Identification Forgery Image and Image Operation by FCM Algorithm

Hanaa Mohsin Ahmed*, Ph.D (Asst. Prof.)* **Huda Mohammed Eid***

Abstract: Digital images are easy to manipulate and edit due to availability of powerful image processing and editing software. It is possible to add or remove important features from an image without leaving any obvious traces of tampering. The structure of detection forgery image in general, includes some of basic stages, and the most important stage is extract features from image because these features is the basic to detected if an image original or not. In this paper, we give a structural to build identification of anti-forensic detecting using steganalytic approaches feature vector. To identify image anti_forensic and image processing. Where several approaches of steganalysis that depend on feature based steganalytic, one of these is Image Quality Measured (IQM). This goal can be achieved by using Fuzzy C Mean (FCM) and Euclidian distains (EU). Results obtained from testing this system for identify forgery image and image operation was with accuracy of 94.8%.

Keywords: Forgery image, Image operation, image quality measure (IQM), Fuzzy C Mean (FCM), build identification, anti-forensic detecting

* University of Technology

1. Introduction

Recently, the digital images have replaced the well-known analog photographs, and forging digital became an easier task quite hard to discover[1]. Images are utilized in a form of authentication proof for crimes and if those images don't stay unique, then it will cause an issue. As forging images is growing daily, it's very important developing ways of detecting as which image is genuine and which one has been forged[2]. The detection of those kinds of forgeries turned out to be a serious issue[3]. Recognizing whether a digital image is genuine or not, is a main task of image forensics[4].

Image Forensics is a significant phase of a great deal of investigations[5]. There are two types of approaches for image forensics: one is known as active protection, and the other one is called passive or (blind)[2]. The general kinds of image forgery methods are image splicing, copy-move, resampling, jpeg compression processing and image processing operation. Forgery is deployed basically in order to make tempered photographs are studied in more detail in this paper, for the sake of determining if a digital image is authentic or not is quite a challenge. Finding the traces of tamper in a digital image is not an easy task.The approach of passive or blind forgery detection deploys the received image only to assess its originality or integrity, with no signature nor water-mark of the original image from the sending party. It relies on the presumption that even though digital forgeries may leave no visually apparent traces of having been altered, they could very possibly affect the underlying statistics feature or image consistency of a natural scene image that shows new artifacts which result in different kinds of inconsistencies. Those inconsistencies might be utilized for detecting the forgery. This approach is widely used due to the fact that it needs no prior information concerning the image. Existing methods identify different traces of alteration and discover them separately with localizing the altered area[2].

As depicted in figure (1), each of the image processing and Anti-forensic procedures are identical to the procedure of data embedding in steganography, due to the fact that each of them has to alter some pixel values in the original (cover) images. The alteration of pixels would eliminate the inherent correlations among the neighboring pixels in the image. For the seek of detecting the pixel alterations in steganography, a

number of steganalytic properties have been suggested by modeling such inherent features. Those properties are typically efficient even for relatively low rates of modification [7].

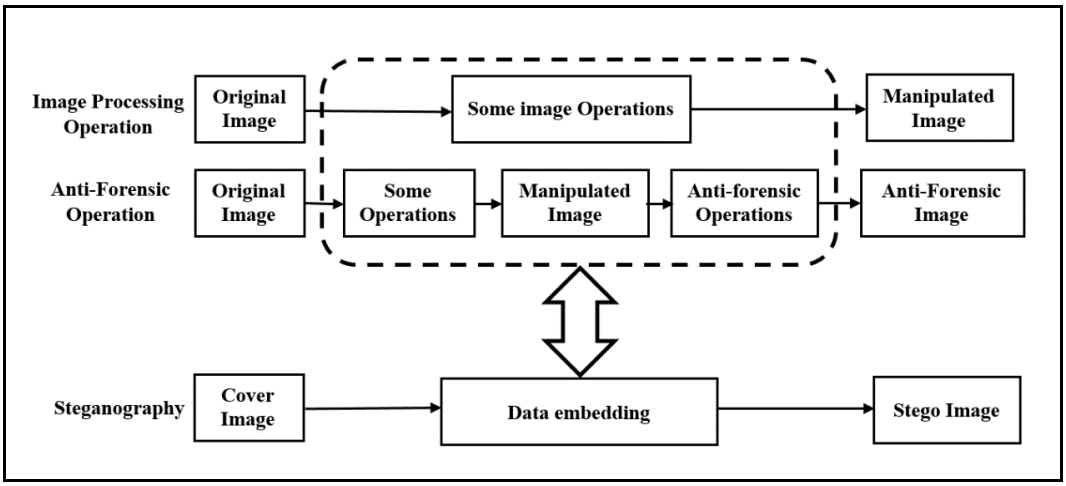


Figure (1): **Image processing operations and anti_forensic operations VS steganography.**(*some operation* in above figure is Contrast enhancement, Sharpening, Spatial Filtering, Lossy compression and Median filtering[7])

2. Related Work

There has been a great deal of studies and literature in the field of identifying forged images and image processing see [8,9]. Below is a group of the most related articles to this paper subject from previous last 6 years:

In 2012 **"Sedighe Ghanbari et al"** has proposed a method to extract features from Gray Level Co-occurrence Matrix (GLCM) which vary between cover image (image with no embedded information) and stego image (image that includes hidden information). In the algorithm that they suggested they first, utilized a combined approach of steganography based on both location and conversion for hiding the data in the original image and name it image-steg1. After that, they hid the data in imagesteg1 again and named it image-steg2. With the use of GLCM matrix features. They research some different properties in the GLCM of the original image and stego image. They show that, they can obtain properties which differ between those images. These properties are

utilized to train the NN and the classification step was performed with the use of 4 layers Multi-Layer Perceptron (MLP) neural network. They performed tests on their algorithm using 800 standard image data-bases and them propose algorithm can have detected 80% of stego images. Thus, their suggested algorithm effectiveness is 80% [10].

