Fusion of near-infrared and RGB images on a FPGA using high level synthesis tool

Alex Noel Joseph Raj*, M. Murugappan** and Arunachalam V***

*Key Laboratory of Digital Signal and Image Processing of Guangdong Province, Department of Electronic Engineering, College of Engineering, Shantou University, Shantou-515063, China

**Department of Electronics and Communication Engineering, Kuwait College of Science and Technology, 13133, Kuwait

***Department of Micro and Nano Electronics,VIT University, Vellore, TamilNadu, India

**Corresponding Author: m.murugappan@gmail.com

Submitted: 02/12/2019
Revised: 25/01/2020
Accepted: 06/02/2020

ABSTRACT

Several applications utilizing a set of red green blue (RGB) and near infrared (NIR) images have been emerging over recent years. The present work proposes a technique of enhancing an image by combining color (RGB) and near infrared information (NIR). In order to fuse the two types of images, the NIR-channel is considered as a luminance counterpart to the visible image. International standard database (RGB-NIR Scene Dataset) is used in this work for image fusion. The objective of the paper is to present a simple and hardware efficient fusion method, where the original RGB image is converted into two different color spaces, namely, HSV and YCbCr. Later, the luminance channel of the RGB image is replaced with the near infrared channel, thereby obtaining a fused enhanced image. The above procedure is effectively implemented on FPGA using the Xilinx HLS tool. RGB-NIR dataset is used in the present work for testing the proposed image fusion algorithm, and the quality of the fused image is measured through peak signal to noise ratio (PSNR). The experimental results indicate that HSV color space is more efficient in image fusion compared to YCbCr color space based on the average PSNR values of approximately 29db for HSV and 25db for YCbCr for various images, respectively. Finally, this complete fusion algorithm is implemented on Xilinx Nexys4 FPGA board to be able to obtain real-time outputs in the form of vivid, contrasted images that are pleasing to the observers. The experimental results illustrate that the Xilinx FPGA utilizes only 50% of the available hardware resources and consumes approximately 5.3 Watts to implement the fusion process.

Keywords: Image fusion; NIR; HSV; FPGA; Xilinx Vivado HLS.

1. INTRODUCTION

Near-infrared images have properties that would provide useful information in images of various scenes, such as the increased contrast between sky and clouds, shadowed and non-shadowed areas, and the increased optical depth. The visible images, on the other hand, contain information such as color, hue, saturation, and brightness. A common use of near-infrared (NIR) imaging is to distinguish between surfaces that appear identical to the human eye. NIR is capable of penetrating through various paints and colorants to reveal the texture of the surface underneath. The reverse is also true; there are surfaces that can be distinguished in the visible spectrum that appear the same in the NIR. The fused result of RGB and NIR therefore would provide us a lot of significant information about the scene and its details. Such a combination also helps in better classification of different materials [1]. The FPGA implementation of such an image fusion would serve as a step towards implementing a system that would be able to simultaneously capture color and near infrared images of a scene, fuse them using the given design and display it based on the requirement of the application such a system would benefit several fields, which use near infrared spectroscopy such as blood alcohol content, forensic analysis, and analysis of progress of reaction and would also enhance the quality of color images.

In specific, Near Infrared spectroscopy finds application in a large number of fields including astronomical spectroscopy, remote monitoring, agriculture and medicine for the detection of tumors [2], etc. As the addition of color information to a near infrared image would help all these fields greatly, a considerable amount of work has been done in the field. A number of cameras have been designed, which simultaneously capture near infrared and RGB images and obtained a combined result. Some techniques modify existing cameras by replacing filters and adding additional sensors to capture a combined image or designing sensors that simultaneously capture a joint image [4, 5, 9, 10]. In [2], the authors interpolated the RGB image; highlight the high intensity pixels of the NIR image before overlapping the two images. This is useful for diagnosing tumors and its more application specific method and it cannot be used to enhance a scenic image [15].

In [3], the principle component analysis (PCA) and Fisher Vectors was used to encode texture or color local descriptors. This work combines visible and near infrared cues for the purpose of image categorization. The present work deals with a more generic approach to enhance an image, which is suitable for multiple applications rather than focusing on any specific application. We consider the NIR channel as the fourth color channel and swap it with one of the existing ones from the original image. In [4], the researchers investigated an image enhancement approach by considering the NIR channel as color, luminance, or frequency counterpart to the visible image. Considering the computational complexity, as well as quality of results obtained in [4], we consider NIR data as spatial information or lightness rather than color or frequency. Thus, it is better to use different color spaces (YCbCr and HSV) rather than the RGB color space so that it can be swapped with the near infrared channel. The key contributions of the paper are (a) compare the results obtained from HSV-based image fusion and those obtained from YCbCr-based image fusion and determine, which produces a better output in most cases and (b) FPGA implementation of the HSV-based fusion using Xilinx Vivado High Level Synthesis (HLS) tool.

The paper is organized as follows. Section 2 describes the fusion process, where the NIR an RGB images are fused after converting them into either HSV or YCbCr format. Later, a detailed comparison of the fusion process based on YCbCr and HSV color spaces and discussion about the FPGA implementation of the fusion process has been presented in Section 2. Section 3 presents the experimental results and discussion on the proposed design, and conclusion of the present work is given in Section 4.

2. METHODS AND MATERIALS

The main objective of image fusion process is to obtain a single image that contains rich information as much as possible from a pair of NIR and RGB images. The fusion process is illustrated in the block diagram shown in Figure 1.

