*Nanotechnology Perceptions*
*ISSN 1660-6795*
*www.nano-ntp.com*

# A Comprehensive Analysis of Algorithms in Multiple-Face Recognition

**Vijaykumar P. Mantri[1,2], Yogesh Deshpande[1,3], SandipThite[1]**

*[1]Vishwakarma University, Pune INDIA,*
*[2]MIT Academy of Engineering, Pune INDIA,*
*[3]Vishwakarma Institute of Information Technology, Pune, INDIA*
*Email: vijay.mantri2000@gmail.com*

When it comes to biometric applications with image analysis and computer vision, face recognition ranks high and is a very important process. In private and public sectors there are various uses of face recognition applications for person authentication, surveillance systems, object detection in autonomous vehicles, etc. Detecting multiple faces in real time on high-resolution CCTVs requires a lot of processing power. If the process is carried out in sequence, the desired real-time performance cannot be attained. This survey paper reviews research work on using various frameworks and methods to improve multi-face detection results using parallel computing.

**Keywords:** Face Recognition, Multiple Face Detection, Deep Learning, Parallel processing, Inter-layer and Intra-layer Parallelism, High-Performance Computing.

## 1. Introduction

Image processing,especially face recognition,has been a very important research field and developing area which has several applications in various fields like biometric authentication, surveillance systems, autonomous vehicles, smartphone cameras, etc. Traditional face recognition applications aim to develop image processing algorithms which are faster and more robust.The accurate face recognition requires images containing faces with high resolution. Today, face recognition algorithms have reached a high level of accuracy, but under particular conditions, these algorithms are still affected by numerous external and internal parameters.Face recognition's major challenge is developing effective feature representation to improve accuracy in different scenarios.Deep learning algorithms implemented on GPU or multicore CPU for faster performance have recently succeeded.For extreme-scale complex computing, these algorithms generally use data parallelism or model parallelism.

A possible strategy for making significant performance enhancements for multiple face recognition is provided by the convergence of high-performance computing (HPC) and data-intensive approaches. This study provides a general description of the relationship between traditional high-performance computing(HPC) and Machine Learning (ML) approaches. The "effective performance" can be accomplished by integrating learning methodologies with simulation-based approaches differentiating it from traditional performance enhancement methods. The researchers propose modelling the diverse ways of concurrency in Deep Neural Networks (DNNs) to support the promise of integrating HPC and learning methods [1][2][3][4][5][6][7][8]. Face recognition is a method for identifying faces with any position or orientation from a photo with various environmental conditions of the image [1].Many studies have been done on face recognition algorithms, and they have significantly improved in terms of both their performance and speed. The development of microprocessors has led to a meteoric rise in the speed at which computations may be performed. However, this growth is not even close to being able to keep up with the rapid improvement in image and video technologies. These include the increasing resolution of cameras, and increasing processing demands of many new real-time applications. The strategy that system designers employed to put concurrent processing notions into practise was the genesis of parallelism.

The importance of Deep Neural Networks (DNNs) hasbeen increased in various fields like big data analysis, autonomous cars, image and object recognition, etc. Deep learning training for the multiple face recognition is computationally very greedy about the resources with respect to the processing cycles and the space complexity too.

This paper aims to study various face detection and face recognition algorithms proposed by various researchers, analyse these models and provide the summary of different multi-face recognition techniques. It also provides in depth analysis of these methods and important technical offering for multi-face recognition.

## 2. Face Recognition Approaches

This section provides a summarized overview of prominent face recognition methods, techniques and architectures developed and used by various researchers.

On the basis of Viola and Jones' framework, Hadi Santoso et al. [1] have described a parallel and multiple-face detection architecture and implementation approach. The researchers have suggested using a parallel technique for face detection. They have carried out several experiments to analyze the acceleration that was obtained. The strategy that was suggested has been demonstrated to be successful. The Ada Boost technique's classifier must calculate each of these sub-windows separately, so parallelizing the method by replicating the classifier offers a significant performance benefit. There could be tens of thousands of sub-windows in input photos.

The Viola and Jone’s framework for face detection has 4 steps like Pre-Processing, Determining Haar-like features, computing integral image and Detection with cascading classifiers as shown in Figure 1.

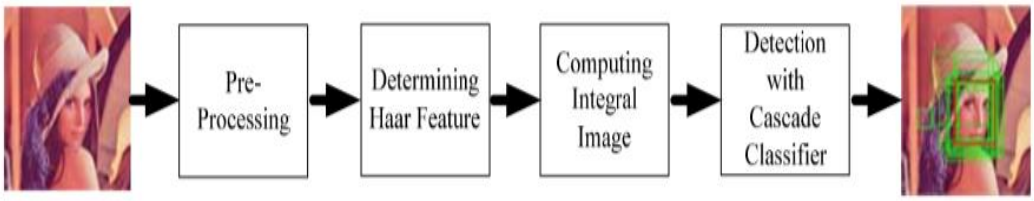


Figure 1:Viola-Jones method for Face detection

The authors proposed the Parallel Architecture to improve performance, efficient data movement and increased energy efficiency with multithreaded parallel programming.This is demonstrated by the experiment that the author carried out. Gains in performance of up to approximately twenty-one times and hundred times, respectively on systems with either two or four cores was observed.

