Digital Video Scenes Recognition using M_{ijn}-EA and Learning Vector Quantization Network

Samaher Al_Janabi
Department of Computer Science, Faculty of Science for Women (SCIW)
University of Babylon, Babylon
samaher@uobabylon.edu.iq

Mahdi A. Salman
Department of Computer Science, Faculty of Science for Women (SCIW)
University of Babylon, Babylon
mahdi.salman@uobabylon.edu.iq

Abstract

This paper presents a hybrid method for digital video scenes recognition which uses recognition to group pixels into known object for each frame. The chromaticity is used as data source for the method because it is normalized. The recognition is carried out by means of an Evolutionary Algorithm from type Mijn-EA, which is employed to obtain the best clusters that represent each frame and number of seeds in each frame. Then, conversion the color space for each cluster from RGB to HSV to fined the textural features of co-occurrence matrix. After that the pixels for each cluster recognition according to the identified classes. The number of classes is a priori unknown and the Evolutionary Algorithm that implements the Mijn-EA is used to determine the main clusters. The detection of the classes in the LVQ3 is done using a texture features. The obtained results substantiate the feasibility of the method and the accuracy.

Key words: Audio/Video Interleaved File, Learning Vector Quantization Network, M_{ijn}-Evalutionary Algorithm, Textures Features.

الخلاصة

يقدم هذا البحث طريقة هجينة لتميز مشاهد الفديو الرقمية التي تستخدم لتميز مجموعة النقاط الى كيانات معروفة لكل اطار .تم استخدام اللون كمصدر للبيانات لتلك الطريقة وذلك لطبيعته. ينفذ التميز باستخدام خوارزمية تطورية من نوع **MIJN-EA** التي تستخدم لايجاد افضل العناقيد التي تمثل كل اطار (حيث ان كل مشهد يحتوي على 24 اطار) وعدد البذور في الاطار ثم نحول الفضاء اللوني لكل عنقود من (احمر, أخضر , ازرق) الى (اللون النقي, الاشباع,الشدة) لايجاد خصائص السطح من مصفوفة الحدوث المتزامن. بعد ذلك كل مجموعة نقاط تميز طبقاً الى الاصناف المحددة.عدد الاصناف غير معروف مسبقاً والخوارزمية التطورية تستخدم لتحديد العناقيد الاساسية. وتحديد الاصناف في شبكة تعليم المتجه الكمي الثالثة تتم باستخدام خصائص السطح . النتائج التي تم الحصول عليها توضح سهولة الطريقة ودقتها.

1. Introduction

Digital video has become very important form of information technology and is now used in many different areas, such as teleconferencing, mobile telephone, surveillance, and entertainment [Holtz, 2004] .

A video sequence is essentially an ordered set of frames and displayed in quick succession to a human viewer, they give the visual illusion of motion. This illusion is achieved by keeping the semantic content mostly the same in each frame of group of successive frames-so that the viewer is able to identify the same critical objects or critical elements of the video in each frame and by introducing only small changes to this content between frames by moving objects relative to one another or by panning, or zooming or moving the camera, etc.

Therefore the fact that the video is intended for human viewer is important because it allow us to assume things about the nature of the video that greatly aid in digital video scene recognition. To give this illusion of motion, successive frames in the video must have very similar semantic content, and this effectively means that successive frames often have very similar pixel have moved slightly between successive frames.

As a result there is a lot of information that is shared between frames in sequence and usually these frames have different types of objects (scenes). Therefore the real challenge in this situation is to be able to automatically recognition these scenes from each frame.

2. M_{ijn} –Evolutionary Algorithm

Since early 60's both in the USA and in Europe researches started considering the relative importance of crossover and mutation operators in the execution of an Evolutionary Algorithm (EA). One of the first attempts to use mutation alone for evolution of good solutions to a given problem was made by Friedberg [Friedberg, 1958] who tried to evolve finite state machines. With time two different opinions because widespread in the scientific community and divided it into schools.

On the first hand, people like Fogel et al. in the USA and Rechenberg supposed mutation to be the key operator for evolution, or at least of high important so they used just selection and mutation to evolve a population(if recombination present, it is a secondary operator). This opinion has historically led to the idea of the Evoluationary Programming (EP) [Fogel, et, al, 1966] and of the Evolution Strategies (ES) [Schwefel, 1977] . On the other hand, the other school believes that crossover does play the vital role and mutation seen as being only a background operator. This can be found in Genetic Algorithms (GAs).

