



SHORT: Segmented histogram technique for robust real-time object recognition

Talal Bonny¹  · Tamer Rabie¹ · Mohammed Baziyad¹ · Walid Balid²

Received: 23 August 2018 / Revised: 5 May 2019 / Accepted: 23 May 2019 /
Published online: 3 June 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

Object recognition is a broad area that covers several topics including face recognition, gesture recognition, human gait recognition, traffic road signs recognition, among many others. Object recognition plays a vital role in several real-time applications such as video surveillance, traffic analysis, security systems, and image retrieval. This work introduces a novel, real-time object recognition approach, namely “SHORT”: segmented histogram object recognition technique. “SHORT” implements segmentation technique applied on the histogram of selected vectors of an image to identify similar image(s) in a database. The proposed technique performance was evaluated by means of two different image databases, namely the Yale Faces and Traffic Road Signs. The robustness was also assessed by applying different levels of distortion on both databases using Gaussian noise and blur, and testing distortion impact on recognition rates. Additionally, the efficiency was evaluated by comparing the recognition execution time of the proposed technique with another well-known recognition algorithm called “Eigenfaces”. The experimental results revealed successful recognition on clear and distorted objects. Moreover, “SHORT” performed 4.5X faster than the “Eigenfaces” algorithm under the same conditions. Furthermore, the “SHORT” algorithm was implemented on FPGA hardware by exploiting data parallelism to improve the execution performance. The results showed that the FPGA hardware version is 28X faster than the “Eigenfaces” algorithm, which makes “SHORT” a robust and practical solution for real-time applications.

Keywords Real-time · Object recognition · FPGA · YALE database · Road signs · Eigenfaces

1 Introduction

The enormous developments in the area of computer vision algorithms along with the massive improvements in processing power of modern computers have encouraged researches

✉ Talal Bonny
tbonny@sharjah.ac.ae

to build algorithms that can deal with real-time scenarios [8, 22, 34]. Moreover, researchers are trying to exploit the advantages of recent developments in image analysis [7, 29] and in processing power of computers to develop systems that are autonomous and can imitate human behaviors. One of such human behaviors is object recognition. An object recognition technique is an algorithm that tries to recognize and identify an object in a scene image or in a sequence of images (videos). Object recognition is a broad term that can include many sub-categories such as face recognition, human gait recognition, road signs recognition, etc. Many applications can be developed using an object recognition system such as video surveillance, traffic analysis, security systems and image retrieval.

Given a test image and a database that has a set of images, an object recognition system elects image from a database that achieves the highest match with the test image, and thus it assumes that it has recognized the test image.

Methods for object recognition can be classified into two main categories, the appearance-based methods, and the feature-based methods [15, 32]. The appearance-based methods are those methods that try to identify and recognize an object by referring it to a well known template, and do a comparison based on the pixel intensity values of both images. These methods are usually simpler and easier to implement. Thus, they are more applicable to be used in a real-time system. However, they can easily suffer from any change in the appearance of an image like having a distorted image, a change in the lighting conditions, or a change in the orientation or size. Examples of such approaches are edge matching, gradient matching, and histogram matching [15].

The other object recognition category namely the feature-based methods. In these methods, the algorithm searches to find sensible matches between image features and object features [32]. Corners, surface patches, and linear edges are all examples of such features. Feature-based methods are less sensitive and have a higher ability to resist the appearance changes such as lighting condition. However, a drawback of these methods is that a single position of the object must consider all feasible matches between the two images.

The “Eigenfaces” is a well-known technique used widely for face detection and recognition [18, 36, 37]. Its popularity comes from its simplicity and its real-time support comparing to other techniques. This approach can easily track and detect a person’s face, and then does a feature comparison with some known faces to recognize the person. The recognition idea is to develop a feature space of the tested face that can span and overcome significant differences between known face images. The significant features are simply the eigenvectors of the set of faces. Unlike other techniques, these features do not necessarily represent the human’s facial features like nose, eyes, and ears. Therefore, it may also be used to detect and recognize objects.

Histogram of intensity, as a statistical image description [9], has been an attractive subject to the research community in the past and recent years [2, 3, 11, 12]. Face recognition was proposed first in [11, 12] by matching the gray-scale intensity histogram of the image with those of the training images. However, direct matching as in [11] does not account for image degradations or changes in lightening conditions. Field programmable gate arrays (FPGAs) provide flexibility, fast prototyping, and capability of being reprogrammed/configured in Bioinformatics [4], Electronics [5, 6], image processing [7, 8], etc.

In this paper, a novel object recognition technique called segmented histogram object recognition technique (SHORT) is proposed, which is robust against distortion and suitable for real-time scenarios. The new technique can fall in the appearance-based category as it is a histogram based technique.

The proposed “SHORT” scheme calculates a metric value (we call it similarity distance) between the tested image and each database image based on computing the histogram for

individual segments of selected vectors of the image. The vectors are selected based on the type of the database being tested. The lowest metric value between the tested and database images refers to the database image which is most close to the tested one.

To evaluate the performance of “SHORT”, two different image databases were used, the Yale Faces Database [14], and the Crown Copyright Traffic Road Sign Database from GOV.UK [35]. The experimental results showed that those images were identified among other images by recording the lowest similarity distance. To evaluate the robustness of “SHORT”, different levels of distortion (impairments) were added to the images of both databases to test its effect on the recognition procedure. The distortion was added using different variances of the Gaussian noise and convolution of a Gaussian blur mask. The experimental results showed that the distorted images were identified among other images by recording the lowest similarity distance. Thus, “SHORT” can be useful for images that suffer from these types of impairments like satellite images, images being transferred using a noisy channel, or images obtained from incorrect camera settings. To evaluate the efficiency of “SHORT”, its recognition time was compared with the time required for recognition by the “Eigenfaces” algorithm. The experimental results showed that the software version of “SHORT” is 4.5X faster than the “Eigenfaces” algorithm, and the FPGA hardware version is 28X faster. This summarizes our novelty in this work which is performing the object recognition in real-time applications using FPGAs with extremely high accuracy (even higher than deep Learning methods as explained later).

The contributions of this work can be described as follows:

1. The paper introduces a novel technique for object recognition which identifies the distorted images with different distortion levels including noise or blur, correctly. It gives a match rate of 95% when a non-uniform distortion is added.
2. The proposed technique can be implemented for different types of databases by using different image segments.
3. The proposed technique was implemented on a hardware platform for practicality.
4. The time required to recognize a query object among different objects in a database was reduced explicitly using “SHORT” in comparison to the well-known face recognition algorithm “Eigenfaces”.
5. The proposed technique does not require huge computational and processing efforts in the training phase as required by state-of-the-art deep learning-based methods.

The rest of the paper is organized as follows: In Section 2, some related work on object recognition is presented. Section 3 introduces the method of our histogram based technique and its vector segmentation to identify the

distorted images. Section 4 presents the hardware implementation of the proposed object recognition technique. Experimental results and discussion are presented in Section 5. The conclusion is presented in Section 6.

2 Related work

In [37], the authors have developed the well-known face recognition method called the “Eigenfaces”. The recognition idea is to develop a feature space of the tested face that can span and overcome significant differences between known face images. The significant features are simply the eigenvectors of the set of faces, and from there the name “Eigenfaces” came from.

There exist a variety of applications on the pedestrian tracking, in which every application comes with its requirements. For example, when tracking the athletic performance, it is required that specific body parts must be thoroughly tracked. However, when tracking pedestrians, all people in the scene can be looked at as one unit. Thus, it is not easy to implement a system that can carry out simultaneous tracking. Traditional systems can only track the body parts of maximally two people at the same time. In [21], a novel method for tracking pedestrians is proposed to track several traffic objects such as bicycles, roller bladders, among others. The assumption is that the directions of the pedestrians do not change as they move. In [21], the image is clustered in different groups with a common color in every group. Using the k-means clustering algorithms, other clusters are developed in the subsequent images. Another assumption is that the legs of the pedestrians form a single cluster. This means if the device can recognize the legs, it can track the pedestrians. This device has an advantage, in that it works even when the camera is in motion. The classification of the tracking device is analyzed first to determine the type of template to use.

In [13], a real-time system was implemented using the “Eigenfaces” as the basis of the system. First, the system computes the eigenvectors of the set of faces being tested forming the “Eigenfaces” set. Then, a new face image obtained from the “Eigenfaces” is projected onto face space. After that, a comparison between this projection and the available projections of all training set images is performed to identify the person using the Euclidian distance.

An omnidirectional vision based technique for object detection is proposed in [10]. This technique adopts the Histogram of Gradients (HOG) features and sliding windows approaches which are used with traditional cameras. Next, the technique modifies these approaches to be used correctly and effectively with omnidirectional cameras. The modification steps include the adjustment of gradient magnitudes using Riemannian metric and conversion of gradient orientations to form an omnidirectional sliding window. This object detection technique has enabled detection directly on the omnidirectional images without the need to convert them to panoramic or perspective images.

In [27], a moving objects detection method is presented. This novel method uses the Frame differencing and W4 algorithm to detect the moving object. Previous work in moving objects detection used either the Frame differencing or the W4 algorithm individually. However, it has been found that because of the foreground aperture and ghosting problems, the detected results of the individual approach are not precise. By using the histogram-based frame differencing technique, the detection method calculates the difference between consecutive frames. Then, the W4 algorithm is performed on frame sequences. Next, a logical 'OR' operation is applied to the outputs of the frame differencing and W4 algorithm. Finally, the moving object is detected using the morphological operation with connected component labeling.

