Synthetic Roof Image Generation using an Extended Gaussian Mixture Model (EGMM)

Sos S. Agaian[1], Shishir P. Rao[1], Rushikesh D. Yeole[1], Cory R.A. Hallam[2]

[1] College of Engineering, [2]Center for Innovation, Technology, and Entrepreneurship (CITE)

The University of Texas at San Antonio.

One UTSA Circle, San Antonio, TX-78249

Abstract: *In this paper, we address the problem of generating synthetic images by using a single image, or multiple images, and present an effective Extended Gaussian Mixture Model (EGMM). We apply this method to roof surface images for investigating and identifying differences, changes, and/or structural damage. This technique allows a single impression of a roof to be randomly generated according to input parameters, and can produce new image models using multiple roof images. We further extend GMMs by mixing color textures of the roof images to generate new models. The best fit statistical model image is obtained by finding the minimum pixel distances as compared to the original image. This approach generates very realistic roof models that are useful in performance evaluation and testing, and offers the potential for further research and application in other surface inspection processes.*

Keywords: *Gaussian Mixture Model, Synthetic Roof Image, EM algorithm, K-means algorithm, SSIM index*

1. Introduction:

In recent years many insurance companies have been using drone images and Google maps to generate databases of insured properties, structures, and roofs in particular. These databases help survey properties that have been in the path of a storm, and using a pre-and post-image comparison, estimate the potential structural damage and insurance liability. These image databases coupled with the appropriate algorithms can lead to new capabilities that serve two purposes, namely verification of an insurable event, and improvement of claim processing time as a result of automated determination of an insurable event. Databases of images can be helpful in determining the type of roof that can be used for structure classification purposes and damage detection, and thus trigger cost estimation algorithms. Collecting large databases of pre and post storm images, especially in the case of large scale natural disasters, can be very arduous, may not be feasible for a long period of time, and can be very expensive. Due to the complexity of varying illumination in the environment, and the presence of camera artifacts, it is not easy to generate a parametric model for the observed image [38]. Synthetic models are required for generating databases of particular input images by varying its parameters, and great efforts are continuously spent in designing new algorithms both in academic and industrial environments for such uses [35].

Image synthesis of natural scenes plays an important role in real life image modeling and computer graphics. Many methods have been developed for gathering and testing large databases, such as for fingerprint images or medical images [35]. However, the authors are not familiar with any synthetic roof image generation method that suits the needs of the insurance industry discussed above. Roof images primarily contain textures, thus we focus on texture synthesis in this work. While recent publications highlight an increase in texture synthesis work, we contend that there are three main texture synthesis approaches discussed, including (i) image *texture replacement* [38], (ii) *image retexturing* [37], and (iii) *texture mixing* [36]. In the process of retexturing, existing textures of the original image are replaced by new ones in the region of interest in the image, thus preserving original image characteristics [38]. It has many applications in industry, interior design, artwork, digital movies and computer graphics [38]. Traditional methods of texture synthesis mainly focus on texture replacement [38-43]. These methods usually require complicated and sometimes labor-intensive processing steps [43].

More recently, the Gaussian Mixture Model (GMM) has been used in image processing applications [29]. A GMM is a probabilistic model in which all the data points are generated from a mixture of a number of Gaussian distributions with unknown parameters which are further estimated [1]. The GMM is a weighted sum of K Gaussian density distributions [1] shown in equation (1) below:

$$s(\mathbf{x}) = \sum_{k=1}^{K} \pi_i N_k(\mathbf{x}, \mu_k, \textstyle\sum_k) \quad (1)$$

Where

$$N(x,\mu,\textstyle\sum) = N(x,\theta) = \frac{1}{\sum_k \sqrt{2\pi}} e^{-\frac{(x-\mu_k)^2}{2\sum_k^2}}, \quad \pi_i = \frac{\pi_k N(x \mid \mu_k, \sum_k)}{\sum_{j=1}^{k} \pi_j N(x \mid \mu_i, \sum_j)} \quad (2)$$

Where π_i are the weights with and $N_k(\mathbf{x}, \mu_k, \sum_k)$ is an i-Gaussian distribution component of the mixture model with its own mean μ_k, variance shape Σ_k, and k is the number of splitting images (or labels). Usually, the parameters (mean μ_k and variance $\sum_k$) of the probabilistic model are calculated using an expectation–maximization (EM) algorithm. These algorithms contain an expectation (E) step that computes the logarithmic likelihood of entire data set with given samples and a maximization (M) step that calculates the parameters by maximizing the logarithmic likelihood function [1, 29].

The standard GMM uses a single adaptation or learning rate that is a compromise of parametric rates. In this approach, each pixel is modeled as a mixture of two or more Gaussians and then each new image frame is generated. The stability of the Gaussian distribution is evaluated to estimate the result of a more stable background process or a short-term foreground process. The training of GMMs can be accomplished using Expectation Maximization. Thus it follows that we investigate its use in new image processing applications, such as the generation of synthetic roof images.

This paper focuses on generating synthetic roof images by using an Extended Gaussian Mixture Model (EGMM). The main contributions of this paper are: (1) generation of synthetic images by using a single image, and (2) generation of synthetic images by using multiple images. We model the underlying image with a mixture model that can capture the different types of image textures with parameters. Each model contains the intrinsic statistical structure of its image texture. These parametric models are flexible, fast, and do not change the size of output model obtained. Furthermore, we use the concept in mixing the texture of the roof images by replacing the parameters of one model with another model to generate a new texture image, such as slate, asphalt shingles, and clay tiles. The remainder of the paper is organized as follows. We discuss the state of the art in algorithms related to texture syntheses in Section II, and present EGMM algorithms for roof image duplication and texture mixing in Section III. We provide computer simulation results in Section IV and conclude the paper with further recommendations in Section V.

2. State of the Art in Algorithms

The three main texture synthesis approaches include *image texture replacement* [38, *image retexturing* [37], and *texture mixing* [36]. When considering the *Texture mixing* method, Ferradans *et al.* (2012) use stationary Gaussian models to generate a mixed color texture model from an input dataset of images [36]. Their method is based on the geodesic path defined by the optimal transport metric in Gaussian models. The barycenter and geodesic path between models are derived according to optimal transport. After observing the interpolated results, the set of texture models can be seen and further texture synthesis can be performed. The newly generated model produces natural looking results while reconstructing the features of the original image.

The *Texture Replacement/Retexture* Method as described by *Yanghai et al.* (1998) replace the specified texture patterns in an image while preserving lighting effects, shadows, and occlusions using the lighting map while detecting the texture patterns of the input image [37]. Using the sample texture plane, the standard tile related to the image is obtained. The mutual information is calculated between this standard tile and the image path to generate candidate texture regions which are used in finding the admissible lighting distribution. The concepts of Markov random fields, Maximum *a posteriori* estimation, and Markov Chain Monte Carlo methods are used in this process. A visually satisfactory texture replacement for the given input is thus obtained. Shen *et al.* (2011) present a color-mood-aware technique to re-texture clothing in a photograph and use it in multimedia applications [38]. A classification algorithm is designed to classify clothing textures using color mood schemes. Gradient maps of the fabric region are calculated, which helps in finding the texture distortion co-ordinates. Then, depending on the target clothing selection from the database of color mood by the user, the lighting and shading effects of the textures is transferred to HSV color space. The re-textured image is obtained without disturbing the geometry of the original image. In order to generate visually convincing replacement, the texture segmentation map and lighting map are obtained separately.

For the *Texture Synthesis* Method Xiaopei *et al.* (2012) have evaluated the illumination and deformation fields on textures from both and analytical and application perspective [38]. Since this estimation requires complex processing, an efficient statistical approach is proposed in this paper. The spatially varying illumination and deformation is inversely estimated according to the variation of the texture statistics. This texture photo is decomposed into an illumination field, a deformation field, and an implicit texture which is illumination and deformation free (using minimal user input). Texture replacement, surface lighting and other synthesis effects are generated by effectively recombining these texture components.

3. Generating Synthetic Roof Images Generation by using an Extended Gaussian Mixture Model (EGMM)

In this Section, we discuss the steps that are implemented for the newly developed EGMM algorithms. Figures 1 and 2 provide the block diagrams for these algorithms.

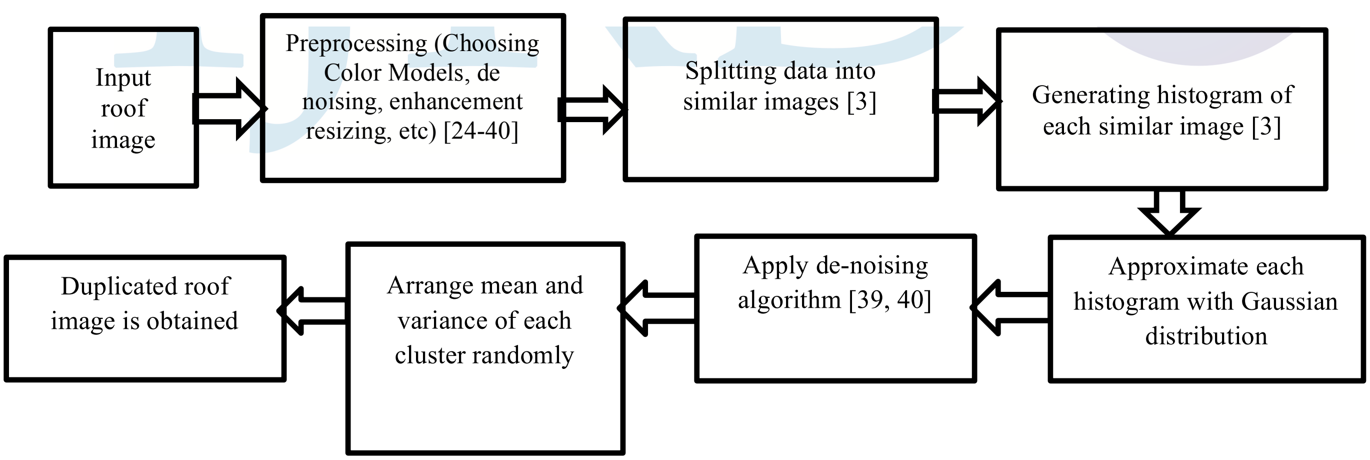


Figure 1: Block diagram for Algorithm 1 - Generating synthetic images by using a single image

