ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse84942020.pdf
https://doi.org/10.30534/ijatcse/2020/84942020

A Comparative Analysis of Filters towards Sign Language Recognition

Fauziah Kasmin[1], Nur Shahirah Abd Rahim[2], Zuraini Othman[3], Sharifah Sakinah Syed Ahmad[4] Zahriah Sahri[5]
[1,2,3,4,5] Centre for Advanced Computing Technologies (CACT), Department of Intelligent Computing and Analytics, Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka (UTeM), Durian Tunggal, Melaka, Malaysia.
fauziah@utem.edu.my, shahirah0397@gmail.com, zuraini@utem.edu.my, sakinah@utem.edu.my, szahriah@utem.edu.my

ABSTRACT

One of the major drawbacks of our society is the barrier that is created between the deaf and the normal. Communication is the only medium by which we can share our message through speech. But, for a person who is having hearing impairment faces difficulties in communication with normal people, sign language is the main medium for deaf people to communicate. There have been few attempts in the past to recognize hand gestures, but with low rate of recognition. Furthermore, the dataset accessible in this subject is still noisy. The objective of this study is to determine filtering techniques that give good performance for hand sign language recognition. Dataset used in this study is Gesture Dataset 2012. In this study, three types of filtering techniques have been chosen which are average filter, median filter, and Laws' masks filters. A comparative analysis has been done and the performance of the filtering techniques are based on recognition performance. Meanwhile, Canny, Sobel, Prewitt, and Roberts algorithms are used for edge detection and features extracted are Histogram of Oriented Gradients (HOG). Finally, for classification, multiclass Support Vector Machine (SVM) classifier is used. At the end of this study, it has been found that using a combination of Laws' masks filters and Canny edge detection algorithm is proven to be a promising combination to increase sign language recognition performance since they give the best results of recognition performance compared to other methods used in the experiments.

Key words: edge detection, histogram oriented gradient, Laws' masks filter, sign language recognition, support vector machine.

1. INTRODUCTION

Sign language is a visual language using hand forms, gestures, facial expressions, and body language, commonly used by deaf people to exchange information among their communities [1][2]. There are many different sign languages used by various countries. This study will be more focused on the recognition of American Sign Language (ASL). ASL is a natural language that has the same linguistic characteristics as verbal languages. Sign language recognition is an expanding research area [3]. Children who are classified as Deaf or Hard of Hearing (DHH) need this recognition and recognition is extremely important to them [2]. For DHH children, they are affected by the language environment and always have problems to access language input [2]. Computer vision recognition of ASL alphabets is a challenging task due to the large intra-class variation, high interclass similarities, constant occlusions and not forgetting of the complexity of ASL signs, [4]. Most of the researchers are using ASL recognition since ASL is single-handed. Thus, ASL is much less complicated compared to other sign languages such as Indian Sign Language (ISL) which using both hands [5].

An ASL recognition system has been developed by [6] and they used artificial neural networks for the translation. ASL word has been translated into English. The system consists of the sensory glove and 3D motion tracker where these devices are used to extract gesture features. Features from sensory glove are the finger joint angles and the tracker gives the hand movements' trajectory. Another research was also done by [7] where these researchers have used Hough transform and neural networks to make the recognition. Their system does not use any gloves and they only use bare hands and interaction exists naturally. The images have been processed and converted into feature vectors. Model has been developed based on the training signs and the feature vectors from test images have been compared with the model and the recognition process achieved a high percentage of accuracy since the features have not been affected by rotation and scaling. In research done by [4], convolutional neural networks (CNN) have been used for the recognition process. They have generated a lot of perspective views which include multi-view augmentation and inference fusion. This will make an effective training process and can overcome potential overfitting. Problems caused by partial occlusions and orientation variations have been solved by applying this approach.

In order to have good recognition of sign language, images retrieved should be clear from noise and other artifacts. One of the approaches that diminish noises is by using filtering methods. Filtering is a technique to refined and modify images [8]. The main reason for using filters is for reducing noise. Apart from that, filters can also be used for emphasizing certain features or removing other features. There is research that has been done in applying filtering methods. Some of the popular filtering methods are average filter [9], median filter [10], Gaussian filter [11], Gabor filter [12], and many more. An algorithm that has been using a non-linear filtering approach is used in [13] to overcome salt and paper noise. These salt and paper noise have caused an actual image to contain white and black spots. So, they proposed a comprehensive median filter (CMF) to enhance the standard median filter. By applying CMF, the resultant image become high quality compared to the standard median filter. Han et. al [14] have proposed adaptive median filtering to eliminate the impulse noise that exists in fingerprint images. Their approach involved three steps. The first step is to initialize the size of the window and then the center pixel is checked whether impulse noise or not. Then, the determination of the size of the window is done based on median, minimum, and maximum values within the size of the window and lastly, median filtering is applied on the determined size of the window. In a research done by [15], they have used neighborhood approach to filter images. They have combined grey level and geometrical configuration information in the area of neighborhood. They have combined adaptive weighted mean and nonlocal means filtering in their approach. Filtering images by using hue, saturation and intensity values have been used by [16]. Each color image has been separated into its hue, saturation and intensity value components. For each component, noise is filtered by using median filter. Work done by [17] have proposed image enhancement by using homomorphic filtering techniques. They produced sub-images by sub-divided the original image into two parts i.e. vertical and horizontal. Then they use homomorphic filtering on sub-images and then they combine together these vertical and horizontal sub-images. As a result, they get clearer image.

