Bulletin of Electrical Engineering and Informatics
Vol. 7, No. 3, September 2018, pp. 386~392
ISSN: 2302-9285, DOI: 10.11591/eei.v7i3.687

Efficient Implementation of Mean, Variance and Skewness Statistic Formula for Image Processing Using FPGA Device

Aqwam Rosadi Kardian, Sunny Arief Sudiro, Sarifuddin Madenda
STMIK Jakarta STI & K,
Jl. BRI No.17 Radio Dalam, Kebayoran Baru, Jakarta Selatan, Indonesia

Article Info

Article history:

Received Sep 04, 2017
Revised May 31, 2018
Accepted Jun 14, 2018

Keywords:

Counter and accumulator
FPGA
Mean
Skewness
Variance

ABSTRACT

Processing statistic formula in image processing and accessing data from memory is easy in software, the other hand for hardware implementation is more dificult considering a lot of constraint. This article proposes an implementation of optimum mean, variance and skewness formula in FGPA Device. The proposed circuit design for all formulas only need three additions component (in three accumulators) and two divisions using two shift-right-registers, two subtractors, one adder and six multipliers. For 8x8 image size need 64 clock cycles to finish the mean, variance and skewness calculations, comparing other approach that need more than 1024 additions component without skewness calculation. Implementation into FPGA needs 68 slices of flip-flops and 121 of 4 input LUTs.

Corresponding Author:

Aqwam Rosadi Kardian,
STMIK Jakarta STI & K,
Jl. BRI No.17 Radio Dalam, Kebayoran Baru, Jakarta Selatan, Indonesia.
Email: aqwam@jak-stik.ac.id

1. INTRODUCTION

Digital image processing is a proccess for manipulating and analyzing image with computer. Some image processing in computer application uses basic statistic formula such as: histogram, mean, variance, skewness, etc. These process has tight correlation with pattern recognition or texture analysis which usually recognizing the object by extracting important information from the image (image features).

Research for histogram, mean and variance calculation base on FPGA device is proposed by [1]. This article proposed a method that needs 1024 addition operations, 64 cycles, 1 division (using shift right register), no need for memory but using 256 register 8 bit for 8x8 image size. This approach consuming more than 9000 slice or logic element of FPGA. Another research for histogram calculation is proposed in [2]. This approach use RAM subcell and FSM for controlling, this approach use 850 logic element of FPGA and can work at more than 100 MHz of frequency clocks and able to process 260 frames per seconds.

Research about hardware implementation for calculating the mean and variance in an iterative manner is proposed in [3]. This approach has been realized efficiently as a hardware architecture. Instead of carrying out combinational divisions, the proposed approach distributes the divisions over several iteration steps. This enables the calculation of running statistics using simple hardware elements. This design used 111 of registers and 2934 of LUT’s. Another approach is called sum and difference histograms (SDHs) for a dense texture analysis algorithm. The implementation of this approach into a reconfigurable architecture needs 73,021 logic elements, required memory is 737,152 bits, and the number of embedded multipliers is 480 [4].

A segmentation process for image analysis in detecting breast cancer using FPGA is proposed by [5]. In this approach skewness calculation is used for extracting the feature, 89,2 % of acuracy is obtained in this research.

The results of comparison with the paper [9], for the histogram calculation process paper requires 320 clock cycles, while the results of this study in accordance with the above calculation requires only 64 clock cycles by producing 3 parameters, namely histogram value, mean and variance. Based on this, it is proved that using the optimization formula is a combination of histogram calculation with mean and combined histogram calculations with variance can reduce the number of clock cycles required. The speed of the process to produce histogram value outputs in the previous paper required 32μs while the process time of this study only requires 6.4 μs with the output of histogram value, mean value and variance value. The weakness of the previous paper is that it takes an additional 256 cycles to process the interface to RAM as a data storage histogram value. The amount of RAM used in the previous paper is 2Kb. The results are more clearly seen in table 1.

