Static Hand Gesture Recognition Using Histogram of Oriented Gradients

Roy Amante A. Salvador
Department of Computer Science
College of Engineering
University of the Philippines Diliman
Quezon City, Philippines 1101

Abstract—**Hand Gesture Recognition is a Computer Vision problem aimed in interpretting and classifying hand gestures captured in video or images. Gestures in general are important to understand as we use them to convey what we mean. Trends in technological devices have included Hand Recognition systems to automatically decode hand gestures and serve as interface. One of the most popular features to use not only in Hand Recognition systems but also in other Computer vision tasks is the Histogram of Oriented Gradients (HOG). In this paper, a real-time static Hand Gesture Recognition System is implemented. With the use of HOG as feature and Support Vector Machine as classifier, the system yields high recognition rate.**

Keywords—Hand Gesture Recognition, Histogram of Oriented Gradients, Support Vector Machine

I. Introduction

Hand Gesture Recognition is one of the interesting topics in Computer Vision and has been an emerging trend in technology. Gestures are now being used as one of the interfaces to various systems as each posture denotes a meaning without additional articulated information. Its application transcends to more than just technological advances in interfacing. Other example applications of a Hand Gesture Recognition System include Automatic Sign Language Transcription. Such system will definitely improve the quality of life of handicapped people. It can also aid in Emotion Detection as non-verbal components of communication give cues on one's emotional state.

Hand gestures can be categorized as static or dynamic. A static hand gesture or posture is one where there's no movement required to convey its meaning. It can be captured by just one image. Dynamic gestures on the other hand is captured by video or series of images and includes the time dimension. This project attempts on automatically classifying a subset of static American Sign Language (ASL) hand postures.

As in most pattern recognition systems, hand gesture recognition involves several steps including data acquisition, data pre-processing, feature extraction, and classification. In this implementation, an interface for capturing static hand postures in real time is included. Video frames are continuously processed by an image preprocessing pipeline and Histogram of Oriented Gradients (HOG) are calculated. HOG has been a widely used feature for image classification and object detection. These HOGs form the feature vector which then becomes input to a Multiclass Support Vector Machine. The system successfully performs recognition with overall accuracy reaching up to 87.07%.

II. Literature Review

Freeman and Roth [8] are said to be the pioneers in testing local orientation for hand gesture recognition. They developed a training set that contained up to 15 histograms with their local orientation of various gestures [10]. Orientation histograms as representation of hand gestures provided invariance to lighting and translations of the hand positions. Histogram of Oriented Gradients (HOG) is a popular feature descriptor used in computer vision particularly in object detection and recognition. This was introduced by Dalal and Triggs [2] which they initially used for pedestrian detection in static images but have expanded to human detection in video as well as detection of common animals and vehicles in images. In their paper, they have shown that HOG performs significantly better than the best Haar wavelet based detector by Mohan et al. [9]. They've concluded that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good performance when using HOG.

Although the trend now with Hand Gesture Recognition as well as other image recognition tasks is to use deep learning, the HOG feature vector is still widely used in Gesture Recognition Systems. Others use it as baseline for measuring performance of proposed methods [1] since it is an established feature. In 2011, the techniques developed for pedestrian detection were applied by Misra et al. [5]. Their system recognizes 7 hand gestures using HOG as descriptors. They've reduced dimensionality of HOG feature descriptors using Partial Least Square (PLS) which they found to be better than Principal Component Analysis (PCA). HOG feature extraction and multivariate SVM classification methods has been used by Feng and Yuan [3]. They've achieved high recognition rate with a system that copes better with illumination changes. Zhang and Liu [4] have used HOG combined with weighted Hu invariant moments as part of a dynamic gesture recognition system. They've also used SVM for the classification layer.

The use of HOG and SVM seems to be a usual combination in object detection and recognition systems. Bristow and Lucey [6] claim that the two have a symbiotic relationship. According to them, the HOG feature induces capacity and adds a prior to a linear SVM trained on pixels.

