A novel Face Recognition System based on Jetson Nano developer kit

Thair A. salih* Mohammad Basman Gh.**
Departement of computer technical engineering, technical college, northern technical University
*thairali59@yahoo.com
**mbg.tcm09@gmail.com

ABSTRACT
Real-time face recognition became more important in the last two decade, it is adopted universally for attacking crime, stopping fraud, ensuring public safety. It proves that it is one of the most reliable biometrics security systems because frames can be taken from cameras without touching or interacting, and those images recorded and spontaneously validated with existing databases. The methodology achieved in this work will deal with several challenges like real-time recognition, cheap, portable, and reliable system. NVIDIA Jetson Nano is used in this work. It is a tiny, powerful AI computer that delivers the estimated performance to run advanced AI workloads with a small size, low-power, and low-cost. The main goal of this research is to deploy a deep learning model for face recognition in Jetson Nano, with the intention of real-time performance and high accuracy. The results of this study have shown great real-time performance in face recognition using Jetson Nano, Where it was processing 8.9 FPS in comparison with the latest raspberry pi edition 2.6 FPS.

Keywords: Face Recognition; Jetson Nano; Raspberry Pi; Surveillance system; real time systems RTS; Security systems.

INTRODUCTION
The key driver for the 4th Industrial Revolution is Artificial Intelligence [1]. It is being tightly interwoven into nearly all aspects of life, from our houses, cars, medical devices, robots, and entertainment. In recent years, systems have delivered the power for AI performance by infusing machine learning abilities with high bandwidth CPUs, GPUs, and specific AI accelerators. As an application for modern AI, the face recognition system has made headlines in the last few years, for the fast rise in companies and government agencies for using it in tracking and surveillance [2]. Face recognition has several advantages over other biometric such as iris and fingerprint: besides being simple and not intrusive, the significant advantage of face recognition is that it can be done at a distance and in a secret way. Between the six biometric characteristics considered by Heitmeyer [3], facial recognition got the highest compatibility in a Machine Readable Travel Documents (MRTD) [4]. The face recognition system based on several evaluation factors, such as recording, renewal, machine conditions, and public perception, shown in Figure1. Face recognition which is one of the leading biometric technologies has become more important due to the fast progress in image capture devices (surveillance systems, cameras in phones), availability of huge amounts of face images on the Web, and increased requests for
higher security. The first automated facial recognition system was produced in 1973 by Kanade in his Ph.D thesis research [5][6]. There was a dormant time in the automatic facial recognition system until Sirovich and Kirby [7] worked on a low-dimensional facial representation using the Principal Component Analysis. The innovation work of Turk and Pentland on Eigenface[8] has completely transformed facial recognition work. Other key milestones in facial recognition include the Fisherface method used by Linear Discriminant Analysis (LDA) after the PCA step to reach better accuracy; the use of local filters, to provide more effective facial features; and the scheme of the AdaBoost learning-based cascade classifier structure for real-time face detection[9]. The question is: how to build a Mobile, small-size, low-power, low-cost, and real-time face-recognition system, to be used in the security application.

Figure 1. A comparison of various biometric traits based on MRTD.
Ishita Gupta explored the feasibility of realizing a Raspberry Pi2 to perform a face recognition system by using conventional methods like Haar detection and PCA. This work was aim at producing a high-security system cost-effective, and simple to use, with high performance[10].

A. Soetedjo and I. K. Somawirata, have proposed a method executed on a low-cost embedded system based on the Raspberry Pi 3 Model B. The results show that the average rate was 98.3% for face detection, with a frame rate of 7.09 FPS[11].

N. R. Borkar and S. Kuwelkar proposed a composite face recognition algorithm by integrating a pair of face recognition methods that combin Linear Discriminant Analysis (LDA), Principle Component Analysis (PCA) based Raspberry pi 3. This combination method has given accuracy of 97%[12].

In 2019, Nafis Mustakim, Noushad Hossain, Mohammad Mustafizur, et al., present a strategy, used in face detection and face recognition in real-time frame steam. They implemented Raspberry Pi in this work. This system was able to process just four frames per second[13].

In 2020, Hansung Lee, So-Hee Park, et al., introduced the design of a friendly, stand-alone entrance control system using human facial recognition. In this work, they reduced the face feature size by 40% of the Gabor LBP facial characteristics. This system achieved facial recognition accuracy of 97.27% on an E-face dataset, with a 5.26 FPS processing speed in an indoor environment[14].

