

Research Article

Optimized K-Means Algorithm

Samir Brahim Belhaouari,¹ Shahnawaz Ahmed,¹ and Samer Mansour²

¹ *Department of Mathematics & Computer Science, College of Science and General Studies, Alfaisal University, P.O. Box 50927, Riyadh, Saudi Arabia*

² *Department of Software Engineering, College of Engineering, Alfaisal University, P.O. Box 50927, Riyadh, Saudi Arabia*

Correspondence should be addressed to Shahnawaz Ahmed; math.shan@gmail.com

Received 18 May 2014; Revised 9 August 2014; Accepted 13 August 2014; Published 7 September 2014

Academic Editor: Gradimir Milovanović

Copyright © 2014 Samir Brahim Belhaouari et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The localization of the region of interest (ROI), which contains the face, is the first step in any automatic recognition system, which is a special case of the face detection. However, face localization from input image is a challenging task due to possible variations in location, scale, pose, occlusion, illumination, facial expressions, and clutter background. In this paper we introduce a new optimized k-means algorithm that finds the optimal centers for each cluster which corresponds to the global minimum of the k-means cluster. This method was tested to locate the faces in the input image based on image segmentation. It separates the input image into two classes: faces and nonfaces. To evaluate the proposed algorithm, MIT-CBCL, BioID, and Caltech datasets are used. The results show significant localization accuracy.

1. Introduction

The face detection problem is to determine the face existence in the input image and then return its location if it exists. But in the face localization problem it is given that the input image contains a face and the goal is to determine the location of this face. The automatic recognition system, which is a special case of the face detection, locates, in its earliest phase, the region of interest (ROI) that contains the face. Face localization in an input image is a challenging task due to possible variations in location, scale, pose, occlusion, illumination, facial expressions, and clutter background. Various methods were proposed to detect and/or localize faces in an input image; however, there are still needs to improve the performance of localization and detection methods. A survey on some face recognition and detection techniques can be found in [1]. A more recent survey, mainly on face detection, was written by Yang et al. [2].

One of the ROI detection trends is the idea of segmenting the image pixels into a number of groups or regions based on similarity properties. It gained more attention in the recognition technologies, which rely on grouping the features vector to distinguish between the image regions, and then

concentrate on a particular region, which contains the face. One of the earliest surveys on image segmentation techniques was done by Fu and Mui [3]. They classify the techniques into three classes: features shareholding (clustering), edge detection, and region extraction. A later survey by N. R. Pal and S. K. Pal [4] did only concentrate on fuzzy, nonfuzzy, and colour images techniques. Many efforts had been done in ROI detection, which may be divided into two types: region growing and region clustering. The difference between the two types is that the region clustering searches for the clusters without prior information while the region growing needs initial points called seeds to detect the clusters. The main problem in region growing approach is to find the suitable seeds since the clusters will grow from the neighbouring pixels of these seeds based on a specified deviation. For seeds selection, Wan and Higgins [5] defined a number of criteria to select insensitive initial points for the subclass of region growing. To reduce the region growing time, Chang and Li [6] proposed a fast region growing method by using parallel segmentation.

As mentioned before, the region clustering approach searches for clusters without prior information. Pappas and Jayant [7] generalized the K-means algorithm to group the

image pixels based on spatial constraints and their intensity variations. These constraints include the following: first, the region intensity to be close to the data and, second, to have a spatial continuity. Their generalization algorithm is adaptive and includes the previous spatial constraints. Like the K-means clustering algorithm, their algorithm is iterative. These spatial constraints are included using Gibbs random field model and they concluded that each region is characterized by a slowly varying intensity function. Because of unimportant features, the algorithm is not that much accurate. Later, to improve their method, they proposed using a caricature of the original image to keep only the most significant features and remove the unimportant features [8]. Besides the problem of time costly segmentation, there are three basic problems that occur during the clustering process which are center redundancy, dead centers, and trapped centers at local minima. Moving K-mean (MKM) proposed in [9] can overcome the three basic problems by minimizing the center redundancy and the dead center as well as reducing indirectly the effect of trapped centers at local minima. In spite of that, MKM is sensitive to noise, centers are not located in the middle or centroid of a group of data, and some members of the centers with the largest fitness are assigned as members of a center with the smallest fitness. To reduce the effects of these problems. Isa et al. [10] proposed three modified versions of the MKM clustering algorithm for image segmentation: fuzzy moving k-means, adaptive moving k-means, and adaptive fuzzy moving k-means. After that Sulaiman and Isa [11] introduced a new segmentation method based on a new clustering algorithm which is Adaptive Fuzzy k-means Clustering Algorithm (AFKM). On the other hand, the Fuzzy C-Means (FCM) algorithm was used in image segmentation but it still has some drawbacks that are the lack of enough robustness to noise and outliers, the crucial parameter α being selected generally based on experience, and the time of segmenting of an image being dependent on its size. To overcome the drawbacks of the FCM algorithm, Cai et al. [12] integrated both local spatial and gray information to propose fast and robust Fuzzy C-means algorithm called Fast Generalized Fuzzy C-Means Algorithm (FGFCM) for image segmentation. Moreover, many researchers have provided a definition for some data characteristics where it has a significant impact on the K-means clustering analysis including the scattering of the data, noise, high-dimensionality, the size of the data and outliers in the data, datasets, types of attributes, and scales of attributes [13]. However, there is a need for more investigation to understand how data distributions affect K-means clustering performance. In [14] Xiong et al. provided a formal and organized study of the effect of skewed data distributions on K-means clustering. Previous research found the impact of high-dimensionality on K-means clustering performance. The traditional Euclidean notion of proximity is not very useful on real-world high-dimensional datasets, such as document datasets and gene-expression datasets. To overcome this challenge, researchers turn to make use of dimensionality reduction techniques, such as multidimensional scaling [15]. Second, K-means has difficulty in detecting the “natural” clusters with nonspherical shapes [13]. Modified K-means

clustering algorithm is a direction to solve this problem. Guha et al. [16] proposed the CURE method, which makes use of multiple representative points to obtain the “natural” clusters shape information. The problem of outliers and noise in the data can also reduce clustering algorithms performance [17], especially for prototype-based algorithms such as K-means. The direction to solve this kind of problem is by combining some outlier removal techniques before conducting K-means clustering. For example, a simple method [9] of detecting outliers is based on the distance measure. On the other hand, many modified K-means clustering algorithms that work well for smaller medium-size datasets are unable to deal with large datasets. Bradley et al. in [18] introduced a discussion of scaling K-means clustering to large datasets.

Arthur and Vassilvitskii implemented a preliminary version, k-means++, in C++ to evaluate the performance of k-means. K-means++ [19] uses a careful seeding method to optimize the speed and the accuracy of k-means. Experiments were performed on four datasets and results showed that that this algorithm is $O(\log k)$ -competitive with the optimal clustering. Experiments also proved its better speed (70% faster) and accuracy (potential value obtained is better by factors of 10 to 1000) than k-means. Kanungo et al. [20] also proposed an $O(n^3 \epsilon^{-d})$ algorithm for the k-means problem. It is $(9 + \epsilon)$ competitive but quite slow in practice. Xiong et al. investigates the measures that best reflects the performance of K-means clustering [14]. An organized study was performed to understand the impact of data distributions on the performance of K-means clustering. Research work also has improved the traditional K-means clustering so that it can handle datasets with large variation of cluster sizes. This formal study illustrates that the entropy sometimes provided misleading information on the clustering performance. Based on this, a coefficient of variation (CV) is proposed to validate the clustering outcome. Experimental results proved that, for datasets with great variation in “true” cluster sizes, K-means lessens (less than 1.0) the variation in resultant cluster sizes. For datasets with small variation in “true” cluster sizes K-means increases (greater than 0.3) the variation in resultant cluster sizes.

