FCM-Based Approach for Locating Visible Video Watermarks

A. Al. Embaby 1,*, Mohamed A. Wahby Shalaby 1,2 and Khaled Mostafa Elsayed 1

1 Information Technology Department, Faculty of Computers and Artificial Intelligence, Cairo University, Giza 12613, Egypt
2 Smart Engineering Systems Research Center (SESC), Nile University, Giza 12588, Egypt
* Correspondence: embaby.ahmed@gmail.com

Received: 8 February 2020; Accepted: 20 February 2020; Published: 27 February 2020

Abstract: The increased usage demand for digital multimedia has induced significant challenges regarding copyright protection, which is the copy control and proof of ownership. Digital watermarking serves as a solution to these kinds of problems. Among different types of digital watermarking, visible watermarking protects the copyrights effectively, since the approach not only prevents pirates but also visually proves the copyright of the broadcasted video. A visible watermark could be in any location on the frame (corner, center, diagonal, etc.). In addition, it could either completely or partially disappear for some frames. The same video also might have multiple watermarks. In order to strengthen the techniques of adding visible watermarks, there is a need to discover the weakness of the used watermarks. Since the major step of attacking a visible watermark is to locate it accurately, in this paper, a Fuzzy C-Means (FCM)-based approach is proposed to locate visible watermarks in video. Broadcasting channels are used to utilize video logos, which can be considered as a form of visible watermark that represents a trademark or symbol to declare the intellectual property ownership. In general, a high-standard video watermark has such properties as a clear background with distinctive shape without additional texture obscuring the watermark area. In addition, the probability of the logo appearing in the four corners of the video frames is higher than in the center. Based on these common properties of the video watermark, the proposed scheme locates the visible watermark using the Fuzzy C-Means technique without any prior information. The proposed technique has two stages: the first stage is positioning, and the second is masking (extracting the watermark mask). Due to real-world limitations such as noise, shadowing, and variations in cameras, the positioning stage is developed by employing gradient and Fuzzy C-Means classifier techniques. By using the dilation and erosion operators, the masking stage is developed to extract the watermark mask. Using a set of trademark videos, the proposed algorithm is tested and evaluated. A comparative study shows that the proposed FCM-based technique is able to achieve higher accuracy at a reasonable computational cost in comparison to the most related and recent published work. The proposed technique could locate different watermarks with high symmetry in their pattern, even if they appeared mutually in the same location. Still, it will be a challenge if the symmetry is low between used watermarks in the same location.

Keywords: dilation; erosion; Fuzzy C-Means; gradient; visible watermark

1. Introduction

In recent years, digital watermarking techniques have been extensively exploited and regarded as a potentially effective solution against illegal reproduction or theft of multimedia contents. Visible watermarking schemes are widely used, marking and protecting the copyright of digital images or videos for specific purposes, such as digital contents used in distant learning or digital libraries, while illegal copying or reproduction is forbidden. Visible watermarking schemes protect intellectual
property rights (IPR) in a more active way. The visibly watermarked content often contains recognizable but unobtrusive copyright patterns indicating the identity of IPR owners.

Different from invisible watermarking, visible watermarking consists of the overlaying of a word mark related to ownership into the original video in a noticeable way. Therefore, visible watermarking can fulfill copyright protection requirements in a straightforward and immediate way than invisible watermarking. In most cases, the embedded visible watermark may affect the quality value of the digital video, in spite of the fact that the watermark is semi-transparent; therefore, recently, several removable visible watermarking techniques were proposed [1–3]. The application scope of permanent visible watermarking become more and more comprehensive. Examples such as the digital libraries, e-commerce, broadcast monitoring, video tracking, as well as digital press require the use of permanent visible watermarking. Digital libraries widely uses visible watermarks with their digital contents, and the users of this digital contents can use them legally for reading and viewing. However, they cannot re-use these digital content for other purposes, such as illegal sale, due to the visible watermark [4].

Visible watermarking techniques should meet some properties [5,6], which are as follows:

1. **Perceptibility:** a visible watermark should be noticeable in gray and color host images/videos.
2. **Distinguishably:** a visible watermark should be clear enough to be recognized or identified as different in any region of the hosted image if the region has a different texture, plain, and edge.
3. **Noticeably:** a visible watermark should not be too noticeable, so the quality value of the hosted image/video remains acceptable.
4. **Transparency:** a visible watermark should not conceal or brighten the host image by a notably large amount; the watermarked area should not have any artifacts or feature loss, and it should remain perceptible by the Human Visual System (HVS), with no impact on non-watermarked areas.
5. **Robustness:** a visible watermark should be optimized to be able to survive against common kinds of attacks.
6. **The watermark embedding process should be automatic for all kinds of images/videos.**

