ISSN 1870-4069

Low-Resource Hardware Performance Analysis for Real-Time Facial Recognition

Edgar Abidán Padilla Luis[1], Fernando Perez-Tellez[2], David Pinto[1]

Benemérita Universidad Autónoma de Puebla,
Faculty of Computer science,
Mexico

Technological University Dublin,
Faculty of Computing Digital and Data,
Ireland

edgar.abidan.pl@gmail.com, david.pinto@correo.buap.mx,
Fernando.PerezTellez@tudublin.ie

Abstract. Real-time facial recognition technology faces challenges on low-resource hardware due to processing and memory limitations. This study analyzes the performance of the Raspberry Pi 4 and 5 compared to a standard desktop computer, evaluating CPU usage, RAM, temperature, and latency across five different models. The results highlight the limitations and feasibility of these devices, providing a guide for selecting appropriate hardware based on performance and resources.

Keywords: Face recognition, real-time processing, low-resource hardware.

1 Introduction

Facial recognition is a biometric technology that identifies or verifies a person's identity by analyzing facial features. This process involves face detection in images, extraction of distinctive features, and comparison with a database [15]. However, it faces ethical challenges related to privacy and bias [15]. Additionally, technical issues such as variability in lighting conditions and facial expressions exist [14]. Its applications range from security and commerce to government and healthcare [8]. Currently, facial recognition is used as a real-time tool [5], [1] and [10]. Nevertheless, choosing a device with limited computational resources to integrate this technology is always challenging. Therefore, this article compares the performance of several facial recognition models between two widely used hardware platforms in real-time applications (Raspberry Pi 4 and Raspberry Pi 5) and a conventional desktop computer.

2 Related Work

Facial recognition is a widely studied task in the field of computer vision. There are numerous state-of-the-art articles that present facial recognition systems, which are implemented on various types of hardware. For example, in [19,6,10,25,7] we can find the implementation and analysis of facial recognition tasks on Jetson platforms. Another commonly used hardware is the Raspberry Pi, as shown in [13,1,19,14], this type of board is frequently used in facial recognition tasks due to its versatility and efficiency. In this paper, we used the Raspberry Pi 4 and Raspberry Pi 5 boards to conduct the experiments.

Dewantoro *et al* [6] conduct a performance comparison of a line-following robot using the Raspberry Pi and Jetson Nano as CPU controllers and find that the accuracy in the task of recognizing line paths is 96% when using the Raspberry Pi, while it is 98% when using the Jetson Nano.

Manni *et al* [11] performed a study at the "Smart Living Technologies" laboratory of the Institute for Microelectronics and Microsystems (IMM) in Lecce, Italy, to validate a proposed approach for real-time heart rate monitoring using various hardware platforms, including Raspberry Pi 4, Odroid N2+, and Jetson Nano. The experiments revealed that Raspberry Pi 4 exhibited the highest CPU consumption due to the lack of a GPU, impacting the execution time of the face detection block using the Dlib library. Additionally, it was observed that Raspberry Pi 4 showed slightly higher memory usage compared to the other platforms, although all displayed similar behaviors in terms of memory usage. Concerning power consumption, Raspberry Pi 4 recorded the lowest consumption, likely due to the absence of a cooling unit. In terms of pipeline accuracy, high accuracy was achieved with a resolution of 640×480 and a distance of 0.5 m across all evaluated platforms. However, Raspberry Pi 4 stood out for its inferior performance compared to Odroid N2+ and Jetson Nano, which demonstrated execution times comparable to a standard PC and slightly lower than a laptop.

Biglaru and Tang [4] evaluated the performance of various machine learning packages when running trained models on different edge hardware platforms. Latency, memory footprint, and energy consumption were compared across the AlexNet and SqueezeNet neural network models. MXNet performed well on MacBook, while TensorFlow showed good performance on FogNode. On Jetson TX2, PyTorch exhibited shorter inference times compared to TensorFlow on AlexNet. Caffe2 demonstrated efficiency in executing SqueezeNet on Raspberry Pi, albeit with memory limitations. It was identified that model loading takes more time than inference in some packages, indicating opportunities for edge optimization. Memory and energy consumption showed a varied trade-off among packages, with MXNet noted for energy efficiency and PyTorch for lower memory usage on Jetson TX2.

