Research Article

SOHAG JOURNAL OF SCIENCES

https://doi.org/10.21608/sjsci.2024.248702.1151

Robust Hand Gesture Recognition Using HOG Features and machine learning approach

Usama Sayed[1], *Samy Bakheet*[2],*Mahmoud A. Mofaddel*[2] *and Zenab El-Zohry*[*,2]

[1] Faculty of Engineering, Assiut University .
[2] Faculty of Computers and Artificial Intelligence, Sohag University.
[*]**Corresponding Author:** zenab_elzohry@yahoo.com.

Received: 27th November 2023, Revised: 15th January 2024, Accepted: 30th January 2024
Published online: 17th March 2024

Abstract:

Physical disability is one aspect of people that cannot be disregarded. A deaf person is someone who is naturally unable to hear.A unique language known as ”Sign-Language” is used to represent their expertise. Sign language is one of the most popular forms of deaf people to learn is American Sign Language (ASL).A collection of images of hands in various hand gestures or shapes are used in American Sign Language. In this study, we introduce feature-based algorithmic analysis to create a significant model for American Sign Language hand gesture identification. This model can be used to effectively learn in order to make a machine intelligent. We create a list of helpful features from digital images of hand gestures for efficient machine learning.For the pre-processing process, a histogram equalization technique and the an-isotropic diffusion filter are used. To extract image features a robust histogram of oriented gradient feature extraction method is proposed then three different machine learning classifiers are performed to achieve the classification process. To test our model, experiments are achieved using the American MNIST sign language dataset. With the use of HOG as feature and Support Vector Machine as classifier, the system yields by achieving high levels of sensitivity, specificity, and accuracy (99.8%, 98.9% and 99.6%, respectively). We can derive that the proposed model is an efficient sign language detection system. **keywords:**computer vision, machine learning,histogram oriented gradient , hand-gesture recognition.

1 Introduction

The field of recognition of patterns, computational vision, and biometrics has recently paid close attention to automatic vision-based hand gesture recognition because of its potential applications in a variety of fields, such as educated user interfaces for Communication between humans and machines, and sign language(SL) machine translation used by individuals With deep or severe hearing or speech deficits [1]. In light of a number of probable inherent obstacles provided by actual settings, including partial blocking, tough lighting alteration, major background disorder, extremely hand pose variability, huge intra-class variability throughout each class, and modifications in scaled, viewpoint, and physical appearance, the assignment of recognizing gestures made with hands in unhindered real-world situations has proven to be persistent, intractable, and especially difficult to accomplish [2] . In general, gestures are thought to be the earliest means of interaction between people because it’s probably that they were used by early humans long before spoken languages appeared.As with the previous point, the foundation of many automation technologies and real-world uses for intelligent vision includes virtual reality, motion gaming, intelligent surveillance, natural user interfaces, and natural user interfaces. These applications all rely on automatic gesture recognition. This is due to the need for automated mathematical model-based semantic interpretation of human gestures[3]. Owing to the structure of vision and the way synapses are structured during the course of human brain development, body language, or gestures, can be recognized by an individual with ease [4]. However, one of the most ambitious and challenging tasks for many researchers in computer vision—a field that is still very much in the forefront of study—is to replicate this kind of behavior, or talent, in computers.To finish this seemingly difficult task, a few potentially difficult issues must be resolved,such as which methods of classification and digital image capture technologies are more appropriate or practical to employ as opposed to others, along with how to precisely differentiate elements that are important from the foreground in emphasize images.

In our work, we propose a recognition model that applies various statistical image processing algorithms to the interpretation of sign language. A Histogram of Gradients features extraction method to extract dataset features then machine learning techniques SVM, KNN and a Decision tree for classification. In stationary human-computer user interface control systems, classical positioning histogram techniques for gesture-based recognizing have proved to be sufficiently useful. However, this usability is still restricted to a small subset of gestures, frequently sets of five or less (stop, right, left, up, down). Similar training data-based flaws plague neural network- and machine learning-based methods nowadays.

The paper's remaining sections are arranged as follows. Section 2 contains relevant works on hand gesture(HG) recognition. After that, Section 3 provides a thorough explanation of the suggested recognition of hand gestures(HG) framework. In Section 4, the experiments that were carried out to assess the performance of the gesture recognition system that is being presented are described in detail, and the results are discussed. Section 5 concludes with a discussion of the scope of future research.

