Proficient Technique for Satellite Image Enhancement Using Hybrid Transformation with FPGA

A dissertation submitted in partial fulfilment of the academic requirement for the award of the degree of

BACHELOR OF ENGINEERING

In

ELECTRONICS AND COMMUNICATION ENGINEERING

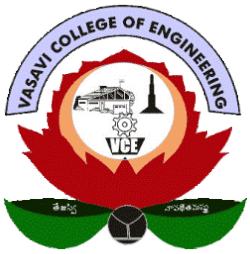
By

P. Mohammad Reala	**1602-19-735-140**
L. Navinder	**1602-19-735-143**
A. Navya	**1602-19-735-144**

Under the guidance of

Mr. S Naveen Kumar

Assistant Professor , E.C.E



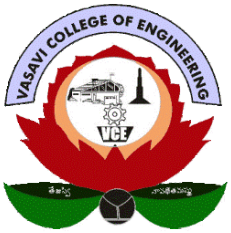
Department of Electronics and Communication Engineering

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

IBRAHIMBAGH, HYDERABAD-500031

2019-2023



Department of Electronics and Communication Engineering

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

IBRAHIMBAGH, HYDERABAD-500031

Date:

CERTIFICATE

This is to certify that the project work titled **"Proficient Technique for Satellite Image Enhancement Using Hybrid Transformation with FPGA"** submitted by

P. Mohammad Reala	**1602-19-735-140**
L. Navinder	**1602-19-735-143**
A. Navya	**1602-19-735-144**

students of Electronics and Communication Engineering Department, Vasavi College of Engineering in partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering in Electronics and Communication Engineering is a record of the bonafide work carried out by them during the academic year 2022-2023. The result embodied in this project report has not been submitted to any other university or institute for the award of any degree

Internal Guide
Mr. S Naveen Kumar
Assistant Professor
E.C.E Department

Head of the Department
Dr . E.SREENIVASA RAO
Professor and HOD
E.C.E Department

External Examiner

DECLARATION

This is to state that the work presented in this thesis titled **"**Proficient Technique for Satellite Image Enhancement Using Hybrid Transformation with FPGA**"** is a record of work done by us in the Department of Electronics and Communication Engineering, Vasavi College of Engineering, Hyderabad. No part of the thesis is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred in the text. The report is based on the project work done entirely by us and not copied from any other source. I hereby declare that the matter embedded in this thesis has not been submitted by me in full or partial thereof for the award of any degree/diploma of any other institution or university previously.

Signature of the students

P. Mohammad Reala	**1602-19-735-140**
L. Navinder	**1602-19-735-143**
A. Navya	**1602-19-735-144**

ACKNOWLEDGEMENTS

This satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mentioning of the people whose constant guidance and encouragement made it possible. We take pleasure in presenting before you our project, which is the result of a studied blend of both research and knowledge.

It is our privilege to express our earnest gratitude and venerable regards to our internal guide **Mr. Sarva Naveen Kumar**, Assistant professor, E.C.E. Department, Vasavi College of Engineering, Ibrahimbagh, for abounding and able guidance during the preparation and execution of the project work. We are grateful for his cooperation and his valuable suggestions.

We record with pleasure our deep sense of gratitude to **Dr. E. SREENIVASA RAO**, Head of the Department, E.C.E. for his simulating guidance and profuse assistance we have received, which helped throughout the project.

We also thank our Project coordinators, **Dr. K. Veera Swamy**, Professor, Department of ECE and **Dr. K. Deepthi**, Associate Professor, Department of ECE, for their continuous guidance and valuable suggestions throughout our project.

Our sincere thanks to the **Principal** and **Management, Vasavi College of Engineering, Hyderabad** for providing all the facilities in carrying out the project successfully.

Also we acknowledge with thanks for the support extended by all the staff members and technical staff in shaping up our project. We are thankful to one and all who cooperated with us during the course of our project work.

ABSTRACT

Techniques for digital photography aid in improving the visual quality of photos. Image improvements' main objective is to change an image so that the result is more appealing for the intended use than the original image.

In this research, we are testing a technique for enhancing the contrast and resolution of satellite photos. The image is processed using the discrete wavelet transform (DWT) and singular value decomposition (SVD) to create a new, enhanced image with improved contrast and resolution after the noise has been removed from the image using various filters. Satellite imaging is used in many different domains, such as geoscience research, astronomy, and geographic information systems. Using DWT, the input filtered image is split into four frequency sub-bands, and the high-frequency sub-band images are then interpolated back into the input image. The method also estimates the singular value matrices of the input filtered image's low-low sub band and the histogram equalised image, normalises both singular value matrices, and finally calculates brightness e. All of these sub bands are combined using inverse DWT to produce a new image with enhanced contrast and resolution. The process that follows employs a variety of noise and filter types, comparing the outcomes with both cutting-edge singular value decomposition (SVD) and discrete wavelet transform (DWT) methods as well as more conventional image equalisation techniques like general histogram equalisation (GHE) and local histogram equalisation (LHE).

The performance of the algorithms is evaluated using the metric Perception based Image Quality Evaluator (PIQE).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

1. DWT – Discrete Wavelet Transform
2. SVD – Singular Value Decomposition
3. GHE – General Histogram Equalisation
4. LHE – Local Histogram Equalisation
5. PIQE – Perception based Image Quality Evaluator
6. FPGA – Field Programmable Gate Array
7. AHE – Adaptive Histogram Equalisation
8. GIF – Guided Image Filtering
9. NLM – Non-local means Filtering
10. HE – Histogram Equalisation
11. WFT – Windowed Fourier Transform
12. FFT – Fast Fourier Transform
13. FWT – Fast Wavelet Transform
14. PSNR – Peak Signal-to-Noise Ration
15. SSIM – Structural Similarity Index
16. EDA – Electronic Design Automation
17. IDE – Integrated Development Environment
18. HDL – Hardware Description Language
19. ESL – Electrical System Level
20. HLS – High Level Synthesis
21. RTL – Register Transfer Level
22. FMC – FPGA Mezzanine Card
23. SDK – Software Development Kit
24. RTOS – Real-time Operating System

1. INTRODUCTION

Agriculture, urban planning, national security, and environmental monitoring are some of the industries where satellite photography has become a vital tool. The sharpness and accuracy of satellite images can, however, be affected by a variety of alterations and alterations. Many image enhancement methods have been proposed to enhance the quality of satellite images. The hybrid transformation is one such method that combines various image enhancement techniques to produce images of greater quality. In this paper, we describe a hybrid transformation-based strategy for enhancing satellite photos that is implemented on an FPGA.

Photographs shot in very dark or very light situations can lose detail in areas that are extremely consistently black or brilliant. The challenging element is figuring out how to increase contrast from the input satellite image, which has every information but is invisible. The main goal of image improvement is to change a picture's attributes to make it more appropriate for a particular task and viewer. One or more picture characteristics will be changed during this phase. There are many ways to enhance digital imagery to increase the visual quality of images. The imaging method, the tasks, and the visual environments all have a big impact on the best choice of such strategies.

To increase an image's visual quality, a sophisticated process called image enhancement uses a variety of techniques like filtering, contrast correction, and noise reduction. These methods can be implemented on FPGAs, which can also be tuned for certain purposes. This enables a more specialized and effective image processing strategy, improving accuracy and speed. Real-time processing is one of the main benefits of employing FPGA-based picture enhancement. FPGAs are perfect for real-time applications where quick and accurate processing are essential, such as in medical imaging, surveillance systems, and industrial inspection, because they can process massive amounts of data at rapid speeds.

1.1 MOTIVATION:

Numerous fields, including geoscience research, astronomy, and geographic information systems, employ satellite imagery. The contrast of satellite photos is one of the most crucial quality criteria. Images from satellites frequently have little contrast. It's common knowledge that contrast enhancement is among the most crucial aspects of picture processing. The difference in luminance reflected from two neighboring surfaces produces contrast. Contrast in visual perception is determined by how different an object's color and

brightness are from those of other objects. Absolute brightness is less responsive to our visual system than contrast. When an image's contrast is evenly and overly focused across a narrow range, information may be lost in those regions.

The hardware requirements for image processing can also be greatly decreased through FPGA-based image enhancement. Complex image processing algorithms can be implemented on an FPGA to cut down on the number of processing components needed, which can save money and reduce power usage. This is crucial in situations where cost and power consumption are major factors.

Real-time image processing operations are a significant benefit of FPGA-based picture enhancement. This is crucial in applications like disaster management, military surveillance, and environmental monitoring, where quick and precise picture processing is essential. FPGAs are perfect for real-time applications since they process enormous amounts of data quickly. Overall, FPGA-based image enhancement provides a strong and adaptable method for processing satellite images. FPGAs are a popular option for satellite image improvement applications because they can do real-time processing, implement customized algorithms, and require less hardware.

1.2 PROBLEM STATEMENT:

Today, a number of picture enhancement methods are accessible. A satellite image can be improved using a variety of models for better visuals. The goal of this research is to develop a method that will improve the resolution and of satellite photographs. The image is altered and given a resolution boost using the discrete wavelet transform (DWT) and singular value decomposition (SVD).

1.3 OBJECTIVES:

- To enhance a satellite image by employing a technique that utilizes discrete wavelet transform (DWT) and singular value decomposition (SVD)
- To analyse the performance with performance metrics.

1.4 PERFORMANCE METRICS:

One of the crucial tasks in image processing for the approach being used is evaluating the effectiveness of an image enhancement strategy. Different metrics—also referred to as performance metrics or evaluation metrics—are used to assess the effectiveness or quality

of the procedure. These performance indicators allow us to evaluate how successfully our enhancement method worked with the provided photos. This will enable us to boost the effectiveness of the image enhancing technique.

1.4.1 Perception based Image Quality Evaluator (PIQE) :

When satellite photos are upgraded using FPGA, the PIQE (Perceptual Image Quality Evaluator) measure is frequently employed to assess the quality of the improved images. The PIQE metric evaluates the quality of the augmented image based on how well the human visual system perceives the image. A popular method for assessing the quality of enhanced images in satellite image enhancement utilising FPGA is the PIQE (Perceptual Image Quality Evaluator) measure. The PIQE metric evaluates the quality of the improved image using a perceptual quality measure that takes into account how the human visual system reacts to the image.

In applications like disaster management, military surveillance, and environmental monitoring, where human interpretation of the image is essential, the PIQE metric is very helpful for assessing the quality of improved photos. Researchers can compare the quality of various improved photos in an impartial manner according to the PIQE metric, which offers a quantitative measure of the image quality. By enhancing contrast, sharpness, and lowering visual noise in the image, FPGA-based implementations of satellite image improvement algorithms can be made to perform better on the PIQE metric. Researchers can choose the most effective and efficient algorithm for the application by using the PIQE metric to assess the performance of several FPGA-based algorithms.

2. LITERATURE SURVEY

The accuracy and quality of satellite imagery are greatly improved through satellite image augmentation, making it a crucial area of remote sensing study. Researchers have looked into several FPGA-based methods for improving satellite images thanks to developments in Field Programmable Gate Array (FPGA) technology. We will examine some recent studies on FPGA-based satellite image improvement in this overview of the literature.

The authors of the 2019 article, "Real-time satellite image enhancement using FPGA-based multi-core architecture," suggested FPGA-based architecture for real-time satellite picture augmentation. Image pre-processing, enhancement, and post-processing are all included in the architecture's pipeline of processing steps. The authors showed that the suggested system accomplished real-time processing with high precision by evaluating the system's performance using various metrics, including processing time, accuracy, and resource utilisation.

The authors of the work "FPGA-based implementation of adaptive histogram equalisation for satellite image enhancement (2017)" suggested an FPGA-based implementation of adaptive histogram equalisation (AHE) for improving satellite images. The scientists showed that the FPGA-based version of the AHE algorithm offers high-speed processing and low power consumption, which makes it a widely used method for enhancing an image's contrast. The performance of the FPGA-based implementation was also contrasted with that of a software-based implementation, and the authors showed that the FPGA-based implementation performed better than the software-based implementation.

With the phrase "FPGA-based implementation of guided image filtering for satellite image enhancement" (2019), the authors of this research suggested an FPGA-based implementation of guided image filtering (GIF) for improving satellite images. The authors showed that the FPGA-based implementation of the widely used GIF technique for noise reduction and edge-preserving smoothing enables high-speed processing and low power consumption. The authors showed that the suggested system accomplished real-time processing with minimal resource utilisation by evaluating the performance of the FPGA-based implementation using a variety of metrics, including processing time and resource utilisation.

In the article "FPGA-based implementation of non-local means filtering for satellite image enhancement" (2020), the authors of this research suggested an FPGA-based non-local

means filtering (NLM) method for improving satellite images. The authors showed that the FPGA-based implementation of NLM offers high-speed processing and low power consumption. NLM is a frequently used approach for denoising and detail retention. The authors showed that the suggested system obtained high accuracy with little processing time by evaluating the performance of the FPGA-based implementation using a variety of metrics, including processing time and accuracy.