In 2013 "**Irene Amerini, et al**" have presented a new approach for the detection of copy-move forgery and localization according to the J-Linkage algorithm that operates a robust clustering in the space of the geometrical transformation. The Results of the experiments, performed on various data-sets, displayed that the suggested approach performs better than other similar state-of-the-art approaches according to both copy-move forgery detection reliability and accuracy in the altered patch localizing [4]. Also in 2013 **"Amani Alahmadi, et al"** have presented a new passive image forgery detecting approach based on Local Binary Pattern (LBP) and Discrete Cosine Transform (DCT) for detecting copy-move and splicing forgery techniques. Initially, from the chrominance component of the input image, discriminative localized properties are deduced via applying two-dimensional DCT in Local Binary Pattern space. After that, Support Vector Machine (SVM) is utilized for the detection. Experiments performed on 3 image forgery benchmark data-sets showed the supremacy of the approach over other previous approaches according to the precision of detection [11].

In 2014 **"Hamza Turabieh, et al**" proposed a detection method for the existence of the LSB insertion in digital images. The least significant bit embedding alters the statistic features of the cover item has been used for designing the detection algorithm. The suggested algorithm benefitted from the properties of the least significant bit planes with the use of the GLCM. Those properties have been utilized for the classification of image portions into stenographic and ordinary cases. The classification stage has been done with the use of 3 layers back-propagation neural network (BPNN). Experimental results were listed in addition to presenting percentile results of each of the false and positive detections [12].

In 2016 **"Gulivindala Suresh and Chanamallu Srinivasa Rao"** they have proposed a work "Copy move forgery detection using GLCM based statistical features" GLCMs are extracted from all the images in the database and statistics such as contrast, correlation, homogeneity and

energy are derived. These statistics form the feature vector. SVM is trained on all these features and the authenticity of the image is decided by SVM classifier. Their proposed work is evaluated on CoMoFoD database, on a whole 1200 forged and processed images are tested. The performance analysis of their work is evaluated with the recent methods [13].

In 2017 **"Ms.G.Clara Shanthi and Dr.V.Cyril Raj"** have proposed, an efficient forgery detection and classification technique by three different stages. At first stage, preprocessing is carried out using bilateral filtering to remove noise. At second stage, extract unique features from forged image by using efficient feature extraction technique namely GLCM. Finally, forged image is detected by classifying the type of image forgery using Multi Class SVM. The performance is analyzed using the following metrics: accuracy, sensitivity and specificity [14].

As concluded from previous work: (1) finding any process would alter any pixel values and therefore definitely eliminate some inherent statistics of the original images, which is identical to the procedure of data hiding. (2) Finding any operation would alter several pixel values and therefore inevitably eliminate some inherent statistics of the original images. (3) Fuzzy system is not used for this prepays as classifier. Accordingly, using the image quality as a measurement of the statistical features of the extraction approaches of steganalysis, as the property vector for identifying the existence of image Anti-forensic or not. fuzzy c mean will be used as a classifier in this paper.

3. General Model of Detecting Image

A generalized model of passive (blind) image forgery detection method is described below. It includes of the following basic stages [15]:

(1) Image pre-processing: Prior to property extracting procedure some processes are done on the considered images, like cropping, transformation RGB image to greyscale, for improving the performance of the classification process.

(2) Extracting Properties: A group of properties are deduced for every one of the classes which helps in distinguishing it from other classes, while staying independent of characteristic differences in the class from the

input forged data. Specifically, extracting informative properties and selecting properties that have to be of high sensitivity to image alterations. One of the preferable properties of the chosen characteristics and constructed feature vector has to be of small dimension that will minimize the computational complexity of training and classifying.

(3) Property pre-processing and the selection of the classifier: according to the obtained group of properties choose or design proper classifiers and select a large group of images for the training of the classifiers. Find some valuable parameters of classifiers that might be used for the classification.

(4) Classification: The goal of this process is discriminating the given images and classifying them into two groups: authentic and tampered images.

(5) Post-processing: In some forgeries such as copying moving and splicing, post-processing procedure deals with localizing the altered area as investigated [15].

Based on the stages depicted above, the framework of blind image forgery detection is displayed in figure (2) [15-16].

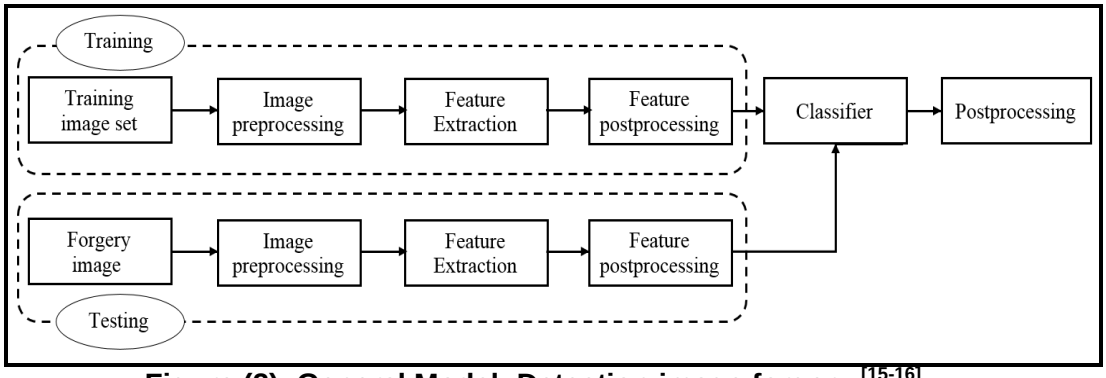


Figure (2): General Model: Detection image forgery [15-16]

4. Image Quality Measured (IQM)

IQM has the objective of using computational structures for measuring the quality of the image consistently with subjective evaluations [17]. A good IQM has to represent the distortion on the image accurately

according to, blurring, compression, additive noise and sensor inadequacy [18]. There are two kinds of IQMs [19]:

4.1 Subjective Image Quality: Subjective measures are the result of human experts giving their opinion concerning image quality but the results are different from one person to another [19].

4.2 Objective Image Quality: Objective measures are done with mathematical algorithms. The objective Image Quality can be classified into full-reference, reduced-reference and no-reference. Full reference criteria are also a function of the original image that is presumed to be distortion-free (known as the “reference image”) [19].

Reduced reference criteria need a partial knowledge of the reference image (this knowledge is known as the “reduced reference”). Finally, the no-reference criteria have no information concerning the reference image [19]. In this paper we used non-reference, and for all IQM equation **s, t** represents row and column of image, and **n*m** is the size of image.

