Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1, No.10 : 2024
ISSN : **1906-9685**

INTELLIGENT CAMPUS SURVEILLANCE SYSTEM

Dr. J. ARUNA DEVI, Associate Professor in Department of Computer Science and Engineering, Vignan's Institute of Information Technology, Duvvada

#M. BHARATH, Student of Computer Science & Engineering
NISSITHA JYOTHI, Student of Computer Science & Engineering
P. YASWANTH SAI, Student of Computer Science & Engineering
M. K. ANSON, Student of Computer Science & Engineering
P. PAWAN PRASAD, Student of Computer Science & Engineering
Vignan's Institute of Information Technology(A), Visakhapatnam, Andhra Pradesh, India

ABSTRACT: In recent times we have used a number of surveillance systems using CCTV cameras for restricted areas and storage of both positive and negative data for months. When it comes to college students' safety, security, and tracking play major roles as they do various different activities so in order to control and have proper evidence to keep these parameters in control this system will keep on monitoring and fulfill the three agendas using computer vision technology and Open CV library. Implementing a Deep stream is also the second primary objective as it will detect through different cameras without any frames dropping and give us accurate detections. The workflow starts with Data collection and Data preprocessing followed by annotating the required data according to the incidents reported making them negative and positive data which will be used to train and validate the MODEL and then followed by testing the validation data of the collage to make it suitable for the environment. This model takes input as a CCTV stream and detects the mentioned objects that going to be dumped in the database smoothly and this is reflected on the dashboard.
Keywords: Surveillance, Object Detection Model, Computer vision Engineering.

1. INTRODUCTION

The basic use case defines to make sure the safety, security, and protection of the students via surveillance through CCTV by detecting any unauthorized situations happening around the campus. Our focus today is on understanding how advancements in Understanding campus conditions is crucial for tailoring treatment. Detecting through these cameras without fail will help to increase security without any hand power. The main development creates a pipeline for different campuses with just pre-installment without any huge data collection and training. This inference will run through using docker in Linux. Deep stream will install all the prerequisites of all Python libraries. Computer vision plays an important role in real-time monitoring with detections and analysis of video streams with detections. Behavior analysis is possible through movements and actions which will be detected through this computer vision. Automated alerts can be generated through this where people can understand the emergency of the situation around them. In the initial stage of development of the project, the data collection for model training is a huge parameter. Open cv is a library for computer vision in which we can implement different model to predict on images and videos. During the collection of data getting negative data and positive data and separating them is necessary. After collecting the data we need to annotate according to the detections we planned.