This paper aims to examine the ability to deploy a pre-trained model for face recognition on an embedded system, and achieving high accuracy while
simultaneously keeping the real-time work. A new development kit by Nvidia produces a small, specialized AI computer that provides the high performance to run new AI workloads, by giving 472 GFLOPs [15] for operating modern AI algorithms fast.

THE STRUCTURAL DESIGN

In this work, three MoC’s will be used and an RPI camera. The Table 1 below shows the specifications for each hardware[15],[16], [17].

1.1. Jetson Nano was implemented which is a low_cost AI computer from NVIDIA. It gives the high performance to work with modern AI workloads at a small size, also, It is power_efficient with 5 watts power consumption. Jetson Nano has an 80mm x 100mm dimension as Figure 2, with a 128_core Maxwell GPU, quad_core Arm Cortex-A57 CPU, and System Memory 4GB.

1.2. Raspberry pi 3 is A tiny, desktop computer that runs Linux, with a 1.4 GHz 64-bit quad-core ARM Cortex-A53 and 1GB of RAM.

1.3. Raspberry pi 4 is the latest product in the Raspberry Pi range of computers. It increases in processor speed, multimedia performance, memory, with a 1.5GHz quad-core processor, 1GB, 2GB, or 4GB of RAM.

1.4. The Raspberry Pi Camera Module V2-8 Megapixel, The v2 Camera Module has a Sony IMX219 8_megapixel and 1080p30 or 720p60 video.

Table 1: hardware specification [15],[16], [17].

| specification  | Jetson Nano | Raspberry Pi 4 | Raspberry Pi 3 |
|---------------|-------------|----------------|---------------|
| CPU           | Quad-core ARM A57 1.43 GHz | quad-core A72 (ARMv8-A) 64-bit @ 1.5GHz | 1.4 GHz 64-bit quad-core ARM Cortex-A53 |
| GPU           | 128-Core Nvidia Maxwell | Broadcom VideoCore VI | Broadcom VideoCore IV |
| RAM           | 4GB DDR4 | 4GB LPDDR4 | 1GB DDR2 |
| Connectivity  | Ethernet 10/100/1000 2.4 GHz and 5 GHz 802.11b/g/n/ac wireless LAN | 802.11ac Wi-Fi, Bluetooth LE 4.1, Ethernet |
| Ports         | 3x USB 2.0, 1x USB | 2 × micro-HDMI, 2 × USB 2.0 | 4x USB 2.0, HDMI |
2. THE PROPOSED METHODOLOGY

The proposed system implements a face recognition security system using a pre-trained deep learning model in the Jetson Nano developer kit.

2.1 facial recognition process
Any face recognition system usually consists of four main modules, as shown in the block diagram below in Figure 3: face detection, face normalization, Face stamp extraction, and matchings.

**Figure 3:** Block diagram of facial recognition.

**Face Detection:** It’s the first step in the face recognition system, and its accuracy has Hugh effects on the performance of the following operations. Face detection used to find human faces localization within the image or the frame.

**Image processing:** It’s techniques that are used to preparing and improving images or video quality. In this part, after detecting the faces within the frame, The faces will be crop using the coordinations for these faces. Face Alignment will be the second stage in this process to increase the accuracy of face recognition. Finally, Resize the image size to 224x224 to speed up the recognition time.

**Face stamp Extraction:** It’s a process to generate 128 measurements from each face. Any different image of the same person should produce approximately the same measurements. These measurements should be collected from each face to create a known face database. This process of training CNN to output the face stamp, requires a lot of data and a powerful computer.

2.2 Important libraries definitions
Nvidia SDK Manager will be used to flash our Jetson nano with the operating system image and install the libraries and APIs. Some additional Python libraries like dlib, NumPy, and face_recognition is required to be installed.

**Dlib**: is a landmark’s facial detector with pre-trained models [18], dlib is used to predict the position of 68 coordinates that map the facial features on a person’s face like the image below in Figure 3(b). This pre-trained model was trained by the ibug 300-W dataset [19].

**OpenCV**: is a real-time open-source library for machine learning and computer vision [20]. OpenCV was created to accommodate a common base for computer vision applications and to speed up the use of machine knowledge in commercial products.

**Face_recognition**: is a library Created using Dlib’s state-of-the-art pre-trained model, which is built by a Deep Convolutional Neural Network training. The model has an accuracy of about 99.38% on the Labeled Faces in the Wild benchmark. This library gave 128 measurements for every face. The measurements are almost the same, as the pictures belong to the same person.