Kanokmon Rujirakul et.al. [2] in their research paper has proposed multiple face recognition using Parallel Expectation-Maximization PCA Face Recognition Architecture (PEM-PCA) methodology. Authors have observed that the efficacy is directly proportional to the no. of cores along with the no. of threads which are running concurrently in computing device. Researchers focused on two parameters that is Eigenvector and epsilon value. As Eigenvector value goes on increasing so shall be the value of processing time, same is the scenario reflected for epsilon values. The researchers have pointed out multiple face recognition PCA mechanism and its drawback. They proposed a new method known as PEM-PCA (Parallel Expectation Maximization - Principal Component Analysis). The proposed method for face recognition consists of three phases namely face preprocessing, extraction of face features, and classification of face which can be done parallel.

When the recognition accuracy did not vary much, this face recognition method was proposed to decrease the time complexity needed for face identification. PCA has drawbacks so a face recognition system's recognition accuracy can be increased using a novel technique called classic PCA. The researchers developed a new technique of using PCA. During the research, a number of concerns were analysed and looked at, in regard to the amount of computing time complexity that occurred while calculating the covariance matrix. One of the possible techniques to enhance PCA is Expectation-maximization (EM) PCA face recognition. When the recognition accuracy did not change significantly, it was suggested to lower time complexity. In addition, as a result of the development of parallelism, new face recognition model was suggested by employing parallelism for the manipulation of large matrices.

This architecture included parallel preprocessing, recognition, and classification, all of which refer to Parallel Expectation-Maximization PCA (PEM-PCA) [2]model proposed by Kanokmon Rujirakul et.al. as shown in figure2.

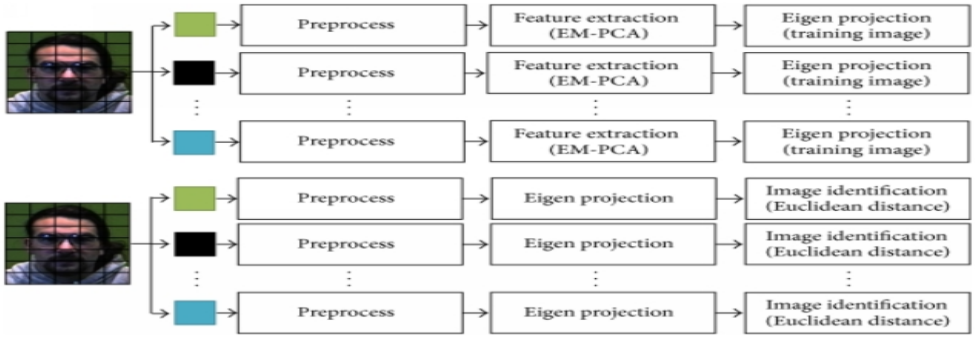


Figure 2: PEM-PCA Architecture

PEM-PCA has a justified parallel method implementation, and it performs better than the other PCA methods. It is important to note that the results are dependent on the total amount of training photos. But this variation does not significantly affect the precision of the recognition. Even though the suggested method can speed up the process significantly compared to PCA, future work can include more in-depth analyses and research. For instance, preprocessing stages can be improved; a high degree of parallelism can be enhanced in areas other than matrix manipulation and a large number of different images can be tested. In addition, independently separating recognition tasks can be done in order to demonstrate the effectiveness of parallelism usages. It is also possible to conduct additional research into the other components of the system, specifically face detection, in order to ensure that the process of face recognition is finished completely.

Hamed Fatemi et.al.[3] have introduced a new technique for multiple face recognition with high performance parallel computing platform using algorithmic skeleton approach for searching skin tone colour using YUV color domain. The authors proposed the detecting skin in the image using two steps. In the first step it finds the skin-tone in the given photo which gives a set of faces as output. In the second step it uses this output as input to theRadial Basis Function (RBF) neural network for identificationof number of person faces recognized and matches with database. The RBF neural network accepts n dimensional input which is further connected to m hidden layer nodes which is connected to output layer o nodes which is number of faces detected in the image. The proposed model is as shown in Figure 3.

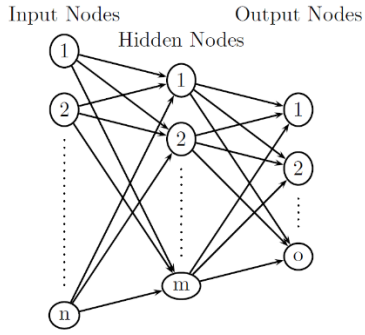


Figure 3:RBF NN Architecture

The experimental analysis involves low, intermediate and high level image operation in which the process of face detection and recognition is analysed with luminance. The true color method is used for face detection and the Radial Basis Function (RBF) neural network method is used for face recognition. In this method the image is converted to Gray Scale with skin region UV Spectrum ranging -127 to +127. The skeletonizing approach involves converting color, binarization, labeling and RBF neural network using IMAP Board assuming identical skeletons. Researchers have shielded parallel processing deployment of this algorithm which reflects the performance penalty below 10%.

Aashna R. Bhatia et.al.[5] have proposed parallel implementation of face detection algorithm on Graphics Processing Units (GPU)and discussed about Viola-Jones face identification algorithm. This algorithm, which was among the first of its kind, is examined in considerable depth. The primary topics of discussion in this article are advantages and disadvantages of the Viola Jones algorithm. Additionally, it explores the factors that make it the most well-known face identification algorithm and how it may be enhanced to better suit contemporary needs. In order to avoid the issues that arise while utilizing the serial implementation method, the Viola-Jones algorithm will be implemented on a parallel platform. The main objective is to accelerate the technique's computing speed by leveraging CUDA and Open CV on GPUs to implement the algorithm in parallel. Another objective is to compare the computational outcomes that can be obtained by the algorithm's serial and parallel implementations. The work focuses on implementing Viola Jones face detection algorithm using Open CV and GPU. This method helps in using the properties of the GPU and comparing the outputs of algorithm's serial and parallel processing. Due to initialization overheads, the first call to the function is very slow. The CUDA runtime API is called indirectly, and the major processing power is spent during the process of switching processes between main memory and GPU memory. It functions most well in environments with limited resources; the parallel implementation requires less time for processing. The detection rate is higher than the usual technique, and using GPUs is the more efficient way to process big amounts of data.

