A New Fusion Feature Selection Model (FFSM) based Feature Extraction System for Hand Gesture Recognition

S. Subhashini[1] and S. Revathi[2*]

[1] Assistant Professor (Senior Grade), Department of Computer Science and Engineering, B.S Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India. subhashini10@gmail.com, Orcid: https://orcid.org/0000-0003-4297-8528

[2*] Professor, Department of Computer Science and Engineering, B.S Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India. revathisathiya2202@gmail.com, Orcid: https://orcid.org/0000-0001-9584-5089

Received: August 22, 2023; Accepted: October 25, 2023; Published: December 30, 2023

Abstract

The research presented here proposes a unique approach for exact feature extraction that makes use of the Fusion Feature Selection Model (FFSM). The method encompasses various stages to preprocess and extract crucial features from first-person hand action (FHPA) images. Preprocessing involves video-to-frame conversion, RGB to grayscale conversion, an improved median filter, and Gaussian blur-based image smoothing. Segmentation is achieved using the Improved SwinNet to identify meaningful regions within the images. Feature extraction employs the Gabor Line Derivative (GLD) method, Active Shape Model (ASM), and Histogram of Oriented Gradients (HOG) to capture texture, edge, and shape information, respectively. Extensive experimental evaluations demonstrate the effectiveness of our proposed approach, achieving remarkable performance in accurate feature extraction tasks.

Keywords: Gabor Line Derivative (GLD), Swin Net, Active Shape Model (ASM), Histogram of Oriented Gradients (HOG).

1 Introduction

Feature extraction is an arduous process that is subsequently processed for classification and recognition based on the applications. For skin color-based feature extraction techniques, the input RGB image is transformed into an HSV image to detect the hand, which has Hue (H), Saturation (S), and Value (V) components depending on skin colour. Each pixel in the image has an HSV value, which is compared to the HSV value of the skin colour. This comparison is based on the value of a skin pixel and a predetermined threshold value, with the threshold value being used to identify which pixel is significant to the accurate feature. (De Dios, J.J., 2007)

To combat skin color noise, skin color-based filtering algorithms are used for the segmentation of the hand region on uniform backdrops. When a hand is surrounded by items with any shade of skin and intricate backdrops, segmentation is still challenging (Sahoo, J.P., 2023). The performance of gesture recognition is impacted by issues such as light variation, hand segmentation, hand shape, and gesture

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 14, number: 4 (December), pp. 149-163. DOI: *10.58346/JOWUA.2023.I4.011*

*Corresponding author: Professor, Department of Computer Science and Engineering, B.S Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India.

poses with inter-class similarity. Grey-world techniques and homomorphic filtering are recommended for light normalization. The reported strategies dramatically lower the gesture recognition performance for a wide number of gesture classes, such as American Sign Language (ASL) (Benitez-Garcia, G., 2021). To improve human-computer interaction, semantic information from hand gestures has been extracted using both traditional computer vision approaches and machine learning techniques over the years. Hand gesture detection, segmentation, and recognition are some of the key topics of research on hand gestures (Lazarou, M., 2021).

Gestures are a form of communication between humans and computers, and gesture recognition is the computer's analysis of each gesture's unique meaning. The expression of humans can be obtained using gesture recognition, which can subsequently be used to achieve the objective of intuitive and intelligent man-machine interaction (Lawal, S., 2020). The gesture structure, contour, edge and other aspects are recognized using a geometric feature-based recognition technique (Rasel, A.A.S., 2019).

2 Related Works

In 2021, Parvathy*et al.* suggested a machine learning-based hand gesture recognition (HGR) system. The three stages of this suggested system are segmentation, feature extraction, and classification. DWT (Discrete Wavelet Transform) with modified SUR (Speed Up Robust) Rotation and scale invariant key descriptors were extracted using the feature extraction technique.

In 2018, Zhang *et al.* presented a novel method for Dynamic Continuous Hand gesture identification using a Frequency Modulated Continuous Wave radar sensor is given to overcome the issues. The lighting, noise, or atmospheric conditions have no impact on the radar system.

In 2019, Zhang *et al.* presented as EMG-based model for real-time hand gesture identification. They capture EMG signals using an armband and then utilize a sliding window method to segment the data before extracting characteristics.

In 2019, Rasel, *et al* presented a system which is utilized to recognize the hand gesture by SVM with an oriented gradient’s histogram.

In 2020, Qi *et al.* introduced electromyography equipment that collects the surface EMG signal of the arm from nine static gestures as samples and analyses them to derive four different types of signal characteristics.

In 2020, RafidMostafiz et *al.* created a technology to aid medical professionals in diagnosing illnesses and making judgements. It was suggested that a chest X-ray image, DCNN and Discrete Wavelet Transform (DWT) may be used to find Covid-19. The image was first submitted for enhancement and the features of DCNN and DWT were extracted after segmentation as part of the pre-processing stage.

3 Proposed Methodology

Our proposed methodology aims to achieve accurate Feature extraction through a multi-step approach. First, we gather a diverse dataset of first-person hand action (FHPA) videos or images, comprising various hand gestures for system recognition. In the pre-processing stage, we convert videos to frames and transform RGB images into grayscale to reduce computational complexity. To improve image quality, we introduce an improved median filter to effectively reduce noise while preserving edges. Additionally, Gaussian blur-based image smoothing is applied to further enhance the image clarity. Segmentation is accomplished using the novel Improved SwinNet, breaking images into meaningful regions. For feature extraction, we employ a Fusion Feature Selection Model (FFSM) incorporating the Gabor Line Derivative (GLD) method to capture texture and edge information. Active Shape Model (ASM) captures shape information, and Histogram of Oriented Gradients (HOG) represents shape and contour details. Our comprehensive methodology promises to achieve superior performance in accurate feature extraction tasks. Figure 1 shows the overall flow diagram of the proposed model.