This article proposes a more efficient method for mean, variance and skewness value calculation of the image intensity and its implementation on a FPGA. First section describes the state of the art of the similar research. Second section is about basic statistic formula for image processing. Third section is proposed algorithm in Matlab. Fourth section is about the efficient algorithm hardware implementation using FPGA device. Final section is conclusion of this article.

2. STATISTIC FORMULA FOR IMAGE PROCESSING

2.1. Image Histogram

Frequency of the appearance every intensity value from the whole image pixel element can be drawn based on image histogram diagram. [7] [8]. Higher value of histogram show the number of pixels with that intensity value is high and vice versa. Histogram can show the brightness and contrast of the image. A lot of use histogram in image texture analysis because of the simplicity of the algorithm. Histogram of an image with NxM pixels can be calculated mathematically as :

$$H(i) = \sum_{n=0}^{N} \sum_{M=0}^{M} k, \tag{1}$$

where *k*=1 if *f (n,m)=i* and *k*=0 if *f (n,m)≠i*, while *f(n,m)* is intensity value of the pixel at coordinate (*n,m*) in the image.

2.2. Mean Formula

From the histogram value we can calculate the 'mean' value from the image. Mathematical formula for 'mean' (*μ*) is: [7]

$$\mu = \sum_{i=0}^{L} i.p(i) \quad \text{where}\ p(i) = \frac{H(i)}{NM} \tag{2}$$

In this formula *i* is grey level of pixel intensity in the image and *p(i)* is the probability of occurrence *i*. *L*=255 is the higher value of grey level *i*. This formula will produce an average brightness of objects.

2.3. Variance Formula

$$\sigma^2 = \sum_{i=0}^{L=255} (i-\mu)^2 p(i) = \frac{1}{N*M} \sum_{i=0}^{255} (i-\mu)^2 H(i), \quad i = f(n,m) \tag{3}$$

σ^2 is the normalized 2nd order statistical formula, these feature shows the contrast level of the image. [7]

2.4. Skewness Formula

Skewness feature is measurement for the symetri of intensity average or brightness distribution. This distribution, or data set, is symmetric if it looks the same to the left and right of the center point. [5] In mathematical formula is stated as equation (4).

$$\text{skewness} = \sum_{i=0}^{L=255} (i-\mu)^3 p(i) = \frac{1}{N*M} \sum_{i=0}^{255} (i-\mu)^3 H(i), \quad i = f(n,m) \tag{4}$$

Skewness is also called normalized 3rd order moment. Negative value stated that brightness distribution tented to the left of intensity average and positive value stated otherwise condition. In practice this skewness value is divided by L^2 to be normalized. [7] Figure 1 show the ilustration of this histogram distribution and skewness value, which tented to left (negative skewness value) or right (positive skewness value).

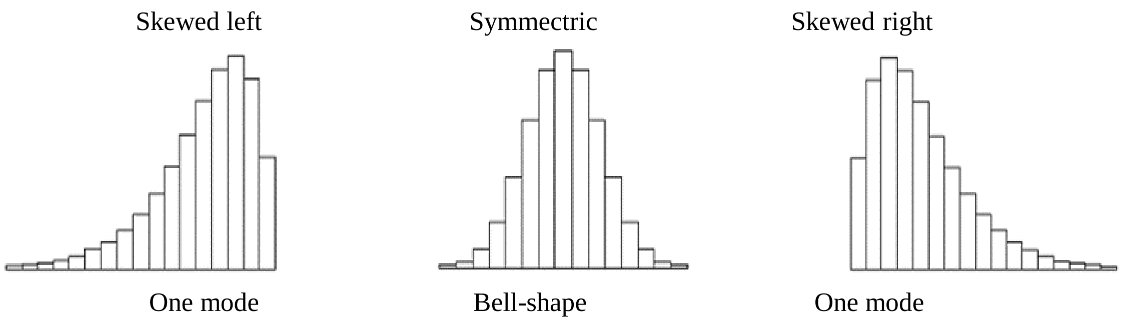


Figure 1. The ilustration of histrogram distribution and skewness [8]