The recent research by me and my colleagues fits well in the frame; in fact, we decided to focus our attention on the mutation behavior, and to attempt to form an evolution model based on selection and mutation only. Furthermore, keeping mind the work carried out by many other researchers, we have aimed to design new mutation operator which could exhibit.

The idea underlying our work is the design of a mutation operator allowing us to take of the information shared between adjacent bits and among neighbouring ones, in many cases they are not completely independent one from another, rather they whole encode in the most general case real variables. What is really important for is that such an operator should be capable of mutating a number of adjacent bit the same time. With this we do not mean we are looking for something which since chooses randomly a substring and applies flip-mutation to all of the therein contained bits.

2.1 M_{ijn} Mutation

Let us consider a genetic alphabet *A* composed by s≥2 different symbols, *A*= $\{a_1,\ldots,a_s\}$ and strings $\vec{\sigma}$ over the universe $\sum = A^{\iota}$ (the set of all the strings $\vec{\sigma}$ of a given finite length L).

In the following we shall simply denote with $\vec{\sigma}$ a generic string $\sigma_{L-1}\ldots\sigma_0$ of any length L, where $\sigma_{q\in A\ \forall_q \in \{0,\ldots,L-1\}}$. If we want to specify the i-th symbol of the alphabet occurring in the generic position q of string $\vec{\sigma}$ we shall use $\sigma_{q,i}$.

Let's start by describing in words the M_{ijn} behavior. Our operator can be represented as a two phase-algorithm, the first phase consisting in search and the second in replacement. **In the search phase**, we randomly choose a point p in the string $\vec{\sigma}$, and we consider σ_p, let it be equal to a_i ($\sigma_{p=}a_{i)}$. Then, starting from p, we go leftbound and we examine the contents of the adjacent position, $\sigma_{p+1;}$ if it is equal to a_i we move leftbound

until we find a position k whose contents is different from the previously found values, let's say it is a value a_j ($\sigma_k=a_j$). Now the search phase has ended and we start **the replacement phase**: σ_k becomes a_i and $\sigma_{k-1,\dots}$, σ_p all become a_j.

Then, formally, the M_{ijn} operator is the following:

$$M_{ijn}:\sigma\in\sum_{,}p\in\{0,...,L-1\}\rightarrow\sigma^{t}\in\sum^{t}\subset\Sigma \qquad (1)$$

Where p is randomly chosen and p=0 means the rightmost position in $\vec{\sigma}$, while p=L-1 means the leftmost one. Denoting with $\sigma_{q,i}^{z}$ the generic sequence of z equal symbols a_i starting from the position q and going left, we have that the application of M_{ijn} to p when n-1 equal symbols are met according to the above procedure yields for n<L-p-1:

$$\sigma=\sigma_{L-1}\dots\sigma_{p}\boxed{\sigma_{p+n-1},i\sigma_{p,j}^{n-1}}\qquad\sigma_{p-1}\dots\sigma_{0}\xrightarrow{M_{ijn}}$$

$$\sigma^{t}=\sigma_{L-1}\dots\sigma_{}\boxed{\sigma_{p+n-1},j\sigma_{p,i}^{n-1}}\qquad\sigma_{p-1}\dots\sigma_{0}\qquad(2)$$

While for n=L-p-1:

$$\sigma=\boxed{\sigma_{p,j}^{L-P}}\quad\sigma_{p-1}\dots\sigma_{0}\xrightarrow{M_{ijn}}\boxed{\sigma_{p,k}^{L-P}}\qquad\sigma_{p-1}\dots\sigma_{0}\qquad(3)$$

With $a_k\neq a_j$ randomly chosen in *A*.

Example:-

Let us consider the binary case, and the following 8-bit string σ=01011010. The length of 8 means that p can vary between 0 and 7. Let us suppose that the application point p is randomly chosen as 3, i.e. p=3. in this case $\sigma_3=1$, so we have to go leftbound and look for the first occurrence of a 0 in σ. For p=4 we find $\sigma_4=1$, so we further move leftbound; for p=5 we have that $\sigma_5=0$ so we stop. Then we replace the string components from 3 to5:precisely,in the 3rd and in the 4th position, where a 1 was contained, we insert a 0, and in the 5th position, where a 0 was held, we put a 1. In conclusion, in such case the new string obtained after the application of M_{ijn} are 01**100**010, where the substring in bold is the part of σ modified according to $M_{ijn.}$

Procedure M_{ijn} Mutation Operator

Begin
  Choose randomly a value p(0≤p≤L-1);let us
  Suppose that $\sigma_{p=}a_i$.
  Index=p;
  /*start search phase*/
  Repeat
    Index=index+1;
  Until $\sigma_{index}<>a_i$;
  /*end search phase and start replacement phase*/
  Save= σ_{index} ;
  $\sigma_{index=}\sigma_p$;
  For count =(index-1) To p step 1 do
  σ_{count}=save ;
  end for
  /* end replacement phase*/
End.

