An Aspect of Biometric Image Matching Using Python

Devraj Roy, Pinaki Pratim Acharjya
Department of Computer Science and Engineering
Durgapur Institute of Advanced Technology and Management, Durgapur, West Bengal, India
roydevraj77@gmail.com, ppacharjya@gmail.com

Abstract—*In this paper we have compared the performance of two image matching techniques – mean square error and structural similarity index, using python. For our paper we have considered the matching of fingerprints.*

Keywords - fingerprints, image matching, mean square error, structural similarity, python

I. INTRODUCTION

Image matching is technique by which we can measure the similarities between images [1]. In this paper we have compared the workings of two digital image matching techniques- mean square error method [2] and structural similarity index. For our comparison we are using fingerprints as they are of much importance in our real life. Now a day's biometrics are used for security reasons, like for authorization of a personnel in an organization. They are also used for identification purposes by the government (UIDAI), investigators, etc.

The images of fingerprints [3], [6-7] used by us were converted into grey scale format using the OpenCV library. We have also used matplotlib, numpy, scikit-image modules for computation and plotting purposes. The images that were originally in RGB format were taken as input using OpenCV and later converted to grey scale (numpy arrays) for computation. The pixel aspect ratio of each biometric image is 2592X1456.

The Mean Square Error (MSE) [4-5] method compares the numpy arrays and returns a positive real value; 0 indicates a perfect match and the greater the value, more distinct is the images. The Structural Similarity Index (SSIM) returns a value in the range of -1 to 1 where 1 indicates a perfect match. In general it is evident that SSIM gives better results but in our study we see that MSE gives comparatively better results.

II. MATHEMETICAL MODELING

Mean Square Error measures the average of the squares of the errors [4-5]. It is always non negative and values closer to zero indicate more accuracy. In our matching algorithms we implement MSE by computing the square of differences of each element of the Numpy arrays and taking their mean. Mathematically we can represent it as shown in equation1.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \quad (1)$$

To compute this we provide corresponding arrays of the two grey scale images as parameters to our python function. We then convert the arrays to float type, to prevent computation errors, and we square the differences of each element at the same positions of the two arrays and sum them up. After that we find its mean by dividing the summation value by the total number of pixels in the images.

The Structural Similarity Index (SSIM) is a perceptual metric that quantifies image quality degradation caused by processing such as data compression or by losses in data transmission. For the computation of SSIM we use scikit-image's inbuilt method ssim(). SSIM takes into account the pixel intensities, variance of intensities along with the covariance which results in a more accurate comparison of images. While MSE focuses on estimating the error, SSIM compares small sub samples of the image to detect any structural variations. Mathematically it is given by the formula as shown in equation 2.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\alpha_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\alpha_x^2 + \alpha_y^2 + c_1)} \quad (1)$$

In the equation represented in equation 2, x and y represent the pixel values present in the matrix form of the two images. Also μ_x represents the average of x, μ_y represents the average of y, α_x^2 is the variance of x, α_y^2 is the variance of y and α_{xy} is the covariance of x and y.

III. FLOWCHART AND EXECUTION

In our study we are also comparing the execution time as a parameter for deciding the better algorithm. In our test data we have considered ten fingerprints [4-5] and have implemented the matching algorithms comparing each image with the other. In Figure 1 you can see a sample of the images we are using along with their grey scale converted forms.

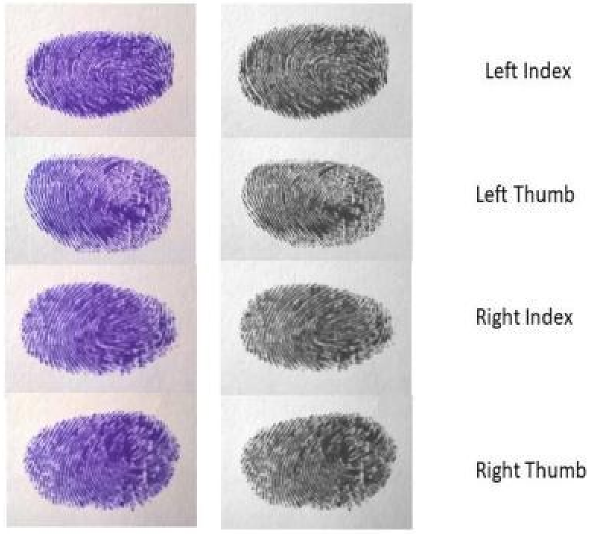


Figure 1. Sample of the fingerprints used for image matching and their corresponding grey scale forms

Execution shows that performing MSE [8-9] for comparing two images takes 0.13065 seconds. The total execution time for the comparison of each possible pair is 11.44938 seconds. After computation we check the MSE value. A value exactly equal to zero indicates a perfect match of the images. Also, more the value of MSE is, more distinct are the images. MSE cannot be negative as we square the difference between the pixel intensity. Figure 2 below represents the working flowchart of the MSE algorithm.

