# Automatic Recognition of Turkish Fingerspelling

Furkan Işıkdoğan and Songül Albayrak
Yıldız Technical University, Computer Engineering Department, Yıldız, Istanbul, Turkey
furkan@isikdogan.com, songul@ce.yildiz.edu.tr

*Abstract*—**Fingerspelling is a manual representation of alphabet letters and used in sign language to spell out words, especially private names. In Turkish Sign Language it is a challenging task due to the ambiguity of the fingerspelling representations of the letters with diacritic marks and complex hand configurations. In this paper we propose a Turkish fingerspelling recognition system based on extraction of the most effective features that help disambiguation of the signs. Our approach is fundamentally based on feature extraction by Histograms of Oriented Gradients (HOG) and dimension reduction by Principal Component Analysis (PCA). Use of internal features instead of outer projection of the image enhances the performance on disambiguation. The test dataset consists of 493 fingerspelling images created by 4 different signers and the test results indicate an average success rate of classification of 99.39%.**

***Turkish fingerspelling recognition; Histograms of Oriented Gradients - HOG; Human Computer Interaction - HCI***

## I. Introduction

A sign Language is a visual language that expressed with gestures and mainly used by deaf people. Fingerspelling can be considered as a subset of the sign language which enables spelling words for private names or clarification of the words by letter-by-letter signing.

Fingerspelling in Turkish Sign Language (TSL) consists of one or two handed signs that representing the 29 letters in the Turkish alphabet. Previous approaches on TSL are mainly based on extraction features from the outer projection of the hand shape. Altun *et al.*[2], developed a TSL fingerspelling recognition system with a letter recognition accuracy of 99.43%. Haberdar and Albayrak[3], developed a word level sign recognition for TSL based on Hidden Markov Model (HMM) framework with an overall success rate of 93.31%.

In this work we developed a fingerspelling recognition system for TSL based on internal feature extraction and enhanced with feature transformation. The image dataset is created by capturing frames from the video database created by Altun *et al*. [2] for fingerspelling recognition. In section 2 we describe the methods that used for selection of key frames and segmentation and preprocessing of the captured frames. Section 3 covers the extraction of gradient based internal features and dimension reduction by feature transformation. Classification of test instances and experimental results are presented in Section 4, and finally, conclusions and future work are discussed in Section 5.

## II. Segmentation and Preprocessing

### A. Key Frame Selection

The frames that most likely to be stable are selected as the key frames in video sequences. We used frame differencing method to detect the temporal changes between successive frames and determine the frame with minimum change as the key frame in a video sequence. Frame differences can be computed by differencing successive frames as in (1) where *D* denotes the frame difference and *I* denotes the pixel intensity matrix of a frame. The key frame selection process is illustrated in "Fig. 1" on the frames of a sample video sequence.

$$D_k(x,y) = |I_{k-1}(x,y) - I_k(x,y)| \quad (1)$$

### B. Segmentation of Hands

Segmentation of hands is one of the essential steps in vision-based fingerspelling recognition. In this work we preferred $YC_bC_r$ color-space due to its convenience for skin color detection. We convert the bitmap frames from RGB color-space to $YC_bC_r$ color-space by using (2) for each pixel of the image.