In the in this paper used the following equation to extract feature from images.

4.2.1 Average (Avg): the most common and familiar is the arithmetic mean, defined by [20]:

$$\mathbf{Avg} = \frac{1}{n*m} \sum_{s,t=0}^{(n,m)-1} \mathrm{pix}(s,t) \qquad (1)$$

4.2.2 The standard deviation (SD): The SD is extremely important**.** It is identified to be the square root of the variance. It is a measure of the image contrast and is computed as [20]:

$$\mathbf{SD} = \sqrt{\frac{\sum_{s,t=0}^{n,m} \mathrm{pix}_{(s,t)} - \mathrm{Avg}}{n*m}} \qquad (2)$$

Where**, Avg** see equation (1).

4.2.3. The information Entropy (Ent): The entropy of a discrete random variable **I.** Which is the average (expected) amount of information obtained from an event. Defined as [21]:

$$\mathbf{Ent} = -\sum_{s,t=0}^{(n,m)-1} \mathrm{pix}(s,t)\mathbf{Log_2}\,\mathrm{pix}(s,t) \quad (3)$$

Less amount of information can be extracted in an event with lesser entropy; and more information would be a consequence of greater entropy.

4.2.4 Inverse Difference Moment (IDM): It measures the local homogeneity of an image. It gives bigger values to smaller grey level differences in pixel pairs which is given by equation (4) [22], moreover, it's affected by the image homogeneity. Due to the weighting factor $(1+(s-t)^2)$-1 IDM will get small contributions from inhomogeneous regions (s <> t). The result is a low value of IDM for non-homogeneous images, and a relatively higher value for homogeneous ones [23]:

$$\mathbf{IDM} = \sum_{s,t=0}^{(n,m)-1} \frac{1}{1+(s-t)^2}\{\mathrm{pix}(s,t)\} \quad (4)$$

4.2.5 Contrast (Ctrt): This measurement of contrast or local intensity variation will favor contributions from P (s, t) away from the diagonal, in other words (s<>t) [23].

$$\mathbf{Ctrt} = \sum_{s,t=0}^{(n,m)-1}(s-t)^2\mathrm{pix}(s,t) \quad (5)$$

4.2.6 Energy (Egy): Energy reaches a maximal value of one. High energy values happen when the distribution of the grey level has a constant or periodical form [24].

$$\mathbf{Egy} = \sum_{s,t=0}^{(n,m)-1} Pix(s,t)^2 \quad (6)$$

4.2.7 Correlation (Corr): It is a measurement of grey shade linear-dependency in an image, particularly, the orientation being considered is the same as vector displacement. The values of high correlation (which are approximate to 1) indicate a linear correlation between the grey levels of pixel pairs [24]:

$$\mathbf{corr} = \frac{\sum_{s,t=0}^{n,m} s*t*\mathrm{pix}(s,t) - \mathrm{Avg_n Avg_m}}{\mathrm{SD_n SD_m}} \quad (7)$$

Where Avg_n, Avg_m, SD_n and SD_m are the average and standard deviations of the margin possibilities *Px*(*s*) and *Py*(*t*) found via summing the rows X or the columns Y of matrix *Pix(s,t)***.**

4.2.8 Shade: Is defined [24] as:

$$\textbf{Shade} = \sum_{s,t=0}^{n,m} (s + t - 2\text{Avg})^3 * \text{Pix}(s,t) \quad (8)$$

4.2.9 Maximum Probability (MP): Is defined [24], as:

$$\textbf{MP} = \textbf{MAX}_{s,t}\ \text{pix}(s,t) \quad (9)$$

4.2.10 Homogeneity (**Hom**): Is defined [24], as:

$$\textbf{Hom} = \sum_{s,t=0}^{n,m} \left[\frac{\textbf{pixel}(s,t)}{(1+(s-t))^2}\right] \quad (10)$$

5. Fuzzy C-Means (FCM)

FCM is utilized in fields such as computation geometry, data compressing and vector quantization, pattern identification and classification [25] and [26,27]. In this paper, a simple and sufficient implementation of Fuzzy C-Means clustering approach has been used. The goal of the Fuzzy C-Means is finding the centers of clusters which minimize a dissimilarity function [28]. It is modeled according to the minimizing of the following objective formula [29]:

$$j_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \left|\left|x_i - c_j\right|\right|^2 \quad (11)$$

Where:

m: represents any real number > 1, it was predetermined to 2.00 by Bezdek (1981)

u_{ij}**:** represents the rating of membership of x_i in the cluster ***j***

x_i**:** The ***ith*** of d-dimensional measured data

||*||: Any norm that expresses the resemblance between the center and any measured data.

Fuzzy partitioning is performed via an iterative optimizing of the objective function that has been depicted above, with the update of membership u_{ij} and the c_j cluster centers by [29]:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left| \left| x_i - c_j \right| \right|}{\left| \left| x_i - c_k \right| \right|} \right)^{\frac{2}{m-1}}} \quad (12)$$

$$C = \frac{\sum_{i=1}^{N} u_{ij}^{m} . x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \quad (13)$$

This iteration will terminate when [29]:

$$max_{ij} \{ |u_{ij}^{k+1} - u_{ij}^{k}| \} < \varepsilon \quad (14)$$

ε : denotes the termination criterion between 0 and 1

k : The iteration steps

This process converges to a local minima or a saddle point of J_m. The algorithm is made up of the following stages [29]:

Step 1: Initializing $U = [u_{ij}]$ matrix, $U(0)$

Step 2: At k-step: compute the vectors of the centers $C(k) = [c_j]$ with $U(k)$

See equation (13)

Step 3: Update U(k), U(k+1)***.***

See equation (12)

Step 4: If $||U(k+1) - U(k)|| < \varepsilon$ then STOP; or else go back to step 2.

6. Euclidean Distance (EU)

Known as the L_2 distance as well. If $\boldsymbol{u} = (x_1, y_1)$ and $\boldsymbol{v} = (x_2, y_2)$ are two points, then the Euclidean distance between them is calculated by [30]:

$$\mathbf{EU}(\boldsymbol{u},\boldsymbol{v}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad (15)$$

Rather than 2 dimensions, in the case where the points have ***n*** dimensions, like *a*= *(x_1, x_2, x_3…. x_n)* and *b=(y_1,y_2, y_3….y_n)* then, equation (15) may be generalized via defining the Euclidean distance between *a* and *b* as [30]:

$$\mathbf{EU}(\boldsymbol{u},\boldsymbol{v}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots\dots (x_n - y_n)^2} = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2} \quad (16)$$

In this paper we used FCM to find classes center and we used EU to classification

7. Some Types of Image Forgery and Image Processing

Digital image forgery has many types [21].