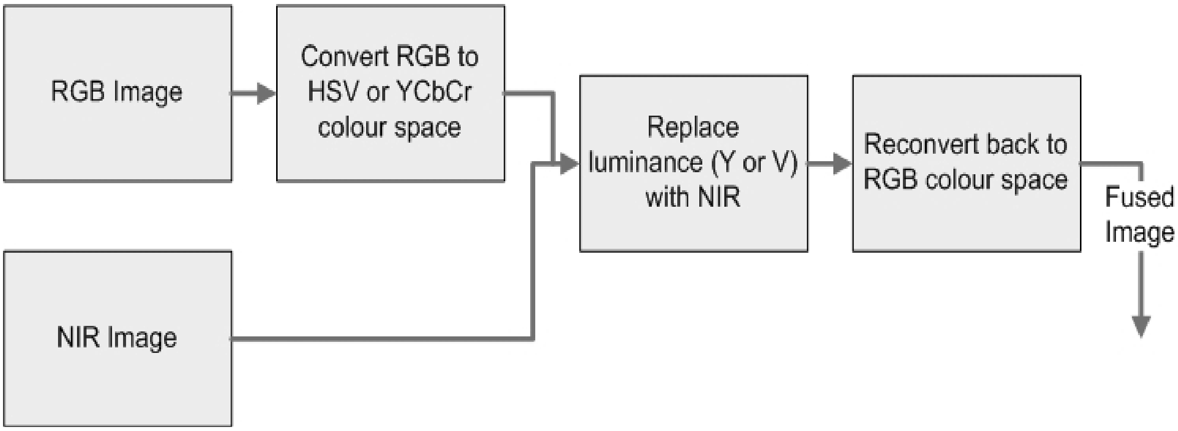

RGB Image
Convert RGB to HSV or YCbCr colour space
NIR Image
Replace luminance (Y or V) with NIR
Reconvert back to RGB colour space
Fused Image

Figure 1. Fusion process.

The steps involved in the design process can be broadly defined as follows:

(1) Obtain dataset of NIR and RGB images of the same scene for different settings.

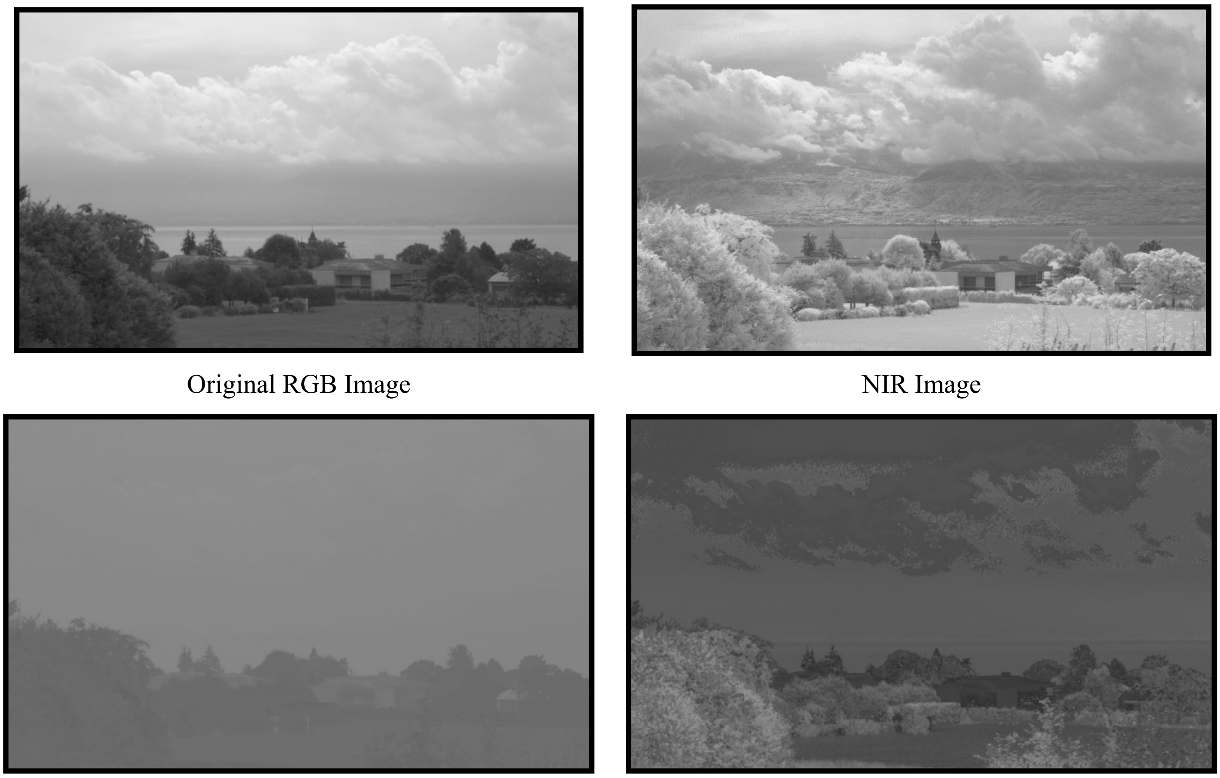
(2) Convert the RGB images into the corresponding YCbCr and HSV color spaces. A detailed comparison between the fusion process based on YCbCr and HSV color spaces is presented in section 2.1.

(3) Replace the luminance channels with the NIR channel.

(4) Reconvert the fused image into the RGB color space.

