

# Biometric face recognition method using graphics processing unit system

Abeer A. Mohamad Alshiha<sup>1</sup>, Mohammed Wajid Al-Neama<sup>2</sup>, Abdalrahman R. Qubaa<sup>1</sup>

<sup>1</sup>Remote Sensing Center, University of Mosul, Mosul, Iraq

<sup>2</sup>Education College for Girls, University of Mosul, Mosul, Iraq

## Article Info

### Article history:

Received Jul 6, 2022

Revised Nov 7, 2022

Accepted Nov 18, 2022

### Keywords:

Biometric face detection

Compute unified device architecture

Face detection

Multi-core system

Parallel algorithm

## ABSTRACT

The expansion of biometric applications and databases is worrying. Processing extensive or sophisticated biometric data results in longer wait times, which might restrict application usefulness. This work focuses on accelerating the processing of biometric data and proposes a parallel method of data processing that exceeds the capabilities of a central processing unit (CPU). The combination of the graphics processing unit (GPU) and compute unified device architecture (CUDA) results in at least three times the processing speed of a published accurate and secure multimodal biometric system. The GPU-assisted approach beats the CPU-only implementation when saturating the CPU-only performance with more people than the available thread count. The GPU-assisted solution is also proven to have the same accuracy as the original system, indicating accuracy and processing performance improvements in the demanding big data environment.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Mohammed Wajid Al-Neama

Education College for Girls, University of Mosul

Mosul, Iraq

Email: mweama@uomosul.edu.iq

## 1. INTRODUCTION

In any country whose population is growing exponentially, the size of government databases of common biometrics such as face and fingerprints have the same metrics [1]. This means that achieving reasonable response time from biometric access control systems nationwide is becoming increasingly challenging from a data processing perspective when using only (CPU) solutions. Moreover, the popularity of access control applications that rely on big data has extended beyond government applications. In fact, more than one study recently reported citizens open a bank account by simply taking a picture using their mobile application, which verifies the user using a combination of facial biometrics and personal information [2]-[4].

For experimental purposes, it is assumed that real-time performance refers to a response time of below three seconds. The best effort in obtaining real-time performance requires utilizing a given hardware architecture as efficiently as possible. The graphical processing unit (GPU) differs from many consumer CPUs in that it prefers to process a large number of instructions simultaneously over a single education's performance. However, this requires careful memory management and algorithm planning.

Furthermore, contemporary GPUs include hundreds of cores and can operate thousands of threads continuously, far more than top-tier consumer CPUs, which contain between four and twelve cores. The Nvidia GeForce GTX 1050 Ti is an example of a low-end GPU with 768 cores and the same computing capacity as the 3584-core 1080 Ti. GPU-assisted methods may handle computational challenges in a rising variety of large data biometric applications, even though CPU and GPU core performance cannot be directly compared. Due to its efficient utilization of low-level memory operations and instruction sets, the compute unified device

architecture (CUDA), native to Nvidia GPUs, has significantly boosted the processing performance of diverse computer vision systems. However, GPU implementation research is often restricted to single biometrics or multimodal biometric systems that combine modalities with minimal acquisition synergy.

Brown and Bradshaw [5], an accurate face and iris multimodal biometric system solved several significant information security challenges in mobile and other digital platforms. Because these two biometrics may be gathered concurrently at a distance using a near-infrared sensor [6], the user experience is comparable to that of using a single biometric but enhanced security and accuracy. This acquisition approach may minimize the time required to transport network data and memory, a significant bottleneck in GPU-assisted biometric systems and other systems using GPU processing.

This research examines the processing performance aspect of the face and iris biometric modalities and their fusion by reimplementing in [5] on the CUDA platform. More modalities, such as the hand and fingerprint, may be handled in this manner. However, because of space constraints, they are not discussed in this study. The reimplementation includes the optimization of preprocessing and feature detection as well as the classification process, such that a maximum number of concurrent individuals are identified per given duration of time. The computational performance of the CPU-only and GPU-assisted implementations of the proposed multimodal biometric system will be compared. The goal is to improve the speed at which biometric data is processed. Still, the accuracy of the GPU-assisted implementation will also be checked to make sure that it is fair to compare it to the original CPU version.

The rest of the paper is organized as follows: Section 2 presents related faces, iris, fused systems, and biometric systems that operate on the CUDA framework. Section 3 explains how the CUDA framework is utilized to improve processing speed. Section 4 discusses the construction of the system. The experimental analyses and results are discussed in section 5. Section 6 concludes the paper and outlines future work.

## 2. RELATED RESEARCH

Many previous studies dealt with the subject of the iris and the face and their common characteristics. These researchers have investigated which information may be particularly important for the processing of face identity. This section review results from recent studies.

