



Design and Implementation of Content-Based Image Retrieval on FPGA card

Hasnae El Khoukhi¹, Youssef Filali¹, My Abdelouahed Sabri¹, Abdellah Aarab¹

¹Laboratoire d'Informatique, Signaux, Automatique et Cognitivisme (LISAC) Laboratory
Faculty of Sciences Dhar-Mahraz, Sidi Mohamed Ben Abdellah University, Morocco, Fez
hasnae.elkhouchi@usmba.ac.ma

ABSTRACT

In recent years, real-time Image Processing (IP) has seen some applications and has posed a fast-growing field of programmable gate arrays (FPGA) implementation that has been the essential subject of the current engineering trend their simplicity and immense potential. One of the main problems of some IP is image extraction, which is that you can retrieve valuable knowledge about the quality of the images from the data effectively. We define the implementation of CBIR in FPGA image processing applications. The fundamental goal behind this initiative is to meet the shortest timeline by the optimum use of services. We are undertaking a review of this embedded approach's benefits and discussing its usefulness with a series of benchmark images. Implementation based on the FPGA indicates the viability of the suggested technique. The practical implementation of all processes using the FPGA Verilog code has been compiled into the ALTERA QUARTUS-II software method. The results obtained show qualitatively and quantitatively the feasibility of the proposed procedure.

Key words: image processing, feature extraction, CBIR, Verilog, FPGA.

1. INTRODUCTION

In the recent past, data must be processed and retrieved more efficiently, particularly in the real-time image processing (IP) area, which has a wide variety of applications and has earned researchers' attention. Further, DI sensor technology and storage devices have increasingly increased in the scale of databases due to the latest innovation in communication systems. The exponential growth of digital evidence has resulted in richer explanations and a powerful recovery method. The content-based image retrieval system (CBIR) has played a significant role in the IR domain to establish this. IR is the Retrieval of images from an extensive archive to scan and retrieve. When the pictures get complicated, it becomes difficult to locate the right images [1]. In the late 1970s, early work can be traced to this area. Since then, researchers have

been drawn to the implementation potential of image database management techniques [2]. There were growing problems with text-based rehabilitation. An urgent issue was the efficient handling of quickly growing visual data.

Content-based imagery retrieval (CBIR) is a non-conventional IR technology with a new layer in the field of DI processing and a big challenge with any IP. It uses visual information in the form of an input test image called a Query Image (QI), to request images from broad image databases based on consumer preferences. CBIR has made considerable strides in the fundamental study and device growth since the 1990s and over the last decade [2].

The visual material is mainly color, form, and texture characteristics and portrays low-grade aspects taken from the image. The Feature Vector FV is very significant since the pictures are retrieved on this attribute. This FV is very critical. The collection of features (FS) and the removal of features (FE) are a central feature in the IR process and are essential for the Retrieval of images (RS) [3]. The RS relies on the picture description's accuracy, which is mostly based on the FE image [4].

Due to several reasons like the extensive image-represented data set and the dynamic operations that may have to be performed on the series processor, it isn't easy to achieve real-time IP. Traditionally the IP has been used for similar design and pipeline processors. However, photographs and videos with a high-resolution, limited delay have proved unsuccessful. Their valuable power output characteristics are also gaining popularity thanks to application-specific hardware accelerators with reconfigurable content like Field Programmable Logic Arrays (FPGA).

In IP algorithms, parallelism is observed in two primary forms: parallel spatial and temporal. A combination of these two types will allow FPGA deployments to be parallel. In reality, the system's operation mode and hardware constraints are subject to this parallelization [5],[6].

IP algorithms are usually categorized into one of three levels: medium, moderate, or high. Low-level algorithms run on various pixels or districts. Middle-level algorithms either transform or work on one of the higher representations pixel data into a particular representation such as the histogram,

co-ordinate, or chain string. High-level algorithms are intended to extract image significance using other level information. The rejection of a part can take the form of recognizing an object in an image. If the representations are shifted from low to high, the exploitable parallelism is decreased accordingly because of the pixel data transition to descriptive words. However, the data volume to be analyzed is limited, allowing more time to complete the collection. Thanks to its nature, FPGAs are ideal for compute-intensive tasks, which constitute the vast majority of low- and mid-level operations. Big data sets should be used, and processes are frequently repeated [7].

Currently, FPGA is progressively used in research and development of applied computer applications for particular tasks. This is because FPGAs have benefits over other programmable machines. These benefits include fast clock rate, high second operations, portability, reusability of code libraries, low cost, parallel processing, and the ability to communicate with high and low interfaces, protection and IP, and others.

