2022 Iraqi International Conference on Communication & Information Technologies (IICCIT-2022) , Basrah University , Basrah , Iraq

GPU-Based Multiple Face Recognition Using YOLOv5x

Ali Ahmed Abed
Department of Computer Engineering
University of Basrah
Basrah, Iraq
ali.abed@uobasrah.edu.iq

Ali Abdulghafoor Jallod
Department of Computer Engineering
University of Basrah
Basrah, Iraq
ali.a.jallod@gmail.com

***Abstract*— The growth of deep learning has led to impressive achievements and challenging issues in artificial intelligence and computer vision fields. Face recognition based on convolutional neural networks (CNNs) is one of the challenging and interesting hot topics in these fields. Recently, YOLO (You Only Look Once) became one of the more robust and attractive frameworks adopted for object detection in real-time. Despite that it is originally designed for object detection, the work of this paper is succeeded to apply it for face recognition with multiple faces recognized simultaneously in real time. The last version of this platform, known as YOLOv5x, is adopted to satisfy the desired task based on embedded GPU (Graphics Processing Unit) edge device. The proposed system is trained using custom-created dataset of 7378 augmented images, tested, and validated using Google Colab Notebook and Jetson Nano developer board. This board is supported by a parallel computing platform called CUDA (Compute Unified Device Architecture) enhanced with PyTorch framework for real-time acceleration without using DeepStream real-time server or Docker container. The developed system can recognize up to 20 faces simultaneously with high inference time of 0.817 s and precision of 99.8%.**

Keywords—YOLOv5x, GPU, Jetson Nano, CUDA, PyTorch, Google Colab notebook.

I. INTRODUCTION

Recently, most of the deep learning algorithms has a CNN infrastructure with very large number of parameters and layers. These deep learning models enhanced the various vision tasks, including image categorization, object detection, and face recognition [1]. The processing layers are activated to learn and process inputs at various abstraction levels emulating human brain. Several tremendous applications for face recognition have been created such as surveillance systems, military, finance, person authentication, biometric identification, content-based data retrieval, access management, and social media [2]. The infrastructure of face recognition systems is deep neural networks adopted for learning effective features and parameters to obtain faster and more reliable algorithms. High accuracy face recognition is basically dependent on high-resolution facial image with no occlusion. Face recognition system automatically identifies or verifies a person from an image or a video stream [3]. The input face image is applied to the neural network to get the face features and parameters. Then, the network accomplishes a comparison between the input face image with a database of faces already stored in memory [4]. Several deep learning platforms were created for face recognition issue with various accuracies, inference time, and computational complexities. YOLO (You Only Look Once) is the most recent and high-speed deep learning platform that created originally for real-time object detection (not face recognition) applications. This paper proposes YOLO-based framework for not only face recognition but also with multiple recognized faces in a single scene. It is known that YOLO is a state-of-the-art object detector primary applicable for recommendation systems but not a face recognition platform, thus, the work of this paper is indeed a challenge. Due to the high computational complexity of deep learning algorithms, multiple face recognition is another challenge addressed in this work. Implementation of the proposed model in real-time on an integrated GPU embedded edge device to recognize several faces simultaneously is the third challenging issue encountered in this context. The YOLO platform came with five versions, YOLOv1-YOLOv5, during the period 2016-2020. The last version, YOLOv5, came with four versions s, m, l, and x, meaning small, medium, large, and extra-large. In this paper, the most recent version, YOLOv5x, is adopted for implementing the desired task. The proposed model is trained on high-speed servers available in Google Colab and inferred on a Notebook and then on a Jetson Nano developer kit supported by CUDA. To adapt the designed system with real-time requirements, PyTorch framework, instead of DeepStream and Docker files, is embedded into the developer board to minimize processing time, hence, increasing processing speed. A custom-created dataset, contains 217 student images augmented to 7378 images, is created for system training and testing. The dataset is partitioned so that 90% is used for training and 10% is used for validation.

The rest of the paper is organized as follows. Section II lists the works related to the paper subject. Section III describes the proposed method used in the proposed face recognition system. Section IV contains the experimental results that prove the system validation. Finally, Section V presents a summary of conclusions and some future work.

II. RELATED WORKS

The following is the past research works related to the subject of the paper.