In conclusion, recent research has produced encouraging findings for FPGA-based methods of satellite picture improvement. Some of the commonly used methods for enhancing satellite images include adaptive histogram equalisation, guided image filtering, and non-local mean filtering. FPGA-based implementations of these methods have demonstrated that they offer fast processing speeds and low power consumption. A promising method for accomplishing real-time processing with high accuracy is the suggested multi-core architecture for real-time satellite picture augmentation. More investigation into FPGA-based methods for satellite image augmentation is anticipated to result in systems that are more precise and efficient for remote sensing applications.

3. THEORETICAL ANALYSIS

3.1 Histogram Equalisation

3.1.1 Introduction

A uniform histogram (i.e., uniform distribution) is produced in the output image by the histogram definition technique known as "histogram equalisation." The aim behind equalising a histogram in image processing is to extend and/or redistribute the original histogram using the entire range of discrete levels in the image, improving visual contrast. HE is a method that is widely applied to enhance the contrast of photographs due to its computational speed and simplicity. The methods based on HE are the most often used for boosting contrast in gray-level images. Despite the fact that it effectively increases visual contrast.

When the image's usable data is represented by close contrast values, this technique often improves the overall contrast of many photographs. As a result, areas that lack local contrast can acquire additional contrast. The method is effective in photos with foregrounds and backgrounds that are either light or dark.

Adaptive histogram equalisation, which divides the image into smaller portions and applies histogram equalisation separately to each region, is one type of histogram equalisation. The use of this adaptive method may be advantageous for images having localised contrast changes.

Histogram equalisation, a widely used technique in image processing, is typically employed as a first step in a number of computer vision and image analysis applications.

This technique is commonly used in medical applications to produce higher-quality black and white images, such as digital X-rays, MRIs, and CT scans.

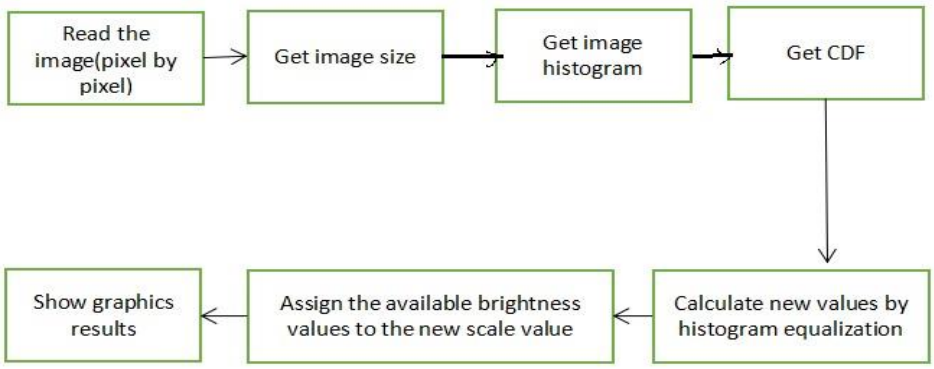


Fig 3.1 Block diagram for Histogram Equalisation

3.1.2 Steps Involved

1. Obtain the source image.

2.Create the image's histogram in step two.

3. Track down the image's local minima.

4. Split the histogram according to the local minima.

5. Specify the specific grey levels for each histogram partition.

6. Each partition should be given the histogram equalisation.

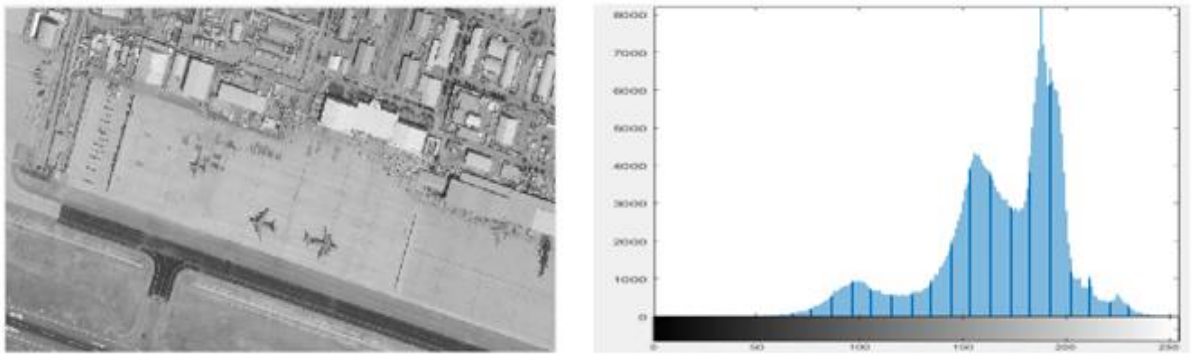


Fig 3.2 Original Image and its Histogram

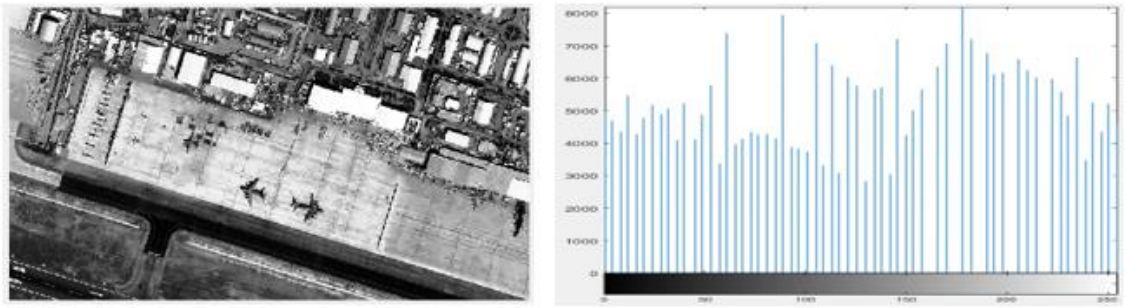


Fig 3.3 Histogram equalised image and its Histogram

The images clearly show that the new image's histogram has been equalised and its contrast has been improved. The general form of the histogram changes during histogram equalisation, however it stays the same during histogram stretching. This is another crucial point to remember.

3.1.3 Advantages and Disadvantages

The method's simplicity and invertibility as an operator make it a strong asset. The original histogram can therefore theoretically be restored if the histogram equalisation function is known. The calculation requires little processing power. The method's indiscriminate nature is a drawback. Although the useable signal is reduced and the contrast of background noise may rise, the brightness of the input image is not maintained on the output image.

Increased contrast in photos can be achieved with the help of histogram equalisation. By distributing the pixel values across the entire dynamic range, it enhances the image's visual appeal. The histogram equalisation approach is straightforward to use. The findings can be easily acquired, and no complicated computations are needed.

There is no information loss in the image as a result of histogram equalisation. The original image is not altered; only the pixel values are. In several software programmes, histogram equalization—a commonly used approach in image processing—has been implemented.

On the other hand, since histogram equalisation is a linear transformation, it is difficult to adapt it to fit particular applications. This may reduce its applicability in specific circumstances. The image's histogram is a key component of the histogram equalisation process. The results might not be sufficient if the histogram is not appropriate for the application.

Although the histogram equalisation procedure is straightforward, it might be computationally challenging for large photos. Longer processing times and more memory consumption may follow from this. Histogram equalisation occasionally causes over-enhancement in some areas of the image. Important details may be lost as a result, and artefacts may be introduced.

In conclusion, histogram equalisation is a potent method that can improve an image's contrast. It is not without its restrictions, though. Although easy to use and popular, it can over-enhance images, is computationally challenging, strongly relies on the image's histogram, and lacks versatility. Therefore, before choosing to implement histogram equalisation in image processing applications, it is crucial to carefully weigh its benefits and drawbacks.

3.2 Discrete Wavelet Transform

3.2.1 Introduction

Wavelets, which are mathematical procedures used to organise data according to frequency, were developed by scientists from many different fields. The data can be sorted after translation at a resolution appropriate for its scale. Data at different levels can be examined to produce a more complete picture. It is possible to discern between small and large characteristics since they are looked at independently. The discrete cosine seven transform cannot handle data discontinuities as well as the wavelet transform since it is not Fourier-based.

The discrete wavelet transform (DWT) is a mathematical transformation used in image and signal processing. By dividing a signal or an image into numerous frequency sub bands, it enables a multi-resolution analysis.

The DWT repeatedly applies a series of filters to the input signal or image. These filters, also known as wavelet filters or analysis filters, separate the input into approximation coefficients (low-frequency components) and detail coefficients (high-frequency components). The DWT has a hierarchical application that might result in a decomposition into numerous scales or levels.

The DWT can be expressed in a two-dimensional format for image processing. In this case, the horizontal and vertical components of the image are first separated. The horizontal decomposition creates approximation and detail coefficients along the rows, and the

vertical decomposition further divides each row's coefficients into approximation and detail coefficients along the columns.

The DWT has a wide range of uses, including signal analysis, time-frequency analysis, feature extraction, image compression (as in the JPEG2000 standard), and image denoising. For analysing and processing signals and images of various sizes and resolutions, it is a potent instrument.

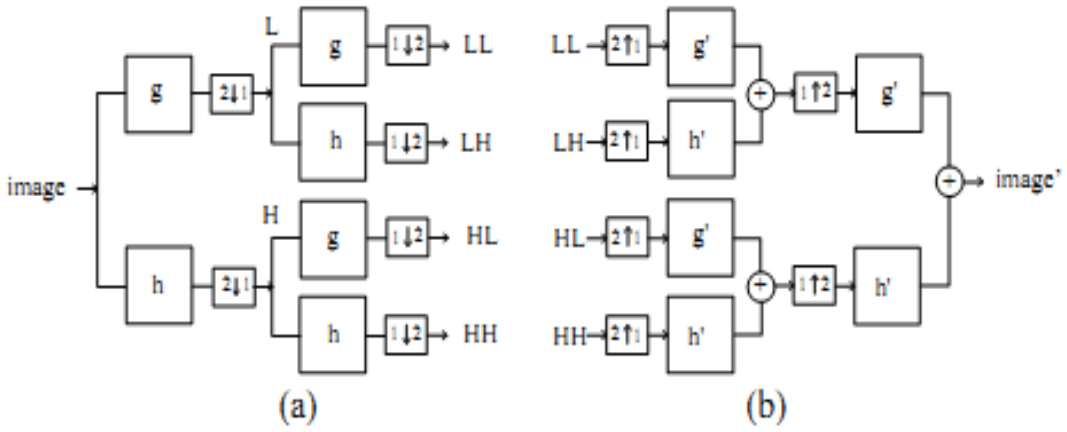


Fig 3.4 Wavelet Transform and Inverse Wavelet Transform

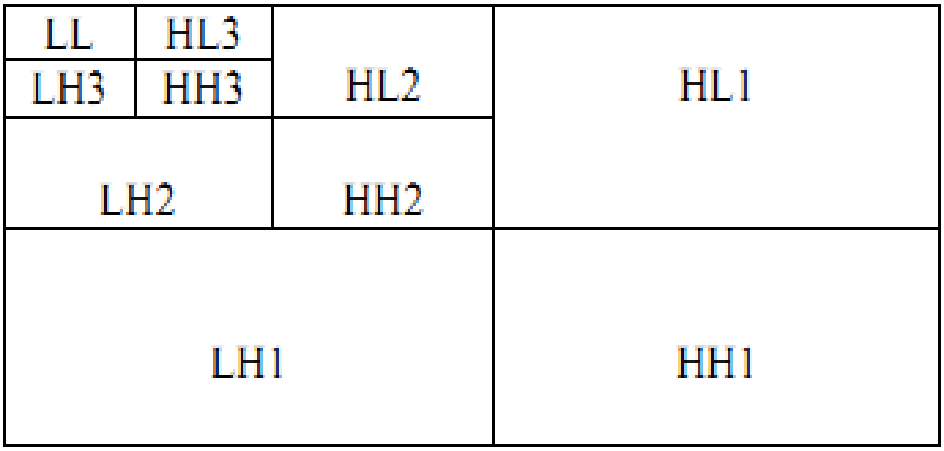


Fig 3.5 Structure of Wavelet Decomposition

3.2 Why Wavelet Transform

In signal processing, it is frequently necessary to see signals in a different way from their typical time-amplitude form.For correct analysis and processing, effective transform

techniques are required because the information frequently resides in the frequency domain or even time-frequency domain. These changes can reveal distinct information in the signals that could otherwise go unnoticed.

Different transformations can be chosen based on the signal at hand because they each have advantages. The frequency content is crucial for the majority of digital signal processing jobs. The most popular method for doing this is the Fourier Transform, however it has limitations because it can only represent stationary signals, or signals whose frequency content does not fluctuate over time. As a result, even though the Fourier Transform can show how much energy there is at a specific frequency point, it cannot determine when this energy enters the system.