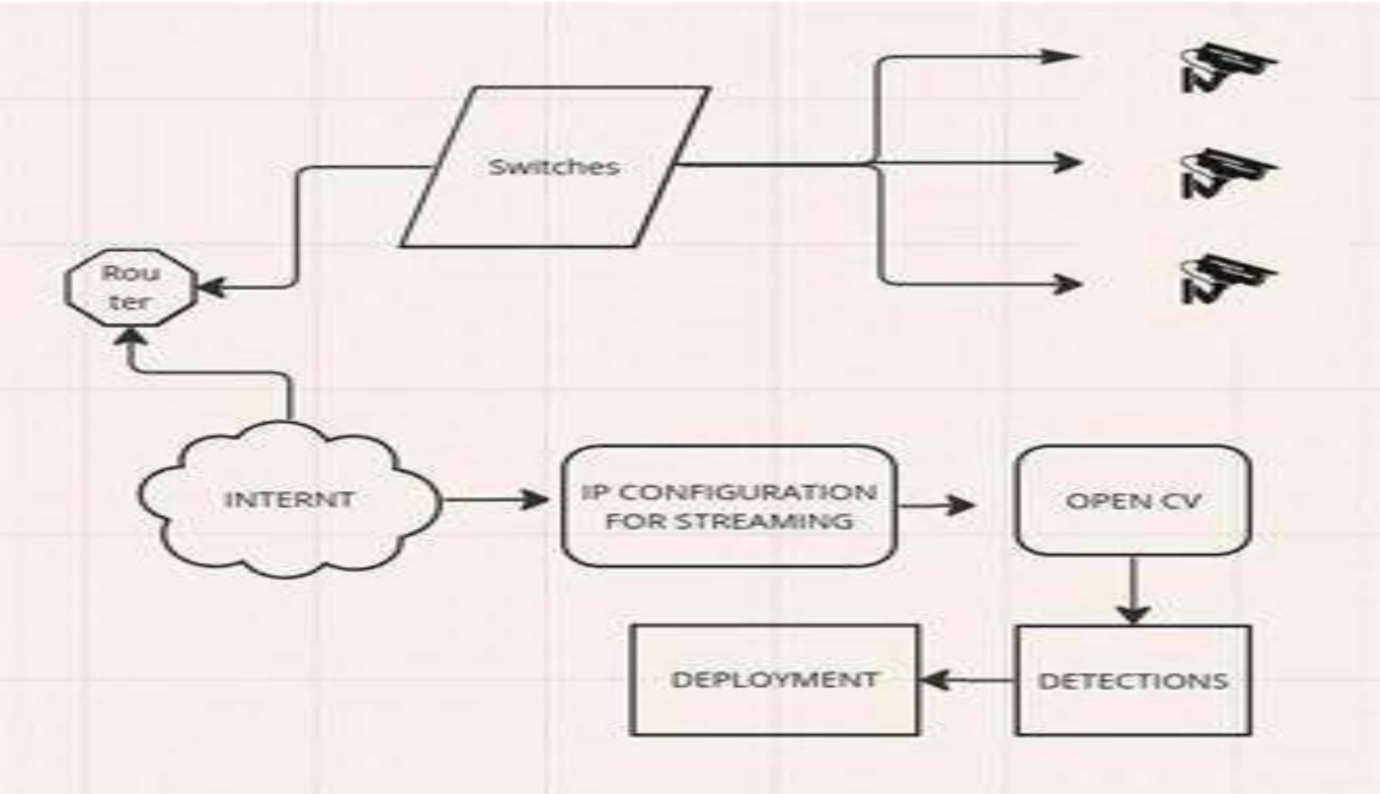

Fig -1*: Flow Chart of Existing System*

After annotation, we can send them to training by taking 70% of the test data and 30% of the validation data. The trained model will be performing its prediction after its training. But the issue which comes is there will be huge frame drop in open CV due to live stream and the detections which are going on. In order to overcome the difficulty we constructed deep stream version 6.2 on a Linux system to make sure there would be no frame drop from the IP camera and stream live data without any type of latency which can also be seen on the web dashboard. The detections can be done using trained models which have been annotated from the data collected.By strategically Mapping or identifying the patterns or similar data patterns from annotation the model gets trained from it and different variants of the data as well as negative data should be identified. This annotation will create a bounding box on respective detection that needs to be taken.

2. REVIEW OF LITERATURE

Deep learning techniques have been extensively studied and applied in various domains of computer vision and image processing. Bodake and Kadam proposed a crowd detection technique utilizing Convolutional Neural Networks (CNN) for counting individuals in static images captured across different environments [1]. Santhini and Gomathi focused on crowd scene analysis using deep learning networks, aiming to analyze crowd scenes in videos across different dimensions [2]. Rahmaniar and Hernawan experimented with real-time human detection in buildings using NVIDIA Jetson boards, demonstrating the effectiveness of these platforms in executing computer vision tasks with deep learning [3]. Wei et al. developed a deep learning-based person detection and classification system for far-field video surveillance applications [4]. Xiong et al. introduced a deep learning approach for driver distraction detection, particularly focusing on identifying mobile phone usage during driving [5]. Ojha et al. developed multiple driver drowsiness detection systems using diverse sensors and algorithms to monitor physical and behavior.

3. METHODOLOGY

Intelligent Campus Surveillance provides various detections like Crowd, Person, Vehicle, Helmet on Vehicle, and Mobile Sleep Detection will be detected through this surveillance using deep stream.

Collection and Annotation: Machine learning Models extensively depend on Data regarding the environment we are providing based on the training set and the type of annotation the model performs according to it. Annotate the dataset with bounding boxes around each particular object detection.

Model Selection: Choosing a suitable model either a Pre – trained or custom model for object detection is crucial. Most of the cases for object detection include more of YOLO (you only look once), and SSD (single shot Multibox Detector). Also, we can use the custom model with training data.

Deepstream Setup: Setting up the deepstream docker setup in your system where it should contain NVIDIA graphic card and NVIDIA SMI drivers. Configure the Deepstream setup to process video streams from CCTV cameras or other sources.

