

Accelerating CBIR System using Graphics Processing Unit in OPEN CV Environment

Bhavneet Kaur
Assistant Professor
CSE Department
SBSSTC,Punjab

Sonika Jindal
Assistant Professor
CSE Department
SBSSTC,Punjab

ABSTRACT

Content Based Image Retrieval is a system for retrieving the images from a dataset of images. The CBIR has several stages including feature extraction at the initial level and then performing the similarity matching. Later on based upon the similarity matching results the required images can be extracted. Images are a complex data to handle as they are composed of matrices and vectors of data and also due to

multi thread execution of algorithms, programmability and low cost, image processing becomes an appropriate field of achieving parallelism. To parallelize the phases of CBIR a special hardware is used known as Graphics Processing Unit. The objectives of the published work is to develop a Content Based Image Retrieval system and test it on benchmark WANG database using suitable and efficient techniques so as to achieve efficient results. As images are complex and there is an approach of achieving parallelism while dealing with images, so after implementing CBIR, the feature extraction phase has been parallelized and the tendency of GPU processor is exploited to obtain the speed up. An accuracy of range between 22 to 67% for individual categories of the database was achieved. Finally on comparing the execution times of serial implementation and parallel implementation a considerable speed up of 15 % on an average was obtained.

Keywords

CBIR,Graphics Processing Unit, Feature extraction,Similarity matching,SURF,KNN,CUDA,OPEN CV

1. INTRODUCTION

CBIR is a process of retrieving the images from the image database. A large collection of images is referred to as image database. A database of images is a system where image data are integrated and stored [1]. Content-based image retrieval uses the visual contents of an image such as colour, shape, edges or contours, texture, and spatial layout to represent and index the image. In a peculiar content-based image retrieval systems, the visual contents of the images in the database are extracted and described by multidimensional feature vectors. The feature vectors of the images in the database form a feature database. To retrieve images, users provide the retrieval system with example images or sketched figures. The system then changes these examples into its internal representation of feature vectors [2]. In CBIR (Content-Based Image Retrieval), visual features such as shape, colour and texture are extracted to characterize images. Each of the features is represented using

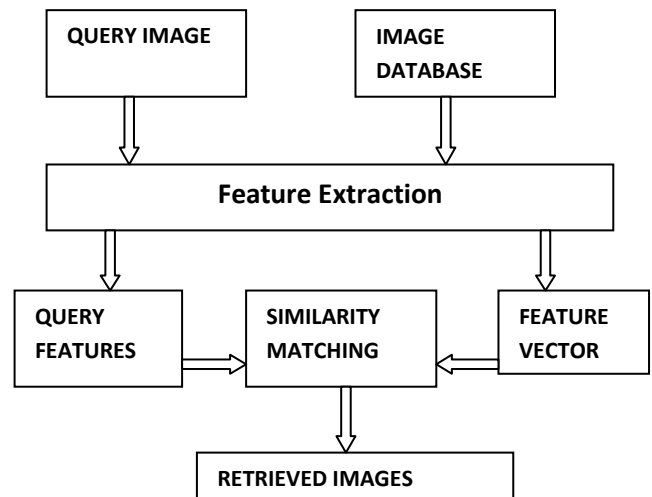


Figure 1. CBIR System

one or more feature descriptors. The main phases of CBIR are feature extraction, similarity matching, indexing, image retrieval and the relevance feedback. During the retrieval, features and descriptors of the query are compared to those of the images in the database in order to rank each indexed image according to its distance to the query. The similarity search is done between the feature vector of the query image and the feature vectors of the database images by using the KNN algorithm [21]. In this, the Euclidian distance formula is used to obtain the nearest neighbours that are the best matching features from those feature vectors. The distance is compared with a threshold distance. Figure 1 shows the phases of Content Based Image Retrieval system.