A time-frequency representation is required because various signals, including picture, music, biological, and geographic signals, have variable properties with time. The Windowed Fourier Transform (WFT) was used as the solution up until this point.With WFT, the input signal is separated into segments, and the frequency content of each segment is examined independently. The chunks of this data are then windowed to converge to zero at the endpoints.

This is achieved by applying a weighted function that places more emphasis on the middle than the ends of the window. By localising the windowed signal in time, it can thus provide both time and frequency information, although it is constrained by the fixed window width. While processing signals at various sizes or resolutions are wavelet algorithms. Gross aspects of the signal can be seen by viewing it via a big window. Similar to this, when seeing a signal through a small window, the signal's minor details become apparent.

As a result, the Wavelet Transform is able to conceptually see both the forest and the trees. The Wavelet Transform currently has the best time-frequency representation of any other transformation thanks to this property.

3.2.3 Comparing Wavelet and Fourier Transform

Fast Fourier Transform (FFT) and Discrete Wavelet Transform (DWT) are linear processes for discrete signals. Numerous mathematical characteristics of their matrices are similar, and the inverse transform matrix for both the FFT and the DWT is simply the original matrix transposed. As a result, one can consider both transforms to be a rotation of function space to a different domain. This means that the new domain for the DWT has more

complex basis functions, such as wavelets, whereas the new domain for the FFT contains basis functions that are simpler sine and cosine waves. Both the DWT and FFT's basis functions are localised in frequency, making them both excellent for visualising power spectra, which demonstrate the amount of power contained within a frequency range.

The underlying wavelet functions' spatial localization is one of the two systems' most striking differences. Sine and cosine of Fourier are not. Many functions and operators that use wavelets are "sparse" when transformed due to this characteristic of wavelets as well as their localization in frequency. Because of this sparsity, wavelet treatments are excellent for data compression, identifying certain features in photos and movies, and removing noise. The fact that Wavelet Transforms use variable size windows is what really sets them apart from Fourier Transforms. The transform employs short windows for high frequency signal components and long windows for low frequency signal components. This leads to multiresolution analysis, in which the signal is examined at several frequencies and resolutions.

This results in multiresolution analysis where the signal is analysed with different resolutions at different frequencies. This means that for increasing frequency the time resolution increases and for decreasing frequency the frequency resolution increases. This results in the Wavelet Transform being able to locate a certain high frequency component in a signal more accurately in time than a low frequency one. At the same time, it can locate a low frequency component more accurately in frequency.

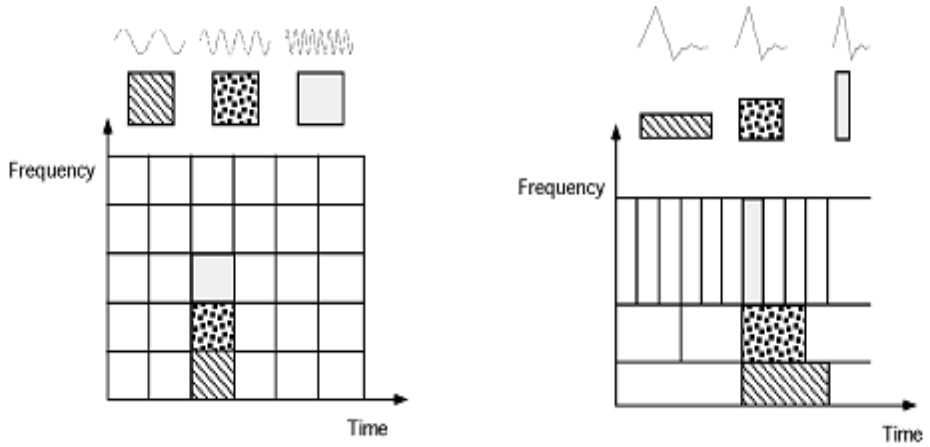


Fig 3.6 Windowed Fourier transform VS Wavelet transform

The time-frequency distinction between Windowed Fourier Transforms and Wavelet Transforms is seen in the above figure. As shown in image a The window function of the Windowed Fourier Transform is a straightforward square wave. The sine/cosine waves are then truncated to fit within this window. This indicates that the analysis's resolution is constant across the whole time-frequency plane. For a Daubechies Wavelet Transform in image b, the window changes over time and frequency. Due to this, we may view signals with fine-grained frequency analysis and isolate signal discontinuities.

The Wavelet Transform is gradually making its way into most fields of study due to how effectively it handles signal processing. It is now employed, to name a few, in the SNOW video codec and the jpeg2000 image file standard due to its exceptional image compression capabilities. In addition to its compression capabilities, it is often used in measuring equipment due to its capacity to detect concealed components in some signals, such as infrasound.

3.2.4 Haar wavelet

By computing the sums and differences of adjacent elements, the Haar wavelet analyses data. Prior to operating on adjacent vertical elements, the Haar wavelet operates on adjacent horizontal elements.

The Haar wavelet transform has the appealing property that it is identical to its inverse. The compact support of the Haar wavelet, which denotes that it is non-zero only in a finite interval, is what distinguishes it. The support range for the Haar wavelet is [0, 1]. Localised features in signals or images can be captured by the Haar wavelet thanks to its compact support property.

A particular application of the Discrete Wavelet Transform (DWT), the Haar transform makes use of the Haar wavelet. A signal or image is broken down into a number of approximation and detail coefficients at various scales or levels by the Haar transform. While the detail coefficients represent the high-frequency components, the approximation coefficients are responsible for the low-frequency components.

A pair of filters known as the Haar filters are applied to the signal or image as part of the hierarchical process known as the Haar transform. These filters perform a low-pass filtering and a high-pass filtering operation to extract the approximation and detail coefficients. The resolution is then decreased by down sampling the obtained coefficients.

Due to its clarity and simplicity, the Haar wavelet is frequently used as an introductory example in wavelet theory. In contrast to more complex wavelets, it has drawbacks. The Haar wavelet does not adequately capture frequency components because of its poor frequency localisation. In image compression applications, it also has staircase artefacts.

3.2.5 Advantages of DWT

- Multi-resolution analysis: The DWT provides a multi-resolution representation of the input signal or image. It allows for the analysis of different frequency components at different scales, which is valuable in many applications, such as image denoising, image compression, and feature extraction.
- Energy compaction: The DWT tends to concentrate the signal or image energy in a few significant coefficients while discarding or reducing less significant coefficients. This property makes it efficient for compression applications where high-energy coefficients can be retained, and low-energy coefficients can be discarded or quantized with less precision.
- Fast algorithms: Efficient algorithms, such as the Fast Wavelet Transform (FWT) and the lifting scheme, have been developed to compute the DWT in a computationally efficient manner. These algorithms exploit the redundancy and symmetry in the wavelet filters to reduce the computational complexity of the transform.
- Orthogonality: The DWT filters are designed to be orthogonal, meaning they preserve energy and do not introduce distortion. Orthogonality ensures that the signal or image can be perfectly reconstructed from its DWT coefficients.

3.3 Singular Value Decomposition

3.3.1 Introduction

Singular value decomposition (SVD) is a useful method for picture analysis and enhancement. It is a mathematical approach for dividing a matrix into three distinct matrices, each with its own set of properties. The method has been applied in a number of domains, including signal processing, computer vision, and image processing. In this research, we will investigate how SVD can be used to improve satellite image quality using FPGA technology.

Noise, haze, and other factors can degrade satellite photos, making them difficult to comprehend. Improving these photos can provide useful information for a variety of applications, such as agriculture, urban planning, and environmental monitoring. SVD is one method for improving satellite photos by dissecting the image into its constituent elements and modifying them selectively.

To use FPGA technology to build SVD for satellite image improvement, we must first gather a satellite image dataset. We will use this dataset to train and test our model. The data will then be pre-processed to remove any noise, and a filter will be applied to improve image quality. After pre-processing the data, we will utilize SVD to deconstruct the picture matrix into its constituent pieces.

After decomposing the image into its constituent pieces, we can choose to edit the singular values to improve the image. We can, for example, enhance the strength of certain single vectors to highlight specific characteristics in a picture. To decrease noise or blur, we can also lessen the intensity of additional single vectors.

3.3.2 Properties of SVD

Singular Value Decomposition (SVD) is a versatile mathematical tool that may be applied to a wide range of tasks, including satellite picture augmentation. In this research, we will investigate the features of SVD and how they might be used to improve satellite image quality using FPGA technology. The SVD algorithm divides a matrix into three parts: U, S, and V. Each of these matrices has distinct qualities that can be used to improve images.

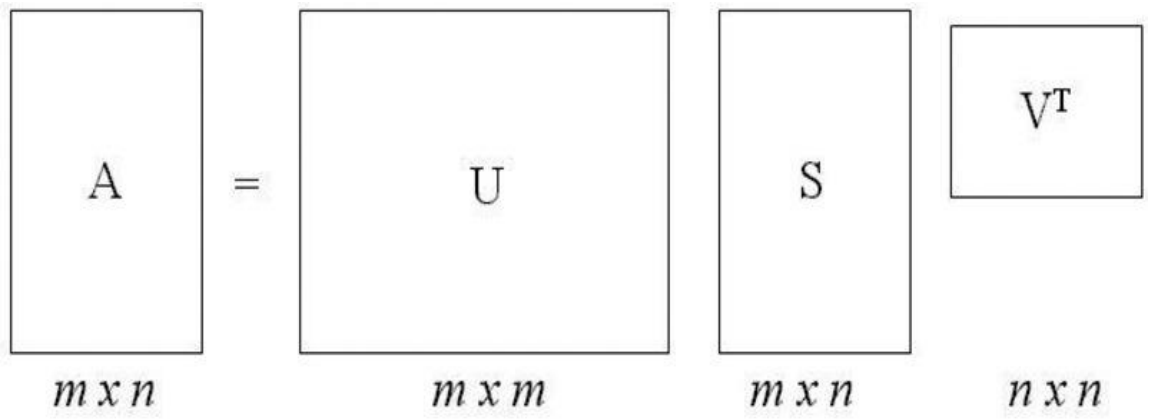


Fig 3.7 SVD Computation Matrices

One of the essential aspects of SVD is that the singular values in the S matrix represent the strength of each singular vector. These unique vectors represent the image's most important features. We can enhance specific features in the image by judiciously changing these individual values.

Another feature of SVD is that the left singular vectors in the U matrix indicate the image's principal components. These primary components represent the image's most critical information. We can emphasize specific features and reduce noise and blur by modifying the left singular vectors.

Furthermore, the right singular vectors in the V matrix describe the orientation of the image's features. We can change the orientation of the image's features by adjusting the singular vectors. The SVD algorithm can be considerably accelerated by using FPGA technology and a hardware accelerator, making it perfect for real-time satellite picture augmentation. The use of FPGA technology also allows for parallel processing, which can help accelerate the method.

To use FPGA technology to build SVD for satellite image improvement, we must first gather a satellite image dataset. The data will then be pre-processed to reduce noise, and filters will be applied to improve image quality. After pre-processing the data, we will utilize the SVD algorithm to deconstruct the picture matrix into its constituent pieces.

We can utilize performance measurements such as peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) to validate the results of the SVD-based satellite image improvement. These metrics can be used to compare the enhanced image to the original image and assess the image enhancement algorithm's efficacy.

SVD features can be used to improve satellite image quality by utilizing FPGA technology. We can enhance specific characteristics in the image and reduce noise and blur by selectively adjusting the singular values, left singular vectors, and right singular vectors. The SVD algorithm can be considerably accelerated by using FPGA technology and a hardware accelerator, making it perfect for real-time image augmentation. This initiative has the potential to yield useful data for a variety of uses, including agricultural, urban planning, and environmental monitoring.

3.3.3 Computing Procedure

Step-1: Image Processing

The first stage is to eliminate noise from the satellite image data and add filters to improve image quality. Techniques like denoising, deblurring, and contrast enhancement may be used. The purpose of this stage is to create a clean, high-quality image for use with the SVD algorithm.

Step-2: SVD Decomposition

The SVD algorithm is then applied to the pre-processed image. This entails breaking down the picture matrix into its constituent pieces, which are the left singular vectors (U), singular values (S), and right singular vectors (V). This can be accomplished by utilizing FPGA technology and a hardware accelerator to accelerate computation.

Step 3: Selective Manipulation of Singular Values

Once the image matrix has been deconstructed, the singular values can be selectively adjusted to enhance specific aspects of the image. This entails altering the strength of the single vectors to highlight or suppress specific aspects of the image.

Step 4: Selective Manipulation of Left Singular Vectors

The U matrix's left singular vectors can also be deliberately modified to highlight specific aspects of the image while removing noise and blur. This entails changing the image's main components to get the desired improvement.