Laws' masks filters are usually used as texture analysis. It has been used for synthetic and natural textures recognition [18]. It has been developed by [19]. Laws' masks filters have been created in image processing as one of the finest techniques for texture analysis and are used in many apps. Moreover, the Laws' algorithm is considered attractive for extracting image texture properties. This is because distinct Laws' algorithm masks seem to investigate distinct picture characteristics [20]. Laws' algorithm has been used previously in vascular ultrasound imaging and dermatological imaging [21]. Besides, Laws' texture energy measure (LTEM) techniques have been widely used in medical imaging, which for diagnosing liver disease and cancer [22]. Plus, LTEM has been also applied in order to detect defective texture such as steel and ceramics. They came out by recommending the uses of the Laws' masks filters with parallel processing methods to improve system efficiency and reducing the time of processing [23]. From the result gained from [22], the general average classification precision was the highest when the regulations used texture energy-based characteristics as the input of the neural network. Here, we can see that most of the study using Laws' masks filters showing a good result. Furthermore, the use of the Laws' masks filters does not require pre-image processing. This may be one of the advantages of this technique [20]. However, the performance evaluated for complex steerable pyramid implemented in the frequency domain based Laws' masks filters shows a better classification compared to an individual Laws' masks filter method [24]. A combination of steerable approach also has been done in [25] and it shows better performance in segmenting retinal blood vessels and binarisation of handwritten images.

The focus of this study is presenting an image recognition framework that will undergo image acquisition, converting colored images into grayscale, image restoration, edge detection, and finally extracting features. After the process is done, the framework will proceed with the classification for image recognition. Generally, image recognition in terms of machine vision is the ability of the system in identifying various things such as objects, people, places, writing, and actions in an image. For image restoration, three types of filtering algorithms will be used which are average filter, median, and Law's filter. In this study, Laws' mask filters will be used as filtering techniques. These three filters will be compared and analyzed based on the resultant images. The result is based on the performance measured after classification. It is important to compare filters as it will help the process of classification images. An image with unwanted noise may slowing down the image processing system.

This paper is organized as follows: Section 2 described the methods used and the results of the experiment are shown in Section 4. Section 5 will discuss and conclude this paper.

2. MATERIALS AND METHODS

This study includes these five major steps as in Figure 1.

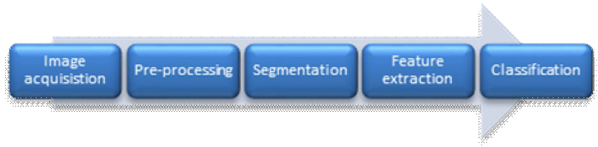

Figure 1: Experimental set-up

1) Image acquisition

In the input image phase, the dataset of the ASL Alphabet is used in this experiment which was fetched from the Internet named Gesture Dataset 2012 [26]. The dataset containing the captured images of American Sign Language alphabets (A-Z), as well as the numerical (0-9). From the original dataset, only the alphabet dataset will be taken completely from A to Z to carry out this experiment. There will be 60 images per alphabet for the training set, and 10 images for the testing set. The overall total was up to 1560 images from A to Z used for the training set, 260 images for the testing set. Figure 2 shows some samples of images in Gesture Dataset 2012.

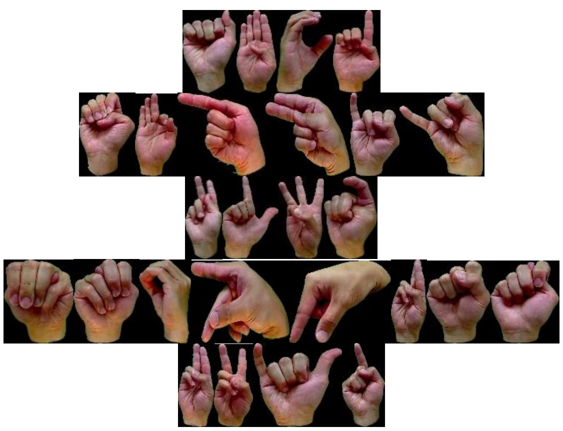

Figure 2: Gesture Dataset 2012

2) Pre-processing

The images were preprocessed after acquiring from the database. To reduce image complexity, it is first changed into grayscale. Grayscale is a transformation of a 3D pixel value (R, G, B) to a 1D value.

Next, the image restoration takes place, which the most important part of this study. Here, in a total of three types of filtering techniques will be used, so that at the end of the study we may conclude which filtering technique gives better recognition performance. The filtering techniques are median filter, average filter, and Law's filter. In image processing, filtering is used for image modification and enhancement. Image processing operations implemented with filtering include edge enhancement, smoothing, and sharpening.

a) Average filtering

An average filter is known to be a simple, intuitive, and easy way to smooth pictures, reducing the variation in intensity between one pixel and the next [27]. There is a 3x3 square kernel. However, a larger kernel which is a 5x5 squares kernel can be used for more effective and good for a critical smoothing. This operation can be performed using a convolution mask $m \times n$ where all coefficients have values of 1 / m. The image is reduced by blurring. The equation is as

$$f'(x,y) = \frac{1}{mn}\sum_{(s,t)\in S_{xy}} g(s,t). \tag{1}$$

How it works:

1. Replacing the mean value of the pixel's neighbors values, including itself.
2. Eliminate pixel values that are not representative of their environment.

b) Median filtering

Unlike the average filter, the median filter replaces the pixel value with the median value of the pixels [28]. The median is calculated by first separating all the pixel values in numerical order from the surrounding neighborhood and then replacing the pixel regarded with the center pixel value. They provide less blurred and very efficient image-based noise removal of impulse, pepper, and salt. The equation is as

$$f'(x,y) = median\{g(s,t)\}. \tag{2}$$

b) Laws filtering

Laws filtering is used for texture feature image extraction [22]. To detect distinct kinds of textures, this strategy utilizes local masks to create distinct texture characteristics. The quantity of variation is evaluated in a window using the energy-textured strategy established by Law.

Image texture is a complicated visual pattern that includes entities or areas with brightness, color, shape, size, etc. Law's filter uses texture filters to filter the input picture. It calculates the energy of texture by summarizing the absolute value of filtering around each pixel in neighborhoods. In this study, there will be a set of 1-dimensional convolutions kernel as shown in Figure 3.

