Proc. of the Fifth Intl. Conf. Advances in Computing, Communication and Information Technology- CCIT 2017

ISBN: 978-1-63248-131-3 doi: 10.15224/ 978-1-63248-131-3-17

Translating Arabic Sign Language (ARSL) To Text Using Artificial Neural Networks

Lina E. A. Elsiddig Mayada Mohamed Ismail Mohamed

***Abstract*— A communication gap exists between the hearing and hearing-impaired communities due to a lack of familiarity with the means of communication of each. This research attempts to bridge this distance by creating Arabic Sign Language (ArSL) datasets, which there is a lack of, image processing, selecting a feature extraction method and designing a machine learning classification system capable of translating Arabic Sign Language (ArSL) to text. The system was implemented on MATLAB 2014a using an Artificial Neural Network that was trained on the morphological features of 100 samples to classify input images into 3 alphabet classes that achieved an accuracy of 73.3%.**

***Keywords*—Arabic Sign Language, image processsing, features, artificial neural network**

I. Introduction

According to World Health Organisation (WHO) statistics, 360 million people have disabling hearing loss [1], 4.436 million of which are in the Arabic speaking world [2]. The fundamental communication method used by individuals who suffer from disabling hearing loss is sign language, which is a visual language that employs hand shapes and positions, boy movements and facial expressions [3]. The Sign language of the adult deaf community of a specific country is recognized as the first language of a deaf individual [4], making it Arabic Sign Language (ArSL) in the Arabic speaking countries. Individual countries' deaf communities have established their own sign language; as a result there are as many sign languages as Arabic-speaking countries, however, a documented standard developed so far is the 30 ArSL alphabets, that is, finger spelling. Finger spelling comprises of static manual gestures spelled by motion-less hand shapes [5].

This research aims to create ArSL alphabet datasets, since there is a lack of documented data sets hence contributing to the lack of ArSL translators. It also aims to select a feature extraction method and a machine learning classifier that achieves a classification accuracy of 70% and ultimately design and implement a system that translates finger-spelled ArSL alphabets to text in a simulation environment based on these findings.

II. Related Work

Copious amounts of researches have been done in the field of gesture recognition using computer vision; some relevant researches are summarized below.

In 2005, [6] used a gloved colour hand appearance-based model to implement a signer variant ArSL recognition system that recognised 50 signs performed by one person having 10 samples per sign. In 2010, [3] extracted distance morphological features from the wrist to the end of the palm based on wrist orientation performing the classification using K-Nearest Neighbour algorithm for their ArSL translator. Their proposed system translates 30 Arabic alphabets with an accuracy of 91.3%. In a different research to classify electronic components in 2009 [7], morphological features; including circularity, solidity, convexity, concavity and aspect ratio among others were utilized to describe the object. A correlation matrix was arranged for every possible pair of features in order to eliminate highly correlating and Support Vector Machines (SVM) were then used to rank the features based on discriminating ability. In 2011, [8] employed a 3-layer artificial neural network trained on the Fourier descriptors of a processed image to translate the finger spelling component of English sign language in real-time with an accuracy of 80%. In 2014, [9] used a similar classifier and image processing approach to [8] however the artificial neural network was trained on an array of masks derived from the processed image. In 2015, [10] used a Microsoft Kinect Sensor as opposed to a webcam in order to provide depth and skeletal information in the design of their Indian Sign Language translator. The dataset comprised of skeletal positions and the sign meaning, the classification was then done using a parse tree. In 2013, [11] created a Video-based signer-independent color glove model similar to [6] for ArSL word recognition system using Hidden Markov Models. Discrete cosine transform was used to extract features from the input gestures by representing the image as a sum of sinusoids of varying magnitudes and frequencies. In 2010, [12] proposed a 3D imagery ArSL translator using PCNN for recognition of 30 ArSL alphabets with a 93% recognition accuracy.

In this paper, we have taken the computer vision approach without the use of any glove or additional hardware.