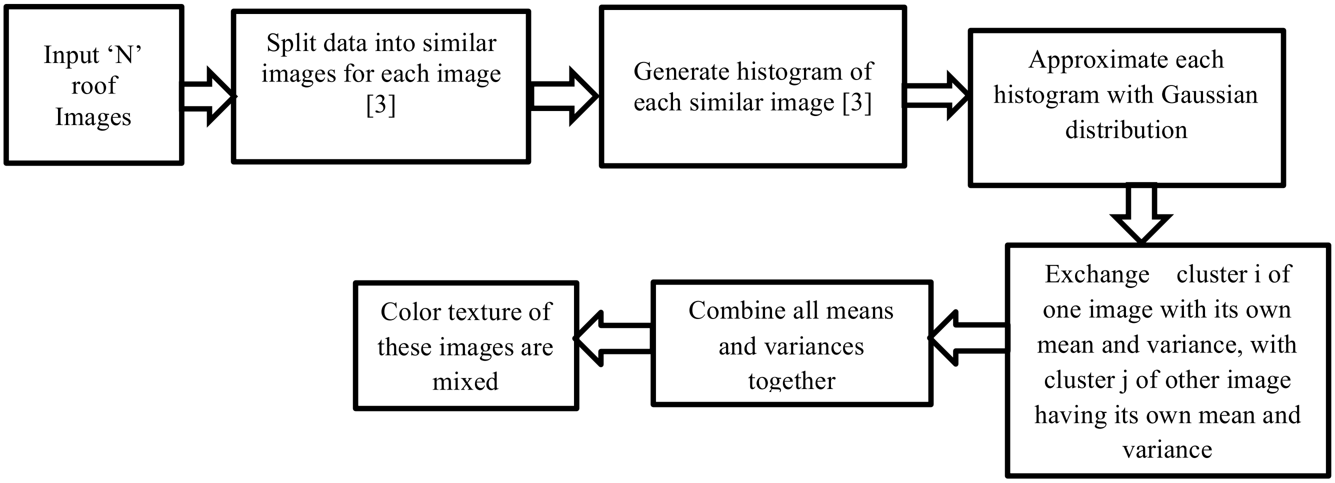


Figure 2 : Block diagram for algorithm 2 - Generating synthetic images by using multiple images

3.1) Algorithm 1: Single image EGMM

For the single image EGMM, assume the image pixels for this image as: Y = $\{y_1, y_2, ..., y_N\}$. The general steps for the EGMM are as follows:

Step 1: Preprocessing techniques like Choosing Color Models, de noising, enhancement resizing, etc.

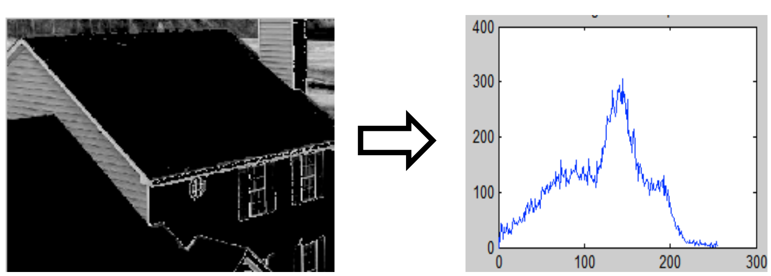
Step 2: Decompose given data Y by combination of "similar" data by using k-mean *(k*-means clustering aims to partition the *n* observations into *k* each clustering sets **S** = $\{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares (sum of distance functions of each point in the cluster to the K center).

We use the following k-mean algorithm to determine the clustered data set:

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)=k}\left\|y_i - y_j\right\|^2 = \sum_{k=1}^{K} C_k \sum_{C(i)=k}\left\|y_i - \mu_k\right\|^2 \qquad (3)$$

where $y_1, \ldots, y_N$ are data points or vectors of observations. Each observation (vector y_i) will be assigned to one and only one cluster, $C(i)$ denotes cluster number for the i^{th} observation

Step 3: Calculate the histogram of each generated cluster



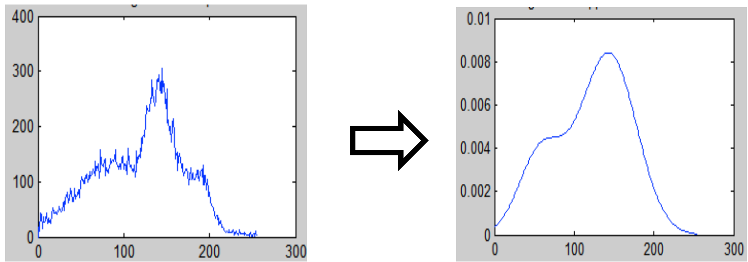
Cluster Histogram

Figure 3: Generation of histogram of the cluster

Step 4: Approximate the generated histogram from step 3 to a statistical Gaussian distribution using the formula:

$$N(x,\mu,\sum) = N(x,\theta) = \frac{1}{\sum_k \sqrt{2\pi}} e^{-\frac{(x-\mu_k)^2}{2\sum_k{}^2}} \quad (4)$$

Where $\mu, \sum$ are the means and variances respectively.



Histogram   Approximation using Gaussian distribution

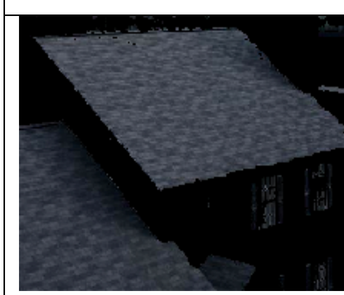
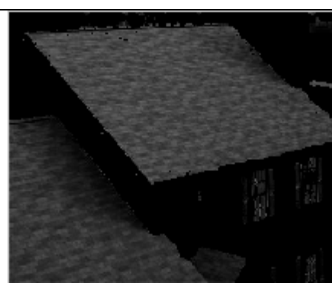
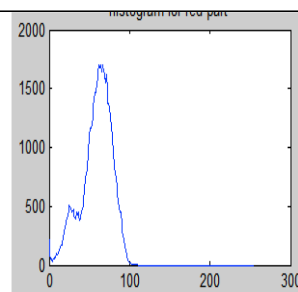
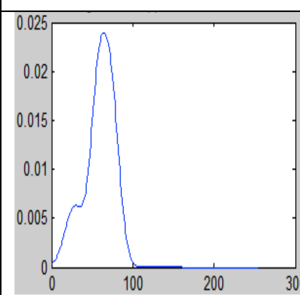
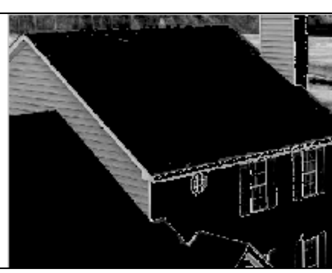
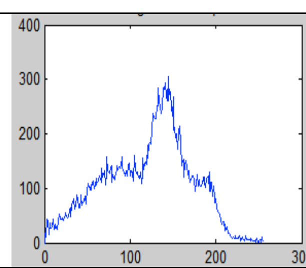
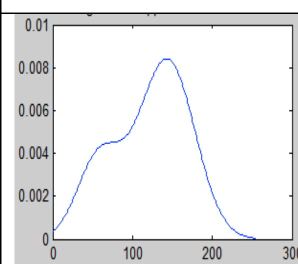
Figure 4: Approximation of histogram by Gaussian distribution

This process of clustering, histogram and its corresponding Gaussian approximation is shown in the table below. Since the input images are RGB, we have shown this separately for Red, Green and Blue part of each cluster

Table 1: Input Images

		
Input Image 1	Input Image 2	Input Image 3

Table 2 : The below table shows results for k-means clustering with the number of clusters equal to 3 for Red part

Total Number of Clusters= 3	Red part	Histogram	Gaussian Approximation
 Cluster 1			
			

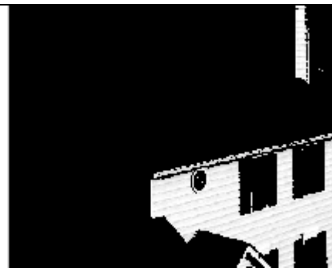
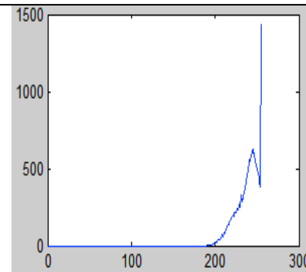
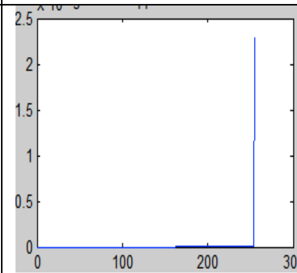
Cluster 2			
 Cluster 3			

Table 3 : The below table shows results for k-means clustering with the number of clusters equal to 3 for Green part

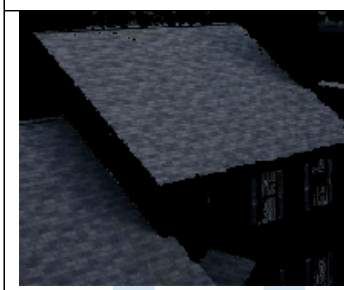
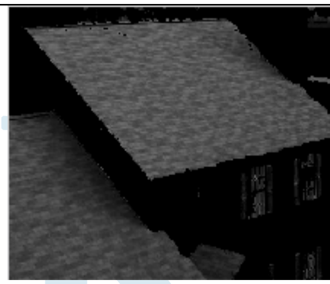
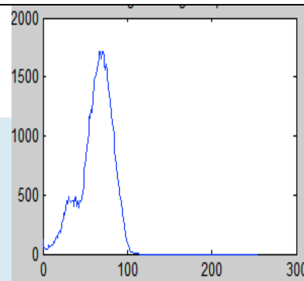
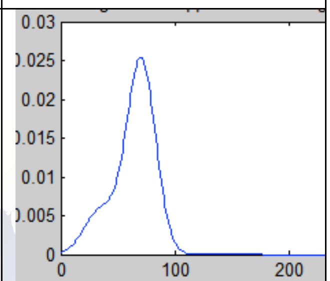
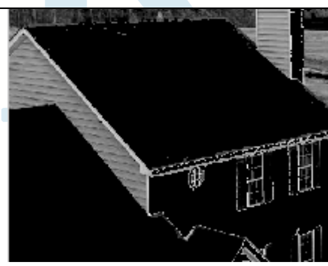
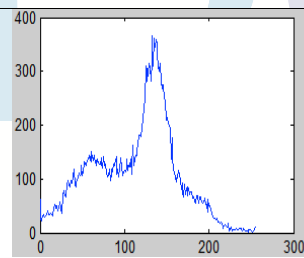
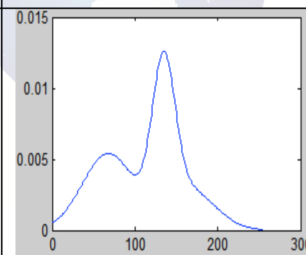
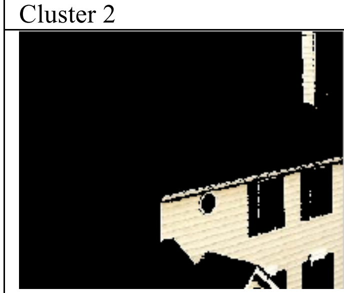
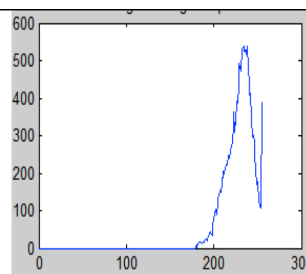
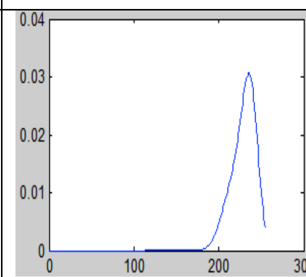
Total Number of Clusters= 3	Green part	Histogram	Gaussian Approximation
 Cluster 1			
 Cluster 2			
 Cluster 3			