$$Y = 0.299R + 0.587G + 0.114B,\ C_b = R - Y,\ C_r = B - Y \quad (2)$$

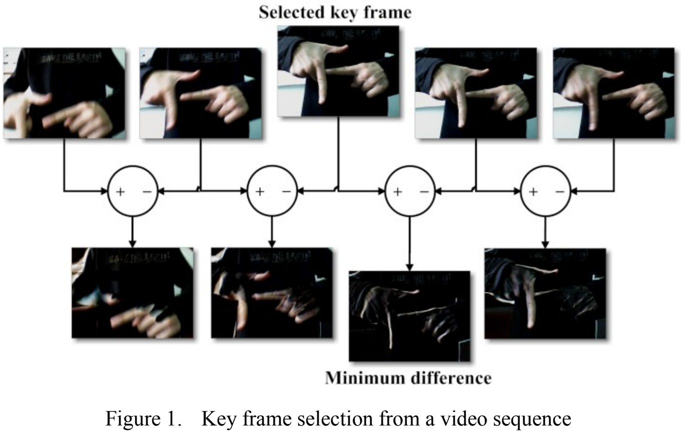


Figure 1. Key frame selection from a video sequence

Working on $YC_bC_r$ color-space, Chai and Ngan [4] report that the most representative color component ranges of human skin color as (3). A skin color segmentation mask is obtained by thresholding the color components of the image within the ranges in (3) as a binary matrix in order to enable the use of binary morphological operations.

$$C_r = [133\ 173],\ C_b = [77\ 127],\ \text{where } Y, C_b, C_r = [0\ 255] \quad (3)$$

In this work we used connected component labeling method to determine the hands as the main components. A threshold value is determined for the area percentage of the connected components and the components with an area below the threshold value are eliminated as illustrated in "Fig. 2". Morphological opening can also be applied for eliminating small skin colored regions, but our results of several trials indicated that the morphological opening with a small sized structuring element might not be sufficient to remove medium sized skin colored regions such as textures of clothes and increase in the size of the structuring element may cause deformations on the shape of the hands.

In order to use internal feature descriptors, a gray level hand image obtained by applying the binary segmentation mask to the luminance component of the image. The resultant image is smoothed with a Gaussian filter in order to reduce the artifacts within the hand segment. Finally the black area on the image is cropped to provide scale invariance for the feature descriptor as shown in "Fig. 3".

## III. Feature Extraction

This section describes the methods applied in this work for gray level feature extraction and dimension reduction by feature transformation.

### *A. Histograms of Oriented Gradients*

In this work we used Histograms of Oriented Gradients (HOG) [1] as the descriptor of hand shapes. This descriptor has previously applied for human detection [1], head and shoulder detection [5] and human posture estimation [6]. The rough shape of an object can usually be characterized by HOG.

Local edge directions and intensities used as the main features while extracting the HOG feature. The implementation of HOG can be summarized as; dividing the image to a fixed sized of spatial cells, computing gradients of the image, obtaining angles and magnitudes of the edges, accumulating weighted votes for gradient orientation over each cell and normalization of contrast over overlapping blocks which are larger spatial regions that consist of cells.

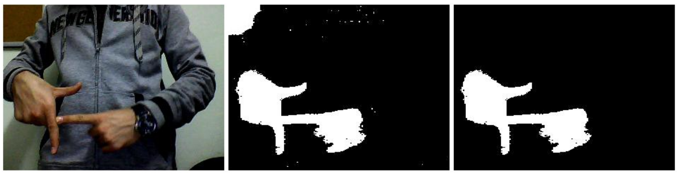


Figure 2. (a) Original image, (b) binary segmentation mask after skin color thresholding, (c) result of the connected component elimination.

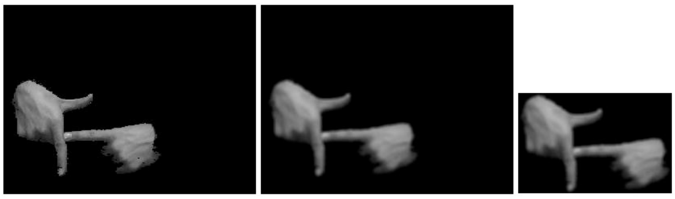


Figure 3. (a) Gray level image after segmentation, (b) gaussian filter applied image, (c) extracted hand area on the image.

Gradient computation can be implemented by applying the 1-D discrete derivative masks as in (4) which are presented as the default gradient computation kernel filters for the HOG descriptor by Dalal and Triggs[1].

$$[-1, 0, 1] \text{ and } [-1, 0, 1]^T \quad (4)$$

The magnitude and orientation for each gray level pixel within the image can be computed by using the corresponding values of the horizontal and vertical components of the image gradient. The histograms of oriented gradients are computed by accumulating the weighted votes for each cell where each pixel within a cell votes for the orientations with a weight of its magnitude. The orientations are represented by a predefined number of angle intervals. Our experiments on test dataset indicated that the use of 6 unsigned orientation bins spaced over 0° - 180° is highly efficient for TSL fingerspelling.

