Feature based Algorithmic Analysis on American Sign Language Dataset

Umair Muneer Butt[1], Basharat Husnain[2], Usman Ahmed[3], Arslan Tariq[4],
Iqra Tariq[5], Muhammad Aadil Butt[6], Dr. Muhammad Sultan Zia[7]
Department of Computer Science
University of Lahore
Chenab Campus, Pakistan

Abstract—**Physical disability is one of the factor in human beings, which cannot be ignored. A person who can't listen by nature is called deaf person. For the representation of their knowledge, a special language is adopted called 'Sign-Language'. American Sign Language (ASL) is one of the most popular sign language that is used for learning process in deaf persons. For the representation of their knowledge by deaf persons, a special language is adopted 'Sign-Language'. American Sign Language contains a set of digital images of hands in different shapes or hand gestures. In this paper, we present feature based algorithmic analysis to prepare a significant model for recognition of hand gestures of American Sign Language. To make a machine intelligent, this model can be used to learn efficiently. For effective machine learning, we generate a list of useful features from digital images of hand gestures. For feature extraction, we use Matlab 2018a. For training and testing, we use weka-3-9-3 and Rapid Miner 9 1.0. Both application tools are used to build an effective data modeling. Rapid Miner outperforms with 99.9% accuracy in auto model.**

Keywords*—*Hand gesture recognition; pre-processing; weka; rapid miner; HOG; LBP; auto model

I. Introduction

Sign language provides a big aid and convenience in human life [1] and used especially by deaf persons and by other people to add weight in conversation. Visual representation by hands, delivers a meaningful message to others [2]. Sign Language consists in three forms: one is called facial expression, second is hand gestures and third is called body postures [1], [2]. In our daily life, we mostly use our body postures and facial expression to deliver meaningful information to others. The goal of the communication is achieved when the senders message fully interpreted by the receiver with full of emotions. Hand gestures and facial expressions play an important role in the learning process of deaf persons. Sign language is greatly influenced by hand gestures recognition. Hand gesture plays vital role in understanding sign language [1]. It can be taken from live camera in the form of moving hand gestures or in the form of still images [3]. In our research we will consider only still images of hand gestures.

The persons who can't listen by birth are called deaf persons. Deaf persons can't listen any voice through their ears.Teaching them verbally is not effective way of communication. There is a need of special language for their learning purpose. That is called a "Sign Language". Sign Language is used to understand the conveyed message from others. American Sign Language has 24 different hand postures. Each posture shows a unique ASL letter. The following Fig. 1 shows American Sign Language alphabets. Sign Language field is very vast. The study of Hand gestures are always being a very tough to learn. A machine could not be recognized gestures until or unless the machine is professionally trained.

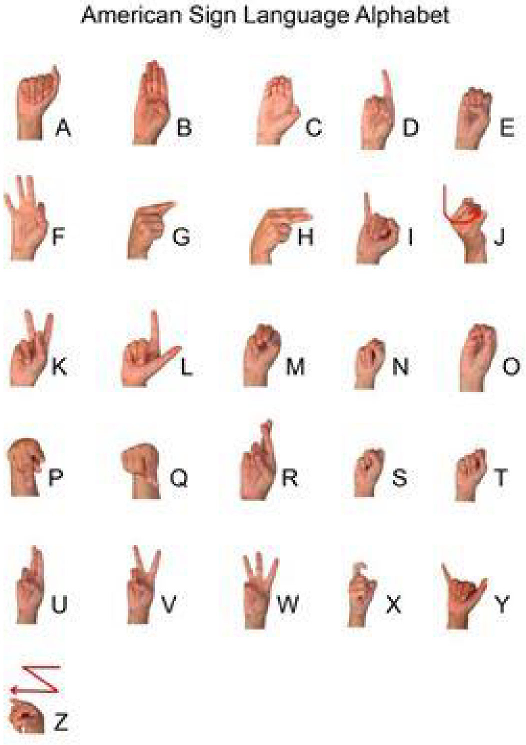


Fig. 1. Hand Gestures[21]

The above data set of sign American Sign Language is taken form a well-known website "kaggle" [21]. The data set did not contain the letter "J" and "Z". It is because visually similarity of these two signs with others.