Dalia Shouman Ibrahim et.al.[6] have proposed two alternative parallel architectures to speed up the training and testing portions of the PCA method using distributed memory architecture. It is important to make face recognition decisions as quickly as possible in real time. The principle component analysis (PCA) technique, a feature extraction method, is extensively used in the field of facial recognition applications. These programs work by projecting images into a new face space. PCA has the ability to lessen the number of dimensions that an image has. However, due to the nature of its highly expensive calculations, PCA requires a major amount of processing power and time. The researchers suggested two distinct parallel designs to speed up training and testing. The summarized steps for these training and testing phases are as follows:

In the training Phase :

1:Converting image to a column of 2D matrix (A) representing images.

2: Finding average of all pixels in all images

3:Finding value of covariance matrix to extract eigenvalues (D) and eigenvectors (V)

4: Sorting the eigenvectors based on their eigenvalues.

5: Projecting the training images to allocated space by taking A's dot product with matrix V.

6: Storing the face space from the above result.

In recognition Phase :

1: Converting the given testing image into a ((MXN)X1) vector.

2:Subtract the calculated mean (from training phase) from above image vector.

3: Load the testing image in the allocated memoryface space.

4: Find the Euclidean distance betweentesting images and that of trainingimages.

5: The images with less than threshold are the images matched with testing image

Both architectures are built on Message Passing Interface (MPI), and they offered to handle a variety of various scenarios in facial recognition systems. When there is single input test image and multiple stored training, the most probable optimal method is to duplicate the test image and distribute multiple copies of the training set. If needed to process a video stream for recognition or big number of test faces, the most efficient way to handle these situations is to centralize the training set and distribute the test photos. These methods achieved super linear and linear speed-ups, bringing the execution time down by 25 percent in the training phase and 5 percent in the recognition phase. In the PCA technique, the experiments took into consideration all P-1 Eigenvectors. However, by using distributed memory architecture and a smaller number of Eigenvectors, it is anticipated that the estimated execution time of phases of the PCA algorithm will be reduced. The findings of these experiments indicate that the proposed architectures were capable of achieving in line speed-up and system scalability on various data sizes taken from the Facial Recognition Technology (FERET) database.

Fang Gao et.al. [7] have proposed low power parallel CPU-Accelerator heterogeneous multicore architecture for implementation of face detection system. A fundamental CPU version based on the cascade classifier and the local binary patterns operator for face identification prototype was developed and put into use. Then the prototype was expanded to include a particular embedded parallel computing platform. This platform is comprised of Xilinx Zynq and Adapteva Epiphany and is named as Parallella. Next the face identification algorithm was improved so that it could be adapted to the parallel architecture.

The steps to implement Parallella face detection model is as in Figure 4.

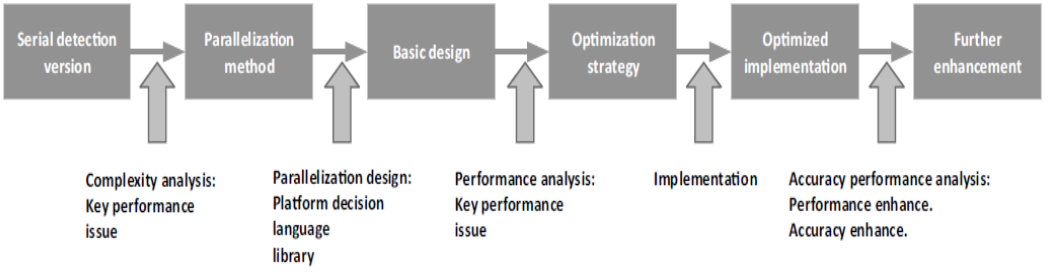


Figure 4:Parallella model implementation steps

The first step is serial face detection using AdaBoost LBP cascade classifier. Next step is assigning host CPU and multicore accelerator to implement hybrid-parallel method. The optimized implementation is done considering performance issue which can be further enhanced with accuracy enhancement. In Parallella platform, the dual-core ARM CPU integrated in Zynq and the OpenCV library is usedfor various image process functions.

This was done in order to increase the speed at which faces could be detected and to increase the amount of computing resources that were utilised. A face detection experiment was implemented to assess the performance of the computer and a solution was proposed in this research. The outcomes of the experiments show that the suggested execution reached a level of precision that was extremely consistent with that of the dual-core ARM, achieving a speedup that was 7.8 times greater. The findings of the experiments demonstrate that the proposed implementation offers significant performance benefits over alternative approaches.

Shivashankar J. Bhutekar et.al.[8] developed a GPU based system that processes parallel face detection and recognition in real time on NVIDIA GeForce GTX 770 GPU, which is much faster than on the CPU. They have developed and implemented the parallel algorithm in graphics processor with the help of CUDA. With CUDA programming model face detection and face recognition task can be done in parallel.

One of the algorithms recognizes a face and sends the information on to the face processing system. In traditional face recognition system using CPU, the face recognizer must wait for all faces in the frame have been identified, which takes a significant amount of time. The findings of the experiments make it abundantly evident that the GPU plays an essential part in the process of parallel computing. When using a CPU, face detection processing takes significantly longer time than a GPU. A parallel system with GPU can recognize any face instantly. This is the main difference between using a CPU to implement anything and using a GPU to perform same thing in real time. The experimental results for face detection and face recognition on GPU and CPU is as shown in table. The technique work in parallel on GPU so the required time to finish on GPU is much less than on CPU.