Lina Elsiddig Abdelrahim Elsiddig
University of Medical Sciences and Technology (UMST)
Sudan

Mayada Mohamed Ismail Mohamed
University of Medical Sciences and Technology (UMST)
Sudan

III. Methodology and Results

Sign language recognition entails multiple disciplines including analysis of the sign language, image processing,

Proc. of the Fifth Intl. Conf. Advances in Computing, Communication and Information Technology- CCIT 2017

ISBN: 978-1-63248-131-3 doi: 10.15224/ 978-1-63248-131-3-17

feature extraction and classifier design. In this research, ArSL alphabet datasets were created, on a black background and on a white background, a comparison of image processing techniques was made on the different datasets, a comparison of the performance of two feature extraction methods namely morphological features and point features when paired with various machine learning classifiers was made. The best performing classifier was then trained on a number of samples of a varying number of characters in order to achieve the highest possible accuracy with the available dataset size. These experiments were done on MATLAB software. Fig. 1 shows the methodology flow chart.

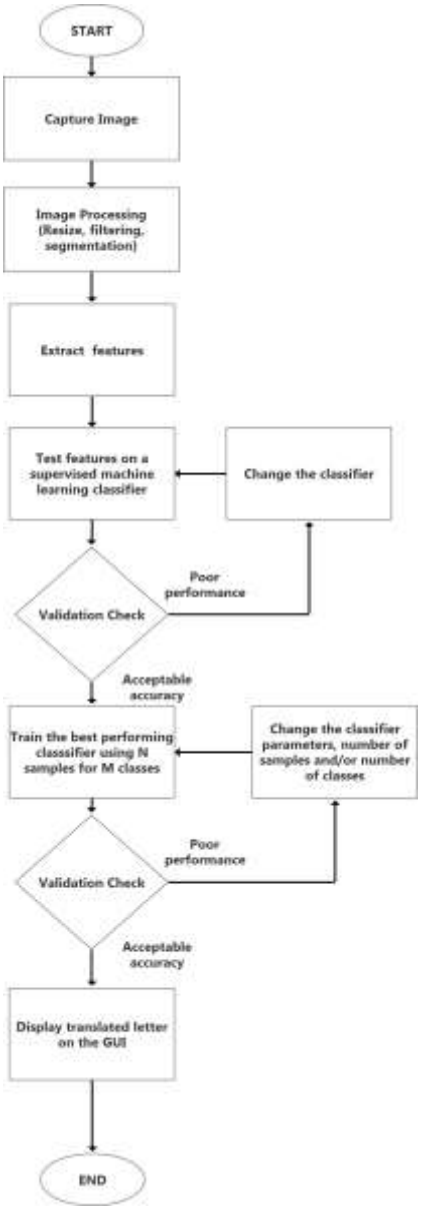


Figure 1. Flowchart describing the methodology followed by this paper

A. *Dataset creation and Image Processing*

The lack of documented ArSL alphabet datasets led us to create our own datasets by placing the subject at 120cm from the lens at the height of their hand and captured all the alphabets. The data was verified by the observation of ArSL speakers. Fig. 2 shows the image capture scenario. A sample of the dataset that was created is shown in Fig. 3.

In order to prepare the image for recognition, it underwent cropping and resizing in order to decrease the run time of the coming processes, low pass filtering using a Gaussian filter to remove noise and fine edges, and segmentation to isolate the object from the background. The best performing technique and background color that was adopted throughout this paper was Simple Thresholding and Boundary Extraction when the signer is on a dark background. The stages are shown respectively in Fig. 4.