2. GRAPHICS PROCESSING UNIT

Graphics processing units (GPUs) have rapidly evolved to become high performance accelerators for data parallel computing. Due to its huge parallel hardware architecture and high performance of floating point arithmetic and memory operations on GPUs make them particularly well-suited to many of the same scientific and engineering workloads that occupy HPC (High Performance Computing) clusters, and that is why they are used as HPC accelerators [3]. Apart from being cost-effective such accelerators, GPUs also significantly reduce space, power, and cooling demands, and reduce the number of operating system images that must be managed in comparison to traditional CPU-only clusters of similar aggregate computational capability [4].

The architecture of GPU is divided into Host and Device. GPU is known as device and CPU is called host. Both of them work

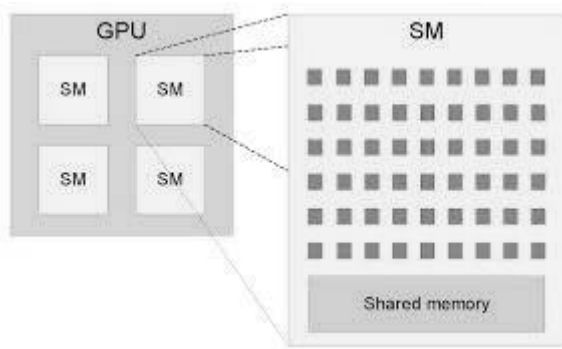


Figure 2. Architecture of GPU

simultaneously [5]. Figure 2 shows the architecture of GPU. Initially the main program is read by CPU and sent to the device (GPU) where it is divided into several threads for further processing. Finally the results are sent back to the host (CPU) and they are displayed.

2.1.1 Compute Unified Device (CUDA)

CUDA is the language tool used for the implementation of the GPU. It is a parallel computing architecture developed by NVIDIA. It was introduced by NVIDIA in November 2006. It allows software developers to access GPUs through standard programming language, C for CUDA (C with NVIDIA extensions and certain restrictions). A well compiled CUDA program can execute on any number of processor cores. CUDA C is the language particularly designed to provide general purpose computing on GPU [6]. CUDA runtime executes multiple copies of our program in parallel on what we called blocks. We called the collection of blocks we launch on the GPU a grid [7].

The objectives of our work is to develop a Content Based Image Retrieval system and test it on benchmark WANG database using suitable and efficient techniques so as to achieve efficient results. As images are complex and there is an approach of achieving parallelism while dealing with images, so after implementing CBIR, we have parallelized the feature extraction phase and exploited the tendency of GPU processor to obtain the speed up.

3. PROBLEM DESCRIPTION

Content-based image retrieval, a technique which uses visual contents to search images from large scale image databases according to users' interests. It includes the phases as feature extraction, similarity matching, and image retrieval. Since, images are a complex data to handle as they are composed of matrices and vectors of data and also due to multi thread execution of algorithms, programmability and low cost, image processing becomes an appropriate field of achieving parallelism. Content Based Image Retrieval process is performed since it is widely used. We define our retrieving the images from WANG database using a CBIR system both sequentially as well as in parallel using GPU. As images are complex and there is an approach of achieving parallelism while dealing with images, so after implementing CBIR, we have parallelized the feature extraction phase and exploited the tendency of GPU processor to obtain the speed up. The images have a stored content which are known as the features of it. Those features that describe directly the images and its data are called low level features that include colour, texture, edges, shape etc. These are basically based on the whole image or on the part of an image as well [8].

The main objective is to process the CBIR system where features are initially extracted and then the similar images are fetched from the system by comparing the query image and the images kept in the data set on the basis of the distance between them. The system should perform well in terms of its accuracy and robustness. The execution time for image retrieval can be reduced by using Graphics Processing Unit whose function is to achieve the parallelism. In our research, we must try to get an efficient CBIR system having good accuracy values and for better speed up GPU shall be used which can be therefore a highlight of our work. So, we need to accelerate the CBIR system so we have computed the efficiency of our serial implementation of CBIR and to parallelize it we must use GPU. The environment used for implementation is OPEN CV [9]. For robust feature extraction we should use SURF technique and KNN for similarity matching and to find the speed up a comparison of the execution results of CBIR on CPU and GPU has to be done at the end.

