Real-Time Image Recognition and Video Analysis Using Deep Learning for Aurangabad Smart City

Syeda Firasatunnisa Begum[1], Priyanka S Shirsath[2], Saurabh Bansod[3]

{s.firasat23@gmail.com[1], shirsathpriyas@gmail.com[2], saurabhbansod@nielit.gov.in[3]}

National Institute of Electronics and Information Technology, Aurangabad 431001[1,2,3]

Abstract. Implementation of Urban IoT in Smart Cities of India has initiated the development of innovative heterogeneous systems to handle and integrate the enormous data traffic flowing on the networks[2]. The local governing bodies look for the most advanced communication technologies to support added-value services for good governance in Smart City. Using artificial intelligence in the public safety domain, the paper focuses on implementing real-time image recognition and video analysis using deep learning for Aurangabad Smart City, India.

Keywords: Smart City, India, Deep Learning, MTCNN.

1 Introduction

The paper is aimed to study the effect of digital intervention within Smart Cities in India [1],[13]. Urbanization has led to dense urban activities, causing an increase in pedestrian traffic, private vehicles, passenger carriers, and goods vehicles. Aurangabad is one of the nominated cities in India under the Smart Cities Mission [16]. Integration of various technologies aimed towards sustainable growth of Indian cities is one of the missions of Smart Cities[13]. The problem ahead for city planners, researchers, and local governing bodies is to try various models, to assist in forecasting, efficient decision making, defining strategy, and assessing the data accumulated from the 700+ CCTV network within the Aurangabad Smart City[17]. If we compare the Indian Smart City with those of the developed world, high population density and unorganized traffic are critical distinguishing factors. A child/ elder citizen getting lost in such a densely populated scenario is traumatic as it is difficult for the family to locate the person due to the absence of a social support

THEETAS 2022, April 16-17, Jabalpur, India

DOI 10.4108/eai.16-4-2022.2318072

system in the new urban towns of India[9]. In our case, image processing using Deep Learning Model, the Multi-task Cascaded Convolution Neural Network (MTCNN) was used on the data set collected from CCTV installed in the city for image identification. Image processing plays a salient role in object detection technology.

Fig.1. Population And Traffic Density On Roads Of Aurangabad City

2 Literature and Field Survey

As a part of a survey, the data from the Command Centre of Aurangabad Smart City Development Corporation Limited (ASCDCL), gathered from live streams from various crossing points, were analyzed. The traffic accidents lead to traffic congestion but cause loss of lives and damage the property. In a severe accident, the driver cannot call for help as the passersby pay no attention to the accident, especially on highways where other drivers will not stop. It results in delays in the victim's medical treatment. Implementing a robust accident detection system will alleviate the problems mentioned above[8],[17].

2.1 Field survey of cameras installed in Aurangabad Smart City

The field survey was undertaken to understand the type of cameras, the effect of local environmental conditions, and the output format. PTZ (Pan Tilt Zoom) camera and fixed, boxed IP cameras are installed. The image output at the analyst post is in jpeg, jpg, Mkv formats with 96 dpi x 96 dpi and a bit depth of 24 and resolution of 1920 X 1080. The live video has a resolution of 1920 x 1080. In the case of the PTZ camera, the focal length can be adjusted between 100 meters to 300 meters. The MTCNN algorithm is capable of adjusting the fpi based on the requirement from 5 to 30 fps[17]. The limitation of the installed cameras are:

a. Due to lower bandwidth/rains, the live stream has a blurring effect.
b. Direct sunlight on the aperture causes reflection/ bright light on the camera at a specific time during the day. Hence instances are not recorded at that specific time.
c. Performance is poor in low light conditions.
d. Manual analyst intervention is high: as the analyst has to adjust the live stream in the case of PTZ Camera, the option of zoom in or zoom out of prerecorded video is not available in the prerecorded video. Auto-scan mode is not activated[16].
e. Cameras fitted at corners; hence the view posts are diagonal and not perpendicular to the road.