2.1. Conversion and Color Swap

A set of forty different images are randomly chosen from the selected dataset, which consists of 477 images in 9 categories captured in RGB and near-infrared (NIR) with a resolution of 1024 × 768. The images were captured using separate exposures from modified SLR cameras (Nicon D90 and Cannon Ti cameras with a cutoff between the visible and NIR filters of 750 nm), using visible and NIR filters. The acquired images are pixel-to-pixel registered and more detail about the dataset can be found in [5]. The scene categories are as follows: country, field, forest, indoor, mountain, old building, street, urban, water. MATLAB software (in specific image processing toolbox) was used to extract and save the red, blue and green values (RGB) of the chosen images. These values are then used to convert the images into the HSV and YCbCr color formats using the algorithm proposed in [6]. The mathematical concept behind the conversion process is presented in Appendix 1. A set of converted images are shown in Figure 2.



Original RGB Image

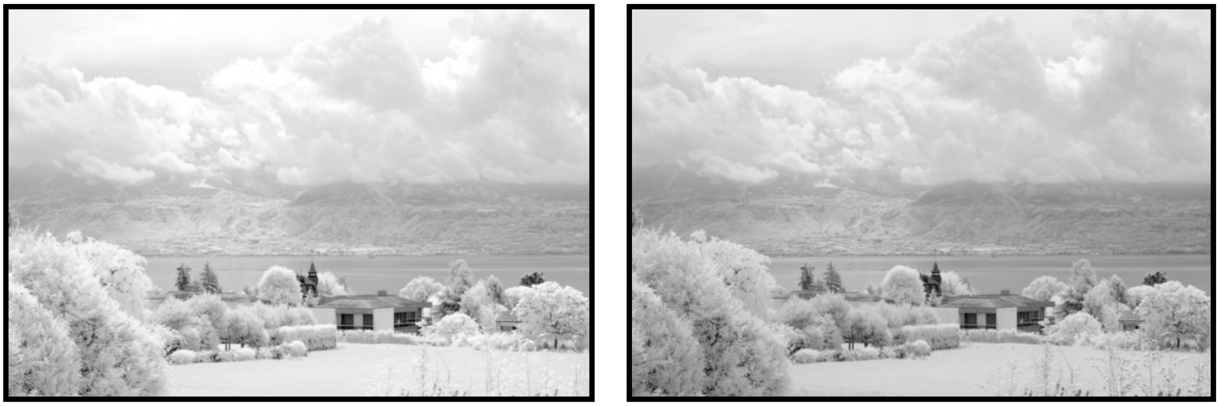
NIR Image

Convert from RGB to YCbCr

Convert from RGB to HSV

Figure 2. RGS, NIR, YCbCr, and HSV representations of a scene.

Later, the V channel of the HSV format and the Y channel of the YCbCr format are replaced with the NIR channel by separation followed by concatenation and reconverted back into the RGB color format, and the resultant image is displayed (Figure 3).



Fused Image (Using YCbCr) Fused Image (Using HSV)

Figure 3. Fused and reconverted images.

3. RESULTS AND DISCUSSION

To analyze and compare the two resultant fused images obtained from the two types of color formats, we first obtain the peak signal to noise ratio (PSNR) for the two images when compared with the original RGB image [17]. PSNR is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale compared to other performance measures. The PSNR values for the fused images were calculated (using equation 1) for the entire set of 40 images and it was found that the value of PSNR was consistently higher for the image fused using the HSV color model. This can be viewed in Figure 4.

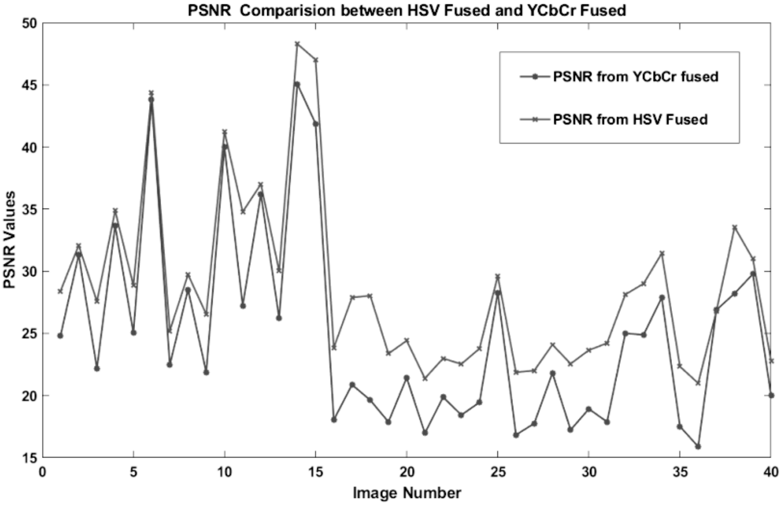


Figure 4. PSNR values of 40 images fused using HSV (green) and YCbCr (blue).

$$PSNR = 10log_{10}\left(\frac{R^2}{MSE}\right) \tag{1-1}$$

$$MSE = \frac{\sum_{M,N}[I_1(m,n)-I_2(m,n)]^2}{M\times N} \tag{1-2}$$

where M, N represent the image resolution, MSE, the mean square error computed between the original image I1 and fused image I2, and the R is 255, denoting the maximum grayscale value. To further analyze the quality of the resultant fused images, we used contour plots to present the depth of information of the fused images. The contour plot gives the height of an aspect of an image with respect to its x − y plane. The details present in the picture are outlined by the contour plot, and it is seen that the plot with the greater number and density of contour lines comes from a more detailed image. From Figure 5, it can be inferred that the images fused using YCbCr and HSV are far more detailed than the original RGB image.