Table 1: Time taken on GPU and CPU

| Time (in ms) | Image Size | | | | | |
|---|---|---|---|---|---|---|
| | 700 X 580 | | 650 X 400 | | 480 X 360 | |
| | On GPU | On CPU | On GPU | On CPU | On GPU | On CPU |
| Face Detection | 350 | 1406 | 160 | 900 | 102 | 780 |
| Face Recognition | 312 | 300 | 127 | 279 | 90 | 207 |

The experimental results on the various size images are shown in table 1. Also, it is observed that the photo frame with a resolution of 1219 X 810 took 893.78 milliseconds and the second frame with 2048 X 1536took 1466.38 milliseconds. This does not include the time it took to load the image or write it to the hard disc. The GPU-based technology detects and recognizes faces in real-time more quickly than a CPU. The algorithms work concurrently, as the first one detecting faces and the second sending them to the face-processing system.

Mohd. Iqbal et.al. [11] proposed a DSP system that is used to carry out an image processing performance analysis utilizing a parallel and overlap segment technique. The analysis is carried out using overlapping segment data. The results that it delivers are far better when compared to those obtained using the conventional segment method. Researchers were able to conduct an analysis of the findings by making use of a variety of characteristics, such as the

MSE (mean square error), PSNR (peak to signal noise ratio), filter type, segment size, and overlap factor. The values that are assigned to these parameters impact on how long it takes for the image processing to be finished and the quality of the image that is produced. The idea of parallel processing is extremely important in this approach as it reduces the total amount of time necessary to finish the process of image processing. This approach proves to be the most successful for sectioning and grouping as it employs an overlap segment strategy. By utilizing the overlap segmentation technique, the complexity associated with filtering procedures can be completely removed. This approach takes advantage of parallel processing to cut down on the amount of time necessary for computation, improving the overall quality of the data that is generated. One of the advantages provided by using this parallel technique is the capability of applying a different algorithm to each individual image frame. The program can find parameters used in image processing that lead to the most successful results and yield the most useful outputs after completing a study on the segment size of the image. In this model, the Gaussian filter is evaluated in comparison to the median filter, the average filter, the motion filter, and the Gaussian filters. The overlap segment technique is a lot quicker than the conventional method of segmentation. Due to this, model becoming more and more popular. In addition, the processing is done using windows of variable sizes and segments of varying sizes for each iteration of the procedure.

Applications that rely on sequential algorithms are no longer able to boost their performance by relying on the scalability of the underlying technology, prompted by Sharanjit Singh et al. [12] who developed a new methodology. Applications that do image processing shows a high degree of parallelism, making them a suitable source for multi-core platforms to draw their processing power. The primary goal of parallel processing is to achieve better performance and faster solutions with more efficient use of available resources. This can be accomplished by making use of multiple processors in conjunction with one another. All are aware that the medical images must have the highest possible level of clarity and it must be recovered as quickly as is physically possible. This demands a higher amount of computing power than a typical sequential computer. Through a method known as parallelization, the researchers were able to successfully complete this task. The pace at which the image was created can be optimized through the utilization of parallelization, which speeds up the process. This research article examines the different kinds of parallelism that can be utilized in image processing, such as data parallelism, task parallelism, and pipeline parallelism. Each of these types of parallelism has its own advantages and disadvantages. In addition, this article discusses the point operators, neighbourhood operators, and global operators that are utilized throughout the many stages of the image processing process. The paper discusses about numerous methods that are used for parallel image processing, and applications on medical related images with the work flow engine Taverna being used for scientific processing. Also the paper covers discussion on the application of parallel image processing. There are few features that the parallel program must have to get high performance as expected. These features are Granularity (Coarse-grained and Fine grained), Synchronization, Latency, Scalability, Speedup &efficiency and Overheads.

The fundamental objective of this project is to investigate how many facets of digital image processing can benefit from the implementation of parallel computing. Parallel programming is discussed, along with the many other options that come with it, and its prerequisites for

greater performance are outlined by the researchers. Data parallelism, task parallelism, and pipeline parallelism are the three forms of parallel processing that were utilized in the image processing pipeline. Pipeline parallelism is the most common form of the three. In addition to this, the processes that may be utilized to carry out image processing activities in parallel have been given in detailed. An equalization of an image utilizing parallel computing have been discussed in detail in the research paper, along with the three different ways. An example of how parallel computing can be used in medical imaging is explored in great depth. Taverna is a workflow engine that is being described, and its design strategy is also being offered, in the context of the development of medical imaging.

Authors I P Skirnevskiy et.al. [13] in their article described the benefits of utilizing GPU in the large amounts of digital image processing. The research paper describes concise explanation of a parallel computing technique and its applications in a variety of fields.

For solving computational problem, Parallel computing uses simultaneously multiple computing resources. A task is divided into several independent subtasks and processed on parallel processors. The process is as shown in figure5 [13][22]. It creates a parallel CPU thread on GPU by copying data from RAM in video memory as GPU don't have direct access to memory. After completing required calculations, the data is moved back to RAM and cleared the video memory.