4. SYSTEM ARCHITECTURE

Here the clear prototype of our implementation is defined that cover the following phases as:

1. To perform feature extraction on Central Processing Unit and also on GPU.
 - CPU part using OPEN CV.
 - GPU part using CUDA on OPEN CV.
2. To perform similarity matching on the extracted features. Use KNN for extracting similar images.
3. To calculate the performance metrics of CBIR. The performance parameters include Acceptance ratio, rejection ratio, Poor matcher, execution times on CPU as well as on GPU, True Positives, False Positives, True Negatives, False Negatives and thus create confusion matrix from it so as to calculate the overall accuracy of the CBIR system.
4. To compare execution times of both serial as well as parallel implementation. Therefore, we can calculate the individual speed up and also an average speed up.

The system architecture is divided into two main parts:

- A. SERIAL IMPLEMENTATION where Feature Extraction is performed on host that is CPU and Similarity Matching is also performed on host.
- B. PARALLEL IMPLEMENTATION where Feature Extraction is performed on Device that is GPU and Similarity Matching is performed on Host.

OPEN CV is a platform provided for image processing implementation using C++ interface. Open CV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision, developed by Intel, and now supported by Willow Garage and Itseez [10]. It is an open source computer vision library in C or C++ and optimized for real time applications. It provides

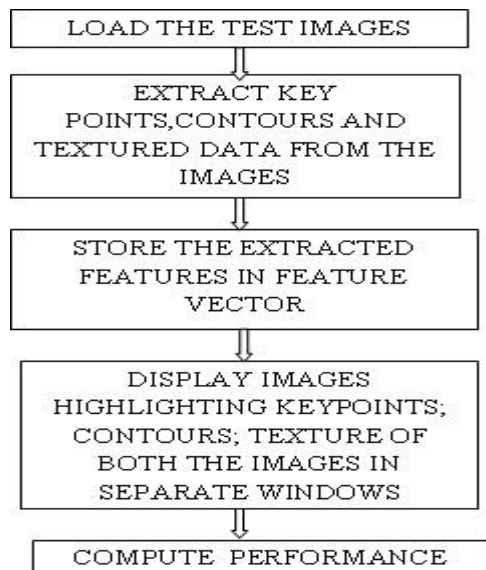


Figure 3. Feature Extraction Algorithm

interface to Intel's Integrated Performance Primitives (IPP) with processor specific optimization (Intel processors) [11].

4.1.1 Feature Extraction Algorithm

Feature extraction not only finds its use in CBIR domain but also in various other fields of image processing, data mining and speech recognition as well. Since the result of this phase

is a set of features that further can be taken as input for other phases of CBIR so, it is very important to choose a correct technique for feature extraction. The technique we have used is SURF that is Speeded Up Robust Features [12] that helps in accelerating feature extraction for images. It is a method that performs a scale and rotation-invariant interest point detector and descriptor [13]. The low level features are extracted since they work on the whole of the image [14]. We have extracted the key points, contours and texture.

This is the starting phase of our work where test images has been loaded from the WANG data set which is a benchmark database [15]. The images are taken in JPEG format and we can also display them using imshow function in OPEN CV [16]. The images are converted into grey scale. Then extraction of the key points from the images is done using SURF detector.

It uses min hessian matrix [17]. For extracting the contours from the images a declaration and definition of a function in OPEN CV is done. The edges are detected using the Canny edge detector [18] and for textures an algorithm provided in OPEN CV is used. The texture of images are what is got as an output. Figure 3 shows the flowchart of feature extraction algorithm that have been used with OPEN CV.

