Multimedia Tools and Applications (2023) 82:26379–26409
https://doi.org/10.1007/s11042-023-14433-x

New combined DT-CWT and HOG descriptor for static and dynamic hand gesture recognition

Salah Eddine Agab[1] · Fatma Zohra Chelali[1]

Received: 23 November 2021 / Revised: 23 February 2022 / Accepted: 22 January 2023 /
Published online: 31 January 2023

Abstract

In recent years, researchers have been focusing on developing Human-Computer Interfaces that are fast, intuitive, and allow direct interaction with the computing environment. One of the most natural ways of communication is hand gestures. In this context, many systems were developed to recognize hand gestures using numerous vision-based techniques, these systems are highly affected by acquisition constraints, such as resolution, noise, lighting condition, hand shape, and pose. To enhance the performance under such constraints, we propose a static and dynamic hand gesture recognition system, which utilizes the Dual-Tree Complex Wavelet Transform to produce an approximation image characterized by less noise and redundancy. Subsequently, the Histogram of Oriented Gradients is applied to the resulting image to extract relevant information and produce a compact features vector. For classification, we compare the performance of three Artificial Neural Networks, namely, MLP, PNN, and RBNN. Random Decision Forest and SVM classifiers are also used to ameliorate the efficiency of our system. Experimental evaluation is performed on four datasets composed of alphabet signs and dynamic gestures. The obtained results demonstrate the efficiency of the combined features, for which the achieved recognition rates were comparable to the state-of-the-art.

Keywords American sign language · Arabic sign language · Artificial neural network · DT-CWT · Gesture recognition · Random Forest · SVM

✉ Salah Eddine Agab
agab11@hotmail.fr

Fatma Zohra Chelali
Chelali_zohra@yahoo.fr

[1] Speech communication and signal processing laboratory, Faculty of Electrical Engineering, University of Sciences and Technology Houari Boumediene (USTHB), Box n°: 32 El Alia, 16111 Algiers, Algeria

1 Introduction

Hand gesture recognition has become a very active research area in machine learning and computer vision [68], it allows natural and simple interaction with machines to make human-computer interaction (HCI) very similar to human-to-human interaction [59]. In addition, hand gesture recognition can be used in many applications, such as:

- Developing devices to help deaf people who use sign language as their normal and natural way of communication [36].
- Robot navigation and robot control in an indoor environment by capturing hand gestures from a camera placed on the robot, and processing the data on a computer [12].
- Game interfaces which allow controlling avatars and cars using gestures, for example, the Kinect camera introduced by Microsoft [31].
- Interactive and teaching systems, for example, a multimedia system developed in [52] teaches the user how to make Tortellini by imitating the instructions of a real cook. This system is able to estimate the user's accuracy in mimicking the correct actions and return positive or negative feedback.

Hand gesture recognition depends on non-vision based or vision-based approaches, the first category utilizes a set of sensors attached to a glove to analyze the position of the hand and the flexion of the fingers, these methods usually provide better characterization of the hand shape with few parameters, since they detect the correct coordinates of the palm and fingers. Furthermore, the gesture is recognized regardless of the background clutter [5, 43]. However, these methods are difficult to use and implement in real-life scenarios, as they require the user to wear a glove, which may be uncomfortable for a long-time use [48].

The second category is based on the use of a camera to capture the gesture, collected data is then transferred to a computer for further analysis such as hand detection and features extraction [47]. Different types of cameras can be used, such as:

- Kinect camera, which is composed of a depth camera, Infrared Emitter (IR), and a color sensor. Depth images are usually used, as they provide information about distances in the scene allowing for better hand detection and tracking. However, the resulting images have low resolution, poor texture and edge information [44, 57].
- Leap Motion Controller (LMC) which is composed of two IR cameras and three IR lights, this sensor provides a precise skeletal representation of the hand. However, the palm should be close and perpendicular to the sensor. Moreover, the IR camera's field of view is around 150°, which introduces strong distortions in the resulting image [32, 35].
- RGB cameras, which capture high resolution and colored images characterized by rich texture and edge information. In addition, RGB cameras are widely available in smartphones and laptops, which makes them suitable and easier to use as they do not require any additional hardware cost. However, these sensors are highly affected by illumination changes and background clutter. Therefore, robust, and invariant features are required to characterize the acquired images [4, 40].

Vision-based gesture recognition is more natural and intuitive, as it does not require the use of any additional hardware. Furthermore, during the current COVID-19 pandemic, where the least interaction with devices and surfaces is necessary to prevent the spread of the virus.

Researchers are developing vision-based recognition systems to reduce physical contact between humans and commonly used devices, for example, implementing a gesture recognition system to control the computer wirelessly [60], or utilizing hand gestures to command an elevator [23].

Gestures can be characterized in the spatial domain in their static form, also known as posture, this form requires less computational complexity, as the gesture is only represented by one image, whereas the dynamic form, called gesture, is characterized in the temporal domain, since it represents a sequence of postures [26].

Different methods have been proposed to characterize and classify gestures in both forms. In [46], authors used a hybrid algorithm with Discrete Wavelet Transform (DWT) and Local Binary Pattern (LBP) to characterize a set of dynamic gestures representing the Indian sign language, obtained results using Adaboost show good recognition rates for which 90.28% accuracy was achieved. In [22], authors proposed a system capable of recognizing a set of static gestures from the Persian sign language (PSL) based on the 7th decomposition level or higher of DWT using Haar wavelet which results in a very compact feature vector, this system achieves 83% accuracy rate using the Multilayer Perceptron (MLP). In [65], a combination of DWT and MLP was used to recognize 24 alphabets from the American Sign Language excluding letters J and Z, this system was able to recognize 97% of the test images. Authors in [49] proposed a new method based on Wavelet Neural Network (WNN), which is an Artificial Neural Network (ANN) with a wavelet-based function as the transfer function, the proposed method gave very satisfying results. In [64], authors proposed a novel method to recognize gestures from American Sign Language by applying edge detection technique using Sobel filter followed by 2D DWT to each image from the dataset, the resulting 2D images were transformed to 1D using the ring projection method followed by 1D DWT, resulting vectors are concatenated to define the feature vector. Generalized Regression Neural Network (GRNN) was used for classification and achieved an accuracy rate of 90.44% on the Massey University dataset. Another gesture recognition system based on DWT and Fisher ratio for feature extraction was proposed in [56], experiments were conducted on two different datasets, first is Massey University dataset where the achieved accuracy rate was around 98%, and second is Jochen Triesch dataset where the accuracy rate is around 95%.

In [17], authors developed a system to translate gestures from the Arabic Sign Language (ArSL) to text using two methods for characterization, namely, Fast Wavelet Network Classifier (FWNC) and Separator Wavelet Network Classifier (SWNC), results showed that FWNC performed better than SWNC. In [7], authors used DWT decomposition for feature extraction and Hidden Markov Model (HMM) or K-Nearest Neighbor (KNN) as classifiers, results show that 2nd level decomposition combined with db5 wavelet provided better results overall.

In [62], authors implemented a deep learning-based recognition system that could differentiate between 10 digits, this system utilizes Haar features and Adaboost classifier to perform hand segmentation in real-time and CamShift algorithm for hand tracking. Furthermore, Convolutional Neural Network (CNN) was implemented for gesture characterization and classification, this system achieved a 98.3% recognition rate. A similar gesture recognition system was implemented in [50], where Haar features and CamShift were utilized for real-time hand gesture segmentation and tracking. However, the recognition is performed by tracking the number of defects generated by the hand, authors evaluated the efficiency of the system in real-life scenarios and achieved satisfying results. In [58], the authors proposed a static hand gesture recognition system based on pre-trained CNN for hand characterization and classification. To test the efficiency of this system, the authors recorded a user-independent dataset composed of 1000 RGB images and 1000 depth maps collected using an Intel RealSense camera, simulation results illustrate that a 99% recognition rate is achieved. Authors in [33] explored multimodal fusion using Convolutional Recurrent Neural Network (CRNN), to combine depth information with 2D skeleton coordinates, this system was tested on two existing datasets and achieved comparable results to previous works.

In [69], authors proposed a new characterization approach that combined HOG and uniform LBP descriptors, this new descriptor extracts the hand shape using HOG, then LBP is applied on the resulting image to characterize the hand textures, this system was tested on NUS hand posture dataset where 97.8% recognition rate was achieved using LIBSVM classifier. An improved version of this descriptor was later developed in [42], where the LBP descriptor is extended to use multi-blocks by calculating the average gray-scale value for each sub-block before applying LBP, this system was tested on Jochen Triesch dataset where 98% recognition rate was achieved using SVM classifier.

This paper introduces a vision-based hand gesture recognition system that is able to recognize gestures in both forms, where static form represents Arabic and American sign language alphabets, and dynamic form is represented by two datasets known as Cambridge and Marcel, these datasets describe gestures with different shapes and motions. Gestures used in our study were recorded in challenging conditions, such as different lighting, low quality, and resolution. Furthermore, the proposed system is user-independent, meaning that the gesture is recognized even when the user changes. Our work considers different architectures to implement the gesture recognition system, where the first architecture applies individual descriptors such as Discrete Wavelet Transform (DWT), Dual Tree Complex Wavelet Transform (DT-CWT) as well as the Histogram of Oriented Gradients (HOG), to characterize the gestures. Whereas, the second architecture combines two individual descriptors to form a new feature vector, namely DWT + HOG and DT-CWT + HOG. The resulting individual and combined features are fed to five classifiers to achieve better performance in terms of recognition rate and processing time.

The rest of this paper is divided into 5 sections, where the architecture of the proposed hand gesture recognition system is presented in section 2. Section 3 details the descriptors used for features extraction; the classification phase is presented in section 4. Obtained results are discussed in section 5. Finally, section 6 concludes the paper.

2 Proposed sign alphabet and hand gesture recognition system

2.1 Proposed approach

In this paper, we investigate the performance of two wavelet-based algorithms, as well as a textural-based descriptor applied to hand images for the recognition of static postures and dynamic gestures. The employed descriptors are:

- Discrete Wavelet Transform (DWT): allows image analysis at different resolutions and reduces the size of an image without affecting the details and edges.
- Dual Tree Complex Wavelet Transform (DT-CWT): was initially developed to overcome the DWT’s limitations. Although this descriptor has been implemented in previous works for face recognition and image compression, it has never been used in gesture recognition.

- Histogram of Oriented Gradients (HOG): is a widely known descriptor that is able to characterize the shape of the articulated gestures successfully and efficiently.

For the classification task, we propose five classifiers: Multilayer Perceptron (MLP), Probabilistic Neural Network (PNN), Radial Basis Neural Network (RBNN), Support Vector Machine (SVM), and Random Forest.

Our main contributions consist of:

- Describing the benefits of DT-CWT such as shift-invariance, better directionality, and noise reduction compared to the regular DWT for hand gesture characterization;
- Proposing a new descriptor known as the combination of HOG and DT-CWT, where HOG is applied on the imaginary part of the approximation image obtained from DT-CWT, so that the result vector is compact, more discriminant, and less redundant, which allows for accurate and fast recognition for both static and dynamic gestures.
- Comparing the efficiency of several classifiers: MLP, PNN, RBNN, Random Forest, and SVM for gesture recognition system, where we demonstrate the efficiency of the Conjugate gradient with Powell-Beale restarts for MLP classifier, as well as, the Extremely randomized trees approach for Random Decision Forest.

The first step in our recognition system concerns the hand image characterization using various descriptors, namely DWT, DT-CWT, HOG, and the combined DWT + HOG, and DT-CWT + HOG. Each descriptor is applied to each image from ASL and ArSL where the resulting histogram represents the feature vector. Whereas for Marcel and Cambridge datasets, the gestures are defined by video sequences, where each sequence is composed of numerous frames. The descriptors are applied to each frame individually and the feature vector is defined by the concatenation of all individual vectors.