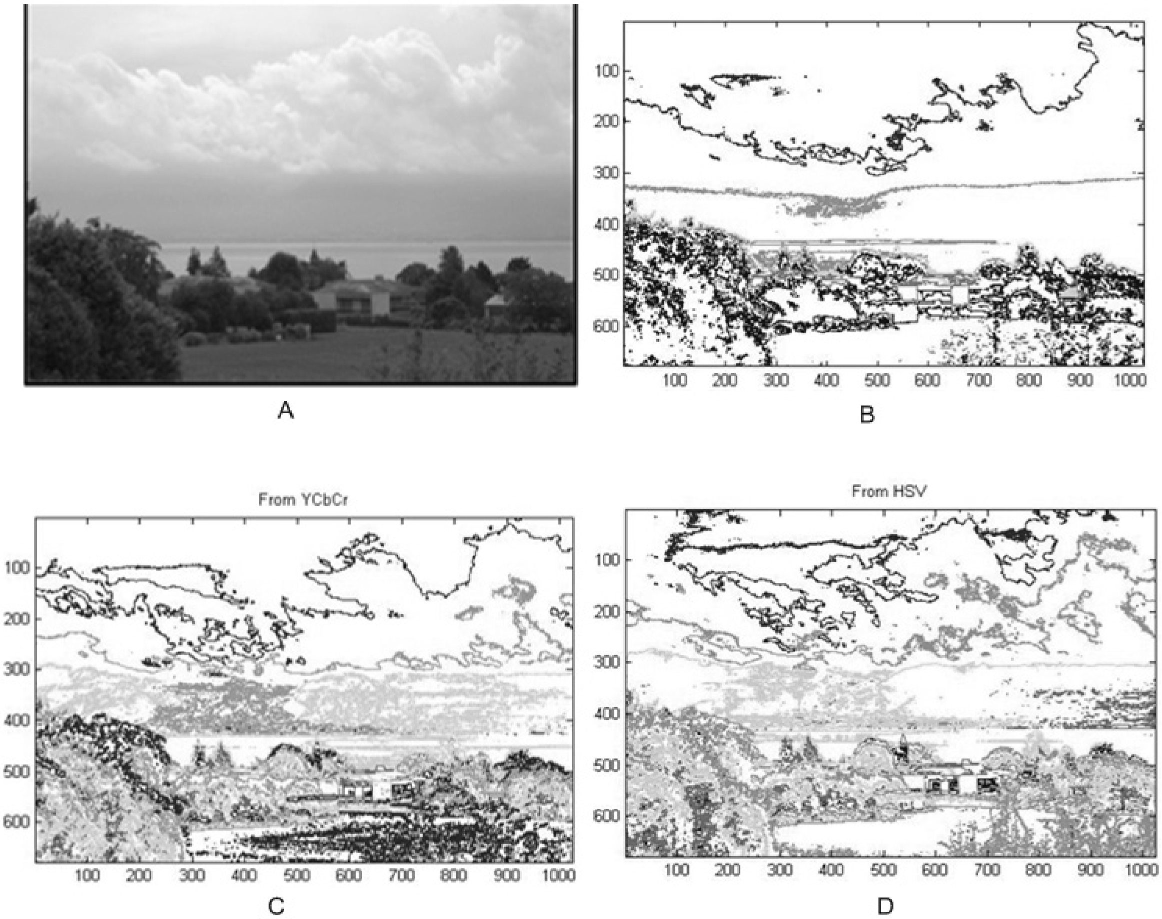


Figure 5. (Clockwise): Original RGB image (A), contour details from original RGB image (B), image fused using YCbCr (C) and image fused using HSV (D)

The above analysis was repeated for the entire set of images (40 numbers) under consideration, and it was found that, in most of the cases, the contour plots of the image fused using HSV provided better results than YCbCr. It can be said that, in general (refer to Figure 6), the image fused using the HSV color conversion shows better results. These images have a higher PSNR, have more details, and are, in general, more pleasing to the viewer and also preserve color information in a better fashion. Therefore, for the FPGA implementation, we will use this method to fuse the two types of images.