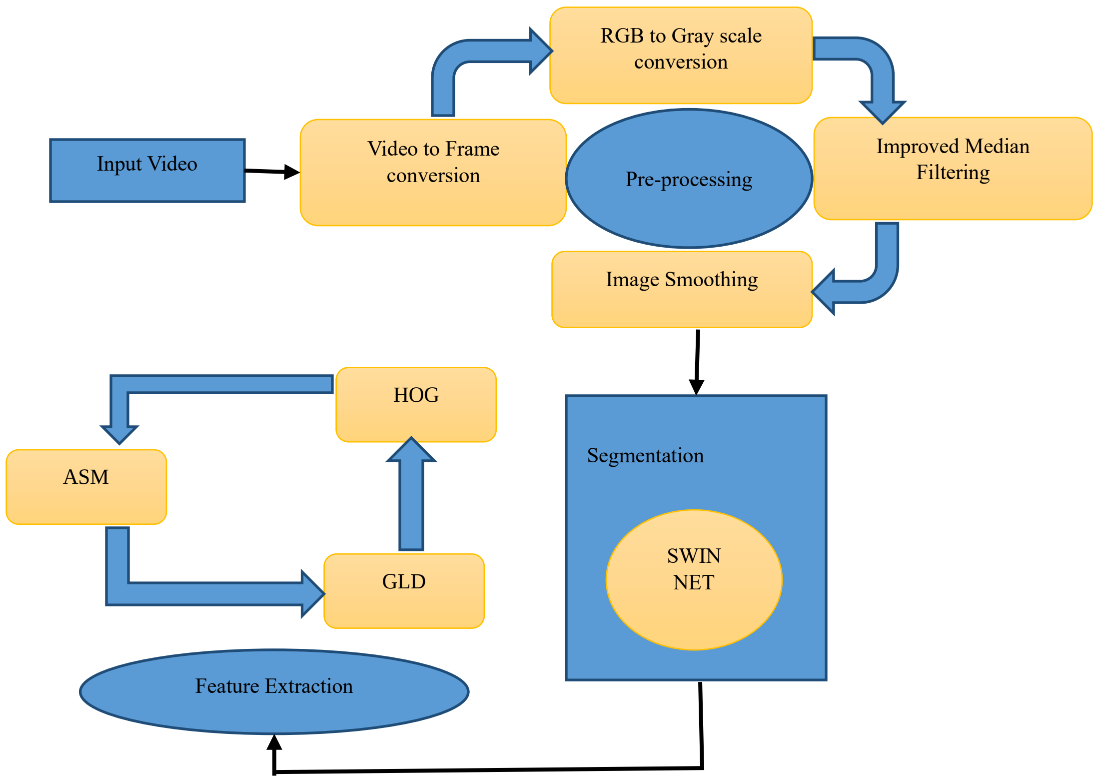

Figure 1: Overall Flow Diagram of the Proposed Model

Pre-processing

In this research work, the collected raw videos are pre-processed via video-to-fame, RGB to gray scale conversion, Median filtering and image smoothing. The architecture of the pre-processing stage is shown in Figure 2.

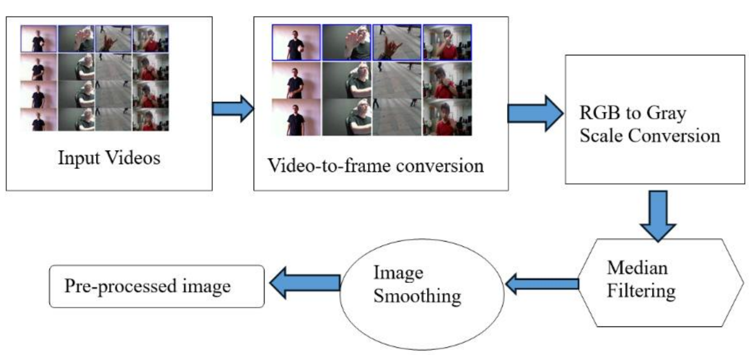

Figure 2: Pre-processing Phase

These pre-processing methods are discussed below

Video-to-frame Conversion

Video-to-frame conversion is the initial step of pre-processing. In this level, the raw video is converted into the frame. A video is a form of electronic media that may be used for recording, broadcasting, sharing, viewing, advertising, and showing visible effects. The picture, sometimes known as frame informally, is an essential component of the video. The most significant multimedia is the video. The term "group of pictures" refers to a grouping of frames. Video analysis and segmentation are more crucial than ever because of the quick development of video indexing, editing, and transcoding systems. A large video clip can be divided into smaller units for further processing using a technique called frame extraction.

The frequency at which an imaging system creates distinct, sequential pictures known as frames is known as frame rate. Both photography and video films record a succession of frames (still pictures). For instance, PAL offers 25 frames per second whereas the NTSC system transmits 29.97 frames per second. A video-to-frame conversion is the most used program. Due to the disparity in frame rates between film (often 24 frames/s) and television (30 or 25 frames/s), playing a film into a television camera alone would result in flickering when the film frame was adjusted in the middle of the TV frame. The smoothness of object motion is improved via frame conversion and the playback speed can be slowed down to bring out the motions of the objects.

