2D Log Gabor And SVD Based Parallel Texture Feature Extraction UsingNVIDIA GPU

Supervisor: Dr. Jia Uddin

Sharif Ahmmad Raja	12101092
AminurRabRatul	12101116
SakibAnjumNiloy	12101085

Department of Computer Science and Engineering

School of Engineering and Computer Science

BRAC University

Submitted on: 20thApril 2016

DECLARATION

We, hereby declare that this thesis is based on the results found by ourselves. Materials of work found by other researcher are mentioned by reference. This Thesis, neither in whole or in part, has been previously submitted for any degree.

Signature of Supervisor :

Dr. Jia Uddin

Signature of Author :

Sharif Ahmmad Raja

AminurRabRatul

SakibAnjumNiloy

Acknowledgements

All thanks to Almighty Allah, the creator and owner of this Universe. This thesis work could not be possible without his blessing. He gives us strength and abilities to complete this thesis. We seek forgiveness from Allah for every mistake, we made, of course, during the period of writing this thesis.

We would also like to mention the name of Prophet Muhammad, who are our inspiration in every rightly guided things we do and every righty guided choice we make.

This thesis work suggested by Dr. Jia Uddin, Assistant Professor, Department of Computer Science and Engineering, BRAC University. We would like to express our sincere gratitude to Sir, for his support, encouragement and guidance throughout this thesis work. He helps us uncountable occasions and this thesis work would not be possible without his help, support, comments and valuable ideas. It has been great honor to work with him.

Next, we would like to thank Peter Kovesi, the Centre for Exploration Targeting at The University of Western Australia, to helping us programming with MATLAB. After that our gratitude towards Dan Doherty from MATHWORKS**,** to helping us in our GPU implementation part.

Finally, we are grateful to the influences and support of faculties, friends from BRAC University and our family members, which is our giant motivation in this thesis journey.

CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

Page

ABSTRACT

Texture feature is one of the most popular technique in image segmentation, classification, retrieval and many others. Now a days, among other ways of texture feature extraction, Gabor filtering has been widely used. Here, we are presenting a well ordered two dimensional texture feature extraction method. First, we convert the image to gray level. Then a 2D Log Gabor filter with different frequencies decomposed with the SVD algorithm applies on each converted part of gray level image to extract appropriate distinctive texture information. To evaluate the performance of proposed model, we utilize singular values of SVD as a feature vector. For classifier, we use Naïve Bayes classifier for training and testing our experimental dataset. In our experimental set up we utilize an NVDIA GeForce GTX780 graphics card. Experimental result showed this parallel implementation of our model is56X faster than conventional CPU implementation.

1. INTRODUCTION

1.1 Motivation

Feature extraction reducing the amount of resources where we actually required to describe this things with a large set of data. Generally it does constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. Texture is important for recognition, interpretation of data, computer vision, and image processing and pattern recognition. By use the extraction it is possible to solve some real life problems. These problem actually solve by combined of Feature detection, feature extraction, and matching.

In recent years, big efforts have been devoted to the attempt to improve the performance of image retrieval systems and research has explored many different directions trying to use with profit results achieved in other areas. We see in the past or in recent years texture feature use in many works. Recently for more faster or accurate result texture feature extraction parallel done with GPU. For example: from [31] we see that a GPU based parallel computing method texture feature extraction method for accelerating of color moments. In [3] they use Texture Feature Extraction in Region-Based Image Retrieval System. Here their work based on the Projection-Onto-Convex-Sets (POCS) theory. Texture feature extraction widely in medical field in this day. From [4] we find that Fractal Dimension Applied in Texture Feature Extraction in X-ray Chest Image Retrieval. They show that the retrieval system with the fractal algorithm has relatively high precision and recall up to about 85%.Based on a new Tamura feature extraction algorithm the forensic image experiment happened in [8]. In [2] Gabor filter and SVD based Texture Feature Extraction system Method for State Monitoring of Induction Motors. Fourier-Bessel expansion with generalized discriminant analysis (GDA) [2], stationary wavelet packet transform (SWPT) [5].Similar to conventional 1D methods, now a days, 2D texture feature extraction methods are used [6, 7, 9-13].

1.2 Aims and objectives

This thesis specially targeted on the issue of establishing a new tactics which can help us to establish efficient operation of feature extraction and classification techniques. The step by step procedure of this thesis are summarized here. This work marked the following issues:

- At first we have to convert the sample to gray level image to achieve the texture information from the converted images.
- In this thesis we use 2d log Gabor and SVD for texture feature extraction. To extract details texture information, we applied a 2D log Gabor filter with various types of frequencies and different orientation angles is applied on each image and extract 2D log Gabor filtered images.
- Then decompose the SVD algorithm on each log Gabor filtered image to convert singular values of SVD as a feature vectors and for classifier we use Naïve Bayes.
- After that find better result and reducing computational complexity our proposed model implemented parallel in NVIDIA GPU.

1.3 Thesis Contribution

In this thesis we implemented a new parallel processing model. We use 2D Log Gabor filter (improved version of Gabor filter) and SVD to establish a new texture feature extraction method.

After that we run our proposed model through parallel processing with GPU to reduce computational cost

We used seven different category picture which was in total 1200 image to test our proposed model. The category we use are marble, upholstery, carpet, wallpaper,corduroy and glass. We first find how much accurate result actually our model show. We find very high accuracy percentage for each and every dataset.Like for marble we find 97.5% accuracy rate, upholstery also show 97.5%, carpet show 93.5% accurate result, wallpaper show 91.5% accuracy rate, glass show 95% correct result and corduroy images show 85.5% accurate result. We compare our Log Gabor and SVD based model with Gabor and SVD based texture feature extraction model and also some other existing model. After comparison with Gabor and SVD based model our proposed model show great result for each and every dataset. We then find some proved texture feature extraction model for same type of dataset. After comparing with that we find an extremely better result.

Parallel computing actually done to reduce our time cost for each even types of image. We take each type of image with different resolution and run parallel processing on them with NVIDIA GPU. This will cut run time of our each image of each dataset.For NVIDIA GPU GeForce GTX780 this GPU version works 56% faster than CPU implemented model.

1.4 Thesis Outline

Orientation of this thesis is as follows:

- ✓ Chapter 1 is the formal introduction of the thesis. We have discussed our motivation and approach towards our proposed topic in this chapter.

- ✓ Chapter 2 is Literature review of our thesis work. Here we also showed some example so similar model. This chapter also focuses on the background study that covers all the important basics needed for this research along with their formal definitions and representations. In this chapter we have described Gabor filter, Log Gabor Filter, SVD Algorithm, Naïve Bayes Classifier and NVIDIA GPU.
- ✓ Chapter 3, focuses on the proposed work. Here, we have described our whole workflow to achieve the results following our proposal.
- ✓ Chapter 4, is basically describes the various features of hardware and software's that was used in our experimentation.
- ✓ Chapter 5, is the result and analysis part where we have explained our acquired results after implementing our approach. We have attached visual of both the input and generated output in this chapter. We have also made a comparative analysis of our results with other model.
- ✓ Chapter 6, is the conclusion part of our paper.

2. LITERATURE REVIEW

2.1 Introduction

Now a days we see there are many model for texture feature extraction. We found several paper about this topic. In every paper we found different types of technique to implementing texture feature extraction. Like, [14] they did designing compact Gabor filter banks for an efficient texture feature extraction.. In [1] they use Gabor filter and SVD based extraction method for state monitoring of induction motor. In [15] they did texture feature extraction using 2-D Gabor Filter to extract the texture features of Inverse Fast Fourier Transform (IFFT), texture energy and transformed IFFT. In [18] used Gabor filter for efficient texture feature extraction. Firstly, to reduce computational complexity of texture feature extraction they use Gabor filter bank and after that Gabor filter bank thus designed produces low-dimensional feature representation with improved extract classification. In [9], Tuan et al. proposed model use scale-invariant feature transform (SIFT). In [16, 17] used spectral textural method for textural analysis and multi-spectral image classification. In [19] Wang, use Gabor wavelet for feature extractor and for high level decision making they use Fuzzy Adaptive Resonance Theory (Fuzzy ART). Weitao [38] designing a compact Gabor filter bank for efficient texture feature extraction. In [20] they done comparison study between Gabor and log Gabor wavelet for texture analysis. In [21] M-band wavelet theory is introduced and for texture extraction M band wavelet histogram technology is applied. Every process have some defects like: SIFT has uncertainty in the number of key points and high computational cost [3], problem in wavelet texture analysis is that the number of features tends to become large, especially for wavelet packet decompositions [22]. Facial expression classification using Gabor and Log Gabor filter in [36]. Xiao [37] use log Gabor wavelet and its application to detect image edge. In [39] Log Gabor filter using iris recognition system.In [23], an efficient image representation approach was proposed using a Log Gabor filter, where Log Gabor filters possesses a number of interesting mathematical properties: (i) it has a smooth and indefinitely differentiable shape, (ii) does not have side lobes and (iii) generates high-dimensional representations. So, we try to use the Log Gabor and SVD based model to find the better result in CPU.

We then find that many GPU implemented texture feature extraction also done in these recent time to get a better result than past. Like: from [31] we see that a GPU based parallel computing method texture feature extraction method for accelerating of color moments. In [33] they proposed an object recognition using SVD and log-log frequency which is sampling with Gabor kernel. In [32] they run performance evaluation on feature extraction algorithm i.e. Canny Edge detector on GPGPU. After that, a fast feature extraction in object recognition using SIFT method parallel processing on both CPU and GPU. From every GPU implemented parallel computing method prove that, whenever we implement any method by parallel implementation it will gives us the better result than CPU implemented version.

2.2Structure of Different Models

In [38] we found following type of model:

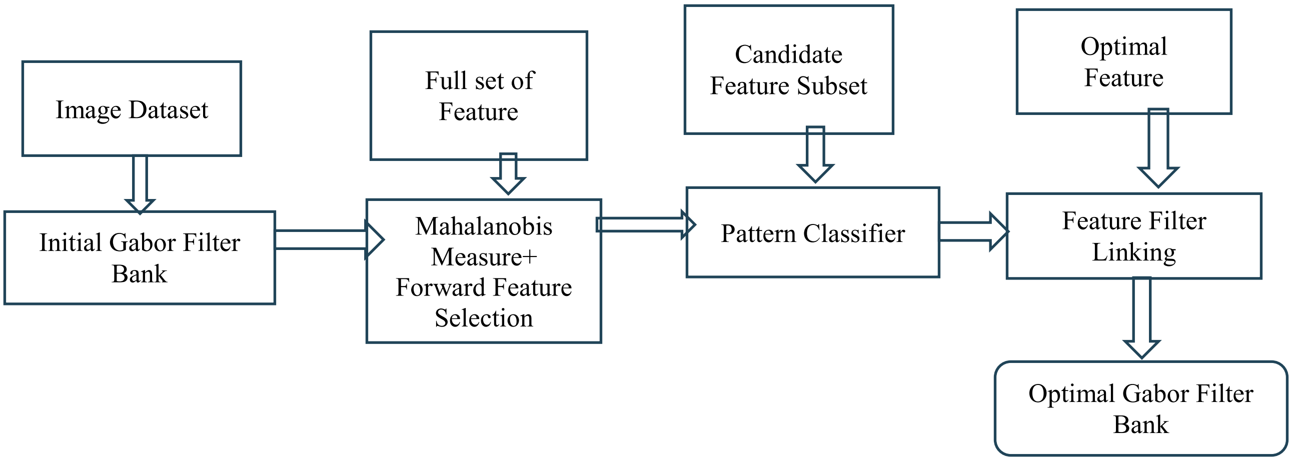


Figure1. Block Diagram of Optimal Gabor Filter Bank Process

From [16] we find the following model:

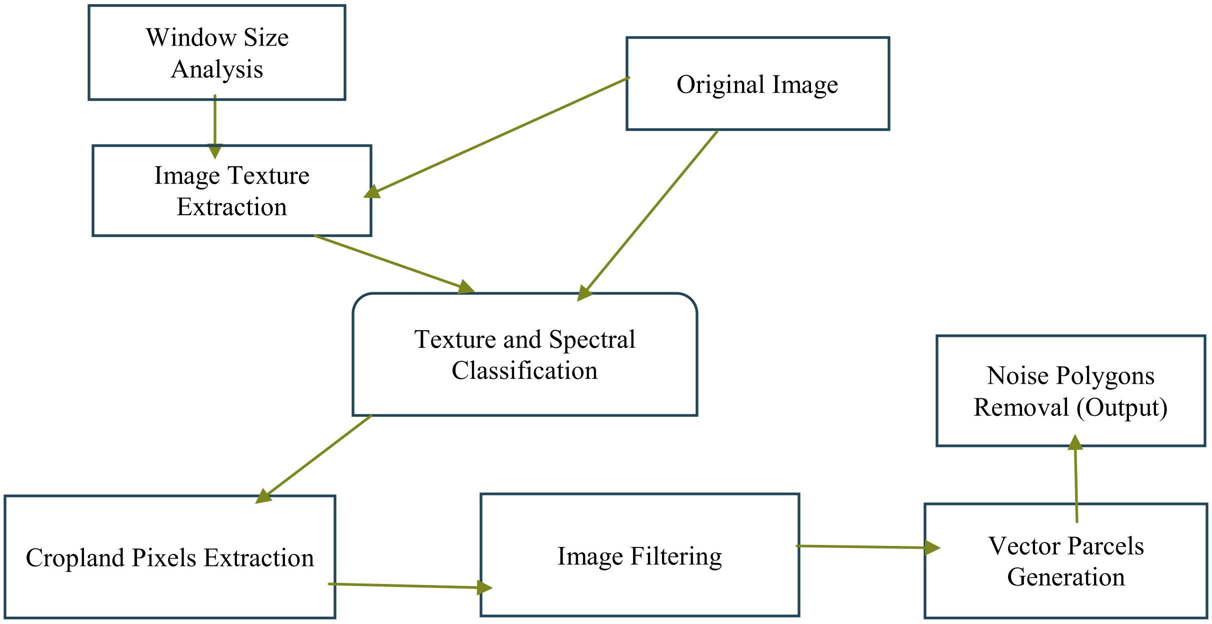

Window Size Analysis
Original Image
Image Texture Extraction
Texture and Spectral Classification
Noise Polygons Removal (Output)
Cropland Pixels Extraction
Image Filtering
Vector Parcels Generation

Figure 2. Block Diagram of Noise Polygons Removal System

2.3Gabor Filter

In image processing, a linear Gabor filter is used for identifying texture information. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation. The 2D representation of time domain vibration signals give additional features compared to the state-of-arts 1D approaches. In this study, in order to extract meaningful texture information in a 2D gray level image, a 2D Gabor filter with various frequencies and orientation angles is used. In general, the 2D Gabor filter consists of a complex exponential centered at a given frequency and modulated by a Gaussian envelop. Because of the complex exponential, the filter has both real and imaginary parts [21-23, 26]. AGabor filter can be formulated as.

$$g(x,y)=\frac{1}{\sqrt{\pi\sigma_x}\sigma_y}e^{-\frac{1}{2}[\frac{(x-x_0)_r^2}{\sigma_x{}^2}+\frac{(y-y_0)_r^2}{\sigma_y{}^2}]}e^{j(\omega_{0x}x+\omega_{0y}y+\theta)} \tag{1}$$

Where $\begin{bmatrix}(x-x_0)_r\\(y-y_0)_r\end{bmatrix}=\begin{bmatrix}\cos\alpha & \sin\alpha\\-\sin\alpha & \cos\alpha\end{bmatrix}\begin{bmatrix}(x-x_0)\\(y-y_0)\end{bmatrix}$, σ_x,σ_y are the variances along x and y axis respectively, ω_0 is the angular frequency of the sinusoidal function, and θ represents the orientation of Gabor filter.