2 Related Work

Throughout the previous 20 years or more, a great accord of discuss has been done—and still is—on the analysis, formalization, and recognition of hand gestures in still photos or video streams. Even after years of intensive work, this problem remains open and difficult for researchers in many other fields, including biometrics, pattern recognition, and computer vision communities. More thorough study is desperately needed to help develop original and inventive vision-based methods and processes that will help address the gesture recognition(HG) problem.

The literature states that there are two prime types of vision based recognition of gestures techniques: static and dynamic [5]. Hand gestures, also referred to as hand postures, are classified into a predetermined number of gesture categories by static hand gesture identity, which solely uses physical appearance and hand posture cues from nevertheless images—avoiding any movement signals. Thus, to adequately recognize static motions, only one image at the classifier's input needs to be processed [5,6].However, historical data is mainly used to simulate and recognize dynamic hand actions in order to identify the motion allusion of the gestures (i.e., hand identifying and tracking) [7,8]. Many studies on hand gesture identification have been published in the literature. These studies begin with image preprocessing and proceed through segmentation, extraction of features, classification, and other standard pattern analysis procedures [9,10]. A successful neural network-based multilayer perceptron (MLP) approach for recognizing static alphabetic gestures in Persian Sign Language (PSL) is shown in [11]. Using the discrete wavelet transform (DWT), wavelet features are extracted, and 94.06 percent recognition rate on average is achieved.

Similar to this, Cao et al. [12] present a method for recognising hand postures that integrates numerous heterogeneous picture characteristics and multiple kernels learning SVMs (SVM). They estimate the proper category for the input unobserved posture using multiple trained SVM kernels. The Jochen-Triesh hand posture dataset was used to test the model, and 99.16% profitable recognition accuracy was attained. Furthermore, in [13] the case of complex background items, a based on features visual attention framework is provided for hand posture detection and recognition. A mixture of high-level (texture and shape) and low-level (colors) image features form the basis of this structure. Posture categorization was achieved with an overall accuracy of 94.36% using the multi-class SVM. Analysis of contour form for hand posture recognition has been the subject of a substantial body of research. For example, in [5], the authors describe a contour-based feature method for identifying 14 sign language(SL) gestures. Utilizing temporal curvature analysis, the hand shape's silhouette is described,and an SVM classifier is trained using the features that were obtained. In numerous gesture recognition studies, convolutional neural networks (CNNs) have been thoroughly examined and have shown state-of-the-art achievement when contrasted to baseline methods [5, 14].For instance, a hybrid CNN-SVM approach is suggested in [14], where CNN and SVM are used as a gesture recognizer and feature extractor, respectively.

3 Proposed Methodology

This section presents a recognition model that applies various statistical computational vision algorithms to sign translation of languages. After extracting dataset features using a Histogram of Gradients, machine learning techniques are applied. For classification, use a decision tree, KNN, and SVM. In static human-computer connect systems of control, the traditional orientation histogram's methods for gesture-based recognition have reached a sufficient level of usability. Similar training data-based drawbacks are encountered by contemporary machine learning and neural network based techniques.

3.1 Pre-processing and Image enhancement

The histogram equalization (HE) technique is employed for the pre-processing step, because it is easy to use and provides good contrast enhancement [15]. This technique has been utilized in consumer electronics, image matching, medical image processing, speech recognition, and texture synthesis [16]. HE transforms the resulting image according to the test image's A measure of probability distribution.The image's dynamic range is distorted and flattened. Consequently, the altered image's overall disparity is enhanced. Then we improve the image quality by using anisotropic diffusion filter for removing image noise while preserving the image features [17]. Since the creation of Pirona and Malik [18] which anisotropic diffusion is used in place of isotropic diffusion, several methods have been proposed in tying adaptive eliminating to nonlinear parabolic differential equation systems in order to remove noise from images while maintaining significant structures. Anisotropic diffusion is related to an energy dissipation process seeking minimum functional energy. When the energy function is the parameter for the entire contrast within the image, an estimated total contrast reduction model can be obtained [19]. Although these techniques have demonstrated their ability to make a good trade-off between eliminating noise and edge safeguarding, the images obtained from applying these techniques in the presence of noise are frequently partly constant ; Thus,fine details and sub-regions of the original image may not be satisfactorily recovered.