The second step represents the classification phase, where extracted features from each dataset are divided into two sets. Training data are learned by all classifiers (MLP, PNN, RBNN, SVM, and Random Forest) to build the predicting models. Testing data are then used to compute the performance of our proposed system in terms of recognition rate and time processing. Figure 1 shows a block diagram describing the general architecture of the proposed system:

2.2 Datasets description

To evaluate the performance of the proposed system, four different datasets were used, two datasets contain static alphabet signs and the other two are composed of dynamic gestures. We present, in this section, the different datasets used in our work.

- The first dataset is known as Arabic Sign Language dataset, it is composed of 30 Arabic signs; each sign is recorded 60 times by a different person, each image from this dataset has a different resolution, we choose to resize all images to the same resolution of 128 × 128 pixels [3].
- The second dataset is Jochen Triesch's dataset, it is composed of 10 American alphabet signs recorded against uniform and complex backgrounds; these postures were realized by 24 persons, and all images have the same resolution of 128 × 128 pixels [66].

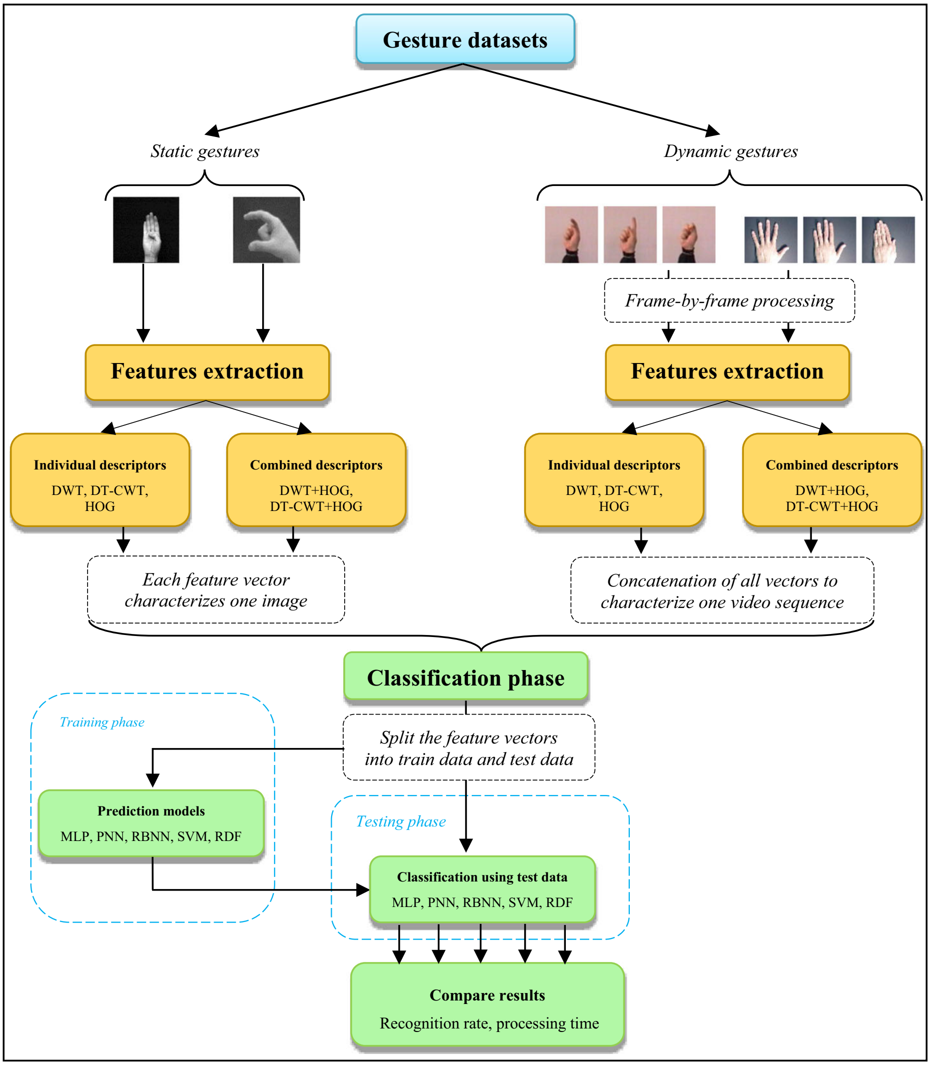
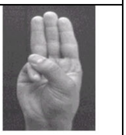
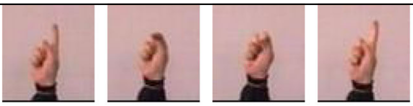


Fig. 1 Block diagram presenting different blocks of our hand gesture recognition system

- The third dataset is Marcel’s dynamic dataset which is composed of four dynamic gestures (Rotate, Stop, No, Clic), each gesture is represented by 12 sequences, and each sequence contains 55 frames with a resolution of 58 × 62 pixels (https://www.idiap.ch/resource/gestures/).
- The fourth dataset consists of 9 dynamic gestures defined by three different hand shapes (Flat, Spread, V-Shape) and three motions (Leftward, Rightward, Contract), each gesture was repeated 10 times by 2 subjects using 5 different illumination conditions, which results in a total of 100 video sequences for each class [27].

Table 1 shows a few images from the datasets used in this paper.

Table 1 Images from each dataset

Dataset		Resolution	Images
Static gestures	Arabic Sign Language (Al-Jarrah dataset) [34]	128×128	Alef أ Ba ب Ta ت  Tha ث
	American Sign Language (Triesch dataset) [35]	128×128	A B C D
Dynamic gestures	Marcel's dataset	58×62	 Clic gesture No gesture
	Cambridge dataset [37]	320×240	Flat Leftward V-Shape Contract 5 illumination conditions

3 Hand gesture characterization

One of the most important operations in pattern recognition is feature extraction, which reduces the size of the raw data by eliminating irrelevant or redundant information allowing faster and better classification.

Three techniques are used in our work to characterize hand gesture images, DWT, DT-CWT, and HOG descriptors. The main advantage of this parameterization phase is to generate a feature vector that best describes our data with minimum redundancy. A combined descriptor called DT-CWT + HOG is applied using HOG and DT-CWT techniques, where the HOG descriptor is calculated from either the real part or the imaginary part resulting from DT-CWT.

Two architectures are proposed for our hand gesture recognition system, where the first architecture employs individual descriptors, and the second combines two descriptors to define a final descriptor, the following paragraph describes the details of the mathematical background for each descriptor.

3.1 Discrete wavelet transform (DWT)

The 2D DWT is a very useful tool in image processing as it allows feature extraction and analysis at different levels of resolution. Indeed, the 2D DWT filters noise and smooth areas without affecting details and edges of the image [39].

The 2D DWT decomposition is performed by passing the input image through high-pass and low-pass filters, the resulting images are downsampled with a factor of 2, then passed through the same two filters generating four sub-images: one approximation and three details, each sub-image has half the dimensions of the original. This approach is known as filter bank-implementation of DWT [39] [34]:

- **LL:** This is the approximation image which resulted from passing the image through two low-pass filters.
- **HH:** This image represents diagonal details of the input image, where both directions were extracted using two high pass filters.
- **LH:** Horizontal details are extracted using the low-pass filter followed by a high-pass filter meaning that the horizontal direction contains lower frequencies whereas the vertical direction has high frequencies.
- **HL:** This image defines vertical details which are extracted using the high-pass filter followed by the low-pass filter, in this case, the horizontal direction has high frequencies and the vertical direction has low frequencies.

Figure 2 summarizes the steps described above. The approximation image (LL1) is then decomposed to generate second level sub-images (LL2, LH2, HL2, and HH2); similarly, the third level is obtained by decomposing the image (LL2) [45].

Figure 3 shows an example of three levels of decomposition using 2D DWT.

Figure 4 shows resulting images from applying 2D DWT to an image from the American Sign Language dataset.

Figure 4 shows that the LL image has a smoother background than the input image without affecting the hand shape and details. Moreover, processing the LL image is faster due to its reduced resolution of 4096 pixels compared to 16,384 pixels for the input image. Consequently, the extracted features using the DWT descriptor are represented by the LL image.

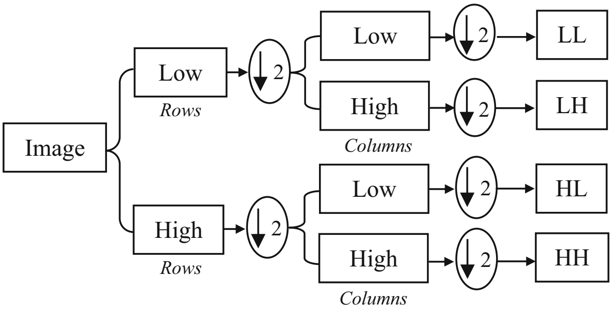


Fig. 2 Generation of the first level approximation and details from an image using 2D DWT

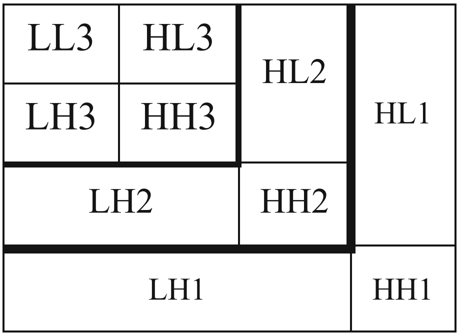


Fig. 3 Three levels decomposition using 2D DWT

3.2 Dual tree complex wavelet transform (DT-CWT)

2D DWT provides a compact representation of the image with limited redundancy and very good reconstruction, but it suffers mainly from two major drawbacks [1]:

- **Lack of shift invariance:** implies that a small translation in the input signal results in major changes in the amplitude of wavelet coefficients at different scales. This lack of shift invariance occurs from the downsampling by a factor of two [29].
- **Poor directional selectivity:** The HH image defines the details of both diagonals, this means that DWT cannot distinguish between opposing orientations [1], because the filters used are real and separable [29].

To overcome these limitations, Kingsbury introduced Dual-Tree Complex Wavelet Transform (DT-CWT) [29]. To solve the problem of shift invariance, the downsampling is removed after the first level which means that two trees are obtained, where one tree is one sample offset from the other as shown in Fig. 5. Also, Kingsbury found that using odd-length filters in the first tree and even-length in the second provides uniform intervals between trees, the filters used are usually chosen from the biorthogonal set because their impulse responses are similar to the real and imaginary parts of a complex wavelet [29].