In our previous work [8], the algorithm is used to recognize face images with different facial expressions and different ambient illumination levels. The distance metric of the algorithm is based on the standard deviation of the histogram of the row in the image. In this work, our algorithm is more general. It can be used to recognize objects (or faces) distorted with different distortion levels including noise or blur. The distance metric here is based on the accumulated difference between the histograms with of segments in the vectors.

Neural network [17] and deep learning methods [30, 31, 38] are also used for imagevideo analysis and recognition. In 2012, Geoffrey Hinton et. al. [17] developed deep learning methods using Convolutional Neural Networks (CNNs), which are a classical type of deep model that can directly extract local features from an input images.

The main disadvantages of deep learning methods in comparison to conventional methods, such as the proposed “SHORT” method, lies mainly in the complexity of the underlying architecture and its inability to adapt to different datasets other than the one it was trained to perform on.

Properly setting up the CNNs’ parallel GPU architecture and parameter optimization are key to successful operation of deep learning methods. This stage requires a lot of effort and time because of the enormous amount of training data involved. If the same specific models are always used to train a specific dataset, then testing a different dataset using this model will typically cause a decrease in performance of deep learning methods. In comparison, our method can be easily implemented using FPGAs and can run in real-time.

3 Method

The proposed technique divides the database images into vectors and calculates the histogram for different sizes of segments in the vectors. The histograms are compared to find the closest object image in the database.

The searched image is called the query image (Q) and the tested image in the database is called the database image (D).

An image of N rows and M columns have a total number of pixels equal to $N \times M$. Each pixel is represented using 8-bit value P such that $P \in [0, 255]$.

An image histogram is a vector H that represents the frequencies of all possible values of the discrete variable P . Hence, the histogram H of an image can be represented using the vector

$$H = [F(0) \ F(1) \ F(2) \ \dots \ F(255)]. \quad (1)$$

The frequency of any value P is given as

$$F(P) = \sum_{i=1}^{NM} \text{Binary}(P_i) \quad (2)$$

where P_i is the pixel value at the position i of the image. $\text{Binary}(P_i) = 1$ if $P_i = P$, and $\text{Binary}(P_i) = 0$ otherwise.

Comparing the histogram of the query image with each database image is not an accurate method to find a matching image in a database. Instead, and one of the main contributions in this work is to propose a new technique for object recognition, which is based on a histogram of selected vectors of the images. The vector-based histogram technique compares the histograms of individual vectors v of the images to find the lowest similarity distance.

The histogram for a certain vector v can be written as

$$H(v) = [F(0, v) \ F(1, v) \ F(2, v) \ \dots \ F(255, v)]. \quad (3)$$

The vectors v of the image are selected in a row-wise $v \in [1, N]$, column-wise $v \in [1, M]$, or diagonal-wise $v \in [1, N + M - 1]$ fashion, based on the type of the images in the database.

The row-wise vectors of an image should not be selected if there exist two images in the database such that one is horizontally flipped from the other. The reason for that is, when two images are horizontally flipped, the difference between histograms for each row will be similar, and consequently, the similarity distance between them shows that they are identical although they are completely different.

The column-wise vectors of an image should not be selected if there exist two images such that one is vertically flipped from the other, for the aforementioned reason.

In case of a database which may have some images flipped vertically or horizontally from others, then the vectors of the image should be selected in diagonal-wise.

Figure 1 shows two different examples of selecting the vectors of the image for histograms comparison. The images in the top are selected from the Yale Faces database. In this database, there are no images which might be flipped vertically or horizontally. Therefore, the vectors v are selected in horizontal-wise fashion, i.e. each vector represents a row of the image.

The images in the bottom are selected from the Traffic Road Signs database. It is clear that one of the images is horizontally flipped from the other. If the vectors are row-wise selected, then the similarity distance will be very small (or equal zero) which means that they are very close and will be recognized wrongly. This database might also have some images which are vertically flipped. Therefore, and to avoid this conflict, the vectors v are selected in diagonal-wise fashion.

After selecting the vectors of images, the similarity distance is calculated.

Figure 2 shows the flowchart of calculating the histograms using the proposed vector-based histogram technique. For each vector of the query and database images, the histogram

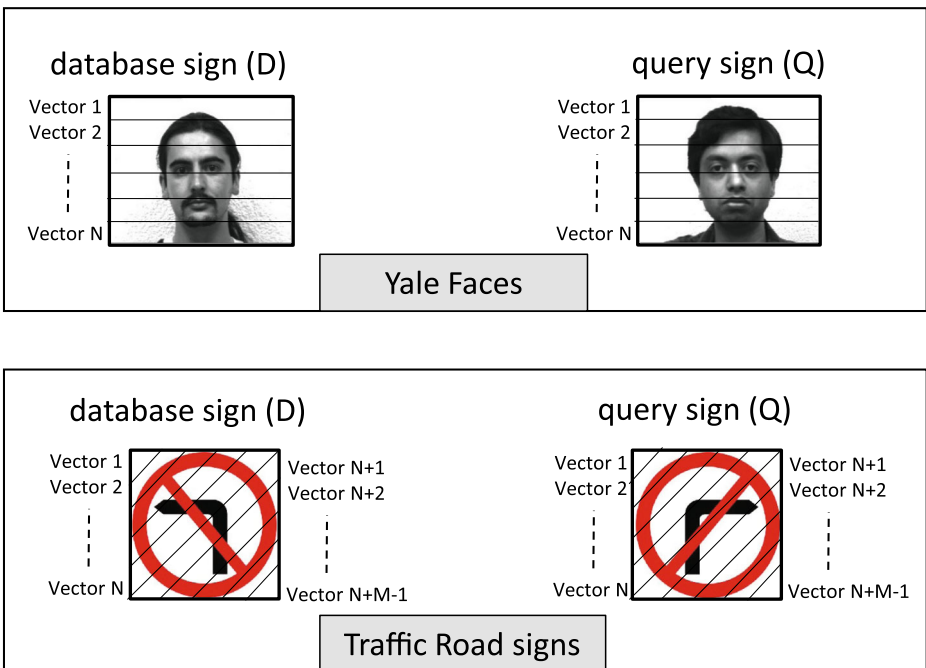


Fig. 1 Two different examples of selecting the vectors of the image for histograms comparison

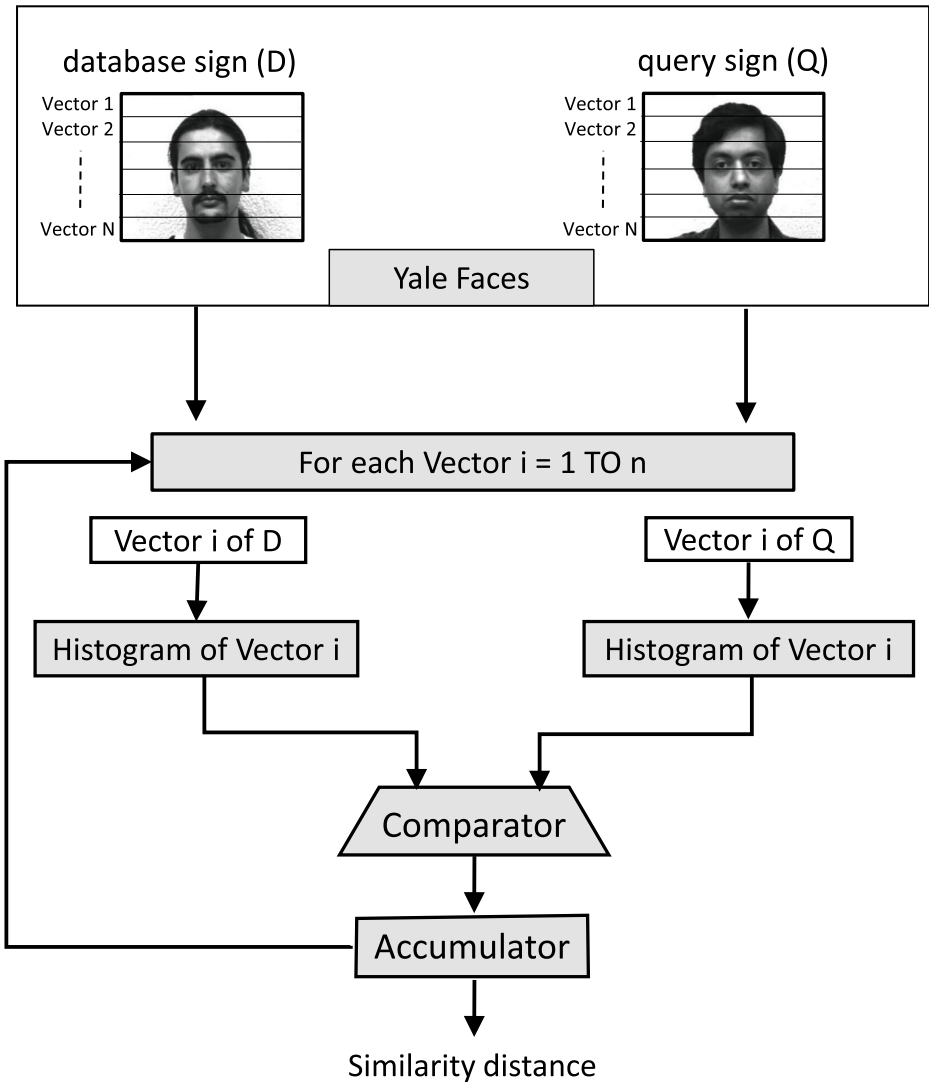


Fig. 2 Flowchart of calculating the similarity distance using our vector-based histogram technique

is calculated. Then the difference between the histograms is found and accumulated for all vectors. This will result in the similarity distance between the query and database images as explained in (4).

$$\begin{aligned}
 \text{Similarity Distance} = & |H(v_1(Q)) - H(v_1(D))| \\
 & + |H(v_2(Q)) - H(v_2(D))|
 \end{aligned}$$

$$\begin{aligned}
& + |H(v_3(Q)) - H(v_3(D))| \\
& + \dots \\
& + |H(v_n(Q)) - H(v_n(D))|
\end{aligned} \tag{4}$$

Where $H(v_1(Q))$ is the histogram of vector 1 of the query image. $H(v_1(D))$ is the histogram of vector 1 of the database image, etc.