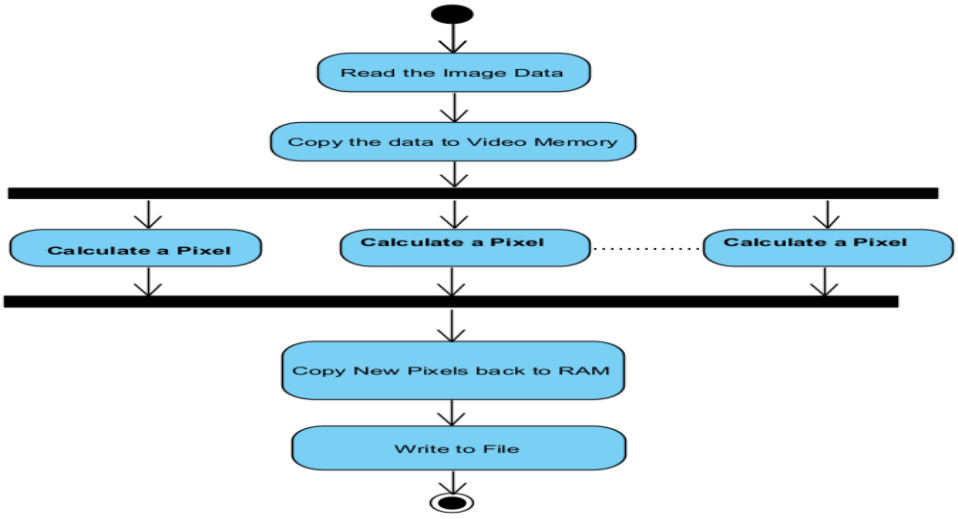


Figure 5:Activity diagram of GPU

The parallel computing on GPU can be done using either Nvidia CUDA technology or AMD's FireStream technology. It is observed by authors that CUDA technology is much faster so the choice of selecting this CUDA technology is suitable for such tasks.The paper also discusses about meaning of image noise and a quick summary of certain noise reduction algorithms. In addition to this, it outlines several fundamental prerequisites that a particular algorithm needs to fulfill for the elimination of noise in the projection to computer tomography. The research paper elaborated the comparison of the performance with the mixed percentages use of GPU with percentage CPU, the result is as shown in table 2 [13][22].

Table 2: Time of calculations

| | Time (In Sec) | |
|---|---|---|
| Proportion | 210 X 182 pixels | 160 X 128 pixels |
| Single CPU thread | 627.9577 | 184.0913 |
| Double CPU thread | 713.9502 | 193.8745 |
| Half CPU + Half GPU | 336.0062 | 98.8670 |
| 1/8 CPU + 7/8 GPU | 139.9920 | 39.2366 |
| Single GPU | 111.7244 | 29.8882 |

The implementation of a program can be greatly speed up by utilizing parallel computing with the graphics processing unit (GPU). The number of independent computations that are carried out concurrently is the primary determinant of degree of parallelism and the acceleration. Nvidia CUDA technology has been introduced in digital image processing computer tomography, however, it can also be utilised for a wide range of other applications.

Image editing is a mechanism that was proposed by Vengayil Nayana Murali et.al. [14], and it involves the procedures of modifying photographs. Modifications are made to the image's pixels in order to improve it. There is a possibility that the image will have noise or that it will be ruined because the camera was not focused properly. Picture restoration is the process of taking an image that has been corrupted or has noise on it and attempting to recreate the original, clean image. A mathematical tool named Principal component analysis (PCA) that takes variables that are correlated and converts them into a number of variables that are uncorrelated. Principal component is a term used to describe variables that are not connected with one another. The direction that exhibits the greatest amount of variation is considered to be the first main component. The location of the second principal component can be found in the region of subspace that is perpendicular to the first principal component's location. The 3rd principle component is determined by moving in the path of the subspace with the highest variance, which is perpendicular to the previous two principal components. The PCA was initially created for a linear transformation that was built on linear algebra. Its initial application was for the reduction of multivariate data. There have been much advancement made in image processing applications, and more research is being done in this field. The GPU serves as a computational coprocessor for the image processing program, which allows for significant time savings. The computing power of both GPUs and multi-processor systems has been utilized by the researchers. An example of how parallel computing can be used in medical imaging is broken down in great depth. Taverna is a workflow engine that is being described, and its design strategy is also being offered, in the context of the development of medical imaging.

Charalambos et al. [15] discussed the concept of parallel image processing which is similar to retinal processing and optical computing. Even though simulation on digital serial computers (DSCs) is more straightforward scenarios, it quickly becomes practically impossible to control and calls for a vast amount of memory in addition to very lengthy programming and execution durations. Recent developments in electronic circuit technology have made it possible to construct vast interconnected parallel logic elements arrays. These arrays are created when the execution of a concurrently statement is carried out on all points utilizing real-time parallel computers.Image Processing Group constructed a machine named parallel cellular logic image processor using MOS memory and TTL logic. The concept of such processor unit is given in Figure 6.

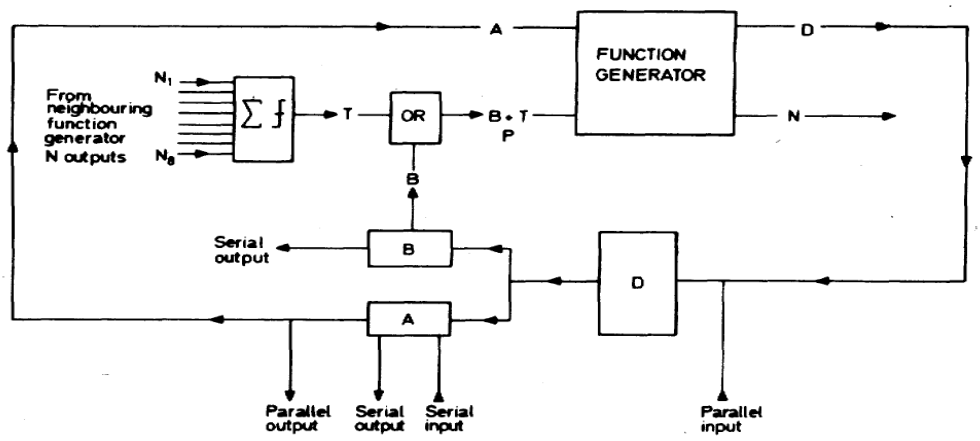


Figure 6 : The cellular logic image processor architecture

Here the cell consists of summation unit and threshold unit along with function generator and OR gate. The cell takes two inputs and gives two independent outputs. All parts of the input image features contribute to all parts of the required output. At every iteration, the parallel processingcontinues at the maximum available speed in the integrated circuit used. A brief description of an example processor of this type is provided, along with an introduction to the processor's code.