In order to achieve a better invariance to illumination and contrast it is recommended [1] to locally normalize the values of histograms within blocks. Let $v$ and $v'$ denote the unnormalized and normalized descriptor vectors respectively; $||v||$ denotes the L2-norm of the vector and ε is a small constant to prevent division by zero error. Normalized values of histograms can be computed as follows.

$$v' = v \,/\, (||v|| + \varepsilon) \quad (5)$$

In this work blocks are defined as unions of 2×2 cells where the image is divided into an 8×8 grid of cells as illustrated in "Fig. 4". The image is divided into a fixed number of cells with variable widths and heights in terms of pixels to accomplish resolution independency. In order to form the final descriptor, block normalization is applied to each cell in the grid of overlapping blocks which are illustrated in "Fig. 5". Use of block normalization has increased the overall performance of our system by reducing the effect of partial shadows and illumination differences on the hand images.

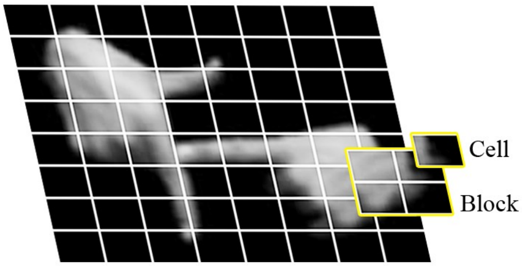


Figure 4. Spatial cells and blocks on a sample image

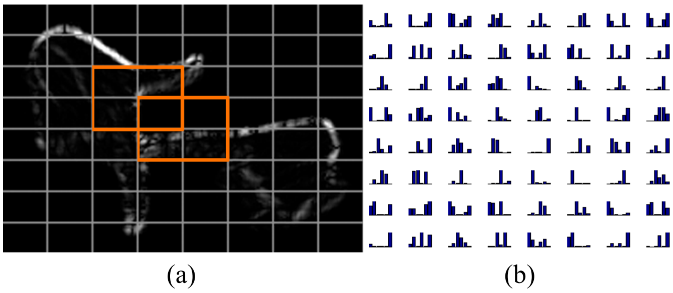


Figure 5. (a) Overlapping blocks on a gradient image, (b) normalized histograms of oriented gradients for a sample image

### B. Dimension Reduction

In this work an image is represented by HOG feature that consists of 6 histogram values for an 8×8 grid which means use of 384 features for describing an image. Redundant features, which usually exist on the blocks in the background, can be eliminated by Principal Component Analysis (PCA).

The image dataset is represented by a feature matrix that consists of feature vectors of each image on corresponding rows. Eigenvalues and eigenvectors are obtained from the covariance matrix of the feature vectors and the features are transformed into a lower-dimensional matrix by the elimination of less significant components. Our experimental results on the fingerspelling dataset indicated that features can be reduced up to 33% by PCA with no loss in the overall classification performance. Hence, we used 257 features per instance instead of 384.

## IV. EXPERIMENTAL RESULTS

### A. Image Dataset

The image dataset is created by capturing frames from the video database created by Altun *et al*. [2]. Our dataset, which is created by using four different signers, consists of 493 images for the 29 static fingerspelling representations of the letters in the Turkish alphabet. A set of fingerspelling images in our dataset, which is representing all 29 letters in Turkish Alphabet, is shown in “Fig. 7”.

### B. Training and Test

The dataset is divided into the training and test datasets with two different methods in different experiments. Table1 summarizes the distribution of training and test datasets.

In the first experiment the 10 images from three different signers are used for training and 6 different images from the same three signers are used for test for each 29 letter representations, which sums up to 290 images for training and 174 images for test. In the second experiment, it is aimed to measure the signer independency of the system. Hence, the training and test sets are selected from completely different signers. Training dataset is created by using 348 images from two different signers, and the remaining 145 images from other three signers are used for test.