Step 5: Selective Manipulation of Right Singular Vectors

The right singular vectors in the V matrix can also be modified selectively to modify the orientation of the image's features. This can be advantageous in certain situations, such as urban planning, where building and road orientation are critical.

Step 6: Reconstruction of Enhanced Image

Once the singular values, left singular vectors, and right singular vectors have been selectively manipulated, the enhanced image can be rebuilt using the updated SVD matrix. This can be accomplished by utilizing FPGA technology and a hardware accelerator to accelerate computation.

Step 7: Performance Evaluation

Performance metrics such as peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) can be used to compare the enhanced image to the original image to assess the success of the SVD-based satellite image enhancement technique. These metrics provide a quantitative measure of the algorithm's increase in image quality.

3.4 Verilog

3.4.1 Introduction

A hardware description language called Verilog is used to simulate, develop, and model digital circuits and systems. Since its first introduction in the 1980s, it has grown in popularity as a language for building and evaluating digital systems in the electronics sector. Engineers can define circuits and systems in Verilog at a high level, which can then be translated into a gate-level representation that can be implemented in hardware.

Modules in Verilog can be instantiated to produce a bigger design. These modules may contain simpler components like flip-flops, logic gates, and registers as well as more sophisticated ones like processors and memory cells. Verilog offers building blocks for creating intricate state machines and other digital systems.

Electronic design automation (EDA) tools that support Verilog include those that enable design simulation, synthesis, and verification. Before committing to hardware implementation, engineers can validate and test their designs using these tools.

Digital circuits for a variety of applications, such as consumer electronics, telecommunications, and aerospace, are frequently designed using Verilog. It continues to develop to satisfy the needs of the electronics industry and is a vital tool for digital circuit designers thanks to its simplicity of use and powerful capabilities.

3.4.2 Tools used

Integrated development environments (IDEs), simulators, synthesisers, and waveform viewers are a few of the tools utilised in Verilog programming. Each tool, from coding through verification and implementation, plays a significant part in the Verilog design pipeline.

Integrated Development Environment : A standard integrated development environment (IDE) comes with a code editor, a debugger, a build automation tool, and other capabilities that make it easier for developers to write, test, and debug their code. An IDE's build automation tool automates the building and compilation of code, cutting down on the time and labour necessary to produce a usable executable. Version control, code restructuring, and code profiling are further tools that IDEs offer to aid developers in managing their code more efficiently.

Simulators: Before committing digital designs to hardware implementation, designers can test and simulate them using simulators, which are software tools. Simulators help designers evaluate the functioning of their designs, spot mistakes early in the design cycle, and study how their creations behave under various scenarios. Event-driven and cycle-based simulators are the two primary categories of Verilog simulators. The most popular kind of Verilog simulators are event-driven simulators. On the basis of the happening of events, they imitate the behaviour of digital circuits. Digital circuits are simulated by cycle-based simulators using a constant clock cycle.

Synthesizers: Verilog synthesisers are computer programmes that translate Verilog code into a gate-level netlist for usage in a real hardware device. Synthesis is the process of transforming Verilog code, which describes a digital circuit's functioning, into a list of gates and flip-flops that actually implement the circuit as physical logic gates. The Verilog synthesizer examines the Verilog code and produces an optimized netlist that satisfies the design's objectives for performance, area, and power. Techniques including placement and routing, technology mapping, and logic optimization may be used during the optimization process.

Waveform Viewers: Software tools called waveform viewers enable designers to monitor and examine the behaviour of digital circuits during simulation. The waveform viewer in Verilog simulation shows the values of the design's signals over time, enabling designers to see and comprehend the design's behaviour. Signals are shown as waveforms in the waveform viewer, which are graphs of the signal values through time. Both analogue and digital waveforms, which display discrete values, can be used by the viewer to display signals. Designers can benefit from the Verilog waveform viewer's many capabilities, including the ability to zoom in and out, to display many signals on one plot, and to identify or annotate the waveform. Tools for measuring signal timing and analysing signal

behaviour, such as rise and fall periods, pulse widths, and frequency, may also be available through the waveform viewer.

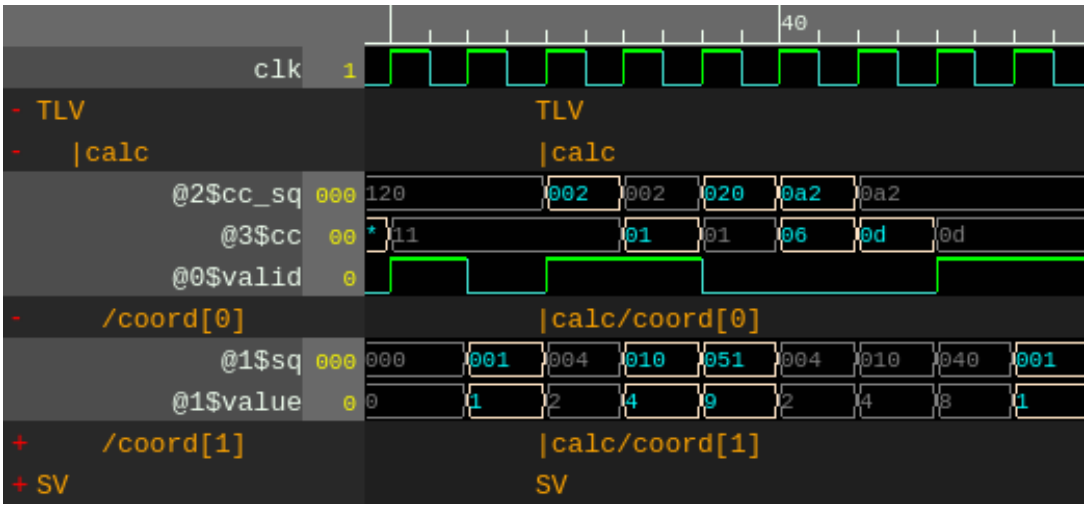


Fig 3.8 A waveform viewer in Verilog Tool

4. METHODOLOGY

4.1 Tools

4.1.1 MATLAB

Satellite image improvement is critical for a wide range of applications, including agriculture, urban planning, and disaster management. Hybrid transformation-based techniques have received a lot of interest in recent years because of their ability to preserve both spatial and spectral information while improving overall image quality. This paper provides an in-depth look at the use of MATLAB and FPGA for satellite picture enhancement utilizing hybrid transformations.

To produce superior enhancement effects, hybrid transformation-based approaches combine several types of transformations. Spatial and spectral transformations are the most commonly employed forms of transformation in hybrid techniques. Spatial transformations change the spatial arrangement of pixels in an image, whereas spectral transformations change the image's spectral characteristics. To improve satellite images, hybrid transformations combine the advantages of spatial and spectral modifications.

MATLAB is a great tool for enhancing satellite images via hybrid transforms. MATLAB has a number of functions and tools that enable users to implement and evaluate various types of transformations. Users can also utilize MATLAB to evaluate the performance of various transformations using metrics such as peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM).

FPGA (Field-Programmable Gate Array) is a hardware platform that can be used to expedite satellite image processing. When compared to typical CPU-based systems, FPGA-based systems can deliver faster processing speeds. FPGA-based systems can also be tailored to implement specific algorithms, making them ideal for image processing jobs like satellite image augmentation.

The first step in implementing satellite picture improvement using hybrid transformations on an FPGA is to build the hybrid transformation algorithm in MATLAB. After that, the MATLAB code can be transformed to a hardware description language (HDL) such as VHDL or Verilog. Using tools like Xilinx Vivado or Altera Quartus, the HDL code may then be synthesized and transferred to the FPGA.

The ability to analyse massive amounts of data in real-time is one of the primary benefits of employing an FPGA for satellite picture augmentation. FPGA-based systems can also be tailored to match specific performance requirements, making them appropriate for a variety of satellite image augmentation jobs.

The improvement of satellite images via hybrid transformations is a difficult undertaking that necessitates a mix of spatial and spectral adjustments. MATLAB is an extremely capable environment for implementing and testing many types of transformations. FPGA-based systems can deliver quicker processing speeds and can be tailored to individual performance needs. The power of MATLAB and FPGA combined can result in efficient and effective satellite picture enhancement solutions.

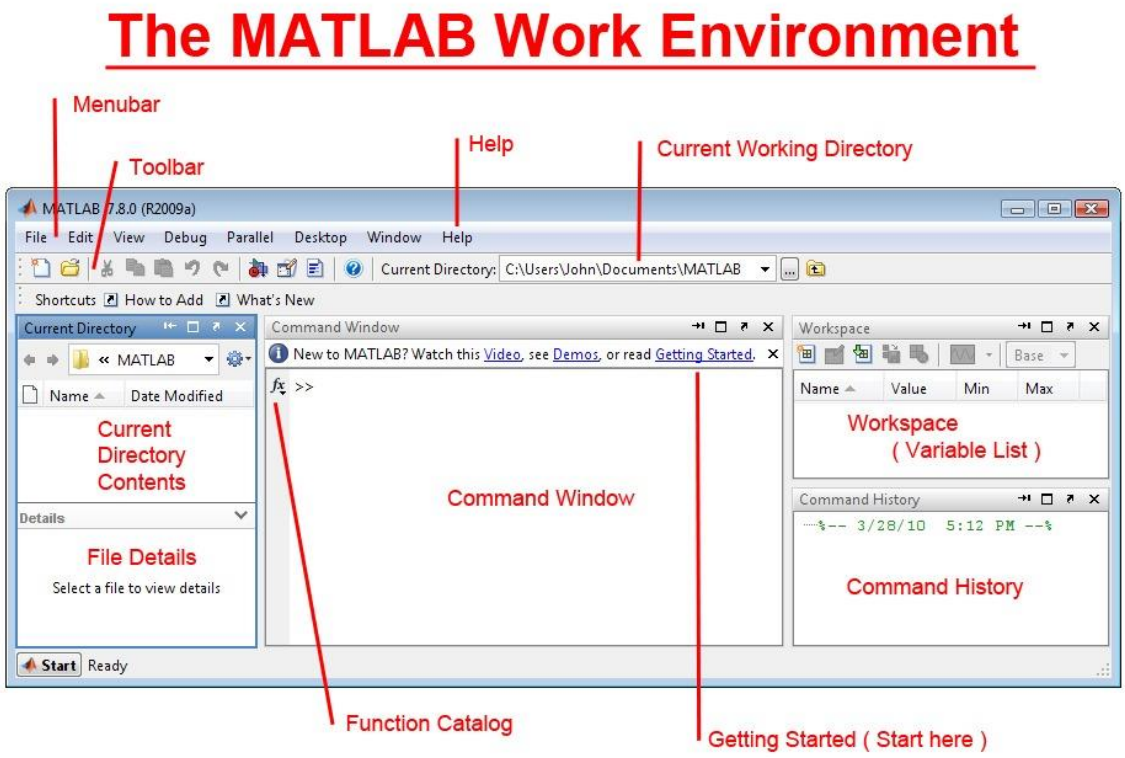


Figure 4.1 MATLAB Work Environment

4.1.2 Vivado

Vivado Design Suite is a software tool developed by Xilinx that offers increased features for system-on-a-chip development and high-level synthesis, replacing the previous Xilinx ISE tool. It is commonly used to synthesize and analyse hardware description language (HDL) designs, and was created to revolutionize the design flow.

Vivado is an integrated design environment (IDE) that offers system-to-IC-level capabilities and was introduced in April 2012. The software tool is based on a scalable shared data model and a common debugging environment. It provides electrical system level (ESL) design tools that enable designers to create and validate C-based algorithmic intellectual property (C-based algorithmic IP). Additionally, Vivado offers standards-based IP stitching, system integration, and verification of all types of system building blocks, as well as standard-based algorithmic and RTL IP packaging for reuse.

A limited version of the design environment is available for free with Vivado's WebPACK Edition. The software tool also includes the Vivado High-Level Synthesis (HLS) compiler, which allows developers to write programs in C, C++, and SystemC and directly target Xilinx devices without having to manually write RTL. Vivado HLS supports C++ classes, templates, functions, and operator overloading, which can increase developer productivity. Moreover, Vivado 2014.1 includes assistance for automatically converting OpenCL kernels to IP for Xilinx devices. Various CPU, GPU, and FPGA systems support the execution of programmes referred to as OpenCL kernels.

The programming and design of FPGAs and SoCs can be done entirely within the development environment provided by the robust and powerful tool known as Vivado. Designers working in a variety of fields and applications frequently use it because of its sophisticated features, integration abilities, and IP core libraries.

Memory controllers, communication interfaces, and video processing modules are just a few of the pre-designed and certified functional blocks that are available with Vivado's IP cores. To hasten development and shorten time-to-market, these IP cores can be altered and included in designs.

A built-in debugging tool for Vivado called Vivado Logic Analyzer enables designers to record and examine signals inside their FPGA design to help with troubleshooting. This instrument can be used to identify and narrow down problems as well as confirm proper operation.

