VI. Results

The detect net used for object detection has good performance for identifying the object in the frames. It identifies objects like buses, pedestrians, cars, trucks, etc. with good accuracy. The detected objects over a period of time are stored in a Time Series Database[26]. The mask detection algorithm complexity is relatively high as compared to the detect net and the algorithm was able to analyze the frames in real-time on the edge device. The same algorithm on standard computing hardware is capable of analyzing twice the frames per second. The data captured using the Mask detection model is also stored in a Time Series Database[26] over a period of time.
It is also observed that downscaling input data results in an improvement in the performance of the model. Interestingly, the model presents a tradeoff between accuracy and runtime if the frame size is modified. A reduction in the size of the input frame results in an increase in the number of frames analyzed per second.

VII. Discussion

Edge-powered Video processing provides many advantages like less bandwidth usage, Real-time analysis, and Increased Data privacy [2]. Over the past years, the major limitation for edge computing to be used in real-time video analytics use cases was the computational complexity of the image processing models[21].
However, with recent advancements and by maintaining an optimal balance between the cloud and edge processing [14], edge computing can be used in a variety of use cases particularly for better planning of smart cities. The paragraphs below highlight some of the use cases where the inference gathered from the results of this paper can be used for better planning of smart cities.
The traffic count/pedestrian data collected at the edge over a period of time can be used for real-time traffic prediction at the centralized cloud and hence improving citizen experience in smart cities.
The traffic vehicle type count can be used for better planning of roads and the entire transport system can be optimized using the data points generated over a period of time.
The insights from the collected data can be used for optimal traffic police personnel allocation and the data collected from the real-time mask detection can be used to check the status of social distancing norms in smart cities.
The key to successful edge deployments lies in a good balance between edge and cloud processing [14]. For smart cities, regular real-time frame processing can be collected at the edge and other machine learning models can be deployed on the cloud to identify the traffic and social distancing trends over a period of time.
Processing the frames at the edge does not require processing at the cloud and hence can be deployed in areas with low internet bandwidth, reducing the cost for cloud deployments. Additionally, The raw data generated by CCTV’s is highly sensitive and edge computing ensures the anonymity of the data captured [29].

VIII. Conclusion and Future Work

In early approaches to video analytics for better planning of smart cities, the cloud-centric approaches were prevalent. The design has shown that the cloud-centric approach has a lot of limitations and hence a relatively new approach has emerged which looks for a balance between cloud and edge processing where less compute-intensive tasks are performed at the edge while critical tasks are performed at the centralized cloud. [14].

In this paper, we used Nvidia Jetson as the edge device for our edge processing. Edge enhanced models perform very well on the edge device and are nearly capable of processing one frame per second and most traffic cameras capture one image per second which makes the edge computing approach appropriate for real-time traffic and cloud monitoring applications. The inference captured at the edge is periodically sent to the cloud where deeper insights are obtained by analyzing the captured data over a period of time. These insights are critical in better planning of cities and efficient allocation of police personnel for better surveillance.

Future work includes identifying effective ways to handle the security issues and Hardware Interoperability issues that arise with the edge computing approach [15].

Workload allocation is another major issue that comes with edge computing as we have multiple layers with different computational capabilities. Hence, various factors like latency, bandwidth, energy, and cost are critical in identifying optimal allocation strategies [16]. In our use case, footage analysis was performed at the edge while future traffic prediction was performed at the centralized cloud.

Edge processing enables us to make critical decisions in near real-time and also provides better data privacy as privacy-sensitive frames can be anonymized and inference can be captured. Hence, the approach of edge analytics gives us all the critical data points required for better planning of smart cities while maintaining minimum latency, improved privacy, and less bandwidth usage which is the best of all the worlds.

REFERENCES

[1] https://en.wikipedia.org/wiki/Smart_city.

[2] Jiasi Chen and Xukan Ran, Deep Learning With Edge Computing: A Review.

[3] Latif U. Khan, Ibrar Yaqoob, Nguyen H. Tran,, S. M. Ahsan Kazmi, Tri Nguyen Dang, Choong Seon Hong, Edge-Computing-Enabled Smart Cities: AComprehensive Survey.