This paper is mainly assisted by the architecture, deployment, and adaption of FPGA-based CBIR hardware (CBIRH). Indeed, we plan to evaluate the efficacy of CBIRH's proposed method, which is used with ALTERA DE2 FPGA in the CYCLONE II FPGA system. For a collection of benchmark images of 488 pictures, we performed an experimental analysis. The pictures close to the QI of the plant are collected from the database image selection in this article. Here is a compilation of 488 images, including cars and vessels, from which QI is successively recovered dithered images. A hardware implementation of the FPGA-based CBIR approach has also been tested. The CBIRH provides good results as an advanced tool.

This paper is mainly assisted by the architecture, deployment, and adaption of FPGA-based CBIR hardware (CBIRH). Indeed, we plan to evaluate the efficacy of CBIRH's proposed method, which is used with ALTERA DE2 FPGA in the CYCLONE II FPGA system. For a collection of benchmark images of 488 pictures, we performed an experimental analysis. The pictures close to the plant's QI are collected from the database image selection in this article. Here is a compilation of 488 images, including cars and vessels, from which QI is successively recovered dithered images. A hardware implementation of the FPGA-based CBIR approach has also been tested. The CBIRH provides good results as an advanced tool.

Most of the open literature designs only display the code in VHDL, Verilog, and more, but there is no graphical explanation. The description of Block Diagrams in this work helps us introduce a CBIR driver's hardware description language quickly.

The remainder of the paper is structured accordingly. The overview of FPGA is connected to Section 2, the CBIR summary of Section 3, and the suggested embedded CBIR is

included in Section 4. The outcomes of the implementation are also given. Finally, the thesis and some potential paths are presented in Section 5.

2. OVERVIEW OF FPGA

Many IP applications need several operations on each pixel in the image to achieve even significant amounts of operations per second. In recent years, the continuous growth of the scale and functionality of Field Programmable Gate Array (FPGA) has led to a growing emphasis on their use as IP application deployment platforms [8], [23].

For computer-vision applications, FPGAs were launched in 1999 and are fundamentally similar devices that enforce the requisite logic for a specific architecture construction with separate hardware blocks for each purpose [9]. They are also configurable devices that allow the hardware to be configured after development.

An FPGA consists of a logic blocks and the network switches are reprogrammable, meaning that unique hardware can be built and that the device's configuration can be modified quickly. FPGA provides a balance between the performance and hardware speed of general-purpose processors. The underlying parallelism of computer hardware is used to make output gains. FPGA could be programmed to partition the picture, and the resulting parts could be spread into several pipelines that could simultaneously process data [10], [11].

Our research attempts to use the FPGA-based hardware implementation's success to build a new CBIR process architecture by inserting it into an FPGA platform.

3. OVERVIEW OF THE CBIR METHOD

CBIR programs are used to extract images from the image archive, based on the various image features. Photos can be obtained by uploading an input query image or a drawing of figures to the device. The general CBIR workflow was seen in Figure 1. Two key steps are demonstrated, the "indexing" step and the "searching or retrieval" step. The selection of features for all photos in the database is carried out in the "indexing" process. However, in the "scan" process, a distinction is made between the vector features of the requested image and those of the other pictures. Based on a practical similarity/distance calculation, the images are rated by their significance or similarity to the query image.

Also, an opportunity to include human insight and then replicate the search for this new information could enhance the results obtained.

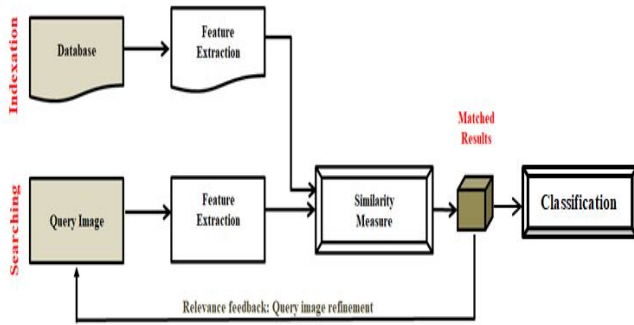


Figure 1: CBIR workflow

The elements derived (FE) from the lesion comprise bone, structure, form, and color attributes. The Retrieval of appropriate images from the image is the essential step by which the QI and the database images can be compared.

In FE, it extracts the QI-set function commonly known as FV or the image signatures that accurately depict each file's contents in the database. More precisely, it is about removing visual information from an image and storing it as an FV in a database of features. The FE considers image information in the form of a feature value (or a value set corresponding to FV) for each pixel. These FVs are used to equate the QI with the other images and to recover them. The frequently used visual features of the picture in the IR process are:

- **Texture:** A collection of statistical texture descriptors has been used to calculate the lesion's texture. The ten features extracted are: Contrast; Correlation; Energy; Homogeneity; Entropy; Inverse Moment; Smoothness; Regular Derivation; Kurtosis; Root Mean Square.
- **Color:** The goal is to measure the color intensity of the lesion. The derived characteristics are maximum, minimum mean, and pixel intensity variation within the R, G, and B range for 12 features.
- **Shape:** The lesion structure plays a significant part in identifying the type of lesion used by the dermatologist. The six features derived from the lesion are area, maximum and minimum diameter; size; perimeter; and eccentricity.