- Copy_move: copying a portion of own image and pasting it in the same image [31]. See figure (3) [32].

A. Original image B. Forgery image

Figure (3): Copy_Move Forgery[32]

- Cut: cutting a portion of image [31]. See figure (4) [32].

A. Original image B. Forgery image

Figure (4): Cut Forgery[32]

- Image splicing: copying a portion of another image and pasting in image [9]. See figure (5).

A. Original image[32] B. Forged image

Figure (5): Image Splicing Forgery

- Image resampling: for creating a forged image of high quality, some of the chosen areas could be geometrically transformed in techniques such as rotation, scaling, stretching, skewing and flipping figure (6) [33].

A. Original image B. Forged image

Figure (6): Image Resampling Forgery[33]

- Image compression figure (7), and for more details see [32].

A. Original image B. Compression image

Figure (7): Image compression[32]

- Noise image: noise is any undesired information that contaminates an image from a variety of sources [34]. See figure (8), and for more details see [32].

A. Normal image B. Forgery image

Figure (8): Image Noisy Operation[32]

- Blurring image: See figure (9), and for more details see[32].

A. Original image B. Forged image

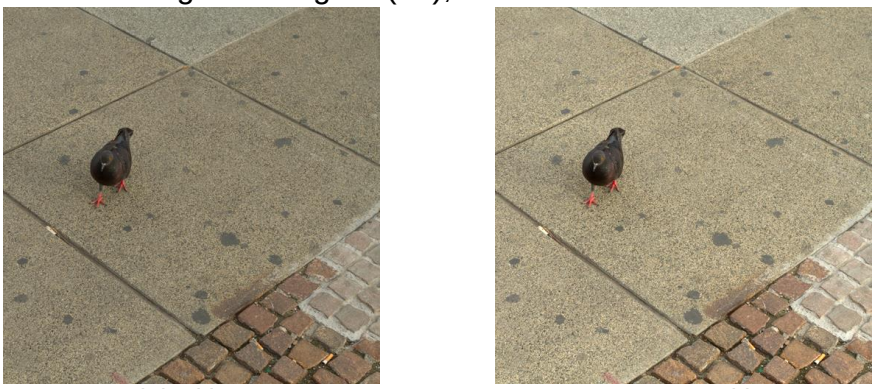
Figure (9): Blurring image Operation[32]

- Brightness image: See figure (10), and for more details see[32]

A. Original image B. Forged image

Figure (10): Brightness enhance Operation[32]

- Contrast image: See figure (11), and for more details see[32]



A. Original image B. Forged image

Figure (11): Contrast enhance image operation[32]

8. System Identification Forensic (SIF) Design

The concept of proposed SIF is by using image quality measured to extract a universal feature vector by using IQM for identification (forgery/anti-forensic) image from original one based on using FCM. The proposed SIF is made up of two stages: the first one (training) which consist of the following steps for data collection and classification. (1) collecting a forgery images to build dataset of some type of it by using previously made forgery images (CoMoFoD database). (2) Preprocessing by, convert input image to grayscale image. (3) Then, extracting features using IQM from this grayscale image. The Image Feature vector is

extracted for 249 images in dataset for some type of forgery image and image operation (copy move, splicing, brightness, contrast, Noise and blurring). The resulted feature vector is stored as classes centres to classify as a final step in this phase by FCM algorithm. The rest of this feature store to used it in the testing phase by EU equation. The second phase: is the identification phase, the input is (test image) goes on some process to extract IQM and used them to determine whether the input image is (anti-forensic image / not) by comparing with stored classes centres. Figure (12) is the framework block diagram of general proposed method, and the steps of propose SIF is follow: see Algorithm (1).

Algorithm (1): SIF Steps
Input: Colour image
Output: Class name
Process:
Step1: In Training Stage
step 1.1: Read colour image
Step 1.2: Convert input colour image to grayscale image
Step 1.3: Using IQM to extract feature vector value from the image result from previous step
Step 1.4: Classification all feature vector from previous step by FCM algorithm **(***training set***)**
Step 2: In Testing Stage // Repeat step 1.1, 1.2, 1.3 in training stage
Step 2.4: Identification any image by difference between feature result from step 1.3 in testing stage and class's centers result from step 1.4 in training stage by EU equation **(***testing* set)
// In implementation like testing stage in all steps (Step 2)
Step 3: End