To perform DT-CWT in 2D, the image is filtered along columns followed by rows, the filter used for rows is the complex conjugate of the first filter [29]. Starting at the first level of decomposition, the 2D DT-CWT transform produces four trees (A, B, C, and D) as shown in

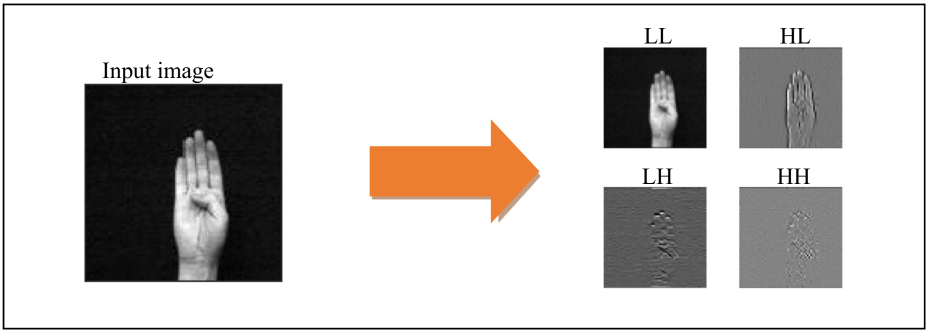


Fig. 4 First level decomposition using 2D DWT

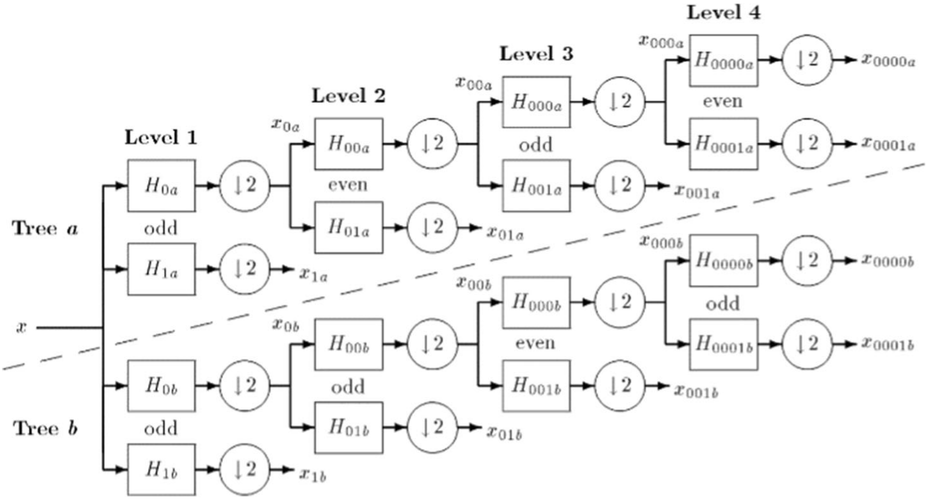


Fig. 5 Four levels decomposition of a 1D signal using DT-CWT [29]

(1), where a^0 represents the input image, h^0 and g^0 are the odd-length filters, a defines the approximation image and d defines the detail image [20]:

$$\begin{array}{lllll}
\text{Tree T} & \text{A} & \text{B} & C & \text{D} \\
\left(a_T^1\right)_{x,y} & \left(a^{\circ}*h^{\circ}h^{\circ}\right)_{2x,\ 2y} & \left(a^{\circ}*h^{\circ}h^{\circ}\right)_{2x+1,\ 2y} & \left(a^{\circ}*h^{\circ}h^{\circ}\right)_{2x+1,\ 2y} & \left(a^{\circ}*h^{\circ}h^{\circ}\right)_{2x+1,\ 2y+1} \\
\left(d_T^{1,1}\right)_{x,y} & \left(a^{\circ}*g^{\circ}h^{\circ}\right)_{2x,\ 2y} & \left(a^{\circ}*g^{\circ}h^{\circ}\right)_{2x+1,\ 2y} & \left(a^{\circ}*g^{\circ}h^{\circ}\right)_{2x+1,\ 2y} & \left(a^{\circ}*g^{\circ}h^{\circ}\right)_{2x+1,\ 2y+1} \\
\left(d_T^{1,2}\right)_{x,y} & \left(a^{\circ}*h^{\circ}g^{\circ}\right)_{2x,\ 2y} & \left(a^{\circ}*h^{\circ}g^{\circ}\right)_{2x+1,\ 2y} & \left(a^{\circ}*h^{\circ}g^{\circ}\right)_{2x+1,\ 2y} & \left(a^{\circ}*h^{\circ}g^{\circ}\right)_{2x+1,\ 2y+1} \\
\left(d_T^{1,3}\right)_{x,y} & \left(a^{\circ}*g^{\circ}g^{\circ}\right)_{2x,\ 2y} & \left(a^{\circ}*g^{\circ}g^{\circ}\right)_{2x+1,\ 2y} & \left(a^{\circ}*g^{\circ}g^{\circ}\right)_{2x+1,\ 2y} & \left(a^{\circ}*g^{\circ}g^{\circ}\right)_{2x+1,\ 2y+1}
\end{array} \tag{1}$$

For other levels (m > 1), even-length filters are added to the transform, to achieve shift invariance, odd-length filters (h^0, g^0) are used in one tree, whereas even-length (h^e, g^e) are used in the other tree giving us 4 possible combinations each one defines a different tree. Complex coefficients of 2D DT-CWT are formed from combining detail coefficients of different trees as shown in (2) [20]:

$$\begin{aligned}
z_{+}^{j,k} &= \left(d_A^{j,k} - d_D^{j,k}\right) + j\left(d_B^{j,k} + d_C^{j,k}\right) \\
z_{-}^{j,k} &= \left(d_A^{j,k} + d_D^{j,k}\right) + j\left(d_B^{j,k} - d_C^{j,k}\right)
\end{aligned} \tag{2}$$

Where j represents the scale, k = 1,2,3 are the detail coefficients following three different directions, and A, B, C, D are the trees.

For a given scale, the 2D DT-CWT transform produces six images of complex coefficients oriented at ±15°, ±45°, and ± 75°. Figure 6 shows a comparison between the details produced by this transform and details obtained using the real 2D DWT [29].

We notice that DT-CWT is able to separate 6 directions compared to 3 for DWT, which allows for better selectivity and representation for the oriented textures. Figure 7 shows the

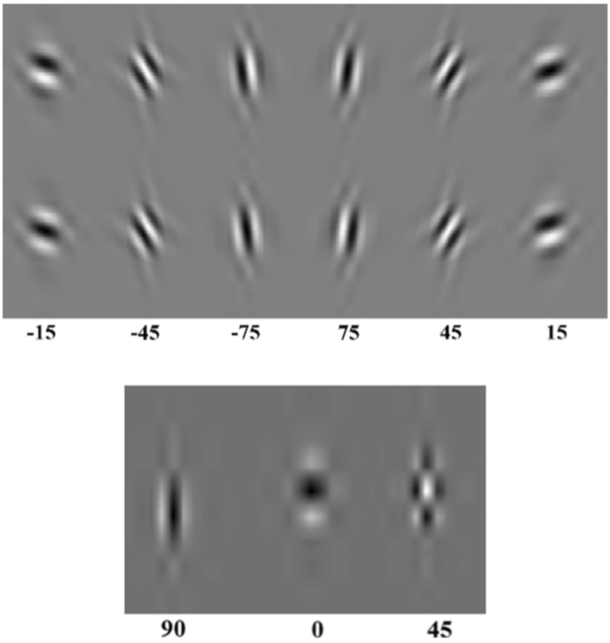


Fig. 6 2D impulse responses of the DTCWT at level 4 (upper part) and equivalent responses of real DWT (lower part) [29]

improvement in directional selectivity provided by 2D DT-CWT compared to real 2D DWT; we notice that the DWT was unable to distinguish between opposing diagonals of the image whereas the 2D DT-CWT can detect both diagonal directions separately.

The odd/even filter implementation of the DT-CWT suffers from a few problems such as a non-symmetric sub-sampling structure, different frequency responses between trees and the used filters must be bi-orthogonal. To overcome these limitations, a new approach was proposed in [30] known as the Q-shift implementation of DT-CWT, which uses odd-length filters in the first level, and even-length for remaining levels. The delay of half sample is reached using the time reverse of the first tree filters in the second tree [30].

The filters of the first tree are designed using a low pass FIR filter $H_{L2}(z)$ of length 4n:

$$\mathrm{H}_{\mathrm{L2}}(z) = H_{\mathrm{L}}\left(z^{2}\right) + z^{-1} H_{\mathrm{L}}\left(z^{-2}\right) \tag{3}$$

Where $H_L(z)$ contains coefficients from z^{n-1} to z^{-n}.

The obtained filters for different values of n are shown in Table 2.

Figure 8 shows the result of applying 2D DT-CWT to the same image used for the 2D DWT example in Fig. 4, we notice that 2D DT-CWT extracts more information compared to the 2D DWT.

When comparing Figs. 4 and 8, we notice that DT-CWT provides a real and imaginary approximations per tree, whereas the regular DWT only provides a single approximation. In addition, DWT only produces three detail images oriented at (0°, 45°, and 90°), which is limited compared to DT-CWT that produces 6 real and imaginary details, this results in a smoother approximation image when using DT-CWT while keeping the same resolution as the approximation image using DWT.

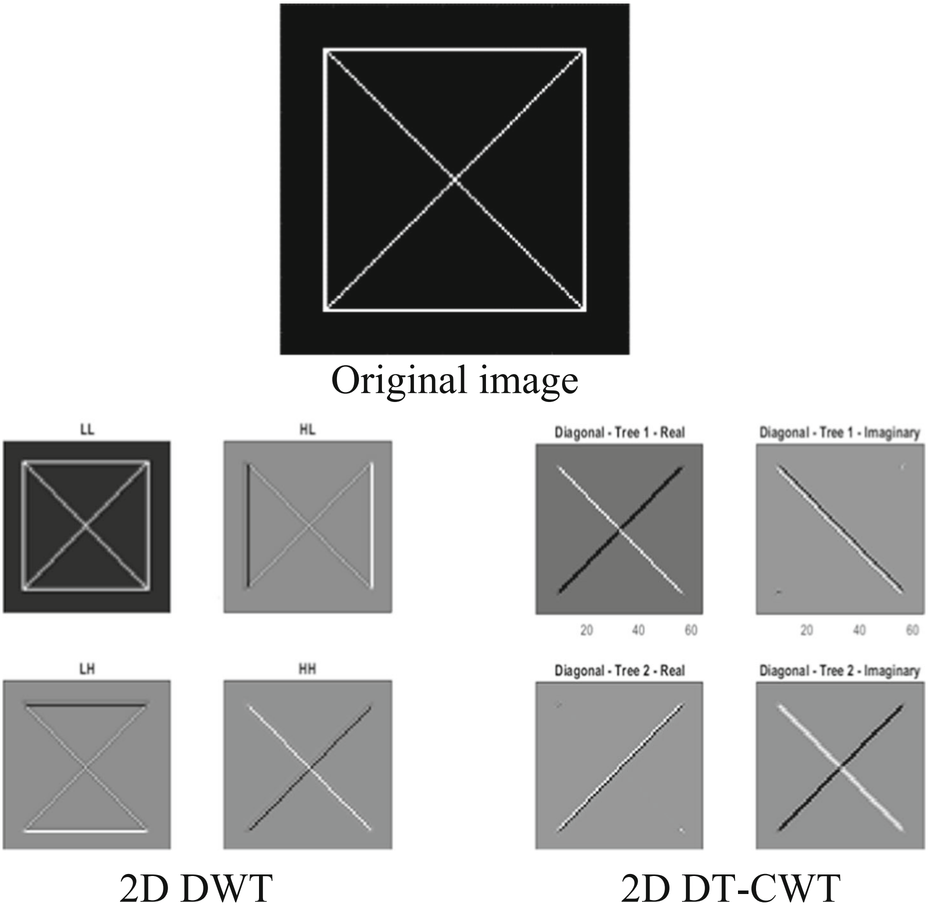


Fig. 7 Directional selectivity of the real 2D DWT compared to 2D DT-CWT

3.3 Histogram of oriented gradients (HOG)

HOG descriptor was first presented in 2005 by Dalal and Triggs. The HOG descriptor characterizes the local structure or the shape of an object by extracting the magnitude and direction of edges without having additional information on edge positions [11].