3.2 Histogram of Oriented Gradients feature extraction

In image processing, the Histogram of Oriented Gradients (HOG) feature description method is frequently employed [20] and developed in computer vision for object detection by [21].The object of form and state is where the usual HOG idea originates. It can be identified by the distribution of pixel values' intensities and directions, and it is depicted as a vector known as a gradient vector. A gradient is a vector made up of components that collectively indicate the speed at which a pixel's value can change.Numerous helpful details can be found in the vector of gradients value. It displays the variation in each pixel's luminance value. A pixel's gradient vector value changes in proximity to an object's corner or edge. Therefore, the posture representation can be chosen with the help of the HOG characteristic.

The fundamental idea behind the HOG technique is to describe nearby objects in an image by using information about the distribution of edge directions or depth gradients. The operators of HOG are executed by partitioning an image into cell-named subregions. For every cell, we will generate a histogram of the gradient instructions for the factors.We obtain an illustration of the original image by the combined histograms. Locally histograms are able to be contrast-adjusted by identifying an intensity threshold in a region that is bigger than the cell calling blocks, which will improve recognition achievements. The threshold value will be used to normalize each cell in the block. A feature vector that is more lighting-invariant will be the result of the normalisation process. The steps to extract HOG's features are as follows:

1.computing each pixel's gradient vector. Pixel values in a grayscale image range from 0 to 255. If a pixel in the gradient vector has values on its left, right, above, and below sides, then that pixel is denoted by a distinct pair. Let I_x and I_y represent the two pairs of right and left, as well as up and down, pixels' respective values. The following formula is utilized to calculate the gradient vectors.:

$$Gradiant = \sqrt{I_X^2 + I_Y^2} \tag{1}$$

$$\theta = arctan \frac{I_X}{I_y} \tag{2}$$

2.constructing blocks.Blocks of the resulting image from the previous stage are equal in size. Every block has four identically sized cells with the same number of pixels in each. As illustrated in Figure 1, the constituent parts are piled on top of one another.. The following formula is used to determine the number of blocks:

$$n_{block} = (\frac{w_i - w_b \quad x \quad w_c}{w_c} + 1) x (\frac{h_i - h_b \quad x \quad h_c}{h_c} + 1) \tag{3}$$

where the image's width and height are represented by the variables w_i,h_i,w_b,h_b ,w_c and h_c correspondingly.

3.The computation of the indicative vector. For every block cell, the characteristic vector is computed. The number of usual vector dimensions for a cell (p) is used to split the directional space. The inclination angle of the pixel at coordinates (x,y) is discrete and divided into p bins. We applied an unsigned HOG differentiation (p=9) using the following formula.:

$$B(x,y) = round(\frac{p \quad x \quad \alpha(x,y)}{\pi}) mod p \tag{4}$$

Unsigning HOG (p =18), we have

$$B(x,y) = round(\frac{p \quad x \quad \alpha(x,y)}{2\pi}) mod p \tag{5}$$

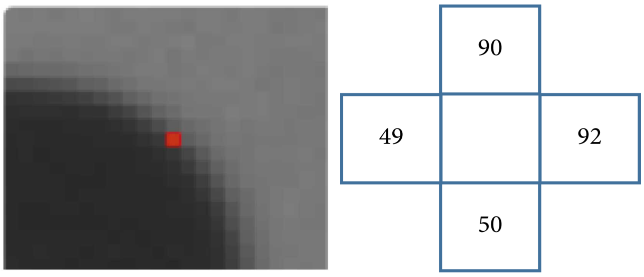

Figure 1: Computing gradient vector blocks.

where the total of the pixels' varying intensities determines the bin value. There are four cells each block. We can obtain a block's feature vector by joining four cells. The block's typical vector dimension is 4 p bin, where p = 9 for (unsigning HOG) or 18 (signing HOG).

4.computing the characteristic vector: By dividing by the magnitude of the blocks, we normalise the feature vector of the blocks. The HOG feature is created by combining the feature vectors from each block to create the image. The number of characteristic vector dimensions of the image is calculated by

$$Size_{featureimage} = n_{block/image} * size_{feature/block} \tag{6}$$

where $n_{block/image}$ is the block and $size_{feature/block}$ is the number of characteristic vector dimensions per block.