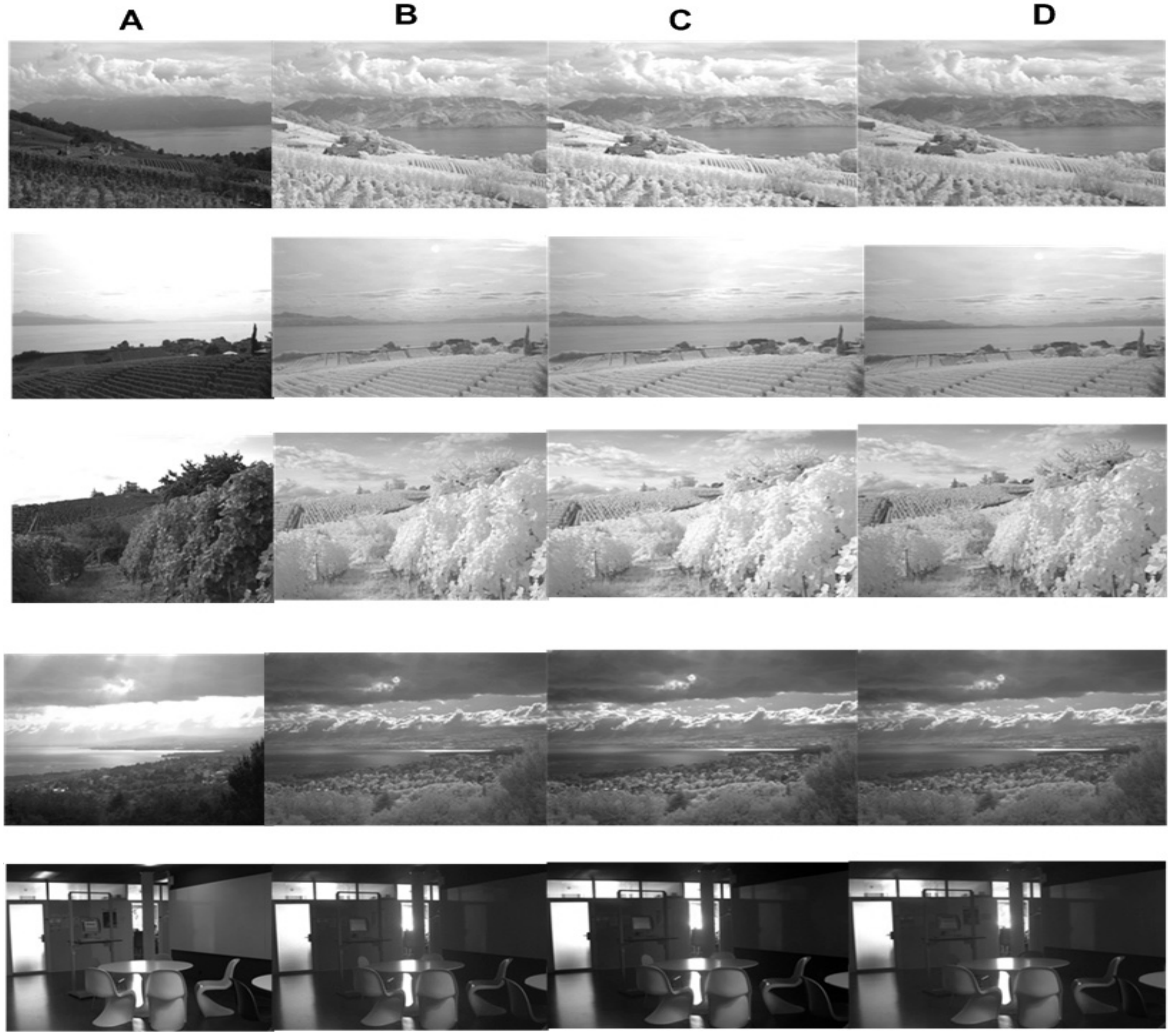


Figure 6. A: RGB image; B: NIR image; C: Image fused using YCbCr; D: Image fused using HSV.

3.1. FPGA Implementation

The second contribution is the implementation of the HSV based fusion algorithm on a Xilinx Nexys4 board, which includes an Artix 7 FPGA. Implementation of image fusion algorithm in FPGA could be useful while designing intelligent systems for different applications, and it replaces the requirement of desktop or personal computer for processing [16]. Also, FPGA performs the fusion process faster than PC with lesser power consumption. In this work, the fusion algorithm was coded using the Xilinx Vivado Design Suite (version 2019.2), which integrated both Xilinx ISE and Vivado HLS [High Level Synthesis tool]. The work includes a top level module, which calculates the H, S and V components from the RGB values, then replaces the V channel with the NIR channel, and converts the color space back into RGB.

The pixel values of RGB and NIR images are stored separately in four different files. Later, from Vivado Design Suite, the Block Memory Generator IP is used as ROMs that are initialized with the values of files. Thus, we generate four Block Memory IPs to store the values of R, G, B and NIR separately. For the experiments, we have used image size 120 x 96 pixels and width of each pixel is 8 bits and therefore the depth of each memory is 11520. The top module is shown in Figure 7 and the output from each block memory; doutr, doutg, doutb, doutnir provide the data of each color channel red, blue, green, and near-infrared, respectively. This data will be used for the color conversion, swap and subsequent reconversion.

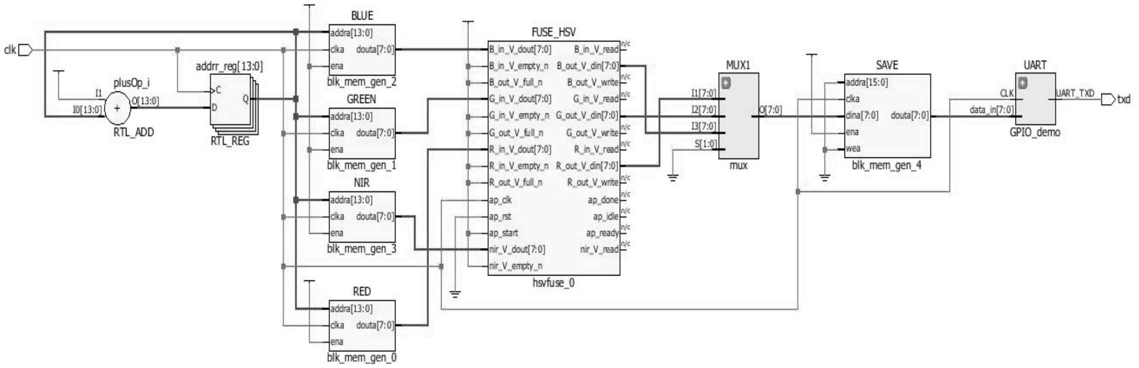


Figure 7. Module diagram of FPGA implementation.