The first step when calculating HOG descriptor is to compute the gradient of the image, this is done by convolving the input image with a derivative mask in both horizontal and vertical directions, the mask is either a 1D kernel such as [−1,1] or a 2D kernel such as Sobel filter [11].

$$\begin{cases} G_x = \text{Kernel} * \text{Image} \\ G_y = \text{Kernel}^T * \text{Image} \end{cases} \tag{4}$$

The resulting horizontal and vertical gradients are then combined to define the magnitude 'G' and orientation 'θ' of the gradient, as shown in (5):

$$\begin{cases} G = \sqrt{G_x^2 + G_y^2} \\ \theta = \arctan\left(G_y / G_x\right) \end{cases} \tag{5}$$

Table 2 Odd-Length filters for n = 5, 7, and 9 [30]

n=5	n=7	n=9
0.03516384	0.003253142763653	−0.002284127440271
0	−0.003883211999158	0.001209894163073
−0.08832942	0.034660346844853	−0.011834794515431
0.23389032	−0.038872801268828	0.001283456999344
0.76027237	−0.117203887699115	0.044365221606617
0.5875183	0.275295384668882	−0.053276108803047
0	0.756145643892522	−0.113305886362143
−0.11430184	0.568810420712123	0.280902863222186
0	0.011866092033797	0.752816038087856
0	−0.106711804686665	0.565808067396459
	0.02382538479492	0.024550152433667
	0.017025223881554	−0.120188544710795
	−0.005439475937274	0.018156493945546
	−0.004556895628475	0.031526377122085
		−0.00662879461243
		−0.00257617430660
		0.001277558653807
		0.002411869456666

The gradient image is divided into N × N regions called "cells", all pixels are weighted by the magnitude and orientation of the gradient, for each cell, a histogram is computed as follow [11]:

$$H(\alpha) = H(\alpha) + N_G(x, y) \tag{6}$$

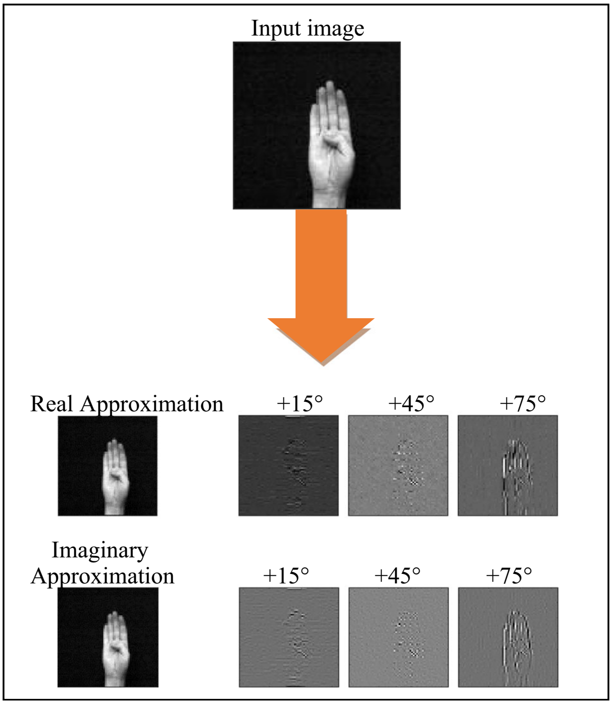


Fig. 8 First level decomposition using 2D DT-CWT

Where N_G is the pixel's intensity and α defines the number of bins in each histogram, this parameter is used to compute the angle step which is uniformly distributed in the range [−180°, 180° [, the angle step is defined as 360°/α.

The HOG feature vector represents the concatenation of the histograms extracted from all cells since there are N × N cells, and each cell is defined by a histogram, therefore, the size of the final vector is $N \times N \times \alpha$.

Finally, the L2-norm is used to normalize the HOG vector, this step is crucial as it allows better invariance to illumination changes. Normalization using the L2-norm is defined as follow:

$$V = H/\sqrt{\|H\|^2 + \varepsilon^2} \tag{7}$$

Where V represents the normalized HOG descriptor, H is the HOG vector without normalization and ε is a small constant used to avoid division by zero.

Figure 9 summarizes the computation steps of the HOG descriptor applied to a sequence of 55 frames of Clic gesture from the dynamic dataset (https://www.idiap.ch/resource/gestures/), HOG descriptor is applied to each frame of the video sequence following the steps described above, then, all 55 histograms are concatenated to build HOG descriptor of the video sequence.

The resulting concatenated vector $HOG_{i\,(i\,=\,1:\,M)}$, where M is the total number of frames in the video sequence, will describe and reduce the amount of information contained in the hand gesture. We note that the concatenated histogram is designed as the total descriptor for the whole video sequence and its size is $N \times N \times \alpha \times M$.

4 Classification phase

The classification task consists of finding the appropriate class for each posture or gesture. For this purpose, five different classifiers were used: three variants of the Artificial Neural Network, which are Probabilistic Neural Network (PNN), Radial Basis Neural Network (RBNN), and Multilayer Perceptron (MLP), as well as, Support Vector Machine (SVM) classifier and Random Forest. In this section, we will briefly present each one of these classifiers.

4.1 Multilayer perceptron (MLP)

The MLP is a feed-forward Artificial Neural Network that consists of three layers, an input layer containing a fixed number of neurons, one or multiple hidden layers containing a variable number of neurons where each represents a nonlinear activation function, and an output layer where the number of its neurons is equal to the number of classes.

The training of MLP starts with initializing the weights of neurons with random values, then, the input data is propagated throughout the network, the output layer computes the mean squared error (MSE) between desired results and the outputs given by the network [19]:

$$E(n) = \frac{1}{2}\sum_{j=1}^{m}\left(d_j(n) - y_j(n)\right)^2 \tag{8}$$

where m represents the total number of output neurons, n is the current training sample, d and y are, respectively, the desired output and the actual output.

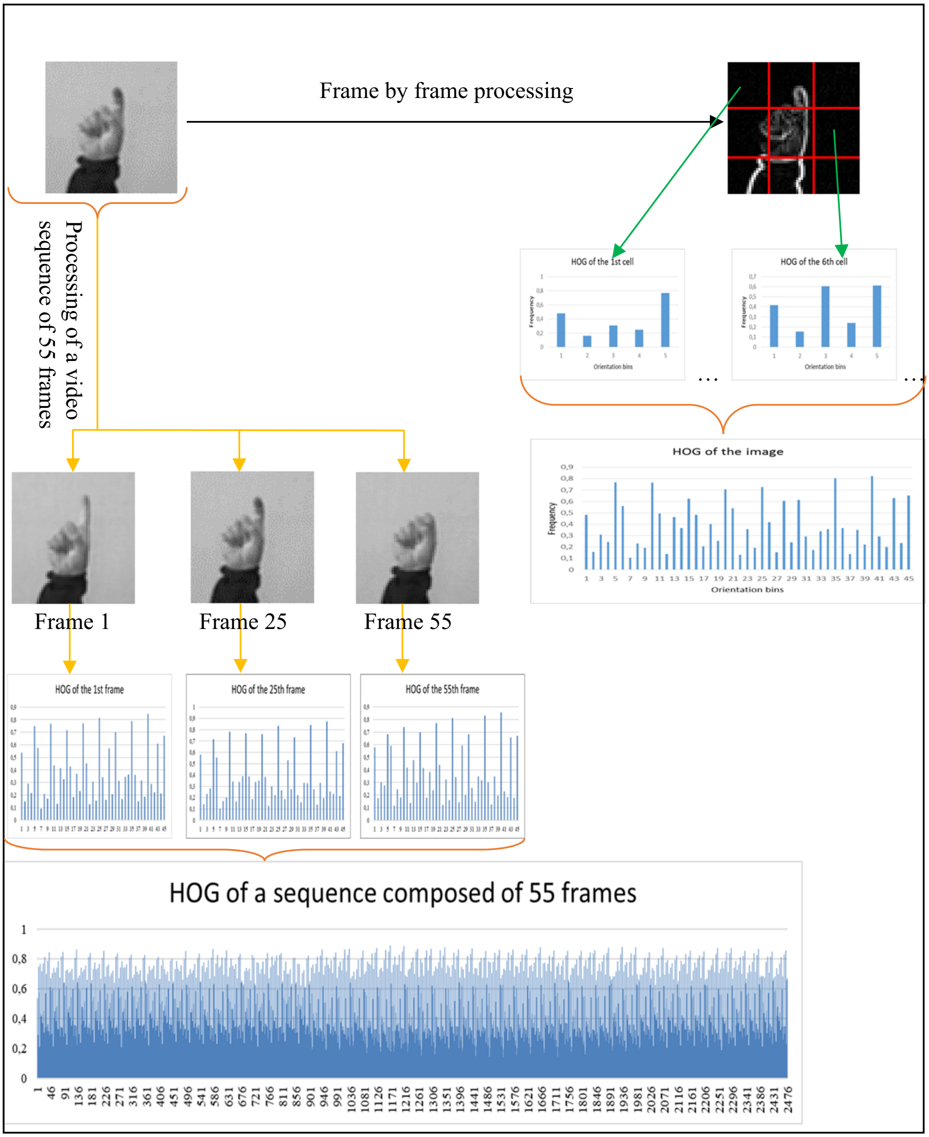


Fig. 9 HOG feature vector computation

The backpropagation approach is used to minimize this error by readjusting weights and biases, each neuron sums the inputs with new weights to give a new output that is closer to the desired output.

4.2 Radial basis neural network (RBNN)

RBNN is also a feed-forward neural network, but unlike MLP, RBNN consists of a single hidden layer composed of nonlinear radial basis functions (RBF) such as a Gaussian function.

Each neuron computes the distance between the center of the function and input data, the output value of the neuron is bigger for smaller distances [13].

Training of RBNN consists generally of two steps, in the first one, the width and center of the Gaussian kernel are determined using unsupervised methods. In the second step, the weights are computed using a supervised approach [13].

4.3 Probabilistic neural network (PNN)

PNN classifier represents an implementation of the kernel discriminate analysis and the Bayesian network, which devises a family of probability density function estimators [38] [61]. The PNN has four layers, where the input and output layers are similar to the previous ANN [61]. The second layer is called the pattern layer, it contains one neuron for each training sample, each neuron computes the distance between the input data, and its dedicated pattern, then, a nonlinear activation function is applied to the distance; pattern units will pass these results to the third layer known as the summation layer. Each class has a dedicated neuron that averages results coming from the pattern layer; the neuron with the largest result defines the class of the input vector [38].

4.4 Support vector machine (SVM)

The main idea of the SVM is to design an optimal hyperplane that classifies data in two classes, but when working with complex problems such as pattern recognition, a linear separation is usually hard or even impossible to achieve. Therefore, a kernel is adopted to map the data in a higher dimensional space [8]. In this paper, the Radial Basis Function (RBF) kernel was used since it provides good performance in terms of accuracy and processing time.

The One-against-all approach was employed to solve multi-class problems with SVM, this method consists of using M-binary classifiers for M-class problems, meaning that, one binary SVM classifier is constructed per class, each SVM classifier is trained to distinguish one class samples from the remaining samples [18].

4.5 Random decision Forest

Random Decision Forest is widely used in classification and regression tasks due to its simplicity and speed; the forest includes a multitude of decision tree classifiers where the training of each classifier starts at the root and is done on randomly selected samples from training data. When the tree grows, each leaf node at its end will represent a class [6].

When training is complete, each tree assigns a class to the new observation from test data, the predicted class given by Random Forest is then determined using a majority vote [6].

An extension to this method known as Extremely Randomized trees or Extra-Trees was later proposed in [16], and it has been demonstrated that this method outperforms other randomized algorithms in both accuracy and efficiency [14]. The main difference between Random Forest and Extra-Trees lies in the fact that the features and splits are selected at random, which leads to many diversified trees [16].

5 Simulation results

We present in this section the simulation results of our hand gesture recognition system for static and dynamic forms, for this purpose, we investigate two architectures. The first architecture consists of characterizing each posture from the static and dynamic datasets using DWT, DT-CWT, and HOG descriptors separately with different classifiers such as MLP, PNN, RBNN, Random Forest, or SVM classifiers to demonstrate the efficiency of our hand gesture recognition system. Whereas, the second architecture investigates a technique known as feature fusion, which consists of combining individual descriptors by applying HOG descriptor to the approximation image from DWT or DT-CWT to ameliorate the recognition rate, namely: DWT + HOG and DTCWT+HOG. For DWT, the approximation image is extracted as shown in Fig. 4, and for DT-CWT, two approximation images are produced representing the real and imaginary parts as shown in Fig. 8. Then, the HOG descriptor is applied to these images as illustrated in Fig. 9.