978-1-6654-7220-3/22/$31.00 ©2022 IEEE

Bochkovskiy, *et al.* [5] presented a detector with two parts: a backbone that pre-trained on ImageNet and a head that predicted object classes and bounding boxes. The most prominent models for one-stage object detectors were YOLO, SSD, and RetinaNet. Redmon [6],[7] published a Darknet deep learning system including a YOLO object detector. Yang *et al.* [8] used YOLO target detection technology. Experimental results proved that the object detection approach based on YOLO had an acceptable resilience and detection speed. Chen, *et al.* [9] suggested YOLO-face, an object detector based on YOLOv3, to increase object identification performance. The method employed more suitable anchor boxes for object identification and a more accurate regression loss function. Garg, *et al.* [10] enhanced the accuracy of detecting faces by utilizing a deep learning model. The accuracy of detecting the face was efficient as compared to the traditional methodology that was available at that time. The suggested model employed the convolutional neural network as a deep learning strategy for detecting faces in videos. Their model was trained and tested using the FDDB dataset. Shinde, *et al.* [11] presented a technique for detecting, defining, and recognizing acts of interest in near real-time using frames derived from a continuous stream of video data taken from a surveillance camera. After a set duration, the model accepted input frames and assigned an action label based on a single frame. Chun, *et al.* [12] used YOLOv3 to solve face detection issues in complicated situations. Jiang, *et al.* [13] used the YOLO series' lite methods based on the complete YOLO. The authors wiped away the whole YOLO method network. Qi, *et al.* [14] treated face detection as a general object detection task, implemented a YOLOv5 object detector as a face detector (not face recognizer), and called it "YOLO5Face."

In this paper, the YOLO version, which is based on YOLOv5 [14],[15], is used to recognize faces in real-time on the Jetson Nano developer kit. This work is trained on a custom-created dataset with 20 classes of people using a high GPU based on CUDA, then inferred on a Nano kit using PyTorch platform, without adopting Docker container and DeepStream technology, to increase the FPS for recognition leading to minimized processing time and increasing frame per second and recognition rate.

III. The Proposed Work

To develop the proposed deep learning system, the following issues are considered:

A. YOLOv5x

Shortly after the release of YOLOv4 in Darknet framework, YOLOv5 is introduced using the Pytorch framework. Most of the previous research works adopt already tested models, but in our work, we forwarded attention to the new YOLOv5x architecture to prove its effectiveness in solving real-time face recognition problems. YOLOv5x, which is launched in 2020, is the biggest platform having 89 million parameters and 284 layers having different accuracy and performance [15]. YOLOv5x guarantees faster inference, higher accuracy for large and tiny objects, less computational resources, and less model files. Therefore, it is expected to work efficiently in embedded system applications on Jetson Nano based on the data stream. This expectation is proved experimentally and practically via the real time implementation and validation of the designed model. Unfortunately, there is no paper on YOLOv5x till June 01, 2022, hence, this work began with YOLOv5s so that it is easy to understand the YOLOv5x. To understand how YOLOv5x improved the performance, one needs to go through the network architecture diagramed in Fig. 1 [16]. Addition of the anchor box selection mechanism making the network never require analyzing the input dataset, but it will automatically "learn" and utilize the ideal anchor boxes for each dataset during training. To proceed with structural details and some coding requirements, one can go through the Yolov5 GitHub repo repository.

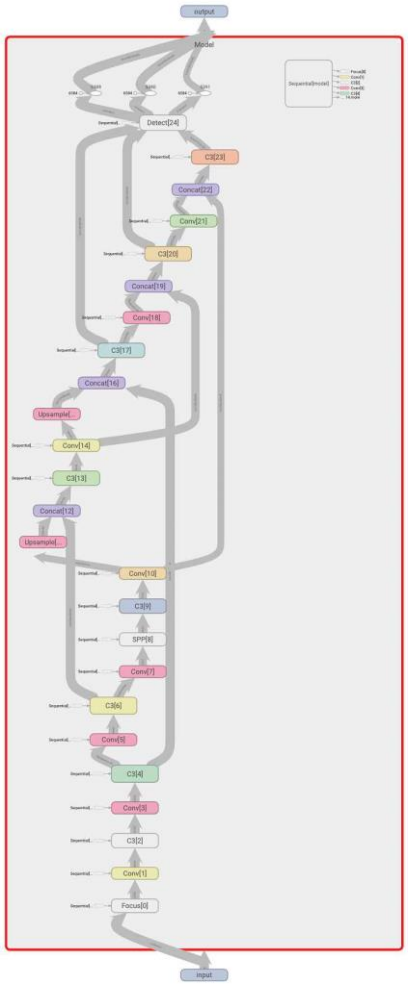

Fig. 1. YOLOv5 model displayed in TensorBoard

B. Dataset

To train a convolutional neural network efficiently, the dataset should contain a large number of images and a diversity of human faces to prevent overfitting and to satisfy an acceptable classification accuracy. For convenience, a custom dataset has been created to fit the memory footprint of the embedded device used in this work. The experimental investigation proved that using of minimized-size dataset has no significant effect on the model accuracy, but the model performance is strongly influenced by the hardware architecture where the model is tested. Jetson Nano is significantly increased the YOLOv5x model performance in relation to fast real-time inference. The dataset is created and stored in a database with 20 classes and labelled with