III. Methodology

A. Architecture

A conventional Hand Gesture Recognition System is often composed of the following stages - Data Acquisition, Preprocessing and Segmentation, Feature Extraction and Classification. This architecture is used as guideline in implementing this project as seen on Figure 1. The data capture and preprocessing stages are implemented in Python OpenCV 3.1.0 Computer Vision library while the Feature extraction and classification phases are implemented using Python SciPy Toolkits.

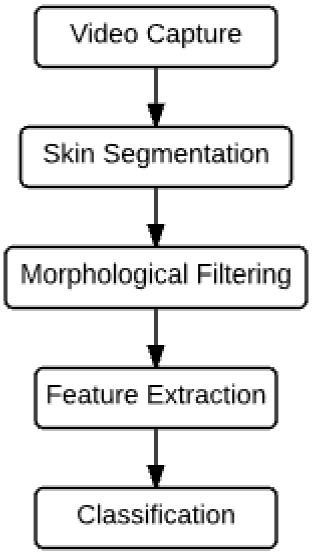


Fig. 1: Overview of Hand Gesture Recognition System Architecture

For a real-time system implementation, a video camera captures and streams raw images to the system. A set of preprocessing steps are performed to effectively determine the hand region and make it more robust. In this project, Skin Segmentation and a series of morphological filtering procedures are applied for each video frame. This allows the next stage to easily extract and represent the frame with Histogram of Oriented Gradients (HOG) feature. A feature vector is created and fed to the classification layer. A Multiclass Support Vector Machine performs the Hand Gesture class prediction which is made for every single frame in the video stream.

B. Video Capture and Dataset Acquisition

The system has an interface for getting the hand gesture of a user. It utilizes the default video capture device of the computer. Users perform the gestures in a controlled region within the entire frame. The defined region is expected to only have the left hand of the user and a simple background which does not contain skin colored areas.

For this project, the hand gesture classes consist of ten hand postures of the American Sign Language (ASL) - A, B, D, E, F, K, L, N, W and Y. Figure 2 shows the corresponding hand posture for each gesture class.

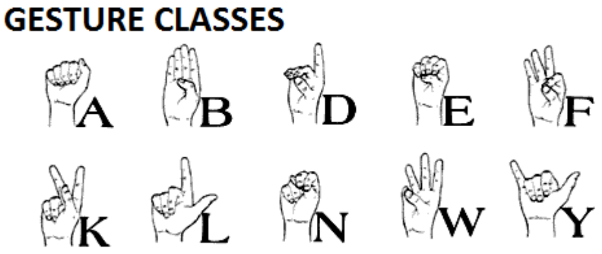


Fig. 2: Subset of the American Sign Language (ASL) [14] chosen in the project

Eight people were asked to hold static gestures using their left hand for more than ten seconds. They were recorded at ten frames per second and were asked to move around the frame and also towards and backwards the camera. This is to introduce positional, scale and some illumination invariance early on in the system. The resulting video clip is then sampled into images and tagged with the appropriate hand gesture class. Half of the collected clips for each class are used as training set while the other remaining half as test set.

C. Preprocessing

The preprocessing stage extracts the region of interest which is the hand and filters out some unwanted information such as noise to improve performance. Figure 3 demonstates the Preprocessing Pipeline. The following procedures are applied for each frame captured by the camera:

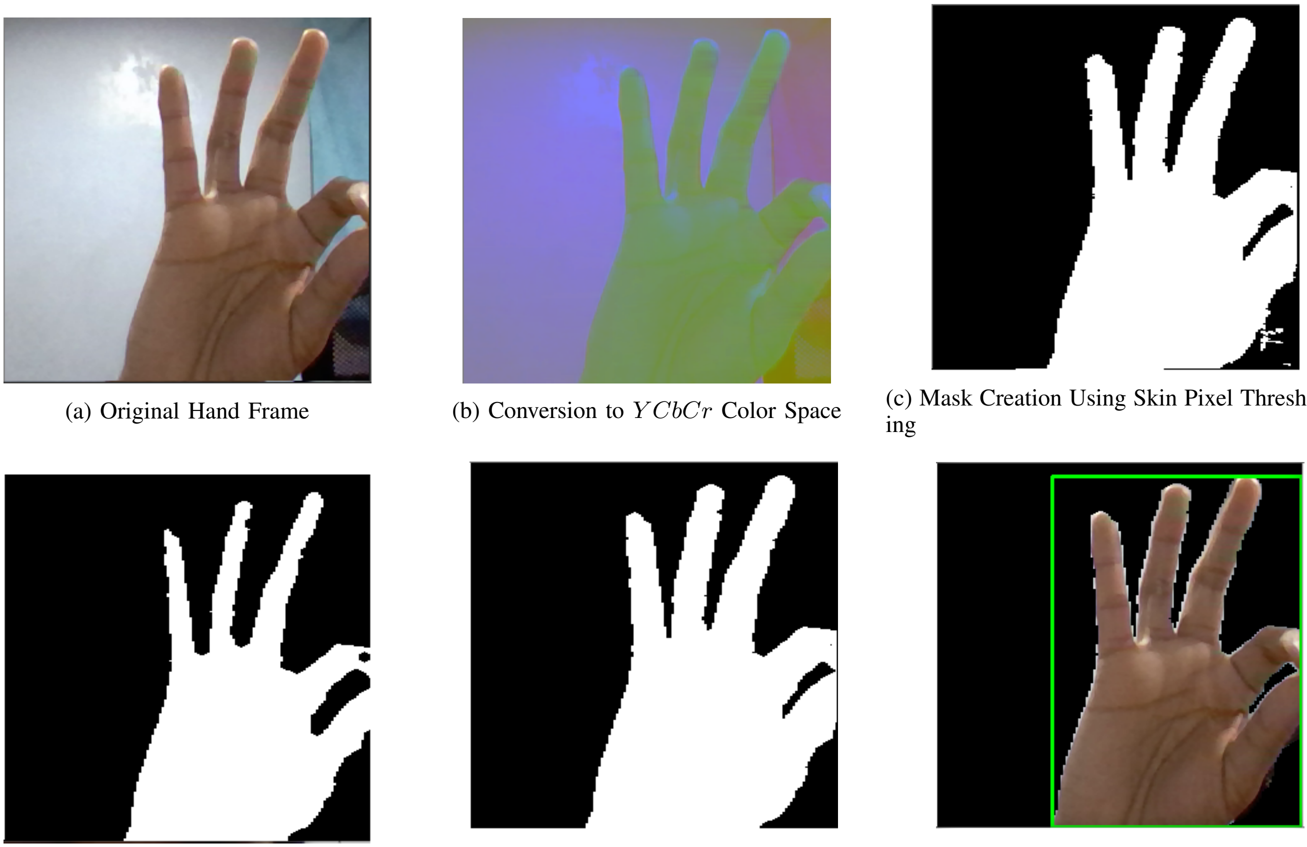
1) Skin Color Segmentation: Skin Segmentation is the process of locating the skin-like region of the image. To start it off, the frame is converted into YC_bC_r color space. From RGB color space , YC_bC_r can be retrieved by the following [10]:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ 37.797 & 74.203 & 112 \\ 112 & 93.786 & 18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

This color space separates chrominance information which are contained in the C_b and C_r channels and brightness information contained in the Y channel. Ignoring brightness information reduces the effect of uneven illumination [11]. It is suitable for skin color detection and is able to handle different types of skin color. A mask is created by thresholding pixel values similar to the ones used by Tang et al. [1] given by

$$77 \leq C_b \leq 127 \quad and \quad 133 \leq Cr \leq 173.$$

2) Morphological Filtering: The binary mask produced by simple thresholding can be distorted by noise and texture. Morphological image processing pursues the goal of removing these imperfections by accounting for the form and structure of the image [10]. A structuring element is a matrix consisting of only 0s and 1s that can have any desired shape and size. The pixels with values of 1 define the neighborhood. Only those portions of the image that fit the structuring element are passed by the filter. Smaller structures are blocked and excluded from the output image.



(a) Original Hand Frame

(b) Conversion to $YCbCr$ Color Space

(c) Mask Creation Using Skin Pixel Thresholding

(d) Two iterations of Erosion

(e) Two iterations of Dilation and Gaussian Blurring

(f) Resulting segmented skin region bounded by a green box.