Integration with Deepstream: Integrate the trained model with Deepstream for detecting the required mentioned objects using Deepstream SDK (software development kit). Configure the different Deepstream elements such as analytics, and tracker to add to your pipeline.

Deep Learning: Deep learning techniques, such as neural networks, are increasingly being employed in drug recommendation systems to analyze complex patient data and make accurate predictions. Deep learning models can learn intricate patterns and relationships in patient data, leading to more precise drug recommendations.

Integration with Deepstream: Integrate the trained object detection model with Deepstream for real-time detection.Configure Deepstream elements like analytics and tracker for comprehensive surveillance.

Real-time Object Detection: Use Deepstream to process live video feeds and detect objects such as crowds, persons, vehicles, etc. Implement post-processing techniques like Non-Maximum Suppression for accurate detections.

Dashboard and Alerts: Develop an admin panel/dashboard to display detailed analytics and reports. Include features like PDF and CSV format reports, incident image capture, and real-time alerts.

4. MISSION VISION (COMPUTER VISION - DEEP LEARNING)

Computer Vision:

Computer Vision algorithms are a subset of machine learning algorithms specifically designed to analyze and interpret video language data. These algorithms are essential for various tasks such as image classification, segmentation on image analysis, named entity recognition, and more. Computer vision algorithms will feed on live videos and images coming from the stream. Reads frames from the video considers the frame extracts the features or pixels and tries to learn the feature from that pixel and validates on different data.

YOLO (you look only once) is a real-time object detection algorithm developed by Redman and Ali Farhadi in 2015. It is a single-shot object detector that uses a convolutional neural network to predict the bounding boxes and class probabilities of objects in input images or videos. This Yolo was previously used in C language which is used in a framework called Dark Net.

Yolo will divide the image into grid cells where each cell represents a bounding box and the probability of the existence of an object. This also predict the class of the object which is an additional feature that is already added to the Yolo model. This Yolo model will be single shot model where everything will be done in single shot without any multiple steps. During the detection, the classification, extraction and feature extraction, and finding out the specific patterns in the Yolo will be.

The basic execution of YOLO is to divide the image to grid cells and predict the pattern and learn according to it. Within a single path, it will detect every object inside it which will frame and make the model to run faster than usual. Yolo has several versions and it will be adopted widely within no time. The below structure defines it.

This Deepstream tries to utilize the live video feed from the camera and applies the object detection model to it in each frame to detect crowds, vehicles, persons, sleep, and other detections. In this Post-process analysis, we can remove unwanted objects that are created or duplicate objects that are created such as **NON-MAXIMUM SUPPRESSION**.

Scalability and Efficiency: YOLO is computationally efficient and scales well to large datasets, making it particularly suitable for Video/Image classification tasks with high-dimensional feature spaces.

Object Detection: Yolo is a frame Classifier that predicts the probability of a given instance to a particular class.

DEEPSTREAM:

Deepstream is basically used to bind all the Python packages in to single project which will be helpful in writing all the functionalities in easy manner. Open CV will be difficult to solve these tracking problems as accuracy and latency increase as the functionality grows so in order to avoid it we use deepstream to solve these types of problems which will bind the features and run asynchronously throughout the pipeline. Alerts will be sent to the dashboard through database dumping. Overall, an intelligent campus surveillance system is recommended system to leverage a combination of live feed and detect the actions for the purpose the safety, security, and tracking. Every campus needs these types of systems in order to have a keep on eye for detecting all these types.

How the detections and website is shown:

The "ADMIN PANEL" will show the detailed analytics and reports which contain PDF and CSV format reports which give detailed report which give us the structure of type of detection if crowd activities occur then it will automatically capture the image and show it on to the display. This will help us to store the incident images compared to different CCTV storage detections. This admin panel also shows the complete detection analytics which are negative data when it detects the person or crowd or vehicle and other detections.