According to a rough counting in a research [3], almost there are 500,000 to 2,000,000 deaf people's uses the sign language for communication with one another. The counting figure may be different from other proved research, but everyone would be agreed that the sign language is at the third most wanted and most used all over the world [3].

We can build a model to recognized hand gestures using

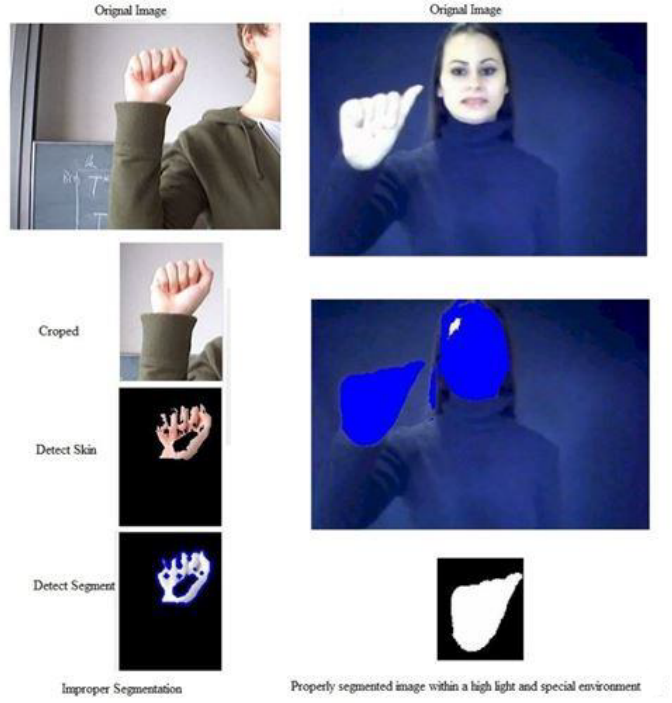


Fig. 2. Skin detection.

different techniques. In the past, developers used a finger technique, in which a user uses a finger mouse to capture fingers [4], using skin colour detection from any useful algorithm [3], gloves technique that was used neural net [5], feature extraction technique using Scale-Invariant Feature Transform(SIFT) algorithm [6]. All of these techniques are very tough to implement. From the above source [3] talked, a skin detection algorithm is used to detect the skin colour.

A special environment is created for skin detection with sufficient lightning conditions. There are some constraints that needs to be satisfied. First, the background color must be different from skin color. Second, algorithms fails to perform well under different backgrounds and colored clothes. For skin detection, the user should be there in a specially created environment in which a sufficient light was required. In case of less light or different background and cloth colours, the skin detection algorithm did not work properly and did not detect skin properly. Fig. 2 shows the detail. Just like skin detection algorithms, the neural net and SIFT techniques are also difficult to implement. A neural net algorithm takes a lot of time to process digital images.

In this paper, we present simple but efficient technique for ASL recognition. We provide a comprehensive analysis on different techniques with feature extraction and different algorithms. we use tools like (Weka and Rapid Miner) and achieved 99% accuracy on test data. Methods used in these techniques, experimental results and assumptions are described in coming Sections II, III, IV and V accordingly.

II. Literature Review

Focus of our research work is sign language recognition using hand gestures. It is important to understand gestures so that true semantics of communication can be grasped. According to a rough counting in a research [3], there are more than 500,000 to 2,000,000 deaf people's using the sign language for communication with one another. The counting figure may be different from other researcher's research, but everyone would be agreed that the sign language is at the third most wanted and most used all over the world [3]. There are different sign languages such as American Sign Language (ASL), Indian Sign Language (ISL), Arabic Sign Language (ArSL), Tamil Sign Language (TSL), Koran Sign Language (KSL), Japanese sign Language (JSL) and many more [1]. In our research work, we are focusing on American Sign Language (ASL).