3.1 Techniques of color features

The quality of the image can be defined in terms of visual and semantic quality. The first focuses on color, texture, shape, relationship with space...etc. Whereas the second is achieved either by textual annotation or by dynamic visual material based inference procedures. Color is the most significant function since we can only recognize a small number of gray levels. Still, our eyes can discern thousands, and even millions of distinct colors can be seen on a computer [12].

A. RGB

Currently, colored pictures are typically in RGB (Red, Green, and blue) three-dimensional. RGB components color any image. In the database [13], [14], the target components' median value is measured and processed. These three mid-values are stored for each image and known as characteristics. Based on RGB elements' mean values, the photographs are classified into the classifications Red, Green, and blue. Many researchers in color-based RGB propose various color models, each aimed at promoting a specific mission or solving a particular problem [15].

B. HSV

HSV CS, also known as the color model of the hex cone. It is essential to calculate the HSV space portion to minimize computation and increase performance. The initial RGB image is transformed into an HSV image since the RGB color model is not ideal for representing colors in words that are realistic for human interpretation [16]. The HSV model describes the CS in terms of three elements. Hue reflects a type of color, such as red, blue, or yellow, varying from 0 to 360 degrees. Saturation is the vibrancy of color that ranges from 0 to 100%. Value (intensity) is the paint's brightness that varies from 0 to 100% [17].

For an image chosen from an image library, since the RGB CS does not fulfill people's visual specifications, the image usually transforms from RGB to other CSS. The HSV space is used in the paper as a more general CS.

C. Color Histogram

Color features are the most intuitive and straightforward IF features. It is also an essential aspect of vision compared to other IFs, such as texture and form. They are very stable, resilient, and not susceptible to shifts in rotation, translation, and size. Moreover, the estimation is reasonably simple [18]. A CI is the most commonly used means of collecting and presenting color information about an image and can be built in other CSS other than RGB, such as HSV. Color bins are displayed in histograms, and the total number of containers in RGB histograms can be calculated based on RGB planes. The CIs are the distribution of the strength of the color in the picture. They are a series of bins where each bin represents the likelihood that the pixels in the image are of a particular color. It acts as an accurate reflection of the color quality of the image if the color pattern is express relative to the rest of the data collection.

Also, the viewing axis's direction and rotation are long-lasting and vary only slowly with the distance, occlusion, and viewing angle. To calculate the CI, the CS must be split into many small groups. Any of the intervals is considered to be a container. The color is thus quantified. The global CI can

be calculated by counting the pixels where the colors are falling at each interval, as follows:

- Phase 1. Conversion of images from RGB to HSV one
- Phase 2. Quantification of images using the formula (1).

$$H = \begin{cases} 0 & h \in [316, 360] \\ 1 & h \in [1, 25] \\ 2 & h \in [26, 40] \\ 3 & h \in [41, 120] \\ 4 & h \in [121, 190] \\ 5 & h \in [191, 270] \\ 6 & h \in [271, 295] \\ 7 & h \in [295, 315] \end{cases} \quad S = \begin{cases} 0 & s \in [0, 0.2] \\ 1 & s \in [0.2, 0.7] \\ 2 & s \in [0.7, 1] \end{cases} \quad V = \begin{cases} 0 & v \in [0, 0.2] \\ 1 & v \in [0.2, 0.7] \\ 2 & v \in [0.7, 1] \end{cases} \quad (1)$$

- Phase 3. Calculation of the importance of each function
- Phase 4. Measurement of similarity of evaluation based on Euclidean distance via formula (2)

$$D = \sum_{i=1}^n (A_i - B_i)^2 \quad (2)$$

With, A and B corresponding to two FVs of dimension n.

D. Color Moments

CMs are used to separate images from their color characteristics. Also, they provide color similarity comparisons between images. The similarity value will then be compared to the image values indexed in the database for image retrieval tasks.

CM is used to address the quantization effect of CI for color distribution and is separated into three central moments: mean (μ), standard deviation (σ), and skewness [19]:

- Mean: Average image color value determined using the format (3):

$$\mu = \frac{1}{N} \sum_{i=1}^N P_i \quad (3)$$

Where, N is the total number of pixels in the image and P_i specifies its i^{th} pixel

- Standard deviation: the square root of the distribution variance, determined using a formula, is provided (4):

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (P_i - \mu)^2} \quad (4)$$

Whether $\mu = [\mu_H, \mu_S, \mu_V]^T$ and $\sigma = [\sigma_H, \sigma_S, \sigma_V]^T$. Each μ and σ portion provides the hue, saturation, and value detail, respectively.