Note:- as it can immediately be seen the M_{ijn} mutation changes at last two symbols at the same time apart from the case p=L-1 in which only the leftmost bit is changed. So it definitely something different from the classical Bit-Flip Mutation (BFM) used in GAs.

3. Kohonen Networks

Kohonen neural network belongs to the (Winner Takes All) WTA group of networks with the competitive learning algorithm (supervised learning). The main advantage of this network is its simple structure and simultaneously large abilities. It is similar to the one-layer network but the possibility of application is widened. The network consists of one layer but this layer may be expanded to the two- or more dimensions.

The outputs of neurons are WTA type. Idea of regularity detection the idea of contiguity was defined for the Kohonen network. In the one-dimensional network the contiguity is described as the sections on both sides of the selected neuron.

The Kohonen network does not require the distinct learning process. The weights in the matrix of weights are evaluated in the adaptive way during the network run [Zurada, 1992] .

The learning vector quantization (LVQ) is a supervised learning extension the kohonen network method which used as pattern recognizer each neuron in output layer represent a class or category (several output may be assigned to each class), the weight vectors are some times referred to as a reference or code book vector. The neuron with closest (Euclidean norm) weight vectors declared to be the winner. Kohonen [1990] is proposed several improvements to LVQ is termed LVQ2, LVQ2.1, LVQ3.All of these devolve around the manner in which the network is trained in particular, training will no longer be exclusively to the winning neuron. The training method rewards a winning neuron if it belongs to the correct category by moving it towards the input vector. Conversely, if the wining neuron does belong to the correct category, it is punished in that it is forced to move a way from the input. The principal idea behind these improvements is that the two neurons are modified rather than only a winning neuron in LVQ. In the other words the winner and runner-up are modified in the same time [Sordo, 2002] .

3.1. Winner take –all learning rule

This learning rule differs from the other rules .This rule is an example of competitive learning; typically it is used for learning statistical properties of inputs.

$$\Delta w_{mj} = \alpha(x_j - w_{mj}) \qquad \text{j=1, 2} \qquad (4)$$

Where α >0 is a small learning constant, typically decreasing as learning progresses

$$w_{pi}^t x = \max_{i=1,2...p}(w_i^t x) \qquad (5)$$

The learning is based on the premise that one of the neurons in the layer has the maximum response due to input x-this neuron is declared the winner- the individual weight adjustment becomes criterion corresponds to finding the weight vector closest to the input x.

3.2.1. LVQ neural network [Karkanis et.al., 2006]

Initialize the weight vectors; wij to random values .It may be beneficial to choose these values when the magnitude is small.

***Step 1 Initialize*:** Initialize the weight vectors .The weight vectors may be initialized randomly. However, there are some other choices which are going to be discussed shortly. Also initialize the learning rate.

Step 2: for each vector $x^{(p)}$ in the training set there are more steps 2a and 2b.

Step 2a: find the winning neuron k such as:

$i(x^{(p)}) = k$ Where $\left\|x^{(p)} - w_k\right\| < \left\|x^{(p)} - w_j\right\|$ j=1,2,…n (6)

Step 2b: update the weights w_k as follows:

$w_k^{new} = \{w_k^{old} + \alpha(x^{(p)} - w_k^{old})$ if T=C_j (7)

$w_k^{new} = w_k^{old} - \alpha(x^{(p)} - w_k^{old})$ if T≠C_j (8)

Where

$x^{(p)}$: All layers

T: The desired class (or category) for training vector

C_j : The actual class of the output neuron

$\|x- w_j\|$: The Euclidean norm between weight vector and input vector.