Gesture recognition was used in 1993 for the first time [3]. Later for recognizing dynamic gestures, Dynamic Time Wrapping (DTW) technique was used [4]. Hidden Markov Model (HMM) was also used for recognizing sign language's shape [5], [6]. They used HMM efficiently and accuracy of sign language recognition reached to 94%. Later, it was found that accuracy dropped to 47.6%, when system was by a person other than those images were used for training. If both person's images are used for training then accuracy level increased [7]. Major limitation of HMM was its context dependency. HMM was used with 3D data to classify 53 ASL and attained accuracy of 89.91% [8].

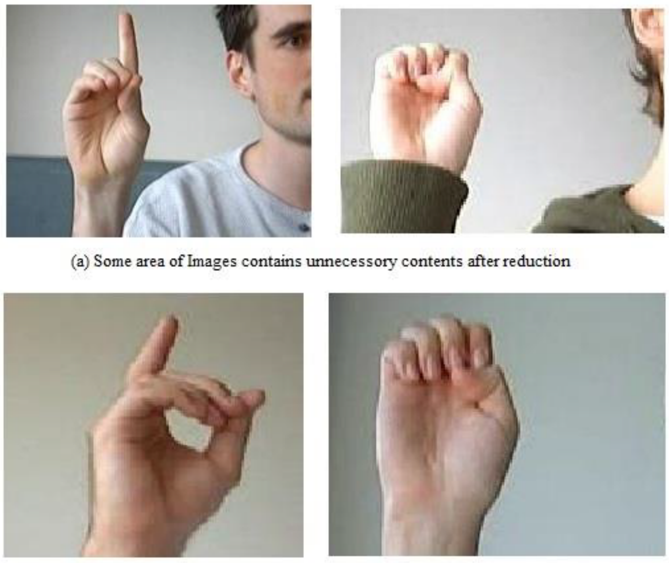
Image acquisition and Pre-processing is the backbone of gesture recognition. In the Past, image is acquired using Leap Motion Controller (LMC), Kinect and vision based approaches [19]. LMC can acquire signals 200 frames per sec [18]. It has been widely used for hand gesture recognition tasks [20].

In the past, researchers used many methods for recognizing hand gestures. Some used a finger technique, in which a user uses a finger mouse to capture fingers [9], using skin colour detection from any useful algorithm [10], [11], gloves technique that used neural nets [12], [13], feature extraction technique using ”SIFT” algorithm [14], [15]. Viola- Jones method was used for detecting skin, skin colour was used to detect hand. After hand detection features are extracted using SIFT and Support Vector Machine(SVM) is used for classification purpose [15].

Skin detection techniques are more sensitive in this process, the user should be there in a specially examined environment, which requires a specific intensity light. There are various other constraint like: The background colour must be different form skin colour, light should be constant and background and clothes should be simple.Various algorithms fails to perform in skin detection, if these condition cannot be fulfilled [9]. Fig. 2 shows the detail. Just like skin detection algorithms, the neural net and SIFT techniques are also not efficient both in accuracy and time. A neural net algorithm takes relatively more time than other techniques to process digital images [4]. So, Neural net is not suitable for real time skin detection [16]. K Nearest Neighbour algorithm was used with PCA and achieve 96% accuracy [17].

III. Proposed Methodology

The proposed method is based on combination of Histogram of Oriented Features (HOG), Local Binary Patterns (LBP) and statistical features. It focuses on the algorithmic



(a) Some area of Images contains unnecessory contents after reduction

(b) Accurate image after reduction

Fig. 3. Accurate and inaccurate image reduction [21].