Next, using the HLS tool, the equations for conversion from RGB color space to HSV color space, the replacement of V channel with NIR, and reconversion back to RGB color space were implemented. A test bench was written to verify the accuracy of the conversion function, and later, after synthesis (which provided the resource usage), the synthesized design obtained from HLS was exported as an IP (hsvfuse_0) for use in Vivado HLS. Figure 8 shows the IP synthesized using HLS, where ap_clk and ap_rst represent the clock and reset signals, R_in_V, G_in_V, B_in_V and nir_V represent the four inputs from the Block ROMs, which are initialized with R, G, B and NIR values. R_out_V, G_out_V and B_out_V represent the reconverted RGB color channels, with V channel replaced by the NIR. From the simulations, it was found that the entire fusion process for an image size 120 x 96 takes about 138 clock cycles. This means that approximately 83 pixels are fused per clock cycle by the newly designed IP (hsvfuse), which required additional FIFOs to be added to the inputs of the hsvfuse IP so that subsequent color data is loaded into the FUSE_HSV block after a sufficient delay.

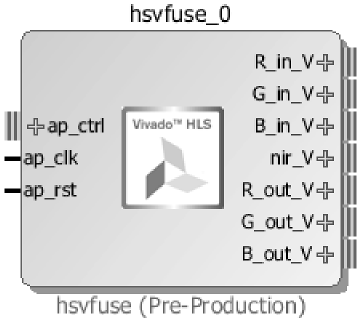


Figure 8. IP synthesized with HLS.

Finally, the R, G, and B results of the hsvfuse IP are then to be saved on to a single Block Memory Generator in a sequence order; that is, red values are saved in the first 11520 address locations, green values are stored in the subsequent 11520 address locations, and so on. Later, through the UART module, color values are transferred to the system and combined to display the fused image as shown in Figure 9.

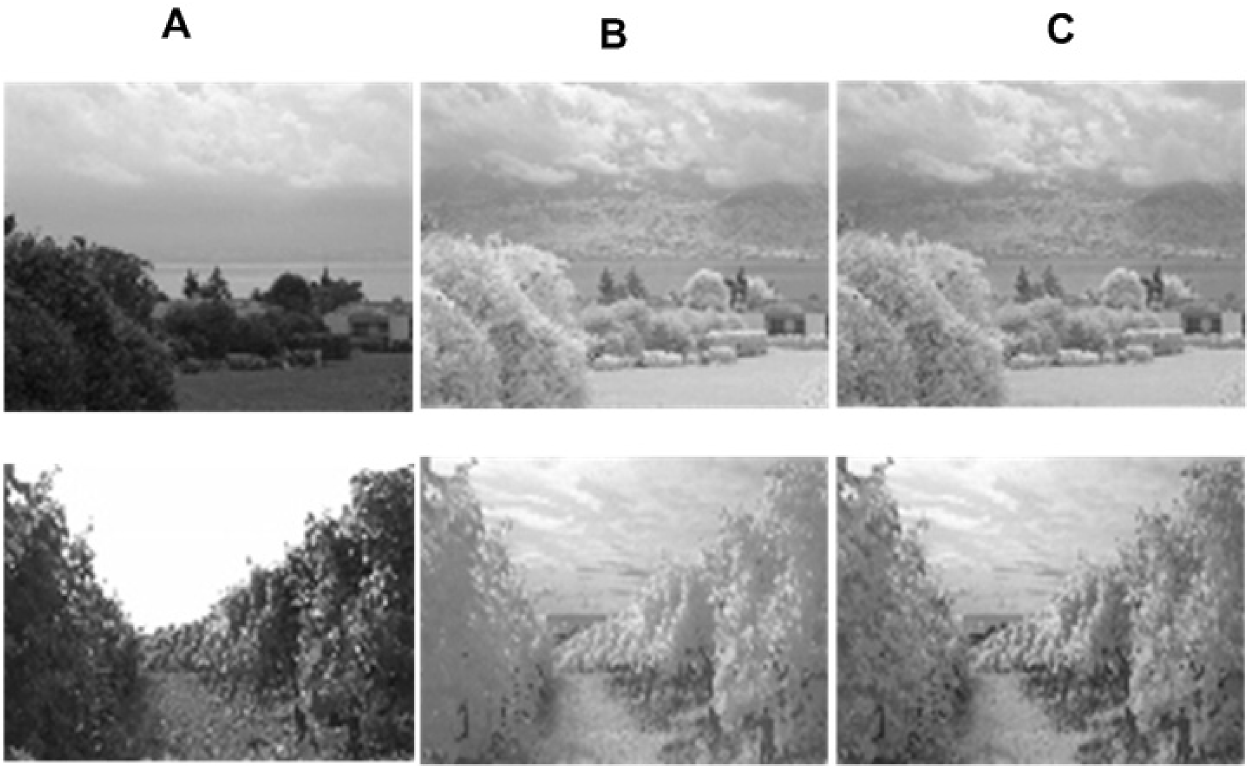


Figure 9. A: RGB Image; B: NIR Image; C: FPGA output of fused image.