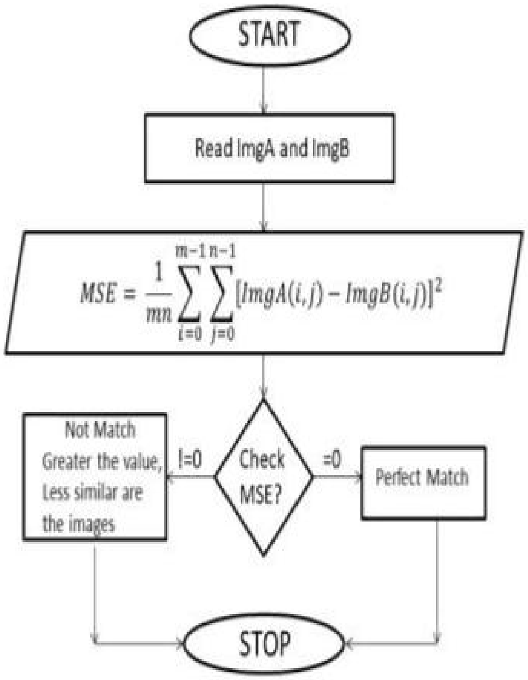


Figure 2. Flowchart of MSE

Similarly for SSIM we note the execution time to be as follows: for comparing two images it takes 0.92054 seconds and the total time for the comparison of all possible pairs is 81.50110 seconds.

This is clearly in accordance with the fact that SSIM takes into account more parameters than MSE does. SSIM yields a value that lies between -1 to 1. 1 indicates a perfect match of the images. The more SSIM value progresses towards -1 the more distinct are the images. Figure 3 shows the working flowchart of SSIM.

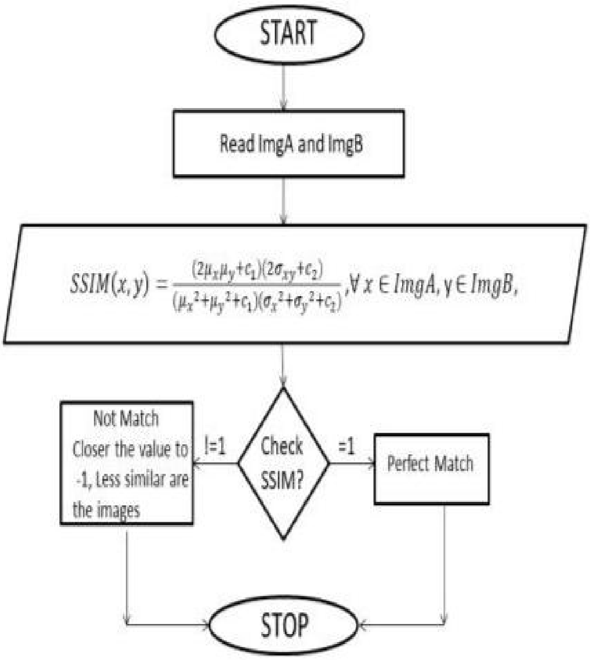
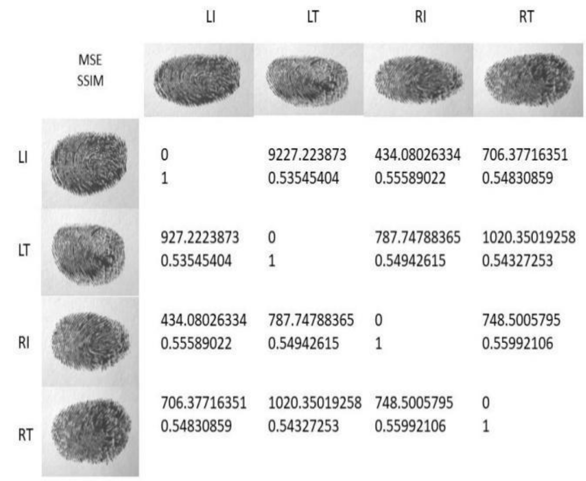


Figure 3. Flowchart of SSIM

IV. Results and Discussion

As it is evident from the results of the image matching process we can see that the execution of MSE is much faster than it is for SSIM. Also it can be seen that SSIM yields values that differ very slightly for each comparison unlike MSE. However this is also due to the fact that SSIM is bounded between -1 to 1 while MSE is not. Viewing the results as shown in Figure 4 we can say that both algorithms produce feasible results.