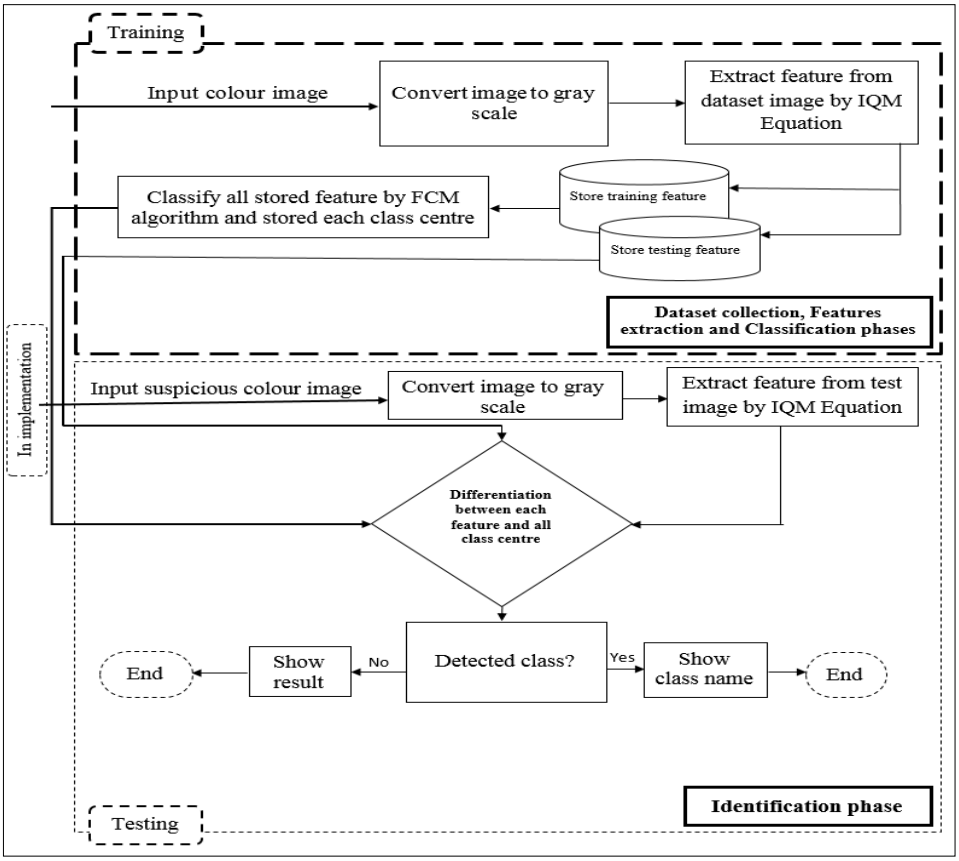


Figure (12): Framework of the general proposed approach

To build dataset of feature vector, selected 249 images (forgery image in most types, see following tables) after complete all dataset images to find the center classes values for each feature as explained in Algorithm (3.3).

Table (1) shows all the features that input to FCM Algorithm approved in the system. The following table (1) shows the feature vector that extracted from dataset image (CoMoFoD database) (it's the input of FCM algorithm). Table (2) shows the centers of classes (it's the Output of FCM algorithm). Table (3) gives the SIF classes dependency in which the EU is used to present the min distance for similar classes.

Note (feature1=AVG, feature2=SD, feature3=ENT, feature4=IDM, feature5=Ctrt, feature6=Egy, feature7=Corr, feature8= Shade, feature9= Hom, feature10= MP).

Table (1): Some Feature Vectors of training Image Dataset, just 64 from 249 image