CMs have been successfully used in many retrieval systems to distinguish objects based on color and be effective and efficient in representing color distributions of images, mainly when they contain only the item. CMs are based on the fact that color distribution in an image can be regarded as a distribution of probability, described by different moments. The FE from a large local window is in the form of statistical parameters like mean, variance, covariance, and kurtosis [22].

In our paper, we focus more specifically on mean and standard deviation.

3.2 Techniques of texture features

The top-ranked images are classified according to their texture features. It measures the different texture parameters such as entropy, contrast, auto-correlation, dissimilarity, standard deviation, mean, and variance of both QI and target images. The picture required from the measured values is extracted from the database. It is essential to find a collection of texture features with a robust discriminating capacity to develop an efficient and successful texture classification algorithm. Any of them are typically derived in a transformed domain by applying a local operator, statistical analysis, or calculation. In general, the characteristics are calculated from the matrices of co-occurrence [20].

Although GLCMs have a rich explanation of space dependency, they cannot be exploited raw. A set of 14 statistical descriptors or attributes is well known in the literature to quantify the textural details found in the GLCMs. The five most frequently found descriptors are homogeneity, correlation, contrast, entropy, and energy. This article reflects on the dimensions of entropy and homogeneity.

- Entropy: it measures the disorder of the image, based on the formula (5):

$$Entropy = \sum_{i=1}^N \sum_{j=1}^N P_{ij} (-\ln P_{ij}) \quad (5)$$

Where, P_{ij} is the occurrence probability of the gray level pair $\{i, j\}$, and N is the total number of pixels in the image.

- Homogeneity: high values are needed if the variations between pairs of pixels analyses are minimal and are more responsive to diagonal GLCM elements, whereas the contrast that is more reliant on distant diagonal elements. The formula (6) is used to calculate this:

$$Homogeneity = \sum_{i=1}^N \sum_{j=1}^N \frac{P_{ij}}{1+(i-j)^2} \quad (6)$$

3.3 Similarity Measurement

In this article, we use the Euclidean distance, which is the most common calculation for distance measurement between two vectors. The image feature vector components are similarly broad, and the feature vectors are independent of each other, so it is definable as the ordinary gap between two variables. Formally, the sum of the squares of the variations between vector components is given by the square root (see formula (2)). For a low degree of difficulty, this distance is easy to use. It provides an exemplary method for comparison of vector features for these purposes.

3.4 Classification

The classification device used for calculating our approach's classification rate is:

- **Vote:** We use to combine various classifiers with improving machine efficiency based on a particular class. Different combiner functions (e.g., average, operation, limit, and majority) are essential in this regard. The voting system only wants, given an object, a specific classifier for which it performs better for the output class [21].

We concentrate on our paper on ArgMax voting in particular. The ArgMax vote: This strategy takes as many arguments as possible of elements to maximize the function value and the output.

4. PROPOSED ADAPTIVE CBIR HARDWARE AND IMPLEMENTATION DETAILS

4.1 Proposed methodology

Generally, FPGAs are mostly used as tools for real-time image processing applications because their configuration can leverage spatial and temporal parallelism. The execution mode and the system's hardware limitations apply for this parallelization. The creator will restructure the algorithm with these limitations. This paper introduces a new CBIR technique architecture based on FPGA hardware, which we refer to as CBIRH (Hardware CBIR). In compliance with RGB-Grayscale and RGB-HSV CS transformations, the proposed hardware architecture has been developed.

Its implementation begins with hardware design, Verilog coding, logic synthesis, position and path, FPGA implementation, and VGA display of results on the monitor. The suggested solution's block diagram is based on color and texture characteristics, as seen in Figure 2.

In addition to Grayscale, we use the HSV CS in our solution. Hue (H) is used to differentiating the colors or corresponds to the actual color. The saturation(S) contains details about the proportion of white light that was applied to the pure color, and the value (V) refers to the perceived light level or gives the light intensity [24]. The mean (μ) of pixel colors provides details on the scene's primary color, while the standard deviation (σ) of pixel colors shows the variance of pixel colors.

A. Pre-processing

- Phase 1: Reading input image and creating image Matrix,
- Phase 2: Converting the images from RGB CS to HSV one, (RGB-HSV) where RGB (Red, Green, Blue) and HSV (Hue, Saturation, Value),

The digital hardware implementation of the translation of

RGB to HSV color space relies on the formula. (7) to (9). The Hue (H) portion of the HSV CS can be derived from the R, G, and B channels by formula (7):

$$H = \begin{cases} 0 + \frac{43 \times |G - B|}{\text{Max}(R, G, B) - \text{Min}(R, G, B)}, & \text{if Max}(R, G, B) = R \\ 85 + \frac{43 \times |B - R|}{\text{Max}(R, G, B) - \text{Min}(R, G, B)}, & \text{if Max}(R, G, B) = G \\ 171 + \frac{43 \times |R - G|}{\text{Max}(R, G, B) - \text{Min}(R, G, B)}, & \text{if Max}(R, G, B) = B \end{cases} \quad (7)$$

Where Max(R, G, B) is a sum of three Red, Green, and Blue pixels.