Our algorithm to extract HOG features from sign language images by calculating edge orientations in a local neighborhood of the image. It divides the image into small cells [2 × 2] and [4 × 4] to get spatial information and keep minor size features. For each cell, an edge orientation graph is to calculate the direction values evenly in bins between -180 and 180 degrees to differentiate between the light to dark and dark to light transitions within the image region as shown in figures 2 and 3. The graph channels are distributed evenly depending on the gradient. It performs the histogram counts to compensate for the illumination. To achieve this, gather more local histogram energy on the areas that are somewhat related, and then use the results to normalize all of the cells in the cluster. A combination of these graphs represents the ultimate HOG descriptor.

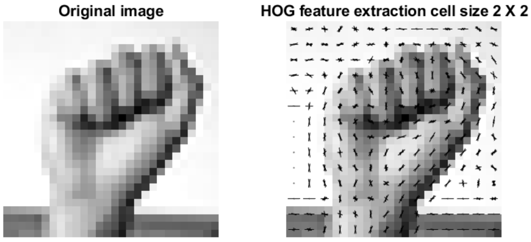

Figure 2: Extract American sign language MNIST dataset image features using HOG feature extraction to [2 × 2] cells.

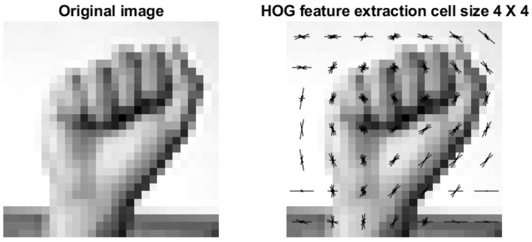

Figure 3: Extract American sign language MNIST dataset image features using HOG feature extraction to [4 × 4] cells.

3.3 Feature Classification

To classify the extracted features, we used the three most common different machine learning classifiers support vector machine [SVM], K-nearest neighbors [Knn] and decision tree. To find the best predictive model, the study compared the Decision Tree, SVM, Knn and Decision

Tree algorithms. After a training phase, a Support Vector Machine (SVM) is a supervised sorting method that can determine with the highest mathematical accuracy whether a new point belongs to a class or not.

3.3.1 Support vector machine [SVM]

Non-linear classifiers like Support Vector Machines (SVM) are known for their ability to yield better classification outcomes than other techniques. The concept underlying this approach is to allocate non-linearly input data to certain high-dimensional regions where linear data separation can produce superior performance (or regression). One difficulties SVM is a significant support vector that the training group uses for regression-based grading. The SVM training algorithm created a model that places new examples in either category if there is a set of training examples that are all marked as falling into one of two categories,As seen in Fig. 4, the SVM model represents examples as points in space that are arranged to divide examples into distinct categories into as wide a gap as feasible. Then, new examples are placed in the same area and are likely to fall into one of the categories according to which side of the gap they are on. By implicitly mapping their inputs into high-dimensional feature distances, SVM tools can effectively perform non-linear classification in addition to linear classification. This is known as the ”kern trick.”

3.3.2 K-nearest neighbors [Knn]

It is a method of organizing objects in the feature space according to the nearest practice examples. The function is only rounded up locally in k-NN, a form of example-based or lazy learning, and all computations are postponed until classification. The object is attributed to the most typical category amidst its closest neighbors, as determined by the majority of its neighbors’ voices (an integer k that is positive and usually small). The object is just placed in the category of the closest neighbor if k = 1. Selecting an odd value for k can be advantageous when dealing with binary sorting problems involving two categories, as it prevents restricted sounds, as illustrated in Figure 5. Regression can be performed using the same method, which entails setting the object’s property value to the average of its neighboring k values. Analyzing neighbor contributions to determine which neighbors are closest to you and which ones contribute more on average could be helpful. Neighbors are selected from a set of objects for which the drug’s value (or, in the case of regression, the correct classification) is known. This can be considered a set of training data for the algorithm, though a specific training step is not necessary. In order to locate neighbors, objects are described by position vectors in a multi-dimensional feature space. Although other distance measures, like Manhattan distance, can be used in its place, Euclidean distance is commonly used. The local data structure affects the nearest k algorithm.