L5 (level)	=	[1 4 6 4 1]
E5 (edge)	=	[-1 -2 0 2 1]
S5 (spot)	=	[-1 0 2 0 -1]
R5 (ripple)	=	[1 -4 6 -4 1]
W5 (wave)	=	[-1 2 0 -2 1]

Figure 3: 1-dimensional of Law's filter

Each of these filters highlights different texture features, so new images are directly related to the values of these features. The 2D convolution mask is then obtained by calculating the output of the vector pairs. For example, L5R5 is the L5 and R5 products. There will be 25 texture maps of 2D 5x5 mask can be obtained from 1-dimensional Law's filter. Figure 4 shown depicts how the 2D convolution mask is obtained, meanwhile, Table 1 shows the 25 texture maps.

```
>> L5 = [1;4;6;4;1]

L5 =

     1
     4
     6
     4
     1

>> R5 = [1 -4 6 -4 1]

R5 =

     1    -4     6    -4     1

>> L5*R5

ans =

     1    -4     6    -4     1
     4   -16    24   -16     4
     6   -24    36   -24     6
     4   -16    24   -16     4
     1    -4     6    -4     1
```

Figure 4: The production of L5R5

Table 1: 25 texture maps of 2D 5 by 5 mask

	L5	E5	S5	R5	W5
L5	L5L5	L5E5	L5S5	L5R5	L5W5
E5	E5L5	E5E5	E5S5	E5R5	E5W5
S5	S5L5	S5E5	S5S5	S5R5	S5W5
R5	R5L5	R5E5	R5S5	R5R5	R5W5
W5	W5L5	W5E5	W5S5	W5R5	W5W5

Next, after creating a texture energy map, the identification of final maps are determined. Finally, fifteen maps are produced after symmetry pairs are combined. For example, vertical wave content is given by the L5W5 texture energy map, so horizontal wave content is given by W5L5. Thus, the mean of L5W5/W5L5 is the total wave content where they show vertically and horizontally wave content. Table 2 shows the 15 final texture maps produced.

Table 2: 15 texture maps of 5 by 5 mask

L5L5	E5L5/L5E5	W5E5/E5W5
E5E5	W5L5/L5W5	W5S5/S5W5
S5S5	S5E5/E5S5	R5E5/E5R5
R5R5	R5L5/L5R5	R5S5/S5R5
W5W5	S5L5/L5S5	W5R5/R5W5

3) Segmentation

Edge detection is conducted after the applying filtering process. This step takes place before the feature extraction phase so that any redundant and noisy data could be avoided and only useful information can be obtained [29]. One of the techniques of image processing, edge detection frequently used to detect object borders in a picture by detecting discontinuity of brightness. In this experiment, Canny, Sobel, Roberts, and Prewitt edge detection will be used.

a) Canny edge detection

One of the most frequently used image processing instruments is Canny edge detection [30]. In a very strong manner, it traces the edges. It is an ideal method for edge detection because it offers excellent detection, clear reaction, and excellent localization. This is how it works:

i. Smoothing the picture by using the Gaussian filter.
ii. Compute the magnitude of the gradient.
iii. Pixels that are not part of an edge are removed by using the non-maximum suppression algorithm.
iv. Hysteresis thresholding is used along edges. Two thresholds are determined that are upper (*u*) and lower (*l*) thresholds.
v. Checking each pixel where:
   - The gradient of the pixel > *u*, then it is an edge.
   - The gradient of the pixel < *l* then discards the pixel.
   - *u* < Gradient of the pixel < *l*, then it is an edge.

Small filters are desirable for tiny and sharp lines to be detected as they cause fewer cases of blurring. Large filters are desirable for bigger and smoother edges to be detected.

b) Sobel edge detection

Sobel edge detection is 3x3 convolution kernels [27]. It is a line edge detector. Figure 5 shows the common 3x3 convolution kernel used in Sobel. As we can see, the kernels able to estimates the x and y direction of the gradient. Images are converted with both kernels to approximate the derivatives in horizontal and vertical modifications [31]. At each given point, the magnitude of the gradient can be approximated with

$$G = \sqrt{G_x^2 + G_y^2} \tag{3}$$

-1	0	1		1	2	1
-2	0	2		0	0	0
-1	0	1		-1	-2	-1
	Gx				Gy	

Figure 5: 3 x 3 convolution kernel used in Sobel

Because of the Gaussian smoothing, it is less susceptible to noise present in pictures. Smoothing impacts edge detection precision. It does not, however, generate high-precision pictures for edge detection. Its quality is sufficient to be used in many applications.

c) Prewitt edge detection

Prewitt edge detection is the correct way to estimate the edge's magnitude and orientation [27]. Prewitt operators are restricted to eight possible orientations. It starts with the calculation of all eight masks. Then select the bulletproof mask with the biggest module. Figure 6 shows the Prewitt mask.

1	2	1		-1	0	1
0	0	0		-2	0	2
-1	-2	-1		-1	0	3
	Px				Py	

Figure 6: Prewitt mask

This is how the Prewitt operator works:

i. Apply the Px and Py Prewitt mask to the picture.
ii. Apply the algorithm and gradient for Prewitt edge detection.
iii. Masks manipulating Px, Py in the input picture individually.
iv. Combine results to find the absolute gradient magnitude.
v. The output edges are the absolute magnitude.

d) Roberts edge detection

Roberts edge detection is somehow similar to Sobel and Prewitt [27]. This is how it works:

i. The pixel value at that stage reflects the absolute magnitude of the input spatial gradient.
ii. The operator has a 2x2 convolution kernel that responds to the running peak edge.
iii. There is one respective kernel for each of the perpendicular orientations.
iv. Magnitude is given by the same formula as of the Sobel, but the orientation of the angle as in equation 4.

$$\text{angle } \theta = \arctan\left(\frac{Gx}{Gy}\right) - \frac{3\pi}{4} \qquad (4)$$

4) Feature extraction

Extracting feature is a form of dimensionality reduction. Feature extraction is performed after an image had gone through the pre-processing and segmentation phase. In order to categorize these gestures, definite characteristics of the image need to be taken out. In this study, the histogram of oriented gradient (HOG) descriptor will be used. HOG is a feature descriptor that has been widely used and successfully used for object detection, which calculates the incidence of gradient orientation on the local area of the image window - detection, or region of interest (ROI). It is calculated by detecting the window detector on the image, where the HOG features are extracted for each filter.