Fig. 2. Customized Data Set for Face Recognition

2.2 Field survey for Vehicle collision within Aurangabad Smart City

In India, the traffic is heterogeneous. It comprises cars, buses, trucks, motorbikes, bicycles, tractors, etc. A Deep Learning-based system for extracting the vehicle's position to predict vehicle collision using a video stream from Command Center. As per the report of road accidents of Maharashtra, in 2018- 2019 has 12,788 deaths in Aurangabad. As per the report, 80% of the fatalities are triggered due to human error[23]. Tragically, the majority of all road accident fatalities in India impact the economically weaker sections of society. About 70% of fatalities constitute a combination of pedestrians, two-wheeler riders and cyclists. The accidents can be classified as A1- victim dies on the spot or within 24 hrs, A2- the victim is seriously injured or dies after 24 h and A3- no injuries occur, there is damage to property damage. It is observed that most fatalities occur between 6.00 pm to 9.30 pm. The max number of vehicles during this time is approximately 70 to 110 per km.

The survey reveals that two-wheeler drivers are vulnerable and predominantly men in the age group of 25 to 36 years[23][24]. The primary causes are rash driving, over-speeding, traffic signal violations ignoring seat belts, helmet rules, wrong side driving, dangerous overtaking, etc. The environmental risk factors are the defective layout of crossroads, narrow roads, use of low-quality material in roads, potholes, poor use of road furniture like road markings, etc. also contribute to road traffic accidents. Based on the field survey, the roads within Aurangabad City limits are classified as roads within the old city (Kilay Arak, Lal Masjid, and Gandhi Putla) and roads in the new city (University gate, Radha Krishna Mangal Karyala).

Fig. 3. Roads within Aurangabad Old and New City

3 Technical Solution

3.1 Data Set For Face Recognition Using Deep Learning

To load a customized dataset Open CV was used, classification data set was split into training data set of 420 individuals, testing data set of 105 images and validation using 5 images. The format of the file can be JPEG/JPG/PNG etc.; it took 20 days to train the model. The accuracy achieved was 89 % on this customized data set. The images were converted to Numpy array with float32 as the data type. The image array is normalized to scaled-down between 0 and 1 from 0 to 255 for a similar data distribution for faster convergence.

3.2 Data Set For Vehicle Collision Using Deep Learning

Deep Neural Network finds application in extracting the trajectory of moving vehicles based on the present position of the vehicle from input video data stream of vehicular traffic. Input from

video/images from data centers in mp4 /Mkv /PNG(images) format . The data size 4 GB Indian Roads Data set was used for training, testing and validation. The training time on the customized data set was 28 days (approx.) The incidence of collision is rare and subjected to police investigation. Hence the availability of such data is difficult. Canny edge detection algorithm compensates for less volume of collision data as it can process a single high-resolution image with an accuracy of 75% in our case with a detection Time of 10 ms.

3.3 Model Description for Face Recognition

As the deep learning model varies depending on the conditional stages of the models, the primary consideration would be providing efficient and most accurate detection on the model. The initialization process for face recognition is followed in a staged manner[9].

3.3.1 Face Detection

a. *Mode Process* – In this, the model is loaded with the pre-trained state of art detection interference.
b. *Color Segmentation* – The segmentation and augmentation preprocessing is followed for the image to provide certain filtering and masking techniques.
c. *Post Processing* – In this stage, the processed model is used to predict the class order defined with the localization bounding boxes and align the region of the area as selected as per the person in the given area.
d. *Template matching* – It is the final stage, it is used to verify whether the template of the bounding box is matched with the given class of the person.

3.3.2 Face Recognition

a. *Preprocessing* – This stage processes the images uses techniques for resizing, filtering and transforming the image into numerical data from the image.
b. *Model* – The model is evaluated with the set of input and outcome in the image to text format or image to detected image format.
c. *Encoder* – The model process is then fetched with the encoder to provide the numerical data to the encoded format for the transformation of a neural network to carry out numerical processing.
d. *Classifier* – In this stage, the classifier classifies the numerical data and set of classes. It

defines the input, which is further highlighted for reference and matches the features of the person to the classes nominated to form the previous scale.

e. *Classifier result* – In this process, the results are then processed in the text and IoU outcome, which is then processed using computer vision and showcased on the same input image.