5.1 The implemented technique

We described in section 3 the techniques used for feature extraction, namely DWT, DT-CWT, and HOG. These descriptors were applied to all datasets, then 50% of the resulting features were used for training the classifiers and the remaining 50% were used to test their efficiency. We note that the percentage of training and testing varied from one researcher to another. Consequently, building a real comparison is a challenging task.

The parameters of all descriptors were tuned to optimize their configuration, where:

- For the DWT, we adopted two wavelet families, which are Daubechies wavelets (db_1, db_2, db_4, db_8, db_{10}) and Biorthogonal wavelets ($bior_{1.1}$, $bior_{1.3}$, $bior_{2.2}$, $bior_{2.4}$, $bior_{3.5}$), for each wavelet, we investigated decompositions up to the third level, and the feature vector represents the approximation image.
- For the DT-CWT, we used the filters described in Table 2 with three level decompositions, the resulting feature vector is either the real part or the imaginary part. Several tests were carried out to demonstrate the efficiency of each component, i.e., the real and the imaginary parts according to recognition results obtained.
- For the HOG descriptor, different values of cells (1:10) and bins (1:20) were investigated to find the combination that provides the best recognition rate and classification speed.

Once the best topology for each descriptor is determined, different parameters for each classifier are investigated to improve the performance of our system, for the MLP, different training functions, as well as different numbers of neurons, were investigated. For the RBNN, PNN, and SVM, we varied the spread of radial function. Moreover, for Random Decision Forest, we explored multiple numbers of Decision Trees so that accuracy can be improved.

5.2 Descriptors parametrization

In this section, we investigate the best topology for each descriptor. For this purpose, we applied each descriptor to the ASL dataset with black background, and we used MLP as a classifier. We use the black background for parametrization due to its low resolution and noisy background.

For the DWT descriptor, we use different wavelets and 3 decomposition levels as shown in Fig. 10.

Figure 10 shows that the third level decomposition computed for either db_1 or $bior_{1.1}$ wavelets performs better in terms of recognition rate where 57% was achieved. Moreover, the classification time is reduced significantly when using the third level decomposition since the resulting image, in this case, has been down sampled by a factor of 2^3.

For the DT-CWT, recognition rates are compared when applying filters shown in Table 2; two even-length filters are included to show that results given by odd-length filters are better. Moreover, using either the real part or imaginary part of approximation image resulting from the third level decomposition is investigated, obtained recognition rates are presented in Fig. 11.

Figure 11 shows that odd-length filters perform better than even-length, in addition, the third set of filters ($n = 7$) achieves the highest recognition rates of 56% and 57% for the real and imaginary parts.

For the HOG descriptor, the image is divided into N-by-N cells and each cell is characterized by an $\propto$ bins histogram, providing a final vector of $N \times N \times \propto$, the purpose is to find the smallest values of N and $\propto$ that allow the best characterization of the image. Experiments show that good recognition rates were obtained when using 4 × 4 cells and at least 10 bins, where 80% recognition rate was achieved.

Finally, we investigate the effectiveness of our gesture recognition system using the HOG technique applied for the approximation image given by DWT and imaginary part of DT-CWT transforms as indicated above. Since the size of the approximation image is reduced by a quarter at each level, we expect the HOG descriptor to require less time and fewer parameters to characterize the images. Experiments show that the combined DWT + HOG and DT-CWT + HOG improve significantly the recognition rate. Furthermore, the HOG descriptor parameters are reduced to 3 × 3 cells and 7 bins, this allows for a faster classification due to the smaller size of the descriptor vector. The best topology for each descriptor is presented in Table 3.

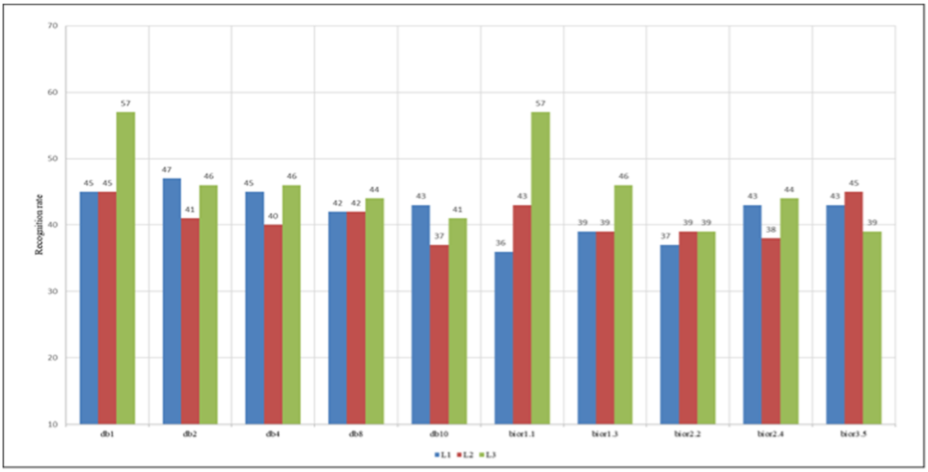


Fig. 10 Recognition rates for different wavelets and decomposition levels

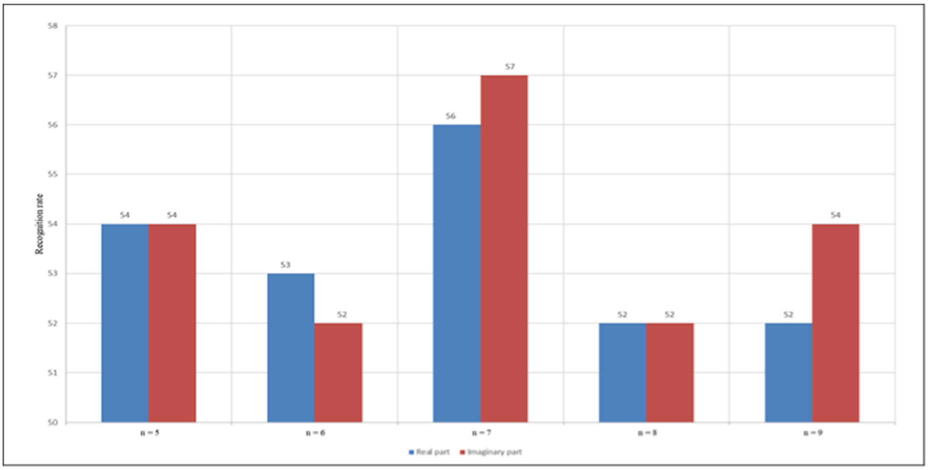


Fig. 11 Recognition rates for DT-CWT using MLP classifier

5.3 Classifiers parametrization

We conducted several experiments to determine the best configuration for each classifier, for this purpose, we used the third level decomposition of DWT on the ASL dataset with a black background; the obtained results for the MLP classifier with MSE as a loss function were as follow.

Where the training functions adopted are:

GDX: Gradient Descent with momentum and adaptive learning rate.
OSS: One-Step Secant.
RP: Resilient Backpropagation.
SCG: Scaled Conjugate Gradient.
CGF: Conjugate Gradient with Fletcher-Reeves updates.

Table 3 Best topology for each descriptor

Descriptor	Parameter	Value	Feature vector size
DWT	Wavelet	db_1	256
	Decomposition level	3rd level	
DT-CWT	Decomposition filter	N=7	256
	Decomposition level	3rd level	
	Approximation	Imaginary part	
HOG	Cells	16	160
	Bins	10	
DWT+HOG	Wavelet	db_1	63
	Decomposition level	3rd level	
	Cells	9	
	Bins	7	
DT-CWT+HOG	Decomposition filter	N=7	63
	Decomposition level	3rd level	
	Approximation	Imaginary part	
	Cells	9	
	Bins	7	

CGP: Conjugate Gradient with Polak-Ribiére updates
CGB: Conjugate Gradient with Powell-Beale restarts.

From Fig. 12, we notice that the CGB training function gave better recognition rates compared to other functions, this is due to Powell-Beale approach, which consists of restarting the search direction when orthogonality between consecutive gradients is lost. Therefore, the CGB function is used in the remaining tests for MLP, we also investigated different numbers of hidden layers, we found that using more than two hidden layers increases the processing time significantly without necessarily improving on recognition rate. Consequently, two hidden layers are used. Cross-entropy will also be used to compare the performance with MSE loss function.

The same procedure was applied for the remaining classifiers where, for the RBNN classifier, both spread and number of neurons in the hidden layer were varied, respectively, in the ranges [0.1 to 10] and [1 to 50]. For the PNN classifier, the spread was varied in the range [0.1 to 10].

For SVM classifier, the spread was varied in the range [0.1 to 200], whereas the regularization parameter C was fixed to 200 to avoid any misclassification in the training stage.

And finally, for Random Forest, the number of decision trees was varied from [1 to 100]. Table 4 presents the best topology of each classifier.

5.4 Simulation results for each dataset

Once the best configuration for each classifier and descriptor is defined, the performance of each configuration in terms of recognition rate and processing time is computed for each dataset, obtained results are presented in this section.

5.4.1 American sign language (Jochen Triesch's dataset)

For this dataset, 120 postures are used for training each classifier (50% of the dataset), and the remaining 120 postures are used to test the performance of each descriptor, obtained recognition rates, and are shown in Table 5.

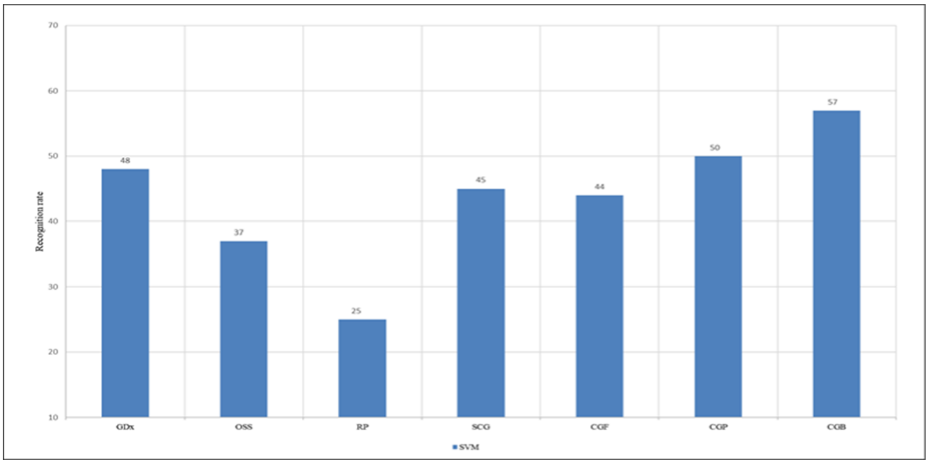


Fig. 12 Recognition rates for different training functions

Table 4 Best topology for each classifier

Classifier	Parameter	Value
MLP	Training function	Conjugate Gradient with Powell-Beale restarts
	Transfer function	Hyperbolic tangent sigmoid
	Number of hidden layers	2
	Number of neurons in each hidden layer	1 to 200
	Performance function	MSE or Cross-Entropy
	Number of epochs	500
RBNN	Number of neurons in hidden layer	1 to 50
	Spread of RBF	0.1 to 10
PNN	Spread σ	0.1 to 10
SVM	Spread σ	0.1 to 200
	Regularization parameter C	200
Random Forest	Number of decision trees	1 to 100

Simulation results show that the highest recognition rate of 99.17% was obtained for the combination of DT-CWT and HOG descriptor where 119 out of 120 samples of test data were correctly classified for the white background dataset using Random Forest classifier, in comparison, the combined DWT + HOG achieved 94.17% for the white background using Random Forest classifier.