Baobaid *et al* [2] conducted a study on facial detection and recognition systems, comparing neural network-based and non-neural network-based algorithms. They found that neural network algorithms, like FaceNet, outperform non-neural ones in terms of accuracy. The performance was tested on different hardware accelerators such as Raspberry Pi, Jetson Nano GPU, and GTX1060 GPU. Raspberry Pi showed limited performance, with a processing rate of less than 1 FPS for detection and an average of 1.5 FPS for recognition using neural network algorithms. However, its performance significantly improved with the use of hybrid accelerators like Intel Movidius. While GPUs proved superior in accuracy and processing time, FPGAs were noted for their attractiveness in power consumption and execution time, suggesting a future approach of heterogeneous systems combining these technologies.

Khan *et al* [9] carry out a comparative study of three facial recognition algorithms on multicore systems, evaluating their speed and accuracy by taking 13 samples per person. The system's compatibility across different machines, including low-spec configurations, was demonstrated, emphasizing the use of Raspberry Pi to eliminate platform dependencies. Results presented in tables and graphs indicated that the LBPH algorithm was the most accurate, achieving up to 90% accuracy but with higher time consumption due to its binary pattern computation. In contrast, the Fisher Face algorithm was the fastest but least accurate, reaching a maximum of 85% accuracy.

Although Jetson Nano models typically outperform Raspberry Pi in performance, they require more computational resources and are more expensive. Motivated by these differences, this study focuses exclusively on the popular Raspberry Pi 4 and Raspberry Pi 5 models.

3 Experiment Description

In the reviewed literature, facial recognition systems conduct experiments in highly controlled environments, using cameras that focus on faces at very short distances, as shown in these papers [19,2,1,26,13,9,22,12], Fig. 1 shows an example of a typical image used in these analyses. This setup allows facial recognition models to achieve precise results. However, in a real-world environment, this design is inadequate, as cameras are not always positioned at the same distance, and people's faces are not close to the lens. For this reason, we designed our experiment to emulate a real-world setting.

The experiment was conducted in an office environment. Fig. 2 shows a floor plan of the location where the experiment took place. The gray areas represent desk sections, while the white areas indicate open spaces where people can walk. The room has only one entrance and exit door. For the experiment, a camera was placed next to Desk Section A, pointing toward the door. Four individuals participated, each performing the following steps: entering the office through the door, walking from the door to the camera, passing in front of the camera while still within its range, moving toward Desk Section B, exiting the camera's range, and moving toward Desk Section C. Fig. 2 illustrates the path taken by the participants.

Among the participants, only two were included in the database with images taken at various distances within the camera's range. Fig. 3 shows some images from the database.

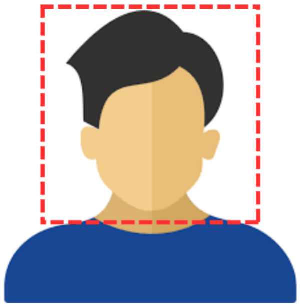


Fig. 1. Example of a typical image analyzed in state-of-the-art papers.

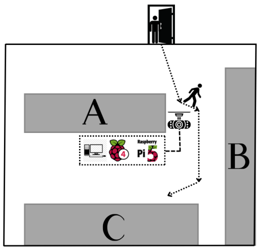
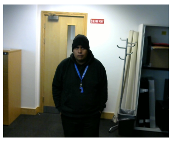


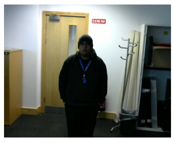
Fig. 2. In this experiment a person opens the door, walks in front of the webcam connected to one of the three devices used.


(a) Image 1 of the person 1.


(b) Image 2 of the person 1.


(c) Image 3 of the person 1.

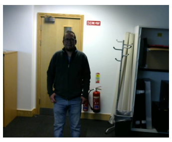

(d) Image 4 of the person 1.


(e) Image 5 of the person 1.


(f) Image 1 of the person 2.


(g) Image 2 of the person 2.


(h) Image 3 of the person 2.


(i) Image 4 of the person 2.


(j) Image 5 of the person 2.

Fig. 3. Images in the Database.