This operation is repeated for all database images. The lowest similarity distance refers to the database image which is most close to the query.

When the query image is distorted with noise or blur, it will not be identified among the database images correctly using the proposed vector-based histogram technique (as shown in the experimental results). Therefore, we introduce our novel and robust technique, named “SHORT”: segmented histogram object recognition technique. In this technique, the histograms of the selected vectors are segmented into different ranges of the same lengths.

Assuming that the histogram for a certain vector v is given in (3), and the segment length is L , then the segmented histograms can be written as

$$\begin{aligned}
H(v, s_1) &= [F(0, v)F(1, v)F(2, v) \cdots F(L - 1, v)] \\
H(v, s_2) &= [F(L, v)F(L + 1, v)F(L + 2, v) \cdots F(2L - 1, v)] \\
&\dots \\
H(v, s_{256/L}) &= [F(255 - L, v)F(255 - L + 1, v)F(255 - L + 2, v) \cdots F(255, v)]
\end{aligned} \tag{5}$$

Where $H(v, s_1)$ is the histogram of the first segment of selected vector v which represents the frequencies of all possible values of the discrete variable P such that $P \in [0, L - 1]$. The last segment $H(v, s_{256/L})$ represents the frequencies of all possible values of the discrete variable P such that $P \in [255 - L, 255]$. The Number of segments S is equal to $256/L$. As all segments should have the same length, therefore the segment length L may have only the values which are divisible by 256, i.e. 2, 4, 6, 16, 32, 64, or 128. When $L = 1$, that means “SHORT” is not using the segmentation but it compares the histograms of the complete vector, as in (3).

Figure 3 shows the general flowchart for calculating the similarity distance using “SHORT”. It might be used for any database but the figure shows that the flowchart is used for road signs database. In our technique, the images are separated into either N vertical vectors, M horizontal vectors, or $N+M-1$ diagonal vectors based on the type of images in the database. In the Yale database, there is no need to separate the images in diagonal-wise fashion because there are no images in this database that are flipped vertically or horizontally. In this technique and for each selected vector (in row-wise, column-wise, or diagonal-wise fashion) of the query and the database images, the histogram is calculated. Then the histogram is segmented into S number of segments based on a selected segment length L . After segmentation, the comparators find the difference between query segments and database segments. The number of comparators is equal to the number of segments. The accumulator accumulates all segment differences of all vectors to result in the similarity distance as shown in (6)

$$\begin{aligned}
\text{Similarity Distance} &= |H(v_1, s_1(Q)) - H(v_1, s_1(D))| + |H(v_1, s_2(Q)) - H(v_1, s_2(D))| + \dots \\
&+ |H(v_2, s_1(Q)) - H(v_2, s_1(D))| + |H(v_2, s_2(Q)) - H(v_2, s_2(D))| + \dots \\
&+ \dots \\
&+ |H(v_n, s_1(Q)) - H(v_n, s_1(D))| + |H(v_n, s_2(Q)) - H(v_n, s_2(D))| + \dots
\end{aligned} \tag{6}$$

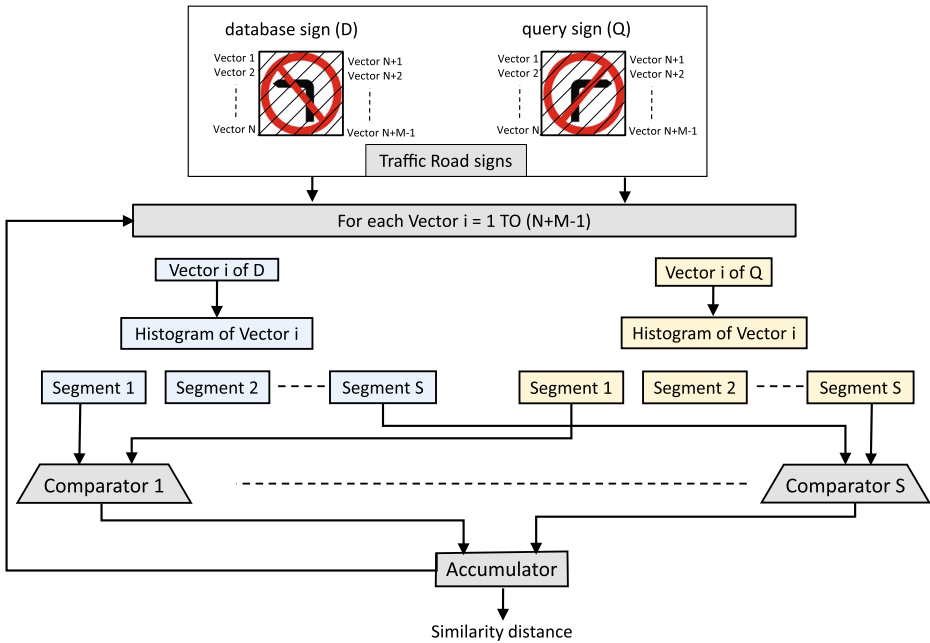


Fig. 3 Flowchart for calculating the similarity distance using our technique “SHORT”

Where $H(v_1, s_1(Q))$ and $H(v_1, s_1(D))$ are the histograms of the first segment of the first vector of the query and database images, respectively.

4 FPGA hardware implementation of the “SHORT”

In this section, a demonstration on how to implement “SHORT” on the prototyping FPGA Zed-Board from Xilinx¹ is presented. It includes ZYNQ-7000 FPGA [39]. The FPGA chip which has 140 BlockRAMs, 13,300 logic slices, and 220 DSP48E1s. The FPGA chip combines dual-core ARM Processing System (PS) with Programmable Logic (PL). An industry standard Advanced eXtensible Interface (AXI) is used to connect the two parts interconnects with low latency connections, and high bandwidth [16]. The board contains also 512 MB DDR3 Memory and some other peripherals to enable users to experiment with various aspects of their embedded designs.

Figure 4 shows the block diagram of the hardware implementation on the FPGA board. It consists of the following IP blocks:

- The Zynq Processing System (PS) which includes the ARM processor and DDR memory controller.
- The FPGA Block RAM.
- The AXI Direct Memory Access (DMA).
- Four Histogram IPs.
- Four Segmentation IPs.

¹<http://www.digilentinc.com/>

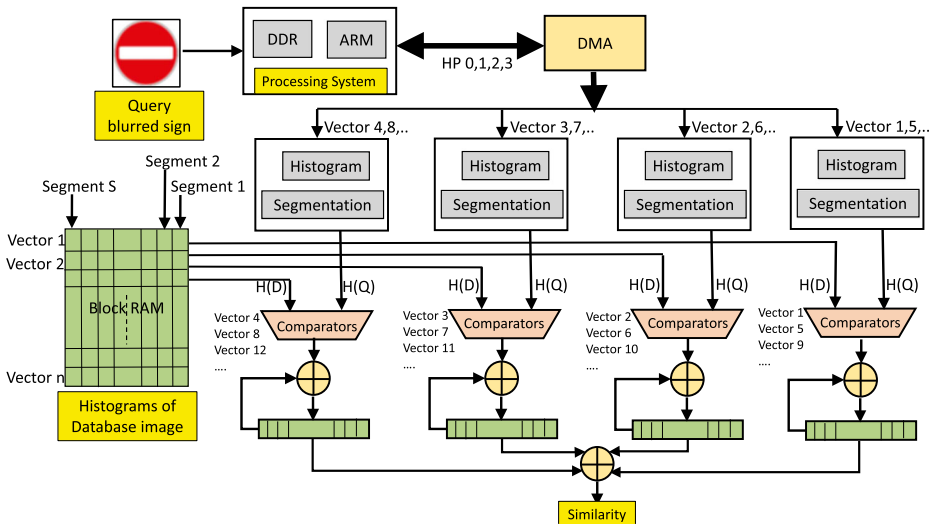


Fig. 4 FPGA Hardware of “SHORT” on FPGA Zed-Board

- Four Comparators IPs.
- Four Accumulators IPs.

The vectors of query image are stored in the DDR memory, while the Block RAMs are used to store the segmented histograms of each vector for each database image. These values were calculated off-line (design time) for only one time. For any new database image, its segmented histograms have to be calculated and stored in the Block RAM. In Fig. 4, the segmented histograms of each vector of the image will be stored in one row of the Block RAM.