– The differential component determined by subtracting from a gray picture the minimum component of the full component is an indicator of the sum of gray in the hue.

By considering three cases, the tint is determined.

- The color of the hue ranged between 0 and 360 degrees and was between 0 and 255.

- The numerical value 85 refers to 120 degrees.

The numerical value 171 is 240 degrees, and 360 or 0 degrees is the numeric value 0.

- The channel has changed to green to 85 and the channel to blue to 171.

- The factor of 60 degrees was adjusted to 43 degrees numerical.

Using the formula (8), the Saturation (S) part of HSV CS can be obtained from R, G, and B channels:

$$S = \left\lfloor \frac{\text{Max}(R, G, B) - \text{Min}(R, G, B)}{\text{Max}(R, G, B)} \right\rfloor \quad (8)$$

- The value Max(R, G, B) represents the quantity of white in the image color.

- The Min(R, G, B) value represents the sum of black in the image color.

– The HSV CS component Value (V) is given by the formula (9):

$$V = \text{Max}(R, G, B) \quad (9)$$

Where Max(R, G, B) represents the value or luminance or brightness of the HSV.

The tint is no longer less than zero.

- Phase 3: Converting the images from RGB CS to Grayscale one (RGB-GRAY),

In the average method, we take the average of the three colors that is R, G, and B, to convert the image color into a grayscale image (RGB to GRAYSCALE).

The formula (10) to calculate the grayscale output is:

$$\text{GRAYSCALE} = \left\lfloor \frac{(R + G + B)}{3} \right\rfloor \quad (10)$$

Where(R+G+B) the sum of Three Red, Green, and Blue pixels matrixes.

B. Extraction of color and texture features(GLCM)

- Phase 4: Extracting the Mean value from RGB components

($\mu R, \mu G, \mu B$), using formula (3).

- Phase 5: Extracting the Standard deviation value from RGB components ($\sigma R, \sigma G, \sigma B$), using formula (4).
- Phase 6: Extracting the histogram value from HSV components (Hhist, Shist, Whist) and using formula (A.c.1).
- Phase 7: Extracting the Mean value from HSV components ($\mu H, \mu S, \mu V$), using formula (3).
- Phase 8: Extracting the Standard deviation value from HSV components ($\sigma H, \sigma S, \sigma V$), using formula (4).
- Phase 9: Extracting the Mean, Standard deviation, Entropy, and Homogeneity values from gray image components (gray_Mean, gray_Standard deviation, gray_Entropy, gray_Homogeneity),
- Phase 10: Result 1 = combine ($\mu R, \sigma R, \mu G, \sigma G, \mu B, \sigma B$),
- Phase 11: Result 2 = combine ($\mu H, \sigma H, \mu S, \sigma S, \mu V, \sigma V, Hhist, Shist, Whist$),
- Phase 12: Result 3 = combine (gray_Mean, gray_Standard deviation, gray_Entropy, gray_Homogeneity),
- Phase 13: Result = combine (Result 1, Result 2, Result 3),
- Phase 14: Counting each feature value,

C. Similarity comparison (RGBTOVGA)

- Phase 15: Finding Euclidean Distance between the QI and database images (see formula (2), for the application of the similarity measure),
- Phase 16: Applying the results, VGA monitor.

D. Classification (LCD1602)

- Phase 17: Applying the classification process using method Voting classification,
- Phase 18: Applying the results, LCD display.

The proposed CBIRH algorithm is an adaptive version of the Standard CBIR method, based on hardware implementation using ALTERA DE2 FPGA. The performed algorithm is constructed using the Verilog code has been compiled into the ALTERA QUARTUS-II software method. The algorithm, with its actual sizes, was evaluated on Database images of 488 images like cars and vessels. By introducing and analyzing the obtained data, we perform an experimental analysis of the applied algorithm's efficacy.