3.1 Methodology

We developed a system based on the methodology shown in Fig. 4. The system flow is as follows: a camera records in real time, capturing images with a size of

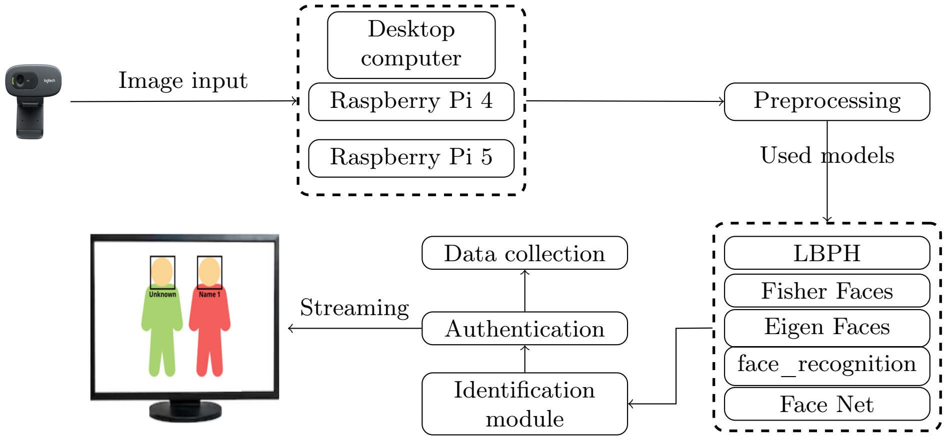


Fig. 4. Methodology used to evaluate performance on hardware with limited resources.

1280px × 720px. Subsequently, one of three devices is used: a desktop computer, a Raspberry Pi 4, or a Raspberry Pi 5. On the selected device, each image goes through a preprocessing module where it is converted to grayscale and resized to 20% of its original size. This resizing is done to reduce latency and enable real-time facial recognition.

Each of the devices runs five facial recognition models: Local Binary Patterns Histogram (LBPH) [12], Fisher Faces (FF) [3] , Eigen Faces (EF) [16], Face_recognition (FR) [7], and Face Net (FN) [13]. Each model is run separately, meaning the five models are not executed simultaneously. The models were previously trained to recognize two people. Each image is analyzed by the models in the identification module. Then, in the authentication module, if a person from the database matches an analyzed face, the name is assigned; otherwise, they are labeled as 'unknown'. Additionally, a rectangle is drawn around the detected face. Finally, the image with the results from the authentication module is displayed on a monitor. At the same time, another module stores the data for later analysis.

Hardware overview Below, we present the characteristics of the devices used to conduct this study.

Webcam The Logitech C270 HD Webcam is a budget-friendly webcam that offers clear HD 720p video calls, noise-canceling microphone, automatic light correction, and a wide field of view. It's easy to set up and use, making it a great choice for basic video calls and streaming.

(a) Configuration of the Raspberry Pi 4 with accessories.

(b) Single-board view of the Raspberry Pi 4.

Fig. 5. Overview of the Raspberry Pi 4 hardware configuration and its single-board computer.

Desktop Computer The workstation is a Dell OptiPlex 5050 with 32GB of RAM, an Intel Core i5-6500 processor, Mesa Intel HD Graphics 530, and 512GB of storage. It runs Ubuntu 22.04.4 LTS.

Raspberry Pi 4 The Raspberry Pi 4, a powerful single-board computer is shown in Fig.5b, running on the latest Bookworm OS, boasts a Broadcom BCM2711 SoC, quad-core Cortex-A72 CPU at 1.8GHz, and LPDDR4-3200 SDRAM memory of 8GB. For enhanced protection and improved thermal performance, the Raspberry Pi 4 can be equipped with a protective case and heatsinks. These additions safeguard the board from physical damage and ensure efficient heat dissipation during operation, extending its lifespan and maintaining stable performance as illustrated in Fig. 5a, especially when running demanding tasks.

Raspberry Pi 5 The Raspberry Pi 5 with Raspberry Pi Bookworm operating system is the latest and most powerful single-board computer from the Raspberry Pi Foundation this board is shown in Fig. 6b. The Raspberry Pi 5 boasts a powerful Broadcom BCM2712 quad-core ARM Cortex-A76 processor at 2.4 GHz, 8GB of LPDDR4X-4267 RAM, and a VideoCore VII GPU for superior performance. For further expansion, it includes a PCIe 2.0 x1 interface. The board is powered by a 5V/5A USB-C port and features a real-time clock. Finally, the Raspberry Pi 5 incorporates a power button for more intuitive device control. This device has been enhanced with a protective case, heat sinks, and a fan for better performance as illustrated in Fig. 6a.