Transferring data from the DDR memory to other system parts is performed through High Performance ports (HP). This will increase the data throughput and will offload the processor from tasks that involve memory transfers. The histogram IP receives the query image vector from the DDR through the DMA port and calculates its histogram.

The segmentation IP splits the vector histogram into the same length of segments.

The comparators IP consists of a number of comparators equal to the number of segments S . Each segment comparator computes the difference between segment histogram of the query and the database images. The accumulators IP sums up the differences of specified vectors to find the final one between query and database images and then it stores the results in a register. The specified vectors for the first comparators IP are Vectors 1, 6, 9, etc.; and for the second comparators IP are vectors 2, 6, 10, etc.; and so on. All computed differences from all comparators IPs are accumulated to compute the similarity distance.

In the FPGA Zed-Board, the high performance ports (HP) are used to access the external memory (DDR3). Those ports are designed for maximum bandwidth of 4 ports \times 64 bits \times 150 MHz = 4.8 GByte/sec. Four 32-bit DMA ports (HP0-HP3) are used to receive data from the DDR. Therefore, the hardware implementation contains four parallel threads. Each thread consists of histogram IP, segmentation IP, comparators IP, and accumulators IP. Hence, our hardware implementation can process four vectors in parallel.

5 Experimental results and discussion

In this section, the experimental results of “SHORT” and its comparison with the “Eigen-faces” technique are presented. For evaluation, two databases were used; the Yale Faces Database [14], and the Crown Copyright Traffic Road Sign Database from GOV.UK [35]. The tested Yale database contains a total of 480 faces (320 x 243 pixels) including eight different facial expressions and four different distortion levels (noise and Blur). The tested Road Sign Database contains 600 sign images categorized in different groups.

Figure 5 shows samples of the Yale faces database. In addition to, a face sample with eight different facial expressions without any distortion. The bottom of Fig. 5 shows face samples after adding different Gaussian noises with variances of 1.1, 2.1, and 3.1, in addition to Gaussian Blur of 13×13 convolution mask.

Figure 6 shows 9 samples of original traffic road signs without distortion (top part), and after adding different Gaussian noises and Blur (bottom part).

The evaluation is performed on all clear and distorted images (noise or blur) of both databases.

5.1 Performance evaluation on clear images

To evaluate the “SHORT” technique, it was first tested with clear images (without distortion), and when the segment length = 1. For that, the first face (Face 1) of the Yale database is considered as a query which needs to be compared with all database faces. The results of “SHORT” are compared with the “Eigenfaces” Algorithm as it is a standard technique used for recognition. Tables 1 and 2 show the closest Yale database faces to the query (when the query is Face 1) using the “Eigenfaces” Algorithm and “SHORT”, respectively. The first column in each table, where Face 1 is a clear image, shows that the closest database

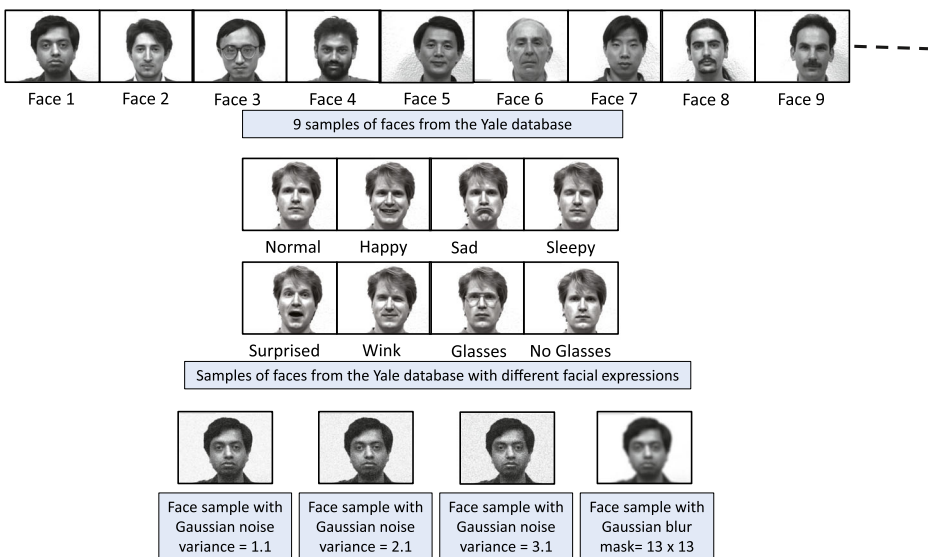


Fig. 5 Samples of faces from the Yale database. In addition, samples of faces with different facial expressions and distortions



9 samples of signs from the traffic road signs database

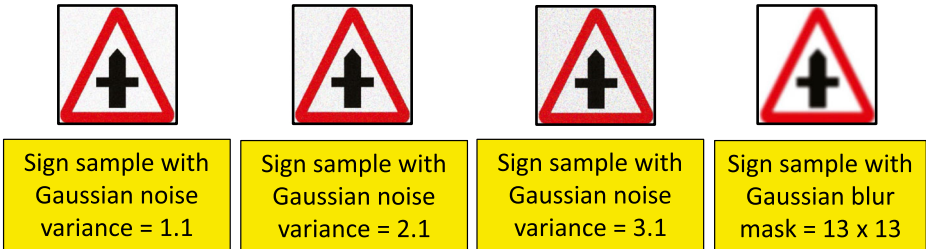


Fig. 6 Samples of road signs from the Traffic Road Sign Database. In addition to samples of distorted signs

faces to the query is the first face (Face 1) and then the third face (Face 3). This proves the performance of “SHORT” when the query face is clear and has no distortion.

This conclusion is not only valid when the query is Face 1, but also for any face in the database. Figure 7 shows samples of the similarity distance between different query faces and all database faces (the last bar shows the average distance through all remaining images in the database). The query in Fig. 7a is Face 1, while it is Face 5, Face 7, and Face 9 in Fig. 7b, c and d, respectively. The query face, in each figure, has no distortion, i.e. it is exactly the same as in the database. It is clear in all figures that the similarity distance between the query face and the same face in the database is zero, but all other distances are greater than zero. This concludes that Face 1 (query) is identified in the database in Fig. 7a, face 5 (query) is identified in Fig. 7b, face 7 (query) is identified in Fig. 7c, and Face 9 (query) is identified in Fig. 7d.

Table 1 Closest database faces to the query (Face 1) using the “Eigenfaces” Algorithm

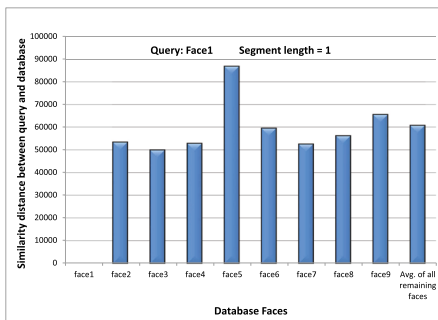
Yale database					
	Face 1				
Query	Clear	Noise = 1.1	Noise = 2.1	Noise = 3.1	Blur
Closest database faces	Face 1	Face 1	Face 1	Face 1	Face 1
	Face 3	Face 3	Face 3	Face 3	Face 3

Table 2 Closest database faces to the query (Face 1) using “SHORT” without segmentation (Segment Length = 1)

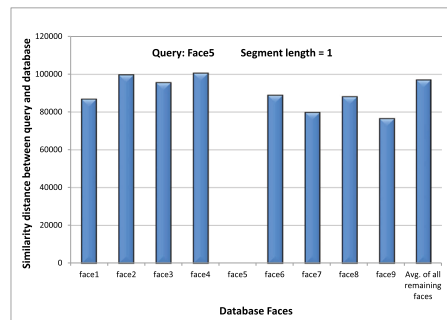
Yale database		Face 1				
Query		Clear	Noise = 1.1	Noise = 2.1	Noise = 3.1	Blur
		Face 1	Face 5	Face 5	Face 5	Face 1
Closest database faces		Face 3	Face 9	Face 9	Face 9	Face 7

5.2 Performance evaluation on distorted images

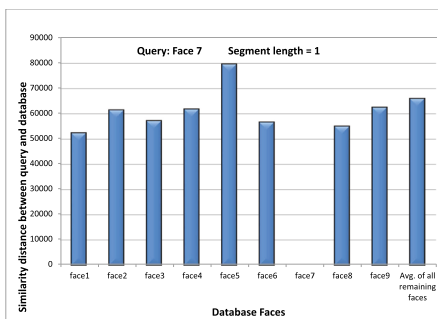
Adding distortion to the query face will not give the same results as the clear faces. In Tables 1 and 2, the second, third, and fourth columns show the results when different Gaussian noises with variances of 1.1, 2.1, and 3.1, are added to the query (Face 1), respectively. The last column shows the results when a Gaussian blur of 13×13 convolution mask is added to the query face. The “Eigenfaces” algorithm in Table 1 show that the closest database face to the query for any noise variance or blur is Face 1 and the second closest one



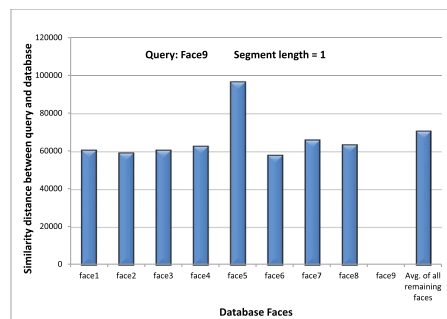
(a) Query: Face 1



(b) Query: Face 5



(c) Query: Face 7



(d) Query: Face 9

Fig. 7 Similarity distance between query face and all database faces (segment length = 1). When query and database faces are the same, the distance is zero

is Face 3, while in case of “SHORT”, Face 5 is the closest database face to the query and Face 9 is the second closest one. This concludes that “SHORT” does not deliver accurate results when the query has noises and the segment length is 1.