Fig. 3: Preprocessing Pipeline

Considered to be the most basic morphological operations, dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The state or value of any pixel in the resulting image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. In erosion, the value of the output pixel is the minimum value of all the pixels in the inputs pixel neighborhood while in dilation it is the maximum.

For this stage, a mask is created using an elliptic structuring element inscribed in a rectangle with width and height equals to five pixels. It's used as kernel for two iterations of Erosion followed by two iterations of Dilation operation. The mask is then smoothened by Gaussian Blurring and applied to the original hand frame. These are implemented using OpenCV out-of-the-box functionalities.

3) Hand Region Extraction: The hand region is extracted after applying skin masking step. This leaves out most blank spaces and offers more robustness in classification. The biggest blob in the final skin mask described previously is extracted by finding the mask's contours. OpenCV's *findContours* method is used which is based on Suzuki's algorithm [13]. The contour with the largest size is determined to be the biggest object in the frame and hence is assumed to be the hand. This contour is then bounded by a box and is considered the hand region. Figure 3f shows a sample segmented hand region.

D. Feature Extraction

The resulting segmented hand region is turned into a grayscale image and resized to n by n pixels. This is to standardize the size of the feature vector that is going to be passed to the classifier. The Histogram of Oriented Gradients (HOG) feature is extracted from the resized gray scale image. Python Scikit's HOG extraction method is used [16]. A sample visualization of the HOG of a hand gesture can be found on Figure 4

Image global normalization is applied first to reduce the effect of changes in illumination. Gamma correction, a nonlinear operation used to code and decode luminance, is used either by computing the square root or the log of the channel. Gamma correction is governed by the following power-law expression.

$$V_{out} = AV_{in}^{\gamma}$$

When gamma value $\gamma < 1$, it is called Gamma compression. This procedure minimizes the effects of local shadowing and variance in illumination. The first order image gradients in x and y axes are then computed. They contain contour, silhouette and some texture information.

The image window of size n by n is divided into blocks which are spatial regions within the image. Blocks are composed of 3 x 3 cells and each cell is made up of 8 x 8 pixels. For each cell, a local 1-D histogram of gradient or

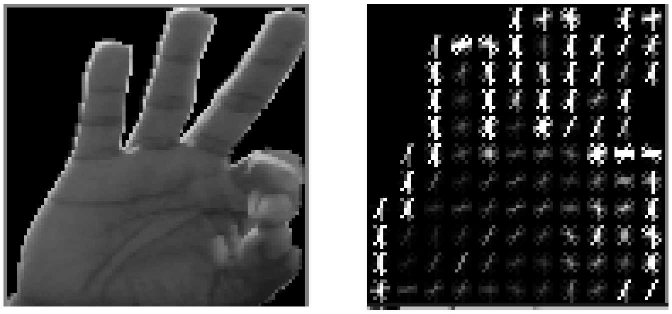


Fig. 4: Visualization of the HOG of a Hand Gesture in Grayscale Image

edge orientations over all the pixels in the cell are accumulated. The combined cell level histogram is the orientation histogram representation of the cell. The gradient of the cells are partitioned into 9 orientation bins and the gradient magnitude is used to vote on the orientation of the histogram. The gradient magnitude is given by

$$||\nabla f|| = \sqrt{\left(\frac{\delta f}{\delta x}\right)^2 + \left(\frac{\delta f}{\delta y}\right)^2}$$

and orientation is given by

$$\theta = \tan^{-1}\left(\frac{\delta f}{\delta y} \div \frac{\delta f}{\delta x}\right)$$

The gradient histograms generated are normalized across blocks. Normalization further offers robustness to shadowing, illumination and edge contrast. An energy measure is accumulated over the cells in the block. This value is then used to normalize each cell in the block. At this point, we now have the Histogram of Oriented Gradient (HOG) descriptors. The collection of HOG descriptors from all the blocks of a dense overlapping grid of blocks covering the window are accumulated and flattened into a feature vector.