4 Experimental Results

This section presents the results of the experiments, focusing on four key resources that devices consume and are critical for software implementation: CPU usage, RAM consumption during task execution, device temperature, and latency in real-time model execution.

(a) Configuration of the Raspberry Pi 5 with accessories.

(b) Single-board view of the Raspberry Pi 5.

Fig. 6. Overview of the Raspberry Pi 5 hardware configuration and its single-board computer.

Tab. 1. Latency on Desktop, Raspberry Pi 4 and Raspberry Pi 5.

Model	Desktop				Raspberry Pi 4				Raspberry Pi 5			
	min	max	mean	std	min	max	mean	std	min	max	mean	std
FF	0.01	0.16	**0.03**	0.01	0.04	0.35	0.05	0.02	0.02	0.12	0.03	0.01
EF	0.01	0.16	0.03	0.01	0.04	0.35	0.05	0.02	0.02	0.12	0.03	0.01
LBPH	0.01	0.17	0.03	0.01	0.04	**3.61**	0.08	**0.31**	0.02	1.95	0.04	0.16
FR	0.02	0.24	0.06	0.06	0.10	0.50	**0.18**	0.15	0.05	0.26	0.08	0.07
FN	0.02	0.24	**0.07**	0.04	0.12	1.09	**0.45**	0.32	0.05	0.92	0.18	0.15

In order to analyze the performance of the devices, the average and standard deviation were obtained from calculated metrics. Additionally, the minimum and maximum range achieved by the data was also determined.

4.1 Latency

Table 1 shows data obtained on latency in seconds across three hardware platforms (Desktop, Raspberry Pi 4, and Raspberry Pi 5). Each row represents one of the five different models that were implemented.

The Desktop consistently shows the lowest latency for all models, with averages ranging from 0.03 to 0.07 seconds, highlighting its superior processing capabilities. Comparing the Raspberry Pi devices, the Raspberry Pi 5 demonstrates better performance than the Raspberry Pi 4, with lower average and maximum latencies, as well as more consistent results. Regarding the models, FF and EF yield very similar results and maintain low latency across all hardware, while LBPH performs well on Desktop and Raspberry Pi 5 but exhibits significantly higher latency on Raspberry Pi 4 (maximum of 3.61 seconds and a standard deviation of 0.31). On the other hand, FR and FN are the most demanding models, especially on Raspberry Pi 4, where they reach averages of 0.18 and 0.45 seconds, respectively, although they perform better on Desktop and Raspberry Pi 5. This indicates that simpler models like FF and EF are suitable even for low-power devices like the Raspberry Pi, while more complex models such as FR and FN may not be ideal for Raspberry Pi 4

Tab. 2. RAM consumed on Desktop, Raspberry Pi 4 and Raspberry Pi 5.

Model	Desktop				Raspberry Pi 4				Raspberry Pi 5			
	min	**max**	**mean**	**std**	**min**	**max**	**mean**	**std**	**min**	**max**	**mean**	**std**
FF	2	1068	949	248	18.88	944	721	252	1	954	786	245
EF	2	1068	947	249	1	933	698	280	1	954	787	236
LBPH	1	1064	941	246	1	950	672	279	1	932	718	277
FR	1	1064	948	246	70	933	**755**	243	65	926	**768**	**240**
FN	1	1070	950	249	1	934	**644**	254	1	984	769	253

due to high latency and variability, though they are more feasible on Raspberry Pi 5. Overall, for real-time or latency-sensitive applications, Desktop remains the best option, while Raspberry Pi 5 could be a viable alternative for simpler models or optimized complex models, and Raspberry Pi 4 may be insufficient for demanding applications.

4.2 RAM Memory Consumed

Table 2 shows data obtained on RAM usage when running the models on the three types of hardware. These data are measured in MB.