Concerning the robustness of watermarks, the visible watermarking essentially provides robustness against common kinds of attacks, because the embedded visible watermark can be recognized and identified easily by the Human Visual System (HVS). There are different types of watermark attacks, namely, geometrical attacks and signal processing-based attacks. The geometrical attacks include the scaling, rotation, and transformation of the image. In addition, signal processing-based attacks could be made using compression, noise addition, filtering, and the modification of brightness and contrast [7,8]. The first step to attack a visible watermark in the video is the localization of a watermark, and thus the watermark will be easily attacked. Videos often have multiple overlapped watermarks in the same place. While it serves the broadcasters’ intention of announcing the ownership, it degrades the viewing experience because of the constant obscuring of part of the content. Consequently, it would be interesting to see how such logos could be localized in order to be attacked (removed). Hence, there is a need to develop a technique for localizing such logos and extracting a mask of the undergoing watermark/s in order to be attacked. Such techniques will help to highlight the weakness in existing schemes for adding watermarks and thus to strengthen the visible watermark against attackers.

In the literature, the watermarks in videos could be broadly categorized as follows:

1. **2D and 3D watermarks:** based on dimensionality, the embedded watermark can be 2D or 3D.
2. **Fixed or moving watermarks:** the watermark can change its location on the screen (i.e., from one corner to another), and other watermarks can be fixed.
3. **Single or multiple watermarks:** in the same video, we can have single or multiple watermarks in different locations on the video frame/s.

Over the past two decades, many techniques have been proposed to handle the video streams in the compressed domain directly for real-time applications [9–12]. D. Lin and G. J. Liao [13] use
FCM and introduced a technique for watermark embedding on compressed video. In [13], they use the FCM to find the best location in the video where the watermark can be embedded.

In the field of watermark detection, some researchers have proposed compressed domain-based techniques [11,12]. However, due to the nature of the visible watermarks in the spatial domain, other researchers have used base-band techniques to process the watermarks [14]. The easiest and most straightforward method of localizing a visible watermark in the video is to depend on prior knowledge, i.e., the corner, the sample, or the template of the watermark. Template matching is widely used for logo recognition because of its simplicity and capability to handle different types of objects. Many state-of-the-art approaches for template matching have been used for logo detection and recognition [15–17]. However, the template matching techniques are not efficient in case of video watermarks with a change of location and appear/disappear partially or totally during the video. Traditional logo detection techniques are often based on object detection approaches, which used statistical learning to build a classifier offline and later used it in real-time for recognition [18].

Until now, different types of visible watermarking techniques are introduced. One of the vital embedding domains is the frequency domain, on which the watermark is embedded into the cover image/video spectrum, thus not directly impacting the selected image/video quality. The most widely used transforms are Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), and Discrete Wavelet Transform (DWT). The watermark embedding system injects the watermark equally on the original data. In [19], a DCT-based visible watermarking technique was introduced. Different types of research are focused on DWT because of its multi-resolution property. In [20], the researchers use the DWT to introduce a contrast-sensitive visible watermark scheme. In addition, the DWT can be joint with other algorithms to increase the robustness of the watermark-embedding techniques. At the same time, the invisible watermarking techniques are still evolving. ref.[21] used a robust watermarking based on DCT and Just Noticeable Distortion (JND). The proposed JND model utilizes both orientation diversity and local color complexity and selects the optimum quantization step while embedding the watermark.

In addition, there are many other techniques that have been developed to achieve better results for watermark attack. Traditionally, the manually-based techniques, which are designed based on the manual selection, identify the region of the video logo first and then select the best logo region among all the frames [18]. In [1], Yan, Wang, and Mohan proposed two different approaches: the first approach depends on the color distance of corresponding pixels between two neighboring frames, and based on that, it builds a binary array using a predefined threshold. This technique is highly sensitive to the chosen threshold and the variation between two frames. The second approach proposed by Yan et al. [1] was based on the Bayesian classifier. It used local features instead of the global features of the logo to build its classifier. Although the problem in [1] can be solved effectively by the multi-frame gray-level technique [7], they [1,7] cannot widely be used for transparent watermark detection and text watermark application. On the other hand, localizing the watermark mask in the entire video sequence is another important issue. Chung-Ming and Jin-Long [2] have proposed a gradient-based average approach for logo detection. This gradient-based approach computes the gradient for all frames; then, by the manually selected threshold, it converts the gradient of each frame to binary (0, 255), averages the gradient of all frames, and then applies a manually selected threshold again on the output after averaging. However, the proposed algorithm in [2] was able to achieve better results in case of a transparent logo; it suffered from the problem with the text watermark and the high computational complexity. Since the watermark logo has a short period of time to stay stable, Xinwei Wang and Shanzhen Lan [22] have proposed a time-averaged edge algorithm based on shot segmentation and gradient map working only on logos located at frame corners. In [23], they observed that the brightness of the logo is generally high for visibility, the lower the brightness of the Region of Interest (ROI). Based on this observation, they proposed an adaptive weighted averaging method depending on the brightness of the ROI of the input image (watermarked frame). In [23], after detection of the logo, they added another step for verification. H. Garcia [24] proposed a new visible watermark detection technique based on the Total
Variation where the Otsu’s threshold selection algorithm was also used to identify the watermarked pixel from the original edge pixels.