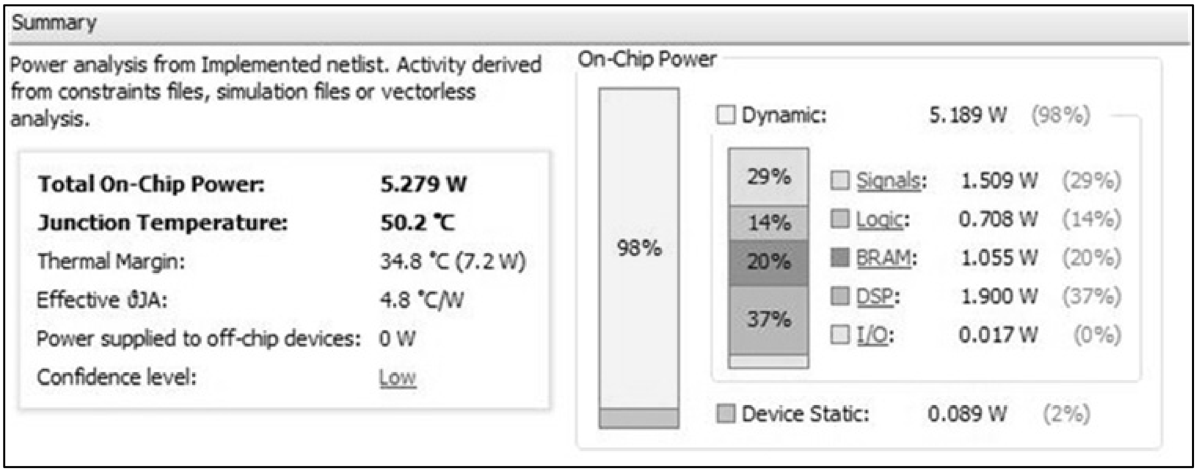
3.2. Design Analysis

The FPGA implementation was performed on Nexys 4 board with an Artix 7 FPGA. The design was synthesized, and the following information was obtained directly from the Vivado Design tool. Two important performance measures were analyzed from the design tool such as resource utilization memory and power utilization. Table 1 shows the resource utilization on FPGA for processing the images.

Table 1. Resource utilization summary of Nexys 4 FPGA.

Resource	Utilization	Available	Utilization in %
Slice LUTs	6568	20800	31.58
Slice Registers	6688	41600	16.08
Memory	22	50	44.00
DSP	15	90	16.67
IO	2	212	0.94

It can be seen from the above table that the process of converting to HSV, swapping the appropriate channels and reconverting to RGB utilizes only 44% of the total memory and 32% of the Look up table (LUT) slices. The utilization of DSPs, slice registers and input/output ports are much lower.



Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**5.279 W**
Junction Temperature:	**50.2 ℃**
Thermal Margin:	34.8 °C (7.2 W)
Effective ϑJA:	4.8 °C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

On-Chip Power

☐ Dynamic:	5.189 W	(98%)
☐ Signals:	1.509 W	(29%)
▨ Logic:	0.708 W	(14%)
■ BRAM:	1.055 W	(20%)
▨ DSP:	1.900 W	(37%)
☐ I/O:	0.017 W	(0%)
▨ Device Static:	0.089 W	(2%)

Figure 10. Power utilization summary of Nexys 4 FPGA.

The power utilization summary in Figure 10 indicates that the total on-chip power utilization is 5.279W, of which most of the power was dissipated by dynamic device resources. We also see details of the thermal properties of the device and the confidence level. The confidence level is a measurement of the accuracy and the completeness of the input data that the power analysis tool uses. Specifying details like the activity rates of clocks and input output signals can increase the accuracy.

From the above results, we can conclude that only 50% of the available resources were used in implementing the fusion algorithm on a FPGA with a power consumption of 5.279 W. The present system described in the present work does not include a camera for real time image capture and only works with existing datasets of RGB and near infrared images; hence, to enable real time image fusion, a camera system that simultaneously captures the two images (RGB and NIR) should be employed. Further, it can be observed that the precise RGB color information of the original visible image is not reproduced in the final fused image. This could be due to fixed precision data width used in the FPGA implementation. Also, optimizations can be brought about in the HLS design to include pipelining in order to reduce the latency in the fusion process.

The future work aims to use different fusion algorithms such as Discrete Wavelet Transform (DWT), Multiwavelet Transform (MWT), Multiresolution fusion methods for improving the quality of fused image and to compare with the present work. The present work considered a limited number of images from the dataset for experimentation. In order to generalize the system, we aim to analyze different databases in future to fine tune the proposed algorithm.

4. CONCLUSIONS

(1) The work attempts to improve the quality of image by fusing together the visible (RGB), as well as near infrared images of the same scene. There are several ways to implement this; however, the method found to be most convenient and, at the same time, providing good results was chosen.

(2) The luminance channel of the original visible image was replaced by the near infrared channel. The YCbCr and HSV color spaces are used to achieve the same, and it is concluded that using the HSV color space yields more visually appealing results on a general basis.

(3) The system is implemented on an FPGA, so that it can function independently and in real time depending on the application, where it needs to be used. Further modifications and optimizations are to be brought about in the system to make it a real-time system. The final fused image obtained contains color information from the RGB image, as well as depth and shadow details from the near infrared image.

REFERENCES

Avcibas, I., Sankur, B. & Sayood, K., 2002. Statistical evaluation of image quality measures. Journal of Electronic imaging, **11**(2): 206-24.