Face detection and alignment in an unconstrained environment is challenging due to various poses, illuminations and occlusions. Deep learning approaches can achieve impressive performance on these two tasks. In this paper, we propose a deep cascaded multi-task framework that exploits the inherent correlation between detection and alignment to boost up their performance. In particular, our framework leverages a cascaded architecture with three stages of carefully designed deep convolutional neural networks to predict face and landmark location in a coarse-to-fine manner. Our method achieves superior accuracy over the custom MTCNN model techniques on the challenging Indian dataset benchmarks for face detection and inference benchmark for face alignment while keeping real-time performance.

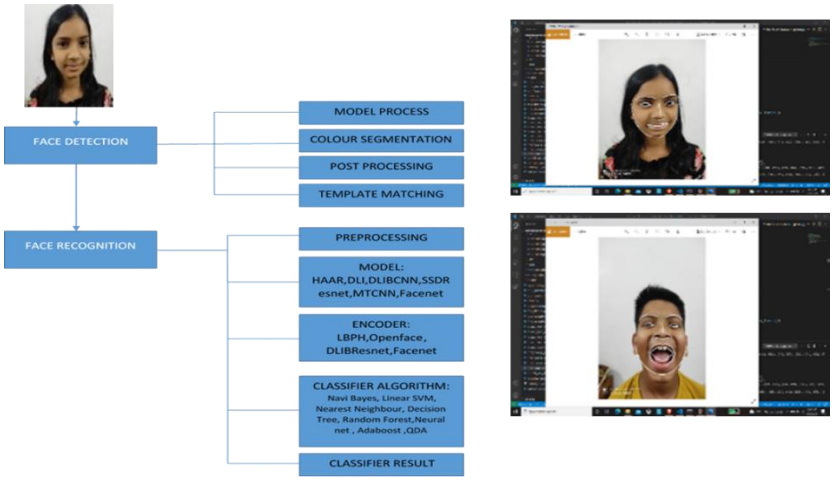


Fig. 4. Block Diagram of Face Recognition System

3.3.4 MTCNN Architecture

The first step is to take the image and resize it to different scales to build an image pyramid, which is the input of the following three-staged cascaded MTCNN: *Stage 1: The Proposal Network (P-Net)*, this stage is a Fully Convolutional Network (FCN). The difference between a CNN and an FCN is that a fully convolutional network does not use a dense layer as part of the architecture. The Proposal Network is used to obtain candidate windows and their bounding box regression vectors. Bounding box regression is used to predict the localization of boxes when the goal is detecting an object of a pre-defined class, in this case, *faces*. On obtaining the bounding box vectors, some

refinement is done to combine overlapping regions. The final output of this stage is all candidate screens after refinement is to downsize the volume of candidates. *Stage 2: The Refine Network (R-Net),* all the candidates from the P-Net are fed into the Refine Network. The R-Net further reduces the number of candidates, performs calibration with bounding box regression, and employs non-maximum suppression (NMS) to merge overlapping candidates. The R-Net gives output whether the input is a face or not, a 4-element vector for bounding box for the face, and a 10-element vector for facial landmark localization. *Stage 3: The Output Network (O-Net),* this stage is similar to the R-Net, the Output Network aims to describe the face in more detail and output the five facial landmarks' positions for eyes, nose, and mouth[8],[21].

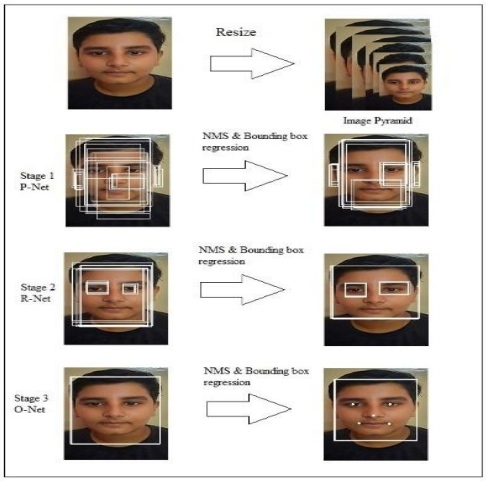


Fig. 5. Face detection using MTCNN Model

3.3.5 Comparision of MTCNN /Haar Cascade /Dlib/DNN for Open CV

The available dataset was trained on the Haar Cascade, Dlib, MTCNN and DNN module for OpenCV to validate the precision in parameter extraction for face recognition [19]. The differentiating features are tabulated below based on the test results:

Fig. 6. Result of comparison of MTCNN / Haar Cascade /Dlib/DNN for Open CV

3.4 Vehicle Collision Detection

3.4.1 System Flow Diagram for Vehicle Collision

The model can predict collision in very near proximity. YOLOv3 was chosen as the object detection algorithm for this study as it has better operational timeliness, detection accuracy, and stability while also being better suited for use with cheaper mid-to-high-end GPUs. YOLOv3 was developed in 2018[22]. It uses one-stage detection, which is suited for the high traffic density of India. At first, the video is converted to frames at a rate r and duration. As seen in the system flow diagram, as the system uses CNN, the prepossessing is the same as in face recognition. It applies a single NN to the whole image. The NN divides the image into grid cells and produces probability for every region. The YOLOv3then predicts the number of bounding boxes and chooses the best based on the probability. SVM technique is used for classification and regression analysis using the label training the data. The model produces binary output 1 in case of occurrence of collision and 0 in case of nonoccurrence of collision.

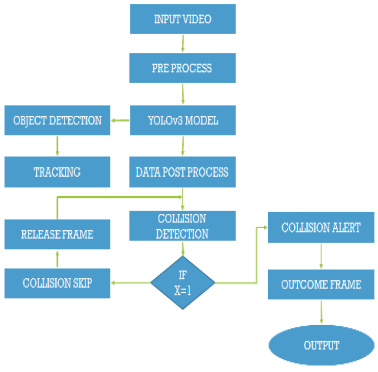


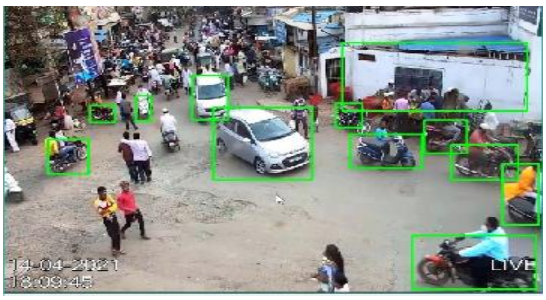
Fig. 7. System Flow Diagram for vehicle collision

3.4.2 Architecture of YOLOv3

The architecture comprises 53 CNN, also called darknet-53. The detection architecture has 53 more layers, thus making a total of 106 layers for YOLOv3[26]. The detection is made at 82, 94 and 106 layers. The essential elements of the YOLOv3 are *residual block, skip connections* and *upsampling.* The CNN layer is followed by *batch formalization* and *leaky Relu* activation function. In yOLOv3, there are no pooling layers, but instead, additional convolution layers with stride 2 are used to downsample feature maps to prevent the loss of low-level features. The prevention of loss of low-level features improves the ability to detect small objects. Input to the network is given as images extracted from the live video stream. An approx. Batch size in our case was 4900 images, it generally defined as (*n,w,h,3*) where n = no of images, w = width of image, h= height of an image, 3 refers to RGB. The width and height can be any number divisible by 32. The model accuracy improves for high- resolution images. At image is resized based on the network size.

Detection at 3 Scales: YOLOv3 does detection at 3 separate layers, namely 82, 94 and 106. The network downsamples the input image by following factors 32-16- 8 respectively, also referred to as *strides* to the networks. The size of the output would be 13x13(large objects), 26x26 (medium objects) and 52x5 (small objects)

Detection Kernels: YOLOv3 applies 1x1 Kernels at layers 82,94,106 in the network with an aim to downsample input images. The resultant feature maps will have the same spatial dimensions. The 1x1 Kernel is given by equation (b*(5+c)) where b= number of bounding boxes and c= 8 classes in our study, the resultant attributes in our case are 39. The model predicts 3 bounding boxes for each cell. *The main task in training the data set is to identify the cell that the image falls in and the centre of the object to predict the trajectory of the moving vehicle in case of a vehicle collision.* During training, it has one ground truth box for one object. In the detection scale 1, the stride of the network is 32. The input image is downsampled, the center cell is assigned to predict the object. Hence *cell objectness is 1*.