For the black background, the highest RR of 97.5% was achieved using DT-CWT + HOG and Random Forest classifier, in comparison, DWT + HOG only achieves 90.83% RR for the same classifier and 94.17% using RBNN classifier.

Finally, we investigate the inter-class similarities by studying the confusion matrix of DT-CWT + HOG with Random Forest classifier applied to ASL dataset, we chose this specific combination since it provides an accuracy rate of 97.5% meaning that 117 of test samples were correctly characterized, and since ASL dataset is composed of 10 postures, it is easier to spot similarities between different classes, the corresponding confusion matrix is shown in Table 6.

We notice from Table 6 that 8 out of 10 postures were correctly classified, and the confusion occurs only between the letters G and H because these letters have similar corresponding gestures. As we can notice from Fig. 13, both postures are identical in hand shape and orientation, the only difference is the number of fingers, where G posture uses one finger and H posture uses two fingers.

Table 5 Recognition rates for ASL dataset

	Black background					White background				
	DWT	DT-CWT	HOG	DWT +HOG	DTCWT +HOG	DWT	DT-CWT	HOG	DWT +HOG	DTCWT +HOG
MLP_{MSE}	45	47.5	82.5	85	85	42.5	56.67	85	78.33	81.67
$\text{MLP}_{\text{Cross-Entropy}}$	61.67	61.67	85.83	88.33	86.67	65.83	65	87.5	83.33	85
RBNN	62.5	65	87.5	**94.17**	95.83	**72.5**	**75.83**	88.33	87.5	94.17
PNN	58.33	65	85.83	88.33	88.33	65.83	68.33	87.5	83.33	83.33
SVM	**67.5**	68.33	85.83	90.83	92.5	57.5	67.5	88.33	85.83	88.33
Random Forest	58.33	**70**	**88.33**	90.83	**97.5**	65.83	67.5	**90**	**94.17**	**99.17**

Table 6 Confusion matrix for DT-CWT + HOG applied to ASL dataset

	A	B	C	D	G	H	I	L	V	Y
A	12	0	0	0	0	0	0	0	0	0
B	0	12	0	0	0	0	0	0	0	0
C	0	0	12	0	0	0	0	0	0	0
D	0	0	0	12	0	0	0	0	0	0
G	0	0	0	0	11	**2**	0	0	0	0
H	0	0	0	0	**1**	10	0	0	0	0
I	0	0	0	0	0	0	12	0	0	0
L	0	0	0	0	0	0	0	12	0	0
V	0	0	0	0	0	0	0	0	12	0
Y	0	0	0	0	0	0	0	0	0	12

5.4.2 Arabic sign language

We conducted the same experiments shown previously on ArSL dataset where 50% of the total postures (900 postures) were used for training and the remaining 50% was employed to test the efficiency of each classifier, best recognition rates for each architecture are summarized in Table 7.

We notice that the combined features performed better than the individual descriptors, the best recognition rate of 94.89% was achieved using the combination of the descriptors DT-CWT + HOG for feature extraction and SVM as classifier. The Random Forest classifier did not perform well for this dataset due to the limited number of trees equal to 100. Therefore, increasing the number of trees would improve the recognition rates but it would result in a much slower model.

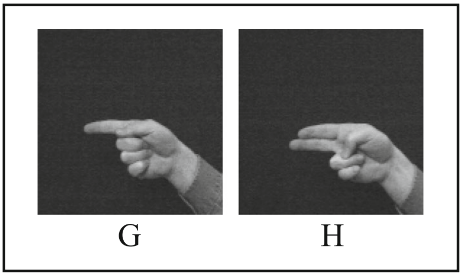


Fig. 13 Letters G and H from Triesch's dataset

Table 7 Recognition rates for ArSL dataset

	DWT	DT-CWT	HOG	DWT +HOG	DTCWT +HOG
MLP_{MSE}	61	67	85.88	88	89.11
$\text{MLP}_{\text{Cross-Entropy}}$	74.22	75.66	89.11	**88.77**	89.44
RBNN	**79.55**	**82.22**	**90.88**	86.33	93.11
PNN	75.44	76	82.55	84.55	88.66
SVM	78.22	80.78	90.22	85.11	**94.89**
Random Forest	75.44	79.55	88.77	86.77	90.44

Table 8 Recognition rates for dynamic gestures dataset

	DWT	DT-CWT	HOG	DWT +HOG	DTCWT +HOG
$\mathrm{MLP}_{\mathrm{MSE}}$	75	75	79.16	87.5	87.5
$\mathrm{MLP}_{\text{Cross-Entropy}}$	79.16	83.33	83.33	91.66	95.83
RBNN	**83.33**	83.33	87.5	**95.83**	95.83
PNN	70.83	75	75	79.16	83.33
SVM	**83.33**	83.33	87.5	91.67	95.83
Random Forest	**83.33**	**87.5**	**91.67**	**95.83**	**100**

5.4.3 Dynamic hand gesture dataset

From previous results, we conclude that the combination of each wavelet transform with HOG descriptor improved recognition rates for static postures, but since these algorithms were designed to characterize 2D images, we evaluate their performance for dynamic gestures where each sequence is defined by a set of continuous postures. The obtained results for Marcel gesture dataset are presented in Table 8, where 24 sequences were used for training and 24 sequences were used for testing.

Obtained results show that the Random Forest provided higher results than other classifiers, moreover, the combined features performed better than individual descriptors, where the combination of DT-CWT + HOG with Random Forest classifier performed best by recognizing all 24 test sequences.

5.4.4 Cambridge hand gesture dataset

For this dataset, we used the same experimental protocol described in [27], meaning that the training of all classifiers was performed on 20 sequences acquired from the first illumination setting, whereas testing was done on the 80 sequences from the remaining illumination settings. The obtained recognition rates are presented in Table 9.

Table 9 shows similar results to other datasets, where the DT-CWT + HOG performed best achieving an average recognition rate of 76.25% using RBNN classifier and 81.11% using Random Forest. Furthermore, we notice that the combined features greatly improved the recognition rates which increased from 32.11% to 69.30 for DWT and DWT + HOG, and from 36.66 to 81.11 for DT-CWT and DT-CWT + HOG.

Table 9 Recognition rates for Cambridge dataset

	DWT	DT-CWT	HOG	DWT +HOG	DTCWT +HOG
$\mathrm{MLP}_{\mathrm{MSE}}$	15.55	19.44	61.38	62.77	66.11
$\mathrm{MLP}_{\text{Cross-Entropy}}$	27.77	34.72	63.19	65.27	74.30
RBNN	29.86	31.11	**74.31**	64.72	76.25
PNN	27.77	33.88	52.22	66.66	72.91
SVM	**32.11**	**36.66**	59.44	64.72	69.58
Random Forest	29.58	29.30	65.28	**69.30**	**81.11**

5.5 Time processing

Our proposed system is composed of three main phases:

- Applying DWT or DT-CWT to the input image.
- Features extraction using HOG descriptor to create a compact vector.
- Training and gesture classification using different classifiers. For this phase, we only present the obtained results for DT-CWT + HOG since this combination provides the best recognition rates for all datasets.

Time processing for each phase was computed separately, we note that our system was implemented using MATLAB R2019b on a laptop equipped with an I3-4030u processor and 4GB of RAM. Obtained results were as follow:

- The average time required to apply DWT or DT-CWT to one image is about 0.7 s.
- Features extraction phase using HOG descriptor takes 0.5 s per image
- Processing times for gesture classification and training are summarized in Fig. 14.

Figure 14 shows that the training times obtained for SVM classifier are considerably faster than the other techniques, where for the Cambridge dataset, training SVM takes 1.17 s compared to 17 s for the Random Forest. In addition, after the training phase, the system takes 1.28 s to characterize and classify one image using SVM classifier, compared to 1.62 s for Random Forest.

5.6 Discussions

In this section, we present and discuss the best results achieved by our system based on recognition rates and processing times, which include training and testing. Figure 15 summarizes the highest RR as well as the lowest processing times achieved by the proposed system.

Simulation results show that the combined features DT-CWT + HOG with Random Forest classifier provide the highest overall RR, where 97.5% and 99.17% were achieved for

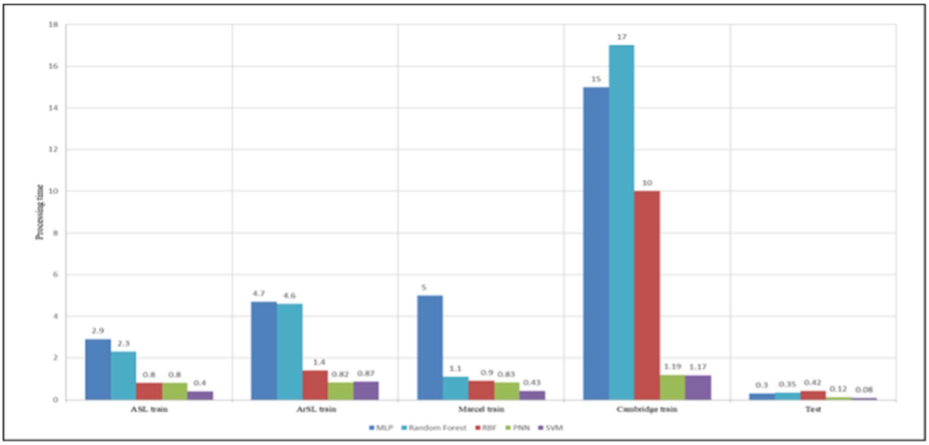


Fig. 14 Processing times for all classifiers

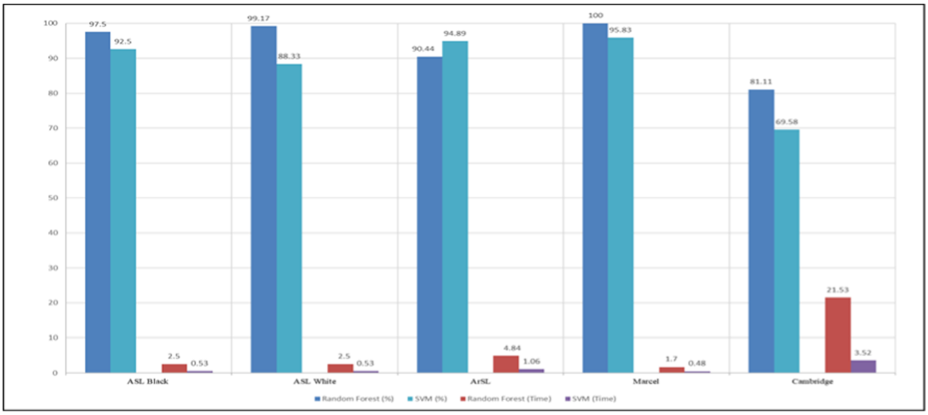


Fig. 15 Recognition rates and processing times for all datasets

Triesch's ASL dataset with white and black backgrounds. Similarly, a 100% RR was reached for Marcel's dynamic dataset and 81.11% for Cambridge dataset. However, when considering processing times, we notice that SVM classifier has better recognition timings, for Triesch's dataset, the classification phase using SVM takes 0.53 s which is 5 times faster than Random Forest classifier. For Marcel's dataset SVM classifier takes 0.48 s while Random Forest spends 1.7 s. Moreover, for Cambridge dataset, SVM classifier takes 3.52 s which is significantly faster than the 21.53 s spent by Random Forest.

These results illustrate that there is a trade-off between speed and performance between the two classifiers, Random Forest performs better in terms of RR, in contrast, SVM is faster but provides lower RR.

However, for ArSL dataset, SVM demonstrates better performance in both RR and processing times, where 94.89% RR is achieved compared to 90% for Random Forest, and the classification time is about 1 s for SVM compared to 4.84 s for Random Forest. Therefore, SVM classifier is better suited for this specific dataset, since it provides higher RR with less processing time.