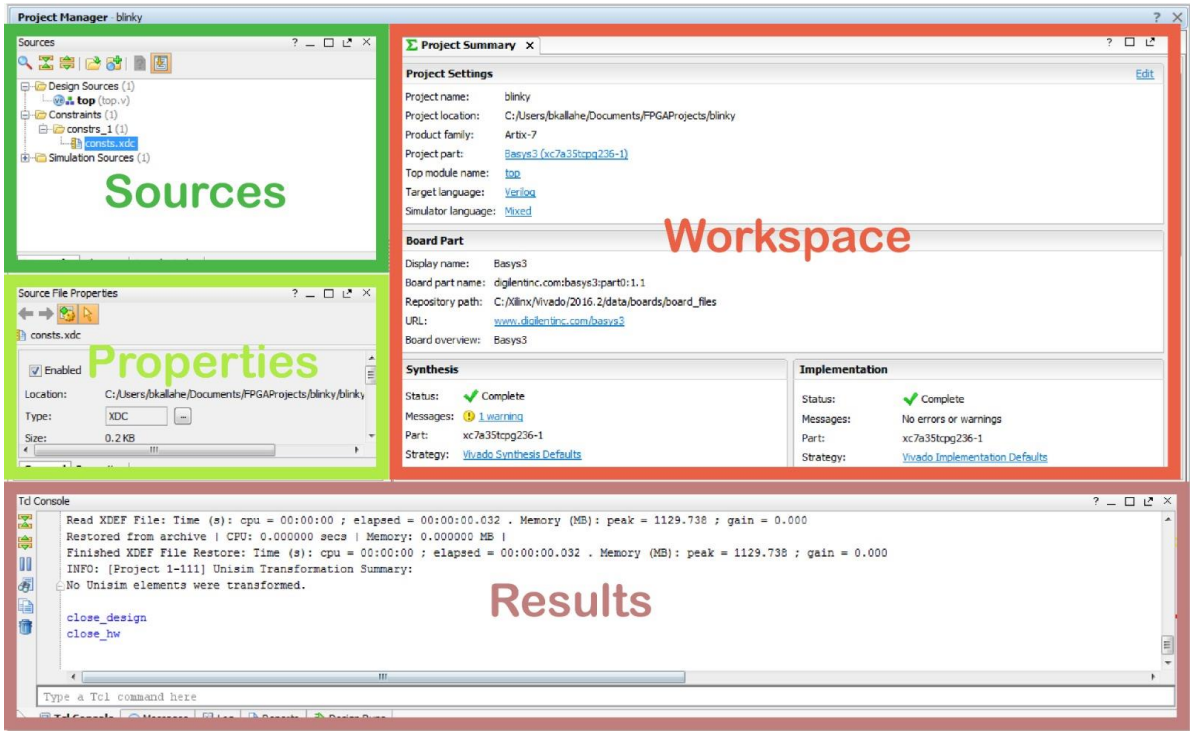


Figure 4.2 Different Tabs in Vivado

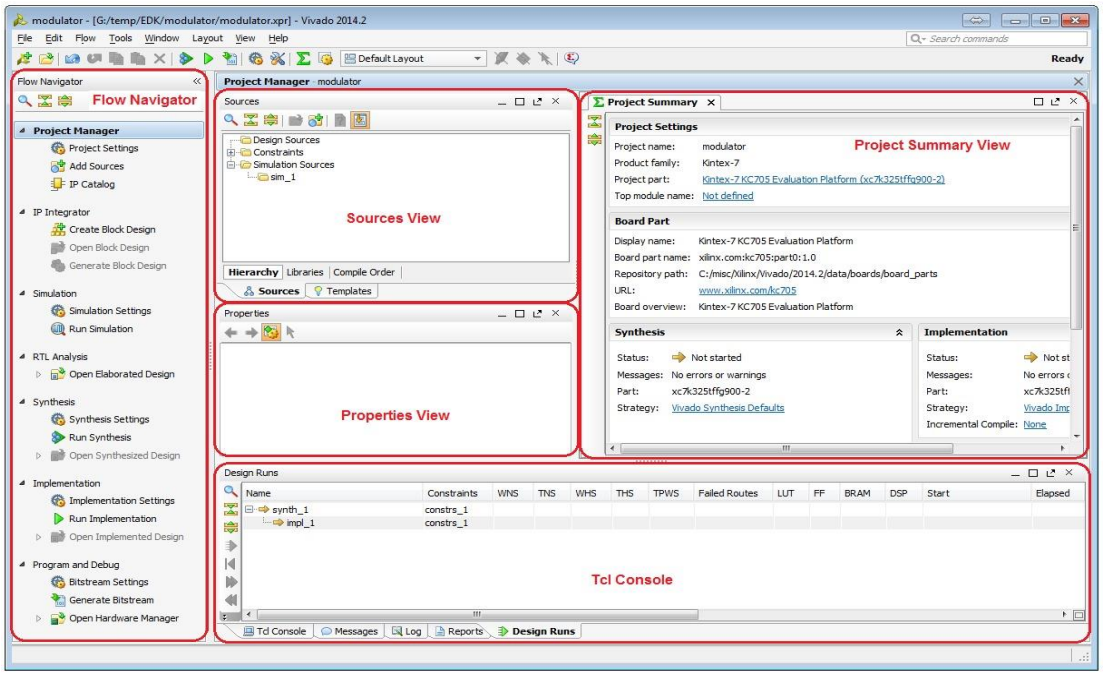


Figure 4.3 A basic image of Vivado Tool

4.1.3 ZedBoard

A cheap development board for the Xilinx Zynq®-7000 SoC is called ZedBoardTM. This board comes equipped with everything required to build a Linux, Android, Windows®, or other OS/RTOS-based design. Additionally, a number of expansion connectors make the processing system and programmable logic I/Os accessible to users. Utilise the tightly integrated ARM® processing system and 7 series programmable logic of the Zynq-7000 SoC to build cutting-edge designs for the ZedBoard.

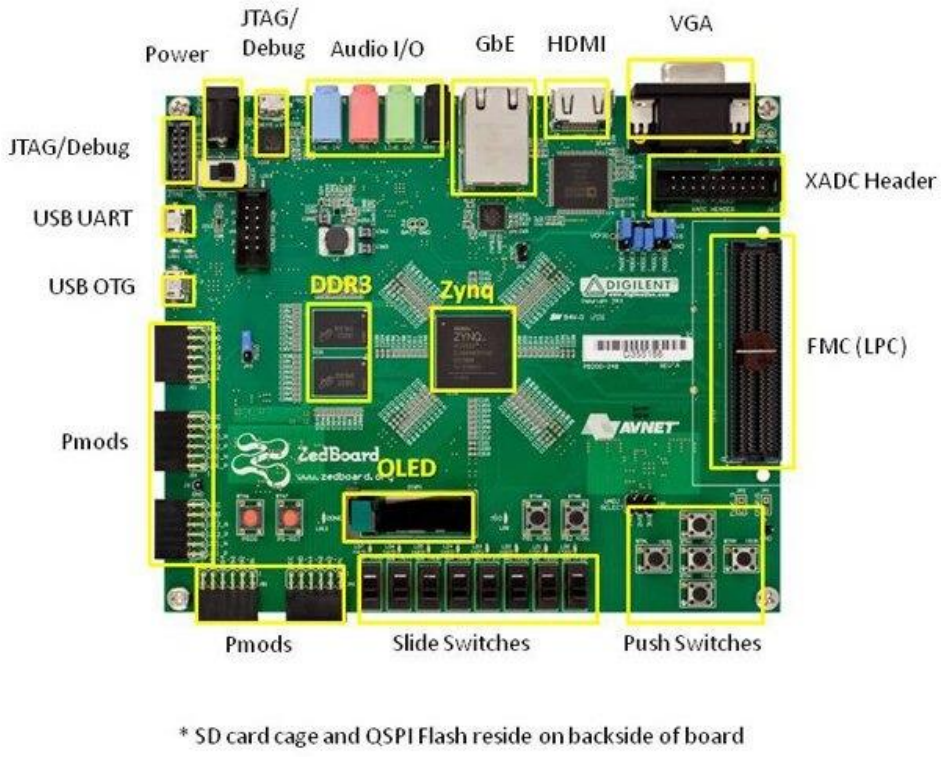


Figure 4.4 ZedBoard

On a single chip, the Zynq-7000 SoC integrates a dual-core ARM Cortex-A9 processor with programmable logic. It combines the processing capability of a CPU with the flexibility and performance of an FPGA. The Xilinx 7-series FPGA fabric is a feature of the Zynq-7000 SoC on the ZedBoard. Reconfigurable logic is offered by the FPGA and can be utilised to construct unique hardware accelerators, interfaces, or specialised operations. The ZedBoard has a variety of memory components, including 256 MB of Quad-SPI Flash memory, 512 MB of DDR3 RAM, and a microSD card port. The board includes a number of interfaces and peripherals, including as USB ports, Ethernet ports, HDMI output, audio

input/output, Pmod connections, and an FMC (FPGA Mezzanine Card) connector for expansion.

Different software tools and development environments can be used to programme and develop the ZedBoard. The ARM Cortex-A9 processors can be programmed using industry-standard software development tools, such as the GNU Compiler Collection (GCC) or Xilinx Software Development Kit (SDK), while FPGA programming is frequently carried out using Xilinx's Vivado Design Suite. Real-time operating systems (RTOS) like FreeRTOS and Linux are among the operating systems that the ZedBoard supports.

The ARM Cortex-A9 processors can be used to run these operating systems, which interact with the programmable logic and make use of their features. The ZedBoard has a vibrant development and user community that exchanges information, projects, and support. Users may get started and explore the board's possibilities with the help of online forums, documentation, tutorials, and example designs. For many different applications, including embedded systems, digital signal processing, motor control, image processing, and IoT (Internet of Things) projects, the ZedBoard is extensively utilised in prototyping, development, and education.

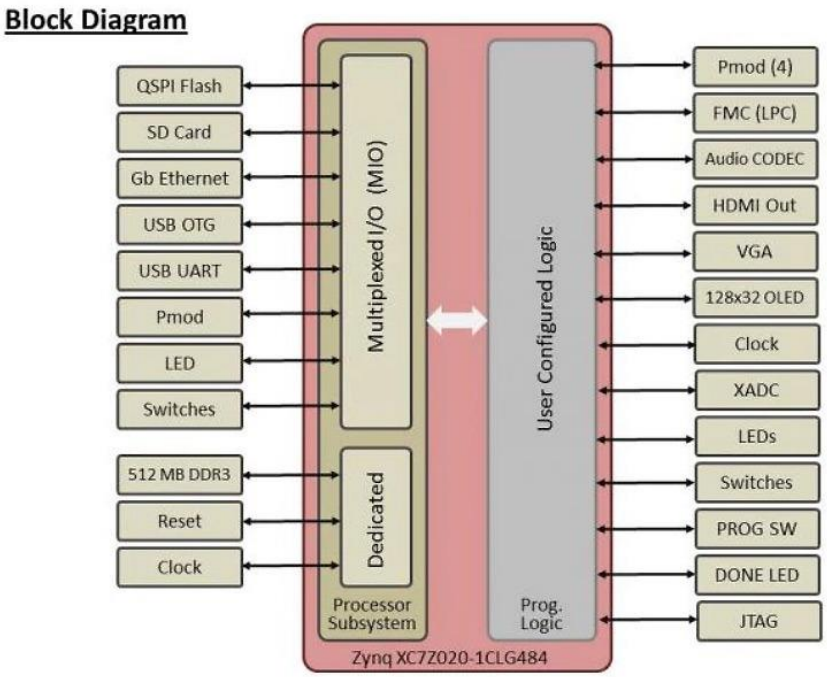


Figure 4.5 Block Diagram of a ZedBoard

Pin Configuration:

- FPGA I/O Pins: The ZedBoard has several FPGA I/O banks, each with a set of input/output pins. These pins can be utilised to connect to external hardware or peripherals. Tools like Xilinx Vivado can be used to define the precise pin assignments in the FPGA design.
- Pmod Connectors: The ZedBoard has two Pmod connectors (JC and JD), which allow for more I/O. Digital I/Os, power, and ground are all included in each Pmod connector is 12 pin count. The documentation for the ZedBoard contains the pin assignments for the Pmod connectors.
- FMC Connector: The High-Speed Mezzanine Card (HSMC), commonly known as the FMC connector, is a component of the ZedBoard. This connector enables the connection of FMC expansion boards or modules, which can offer specialised functionality, more I/O interfaces, or high-speed data transfer. The FMC connector's pin assignments can change based on the particular FMC module being utilised.
- HDMI Connector: To display videos, the ZedBoard contains an HDMI output port. The HDMI connector's pin assignments follow the HDMI standard and are used to transmit both visual and audio signals.
- USB Ports: Multiple USB ports are available on the ZedBoard for connectivity. These ports' pin assignments are predetermined and adhere to the USB specification.
- Ethernet Ports: For network communication, the board has one or more Ethernet ports. Ethernet ports typically have set pin assignments that adhere to accepted Ethernet protocols.