4.1.2 Similarity Matching Algorithm

The similarity matching is one of the important phases that is a part of different applications and CBIR. It helps in finding the similarity between different features on the basis of distance. There are several algorithms available for similarity matching. The similarity is performed by taking the feature descriptors of query images and those of the database images and then it computes the distances between them based upon certain formulae [19]. On the basis of which it finds the similar matching images from the database. One can compute the average distance and the quadratic histogram distance as well. The K Nearest Neighbour search is one of the popular methods

that compute the distances between the features or points and find out the similarity between them. It is also known as Brute Force method [20]. The algorithm has been depicted in figure 4.

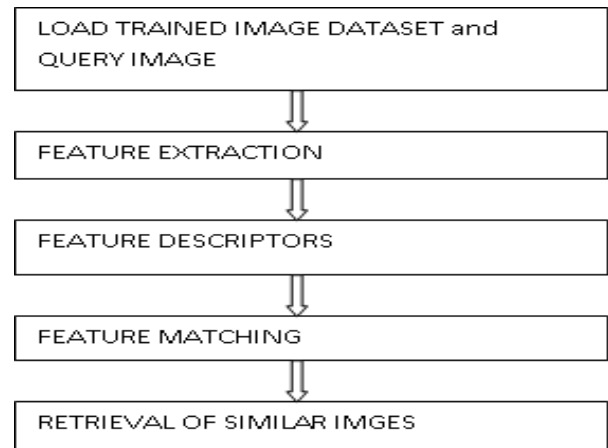


Figure 4. Similarity Matching Algorithm

The similarity algorithm goes as follows:

1. The algorithm starts with loading the WANG dataset. The dataset contains the different category images in the jpeg format like dinosaurs, horses, food, elephants, etc. Since to fetch the similarity between the images it needs to have their features. So, it extract the features of the images of dataset that become the part of our training set and the features of the query image that it inputs to the system.
2. The features that were initially extracted in the first phase are carried forward here. The descriptors of the images are calculated known as feature descriptors.
3. Then finally the query image feature descriptor and the training set descriptors are matched and thus the similar images are retrieved.

4.1.3 Serial Implementation

In the serial implementation of the work as the figure 5 depicts, the system initially inputs one of the image randomly of any category from the WANG database which is the test image and also on the other hand it has input the WANG data set containing different category's images. Then it has extracted the features of query image and the dataset images using SURF feature extractor. After calculating the feature descriptors we have performed similarity matching using KNN method. From this we had retrieved the a-like images from the dataset. It has performed the loading of the images, feature extraction and similarity matching using OPEN CV platform. Then it could find the performance of our CBIR system. So, the image loading, feature extraction and similarity matching are performed on Central Processing Unit that is host. The execution time of CPUs also calculated.

4.1.4 Parallel Implementation

In the parallel implementation of our work as the figure 6 depicts, the system has performed the image loading and feature extraction on the device and similarity matching on the device.

Initially it gives the path of the WANG dataset to the host where it further uploads it to the device so that it can read the

corresponding images from the path location. After loading and reading of the image dataset, the device extracts the features and stores it in the feature descriptors. These are then sent to the host where it downloads the descriptors. Meanwhile the host also calculates the descriptors of the query image. The matcher performs similarity matching

between query image and trained dataset. The host retrieves the similar images. After that we have calculated the

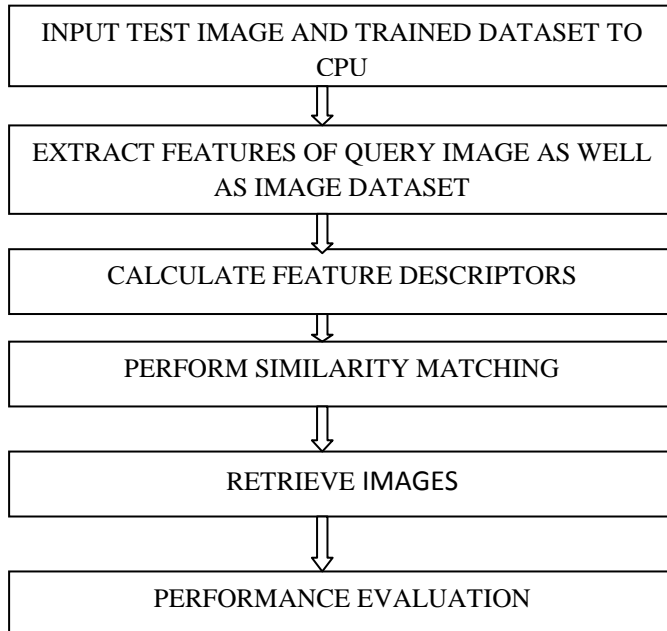


Figure 5. Serial Implementation

execution time of GPU. At the end a comparison is done between the execution times of CPU & GPU and hence found out the speed up of our CBIR system.