It is clearly seen from the previous literature that the real-world limitations, such as noise, shadowing, and variations in cameras, are significant factors that make the problem of localizing watermarks more challenging. For decades, fuzzy logic and fuzzy-based techniques have been used successfully to deal with a wide variety of challenges in different research areas [25–27]. In addition, it is well known that the Fuzzy C-Means classifier is an efficient classifier that has been used in different research areas to improve the accuracies of clustering problems [27,28]. In this research work, a novel FCM-based approach for locating visible video watermarks is proposed. In this novel approach, the visible video watermark is located using the Fuzzy C-Means technique without any prior information about the position of the watermark. The proposed technique contains two stages: the first stage is positioning, and the second is masking (extracting a watermark mask). The positioning stage is designed by employing gradient and Fuzzy C-Means Classifier techniques. Then, by using the dilation and erosion operators, the masking stage is developed to extract the watermark mask.

The rest of this paper is organized as follows: In Section 2, the overall proposed approach is discussed. The proposed FCM-based technique is detailed in Section 3. The experimental results and comparisons are presented in Section 4. Finally, the conclusion and future work are discussed in Section 5.

2. Proposed FCM-Based Approach for Locating a Visible Video Watermark

A video watermark is a picture superimposed on several consecutive video frames. Thus, it can be characterized as a spatial region in a video where the edge locations remain constant in consecutive frames, and the temporal variance of the appearance of the pixels is lower than that from the non-watermark pixels (frame background). It is noted that the transparent watermarks also fulfill previous properties [28]. In an image, there is normally a significant correlation between any pixel value and the sum or variance of its adjacent pixels. When an image undergoes some attacks, such as compression or distortion, this relationship is unchanged or changes insignificantly. In the proposed approach, this type of correlation provides a significant factor for accurate locating/extracting of the watermark.

A video watermark that meets the high standard should offer a plain background with distinct contrast without artifacts or feature loss on the watermarked area [1]. There are different categories of the visible watermark based on its properties:

- **Shape-based category**: opaque, transparent, and animated.
- **Dimensionality-based category**: 2D and 3D.
- **Image-based category**: text or image.
- **Location-based category**: fixed frame location (the watermark displayed on a fixed location on the frame) or different location on different frames.

Based on the above properties, a novel FCM-based approach for locating visible watermarks is proposed. The proposed FCM based technique contains two stages: (1) locating the watermark, and (2) the masking stage. In the first stage, the visible watermark is automatically located without any prior knowledge based on the FCM technique. Then, the second stage is the masking stage, which prepares a mask for the watermark to be inpainted for removal purposes. The overall block diagram of our proposed scheme is shown in Figure 1. It is seen from Figure 1 that the above-mentioned two stages contain subtasks. The first subtask is to employ a smoothing filter by applying the averaging on the color intensity (in RGB color space, in R Channel). This smoothing filter uses the constraints of the watermark; in other words, a good video watermark displays a clear background with distinct contrast without additional texture obscuring the watermark area. So, when smoothing all the frames, the watermark will be the only object that is clearly maintained. At that moment, the watermark is dominated but still surrounded by non-deterministic noises, since it depends on the video content, camera position, shadow, and other factors. Then, a Fuzzy partitioning subtask is carried out through
an iterative optimization using FCM to separate the watermark from this non-deterministic noise. In the second phase, morphological dilation operators are employed for the construction of the watermark region.

![Flowchart](image)

**Figure 1.** The proposed Fuzzy C-Means (FCM)-based scheme flowchart.

3. Two Stages of FCM-Based Technique for Locating Visible Watermark in Video

3.1. **Edge Extraction Stage**
Edge extraction is the first stage of the efficient technique for locating visible watermarks in video. In this stage, we depend on the constraints of the visible watermarks in video. A good video watermark displays a clear background with distinct contrast without additional texture obscuring the watermark area. Stage 1 consists of three steps: smoothing, gradient, and FCM clustering.

3.1.1. Smoothing

Since the watermark has a stable position for a longer duration within a video so that it can draw the attention of viewers and thus be identified as a trademark [30], the averaging step of Stage 1 is applied. Watermarks that change location are out of our scope.