HOG was used to calculate the input image gradient level and gradient path [32]. The fundamental hypothesis behind the angular descriptor's histogram is calculated that the dispersion of force curves or edge bearings can illustrate the presence and shape of protrusions in the picture. The steps in HOG feature extraction are gradient computation, oriented binning, descriptors block, and block normalization [33]. The details are as below:

i. Gradient computation - The gradient of the image is gained by shifting it vertical and horizontal 1D distinctive derivative mask such as $Dx = [-0]$ and $Dy = [0]$. Next is to obtain X and Y derivatives by using convolution operation where $I_x = I{*}D_x$ and $I_y = I{*}D_y$. Finally, is to compute the magnitude of the gradient with

$$|G| = \sqrt{I_x^2 + I_y^2} \qquad (5)$$

ii. Oriented binning - This is where the histograms of cells are formed. Each pixel calculates a biased vote based on its gradient computation orientation for a histogram channel. The shape of the cells was rectangular. The histogram channels were distributed uniformly between 0 and 180 degrees or between 0 and 360 degrees. It's either signed or unsigned based on the gradient.

iii. Descriptor block - The gradient strength should be normalized regionally to change the contrast and illumination. It requires grouping cells together into larger blocks that are interconnected. Done with normalization, all histograms are integrated into a single feature vector.

iv. Block normalization - It has different approaches. The normalization factors f was gained with:

$$L - norm: f = \frac{v}{[v] + \varepsilon} \qquad (6)$$

$$L - norm: f = \frac{v}{\sqrt{[v]^2 + \varepsilon^2}} \qquad (7)$$

In this case, the image is divided into small areas called cells and for each pixel, in each cell, a gradient direction histogram is collected.

5) Classification

Features extracted from the image are given as input to the classification phase. There are different classifiers available that are helpful for gestures of acceptance. The stage of classification comprises of two phases, namely training and testing. In digital picture assessment, the most significant aspect is the classification of images. In this experiment, the support vector machine (SVM) is used for the recognition of the ASL alphabets. SVM is a theoretical superior machine learning methodology with great results in classifying high dimensional datasets and has been found to compete with the best machine learning algorithm [34]. Recently, SVM receives special attention as it provides better classification results used by other widely used pattern recognition methods.

Multiclass SVM is a controlled learning strategy [35]. Each item of information is expressed as a point in an n-dimensional space where each element is estimated to be a specific coordinate. SVM develops hyperplanes that have the largest partition to the closest providing information to any class because as a larger margin rule, lower errors of classifier generalization. Full implementation will be done through the following steps:

i. Data loading and preprocessing - Normalizing the input features is a standard procedure in machine learning. The data is centered, and the mean is derived. A mean image is calculated out of all training images. Then, it is subtracted from the dataset. Bias matrix is also added so that our optimizer will treat both the biases and the weights at the same time.

ii. SVM classifier implementation - SVM classifiers are then built by filling in the gradient computation. Then, its vectorized version is implemented. To implement the code for the gradient, we simply return to the representation of our loss function in SVM:

$$L_i = \sum_{j \neq y_i} max\ (0, w_j^T x_i - w_{y_i}^T x_i + \Delta) \quad (8)$$

Then, differentiate the function with respect of w_{y_i} *and* w_j as shown below:

$$\nabla_{w_{y_i}} Li = -(\sum_{j \neq y_i} 1(w_j^T x_i - w_{y_i}^T x_i + \Delta > 0))\ x_i$$

$$\nabla_{w_j} Li = 1(w_j^T x_i - w_{y_i}^T x_i + \Delta > 0) x_i \quad (9)$$

iii. With this, the number of classes that did not pass through the margin $j \neq y_i$ can be counted. However, the given example's classes will not always be classified correctly. Therefore, the SVM loss function penalizes the missed classified which the incorrect classes are summed and xi is scaled by that number. Scores will be computed for each data example (score is the correct class). The scores are then compared. Next, is to compute the margin to see if it is larger than 0. The incorrectly classified will contributes as a loss.

iv. Continue the process by implementing vectorized computation for the loss. Firstly, calculates the input score with respect to weight. Secondly, keep track of the score and get the maximum as it is saved to the variable margin. To implement a vector version of the gradient computation, a mask that flags the examples is first created if when the margin is greater than 0. The number of these examples is counted later.

v. Stochastic Gradient Descent - It is to minimize the loss that was calculated by the SVM. By using the gradient vector that was obtained previously, the parameters will be changed to the direction that the gradient pointing at. We can now train SVM classifiers by using the gradient descent and plot the loss with respect to the number of iterations.

vi. Hyperparameter tuning - where we may improve the accuracy by tuning our learning rate. Also, a good way to tune the hyperparameters are by train and test our classifier. The SVM classifier is trained with a matrix containing learning values and regulated values. The accuracies are recorded, and the maximum value countered is taken. A good hyperparameters setting are considered if the set of parameters that obtained the maximum accuracy.