### 2.1. Characteristics of face and iris

Face segmentation eliminates dynamic elements such as an individual's hair and backdrop. The difficulty of consistently deleting these emotional traits during segmentation has been extensively studied. Karras *et al.* [7], the author suggested a new way to divide a face using gradient boosting to learn an ensemble of regression trees. The system is accurate and significantly exceeds real-time performance at less than 1 millisecond per image on a single CPU. The facial landmarks are automatically generated from a sparse subset of image intensities. In addition, the system manages incomplete or partially labeled data. The resultant facial landmark coordinates are utilized to segment the face.

A new method by Hezil and Boukrouche [8] successfully executes facial recognition with a single training sample without 3D modeling or deep learning, which may exacerbate the massive data processing challenge. Their method tries to frontal all segmented faces using a basic mesh for each person without face detection preprocessing. The problem with their approach is that it takes more time because it uses 40 Gabor filters with 8 different orientations and 5 different scales.

On the other hand, the iris is known as the most accurate external biometric, but this can only be done using sensors close to the eye [9]. Improving iris identification while employing sensors at a distance is a difficulty. There are two primary ways of segmenting the iris: the integral operator and the hough transform (HT) [10]. Calculating the maximum in the blurred derivative concerning the rising radius of normalized contour integrals along circular trajectories, the Integrodifferential operator performs an exhaustive search for the iris's center and radius. In contrast, HT utilizes binary edge maps to identify pupil and iris borders. To estimate the dimensions of the concentric border circles, votes are tallied.

Chaki *et al.* [11] used an HT algorithm appropriate for iris extraction. Restricted circular hough transformation (RCHT) is a method that looks for circles bordered by the upper and lower eyelids. The extracted texture inside two concentric rings constitutes the iris's feature vectors. The more significant vector, composed of the feature vectors for each eye, is classified using support vector machines (SVMs).

### 2.2. Iris and the face merge

Ammour *et al.* [12] devised a thorough face and iris fusion approach at the feature level. All throughout the preprocessing stage, they stress the significance of feature alignment. Extensive experiments show that alignment significantly improves recognition accuracy when using biometric data. The two eye

sockets serve as a dividing line to split the face in half and provide a vertical axis. Iris alignment is done by cutting the iris and rotating each half to the right place for the face's posture.

The accuracy was verified using the CASIA-Iris-Distance dataset, which comprises close-up near-infrared face images. It stores near-infrared images of the face, which need a memory transfer to the GPU device to decompose into face and iris representations. The analysis focused on feature alignment and limiting false matches, as determined by a false acceptance rate (FAR) of 0.01%. The alignment of both irises improved verification accuracy by 44%. Aligning the faces enhanced verification accuracy by 7%. Furthermore, combining the face and both irises increased accuracy by 14%. The actual acceptance rate (GAR) for the face was 87.33%, 71.55% for fused irises, and 94.44% for all three fused irises. For more details see [13], [14].

**2.3. Instances of iris using GPU**

Using the CASIA iris dataset, Jamaludin *et al.* [15] have shown the effectiveness of GPU parallel processing. The noise was reduced in the iris segmentation by omitting the whole iris and confining the upper and lower eyelid areas. This yielded a GAR of around 96%. The iris recognition technique takes 719 ms to complete, which is 200ms faster than the existing gold standard. All image processing methods were executed on the GPU, which was twice as fast as the CPU.

**3. GPU-AIDED COMPUTING UTILIZING CUDA**

CUDA offers rapid parallel programming and enables general-purpose GPU to compute (GPGPU). Using a cluster of multiprocessors with hardware-level task scheduling, as in the Single Instruction Multiple Threads architecture, makes this feasible. This is ideal for running the same program instruction on a large population's biometric data all at once. The shared device memory is how the multiprocessors talk to each other. All processor cores can use it, but it depends on the central processing unit and main memory to work [16], [17].

The CUDA framework's software component advances the C/C++ programming language [18], [19]. Planning a GPU-assisted biometric system from the perspective of a CUDA programmer needs the utilization of both host code and device code running on the CPU and GPU, respectively. This research generates host code for algorithms like Gaussian blur that exhibit little data parallelism or run more slowly on the device due to low complexity. On the other hand, algorithms with a lot of data parallelism are loaded into the device's memory and performed in device code.

The left side (host) of Figure 1 illustrates the host's organization of device code into kernels for low-level hardware communication [20]. By making a kernel call on the host, we may have a piece of code running on the device that can be called again, treated as a standalone function, and used with new data each time. The ultimate result is a novel biometric system that runs in numerous concurrent threads in the device's memory, despite its sequential origins. A programmer must still explicitly declare these memory allocations. However, the computer vision algorithms are often configured to automatically specify memory allocations, except in the classification phase and during the processing of many users simultaneously.