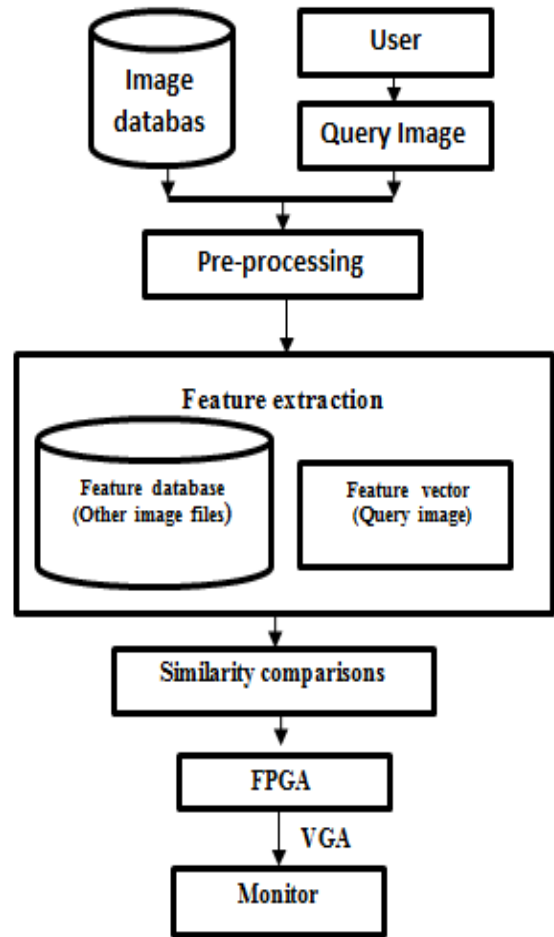


Figure 2: Block diagram of the proposed approach based on color and texture features

4.2 Experimental Results

In order to test the performance of the proposed architecture, statistical experimentation has been carried out on the Intel(R) Core(TM) PC i5-7200U Processor @ 2.50GHz 2.70GHz, 4 GB RAM, Windows 10 Technical. Statistical tests have been performed on the Intel(R) Core(TM) PC i5-7200U Processor @ 2.50GHz 2.70GHz, 4 GB RAM, Windows 10 Technical, to measure the performance of the proposed architecture.



(a)



(b)

Figure 3: A sample of Database images of tow categories (a) for cars and (b) for boats

Figure 4 resumes the corresponding Dataflow of the proposed design, and Figure 5 represents the Wave.

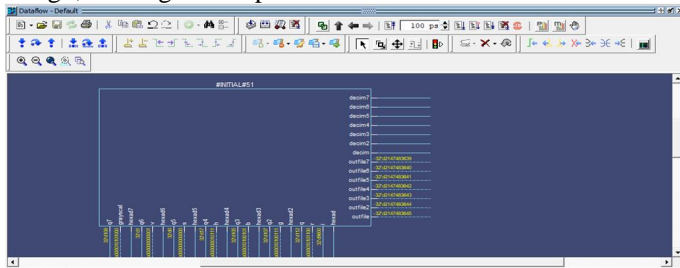


Figure 4: Data flow of the proposed design

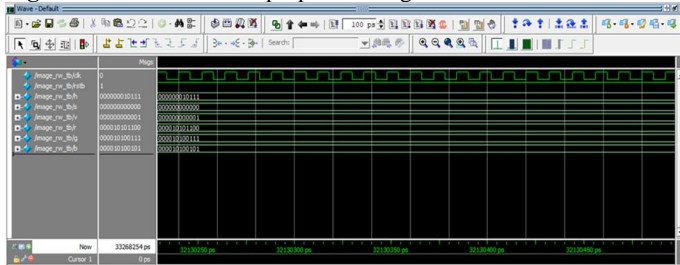


Figure 5: Wave of the proposed design

Figure 6 displays the highest signal graph for the RGB-HSV, RGB-GRAY, GLCM, RGB-VGA, and module LCD1602. The proposed embedded CBIR algorithm is a standard CBIR adaptive variant based on FPGA hardware implementation. The algorithm is used on a 488-pixel database pixel with actual sizes of the cars and vessels, with ALTERA DE2 FPGA as a CYCLONE II FPGA method. By introducing and analyzing the findings, we carry out an experimental analysis of the algorithm's usefulness.

The process "Indexing" was carried out in Matlab, and the function "Searching" was carried out on the FPGA card. Figure 7 displays the block diagram of the solution.

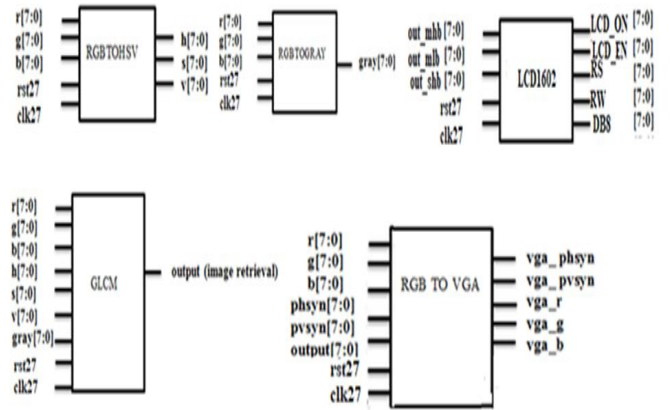


Figure 6: Signal Diagram of RGB to HSV(rgbtohsv.v), RGB to GRAY(rgbtogray.v), LCD1602(LCD.v), GLCM(GLCM.v), RGB toVGA(rgbtojpg.v) module .