6) Performance measure

In order to perform a comparative study between the proposed techniques, performance measurement needs to be carried out. The recognition performance, which is the percentage of accuracy is calculated based on the amount of correctly predicted labels. As for Laws' masks filters will produce 15 texture maps, and the process will be tested on each of the texture maps, average and standard deviation is computed for further observation. The performance of each experiment held is computed as below:

$$accuracy = \sum \frac{correctly\ identified\ testing\ image}{total\ number\ of\ testing\ image}\ x\ 100 \quad (10)$$

Figure 7 shows the analysis flowchart of the experiment. From Figure 7, the flowchart of analysis has been shown in more detail. For the training phase, the images from the training set will be converted into grayscale. Then, the image is filtered by using three proposed filtering techniques which are average filter, median filter, and Laws' masks filters. As discussed previously, 15 texture maps will be produced when Laws' masks filters are applied to the image. All the texture maps will carry out the same process as other techniques. The filtered image is then will be applied edge detection technique. Here, four edge detection algorithms are used. Then, the HOG descriptor will be applied to the edged image, and use the HOG value to train the multiclass SVM model.

For the testing phase, the images from the testing set will be carrying out the same process as the training phase. However, after gaining the HOG value, the trained multiclass SVM model will be used to predict the class. Finally, the recognition performance of each process will be computed for observation.

3. RESULTS AND DISCUSSION

In this section, the results of all experiments done will be discussed.

i. Filtering algorithm

The purpose of this analysis is to study on filtering techniques. In this analysis, there will be three filtering algorithms which are average filter, median filter, and Laws' masks filters. Figures 8, 9, and 10 are the transformation after applying the mentioned filtering techniques respectively. Here, we can see that the average filter gives much blurring effects to the image compared to the median filter. While Laws' masks filters are producing 15 texture maps, resulting from the 5 by 5 mask. Table 3 is a guide to be referred for texture maps produced.

Table 3: Texture maps of Laws' masks filters

1 =	L5L5	6 =	E5L5/L5E5	11 =	W5E5/E5W5
2 =	E5E5	7 =	W5L5/L5W5	12 =	W5S5/S5W5
3 =	S5S5	8 =	S5E5/E5S5	13 =	R5E5/E5R5
4 =	R5R5	9 =	R5L5/L5R5	14 =	R5S5/S5R5
5 =	W5W5	10 =	S5L5/L5S5	15 =	W5R5/R5W5

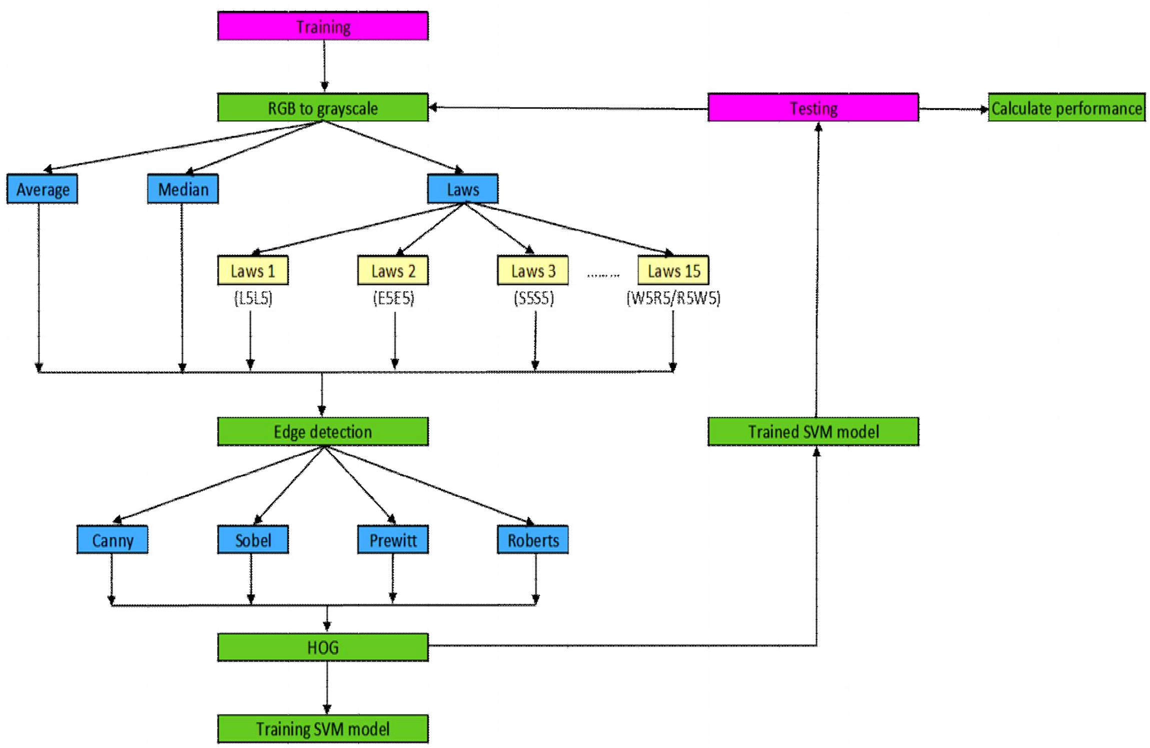

Figure 7: Analysis flowchart

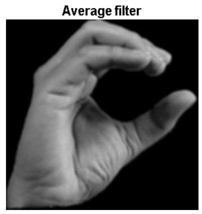

Figure 8: Average filter

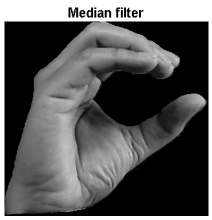

Figure 9: Median filter

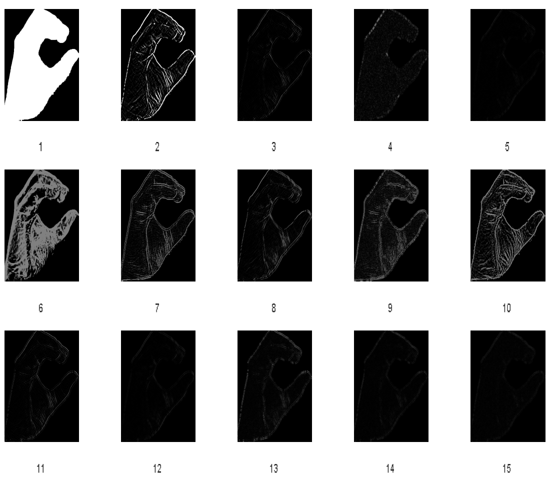

Figure 10: Laws' masks filters

ii. Edge detection algorithm

Figures 11, 12, 13, and 14 show the effects of an image after applying Canny, Sobel, Prewitt, and Roberts edge detection algorithm respectively. From here we can see that, Canny edge detection able to detect edges in a very robust manner. This is because Canny is capable of detecting more detail edges compared to Sobel, Prewitt, and Roberts algorithm.