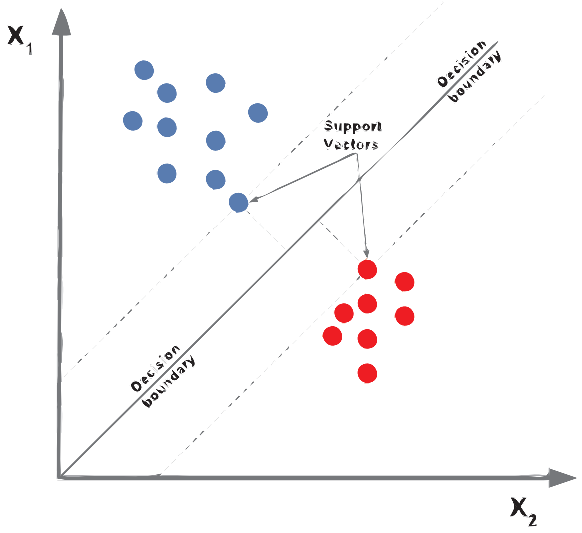

Figure 4: Support Vector Machines [SVM].

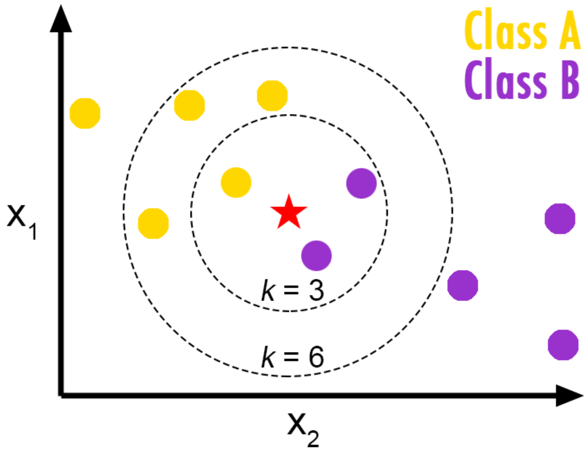

Figure 5: K-nearest neighbors [Knn].

3.3.3 Decision Tree[]

A classifier that is represented as a recursive dividing of an instance's space is called a decision tree. TThe decision tree's nodes work together to create a directed tree known as a ”rooted tree,” which has a node known as the ”root” that is devoid of any arriving edges. Every other node has exactly one incoming edge. A node with outgoing edges is called an internal or test node. The other nodes are decision nodes, also known as leaves. The request space was divided into two or more sub spaces by the internal nodes of a decision tree depending on a particular discrete function of the input attribute values. Each test examines a single attribute in the most basic and common scenario, partitioning the instance space based on the attribute's value. The condition with numerical attributes denotes a range. Every leaf has a class assigned to it that corresponds to the ideal target value. As an alternative, a probability vector indicating the likelihood that the target attribute will have a particular value may be stored in the leaf. Making use of the test results along the way, instances are categorized by working their way down from the tree's base to a leaf.

4 Experimental Results

4.1 American sign language (ASL) MNIST dataset

ASL dataset style formats closely fit the classic MNIST dataset as shown in Fig 6. Each test case and training reflects (0-25) as a single map for each alphabet A-Z as shown in (figure 7) data distribution.Though roughly half the size of a standard MNIST, the training data (27,455 cases) and test data (7,172 cases) are otherwise comparable to the header row, pixel1, pixel2,....784, representing an image measuring 28 × 28 pixels with values for grayscale among 0-255. The first hand gesture image data shows several users repeating gestures against various backgrounds. The expansion of the small number (1704) of color images listed as not clipped around the important hand area provided the MNIST sign language data. An ImageMagick-based image pipeline was used to generate new data, which involved manual scaling, gray-scaling, and cropping, followed by the creation of at least 50 variations to increase the quantity. The quantity of distinct letters found in the training set. Keep in mind that the dataset does not contain the letters J (9) or Z (25). It is evident that the data have an approximate distribution.

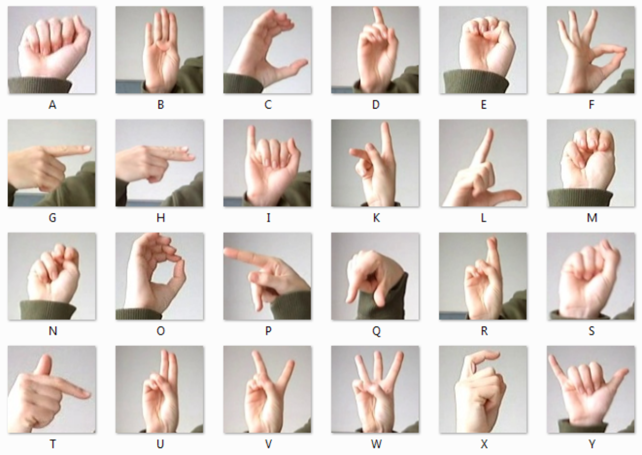

Figure 6: American sign language MNIST dataset distribution.