[4] Andrea Hamm, Alexander Willner, Ina Schieferdecker, Edge Computing: A Comprehensive Survey of Current Initiatives and a Roadmap for a Sustainable Edge Computing Development.

[5] Camille Ballas, Mark Marsden, Dian Zhang, Noel E. O'Connor, and Suzanne Little Performance of video processing at the edge for crowd-monitoring applications.

[6] M.Ali, A.Anjum, M.U Yaseen A.R. Zamani, D.Balouek-Thomert, O.Rana and M.Parashar, Edge Enhanced Deep Learning System for Large-scale Video Stream Analytics.

[7] Elarbi Badidi, Zineb Mahrez, and Essaid Sabir, Fog Computing for Smart Cities' Big Data Management and Analytics: A Review

[8] Ganesh Ananthanarayanan, Victor Bahl, Landon Cox, Alex Crown, Shadi Nogbahi, Yuanchao Shu, Microsoft Research, Demo: Video Analytics - Killer App for Edge Computing.

[9] Johan Barthélemy , Nicolas Verstaevel , Hugh Forehead and Pascal Perez, Edge-Computing Video Analytics for Real-Time Traffic Monitoring in a Smart City.

[10] OpenCV, https://docs.opencv.org/master/

[11] Keras API Docs - https://keras.io/

[12] Nvidia Detectnet Https://developer.nvidia.com/blog/detectnet-deep-neural-network-object-detection-digits/

[13] Florian Schroff, Dmitry Kalenichenko, James Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering.

[14] Zeinab Nezami, Kamran Zamanifar, Karim Djemame, Evangelos Pournaras, Decentralized Edge-to-Cloud Load-balancing: Service Placement for the Internet of Things.

[15] Shalin Parikh, Dharmin Dave, Reema Patel, Nishant Doshi, Security and Privacy Issues in Cloud, Fog and Edge Computing.

[16] Weishong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu "Edge Computing: Vision and Challenges" IEEE Internet of Things Journal, Vol. 3, No. 5, 2016, pp. 637-646.

[17] Abu Sufyan Aniruddha Ghosh Ali Safa Sadiq FlorentinSmarandache, A Survey on Deep Transfer Learning to Edge Computing for Mitigating the COVID-19 Pandemic.

[18] Nvidia Jetson Docs - https://docs.nvidia.com/jetson/

[19] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks

[20] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, DOI: 10.1109/CVPR.2009.5206848.

[21] Varghese, Blesson & Wang, Nan & Barbhuiya, Sakil & Kilpatrick, Peter & Nikolopoulos, Dimitrios. (2016). Challenges and Opportunities in Edge Computing. 10.1109/SmartCloud.2016.18.

[22] Chiehyeon Lima, Kwang-Jae Kim, Paul P. Maglio, Smart cities with big data: Reference models, challenges, and considerations.

[23] G. Sreenu and M. A. Saleem Durai, Intelligent video surveillance: a review through deep learning techniques for crowd analysis.
[24] Jianguo Chen, Kenli Li,Senior Member, IEEE, Qingying Deng, Distributed Deep Learning Model for IntelligentVideo Surveillance Systems with Edge Computing.
[25] Diederik P. Kingma, Jimmy Ba, Adam: A Method for Stochastic Optimization.
[26] Dmitry Namiot, Time Series Databases,https://www.researchgate.net/publication/286732446_Time_Series_Databases.
[27] Adnane Cabani, Karim Hammoudi, Halim Benhabiles, Mahmoud Melkemi, MaskedFace-Net -- A Dataset of Correctly/Incorrectly Masked Face Images in the Context of COVID-19
[28] Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. J Big Data **6,** 60 (2019). https://doi.org/10.1186/s40537-019-0197-0.
[29] Abdulmalik Alwarafy and Khaled A. Al-Thelaya and Mohamed Abdallah and Jens Schneider and Mounir Hamdi, A Survey on Security and Privacy Issues in Edge Computing-Assisted Internet of Things.