5. RESULTS AND DISCUSSION

Visual Content-Based Image Retrieval consists in searching an image database to retrieve images which are visually similar to a given query image [21]. Content-based image retrieval (CBIR) is the set of techniques for retrieving semantically-relevant images from an image database based on automatically-derived image features. The main phases of CBIR on which the system has worked is feature extraction which is the initial phase and the finally the similarity matching phase. It worked on WANG database [22]. and calculated different metrics to validate the CBIR system as processed on OPEN CV platform. Initially the accuracy of the sequentially processed CBIR system and then also calculated the speed up of the parallel system.

1. Detect-ability: The system has initially taken Wang data set having several category images. Initially we have extracted the features form the images using SURF Feature Extractor. The features extracted are low level features that include key points, contours and texture. For checking the extent of features extracted it has used a parameter called Detect ability. It means how much efficiently does the technique detects the features. It also extracted the features of the query image. It measures the ability to obtain sufficient feature point correspondences in the images [23]. After getting and observing all the features as

shown in figure 7 a conclusion tells that the technique so use has a good detect-ability.

2. Confusion Matrix:-A confusion matrix [24] contains information about actual and predicted classifications done by a classification system. Performance of such systems is commonly evaluated using the data in the matrix. Figure 8 has shown the confusion matrix .
3. Accuracy:- The algorithm a has used the above said parameters which are True Positives, True Negatives, False Positives & False Negatives to calculate the accuracy of our CBIR system. It basically shows how efficiently the system is able to detect the images based on the extracted features. The retrieval scheme must be accurate, i.e. the retrieved images must resemble the query image. We classify a retrieval as accurate if for a given query image the perceptually (to a human) most similar image in the database is retrieved by the system as the topmost [25]. The results of accuracy of all the categories has been summarized the results in a graph shown in figure 9. The results so shown, validates the proposed CBIR system according to their respective values.