On Desktop, all models show average values close to 950 MB with consistent standard deviations between 246 and 249 MB, indicating lower variability compared to other platforms. On Raspberry Pi 4, the average RAM consumption is lower than on Desktop, ranging from 644 MB (FN) to 755 MB (FR), but with higher variability, reaching standard deviations of up to 280 MB, suggesting less stable performance; additionally, some models, such as EF, have minimum RAM values of 1. On the other hand, Raspberry Pi 5 demonstrates better performance in terms of RAM consumption compared to Raspberry Pi 4, with higher but more consistent average values and lower standard deviations, indicating more stable behavior; for instance, the FR model has an average consumption of 768 MB with a low standard deviation of 240 MB, while LBPH shows the lowest average consumption across all platforms, standing out for its RAM efficiency. Comparing the platforms, Desktop uses more RAM on average, likely due to its higher processing capacity, while Raspberry Pi 5 proves to be more efficient than Raspberry Pi 4 by achieving a better balance between average consumption and stability. Regarding the models, FF shows high average consumption on Desktop and Raspberry Pi 5, while LBPH stands out as the most efficient in terms of RAM across all platforms, suggesting that the choice of model and hardware will depend on system priorities, such as stability, lower RAM consumption, or overall efficiency, with Raspberry Pi 5 being a notable improvement over Raspberry Pi 4.

4.3 CPU Usage

Table 3 present the CPU usage percentages for the models across the platforms.

Tab. 3. Usage CPU on Desktop, Raspberry Pi 4 and Raspberry Pi 5 with the five models.

CPU	**Desktop**				**Raspberry Pi 4**				**Raspberry Pi 5**			
	min	**max**	**mean**	**std**	**min**	**max**	**mean**	**std**	**min**	**max**	**mean**	**std**
FF model												
CPU 1	0	100	17.31	10.29	0	94.9	17.14	13.04	0	60	8.57	7.73
CPU 2	7.10	66.7	20.39	9.29	0	70	17.65	10.13	0	40	8.80	6.32
CPU 3	0	100	21.33	15.23	0	100	41.62	17.17	0	82.5	17.45	11.54
CPU 4	4.20	91.7	18.14	10.19	0	75	18.98	13.70	0	50	22.45	9.14
EF model												
CPU 1	0.00	100.00	17.57	10.03	7.10	86.70	22.79	14.76	0.00	75.00	29.09	16.98
CPU 2	4.00	76.90	22.19	10.84	0.00	58.30	19.06	10.69	0.00	57.10	18.19	11.26
CPU 3	0.00	100.00	18.53	10.82	7.10	100.00	29.62	19.39	0.00	40.00	11.90	10.82
CPU 4	0.00	100.00	17.37	12.82	0.00	75.00	29.09	16.98	0.00	89.50	13.33	11.64
LBPH model												
CPU 1	0.00	50.00	16.64	7.33	0.00	70.00	22.79	15.62	0.00	100.00	26.42	18.85
CPU 2	0.00	100.00	22.49	12.76	0.00	61.10	26.84	15.27	0.00	60.00	10.59	9.75
CPU 3	7.70	66.70	18.78	8.84	0.00	68.80	17.01	9.04	0.00	40.00	7.93	7.59
CPU 4	7.70	100.00	18.49	12.84	7.10	100.00	31.42	21.81	0.00	50.00	12.29	10.12
FR model												
CPU 1	0	100	15.10	16.93	0	100	15.10	16.93	0	35.70	7.42	7.20
CPU 2	0	100	24.74	29.36	0	100	24.74	29.36	0	72.70	7.13	12.58
CPU 3	0	83.30	10.71	13.34	0	83.30	10.71	13.34	0	100	21.77	29.17
CPU 4	0	100	18.26	23.64	0	100	18.26	23.64	0	100	39.84	32.72
FN model												
CPU 1	19.00	100.00	61.73	19.39	0.00	100.00	18.26	23.64	41.90	100.00	99.27	5.81
CPU 2	20.10	100.00	62.44	19.26	20.10	100.00	62.44	19.26	37.10	100.00	99.39	6.17
CPU 3	30.80	100.00	61.10	20.30	30.80	100.00	61.10	20.30	36.60	100.00	89.35	19.41
CPU 4	37.50	100.00	64.86	18.19	37.50	100.00	64.86	18.19	36.90	100.00	98.78	8.64

When analyzing the data from the tables showing CPU usage for the face recognition models, it was observed that CPU usage on the Desktop is generally higher compared to the Raspberry Pi 4 and Raspberry Pi 5, reflecting the greater processing power of the desktop hardware.