PyTorch format. 217 Images are stretched to 416×416 pixels and augmented with more than 24 operations: add_light, add_light_color, fliping, inverting, saturation, superpixel, averaging, bilateralBlur, dilation, erosion, GausianBlur, medianBlur, multiplying, openingImage, addeptiveNoising, contrasting, edgeDetectCanny, edging, embossing, grayscaling, peppering, salting, salt-and-peppering, and sharpening. Different parameter values for preprocessing, data augmentation, and post-processing are applied for the proposed model architecture to increase the number of images ready for training to 7378, eliminating false positives, and increasing true positives, where false/true positives refer respectively to pictures that have been labeled as false/true and produced a false/true result as predicted by the model [17].

C. Training

The proposed software package is developed with OpenCV and Python 3.9.5 for PC and Python 3.6.9 for Jetson Nano. The PyTorch is used for training and detection with YOLOv5. The developed model is trained using Google Collaboratory Notebook to benefit the high-performance Tesla GPUs of Google Cloud overcoming the limitation of the available PC local machine. The training with Google Colab used the NVIDIA Tesla T4 16G GPU with CUDA version 11.2 for 1000 epochs using torch 1.12.0+cu113. This high-speed GPU saved a considerable amount of "Wall time" throughout training process as compared to the time consumed for training by an ordinary Core i7, 2.4GHz laptop or even by the Jetson Nano AI developer kit. Depending on different thresholds, performance metrics curves are obtained to extract some important evaluation parameters such as: precision, recall, and mean Average Precision (mAP) [18].

D. System Setup

Since the computational cost of the CNN network architecture of YOLOv5x is high, the hardware board has a hard constraint in that it must contain high GPU and RAM modules. Also, the selected board should support parallel programming to enable several threads to be processed concurrently. In this paper, the powerful embedded board NVIDIA Jetson Nano 4GB shown in Fig. 2 is technically chosen since it has 128 CUDA cores GPU and a quad-core ARM Cortex-A57 high-performance CPU.

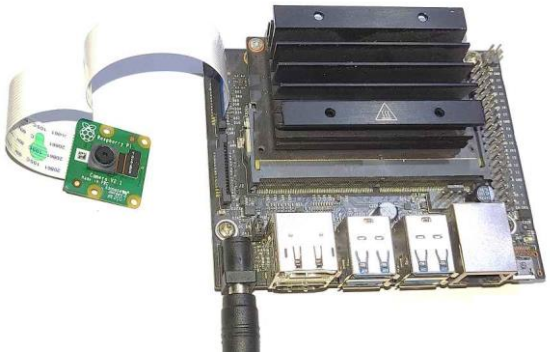

Fig. 2. NVIDIA Jetson Nano 4GB with an IR camera

This low-cost, small-sized, and power-efficient architecture led to the decision of choosing it for our artificial intelligent face recognition commercial-like product. The fundamental software tools and packages used to support Jetson Nano are CUDA 11.2, Cudnn 8.0, TensorRT 7.1.3. TensorRT is a deep learning inference SDK with excellent performance that contains an inference optimizer and a runtime for deep learning applications with low latency and high throughput. In addition, Python and OpenCV are already installed on the board. Furthermore, Virtual Network Computing (VNC) server is installed on the board to monitor its Linux environment on the laptop's screen. Accompanying to the Jetson board, an IR Camera Pi module 2 (shown in Fig.2) is plugged-in to capture real-time still (stationary) images, archived videos, or live video streaming. The YOLOv5x platform is then applied along with the input images to the bounding box and labeling to monitor the recognized faces with their labels and recognition accuracies.

IV. Experimental Results

The experimental results are classified into two categories: First, results of training and Second, results of inference.

A. Training Results

The performance metrics and statistics considered for the proposed model evaluation during training is calculated and displayed in Fig. 3. Status of the dataset during training is monitored in Fig.4.