4.2 Procedure

The hybrid transformation stage entails applying numerous image enhancement techniques to the pre-processed image to obtain a high-quality, improved image. The hybrid transformation can comprise histogram equalization, contrast stretching, filtering, and other approaches. The choice of image-enhancing techniques is determined by the application's specific requirements.

The image enhancing technique is as follows. The low contrast image (let us assume A) is duplicated twice. On one copy, generalised histogram equalisation is carried out. Let H

represent the equalised image. Next, A and H are both subjected to a discrete wavelet transform. Four sub bands—LL, LH, HL, and HH—are obtained when DWT is used.

Let LL_H, LH_H, HL_H, HH_H and LL_A, LH_A, HL_A, HH_A stand in for the sub bands of H and A, respectively. Since the LL band contains the majority of the available information, it is all we are interested in. For the two LL bands, LL_A and LL_H, we compute SVD.

In both cases, the singular matrix is obtained by computing the SVD. Singular values are found in the diagonal of the singular matrix, which is a diagonal matrix. The lighting information is contained in singular values.

$LL_A = U_A\ S_A\ V_A^T$

$LL_H = U_H\ S_H\ V_H^T$

The singular matrices of the LL sub band of A and H are thus denoted by S_A and S_H. Following the acquisition of the singular matrices, we must determine the correction factor E. E is computed by dividing the highest value in S_A by the highest value in S_H. Forming a new singular matrix $S_A' = E\ S_A$ is the next stage.

A new LL band called LL_A' is created with the formula $LL_A' = U_A\ S_A'\ V_A^T$. The upgraded LL band that we need is this. In order to create the enhanced image, this band is now paired with the original LH_A, HL_A, and HH_A bands. The original high frequency bands are retained because they have edge data.

The post-processing procedure entails removing additional noise and aberrations from the enhanced image. Filtering techniques such as median filtering, Gaussian filtering, and Laplacian filtering can be used at the post-processing stage. Edge detection techniques such as the Sobel, Prewitt, and Roberts operators can also be used in the post-processing step. The post-processing stage can increase the augmented image's visual quality and prepare it for display.

The code is converted to Verilog code and is tested on the Vivado Tool using ZedBoard as reference FPGA. A testbench is written to give an input image to test the functionality so the verilog code that is to be implemented.

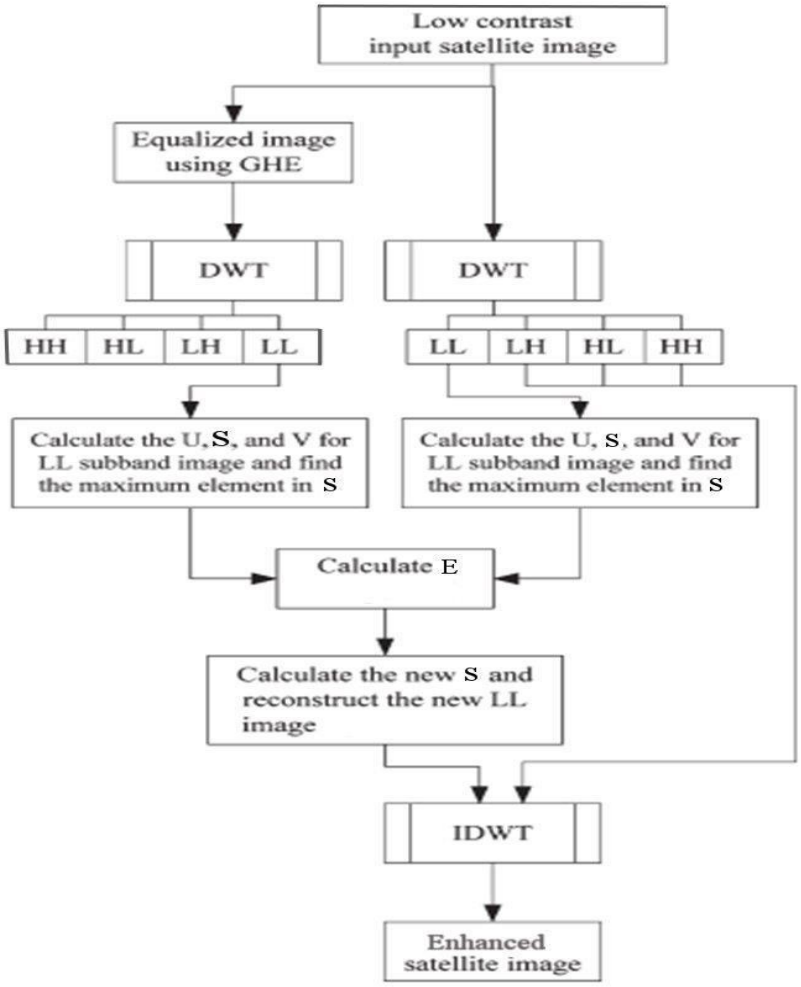


Figure 4.6 Flowchart of the methodology

The proposed methodology can be applied to FPGAs to obtain real-time performance while consuming less power. The FPGA implementation entails creating a hardware architecture that includes phases for pre-processing, hybrid transformation, post-processing, and display. To enable parallel processing and optimize power consumption, the hardware architecture may include DSPs and FPGAs.

The amount of image-enhancing techniques employed in the hybrid transformation, the number of processing units, and the optimization of the design for low power consumption are all factors to consider while designing the hardware architecture. Techniques like pipelining, parallelism, and clock gating can be used to optimize the architecture.

Advantages of Proposed methodology:

The suggested FPGA-based satellite image enhancement methodology has various advantages over existing solutions. For starters, the process may generate high-quality upgraded photos with greater contrast, noise reduction, and detail. Second, the methodology can be applied to FPGAs to attain real-time performance while consuming less power. Third, the methodology can be tailored to unique application needs, providing flexibility and versatility.

Limitations of Proposed Methodology:

There are certain drawbacks to the suggested technology for satellite image augmentation utilizing FPGAs. For starters, the process may necessitate large computational resources, raising the cost of implementation. Second, the process may necessitate specialized hardware and software knowledge, which may limit its use. Finally, extensive testing and validation may be required to confirm the methodology's effectiveness and reliability.

5. EXPERIMENTAL RESULTS

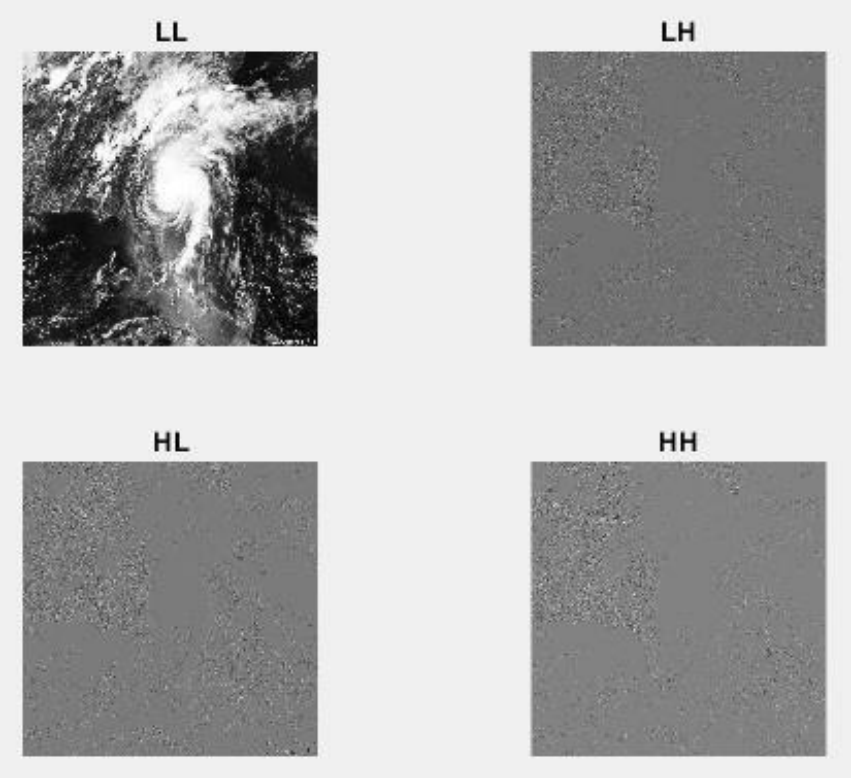


Figure 5.1 Sub-bands of the image

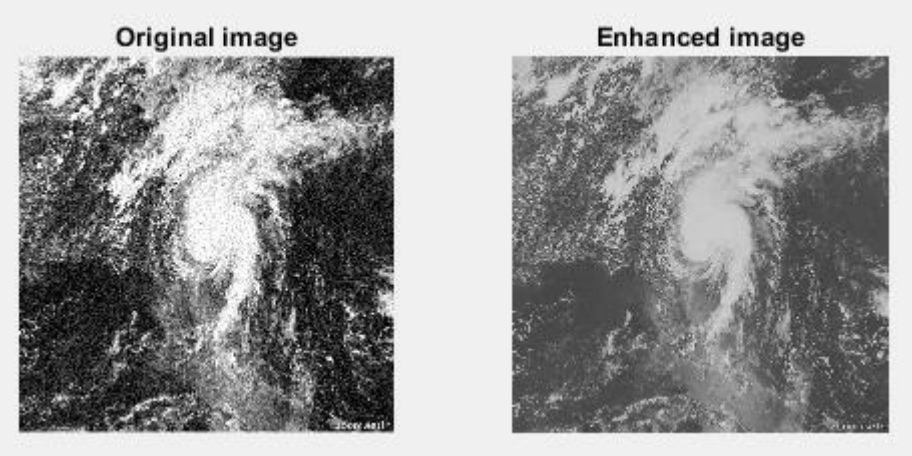


Figure 5.2 Original and Enhanced Images

PIQE of Original image : 24.8199

PIQE of Enhanced image : 20.8100

Detailed RTL Components Information:

Adders :
2 Input 32 Bit Adders := 2
4 Input 17 Bit Adders := 1
5 Input 17 Bit Adders := 2
2 Input 17 Bit Adders := 1
2 Input 16 Bit Adders := 1
2 Input 11 Bit Adders := 1
2 Input 9 Bit Adders := 1
2 Input 8 Bit Adders := 1

Registers :
32 Bit Registers := 3
17 Bit Registers := 10
15 Bit Registers := 1
14 Bit Registers := 2
11 Bit Registers := 5
9 Bit Registers := 12
8 Bit Registers := 2
3 Bit Registers := 6
2 Bit Registers := 4
1 Bit Registers := 9

RAMs :
272K Bit RAMs := 1
176K Bit RAMs := 1

Muxes :
2 Input 17 Bit Muxes := 5
2 Input 15 Bit Muxes := 10
2 Input 11 Bit Muxes := 6
2 Input 8 Bit Muxes := 1
2 Input 1 Bit Muxes := 5

Source	Voltage (V)	Total (A)	Dynamic (A)	Static (A)
Vccint	1.000	6.697	6.390	0.307
Vccaux	1.800	1.083	0.983	0.100
Vcco33	3.300	0.000	0.000	0.000
Vcco25	2.500	0.000	0.000	0.000
Vcco18	1.800	5.688	5.687	0.001
Vcco15	1.500	0.000	0.000	0.000
Vcco135	1.350	0.000	0.000	0.000
Vcco12	1.200	0.000	0.000	0.000
Vccaux_io	1.800	0.000	0.000	0.000
Vccbram	1.000	0.169	0.127	0.042
MGTAVcc	1.000	0.000	0.000	0.000
MGTAVtt	1.200	0.000	0.000	0.000
MGTVccaux	1.800	0.000	0.000	0.000
Vccpint	1.000	0.473	0.000	0.473
Vccpaux	1.800	0.010	0.000	0.010
Vccpll	1.800	0.003	0.000	0.003
Vcco_ddr	1.500	0.000	0.000	0.000
Vcco_mio0	1.800	0.000	0.000	0.000
Vcco_mio1	1.800	0.000	0.000	0.000
Vccadc	1.800	0.020	0.000	0.020

Figure 5.3 Power Supply Summary

On-Chip	Power (W)	Used	Available	Utilization (%)
Slice Logic	1.548	700	---	---
LUT as Logic	1.323	258	53200	0.48
CARRY4	0.148	40	13300	0.30
Register	0.069	301	106400	0.28
BUFG	0.006	1	32	3.13
LUT as Shift Register	0.002	18	17400	0.10
Others	0.000	9	---	---
Signals	3.195	595	---	---
Block RAM	1.589	14	140	10.00
I/O	12.190	117	200	58.50
Static Power	1.064			
Total	19.586			