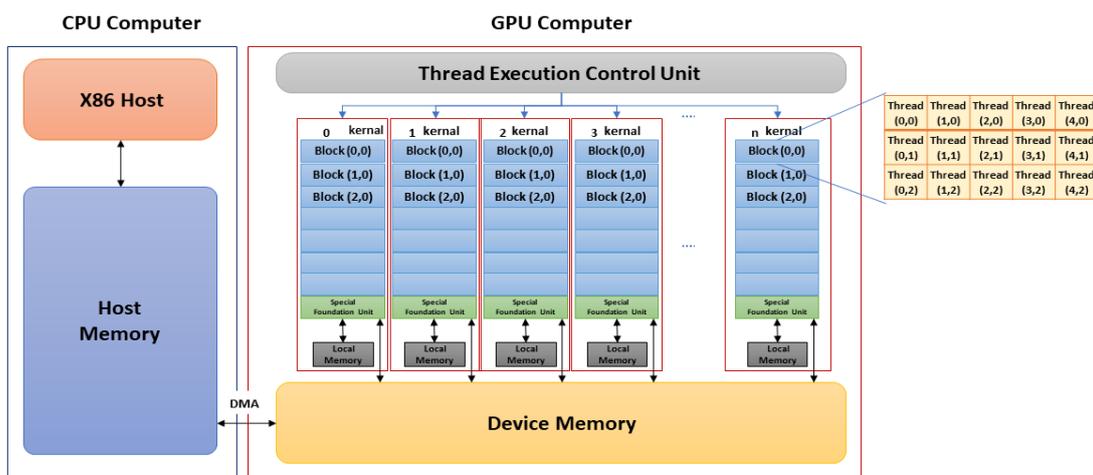


Figure 1. Shows the CUDA framework

Hardware threads are stored in the multiprocessors and run in a coordinated fashion as a set of threads inside a warp. Warps are 32-threaded hardware-managed supercomputers. However, different threads in device

memory may be necessary for machine vision and other GPGPU applications. This is particularly true when doing a parallel analysis of biometric data from different people. Each block, as shown at the bottom of Figure 1, groups threads and may accommodate up to 1,024 threads. A hardware scheduler coordinates the coordinated operation of a group of warps inside a single block. This could be called a zero-latency scheduler because it can hide memory bottlenecks in the pipeline and communication delays between devices by using warp switches in the right places [21], [22].

The memory transfer overhead is introduced when the computed result is returned to the host's memory. This becomes obvious inside code loops where the transfer is happening and is thus not a "recommended practice" [22]. The relative processing performance of the CPU-only and GPU-assisted solutions is predicted to significantly affect the number of selected people. In this research, we are trying to find ways to make it easier for a GPU-assisted implementation to transfer memory:

- Since both the face and the iris can be easily acquired as a single data image and put into device memory for training and testing (as explained in the following two bullet points), we may use their complementary nature.
- Before using computing with the help of a GPU, a certain number of people are put into the device's memory so they can be used to train the computer to put data into categories.
- Before data classification testing, the data model is loaded into the device's memory. GPU-assisted computing is then used to make predictions about the class of a sample set of people.
- The gadget does as much computing as possible before sending the results back to the host.
- Since the preparation procedures are the first to be performed, the host machine may be used to carry them out.

#### 4. THE SUGGESTIVE SYSTEM

This testing was performed on a computer equipped with an Intel 6400 quad-core CPU and an Nvidia GTX 1050 Ti GPU. The C/C++ code for the CPU-only version uses OpenCV and Dlib for image processing and Liblinear for machine learning. The CUDA libraries are used in the GPU-accelerated version. The C++ Chrono high-resolution clock technique measured CPU and GPU timings. This approach demonstrates a multimodal biometric system that is fast and accurate. Figure 2 shows the outlines of the method by dividing the face from the iris to show how the two may be processed in parallel, this figure is used a lot to offer the different stages of the system.