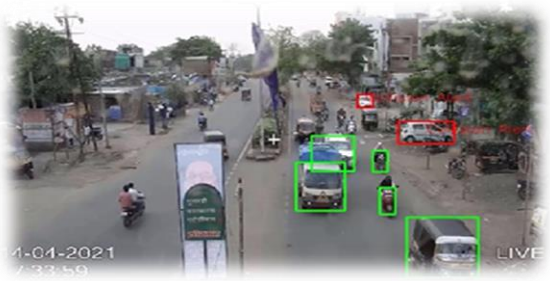


Fig. 8. Training of Vehicle collision detection using YOLOv3 model using customized data set of Aurangabad city

Anchor Boxes (priors) are used to calculate the real width and height of the predicted bounding box. In the case of our testing, 9 anchor boxes are used at each layer 82,94,106.K means clustering is applied to calculate the anchor boxes. To extract the bounding boxes, the scale and strides are computed the element product to find maximum probability (0.55/0.89/0.67). This probability is applied to all 13x13 cells across 3 bounding boxes and 39 classes. A YOLOv3 model for the COCO dataset consist 10,647 bounding boxes that are filtered with a non- maximum filtering technique. To predict the real width and height of the bounding boxes, YOLOv3 calculates the offsets to the defined anchors, also called log space transform. It passes the center through the sigmoid function.

Where in function y:

Pc = presence of object

bx,by,bh,bw = coordinates of bounding box c1,c2,c3 = classifier

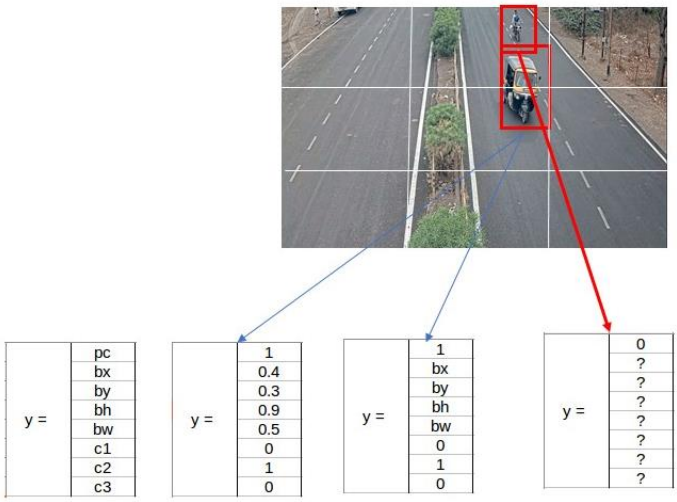


Fig. 9. Area of Intersection of Bounding Boxes

4 Conclusion

Deep learning is extensively applied in the development of image recognition and object detection technologies, which utilize convolutional neural networks (CNNs) as a basis for extracting the features of an image. The objective of image recognition in Aurangabad Smart City is to acquire more information about multiple objects in a frame. This study was subjected to the utilization of enormous data from video streams from the CCTV system installed for developing citizen-facing services. The MTCNN model was implemented to evaluate the accuracy in finding a lost person using face recognition. The available dataset was trained on the Haar Cascade, Dlib, MTCNN and DNN module for OpenCV to validate the precision in parameter extraction for face recognition. MTCNN model outperformed other models in terms of better accuracy, less detection time and high precision. The YOLOv3 was implemented to identify vehicle collision detection to save the life of a citizen. The system uses supervised learning with SVM classifiers and canny edge detection to train a model on fewer accidents. As mentioned above, a solution used the state of the art deep learning-based model on customized data set to assist planners and administrators in making a quick decision in smart city scenarios by utilizing appropriate local resources efficiently. It is an executable solution that is ready for implementation.