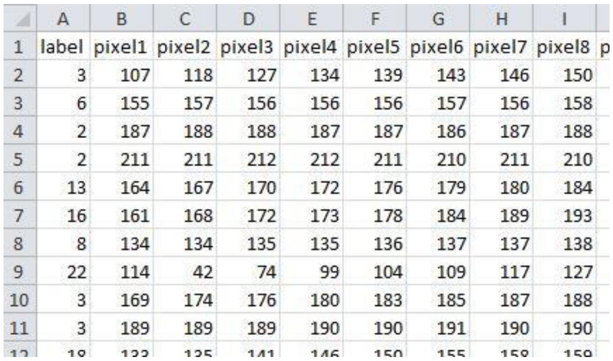
analysis of different tools and techniques with respect to time and accuracy. Sign language recognition can be divided into four major steps [16].

1) Image Acquisition
2) Image Segmentation
3) Feature Extraction
4) Hand Gesture Recognition

A. Image Acquisition and Pre-Processing

Data set for proposed work are taken from a well-known source “Kaggle”. Other Data sources are also visits but we could not found enough data for hand gestures in digital images or Comma Separated Value (CSV) file format. At Kaggle, we found two data sets for hand gestures one is in the form of set of colour images and other is in the form of CSV file as shown in Fig. 5. Colour images has 9 folders and each folder has 241 colour images with an excel file. Excel file contains image name and the images dimensions from x1, y1, x2 and y2. After Pre-processing (reduction of images according to x1, y1, x2 and y2 given points in excel file) in MATLAB 2018a, we found images are not good as they still contains some unnecessary contents. In these images we found some images did not have proper cutting contents. According to the given dimensions in excel file, some hand gestures was cute and they did not express the accurate meaning of sign language. Fig. 3 shows the cute area of hand gestures.

Data set in the form of CSV files with the following name “signministtrain” and “signministtest” are checked. The training data set contains 27,455 digital images record and test data set contains 7,172 digital images records. These files contain pixel values of a grey scale digital image in the form 785 columns and the last column contains the class of each image. First 784 columns have pixel values of each image with dimension 28x28. Have a look of these CSV file in the following Fig. 4. ‘ First and very important task is to separate each file into its original graphical form, from its pixel values. For this purpose we use set of instruction in MATLAB to convert each pixel into an image. After analysing the “signministtrain” file, we have the following labels or classes for image dataset. In CSV file 5 each record has a label in numeric format, which means that each numeric digit is represent a sign language letter.



	A	B	C	D	E	F	G	H	I
1	label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8
2	3	107	118	127	134	139	143	146	150
3	6	155	157	156	156	156	157	156	158
4	2	187	188	188	187	187	186	187	188
5	2	211	211	212	212	211	210	211	210
6	13	164	167	170	172	176	179	180	184
7	16	161	168	172	173	178	184	189	193
8	8	134	134	135	135	136	137	137	138
9	22	114	42	74	99	104	109	117	127
10	3	169	174	176	180	183	185	187	188
11	3	189	189	189	190	190	191	190	190
12	18	133	135	141	146	150	155	158	159

Fig. 4. Data Set with labels

Label in CSV File	Assigning Letters	Label in CSV File	Assigning Letters	Label in CSV File	Assigning Letters
0	A	8	I	17	R
1	B	9	K	18	S
2	C	11	L	19	T
3	D	12	M	20	U
4	E	13	N	21	V
5	F	14	O	22	W
6	G	15	P	23	X
7	H	16	Q	24	Y

Fig. 5. Labels in CSV File

The following algorithm takes each row form csv or excel file from very first record to end of the file and reshape the each row vector from 1x784 columns to 28x28 columns vector. 28x28 column vector stores in an array and convert it into a graphical image file and store it on the given location.

1) Read CSV file and convert into an excel file format.
2) Resize each 1x784 column to 28x28 vector column by reading each record in excel file.
3) Generate digital images by reading each 28x28 vector.
4) Store the file at given location in digital form.