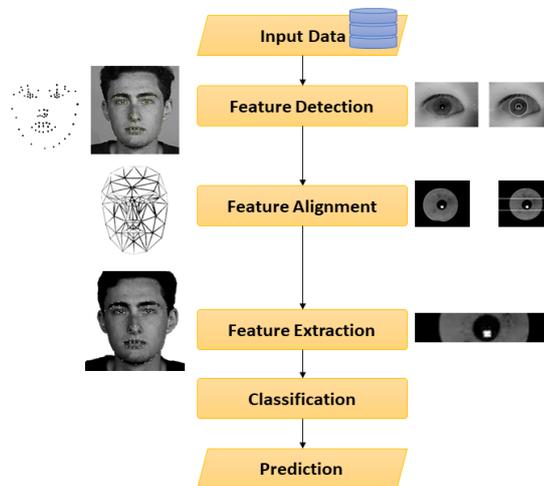


Figure 2. Shows the proposed face and iris recognition solution

##### 4.1. Steps before detecting features

As part of the preprocessing phase, a Gaussian blur with a  $5 \times 5$  kernel is performed on the whole image, including the face and iris. Given that the face and iris algorithms are separate, they may be run in parallel, as shown in Figure 2. During preprocessing, a single image undergoes several other elementary processes, such as scaling and pixel normalization, which run more quickly on the CPU than on the GPU. The initial objective is to identify the eyeballs from the facial areas sent to the GPU. Thus, the face and iris may be processed in parallel after loading only once. Haar object detection [23] is used to isolate the eyes from the rest of the face, and it runs twice as quickly on the GPU as it does on the CPU. Additionally, the transmission time between the host and device memory requires around  $60\mu\text{s}$  per image.

Detecting features of face and iris: The first step is to settle on a target area region of interest (RoI). Facial landmark detection utilizes a linear support vector machine and a sliding window classifier based on the histogram of Gaussian (HoG) characteristics. face detection is done by going back and forth along the x-axis by 5 degrees in both directions until a match is found.

Figure 2 shows the results of the model-based detection of 68 interpolated landmarks inside the original face RoI. The landmark model employed in this work was trained on the iBUG 300-W face annotated dataset in the form of a cascade of regression trees [24]. The time complexity is decreased for training and testing the landmark detector by computing the transform and warping it just once at each level of the cascade. Furthermore, the exercise is only completed once, enabling the landmark detector to anticipate unknown faces included in distinct datasets. A graphics processing unit to help identify HoG characteristics sped up the process by order of magnitude. Because the complete face detector runs in less than a millisecond per person, the memory copy to and from the GPU device decreases this speedup from 800% to 50%.

When capturing the RoI around the pupil in a visual image, the RCHT technique is applied to the input data [9]. A Laplacian of Gaussian (LoG) filter with 80 and 1900 kernels improves the RCHT method's consistency. The iris border is therefore identified by comparing the sclera with the iris area below or horizontal to the pupil, depending on the boundary threshold. This barrier is designed for those who stare over to the side or have to squint. The concentric circles at the edges represent the solution. The GPU-assisted version of RCHT resulted in a spectacular 50-fold performance gain.

#### 4.2. Matching features

We employ the confidence score generated by the fast local binary pattern histogram (LBPH) approach for failsafe feature alignment in conjunction with a sliding window. High-security settings allow for a more stringent rejection of low-quality data by adjusting this threshold. Before resorting to the fallback strategy, the following feature alignment techniques are used on the face and the iris, respectively. However, because of the already fast computation time on the CPU, none of these techniques are run on the GPU.

Following the identification of critical anatomical landmarks, a facial mesh is generated. The face mesh is built using delaunay triangulation so that no landmark is within the circumcircle of any triangle [25]. The leftward-posing mesh is brought into a frontal orientation, as seen in Figure 2. faces at angles of up to 30 degrees may be quickly formalized using this approach.

An inverted Gaussian filter with an  $11 \times 11$  kernel is used to enhance the sharpness of the iris texture included inside the concentric circles that make up the RoI. To again accommodate squinting and off-center eye gazing, the top and bottom iris areas are trimmed near the margin of the pupil. However, before the data is pruned, the iris texture is aligned by rotating the image of the query image according to the database image. This allows for the alignment of textures that are not obscured.

#### 4.3. Extracting features

The preceding procedures eliminated intra-class variance, which is necessary for adequate classification. However, it is also essential to maximize inter-class variance because of the role it plays in minimizing false matches. In the GPU-assisted version of the system, the feature extraction process is not run on the GPU.

The studies [25]-[27] have been shown to work well together to maximize inter-class variance for image-based biometrics. The LoG filter eliminates undesirable characteristics in the low and high-frequency spectrum before boosting the remaining features, thus improving the mean signal. The Gaussian and Laplacian kernels were  $15 \times 15$  and  $7 \times 7$ , respectively. This was applied to all of the normalized and segmented data. Using the local line binary pattern (LLBP) operator, we can attenuate brightness variations without increasing background noise. The final images are downsized to  $75 \times 75$  pixels so they may be used in categorization.