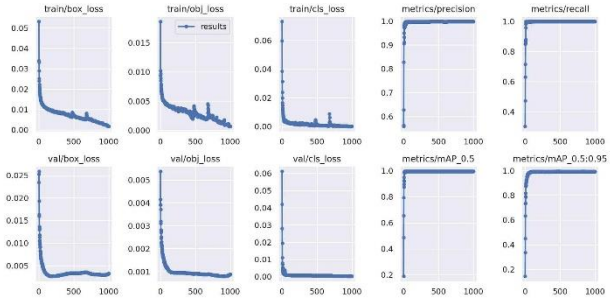

Fig. 3. All required performance measures of the proposed system

train-batch0 train-batch1

val_batch0_labels val_batch0_pred

val_batch1_labels val_batch1_pred

Fig. 4. Samples of intermediate training results

The main and important measures that mostly focused are precision, recall, mAP, and wall time [19]. The obtained result of training for these measures is summarized in Table I.

TABLE I. IMPORTANT PERFORMANCE MESURES

Performance	*YOLOv5x on TeslaT4*
Precision	0.99814
Recall	0.99894
mAP@.5	99.492%
mAP_0.5:0.95	99.222%
Wall Time	2d 14h 30m 43s

From Table I, the high value of the mean Average Precision (mAP) proved the robustness of the proposed face recognizer. Since we used Tesla T4 GPU, the wall time (the total time spent for training) is somewhat short.

B. Inference Results

After training of the proposed model with the custom-created dataset, the weight files, YAML files (used to specify the whole proposed model configuration), and all other necessary files are converted to suitable formats and uploaded into the Jetson Nano. The designed system could recognize about 20 faces simultaneously with recognition and accuracy reaches more than 98% in real time for a large diversity of face images as shown in Fig.5. A movie-like video streaming for different number of faces inside images with various appearances and positions are found in the following private Google Drive link:

https://drive.google.com/open?id=1vCPp0bvLglb8OSyv3lmM-cgscfi4m6Sk&authuser=0

The real average inference time is measured to be 0.814s which means a 1.2285 FPS for the developed system based on Jetson Nano. This could lead to strongly recommend the designed system commercially.

V. CONCLUSION AND FUTURE WORK

In this paper, YOLOv5x, a deep learning-based object detection platform, is introduced unprecedentedly as a face recognizer for multiple faces in real-time. The framework is implemented on the NVIDIA Jetson Nano board, which is built with CUDA to realize parallel processing. PyTorch platform are used to accelerate real-time fast inference. A custom dataset is created for training with Google Colab and Jetson Nano. The measures showed that YOLOv5x performs better than previous versions in relation to inference on Jetson Nano and could recognize small and tiny faces easily. The metrics proved the robustness of the designed model for dealing with images, stored videos, and real-time live video streaming. This system could be considered as the basic building block for all commercial applications that need face recognizers such as security and surveillance civil and military systems. Therefore, as a future work the low-cost, energy efficient real-time multiple face recognizer designed in this work can be integrated successfully with any security, attendance, or surveillance system.