3.2.2. Variation of LVQ [Pandya, 1996]

A- LVQ2

Can be extended by training both the winning vector and the first runner –up under appropriate conditions. The principal idea is that when the winning neuron does not represent the correct category and the first runner-up does represent the correct category, we may wish to train both, the winner is punished and the runner – up is rewarded LVQ2 imposes the further condition that distance, dc from the input vector to the winner,dr the distance between the input vector and the runner-up the window is define as follows:

$dc/dr = 1-\varepsilon$ (9)

$dr/dc = 1+\varepsilon$ (10)

dc: The distance from the input vector to the winner. dr: The distance from the input vector to the runner-up.

ε :Constant

When the above conditions are not met, the training proceeds exactly as before for LVQ. When all of the above conditions are met, training proceeds as follows:

$w_R^{new} = w_R^{old} + \alpha(x^{(p)} - w_R^{old})$ (11)

$w_c^{new} = w_c^{old} - \alpha(x^{(p)} - w_c^{old})$ (12)

w_R: The weights of runner-up.

w_c : The weights of winner.

α : Learning rate.

B- The LVQ2.1 [Sordo, 2002]

The fine the modification of LVQ , the LVQ2.1, here the best two references vectors are trained ,provided that one of them belongs to the correct class and the an other does not best or runner-up , represent the correct class the windowing requirement is as follows:

$\min[dc_1/dc_2, dc_2/dc_1] > 1-\varepsilon$ (13)

$\max[dc_1/dc_2, dc_2/dc_1] < 1+\varepsilon$ (14)

Where the above conditions met, training proceeds as follows:

Where dc_1 is the distance from $x^{(p)}$ to wc_1 and dc_2 is the distance from $x^{(p)}$ to wc_2

$$w_{c1}^{new} = w_{c1}^{old} + \alpha(x^{(p)} - w_{c1}^{old}) \quad (15)$$

$$w_{c2}^{new} = w_{c2}^{old} - \alpha(x^{(p)} - w_{c2}^{old}) \quad (16)$$

dc_1 : The distance from $x^{(p)}$ to w_{c1}.

dc_2 : The distance from $x^{(p)}$ to w_{c2}.

w_{c1} : The weight of runner-up.

w_{c2} : The weight of winner.

α : Learning rate.

C- The LVQ3 [Sordo, 2002]

The LVQ3 algorithm a gain varies the windowing condition under which both the winner and the runner – up neurons are trained, where the both neurons belong to different classes, training proceeds as in LVQ2.1, when both neurons belong to the same class and the windowing criteria in LVQ3 is as follows:

$$\min[dc_1 / dc_2, dc_2 / dc_1] > 1 - \varepsilon / 1 + \varepsilon \quad (17)$$

Training is accomplished as follows:

$$w_c^{new} = w_c^{old} + \beta(x^{(p)} - w_c^{old}) \quad (18)$$

Where

$\beta = m1.\alpha(t)$

w_c : The weight of winner

β : Learning rate in LVQ improvements

$0.1 < m1 < 0.5$ is typical value of $m1$ is about 0.2.

4. Suggested Digital Video Scenes Recognition Approach

Digital Video Scenes Recognition is one of the important areas in computer vision. This part of paper we represent the structure of the suggested approach and show how we can implementing this approach to detection the path of any object in digital video frame.