4.2 Results

To achieve the optimum efficacy of our proposed model, we use two various cell sizes of HOG features and classify the model performance with three machine learning classifiers. we have provided a complete view of how our model performs for different gestures selected as the signs language by the user. we test the models on the American Sign Language dataset. It is noticed that no pre-trained models were applied and the training process of the model was performed from scratch. The dataset is not balanced, as all the different gestures had almost a non similar number of training samples. Hence, we have used the accuracy and computation time of the sign detection of the model as the performance measure which can be represented as shown in figure 8. It observes that the machine learning model accuracy performs better and computation time processing in terms of using HOG feature extraction at [2 × 2] and [4 × 4] cell sizes using the support vector machine classifier.

The proposed technique yields results that are quantitatively evaluated in terms of three commonly used performance indices for performance evaluation: accuracy (AC), specificity (SP), and sensitivity (SN). The following is a definition of the three indices. In general, sensitivity measures the percentage of positives that are correctly identified; it is also known as recall or true positive rate.

$$SN = \frac{TP}{TP+FN} X100(\%) \tag{7}$$

Specificity (also called true negative rate)

$$SP = \frac{TN}{TN+FP} X100(\%) \tag{8}$$

This is how the accuracy of a given gesture recognition was calculated:

$$Accuracy(\%) = \frac{correctclassification}{totaltestdata} X100 \qquad (9)$$

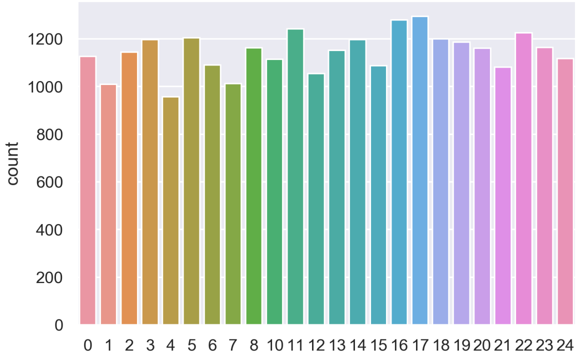

Figure 7: : American sign language MNIST dataset distribution.

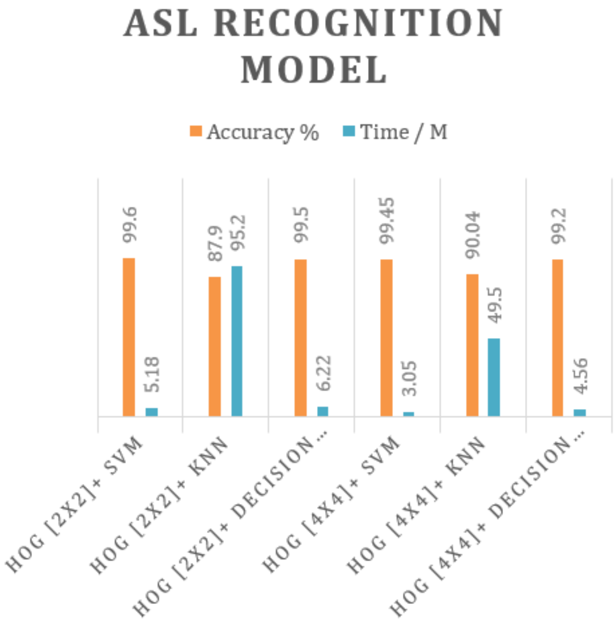

Figure 8: The Proposed sign language MNIST model Performance.

5 Conclusion

A conventional multi-class recognition is performed to the American sign language detection problem. For the pre-processing process, a histogram equalization technique and the anisotropic diffusion filter are used. To extract image features we used a robust histogram of oriented gradient feature extraction method is proposed then three different machine learning classifiers are performed to achieve the classification process. To test our model, experiments are achieved using the American MNIST sign language dataset. We can conclude that the proposed model is an efficient sign language detection system. It could capture the variations in various signs which look similar to the human eye. It has shown by experimental comparisons that the proposed SVM machine learning model accuracy outperforms the prior related works as shown from table 1 and other proposed KNN and decision tree classifiers of recognizing static hand gestures. While the proposed model performs better, in the time cost complexity when we extract dataset features at [4 × 4] cell sizes using the support vector machine classifier.With the use of HOG as feature and Support Vector Machine as classifier, the system yields by achieving high levels of sensitivity, specificity, and accuracy 99.8%, 98.9% and 99.6%, respectively). We can conclude that the proposed model is an efficient sign language detection system.