RGB to Grayscale Conversion

Onto the image frames, the RGB to gray conversion is applied. This is the second step of pre-processing. This conversion turns RGB images into grayscale by keeping the brightness but removing saturation and hue data. A grayscale picture may be seen as a single layered image, as opposed to an RGB image, which can be seen as three images (a red, green, and blue scale image) piled on top of one another. Our main concept is to calculate a weighted total of all three colours to convert an RGB image pixel, which is a triplet value corresponding to the red, blue, and green colour components of an image at a particular spatial location, to a single value. It aids in the simplification of algorithms and also gets rid of difficulties brought on by computing demands. It provides an opportunity for individuals who are new to image processing to learn more quickly. This is because grayscale compression reduces a picture to its most basic pixel. It improves simple visualization. Some benefits of RGB to greyscale conversion are given below:

- While storing a single colour pixel of an RGB image requires 8*3 = 24 bits (8 bits for each colour component), doing so when an RGB image is turned into a grayscale image just requires 8 bits. A grayscale image will therefore take up 33% less memory than an RGB image.
- The usage of grayscale images in a variety of tasks, such as image segmentation and morphological procedures, where it is easier to deal with a single-layered grayscale image than a three-layered RGB colour image.

Fractional-Order Adaptive Median Filter with Thresholding

This concept combines fractional-order calculus, adaptive thresholding, and rank-based filtering to achieve robust noise reduction and edge preservation.

Mathematical Expression

Given an input image $I(\mathrm{x,y})$, the proposed Fractional-Order Adaptive Median Filter with Thresholding operates as follows:

For each pixel (x,y) in the image: Calculate the fractional-order gradient magnitude $G_\alpha(x,y)$ using Grünwald-Letnikov fractional differ integral operators

$$G_\alpha(x,y)=sqrt(D_{x\alpha}(x,y)^\wedge 2 + D_{y\alpha}(x,y)^\wedge 2) \tag{1}$$

Calculate the adaptive threshold $T(x,y)$ using the median of gradient magnitudes in the local window $W(x,y)$ for effective image processing.

$$T(x,y) = median \{G_\alpha(i,j):i,j)in\ W(x,y)\} \tag{2}$$

Retrieve pixel data $P(i,j)$ from the surrounding window W(x,y)

When $G_\alpha(x,y)>T(x,y)$, compute the median from $P(i,j)$ for (x,y) in the filtered image. Otherwise, employ thresholded median calculation.

$$Median_{th}(x,y) = median\{P(i,j): G_\alpha(x,y)>T(x,y))\} \tag{3}$$

Assign thresholded median $Median_{th}(x,y)$ to pixel (x,y) in the output image.

Incorporating thresholding makes the filter adjust median computation according to local image details. When the gradient magnitude surpasses the adaptive threshold, a regular median calculation is applied. Alternatively, a thresholded median calculation is employed, considering solely pixel values with gradients exceeding the threshold.

Image Smoothing

The final step of pre-processing is image smoothing. In this stage, Noise-free images which are obtained from median filtering are applied for image smoothing. It is a digital image processing method that minimizes and suppresses image noises, clutter, and sharpness from the image to provide a much more blended and smooth appearance. A crucial technique for improving images is called image smoothing, which may take out noise from the photographs. As a result, it is a necessary functional element in many image-processing programs. The quality of photos may be enhanced by using image smoothing. For human eyesight, image quality is a key component. The noise in the image is typically difficult to remove using image processing. The presence of noise has an impact on the image's quality. There are several ways to remove noise from photos. An image filter is used as a preprocessing module since many image-processing algorithms struggle in noisy environments. However, when placed in a very noisy environment, typical filters relying on pure numerical calculation quickly lose their capacity. The uses of image smoothing are described below:

An essential stage of image processing is image smoothing. A digital image with high spatial frequencies (Hz) should have the noise removed using the smoothing filter in low-pass filters (low pass filters). It is possible to emphasize and sharpen edges and minute features in images by using an image-sharpening approach. The image becomes less distinct or blurry as it grows smoother. Low-pass filters are utilized in smoothing techniques, whereas pixels that move through an image with a kernel average or median value are also employed. By reducing the sharpness of an image's corners and edges, image smoothing may be completed rapidly. The picture, on the other hand, has a problem with random noise. Contrarily, sharpening involves bringing edges into focus in order to enhance the image, even if it only has small defects. These actions will show the image enhancement. The advantages of image smoothing are as follows:

- In image processing, smoothing is used to lower noise and raise the image's general quality.
- The number of artifacts that could be present in an image can be decreased with the use of smoothing.
- A picture gets less blurry and clearer when a smoothing algorithm is applied to it.

Various techniques are employed for image smoothing. Gaussian blurring is the most effective method. In this method, a box filter is swapped out for a Gaussian kernel. By utilizing the cv.GaussianBlur() method, this is accomplished. The kernel should have a positive width and an odd height. Furthermore, we need to include sigmaX and sigmaY, which stand for the respective standard deviations of the X and Y axes. If only sigmaX is given, it is presumed that Sigma is the same as sigmaX. If one or both of them are given as zeros, the kernel size is used to determine them. Gaussian blurring is a useful technique for effectively removing Gaussian noise from images.