Despite the satisfying results and advantages provided by the proposed system, such as the compact feature vector, feature analysis on different resolutions, noise resistance, hand shape characterization, and user independence. However, it presents some limitations, namely:

- The system robustness to illumination changes could be enhanced.
- The HOG descriptor efficiently captures the hand structure if there is no background clutter. Therefore, an accurate segmentation is required for real-life applications.
- The characterization time of 1.2 s is slow for real-time applications.
- Random Forest classifier requires many trees to achieve good performance, this results in a slower model overall.
- Our system should be improved to achieve good accuracy, especially against a complex background.

5.7 Comparison with previous works

To showcase the efficiency of the proposed method which is the combination of DT-CWT and HOG descriptors, we compared obtained recognition rates of each dataset

Table 10 Comparison of results with previous works

Reference	Dataset	Method	Training/testing percentages	RR (%)
Kaur et al. 2016 [24]	ASL	Krawtchouk features till 3rd order	50% / 50%	84.9
Sahoo et al. 2018 [56]	ASL (White Background)	DWT+Fisher ratio	Leave 4 out validation	95
Sadeddine et al. 2021 [54]	ASL (White Background)	FORI + KNN	75%/25%	98.33
	ASL (Black Background)	FGWT + KNN	75%/25%	100
Proposed Method	ASL (White Background)	***DT-CWT+HOG+Random Forest***	50% / 50%	**99.17**
	ASL (Black Background)	***DT-CWT+HOG+Random Forest***	50% / 50%	**97.5**
Al-jarrah et al. 2001 [3]	ArSL	ANFIS Network	67% / 33%	93.55
Dahmani et al. 2014 [9]	ArSL	Tchebichef moments + Hu moments	–	94.67
Sadeddine et al. 2018 [53]	ArSL	Local Binary Pattern	75% / 25%	90.41
Sadeddine et al. 2021 [54]	ArSL	FORI + KNN	75%/25%	95.83
Proposed Method	ArSL	***DT-CWT+HOG+SVM***	50% / 50%	**94.89**
Nasri et al. 2015 [41]	Marcel	Contour-Based Similarity Images	50% / 50%	85.4
Agab et al. 2019 [2]	Marcel	Edge Histogram Descriptor	50% / 50%	95.83
Proposed Method	Marcel	***DT-CWT+HOG+Random Forest***	50% / 50%	**100**
Kim et al. 2007 [28]	Cambridge	Canonical Correlation Analysis	20% / 80%	65
Kim et al. 2009 [27]	Cambridge	Tensor Canonical Correlation Analysis	20% / 80%	82
Gamal et al. 2013 [15]	Cambridge	Fourier Descriptor	50% / 50%	96
Sagayam et al. 2018 [55]	Cambridge	Hidden Markov Model + CSA	50% / 50%	81.21
Tang et al. 2019 [63]	Cambridge	Key Frames + Features Fusion	50% / 25%	98.23
Proposed Method	Cambridge	***DT-CWT+HOG+Random Forest***	20% / 80%	**81.11**
			50% / 50%	**92.89**

with previous works that used different techniques to characterize and classify the gestures.

We clarify that the percentage of training and testing data are different for the majority of works in gesture recognition while using the same dataset. Table 10 shows a comparison of the obtained results in terms of recognition rate for previous works using the same datasets while precising the employed training and testing percentages.

It is clear from Table 10, when comparing recognition rates obtained for each architecture, that our proposed method applied on Triesch's dataset, achieved a better score varying from 97.5 to 99.17%, which is motivating and encouraging compared to those obtained in [24, 35], we also notice that better results were achieved in [54] for the black background by increasing the training percentage to 75%.

For Arabic Sign language dataset, the combined DT-CWT + HOG achieved a 94.89% recognition rate which is better than the results cited in [3, 9, 53]. However, better a recognition rate of 95.83% was achieved in [54] due to employing a training percentage of 75%. Besides, a score of 100% was obtained for the dynamic dataset using the combined descriptor DT-CWT + HOG and the Random Forest classifier, which is better than the results obtained in [2, 41].

In addition, 81.11% RR was achieved for Cambridge dataset, which is slightly lower than the score obtained in [27], but superior to one cited in [28]. Furthermore, increasing the training percentage to 50% greatly improves the RR of our system from 81.11% to 92.89%. The achieved RR exceeds the performance of HMM used in [55], yet the methods used in [63] achieved a higher RR of 98.23%.

From our overall analysis, our proposed method achieved significant improvement in terms of recognition rate where 97.5 to 99.17% was obtained using DT-CWT + HOG combined with Random Forest classifier for American Sign Language, and 100% RR for the same architecture applied for Marcel dataset and 81.11% for Cambridge dataset. We can say that there is no universal descriptor or classifier that could be the best for any data with complex or uniform background.

Table 11 Comparison of ASL results to previous works

Protocol	Reference	Method	RR (%)
P1	Triesch et al. 2002 [67]	Elastic Graph Matching	93.8
	Kelly et al. 2010 [25]	Eigenspace Size Function + Hu moments	85.1
	Moghaddam et al. 2011 [37]	Kernel Principle Component Analysis + SVM	89.5
	Dahmani et al. 2014 [9]	Tchebichef moments + Hu moments + Geometric features	85.33
	Reddy et al. 2018 [51]	Local Histogram Feature Descriptor + SVM	91.9
	Proposed Method	***DT-CWT+HOG+Random Forest***	***89.76***
P2	Just et al. 2006 [21]	Modified Census Transform	89.9
	Kelly et al. 2010 [25]	Eigenspace Size Function + Hu moments	91.8
	Moghaddam et al. 2011 [37]	Kernel Principle Component Analysis + SVM	95.3
	Dahmani et al. 2019 [10]	Size Functions based moments	94.58
	Dahmani et al. 2014 [9]	Tchebichef moments + Hu moments + Geometric features	96.88
	Reddy et al. 2018 [51]	Local Histogram Feature Descriptor + SVM	95
	Proposed Method	***DT-CWT+HOG+Random Forest***	***96.56***

Moreover, for the ASL dataset, two benchmark protocols are, usually, employed in literature to facilitate the comparison between different techniques. The first protocol, P1, uses only 3 images for training and 21 images for testing, whereas, the second protocol, P2, uses 8 images for training and the remaining 16 images are used for testing, protocol P1 was first introduced in [67] and P2 in [21]. Table 11 shows a comparison between our proposed system and previous works using protocols P1 and P2.

Table 11 shows that the proposed method performed well using both protocols, for protocol P2, obtained results are better than previous works, where 309 test images were correctly classified from a total of 320 images. However, for protocol P1, the Elastic Graph Matching method used in [67] performed better than our method achieving 93.8%, where our system achieved 89.76% recognition rate, this is due to the reduced number of training images (3 per class).

6 Conclusion

In this paper, we have presented an alphabet sign and a gesture recognition system using DWT, DT-CWT, and HOG descriptors for gesture parametrization and MLP, RBNN, PNN, SVM, as well as Random Forest classifiers to recognize the correct gestures and illustrate the efficiency of our system. We have demonstrated that the combined features DT-CWT + HOG performed much better in terms of both accuracy rate and execution times than the individual descriptors. To evaluate these performances, we have used several datasets composed of static and dynamic gestures. For the static form, two sign language datasets with simple background were used, the first contains 10 alphabets from the American Sign Language, and the second contains 30 alphabets from the Arabic Sign Language, for the dynamic form, we used two different datasets, the first contains 4 different classes, where each class is defined by 12 video sequences with 55 frames each, the second contains 3 different gestures, each gesture is repeated in 3 different motions, which provides 9 classes recorded with 5 different illumination settings.

From simulation results, we conclude that applying the HOG descriptor to the imaginary part of the approximation image computed by DT-CWT provides much better results than using the individual features. Furthermore, for the classification task, SVM and Random Forest classifiers provided better results overall when compared with other classifiers, where for ASL, Marcel, and Cambridge datasets, Random Forest achieved higher recognition rates than all other classifiers, whereas SVM provided lower processing times. However, for ArSL dataset, the SVM classifier achieved better results in terms of recognition rates and processing time. The comparison with some previous works shows the efficiency of the proposed method, where comparable performance was achieved by our proposed system.

For future work, we can investigate improving this system by adding segmentation and hand tracking steps for real-time acquisition and recognition.

Acknowledgments Authors would like to thank Dr. Alaa Halawani for providing the Arabic Sign Language (ArSL) dataset. Authors would also like to acknowledge the General Directorate of Scientific Research and Technological Development (DGRSDT).

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

References

1. Adam I (2010) “Complex Wavelet Transform: application to denoising,” PhD Thesis, Politehnica University of Timisoara and Telecom Bretagne, Brest
2. Agab SE, Chelali FZ (2019) “Dynamic hand gesture recognition based on textural features”, 2019 International Conference on Advanced Electrical Engineering (ICAEE), Algiers, Algeria. https://doi.org/10.1109/ICAEE47123.2019.9014683
3. Al-jarrah O, Halawani A (2001) Recognition of gestures in Arabic sign language using neuro-fuzzy systems. Artif Intell 133:117–138. https://doi.org/10.1016/S0004-3702(01)00141-2
4. Al-Shamayleh AS, Ahmad R, Abushariah MAM, Alam KA, Jomhari N (2018) A systematic literature review on vision based gesture recognition techniques. Multimed Tools Appl 77:28121–28184. https://doi.org/10.1007/s11042-018-5971-z
5. Badi H, Sameem AK, Husien S (2013) Gesture feature extraction for static gesture recognition. Arab J Sci Eng 38(12):3349–3366. https://doi.org/10.1007/s13369-013-0654-6
6. Camgöz N, Kindiroglu A, Akarun L (2014) “Gesture Recognition Using Template Based Random Forest Classifiers”, In ECCV Workshops; Springer, Cham, Switzerland, 579–594. https://doi.org/10.1007/978-3-319-16178-5_41
7. Candrasari EB, Novamizanti L, Aulia S (2019) Discrete wavelet transform on static hand gesture recognition. J Phys Conf Ser 1367:1–13. https://doi.org/10.1088/1742-6596/1367/1/012022
8. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297. https://doi.org/10.1007/BF00994018
9. Dahmani D, Larabi S (2014) User-independent system for sign language finger spelling recognition. J Vis Commun Image Represent 25(4):–1240, 1240, 1250. https://doi.org/10.1016/j.jvcir.2013.12.019
10. Dahmani D, Larabi S, Cheref M (2019) Efficient representation of size functions based on moments theory. Multimed Tools Appl 78:27957–27982. https://doi.org/10.1007/s11042-019-07859-9
11. Dalal N, Triggs B (2005) “Histograms of oriented gradients for human detection”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 886-893. https://doi.org/10.1109/CVPR.2005.177
12. Faudzi AAM, Ali MHK, Azman MA, Ismail ZH (2012) Real-time Hand Gestures System for Mobile Robots Control. Procedia Engin 41:798–804. https://doi.org/10.1016/j.proeng.2012.07.246
13. Foody GM (2004) Supervised image classification by MLP and RBF neural networks with and without an exhaustively defined set of classes. Int J Remote Sens 25(15):3091–3104. https://doi.org/10.1080/01431160310001648019
14. Galelli S, Castelletti A (2013) Assessing the predictive capability of randomized tree-based ensembles in streamflow modelling. Hydrol Earth Syst Sci 17(7):2669–2684. https://doi.org/10.5194/hess-17-2669-2013
15. Gamal HM, Abdul-Kader HM, Sallam EA (2013) “Hand gesture recognition using fourier descriptors”, International Conference on Computer Engineering & Systems (ICCES). https://doi.org/10.1109/ICCES.2013.6707218
16. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. Mach Learn 63:3–42. https://doi.org/10.1007/s10994-006-6226-1
17. Guesmi F, Bouchrika T, Jemai O, Zaied M, Ben Amar C (2016) “Arabic sign language recognition system based on wavelet networks”, 2016 IEEE international conference on systems, man, and cybernetics (SMC), Budapest, 3561-3566. https://doi.org/10.1109/SMC.2016.7844785
18. Hsu CW, Lin CJ (2002) “A comparison of methods for multiclass support vector machines”, In: IEEE Transactions on Neural Networks, 13 (2), 415–425. https://doi.org/10.1109/72.991427
19. Jain AK, Mao J, Mohiuddin KM (1996) Artificial neural networks: a tutorial. IEEE Comput 29(3):31–44. https://doi.org/10.1109/2.485891
20. Jalobeanu A, Blanc-Feraud L, Zerubia J (2003) Natural image modeling using complex wavelets. Proc SPIE 5207:480–495. https://doi.org/10.1117/12.507945
21. Just A, Rodriguez Y, Marcel S (2006) “Hand Posture Classification and Recognition using the Modified Census Transform”, IEEE 7th International Conference on Automatic Face and Gesture Recognition (FGR06). https://doi.org/10.1109/FGR.2006.62
22. Karami A, Zanj B, Sarkaleh AK (2011) Persian sign language (PSL) recognition using wavelet transform and neural networks. Expert Syst Appl 38:2661–2667. https://doi.org/10.1016/j.eswa.2010.08.056
23. Katti J, Kulkarni A, Pachange A, Jadhav A, Nikam P (2021) “Contactless Elevator Based on Hand Gestures During Covid 19 Like Pandemics”, 7th International Conference on Advanced Computing and Communication Systems (ICACCS). https://doi.org/10.1109/ICACCS51430.2021.9441827
24. Kaur B, Joshi G (2016) “Lower order Krawtchouk moment-based feature-set for hand gesture recognition”, Advances in Human Computer Interaction. https://doi.org/10.1155/2016/6727806