3. PROPOSED ALGORITHM

Mean is the first-order and variance is the 2nd-order of statistical analysis methods used for segmentation and feature extraction processes an image. Mean represents the mean value of the intensity of the entire pixel in an image, while variance represents the image contrast, and skewness is brightness distribution. Referring to the Equations 1-4 we can derive the Equation 5-7 that is optimum formula that proposed to be implemented on the FPGA.

$$\mu = \sum_{i=0}^{255} i.P(i) = \frac{1}{N*M}\sum_{n=0}^{N}\sum_{m=0}^{M} f(n,m) \quad (5)$$

$$\sigma^2 = \left(\frac{1}{N*M}\sum_{n=0}^{N}\sum_{m=0}^{M} i^2\right) - \mu^2 \quad ,\text{if} \quad i = f(n,m) \quad (6)$$

$$skewness = \left(\frac{1}{N*M}\sum_{n=0}^{N}\sum_{m=0}^{M} i^3\right) - \left(3*\mu*\frac{1}{N*M}\sum_{n=0}^{N}\sum_{m=0}^{M} i^2\right) + 2*\mu^3 \quad ,\text{if} \quad i = f(n,m) \quad (7)$$

Pseudo code 1. Mean formula (1st version in equations 2 and 3)

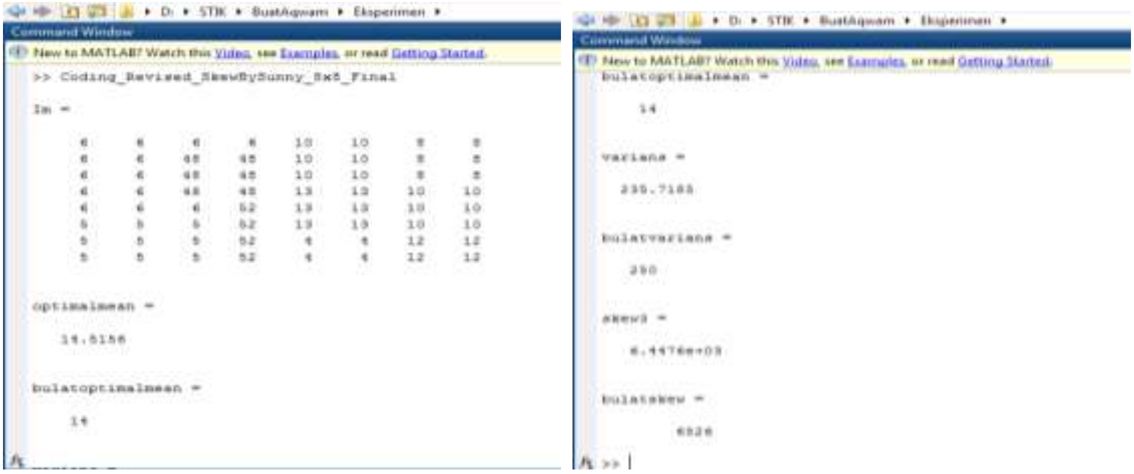
```
f=% data %
h=zeros(256,1);
[N,M]=size(f);
for i=1:N                % Histogram H(i) calculation%
   for j=1 : M
         h(f(i,j)+1)=h(f(i,j)+1)+1;
   end
end
N1=(N*M);
for i=1 : 256          % Probability p(i) calculation%
   p(i)=h(i)/N1;
end
Mean=0;
for i=1 : 256         % Mean calculation%
   Mean=Mean + p(i) * (i-1);
end
Mean
```

The pseudo code of mean calculation based on Equation 2 is showed in Pseudo Code-1. There are three steps of calculation: the first one is histogram *H(i)*, the second one is probability *p(i)* and the last one is mean value. The algorithm of the mean formula in the Equation 5, optimal variance Equation 6 and skewness Equation 7 is presented in Pseudo Code-2. In this algorithm, there is just one step of calculation that has the same complexity as histogram *H(i)* calculation. It means that the proposed algorithm can reduce two steps of calculation (2 x 256 clock cycles) and some number of arithmetic operations e.g. addition, multiplication and division.