TABLE I. DISTRIBUTION OF DATASET FOR TRAINING AND TEST

| **Signer Number** | **Number of Images for each 29 letter** | | | |
|---|---|---|---|---|
| | ***Experiment 1 (Signer Dependent)*** | | ***Experiment 2 (Signer Independent)*** | |
| | ***Training Data*** | ***Test Data*** | ***Training Data*** | ***Test Data*** |
| 1 | 4 | 2 | 6 | - |
| 2 | 4 | 2 | 6 | - |
| 3 | 2 | 2 | - | 4 |
| 4 | - | - | - | 1 |
| **All** | **10** | **6** | **12** | **5** |

Finally, in the third experiment leave-one-out cross-validation is applied for calculating the overall classification performance. At each step, one different image is used for test and the remaining 492 images are used for training.

We used the 3-nearest neighbor classification with Euclidean distance to train and classify the fingerspelling representations of the letters. The signer dependent experiment has resulted with 173 true and one false classification. The signer independent experiment which is more challenging has resulted with 141 true and 4 false classifications, and the leave-one-out cross validation has resulted with 490 true and 3 false classifications. Test results of all three experiments are reported in Table2 and illustrated in “Fig. 6”.

Letter recognition errors usually occur when the characters are very similar to each other in shape and orientation. In the third experiment, the most frequently confused characters were ‘S’ and ‘Ş’ and ‘Y’ and ‘K’ respectively as shown in “Fig. 8”. Confusion matrix of most frequently confused characters is presented in Table3.

TABLE II. CLASSIFICATION PERFORMANCE

| **Experiment Number** | **Number of Images** | | **Classification** | |
|---|---|---|---|---|
| | ***Training Data*** | ***Test Data*** | ***#True*** | ***#False*** |
| Experiment1 (Signer dependent) | 290 | 174 | 173 | 1 |
| Experiment2 (Signer independent) | 348 | 145 | 141 | 4 |
| Experiment3 (Leave-one-out cross validation) | 493 | 493 | 490 | 3 |

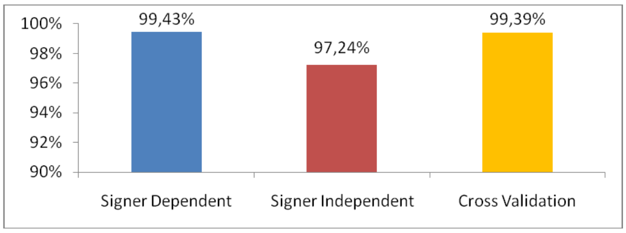


Figure 6. Success rates of classification

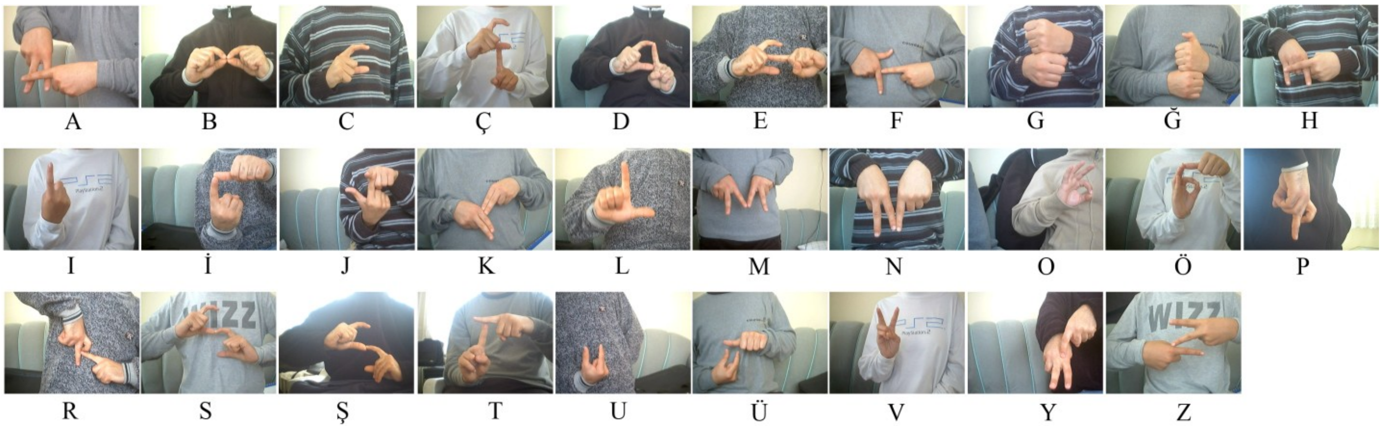
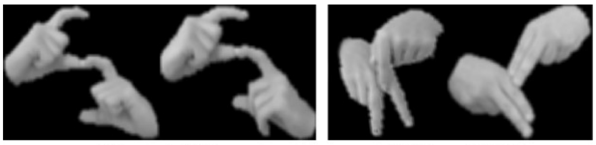