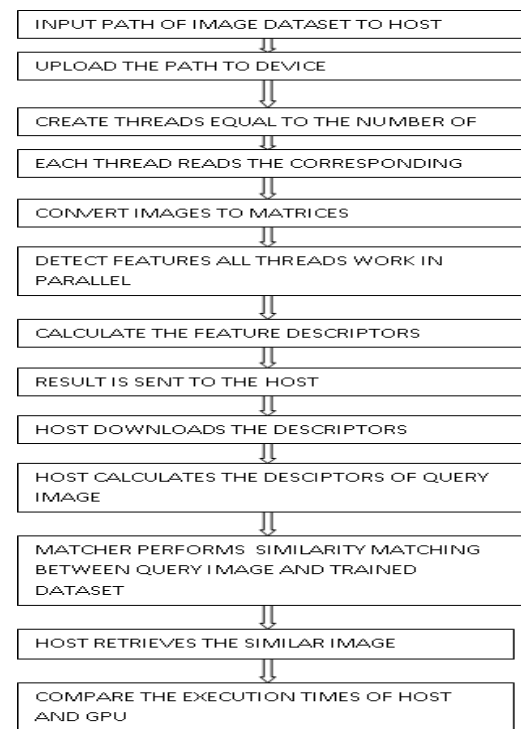


Figure 6. Parallel Implementation

$$\text{Accuracy} = \frac{TP + FP}{TP + TN + FP + FN}$$

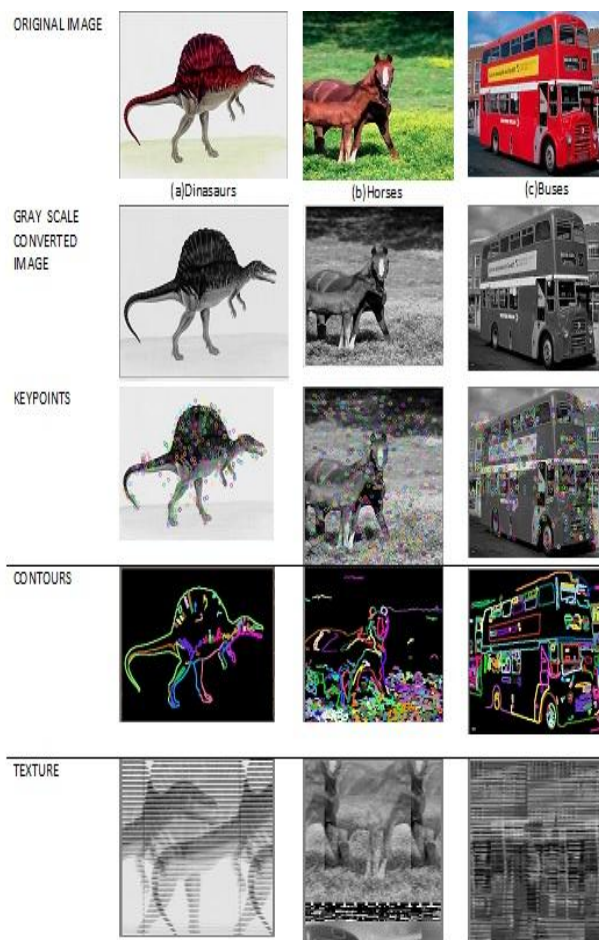
4. Speed Up: Speed up means how faster the execution has been. At last it used the GPU for parallel implementation of feature extraction and CPU for implementing the retrieval of similar images. It has calculated the execution times from the serial implementation of the system and parallel implementation of the same and on comparing these two, speed up could be fetched. An observation can be made that the CBIR system has obtained an accuracy of range between 22 to 67% for individual categories and an average speed up of 15%. The figure 10 shows the comparison of execution times of

the system on CPU as well as on GPU. One can also observe from the displayed results the individual speed ups of all the categories of the database in figure 11.

6. CONCLUSION

The algorithm has performed the research on Content Based Image Retrieval which is the system used for image retrieval from a dataset of images. In this case it has worked on the benchmark database that is the WANG database. The platform that is used for our implementation is the OPEN CV that is one of the tool used for image processing algorithms composing of C++ libraries. Initially it has performed the feature extraction on the image dataset and the query image as well using the SURF feature extractor and using these features we have performed similarity matching with KNN algorithm. All of the processing is performed using the host that is Central Processing Unit. To validate the proposed system a checking of the accuracy of our system is done. An accuracy of range between 22 to 67% for individual categories of the database has been fetched.

The algorithm has also tried to reduce the execution time by using GPU. In this case it has performed the feature extraction using device that is GPU using SURF with GPU and finally retrieved the similar images using KNN itself. Finally on comparing the execution times of serial implementation and parallel implementation it has obtained a considerable speed up of 15 % on an average.



	buses	butterfly	dinosaurs	elephants	flaura	food	horses	mountain	people	space	sunset	Unmatched
buses	32	2	60	38	1	77	64	6	4	3	1	143
butterfly	4	18	15	10	7	14	8	19	5	5	7	319
dinosaurs	14	5	80	14	5	30	20	7	3	3	4	246
elephants	27	8	34	38	10	72	61	10	8	5	9	149
flaura	4	8	5	8	18	8	8	24	17	15	8	308
food	17	5	47	42	3	73	65	8	3	3	3	162
horses	22	9	25	39	8	58	64	9	6	4	2	185
mountains	8	15	9	9	20	8	9	17	13	18	8	297
people	9	9	8	13	16	27	18	15	15	17	8	276
space	10	9	15	14	15	13	15	10	8	15	10	303
sunset	10	5	9	9	5	9	9	8	6	6	17	338