Segmentation

The pre-processed image is applied to segmentation. Image segmentation is the technique of dividing a digital image into several subgroups (pixels) known as Image Objects, which can reduce image complexity and facilitate image analysis. The most crucial step in image processing is segmentation. A complete image may be divided into many portions, making it easier to interpret and more meaningful. These various pieces will cover the complete image when they are reconnected. The segmentation process may also be influenced by other characteristics present in the picture. It might be either the texture or the color. A picture is segmented before it is denoised so that the original image may be recovered. The basic goal of segmentation is to simplify the data for easy analysis. Image analysis and image compression both benefit from segmentation. Various types of segmentation such as region, edge, model, feature-based, and thresholding are in practice. Among these edge-based segmentation is a more effective method.

Edge detection methods can also be used for segmentation. This method identifies the edges to segment. To find the visual discontinuities, edges are discovered. By recognizing the pixel value and comparing it to the surrounding pixels, the region's edges may be identified. Support Vector Machine (SVM) is used for this classification and has both fixed and adaptable features. The segmentation method utilized for remote sensing pictures offers a high spatial resolution. Segmentation is a two-step process that begins with the extraction of edge data from the edge detector and ends with the labeling of the pixels. The benefit of this method is that it can also get data from weak boundaries. Positional accuracy is increased by segmentation's spatial resolution. The picture is split based on edge flow. To segment a picture, it determines the direction in which a pixel's color and texture change. Edges can also be used for segmentation. Since it is not closed, there will be a little gap between the edges. Thus, edge connecting fills the gap in order to provide connectivity for segmentation, the broken edges are stretched along the link's slope-direction. In this research work, the SwinNet is proposed for the segmentation process.

Hierarchical Adaptive Swin Net with Adaptive Contextual Segmentation: Mathematical Expression

The algorithm for the proposed segmentation process is shown in Table 1.

Table 1: Algorithm for the Proposed Segmentation Process

Input image : *I (x ,y)*
Step1: //Hierarchical Feature Extraction and Adaptive Depth Selection
 	
for each (x,y) in $input_{image}$:
        $complexity_{measure}$ = CalculateComplexityMeasure(I(x,y))
        optimal $_{stages}$ = MapComplexityToStages(complexity_measure)

```
    for each stage in optimal stages :
            feature map = ConvLayers(input,num layers)
            input = feature map

Step 2://Adaptive Context Aggregation

for each stage in  Stages :
    context Kernel Size = ComputerContextKernelSize(input,stage)
    dilation rate = ComputeDilationRate(stage)
      feature map = ConvLayers(input,kernel size,dilation rate)
      input = feature map

Step 3: //Hierarchical Feature Refinement

for each stage in  Stages :
    feature map = ConvLayers(input,num layers)
    input = feature map

Step 4://Hierarchical Adaptive Contextual Segmentation

for each (x,y) in input image:
      complexity_measure = CalculateComplexityMeasure(I(x,y))
      optimal stages = MapComplexityToStages(complexity_measure)
       for each (x,y) in optimal Stages :
            feature_map = ConvLayers(input,num layers)
            context_ kernel_ size = ComputerContextKernelSize(feature_map,stage)
            dilation rate = ComputeDilationRate(stage)
              contextual_feature_map = ConvLayers(feature_map,context_kernel_size,dilation rate)
              refined_feature_map = ConvLayers(contextual_feature_map,num_layers)
              input = refined_feature_map
              segmentation_map = SegmentationLayer(refined_feature_map)
```

Utilizing input image *I(x,y)* the integrated method employs hierarchical feature extraction, dynamic depth choice, context aggregation, and iterative feature enhancement for comprehensive image analysis.

Step 1- Hierarchical Feature Extraction and Adaptive Depth Selection

The emphasis at this stage is on obtaining hierarchical features from the input image while dynamically determining the most suitable network depth according to the image's complexity. For each pixel (x,y) within the input image, the method calculates a complexity measure using functions like Calculate Complexity Measure, factoring in aspects such as texture, edge density, and more. This complexity measure guides the selection of optimal stages for feature extraction, labeled as optimal_stages, using a function like MapComplexityToStages. At each stage, ConvLayers are applied with specific layer counts, extracting features with diverse abstraction levels. This approach to hierarchical feature extraction empowers the network to capture intricate details and higher-level structures, customizing its operation to the image's intricacy.

Step 2-Adaptive Context Aggregation

The objective is to flexibly adjust convolutional layer receptive fields based on input image intricacy. Employing multiple stages, each applies ConvLayers with changing context_kernel_size and dilation_rate. Functions like ComputeContextKernelSize and ComputeDilationRate determine these parameters, factoring image complexity and stage index. This dynamic context aggregation directs the network to capture precise local details in simpler zones and broader spatial contexts in intricate areas. The outcome: a feature map enriched with context-aware insights.

In this context, ComputeContextKernelSize and ComputeDilationRate are functions that compute suitable context kernel size and dilation rate, contingent on image complexity and stage index

Step 3- Hierarchical Feature Refinement

Expanding on the prior stage, this phase improves the features obtained from adaptive context aggregation. As in earlier stages, ConvLayers with designated layers enhance the feature map, acquiring abstract and context-rich attributes. Hierarchical refinement ensures the network systematically enhances features to encompass intricate patterns and context-aware insights.

Step 4-Hierarchical Adaptive Contextual Segmentation

This pivotal phase amalgamates adaptive depth selection, contextual aggregation, and hierarchical feature refinement to enhance image segmentation precision. For every pixel (x, y), complexity measure and optimal_stages are recalculated, steering multiple stages. These entail ConvLayers, contextual adaptation (via context_kernel_size and dilation_rate), and hierarchical refinement. The outcome: a polished feature map encompassing in-depth insights into image characteristics and context. This refined map then enters the SegmentationLayer, culminating in further feature processing and the ultimate generation of the segmentation map.