Figure 7. A set of fingerspelling images for all 29 letters in Turkish Alphabet

TABLE III. CONFUSION MATRIX OF CHARACTERS

| | | **Actual** | | | |
|---|---|---|---|---|---|
| | | ***K*** | ***S*** | ***Ş*** | ***Y*** |
| **Predicted** | ***K*** | 17 | 0 | 0 | 1 |
| | ***S*** | 0 | 15 | 0 | 0 |
| | ***Ş*** | 0 | 2 | 17 | 0 |
| | ***Y*** | 0 | 0 | 0 | 16 |



'S' and 'Ş'

'Y' and 'K'

Figure 8. Most frequently confused characters

## V. CONCLUSIONS AND FUTURE WORK

In this work we have proposed an inner feature extraction based fingerspelling recognition system by using HOG for Turkish Sign Language. Signer independent test results indicate a success rate of classification over 97% and signer-dependent tests and cross validation tests indicate a success rate over 99%. Use of internal features has provided a significant success rate even on the hand postures with very similar shapes and orientations. Feature extraction on gray level segmented images provides a recognition system that robust to occlusions of the fingers and hands.

One of the main limitations of our work is that our dataset consists of captured frames of videos and the feature extraction and classification are performed by processing static images. Hence, it is not possible to detect the trajectories of the gestures. In TSL, representation of some characters involves some movements. Different captures of same fingerspelled characters can cause a decrease in the classification performance. In future work, a multilevel structure with finger movement tracing and static frame processing can be implemented to handle the movements of fingers.

Another limitation of our work is color based segmentation of hand in corresponding frame. Our recognition system is robust to small skin colored areas in the frames such as textures of clothes or partial faces which can be eliminated as mentioned in Section 2. However, the segmentation may fail when the face appears in the frame. Face detection or spatial ordering can be performed on frames to distinguish face from hand. Segmentation of hand on skin colored backgrounds including skin colored clothes is still challenging after all. In future work, another approach to enhance the performance can be implementing a fingerspelling recognition system with a word level spelling correction by using a dictionary.

## REFERENCES

[1] N. Dalal, and B. Triggs, "Histograms of oriented gradients for human detection," IEEE Computer Vision and Pattern Recognition, pp. 886-893, 2005.

[2] O. Altun, S. Albayrak, A. Ekinci, and B. Bükün, "Turkish fingerspelling recognition system using axis of least inertia based fast alignment," The 19th Australian Joint Conference on Artificial Intelligence, 2006.

[3] H. Haberdar, and S. Albayrak, "Real time isolated Turkish Sign Language recognition from video using hidden Markov models with global features," Lecture Notes in Computer Science, vol. 3733, pp. 677-687, September 2005.

[4] D. Chai, and K. N. Ngan, "Face segmentation using skin-color map in videophone applications," IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 4, June 1999.

[5] C. Zeng, and H. Ma, "Robust head-shoulder detection by PCA based multilevel HOG-LBP for people counting," 20th International Conference on Pattern Recognition, 2010.

[6] K. Onishi, T. Takiguchi, and Y. Ariki, "3D human posture estimation using the HOG features from monocular image," 19th International Conference on Pattern Recognition, 2008