Scalability of the k-means for large datasets is one of the major limitations. Chiang et al. [21] proposed a simple and easy to implement algorithm for reducing the computation time of k-means. This pattern reduction algorithm compresses and/or removes iteration patterns that are not likely to change their membership. The pattern is compressed by selecting one of the patterns to be removed and setting its value to the average of all patterns removed. If $D[r_l^i]$ is the pattern to be removed then we can write it as follows:

$$D[r_l^i] = \sum_{j=1}^{|R_l^i|} D[r_{ij}^i], \quad \text{where } r_l^i \in R_l^i. \quad (1)$$

This algorithm can be also applied to other clustering algorithms like population-based clustering algorithms. The computation time of many clustering algorithms can be reduced using this technique, as it is especially effective for large and high-dimensional datasets. In [22] Bradley

et al. proposed Scalable k-means that uses buffering and a two-stage compression scheme to compress or remove patterns in order to improve the performance of k-means. The algorithm is slightly faster than k-means but does not show the same result always. The factors that degrade the performance of scalable k-means are the compression processes and compression ratio. Farnstrom et al. also proposed a simple single pass k-means algorithm [23], which reduces the computation time of k-means. Ordonez and Omiecinski proposed relational k-means [24] that uses the block and incremental concept to improve the stability of scalable k-means. The computation time of k-means can also be reduced using Parallel bisecting k-means [25] and triangle inequality [26] methods.

Although K-means is the most popular and simple clustering algorithm, the major difficulty with this process is that it cannot ensure the global optimum results because the initial cluster centers are selected randomly. Reference [27] presents a novel technique that selects the initial cluster centers by using Voronoi diagram. This method automates the selection of the initial cluster centers. To validate the performance experiments were performed on a range of artificial and real world datasets. The quality of the algorithm is assessed using the following error rate (ER) equation:

$$ER = \frac{\text{Number of misclassified objects}}{\text{Total number of objects}} \times 100\%. \quad (2)$$

The lower the error rate the better the clustering. The proposed algorithm is able to produce better clustering results than the traditional K-means. Experimental results proved 60% reduction in iterations for defining the centroids.

All previous solutions and efforts to increase the performance of K-means algorithm still need more investigation since they are all looking for a local minimum. Searching for the global minimum will certainly improve the performance of the algorithm. The rest of the paper is organized as follows. In Section 2, the proposed method is introduced that finds the optimal centers for each cluster which corresponds to the global minimum of the k-means cluster. In Section 3 the results are discussed and analyzed using two sets from MIT, BioID, and Caltech datasets. Finally in Section 4, conclusions are drawn.

2. K-Means Clustering

K-means, an unsupervised learning algorithm first proposed by MacQueen, 1967 [28], is a popular method of cluster analysis. It is a process of organizing the specified objects into uniform classes called clusters based on similarities among objects based on certain criteria. It solves the well-known clustering problem by considering certain attributes and performing an iterative alternating fitting process. This algorithm partitions a dataset X into k disjoint clusters such that each observation belongs to the cluster with the nearest mean. Let \bar{x}_i be the centroid of cluster c_i and let $d(x_j, \bar{x}_i)$ be the

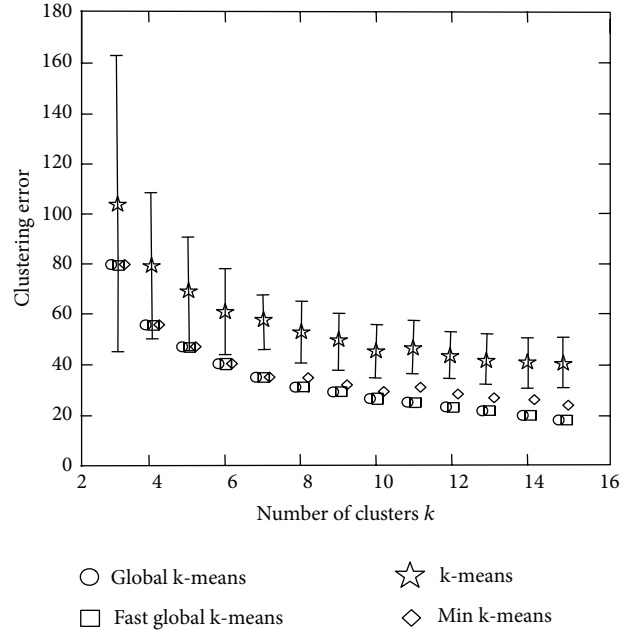


FIGURE 1: Comparison of k-means.

dissimilarity between \bar{x}_i and object $x_j \in c_i$. Then the function minimized by the k-means is given by the following equation:

$$\min_{\bar{x}_1, \dots, \bar{x}_k} E = \sum_{i=1}^k \sum_{x_j \in c_i} d(x_j, \bar{x}_i). \quad (3)$$

K-means clustering is utilized in a vast number of applications including machine learning, fault detection, pattern recognition, image processing, statistics, and artificial intelligent [11, 29, 30]. k-means algorithm is considered as one of the fastest clustering algorithms with a number of variants that are sensitive to the selection of initial points and are intended for solving many issues of k-means like the evaluation of the clusters number [31], the method of initialization of the clusters centroids [32], and the speed of the algorithm [33].

2.1. Modifications in K-Means Clustering. Global k-means algorithm [34] is a proposal to improve global search properties of k-means algorithm and its performance on very large datasets by computing clusters successively. The main weakness of k-means clustering, that is, its sensitivity to initial positions of the cluster centers, is overcome by global k-means clustering algorithm which consists of a deterministic global optimization technique. It does not select initial values randomly; instead an incremental approach is applied to optimally add one new cluster center at each stage. Global k-means also proposes a method to reduce the computational load while maintain the solution quality. Experiments were performed to compare the performance of k-means, global k-means, min k-means, and fast global k-means as shown in Figure 1. Numerical results show that the global k-means algorithm considerably outperforms other k-means algorithms.

Bagirov [35] proposed a new version of the global k-means algorithm for minimum sum-of-squares clustering problems. He also compared three different versions of the k-means algorithm to propose the modified version of the global k-means algorithm. The proposed algorithm computes clusters incrementally and $k - 1$ cluster centers from the previous iteration are used to compute k-partition of a dataset. The starting point of the k th cluster center is computed by minimizing an auxiliary cluster function. Given a finite set of points A in the n -dimensional space the global k-means compute the centroid X^1 of the set A as the following equation:

$$X^1 = \frac{1}{m} \sum_{i=1}^m a^i, \quad a^i \in A, \quad i = 1, \dots, m. \quad (4)$$

Numerical experiments performed on 14 datasets demonstrate the efficiency of the modified global k-means algorithm in comparison to the multistart k-means (MS k-means) and the GKM algorithms when the number of clusters $k > 5$. Modified global k-means however requires more computational time than the global k-means algorithm. Xie and Jiang [36] also proposed a novel version of the global k-means algorithm. The method of creating the next cluster center in the global k-means algorithm is modified and that showed a better execution time without affecting the performance of the global k-means algorithm. Another extension of the standard k-means clustering is the Global Kernel k-means [37] which optimizes the clustering error in feature space by employing kernel-means as a local search procedure. A kernel function is used in order to solve the M-clustering problem and near-optimal solutions are used to optimize the clustering error in the feature space. This incremental algorithm has high ability to identify nonlinearly separable clusters in input space and no dependence on cluster initialization. It can handle weighted data points making it suitable for graph partitioning applications. Two major modifications were performed in order to reduce the computational cost with no major effect on the solution quality.

Video imaging and Image segmentation are important applications for k-means clustering. Hung et al. [38] modified the k-means algorithm for color image segmentation where a weight selection procedure in the W-k-means algorithm is proposed. The evaluation function $F(I)$ is used for comparison:

$$F(I) = \sqrt{R} \sum_{i=1}^R \frac{(e_i/256)^2}{\sqrt{A_i}}, \quad (5)$$

where I = segmented image, R = the number of regions in the segmented image, A_i = the area, or the number of pixels of the i th region, and e_i = the color error of region i .

e_i is the sum of the Euclidean distance of the color vectors between the original image and the segmented image. Smaller values of $F(I)$ give better segmentation results. Results from color image segmentation illustrate that the proposed procedure produces better segmentation than the random initialization. Maliatski and Yadid-Pecht [39] also propose an adaptive k-means clustering, hardware-driven approach.