Suppose a video with a watermark is denoted by

\[ V = \left( v_{i,j,k} \right)_{L \times W \times H}, \]  

where \( V \) is one of the color channels, \( L \) is the total number of frames in the video, and \( W \) and \( H \) are the width and height of the video frames, respectively. The average will be computed as shown in Equation (2):

\[ A(i,j) = \frac{1}{L} \sum_{k=1}^{L} v(i,j,k). \]  

Figure 2 shows samples of watermarked frames from one video, and Figure 3 shows an example of the output from the averaging step when applied to one channel (R Channel) of the given video, from which we can prove the efficiency of locating different kinds of the watermark at the same time on the same video. For example, regarding the text watermark displayed in the middle (www.videoyoum7.com), although it is appearing and disappearing gradually during the video time, we are able to extract it by this step. In addition, the watermark in the top left corner is an animated watermark, but also we succeed in locating it by all its parts. There is another kind of watermark on the given example that appears for a while in the middle of the video; then, it disappears, similar the one in the bottom left corner.
3.1.2. Gradient

The gradient of an image can be defined as the directional change in the pixel intensities of an image. Mathematically, the gradient is computed by the derivatives in the horizontal and vertical directions at each image pixel. At each image location, the direction of the gradient would be the direction of the increasing intensity, and the magnitude of the gradient would define the rate of change in that direction.

The edge detector is important to enhance the connected gradients in an image, which may take the form of an edge, contour, line, or some connected set of edges. Many edge detectors are simply implemented as kernel operations or convolutions. We use a Sobel kernel to compute the gradient of the output image (A) from Step 3.1.1. A Sobel operator uses two $3 \times 3$ kernels, which are convolved with the original image to calculate approximations of the derivatives: one for horizontal changes, and one for vertical. The Sobel edge operator masks are given as $\Delta_x$ and $\Delta_y$. 
\[
\Delta_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \tag{3}
\]

\[
\Delta_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \tag{4}
\]

\[
G_x = \Delta_x \ast A, \tag{5}
\]

\[
G_y = \Delta_y \ast A. \tag{6}
\]

where * is the 2D convolution operation.

The operator calculates the gradient of the image intensity at each location, giving the direction of the most substantial possible increase from light to dark and the rate of change in that direction. Therefore, the result shows how “abruptly” or “smoothly” the image changes at that point and, therefore, how likely it is that part of the image represents an edge, as well as how that edge is likely to be oriented [31]. In practice, the magnitude (likelihood of an edge) calculation is more reliable and easier to interpret than the direction calculation. The combined results of Equations (3) and (4) find the absolute magnitude of the gradient as follows:

\[
|G| = \sqrt{G_x^2 + G_y^2}. \tag{7}
\]

Figure 4 shows an example of the gradient of averaging output in Figure 2, from which we can see how efficiently the edges of the watermark objects are extracted. We can use the MATLAB function \textit{imgradient} to implement this step.

![Gradient after averaging](image)

\textbf{Figure 4.} Gradient after averaging.

3.1.3. Fuzzy C-Means Clustering (FCM)

Now, after getting the gradient and identifying clear watermark edges, we need to classify the output from the gradient step in order to remove spurious pixels and dominate the watermark for
the next stage (masking stage). Unlike [2,18], we did not depend on thresholds to classify the watermark pixels; we used FCM, which give us the best result for the overlapped dataset and is comparatively better than the K-Means algorithm. The Fuzzy C-Means technique is a method of clustering that allows each piece of data to belong to two or more clusters with different degrees of membership [32]. This method is based on the minimization of the following objective function

\[ J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} \mu_{ij} m \left\| x_i - c_j \right\|^2, \quad (8) \]

where \( m \) is any real number greater than 1, \( x_i \) is the \( i \)th d-dimensional sample data, \( c_j \) is the \( d \)-dimension center of the cluster, \( \|*\| \) is any norm expressing the similarity between any measured data and the center, \( \mu_{ij} \) is the degree of membership of \( x_i \) in the cluster \( j \), which is called subject degree, \( N \) is the number of sample data, and \( C \) is the number of clusters.

FCM works by assigning membership to each data point corresponding to each cluster center based on the distance between the cluster center and the data point. The closer the data is to the cluster center, the more its membership is toward a particular cluster’s center. Clearly, the summation of membership of each data point should be equal to one. After each iteration, the membership and cluster centers are updated according to Equations (10) and (11).

\[ \mu_{v} = \frac{1}{\sum_{c=1}^{C} \left( \frac{d_{ij}}{d_{ik}} \right)^{m-1}} \quad (9) \]

\[ c_k = \frac{\sum_{i=1}^{N} \mu_{ik}(x)^m x}{\sum_{i=1}^{N} \mu_{ik}(x)^m} \quad (10) \]

In the proposed approach, we divided the gradient image into non-overlapping blocks, such that each block size is 3 x 3. Since we have to classify the image pixels to be either watermark pixels, therefore, we set \( C = 2 \). Then, the two cluster centers are initialized randomly. We tested this technique using different datasets. It is found that on average, the FCM converges after 15 iterations. The datasets have been selected to cover all possible varieties of the watermarks. Then, these datasets are classified based on the dimensionality, camera status, and watermark category. Based on dimensionality, we selected the following dimensions [720 x 1280 x 3], [640 x 360 x 3], and [320 x 240 x 3]. Some of these selected datasets have been captured by a fixed camera, and others have been captured by a movable camera. Finally, these different datasets have different types of watermark category: namely, static, animated, 2D, and 3D watermarks.