By merging hierarchical feature extraction, adaptive depth selection, context aggregation, and feature refinement, the integrated method achieves precise and efficient image segmentation.

Feature Extraction via Fusion Feature Selection Model (FFSM)

Active Shape Model (ASM)

The shape of an item is represented by landmarks, which are a series of related locations that are each significant and present in the majority of the pictures under consideration, such as the position of the right eye. A sufficient quantity of characteristic points should be offered to cover the intricate features and form. A collection of landmarks makes a shape. The points in the forms are represented as vectors, with all of the x coordinates being followed by the y coordinates. Reduce the average Euclidean distance between shape points by using a correspondence transform to align one shape to another. This transform supports rotation, scaling, rotation, and translation. The midpoint of the stratified training forms is defined as the mean shape. The average form that is aligned to the location and size of the face as determined by a global face detector is where the ASM starts its search for facial landmarks. The ASM model's algorithm consists of the following steps:

(i) Identify the points
(ii) Giving all landmarks a grey profile.
(iii) Align the training regimen for ASM.

(iv) Compiling statistics for PCA at each resolution using an aligned training set.

(v) Repeat steps one through four for every resolution level.

After these procedures, X can be represented as follows:

$$X_i = \bar{X} + P.\, b \tag{4}$$

In above Eq. (4), $\bar{X}$ represents the ASM mean of the training set, P indicates that the most effective significant t techniques of variation are eigenvectors, and b is a coefficient vector. A collection of parameters for a flexible model is defined by the vector b. Using formulation (5), the shape by changed by varying the components of b. By applying bounds, it becomes

$$-3\mathrm{sqrt}(\lambda_i) < b_i < -3\mathrm{sqrt}(\lambda_i) \tag{5}$$

Where λ_i is the eigenvalue, it is guaranteed that the produced shapes will resemble the training samples. Iterations are used in the ASM search process. It searches for a new shape using the local appearance model in each iteration and then modifies the model's parameters to best match the newly discovered form.

Histogram of Oriented Gradients (HOG)

The HOG is a feature descriptor that describes how the gradient orientations are distributed throughout an image. By examining the local gradients, it can efficiently depict the shape and contour information of the hand motions. Image pixels are separated into cells that are each an identical size of 8x8 pixels. A 9-bin histogram with a starting range of 0 to 180 or 0 to 360 degrees ('unsigned' or' signed') is initialized for each cell.

Each pixel will cast a vote for the 9-bin histogram according to its orientation after the magnitude and orientation have been determined. Its equivalent magnitude will determine the total number of votes. Stronger magnitudes will, therefore, have a greater effect on the histogram. It may simplify the gradient components by arranging each cell's magnitude and orientation into a histogram, which yields a vector of just 9 values that represents the sum of the magnitudes of each bin. The gradient histogram quantizes the components of each cell in the picture, to put it another way. It's also crucial to understand that HOG stores distributions of local intensity gradients or edge orientations rather than information about gradient or edge placements. Normalization is required to solve this problem since light changes often influence the gradient. The cells are initially grouped into blocks, and normalization is then performed using all of the histograms in the block rather than normalizing each histogram separately. According to Dalal and Triggs, each block is constructed from 2x2 cells, with a 50% overlap between each block. Concatenating the histograms of the four cells in a block result in a vector with 36 components (4 histograms x 9 bins per histogram), which is then normalized by dividing it by its magnitude. L2-normalization, which is shown in Eq. (6) is the most often employed block normalization.

$$f = v / \mathrm{sqrt}(\|v\|^2{}_{2+}\, e^2) \tag{6}$$

Panning the window across the whole picture will decrease the calculation time for obtaining HOG descriptors. Additionally, it modifies the image's size so that hyperbolas may be seen at various sizes. This is significant since the size of hyperbolas varies depending on several factors, including the size of the target, the environment of the medium, and the actual setup of the system.

4 Results and Discussion

In this study, a new version of the Feature extraction technique, the Fusion Feature Selection Model (FFSM) is suggested for hand gesture recognition. This chapter includes the proposed model is compared with several existing algorithms, ASM and Grayscale. For the analyzing purpose, the metrics such as

Accuracy, Precision, Sensitivity, Specificity, F- measure, MCC, NPV, FPR, and FNR were considered for evaluation.

Performance Analysis

The various metrics are considered to evaluate the proposed technique with the existing algorithm. Those metrics are discussed below:

Accuracy

Accuracy is a statistic frequently used to assess a machine learning model's performance, particularly in classification tasks. It calculates the percentage of the model's total predictions that were accurate forecasts. In other words, accuracy indicates how frequently the model categorizes the data points correctly.

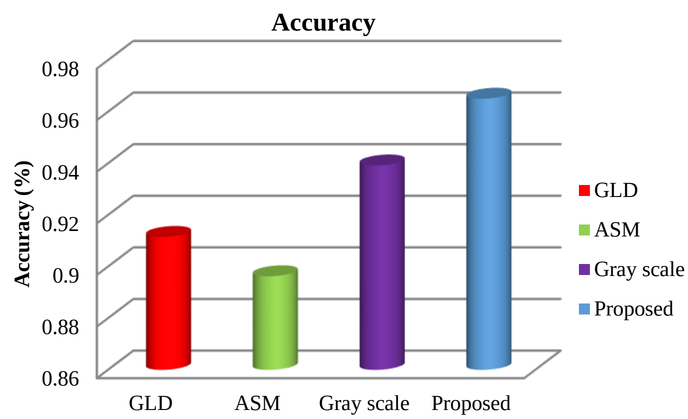

Figure 3: Accuracy of the Proposed System FFSM Compared with GLD, ASM, and Gray Scale

Comparing the proposed feature extraction to existing feature extraction algorithms like Gabor Line Derivative, Active Shape Model, and Grey Scale, the proposed feature extraction achieves 96.5% accuracy. Compared to the Gabor Line Derivative, Active Shape Model, and Grey Scale, our proposed method achieves 5%, 7%, and 2.58% higher accuracy respectively because of the Hierarchical Feature Extraction and Adaptive Depth Selection which is shown in figure 3.