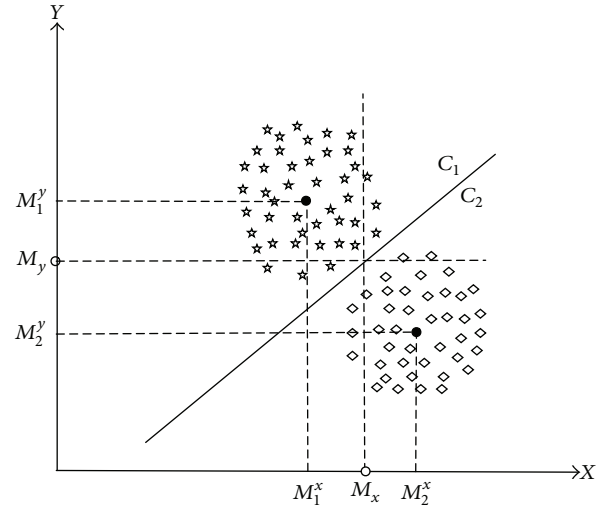


FIGURE 2: The separation points in two dimensions.

The algorithm uses both pixel intensity and pixel distance in the clustering process. In [40] also a FPGA implementation of real-time k-means clustering is done for colored images. A filtering algorithm is used for hardware implementation. Suliman and Isa [11] presented a unique clustering algorithm called adaptive fuzzy-K-means clustering for image segmentation. It can be used for different types of images including medical images, microscopic images, and CCD camera images. The concepts of fuzziness and belongingness are applied to produce more adaptive clustering as compared to other conventional clustering algorithms. The proposed adaptive fuzzy-K-means clustering algorithm provides better segmentation and visual quality for a range of images.

2.2. The Proposed Methodology. In this method, the face location is determined automatically by using an optimized K-means algorithm. First, the input image is reshaped into vectors of pixels values, followed by clustering the pixels, based on applying a certain threshold determined by the algorithm, into two classes. Pixels with values below the threshold are put in the nonface class. The rest of the pixels are put in the face class. However, some unwanted parts may get clustered to this face class. Therefore, the algorithm is applied again to the face class to obtain only the face part.

2.2.1. Optimized K-Means Algorithm. The proposed K-means modified algorithm is intended to solve the limitations of the standard version by using differential equations to determine the optimum separation point. This algorithm finds the optimal centers for each cluster which corresponds to the global minimum of the k-means cluster. As an example, say we would like to cluster an image into two subclasses: face and nonface. So we look for a separator which will separate the image into two different clusters in two dimensions. Then the means of the clusters can be found easily. Figure 2 shows the separation points and the mean of each class. To find these

points we proposed the following modification in continuous and discrete cases.

Let $\{x_1, x_2, x_3, \dots, x_N\}$ be a dataset and let x_i be an observation. Let $f(x)$ be the probability mass function (PMF) of x

$$f(x) = \sum_{-\infty}^{\infty} \frac{1}{N} \delta(x - x_i), \quad x_i \in \mathbb{R}, \quad (6)$$

where $\delta(x)$ is the Dirac function.

When the size of the data is very large, the question of the cost of the minimization of $\sum_{i=1}^k \sum_{x_i \in s_i} |x_i - \mu_i|^2$ arises, where k is the number of clusters and μ_i is the mean of cluster s_i .

Let

$$\text{Var}_k = \sum_{i=1}^k \sum_{x_i \in s_i} |x_i - \mu_i|^2. \quad (7)$$

Then it is enough to minimize Var_1 without losing the generality since

$$\min_{s_i} \sum_{i=1}^d \sum_{x_j \in s_i} |x_j - \mu_i|^2 \geq \sum_{i=1}^d \min_{s_i} \left(\sum_{j=1}^k \sum_{x_j \in s_i} |x_j^i - \mu_i^i|^2 \right), \quad (8)$$

where d is the dimension of $x_j = (x_j^1, \dots, x_j^d)$ and $\mu_j = (\mu_j^1, \dots, \mu_j^d)$.

2.2.2. Continuous Case. For the continuous case $f(x)$ is like the probability density function (PDF). Then for 2 classes, we need to minimize the following equation:

$$\begin{aligned} \text{Var}_2 = & \int_{-\infty}^M (x - \mu_1)^2 f(x) dx \\ & + \int_M^{\infty} (x - \mu_2)^2 f(x) dx. \end{aligned} \quad (9)$$

If $d \geq 2$ we can use (8)

$$\begin{aligned} \min \text{Var}_2 \geq & \sum_{k=1}^d \min \left(\int_{-\infty}^M (x^k - \mu_1^k)^2 f(x^k) dx \right. \\ & \left. + \int_M^{\infty} (x^k - \mu_2^k)^2 f(x^k) dx \right). \end{aligned} \quad (10)$$

Let $x \in \mathbb{R}$ be any random variable with probability density function $f(x)$; we want to find μ_1 and μ_2 such that it minimizes (7) as follows:

$$(\mu_1^*, \mu_2^*) = \arg \left[\min_{(\mu_1, \mu_2)} \left[\int_{-\infty}^M (x - \mu_1)^2 f(x) dx + \int_M^{\infty} (x - \mu_2)^2 f(x) dx \right] \right], \quad (11)$$

$$\begin{aligned} \min_{M, \mu_1, \mu_2} \text{Var}_2 = & \min_M \left[\min_{\mu_1} \int_{-\infty}^M (x - \mu_1)^2 f(x) dx \right. \\ & \left. + \min_{\mu_2} \int_M^{\infty} (x - \mu_2)^2 f(x) dx \right], \end{aligned} \quad (12)$$

$$\begin{aligned} \min_{M, \mu_1, \mu_2} \text{Var}_2 = & \min_M \left[\min_{\mu_1} \int_{-\infty}^M (x - \mu_1(M))^2 f(x) dx \right. \\ & \left. + \min_{\mu_2} \int_M^{\infty} (x - \mu_2(M))^2 f(x) dx \right]. \end{aligned} \quad (13)$$

We know that

$$\min_x E[(X - x)^2] = E[(X - E(x))^2] = \text{Var}(x). \quad (14)$$

So we can find μ_1 and μ_2 in terms of M . Let us define Var_1 and Var_2 as follows:

$$\begin{aligned} \text{Var}_1 &= \int_{-\infty}^M (x - \mu_1)^2 f(x) dx, \\ \text{Var}_2 &= \int_M^{\infty} (x - \mu_2)^2 f(x) dx. \end{aligned} \quad (15)$$

After simplification we get

$$\begin{aligned} \text{Var}_1 &= \int_{-\infty}^M x^2 f(x) dx \\ &+ \mu_1^2 \int_{-\infty}^M f(x) dx - 2\mu_1 \int_{-\infty}^M xf(x) dx, \\ \text{Var}_2 &= \int_M^{\infty} x^2 f(x) dx \\ &+ \mu_2^2 \int_M^{\infty} f(x) dx - 2\mu_2 \int_M^{\infty} xf(x) dx. \end{aligned} \quad (16)$$

To find the minimum, Var_1 and Var_2 are both partially differentiated with respect to μ_1 and μ_2 , respectively. After simplification, we conclude that the minimization occurs when

$$\begin{aligned} \mu_1 &= \frac{E(X_-)}{P(X_-)}, \\ \mu_2 &= \frac{E(X_+)}{P(X_+)}, \end{aligned} \quad (17)$$

where $X_- = X \cdot 1_{x < M}$ and $X_+ = X \cdot 1_{x \geq M}$.