#### 4.4. Classification

The eigenfaces approach [28] is used by the  $k - NN$  classifier, and it is the most effective method for image-based biometrics. This classifier uses a linear combination of characteristics to optimize the overall variance in data. The first five main components used to create a model include the majority of the variation in the data. This is a common technique that effectively decreases background noise in data. During the prediction phase on the GPU, many threads run at the same time to handle a large number of samples. The total scatter matrix for a set of  $N$  sample images of type  $a_s$  is defined as:

$$X_t = \sum_{k=1}^N (a_s - m)(a_s - m)^t$$

where  $m \in R^N$  is the mean image obtained from the samples.

## 5. EVALUATION AND RESULTS OF AN EXPERIMENT

Below, especially in section 2, we analyze and compare the results of two experiments designed to evaluate an at-a-distance dataset's accuracy and processing speed. Related state-of-the-art research is filled with challenging datasets, as was described. When assessing a case, GAR is always used. It has been established that FAR equals 100% of GAR for verification purposes.

### 5.1. Datasets from experiments

We utilized the following dataset for empirical testing: The experiments presented in the previous section employ the tough CASIA-Iris-Distance dataset. Information was collected from a distance of three meters using a sensor with a resolution of  $2.352 \times 1.728$  designed and built by the authors. Because the iris worked well on this data set, it shows that it can be a powerful biometric that is easy for the user to use.

These categories comprise the CASIA-Iris-Distance dataset: 90 people are chosen compared to Ammour *et al.* [12]. This information will be used as a trial run for near-infrared camera sensors built into smartphones and other digital devices that can outsource processing to a central server. In a group of 90 people, five samples are utilized in instruction and five in evaluation.

### 5.2. The experiment in accuracy

Verification ensures authenticity by utilizing k-nearest neighbor (k-NN) to determine whether a training sample from a known class and a test sample has an eigenvalue distance below a threshold. Figure 3 shows how the suggested face and iris systems fared compared to Ammour *et al.* [12] using the CASIA-Iris-Distance dataset for verification purposes. The verification is expected to be accurate to 0.01% FAR. The suggested approach improved the results for the face, the two irises, and the combination of the three. The improvement in the irises is significantly more pronounced than that on the face. The iris algorithm's design with long-range functionality is responsible for this.

### 5.3. Test of computer efficiency

The second verification round is performed, but this time on the GPU-assisted implementation. The results were presented in Figure 3, meaning that the system successfully moved to the CUDA environment. Figure 4 compares the processing speeds of the CPU-only and GPU-assisted variants of the same method as well as the speed-up. The speedup is around 50 times larger with the HoG filter face detection approach. This is followed by the RCHT, a Hough-based algorithm, and Haar. The 588ms execution time was faster than the GPU iris system that Jamaludin *et al.* [15] came up with.

Due to time restrictions, the training portion of classification was not moved to the GPU. Prediction speed for a single person is comparable to the CPU and GPU versions. Figure 5 shows that the transmission time between the device and host is about 150 milliseconds, multiplied by the number of subjects tested (1, 4, 6, 10, or 18). An exponential trend is noticed for the prediction phase after depleting the (four) thread count on the CPU. This demonstrates that the processing speed of biometric data may be determined, in part, by the number of physical cores available in a computer rather than by software-level threads in a parallel processing architecture. The GPU-assisted version achieves real-time performance while concurrently categorizing (including all accumulated timings) up to 18 people. This is a promising result, particularly given the usage of a low-end GPU.

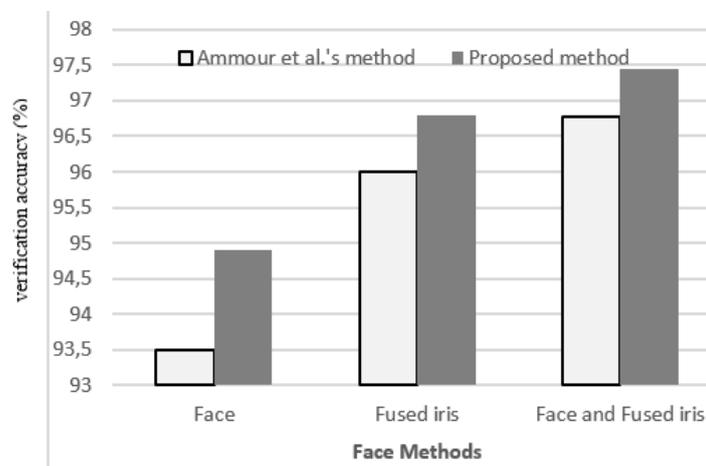


Figure 3. Shows the results of validation tests conducted on the CASIA-iris-distance dataset