Figure 8 shows the similarity distance between the query (Face 1) and the first nine faces of the database (first nine bars), and the average of remaining faces in the database (last bar). The query face is distorted with a Gaussian noise variance of 1.1 (column 1), the variance of 2.1 (column 2), the variance of 3.1 (column 3), and blur (column 4). In this figure, the segment length used in “SHORT” is 1. As shown in this figure, the lowest distance between the distorted query face and the database faces is not Face 1. Instead, it is Face 5 and then Face 9 for any noise level. Another important note in Fig. 8 is that increasing the noise variance of the query face (Face 1) from 1.1 to 3.1 will increase the similarity distance between the distorted query and Face 1 in the database.

The distorted query can be identified when “SHORT” uses large segment length, as shown in Fig. 9. In this figure, the query face (Face 1) is distorted with the three Gaussian noise variances (one column for each variance), and then “SHORT” is applied with different segment lengths starts from 1 till 128. When segment length is 1, 2, or 4, the closest database faces to the noisy query is Face 5, for any noise variance. Starting from segment length = 16, the closest database face becomes Face 1 but only when the level of noise is low (variance = 1.1). When the segment length increases, “SHORT” delivers accurate results for a higher noise variance. When the segment length is the highest (128), the closest database faces to the noisy query is Face 1, for any noise variance.

Figure 10 shows the similarity distances between query and database faces, for all noise variances and blur, when segment length = 128. It is clear that the lowest similarity distance is for Face 1 of the database even when the query face is distorted with any variance of

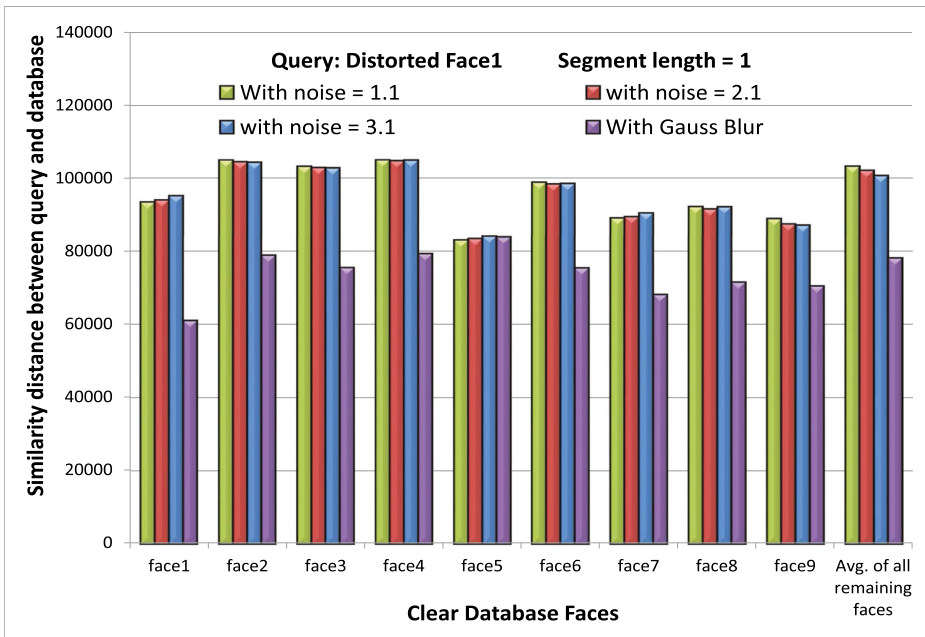


Fig. 8 Results of comparing distorted Face 1 with all faces when segment length = 1

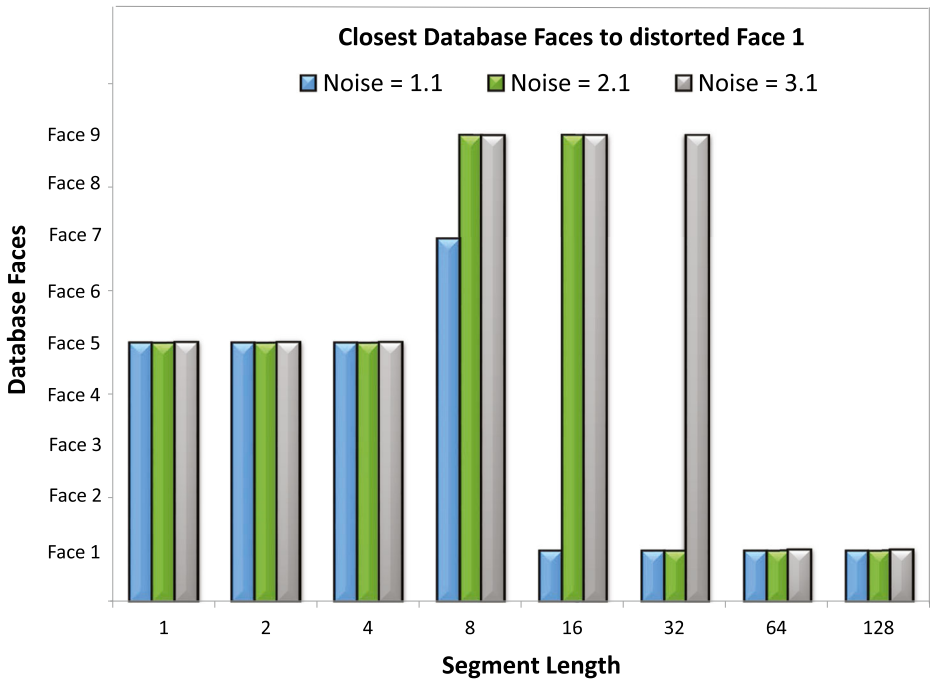


Fig. 9 Closest database faces to distorted Face 1 for different segment lengths and under different levels of distortion

the Gaussian noise. This shows how “SHORT” is robust against any noisy or blurred query when segment length = 128.

Using the segment length of 128 decreases the similarity distance between the distorted query and database faces more than other segment lengths. This is shown in Fig. 11. This figure shows how the distance between the query and database faces decreases when the segment length increases from 1 to 128. On the other hand, higher segment length requires more time for computing the differences between histograms of the query and database images. As all segments should have the same length, therefore the segment length L may have only the values which are divisible by 256, i.e. 2, 4, 6, 16, 32, 64, or 128. The query in Fig. 11 is the noisy and blurred Face 1, and the database face is the clear Face 1. When segment length = 1, the similarity distance starts from 95000 (for noisy query), and from 61000 (for blurred query). It decreases till 1400 when segment length = 128. This figure shows also that the distance is higher when the noise variance is high, and the distance for the blurred face is less than the noisy one.

The proposed “SHORT” method identifies any query face distorted with any noise or blurred level when segment length = 128. Figure 12 shows other examples for different distorted query faces, such as Face 2 in Fig. 12a, Face 5 in Fig. 12b, Face 7 in Fig. 12c, and Face 9 in Fig. 12d. In any of these figures, the similarity distance is the lowest when the clear face of the database is the same as the distorted face for any distortion level.

The proposed “SHORT” methodology is not only a technique for face recognition, but it may also be used for object recognition, which may be used to identify any object in



Fig. 10 Results of comparing distorted Face 1 with all faces for range length = 128

a database of different objects. Therefore, “SHORT” is also verified to detect traffic road signs.

When “SHORT” is used, the experimental results show the same conclusion as we got in the Yale Faces Database (Figures are not included for brevity purposes):

- In the case of clear images, the similarity distance between the query road sign and the same sign in the database is zero, but all other distances are greater than zero.
- In the case of distorted images, when the segment length increases, the similarity distance between the query and database signs decreases.
- The proposed “SHORT” algorithm delivers accurate results when the segment length is high (64 or 128)

Figure 13 shows samples of results obtained for similarity distance between different distorted query signs and some clear database signs. The last bar shows the average of remaining database signs. The segment length in these samples is 128. In Fig. 13a, the query is Sign 1 and the closest database sign is Sign 1 (for any added distortion). The next closest sign is Sign 2 (as it is close to it). When the query becomes Sign 3 (Fig. 13b), the closest and second closest database signs are Sign 3 and Sign 4, respectively (as they are close to each other). And when the query is Sign 9, the closest database sign is Sign 9 (Fig. 13d).

In Fig. 13c, the query is Sign 7. In this case, the similarity distance between the query and database Sign 7, and between the query and database Sign 8 are almost the same because Sign 7 and Sign 8 are very close (horizontally flipped). This would create conflict when “SHORT” is used to detect the traffic road signs because there are many signs that are horizontally or vertically flipped.