Figure 7. Confusion Matrix



Figure 8. Extracted Features-Keyoints,Contours & Texture

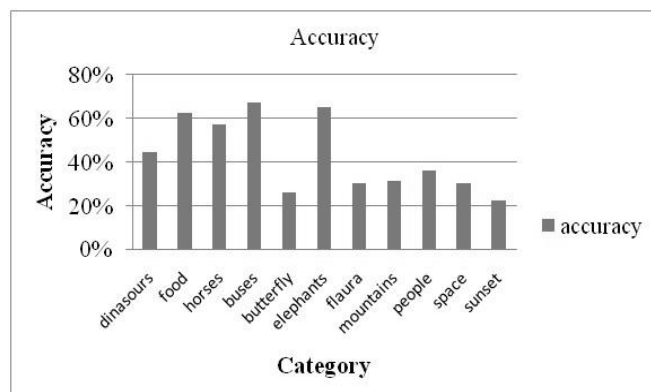


Figure 9. Accuracy Results

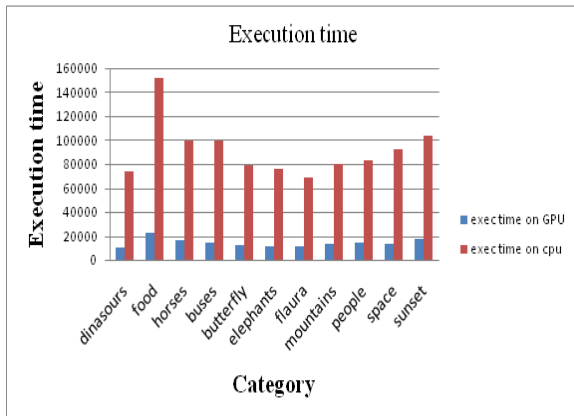


Figure 10. Comparison of execution times of the system on CPU as well as on GPU

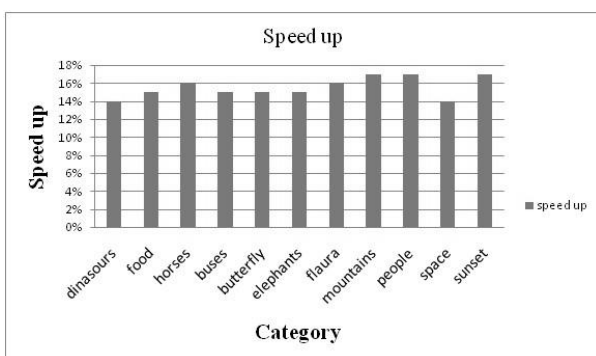


Figure 11. Speed Up

7. FUTURE SCOPE

The algorithm can be performed on various other benchmark databases. We can also use the other feature extraction techniques provided in OPEN CV like SIFT,ORB,FAST etc. and GPU can also be used on the similarity search phase of CBIR and a speed up can be calculated.

8. REFERENCES

[1] Nezamabadi-pour, H. and Kabir,2004,Image retrieval using histograms of uni-color and bi-color blocks and directional changes in intensity gradient, Pattern Recognition Letters.

[2] Muller, H., Michoux, N., Bandon, D., and Geissbuhler, A. (2004). A review of content based image retrieval systems in medical applications clinical benefits and future directions.International journal of medical informatics, 73(1):1{23}.

[3] Samant, S. S., Xia, J., Muyan- Ozcelik, P., and Owens, J. D. (2008). High performance computing for deformable image registration: Towards a new paradigm in adaptive radiotherapy. Medical physics, 35(8):3546{3553}.

[4] Kindratenko, V. V., Enos, J. J., Shi, G., Showerman, M. T., Arnold, G. W., Stone, J. E., Phillips, J. C., and Hwu, W.-m. (2009). Gpu clusters for high-performance computing. In Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on, pages 1{8}. IEEE.