The filtering operation with a Gabor filter can be expressed as

$$G(x,y)=I(x,y)*g(x,y), \tag{2}$$

Where $G(x,y)$ denotes the complex convolution result. The magnitude of each pixel in a Gabor filtered image is calculated using the following equation [13].

$$m(x,y)=\sqrt{k_c^2(x,y)+k_s^2(x,y)}, \tag{3}$$

Where $k_c=re(g(x,y)*i(x,y))$, $k_s=\mathrm{Im}(g(x,y))*i(x,y)$ and $*$ denotes a convolution operator.

The frequency and orientation angle of the Gabor filter highly effect on the classification accuracy. Therefore, in order to extract more distinct texture features, a bank of Gabor filters with different scales and orientations are applied to the high frequency image to obtain Gabor filtered images. However, no formal techniques exist for choosing the parameters and experience-guided intuition, trial and error, or heuristic search is used. In this study, the frequency to 4 and orientation angle to $\pi/4$ be set.

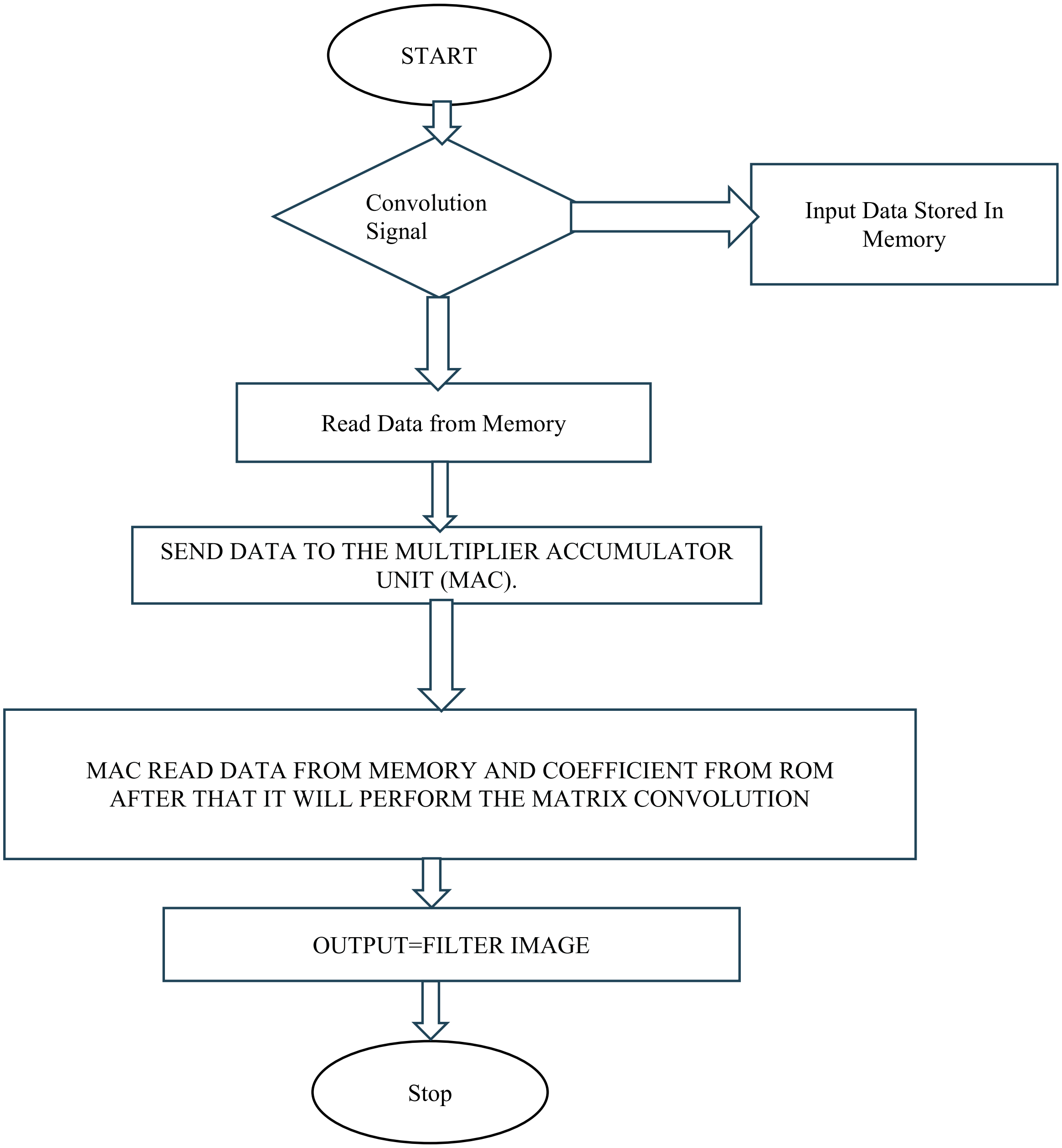

START
Convolution Signal
Input Data Stored In Memory
Read Data from Memory
SEND DATA TO THE MULTIPLIER ACCUMULATOR UNIT (MAC).
MAC READ DATA FROM MEMORY AND COEFFICIENT FROM ROM AFTER THAT IT WILL PERFORM THE MATRIX CONVOLUTION
OUTPUT=FILTER IMAGE
Stop

Figure 3. Flow-Chart of Gabor Filter

2.42D Log-Gabor Filter

2D Log Gabor filter basically use for find the texture information. 2D Log Gabor filter with various frequencies and orientation angles is used. It is the latest version of Gabor filter. It has complex exponential frequency. 1D Log-Gabor frequency is:

$$G(f) = exp\left(\frac{-\left(\log\left({}^{f}/_{f_0}\right)\right)^2}{2\left(\log\left(\frac{\sigma}{f_0}\right)\right)^2}\right) \qquad (4)$$

f_0 Is center frequency of the used filter and σ actually show effects of the bandwidth of this filter.

After Weber–Fechner law applies:

$$G(f) = \frac{f}{f_0} exp\left(\frac{-\left(\log\left({}^{f}/_{f_0}\right)\right)^2}{2\left(\log\left(\frac{\sigma}{f_0}\right)\right)^2}\right)(5)$$

The orientation component is a Gaussian distance function according to the angle in polar coordinates:

$$G(f,\theta) = exp\left(\frac{-\left(\log\left({}^{f}/_{f_0}\right)\right)^2}{2\left(\log\left(\frac{\sigma}{f_0}\right)\right)^2}\right) \exp\left(\frac{-(\theta-\theta_0)^2}{2\sigma_\theta^2}\right)(6)$$

Here: f_0 is center frequency, σ_f the width parameter for the frequency, θ_0 is center orientation, and σ_θ the width parameter of the orientation.

The filtering operation with a log Gabor filter can be expressed as:

$$B = 2\sqrt{{}^{2}/_{\log(2)}}\left(\left|\left|log\left(\frac{\sigma_f}{f_0}\right)\right|\right|\right)(7)$$

The angular bandwidth is given by:

$$B_\theta = 2\sigma_\theta\sqrt{2log2}(8)$$

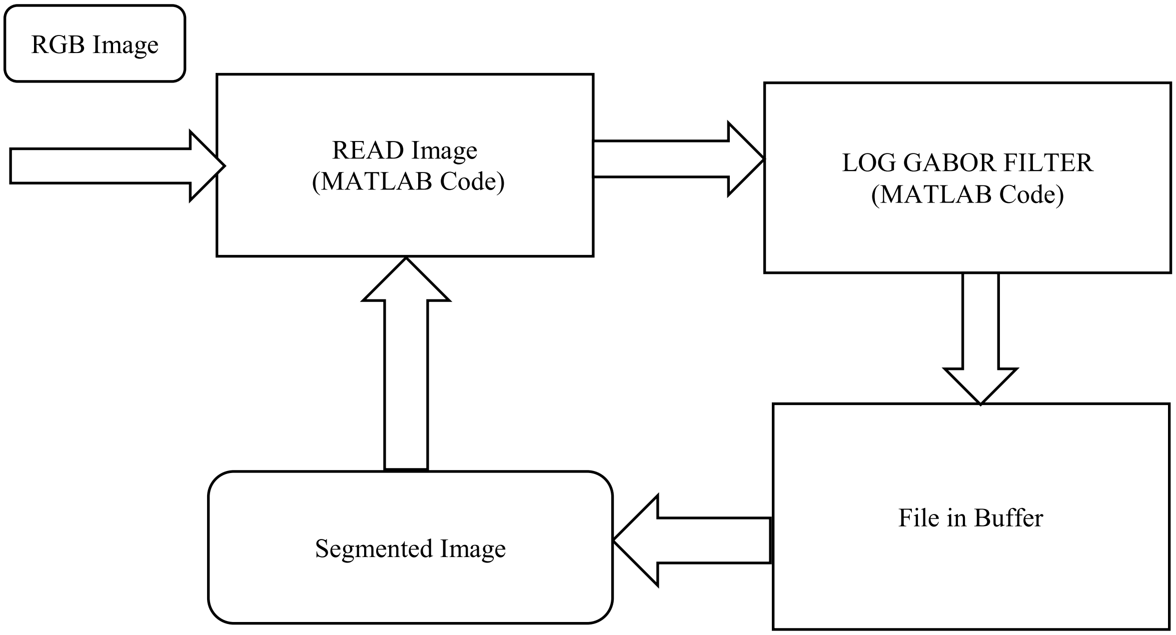


Figure 4. Flow-Chart of 2D LOG Gabor

2.5Singular Value Decomposition

The singular value decomposition (SVD) is a factorization of a matrix (real or complex). SVD [6] decreases a huge dimensional variable data points to a lower dimensional space. Generally original data more generally expose by SVD [30]. The singular values are used as a feature vector of the corresponding Log Gabor filtered image. If we have a rectangular matrix of size $m \times n$ that can be expressed by its SVD is:

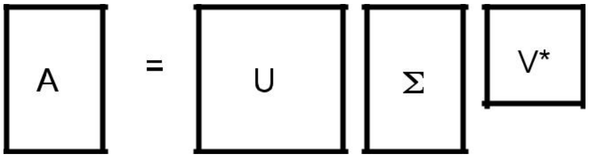
$A_{mn} = U_{mm} S_{mm} V_{nn}^T$ (9)

Where $U^T U=I$, $V^T V=I$.Here we find columns of U are orthonormal eigenvectors of and the columns of V are orthogonal eigenvectors of $A^T A$ and S is a diagonal matrix which actually containing the square roots of eigenvalues, $\sigma_1, \sigma_2, \sigma_3, ... \sigma_m$ from U or V in descending order. The matrix S is represented as:

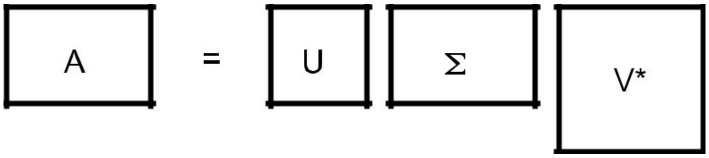
$$S = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & . & 0 & 0 \\ 0 & 0 & 0 & . & 0 \\ 0 & 0 & 0 & 0 & \sigma_m \end{bmatrix}$$

The dimensions from σ_1 to σ_8 are stand for computational complexity of a classifier increases with the increment of feature size.

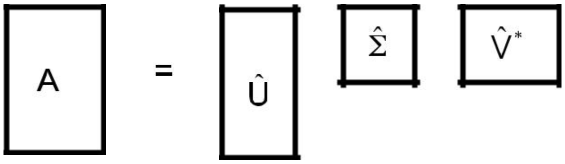
The first form of the singular value decomposition where$m<n$.



The second form of the singular value decomposition where$m \geq n$.



The third form of the singular value decomposition where $r \leq m < n$.



S, U, V Matrixafter Implementing SVD Algorithm

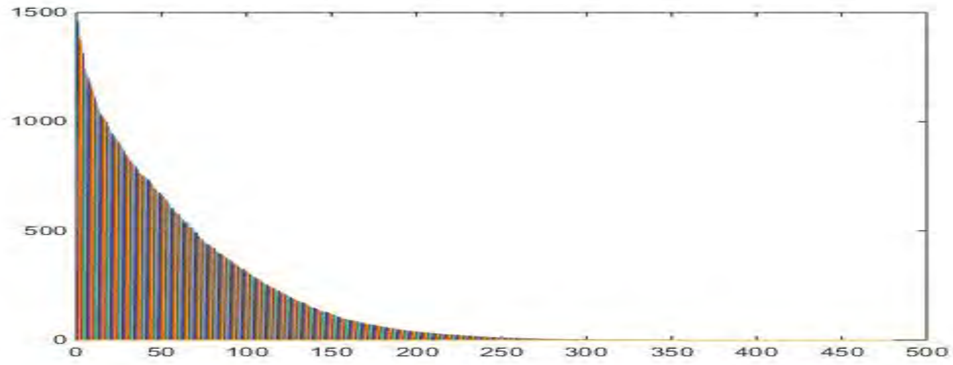


Figure5.S Matrix

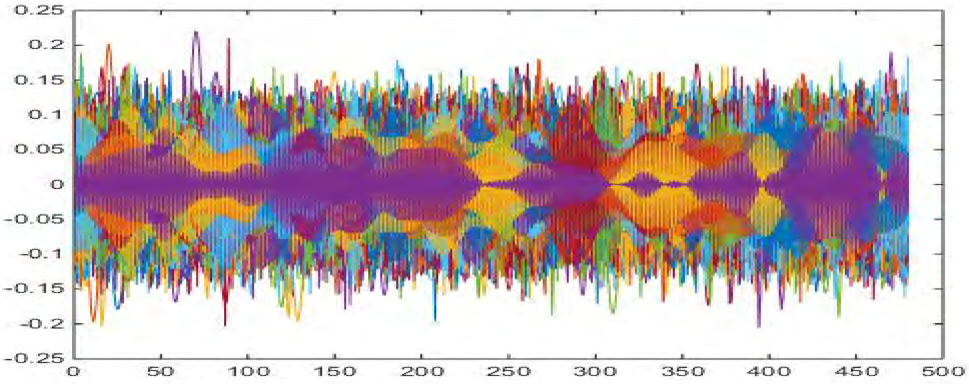


Figure6.U Matrix

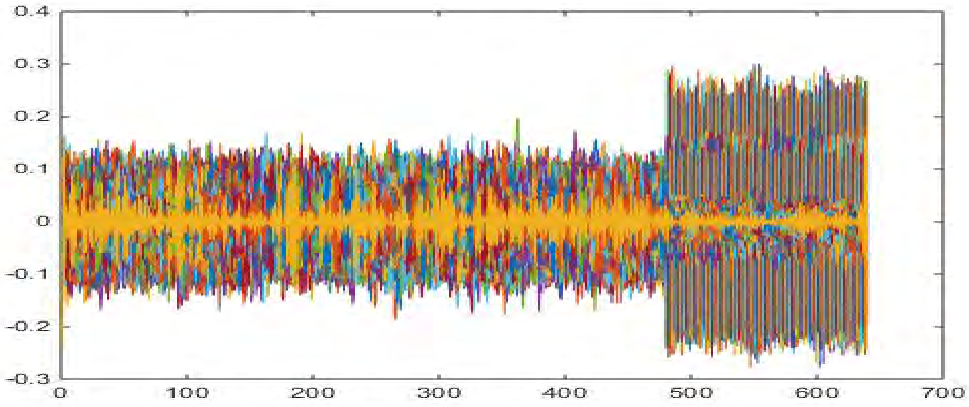


Figure7.V Matrix

2.6Texture Feature

Due to constant improvements on Multimedia and network technology which also made memory much cheaper than before along with making processors much more powerful than ever before, usage of images with large dataset and size has been increasing rapidly. Be it in medical sector or geography or architecture or medicine or media, its being used more than ever before. So, as a result, constant and efficient search of a desired image of a large set of data became much more important now. Which resulted as old technology of text-image retrieval obsolete. Content-Base Image retrieval has been using now a days to meet this demand as it's much more efficient and faster than previous technologies.