Precision

In the context of binary classification issues, precision is a widely used metric in the fields of machine learning and statistics. It serves as a gauge for how well a model predicts the future. The ratio of true positive (TP) predictions to all of the model's positive predictions is used to calculate precision.

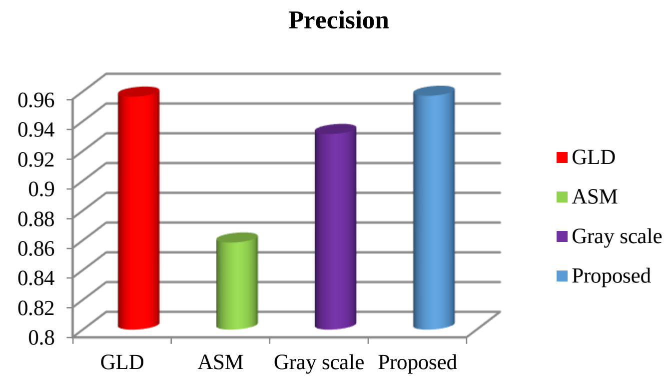

Figure 4: Precision of the Proposed System FFSM Compared with GLD, ASM and Gray Scale

When compared to existing feature extraction methods such as Gabor Line Derivative, Active Shape Model, and Grey Scale, the suggested feature extraction achieves 95.7%. Figure 4 shows that the precision value is enhanced to 0.1% over the GLD.

Sensitivity

Sensitivity, commonly referred to as recall or true positive rate, is a crucial statistic used to binary classification issues. It gauges a model's capacity to accurately pick out positive examples from among the real positive cases in the dataset.

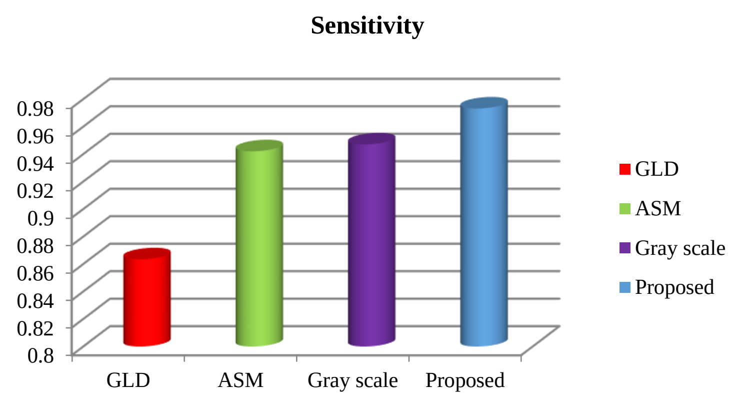

Figure 5: Sensitivity of the Proposed System FFSM Compared with GLD, ASM and Gray Scale

When compared to existing feature extraction methods such as Gabor Line Derivative, Active Shape Model, and Grey Scale, the suggested feature extraction achieves 97.3% sensitivity. Figure 5 shows that our suggested method outperforms the Gabor Line Derivative, Active Shape Model, and Grey Scale by 11%, 3.1%, and 2.6%, respectively.

Specificity

In binary classification issues, specificity is an essential additional parameter. It assesses a model's capacity to accurately detect negative occurrences among all the real negative examples in the dataset.

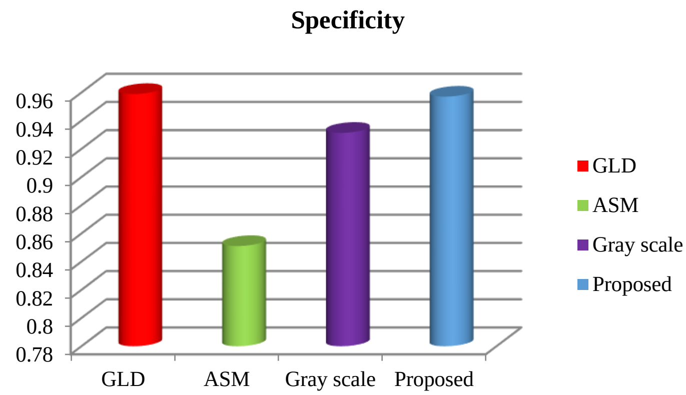

Figure 6: Specificity of the Proposed System FFSM Compared with GLD, ASM and Gray Scale

The proposed feature extraction obtains 95.7% when measured against existing feature extraction techniques such as GLD, ASM and Grayscale. Figure 6 demonstrates that the specificity value is 10% and 3% greater than ASM and Grayscale, respectively, and almost similar to GLD.

F-Measure

The F1 score, sometimes referred to as the F-measure or F-score, is a well-liked statistic used in binary classification issues to balance the trade-off between recall (sensitivity) and precision.