The paper also includes several symmetrical examples and directed functions along with parallel techniques. A discussion is provided on the potential uses of such processors in the fields of image processing, pattern recognition, and AI. The creation of DPPs appears to be relatively possible in today's world because of the development of new circuit techniques. These techniques make it possible to manufacture integrated circuits on a big scale at a reasonable cost. The gain of knowledge concerning the building of such processors, which will be reflected in future advancements, is an additional advantage. The amount of real-time applications that can be accomplished by parallel processing is only restricted by the user's creativity.

S. Vijayakumar et.al. [17] proposed the use of GPU that plays a significant part in parallel processing. The majority of computers today come equipped with GPUs, which allow for significantly quicker graphics processing. Parallel algorithms that are executed on GPUs can frequently achieve up to a one hundred times faster than CPU algorithms. This has various practical applications, including image processing, neural networks, signal processing, financial modeling, and many areas. In this study, researchers explore the ways in which GPU programming can be used in the medical image processing. These days, GPU is considered one of the prominent tools for high-performance computing. GPUs are increasingly being used in the analysis of medical images. Many researchers working in the field of medical image processing will find that making concurrent use of the capability of GPUs will be beneficial. In medical image processing, the development of parallel methods for GPU has been resulted in higher processing speed as well as increased accuracy.

Thomas Bräunl et. al. [18] developed a novel method that focuses on low-level real-time image processing for parallel active vision systems. Point operators, dithering, smoothing, local operators, morphological operators, edge detection, and image segmentation are some of the image operator types that are covered in this discussion. The processing of images is becoming increasingly significant in a wide number of application domains. Active vision, which is used in autonomous vehicles, demands a significant amount of computer power to function effectively in real time. Vision enables the construction of sensor systems that are more versatile than other sensor systems. In addition, there is a need faster image processing processes that are mission-critical, such as those used in the evaluation of medical or satellite image data. The benefits can be gained from real-time image processing by using synchronous parallel processing. Several low-level routines and the data parallel implementations of those routines have been investigated by the researchers. Active vision, particularly for imaging systems and mobile robot systems, can obtain a large speed by adopting SIMD systems. The utilization of a data parallel system should be completely hidden from the user, who is only responsible for utilizing an image processing library. It is only necessary to do a single conversion of current sequential image processing library into a parallel operating environment. This can be done by knowledgeable individuals to guarantee that the parallel resources are utilized most effectively. The application programmer, on the other hand, does not need to be concerned with parallel processing in any way. All the parallel algorithms have been implemented using Parallaxis model. These may be run either on sequential workstations or on a data parallel system using the Parallaxis simulator. The creation of a SIMD vision sub-system is going to be the focus of future development.

Preeti Kaur et.al. [19] proposed application program method that do high degree of parallelism for image processing and are an ideal source on multicore platforms. The researchers mentioned that application program that use sequential algorithms can no longer rely on speed improvements brought about by scaling up their technology. The primary objective of parallel processing is to achieve high levels of operation speed. Also to provide solutions those require less time and make more efficient use of available resources. Finding the various parameters, including serial time, parallel time, fork time, join time, and overheads, is the primary topic of this research study paper. The findings indicate that the utilization of parallel computing capabilities in MATLAB, in conjunction with multicore platforms, can significantly increase the rate at which images are processed. In this study, a parallel version of a sequential image processing technique, known as the contrast algorithm, is presented.

Parallel version of the technique was constructed using MATLAB library threads in order to harness the parallel processing capacity of modern processors with multiple cores. When image processing application need to process many images, one can use pipeline processing of images. In the proposed pipeline processing, these images will be in different phases during the same time as shown in figure 7.

| Time | Phase 1 | Phase 2 | Phase 3 | Phase 4 | Phase 5 |
|---|---|---|---|---|---|
| T1 | Image 5 | Image 4 | Image 3 | Image 2 | Image 1 |
| T2 | Image 6 | Image 5 | Image 4 | Image 3 | Image 2 |
| T3 | Image 7 | Image 6 | Image 5 | Image 4 | Image 3 |
| T4 | Image 8 | Image 7 | Image 6 | Image 5 | Image 4 |

Figure 7: Parallel Pipeline Processing

There are few features that the parallel program must have to get high performance as expected. These features are Granularity (Coarse-grained and Fine grained), Synchronization, Latency, Scalability, Speedup & efficiency and Overheads.The objective of the work done by authors is develop parallel program on MATLAB. In order to achieve this they have worked on various parameter like Mean Square Error (MSE) (Need to be Minimum), Peak Signal-To-Noise Ratio (PSNR) (Need to be Maximum), similarity of the structure of two signals, Normalized Absolute Error (NAE), Efficiency, Overheads, Fork Time, Join Time, etc. of digital images using author's proposed technique compared to other existing technique, and the results were analyzed. The enhancement of the image processing algorithm's performance as well as the usage of the multicores to their fullest potential was the primary goals of this implementation. The implementation of parallel processing was roughly 2.5 times faster in terms of performance than the processing of sequential data, as shown in fig. 8.