	Feature_1	Feature_2	Feature_3	Feature_4	Feature_5	Feature_6	Feature_7	Feature_8	Feature_9	Feature_10
1	0.000492645	8.74813	0.234064	0.955354	-76435.3	268.822	11773.6	0.00672318	144.455	0.222688
2	7.31617E-05	10.1241	0.0638684	0.776554	-649762	1574.27	12516.5	0.000959363	168.425	0.0649703
3	0.000130844	9.37745	0.122973	0.932455	-95392.6	392.957	11242.5	0.000554214	116.372	0.12436
4	0.0112349	7.5615	0.388314	0.986553	2432830	119.843	17704.1	0.102063	109.234	0.27064
5	0.000267144	8.80672	0.13461	0.921704	248337	217.514	5338.67	0.00122309	108.79	0.13899
6	0.00106708	8.62976	0.288065	0.960617	-852723	330.675	16462.1	0.0102702	147.5	0.259619
7	0.000497587	8.26452	0.298454	0.987417	-1298900	88.9208	14044.9	0.00204103	149.6	0.296679
8	0.000140228	9.38583	0.116939	0.925737	548140	369.072	9570.5	0.00318768	106.212	0.117483
9	0.00109956	8.24762	0.35875	0.960697	-7444.23	217.098	10830.4	0.0226731	125.693	0.308011
10	0.000398386	8.57435	0.214362	0.95845	-749693	179.329	8452.62	0.00139891	161.34	0.214324
11	0.0304844	7.60817	0.422767	0.987058	-369070	150.632	23127.3	0.173794	154.197	0.237189
12	0.00190382	7.22446	0.252755	0.949073	-565844	119.557	4575.63	0.00591289	150.054	0.248504
13	0.0163632	7.49001	0.311452	0.986839	-749121	81.0206	12231.2	0.125126	170.616	0.197341
14	0.00170949	7.35888	0.375468	0.996408	1808240	48.27	26828.9	0.00855018	124.436	0.355881
15	0.0917895	5.85131	0.572214	0.993899	-764181	70.1638	22930	0.281422	175.942	0.172245
16	0.000219136	8.8622	0.121835	0.885957	116692	245.594	4061.43	0.000630657	110.457	0.123831
17	0.000379928	8.32814	0.093849	0.76801	-92012.7	212.095	1616.39	0.000911586	125.353	0.0979576
18	0.000134946	9.32627	0.083986	0.906908	-518934	384.49	7875.91	0.00119634	158.345	0.0858357
19	0.000645791	9.57619	0.0994259	0.903949	969065	577.93	11455.9	0.0236936	111.266	0.0786639
20	0.000163459	9.64481	0.0753722	0.859176	497940	633.281	8360.63	0.00890946	108.958	0.0694634
21	0.000518619	8.11367	0.182219	0.894179	-31619.4	118.659	2123.98	0.00282458	88.4655	0.184431
22	0.00034163	8.69479	0.178777	0.966475	-48508	153.51	9004.33	0.00278253	124.859	0.175694
23	0.00073007	8.34718	0.279303	0.922753	-444688	273.244	6801.36	0.00344759	114.314	0.270418
24	0.000201763	9.47712	0.1879	0.83533	42329.8	873.693	9737.76	0.00128042	128.308	0.185305
25	0.000302428	8.63838	0.220741	0.968981	130831	141.354	8972.55	0.00155562	139.651	0.221115
26	0.000647474	8.22688	0.245649	0.957725	-213499	138.229	6401.35	0.00261436	144.641	0.244778
27	0.000192303	9.66507	0.0842248	0.884892	766301	647.405	10601.3	0.0100179	128.595	0.0781867
28	5.92512E-05	10.249	0.0578127	0.811373	-439077	1640.68	15755.3	0.00104345	152.566	0.0590125
29	0.00126664	8.13855	0.337732	0.985459	-1265820	156.157	21321.4	0.00845845	160.885	0.310055
30	0.000635361	8.61867	0.255929	0.969901	-260088	230.97	15116.5	0.00770166	153.735	0.249036
31	0.00108656	8.27397	0.354968	0.959786	-555.338	223.062	10870.6	0.0226731	125.496	0.304475
32	0.000406653	8.53994	0.216088	0.962686	-763809	160.266	8429.89	0.00137216	161.804	0.216188
33	0.029034	7.64033	0.419323	0.986731	-348225	153.625	23002.5	0.169559	153.83	0.238063
34	0.00189828	7.23631	0.251772	0.952125	-608563	120.003	4893.21	0.00591289	149.327	0.247345
35	0.0163592	7.49365	0.311194	0.986784	-745327	81.3875	12234.8	0.125126	170.549	0.197072
36	0.00165626	7.37751	0.374979	0.996359	1806670	48.9256	26827.1	0.00821	124.445	0.35473
37	0.0899248	5.90249	0.566619	0.993705	-715584	72.0765	22828	0.279847	175.427	0.172164
38	0.000219562	8.84779	0.121243	0.883296	87997.3	241.81	3902.19	0.000645945	110.687	0.123287
39	0.000353092	8.39745	0.0944061	0.798438	-128734	217.054	1936.67	0.000896297	123.886	0.098224
40	0.000135101	9.32677	0.0837182	0.906638	-522589	386.503	7893.15	0.00119251	158.452	0.0855391
41	0.00079708	9.55963	0.102748	0.906509	1024310	580.274	11833.2	0.0266825	112.04	0.0787447
42	0.000211919	9.6388	0.0778229	0.864289	571681	636.423	8742.67	0.0113442	109.678	0.0695073
43	0.000524532	8.09467	0.182201	0.890586	-19518.7	116.609	2014.92	0.0018117	88.7818	0.185299
44	0.000342038	8.69173	0.178727	0.966594	-49833.1	152.454	8974.79	0.00278253	124.922	0.175773
45	0.000734714	8.33823	0.279398	0.922979	-436556	269.544	6729.65	0.0034667	114.523	0.27063
46	0.000160562	9.62367	0.169278	0.810356	-127508	1040.94	9936.87	0.0011734	131.764	0.16647
47	0.000300895	8.64256	0.220772	0.968667	118309	142.075	8926.66	0.00157091	140.069	0.221095
48	0.000639559	8.24488	0.24442	0.956579	-213194	144.396	6506.58	0.00261818	143.754	0.243636
49	0.000189234	9.69721	0.0847103	0.8873	747488	671.868	11251.3	0.0100561	131.443	0.0788712
50	5.84929E-05	10.2559	0.0570614	0.807512	-409279	1666.63	15650.1	0.00105874	152.312	0.058218
51	0.00148239	8.07192	0.341494	0.985923	-1325620	152.218	21474.9	0.0100561	162.077	0.313306
52	0.00054481	8.67093	0.250317	0.968014	-220750	242.946	14947.6	0.00637919	152.844	0.244168
53	0.00343555	7.36719	0.477717	0.987075	-1363890	80.572	12386.6	0.0203836	166.207	0.369133
54	0.00124497	8.03878	0.287253	0.966556	-1294560	186.62	10973.5	0.00611928	150.768	0.285313
55	0.000136374	9.22982	0.0743168	0.785851	-31453.3	475.007	3961.22	0.000351639	132.817	0.0765842
56	0.000152228	9.24536	0.155963	0.953762	121414	298.8	12625.7	0.000680345	107.641	0.158929
57	0.000233554	8.8797	0.115538	0.846042	-271303	357.302	4284.25	0.000657412	143.774	0.119175
58	0.000375272	8.59201	0.212351	0.958832	122676	176.955	8419.67	0.00366163	134.053	0.209093
59	0.000365237	8.60997	0.158944	0.905382	-107726	196.915	3965.37	0.00121354	125.556	0.164327
60	8.08697E-05	9.78575	0.0761198	0.84831	401850	793.036	9663	0.000324884	95.7614	0.0784998
61	0.00241343	8.51252	0.32352	0.957	-852474	320.789	14599.5	0.0452468	153.341	0.266938
62	0.00675962	8.28441	0.247707	0.95923	-1571690	257.661	12382	0.0795201	198.853	0.168449
63	0.0526157	7.2874	0.445737	0.983583	-1734460	189.503	22896.9	0.228691	173.468	0.197429
64	0.00132148	7.64559	0.21257	0.949025	-1147590	191.674	7328.65	0.00550774	186.849	0.201413