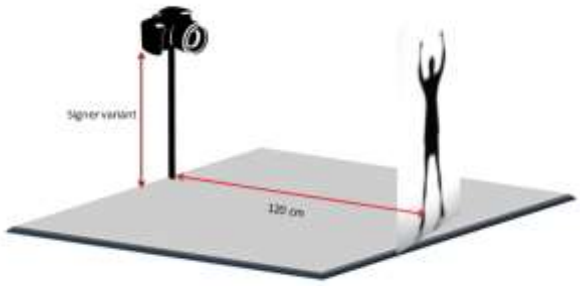


Figure 2. Image capture scenario

Figure 3. The created dataset of 30 ArSL alphabets

Figure 4. From right to left: Cropped and resized image, filtered image, segmented image (simple threshold), extracted boundary

B. *Feature Extraction and Classifier Selection*

Features are vectors used to express the image in order to reduce the dimensionality of data in the interest of avoiding the curse of dimensionality [13]. In this paper, since the image is in binary form after processing, Morphological Features were extracted by the means of equations and

Proc. of the Fifth Intl. Conf. Advances in Computing, Communication and Information Technology- CCIT 2017

ISBN: 978-1-63248-131-3 doi: 10.15224/ 978-1-63248-131-3-17

SURF (Speeded-up Robust Features) Points were extracted using MATLAB functions. The morphological features calculated were orientation, thinness, irregularity, horizontal and vertical projections and aspect ratio.

Supervised machine learning classifiers are algorithms used in applications, such as computer vision, where heuristics perform poorly. They predict the class of input data based on training examples [14]. In this paper, an experiment was conducted where Support Vector Machines (SVM), classification tree, naïve Bayes and artificial neural networks were each trained on 10 samples of morphological features and SURF points separately to classify into 2 classes and the experiment determined the classifier and features used for the rest of the paper. The classifiers were each tested with 10 test set samples, 5 for each class. Table 1 shows the classification accuracies. Fig. 5 shows a graphical representation of the result of this experiment. (Note that in Fig. 4 CT represents Classification Tree, NB represents Naïve Bayes and NN represents Neural Network)

TABLE I. TABLE SHOWING THE PERFORMANCE OF EACH CLASSIFIER

Criteria	Features	
	SURF point	***Morphological***
Feature vector size of 1 sample	1 x 640	1 x 6
SVM classification	80% accuracy	70% accuracy
Classification Tree Result	70% accuracy	60% accuracy
Naïve Bayes classification	N.A.[a]	50% accuracy
Neural Network classification	80% accuracy	80% accuracy

a. Naïve Bayes classifier did not work using SURF points due to the presence of negative valued features

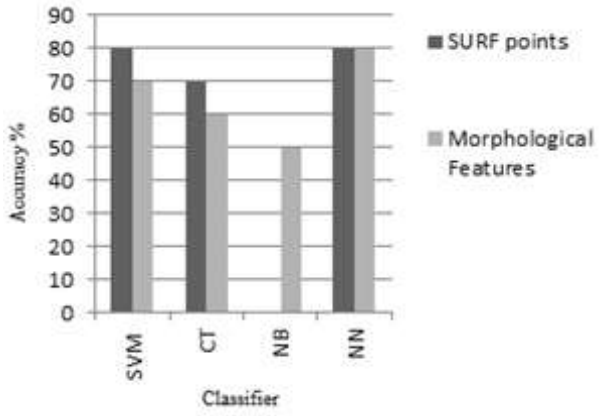


Figure 5. Bar chart representation of the classification accuracies

It can be deduced that the Neural Network and SVM performed best using SURF points and the Neural Network performed best using morphological features. It was also observed that the size of the SURF point vector created a risk of falling into the curse of dimensionality; consequently, morphological features were adopted.

C. *Design and Training of the Artificial Neural Network*

The feature vectors of each of the samples of each class are combined into one matrix that forms the neural network's training set. Each feature vectors corresponds to 1 column. The target matrix consisted of a pattern of 0's and 1's corresponding to each class. The created networks were trained on 20, 50 and 100 samples to classify into 3 and 5 classes. The structures of these networks vary however and the best performing network performances for each combination is shown in the TABLE II. The network structure in TABLE II shows the number of neurons in the network's hidden layers. Fig. 6 shows a graph of the classification accuracies in TABLE II.