CBIR uses image feature to detect/retrieve an image from a large database. Sometime its uses color feature, sometime texture feature or sometimes its uses shape feature it identify an image content. In our proposed model we used texture feature of images to do that.

Texture basically indicates of some sort of repeated pattern over and over again in an image. Sometimes there's maybe some randomness in it. To be more precise, we can say texture feature of an image is a collection of matrix which is calculated during image processing, whose goal is to find out the texture of an image. Its gives us the information about content of an image like intensity or color of particular area of an image.

Texture property [40] of an image includes:

- Coarseness
- Contrast
- Directionality
- Line-likeness

- Regularity
- Roughness
- Entropy
- Energy

Texture feature can be used to classification of images. It also helps with the segmentation of image. But that’s not the focus of our paper. Here, in our proposed model we will use the texture feature of image in classification.

There are two ways to find out texture feature of an image.

1. Structured Approach
2. Statistical Approach

Structured Approach:

It looks for some regular and repeated pattern

Statistical Approach:

It looks for quantitate arrangement of intensities of a region of an image. Much more popular and vastly used approach it is than structured approach.

2.7 Feature vector

A feature vector is a vector that contains some information that describes the important characteristics of an object. In other words, we can say that the numerical values that form an image is actually feature vector of that particular image. In short, it is a vector which contains multiple elements (features). In the context of image processing a feature take many forms and feature representation of an image would be raw intensity value of each pixel. Representation of a feature of an image can be a pixel or the whole object in that image. Color components, length, area, circularity, gradient magnitude, gradient direction or gray level intensity are some examples of feature.

As for example: x=[R, G, B]; is a feature vector. It contains components of a pixel or an object. In a finger print recognition a feature vector is a vector formed from the collection of trivia extracted from the input given.

In an application for object recognition the feature vector of the query object is compared to the data stored in the database to find matches.

For most of the samples it is not sufficient to extract only one type of feature from the image data. We gathered all the information in one single vector which is commonly referred to as a feature vector.

2.8Naive-Bayes classifiers

These classifiers are a family of very simple probabilistic classifiers in machine learning, based on the Bayes theorem. They apply this theorem with strong independence supposition between different features. It is a popular method for text categorization which has been studied since 1950s extensively. It finds application in the field of medical diagnosis. The classifiers which require a number of linear parameters in the number of variables for learning a problem are highly scalable. Simple Bayes and independence Bayes are some other names of this classifier which is found in the field of computer science and statistics.

2.9NVIDIA GPU

An American technology company known as NVIDIA Corporation is the producer of NVIDIA GPU. It is very popular as a graphical chipset. This chipset can either be used as open-source driver or proprietary driver. For some 3D applications and games which are little bit heavy in terms of computation this proprietary driver is a must. The company designs the chips for a variety of customers i.e. for mobile computation in the form of system on chip units, the gaming market and automotive market as well. GeForce is one of the productions of NVIDIA. This GPU is in a competition directly with Radeon which is an Advanced Micro Devices (AMD) product. Professional visualization, Data centers, gaming and Auto are the four markets where NVIDIA is focusing from 2014.

The researchers and scientists are given the capability of parallel processing by NVIDIA. Thus the applications which need high performance can be run efficiently using NVIDIA GPUs. This NVIDIA GPU power millions of devices like desktops, notebooks, workstations and supercomputer all over the world.

There are some fields which describes about a GPU. They are:

- Model
- Launch
- Codename
- Fabrication process
- Bus interface
- Memory
- SM count
- Core clock

- Memory clock
- Core configuration
- Fillrate
- Memory subsection

  -Bandwidth

  -Bus type

  -Bus width
- API support section

  -Direct 3D

  -OpenGL

The researchers and scientists are given the capability of parallel processing by NVIDIA. Thus the applications which need high performance can be run efficiently using NVIDIA GPUs. We use NVIDIA GPU for parallel computing. We use four different resolution of each type images. We use 640*480, 480*320, 320*240 and 240*180 resolutions of each image. Then we parallel implement those resolutions of each image with our CPU implemented part. And a great amount of time cost for every single picture. We use NVIDIA GEFORCE GTX780 GPU card for this part.

2.9.1 CUDA

CUDA is a parallel computing platform developed by NVIDIA Corporation which is an acronym for Compute Unified Device Architecture. It was first released eight years back in 2007. It is a freeware product which can be run on the popular operating systems i.e. windows

XP and later versions, Mac OS X, Linux. It is a standard feature in all GPUs i.e. NVIDIA GeForce, Quadro, Tesla GPUs and NVIDIA GRID solutions as well.

Basically it is a platform for computation and parallel programming model which increases performance dramatically with the power of GPU. It has been widely used through uncountable applications and published researches. The different applications that are used in the fields of astronomy, biology, chemistry, physics, data mining, finance and other heavy fields are doing increased use of CUDA for delivering benefits of GPU acceleration.

CUDA allows doing GPGPU commonly known as general purpose processing in GPUs. Software developers can perform tasks in CUDA enabled graphics processing unit (GPU) that allows the developers to have direct access to the GPUs parallel computational elements and instruction sets as well.

NVIDIA has brought world's first teraflop many core processor called Tesla. It enables energy efficient parallel computing power. Tesla solves most important computing challenges of present times in a quick and accurate fashion with the help of thousands of CUDA cores per processor.

Programming languages such as C, C++ and FORTRAN can commonly be used in CUDA platform. This kind of accessibility gives the scope of utilizing the resources of GPU to the parallel programmers whereas previous APIs like Direct3D and OpenGL needed to have advanced skills in graphic programming. Programming frameworks known as OpenCL and OpenACC are also supported by CUDA. CUDA offers both high and low level APIs.

Codes written in C or FORTRAN can be run in parallel on multi CPU and GPU accelerated systems. OpenACC gives this kind of flexibility. It is an open industry for compiling where codes can be inserted and compiler codes can be generated.

There are several advantages of CUDA over traditional general purpose computation on GPUs using graphics API. They are:

- Scattered reads: it allows reading from arbitrary addresses in memory.
- Unified virtual memory
- Unified memory
- Shared memory
- Faster downloads and readbacks
- Integer and bitwise operations supported which includes integer texture lookups too
- Multiple graphics card in single system: the several applications can be distributed to the other GPUs available. It is not automatic. The complete control is given to the application.

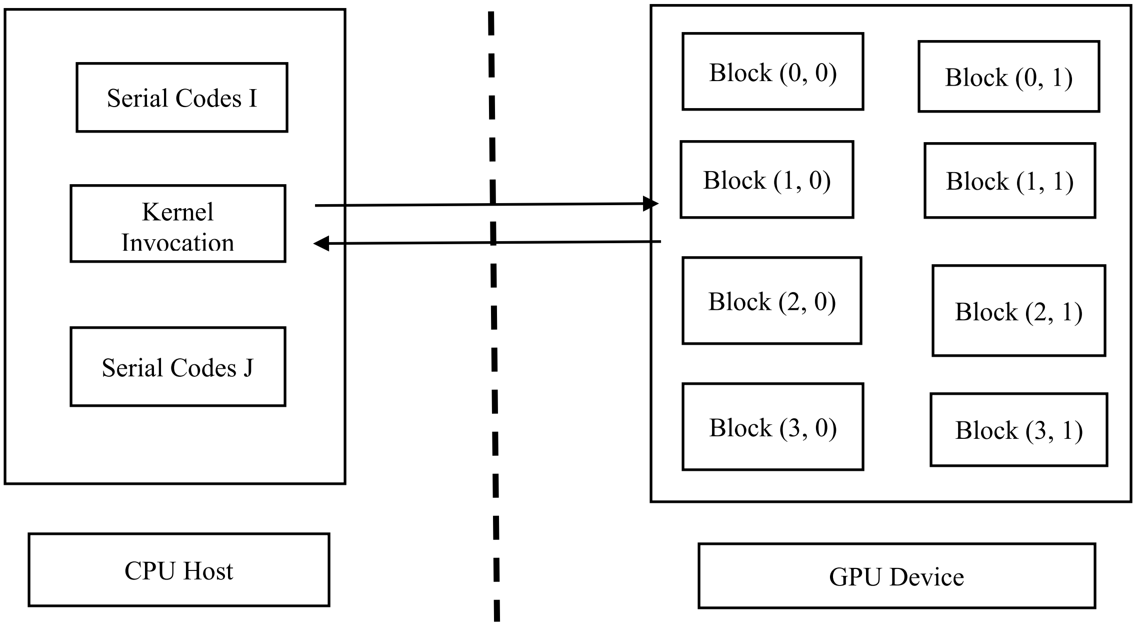


Figure8. Parallel Computing between CPU Host and GPU Device

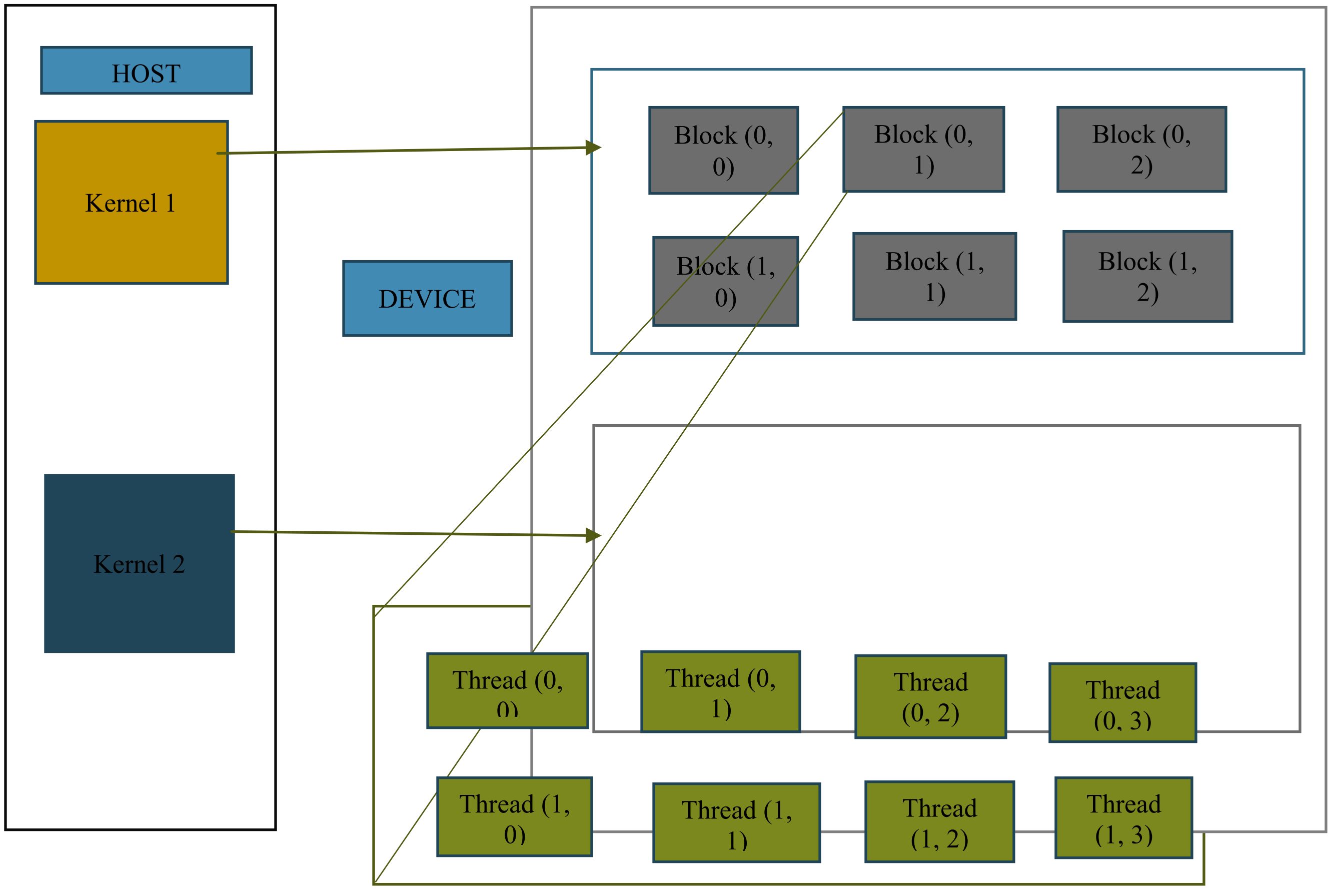


Figure9. NVIDIA CUDA Architecture

3. PROPOSED MODEL

3.1 Workflow

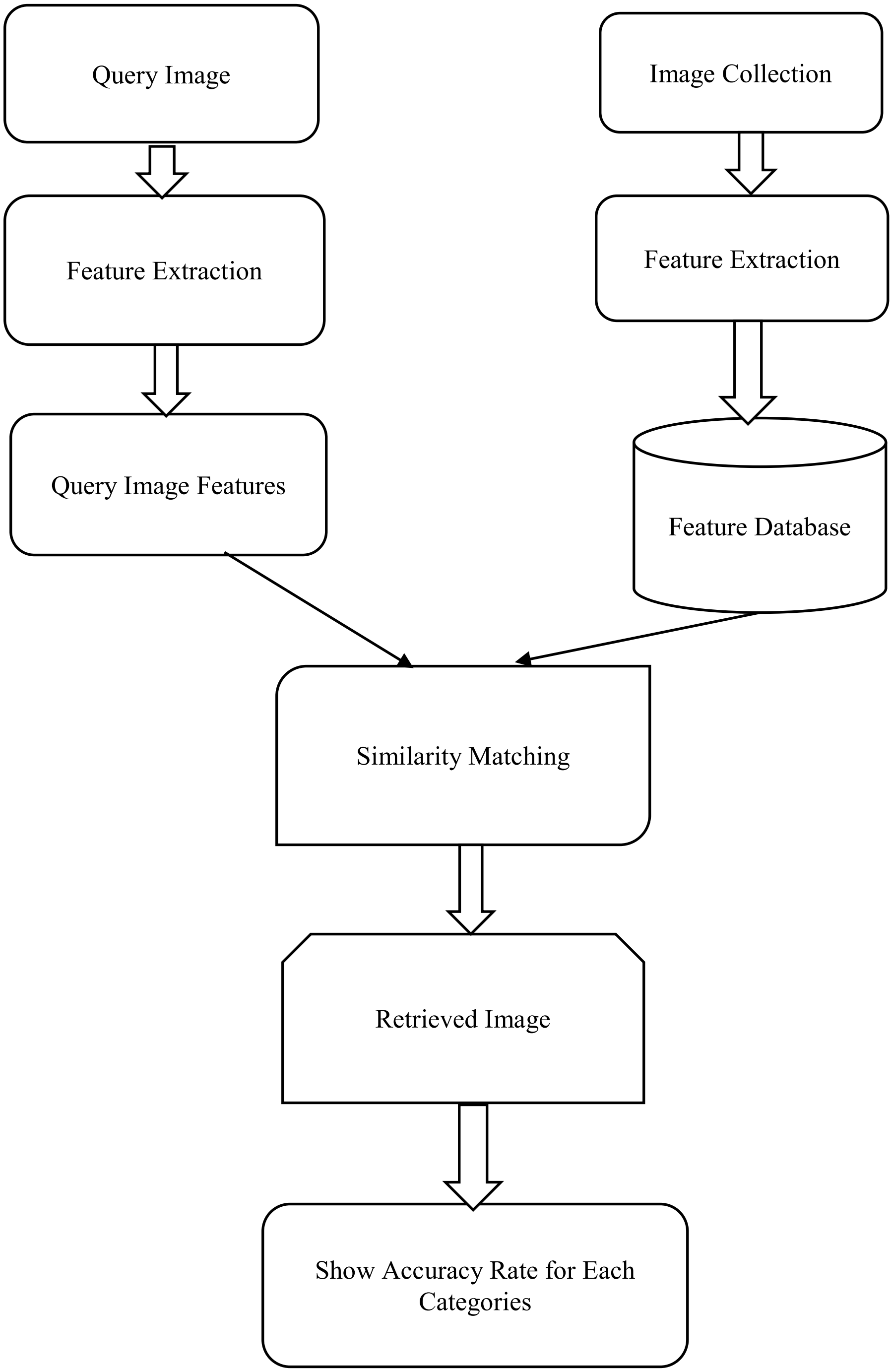


Figure 10. Basic Block Diagram of Our Thesis

In Figure 10 we show the basic work flow part. We actually trying to describe what are we really want to do. We 1st take 7 categories image collection. Then we done feature extraction on those images. After that, we keep those feature extracted images on feature database. On the other side, we several query images. Then done feature extraction of those images. After that we match the query images feature extracted sample with our feature database sample. And lastly, we try to find how accurately this system define the each of the query images categories.