Table 4 : The below table shows results for k-means clustering with the number of clusters equal to 3, for Blue part

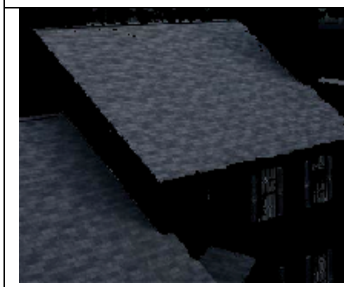
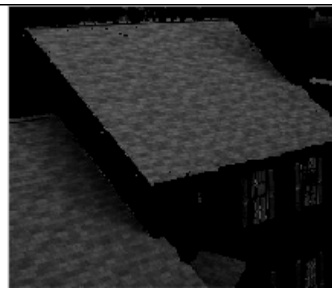
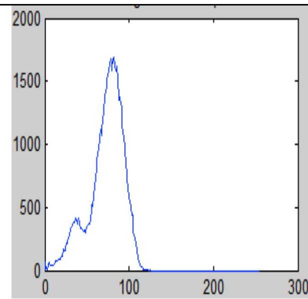
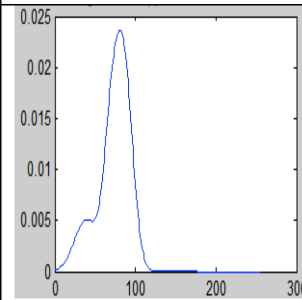
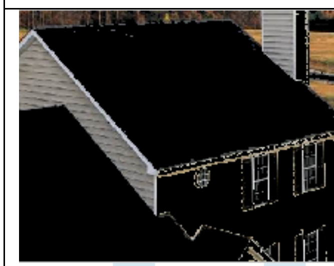
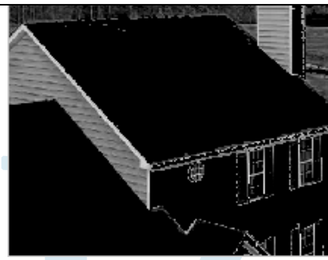
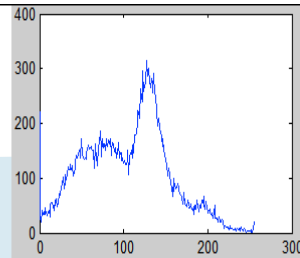
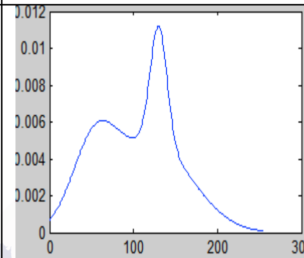
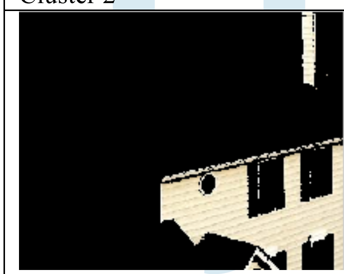
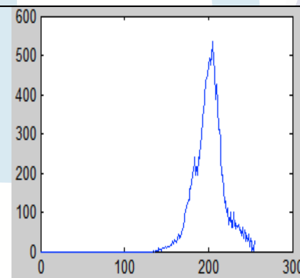
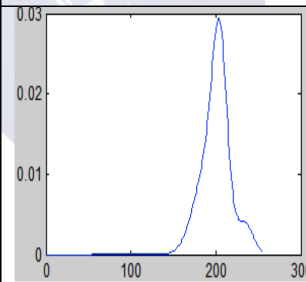
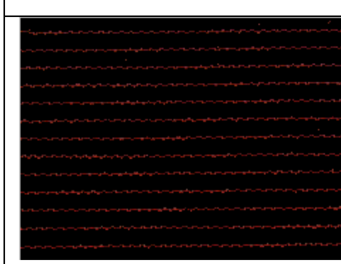
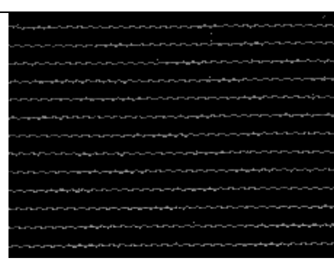
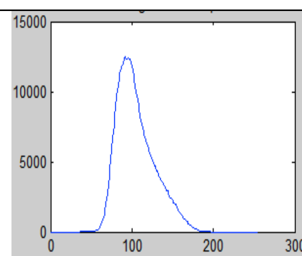
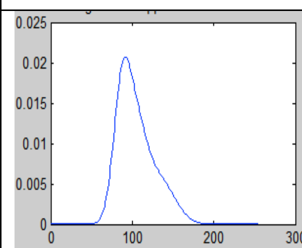
Total Number of Clusters= 3	Blue part	Histogram	Gaussian Approximation
 Cluster 1			
 Cluster 2			
 Cluster 3			

Table 5 : The below table shows results for k-means clustering with the number of clusters equal to 3, for Red part

Total Number of Clusters= 3	Red part	Histogram	Gaussian Approximation
 Cluster 1			

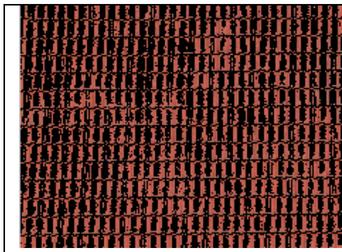
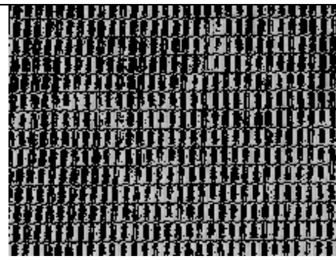
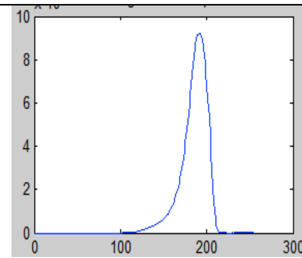
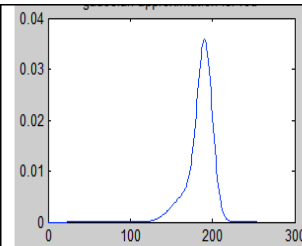
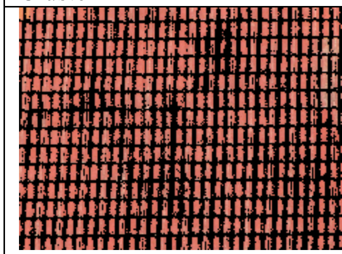
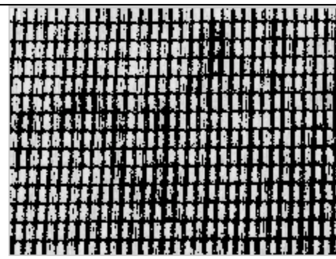
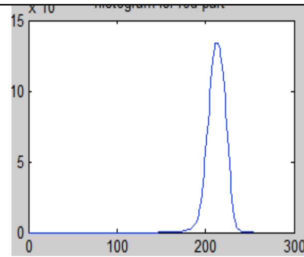
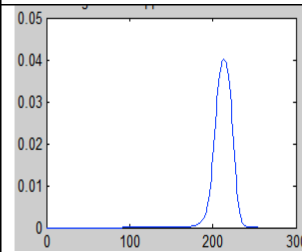
 Cluster 2			
 Cluster 3			

Table 6 : The below table shows results for k-means clustering with the number of clusters equal to 3, for Green part

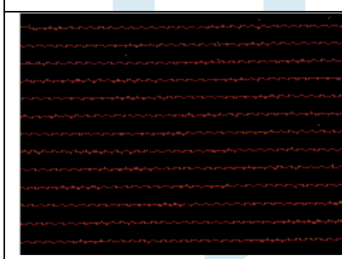
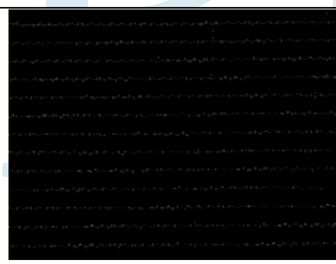
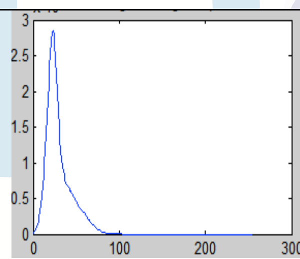
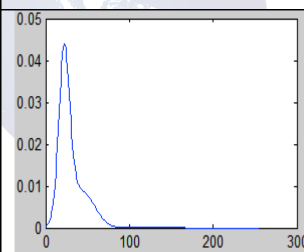
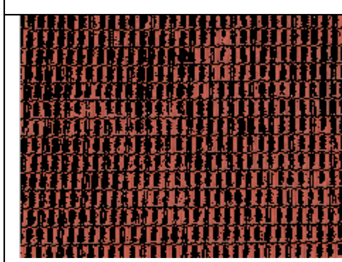
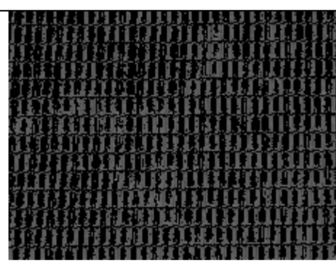
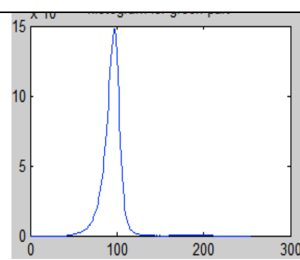
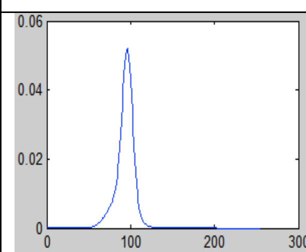
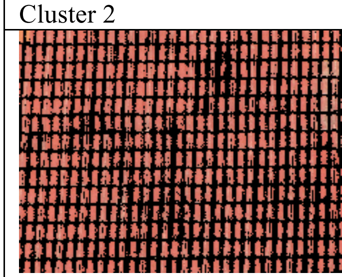
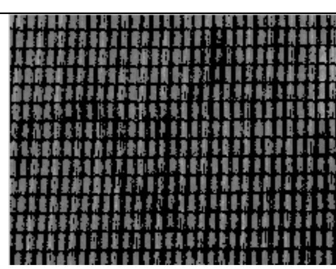
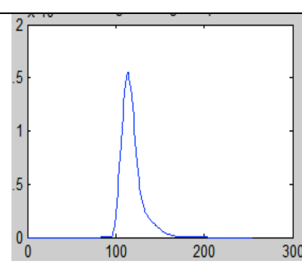
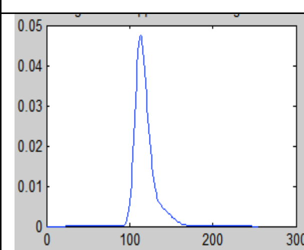
Total Number of Clusters= 3	Green part	Histogram	Gaussian Approximation
 Cluster 1			
 Cluster 2			
 Cluster 3			