Figure 4.4 On- Chip Components Power Analysis

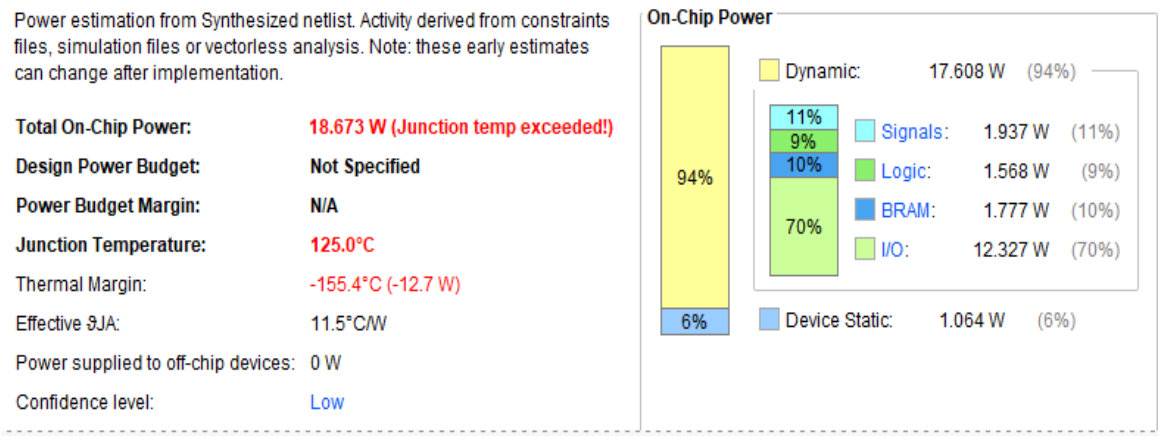


Figure 5.5 Power Summary Synthesis

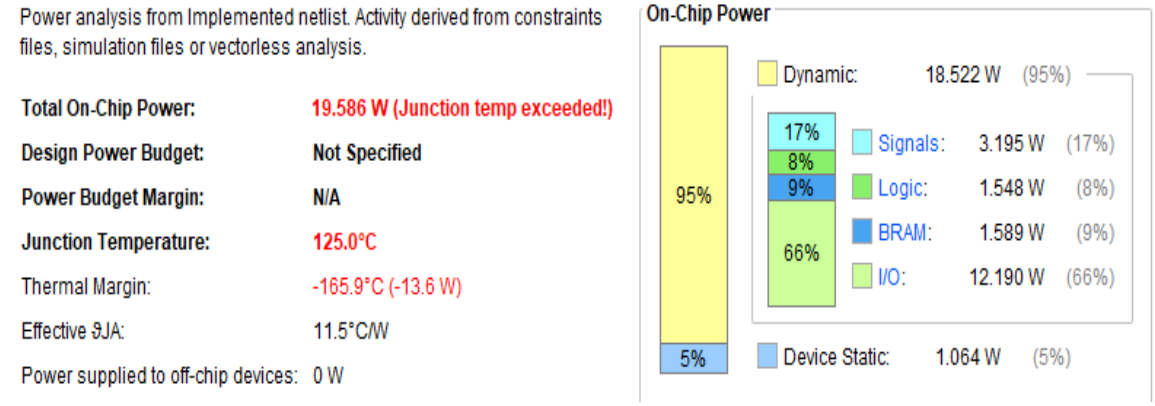


Figure 5.6 Power Summary Implementation

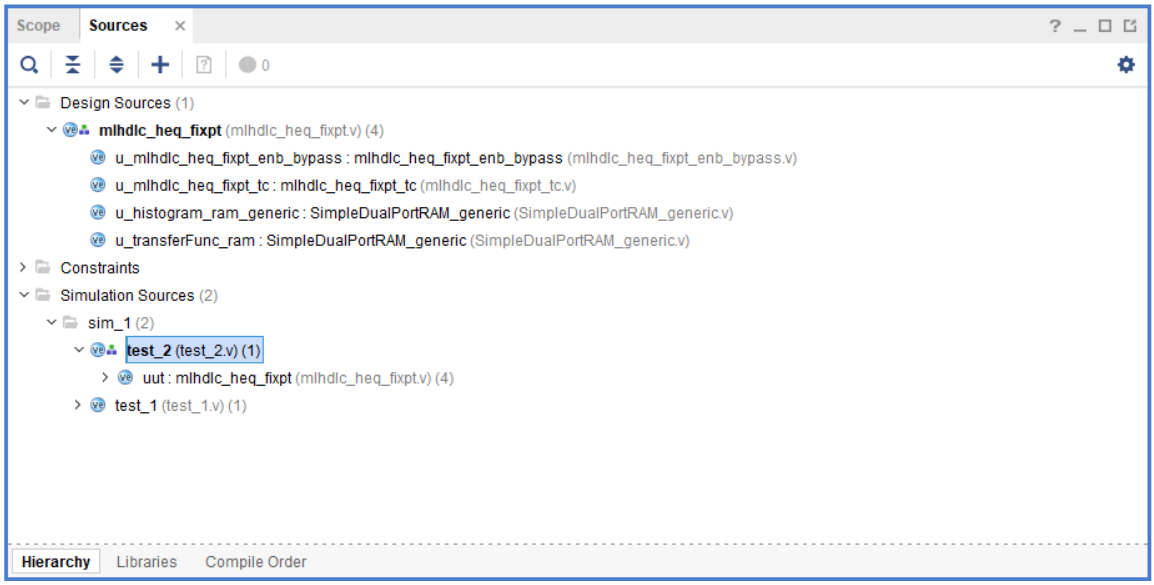


Figure 5.7 Sources in Vivado

Timing Check	Count ↓ 1	Worst Severity
unconstrained_internal_endpoints	1409	High
no_clock	349	High
no_output_delay	70	High
no_input_delay	46	High
constant_clock	0	
pulse_width_clock	0	
multiple_clock	0	
generated_clocks	0	
loops	0	
loops	0	
partial_input_delay	0	
partial_output_delay	0	
latch_loops	0	

Figure 5.8 Timing Analysis

clk	0	Logic
reset	0	Logic
clk_enable	1	Logic
x_in[8:0]	1ff	Array
y_in[8:0]	09f	Array
pixel_in[7:0]	a6	Array
width[8:0]	1ff	Array
height[8:0]	09f	Array
ce_out	1	Logic
x_out[8:0]	1ff	Array
y_out[8:0]	09f	Array
pixel_out[8:0]	11f	Array
original_pi...	118	Array
result_piq[...	112	Array

Figure 5.9 Inputs and outputs

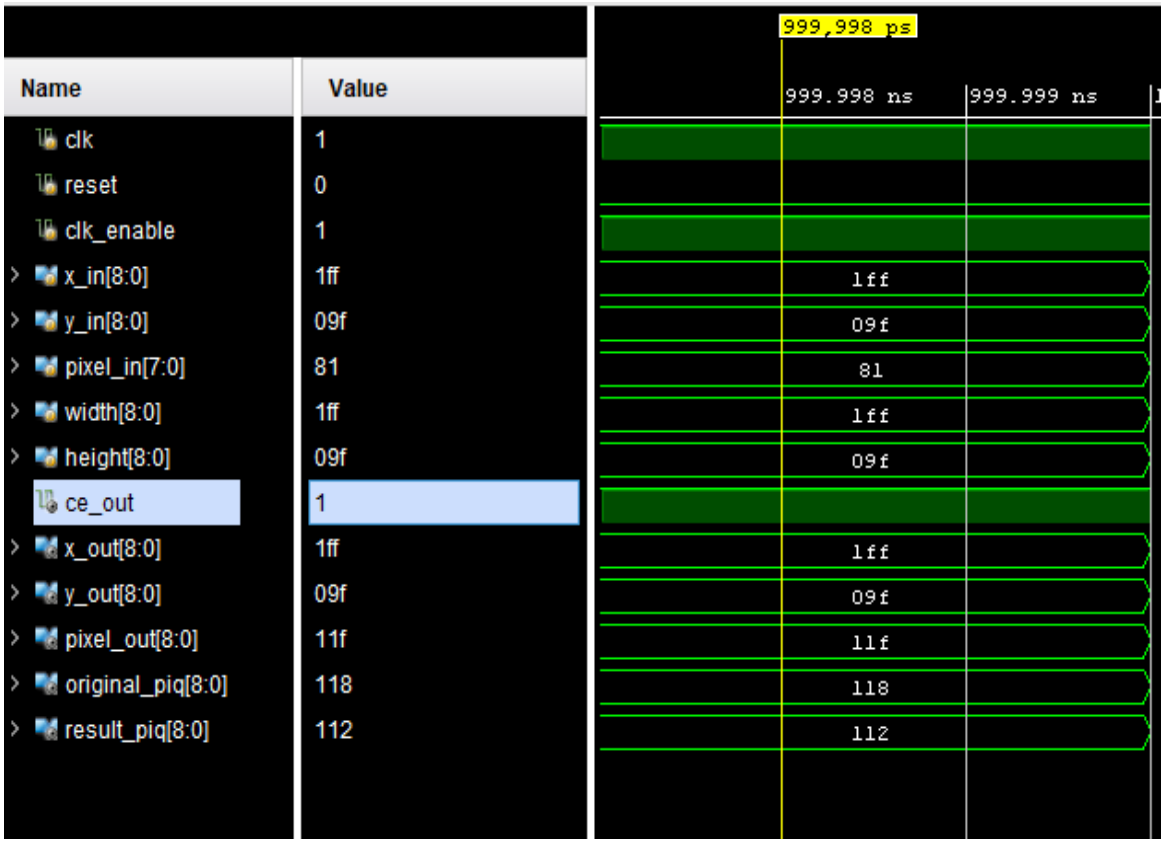


Figure 5.10 Output Waveform

```
Original image PIQ value =000000000,Resultant image PIQ calculated =000000000
Original image PIQ value =111111001,Resultant image PIQ calculated =111110011
Original image PIQ value =010010111,Resultant image PIQ calculated =010010001
Original image PIQ value =110000001,Resultant image PIQ calculated =101111011
Original image PIQ value =011000011,Resultant image PIQ calculated =010111101
Original image PIQ value =101100000,Resultant image PIQ calculated =101011010
Original image PIQ value =011111111,Resultant image PIQ calculated =011111001
Original image PIQ value =011101100,Resultant image PIQ calculated =011100110
Original image PIQ value =100011000,Resultant image PIQ calculated =100010010
```

Figure 5.11 Input and Output PIQE values

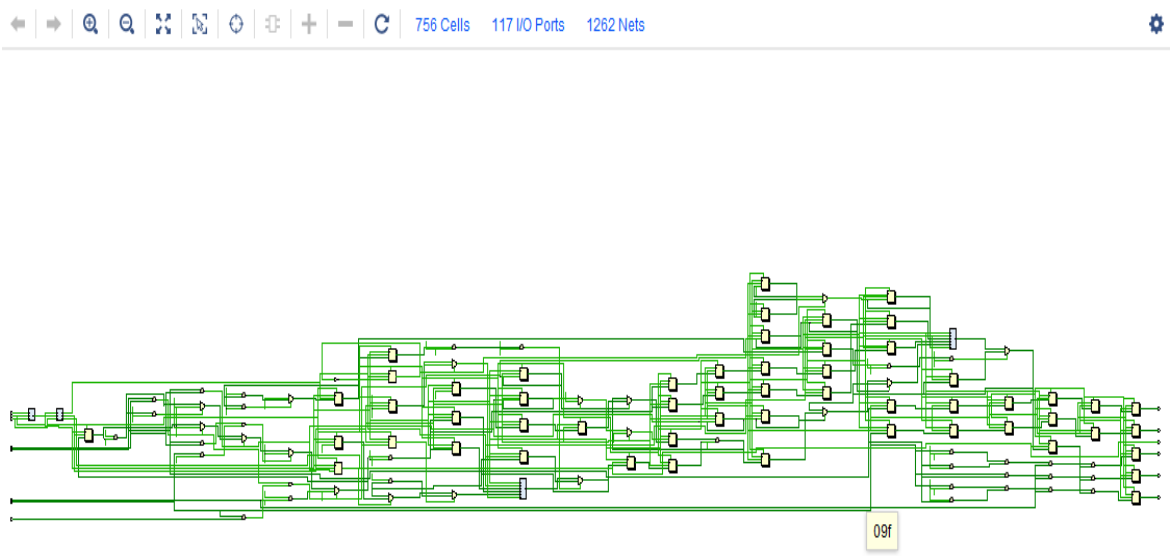


Figure 5.12 RTL Schematic

	# nets
# of logical nets	991
# of nets not needing routing	394
# of internally routed nets	394
# of routable nets	597
# of fully routed nets	597
# of nets with routing errors	0

Figure 5.13 Design Route Status

Site Type	Used	Fixed	Available	Util%
Block RAM Tile	14	0	140	10.00
RAMB36/FIFO*	13	0	140	9.29
RAMB36E1 only	13			
RAMB18	2	0	280	0.71
RAMB18E1 only	2			

Figure 5.14 Memory Utilisation

6. SUMMARY

In order to improve the satellite image, we wrote the code for the project titled "Satellite Image Enhancement using FPGA" in MATLAB. The PIQE (perception-based image quality evaluator) metric has been utilised to assess how enhanced our image is. Since the PIQE metric values of the original and improved photos differed by at least four points, with the enhanced image's PIQE metric being 4 points lower than the original image, we have come to the conclusion that our image has been enhanced. Discrete Wavelet Transform (DWT) and Singular Value Decomposition (SVD) were utilised for this image enhancement.