3.2 CPU Implemented Work Flow

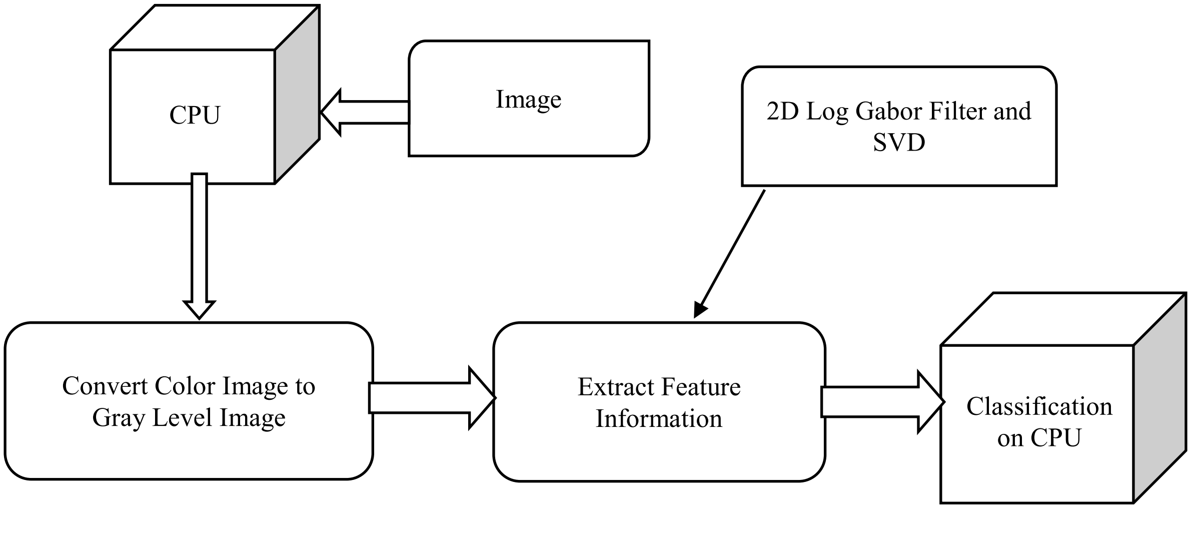

CPU
Image
2D Log Gabor Filter and SVD
Convert Color Image to Gray Level Image
Extract Feature Information
Classification on CPU

Figure 11. CPU Implemented Version

In Figure 11 is CPU implemented version. Here we first take images. After that we see image are in gray level or not. If they are not in gray level we convert this image to gray level.

Then we use 2D Log Gabor filter and SVD based model to extract feature information. Finally, The classification of image shown on CPU.

3.3 CPU and GPU Implemented Processing

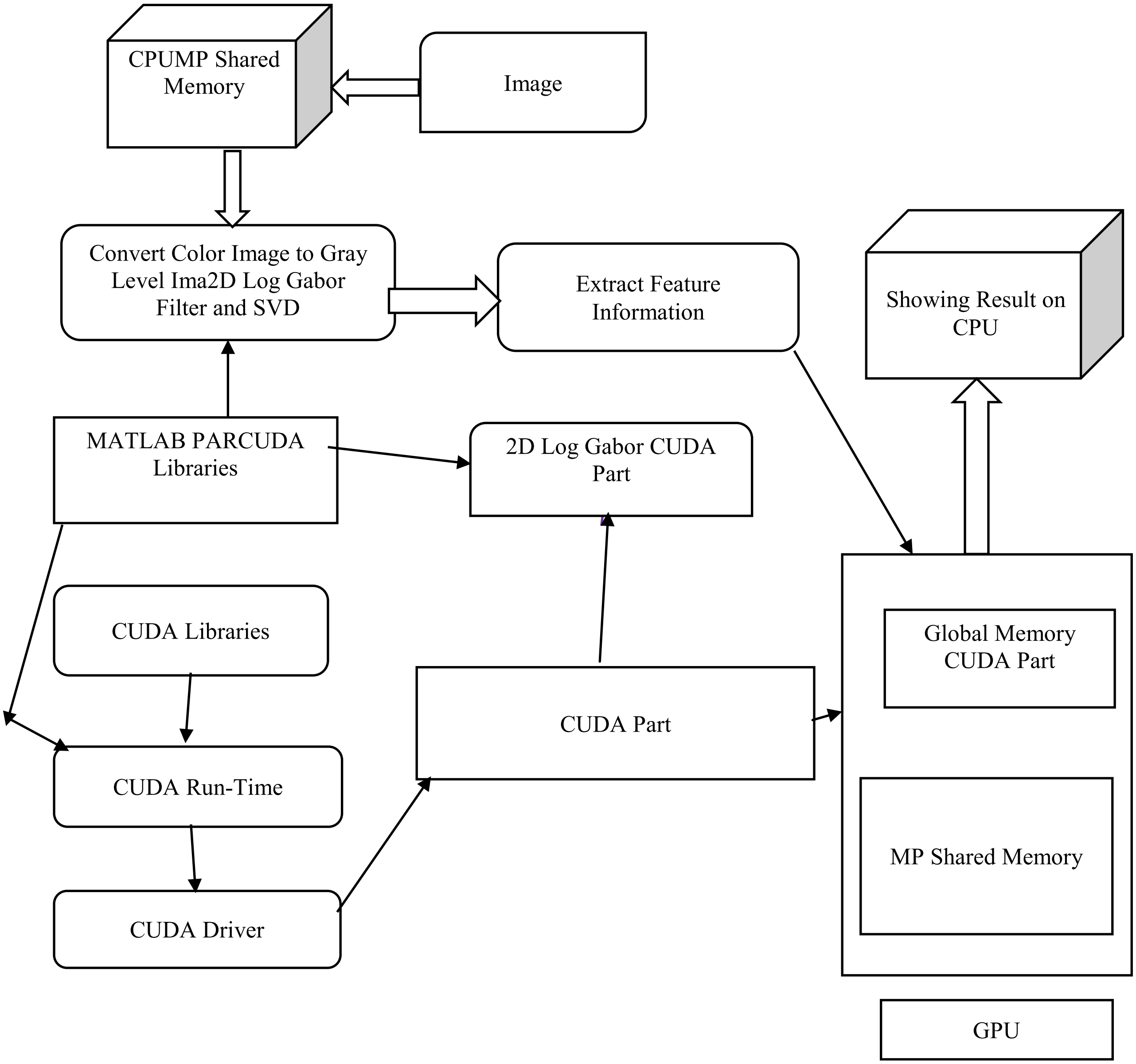


Figure 12. Parallel Implemented Version

In Figure 12we parallel implement our model in CPU and GPU. It will cut our run time cost remarkably. We showed that GPU version is 56% faster than CPU implemented version.

4. EXPERIMENTAL SETUP

Our implementation running on Core i7 with 3.40 GHZ, 16 GB RAM with NVIDIA GEFORCE GTX780 GPU card. We use MATLAB 2014b for our coding part.

4.1 Core i7 3.40 GHZ

It is one of the highest selling processor of Intel. Number of cores, memory cache and processor clock rate actually faster your image processing run time. This processor 4 cores (2 logical cores per physical), Clock-Speed 3.4 GHZ and 8M memory cache.

4.2 MATLAB

MATLAB is numerical computing environment and fort generation programming language. Its language developed by MATHWORKS, it allows matrix manipulations, plotting offunctions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MUPAD symbolic engine, allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

It also has easy to use graphics commands that make the visualizations of results immediately available. These factors make MATLAB and excellent tool for teaching and research [38].

We used MATLAB 2014b for our coding. It has some great features for GPU implementation. Like, it has parallel computing toolbox. Again it has also some great features including,

- Packaging of custom MATLAB toolboxes into a single, installable file
- Preference for controlling the initial working folder, with the option to start in the folder from your previous MATLAB session
- Workflow improvements when editing classdef files, including immediate impact on existing and new workspace variables
- matlab.wsdl.createWSDLClient function for accessing SOAP-based Web services
- cummin and cummax functions for computing cumulative minimum and maximum of an array
- Reverse accumulation option for the cumsum, cummin, cummax, and cumprod functions
- Median and mode calculations of categorical data
- New look of MATLAB graphics with improved clarity and aesthetics
- Improved infrastructure based on MATLAB objects
- Rotatable axis tick labels
- histogram function for plotting histograms

4.3NVIDIA GEFORCE GTX780 GPU Card

We used NVIDIA GEFORCE GTX780 GPU card for our parallel processing. We actually use these GPU card to reduce our time cost or run cost for each and every images. We take 7 categories images with 4 different resolution and implemented them with our proposed model. We collect some features of this GPU card. In bellow we show those features:

Table 1. GTX 780 GPU Engine Specifications

CUDA Cores	2304
Base Clock (MHz)	863
Boost Clock (MHz)	900
Texture fill rate (billion/sec)	160.5

Table 2.GTX 780 Memory Specifications

Memory speed	6.0 Gbps
Standard memory config	3072 MB
Memory interface	GDDR5
Memory interface width	384-bit
Memory bandwidth (GB/sec)	288.4

Table 3.GTX 780 Support

Important technologies	GPU Boost 2.0, PhysX, TXAA, NVIDIA G-SYNC-ready, SHIELD-ready
Other supported technologies	3D vision, CUDA, Adaptive VSync, FXAA, NVIDIA Surround, SLI-ready
OpenGL	4.3
Bus support	PCI express 3.0
Certified for windows 7, 8, vista, XP	Yes
3D vision ready	Yes
3D gaming	Yes
Blu ray 3D	Yes
3D vision live (photos and videos)	Yes
NVIDIA PhysX technology	Yes
Microsoft directX	12 API

Table 4. Display Support

Maximum digital resolution	4096 x 2160
Maximum VGA resolution	2048 x 1536
Standard Display Connectors	One dual link DVI-I, one dual link DVI-D, one HDMI, one display port
Multi monitors	4 displays
HDCP	Yes
HDMI	Yes
Audio input for HDMI	Internal

Table 5.GTX 780 Graphics Card Dimensions

Height	4.376 inches
Length	10.5 inches
Width	Dual-slot

Table 6. Thermal and power specifications

Maximum GPU temperature (in C)	95 C
Graphics card power (W)	250 W
Minimum recommended system power (W)	600 W
Supplementary power connectors	One 8 pin and one 6 pin

5. EXPERIMENTAL RESULTS AND ANALYSIS

In this chapter, we will show our experimental result and will provide enough description to make it easier to understand.

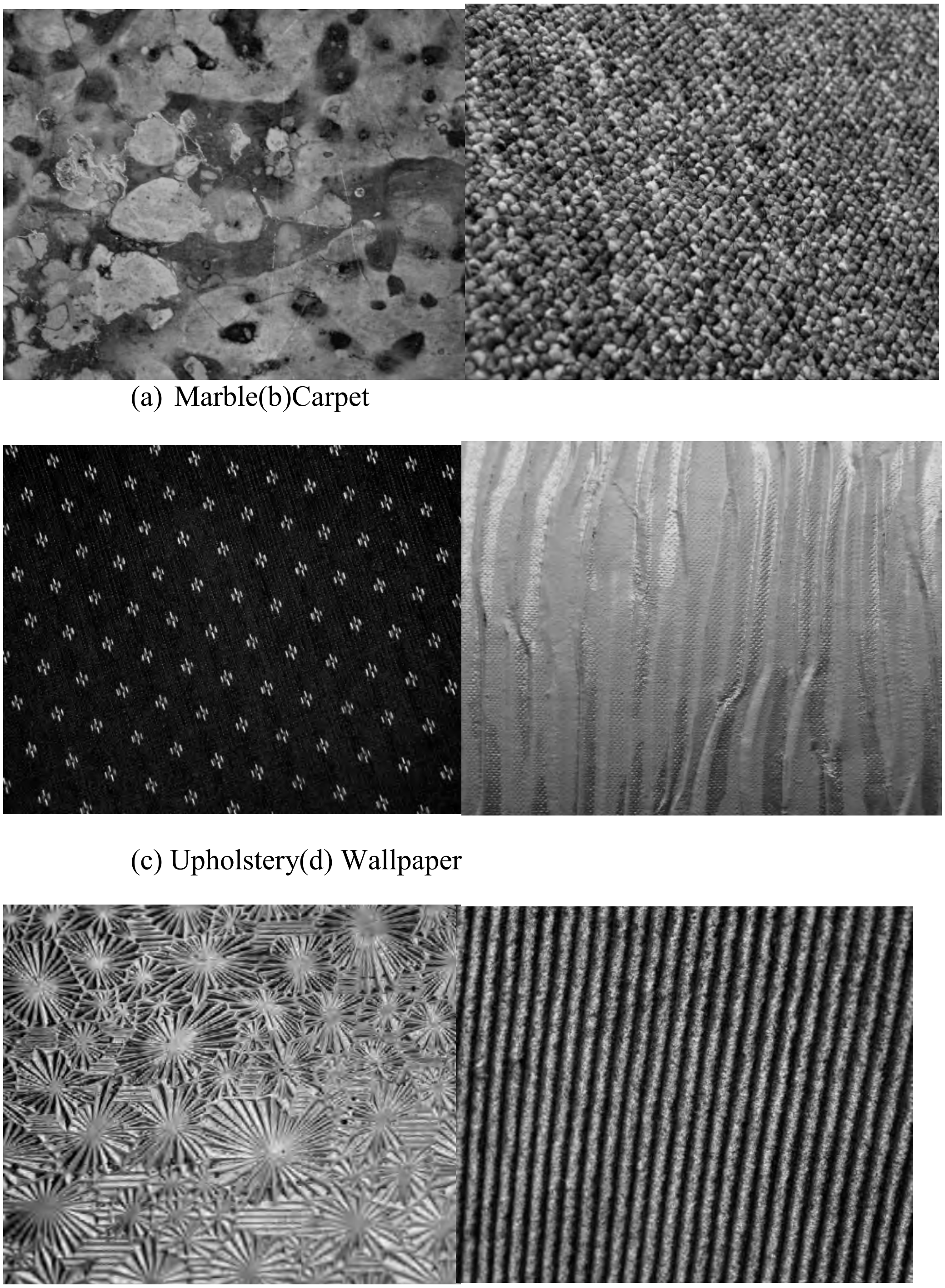
5.1 Experimental environment

On the first part of our experiment, we implemented our proposed model on CPU. Which consists of Intel Corei7 and 3.40 GHz processor. Later part of our experiment we bought GPU on the equation. We used NVDIA GPU card for this part our experiment. Which to be precise is NVIDIA GPU GeForce GTX780. Rest of the Configuration remains the same.