References

[1] S. Bakheet, A. Al-Hamadi, In J. Comput. Theor. Nanosci. 14(2), 1–10 (2017).

[2] S. K. Leem, F. Khan, S. H. Cho, In IEEE Trans. Instrum. Meas. 69(4), 1066–1081 (2020).

[3] R. Faugeroux, T. Vieira, D. Martinez, T. Lewiner, In 27th SIBGRAPI Conference on Graphics, Patterns and Images, (2014), pp. 133–140.

[4] J. Bransford, In National Academies Press, Washington, DC, (2000)

[5] Y. Ren, X. Xie, G. Li, Z. Wang, In IEEE Trans. Circ. Syst. Video Technol. 28(2), 364–377 (2018).

[6] Y. Zhang, C. Cao, J. Cheng, H. Lu, In IEEE Trans. Multimedia. 20(5), 1038–1050 (2018).

[7] . L. Baraldi, F. Paci, G. Serra, L. Benini, R. Cucchiara, in 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, (2014), pp. 702–707.

[8] . S. Mohatta, R. Perla, G. Gupta, E. Hassan, R. Hebbalaguppe, in IEEE Winter Conference on Applications of Computer Vision (WACV), (2017), pp. 330–335.

[9] S. Rautaray, A. Agrawal, In Intell. Rev. 43, 1–54 (2015).

[10] T. Orazio, R. Marani, V. Renò, G. Cicirelli, In Image Vis. Comput. 52, 56–72 (2016).

[11] A. Karami, B. Zanj, A. Kianisarkaleh, In Expert Syst. Appl. 38, 2661–2667 (2011).

Table 1: A comparison between our model performance and the prior related works on ASL MNIST dataset.

Related work	**Dataset**	**Method**	**Accuracy %**
Rathi [22]	ASL MNIST	MobileNet	95.06
		Inception$_V$3	93.36
Bakheet A. Al-Hamadi [23]	Senz3d	multiple shape cues	93.3
zhao and Wang [25]	ASL MNIST	CNN Networks	90
Proposed	ASL MNIST	HOG + SVM	**99.6**
		HOG + KNN	95.2
		HOG + D-Tree	99.5

[12] J. Cao, Y. Siquan, H. Liu, P. Li, In Multimedia Tools Appl. Springer. 75(19), 11909–11928 (2016).

[13] P. Pisharady, P. Vadakkepat, A. P. Loh, In Int. J. Comput. Vis. 101, 403–419 (2013).

[14] P. Ji, A. Song, P. Xiong, P. Yi, X. Xu, H. Li, J. Intell. Robot. Syst. 87(3-4), 583–599 (2017).

[15] Chen, S.-D.& Ramli, A. R. In IEEE transactions on Consumer Electronics 49, 1310–1319 (2003).

[16] Kim, M. & Chung, M. G. In IEEE Transactions on Consumer Electronics 54, 1389–1397 (2008).

[17] Bai, J. & Feng, X.-C. In IEEE transactions on image processing 16, 2492–2502 (2007).

[18] Perona, P. & Malik, J., In IEEE Transactions on pattern analysis and machine intelligence 12, 629–639 (1990).

[19] Rudin, L. I., Osher, S.& Fatemi, E. In Physica D: nonlinear phenomena 60, 259–268 (1992).

[20] Dalal, N. & Triggs, In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) 1 (2005), 886–893.

[21] Mizuno, K. et al. In IEEE Workshop on Signal Processing Systems (2012), 197–202.

[22] Rathi, In arXiv preprint arXiv:1805.06618 (2018).

[23] S. Bakheet, A. Al-Hamadi, In EURASIP Journal on Image and Video Processing (2021) .

[24] Rao, G. A., Syamala, K, Kishore, P. & Sastry, In Conference oF Signal Processing And Communication Engineering Systems (SPACES) (2018), 194–197.

[25] Zhao, Y. & Wang, L. In Ninth International Conference on Intelligent Control and Information Processing (ICICIP) (2018), 269–272.