Table 7 : The below table shows results for k-means clustering with the number of clusters equal to 3, for Blue part

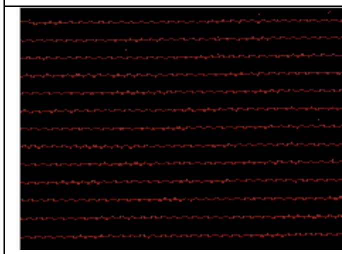
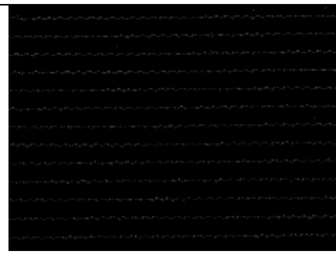
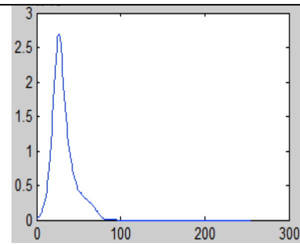
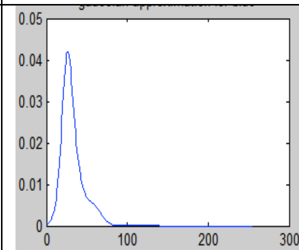
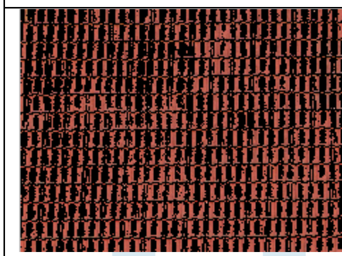
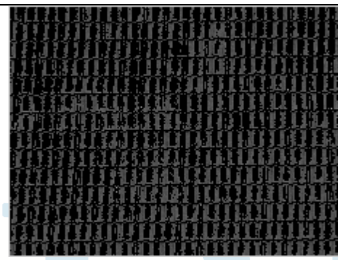
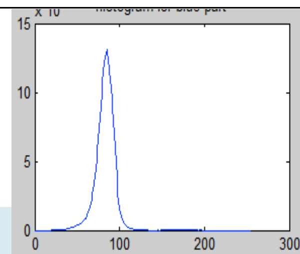
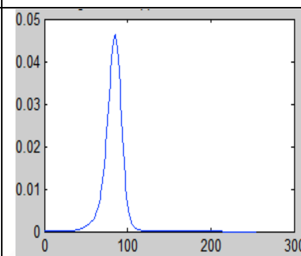
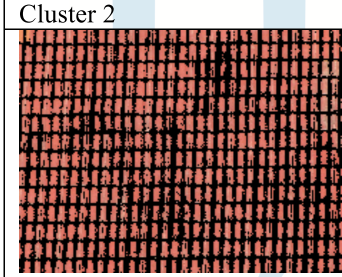
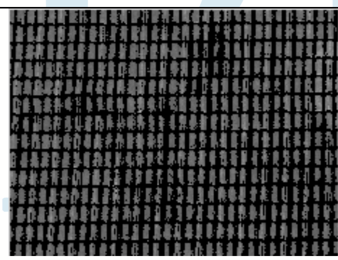
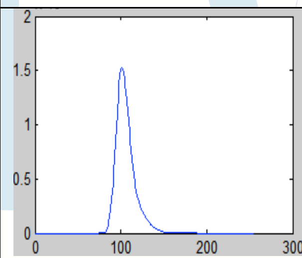
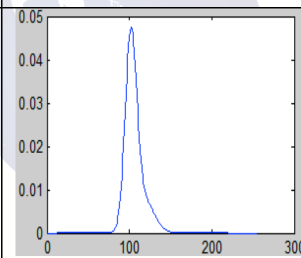
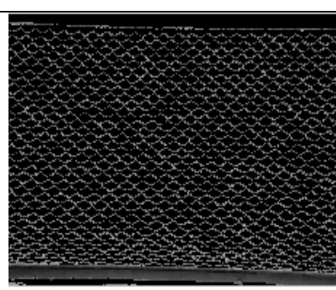
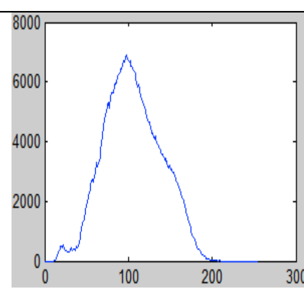
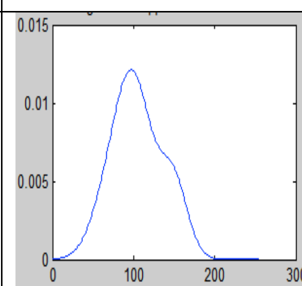
Total Number of Clusters= 3	Blue part	Histogram	Gaussian Approximation
 Cluster 1			
 Cluster 2			
 Cluster 3			

Table 8: The below table shows results for k-means clustering with the number of clusters equal to 3, for Red part

Total Number of Clusters= 3	Red part	Histogram	Gaussian Approximation
 Cluster 1			

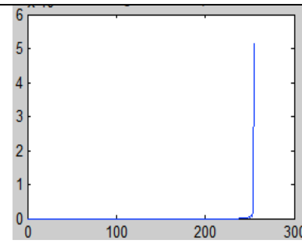
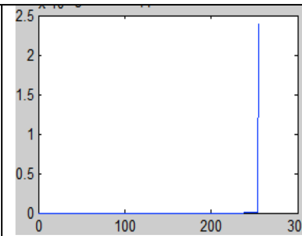
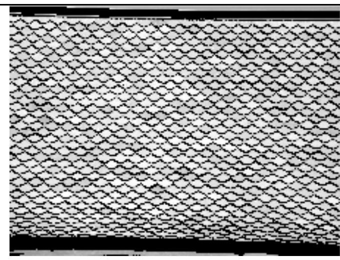
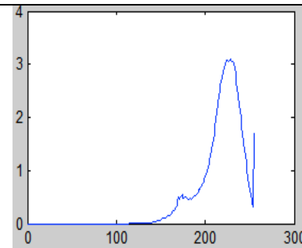
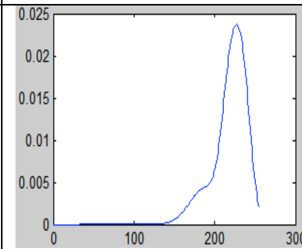
 Cluster 2			
 Cluster 3			

Table 9 : The below table shows results for k-means clustering with the number of clusters equal to 3, for Green part

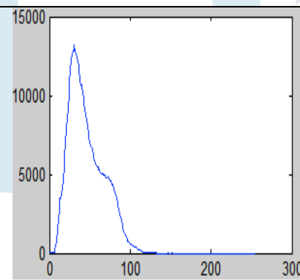
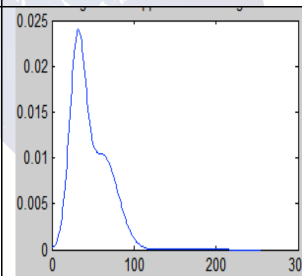
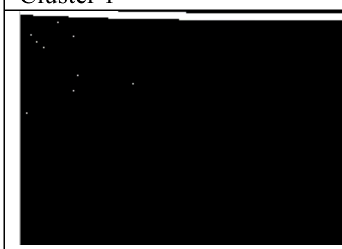
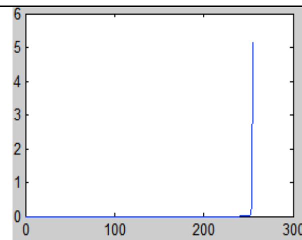
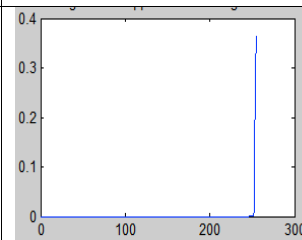
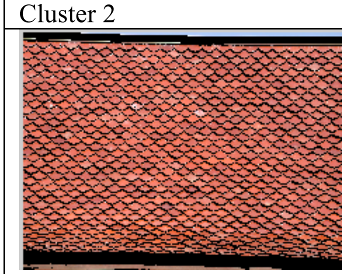
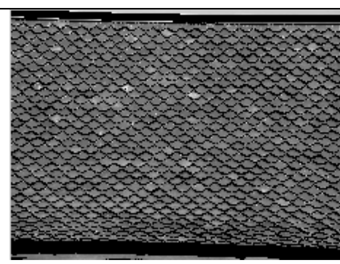
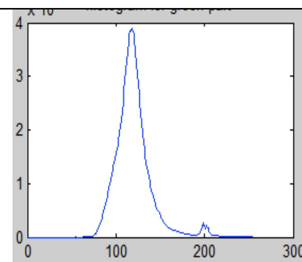
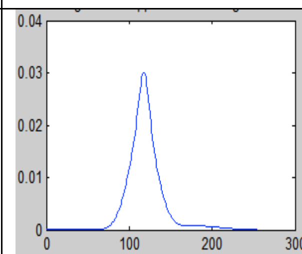
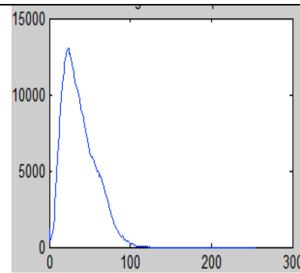
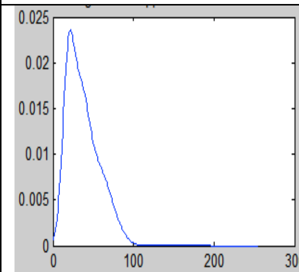
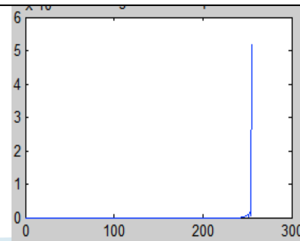
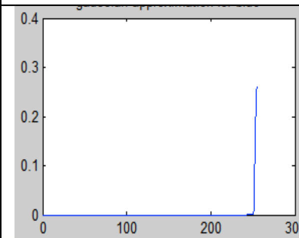
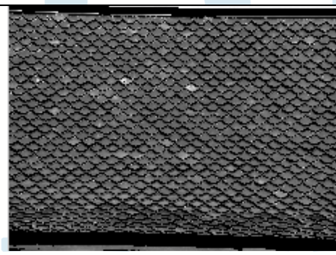
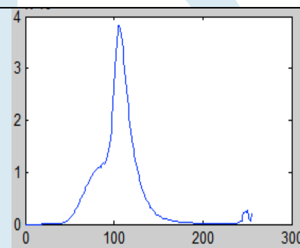
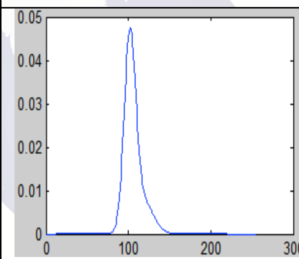
Total Number of Clusters= 3	Green part	Histogram	Gaussian Approximation
 Cluster 1			
 Cluster 2			
 Cluster 3			