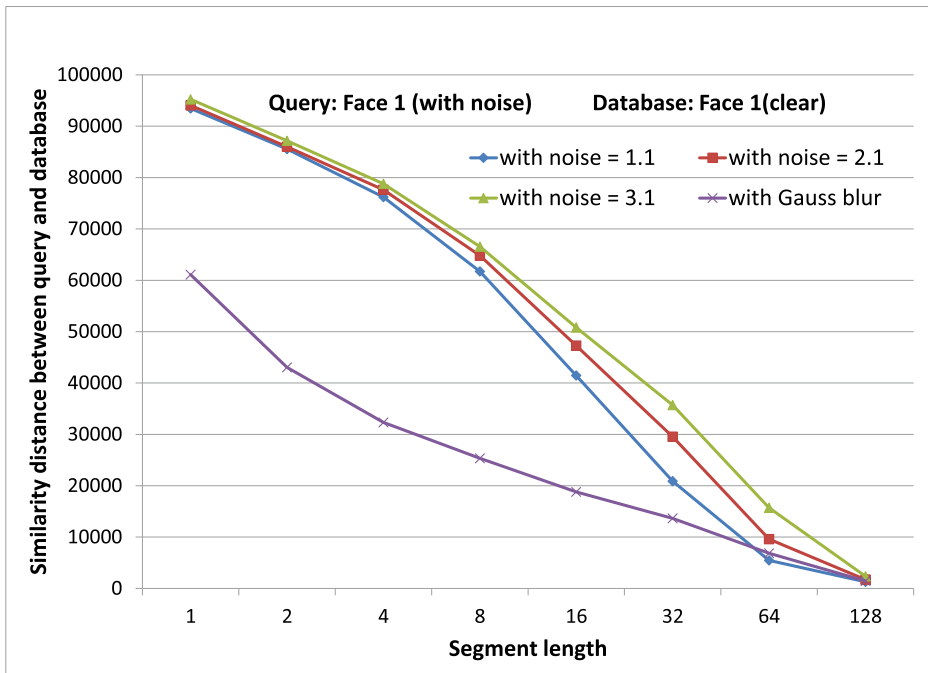


Fig. 11 Results of comparing Face 1 under different levels of distortion with clear Face 1 for different range lengths

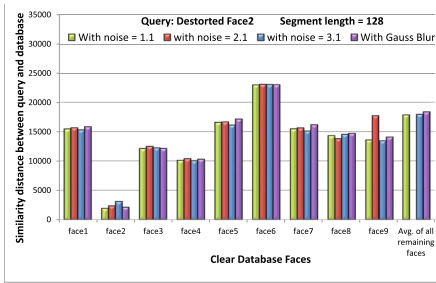
As discussed in Section 3 (Fig. 1), the solution to this problem is to compare the segmented histograms in a diagonal-wise.

Figure 14 shows the results of comparing the distorted flipped Sign 7 with the clear database signs but in a diagonal-wise. In this case, “SHORT” delivers correct result without conflict, because the distance between the query and database Sign 7 is the lowest and much far from other distances. These results show solid robustness of “SHORT” to detect objects for any type of distortion.

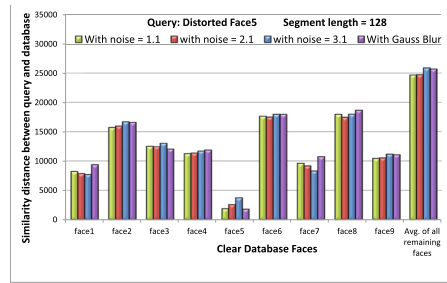
5.3 Performance evaluation on non-uniform distorted images

All results in the previous section are presented when the faces are distorted with a uniform Gaussian noise or blur. In real-world scenarios, the images may be distorted with non-uniform noise or blur. Figure 15 shows samples of faces from the Yale database that are distorted with non-uniform noise and blur.

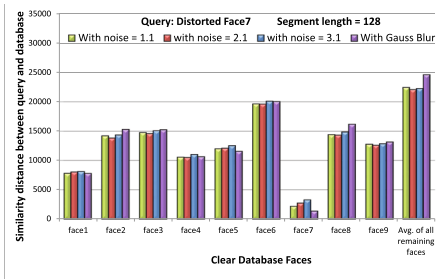
To evaluate the robustness of “SHORT” to non-uniform distortion, all face images from the Yale database are distorted with non-uniform noise and blur. The non-uniform distortion attack applied to the test images is the “Salt & Pepper” attack. The “Salt & Pepper” attack replaces random pixels with black and white pixels of a density N , where N is the percentage of the total additive noise pixels with respect to the total number of pixels. On the other hand, a spatially variant Gaussian distributed filter kernel is used to perform the non-uniform blurring effects to an image. The blurred image is obtained by convolving the input image with the Gaussian filter kernel. The standard deviation of this Gaussian kernel is responsible of the blurring strength effect.



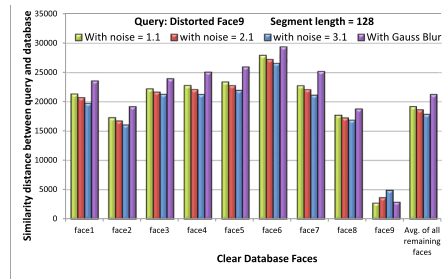
(a) Query: Distorted Face 2



(b) Query: Distorted Face 5



(c) Query: Distorted Face 7



(d) Query: Distorted Face 9

Fig. 12 Samples of results show the similarity distance between distorted query faces and clear database faces (segment length = 128)

After applying the non-uniform noise and blur, the “SHORT” is applied with different segment lengths starts from 1 till 128. The match rate is computed for all distorted faces and for each distortion type.

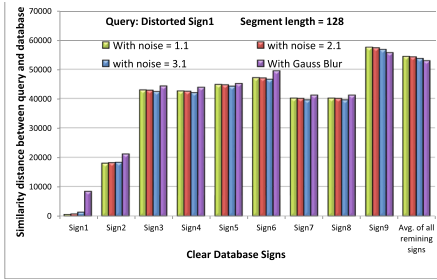
Figure 16 shows the match rate for non-uniform distorted images and different segment lengths the images are distorted with non-uniform noise (first column) and non-uniform blur (second column).

As shown in this figure, when the segment length increases, the match rate improves to reach the maximum in the case of segment length equals 128. The match rate in case of non-uniform noise is better than the rate in the non-uniform blur. For noise distortion, the match rate improves from 30% (for length equals 1) till 95% (for length equals 128). For blur distortion, the match rate starts from 50% and increases to reach 95%.

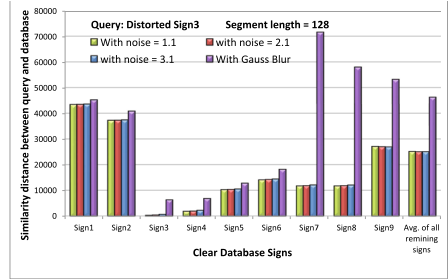
5.4 Execution time evaluation

The execution time evaluation is performed by comparing the time required to execute “SHORT” with the “Eigenfaces” algorithm, when both are running on the same platform (Intel core-I5 @ 350 GHz, 64-bit operating system). The proposed “SHORT” algorithm is written in C++ using the ”OpenCV” library as the “Eigenfaces” uses this library to assure fair comparison. The time required to run the “Eigenfaces” was 280 ms while “SHORT” required only 60 ms (which is 4.5X faster).

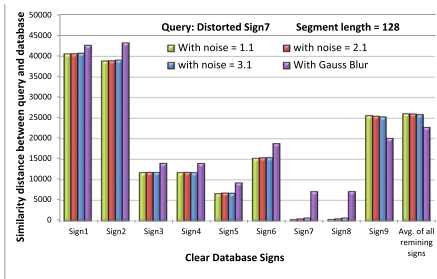
For hardware evaluation, The proposed “SHORT” algorithm was re-written in VHDL and simulated using the ModelSim. The design is synthesized, implemented, and verified on the FPGA prototyping board (Zed-Board), using the Xilinx Vivado design suite [1]. All IPs



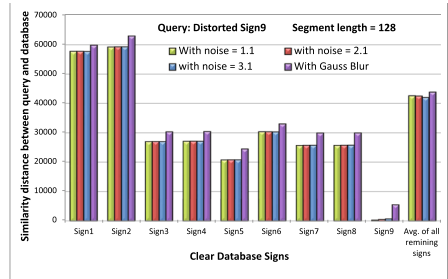
(a) Query: Distorted Sign 1



(b) Query: Distorted Sign 3



(c) Query: Distorted Sign 7



(d) Query: Distorted Sign 9

Fig. 13 Samples of results show the similarity distance between different distorted query signs and clear database signs (segment length = 128)