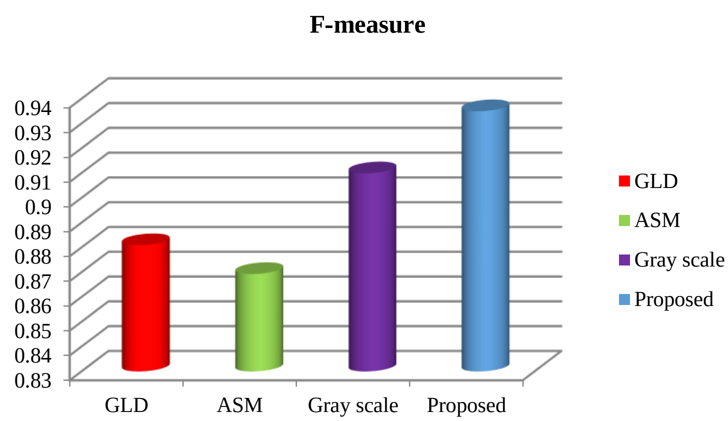

Figure 7: F-measure of the Proposed System FFSM Compared with GLD, ASM and Gray Scale

By accounting for both false positives and false negatives, it offers a single number that sums up a model's performance. When compared to existing feature extraction techniques such as GLD, ASM, and Grey Scale, the suggested feature extraction obtains 93.5% F-measure. Figure 7 shows that our suggested method outperforms the Gabor Line Derivative, Active Shape Model, and Gray Scale by 5%, 7%, and 2%, respectively.

Matthews Correlation Coefficient (MCC)

It is another statistic used to assess how well binary classification models work. The MCC provides a balanced measure of classification performance, especially in unbalanced datasets, by accounting for all four components of the confusion matrix (true positives, false positives, true negatives, and false negatives).

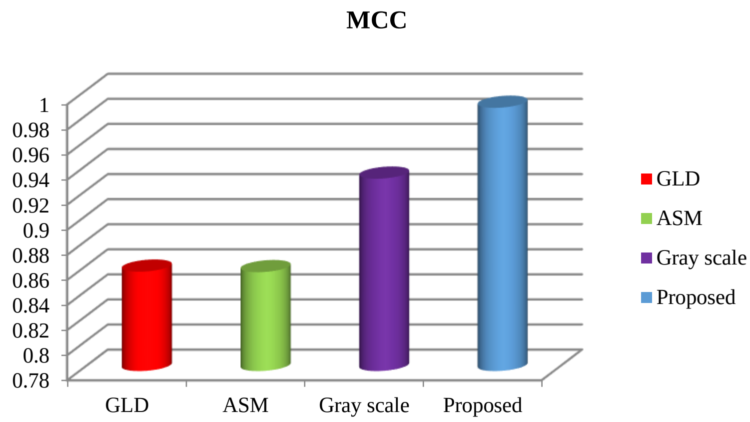

Figure 8: MCC of the Proposed System FFSM Compared with GLD, ASM and Gray Scale

The suggested feature extraction achieves 98.9% MCC when compared to existing feature extraction methods such as GLD, ASM, and Grey Scale. Figure 8 illustrates how our suggested solution outperforms the Gabor Line Derivative, Active Shape Model, and Gray Scale by 13%, 13%, and 5%, respectively.

Negative Predictive Value (NPV)

It is a statistic used to assess how well a binary classification model is working. Out of all the occasions when the model predicted a negative outcome, NPV represents the percentage of genuine negative predictions.

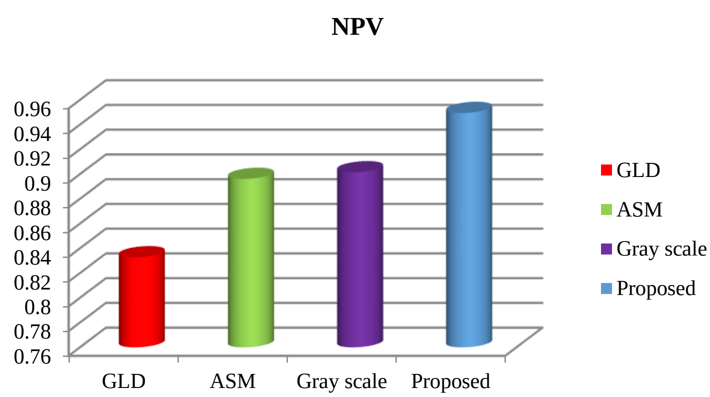

Figure 9: NPV of the Proposed System FFSM Compared with GLD, ASM and Gray Scale

The suggested feature extraction yields 94.9% NPV when compared to popular feature extraction methods like GLD, ASM, and Grey Scale. Our suggested solution delivers results that are correspondingly 11.7%, 5.3%, and 4.4% greater than the Gabor Line Derivative, Active Shape Model, and Grey Scale, as shown in Figure 9.

False Positive Rate (FPR)

It is a statistic used to assess how well a binary classification model is working. FPR calculates the percentage of false positive predictions among all truly negative occurrences.

False Negative Rate (FNR)

It is a statistic used to assess how well a binary classification model is working. FNR calculates the percentage of incorrect negative predictions among all cases where a positive outcome occurred.

The performance of the suggested technique is evaluated from the obtained data. Table 2 shows a feature extraction analysis of the suggested and existing algorithms such as GLD, ASM and Grayscale.