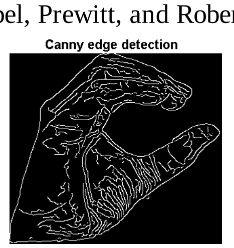

Figure 11: Canny edge detection

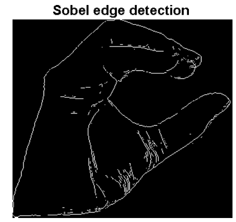

Figure 12: Sobel edge detection

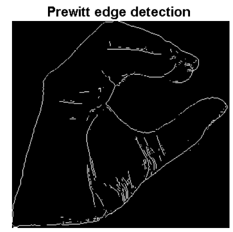

Figure 13: Prewitt edge detection

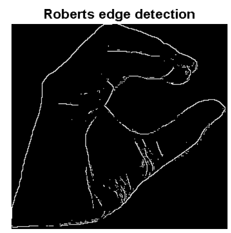

Figure 14: Roberts edge detection

Our objective is to compare which filtering techniques that work best with edge detection algorithm in classifying the ASL. Hence, the comparison also has been made without using any filtering techniques. The results are as follows.

a. Without filtering techniques

Table 4: Performance without filtering

Edge detection	Performance
Canny	49.6%
Sobel	38.1%
Prewitt	38.5%
Roberts	41.9%

Table 4 shows the performance result from the experiments held which is without any filtering techniques is used. The pre-processing phase in this experiment was just going through edge detection techniques. From here we can see that without using any filtering algorithm, the highest percentage of the recognition performance is 49.6%, the pre-processing which applied the Canny algorithm. Meanwhile, the lowest percentage of the recognition performance is 38.1%, the pre-processing which applied the Sobel algorithm. Thus, this has proved that the Canny algorithm is the best edging algorithm compared to Roberts, Prewitt, and Sobel algorithms.

b. With average filter

Table 5: Performance with Average filter

Edge detection	Performance
Canny	48.8%
Sobel	42.3%
Prewitt	41.5%
Roberts	43.5%

Table 5 shows the performance result from experiments held which is with average filter is used. From here, we can see that with the use of the average filter, the highest percentage of recognition performance is 48.8%. We also may notice that the highest recognition performance is the one using the Canny algorithm. Meanwhile, the lowest recognition performance is the one using the Prewitt algorithm. Thus, we can conclude that the Canny algorithm is the best edging algorithm to be used with an average filter compared to Roberts, Sobel, and Prewitt.

c. With median filter

Table 6: Performance with the median filter

Edge detection	Performance
Canny	51.2%
Sobel	39.2%
Prewitt	38.8%
Roberts	41.5%

Table 6 shows the performance result from the experiments held which is with the median filter. From the observation, we can see that with the use of the median filter, the highest percentage of recognition performance is 51.2%. Meanwhile, the lowest percentage of recognition performance is 38.8% which the process that uses the Prewitt algorithm. Besides, we may notice that the highest recognition performance is the one that also using the Canny algorithm. Thus, the Canny algorithm also can be said as the best edging algorithm to be used with the median filter compared to Roberts, Sobel, and Prewitt.

d. With Laws' mask filters

Table 7 shows the result of the recognition performance when Laws' mask filters are used. As said previously, there will be 15 texture maps produced after the Laws' mask filters are applied to the image. Therefore, the experiment is held for each texture map. When we examine the highest recognition performance for each edge detection algorithm, we can see that the highest percentage of recognition performance is 59.6%, the experiment which is held with the use of the Prewitt algorithm and followed by the Canny algorithm which is 58.8%. Next is the Sobel algorithm which is 55.4% and lastly the Roberts algorithm which is 54.6%. Here, we also may notice that the recognition performance is getting better with the aid of Laws' masks filters if compared with recognition performance which with the aid of the average filter and the median filter.

Table 7: Performance with Laws' masks filters

		Performance			
No	Energy maps	Canny	Sobel	Prewitt	Roberts
1	L5L5	37.3%	38.8%	39.2%	40.0%
2	E5E5	55.8%	40.0%	38.8%	38.1%
3	S5S5	53.5%	40.4%	40.4%	39.6%
4	R5R5	58.1%	55.4%	57.3%	53.1%
5	W5W5	56.9%	41.5%	39.6%	44.2%
6	E5L5/L5E5	57.7%	**55.4%**	55.4%	**54.6%**
7	W5L5/L5W5	51.2%	46.9%	45.4%	48.8%
8	S5E5/E5S5	56.9%	40.8%	**59.6**%	38.8%
9	R5L5/L5R5	54.6%	47.7%	49.2%	47.3%
10	S5L5/L5S5	54.2%	50.4%	49.6%	50.0%
11	W5E5/E5W5	55.8%	41.9%	41.9%	38.8%
12	W5S5/S5W5	56.9%	42.3%	41.9%	38.1%
13	R5E5/E5R5	**58.8%**	39.2%	41.2%	37.3%
14	R5S5/S5R5	56.2%	39.6%	39.6%	38.8%
15	W5R5/R5W5	56.9%	52.7%	51.2%	51.2%
AVERAGE :		**54.7%**	44.9%	46.0%	43.9%
STD DEV :		**0.0519**	0.0603	0.0716	0.0625

Average and standard deviation is computed to undergo further observation. The low standard deviation shows that the information points tend to be near to the average while the high standard deviation shows that the information points are spread over a wider range of values.. From the previous observation, the Prewitt algorithm gives the highest percentage of recognition performance. However, the average percentage of recognition performance of the Canny algorithm shows the highest compared to Sobel, Prewitt, and Roberts algorithms. Meanwhile, the average percentage of recognition performance of the Roberts algorithm shows the lowest. Besides, the standard deviation of the recognition performance of the Canny algorithm is the lowest among all edge detection techniques. Therefore, we can conclude that the Canny algorithm works very well with the Laws' masks filters. Furthermore, with the combination of Laws' masks filters and, Canny algorithm, the thirteenth texture maps which are the R5E5/E5R5 give the best results. Meanwhile, the first texture maps which are the L5L5 give the worst results.