After executing this algorithm we got 27,455 training and 7,172 test images for hand gesture data set. After pre-processing the following Fig. 6 shows data set is generated in digital images. After analysing the “signministtrain” file, we have the following labels or classes for image dataset.

B. Feature Extraction

We use simple feature extraction technique in this paper to make is simple to simplest. MATLAB 2018a is used for feature extraction techniques. HOG (Histogram of Oriented Gradient) and LBP (Local Binary Pattern) are the important feature extraction techniques using in MATLAB.

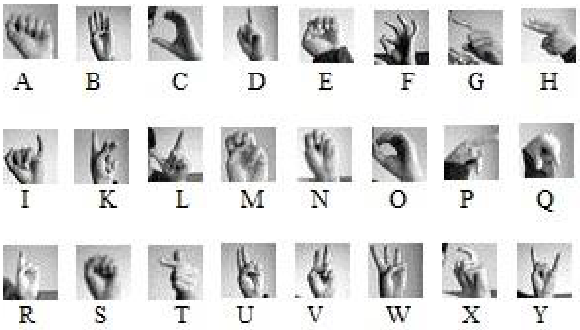


Fig. 6. After pre-processing, hand gestures data set.

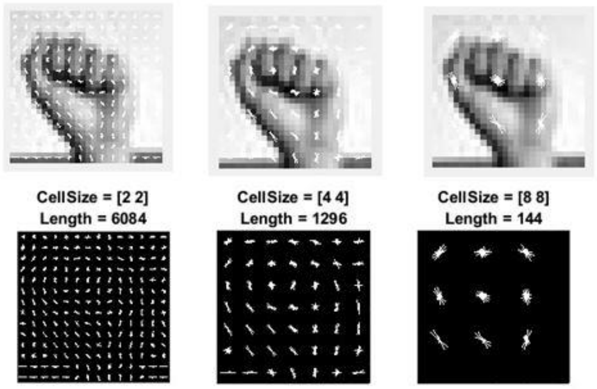


Fig. 7. HOG cell size and features.

1) HOG (Histogram of Oriented Gradient): Histogram of Oriented Gradient is a very useful technique to extract features in digital image processing. Histogram of Oriented Gradient has a variety of parameter to extract efficient features. Here, we discuss some of these parameters.

a) HOG Cell Size: HOG cell size is 2-element vector. It specified the number of pixel in digital image. Cell size [2 2] show that HOG get the features of 2-by-2 pixels. Cell size may vary on different values, it may be 4-by-4, 8-by-8, 16-by16 or 31-by-32 as shown in Table I.In Larger images, we set the cell size large. As we increase the cell size, number of features are also increased. In our feature extraction technique, we use cell size [2, 2], [4, 4] and finally [8, 8] for better experiments. Experiments are shown in experiments and results section.Default value of cell is [8 8]. Fig. 7 shown the number of features on cell size [2 2], [4 4] and [8 8].

b) Num Bins: Num Bins describes the number of features value contained in a cell. Default value of Num Bins is 9, which means total 9 number of features will collected from a cell. The following command is used for HOG feature extraction: extractHOGFeature (image path, 'Cell Size' [8 8], 'Block Size' [2 2], Num bins=9).

2) LBP (Local Binary Pattern): To encode the texture information, LBP (Local Binary Pattern) technique is used vastly. For LBP feature extraction, the following command is used: No-of-Features = extractLBPFeatures (Image Path).