Pseudo code 2. Mean and variance formula (proposed version in *equations 4 and 5*)

```
[M,N,L]=size(f);              % f is the data
htsum=0; vsum=0; i3=0; skw=0;
for i=1:M;                    % Mean calculation%
  for j=1:N;
       htsum=htsum+ f(i,j);
       vsum=vsum+( f(i,j))^2;
       i3=i3+(f(i,j))^3;
  end;
end;
optimalmean=htsum/(M*N).
bulatoptimalmean=fix(htsum/(M*N))
varians=vsum/(M*N)-optimalmean^2      % variance $\sigma^2$ calculation
bulatvarians=fix(vsum/(M*N))-bulatoptimalmean^2
skew3=i3/(M*N)-3*optimalmean*(vsum/(M*N))+ 2*(optimalmean^3)
bulatskew=fix(i3/(M*N))-3*bulatoptimalmean*fix(vsum/(M*N))+ 2*(bulatoptimalmean^3)
```

The results of mean calculation using basic algorithms and optimization algorithms for the same data (Im) are the same, as shown in Figure 2. The values resulting from the evaluation using Matlab will be used for comparison with values that are processed using FPGA components, including the value of *bulatoptimalmean*, *bulatvarians* and *bulatskew* that using fix value (not concerning the floating point value).


D: ▸ STIK ▸ BuatAqwam ▸ Eksperimen ▸
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

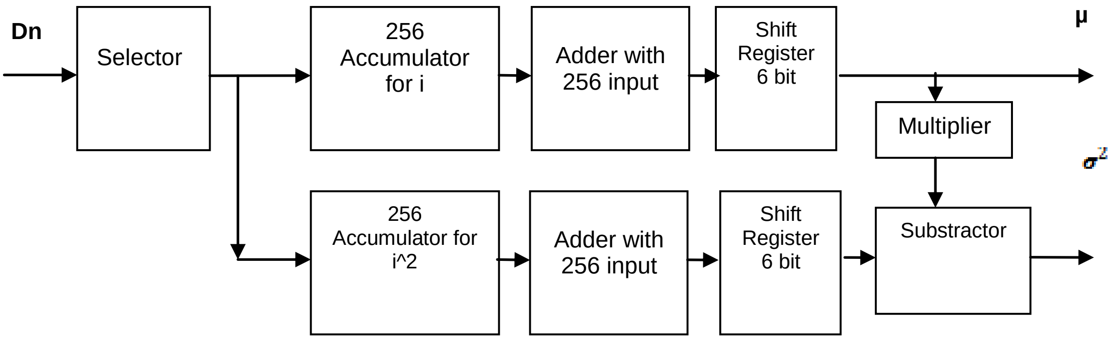
```
>> Coding_Revised_SkewBySunny_8x8_Final

Im =

     6     6     6     6    10    10     8     8
     6     6    48    48    10    10     8     8
     6     6    48    48    10    10     8     8
     6     6    48    48    13    13    10    10
     6     6     6    52    13    13    10    10
     5     5     5    52    13    13    10    10
     5     5     5    52     4     4    12    12
     5     5     5    52     4     4    12    12


optimalmean =

   14.5156


bulatoptimalmean =

    14
```

```
bulatoptimalmean =

    14


varians =

  235.7185


bulatvarians =

   250


skew3 =

   6.4476e+03


bulatskew =

        6526

>> |
```