To find the minimum of Var₂, we need to find $\partial\mu/\partial M$

$$\begin{aligned}\mu_1 &= \left(\frac{\int_{-\infty}^M x f(x) dx}{\int_{-\infty}^M f(x) dx} \right) \\ \Rightarrow \frac{\partial\mu_1}{\partial M} &= \frac{M f(M) P(X_-) - f(M) E(X_-)}{P^2(X_-)},\end{aligned}\quad (18)$$

$$\begin{aligned}\mu_2 &= \left(\frac{\int_M^{\infty} x f(x) dx}{\int_M^{\infty} f(x) dx} \right) \\ \Rightarrow \frac{\partial\mu_2}{\partial M} &= \frac{M f(M) P(X_+) - f(M) E(X_+)}{P^2(X_+)}.\end{aligned}$$

After simplification we get

$$\begin{aligned}\frac{\partial\mu_1}{\partial M} &= \frac{f(M)}{P^2(X_-)} [M P(X_-) - E(X_-)], \\ \frac{\partial\mu_2}{\partial M} &= \frac{f(M)}{P^2(X_+)} [M P(X_+) - E(X_+)].\end{aligned}\quad (19)$$

In order to find the minimum of Var₂, we will find $\partial\text{Var}_1/\partial M$ and $\partial\text{Var}_2/\partial M$ separately and add them as follows:

$$\begin{aligned}\frac{\partial}{\partial M} (\text{Var}_2) &= \frac{\partial}{\partial M} (\text{Var}_1) + \frac{\partial}{\partial M} (\text{Var}_2) \\ \frac{\partial\text{Var}_1}{\partial M} &= \frac{\partial}{\partial M} \left(\int_{-\infty}^M x^2 f(x) dx \right. \\ &\quad \left. + \mu_1^2 \int_{-\infty}^M f(x) dx - 2\mu_1 \int_{-\infty}^M x f(x) dx \right) \\ &= M^2 f(M) + 2\mu_1 \mu_1' P(X_-) \\ &\quad + \mu_1^2 f(M) - 2\mu_1' E(X_-) - 2\mu_1 M f(M).\end{aligned}\quad (20)$$

After simplification we get

$$\frac{\partial\text{Var}_1}{\partial M} = f(M) \left[M^2 + \frac{E^2(X_-)}{P(X_-)^2} - 2M \frac{E(X_-)}{P(X_-)} \right]. \quad (21)$$

Similarly

$$\begin{aligned}\frac{\partial\text{Var}_2}{\partial M} &= -M^2 f(M) + 2\mu_2 \mu_2' P(X_+) \\ &\quad - \mu_2^2 f(M) - 2\mu_2' E(X_+) + 2\mu_2 M f(M).\end{aligned}\quad (22)$$

After simplification we get

$$\frac{\partial\text{Var}_2}{\partial M} = f(M) \left[-M^2 + \frac{E^2(X_+)}{P(X_+)^2} - 2M \frac{E(X_+)}{P(X_+)} \right]. \quad (23)$$

Adding both these differentials and after simplifying, it follows that

$$\begin{aligned}\frac{\partial}{\partial M} (\text{Var}_2) &= f(M) \left[\left(\frac{E^2(X_-)}{P(X_-)^2} - \frac{E^2(X_+)}{P(X_+)^2} \right) \right. \\ &\quad \left. - 2M \left(\frac{E(X_-)}{P(X_-)} - \frac{E(X_+)}{P(X_+)} \right) \right].\end{aligned}\quad (24)$$

Equating this with 0 gives

$$\begin{aligned}\frac{\partial}{\partial M} (\text{Var}_2) &= 0, \\ \left[\left(\frac{E^2(X_-)}{P(X_-)^2} - \frac{E^2(X_+)}{P(X_+)^2} \right) - 2M \left(\frac{E(X_-)}{P(X_-)} - \frac{E(X_+)}{P(X_+)} \right) \right] &= 0, \\ \left[\frac{E(X_-)}{P(X_-)} - \frac{E(X_+)}{P(X_+)} \right] \left[\frac{E(X_-)}{P(X_-)} + \frac{E(X_+)}{P(X_+)} - 2M \right] &= 0.\end{aligned}\quad (25)$$

But $[(E(X_-)/P(X_-)) - (E(X_+)/P(X_+))] \neq 0$, as in that case $\mu_1 = \mu_2$, which contradicts the initial assumption. Therefore

$$\begin{aligned}\left[\frac{E(X_-)}{P(X_-)} + \frac{E(X_+)}{P(X_+)} - 2M \right] &= 0 \\ \Rightarrow M &= \left[\frac{E(X_-)}{P(X_-)} + \frac{E(X_+)}{P(X_+)} \right] \cdot \frac{1}{2}, \\ M &= \frac{\mu_1 + \mu_2}{2}.\end{aligned}\quad (26)$$

Therefore, we have to find all values of M that satisfy (26), which is easier than minimizing Var₂ directly.

To find the minimum in terms of M , it is enough to derive

$$\begin{aligned}\frac{\partial\text{Var}_2(\mu_1(M), \mu_2(M))}{\partial M} &= f(M) \left[\frac{E^2(X_-)}{P(X < M)^2} - \frac{E^2(X_+)}{P(X \geq M)^2} \right. \\ &\quad \left. - 2M \left[\frac{E(X_-)}{P(X \leq M)} - \frac{E(X_+)}{P(X > M)} \right] \right].\end{aligned}\quad (27)$$

Then we find

$$M = \left[\frac{E(X_+)}{P(X_+)} + \frac{E(X_-)}{P(X_-)} \right] \cdot \frac{1}{2}. \quad (28)$$

2.2.3. Finding the Centers of Some Common Probability Density Functions

(1) *Uniform Distribution.* The probability density function of the continuous uniform distribution is

$$f(x) = \begin{cases} \frac{1}{b-a}, & a < x < b \\ 0 & \text{otherwise,} \end{cases} \quad (29)$$

where a and b are the two parameters of the distribution. The expected values for $X \cdot 1_x \geq M$ and $X \cdot 1_x < M$ are

$$\begin{aligned} E(X_+) &= \frac{1}{2} \left(\frac{b^2 - M^2}{b - a} \right), \\ E(X_-) &= \frac{1}{2} \left(\frac{M^2 - a^2}{b - a} \right). \end{aligned} \quad (30)$$

Putting all these in (28) and solving, we get

$$\begin{aligned} M &= \frac{1}{2} (a + b), \\ \mu_1(M) &= \frac{E(X_-)}{P(X_-)} = \frac{1}{4} (3a + b), \\ \mu_2(M) &= \frac{E(X_+)}{P(X_+)} = \frac{1}{4} (a + 3b). \end{aligned} \quad (31)$$

(2) *Log-Normal Distribution*. The probability density function of a log-normal distribution is

$$f(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-(\ln x - \mu)^2 / 2\sigma^2}, \quad x > 0. \quad (32)$$

The probabilities in left and right of M are, respectively,

$$\begin{aligned} P(X_+) &= \frac{1}{2} - \frac{1}{2} \operatorname{erf} \left(\frac{1}{2} \frac{\sqrt{2} (\ln M - \mu)}{\sigma} \right), \\ P(X_-) &= \frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{1}{2} \frac{\sqrt{2} (\ln M - \mu)}{\sigma} \right). \end{aligned} \quad (33)$$

The expected values for $X \cdot 1_x \geq M$ and $X \cdot 1_x < M$ are

$$\begin{aligned} E(X_+) &= \left[\frac{1}{2} e^{\mu + (1/2)\sigma^2} \left\{ 1 + \operatorname{erf} \left(\frac{1}{2} \frac{(\mu + \sigma^2 - \ln M) \sqrt{2}}{\sigma} \right) \right\} \right], \\ E(X_-) &= \left[\frac{1}{2} e^{\mu + (1/2)\sigma^2} \left\{ 1 + \operatorname{erf} \left(\frac{1}{2} \frac{(-\mu - \sigma^2 + \ln M) \sqrt{2}}{\sigma} \right) \right\} \right]. \end{aligned} \quad (34)$$

Putting all these in (28) and assuming $\mu = 0$ and $\sigma = 1$ and solving, we get

$$\begin{aligned} M &= 4.244942583, \\ \mu_1(M) &= \frac{E(X_-)}{P(X_-)} = 1.196827816, \\ \mu_2(M) &= \frac{E(X_+)}{P(X_+)} = 7.293057348. \end{aligned} \quad (35)$$