E. *Classification*

Support Vector Machine (SVM) is used as classifier for this project. It is a linear classifier which aims on maximizing the distance of near miss examples called Support Vectors from decision hyperplanes. Its power to fit complex data is due to the kernel method where a kernel ϕ maps the input to a higher dimensional space. The SVM, in essence, is a binary classifier. To handle multiclass classification, a binary model with linear kernel is trained for each class and is fitted against all other classes. This is the one-vs-all strategy and is implemented using Scikit-learn's *OneVsRestClassifier* [15].

IV. Results

A. *Size of Segmented Hand Region*

The reason why the segmented hand region needs to be resized is that the larger the image is the longer the HOG feature vector becomes. Its original size of 295 by 295 pixels generates more than 96,000 feature values. This isn't economical in terms of space and training time for the SVM. Table I shows the growth of dimensionality of the feature vector and training time as the size of the segmented hand region frame increases. Size $n = 24$ to $n = 96$ are checked.

TABLE I: Descriptor Size and Training Time

Hand Region Frame Size (n x n)	Descriptor Size	Training Time (sec)
24	81	1
32	324	2
40	729	3
48	1296	5
56	2025	7
64	2916	11
72	3969	18
80	5184	23
88	6561	29
96	8100	36

Sizing down the image however lowers resolution so there's loss of detail. The appropriate n by n frame size is determined by checking the Accuracy for different sizes 5.

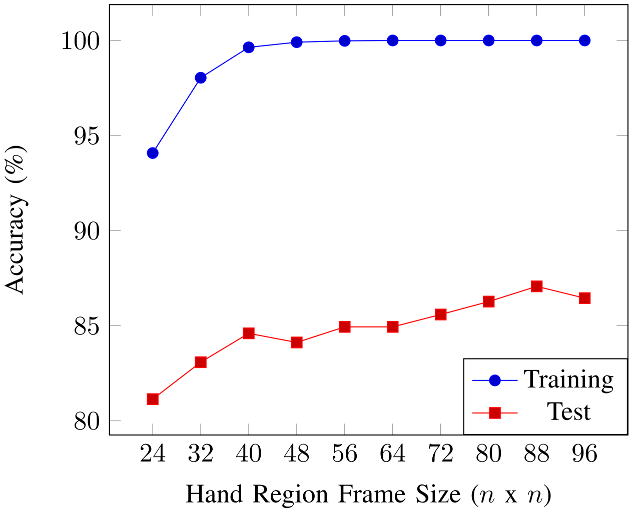


Fig. 5: Hand Region Frame Size vs Accuracy (%)

From smaller hand region frame of $n = 24$, accuracy is already up to 81.14%. This coincides with the human's ability to perform image recognition. At 24 x 24 resolution condition, human recognition can already achieve high recognition rate on processed images [7]. Starting at $n = 40$, all examples in the training set have been correctly classified. Testset Accuracy generally increases as the hand region frame size increases. Performance peaks at $n = 88$ with 87.07% so this is the size used for the system.

B. *Hand Gesture Class Performance*

Performance for each hand gesture class is depicted in Table III. Majority of the gestures achieved more than 90% accuracy. Gesture classes A, E, K and N scored below 90% with N having the most misclassifications.

TABLE II: Hand Gesture Classification Confusion Matrix

		Ground Truth									
		A	B	D	E	F	K	L	N	W	Y
Classification	A	377	0	0	41	0	0	3	70	0	3
	B	0	494	17	0	16	11	1	6	0	4
	D	6	6	446	5	1	29	17	18	1	9
	E	32	21	16	468	13	0	0	34	3	3
	F	0	1	0	0	558	1	0	2	4	5
	K	2	1	0	0	3	409	2	0	8	5
	L	0	6	13	0	2	30	448	0	0	0
	N	42	0	0	17	3	0	0	261	0	0
	W	0	1	0	0	0	62	2	25	436	0
	Y	0	4	0	10	0	9	0	0	0	452

TABLE III: Accuracy of the Hand Gesture Classes

Gesture	Accuracy (%)
A	82.14%
B	92.51%
D	90.65%
E	86.51%
F	93.62%
K	74.23%
L	94.71%
N	62.74%
W	96.46%
Y	93.97%
Overall	87.07%

The confusion matrix on Table II describes how the misclassifications went. Due to similarity in shape, misclassification for the gesture classes A, E and N are mostly predictions for the other two. Misses on the K class could be attributed to one test video sample where there's rotational movement seen.