Figure 5 shows the output after classifying the input gradient using FCM. As it is clear from the output, all watermarks are dominated perfectly. As a result of the FCM step, we will have a binary image based on the classified two clusters. This binary image will be the input to Stage 2, as shown in Figure 1. We can use the MATLAB function \texttt{fcm} in the implementation of this step as

\[ [\text{center}, \text{U}, \text{obj}_\text{func}] = \text{fcm}(I, C); \]

where \( I \) is the gradient after averaging and reshaped as per the required block size, and \( C \) is the number of classes; in our case, \( C \) must equal to 2.
3.2. Masking Stage

Our objective in this step is to dominate the watermarked area/s (Region of Interest) in order to facilitate other operations such as the inpainting of those ROI.

3.2.1. Binary Dilation

Normal dilation, as implemented in MathWorks [33], is used. The binary dilation of a set $X$ by structure element $B$ is denoted as $X \oplus B$, and it can be defined as Equation (11):

$$ (X \oplus B) = \{ z \mid \hat{B}_z \cap X \neq \emptyset \}, $$(11)

where we need to find the set of pixels $Z$, such that the shifted structure element $\hat{B}$ has any overlap with the foreground pixels in $X$.

Generally, the gray-scale dilation of $X(i, j)$ by $B(i, j)$ is defined as:

$$ (X \oplus B)(i, j) = \max \{ X(i-i', j-j') + B(i', j') \mid (i', j') \in D_B \} $$

(12)

where $D_B$ is the domain of the structuring element $B$ and $X(i, j)$ is assumed to be $\infty$ outside the domain of the image.

To be sure that all the watermark areas are dominated, in case of an animated watermark, we may lose small gaps in the border of the watermark area. In this step, we are trying to probe and expand the watermark shapes in order to include the missing small areas. To reach our objective, we applied two flat structuring elements (se1, se2).

To be sure that all the watermark areas are dominated, in case of an animated watermark, we may lose small gaps in the border of the watermark area. In this step, we are trying to probe and expand the watermark shapes in order to include the missing small areas. To reach our objective, we applied two flat structuring elements (se1, se2):

$$ se1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} $$

(13)

$$ se2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} $$

(14)

We can use the MATLAB function `imdilate` to implement this step as

`imdilate(B, [se1 se2]),`
where B is the binary output from the FCM. Figure 6 shows the output from the dilation step.

![Figure 6. Flat dilation output.](image)

3.2.2. Filling Interior Gaps

In order to build a watermark mask, we need to fill the watermarks’ holes. Unlike Flood fill, we need to fill binary mask holes without a starting point. We used the implemented MathWorks method in MATLAB [33], where the unreachable background pixels from the edge of the mask are considered as a hole and will be filled in. We can use MATLAB function *infill* as

\[ F = \text{imfill}(D, \text{‘holes’}), \]

where D is the dilated mask. Figure 7, shows the output after filling step.

![Figure 7. Filled watermarks.](image)

3.2.3. Smooth Watermark Objects Mask

In this step, we applied two morphological operations; in the first operation, we cleared the border of objects. In the second step, we smoothed the watermark mask by applying binary erosion. In both steps, we use the MathWorks morphological package [33]. The MATLAB function for the first step and second step respectively are

\[ S = \text{imclearborder}(F, \text{Con}); \]  
\[ \text{Maskfinal} = \text{imerode}(S, \text{seD}); \]

where Con is the connectivity, we set it to 4. seD is the structure element, we used ‘diamond’ in our implementation. Figure 8, shows the output final mask after smoothing step.
4. Experimental Results and Comparisons

Based on the previous section, the proposed FCM-based technique, for watermark removal, proved its capability using a test case. This test case is a representation of a movable camera, movable objects, animated watermark, and multiple watermarks in the same video. In this section, to show the strength of the proposed technique, our technique has been validated against different videos with different watermark properties. We validated the FCM-based technique using four different categories of video sets:

1. Fixed camera, fixed object.
2. Fixed watermark location without animation.
3. Multiple animated watermarks in the same video.
4. 3D animated watermark.

Figure 9 shows the watermark mask extracted from a video with a fixed camera and a fixed object. From the result, we realized how accurate the FCM-based technique is comparable to the gradient-based algorithm in [2,16]; using our FCM-based algorithm, we can locate the tiny logo of the CNN news channel in the right bottom, while it is not extracted with the gradient-based algorithm. In addition to that, the FCM-based algorithm computation complexity is not functional in the number of video frames, which boosts the speed of locating the watermark mask.