(3) *Normal Distribution*. The probability density function of a normal distribution is

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 / 2\sigma^2}. \quad (36)$$

The probabilities in left and right of M are, respectively,

$$\begin{aligned} P(X_+) &= \frac{1}{2} - \frac{1}{2} \operatorname{erf} \left(\frac{1}{2} \frac{\sqrt{2} (M - \mu)}{\sigma} \right), \\ P(X_-) &= \frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{1}{2} \frac{\sqrt{2} (M - \mu)}{\sigma} \right). \end{aligned} \quad (37)$$

The expected values for $X \cdot 1_x \geq M$ and $X \cdot 1_x < M$ are

$$\begin{aligned} E(X_+) &= -\frac{1}{2} \mu \left\{ -1 + \operatorname{erf} \left(\frac{1}{2} \frac{\sqrt{2} (M - \mu)}{\sigma} \right) \right\}, \\ E(X_-) &= \frac{1}{2} \mu \left\{ 1 + \operatorname{erf} \left(\frac{1}{2} \frac{\sqrt{2} (M - \mu)}{\sigma} \right) \right\}. \end{aligned} \quad (38)$$

Putting all these in (28) and solving we get $M = \mu$. Assuming $\mu = 0$ and $\sigma = 1$ and solving, we get

$$\begin{aligned} M &= 0, \\ \mu_1(M) &= \frac{E(X_-)}{P(X_-)} = -\sqrt{\frac{2}{\pi}}, \\ \mu_2(M) &= \frac{E(X_+)}{P(X_+)} = \sqrt{\frac{2}{\pi}}. \end{aligned} \quad (39)$$

2.2.4. *Discrete Case*. Let X be a discrete random variable and assume we have a set of N observations from X . Let p_j be the mass density function for an observation x_j .

Let $x_{i-1} < M_{i-1} < x_i$ and $x_i < M_i < x_{i+1}$. We define $\mu_{x \leq x_i}$ as the mean for all $x \leq x_i$, and $\mu_{x \geq x_i}$ as the mean for all $x \geq x_i$. Define $\operatorname{Var}_2(M_{i-1})$ as the variance of two centers forced by M_{i-1} as a separator

$$\begin{aligned} \operatorname{Var}(M_{i-1}) &= \sum_{x_j \leq x_{i-1}} (x_j - \mu_{x_j \leq x_{i-1}})^2 p_j \\ &\quad + \sum_{x_j \geq x_i} (x_j - \mu_{x_j \geq x_i})^2 p_j, \end{aligned} \quad (40)$$

$$\begin{aligned} \operatorname{Var}(M_i) &= \sum_{x_j \leq x_i} (x_j - \mu_{x_j \leq x_i})^2 p_j \\ &\quad + \sum_{x_j \geq x_{i+1}} (x_j - \mu_{x_j \geq x_{i+1}})^2 p_j. \end{aligned} \quad (41)$$

The first part of $\text{Var}(M_{i-1})$ simplifies as follows:

$$\begin{aligned}
 & \sum_{x_j \leq x_{i-1}} (x_j - \mu_{x_j \leq x_{i-1}})^2 p_j \\
 &= \sum_{x_j \leq x_{i-1}} x_j^2 p_j - 2\mu_{x_j \leq x_{i-1}} \sum_{x_j \leq x_{i-1}} x_j p_j + \mu_{x_j \leq x_{i-1}}^2 \sum_{x_j \leq x_{i-1}} p_j \\
 &= E(X^2 \cdot 1_{x_j \leq x_{i-1}}) - \frac{E^2(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})}.
 \end{aligned} \quad (42)$$

Similarly,

$$\begin{aligned}
 \sum_{x_j \geq x_i} (x_j - \mu_{x_j \geq x_i})^2 p_j &= E(X^2 \cdot 1_{x_j \geq x_i}) - \frac{E^2(X \cdot 1_{x_j \geq x_i})}{P(X \geq x_i)}, \\
 \text{Var}_2(M_{i-1}) &= E(X^2 \cdot 1_{x_j \leq x_{i-1}}) \\
 &\quad - \frac{E^2(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})} + E(X^2 \cdot 1_{x_j \geq x_i}) \\
 &\quad - \frac{E^2(X \cdot 1_{x_j \geq x_i})}{P(X \geq x_i)}, \\
 \text{Var}_2(M_i) &= E(X^2 \cdot 1_{x_j \leq x_i}) \\
 &\quad - \frac{E^2(X \cdot 1_{x_j \leq x_i})}{P(X \leq x_i)} + E(X^2 \cdot 1_{x_j \geq x_{i+1}}) \\
 &\quad - \frac{E^2(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})}.
 \end{aligned} \quad (43)$$

Define $\Delta \text{Var}_2 = \text{Var}_2(M_i) - \text{Var}_2(M_{i-1})$

$$\begin{aligned}
 \Delta \text{Var}_2 &= \left[-\frac{E^2(X \cdot 1_{x_j \leq x_i})}{P(X \leq x_i)} - \frac{E^2(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})} \right] \\
 &\quad - \left[-\frac{E^2(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})} - \frac{E^2(X \cdot 1_{x_j \geq x_i})}{P(X \geq x_i)} \right].
 \end{aligned} \quad (44)$$

Now, in order to simplify the calculation, we rearrange ΔVar_2 as follows:

$$\begin{aligned}
 \Delta \text{Var}_2 &= \left[\frac{E^2(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})} - \frac{E^2(X \cdot 1_{x_j \leq x_i})}{P(X \leq x_i)} \right] \\
 &\quad + \left[\frac{E^2(X \cdot 1_{x_j \geq x_i})}{P(X \geq x_i)} - \frac{E^2(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})} \right].
 \end{aligned} \quad (45)$$

The first part is simplified as follows:

$$\begin{aligned}
 & \frac{E^2(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})} - \frac{E^2(X \cdot 1_{x_j \leq x_i})}{P(X \leq x_i)} \\
 &= \frac{E^2(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})} - \frac{[E(X \cdot 1_{x_j \leq x_{i-1}}) + x_i p_i]^2}{P(X \leq x_{i-1}) + p_i} \\
 &= \frac{1}{P(X \leq x_{i-1}) [P(X \leq x_{i-1}) + p_i]} \\
 &\quad \times \{ E^2(X \cdot 1_{x_j \leq x_{i-1}}) P(X \leq x_i) \\
 &\quad + p_i E^2(X \cdot 1_{x_j \leq x_{i-1}}) \\
 &\quad - E^2(X \cdot 1_{x_j \leq x_{i-1}}) P(X \leq x_{i-1}) \\
 &\quad - 2x_i p_i E(X \cdot 1_{x_j \leq x_{i-1}}) P(X \leq x_{i-1}) \\
 &\quad - x_i^2 p_i^2 P(X \leq x_{i-1}) \} \\
 &= \frac{1}{P(X \leq x_{i-1}) [P(X \leq x_{i-1}) + p_i]} \\
 &\quad \times \{ p_i E^2(X \cdot 1_{x_j \leq x_{i-1}}) \\
 &\quad - 2x_i p_i E(X \cdot 1_{x_j \leq x_{i-1}}) P(X \leq x_{i-1}) \\
 &\quad - x_i^2 p_i^2 P(X \leq x_{i-1}) \}.
 \end{aligned} \quad (46)$$

Similarly, simplifying the second part yields

$$\begin{aligned}
 & \frac{E^2(X \cdot 1_{x_j \geq x_i})}{P(X \geq x_i)} - \frac{E^2(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})} \\
 &= \frac{[E(X \cdot 1_{x_j \geq x_{i+1}}) + x_i p_i]^2}{P(X \geq x_{i+1}) + p_i} - \frac{E^2(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})} \\
 &= \frac{1}{P(X \geq x_{i+1}) [P(X \geq x_{i+1}) + p_i]} \\
 &\quad \times \{ E^2(X \cdot 1_{x_j \geq x_{i+1}}) P(X \geq x_{i+1}) \\
 &\quad - p_i E^2(X \cdot 1_{x_j \geq x_{i+1}}) \\
 &\quad - E^2(X \cdot 1_{x_j \geq x_{i+1}}) P(X \geq x_{i+1}) \\
 &\quad + 2x_i p_i E(X \cdot 1_{x_j \geq x_{i+1}}) P(X \geq x_{i+1}) \\
 &\quad + x_i^2 p_i^2 P(X \geq x_{i+1}) \}
 \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{P(X \geq x_{i+1}) [P(X \geq x_{i+1}) + p_i]} \\
&\times \left\{ -p_{i+1} E^2(X \cdot 1_{x_j \geq x_{i+1}}) \right. \\
&\quad + 2x_i p_i E(X \cdot 1_{x_j \geq x_{i+1}}) P(X \geq x_{i+1}) \\
&\quad \left. + x_i^2 p_{i+1}^2 P(X \geq x_{i+1}) \right\}. \tag{47}
\end{aligned}$$

In general, these types of optimization arise when the dataset is very big.