V. Conclusion

This paper presents a successful implementation of a Hand Gesture Recognition system using the Histogram of Oriented Gradients (HOG) feature trained on a multi-class SVM. The system has achieved an overall classification performance of 87.07%. One downside seen to using HOG is that its feature vector size increases exponentially as the size of the image increases. Possible improvement on this work is to apply techniques to lower dimensionality of the feature vector such as PLS or PCA [5]. Nonetheless HOG is verified to be a good feature for a Hand Gesture Recognition System.

Another approach to Vision Based Gesture Recognition is Deep Learning. Convolutional Neural Network architectures have recently been winning various Computer Vision and object recognition tasks and are continually improved. With Deep Learning, the preprocessing and feature extraction stage will be eliminated as they are automatically learned. This would be a logical next step and experimental results here can be used as baseline performance.

Acknowledgment

I would like to thank Dr. Prospero Naval for allowing me to work on this topic. This is done as a Mini Project Requirement for CS 282 Computer Vision course.

References

[1] Tang, A., Lu, K., Wang, Y., Huang, J., Li, H.,*A Real-time Hand Posture Recognition System Using Deep Neural Networks*, ACM Transactions on Intelligent Systems and Technology, Vol. 9, No. 4, Article 39, 2013.

[2] Dalal, D., Triggs, B., *Histograms of oriented gradients for human detection*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2, 2005.

[3] K. p. Feng, F. Yuan, *Static hand gesture recognition based on HOG characters and support vector machines*, Instrumentation and Measurement, Sensor Network and Automation (IMSNA), 2013.

[4] Jiali Zhang, Guixi Liu, *Dynamic gesture recognition and human-computer interaction* , International Industrial Informatics and Computer Engineering Conference (IIICEC), 2015.

[5] Misra, A., Okatani, T., Deguchi, K, *Hand gesture recognition using histogram of oriented gradients and partial least squares regression* , MVA2011 IAPR Conference on Machine Vision Applications, 479482 , 2011.

[6] Hilton Bristow and Simon Lucey , *Why do linear SVMs trained on HOG features perform so well?* , arXiv:1406.2419v1 [cs.CV], 2014.

[7] Li, S., Hu, J., Chai, X. and Peng, Y, *Image Recognition With a Limited Number of Pixels for Visual Prostheses Design*, Artificial Organs, 36: 266274. doi: 10.1111/j.1525-1594.2011.01347.x, 2012.

[8] Freeman, W., Roth, M. *Orientation histograms for hand gesture recognition*, Mitsubishi Research Laboratory Report, 1994.

[9] A. Mohan, C. Papageorgiou, T. Poggio. *Example-based object detection in images by components.* PAMI, 23(4):349 361, 2001.

[10] Premaratne, P. *Human Computer Interaction Using Hand Gestures*, Springer Science+Business Media Singapore, 2014.

[11] Sebastian, P., Voon, Y., Comley, R., *Colour Space Effect on Tracking in Video Surveillance*, International Journal on Electrical Engineering and Informatics - Volume 2, Number 4, 2010.

[12] Douglas Chai and King N Ngan. *Face segmentation using skin-color map in videophone applications*, Circuits and Systems for Video Technology, IEEE Transactions on 9, 4 (1999), 551564.

[13] Suzuki, S. and Abe, K., *Topological Structural Analysis of Digitized Binary Images by Border Following*, CVGIP 30 1, pp 32-46, 1985.

[14] *American Sign Language*, http://www.linguistics.uconn.edu/asl/, Accessed: 2016 May 4.

[15] *Scikit Learn Support Vector Machines*, http://scikit-learn.org/stable/modules/svm.html 2016. Last accessed 2016 May 4.

[16] *Histogram of Oriented Gradients*, http://scikit-image.org/docs/dev/auto_examples/plot_hog.html 2016. Last accessed 2016 May 4.