Figure 10 show an extracted watermark mask from a video where the watermark has a fixed location and does not have any animation.
**Figure 10.** Fixed location of watermark without animation.

Figure 11 shows a result of multiple animated watermarks in the same video.

**Figure 11.** Multiple animated watermarks in the same video.

Regarding the 3D watermark category, we validated our algorithm against this case. Figure 12 shows the extracted watermark mask from a 3D animated watermark. As a result, we found that the FCM-based technique achieves an outstanding result in that domain.

**Figure 12.** 3D animated watermark mask.

**Qualitative Comparisons.** Figure 13 shows the comparisons of test videos, where FCM Based represents our FCM-based technique. We compare with a gradient-based technique [2], and it can be seen that the gradient-based technique can lose the tiny watermarks as shown in Case 4, but it still able to allocate those that appear partially during the movie, as shown in Case 1. However, in general, it is not showing the good quality of locating the logo (watermark area). We also compare with the OTSU-based technique [35]. In comparison to the proposed FCM-based technique, the OTSU algorithm was used to determine the watermark pixels based on a selected optimum threshold. It is clearly seen that the OTSU-based technique suffers from too much noise in case of detecting the 3D watermark. In addition, the OTSU-based technique is unable to locate those watermarks that appear for a short period of time, as it is seen in the first case of Figure 13.
Figure 13. Comparisons of testing results.
From the above results and comparisons, it is seen that the proposed fuzzy-based scheme is capable of extracting all the watermark masks from different types of videos efficiently.

**Performance Measure**

In order to analyze the effectiveness of the proposed FCM-based approach for locating visible watermarks, four different categories of video sets have been used. As described in this section, these video sets have different watermark types, and a different amount of watermarks per video that have varieties of difficulties. Since there are no well-defined measuring criteria for measuring the performance of the watermark localization approach, we used the measurements commonly used in the literature to measure the performance of the proposed approach. As in [27], the performance can be measured in terms of the next three criteria.

\[
Recall = \frac{TP}{TP + FN} \quad (15)
\]

\[
Precision = \frac{TP}{TP + FP} \quad (16)
\]

\[
F\text{-measure} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \quad (17)
\]

where True Positive (TP) is the number of localized areas of the true watermark. False Positive (FP) is the number of localized areas where no watermark exists. False Negative (FN) is the number of areas we missed, while it represents a real watermark. Therefore, **Recall (or detection rate)** is the ratio of correctly locating watermark/s area on the video to all the correct watermark areas. **Precision** is the ratio of correctly detected watermark areas to all localization made by the approach. The F-measure is the trade-off between detection rate and precision, giving equal importance to both. Increasing the precision may minimize the Recall, while the F-Measure can be used to join both properties (Recall and Precession) in one measure. The F-Measure concept was derived from the harmonic mean [37] of Precision and Recall.

Table 1 shows the performance measurement results of the proposed FCM-based technique at different FCM block sizes. From the reported measures in Table 1 and Figure 14, the effectiveness of the block size 3 × 3 is clearly seen.

| Block Size | False Positive | False Negative | True Positive | Recall     | Precision   | F-Measure   |
|------------|----------------|----------------|--------------|------------|-------------|-------------|
| 3 × 3      | 10             | 3              | 212          | 0.986047   | 0.954955    | 0.970252    |
| 5 × 5      | 19             | 5              | 201          | 0.975728   | 0.913636    | 0.943662    |
| 9 × 9      | 19             | 15             | 191          | 0.927184   | 0.909524    | 0.918269    |
Experimental work is done using a machine with windows 10(Inteli7) and using MATLAB R2015a (64-bit) with different video lengths (30, 60, and 120 s). The execution time of different techniques is reported in Table 2. All reported execution times include morphological operation. From the reported results, we can prove that the proposed technique processing time is not linear to the video length. However, OTSU requires less processing time but also less quality. The proposed technique requires 5% extra execution time over the OTSU, but with much better quality. At the same time, the proposed technique enhances the processing time over the gradient-based technique by more than 20% and is of better quality. Moreover, the processing time in the proposed technique does not linearly increase as the video length increases as in the gradient-based algorithm. The proposed technique minimizes the processing time by minimizing the operation required to be done in all video frames, without affecting the accuracy of watermark mask extraction. Therefore, the proposed technique can be used as a preprocessing phase of a watermark removal scheme, which helps develop efficient watermark attacking strategies.

Table 2. Comparison of execution time (in seconds).

|          | OTSU    | Gradient Based | Proposed |
|----------|---------|----------------|----------|
| Video 30 s | 115.16  | 130.96         | 128.07   |
| Video 60 s | 180     | 240.66         | 190.9    |
| Video 120 s | 250     | 380.29         | 280.02   |

In our experimental study, the datasets have been selected from different types of visible watermarks in order to have a generalization of the solution. It is clearly seen from the experimental results that the performance of the proposed technique consistently provides high accuracy for different types of visible watermarks.