The proposed methodology uses HDL Coder to extract Verilog code for FPGAs from MATLAB. On vivado, the created code is executed. PIQE is the metric in use. The input image is likewise provided as binary values, and the output generated here has binary values that completely depict an image. According to the aforementioned methods, the improved image value is significantly lower than the original image when the binary values in the output image are converted to numeric values.

The test bench uses binary input pixels to check the functioning of the created code. The ZedBoard is assumed to be the FPGA used to run the code virtually. PIQE values are extracted from the output. The observed PIQE values of the produced image are lower than those of the original images when the PIQE values are observed in binary format.

We get an RTL schematic. The tool provides reports on utilisation, power analysis, and timing analysis. These reports include information on the individual parts of the FPGA, the power they use, the timing delays, and the input and output ports.

7. CONCLUSION AND FUTURE SCOPE

In this study, we investigated the use of FPGA for satellite image enhancement using hybrid transformations. We have seen how hybrid transformations can improve results by combining the advantages of spatial and spectral transformations.

MATLAB was used to construct and test the efficacy of various hybrid transformations. We have seen that MATLAB has a variety of functions and tools that make it simple to implement and analyse various types of transformations. MATLAB also includes measures like PIQE to analyse the performance of various transforms.

FPGAs were utilized to accelerate satellite picture processing. When compared to typical CPU-based systems, FPGA-based systems can deliver faster processing speeds. FPGA-based systems can also be tailored to implement specific algorithms, making them ideal for image processing jobs like satellite image augmentation.

The ability to process massive amounts of data in real-time is the primary advantage of employing an FPGA for satellite picture augmentation. FPGA-based systems can also be tailored to match specific performance requirements, making them appropriate for a variety of satellite image augmentation jobs.

The use of MATLAB and FPGA together can result in efficient and effective satellite picture enhancement solutions. By maintaining both spatial and spectral information, hybrid transformations can deliver better enhancement results. FPGA-based systems can deliver quicker processing speeds and can be tailored to individual performance needs. This experiment demonstrated the potential of using FPGAs for satellite picture improvement, opening up new paths for future research in this sector.

Optimization of FPGA-based systems:

FPGA-based systems can be adjusted to boost performance and reduce power consumption. Future research can concentrate on establishing novel optimization approaches and algorithms to improve the efficiency of FPGA-based systems.

Real-time processing of satellite images:

Many applications, such as disaster management and weather forecasting, require the real-time processing of satellite pictures. Future research could concentrate on developing FPGA-based systems that can process satellite photos in real time.

Integration of machine learning:

To improve the accuracy and efficiency of satellite image augmentation, machine learning approaches can be linked with FPGA-based systems. Future research could concentrate on constructing machine learning models that can learn to autonomously conduct satellite image enhancement tasks.

Integration with other technologies:

To improve the performance of other technologies, such as drones and IoT devices, the FPGA-based system can be incorporated. For example, an FPGA-based system can be utilized to improve the quality of real-time photographs captured by a drone.

REFERENCES

1. Muna F. Al-samaraie ―Color Satellites Image Enhancement Using Wavelet and Threshold Decomposition‖ IJCSI, 8(5).
2. Bin Wang D, Rose M and Aly A Farag, ―Local Estimation of Gaussian-Based Edge Enhancement Filters Using Fourier Analysis‖, IEEE Transactions on Acoustics, Speech, and Signal Processing,5,13-16(1993).
3. G. M. Hemes, S. Danaher, and A. Murray, ―Characterization of Forestry Species - A Comparison Using Singular Value Decomposition (SVD) and Artificial Neural Networks (ANN)‖, Proceeding of IEEE Conference on image Processing and its Applications,815-819(1995)
4. Day-Fann Shen, Chui-Wen Chiu and Pon-Jay Huang, ―Modified Laplacian Filter and Intensity Correction Technique for Image Resolution Enhancement‖, IEEE International Conference on Multimedia and Expo, 7(9), pp. 457-460(2006)
5. Arun R, Madhu S. Nair, R. Vrinthavani and Rao Tatavarti. ―An Alpha Rooting Based Hybrid Technique for Image Enhancement‖. Online publication in IAENG, 24th August 2011
6. Agaian, SOS S., Blair Silver, Karen A. Panetta, ―Transform Coefficient Histogram-Based Image Enhancement Algorithms Using Contrast Entropy‖, IEEE Transaction on Image Processing, 16(3), (2007).
7. Jinshan Tang Eli Peli, and Scott Acton, ―Image Enhancement Using a Contrast Measure in the Compressed Domain‖, IEEE Signal processing Letters,10(10),(2003).
8. R. C. Gonzalez, and R. E. Woods, ―Digital Image Processing‖, Englewood Cliffs, NJ: Prentice-Hall (2007)
9. H. Demirel, G. Anbarjafari, and M. N. S. Jahromi, ―Image Equalization Based On Singular Value Decomposition‖, Proceeding of IEEE Conference on Computer and Information Sciences,1-5(2008).
10. H. Demirel, C. Ozcinar, and G. Anbarjafari, ―Satellite Image Contrast Enhancement Using Discrete Wavelet Transform and Singular Value Decomposition‖, IEEE Geosciences and Remote Sensing Letters,7(2),pp. 333-337(2008).

11. P. S. Murty, and K.P. Rajesh, ―A Robust Digital Image Watermarking Scheme Using Hybrid DWT-DCT-SVD Technique‖, International Journal of Computer Science and Network Security,10(1),pp. 185-192(2010).
12. Hasan Demirel and Gholamreza Anbarjafari,―IMAGE Resolution Enhancement by Using Discreteand Stationary Wavelet Decomposition‖ IEEE transactions on IMAGE PROCESSING,20(5).
13. H. Demirel and G. Anbarjafari, ―Satellite image resolution enhancement using complex wavelet transform,‖ IEEE Geosciences and Remote Sensing Letter,7(1), pp. 123– 126(2010).
14. Kirk Baker,‖ Singular Value Decomposition Tutorial‖(2005).
15. Y. Piao, I. Shin, and H. W. Park, ―Image resolution enhancement using inter-sub-band correlation in wavelet domain,‖ in Proc. Int. Conf. Image Process.,1,pp. I-445– 448(2007)
16. J. W. Wang and W. Y. Chen, __Eye detection based on head contour geometry and wavelet subband projection,‟ Opt. Eng.,45(5) pp. 057001-1--057001-12(2006).
17. E. Hall, ―Almost Uniform Distributions for Computer Image Enhancement‖, Computers, IEEE Transactions, C-23, 207- 208(1974).
18. H. Keshavan, M. Srinath, ―Interpolative models in restoration and enhancement of noisy images‖, Acoustics, Speech and Signal Processing,25,525- 534(1977).
19. J. Lee, ―Digital Image Enhancement and Noise Filtering by Use of Local Statistics‖, Pattern Analysis and Machine Intelligence, Volume: PAMI-2,165-168(1980).
20. Algazi, ―FIR Anisotropic filters for image enhancement‖, Acoustics, Speech, and Signal Processing,11,2471-2474(1986).
21. D. Keren, S. Peleg and R. Brada, ―Image sequence enhancement using sub-pixel displacements‖, Computer Vision and Pattern Recognition,742-746(1988).
22. R. Malladi, J. A. Sethian, ―A unified approach to noise removal, image enhancement, and shape recovery‖, Image Processing, 5, 1554-1568(1996).
23. Fan, F. S. Cohen, and M. A. Patel, ―Classification of rotated and scaled textured images using gaussian markov random field models,‖ IEEE Pattern Anal. Mach. Intell.13(2),192–202(1991).
24. J. Zhang and T. Tan, ―Affine invariant texture signatures,‖ in Proc. Int. Conf. Image Processing, Thessaloniki, Greece,02,618–621(2001).

APPENDIX

MATLAB code:

```
a=imread('proj.jpeg');
b=rgb2gray(a);
c=histeq(b);

[LL,LH,HL,HH]=dwt2(b,'haar');
[LL1,LH1,HL1,HH1]=dwt2(c,'haar');

%svd
[U,S,V]=svd(LL);
[U1,S1,V1]=svd(LL1);

%new LL band
cor=max(S)/max(S1);

LL2=15*cor*LL;

figure()
subplot(2,2,1)
imshow(LL,[])
title('LL');
subplot(2,2,2)
imshow(LH,[])
title('LH');
subplot(2,2,3)
imshow(HL,[])
title('HL');
subplot(2,2,4)
```

```
imshow(HH,[])
title('HH');


%idwt
out=idwt2(LL2,LH,HL,HH,'haar');

figure()
subplot(1,2,1)
imshow(b,[])
title('Original image');
subplot(1,2,2)
imshow(out,[])
title('Enhanced image');

%psnr metric
noisy = imnoise(b,'Gaussian',0,0.08);
psnr(b, noisy)
out=uint8(out);
psnr(noisy,out)

%piqe metric
piqe(b)
piqe(out)
```

Testbench Code:

```
module test_2;

        // Inputs
        reg clk;
        reg reset;
        reg clk_enable;
        reg [8:0] x_in;
        reg [8:0] y_in;
        reg [7:0] pixel_in;
        reg [8:0] width;
        reg [8:0] height;

        // Outputs
        wire ce_out;
        wire [8:0] x_out;
        wire [8:0] y_out;
        wire [16:0] pixel_out;
        wire [16:0] original_piq;
        wire [16:0] result_piq;

        // Instantiate the Unit Under Test (UUT)
        mlhdlc_heq_fixpt uut (
                .clk(clk),
                .reset(reset),
                .clk_enable(clk_enable),
                .x_in(x_in),
                .y_in(y_in),
                .pixel_in(pixel_in),
                .width(width),
                .height(height),
                .ce_out(ce_out),
                .x_out(x_out),
```

```
        .y_out(y_out),
        .pixel_out(pixel_out),
        .original_piq(original_piq),
        .result_piq(result_piq)
    );

    initial begin
        // Initialize Inputs
        clk = 0;
        reset = 0;
        clk_enable = 0;
        x_in = 0;
        y_in = 0;
        pixel_in = 0;
        width = 0;
        height = 0;
//final image generated with PIQ values based on noise ration and enhancement of the
image
$monitor("Original image PIQ value =%b,Resultant image PIQ calculated =%b",
original_piq, result_piq);

        // Wait 100 ns for global reset to finish
        // Wait 100 ns for global reset to finish
                #100;reset = 1;clk_enable = 0;x_in = 0;y_in = 0;pixel_in =
0;width = 0;height = 0;
    #100;reset = 0;clk_enable = 1;x_in = 0;y_in = 0;pixel_in = 0;width = 'h1FF;height
= 'h9F;
                #100 x_in='h1FF;y_in='h9F; #100; pixel_in=01100010;
#100; pixel_in=01101100;
#100; pixel_in=01101001;
#100; pixel_in=10001000;
#100; pixel_in=01110101;
#100; pixel_in=10000001;
#100; pixel_in=01011110;
```

#100; pixel_in=01110001;
#100; pixel_in=01101100;
#100; pixel_in=01111100;
#100; pixel_in=01110001;
#100; pixel_in=01111001;
#100; pixel_in=01101010;
#100; pixel_in=01110111;
#100; pixel_in=01110010;
#100; pixel_in=01110000;
#100; pixel_in=01110011;
#100; pixel_in=01111000;
#100; pixel_in=10000011;
#100; pixel_in=01110011;
#100; pixel_in=01111011;
#100; pixel_in=01111011;
#100; pixel_in=10010100;
#100; pixel_in=10010011;
#100; pixel_in=10100110;
#100; pixel_in=01111100;
#100; pixel_in=10010001;
#100; pixel_in=10011001;
#100; pixel_in=10111010;
#100; pixel_in=10101111;
#100; pixel_in=10110110;
#100; pixel_in=01011101;
#100; pixel_in=10011110;
#100; pixel_in=10101011;
#100; pixel_in=10100111;
#100; pixel_in=10110000;
#100; pixel_in=10011111;
#100; pixel_in=10100001;