5.2 CPU Implementation

During this part, we processed about 1200 image on CPU to

- Firstly implement Log Gabor Filter
- To figure out the Singular matrix of the image after implanting SVD algorithm on the result of Log Gabor Filtered Image
- Lastly, we used to Naïve Bayes Classifier to find out efficiency of our model. We used around 40% image of each category to test. Then, we used all of the image of each category to test to see how many of them our system can correctly guess which category they belongs to.



(a) Marble(b)Carpet

(c) Upholstery(d) Wallpaper

(e) Glass

(f) Corduroy

Figure 13. Six types of images that was used in Experimentation

Below, in the table we are showing result we got, while implementing on CPU

Table 7. Accuracy Rate of Our proposed Model on Testing Different Category Images

Category	**Total Image**	**Percentage of Image Used in training**	**Accuracy Rate**
Carpet	200	40%	93.5 %
Upholstery	200	40%	97.5%
Marble	200	40%	97.5%
Wallpaper	200	40%	91.5%
Glass	200	40%	95%
Corduroy	200	40%	85.5%

5.3 GPU Implementation

During Second part which is also eventually the last part of our experiment, we implemented the very same model on GPU. We parallel processed our model on GPU to reduce the time complexity as well cycle number of the algorithms. What we did was

- Firstly, calculated the time that was needed to process every image in CPU.
- Then we calculated the average time of every category that was need to run it in CPU
- Then we did the same thing while running our model on GPU which finding time need for each category to run in GPU
- To make things more interesting, we also tried to compare with both implementation of our model with different size/resolution of image.

On every steps on GPU, the improvement on time complexity and cycle reduction over CPU was significant. It took lot less time and cycle to implement on GPU. Below on different table we will try to show that.

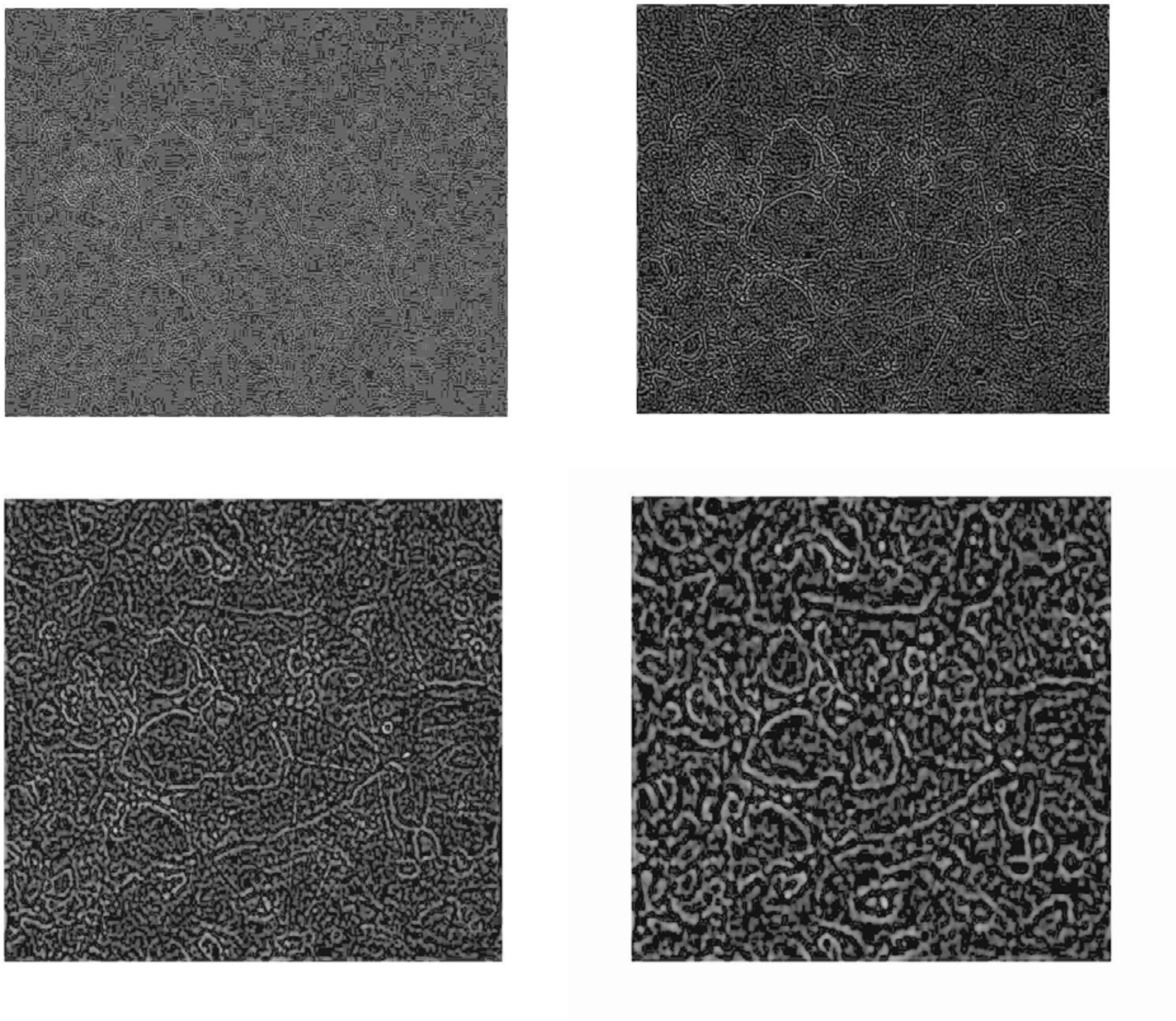


Figure 14. Output of a processed image for Marble after implementing Log Gabor Filter

Table 8. CPU, GPU Speed up Difference on marble category

Image Resolution	CPU (milliseconds)	GPU(milliseconds)	Speedup (per image)
640 X 480	315.663417	161.240776	1.95771
480X 360	294.648129	131.195821	2.24586
320X240	168.470668	90.165657	1.86845
240X180	157.740241	83.758976	1.88326
Total Speedup:	2.00814		

Average speed up for per Marble image: 1.9856

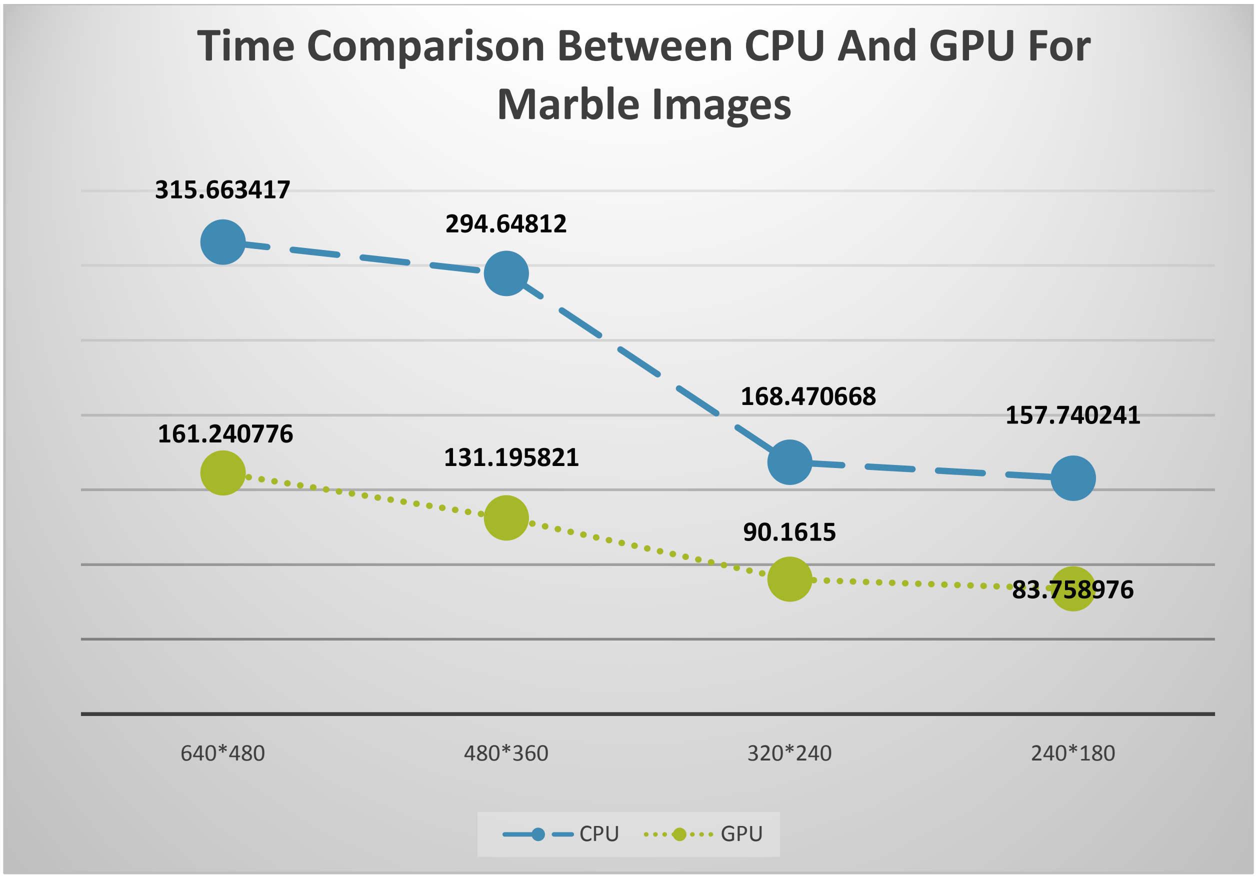
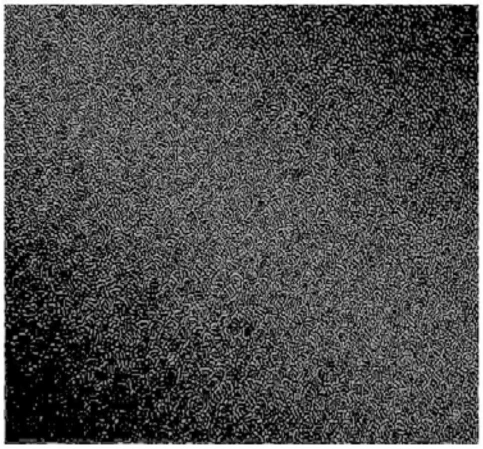
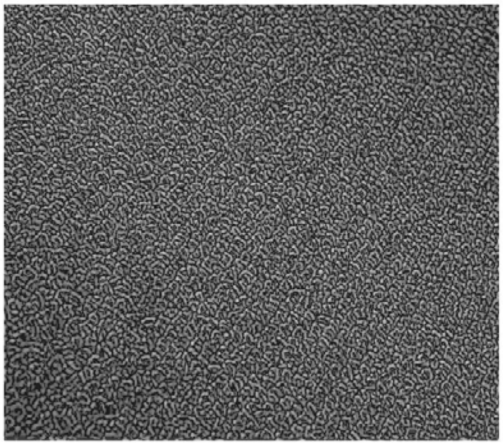


Figure 15. Comparison between CPU and GPU Implementation of Marble Image

Secondly, we show same GPU and CPU implemented comparison for carpet in the following table.





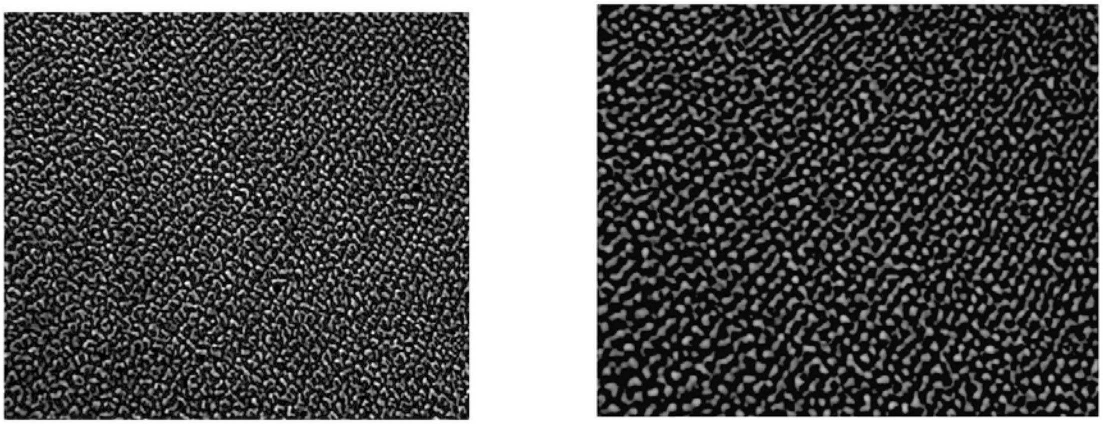


Figure 16. Output of a processed image for Carpet after implementing Log Gabor Filter

Table 9. CPU, GPU Speed up Difference on Carpet category

Image Resolution	CPU (milliseconds)	GPU(milliseconds)	Speedup (per image)
640 X 480	325.924430	162.635436	2.00401
480X 360	298.564570	132.258231	2.25743
320X240	170.568921	92.564781	1.84269
240X180	158.452568	84.214579	1.884576
Total Speedup:	2.021507		

Average speed up for per Carpet image: 1.99717

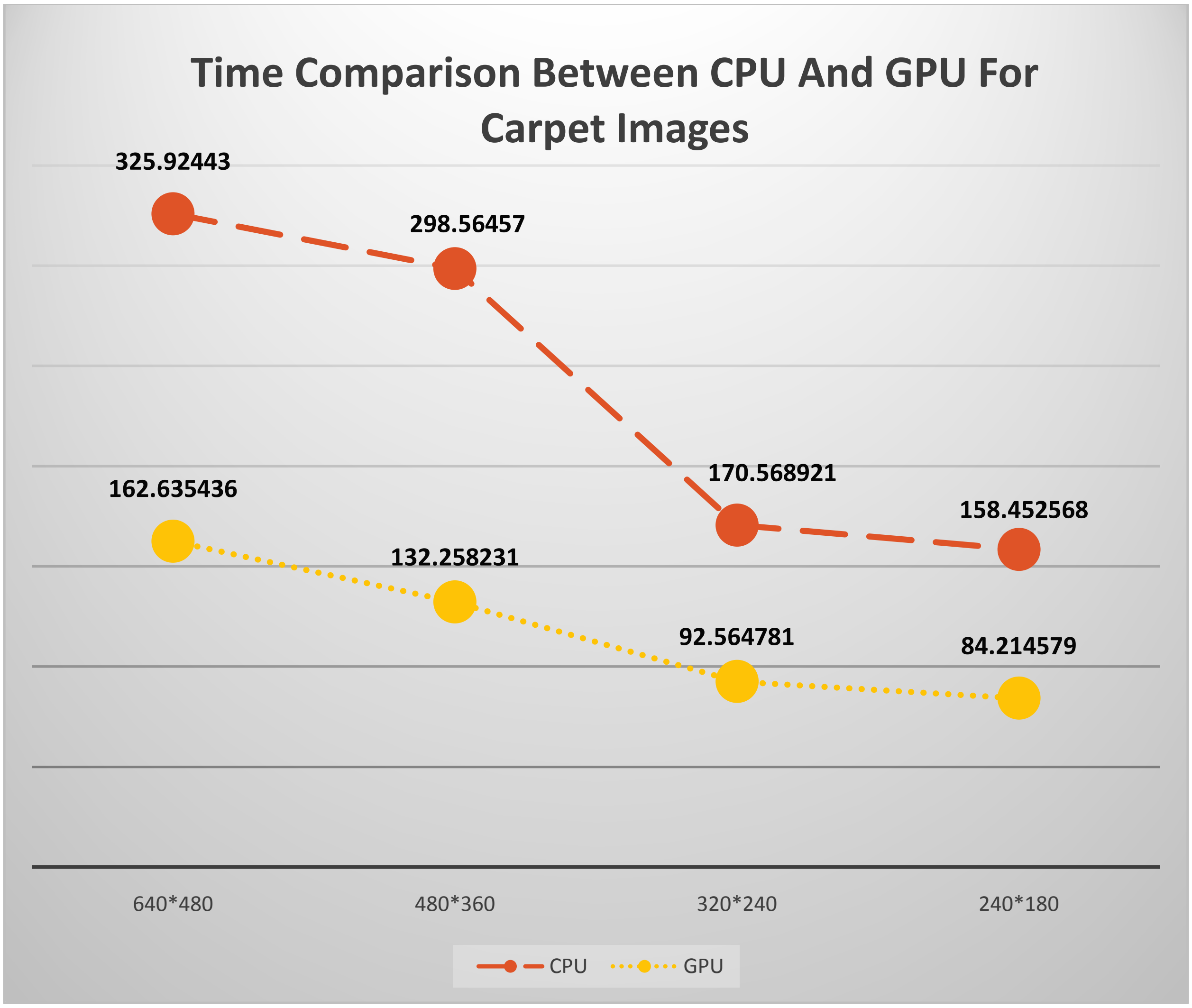


Figure 17.Comparison between CPU and GPU Implementation of Carpet Image

Thirdly, comparison for upholstery in between our CPU and GPU implemented method.