Figure 15 shows briefly how each filter and edge detection affects the effectiveness of recognition performance. Here, we can conclude that Laws' masks filters are the best filtering techniques compared to the average filter and the median filter for ASL recognition. Besides, we also can say that the Canny algorithm is the best compared to Sobel, Prewitt, and Roberts algorithm. Overall, the combination of Laws' masks filters and the Canny algorithm is the best technique to be used as the combination gives the highest performance among all which is 54.7%.

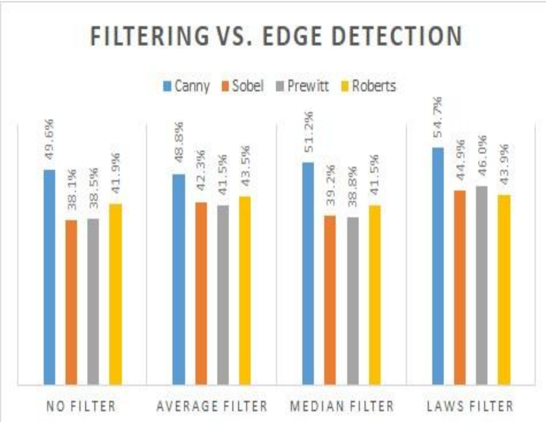

Figure 15: Filtering vs edge detection

4. CONCLUSION

The Laws' masks filters were created as one of the methods for texture analysis in image processing and are used in many applications [36]. In some applications, the Laws' masks filters are considered to be interesting to extract the image's texture characteristics. They also added that the Laws' masks filters seem capable of revealing subtle variations due to different masks that seem to explore different properties of the image. Meanwhile, the ideal detector is commonly recognized as Canny. Canny is better detection of edges and capable of detecting edges very robustly [37]. The Canny algorithm extracts picture features without disturbing picture characteristics. Due to the property of revealing variations in an image by Laws' masks filters and the advantage of Canny detection, then the combination of Laws' masks filters and the Canny algorithm gives better performance in this analysis. Specifically, in Laws' masks filters, the 2D mask produced from the product of ripple and edge gives the highest result of recognition performance. These experiments have also proven that the combination of Laws' masks filters and the Canny algorithm is robust, and this can be seen from the value of lower standard deviation compared to other combinations. Although the experiments in this study give a low performance, a good classifier such as deep learning may increase the recognition rate. In future work, we can look at the ensemble of Laws' masks filters to increase recognition accuracy.

ACKNOWLEDGMENT

Our deepest gratitude and thanks to Universiti Teknikal Malaysia Melaka (UTeM) and the Ministry of Higher Education Malaysia for funding this research grant under Short Term Research Grant (Grant no: PJP/2019/FTMK(6B)/S01678).

REFERENCES

[1] A. Young, R. Oram, and J. Napier, "**Hearing people perceiving deaf people through sign language interpreters at work: on the loss of self through interpreted communication**," *J. Appl. Commun. Res.*, vol. 47, no. 1, pp. 90–110, 2019. https://doi.org/10.1080/00909882.2019.1574018

[2] M. L. Hall, W. C. Hall, and N. K. Caselli, "**Deaf children need language, not (just) speech**," *First Lang.*, vol. 39, no. 4, pp. 367–395, 2019.

[3] I. Journal, A. Engineering, M. Kumar, and S. Grade, "**Conversion of Sign Language into Text**," vol. 13, no. 9, pp. 7154–7161, 2018.

[4] W. Tao, M. C. Leu, and Z. Yin, "**American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion**," *Eng. Appl. Artif. Intell.*, vol. 76, no. July, pp. 202–213, 2018.

[5] D. M. Viswanathan and S. M. Idicula, "**Recent Developments in Indian Sign Language Recognition : An Analysis**," *Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 1, pp. 289–293, 2015.

[6] C. Oz and M. C. Leu, "**American Sign Language word recognition with a sensory glove using artificial neural networks**," *Eng. Appl. Artif. Intell.*, vol. 24, no. 7, pp. 1204–1213, 2011. https://doi.org/10.1016/j.engappai.2011.06.015

[7] Q. Munib, M. Habeeb, B. Takruri, and H. A. Al-Malik, "**American sign language (ASL) recognition based on Hough transform and neural networks**," *Expert Syst. Appl.*, vol. 32, no. 1, pp. 24–37, 2007.

[8] L. Cadena, A. Zotin, F. Cadena, A. Korneeva, and A. Legalov, "**Noise Reduction Techniques for Processing of Medical Images**," in *Proceedings of the World Congress on Engineering*, 2017, vol. I, pp. 5–9.

[9] H. P. Singh, A. Nigam, A. K. Gautam, A. Bhardwaj, and N. Singh, "**Noise Reduction in Images using Enhanced Average Filter**," *Int. J. Comput. Appl.*, pp. 25–28, 2014.

[10] A. S. Alon, J. Rufo I. Marasigan, J. G. N.- Mindoro, and Cherry D. Casuat, "**An Image Processing Approach of Multiple Eggs' Quality Inspection**," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 6, pp. 2795–2799, 2019. https://doi.org/10.30534/ijatcse/2019/18862019

[11] D. Sherlin and D. Murugan, "**Brain tumor segmentation using modified fuzzy metric based approach with adaptive technique**," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 6, pp. 2730–2734, 2019. https://doi.org/10.30534/ijatcse/2019/08862019

[12] G. H.D. and D. D.K.Kirange, "M**achine Learning Approach towards Tomato Leaf Disease Classification**," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 1, pp. 490–495, 2020.