5. Conclusions and Future Work

In this paper, the research goal is to provide an efficient technique to locate a visible watermark/s in the video without prior information. The challenge in all the literature is how to distinguish between background pixels and watermark pixels. Despite seeing it as a classification problem, all of them have tried to resolve it with either a simple solution by applying a fixed threshold or a very complex solution that requires too much computational cost. Based on the Fuzzy C-means clustering technique, a novel approach has been proposed to classify the watermark pixels from the background ones efficiently. Since the localization of visible watermarks is strongly related to the Human Visual
System (HVS), the proposed approach used the fuzzy classifier because it is one of the classifications algorithms that is more closely related to a human being.

It is shown that the proposed FCM-based technique was not limited to the corner location watermarks. It is able to determine the watermarks in any location without prior knowledge. By experimental results, the best block size to be used with the FCM is $3 \times 3$ pixels. We also noticed that the fixed text displayed on the news bar is considered as a watermark, and it is included on the mask. It is noticed that the proposed technique execution time increased as the size of the frame increased, but it is not exponentially affected by the number of frames on the video. Finally, it is recommended for the video watermarking techniques to use different watermark shapes (asymmetric in their structure) in the same video to be displayed alternately in the same location, not as it is now in most, if not all, which display animated watermarks but consisting of the same shape in the same location.

As future work, it is recommended to depend on the "keyframe window" technique in order to make the process independent of the video length as much as possible. In the keyframe window technique, we will determine the keyframes from the video and then use a window of 10 frames (before and after the keyframe) as an input to our technique instead of the whole video frame, as it is now.