Table 10: The below table shows results for k-means clustering with the number of clusters equal to 3, for Blue part

Total Number of Clusters= 3	Blue part	Histogram	Gaussian Approximation
 Cluster 1			
 Cluster 2			
 Cluster 3			

Step 5: De-noising algorithm and image enhancement algorithm like histogram equalization is applied at this step. We also use EM algorithm for this purpose

Step 6: The Probabilistic Parametric Model or Gaussian Mixture Models component ($\mu,\Sigma,\Pi k$) are calculated using Expectation Maximization (EM) algorithm.
EM consists of two steps:
6.1 Expectation step: the new parameters are estimated using the observed data and current estimates of model parameters
6.2 Maximization step: The likelihood function is maximized under the assumption that we know the old parameters
We describe the EM algorithm by the following [33]:

Initialize parameters:

1) Start with initial parameter set $\theta(0)$.

$$\theta= \{u_0 , \Sigma_0 , m_1 , \Sigma_1 , \ldots u_k , \Sigma k, u_2 , \Sigma_2 , \Pi_{k+1},\ldots,\Pi_0\}$$

2)E-step: At the *i*th iteration, we have $\theta^{(t)}$,the conditional expectation is, $Q(\theta \mid \theta') = E[\ln P(X,Y \mid \theta) \mid Y,\theta']$, Where X are the configuration of labels,$X = \{x_1, x_2, ..., x_N\}$.

$$\Sigma_k = \frac{1}{N_k}\sum_{i=1}^{n}\pi_i (x_i - \mu_k) x_i (x_i - \mu_k)^T \qquad \mu_k^n = \frac{1}{N_k}\sum_{i=1}^{n}\pi_i x_i$$

(5)

3) M-step: Now maximize $Q(\theta \mid \theta^t)$ to obtain the next estimates $\theta^{t+1} = \arg\max[Q(\theta \mid \theta^t)]$

***Step** 7*: Arrange these means and variances generated randomly.

Step 8: Duplicated roof image is obtained as an output

3.2) Algorithm 2: Multiple image EGMM

In this section, we present algorithm 2 which generates synthetic images by using multiple images:

***Step 1*:** Consider 'N' Input roof images.

***Step 2*:** Perform k-means clustering and generate clusters [3].

Step 3: Generate histogram of each cluster of each input image [3].

Step 4: Approximate each histogram with Gaussian distribution

Step 5: Exchange cluster i of one image with its own mean and variance, with cluster j of other image having its own mean and variance.

Step 6: Combine all these means and variances together for that particular image that we want the textures of other images to be mixed with.

Step 7: Color texture of these images are mixed.

4. Computer Simulations:

The performance of the algorithm for image duplication was evaluated by an Image Similarity Measure. There are many image similarity measures to compare two images [11, 25-28]. These measures are considered improvements on commonly used measure, such as MSE and $PSNR = 10\log_{10}\frac{L^2}{MSE}$, which have been shown to be inconsistent with perception of the human eye. In this article, we use a Structural Similarity Image measure [11]. An 8x8 image block moves a single pixel x_i at a time across an image. At pixel x_i, a local SSIM score is calculated.

$$SSIM = |L(x,y)|^{\alpha} |C(x,y)|^{\beta} |S(x,y)|^{\chi} \qquad (6)$$

Where

$\mu_k^n = \frac{1}{N_k}\sum_{i=1}^{n} x_i$ is the mean of the image (practically it is the estimate of image intensity).

$\sigma_x = (\frac{1}{N-1}\sum_{i-1}^{N}(x_i - \mu_x)^2)^{0.5}$ is the standard deviation at pixel x_i (practically it is the estimate of image contrast)

$$L(x,y)=\frac{(2\mu_x\mu_y+C_1)}{(\mu_x^2+\mu_y^2+C_1)}$$ is the luminance comparison component

$$C(x,y)=\frac{(2\sigma_x\sigma_y+C_2)}{(\sigma_x^2+\sigma_y^2+C_2)}$$ is the contrast comparison component

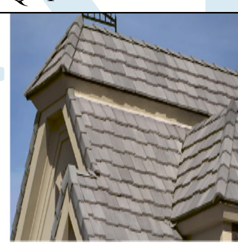
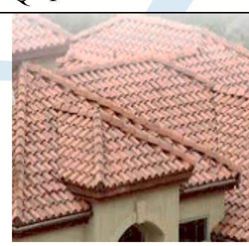
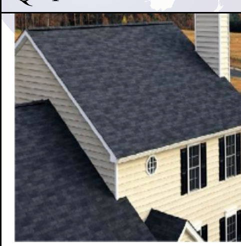
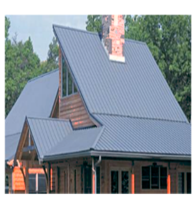
$$S(x,y)=\frac{(\sigma_x\sigma_y+C_3)}{(\sigma_x^2+\sigma_y^2+C_3)}$$ is the contrast structural component

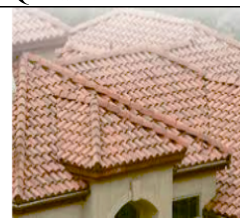
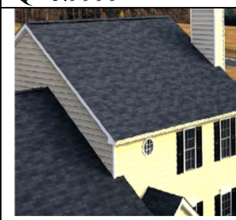
and where C_1, C_2 , C_3 are close to zero constant.

Note that

S(x,y) = 1 if and only if x=usually parameters α,β,γ are used to regulate the relative importance of the three components

Table 11: Gaussian Mixture Models for image duplication output by varying the parameters k (Number of clusters), g (number of Gaussians used for approximation) and EM (number of iterations for the Expectation Maximization algorithm)

Sr. No.	Algorithm	Image 1	Image2	Image 3	Image 4
1	Original Image				
SSIM Index		Q=1	Q=1	Q=1	Q=1
2	GMM with k=10,g=10 and em=1				
SSIM Index		Q=0.9857	Q=0.9993	Q=0.9935	Q=0.9857
3	GMM with k=3,g=10 and em=1				
SSIM Index		Q=0.9999	Q=0.9999	Q=0.9987	Q=0.9999
4	GMM with k=10,g=3 and em=1				

SSIM Index		Q=0.9961	Q=0.9999	Q=0.9870	Q=0.9961
5	GMM with k=5,g=5and em=3				
SSIM Index		Q=0.9999	Q=0.9999	Q=0.9994	Q=0.9999
6	GMM with k=4,g=5 and em=4				
SSIM Index		Q=0.9948	Q=0.9999	Q=0.9994	Q=0.9948

From the above table we can see that the models obtained are visually similar to the original image. In fact, it is not possible to determine if the model is artificial or original when it is viewed. The database for these images can be obtained by varying the parameters like number of clusters, number of Gaussians and number of iterations required for the EM algorithm. We get the best fit statistical model for the roof images which have the minimum pixel distances as compared to the original images. The algorithm implemented gives the best results up to 99% of the original image for its similarity. While this algorithm forms a starting point for application purposes, refinements can be envisioned for future work to improve the quality.

The below table shows, the mixing of textures for 3 input images by exchanging their clusters. The models are named as A1->B1C1 which implies that the first cluster of image 1 having its own mean and variance is replaced by the addition of the first clusters of image2 and image3 with their own means and variances