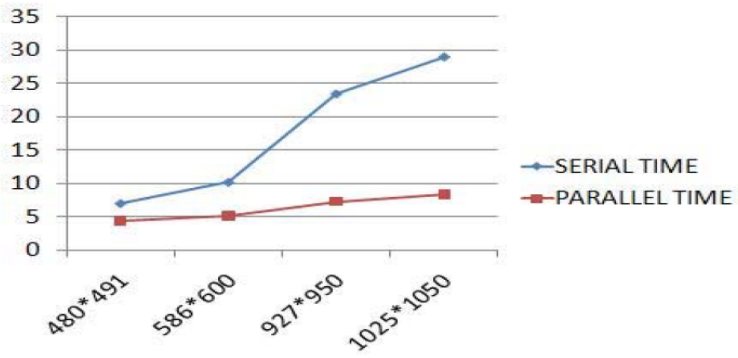


Figure 8 : Results of speedup obtained from different images

This solution is possible with usage of the enormous processing capacity of modern processors that have multiple cores. The goal is to calculate an increasing number of parameters using the same work division in tiles approach in the future and to recommend alternative ways to use the current resources more effectively.

## 3. Classification and Summarization

Table 3 shows the summary and outcome of the Literature Survey along with the model used and the accuracy of these models for face verification and face identification [21] proposed by

various researchers for face verification and face identification. Each method used different dataset and loss function to implement their functions for face verification and face identification.

Table 3 : Summary of various face verification and identification methods.

| Sl. No | Proposed Model | Training Dataset | Face Verification | Face Identification | Loss function used |
|---|---|---|---|---|---|
| 1 | ArcFace | Refined MS-Celeb1M, VGG2 | 99.83% | 83.27% | Additive Angular Margin Loss |
| 2 | CosFace | CASIA - WebFace | 99.73% | 79.54% | Large Margin Cosine |
| 3 | FaceNet | FaceNet | 99.63% | 70.49% | Harmonic triplet loss |
| 4 | DCFL | CASIAWebFace | 99.55% | 75.20% | Correlation loss |
| 5 | DeepID2+ | CelebFaces+, WDRef | 99.47% | 80.10% | Correlation loss |
| 6 | SphereFace | CASIAWebFace | 99.47% | 75.77% | Angular softmax |
| 7 | CenterFace | CASIAWebFace, CACD, Celebrity+ | 99.28% | 65.23% | Center Loss |
| 8 | DeepFR | VGG-Face | 98.95% | 81% | Triplet |
| 9 | DeepFace | SFC | 97.35% | 75% | Cross entropy loss |

The comparative graph of various methods mentioned in table 3 is as shown in figure 9. It is observed that there are high levels of accuracy achieved by various methods reaching nearer to 100% for face verification but coming to face identification, the accuracy is around 80% and there is scope of improvement in it.

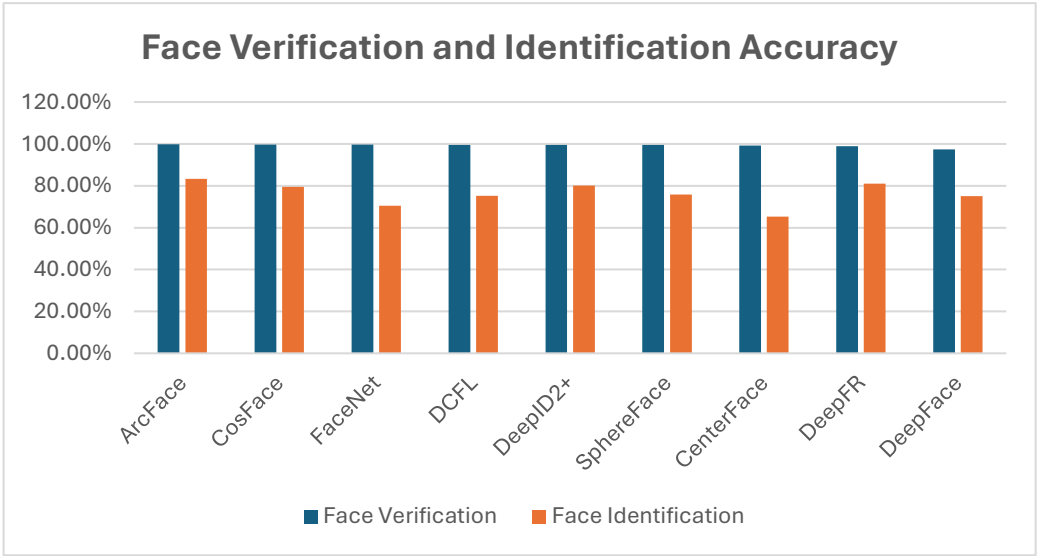


Figure 9 : Comparisan graph for various face verification and identification methods

With the review based on the various research papers, many of them proposed fine grained parallelism to improve the accuracy for face recognition. The review shows that more work needs to be done for fine-grained parallelism at the layer level for DNNs by considering both inter-layer and intra-layer issues.Most of the researchers proposed using either data or model parallelism for extreme-scale computing as future their work. By using coarse-grained parallelization schemes performance can be improved. Coarse-grained data and model parallelism strategies are the two main ways they pay their main attention, which makes their acceleration by adding more hardware resources.