Table (2) Extracetd feature vector of testing

feature_1	feature_2	feature_3	feature_4	feature_5	feature_6	feature_7	feature_8	feature_9	feature_10
0.0001402	9.38583	0.116939	0.925737	548140	369.072	9570.5	0.0031877	106.212	0.117483
0.0006458	9.57619	0.0994259	0.903949	969065	577.93	11455.9	0.0236936	111.266	0.0786639
0.0001635	9.64481	0.0753722	0.859176	497940	633.281	8360.63	0.0089095	108.958	0.0694634
0.0001923	9.66507	0.0842248	0.884892	766301	647.405	10601.3	0.0100179	128.595	0.0781867
8.123E-05	9.78105	0.0748588	0.843432	365624	794.708	9356.89	0.0002848	94.8953	0.0774088
9.346E-05	10.0381	0.0753858	0.785912	-748338	1500.22	12514.8	0.0008562	170.971	0.0765017
0.0009601	8.68821	0.280017	0.958508	-826235	348.587	16453.9	0.00993	146.624	0.253562
0.0004067	8.53994	0.216088	0.962686	-763809	160.266	8429.89	0.0013722	161.804	0.216188
0.0018983	7.23631	0.251772	0.952125	-608563	120.003	4893.21	0.0059129	149.327	0.247345
0.0163592	7.49365	0.311194	0.986784	-745327	81.3875	12234.8	0.125126	170.549	0.197072
0.0899248	5.90249	0.566619	0.993705	-715584	72.0765	22828	0.279847	175.427	0.172164
0.0001351	9.32677	0.0837182	0.906638	-522589	386.503	7893.15	0.0011925	158.452	0.0855391
0.0015924	9.0628	0.135201	0.905769	-1104410	569.714	11522.2	0.025425	195.158	0.10719
0.0104438	9.16495	0.172944	0.787287	-1376580	2204.08	18519.4	0.0986577	180.036	0.0834874
0.0002234	8.85784	0.103555	0.906848	-255674	240.121	4915.36	0.0029201	127.546	0.103692
0.0011922	7.887	0.328426	0.922695	-219024	170.656	4244.45	0.0074456	92.7741	0.296096
9.439E-05	9.79695	0.0718875	0.811358	-216547	1023.89	9831.5	0.0014257	122.979	0.0732125
0.007899	6.38296	0.438467	0.995037	1716270	63.5716	25553.8	0.0416692	124.092	0.167653
0.0136954	6.57588	0.430169	0.98497	2255820	128.358	16951.8	0.0952139	109.147	0.126215
0.0056957	8.45873	0.35917	0.965468	-840172	288.209	16404.1	0.0581198	148.02	0.119413
0.0041551	7.4327	0.512213	0.987803	-1301420	86.1983	14048.5	0.0261321	150.073	0.159801
0.0001489	9.32164	0.123904	0.939322	534430	296.677	9482.04	0.0018232	106.747	0.112616
0.0023005	7.90068	0.497915	0.964213	-22020.9	195.705	10741.6	0.0232158	126.176	0.1053
0.0056876	7.497	0.475574	0.972532	-745531	117.155	8413.04	0.0362073	161.856	0.132855
0.0306955	7.1875	0.547665	0.988569	-385973	131.247	22833	0.169249	154.451	0.0821189
0.112543	4.38434	0.748401	0.9643	-557250	81.7196	4496.44	0.221299	150.991	0.0458287
0.0012093	7.39213	0.383607	0.993002	185083	16.7829	4779.79	0.0054504	108.826	0.350255
0.0041636	6.22106	0.403412	0.985256	-75957.8	8.0532	1084.38	0.0111836	125.333	0.393798
0.0016183	6.82641	0.367406	0.985747	-17217.1	10.56	1471.23	0.0045522	88.4676	0.350411
0.0011249	7.40781	0.413691	0.996866	-32289	12.2654	7813.99	0.0056071	124.859	0.358979
0.0005784	8.26862	0.345901	0.986415	-7025.75	55.4484	8107.47	0.0047318	128.311	0.25499
0.0007467	7.7385	0.402089	0.994515	103325	21.7509	7909.19	0.0041432	139.654	0.340474
0.0003607	8.33185	0.288025	0.9925	106121	40.5649	10776.7	0.0012919	106.29	0.280548
0.0001402	9.08236	0.11006	0.961124	45476.6	160.736	8108.51	0.0003937	152.569	0.11337
0.0011975	7.90681	0.405047	0.992143	-26179.3	40.7736	10337.9	0.0117188	144.457	0.321699

Table (3) The seven classes center

feature_1	feature_2	feature_3	feature_4	feature_5	feature_6	feature_7	feature_8	feature_9	feature_10
0.000616012	9.192054465	0.128617782	0.904077832	644800.502	541.3419861	10549.15228	0.013578761	116.4448636	0.108399711
0.021170907	7.747935147	0.334740698	0.949890359	-713621.6148	295.9959252	12434.01348	0.079164548	161.5576406	0.206628092
0.018724355	7.544243121	0.393079814	0.966907678	-1309300.622	258.8895055	14842.67946	0.072097708	169.6519621	0.219036496
0.009097518	8.198348452	0.270429948	0.928009261	-309965.4162	330.921687	9796.843724	0.044617453	142.4458809	0.185440989
0.019279873	6.853082148	0.488735272	0.994225575	1994756.679	60.96575738	23663.0385	0.086023504	120.955835	0.26061454
0.054535186	7.059875425	0.436885696	0.979052778	-2374750.028	189.6555845	18691.76541	0.171783233	192.8682145	0.21685829

9. Testing and Discussions

In the presented SIF, the result obtained from the testing stage is 94.8%. This Rate was obtained, where the number of images entered in the test stage was 39 images (15% of the total number of images dataset).

As in table (4), only 37 images (True Positive (TP)) were correctly identifying by the proposed SIF According to precision equation (17) [30]:

Precision= (True Positive / Total no of images) *100…….. (17)

Precision = (37/39) *100=94.8

Table (4): Results ratios for the proposed work

True = T Posetave = P	True = T Nagative = N
TP 37	TN 2

Which indicate that the proposed SIF is an applicable promising means for identification both image anti-forensics and image operation.

10. Conclusions and Recommendation

In the proposed SIF, the result obtained from the testing stage is 94.8%. This Rate was obtained, where the number of images entered in the test stage was 39 images (15% of the total number of images dataset). Only 37 images were correctly identifying by the proposed SIF.

In comparison with the previous works, the propose SIF is the only work that uses IQM, FCM and EU. To identify image forgery and image operation.

The proposed SIF can identify forgery image type and image operation types, while there is previous work can identify the universal forgery. Which conceder an improvement to the previous work.

The proposed SIF can be developed and improved by using a better classifier rather than EU, for example SVM algorithm, or any templet matching algorithm.

References

[1] K.Anusudha, Et Al, "Image Splicing Detection Involving Moment-Based Feature Extraction And Classification Using Artificial Neural Networks", Aceee Int. J. On Signal & Image Processing, Vol. 01, No. 03, Dec 2010

[2] S. K. Mankar and P. A. a Gurjar, "Image Forgery Types And Their Detection: A Review", Volume 5, Issue 4, April 2015 International Journal of Advanced Research In Computer Science And Software Engineering.

[3] A. Gupta, et al, "Detecting Copy Move Forgery Using DCT", International Journal Of Scientific And Research Publications, Volume 3, Issue 5, May 2013.

[4] I. Amerini, et al, "Copy-Move Forgery Detection And Localization By Means Of Robust Clustering With J-Linkage", Signal Processing: Image Communication 28 (2013) 659–669

[5] P. D. Pandit and M. Rajput, "Survey On Anti-Forensics Operations In Image Forensics", (IJCSIT) International Journal Of Computer Science And Information Technologies, Vol. 5 (2), 2014, 1570-1573.

[6] V. Christlein, et al, "An Evaluation Of Popular Copy-Move Forgery Detection Approaches", IEEE Transactions On Information Forensics And Security, 2012.

[7] H. Li, et al, "Identification Of Image Operations Based On teganalytic Features", arXiv:1503.04718v2 [cs.MM] 17 Mar 2015

[8] S. Ye, et al,"Detecting Digital Image Forgeries by Measuring Inconsistencies of Blocking Artifact",1-4244-1017-7/07 ©2007 IEEE, ICME 2007 .