5. RESULTS

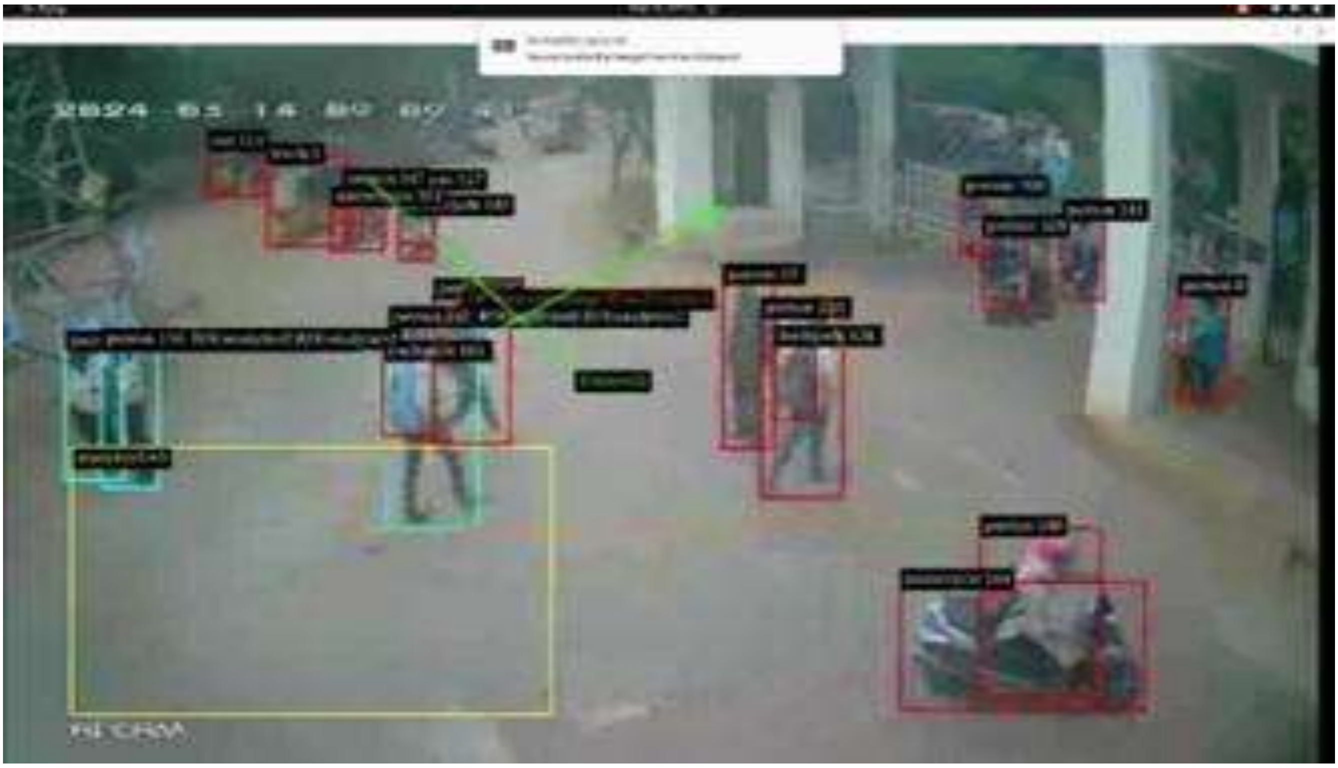

***Fig -2**: Person and Vehicle Detections.*

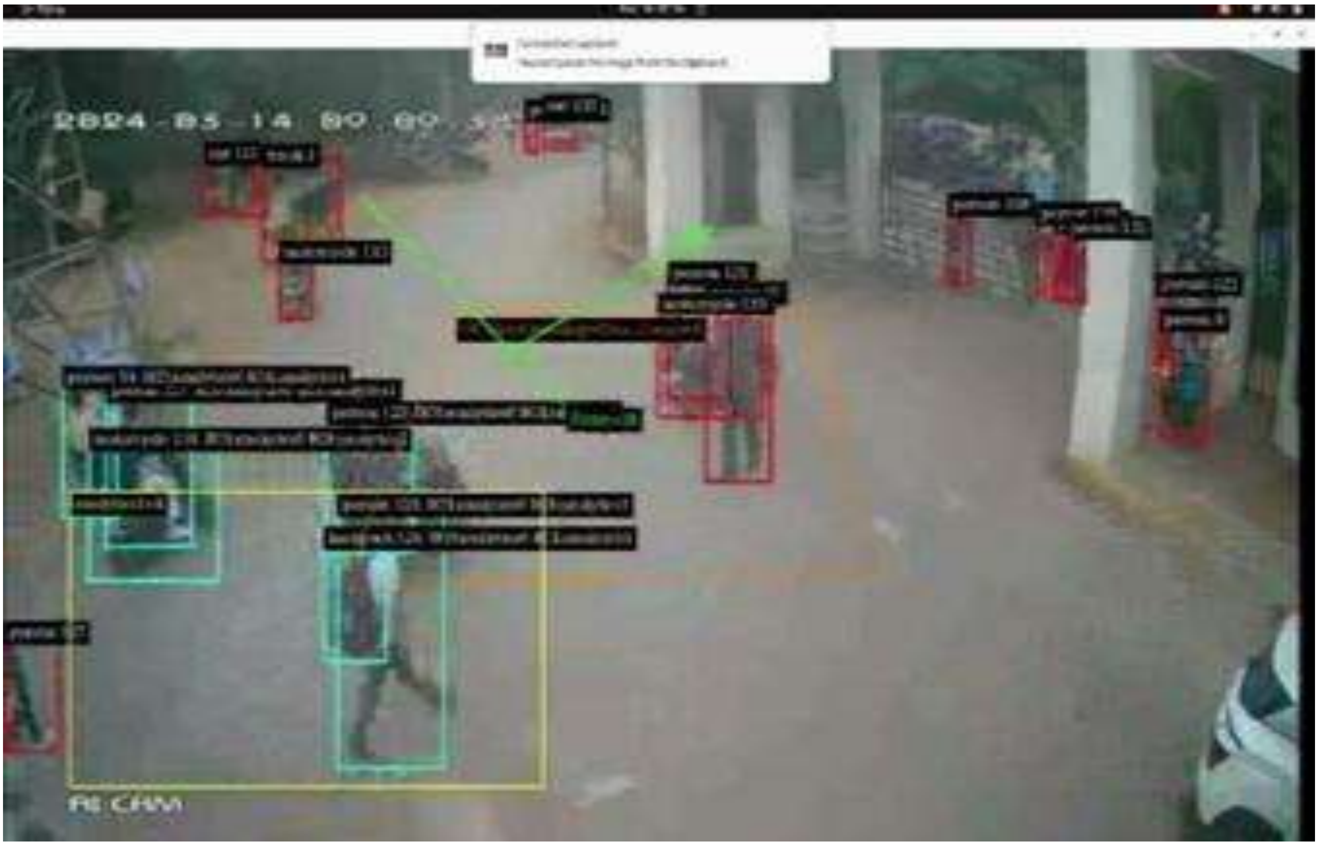

***Fig -3**: Result-1*

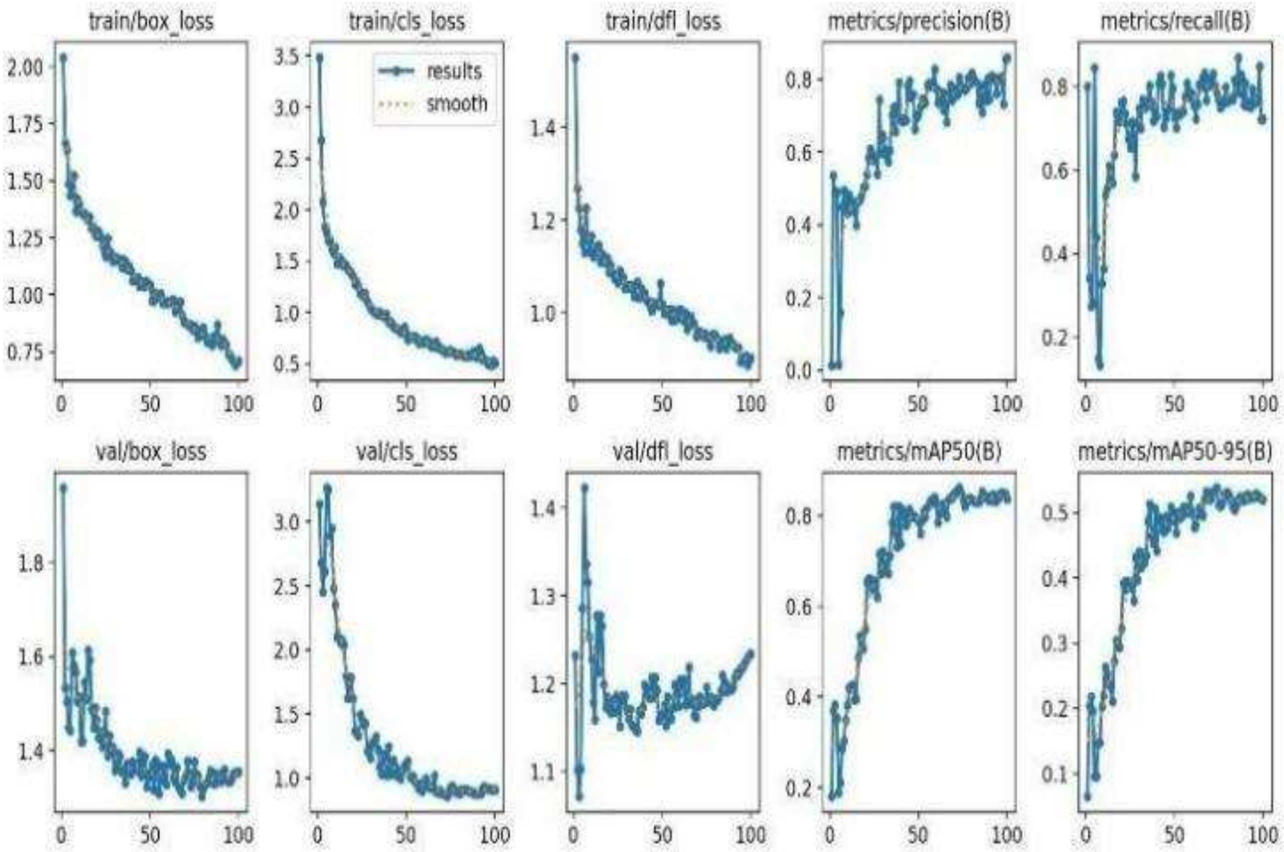

Fig -4*: Output of tuning*

Fig -5*: Output of Dashboard*

6. CONCLUSION

In the given code, we've created a sophisticated computer vision-based surveillance system adept to detecting persons, vehicles, and other 75+ classes according to the required situation. This surveillance system brings proper safety and protection for the students throughout the campus. By intersecting advanced techniques in Web-based applications our system ensures accurate detections while considering the relevance and quality of user feedback. With this end-to-end technology, it will give an accuracy of 85% during the inference. By taking the live feed from the Industry IP Camera there is 0 latency during the detection which leads to proper detections which helps for accurate detection data. This end-to-end