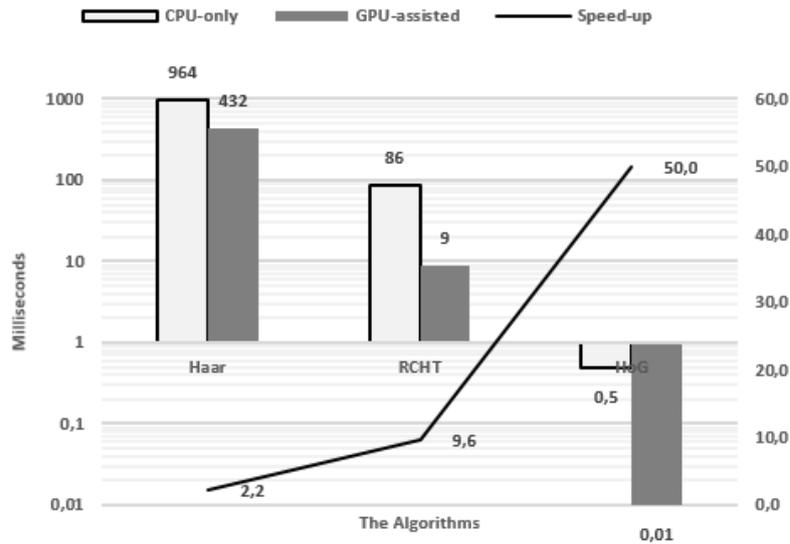


Figure 4. Shows the comparison of the processing speeds of algorithms that just use the CPU and those that also use the GPU (in milliseconds) with the speed-up

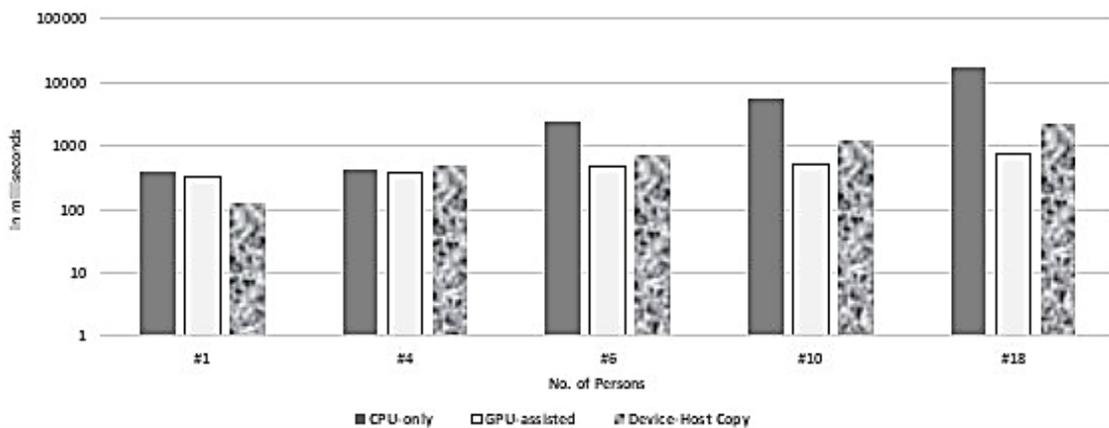


Figure 5. Shows the comparison of the speed of classifier prediction when using only the CPU or GPU-assisted versions (in milliseconds)

## 6. CONCLUSIONS AND FUTURE WORK

Using many biometric modalities increases security but may slow down processing time for large data sets. This article demonstrates a dramatic increase in the identification accuracy of both authentic and imposter users, making it suitable for usage in high-security settings. Studies have proven that remote iris recognition is more accurate than the more conventional facial identification method. The suggested method significantly improves recognition accuracy compared current gold standard iris verification system. Combining facial and iris features led to near-perfect accuracy and additional protection against counterfeiting. Facial and iridescent traits may be quickly acquired and supplemented in biometric systems. They also make it easier to send data, which is essential for getting information to the GPU as quickly as possible.

To take advantage of the GPU's superior single-instruction speed, a GPU-accelerated variant had to be developed by reimplementing the original system to avoid performing basic operations. Overall performance is improved when it is realized that not all algorithms need to be run on the GPU. In addition, the HoG algorithm with GPU support improved processing performance by 50 times, the most of any single method. The GPU's overall performance went up by a factor of three when only one subject was being looked at. But the benefit multiplies as more and more people are evaluated at once.

The GPU's high thread count enabled a 24-fold boost while assessing 18 people. Last but not least, it was determined that the GPU-assisted version was just as accurate as the original system, which thus illustrates the effective reimplementing of a reliable biometric system on a GPU. We want to research more

improvements for the prediction phase and migrate the training phase over to the CUDA framework shortly. Other classifiers will be added and adapted to the CUDA framework as well. Finally, an iris recognition technology that is more effective in parallel, such as a HoG filter with regression trees, may be utilized in place of Haar cascades.