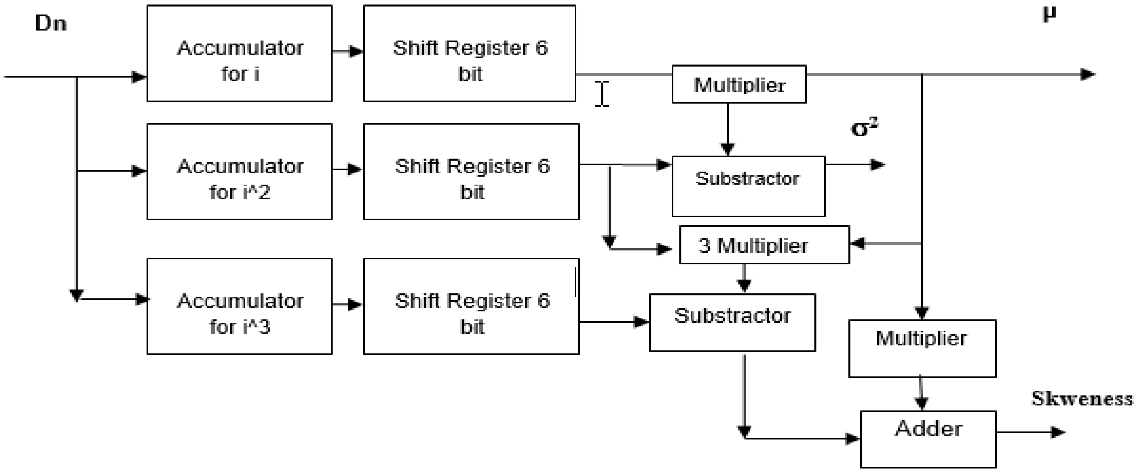
Figure 2. Results form both algorithm in Matlab with the same data

4. HARDWARE IMPLEMENTATION

Figure 3(a) is the first design based on first pseudo code, this design need a selector to select intensity value of pixel from the image between 0 to 255 and send to their own respective accummulator (256 accumulators with 256 addition components). All these accumulator value will be sent to 256 input addition (or another 256 additions) so this approach need 512 additions [1]. And finnally to obtain mean value we divided with the number of pixels wich is 8x8=64. We use shift right register 6 bits to do this division operation. Figure 3(b) is the new propose design using optimal formula, this design give us reduction of aritmatic component.



(a)



(b)

Figure 3. The block diagram of component (a) reference design [1] and (b) proposed design

Figure 4 is entity diagram (component). This design is the new design based on the design in Figure 3.b. For calculating mean and variance value actually we don't have to calculate histogram, we can directly using one accummulator to hold the total value of pixel element value (htsum) in the 2nd pseudo code. One other accumulator to hold 2nd order value of pixel element (i^2). This approach give us the reduction of mathematical operation (component). All design only need two additions in two accummulators and two shift right register, multiplier and subtractor, with the same processing time (NxM=64 clock cycles) for 8x8 image size and the same result of mean and variance value. In this design we use counter and buffer to send the final value of mean and variance after 64 clocks.

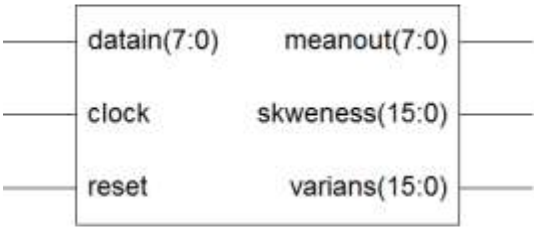


Figure 4. Component entity

5. SIMULATION RESULT

Figure 5 show the simulation result based on behavioral simulation using clock period of 100ns per cycles, this simulation shows for 70 cycles (7000ns) or 70 calculation. This figure shows the result for the same data which is 14 (mean), 250 (variance), and 6526 (skweness). Comparing to the Matlab evaluation result wich is 14,5 with the difference is 0,5 or an error of ((0,5/14)*100%=3,45 %), and 235,72 with the difference is 14,28 or an error of 6,05 % (14,28/235,72*100%). For skewness value the diferrerence is 78,4 (6526-6447,6) or an error of 1,21 % (78,4/6447,6*100%).

The speed of process is paralel with the comming of data input (data in), and final result will be obtined at the end of 64 clock cycles for 64 data. From Figure 6 we can see the design summary of the component that provide the information of logic element accupation in FPGA device. If we calculate in Matlab with integer value (not concerning the floating point value) the result are exactly the same (14 for mean value, 250 for variance value and 6526 for skewness value).