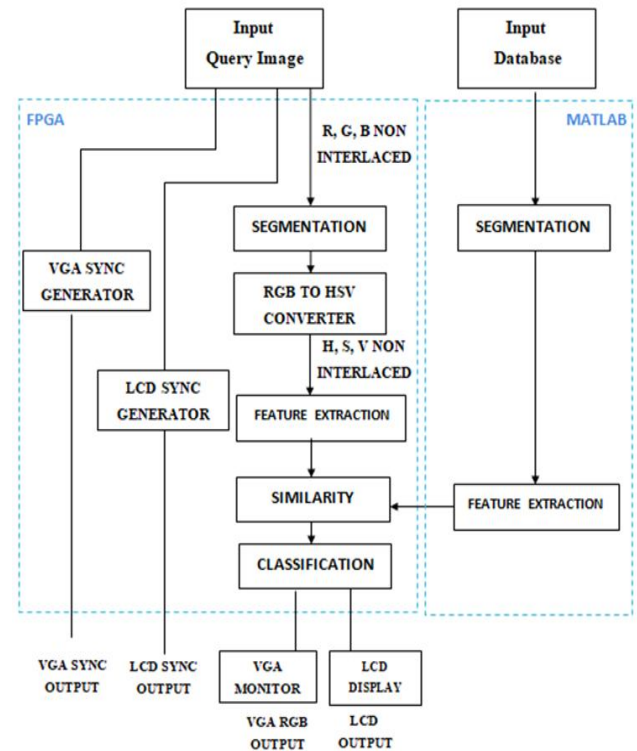


Figure 7: Global architecture of the proposed CBIRH algorithm (case of "car" Query Image)

Figure 8 shows the simulation output images displayed on the monitor via VGA and the effects of the view on the monitor via LCD for towing input "car" and input "boat" samples like question images. The results obtained well demonstrate that the proposed optimized GLCM hardware-based CBIR algorithm using FPGA improves.

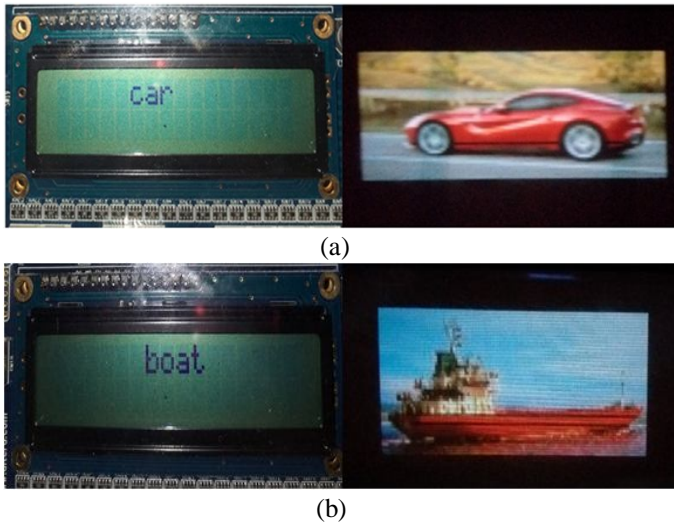


Figure 8: Output result displaying on the LCD for two samples of Input "car" (a) and Input "boat" (b)

5. CONCLUSION

FPGAs provide relatively broad capabilities for parallel processing. It is not mandatory to require such a high clock frequency of the digital logic for this reason, as is the case for a traditional device. Nonetheless, FPGAs with slightly lower frequency can be several times more potent than a widely used tool.

The key emphasis of this paper is the hardware implementation of the CBIR approach using FPGA. We also suggested a CBIRH algorithm, and this report also introduces a hardware implementation of the GLCM technique. We perform an experimental review and a qualitative review of the applied system's efficacy on ALTERA QUARTUS-II applications.

As future directions of this work, a significant part of progress in the considered database and the algorithm itself is feasible by incorporating new extraction techniques for applications. We also plan to suggest an enhanced CBIR algorithm tailored to the video surveillance case and create an associated application framework for FPGA hardware, taking complex and actual data into account.