## ACKNOWLEDGMENTS

The authors appreciate Mosul University for allowing them to use its computing resources.

## REFERENCES

- [1] N. Alay and H. H. Al-Baity, "Deep learning approach for multimodal biometric recognition system based on fusion of iris, face, and finger vein traits," *Sensors*, vol. 20, no. 19, p. 5523, 2020, doi: 10.3390/s20195523.
- [2] J. Coetzee, "Strategic implications of Fintech on South African retail banks," *South African Journal of Economic and Management Sciences*, vol. 21, no. 1, pp. 1–11, 2018.
- [3] H. Baber, "Impact of FinTech on customer retention in Islamic banks of Malaysia," *International Journal of Business and Systems Research*, vol. 14, no. 2, pp. 217–227, 2020, doi: 10.1504/IJBSR.2020.106279.
- [4] C. Kohardinata, N. Soewarno, and B. Tjahjadi, "Indonesian peer to peer lending (P2P) at entrant's disruptive trajectory," *Verslas: Teorija ir praktika/Business: Theory and Practice*, vol. 21, no. 1, pp. 104–114, 2020.
- [5] D. Brown and K. Bradshaw, "Efficient biometric access control for larger scale populations," doi: 10.3846/btp.2020.11171.
- [6] J. Gong, G. Zhang, and K. Wang, "Human iris localization combined with ant colony and improved hough circle detection," *Journal of Mathematics and Informatics*, vol. 16, pp. 23–39, 2019, doi: 10.22457/jmi.138av16a3.
- [7] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.
- [8] N. Hezil and A. Boukrouche, "Multimodal biometric recognition using human ear and palmprint," *IET Biom*, vol. 6, no. 5, pp. 351–359, 2017, doi: 10.1049/iet-bmt.2016.0072.
- [9] S. A. R. Shah and B. Issac, "Performance comparison of intrusion detection systems and application of machine learning to snort system," *Future Generation Computer Systems*, vol. 80, pp. 157–170, 2018, doi: 10.1016/j.future.2017.10.016.
- [10] M. Arsalan, R. A. Naqvi, D. S. Kim, P. H. Nguyen, M. Owais, and K. R. Park, "IrisDenseNet: Robust iris segmentation using densely connected fully convolutional networks in the images by visible light and near-infrared light camera sensors," *Sensors*, vol. 18, no. 5, p. 1501, 2018, doi: 10.3390/s18051501.
- [11] J. Chaki, N. Dey, F. Shi, and R. S. Sherratt, "Pattern mining approaches used in sensor-based biometric recognition: a review," *IEEE Sens J*, vol. 19, no. 10, pp. 3569–3580, 2019, doi: 10.1109/JSEN.2019.2894972.
- [12] B. Ammour, L. Boubchir, T. Bouden, and M. Ramdani, "Face-iris multimodal biometric identification system," *Electronics (Basel)*, vol. 9, no. 1, p. 85, 2020, doi: 10.3390/electronics9010085.
- [13] C. Kamlaskar and A. Abhyankar, "Multilinear principal component analysis for iris biometric system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 3, pp. 1458–1469, 2021, doi: 10.11591/ijeecs.v23.i3.pp1458-1469.
- [14] A. S. Akinfende, A. L. Imoize, and O. S. Ajose, "Investigation of iris segmentation techniques using active contours for non-cooperative iris recognition," *Indonesian J. of Electrical Engineering and Computer Science*, vol. 19, no. 3, pp. 1275–1286, 2020, doi: 10.11591/ijeecs.v19.i3.pp1275-1286.
- [15] S. Jamaludin, N. Zainal, and W. Zaki, "Sub-iris technique for non-ideal iris recognition," *Arab J Sci Eng*, vol. 43, no. 12, pp. 7219–7228, 2018, doi: 10.1007/s13369-017-3051-8.
- [16] J. Fang, C. Huang, T. Tang, and Z. Wang, "Parallel programming models for heterogeneous many-cores: a comprehensive survey," *CCF Transactions on High Performance Computing*, vol. 2, no. 4, pp. 382–400, 2020, doi: 10.1007/s42514-020-00039-4.
- [17] S. Markidis, S. W. der Chien, E. Laure, I. B. Peng, and J. S. Vetter, "Nvidia tensor core programmability, performance & precision," in *2018 IEEE international parallel and distributed processing symposium workshops (IPDPSW)*, 2018, pp. 522–531, doi: 10.1109/IPDPSW.2018.00091.
- [18] D. Chen, Y. Hu, C. Cai, K. Zeng, and X. Li, "Brain big data processing with massively parallel computing technology: challenges and opportunities," *Softw Pract Exp*, vol. 47, no. 3, pp. 405–420, 2017, doi: 10.1002/spe.2418.
- [19] A. Abdelhafez, G. Luque, and E. Alba, "Parallel execution combinatorics with metaheuristics: Comparative study," *Swarm Evol Comput*, vol. 55, p. 100692, 2020, doi: 10.1016/j.swevo.2020.100692.
- [20] L. Qu, H. Ltaief, and D. Keyes, "Toward high performance asynchronous RTM with temporal blocking and buffered I/O," in *Fifth EAGE Workshop on High Performance Computing for Upstream*, vol. 2021, no. 1, pp. 1–5, 2021, doi: 10.3997/2214-4609.2021612022.
- [21] D. B. Kirk and W. H. Wen-Mei, *Programming massively parallel processors: a hands-on approach*. Morgan kaufmann, 2016.
- [22] J. Luo, S. Fujimura, D. el Baz, and B. Plazolles, "GPU based parallel genetic algorithm for solving an energy efficient dynamic flexible flow shop scheduling problem," *J Parallel Distrib Comput*, vol. 133, pp. 244–257, 2019, doi: 10.1016/j.jpdc.2018.07.022.
- [23] D. G. Ganakwar and V. K. Kadam, "Face detection using boosted cascade of simple feature," in *2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC)*, 2019, pp. 1–5, doi: 10.1109/ICRAECC43874.2019.8994977.
- [24] F. Shi *et al.*, "Review of artificial intelligence techniques in imaging data acquisition, segmentation, and diagnosis for COVID-19," *IEEE Rev Biomed Eng*, vol. 14, pp. 4–15, 2020, doi: 10.1109/RBME.2020.2987975.
- [25] D. Brown, "Deep face-iris recognition using robust image segmentation and hyperparameter tuning," in *Computer Networks and Inventive Communication Technologies*, Springer, 2022, pp. 259–275, doi: 10.1007/978-981-16-3728-5\_19.
- [26] D. Brown and K. Bradshaw, "Extended feature-fusion guidelines to improve image-based multi-modal biometrics," in *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists*, 2016, pp. 1–9.
- [27] M. Taskiran, N. Kahraman, and C. E. Erdem, "Face recognition: Past, present and future (a review)," *Digit Signal Process*, vol. 106, p. 102809, 2020, doi: 10.1016/j.dsp.2020.102809.
- [28] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