The Raspberry Pi 4 and Raspberry Pi 5 have lower CPU usage peaks and greater variations, indicating less stable and more fluctuating processing capabilities, with a higher standard deviation, suggesting that their workload is less consistent than on the Desktop.

Although the Raspberry Pi 5 shows slight performance improvements compared to the Raspberry Pi 4, both still exhibit inferior performance compared to the Desktop, with the Raspberry Pi 5 reaching an average performance close to the Desktop in some models, but with greater variability and higher maximum loads.

In terms of CPU demand, the FN model (which is more resource-intensive) shows the highest CPU usage values, especially on the Desktop, indicating that this model is more demanding in terms of processing. In contrast, the LBPH and FR models present a more moderate CPU load, making them more suitable for resource-limited platforms like the Raspberry Pi.

We can conclude that the Raspberry Pi 4 and 5, although more affordable and energy-efficient, exhibit more variable and less powerful performance compared to the Desktop. More demanding models like FN are better suited for the Desktop, while models like FF, EF, and LBPH can be run efficiently on the Raspberry Pi, although with somewhat less stable performance.

4.4 Temperature

The temperature data, measured in degrees Celsius, is presented in Table 4.

Tab. 4. Temperature on Desktop, Raspberry Pi 4 and Raspberry Pi 5.

Model	Desktop				Raspberry Pi 4				Raspberry Pi 5			
	min	**max**	**mean**	**std**	**min**	**max**	**mean**	**std**	**min**	**max**	**mean**	**std**
FF	28	28	28	0	53	57	54	0.88	53	61	55	1.61
EF	28	28	28	0	56	60	57	0.92	54	62	56	1.60
LBPH	28	28	28	0	**48**	51	50	0.57	**52**	58	54	1.06
FR	28	28	28	0	60	65	62	0.90	57	62	59	0.95
FN	28	28	28	0	66	**72**	**69**	1.90	64	**73**	70	2.18

Based on the temperature data from the table, it can be observed that the temperatures on the Desktop remain constant at 28°C across all models, suggesting that the Desktop hardware maintains a stable and low temperature during operation, likely due to a superior cooling system. On the other hand, the temperatures on Raspberry Pi 4 are significantly higher, ranging from 48°C to 72°C, particularly for the FN model, which is more demanding. The average temperatures for this model are between 49°C and 69°C, with a standard deviation ranging from 0.57 to 1.90. Raspberry Pi 5, in contrast, shows slightly lower temperatures than Raspberry Pi 4, with minimum temperatures around 52°C and maximum temperatures reaching up to 73°C. The average temperatures range from 54°C to 69°C, and the standard deviation varies between 0.95 and 2.18, reflecting a level of thermal variability similar to that of Raspberry Pi 4. The FN model generates the highest temperatures on both Raspberry Pi 4 and Raspberry Pi 5, surpassing 70°C, while the FF and LBPH models maintain lower and more stable temperatures across all platforms. The FR and EF models exhibit intermediate temperatures, with Raspberry Pi 4 reaching higher temperatures compared to Raspberry Pi 5. In general, we can observe that while the Desktop maintains low and stable temperatures, the Raspberry Pi 4 and Raspberry Pi 5 exhibit higher and more variable temperatures, particularly during intensive processing, with Raspberry Pi 5 providing slightly better thermal performance than Raspberry Pi 4.

5 Conclusions and Future Work

Our study demonstrates that desktop computers outperform Raspberry Pi platforms in terms of latency, CPU usage, and RAM consumption, as anticipated. Desktop systems consistently maintain low and stable latencies across all evaluated models, while Raspberry Pi devices, particularly the Raspberry Pi 4 exhibit higher latency and greater variability. This suggests that Raspberry Pi platforms are less suitable for high-demand or time-sensitive applications. Although the Raspberry Pi 5 offers a more stable performance compared to the Raspberry Pi 4, it still falls short of the desktop computer in both latency and CPU usage.

In terms of RAM consumption, the desktop computer records higher average usage but with lower variability, while the Raspberry Pi 5 demonstrates improved efficiency relative to the Raspberry Pi 4, which shows more inconsistent performance and greater fluctuations. Regarding thermal behavior, the desktop computer maintains low and stable operating temperatures, whereas Raspberry Pi devices experience higher and more variable temperatures, particularly when running more demanding models such as FN.