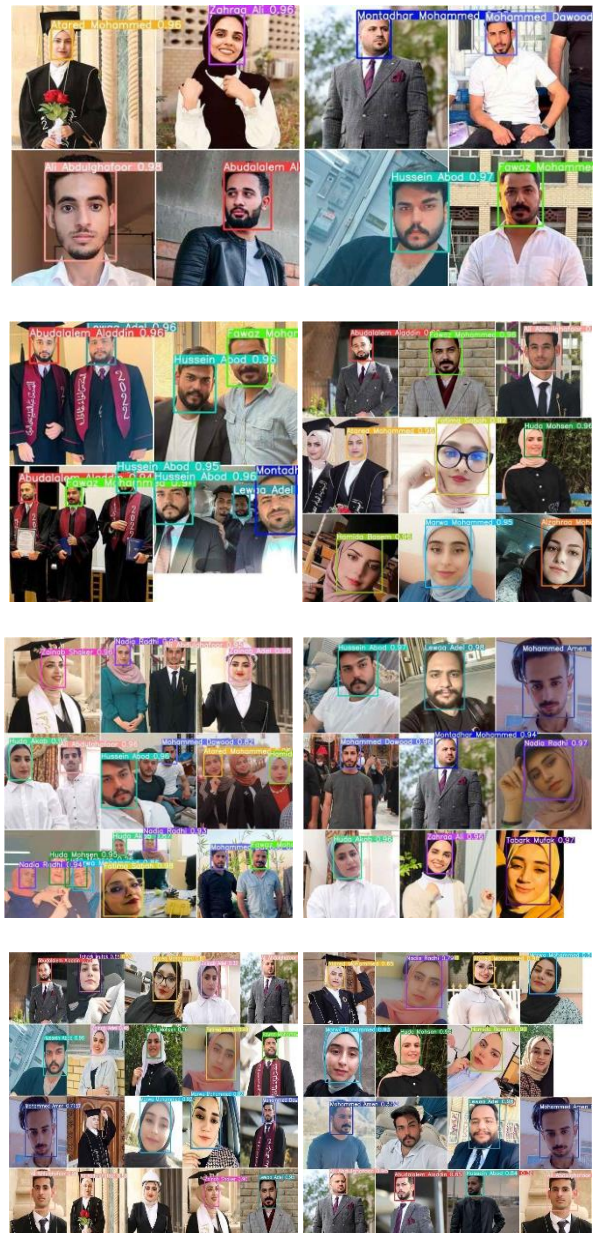

Fig.5. Real-time validation for the proposed multiple face recognizer

REFERENCES

[1] P. S. Prasad,., Pathak, R., Gunjan, V. K., & Rao, H. R., "Deep learning-based representation for face recognition," In ICCCE 2019, Springer, Singapore, pp. 419-424, 2020.

[2] A. Voulodimos, N. Doulamis, A. Doulamis, & E.Protopapadakis, "Deep learning for computer vision: A brief review," Computational Intelligence and Neuroscience, 2018.

[3] P. Kumar, "A multiple face recognition system with DLIB'S ResNet network using deep metric learning," Journal of Critical Reviews, 7(6), 856-859, 2020.

[4] T. Keerthana, K. Kaviya, S. D. Priya, & A. S. Kumar, "AI-enabled smart surveillance system," Journal of Physics: Conference Series, IOP Publishing, vol. 1916, no. 1, p. 012034, 2021.

[5] A. Bochkovskiy, C. Y. Wang, & H. Y. M.. Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[6] J. Redmon, ;A. Farhadi, , "YOLO9000: Better, faster, stronger," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, pp. 7263–7271, 2017.

[7] J. Redmon, A. Farhadi, "Yolov3: An incremental improvement," arXiv 2018, arXiv:1804.02767, 2018.

[8] W. Yang, & Z. Jiachun, "Real-time face detection based on YOLO," 2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII), pp. 221-224, 2018.

[9] W. Chen, H. Huang, S. Peng, C. Zhou, , & Zhang, C., "YOLO-face: a real-time face detector," The Visual Computer, 1-9., 2020.

[10] D. Garg, P. Goel, S. Pandya, A. Ganatra, & K. Kotecha, "A deep learning approach for face detection using YOLO," 2018 IEEE Punecon, pp. 1-4, 2018.

[11] S. Shinde, A. Kothari, & V. Gupta, "YOLO based human action recognition and localization," Procedia Computer Science, 133, 831-838, 2018.

[12] L. Z. Chun, L. Dian, J. Y. Zhi,., W. Jing, & Zhang, C., "YOLOv3: face detection in complex environments," International Journal of Computational Intelligence Systems, 13(1), 1153-1160, 2020.

[13] Z. Jiang, Zhao, L., Li, S., & Jia, Y., "Real-time object detection method based on improved YOLOv4-tiny," arXiv preprint arXiv:2011.04244, 2020.

[14] Qi, D., Tan, W., Yao, Q., & Liu, J., "YOLO5Face: Why reinventing a face detector," arXiv preprint arXiv:2105.12931, 2021.

[15] H. Wang, X. Tong, & Lu, F., "Deep learning-based target detection algorithm for motion capture applications," Journal of Physics: Conference Series, IOP Publishing , vol. 1682, no. 1, p. 012032, 2020.

[16] S. Gutta, "Object detection algorithm - YOLOv5 architecture: History and architecture of YOLOv5," Published in Analytics Vidhya, Aug 2, 2021, https://medium.com/analytics-vidhya/object-detection-algorithm-yolo-v5-architecture-89e0a35472ef.

[17] A. al Mamun, P. P. Em, and J. Hossen, "Lane marking detection using simple encode decode deep learning technique: SegNet," International Journal of Electrical and Computer Engineering, vol. 11, no. 4, pp. 3032–3039, 2021, doi: 10.11591/ijece.v11i4.pp3032-3039.

[18] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," International Journal of Computer Vision, vol. 88, no. 2, 2010, doi: 10.1007/s11263-009-0275-4.

[19] S. Saypadith and S. Aramvith, "Real-Time multiple face recognition using deep learning on embedded GPU system," 2018 Asia-Pacific Signal and Information Processing, Association Annual Summit and Conference (APSIPA ASC), pp. 1318-1324, 2018, doi: 10.23919/APSIPA.2018.8659751.