[13] A. Odat, M. A. Oair, and F. Shehadeh, "**Image denoising by comprehensive median filter**," *Int. J. Appl. Eng. Res.*, vol. 10, no. 15, pp. 36016–36022, 2015.

[14] Z. Wang and Z. Chen, "**Fingerprint Image Enhancement Method based on Adaptive Median Filter**," *2018 24th Asia-Pacific Conf. Commun.*, pp. 40–44, 2018.

[15] S. K and R. T, "**Enhanced Adaptive Weighted Mean Filter With Non-Local Means To Denoise Image Corrupted By Salt And Pepper Noise**," *Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 5, pp. 4751–4752, 2015.

[16] T. Sadhukhan, S. Chatterjee, R. K. Das, S. Ghosh, D. Das, and J. Sanyal, "**Efficient Removal of Noise from an Image using HSV Filtering**," *2019 Glob. Conf. Adv. Technol. GCAT 2019*, pp. 6–9, 2019.

[17] M. Sodanil, S. Nuchitprasitchai, and C. Intarat, "**Night image enhancement using selective filters**," *Proc. - 2017 2nd Int. Conf. Multimed. Image Process. ICMIP 2017*, vol. 2017-Janua, no. 3, pp. 151–155, 2017.

[18] R. Shenbagavalli and K. Ramar, "**Classification of Soil Textures Based on Law ' s Features Extracted from Preprocessing Images on Sequential and Random Windows**," no. August, pp. 4–8, 2017.

[19] Kenneth I. Laws, "**Texture Energy Measures**," in *Image Understanding Workshop*, 1979.

[20] S. Dash and U. R. Jena, "**Multi-resolution Laws ' Masks based texture classification**," *J. Appl. Res. Technol.*, vol. 15, no. 6, pp. 571–582, 2018. https://doi.org/10.1016/j.jart.2017.07.005

[21] A. Murali, W. V Stoecker, and R. H. Moss, "**Detection of Solid Pigment in Dermatoscopy Images using Texture Analysis**," *Ski. Res Technol.*, vol. 6, no. 4, pp. 193–198, 2000.

[22] K. Kamal, R. Qayyum, S. Mathavan, and T. Zafar, "**Advanced Engineering Informatics Wood defects classi fi cation using laws texture energy measures and supervised learning approach**," *Adv. Eng. Informatics*, vol. 34, no. September, pp. 125–135, 2017.

[23] N. R. Hernández, "**Structural analysis of textures based on LAW ´ s filters**," in *IEEE*, 2016, pp. 1–5.

[24] S. Dash and U. R. Jena, "**Texture classification using Steerable Pyramid based Laws ' Masks**," *J. Electr. Syst. Inf. Technol.*, vol. 4, no. 1, pp. 185–197, 2017.

[25] F. Kasmin, A. Abdullah, and A. S. Prabuwono, "**Ensemble of steerable local neighbourhood grey-level information for binarization**," *Pattern Recognit. Lett.*, vol. 98, pp. 8–15, 2017.

[26] A. L. C. Barczak, N. H. Reyes, M. Abastillas, A. Piccio, and T. Susnjak, "**A New 2D Static Hand Gesture Colour Image Dataset for ASL Gestures**," *Res. Lett. Inf. Math. Sci.*, vol. 15, pp. 12–20, 2011.

[27] L. Shapiro and G. Stockman, **Computer Vision**. Pearson, 2001.

[28] N. R. Kumar and J. U. Kumar, "**A Spatial Mean and Median Filter For Noise**," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 4, no. 1, pp. 246–253, 2015.

[29] J. Ekbote and M. Joshi, "**Indian sign language recognition using ANN and SVM classifiers**," *Proc. 2017 Int. Conf. Innov. Information, Embed. Commun. Syst. ICIIECS 2017*, vol. 2018-Janua, pp. 1–5, 2018.

[30] J. Canny, "A **Computational Approach to Edge Detection**," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[31] R. Chandwadkar, S. Dhole, V. Gadewar, D. Raut, and S. Tiwaskar, "**Comparison Of Edge Detection Techniques**," *6th Annu. Conf. IRAJ*, no. 6, pp. 133–136, 2013.

[32] N. Dalal and B. Triggs, "**Histograms of Oriented Gradients for Human Detection**," *2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, pp. 886–893, 2005.

[33] I. Mahmud, T. Tabassum, P. Uddin, E. Ali, A. M. Nitu, and M. I. Afjal, "**Efficient Noise Reduction and HOG Feature Extraction for Sign Language Recognition**," *2018 Int. Conf. Adv. Electr. Electron. Eng.*, pp. 1–4, 2018. https://doi.org/10.1109/ICAEEE.2018.8642983

[34] C. Cortes and V. Vapnik, "**Support-vector Networks,**" *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[35] F. F. Chamasemani and Y. P. Singh, "**Multi-class Support Vector Machine (SVM) classifiers - An application in hypothyroid detection and classification**," *Proc. - 2011 6th Int. Conf. Bio-Inspired Comput. Theor. Appl. BIC-TA 2011*, pp. 351–356, 2011.

[36] M. Rachidi, A. Marchadier, C. Gadois, E. Lespessailles, C. Chappard, and C. L. Benhamou, "**Laws' masks descriptors applied to bone texture analysis: An innovative and discriminant tool in osteoporosis**," *Skeletal Radiol.*, vol. 37, no. 6, pp. 541–548, 2008.

[37] S. Kaur, "**Comparison between Edge Detection Techniques**," vol. 145, no. 15, pp. 15–18, 2016. https://doi.org/10.5120/ijca2016910867