Overall, while the Raspberry Pi 4 and 5 present more affordable and energy-efficient alternatives, their performance remains more variable and generally inferior compared to desktop systems. Consequently, more demanding models like FN are better suited for execution on desktop computers, whereas simpler models such as FF, EF, and LBPH can be deployed effectively on Raspberry Pi devices, albeit with slightly less stable performance.

From these findings, it can be inferred that security systems, which require low latency and high stability, are better suited to desktop platforms. Conversely, simpler facial recognition models can be effectively implemented on Raspberry Pi devices, accepting some degree of reduced stability. Similarly, home automation applications may benefit from the use of Raspberry Pi platforms, particularly when utilizing less demanding models, thereby taking advantage of their energy efficiency and low cost.

Future work will focus on expanding the dataset by incorporating a larger number of identities and images to improve the generalizability of the results. Advanced facial recognition models will be evaluated to enhance performance, particularly on low-resource devices. Additionally, the hardware comparison will be broadened to include popular edge platforms such as the Jetson Nano and Odroid N2+, providing a more comprehensive understanding of system performance across different environments. Optimization techniques will be explored, along with the integration of hybrid hardware accelerators, to improve the efficiency of more complex models. Furthermore, a deeper thermal analysis will be conducted, and energy consumption metrics will be incorporated, recognizing their critical role in the evaluation of embedded systems.

References

1. Anusudha, K., et al.: Real time face recognition system based on yolo and insightface. Multimedia Tools and Applications, vol. 83, no. 11, pp. 31893–31910 (2024)
2. Baobaid, A., Meribout, M., Tiwari, V. K., Pena, J. P.: Hardware accelerators for real-time face recognition: A survey. IEEE Access, vol. 10, pp. 83723–83739 (2022)
3. Belhumeur, P. N., Hespanha, J. P., Kriegman, D. J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In: Computer Vision—ECCV'96: 4th European Conference on Computer Vision Cambridge, UK, April 15–18, 1996 Proceedings, Volume I 4. pp. 43–58. Springer (1996)
4. Biglari, A., Tang, W.: A review of embedded machine learning based on hardware, application, and sensing scheme. Sensors, vol. 23, no. 4, pp. 2131 (2023)
5. Deeba, F., Memon, H., Dharejo, F. A., Ahmed, A., Ghaffar, A.: Lbph-based enhanced real-time face recognition. International Journal of Advanced Computer Science and Applications, vol. 10, no. 5 (2019)

6. Dewantoro, G., Mansuri, J., Setiaji, F. D.: Comparative study of computer vision based line followers using raspberry pi and jetson nano. Jurnal Rekayasa Elektrika, vol. 17, no. 4 (2021)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
8. Kaur, P., Krishan, K., Sharma, S. K., Kanchan, T.: Facial-recognition algorithms: A literature review. Medicine, Science and the Law, vol. 60, no. 2, pp. 131–139 (2020)
9. Khan, M. A., Shaikh, M. K., bin Mazhar, S. A., Mehboob, K., et al.: Comparative analysis for a real time face recognition system using raspberry pi. In: 2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA). pp. 1–4. IEEE (2017)
10. Kocacinar, B., Tas, B., Akbulut, F. P., Catal, C., Mishra, D.: A real-time cnn-based lightweight mobile masked face recognition system. Ieee Access, vol. 10, pp. 63496–63507 (2022)
11. Manni, A., Caroppo, A., Rescio, G., Siciliano, P., Leone, A.: Benchmarking of contactless heart rate measurement systems in arm-based embedded platforms. Sensors, vol. 23, no. 7, pp. 3507 (2023)
12. Rodriguez, Y., Marcel, S.: Face authentication using adapted local binary pattern histograms. In: Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part IV 9. pp. 321–332. Springer (2006)
13. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
14. Singh, S., Prasad, S.: Techniques and challenges of face recognition: A critical review. Procedia computer science, vol. 143, pp. 536–543 (2018)
15. Smith, M., Miller, S.: The ethical application of biometric facial recognition technology. Ai & Society, vol. 37, no. 1, pp. 167–175 (2022)
16. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of cognitive neuroscience, vol. 3, no. 1, pp. 71–86 (1991)