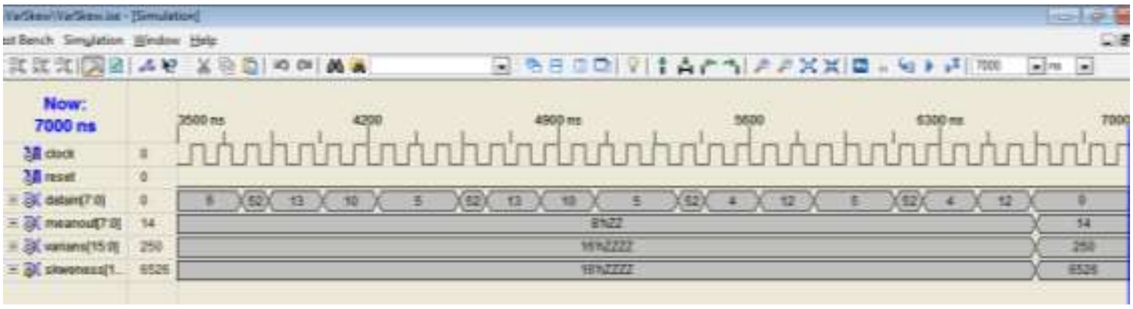
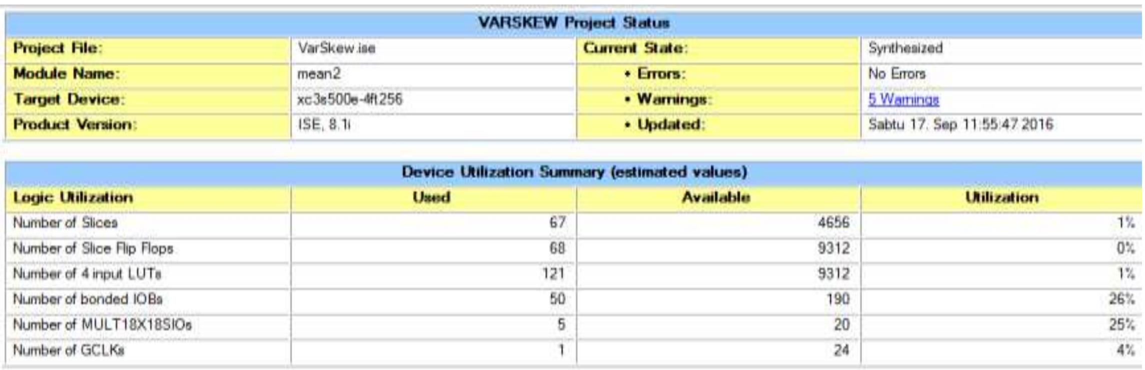


Figure 5. Behavioral simulation results during 70 cycles (7000 ns)



VARSKEW Project Status			
Project File:	VarSkew.ise	**Current State:**	Synthesized
Module Name:	mean2	• **Errors:**	No Errors
Target Device:	xc3s500e-4ft256	• **Warnings:**	5 Warnings
Product Version:	ISE, 8.1i	• **Updated:**	Sabtu 17. Sep 11:55:47 2016

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	67	4656	1%
Number of Slice Flip Flops	68	9312	0%
Number of 4 input LUTs	121	9312	1%
Number of bonded IOBs	50	190	26%
Number of MULT18X18SIOs	5	20	25%
Number of GCLKs	1	24	4%

Figure 6. Summary of logic utilization for mean, variance and skewness calculation component

Table 1, shows the summary of resources as a comparison of the results of the previous research method implementation and the proposed method into the FPGA. The previous method of research only implements the mean equations and variance, while the proposed method implements the equations of mean, variance and skewness. In terms of the use of slice number of previous research methods using 12461 is the proposed method using only 128 slices. This means that the proposed method can reduce the use of slice amounts of 12221 slices. Similarly, for the number of flip-flops, the proposed method can reduce the use of flip-flop counts of 9889 flip-flops.