It can be seen from the test set column in TABLE II that as we increase the number of samples, the classification accuracy increases for a constant number of classes and the classification accuracy decreases as the number of classes is increased for a constant number of samples. This proves that more samples are required to classify into more classes.

It has been observed that there is a non-uniform relation between the numbers of samples and classes with the architecture besides that the more the samples for a certain class, the more the neurons per hidden layer. The best performing number of hidden layers for this application was found to be 3 layers.

TABLE II. THE PERFORMANCE OF THE NETWORKS

Samples	*Classes*	*Network structure*	Classification accuracy	
			Training set	***Test set***
20 samples	3 classes	[6-40-40]	88.3%	60.0%
20 samples	5 classes	[6-65-65]	74.0%	20.0%
50 samples	3 classes	[6-50-50]	68.7%	60.0%
50 samples	5 classes	[6-100-100]	65.6%	32.0%
100 samples	3 classes	[6-25-25]	71.0%	73.3%
100 samples	5 classes	[6-65-65]	62.0%	59.8%

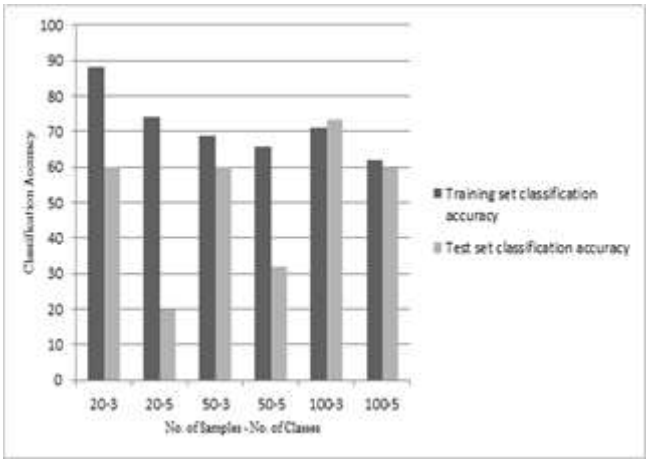
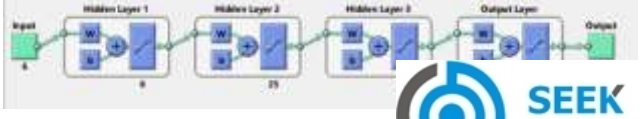


Figure 6. Bar chart showing the networks' classification accuracies

The highest test set accuracy was achieved by using 100 samples to classify into 3 classes. This model also shows decent generalisation since the test set accuracy was greater than the training set, i.e. the network classifies general data from outside its familiar data accurately. This network model was adopted for the remainder of the project. Fig.8 shows the performance plot obtained when training network [6-25-25] and Fig. 7 shows it's the structure.



Proc. of the Fifth Intl. Conf. Advances in Computing, Communication and Information Technology- CCIT 2017

ISBN: 978-1-63248-131-3 doi: 10.15224/ 978-1-63248-131-3-17

Figure 7. Structure of network [6-25-25]

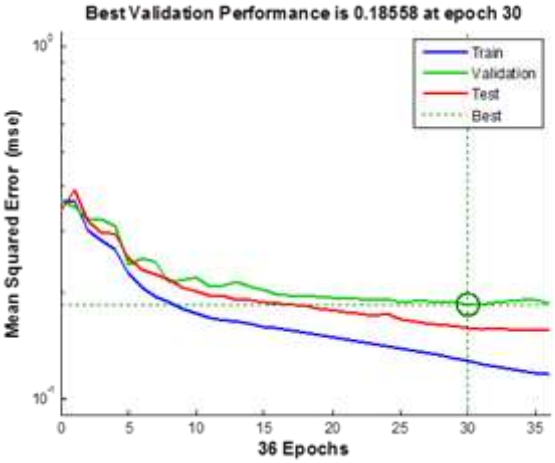


Figure 8. Network [6-25-25]'s performance plot

D. *GUI creation*

The graphical user interface was designed using the "Guide" tool in MATLAB. Since displaying Arabic characters is challenging using MATLAB, image forms of the letters were used and placed on an axis as if it were a figure. Fig. 9 shows the GUI after translating the letter "Alef".