[9] W. Chen, et al, "Image Splicing Detection Using 2D Phase Congruency And Statistical Moments Of Characteristic Function_Art". No. 65050r", Article in Proceedings of SPIE - The International Society for Optical Engineering, February 2007, DOI: 10.1117/12.704321.

[10] S. Ghanbari, et al, "New Steganalysis Method using GLCM and Neural Network", International Journal of Computer Applications (0975 – 8887), Volume 42– No.7, March 2012.

[11] Amani Alahmadi, et al, "Passive Detection of Image Forgery using DCT and Local Binary Pattern ", Article : April 2016, DOI: 10.1007/s11760-016-0899-0, 20

[12] H. Turabieh, et al, "Steganalysis of LSB Encoding in Digital Images Using GLCM and Neural Networks", 2014.

[13] G. Suresh and C. Srinivasa Rao, “Copy Move Forgery Detection Using GLCM Based Statistical Features,” Int. J. Cybern. Informatics, vol. 5, no. 4, pp. 165–171, 2016.

[14] C. Shanthi and T. Nadu, “Image Forgery Detection Based On Local Texture Descriptors”, vol. 4, no. June, pp. 268–273, 2017.

[15] G. K, et al, "Digital Image Forgery Detection Using Passive Techniques: A Survey", Digital Investigation 10 (2013) 226–245.

[16] S. Mushtaq and A. Hussain Mir, "Digital Image Forgeries And Passive Image Authentication Techniques: A Survey", International Journal of Advanced Science and Technology, Vol.73 (2014), pp.15-32.

[17] A. C. Brooks, et al, "Structural Similarity Quality Metrics In a Coding Context: Exploring the Space of Realistic Distortions", IEEE Transactions On Image Processing, Vol. 17, No. 8, Aug. 2008.

[18] M. Desai and S. Patel, "Survey on Universal Image Steganalysis.," nternational J. Comput. Sci. Inf. Technol., vol. 5, no. 3, pp. 4752–4759, 2014.

[19] A. G. Khandizod, et al, "Comparative Analysis of Image Enhancement Technique for Hyperspectral Palmprint Images", International Journal of Computer Applications (0975 – 8887), Volume 121 – No.23, July 2015

[20] M. Fedias And D. Saigaa, "A New Approach Based In Mean And Standard Deviation For Authentication System Of Face", 2010.

[21] N. Pandey, "Managing Information Entropy With Probability Theory", 2017.

[22] F. E. M. Al-Obaidi And A. Jassim, "How Does The Statistical Features Distribute Within Human Iris Texture?",ISSN: 1819-544X Published BY AENSI Publication, EISSN: 1816-157X , 2016 September; 12(9): pages 23-29 Open Access Journal.

[23] F. Albregtsen, "Statistical Texture Measures Computed From Gray Level Co-Occurrence Matrices",University of Oslo, November 5, 2008.[24] D. Gadkari, "Image Quality Analysis Using Glcm", pp. 1–120, 2004.

[24] T. Velmurugan and T. Santhanam "Implementation Of Fuzzy C-Means Clustering Algorithm For Arbitrary Data Points", International Conference on Systemics, Cybernetics and Informatics, 2011.

[25] S.B. Kotsiantis, "Supervised machine learning: a review of classification techniques",Informatica 31 (2007) 249-268

[26] M. Weber, et al, "Unsupervised learning of models for recognition", Proc. 6th Europ. Conf. Comput. Vision 2 (2000) 101-108.

[27] H. R.M Shaaban, et al, "Performance Evaluation Of K-Mean And Fuzzy C-Mean Image Segmentation Based Clustering Classifier", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 6, No. 12, 2015.

[28] M. Alata, et al, "Using GA for Optimization of the Fuzzy C-Means Clustering Algorithm," Res. J. Appl. Sci. Eng. Technol., vol. 5, no. 3, pp. 695–701, 2013.

[29] A Vadivel et al, "Performance Comparison Of Distance Metrics In Content-Based Image Retrieval Applications", 2003.

[30] N.Parashar1 And N.Tiwari, "A Survey Of Digital Image Tampering Techniques",International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol.8, No.10 (2015), pp.91-96.

[31] http://www.vcl.fer.hr/comofod/download.html, (dataset)

[32] M. Ali Qureshi, et al, "A Review On Copy Move Image Forgery Detection Techniques", 2014, DOI: 10.1109/SSD.2014.6808907.

كشف الصور المزورة والمعالجة للصور عن طريق خوارزميه العنقدة المضببه

أ. م. د. هناء محسن احمد* الباحث: هدى محمد عيد*

المستخلص: الصور الرقمية سهل التعديل والتلاعب بها نظرا لتوافر برامج قوية لمعالجة و التحرير الصور. في الوقت الحاضر، من الممكن أضافة أو إزالة الميزات الهامة من صورة دون ترك أي اثر واضحة من العبث (خلق صور مزوره). هيكليه الكشف عن الصور المزوره بشكل عام تشمل بعض المراحل الأساسية، وأهم مرحلة هو استخراج الخصائص الاحصائيه من الصورة لأن هذه الخصائص تكون أساسية للكشف عن إذا الصورة الأصلية أم لا. في هذه الورقة ،تم بناء هيكل لمعرفه الصورة المزوره وعمليات الصور باستخدام نهج تحليل الاخفاء السري والخصائص الاحصائيه. لتحديد صورة المزوره وعمليات الصور. حيث العديد من نهج تحليل الاخفاء السري التي تعتمد على الخصائص الاحصائيه، واحدة من هذه هي مقياس جودة الصورة (IQM). ويمكن تحقيق هذا الهدف عن طريق استخدام خوارزميه العنقدة المضببة (FCM) والمسافه الاقليديه (EU). وكانت النتائج التي تم الحصول عليها من اختبار هذا النظام لتحديد التزوير وعمليات الصورة بدقة 94.8٪.

الكلمات المفتاحية: صوره المزورة، عمليات الصورة، مقياس جودة الصورة (IQM)، خوارزميه العنقدة المضببه (FCM)، كشف المضادة للطب الشرعي.

*الجامعة التكنولوجية