Table 2: Feature Extraction Analysis of Suggested and Existing Algorithm

	GLD	ASM	Grayscale	Proposed
Accuracy	0.911514	0.896256	0.939199	0.965073
Precision	0.956547	0.858508	0.931478	0.957130
Sensitivity	0.863460	0.942054	0.947107	0.973189
Specificity	0.958932	0.851065	0.931395	0.957066
F-Measure	0.880946	0.869312	0.909945	0.935004
MCC	0.859271	0.858843	0.933440	0.989983
NPV	0.832817	0.896169	0.901503	0.949475
FPR	0.020940	0.025597	0.027303	0.020643
FNR	0.004336	0.006857	0.005847	0.003618

Comparing the proposed feature extraction to existing feature extraction algorithms like Gabor Line Derivative, Active Shape Model, and Gray Scale, the proposed feature extraction achieves 96.5% accuracy, 95.7% precision, 97% sensitivity, 95.7% specificity, 93.5% F-measure, 98.99% MCC, and 94.9% NPV.

5 Conclusion and Future Work

In conclusion, this research proposes a novel and effective approach for precise feature extraction using the FFSM model. The methodology involves comprehensive preprocessing steps, including video-to-frame conversion, RGB to grayscale conversion, an improved median filter, and Gaussian blur-based image smoothing. The feature extraction process incorporates the Gabor Line Derivative (GLD) method, Active Shape Model (ASM), and Histogram of Oriented Gradients (HOG) to capture essential information from the segmented images. The accuracy of 96% is achieved through the improved version of the feature extraction and segmentation process. Extensive experimental evaluations demonstrate the effectiveness of the proposed approach, outperforming various existing techniques in accurate feature extraction tasks. The results affirm the potential of the suggested methodology in real-world applications and pave the way for more sophisticated and robust hand gesture recognition systems in the future.

The work further extends with deep learning network architecture for the classification and recognition of images.

References

[1] Benitez-Garcia, G., Prudente-Tixteco, L., Castro-Madrid, L.C., Toscano-Medina, R., Olivares-Mercado, J., Sanchez-Perez, G., & Villalba, L.J.G. (2021). Improving real-time hand gesture recognition with semantic segmentation. *Sensors*, *21*(2), 1-16.

[2] De Dios, J.J. (2007). Skin color and feature-based segmentation for face localization. *Optical Engineering*, *46*(3).

[3] Lawal, S., & Krishnan, R. (2020). Policy Review in Attribute Based Access Control-A Policy Machine Case Study. *Journal of Internet Services and Information Security (JISIS), 10*(2), 67-81.

[4] Lazarou, M., Li, B., & Stathaki, T. (2021). A novel shape matching descriptor for real-time static hand gesture recognition. *Computer Vision and Image Understanding*, *210*, 1-19.

[5] Parvathy, P., Subramaniam, K., Prasanna Venkatesan, G.K.D., Karthikaikumar, P., Varghese, J., & Jayasankar, T. (2021). Development of hand gesture recognition system using machine learning. *Journal of Ambient Intelligence and Humanized Computing*, *12*, 6793-6800.

[6] Qi, J., Jiang, G., Li, G., Sun, Y., & Tao, B. (2020). Surface EMG hand gesture recognition system based on PCA and GRNN. *Neural Computing and Applications*, *32*, 6343-6351.

[7] Rasel, A.A.S., & Yousuf, M.A. (2019). An efficient framework for hand gesture recognition based on histogram of oriented gradients and support vector machine. *Int J Inform Technol Comput Sci*, *12*, 50-56.

[8] Reza, M. M. R. (2020). Covid-19 detection in chest X-ray through random forest classifier using a hybridization of deep CNN and DWT optimized features.

[9] Sahoo, J.P., Sahoo, S.P., Ari, S., & Patra, S.K. (2023). DeReFNet: Dual-stream Dense Residual Fusion Network for static hand gesture recognition. *Displays*, *77*.

[10] Singha, J., Roy, A., & Laskar, R.H. (2018). Dynamic hand gesture recognition using vision-based approach for human–computer interaction. *Neural Computing and Applications*, *29*(4), 1129-1141.

[11] Wang, Y., Shen, J., & Zheng, Y. (2020). Push the limit of acoustic gesture recognition. *IEEE Transactions on Mobile Computing*, *21*(5), 1798-1811.

[12] Zhang, Z., Tian, Z., & Zhou, M. (2018). Latern: Dynamic continuous hand gesture recognition using FMCW radar sensor. *IEEE Sensors Journal*, *18*(8), 3278-3289.

[13] Zhang, Z., Yang, K., Qian, J., & Zhang, L. (2019). Real-time surface EMG pattern recognition for hand gestures based on an artificial neural network. *Sensors*, *19*(14), 1-15.

Authors Biography

Subhashini. S received the Master degree (M.E) in Computer Science and Engineering from SSN College of Engineering, Chennai, in 2008, and Bachelor degree (B.E) in Electronics and Communication Engineering from Noorul Islam College of Engineering, Kanyakumari district, Tamil Nadu, in 2005.She is currently pursuing PhD in Computer Science and Engineering and working as Assistant Professor(Senior Grade) in Department of Computer Science and Engineering at B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai since July 2008. Her area of interest are Computer Vision & Image Processing, Multimedia, Data Science, Artificial Intelligence and Psychology.

S. Revathi received the Ph. D degree in Information and Communication Engineering from Anna University, Chennai, in 2014.She is currently working as Professor in Department of Computer Science and Engineering at B.S.A Crescent Institute of Science and Technology, Chennai. She has published research work in reputed journals, conferences and book chapters. She guided and guiding PhD research scholars and engineering degree students for academic projects. Her area of research interest is Mobile Adhoc networks, Internet of Things and Deep learning.