solution will lead to significant change in the campus as well as society which reduces the manual people surveillance. Reducing the cost of greater then 30 thousand per month by avoiding unexpected accidents for the people who are working near gates. This system will cover throughout region replacing three men throughout that region. Not only detecting

REFERENCES:

1. Qiu, Y. (2014) Video Based Vehicle Detection in Intelligent Transportation System. Master Thesis, Jilin

2. K. Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference over and pattern recognition (pp. 770778). [8] Lin, T. Y., Dollar, P., Girshick, R. B., He, K.Hariharan, B., & Belongie, S. J. (2017, July). Feature Pyramid Networks for Object Detection. In CVPR (Vol1)

3. https://www.researchgate.net/publication/363577111_YOLO_Based_Multi_Objective_Vehicle_Detection_and_Tracking

4. Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. ArXivpreprint ar Xiv: 1412.6806.Y. Bao and X. Jiang, ‘‘an intelligent medicine recommender system framework,’’ in Proc. IEEE 11th Conf. Ind. Electron. Appl. (ICIEA), Jun. 2016, pp. 1383– 1388.

5. Redman, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition(pp. 779-788)U. Bhimavarapu, N. Chintalapudi, and G. Battening, ‘‘A fair and safe usage drug recommendation system in medical emergencies by a stacked ANN,’’ Algorithms, vol. 15, no. 6, p. 186, May 2022.

6. J. Chen, K. Li, H. Rong, K. Bilal, N. Yang, and K. Li, ‘‘A Disease diagnosis and treatment recommendation System based on big data mining and cloud computing,’’ Inf. Sci., vol. 435, pp. 124–149, Apr. 2018.

7. https://link.springer.com/article/10.1007/s11042-022- 13644-y

8. https://docs.nvidia.com/metropolis/deepstream/dev- guide/text/DS_ref_app_deepstream.htmlA. S. Hussein,W. M. Omar, X. Li, and M. Ati, ‘‘Efficient Chronic Disease diagnosis prediction and recommendation system,’’ in Proc. IEEE-EMBS Conf. Biomed. Eng. Sci., Dec. 2012, pp. 209–214.

9. https://github.com/NVIDIA-AI- IOT/deepstream_reference_apps.