Table 1. FPGA Design Summary Comparison

No.	FPGA resources	Previous research [9]	Previous research [1]	Proposed method
1.	Number of slice	NA	12461	67
2.	Number of slice flip flop	NA	9970	68
3.	Number of LUTs	NA	20157	121
4.	Latency	320 clock	64 clock	64 clock
5.	Number of Multiplier	NA	2048	3
6.	Number of Addition	NA	256	1024
7.	Function	Histogram	Histogram, mean, Varians,	Mean, Varians and Skewness

The proposed method is also capable of reducing the use of LUTs as much as 19905 LUTs. The processing time is 64 clocks from, with the error rate (MSE) difference of 0.07 from the previous research method (Mean and Variance) i.e. where results are much more accurate.[1] Compare to research result [9], it need 320 clock for histogram calculation only, instead of our result need 64 clock to calculate mean, varians and skewness value.

6. CONCLUSION

The efficient implementation of mean, variance and skewness statistic formula into hardware base component using FPGA device has been proposed. This component use three additions component (in three accumulators) and two divisions using two shift-right-registers, two subtractors, one adder and six multipliers. Calculating mean and avariance value for 8x8 image size only needs 64 clock cycles. This design (with counter and buffer) needs only 68 slices of flip-flops and 121 of 4 input LUTs. The difference (as an error) of mean value commparing to Matlab result is 3,57% in average due to floating point problem in FPGA. The FPGA result is exactly the same with Matlab result for integer value (not concerning the floating value, *optimal rounded* and *varians rounded*) which are 14, 250 and 6526.

ACKNOWLEDGEMENT

Thanks to Yayasan Pendidikan Gunadarma and Yayasan Ilmu Komputer for supporting this research as one part of research dissertation in Doctoral Program of Gunadarma University, Jakarta-Indonesia.

REFERENCES

[1] A. Pertiwi. Optimalisasi Metode Implementasi Algoritma Histogram, Mean Dan Variansi Ke Dalam IC-FPGA Untuk Aplikasi Analisis Tekstur Citra Real Time. Disertasi.Teknologi Informasi. Universitas Gunadarma. 2015.

[2] L. Maggiani, C. Salvadori, M. Petracca, P. Pagano, R, Saletti. *Reconfigurable Archictecture for Computing Histograms in Real-Time Tailored to FPGA-based Smart Camera.* IEEE 23rd International Symposium on Industrial Electronics (ISIE). Istambul Turkey. 2014.

[3] D.G. Bailey, M.J. Klaiber. *Efficient Hardware Calculation of Running Statistics, 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013)*. Page(s): 1-6.2013; ISSN: 151-2191; Print ISBN: 978-1-4799-0882-0.

[4] M.A. Ibarra-Manzano, D.L.A. lmanza-Ojeda, J.M. López-Hernández, *Design and Optimization of Real-Time Texture Analysis using Sum and Difference Histograms Implemented on an FPGA*. Electronics. Robotics and Automotive Mechanics Conference. Mexico. 2010.

[5] Stuti Sharon, Vani V, Shobha B.N. *FPGA Implementation for Automated Classification of Breast Cancer Using Support Vector Machines.* International Journal for Research in Applied Science & Engineering Technology (IJRASET). Volume 3. June 2015. Special Issue-I1; ISSN: 2321-9653.

[6] W.L. Martinez, A.R. Martinez. Computational Statistics Handbook with MATLAB. Chapman & Hall/CRC. USA. 2002.

[7] R.E. Woods, S.L. Eddins, R.C. Gonzales. Digital Image Processing Using MATLAB. Pearson Education. 2005.

[8] David P. Doane, Lori E. Seward, *Measuring Skewness: A Forgotten Statistic?* Journal of Statistics Education Volume 19. 2011; Number 2.

[9] Nitin Sachdeva, Tarun Sachdeva. *An FPGA Based Real-time Histogram Equalization Circuit for Image Enhancement*. International Journal of Electronics Communication Technology (IJECT). Vol. 1. Issue 1. December 2010; ISSN : 2230-7109 (Online).