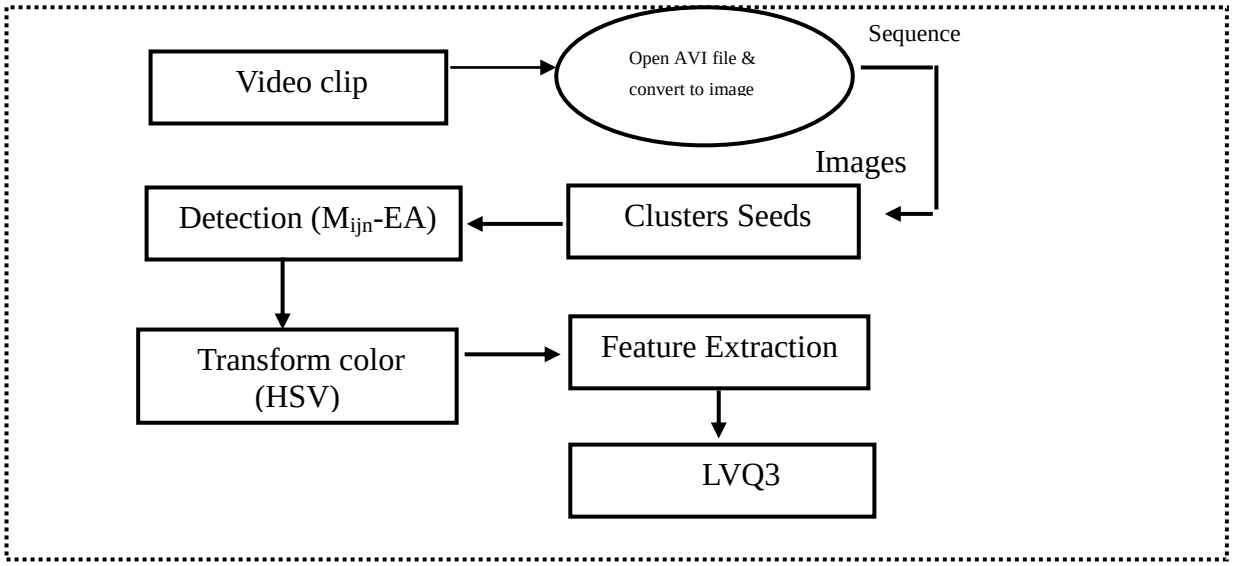

Figure (1): presents the structure of the proposed system

4.1. Open AVI file & convert to image

In this stage we used algorithm to open AVI structure and search to list movie (i.e. sequence frames/images) which are needed in the next stages to recognition it.

In this algorithm the function (ASCII to long)was used to convert FCC ASCCII character to long for example convert FCC RIFF to long by calling ASCII to long ("R","I","F","F") to open AVI file, then after that take information from AVI header such as total frame, height, and width of the image and then search for list movie using search function about key word "MOVI", when we search make a loop from one to sequence of image(frames) then search about keyword "00db" which is the initial of each frame, then make two loops from height and from width respectively to read RGB pixel image, and split it into three bands that save it into three individual matrices called (red, green, blue) which are used in the next stage[Wafaa, 2006] .

4.2. Clusters Seeds

Clustering [Kalyani and Sushmita, 2003] is a useful technique for the discovery of some knowledge from a data set. It maps a data item into one of several clusters, where clusters are natural groupings of data items based on similarity metrics or probability density models. Clustering of data is broadly based on two approaches: hierarchical and partitive. Hierarchical methods can again be categorized as agglomerative and divisive algorithms.

In most real life situations the number of clusters in a data set is not known a priori. The real challenge in this situation is to be able to automatically evolve a proper value of clusters as well as providing the appropriate clustering. So, before we train this network we need to determine the number of clusters and the optimal weight can be initialzed to the network.

In this papar we used the seeds to determined the desired output of the training stage in neural network.

4.3. Detection (M_{ijn}-EA)

Our Evolutionary Algorithm M_{ijn}-EA [De Falco, 1997] works as follows: we start with a randomly generated initial population of, say, P elements each with L positions. Each position contains a symbol in color matrices. We let this population undergo

selection according to the fitness and we apply M_{ijn} as the only operator affecting the string in the population. To each string M_{ijn} is applied once and only once. There is no a priori chosen selection scheme, rather we may choose one among the most commonly used (proportional, truncation, tournament, exponential, and so on).

In this work we deal with the truncation selection as explaining later. And Elitism scheme.

4.3.1. Selection

The selection scheme in M_{ijn}-EA is the truncation model in it, which starts from a population of p individuals. Only the T% elements showing the best fitness are chosen to give origin to the individuals of the next generation (Elitism). Usually truncation ratio lies in rang [10% to 50%].Example: Let p =100 and truncation ratio =50% then Elitism=100*50%=50 individuals.

4.3.2. Fitness Function [Nadia, 2004]

In this paper we use a Dunn's Validity Index as the fitness function and this technique is based on the idea of identifying the cluster sets that are compact and well separated. For any partition of clusters, where c_i represent the *i*-cluster of such partition, the Dunn's validation index, *D*, could be calculated with the following formula:

$$D = \min_{1\le i\le n}\left\{\min_{1\le j\le n}\left\{\frac{d(c_{i,}c_j)}{\max_{1\le k\le n}\{d\ (c_k)\}}\right\}\right\} \qquad (19)$$

Where

$d(c_i, c_j)$:-distance between clusters c_i, and c_j (intercluster distance);

$d(c_k)$:- intracluster distance of cluster c_k;

n:- number of clusters.

The minimum is calculating for number of clusters defined by the mentioned partition. The main goal of the measure is to maximise the intercluster distances and minimise the intracluster distances. Therefore, the number of cluster that maximise *D* is taken as the optimal number of the clusters.