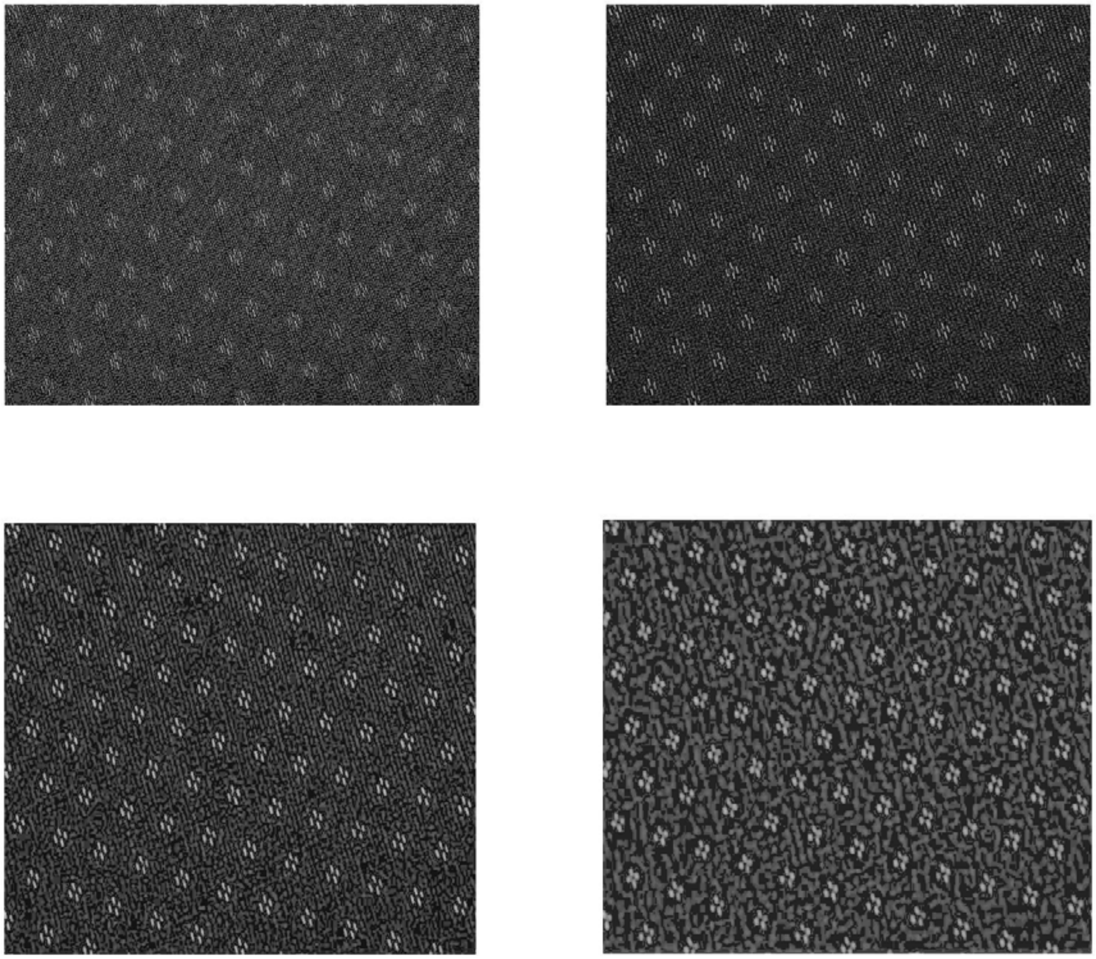


Figure 18. Output of a processed image for Upholstery after implementing Log Gabor Filter

Table 10. CPU, GPU Speed up Difference on upholstery category

Image Resolution	CPU (milliseconds)	GPU(milliseconds)	Speedup (per image)
640 X 480	383.602666	194.552314	1.97171
480X 360	315.258469	148.35689	2.12896
320X240	181.478963	99.546846	1.82056
240X180	165.213954	88.341028	1.87246
Total Speedup:	1.96778		

Average speed up for per upholstery image: 1.94842

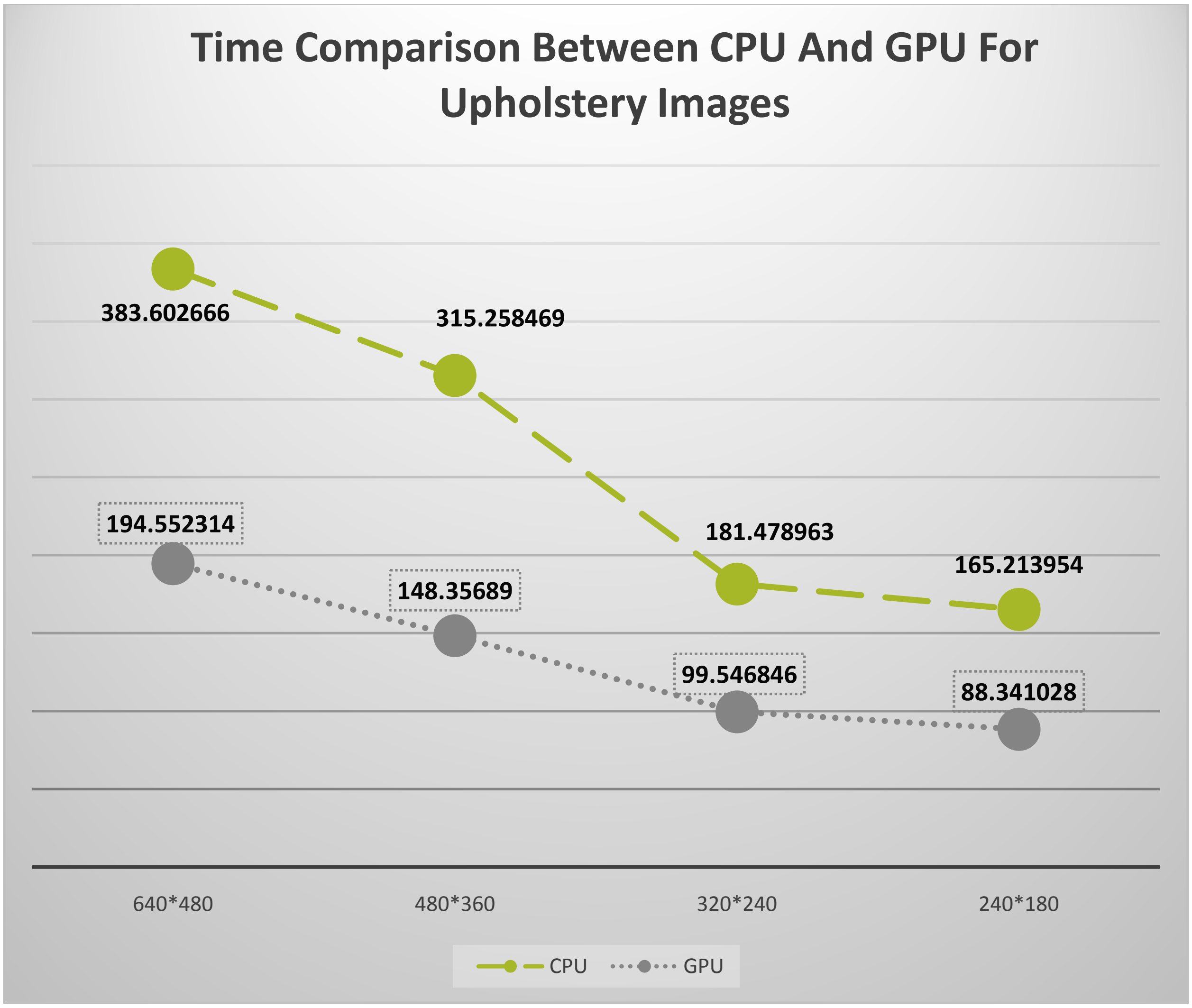


Figure 19.Comparison between CPU and GPU Implementation of Upholstery Image

Fourthly for wallpaper we did the same things.

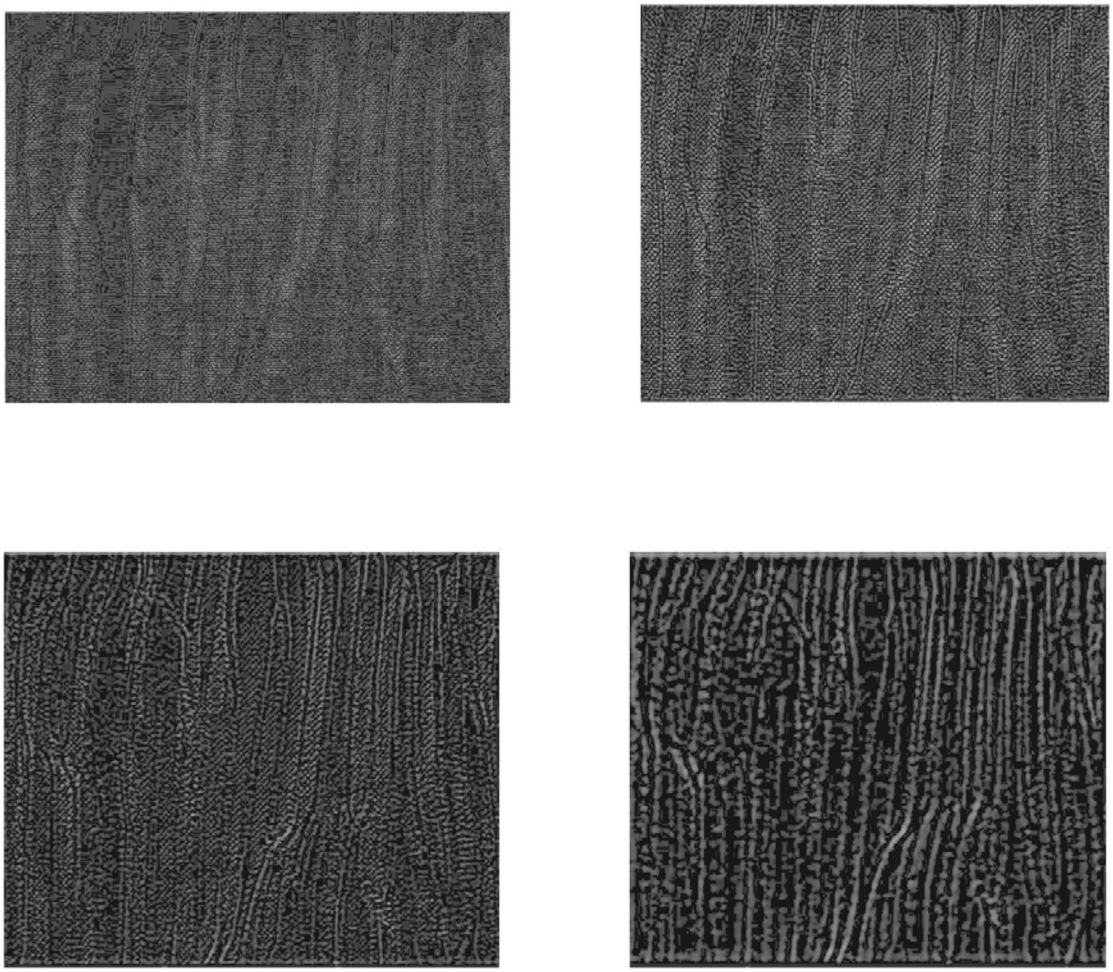


Figure 20.Output of a processed image for Wallpaper after implementing Log Gabor Filter

Table 11. CPU, GPU Speed up Difference on Wallpaper category

Image Resolution	CPU (milliseconds)	GPU(milliseconds)	Speedup (per image)
640 X 480	392.248952	166.255212	2.35934
480X 360	318.54694	149.47896	2.13104
320X240	183.45644	100.45789	1.82620
240X180	167.413236	90.369784	1.85253
Total Speedup:		2.095945	