References

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

[2] Paolo Bellavista, Senior Member, IEEE, Giuseppe Cardone, Member, IEEE,Antonio Corradi, Member, IEEE, and Luca Foschini, Member, IEEE 'Convergence of MANET and WSN in IoT Urban Scenarios'IEEE SENSORS JOURNAL, VOL. 13, NO. 10, OCTOBER 2013.

[3] Andrea Zanella, Senior Member, IEEE, Nicola Bui, Angelo Castellani,Lorenzo Vangelista, Senior Member, IEEE, and Michele Zorzi, Fellow, IEEE 'Internet of Things for Smart Cities' IEEE INTERNET OF THINGS JOURNAL, VOL. 1, NO. 1, FEBRUARY 2014.

[4] Yunchuan Sun1, (Member, IEEE), Houbing Song2, (Senior Member, IEEE),Antonio J. Jara3, (Member, IEEE), and Rongfang Bie4, (Member, IEEE) 'Internet of Things and Big Data Analytics for Smart and Connected Communities' Digital Object Identifier 10.1109/ACCESS.2016.2529723.

[5] Weisong Shi, Fellow, IEEE, Jie Cao, Student Member, IEEE, Quan Zhang, Student Member, IEEE, 'Edge Computing: Vision and Challenges'IEEE INTERNET OF THINGS JOURNAL, VOL. 3, NO. 5, OCTOBER 2016.

[6] Ahmad AFANEH,Faculty of Architecture and Design, Middle East University "MEU", Isam SHAHROUR Laboratory for Civil Engineering and Geo- Environment University of Lille 1 for Sciences and technologies, Lille, France 'Use of GIS for SunRise Smart City project, large scale demonstrator of the Smart City' 978-1-5090-6011-5/17/$31.00 ©2017 IEEE.

[7] Mehdi Mohammadi, Graduate Student Member, IEEE, Ala Al-Fuqaha, Senior Member, IEEE,Mohsen Guizani, Fellow, IEEE, and Jun-Seok Oh 'Semisupervised Deep Reinforcement Learning in Support of IoT and Smart City Services' IEEE INTERNET OF THINGS JOURNAL, VOL. 5, NO. 2, APRIL 2018.

[8] Shilpa Jahagirdar, Sanjay Koli 'Automatic Accident Detection Techniques using CCTV Surveillance Videos: Methods, Data sets and Learning Strategies' International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-3, February, 2020.

[9] Anil Kumar Nayak Supervising Professor Dr. Farhad Kamanga 'Face Detection And Recognition Using Moving Window Accumulator With Various Deep Learning Architecture'.

[10] https://org/abs/1503.03832https://arxiv.org/abs/1409.4 842

[11] https://cmusatyalab.github.io/openface/models-and- accuracies/#model defination

[12] https://github.com/iwanttooxxoox/Keras-OpenFace

[13] https://smartcities.gov.in/

[14] https://realpython.com/python-ides-code-editors- guide/

[15] https://www.impactqa.com/blog/what-is-the-role-of- iot-machine-learning-in-smart-cities/

[16] https://en.wikipedia.org/wiki/Smart_Cities_Mission

[17] https://www.swann.com/blog/types-of-security- cameras/

[18] https://www.researchgate.net/publication/325236448

[19] http://krasserm.github.io/2018/02/07/deep-face- recognition/

[20] https://docs.python.org/3/tutorial/venv.html

[21] https://towardsdatascience.com/face-detection-using- mtcnn-a-guide-for-face-extraction-with-a-focus-on- speed-c6d59f82d49

[22] Application_of_a_Model_that_Combines_the_YOLO v3_Ob.pdf

[23] highway-traffic-book-2018.pdf

[24] highway-traffic-book-2019.pdf

[25] https://www.programmersought.com/article/83616031 867/

[26] https://www.analyticsvidhya.com/blog/2018/12/practical-guide-object-detection-yolo-framewor-python/#:~:text=The%20biggest%
20advantage%20of%20using,to%20the%20R%2DCNN%20algorithms