MSE SSIM	LI	LT	RI	RT
LI	0 1	9227.223873 0.53545404	434.08026334 0.55589022	706.37716351 0.54830859
LT	927.2223873 0.53545404	0 1	787.74788365 0.54942615	1020.35019258 0.54327253
RI	434.08026334 0.55589022	787.74788365 0.54942615	0 1	748.5005795 0.55992106
RT	706.37716351 0.54830859	1020.35019258 0.54327253	748.5005795 0.55992106	0 1

Figure 4. Sample MSE and SSIM values

As mentioned earlier the execution time of SSIM is slower than that of MSE. This is evident from the results as plotted in Figure 5 and Figure 6. For 100 comparisons SSIM takes as much as 82 seconds (approx.) while MSE does the same in 11 seconds (approx.).

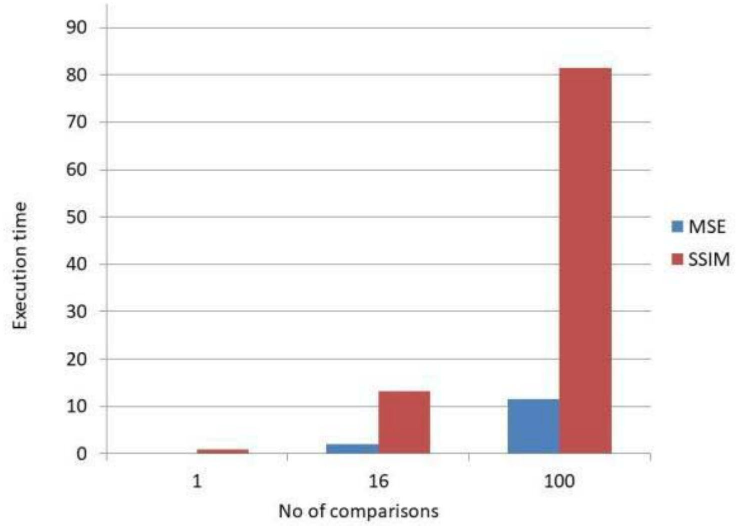


Figure 5. Bar graph representing the execution time

Considering the fact that SSIM includes parameters like mean, variance, covariance of the pixels, it is common to overlook the higher execution time for better accuracy. However due to the conversion of images to gray scale these factors do not hold as much importance as they do in case of RGB images.

		Execution Time	
		MSE	SSIM
no of comparisons	1	0.13065	0.92054
	16	2.06448	13.25456
	100	11.44938	81.5011

Figure 6. Execution time in seconds for different no. of comparisons

V. Conclusion and Future Schope

From the results in Figure 4 and the execution time in Figure 6 we can say that for our grey scale converted fingerprint images MSE does give better performance with satisfactory results. However this does not mean that SSIM is not worth it. In our day to day to life any image captured, including biometrics are in the RGB format where SSIM would give much better results than MSE for just a little cost in performance. So the algorithm to be used for biometric image matching is actually situational. Both give satisfactory results but ultimately it all depends on the need of the user.

References

[1] Pinaki Pratim Acharjya, Dibyendu Ghoshal, “An Image Matching Method for Digital Images Using Morphological Approach”, International Journal of Computer, Information, Systems and Control Engineering, Vol. 8 Issue. 5, 2014.

[2] Dibyendu Ghoshal, Pinaki Pratim Acharjya, “Effect of Various Spatial Sharpening Filters on the Performance of the Segmented Images Using Watershed Approach Based on Image Gradient Magnitude and Direction”, International Journal of Computer Applications, Vol. 82, Issue. 6, pp. 10-26, 2013.

[3] Pinaki Pratim Acharjya, Sagnik Dutta, “An Aspect of Biometric Image Segmentation And Authentication For Digital Images”, In the proceedings of the 2016 International Conference on Computational Intelligence and Computing Research, pp. 216-223, 2017.

[4] Pinaki Pratim Acharjya, Soumya Mukherjee, Dibyendu Ghoshal, “Digital image segmentation using median filtering and morphological approach”, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, Issue. 1, pp. 552-557, 2014.

[5] M. A. Gonzalez, V. L. Ballarin, “Automatic marker determination algorithm for watershed segmentation using clustering”, Latin American Applied Research, Vol. 39, pp. 225-229, 2009.

[6] J. Berry, “The history and development of fingerprintinfg”, in Advances in Fingerprint Technology, (H. C. Lee and R. E. Gaensslen, ed.s), CRC Press, Florida, pp. 1-38, 1994.

[7] M. Owis, A. Abou-Zied, A. B. Youssef, Y. Kadah, “ Robust feature extract ion from ECG signals based on nonlinear dynamical modeling”, In the Proceedings of the 23rd Annual International Conference IEEE Engineering in Medicine and Biology Society, (EMBC’01), Vol. 2, pp. 1585-1588, 2001.

[8] F. Meyer, S. Beucher, “Morphological Segmentation”, Journal of Visual Communication and Image Representation, Vol. 1, pp. 21-46, 1990.