Average speed up for per Wallpaper image: 2.04225

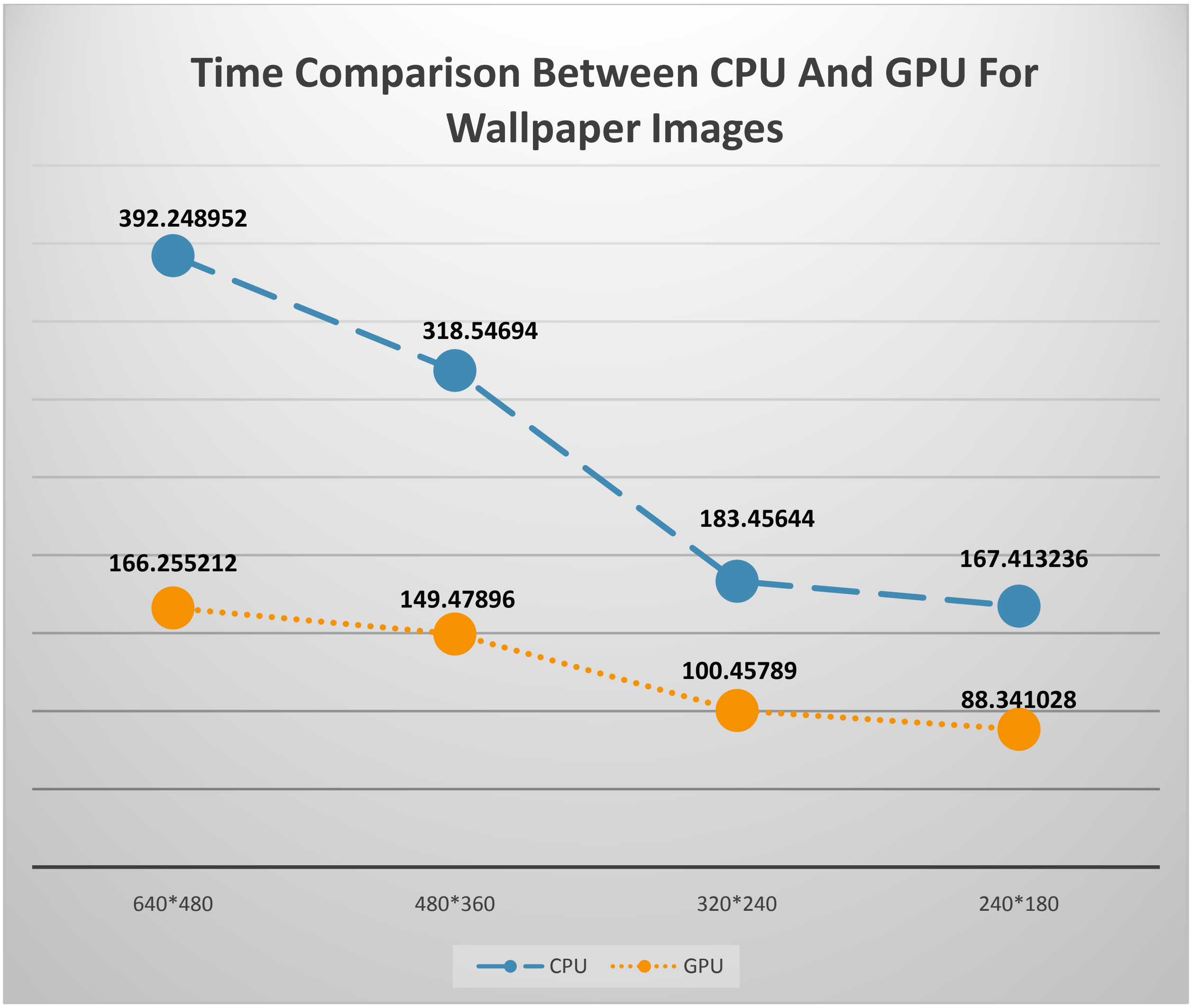


Figure 21. Comparison between CPU and GPU Implementation of Wallpaper Image

Then we did it for Glass category

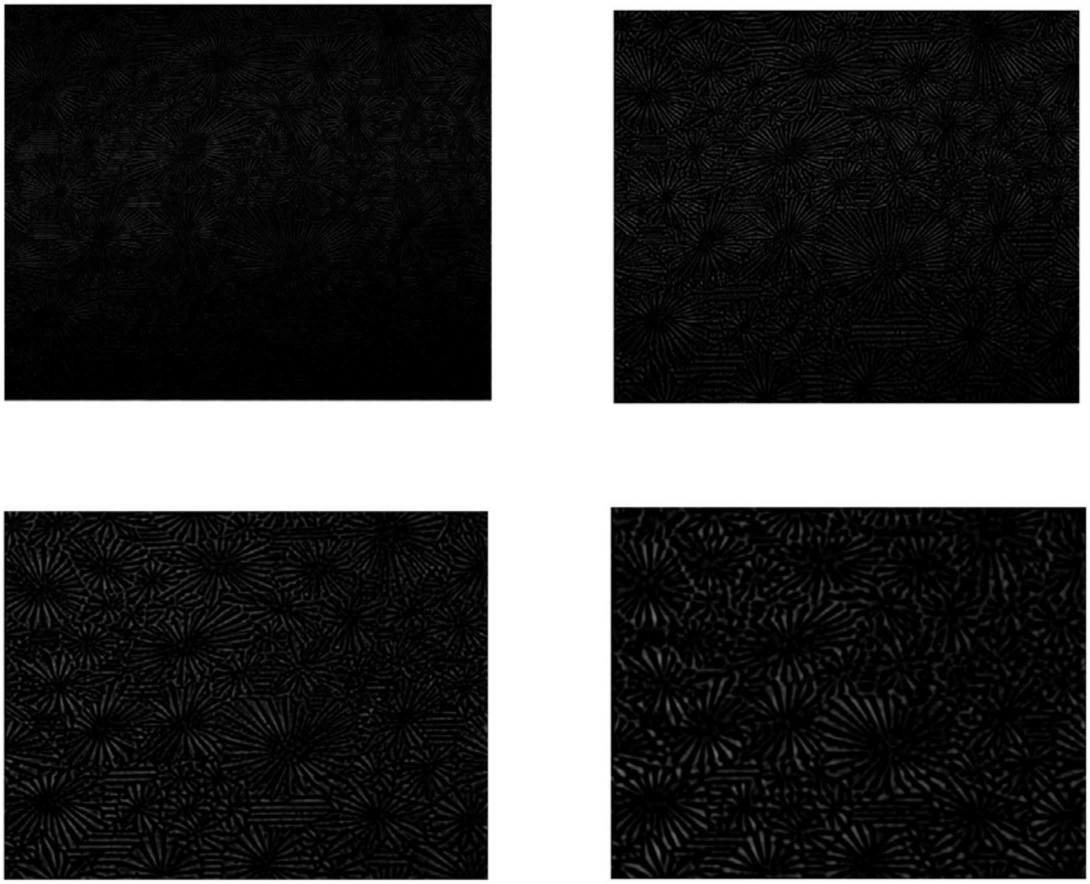


Figure 22.Output of a processed image for Glass after implementing Log Gabor Filter

Table 12. CPU, GPU Speed up Difference on Glass category

Image Resolution	CPU (milliseconds)	GPU(milliseconds)	Speedup (per image)
640 X 480	329.095728	182.592681	1.8023
480X 360	269.751235	156.390928	1.7248
320X240	153.787869	107.222908	1.4342
240X180	144.762140	71.179763	2.0337
Total Speedup:		1.7344	

Average speed up for per Glass image: 1.7344

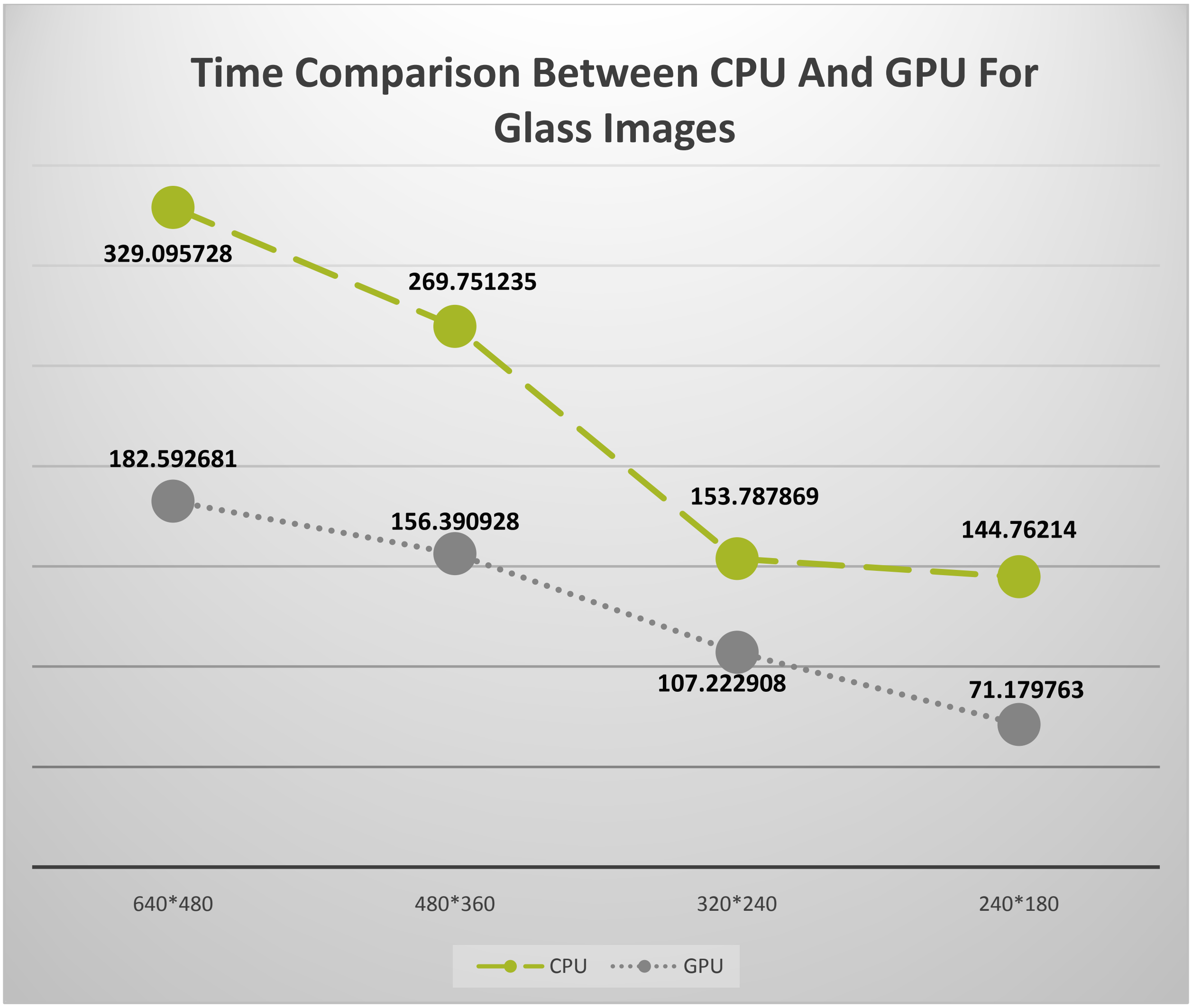


Figure 23. Comparison between CPU and GPU Implementation of Glass Image

Lastly, we repeated the same process with corduroy image

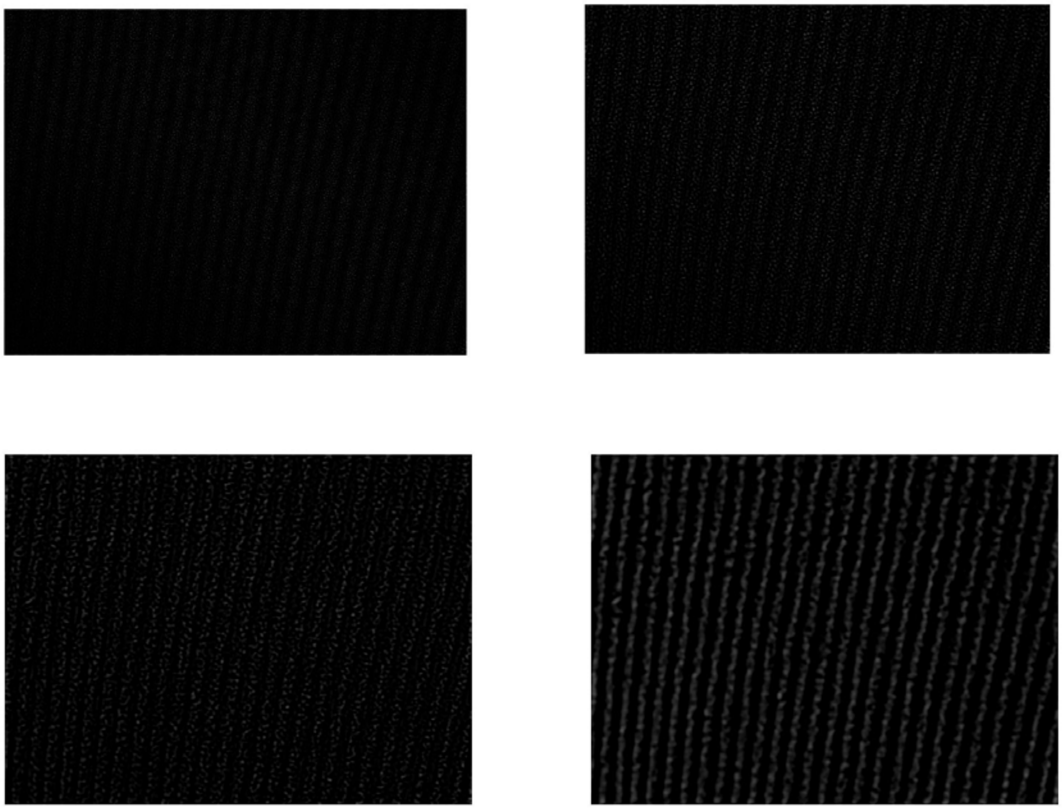


Figure 24.Output of a processed image for corduroy after implementing Log Gabor Filter

Table 13. CPU, GPU Speed up Difference on Corduroy category

Image Resolution	CPU (milliseconds)	GPU(milliseconds)	Speedup (per image)
640 X 480	370.238731	168.367475	2.19889
480X 360	255.641345	155.761823	1.641232
320X240	205.829729	122.102576	1.685711
240X180	123.949743	69.579529	1.781411
Total Speedup:	1.85274		

Average speed up for per corduroy image: 1.85274

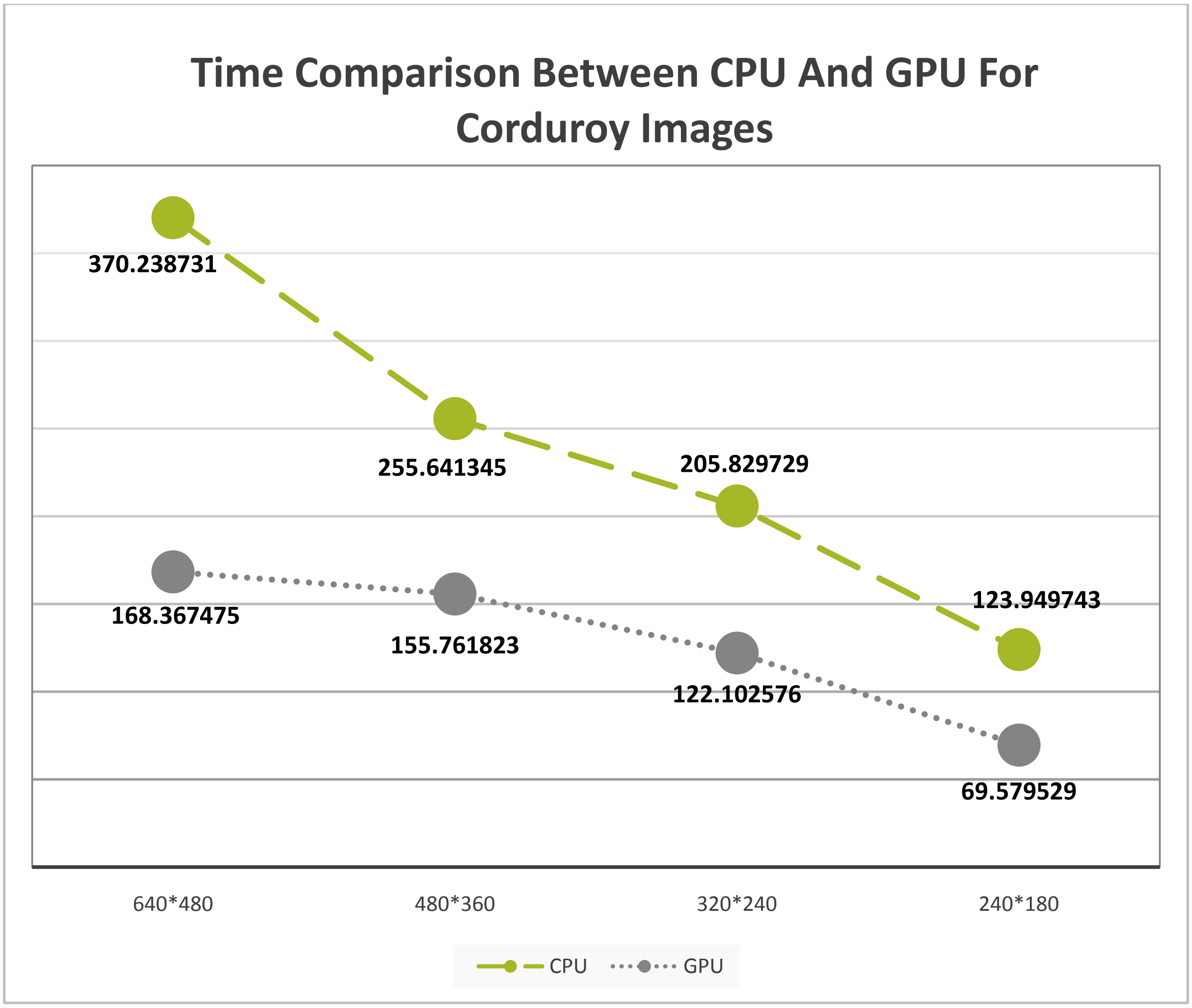


Figure 25.Comparison between CPU and GPU Implementation of Corduroy Image

5.4 Comparison

Now we will compare our proposed model to another similar model. We test and trained with the same images that we used in our experiment with Gabor filter and SVD algorithm. We followed the same procedure that we used in CPU implementation of our model except a slight difference. Which is, instead of Log Gabor Filter we used here Gabor Filter. The rest remained the same.