4.3.3. Scheme for M_{ijn}-EA

The basic scheme for M_{ijn}-EA that use in this work can be explains in the following:

Procedure M_{ijn}-Evolutionary Algorithm

```
Begin
    t=0;
    Initialise randomly p (t) with p elements;
    Evaluate p (t) by using fitness function;
   While not terminated do
        For j=1 to p do
            Select randomly one element among the best T% in p (t)
             Mutate it;
             Evaluate the obtained offspring;
             Insert it into p/(t);
         End for
         P (t+1) = p/(t);
       t=t+1;
    End while
End.
```

After we fined the number of seeds that represented each cluster we can using it as a pointer to determine the desired output for each scene or object in frame.

4.4. Transform color (HSV)

While the RGB color model is good for supporting hardware, it is unnatural for humans; therefore transform color HSV perform some of equation to convert RGB pixels into HSV. Hue (H) describes pure color in terms of the dominant wavelength (e.g., red, orange, yellow, ect.), whereas the saturation(S) gives the measure of degree to which a pure color is diluted by white light (e.g., pink is diluted red), brightness describes overall intensity or strength of light (e.g., dark pink vs. light pink). And in this work we take the value component to extraction the texture features from the image [Beatriz, 2004] .

$$V = \max(R,G,B) \quad (20a)$$

$$S = \frac{\max(R,G,B) - \min(R,G,B)}{delta} \quad (20b)$$

$$H = \begin{Bmatrix} \theta & if\ S \neq 0\ \ and\ \theta \geq 0 \\ \theta + 360 & if\ S \neq 0\ \ and\ \theta < 0 \\ undefined & if\ S = 0 \end{Bmatrix} \quad (20c)$$

Where

$$delta = \max(R,G,B) - \min(R,G,B) \quad (20d)$$

$$\theta = 60 \times \begin{Bmatrix} \frac{(G-B)}{delta} & for\ R = \max(R,G,B) \\ 2 + \frac{(B-R)}{delta} & for\ G = \max(R,G,B) \\ 4 + \frac{(R-G)}{delta} & for\ B = \max(R,G,B) \end{Bmatrix} \quad (20$$

4.5. Texture Features

In this work we use the coefficients of the co-occurrence matrix as feature to training the neural network and these features can be description as the following [Jirari, 2003] :-

- **Entropy**

$$\sum_{i,j} P_{\phi,d}(i,j) \log_2 P_{\phi,d}(i,j) \quad (21)$$

- **Energy or angular second moment:**

$$\sum_{i,j} P^2_{\phi,d}(i,j) \quad (22)$$

- **Maximum Probability:**

$$\max_{i,j} P_{\phi,d}(i,j) \quad (23)$$

- **Inverse Difference moment: κ=2, λ=1**

$$\sum_{i,j;i\neq j} \frac{P_{\phi,d}^{\lambda}(i,j)}{|i-j|^{\kappa}} \quad (24)$$

- **Contrast:**

$$\sum_{i,j} |i-j|^{\kappa} P_{\phi,d}^{\lambda}(i,j) \quad (25)$$

- **Homogeneity:**

$$\sum_i \sum_j \frac{1}{1+(i-j)^2} P_{\phi,d}(i,j) \quad (26)$$

- **Inertia or variance:**

$$\sum_i \sum_j (i-j)^2 P_{\phi,d}(i,j) \quad (27)$$

- **Correlation:**

$$= \frac{\sum_{i,j}\left[(ij)P_{\phi,d}(i,j)\right] - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (28)$$

$$\mu_x = \sum_i i \sum_j P_{\phi,d}(i,j)$$

$$\mu_y = \sum_j j \sum_i P_{\phi,d}(i,j)$$

$$\sigma_x = \sum_i (i-\mu_x)^2 \sum_j P_{\phi,d}(i,j)$$

$$\sigma_y = \sum_j (j-\mu_y)^2 \sum_i P_{\phi,d}(i,j)$$

4.6. Application of LVQ3

After a number of clusters expected in the each frame and the features that represent it which have been determined one can train the supervised Kohonen neural network using winner-take-all learning rule.

The winner- take-all learning rule updates weights of one node in the network called winner node. The winner node has the weight vector similar to the current input. After that the weights update based on the comparing between the class of winner node and the true class that related to input vector. The weights update schema for this learning is [Zurada, 1992] :

$$w' = w + \alpha(x-w) \quad (29)$$

where w' is the altered weight vector, w is the weight vector of the winner node, x the current input vector and the α is the small number, always between 0 and 1, used to make weight change finner.