**Pickle**: it is a library that used to serialize and deserialize Any object in python so that it can be saved on any disk. Pickling is a process that transforms a python object like (list, dict) into a character stream. This character stream carries all the data that required to reconstruct the object.

### 2.3 Experimental Setup

The proposed system is based on the Jetson Nano developer kit, which has a full Linux desktop called Linux4Tegra OS as an operating system. To get real-time videos, the v2 Camera Module with a Sony IMX219 8_megapixel, and 1080p30 or 720p60 video is used. Using the Python programming language in Jetson Nano for image processing, the OpenCV library [20] frequently captures frames from the V2 camera, see Figure4(a). These frames are read one by one to find faces locations. Face detection discovered if there are faces in a given frame or not. If present, it returns the location and the contents of each face within the frame using the dlib library [19], and cropped these faces from the frames, Figure4(c).

![Figure 4](image)

Figure 4: (a) capture the frame. (b) detect the face, (c) landmark detection.

The face normalization module is used to normalize the face feature. This is important because the state-of-the-art recognition systems are expected to identify face images with a
different pose. The normalization method reconstructs the face into a standard frame by face cropping. Warping may be used for more elaborate geometric normalization. Face alignment is the next step after detecting the face and cropping it from the frame. Google announced that face alignment improves the accuracy of its facial recognition model FaceNet from 98.87% to 99.63% [21]. This is nearly a 1% accuracy enhancement. the Eye's position should be detected using the dlib library [19]. Draw a line between these two points, Figure 5(a). Determine the angle between the horizontal line and the drawn line, the rotation for the image based on this phase, Figure 5(b). In the last step, resize the output image to 224*224 size, Figure 5(c).

Figure 5 : (a) Eyes’ line, (b) face alignment, (c) image resize, (d) face stamp.

The face stamp extraction is executed on the normalized face to extract remarkable information to recognizing faces of different persons. This process is used to generate 128 special measurements from each face, Figure 6. This face stamp can estimate the similarity within the database with 99.38 % accuracy. In face matching, the extracted face stamp from the frame matched to many of the registered faces in the database. The output is identity the input face when the match is found with 65% confidence at least.

Figure 6 :Face matching.
If the system detects the face that isn’t in the database, the system will displays “Unregistered”, and if the face has successfully recognized and shows the name and the confidence rate of the recognition, the face already stored in the database.

3- RESULTS

Table 2 shows the performance in frame per second, with 1280*720 camera resolution. Where The calculation for the performance in frame per second by the equation:

$$FPS = \frac{C}{(T_{s2} - T_{s1})}$$

Where;

$T_{s1}$ = start time

$T_{s2}$ = end time

C is the frame counter within this period

| Embedded system | Face detection/FPS | Face recognition/FPS |
|-----------------|-------------------|---------------------|
| Raspberry pi 3  | 4.4               | 1.1                 |
| Raspberry pi 4  | 10.8              | 2.6                 |
| Jetson Nano     | 12.1              | 8.9                 |

The real-time is necessary to make the facial recognition system efficient to be adopted in many security applications. The proposed system works excellent with the Jetson Nano in real-time in comparison with the raspberry pi platform, as shown in Figure 7. The accuracy shows more than 96% in facial recognition under normal conditions such as lightning, and the distance from the camera. However, The best distance from the camera with the same resolution from 1 to 3 meters. Unlike the security systems available in the market, this solution gave a real-time and secure automated approach at a lower price. Finally, the results of this system were validated and compared with many previous systems. This system shows higher performance in real-time in terms of speed and robustness.

Figure 7: Face detection and recognition in ms.
CONCLUSION

This system was successfully implemented and tested in real-time using three MoC’s. The results that taken were satisfying. This system is built on the Jetson Nano, Raspberry pi 4, and Raspberry pi 3; these MoC’s consumes low power. These results shows deffirent performace according to there hardware specifications. The Raspberry Pi 4 was recently released with some excellent upgrades, compared to the previous Raspberry Pi 3 B+. In comparison, the NVIDIA Jetson Nano Developer Kit provides a board that offers a similar feature set. The Jetson Nano has a much more powerful processor than the Raspberry Pi3 and having an additional dedicated GPU in comparison with Raspberry Pi4. The Jetson Nano is much faster at AI tasks, due to the additional GPU supported by NVIDIA’s TensorRT, Which has accelerator library included with JetPack 4.2 NVIDIA. TensorRT is a deep learning platform. It introduces a deep learning optimizer and run-time that achieves lower latency and higher throughput for deep learning applications.