**BIOGRAPHY OF AUTHORS**

**Abeer A. Mohamad Alshiha**    received the BSc in Computing Science and the MSc degrees (first Class Hons.) in Computing Science/Remote Sensing from the Mosul University, Iraq, in 1995 and 2002, respectively. Her Ph.D. Degree in Artificial intelligence from the School of Electrical and Electronics Engineering, Newcastle University, Newcastle Upon Tyne, UK, in 2013. She works as a lecturer & researcher in the Remote Sensing Center, Mosul University, Iraq since 1996 yet. And also, works as a lecturer in different colleges such as the College of Computing and Mathematical Science and the College of Petroleum and Mining, Mosul University, Iraq. Her current research interests include biometrics, artificial Intelligence, tensor object processing, and statistical pattern recognition, with applications in human recognition, tracking, and detection. She can be contacted at email: [abeer.allaf@uomosul.edu.iq](mailto:abeer.allaf@uomosul.edu.iq).



**Mohammed Wajid Al-Neama**    was born in Baghdad, Iraq. He got his bachelor from University of Mosul, Iraq in 1995. He got his M.Sc. from University of Mosul in 2004 and got his Ph.D in High performance computing (Parallel tools and applications) in 2014, Al-Azhar University-Egypt. He is currently a Assistant Professor at the Education College for Girls, University of Mosul, Mosul, Iraq. His main research area is high performance computing, bioinformatics, parallel computing. His research interests are in exact string-matching algorithms, parallel and distributed processing. He can be contacted at email: [mwneama@uomosul.edu.iq](mailto:mwneama@uomosul.edu.iq).



**Abdalrahman R. Qubaa**    was born in Mosul, Iraq. He got his bachelor from University of Mosul, Iraq in 1995. He got his M.Sc. from University of Mosul in 2004 and got his Ph.D in Information Technology (Image Processing) in 2013, from Electrical, Electronic and Computer Engineering Department, Newcastle University, UK. He is currently a Head of UAVs Unite in Remote Sensing Center, University of Mosul, Mosul, Iraq. His main research area is Image processing and Remote Sensing Applications. He can be contacted at email: [abdqubaa@uomosul.edu.iq](mailto:abdqubaa@uomosul.edu.iq).