**Author Contributions:** Conceptualization, A.A.E. and K.M.E.; methodology, M.A.W.S. and A.A.E.; software, A.A.E., K.M.E., and M.A.W.S.; writing—original draft preparation, A.A.E.; writing—review and editing, M.A.W.S.; visualization, A.A.E.; supervision, K.M.E. and M.A.W.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Yan, W.; Wang, J.; Kankanhalli, M.; Automatic video logo detection and removal. *ACM Trans. Multimed. Syst.* **2005**, 10, 5.
2. Kuo, C.; Chao, C.; Chang, W.; Shen, J. Broadcast Video Logo Detection and Removing. In Proceedings of the IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Harbin, China, 15–17 August 2008.
3. Hu, Y.; Kwong, S.; Huang, J. An algorithm for removable visible watermarking. *IEEE Trans. Circuits Syst. Video Technol.* **2006**, 16, 129–133.
4. Garcia, H.; Navarro, E.; Reyes, R.; Ramos, C.; Miyatake, M. Visible Watermarking Technique Based on Human Visual System for Single Sensor Digital Cameras. *Secur. Commun. Netw.* **2017**, 2017, 7903198.
5. Huang, C.; Wu, J. Attacking visible watermarking schemes. *IEEE Trans. Multimed.* **2004**, 6, 16–30.
6. Kankanhalli, M.R.; Ramakrishnan, K. Adaptive visible watermarking of images. In Proceedings of the 6th IEEE International Conference on Multimedia Computing and Systems, Florence, Italy, 7–11 June 1999.
7. Langelaar, G.; Setyawan, I.; Lagendijk, R. Watermarking digital image and video data. A state-of-the-art overview. *IEEE Signal Process. Mag.* **2000**, 17, 5.
8. Voloshynovskiy, S.; Pereira, S.; Pun, T.; Eggers, J.; Su, J.K. Attacks on digital watermarks: Classification, estimation-based attacks, and benchmarks. *IEEE Commun. Mag.* **2001**, 39, 118–125.
9. Darwish, A.M. A video coprocessor: Video processing in the DCT domain. *Proc. SPIE Media Processors* **1999**, 3655, [DOI: 10.1117/12.334761](https://doi.org/10.1117/12.334761).
10. Wahby, A.M.; Mostafa, K.; Darwish, A.M. DCT-based MPEG-2 programmable coprocessor. In Proceedings of the SPIE International Society for Optical Engineering, San Jose, CA, USA, 2002(20 Dec 2001); Volume 4674, pp. 14–20, [DOI: 10.1117/12.451072](https://doi.org/10.1117/12.451072).
11. Fallahpour, M.; Shirmohammadi, S.; Semsarzadeh, M. Tampering Detection in Compressed Digital Video Using Watermarking. *IEEE Trans. Instrum. Meas.* **2016**, 5, 1057–1072.
12. Lee, M.; Im, D.; Lee, H.; Kim, K.; Lee, H. Real-time video watermarking system on the compressed domain for high-definition video contents: Practical issues. *Digit. Signal Process.* **2012**, 22, 190–198.
13. Lin, D.; Liao, G. Embedding Watermarks in Compressed Video using Fuzzy C-Means Clustering. In Proceedings of the 2008 IEEE International Conference on Systems, Man and Cybernetics, Singapore, 12–15 October 2008.
14. Li, Z. The study of security application of LOGO recognition technology in sports video. *Eurasip J. Image Video Process.* 2019, 1, 46.
15. Lowe, D.G. Distinctive image features from scale invariant key points. *Int. J. Comput. Vis.* 2004, 60, 91–110.
16. Bay, L.V.; Tuytelaars, H.; Gool, T. Surf: Speeded up robust features. In *ECCV*; Springer: Berlin/Heidelberg, Germany, 2006.
17. Hinterstoisser, S.; Lepetit, S.; Ilic, S.; Fua, P.; Navab, N. Dominant orientation templates for real-time detection of texture-less objects. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
18. Yan, W.Q.; Kankanhalli, M.S. Erasing video logos based on image inpainting. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2002), Lausanne, Switzerland, 26–29 August 2002.
19. Kankanhalli, M.R. A DCT domain visible watermarking technique for images. In Proceedings of the IEEE International Conference on Multimedia and Expo, New York, NY, USA, 30 July–2 Aug 2000.
20. Huang, B.; Tang, S. A contrast-sensitive visible watermarking scheme. *IEEE Multimed.* 2006, 13, 60–66.
21. Wang, J.; Wan, W.; Li, X.; Sun, J.; Zhang, H. Color image watermarking based on orientation diversity and color complexity. *Expert Syst. Appl.* 2020, 140, 112868.
22. Wang, X.; Li, D.; Li, S.; Lan, S. Video Corner-Logo Detection Algorithm based on Gradient Map of HSV. In Proceedings of the 2nd IEEE International Conference on Computer and Communications, 2016.
23. Kim, H.; Kang, M.; Ko, S. An Improved Logo Detection Method with Learning-based Verification for Video Classification. In Proceedings of the 2014 IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-Berlin), Berlin, Germany, 7–10 September 2014.
24. Garcia, H.; Navarro, E.; Reyes, R.; Perez, G.; Miyatake, M.; Meana, H. An Automatic Visible Watermark Detection Method using Total Variation. In Proceedings of the 5th International Workshop on Biometrics and Forensics (IWBF), Coventry, UK, 4–5 April 2017.
25. Shalaby, M.A.W.; Ortiz, N.R.; Ammar, H.H. (2020) A Neuro-Fuzzy Based Approach for Energy Consumption and Profit Operation Forecasting. In *Advances in Intelligent Systems and Computing*; Hassanien, A., Shaalan, K., Tolba, M., Eds.; *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics, Cairo, Egypt, 26–28 October 2019*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 1058.
26. Khaled, K.; Shalaby, M.A.W.; El Sayed, K.M. Automatic fuzzy-based hybrid approach for segmentation and centerline extraction of main coronary arteries. *Int. J. Adv. Comput. Sci. Appl.* 2017, 8, 258–264.
27. Shalaby, M.A.W. Fingerprint Recognition: A Histogram Analysis based Fuzzy C-Means Multilevel Structural Approach. Ph.D. Thesis, Concordia University, Montreal, QC, Canada, 20 April 2012.
28. Bezdek, J.C.; Keller, J.; Krishnapuram, R.; Pal, N.R. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*; Springer: New York, NY, USA, 2005.
29. Cozar, J.R.; Nieto, P.; Hernández-Heredia, Y. Detection of Logos in Low Quality Videos. In Proceedings of the 11th International Conference on Intelligent Systems Design and Applications, Córdoba, Spain, 22–24 November 2011.
30. Holland, R.J.; Hanjalic, A. In Proceedings of the 2013 IEEE International Conference on Image Processing, Melbourne, Australia, 15–18 September 2003.
31. Vincent, O.R.; Folowusoro, O. A Descriptive Algorithm for Sobel Image Edge Detection. In Proceedings of the Informing Science & IT Education Conference (InSITE), Macon, France, 12–15 June 2009.
32. Dunn, J.C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well Separated Clusters. *J. Cybern. Vol 3, 1973*, 32–57.
33. The MathWorks, Inc. Available online: https://www.mathworks.com/ (accessed on 1 Feb 2020).
34. Albion, M.J.; Ch, F.A.; Torres, A. Detection of TV commercials. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Montreal, QC, Canada, 17–21 May 2004; Volume 3.
35. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. ManCybern.* 1979, 9, 62–66.
36. Lin, P.-Y.; Chen, Y.-H.; Chang, C.-C.; Lee, J.-S. Contrast adaptive removable visible watermarking (CARVW) mechanism. *Image Vis. Comput.* 2013, 31, 311–321.

37. Xia, D.; Xu, S.; Feng, Q. A proof of the arithmetic mean-geometric mean-harmonic mean inequalities. *Res. Rep. Collect.* 1999, 2, 85-87.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).