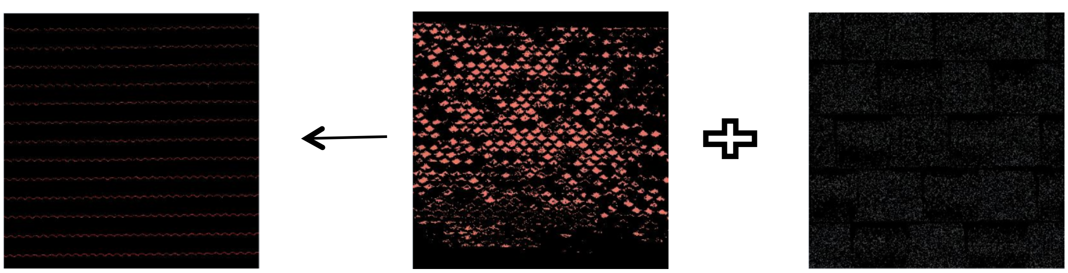
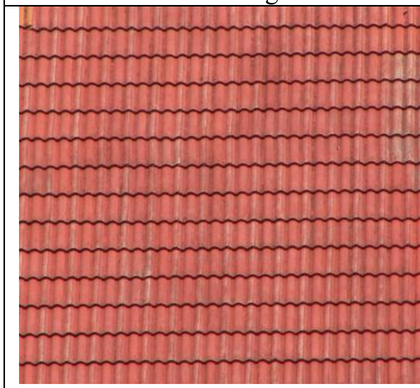
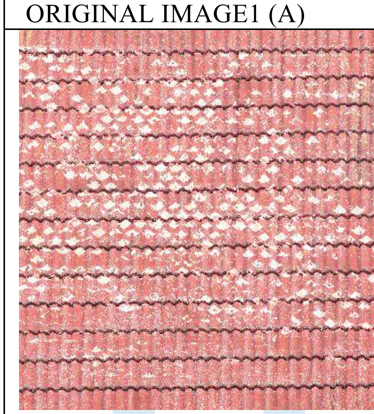
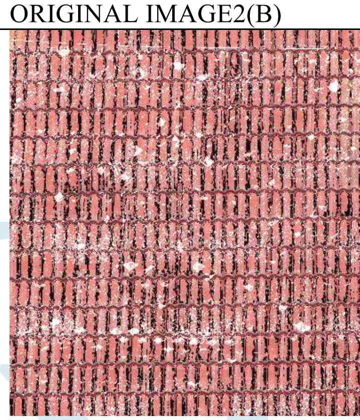
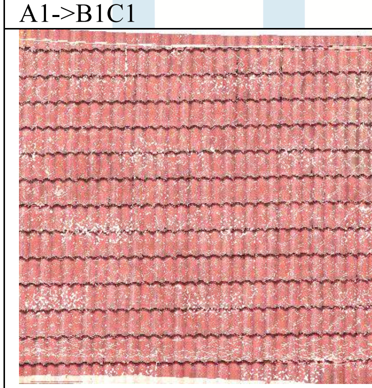
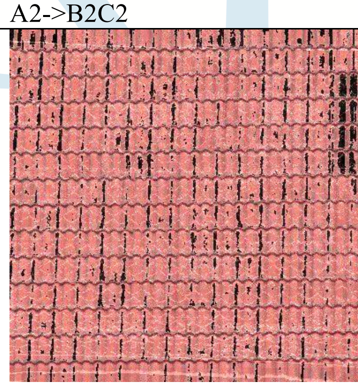
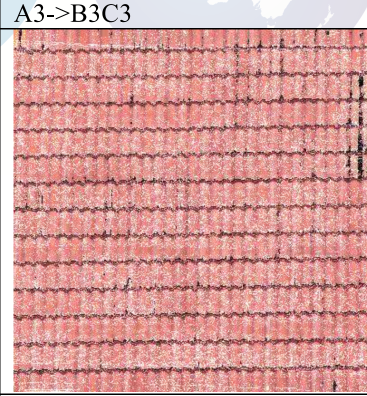
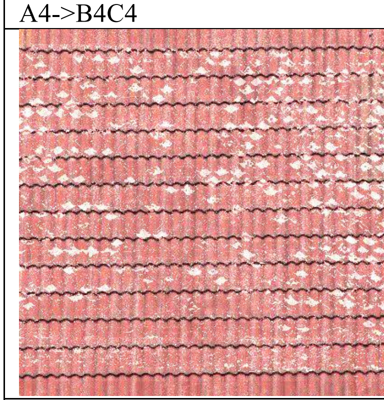
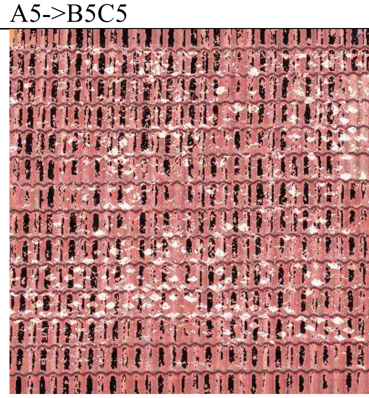
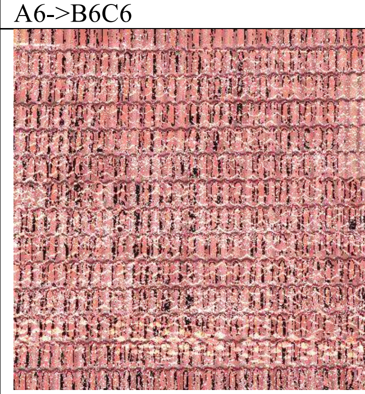
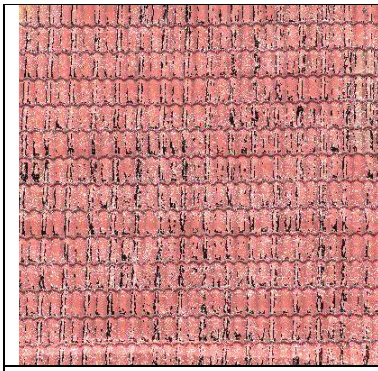
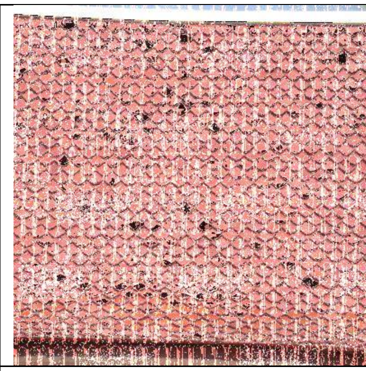
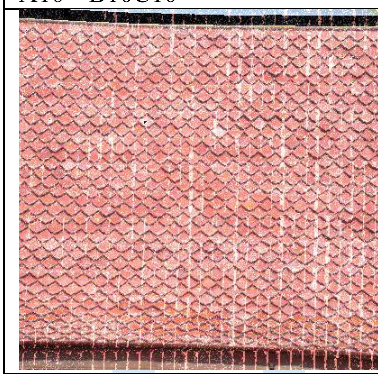
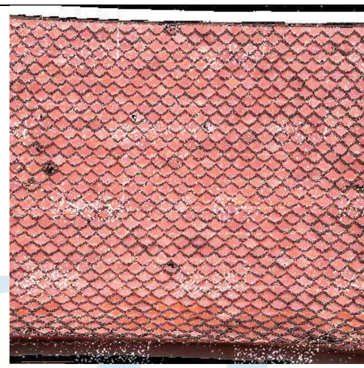
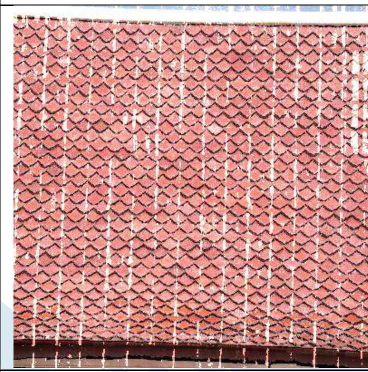
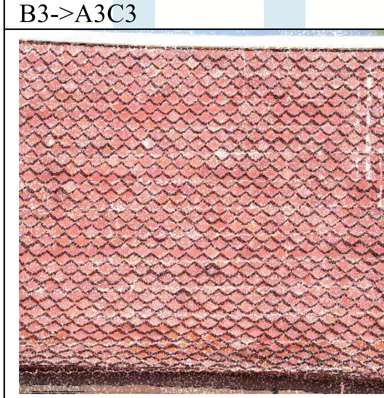
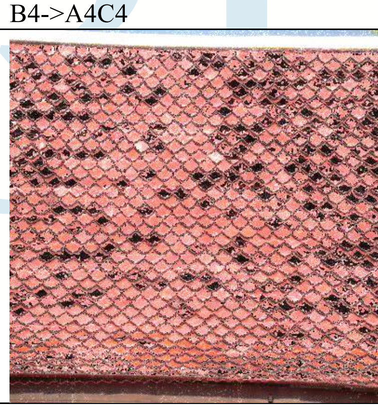
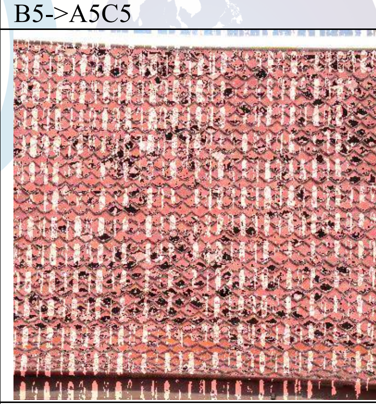
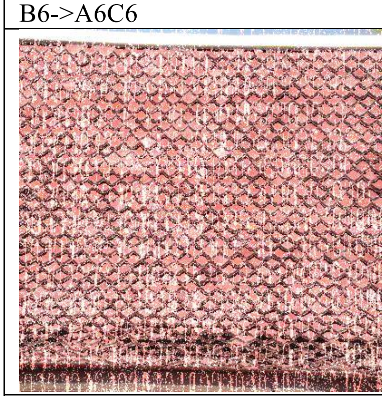
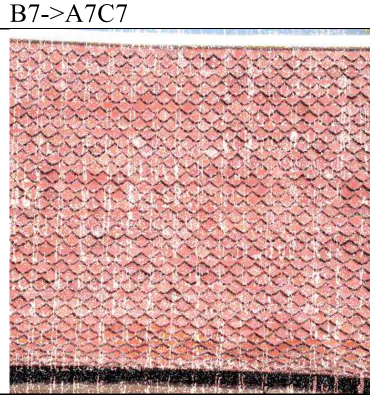
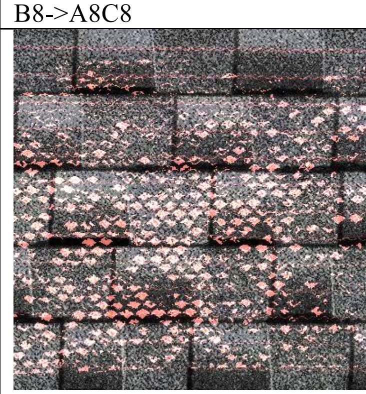
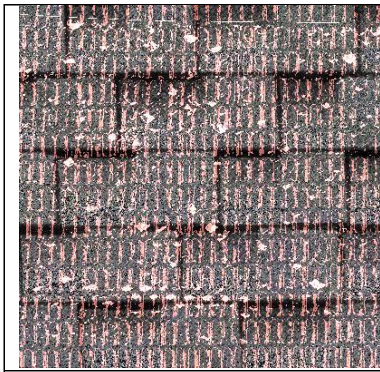
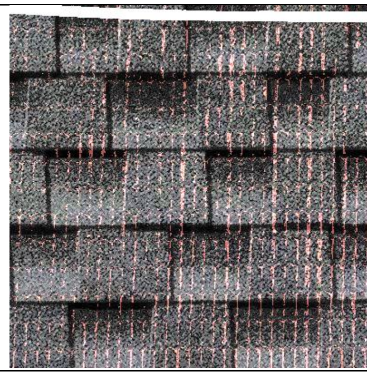
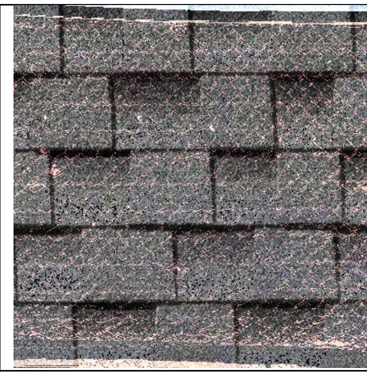
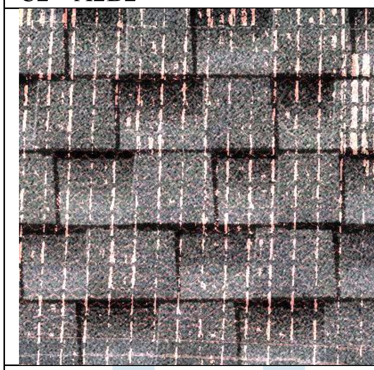
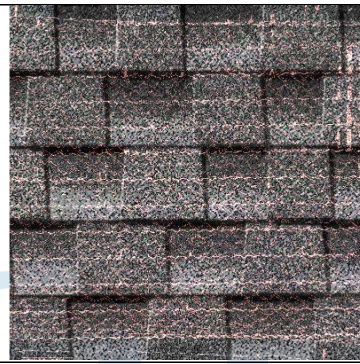
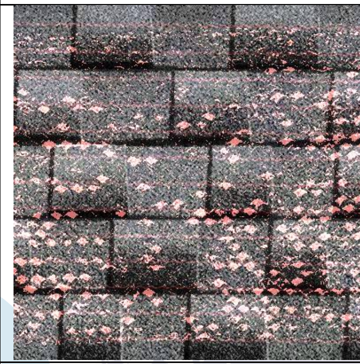
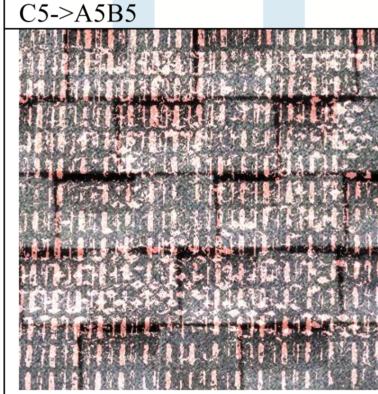
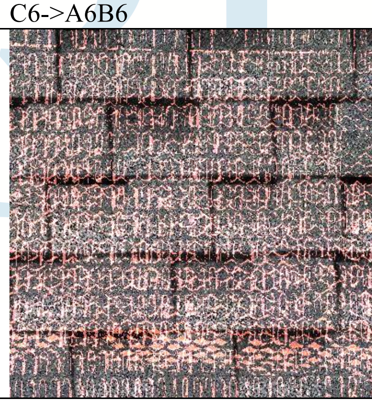
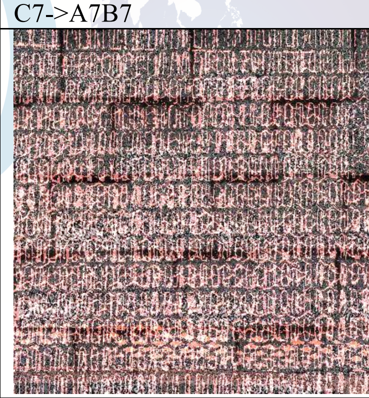