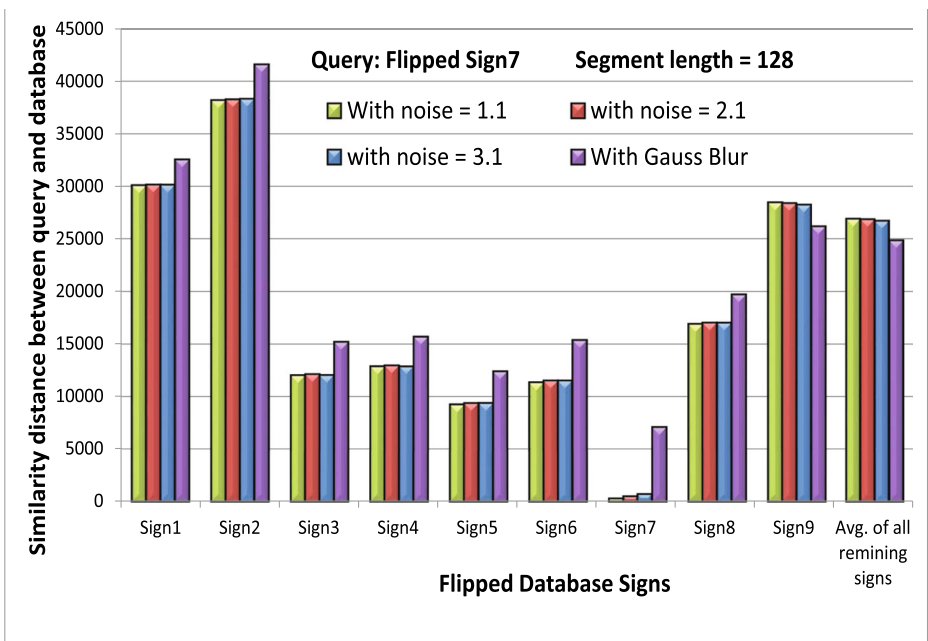


Fig. 14 Results of comparing distorted flipped Sign 7 with the clear database signs in a diagonal-wise

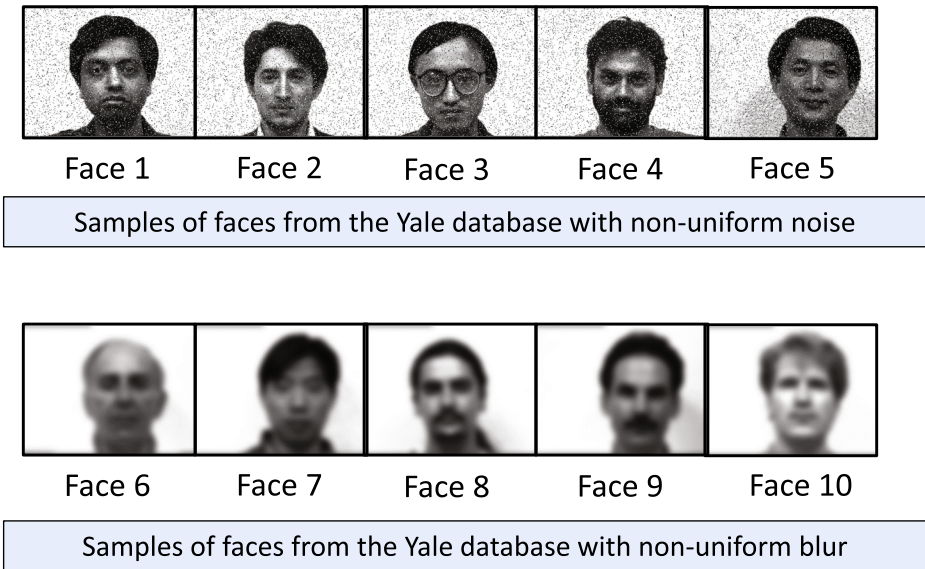


Fig. 15 Samples of faces from the Yale Database with non-uniform noise and blur distortion

are clocked using a 100 MHz frequency clock signal generated by the processing system. Running “SHORT” on the FPGA Zed-board improves the execution time to 10 ms, which is 6X faster than the software version, and 28X faster than the “Eigenfaces” algorithm. This high speed of “SHORT” execution would be enough to support the real-time applications.

5.5 Comparison with state-of-the-art recognition techniques

Table 3 presents maximum average match rates reported by various state-of-the-art recognition techniques and showing the highest average match rate for the proposed “SHORT” recognition method. As shown in Table 3, the recognition techniques are deep learning and neural network based. They used different training models with different dataset. The LT-FHist technique [23] introduced a framework to address the high computational cost associated with the multidimensional histogram processing and develops a training-less color object recognition and localization scheme. This LT-FHist method depends heavily on color features to augment the multidimensional Laplacian feature histogram pyramid representation derived, which contributes to the high percentage value of 96% for its true recognition match rate. Nevertheless, the proposed “SHORT” technique is only based on gray-scale image histograms, thus improving the speed of processing, yet the true recognition match rate for the proposed “SHORT” method is still very competitive at 95%. The One-Shot Learning method of [19] used the Caltech 101 data set², while the Zero-Shot Learning method of [24] used the SUN Data Set³. The Deep-Learning-based methods of [17] and [28] train a large deep convolutional neural network to classify

²http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html

³<http://vision.cs.princeton.edu/projects/2010/SUN/>

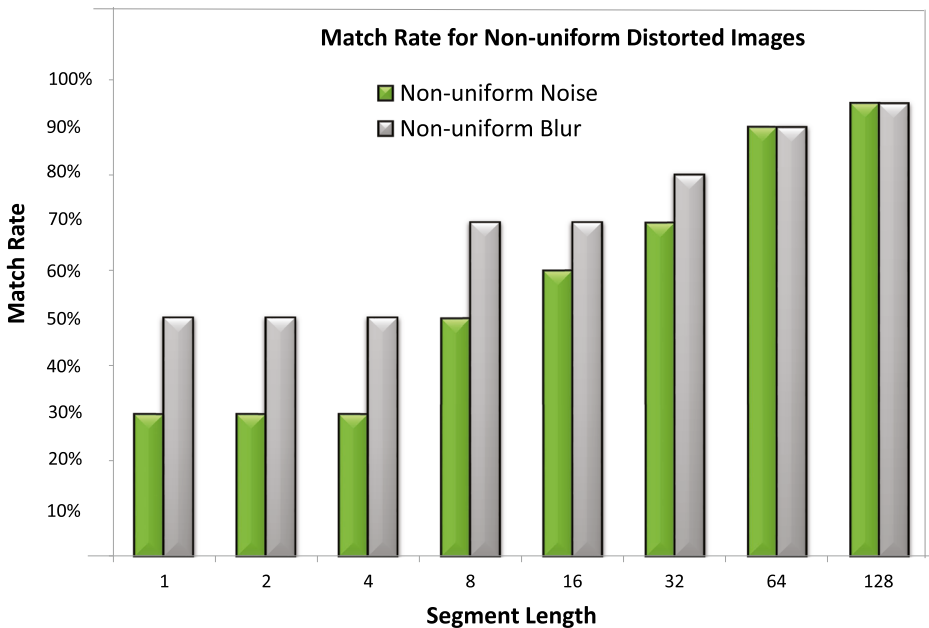


Fig. 16 Match rate for non-uniform distorted images and for different segment lengths

1.2 million high-resolution images in the ImageNet ILSVRC-2012 and ILSVRC-2013 competitions.

Schiele and Crowley [26] exploit the use of multidimensional histogram matching to perform recognition based on histograms of local shape properties which are modeled using receptive field vectors. Their technique uses a large number of training samples for the model in order to obtain a reliable histogram. Rothganger et. al. [25] introduce a novel representation for 3D objects in terms of local affine-invariant descriptors of their images and the spatial relationships between the corresponding surface patches. A limitation of their approach is its reliance on texture thus affecting recognition rates for poorly textured model images. David Lowe, in [20], describes a method for extracting distinctive invariant features, named SIFT key points, from the images, which enables the correct match for a key point to be selected from a large database of other key points. The early work of Swain and Ballard in [33] introduce a novel color object indexing and localization technique, known as intersection and back projection, based on 3D color histograms. One major drawback of their color histogram based method is its sensitivity to lighting conditions such as the color and the intensity of the light source.

The recognition techniques shown in the table require huge computational and processing efforts in the training phase. Moreover, the more complex the design of the deep model, the more time it needs to train and obtain a functional deep learning system. On the other hand, the “SHORT” recognition technique requires no training as it is not based on neural network models and it achieves high match rate (95%) in comparison to other work.

Table 3 The maximum true recognition match rates achieved by various state-of-the-art recognition techniques and the SHORT

Method	No. of Training Models	True-Positive Match Rate Achieved	Image Database Used
LT-FHIST [23]	Single color model image	96%	Object Recognition Database
SHORT	Single model (no training)	95%	Yale database
One-Shot Learning [19]	5 training model images	95%	Caltech 101 Data Set
Schiele & Crowley [26]	Single gray-scale model image	94%	Database of 50 Images [26]
Rothganger et. al. [25]	20 color training model images	94%	Object Recognition Database
Lowe's SIFT [20]	20 gray-scale training model images	88%	Object Recognition Database
Swain's Color Histograms [33]	Single color model image	87%	Object Recognition Database
OverFeat (Deep Learning) [28]	1.2 million high-resolution images	86%	ImageNet LSVRC-2013
Generic Deep Learning [17]	1.2 million high-resolution images	85%	ImageNet LSVRC-2012
One-Shot Learning [19]	Single training model image	82%	Caltech 4 Data Set
Zero-Shot Learning [24]	Description of model	66%	SUN Data Set

6 Conclusion

This paper introduced “SHORT”, a real-time object recognition technique that can be used to accurately identify a query image from a pool of database images under various distortion levels. A demonstrative comparison of the proposed technique with the well-known face recognition algorithm “Eigenfaces” was presented. The experimental study showed that “SHORT” is 4.5X faster than the “Eigenfaces” algorithm. “SHORT” was also implemented on an FPGA platform. The hardware implementation showed a significant improvement in processing speed. Experimental results demonstrated the efficiency of the proposed methodology in terms of robustness and support for real-time applications.