If we consider N to be very large then $p_k \ll \max(\sum_{i \geq k+1} P_i; \sum_{i \leq k-1} P_i)$, $P(X \leq x_{i-1}) + p_i \approx P(X \leq x_i)$, and $P(X \geq x_{i+1}) + p_i \approx P(X \geq x_{i+1})$:

$$\begin{aligned}
\Delta \text{Var}_2 \approx & \left[\frac{E^2(X \cdot 1_{x_j \leq x_{i-1}}) p_i}{P^2(X \leq x_{i-1})} - \frac{E^2(X \cdot 1_{x_j \geq x_{i+1}}) p_i}{P^2(X \geq x_{i+1})} \right] \\
& - 2p_i x_i \left[\frac{E(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})} - \frac{E(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})} \right] \\
& - p_i^2 x_i^2 \left[\frac{1}{P(X \leq x_{i-1})} - \frac{1}{P(X \geq x_{i+1})} \right] \tag{48}
\end{aligned}$$

and, as $\lim_{N \rightarrow \infty} \{p_i^2 x_i^2 [(1/P(X \leq x_{i-1})) - (1/P(X \geq x_{i+1}))]\} = 0$, we can further simplify ΔVar_2 as follows:

$$\begin{aligned}
\Delta \text{Var}_2 & \approx \left[\frac{E^2(X \cdot 1_{x_j \leq x_{i-1}}) p_i}{P^2(X \leq x_{i-1})} - \frac{E^2(X \cdot 1_{x_j \geq x_{i+1}}) p_i}{P^2(X \geq x_{i+1})} \right] \\
& - 2p_i x_i \left[\frac{E(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})} - \frac{E(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})} \right] \\
& = p_i \left[\frac{E(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})} - \frac{E(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})} \right] \\
& \times \left[\left(\frac{E(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})} + \frac{E(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})} \right) - 2x_i \right]. \tag{49}
\end{aligned}$$

To find the minimum of ΔVar_2 , let us locate when $\Delta \text{Var}_2 = 0$:

$$\begin{aligned}
\Rightarrow p_i & \left[\frac{E(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})} - \frac{E(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})} \right] \\
& \times \left[\left(\frac{E(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})} + \frac{E(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})} \right) - 2x_i \right] = 0. \tag{50}
\end{aligned}$$

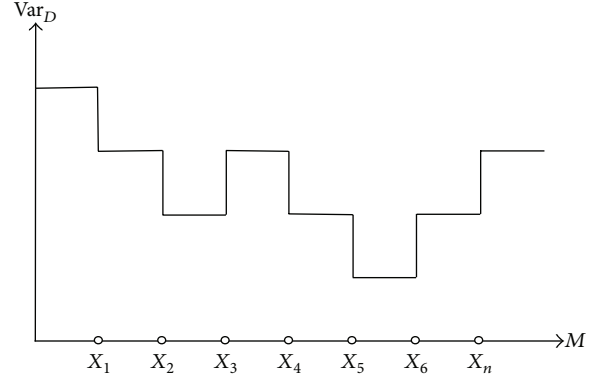


FIGURE 3: Discrete random variables.

But $p_i \neq 0$, and so is $[(E(X \cdot 1_{x_j \leq x_{i-1}}))/P(X \leq x_{i-1})] - (E(X \cdot 1_{x_j \geq x_{i+1}}))/P(X \geq x_{i+1}) < 0$, since x_i is an increasing sequence, which implies that $\mu_{i-1} < \mu_i$ where

$$\begin{aligned}
\mu_{i-1} &= \frac{E(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})}, \\
\mu_i &= \frac{E(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})}. \tag{51}
\end{aligned}$$

Therefore,

$$\left[\frac{E(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})} + \frac{E(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})} \right] - 2x_i = 0. \tag{52}$$

Let us define

$$\begin{aligned}
F(x_i) &= 2x_i - \left[\frac{E(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})} + \frac{E(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})} \right] \\
F(x_i) &= 0 \\
\Rightarrow x_i &= \left[\frac{E(X \cdot 1_{x_j \leq x_{i-1}})}{P(X \leq x_{i-1})} + \frac{E(X \cdot 1_{x_j \geq x_{i+1}})}{P(X \geq x_{i+1})} \right] \cdot \frac{1}{2} \\
\Rightarrow x_i &= \frac{\mu_{i-1} + \mu_i}{2}, \tag{53}
\end{aligned}$$

where μ_{i-1} and μ_i are the means of the first and second parts, respectively.

To find the minimum for the total variation, it is enough to find the good separator M between x_i and x_{i-1} such that $F(x_i) \cdot F(x_{i+1}) < 0$ and $F(x_i) < 0$ (see Figure 3).

After finding few local minima, we can get the global minimum by finding the smallest among them.

3. Face Localization Using Proposed Method

Now, it becomes clear the superiority of the K-means modified algorithm over the standard algorithm in terms of finding



FIGURE 4: The original image.



FIGURE 6: The face without illumination.



FIGURE 5: The face with shade only.



FIGURE 7: The filtered image.

a global minimum and we can explain our solution based on the modified algorithm. Suppose we have an image with pixels values from 0 to 255. First, we reshaped the input image into a vector of pixels; Figure 4 shows an input image before the operation of the K-means modified algorithm. The image contains a face part, a normal background, and a background with illumination. Then we split the image pixels into two classes: one from 0 to x_1 and another from x_1 to 255 by applying a certain threshold determined by the algorithm.

Let $[0, \dots, x_1]$ be the class 1-which represents the nonface part and let $[x_1, \dots, 255]$ be the class 2-which represents the face with the shade.

All pixels with values less than the threshold belong to the first class and will be assigned to 0 values. This will keep the pixels with values above the threshold and these pixels belong to both the face part and the illuminated part of the background, as shown in Figure 5.

In order to remove the illuminated background part, another threshold will be applied to the pixels using the algorithm meant to separate between the face part and the illuminated background part. New classes will be obtained, from x_1 to x_2 and from x_2 to 255. Then, the pixels with values less than the threshold will be assigned a value of 0 and the nonzero pixels left represent the face part. Let $[x_1, \dots, x_2]$ be the class 1 which represents the illuminated background part and let $[x_2, \dots, 255]$ be the class 2 which represents the face part.

The result of the removal of the illuminated back-ground part is shown in Figure 6.

One can see that some face pixels were assigned to 0 due to the effects of the illumination. Therefore, there is a need to return to the original image in Figure 4 to fully extract the image window corresponding to the face. A filtering process is then performed to reduce the noise and the result is shown in Figure 7.

Figure 8 shows the final result which is a window containing the face extracted from the original image.

4. Results

In this Section, the experimental results of the proposed algorithm will be analyzed and discussed. First, it starts with presenting a description of the datasets used in this work, followed by the results of the previous part of the work. Three popular databases were used to test the proposed face localization method. The MIT-CBCL database has image size 115×115 , and 300 images were selected from this dataset based on different conditions such as light variation, pose, and background variations. The second dataset is the BioID dataset with image size 384×286 and 300 images from the database were selected based on different conditions such as indoor/outdoor. The last dataset is the Caltech dataset with image size 896×592 and the 300 images from the database were selected based on different conditions such as indoor/outdoor.