REFERENCES

1. J.Kaur, M.Kumar, **REVIEW PAPER ON CONTENT BASED IMAGE RETRIEVAL FOR DIGITAL IMAGES**, INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS, Vol.3 Issue 9, Pg.: 108-112 September 2015.
2. A.khokher, R.Talwar, **Content-based Image Retrieval: Feature Extraction Techniques and Applications**, International Conference on Recent Advances and Future Trends in Information Technology (iRAFIT2012)2012.
3. K.Kumar, G.Kumar, Y.Kumar, **Feature selection approach for intrusion detection system**, International

- Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), 2013, 2(5), 47-53.
4. K. Shubhankar Reddy, K. Sreedhar, **Image Retrieval Techniques**, A Survey, International Journal of Electronics and Communication Engineering. ISSN 0974-2166 Volume 9, Number 1 (2016), pp. 19-27
5. K .C.Sudeep, J. Majumdar, **Textural A Novel Architecture for Real Time Implementation of Edge Detectors on FPGA**, International Journal of Computer Science(IJCSI), Issues, Vol. 8, Issue 1, January 2011.
6. J.Christopher, K.T. Gribbon, D.G.Bailey, **Implementing image processing algorithms on FPGAs**, 28-Feb-2018.
7. D.Feng, W.C. Siu, H.J.Zhang, **Multimedia Information Retrieval, and Management**, Springer Science & Business Media, 17 Avr. 2013 - 476 pages.
8. H. El Khoukhi, Y. Filali, A. Yahyaouy, M. A. Sabri and A. Aarab, **A hardware Implementation of OTSU Thresholding Method for Skin Cancer Image Segmentation**, 2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS), April 03-04, Fez, Morocco, 2019, pp. 1-5. doi: 10.1109/WITS.2019.8723815.
9. K.T. Gribbon, C.T. Johnston, D.G. Bailey, **A Real-time FPGA Implementation of a Barrel Distortion Correction Algorithm with Bilinear Interpolation**, Image and Vision Computing NZ, Palmerston North, November 2003.
10. S. Mittal, S. Gupta, S. Dasgupta, **FPGA: An Efficient And Promising Platform For Real-Time Image Processing Applications**, Proceedings of the National Conference on Research and Development in Hardware & Systems (CSI-RDHS 2008) June 20-21, 2008, Kolkata, India
11. H. El Khoukhi, F.M. Idriss, A. Yahyaouy, M.A., Sabri, **An Efficiency Study of Adaptive Median Filtering for Image Denoising, Based on a Hardware Implementation**, Advances in Intelligent Systems and Computing. 2020.
12. A. Mukherjee, K. Gaurav, A. Verma, H. Kumar, R. Thakur, **Content Based Image Retrieval using GLCM**, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 11, November 2016.
13. X.Wang, J. Wu, H. Yang, **Robust image retrieval based on the color histogram of local feature regions**. Multimed Tools Appl 49, 323–345 (2010).
14. V.Mohan, A.Kannan., **Color Image Classification, and Retrieval using Image mining Techniques**, International Journal of Engineering Science and Technology, Vol. 2(5), 2010, 1014-1020
15. J. Khan, S.S. Kulkarni, **Implementation of Content Base Image Retrieval Using Clustering Technique**, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.5 May- 2014, pg. 523-529
16. S.Ramadass, **An Efficient CBIR Technique with YUV Color Space and Texture Features**.2012.

17. P. Venkata Krishna, M. Rajasekhara Babu, E. Ariwa, **Global Trends in Information Systems and Software Applications**, 4th International Conference, ObCom 2011, Vellore, TN, India, December 9-11, 2011, Part II. Proceedings.
18. L. Wang, K. Chen, **Advances in Natural Computation: Pt. 2**, First International Conference, ICNC 2005, Changsha, China, August 27-29, 2005, Proceedings.
19. S. Mangijao Singh, K. Hemachandran, **Content-Based Image Retrieval using Color Moment and Gabor Texture Feature**, IJCSI International Journal of computing Issues, Vol. 9, Issue 5, No 1, September 2012.
20. B. Uma Shankar, K. Ghosh, D. Prasad Mandal, S. S.r Ray, D. Zhang, S. K. Pal, **Pattern Recognition and Machine Intelligence**, 7th International Conference, Premi 2017, Kolkata, India, December 5-8, 2017, Proceedings, Springer, 22 Nov. 2017 - 695 pages
21. P. Gangadhara Reddy, **Extraction of image features for an effective CBIR system**, Recent Advances in Space Technology Services and Climate Change 2010 (RSTS & CC-2010), Chennai, 2010, pp. 138-142, DOI: 10.1109/RSTSCC.2010.5712832.
22. S.S.Amiripalli, V.Bobba, (2019), **Trimet graph optimization (TGO) based methodology for scalability and survivability in wireless networks**, International Journal of Advanced Trends in Computer Science and Engineering, 8(6), 3454-3460.
23. B. Murali Krishna, K. Ch. Sri Kavya, P.V. S. Sai Kumar, K. Karthik, Y. Siva Nagababu, **FPGA Implementation of Image Cryptology using Reversible Logic Gates**, International Journal of Advanced Trends in Computer Science and Engineering, ISSN 2278-3091, Volume 9, No.3, May-June 2020.
24. M.Sharvari Waikar, K.B.Khanchandani, **Determination of Image Features for Content-based Image Retrieval using Interactive Genetic Algorithm**, International Journal of Computer Applications (0975 -8887) Volume 89 -No 17, March 2014