References

1. Xilinx. Vivado design suite - hlx editions (2016)
2. Bateux Q, Marchand E (2017) Histograms-based visual servoing. *IEEE Robot Autom Lett* 2(1):80–87
3. Beheshti I, Maikusa N, Matsuda H, Demirel H, Anbarjafari G (2017) Histogram-based feature extraction from individual gray matter similarity-matrix for alzheimers disease classification. *Journal of Alzheimer's Disease*, (Preprint), pp 1–12
4. Bonny T, Affan Zidan M, Salama KN (2010) An adaptive hybrid multiprocessor technique for bioinformatics sequence alignment. In: 2010 5th cairo international biomedical engineering conference, pp 112–115
5. Bonny T, Debsi RA, Majzoub S, Elwakil AS (2019) Hardware optimized fpga implementations of high-speed true random bit generators based on switching-type chaotic oscillators. *Circ Syst Signal Process* 38(3):1342–1359
6. Bonny T, Elwakil AS (2018) Fpga realizations of high-speed switching-type chaotic oscillators using compact vhdl codes. *Nonlinear Dyn* 93(2):819–833
7. Bonny T, Henno S (2018) Image edge detectors under different noise levels with fpga implementations. *J Circ Syst Comput* 27(13):1850209
8. Bonny T, Rabie T, Hafez AHA (2018) Multiple histogram-based face recognition with high speed fpga implementation. *Multimedia Tools and Applications*
9. Cha S-H, Srihari SN (2002) On measuring the distance between histograms. *Pattern Recogn* 35(6):1355–1370
10. Cinaroglu I, Bastanlar Y (2016) A direct approach for object detection with catadioptric omnidirectional cameras *Signal. Image Video Process* 10(2):413–420
11. Demirel H, Anbarjafari G (2008) Pose invariant face recognition using probability distribution functions in different color channels. *IEEE Signal Process Lett* 15:537–540
12. Déniz O, Bueno G, Salido J, De la Torre F (2011) Face recognition using histograms of oriented gradients. *Pattern Recogn Lett* 32(12):1598–1603
13. Georgescu D (2011) A real-time face recognition system using eigenfaces. *J Mob Embedded Distrib Syst* 3(4):193–204
14. Georgiades A, Belhumeur PN, Kriegman DJ (1997) Yale face database. In: Center for computational Vision and Control at Yale University, pp 2
15. Gross R, Matthews I, Baker Simon (2004) Appearance-based face recognition and light-fields. *IEEE Trans Pattern Anal Mach Intell* 26(4):449–465
16. Inc. Xilinx. AXI Reference Guide, volume 14. Xilinx (2012)
17. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
18. Kshirsagar VP, Baviskar MR, Gaikwad ME (2011) Face recognition using eigenfaces. In: 2011 3rd international conference on computer research and development (ICCRD). IEEE, vol 2, pp 302–306
19. Li F-F, Fergus R, Perona P (2006) One-shot learning of object categories. *IEEE Trans Pattern Anal Mach Intell* 28(4):594–611
20. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
21. Masoud O, Papanikolopoulos NP (2001) A novel method for tracking and counting pedestrians in real-time using a single camera. *IEEE Trans Veh Technol* 50(5):1267–1278

22. Or-El R, Rosman G, Wetzler A, Kimmel R, Bruckstein AM (2015) Rgb-d-fusion: Real-time high precision depth recovery. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5407–5416
23. Rabie T (2017) Training-less color object recognition for autonomous robotics. *Inf Sci* 418:218–241
24. Romera-Paredes B, Torr PHS (2015) An embarrassingly simple approach to zero-shot learning. In: *ICML*, pp 2152–2161
25. Rothganger F, Lazebnik S, Schmid C, Ponce J (2006) 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *Int J Comput Vis* 66(3):231–259
26. Schiele B, Crowley JL (2000) Recognition without correspondence using multidimensional receptive field histograms. *Int J Comput Vis* 36(1):31–50
27. Sengar SS, Mukhopadhyay S (2017) Moving object detection based on frame difference and w4. *SIViP*, pp 1–8
28. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y (2013) Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv:1312.6229
29. Song J, Gao L, Nie F, Shen HT, Yan Y, Sebe N (2016) Optimized graph learning using partial tags and multiple features for image and video annotation. *IEEE Trans Image Process* 25(11):4999–5011
30. Song J, Guo Y, Gao L, Li X, Hanjalic A, Shen HT (2018) From deterministic to generative: Multi-modal stochastic rnns for video captioning. *IEEE Transactions on Neural Networks and Learning Systems*
31. Song J, Zhang H, Li X, Gao L, Wang M, Hong R (2018) Self-supervised video hashing with hierarchical binary auto-encoder. *IEEE Trans Image Process* 27(7):3210–3221
32. Sun Z, Bebis G, Miller R (2004) Object detection using feature subset selection. *Pattern Recogn* 37(11):2165–2176
33. Swain M, Ballard D (1991) Color Indexing. *Inter J Comput Vis* 7:11x–32
34. Tompson J, Stein M, Lecun Y, Perlin K (2014) Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans Graph (TOG)* 33(5):169
35. Traffic road sign database (2016)
36. Turk M (2013) Over twenty years of eigenfaces. *ACM Trans Multimed Comput Commun Appl (TOMM)* 9(1s):45
37. Turk M, Pentland A (1991) Eigenfaces for recognition. *J Cogn Neurosci* 3(1):71–86
38. Wang X, Gao L, Wang P, Sun X, Liu X (2018) Two-stream 3-d convnet fusion for action recognition in videos with arbitrary size and length. *Trans Multi* 20(3):634–644
39. Xilinx Inc. 7 Series FPGAs Overview, volume 1. Xilinx (2014)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Talal Bonny received the M.Sc. degree in Computational Sciences and Engineering (CSE) from the Technical University of Braunschweig-Germany in 2002, and the Ph.D. degree in computer engineering from Karlsruhe Institute of Science (KIT), Karlsruhe, Germany in 2009. He worked as Post-Doc fellow for two years at King Abdullah University for Science and Technology (KAUST) in 2011. He joined University of Sharjah, UAE, in 2013 as assistant professor in the department of Electrical and Computer Engineering. Dr. Bonny served as reviewer/TPC member in many IEEE/ACM journals/conferences. His current research interests include embedded systems and hardware digital design, optimization algorithms, and bioinformatics.



Tamer Rabie received the M.Sc. degree in adaptive image restoration in 1993 from the Department of Electrical and Computer Engineering at The University of Calgary in Canada, and the Ph.D. degree in active computer vision applied to dynamic virtual environments in January 1999 from the Edward S. Rogers Sr. Department of Electrical and Computer Engineering at the University of Toronto in Canada, where, in collaboration with world-renowned computer scientist Professor Demetri Terzopoulos, they pioneered the Animat Vision Paradigm, widely cited today in the virtual vision literature. During the period from May 1998 to September 1998 Dr. Rabie held a five-month software engineer contract with the visualization and image processing group at ISG Technologies Inc. in Mississauga, Ontario, Canada, where he was involved in a critical project to develop image quality enhancements for leading edge volume rendering servers for the medical imaging industry. From October 1998 to December 1999 he held a postdoctoral fellow in the Department of Computer Science at the University of Toronto where he was involved in further research and development of the Animat Vision project. From January 2000 to August 2001 he held an assistant professor appointment in the Department of Electrical and Computer Engineering at Ryerson University in Toronto, Canada, and an associate professor position in the Faculty of Information Technology at the UAE University from September 2001 to 2011. He is currently Associate Professor of Computer Engineering at the University of Sharjah. He was also an adjunct professor at the University of Toronto, Intelligent Transportation Systems Centre, conducting collaborative research work in active computer vision-based traffic surveillance and control. His current research interests include digital image processing, computer vision, image/speech watermarking, and intelligent robotic systems. Dr. Rabie has published papers in digital image processing, computer vision, and artificial intelligence. Dr. Rabie is a Senior Member of the IEEE, and a member of the IEEE Signal Processing Society. He is also a member of the Professional Engineers of Ontario (PEO) Association in Toronto, Canada, and licensed as a professional engineer in the province of Ontario since 2001.



Mohammed Baziyad received his Bachelor degree in Network Engineering from the Canadian University of Dubai in 2015. Currently he is Research Assistant at the Autonomous Robotics & Active Vision Research Group at the University of Sharjah. He is currently involved in research projects related to robotics. Research interests include information security, steganography, active computer vision integration, nonlinear robotic control, robot tracking/path planning, simulating human emotion for robotics, stereo and color analysis for dynamic navigation.

Walid Balid received the Ph.D. in Electrical and Computer Engineering from the University of Oklahoma (OU) in 2016, the B.S., Dip.Sc, and M.Sc. degrees in electronic engineering from the University of Aleppo-Syria, in 2006, 2009, and 2011, respectively. Currently, he is Principal Research Engineer at Panduit Corporate R&D. His main research interests include Embedded Systems Development, Intelligent Transportation Systems, Wireless Sensor Networks, IoT, Hardware Design, and Data Analysis.

Affiliations

Talal Bonny¹  · Tamer Rabie¹ · Mohammed Baziyad¹ · Walid Balid²

Tamer Rabie
trabie@sharjah.ac.ae

Mohammed Baziyad
mbaziyad@sharjah.ac.ae

Walid Balid
walid@ou.edu

¹ Department of Electrical and Computer Engineering, University of Sharjah, Sharjah, UAE

² Department of Electrical and Computer Engineering, University of Oklahoma, Norman, OK, USA