Figure 9. ArSL translator GUI

IV. Conclusion and Recommendation

Arabic Sign Language alphabet data sets were gathered from a variety of signers all finger-spelling using their right hand on a darker and uniform background. The best performing features for this application were found to be binary morphological features. The best performing classifier for this application was found to be an Artificial Neural Network when created particularly with 4 layers. An artificial neural network was designed to classify into 3 classes, i.e. recognize 3 alphabets, with an accuracy of 73.3% offline. This was achieved when the model is trained on 100 samples, thus leaving us with the conclusion that the number of samples required for classification increases with the number of classes and as the number of samples increases for a constant number of classes, the test set classification accuracy increases until over fitting occurs. Our recommendations consist of increasing the translation capacity by adding more training samples thus enabling the classification into all 30 classes and implementing a real time translation by coding a c program to segment the capturing times and queue the images.

Acknowledgment

We deeply express our gratitude to all those who have supported this research and its authors. To our families, friends and university academics, much thanks.

This research was supervised by Dr Sami Abbas Alnagar and Ali.M.A.Ibrahim

References

[1] *WHO. Available: http://www.who.int/mediacentre/factsheets/fs300/en/*

[2] RightDiagnosis.Available: http://www.rightdiagnosis.com/d/deafness/stats-country.htm

[3] N. El-Bendary, H. M. Zawbaa, M. S. Daoud, A. E. Hassanien, and K. Nakamatsu, "ArSLAT: Arabic sign language alphabets translator," in Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on, 2010, pp. 590-595.

[4] W. F. o. t. Deaf. Available: https://wfdeaf.org/human-rights/crpd/sign-language/

[5] M. A. Abdel-Fattah, "Arabic sign language: a perspective," Journal of Deaf Studies and Deaf Education, vol. 10, pp. 212-221, 2005.

[6] M. Mohandes, S. Quadri, and M. Deriche, "Arabic sign language recognition an image-based approach," in Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on, 2007, pp. 272-276.

[7] D. Lefkaditis and G. Tsirigotis, "Morphological feature selection and neural classification for electronic components," Journal of Engineering Science and Technology Review, vol. 2, pp. 151-156, 2009

[8] C. Lungociu, "REAL TIME SIGN LANGUAGE RECOGNITION USING ARTIFICIAL NEURAL NETWORKS," Studia Universitatis Babes-Bolyai, Informatica, vol. 56, 2011.

[9] S. Đogić and G. Karli, "Sign Language Recognition using Neural Networks."

[10] A. S. Ghotkar and G. K. Kharate, "Dynamic hand gesture recognition and novel sentence interpretation algorithm for indian sign language using microsoft kinect sensor," Journal of Pattern Recognition Research, vol. 1, pp. 24-38, 2015.

[11] M. Al-Rousan, K. Assaleh, and A. Tala'a, "Video-based signer-independent Arabic sign language recognition using hidden Markov models," Applied Soft Computing, vol. 9, pp. 990-999, 2009.

[12] M. F. Tolba, A. Samir, and M. Aboul-Ela, "Arabic sign language continuous sentences recognition using PCNN and graph matching," Neural Computing and Applications, vol. 23, pp. 999-1010, 2013.

[13] A. K. Jain and B. Chandrasekaran, "39 Dimensionality and sample size considerations in pattern recognition practice," Handbook of statistics, vol. 2, pp. 835-855, 1982.

[14] J. Von Neumann and R. Kurzweil, The computer and the brain: Yale University Press, 2012.

ISBN: 978-1-63248-131-3 doi: 10.15224/ 978-1-63248-131-3-17

About Authors:

"..as we increase the number of samples, the classification accuracy increases for a constant number of classes."

"..as we increase the number of samples, the classification accuracy increases for a constant number of classes."