Figure: Cluster 1 of each input image

As seen from above figure, cluster one of image A is replaced by the addition of first clusters of image B and image C

Table No. 12 : Mixing of texture colors to generate new texture		
 ORIGINAL IMAGE1 (A)	 ORIGINAL IMAGE2(B)	 ORIGINAL IMAGE3(C)
		
A1->B1C1	A2->B2C2	A3->B3C3
		
A4->B4C4	A5->B5C5	A6->B6C6
		
A7->B7C7	A8->B8C8	A9->B9C9

		
A10->B10C10	B1->A1C1	B2->A2C2
		
B3->A3C3	B4->A4C4	B5->A5C5
		
B6->A6C6	B7->A7C7	B8->A8C8
		
B9->A9C9	B10->A10C10	C1->A1B1

		
C2->A2B2	C3->A3B3	C4->A4B4
		
C5->A5B5	C6->A6B6	C7->A7B7
		
C8->A8B8	C9->A9B9	C10->A10B10

The above table shows various mixing of textures where we can see some physical features of two images appearing in the third image using the concept of the Gaussian Mixture Model. Roof images mostly contain texture, so we focus on this texture part and generate new textures that are the combination of several other textures containing visual features of them. The different textures for the images that are affected by storms or any other alterations can be generated with this technique.

5.1) Visual evaluation:

In this experiment, 16 images (shown in the appendix) were taken to perform the survey. Each of these 16 images was considered separately. Participating subjects were trained with other images about which one is real and which one is synthetic. For each image in the experiment, its GMM was generated, which we call a non-real image for a particular value of k, EM & no. of Gaussians. We varied these parameters to generate 2 more models, resulting in a total of 3 GMM's (Non-Real images).We arranged these 3 images randomly in a row of 5 images, with the remaining 2 places filled with original images (Real image). This procedure was repeated for all the 16images, resulting in a total of 40 Real and 60 Non-Real images. We ask each participant in the experiment to visually analyze these 100 images and identify them as real or synthetic. The table below shows the number of images that were identified as real or synthetic from the set of images shown in the appendix using both algorithms 1 and algorithm 2. The experiment showed that only 49.65% and 51.56% of students identified synthetic roof images. However, 42.5% and 42.91% of people identified real roof images.

Table 14: Visual evaluation results

Generation of synthetic images by using a single image					Generation of synthetic images by using multiple images				
Image	Identified synthetic roof images (out of 40)	Percentage	Identified real roof images (out of 60)	Percentage	Image	Identified synthetic roof images (out of 40)	Percentage	Identified Real roof images (out of 60)	Percentage
Image 1	16	40	24	40	Image 1	19	47.5	31	51.6
Image 2	22	55	22	36.6	Image 2	20	50	24	40
Image 3	17	42.5	26	43.3	Image 3	17	42.5	24	40
Image 4	19	47.5	28	46.6	Image 4	19	47.5	22	36.6
Image 5	22	55	24	40	Image 5	20	50	30	50
Image 6	26	65	26	43.3	Image 6	26	65	22	36.6
Image 7	22	55	26	43.3	Image 7	17	42.5	30	50
Image 8	15	37.5	28	46.6	Image 8	27	67.5	23	38.3
	Average	49.65		42.5		Average	51.56		42.91

5. Conclusion:

In this article, we have generated synthetic roof images by using single or multiple roof images effectively using Extended Gaussian Mixture Model which visually look similar to that of original images. This technique can be used in generating database of roofs affected by storms or other calamities. A Gaussian mixture model is a probabilistic model in which all the data points are generated from a mixture of a number of Gaussian distributions with unknown parameters which are further estimated. We get the best fit statistical model for the roof images which have the minimum pixel distances as compared to the original images. The algorithm implemented gives the best results up to 99% for SSIM index which is a measure used in evaluating the performance of the duplicate models. These models are further used in mixing the textures of images and generate several multiple textures using multiple input images. From the performance evaluation survey, we found students have identified synthetic images as real images are more. The future work includes comparing these textures with the state of art algorithms using some kind of similarity measure.

References:

1. C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, “Pfinder: Real-time tracking of the human body,” IEEE Trans. on PAMI, vol. 19, no. 7, pp. 780–785, 1997.
2. N. Friedman and S. Russell, “Image segmentation in video sequences: A probabilistic approach,” In Proceedings Thirteenth Conf. on Uncertainty in Artificial Intelligence, 1997.

3. C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," In Proceedings CVPR, pp. 246–252, 1999.
4. P. J. Withagen, K. Schutte, and F. Groen, "Likelihood-based object tracking using color histograms and EM," In Proceedings ICIP, USA, pp. 589–592, 2002.
5. S. Agaian, S. Bakhtiari and M. Jamshidi, "Local Fingerprint Image Reconstruction Based on Gabor Filtering". Proc. SPIE 8406, Mobile Multimedia/Image Processing, Security, and Applications, 10.1117/12.918384, 2012
6. A. Grigoryan, S.Agaian, "Transform-based image enhancement algorithms with performance measure", Advances in Imaging and Electron Physics- 130,165-242,2004
7. E.A Silva, K Panetta, S. Agaian," Quantifying image similarity using measure of enhancement by entropy", Defense and Security Symposium, 65790U-65790U-12,2007
8. E Wharton, S Agaian, K. Panetta, "Comparative study of logarithmic enhancement algorithms with performance measure", Electronic Imaging International Society for Optics and Photonics, 606412-606412-12,2006
9. B.Silver, S.Agaian, and K. Panetta, "Contrast Entropy Based Image Enhancement and Logarithmic Transform Coefficient Histogram Shifting," Proceedings, IEEE ICASSP, 7803-8874, 2005.
10. B.Silver, S. Agaian, and Karen A. Panetta, "Logarithmic Transform Coefficient Histogram Matching with Spatial Equalization," Proceedings, SPIE Defense and Security Symposium, pp237-249, 2005.
11. Wang, Zhou, and Alan C. Bovik. "A universal image quality index." Signal Processing Letters, IEEE 9.3,pp 81-84,2002.
12. S. Agaian, Karen Panetta, and. A.Grigoryan, "A New Measure of Image Enhancement," IASTED International Conference on Signal Processing and Communication, Marbella, Spain, 19-22, 2000.
13. A. Grigoryan and S. Agaian, "Image Enhancement", Advances in Imaging and Electron Physics, Academic Press, pages 165-243, May 2004.
14. A.Monnet, A.Mittal, N.Paragios and V. Ramesh, "Background Modeling and Subtraction of Dynamic Scenes", In Proceedings ICCV'03, pp.1305–1312, 2003.
15. Z.Zivkovic and F.van der Heijden, "Recursive Unsupervised Learning of Finite Mixture Models", IEEE Trans. on PAMI, vol.26 pp 651-656, 2004.
16. Efros, Alexei A., and Thomas K. Leung. "Texture synthesis by non-parametric sampling." Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on. Vol. 2. pp 1033-1038, 1999.
17. Elguebaly, Tarek, and Nizar Bouguila. "Bayesian learning of finite generalized Gaussian mixture models on images." Signal Processing 91.4 801-820.,2011
18. Lee, Te-Won, and Michael S. Lewicki. "The generalized Gaussian mixture model using ICA." International Workshop on Independent Component Analysis (ICA'00).pp 239-244, 2000.
19. Mitra, Sinjini. "Gaussian mixture models for human face recognition under illumination variations." , Applied MathematicsVol.3 No.12A, pp 2071-2079,2012
20. Elgammal, Ahmed, David Harwood, and Larry Davis. "Non-parametric model for background subtraction." Computer Vision—ECCV 2000. Springer Berlin Heidelberg,pp 751-767, 2000
21. S.Bakhtiari,"A Novel Method for Orientation Estimation in Highly Corrupted Fingerprint Images". International Journal of Intelligent Computing in Medical Sciences and Image Processing, 6(1), 45-63.,2014
22. K.Egiazarian, K. Panetta., S.Agaian. "Image Processing: Algorithms and Systems X; and Parallel Processing for Imaging Applications II." Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. Vol. 8295. 2012.
23. Erar, Bahar. "Mixture model cluster analysis under different covariance structures using information complexity." (2011).
24. S.Nercessian , S.Agaian and K. Panetta.," An image similarity measure using enhanced human visual system characteristics", InSPIE Defense, Security, and Sensing . International Society for Optics and Photonics. (pp. 806310-806310),2011
25. S.Nercessian , S.Agaian and K. Panetta.,"A non-reference measure for objective edge map evaluation", In Systems, Man and Cybernetics,. SMC. IEEE International Conference on (pp. 4563-4568). ,2009
26. C.Gao, S.Agaian and K. Panetta. "No reference color image contrast and quality measures." Consumer Electronics, IEEE Transactions on 59.3 643-651. (2013)
27. C.Gao, S.Agaian and K. Panetta,'' Color image attribute and quality measurements.'', In SPIE Sensing Technology+ Applications (pp. 91200T-91200T), 2014.