The updating of the weights continous until the MSE (Mean Square Error) reaches an accepted threshold.

5. Results

In Digital Video Scenes Recognition system two video file movies were taken which are different in natural complexity, scenes quality, and size.

By applying the proposed method on these movie we noticed the recognition ratio increased if the files have few complexity with stable scene (i.e., the file contain one scene) such as the movie number one which contain 16 frames. And by applying the M_{ijn}-EA we found number of cluster=2 the first represented number and the second represented the back ground of object. For M_{ijn}-EA we has used the following parameter (a population size of 50, a maximum number of generation equal to 100, a truncation selection with T=30%.Also in this case we used the following parameters of neural network (m1=0.3,,α=3.67188.10^{-1} , β(t)=m1. α(t), ε=0.35, number of frames used in training stage=2[frame number 1and frame number 5], number of frames used in testing stage =14, maximum number of epoch allowed to the network for training is 1000.

While when applying the proposed method on movie number two which contain 102 frames distributed over several scenes such as tree, river, road, personals, sky, grass,…,ect. We used the following parameters of Mijn-EA(a population size of 80, a maximum number of generation equal to 80, a truncation selection with T=50%), and we found the number of clusters=9 distributed over several scenes. Also in this case we used the following parameters of neural network (m1=0.1,,α=4.22158.10^{-1} , β(t)=m1. α(t), ε=0.35, number of frames used in training stage=32, number of frames used in testing stage =60, maximum number of epoch allowed to the network for training is 5000.

6. Conclusion

The hybrid technique for digital video scenes recognition problem proposed in this research succeeds in recognition scenes for all frames

Using Mijn-EA based clustering; we can obtain the optimal number of seeds that represent each frame and given Kohonen network suitable initialization parameters,

The transform of color from RGB to HSV reduced the inter pixels redundancy, subsequently increased similarities between them, and this simplify the searching in co-occurrence matrix.

The texture features that used in this search is succeed in description all scenes in each frame where the network enable from known all scenes in training set and testing set. And we can consider this point as one of the most conclusions in this work.

References

Beatriz.D, (2004), "*Experiments in Image Segmentation for Automatic us Licens Plate Recognition*", M.Sc.Thesis, the Faculty of the Virginia Polytechnic Institute and State University.

De Falco. I (1997) ," *An introduction to evolutionary algorithms and their application to the aerofoil design problem*", Part I: the algorithms, von Karman lecture series on Fluid Dynamic, Bruxelles,Belgium.

Fogel, L. J., Owens A. J. and Walsh M. J., (1966), "*artificial Intelligence through simulated evolution* ", Wiley, New

Friedberg R. M,(1958) "*A learning machine* ", part I, IMB Journal of Research and Development, 2.

Holtz K., (2004), "*image and video compression techniques*" Site: http://www.Autosophy.com/video com.html

Jirari M., (2003), "*Texture Classification Based on Co-Occurrence Matrices*", Presentation III, Pattern Recognition, spring.

Kalyani. M, Sushmita. M, (2003), " *clustering and its validation in a symbolic framework*", Pattern Recognaition Letters. Pp 2367-2376.

Karkanis S.A , Iakovidis D.K, Maroulis D.E ,Magoulas G.D, Theofanous N.G, (2006), " *Tumor Recognition in Endoscopic Video Images using Artificial Neural Network Architectures*", Dept. of Informatics, University of Athens, TYPA Build. Panepistimiopolis, 15784 Athens.

Nadia. B, (2004), "*Cluster Validity Algorithms*".

Pandya .A.S, Macy .R.B, (1996), "Pattern Recognition with Neural Network in C++", CRC. Press, Florida.

Schwefel, H. P., (1977), "*numerical optimization of computer models*", John Wiley and Sons, New York.
Site:http://www.cs.tcd.ie/Nadia.Bolshakova

Sordo M., (2002)," *Introduction to Neural Networks in Healthcare*", msordo@dsg.bwh.harvard.edu for Open Clinical October.

Wafaa. H.A. , (2006), "*Video clip compression using DCT technique*" Dep. Of computer science, University of Babylon..
York.

Zurada, J. M.,(1992), "*Introduction to Artificial Neural Systems*," ST. Paul, MN: West publishing company.