## 4. Conclusion

Many image processing, face detection, verification and recognition applications, if implemented on conventional architectures with single general-purpose processor, cannot meet real-time deadlines.This is due to these applications' complexity and stringent performance requirements. Parallel computing presents an opportunity to speed up the execution of certain algorithms. The concept of parallelism can be implemented in either the hardware or the software levels of a system using a wide variety of tools and techniques. The time complexity can be reduced using GPU which acts as a main computing processor for image processing application speedup. Researchers have a way to utilize the GPU and multi-CPU systems computing capabilities. The many ways that have been mentioned have been shown to be effective, and as a result, a combination of specialized hardware and software tools proves to be beneficial in terms of accelerating the image processing algorithm.Many researchers also suggested utilizing layer-wise fine-grained parallelism for multiple face detection using deep learning as their future work.

In next research work, we propose modelling concurrencies in Deep Neural Networks (DNNs) using inter layer parallelism in network inference and training using deep learning. Optimized performance can reduce the time complexity and make use of current hardware more efficiently without need of more hardware resources such as more GPU/TPU for data and model parallelisms, the DL training speed can be improved. Because of the reduced time complexity, the response time of object detection becomes real time.

## References

1. Santoso, Hadi, and Reza Pulungan. "A Parallel Architecture for Multiple-Face Detection Technique Using AdaBoost Algorithm and Haar Cascade." ISICO 2013 2013 (2013).
2. Rujirakul, Kanokmon, Chakchai So-In, and Banchar Arnonkijpanich. "PEM-PCA: A parallel expectation-maximization PCA face recognition architecture." The Scientific World Journal 2014 (2014).
3. Fatemi, Hamed, Henk Corporaal, Twan Basten, P. P. Jonker, and R. P. Kleihorst. "Implementing face recognition using a parallel image processing environment based on algorithmic skeletons." In 10th Annual Conference of the Advanced School for Computing and Imaging (ASCI 2004), Port Zélande, Ouddorp, June 2-4, 2004, pp. 197-202. Advanced School for Computing and Imaging (ASCI), 2004.
4. Pande, Varun, Khaled M. Elleithy, and Laiali Almazaydeh. "Parallel processing for multi face detection and recognition." (2012).

5. Bhatia, Aashna R., Narendra M. Patel, and Narendra C. Chauhan. "Parallel implementation of face detection algorithm on GPU." In 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), pp. 674-677. IEEE, 2016.
6. Ibrahim, Dalia Shouman, and Salma Hamdy. "Parallel architecture for face recognition using MPI." International Journal of Advanced Computer Science and Applications (IJACSA) 8, no. 1 (2017): 425-430.
7. Gao, Fang, Zhangqin Huang, Shulong Wang, and Xinrong Ji. "Optimized parallel implementation of face detection based on embedded heterogeneous many-core architecture." International Journal of Pattern Recognition and Artificial Intelligence 31, no. 07 (2017): 1756011.
8. Bhutekar, Shivashankar J., and Arati K. Manjaramkar. "Parallel face Detection and Recognition on GPU." International Journal of Computer Science and Information Technologies 5, no. 2 (2014): 2013-2018.
9. Guerfi, Imene, Lobna Kriaa, and Leïla Azouz Saïdane. "An Efficient GPGPU based Implementation of Face Detection Algorithm using Skin Color Pre-treatment." In ICSOFT, pp. 574-585. 2020.
10. Watanabe, Eriko, and Kashiko Kodate. "Implementation of a high-speed face recognition system that uses an optical parallel correlator." Applied optics 44, no. 5 (2005): 666-676.
11. Iqbal, Mohd, and Sandeep Raghuwanshi. "Analysis of digital image processing with parallel with overlap segment technique." International Journal of Computer Science and Network Security (IJCSNS) 14, no. 6 (2014): 52.
12. Singh, Sharanjit, Parneet Kaur, and Kamaldeep Kaur. "Parallel computing in digital image processing." International Journal of Advanced Research in Computer and Communication Engineering 4, no. 1 (2015): 183-186.
13. Skirnevskiy, I. P., A. V. Pustovit, and Mariya Ovseevna Abdrashitova. "Digital image processing using parallel computing based on CUDA technology." In Journal of Physics: Conference Series, vol. 803, no. 1, p. 012152. IOP Publishing, 2017.
14. Murali, Vengayil Nayana., C, Rahul., "Image Processing Application Using Parallel Computing." International Journal of Science and Research (IJSR) (2016).
15. Stamopoulos, Charalambos D. "Parallel image processing." IEEE Transactions on Computers 100, no. 4 (1975): 424-433.
16. Sathesh, A., and Edriss Eisa Babikir Adam. "Hybrid Parallel Image Processing Algorithm for Binary Images with Image Thinning Technique." Journal of Artificial Intelligence 3, no. 03 (2021): 243-258.
17. "Parallel Algorithm for Medical Image Processing using GPU Computing", International Journal of Emerging Technologies and Innovative Research, ISSN:2349-5162, Vol.5, Issue 8, page no.640-642, August-2018,
18. Bräunl, Thomas. "Tutorial in data parallel image processing." Australian Journal of Intelligent Information Processing Systems (AJIIPS) 6, no. 3 (2001): 164-174.
19. Kaur, Preeti. "Implementation of image processing algorithms on the parallel platform using Matlab." Int. J. Comput. Sci. Eng. Technol 4, no. 06 (2013): 696-706.
20. Rawas, Soha, and Ali El-Zaart. "Precise and parallel segmentation model (PPSM) via MCET using hybrid distributions." Applied Computing and Informatics (2020).
21. Shepley, Andrew Jason. "Deep learning for face recognition: a critical analysis." arXiv preprint arXiv:1907.12739 (2019).
22. Goyat, Suman, and A. K. Sahoo. "Evolution of Remote Sensing and Graphical Information System Using CPU-GPU Platform."