28. Bensmail, Halima, and Jacqueline J. Meulman. "Model-based clustering with noise: Bayesian inference and estimation." Journal of Classification 20, no. 1 (2003): 049-076.
29. Yang, Ming-Hsuan, and Narendra Ahuja. "Gaussian mixture model for human skin color and its applications in image and video databases." Electronic Imaging. International Society for Optics and Photonics, pp 458-466, 1998.
30. Greggio, N., Bernardino, A., Laschi, C., Dario, P. and Santos-Victor, J. (2010, October)," Unsupervised greedy learning of finite mixture models". In Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on (Vol. 2, pp. 219-224). IEEE.
31. Permuter, Haim, Joseph Francos, and Ian Jermyn. "A study of Gaussian mixture models of color and texture features for image classification and segmentation." Pattern Recognition 39.4 (2006): pp 695-706.
32. Zivkovic, Zoran. "Improved adaptive Gaussian mixture model for background subtraction." In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 2, pp. 28-31. IEEE, 2004.
33. Wang, Quan. "GMM-based hidden markov random field for color image and 3d volume segmentation." arXiv preprint arXiv: 1212.4527 (2012).
34. Reynolds, D.A.: A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification. PhD thesis, Georgia Institute of Technology (1992)
35. Cappelli, Raffaele, D. Maio, and D. Maltoni. "SFinGe: an approach to synthetic fingerprint generation." In International Workshop on Biometric Technologies (BT2004), pp. 147-154. 2004.
36. Ferradans, Sira, Gui-Song Xia, Gabriel Peyré, and Jean-François Aujol. "Optimal transport mixing of gaussian texture models." (2011).
37. Tsin, Yanghai, Yanxi Liu, and Visvanathan Ramesh. "Texture replacement in real images." In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 2, pp. II-539. IEEE, 2001.
38. Shen, Jianbing, Hanqiu Sun, Xiaoyang Mao, Yanwen Guo, and Xiaogang Jin. "Color-Mood-Aware Clothing Re-texturing." In Computer-Aided Design and Computer Graphics (CAD/Graphics), 2011 12th International Conference on, pp. 151-154. IEEE, 2011.
39. Y. Tsin, Y. Liu, and V. Ramesh, "Texture Replacement in Real Images, "Proc. IEEE Conf. Computer Vision and Pattern Recognition,vol. 2, pp. 539-544, Dec. 2001.
40. Liu, Yanxi, Wen-Chieh Lin, and James Hays. "Near-regular texture analysis and manipulation." In ACM Transactions on Graphics (TOG), vol. 23, no. 3, pp. 368-376. ACM, 2004.
41. Fang, Hui, and John C. Hart. "Textureshop: texture synthesis as a photograph editing tool." ACM Transactions on Graphics (TOG) 23, no. 3 (2004): 354-359.
42. Eisenacher, Christian, Sylvain Lefebvre, and Marc Stamminger. "Texture synthesis from photographs." In Computer Graphics Forum, vol. 27, no. 2, pp. 419-428. Blackwell Publishing Ltd, 2008.
43. Liu, Xiaopei, Lei Jiang, Tien-Tsin Wong, and Chi-Wing Fu. "Statistical invariance for texture synthesis." IEEE transactions on visualization and computer graphics 18, no. 11 (2012): 1836-1848.
44. Fang, Hui, and John C. Hart. "Textureshop: texture synthesis as a photograph editing tool." ACM Transactions on Graphics (TOG) 23, no. 3 (2004): 354-359.
45. Liu, Yanxi, Wen-Chieh Lin, and James Hays. "Near-regular texture analysis and manipulation." In ACM Transactions on Graphics (TOG), vol. 23, no. 3, pp. 368-376. ACM, 2004.
46. White, Ryan, and David Forsyth. "Retexturing single views using texture and shading." In European Conference on Computer Vision, pp. 70-81. Springer Berlin Heidelberg, 2006.
47. Agaian, Sos, Jaakko Astola, and Karen Egiazarian, eds. "Binary polynomial transforms and non-linear digital filters" Vol. 191. CRC Press, 1995.
48. Smith, Christopher B., Sos Agaian, and David Akopian. "A wavelet-denoising approach using polynomial threshold operators." IEEE Signal Processing Letters 15 (2008): 906-909.
49. Agaian, Sos S., Rushikesh D. Yeole, Shishir P. Rao, Marzena Mulawka, MIke Troy, and Gary Reinecke. "Missing data reconstruction using Gaussian mixture models for fingerprint images." In SPIE Commercial+ Scientific Sensing and Imaging, pp. 986905-986905. International Society for Optics and Photonics, 2016.
50. Agaian, Sos S., Rushikesh D. Yeole, Shishir P. Rao ,Gary Reinecke, and Mike Troy. "Comparison Study of Gaussian Mixture Models for Fingerprint Image Duplication" Electronic Imaging 2016, no. 15 (2016): 1-7.

APPENDIX

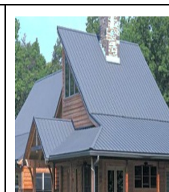
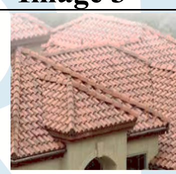
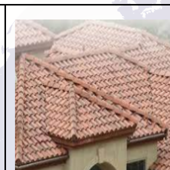
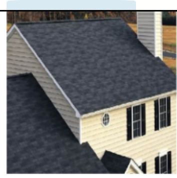
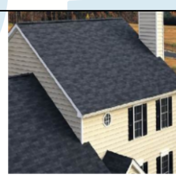
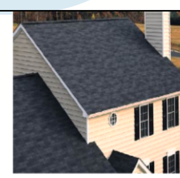
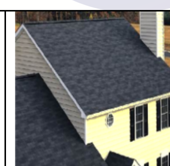
Table No. 15: Images used for visual evaluation of image duplication algorithm				
Image 1				
				
Real	Non-Real	Real	Non-Real	Non-Real
Image 2				
				
Non-Real	Real	Non-Real	Real	Non-Real
Image 3				
				
Non-Real	Non-Real	Non-Real	Real	Real
Image 4				
				
Real	Real	Non-Real	Non-Real	Non-Real
Image 5				
				
Non-Real	Non-Real	Real	Real	Non-Real

Image 6				
				
Non-Real	Non-Real	Non-Real	Non-Real	Non-Real
Image 7				
				
Real	Real	Real	Real	Real
Image 8				
				
Non-Real	Non-Real	Non-Real	Non-Real	Non-Real

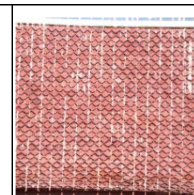
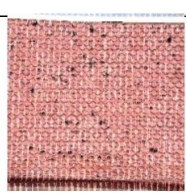
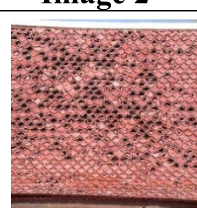
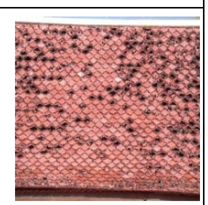
Table No. 16:Images used for visual evaluation for texture mixing algorithm				
Image 1				
				
Real	Non-Real	Real	Non-Real	Non-Real
Image 2				
				
Non-Real	Real	Non-Real	Real	Non-Real

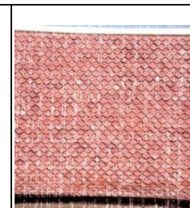
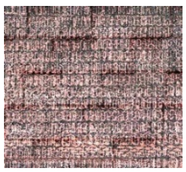
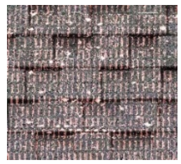
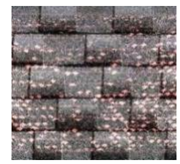
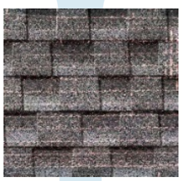
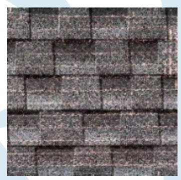
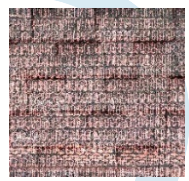
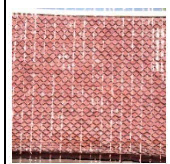
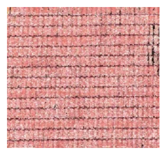
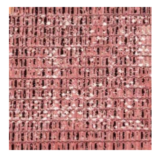
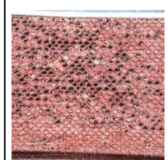
Image 3				
				
Real	Real	Non-Real	Non-Real	Non-Real
Image 4				
				
Real	Non-Real	Non-Real	Real	Non-Real
Image 5				
				
Non-Real	Real	Non-Real	Non-Real	Real

Image 6				
				
Non-Real	Real	Real	Non-Real	Non-Real
Image 7				
				
Real	Real	Non-Real	Non-Real	Non-Real
Image 8				
				
Non-Real	Non-Real	Non-Real	Real	Real