The difference is shown below in the table.

Table 14. Accuracy Rate of Gabor Filter and SVD based Model on Testing Different Category Images

Category	Total Image	Percentage Of Image Used in training	Accuracy Rate
Carpet	**200**	**40%**	**85.5 %**
Upholstery	**200**	**40%**	**83.0%**
Marble	**200**	**40%**	**90.0%**
Wallpaper	**200**	**40%**	**82.0%**
Glass	**200**	**40%**	**89.25%**
Corduroy	**200**	**40%**	**81.65%**

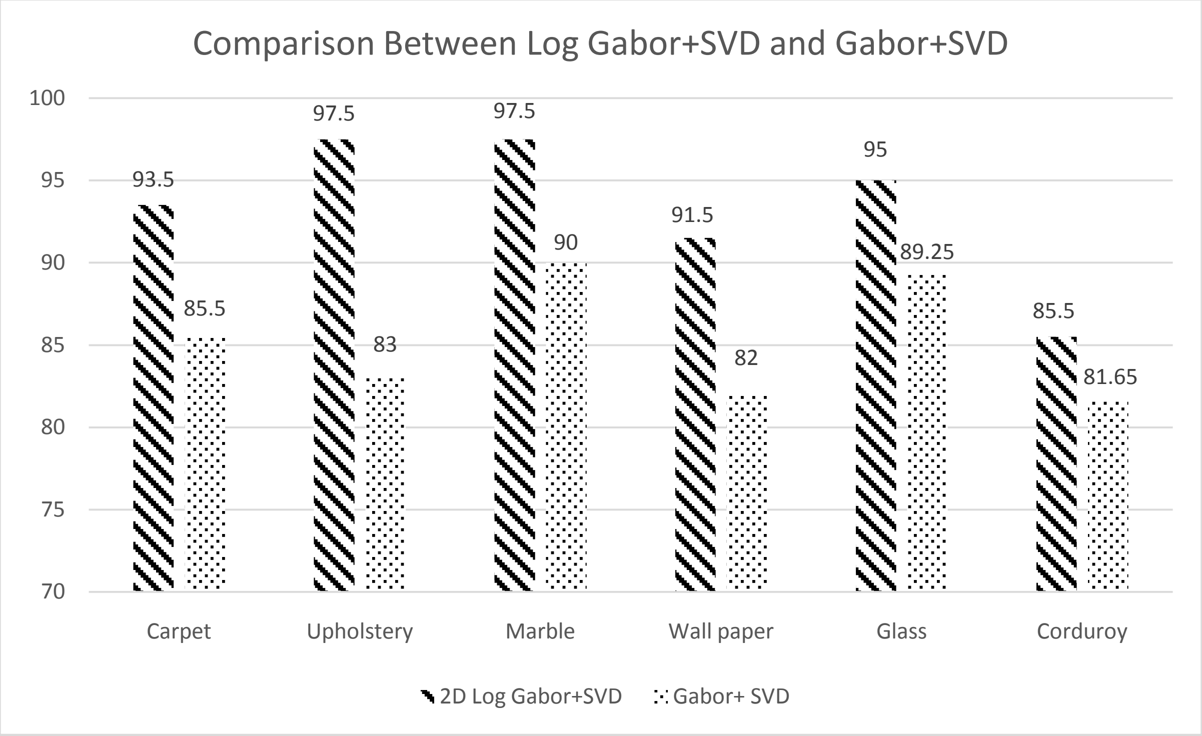


Figure 26. Comparison between Log Gabor, SVD and Gabor, SVD model

Now, we will show some difference between some other model which used some of the very category of image that we used in our experimentation here.

Firstly, we will talk about for marble type of images. Support vector machines (SVMs) this method originally invented by Vapnik [25]. In [26, 27] they used SVM Classifier for texture analysis. We find from [24] with this classifier they correctly guessed 96.01% using marble type of images. Also in [24] we find ANN Classifier produce 94.44% correct result and MMD Classifier will produce 83.61% correct result.From [28] we find K-means algorithm produce correctly solve 74% from total and PAM (Partitioning around Medoids) algorithm show 75% average correct result.From[28] we find K-means algorithm produce correctly solve 74% from total and PAM (Partitioning around Medoids) algorithm show 75% average correct answer.

But in our model, we showed for marbles our system accurately predicted 97.5% from all images. Secondly, for carpet wear we compare our system with existing system. In [29] we see that they use on Support Vector Machine Pattern Recognition Approach for Carpet Wear Classification. They show that SVM classifier is better than Polynomial Kernel approach because SVM show 92% correct labeling but in same place Polynomial Kernel approach only show 89.56% correct answer. But in this paper we show that in practically our proposed system show 93.5% correct answer for carpet wear.

Thirdly, we also find very high percentage of correct result for Upholstery. Here we find 97.5% correct answer. But we cannot find any proposed model to compare with it. After that, same things goes for wallpaper. It shows 91.5% correct result but cannot find any existing model for comparison.

6. CONCLUSIONAND FUTURE WORKS

6.1 Concluding Remarks

Due to continuous improvements of technology, usage of largest image dataset and resolution has been increasing rapidly. Medical sector to geography or in space, everywhere is has its uses. Finding image from large dataset using old technology such as text-image retrieval becomes almost impossible. So. In order to solve this problem, now a days Content-Base Image retrieval has been used instead of it. One of many ways of this technology, its uses texture feature of image classification and segmentation. There are many working model that use texture feature of image in image classification. But, in order improve the efficiency and save precious time, we proposed this model. First of all, we used different combination of algorithm which are Log Gabor Filter and SVD algorithm to improve the efficiency rate then we implemented this model on GPU to reduce to time and computational complexity. As, showed above in our experimental result and analysis chapter, we were successful in doing so.

6.2 Future works

In future, we would like to test our model with much more larger dataset to see its efficiency. Plus, we would also like to test our model with medical images from different test or diagnostics.

7. REFERENCES

[1] Jia Uddin, Md. Rashedul Islam, and Jong-Myon Kim, “An Efficient Gabor Filter and SVD based Texture Feature Extraction Method for State Monitoring of Induction Motors”, International Journal of Multimedia and Ubiquitous Engineering vol.9, No. 10 ,November,2014.

[2] V. T. Tran, F. Thobiani, A. Ball and B. K. Choi, “An application to transient current signal based induction motor fault diagnosis of Fourier–Bessel expansion and simplified fuzzy ARTMAP,” International Journal of Expert Systems with Application, vol. 40, (2013), pp. 5372-5384.

[3] Ying Liu , Dengsheng Zhang ; Guojun Lu ; Wei-Ying Ma, “Study on Texture Feature Extraction in Region-Based Image Retrieval System”, Multi-Media Modelling Conference Proceedings, 2006 12th International.

[4] Shao Hong, Duan Huidong, “Fractal dimension applied in texture feature extraction in x-ray chest image retrieval,” Information and Automation (ICIA), 2012 International Conference on, (2012) June 6-8, Shenyang.

[5] H. Keskes, A. Braham and Z. Lachiri, “Broken rotor bar diagnosis in induction machines through stationary wavelet packet transform and multiclass wavelet SVM,” International Journal of Electric Power Systems Research, vol. 97, (2013), pp. 151-157.

[6] M. Kang and J. Kim, “Singular Value Decomposition Based Feature Extraction Approaches for Classifying Faults of Induction Motors,” International Journal of Mechanical Systems and Signal Processing, vol. 41, no. 1-2, (2013), pp. 348-356.

[7] D. Nguyen, M. Kang, C. H. Kim and J. Kim, “Highly Reliable State Monitoring System for Induction Motors Using Dominant Features in a 2-Dimension Vibration Signal,” New Review of Hypermedia and Multimedia, vol. 13, no. 3-4, (2013), pp. 245-258.

[8] Ying Liu, Zong Li, “Study on texture feature extraction from forensic images with watermark,” Industrial Electronics and Applications (ICIEA), 2014 IEEE 9th Conference on, (2014) June 9-11, Hangzhou.

[9] V. T. Do and U. Chong, “Signal Model-Based Fault Detection and Diagnosis for Induction Motors Using Features of Vibration Signal in Two Dimension Domain,” Journal of Mechanical Engineering, vol. 57, no. 9, (2011), pp. 655-666.

[10] R. Shahriar, T. Ahsan and U. Chong, “Fault diagnosis of induction motors utilizing local binary pattern-based texture analysis,” EURASIP Journal on Image and Video Processing, 2013, vol. 29, (2013), pp. 1-11.

[11] W. C. Jang, M. Kang and J. M. Kim, “Fault Classification of an Induction Motor Using Texture Features of Vibration Signals,” LNEE, vol. 314, (2014), pp. 177-183.

[12] R. Nava, “Overcomplete Image Representations for Texture Analysis,” Electronic Letters on Computer Vision and Image Analysis, vol. 13, no. 2, (2014), pp. 40-41.

[13] M. Kang and J. M. Kim, “Reliable Fault Diagnosis of Multiple Induction Motor Defects Using a Two-Dimensional Representation of Shannon Wavelets,” IEEE Transactions on Magnetics, 2014 (accepted).

[14] Weitao Li, KeZhi Mao, Hong Zhang, and Tianyou Chai, “Designing Compact Gabor Filter Banks for Efficient Texture Feature Extraction,” Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on, (2010) Dec 7-10, Singapore.

[15] Rosniza Roslan, Nursuriati Jamil, “Texture Feature Extraction using 2-D Gabor Filters,” Computer Applications and Industrial Electronics (ISCAIE), 2012 IEEE Symposium on, (2012) Dec 3-4, Kota Kinabalu.

[16] Jianhong Liu, Wenquan Zhu, Minjie Mou, Lingli Wang, “Cropland Parcels Extraction Based on TextureAnalysis and Multi-spectral Image Classification,” Geoinformatics, 2010 18th International Conference on, (2010) Dec 18-20, Beijing.

[17] Maciel Zortea, Devis Tuia, Fabio Pacifici and Antonio Plaza, “SPECTRAL-TEXTURAL ENDMEMBER EXTRACTION,” Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2010 2nd Workshop on, (2010) June 14-16, Reykjavik.

[18] Weitao Li, KeZhi Mao, Hong Zhang, and Tianyou Chai, “SELECTION OF GABOR FILTERS FOR IMPROVED TEXTURE FEATURE EXTRACTION,” Image Processing (ICIP), 2010 17th IEEE International Conference on, (2010) Sept. 26-29, Hong Kong.

[19] J. Wang, G. Naghdy, and P. Ogunbona, “A New Wavelet Based ART Network For Texture Classification,” Intelligent Information Systems, 1996., Australian and New Zealand Conference on, (1996) Nov. 18-20, Adelaide, SA.

[20] Comparison study of Gabor and log-Gabor wavelets for texture segmentation,” Image and Signal Processing and Analysis (ISPA), 2011 7th International Symposium on, (2011) Sept. 4-6, Dubrovnik.

[21] Qimin Cheng ,Chongjun Yang , Feixiang Chen , Zhenfeng Shao, “Application of M-band wavelet theory to texture analysis in content-based aerial image retrieval,” Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International (Volume:3), (2004) Sept. 20-24, Anchorage, AK.

[22] S. Livens, P. Scheunders, G. Van de Wouwer, D. Van Dyck, “Wavelets for Texture Analysis”, First International Conference on Networking and Computing. Vol.2010.

[23] M. Kang and J. Kim, “Singular Value Decomposition Based Feature Extraction Approaches for Classifying Faults of Induction Motors,” International Journal of Mechanical Systems and Signal Processing, vol. 41, no. 1-2, (2013), pp. 348-356.

[24] J. Martinez-Alajarin , “Supervised classification of marble textures using support vector machines,” Electronics Letters (Volume: 40 , Issue: 11).

[25] Vapnik, V.: ‘The nature of statistical learning theory’ (Springer-Verlag, New York, 1995)

[26] Kim, K.I., Jung, K., Park, S.H., and Kim, H.J.: ‘Supervised texture segmentation using support vector machines’, Electron. Lett., 1999, 35, (22), pp. 1935–1937.

[27] Li, S., Kwok, J.T., Zhu, H., andWang, Y.: ‘Texture classification using the support vector machines’, Pattern Recognit., 2003, 36, pp. 2883–2893.

[28] M. Alper Selver , Olcay Akay, “Evaluating clustering methods for classification of marble slabs in an automated industrial marble inspection system,” Electrical and Electronics Engineering, 2009. ELECO 2009. International Conference on,(2009) Nov. 5-8, Bursa.

[29] Svetlana Lazebnik, Cordelia Schmid, Jean Ponce. “A Sparse Texture Representation Using Local Affine Regions,”IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1265-1278, (2005) August.

[30] G. F. Bin, J. J. Gao, X. J. Li. and B. S. Dhillon, “Early fault diagnosis of rotating machinery based on wavelet packets- Empirical mode decomposition feature extraction and neural network,” International Journal of Mechanical Systems and Signal Processing, vol. 27, (2012), pp. 696-7.

[31] Hadis Heidari, Abdolah Chalechale and Alireza Ahmadi Mohammadabadi, “Accelerating of Color Moments and Texture Features Extraction Using GPU Based Parallel Computing,”

Machine Vision and Image Processing (MVIP), 2013 8th Iranian Conference on,(2013) Sept.10-12, Zanjan.

[32] Namdev Sawant and Dinesh Kulkarni, “Performance Evaluation of Feature Extraction Algorithm on GPGPU,” Communication Systems and Network Technologies (CSNT), 2011 International Conference on, (2011) June 3-5, Katra, Jammu.

[33] Zhiqian Wang and Jezekiel Ben-Arie, “SVD and Log-Log Frequency Sampling with Gabor Kernels for Invariant Pictorial Recognition,” Image Processing, 1997. Proceedings., International Conference on (Volume:3),(1997) Oct 26-29, Santa Barbara, CA.

[34] P. D. Kovesi."MATLAB and Octave Functions for Computer Vision and Image Processing" .2000. Web. 31 July. 2015 "Available from: http://www.peterkovesi.com/matlabfns".

[35] Log Gabor filter.(n.d). In Wikipedia. Retrived December 16, 2015, from https://en.wikipedia.org/wiki/Log_Gabor_filter.

[36] N. Rose, “Facial Expression Classification using Gabor and Log-Gabor Filters,”Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference,(2006) April 2-6, Southampton.

[37] Xiao Zhitao,Guo Chengming , Yu Ming , Li Qiang, “Research on log Gabor Wavelet and its Application in Image Edge Detection,” Signal Processing, 2002 6th International Conference on (Volume:1), (2002) Aug 26-30.

[38] David Houcque. (2005, August) Introduction to Matlab for Engineering Students. Document.[Online].https://www.mathworks.com/matlabcentral/linkexchange/links/2089 introduction-to-matlab-for-engineering-students.

[39] Gabor filter. (n.d). In Wikipedia. Retrived December 16, 2015, from https://en.wikipedia.org/wiki/Gabor_filter.

[40] Nikhil Chaturvedi, Saurabh Agarwal, Punit Kumar Johari, “A novel approach of image retrieval based on texture,” International Journal of Enhanced Research in Management & Computer Applications, ISSN: 2319-7471 Vol. 3 Issue 1, (2014) January.

[41] P. S. Hiremath, Jagadeesh Pujari, “Content Based Image Retrieval based on Color, Texture and Shape features using Image and its complement,” International Journal of Computer Science and Security, Volume (1) : Issue (4) 35.