25. Kelly D, McDonald J, Markham C (2010) A person independent system for recognition of hand postures used in sign language. Pattern Recogn Lett 31(11):1359–1368. https://doi.org/10.1016/j.patrec.2010.02.004
26. Khan RZ, Ibraheem N (2012) Hand gesture recognition: A literature review. Int J Artificial Intel Appl (IJAIA) 3:161–174. https://doi.org/10.5121/ijaia.2012.3412
27. Kim T, Cipolla R (2009) Canonical correlation analysis of video volume tensors for action categorization and detection. IEEE Trans Pattern Anal Mach Intell 31(8):1415–1428. https://doi.org/10.1109/TPAMI.2008.167
28. Kim T, Kittler J, Cipolla R (2007) Discriminative learning and recognition of image set classes using canonical correlations. IEEE Trans Pattern Anal Mach Intell 29(6):1005–1018. https://doi.org/10.1109/TPAMI.2007.1037
29. Kingsbury N (1998) "The dual-tree complex wavelet transform: a new efficient tool for image restoration and enhancement," 9th European Signal Processing Conference (EUSIPCO 1998), Rhodes
30. Kingsbury N (2000) "A dual-tree complex wavelet transform with improved orthogonality and symmetry properties", International Conference on Image Processing, Canada, 375-378. https://doi.org/10.1109/ICIP.2000.899397
31. Ma X, Peng J (2018) Kinect sensor-based long-distance hand gesture recognition and fingertip detection with depth information. J Sensors 2018:5809769. https://doi.org/10.1155/2018/5809769
32. Mahdikhanlou K, Ebrahimnezhad H (2020) Multimodal 3D American sign language recognition for static alphabet and numbers using hand joints and shape coding. Multimed Tools Appl 79:22235–22259. https://doi.org/10.1007/s11042-020-08982-8
33. Mahmud H, Morshed MM, Hasan MdK (2021) "A Deep Learning-based Multimodal Depth-Aware Dynamic Hand Gesture Recognition System", Computer Vision and Pattern Recognition. https://doi.org/10.48550/arXiv.2107.02543
34. Mallat SG (2009) A wavelet tour of signal processing: the sparse way. Elsevier Ltd, USA
35. Mantecón T, del-Blanco CR, Jaureguizar F, García N (2016) "Hand Gesture Recognition Using Infrared Imagery Provided by Leap Motion Controller," International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS), 10016, Springer. https://doi.org/10.1007/978-3-319-48680-2_5
36. Mitra S, Acharya T (2007) "Gesture Recognition: A Survey," In: IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 37 (3), 311–324. https://doi.org/10.1109/TSMCC.2007.893280
37. Moghaddam M, Nahvi M, Pakc RH (2011) "Static Persian Sign Language Recognition Using Kernel-Based Feature Extraction", IEEE 7th Iranian Conference on Machine Vision and Image Processing. https://doi.org/10.1109/IranianMVIP.2011.6121539
38. Mohebali B, Tahmassebi AH, Meyer-Baese A, Gandomi AH (2020) "Probabilistic neural networks: a brief overview of theory, implementation, and application", Handbook of Probabilistic Models, Pijush Samui, 347-367. https://doi.org/10.1016/B978-0-12-816514-0.00014-X
39. Moulin P (2009) "Multiscale image decompositions and wavelets", The essential guide to image processing, Al Bovik, 123-142. https://doi.org/10.1016/B978-0-12-374457-9.00006-8
40. Nalepa J, Grzejszczak T, Kawulok M (2014) "Wrist Localization in Color Images for Hand Gesture Recognition," Man-Machine Interactions 3, 242, Springer. https://doi.org/10.1007/978-3-319-02309-0_8
41. Nasri S, Behrad A, Razzazi F (2015) A novel approach for dynamic hand gesture recognition using contour-based similarity images. Int J Comput Math 92(4):662–685. https://doi.org/10.1080/00207160.2014.915958
42. Nie G, Zhao J (2019) "Gesture Recognition Based on Improved HOG-LBP Features", International Conference on Computer, Network, Communication and Information Systems (CNCI), Qingdao, China. https://doi.org/10.2991/cnci-19.2019.39
43. Oudah M, Al-Naji A, Chahl J (2020) Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. J Imaging 6(8). https://doi.org/10.3390/jimaging6080073
44. Oudah M, Al-Naji A, Chahl J (2021) Elderly Care Based on Hand Gestures Using Kinect Sensor. Computers 10(1):5. https://doi.org/10.3390/computers10010005
45. Patil R, Patil S (2015) Static hand gesture recognition system based on DWT feature extraction technique. Int J Innov Res Sci Technol 02(05):23–27
46. Praveen Kumar P, Prasad Reddy PVGD, Srinivasa Rao P (2018) "Sign language recognition with multi feature fusion and Adaboost classifier," ARPN Journal of Engineering and Applied Sciences, 13 (4)
47. Premaratne P (2014) Historical Development of Hand Gesture Recognition In "Human computer interaction using hand gestures," Cognitive Science and Technology, Singapore, Springer. https://doi.org/10.1007/978-981-4585-69-9
48. Rahim MA, Shin J, Islam MR (2020) Hand gesture recognition-based non-touch character writing system on a virtual keyboard. Multimed Tools Appl 79:11813–11836. https://doi.org/10.1007/s11042-019-08448-6
49. Rashed JR, Hasan HA (2017) "New method for hand gesture recognition using neural network," Journal of Engineering and sustainable development, 21 (01)
50. Rautaray SS, Agrawal A (2012) Real time gesture recognition system for interaction in dynamic environment. Procedia Technol 4:595–599. https://doi.org/10.1016/j.protcy.2012.05.095

51. Reddy DA, Sahoo JP, Ari S (2018) "Hand Gesture Recognition Using Local Histogram Feature Descriptor", 2nd International Conference on Trends in Electronics and Informatics (ICOEI). https://doi.org/10.1109/ICOEI.2018.8553849
52. Roccetti M, Marfia G, Zanichelli M (2010) The art and craft of making the Tortellino: playing with a digital gesture recognizer for preparing pasta culinary recipes. Comput Entertain 8(4):1–20. https://doi.org/10.1145/1921141.1921148
53. Sadeddine K, Djeradi R, Chelali FZ, Djeradi A (2018) "Recognition of Static Hand Gesture", 2018 6th International Conference on Multimedia Computing and Systems (ICMCS), Rabat. https://doi.org/10.1109/ICMCS.2018.8525908
54. Sadeddine K, Chelali FZ, Djeradi R, Djeradi A, Benabderrahmane SA (2021) Recognition of user-dependent and independent static hand gestures: application to sign language. J Vis Commun Image Represent 79:103193. https://doi.org/10.1016/j.jvcir.2021.103193
55. Sagayam KM, Hemanth DJ, Vasanth XA, Henesy LE, Ho CC "Optimization of a HMM-Based Hand Gesture Recognition System Using a Hybrid Cuckoo Search Algorithm", Hybrid Metaheuristics for Image Analysis. Springer, Cham. https://doi.org/10.1007/978-3-319-77625-5_4
56. Sahoo JP, Ari S, Ghosh DK (2018) Hand gesture recognition using DWT and F-ratio based feature descriptor. IET Image Process 12(10):1780–1787. https://doi.org/10.1049/iet-ipr.2017.1312
57. Sahoo JP, Prakash AJ, Pławiak P, Samantray S (2022) Real-Time Hand Gesture Recognition Using Fine-Tuned Convolutional Neural Network. Sensors 22(3):706. https://doi.org/10.3390/s22030706
58. Satybaldina D, Kalymova G (2021) Deep learning based static hand gesture recognition. Indo J Electri Eng Comput Sci (IJEECS) 21(1). https://doi.org/10.11591/ijeecs.v21.i1.pp398-405
59. Sharma R, Pavlovic VI, Huang TS (1998) Toward multimodal human-computer interface. Proc IEEE 86(5): 853–869. https://doi.org/10.1109/5.664275
60. Shriram S, Nagaraj B, Jaya J, Shankar S, Ajay P (2021) "Deep learning-based real-time AI virtual mouse system using computer vision to avoid COVID-19 spread", Journal of Healthcare Engineering. https://doi.org/10.1155/2021/8133076
61. Specht DF (1990) Probabilistic neural networks. Neural Netw 3(1):109–118. https://doi.org/10.1016/0893-6080(90)90049-Q
62. Sun JH, Ji TT, Zhang SB, Yang JK, Ji GR (2018) "Research on the Hand Gesture Recognition Based on Deep Learning," 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), Hangzhou, China https://doi.org/10.1109/ISAPE.2018.8634348
63. Tang H, Liu H, Xiao W, Sebe N (2019) "Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion", Computer Vision and Pattern Recognition
64. Taskiran M, Cimen S, Cam Taskiran ZG (2018) The novel method for recognition of american sign language with ring projection and discrete wavelet transform. World J Eng Res Technol (WJERT) 04(01): 92–101
65. Thalange A, Dixit S (2015) "Sign language alphabets recognition using wavelet transform", Conference on Intelligent Computing Electronics Systems and Information Technology. Kuala Lumpur (Malaysia).
66. Triesch J, von der Malsburg C (1996) "Robust classification of hand postures against complex backgrounds, " Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, Killington, VT, USA, 170-175. https://doi.org/10.1109/AFGR.1996.557260
67. Triesch J, von der Malsburg C (2002) Classification of hand postures against complex backgrounds using elastic graph matching. Image Vis Comput 20:937–943. https://doi.org/10.1016/S0262-8856(02)00100-2
68. Trigueiros P, Ribeiro F, Reis LP (2013) "Vision-based gesture recognition system for human-computer interaction", IV ECCOMAS thematic conference on computational vision and medical image processing. Taylor and Francis, Publication, Funchal, Madeira
69. Zhang F, Liu Y, Zou C, Wang Y (2018) "Hand gesture recognition based on HOG-LBP feature", in IEEE international instrumentation and measurement technology conference (I2MTC). https://doi.org/10.1109/I2MTC.2018.8409816

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