Awad, M., Elliethy, A. & Aly, H.A., 2018, October. A Real-Time FPGA Implementation of Visible/Near Infrared Fusion Based Image Enhancement. In 2018 25th IEEE International Conference on Image Processing (ICIP) (pp. 3968-3972). IEEE.

Brown, M. & Süsstrunk, S., 2011, June. Multi-spectral SIFT for scene category recognition. In Computer Vision and Pattern Recognition 2011, 177-84. IEEE.

Desmouliers, C., Oruklu, E., Aslan, S., Saniie, J. & Vallina, F.M., 2012. Image and video processing platform for field programmable gate arrays using a high-level synthesis. IET Computers & Digital Techniques, **6**(6): 414-25.

Feist, T., 2012. Vivado design suite. White Paper, 5, p.30.

Firmenichy, D., Brown, M. & Süsstrunk, S., 2011, September. Multispectral interest points for RGB-NIR image registration. In 2011 18th IEEE International Conference on Image Processing, 181-84. IEEE

Fredembach, C. & Süsstrunk, S., 2008, January. Colouring the near-infrared. In Color and Imaging Conference, Society for Imaging Science and Technology, **2008**(1): 176-82.

Payette, B., 2002. Color Space Converter: R’G’B’to Y’CbCr. Xilinx, XAPP637 (v1. 0), Sep.

Salamati, N., Fredembach, C. & Süsstrunk, S., 2009, January. Material classification using color and NIR images. In Color and Imaging Conference, Society for Imaging Science and Technology, **2009**(1): 216-22.

Salamati, N., Larlus, D. & Csurka, G., 2011. Combining visible and near-infrared cues for image categorisation. In Proc. of the 22nd British Machine Vision Conference (BMVC 2011). DOI:10.5244/C.25.49

Sadeghipoor, Z., Lu, Y.M. & Süsstrunk, S., 2011, September. Correlation-based joint acquisition and demosaicing of visible and near-infrared images. In 2011 18th IEEE International Conference on Image Processing (pp. 3165-3168). IEEE.

Sadeghipoor, Z., Lu, Y.M. & Süsstrunk, S., 2013, May. A novel compressive sensing approach to simultaneously acquire color and near-infrared images on a single sensor. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 1646-50. IEEE.

Sklyarov, V., Skliarova, I., Barkalov, A. & Titarenko, L., 2014. Embedded Blocks and System-Level Design. In Synthesis and Optimization of FPGA-Based Systems (pp. 143-203). Springer, Cham.

Soria, X., Sappa, A.D. & Akbarinia, A., 2017, November. Multispectral single-sensor RGB-NIR imaging: New challenges and opportunities. In 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)(pp. 1-6). IEEE.

Sun, T., Jung, C., Fu, Q. & Han, Q., 2019. NIR to RGB Domain Translation Using Asymmetric Cycle Generative Adversarial Networks. IEEE Access, 7, pp.112459-112469.

Tang, H., Zhang, X., Zhuo, S., Chen, F., Kutulakos, K.N. & Shen, L., 2015, April. High resolution photography with an RGB-infrared camera. In 2015 IEEE International Conference on Computational Photography (ICCP), (pp. 1-10). IEEE.

Zhang, Y., 2014. Real-time Image Processing on an FPGA for an Intraoperative Goggle Device. DOI: 10.7936/K7SF2T4B.

APPENDIX I

1.Colour Space Conversions

Assume that we have RGB values from 0 to 255

RGB to HSV:

R' = R/255

G' = G/255

B' = B/255

Cmax = max(R', G', B')

Cmin = min(R', G', B')

Δ = Cmax – Cmin

$$H = \left\{ \begin{array}{c} 0°, \qquad \Delta = 0 \\ 60° * \left(\dfrac{G' - B'}{\Delta} \mathrm{mod} 6\right), \;\; \mathrm{Cmax} = R' \\ 60° * \left(\dfrac{B' - R'}{\Delta} + 2\right), \;\; \mathrm{Cmax} = G' \\ 60° * \left(\dfrac{G' - B'}{\Delta} + 4\right), \;\; \mathrm{Cmax} = B' \end{array} \right\}$$

$$S = \left\{ \begin{array}{cc} 0, & \mathrm{Cmax} = 0 \\ \dfrac{\Delta}{\mathrm{Cmax}}, & \mathrm{Cmax} \neq 0 \end{array} \right\}$$

V = Cmax

HSV to RGB:

When 0 ≤ H < 360, 0 ≤ S ≤ 1 and 0 ≤ V ≤ 1:

C = V × S

X = C × (1 - |(H / 60º) mod 2 - 1|)

m = V – C

$$(R', G', B') = \left\{ \begin{array}{cc} (C, X, 0), & 0° \leq H < 60° \\ (X, C, 0), & 60° \leq H < 120° \\ (0, C, X), & 120° \leq H < 180° \\ (0, X, C), & 180° \leq H < 2400° \\ (X, 0, C), & 240° \leq H < 300° \\ (C, 0, X), & 300° \leq H < 360° \end{array} \right\}$$

(R,G,B) = (R'+m, G'+m, B'+m)

YCbCr to RGB:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & 0.291 & 0.439 \\ 0.439 & -0.368 & 0.071 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

RGB to YCbCr:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 1.164 & 0.000 & 1.596 \\ 1.164 & -0.392 & -0.813 \\ 1.164 & 2.017 & 0.000 \end{bmatrix} \cdot \begin{bmatrix} Y - 16 \\ Cb - 128 \\ Cr - 128 \end{bmatrix}$$