References

[1] and F. H. Skilton, Mark, The 4th industrial revolution: Responding to the impact of artificial intelligence on business. 2017.
[2] S. Singh and S. V. A. V. Prasad, “Techniques and challenges of face recognition: A critical review,” *Procedia Comput. Sci.*, vol. 143, pp. 536–543, 2018, doi: 10.1016/j.procs.2018.10.427.
[3] R. Heitmeyer, “Biometric identification promises fast and secure processing of airline passengers,” *ICAO J.*, vol. 55, no. (9), pp. 10–11, 2000.
[4] T. Lonsdale, “Technical Advisory Group on Machine Readable Travel Documents,” *Work*, 2007.
[5] T. Kanade, “Picture processing by computer complex and recognition of human faces,” Kyoto University, 1973.
[6] T. Huang, Z. Xiong, and Z. Zhang, *Face Recognition Applications*. 2011.
[7] M. A. O. Vasilescu and D. Terzopoulos, “Multilinear analysis of image ensembles: Tensorfaces,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2350, pp. 447–460, 2002, doi: 10.1007/3-540-47969-4_30.
[8] M. H. Yang, “Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods,” *Proc. - 5th IEEE Int. Conf. Autom. Face Gesture Recognition, FGR 2002*, pp. 215–220, 2002, doi: 10.1109/AFGR.2002.1004157.
[9] P. Viola and M. Jones, “Haar-like,” *Cvpr*, vol. 1, pp. I-511-I-518, 2001, doi: 10.1109/CVPR.2001.990517.
[10] Gupta, “Face Detection and Recognition using Local,” no. December, pp. 7923–7929, 2016, doi: 10.15662/IJAREEIE.2016.0510018.
[11] A. Soetedjo and I. K. Somawirata, “Implementation of face detection and tracking on a low cost embedded system using fusion technique,” *ICCSE 2016 - 11th Int. Conf. Comput. Sci. Educ.*, no. Iccse, pp. 209–213, 2016, doi: 10.1109/ICCSE.2016.7581582.
[12] N. R. Borkar and S. Kuwelkar, “Real-Time implementation of face recognition system,” *Proc. Int. Conf. Comput. Methodol. Commun. ICCMC 2017*, vol. 2018-
[13] N. Mustakim, N. Hossain, M. M. Rahman, N. Islam, Z. H. Sayem, and M. A. Z. Mamun, “Face Recognition System Based on Raspberry Pi Platform,” *1st Int. Conf. Adv. Sci. Eng. Robot. Technol. 2019, ICASERT 2019*, vol. 2019, no. Icasert, pp. 1–4, 2019, doi: 10.1109/ICASERT.2019.8934485.

[14] H. Lee, S. H. Park, J. H. Yoo, S. H. Jung, and J. H. Huh, “Face recognition at a distance for a stand-alone access control system,” *Sensors (Switzerland)*, vol. 20, no. 3, 2020, doi: 10.3390/s20030785.

[15] “Jetson Nano Developer Kit.” https://developer.nvidia.com/embedded/jetson-nano-developer-kit.

[16] “Raspberry Pi 4 Tech Specs.” https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/.

[17] “Raspberry Pi 3 Tech Specs.” https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/.

[18] N. Boyko, O. Basystiuk, and N. Shakhovska, “Performance Evaluation and Comparison of Software for Face Recognition, Based on Dlib and Opencv Library,” *Proc. 2018 IEEE 2nd Int. Conf. Data Stream Min. Process. DSMP 2018*, pp. 478–482, 2018, doi: 10.1109/DSMP.2018.8478556.

[19] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “300 Faces In-The-Wild Challenge: database and results,” *Image Vis. Comput.*, vol. 47, pp. 3–18, 2016, doi: 10.1016/j.imavis.2016.01.002.

[20] A. A. Wazwaz, A. O. Herbawi, M. J. Teeti, and S. Y. Hmeed, “Raspberry Pi and computers-based face detection and recognition system,” *2018 4th Int. Conf. Comput. Technol. Appl. ICCTA 2018*, pp. 171–174, 2018, doi: 10.1109/CATA.2018.8398677.

[21] X. Wen, G., Chen, H., Cai, D., & He, “Improving face recognition with domain adaptation,” *Neurocomputing*, vol. 287, no. 2018, pp. 45–51, 2018.