4.1. MIT-CBCL Dataset. This dataset was established by the Center for Biological and Computational Learning (CBCL)



FIGURE 8: The corresponding window in the original image.



FIGURE 9: Samples from MIT-CBCL dataset.

at Massachusetts Institute of Technology (MIT) [41]. It has 10 persons with 200 images per person and an image size of 115×115 pixels. The images of each person are with different light conditions, clutter background, and scale and different poses. Few examples of these images are shown in Figure 9.

4.2. BioID Dataset. This dataset [42] consists of 1521 gray level images with a resolution of 384×286 pixels. Each one shows the frontal view of a face of one out of 23 different test persons. The images were taken under different lighting conditions, with a complex background, and contain tilted and rotated faces. This dataset is considered as the most difficult one for eye detection (see Figure 10).

4.3. Caltech Dataset. This database [43] has been collected by Markus Weber at California Institute of Technology. The database contains 450 face images of 27 subjects under different lighting conditions, different face expressions, and complex backgrounds (see Figure 11).

4.4. Proposed Method Result. The proposed method is evaluated on these datasets where the result was 100% with a

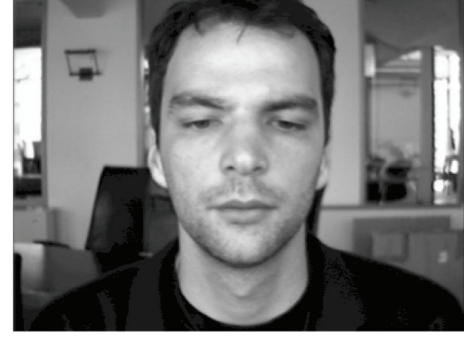


FIGURE 10: Sample of faces from BioID dataset for localization.



FIGURE 11: Sample of faces from Caltech dataset for localization.

TABLE 1: Face localization using K-mean modified algorithm on MIT-CBCL, BioID, and Caltech; the sets contain faces with different poses and faces with clutter background as well as indoors/outdoors.

Dataset	Image	Accuracy (%)
MIT-Set 1	150	100
MIT-Set 2	150	100
BioID	300	100
Caltech	300	100

localization time of 4 s. Table 1 shows the proposed method results with the MIT-CBCL, Caltech, and BioID databases. In this test, we have focused on the use of images of faces from different angles from 30 degrees left to 30 degrees right and the variation in the background as well as the cases of indoor and outdoor images. Two sets of images are selected from the MIT-CBCL dataset and two other sets from the other databases to evaluate the proposed method. The first set from MIT-CBCL contains the images with different poses and the second set contains images with different backgrounds. Each set contains 150 images. The BioID and Caltech sets contain images with indoor and outdoor settings. The results showed the efficiency of the proposed K-mean modified algorithm to locate the faces from the image with high background and poses changes. In addition, another parameter was considered in this test which is the localization time. From the results, the proposed method can locate the face in a very short time because it has less complexity due to the use of differential equations.

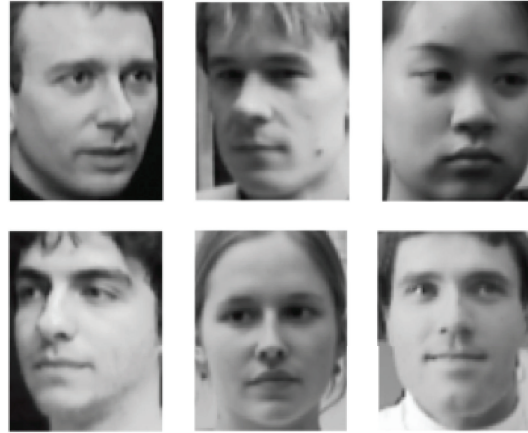


FIGURE 12: Examples of face localization using k-mean modified algorithm on MIT-CBCL dataset.

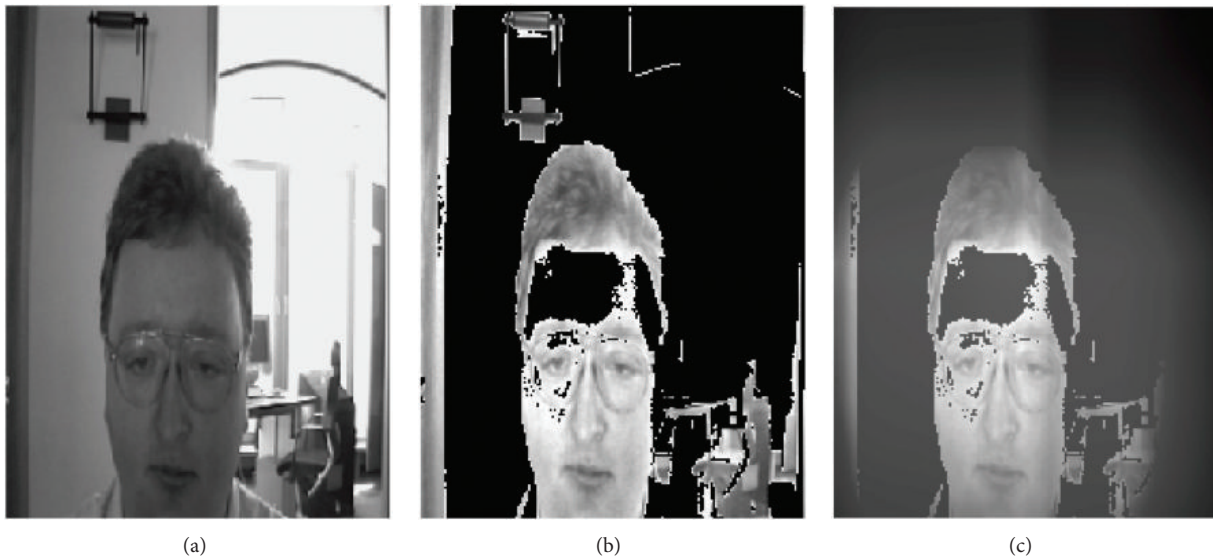


FIGURE 13: Example of face localization using k-mean modified algorithm on BioID dataset: (a) the original image, (b) face position, and (c) filtered image.

Figure 12 shows some examples of face localization on MIT-CBCL.

In Figure 13(a), an original image from BioID dataset is shown. The localization position is shown in Figure 13(b). But there is need to do a filtering operation in order to remove the unwanted parts and this is shown in Figure 13(c) and then the area of ROI will be determined.

Figure 14 shows the face position on an image from the Caltech dataset.

5. Conclusion

In this paper, we focus on developing the RIO detection by suggesting a new method for face localization. In this method of localization, the input image is reshaped into a vector and then the optimized k-means algorithm is applied twice to cluster the image pixels into classes. At the

end, the block window corresponding to the class of pixels containing the face only is extracted from the input image. Three sets of images taken from MIT, BioID, and Caltech datasets and differing in illumination, and backgrounds, as well as in-door/outdoor settings were used to evaluate the proposed method. The results show that the proposed method achieved significant localization accuracy. Our future research direction includes finding the optimal centers for higher dimension data, which we believe is just generalization of our approach to higher dimension (8) and to higher subclusters.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