[5] Owens, J. D., Houston, M., Luebke, D., Green, S., Stone, J. E., and Phillips, J. C. (2008). Gpu computing. Proceedings of the IEEE, 96(5):879{899}.

[6] Sanders, J. and Kandrot, E. (2010). CUDA by example: an introduction to general-purpose GPU programming. Addison-Wesley Professional.

[7] KUMAR, D. P. (2012). Interactive Visualization and Tuning of Multi-Dimensional Clusters for Indexing. PhD thesis, International Institute of Information Technology Hyderabad.

[8] Zhou, X. S. and Huang, T. S. (2000). Cbir: from low-level features to high-level semantics. In Electronic Imaging, pages 426{431}. International Society for Optics and Photonics.

[9] Culjak, I., Abram, D., Pribanic, T., Dzapo, H., and Cifrek, M. (2012). A brief introduction to opencv. In MIPRO, 2012 Proceedings of the 35th International Convention, pages 1725{1730}. IEEE.

[10] Lagani{\`e}re, Robert (2011), OpenCV 2 Computer Vision Application Programming Cookbook: Over 50 Recipes to Master this Library of Programming Functions for Real-time Computer Vision, Packt Publishing Ltd.

[11] Agam, G. (2006). Introduction to programming with opencv. Online Document, 27:itsindia. initsindia.in.

[12] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2):91{110}.

[13] Malik, J., Belongie, S., Leung, T., and Shi, J. (2001). Contour and texture analysis for image segmentation. International journal of computer vision, 43(1):7{27}.

[14] Zhang, Q., Chen, Y., Zhang, Y., and Xu, Y. (2008). Sift implementation and optimization for multi-core systems. In Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on, pages 1{8}. IEEE.

[15] Li, J. and Wang, J. Z. (2003). Automatic linguistic indexing of pictures by a statistical modeling approach. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 25(9):1075{1088}.

[16] Bhavneet Kaur ,Sonika Jindal,(2014),An implementation of Feature Extraction over medical images on OPEN CV Environment, in Proceedings of, IEEE Devices, Circuits and Communications (ICDCCom), 2014 Ranchi, India.

[17] Rosten, E., Reitmayr, G., and Drummond, T. (2005). Real-time video annotations for augmented reality. In Advances in Visual Computing, pages 294{302}. Springer.

[18] del Blanco, C. R., Jaureguizar, F., Salgado, L. (2008). Motion estimation through efficient matching of a reduced number of reliable singular points. In Electronic Imaging 2008, pages 68110N{68110N}. International Society for Optics and Photonics.

[19] Bhavneet Kaur,Sonika Jindal (2014),Study of K Nearest Neighbour Applications in Image Processing with Graphics Processing Unit , International Journal on Advanced Computer Theory and Engineering (IJACTE), ISSN (Print): 2319-2526, Volume -3, Issue -3.

[20] Garcia, V., Debreuve, E., and Barlaud, M. (2008). Fast k nearest neighbor search using gpu. In Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on, pages 1{6}. IEEE.

[21] Veltkamp, R. C., Tanase, M., and Sent, D. (2001). Features in content-based image retrieval systems: a

- survey. In *State-of-the-art in content-based image and video retrieval*, pages 97{124. Springer.
- [22] Wang, J. Z., Li, J., and Wiederhold, G. (2001). Simplicity: Semantics-sensitive integrated matching for picture libraries. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(9):947{963.
- [23] Dwarakanath, D., Eichhorn, A., Halvorsen, P., and Griwodz, C. (2012). Evaluating performance of feature extraction methods for practical 3d imaging systems. In *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*, pages 250{255. ACM.
- [24] Hamel, L. (2009). Model assessment with roc curves.
- [25] Muller, H., Müller, W., Squire, D. M., Marchand-Maillet, S., and Pun, T. (2001). Performance evaluation in content-based image retrieval: overview and proposals. *Pattern Recognition Letters*, 22(5):593{601.