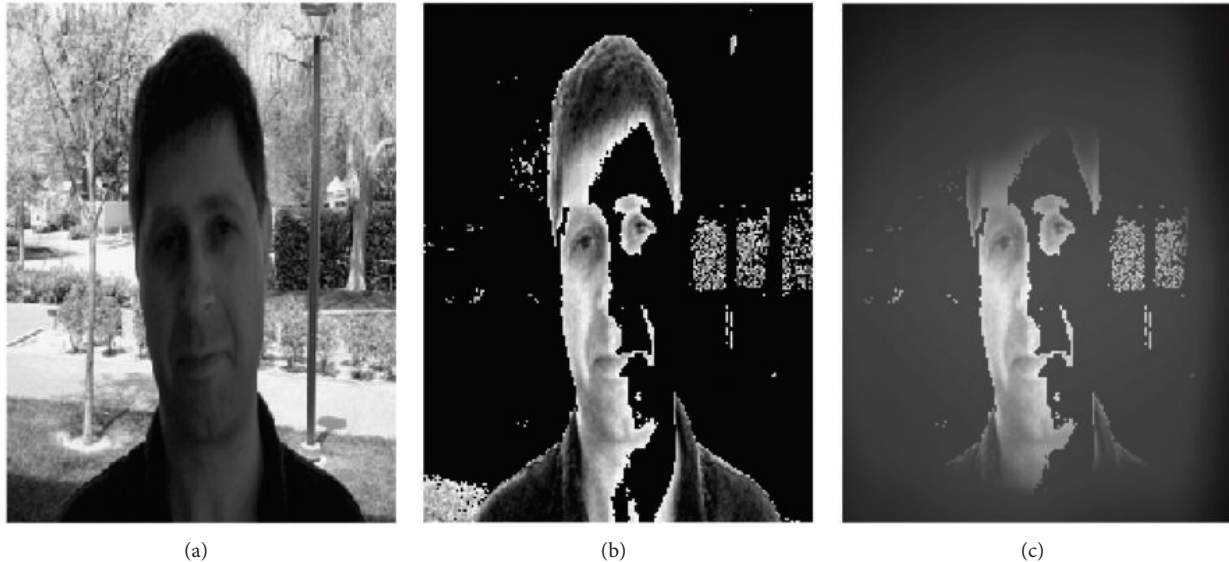


FIGURE 14: Example of face localization using k-mean modified algorithm on Caltech dataset: (a) the original image, (b) face position, and (c) filtered image.

Acknowledgments

This study is a part of the Internal Grant Project no. 419011811122. The authors gratefully acknowledge the continued support from Alfaisal University and its Office of Research.

References

- [1] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: a survey," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 705–740, 1995.
- [2] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, 2002.
- [3] K. S. Fu and J. K. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, no. 1, pp. 3–16, 1981.
- [4] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.
- [5] S.-Y. Wan and W. E. Higgins, "Symmetric region growing," *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1007–1015, 2003.
- [6] Y.-L. Chang and X. Li, "Fast image region growing," *Image and Vision Computing*, vol. 13, no. 7, pp. 559–571, 1995.
- [7] T. N. Pappas and N. S. Jayant, "An adaptive clustering algorithm for image segmentation," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '89)*, vol. 3, pp. 1667–1670, May 1989.
- [8] T. N. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 901–914, 1992.
- [9] M. Y. Mashor, "Hybrid training algorithm for RBF network," *International Journal of the Computer, the Internet and Management*, vol. 8, no. 2, pp. 50–65, 2000.
- [10] N. A. M. Isa, S. A. Salamah, and U. K. Ngah, "Adaptive fuzzy moving K-means clustering algorithm for image segmentation," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 2145–2153, 2009.
- [11] S. N. Sulaiman and N. A. M. Isa, "Adaptive fuzzy-K-means clustering algorithm for image segmentation," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2661–2668, 2010.
- [12] W. Cai, S. Chen, and D. Zhang, "Fast and robust fuzzy *c*-means clustering algorithms incorporating local information for image segmentation," *Pattern Recognition*, vol. 40, no. 3, pp. 825–838, 2007.
- [13] P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Pearson Addison-Wesley, 2006.
- [14] H. Xiong, J. Wu, and J. Chen, "K-means clustering versus validation measures: a data-distribution perspective," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 39, no. 2, pp. 318–331, 2009.
- [15] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling—Theory and Applications*, Springer Science + Business Media, 2005.
- [16] S. Guha, R. Rastogi, and K. Shim, "CURE: an efficient clustering algorithm for large databases," *ACM SIGMOD Record*, vol. 27, no. 2, pp. 73–84, 1998.
- [17] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications," *VLDB Journal*, vol. 8, no. 3-4, pp. 237–253, 2000.
- [18] P. S. Bradley, U. Fayyad, and C. Reina, "Scaling clustering algorithms to large databases," in *Proceedings of the 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '98)*, pp. 9–15, 1998.
- [19] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, 2007.
- [20] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "A local search approximation algorithm for k-means clustering," *Computational Geometry: Theory and Applications*, vol. 28, no. 2-3, pp. 89–112, 2004.

- [21] M. C. Chiang, C. W. Tsai, and C. S. Yang, "A time-efficient pattern reduction algorithm for k-means clustering," *Information Sciences*, vol. 181, no. 4, pp. 716–731, 2011.
- [22] P. S. Bradley, U. M. Fayyad, and C. Reina, "Scaling clustering algorithms to large databases," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '98)*, pp. 9–15, 1998.
- [23] F. Farnstrom, J. Lewis, and C. Elkan, "Scalability for clustering algorithms revisited," *SIGKDD Explorations*, vol. 2, no. 1, pp. 51–57, 2000.
- [24] C. Ordóñez and E. Omiecinski, "Efficient disk-based K-means clustering for relational databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 8, pp. 909–921, 2004.
- [25] Y. Li and S. M. Chung, "Parallel bisecting k-means with prediction clustering algorithm," *Journal of Supercomputing*, vol. 39, no. 1, pp. 19–37, 2007.
- [26] C. Elkan, "Using the triangle inequality to accelerate k-means," in *Proceedings of the 20th International Conference on Machine Learning*, pp. 147–153, August 2003.
- [27] D. Reddy and P. K. Jana, "Initialization for K-means clustering using voronoi diagram," *Procedia Technology*, vol. 4, pp. 395–400, 2012.
- [28] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, University of California Press, 1967.
- [29] R. Ebrahimpour, R. Rasoolinezhad, Z. Hajiabolhasani, and M. Ebrahimi, "Vanishing point detection in corridors: using Hough transform and K-means clustering," *IET Computer Vision*, vol. 6, no. 1, pp. 40–51, 2012.
- [30] P. S. Bishnu and V. Bhattacharjee, "Software fault prediction using quad tree-based K-means clustering algorithm," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 1146–1150, 2012.
- [31] T. Elomaa and H. Koivistoinen, "On autonomous K-means clustering," in *Proceedings of the International Symposium on Methodologies for Intelligent Systems*, pp. 228–236, May 2005.
- [32] J. M. Peña, J. A. Lozano, and P. Larrañaga, "An empirical comparison of four initialization methods for the K-Means algorithm," *Pattern Recognition Letters*, vol. 20, no. 10, pp. 1027–1040, 1999.
- [33] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithms: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.
- [34] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, 2003.
- [35] A. M. Bagirov, "Modified global k-means algorithm for minimum sum-of-squares clustering problems," *Pattern Recognition*, vol. 41, no. 10, pp. 3192–3199, 2008.
- [36] J. Xie and S. Jiang, "A simple and fast algorithm for global K-means clustering," in *Proceedings of the 2nd International Workshop on Education Technology and Computer Science (ETCS '10)*, vol. 2, pp. 36–40, Wuhan, China, March 2010.
- [37] G. F. Tzortzis and A. C. Likas, "The global kernel κ -means algorithm for clustering in feature space," *IEEE Transactions on Neural Networks*, vol. 20, no. 7, pp. 1181–1194, 2009.
- [38] W.-L. Hung, Y.-C. Chang, and E. Stanley Lee, "Weight selection in W-K-means algorithm with an application in color image segmentation," *Computers and Mathematics with Applications*, vol. 62, no. 2, pp. 668–676, 2011.
- [39] B. Maliatski and O. Yadid-Pecht, "Hardware-driven adaptive κ -means clustering for real-time video imaging," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 164–166, 2005.
- [40] T. Saegusa and T. Maruyama, "An FPGA implementation of real-time K-means clustering for color images," *Journal of Real-Time Image Processing*, vol. 2, no. 4, pp. 309–318, 2007.
- [41] "CBCL face recognition Database," Vol 2010, <http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html>.
- [42] BioID: BioID Support: Downloads: BioID Face Database, 2011.
- [43] <http://www.vision.caltech.edu/html-files/archive.html>.