TABLE I. HOG AND LBP FEATURES

HOG Features			LBP Features
Cell Size	**Block Size**	**Num of Bins**	**Cell Size**
[8 8]	[2 2]	9	[28 28]
[8 8]	[2 2]	9	[28 28]
[8 8]	[2 2]	9	[28 28]
[8 8]	[2 2]	9	[28 28]

TABLE II. STATISTICAL FEATURES

Statistical Features			
A.M	**STD**	**Variance**	**Skewness**
Yes	No	No	No
Yes	Yes	No	No
Yes	Yes	Yes	No
No	No	Yes	No

3) Statistical Feature Measurements: Based on above two techniques HOG and LBP, we use some additional statistical techniques for better feature extraction as shown in Table II. We use Mean, Standard deviation, variance and skewness for additional feature extraction techniques. The algorithm reads the files directory which contains training data set images. Get each file one by one and extract the HOG, LBP and other statistical feature measurements. Store this features into a CSV file on the specified location. The following algorithm is used for feature extraction in MATLAB 2018a.

1) Read stored images one by one from specified directory.
2) Generate HOG, LBP and other statistical feature measurements.
3) Set labels against each feature.
4) Store features into a CSV file in specified directory.

IV. EXPERIMENTS AND RESULTS

Weka-3-9-3 and Rapid Miner 9.1.0 are used for experiments of training and test data models as shown in Fig. 8.

A. Weka

Weka has a large collection of algorithms for creating effective models in machine learning techniques. Weka provides important facilities in regard of data preparation and classification. Weka also has regression algorithms and clustering algorithms for unsupervised learning. Decision trees and random forest algorithm are also include in weka for supervised learning. We use Navie Bayes, Lazy IBK and Random Forest algorithms.

B. Rapid Miner

Rapid Miner is the most latest software used for machine learning, data mining, deep learning and text mining. Rapid Miner introduced in 2006 and it has wonderful GUI and provides a lot of options to build a model for machine learning [22]. The algorithms KNN , Neural Net, Generalized Leaner Model, Deep Learning, Naïve Bayes, Random Forest and Decision Trees are used in Rapid Miner 9.1.0 for effective machine learning model.

TABLE III. EXPERIMENT1 RESULTS

HOG Features			LBP Features
Cell Size	Block Size	Num Bins	Cell Size
[4 4]	[2 2]	9	[28 28]
[8 8]	[2 2]	9	[28 28]
[8 8]	[2 2]	9	No
Extracted Features	Results		
No of Features	Lazy.IBK	Random Forest	
1355	95.24%	94.11%	
204	96.75%	94.95%	
145	96.50%	94.24%	

TABLE IV. EXPERIMENT2 RESULTS

Extracted Features	Results	
No. of Features	Lazy.IBK	Random Forest
232	96.83%	95.60%
260	96.40%	95.20%
288	96.19%	94.30%
232	96.87%	95.39%

C. Experiment 1

The following Table III shows the detail of experiment 1. Different results are showing on different parameter settings of cell size as shown in Fig. 9. Lazy.IBK and Random forest gives the accuracy of 96.75% and 94.95% on cell size [8,8] in weka.

D. Experiment2

In our 2nd experiment we use some statistical measurement for extracting effective features. Table IV shows the detail of experiment 2. Lazy IBK gives the highest accuracy 96.87% on combination of HOG, LBP, Arithmetic mean and Variance. Random Forest gives the highest accuracy at HOG, LBP and Arithmetic mean.

E. Experiment3

In 3rd experiment we use only HOG features with LBP and increase the HOG cell size from [2,2] or [4,4] to [8,8] with Numbins 12. We got much better accuracy using Lazy.IBK up to 97.37%. Whereas Random Forest did not achieve much better accuracy. At NumBins 15, Lazy.IBK achieve better accuracy up to 97.70% whereas Random forest achieve accuracy 95.70%. Table V shows the detail of experiment.

F. Experiment4

The final experiment in weka we use NumBins 25 with cell size [8,8] and Lazy.IBK achieved 98.24% accuracy. Random Forest did not achieve much better accuracy than previously achieved 95.70%. Table VI shows the detail of experiment 4 and different graphs represents the data set accuracy of confusion matrix generated by Lazy.IBK. The below Fig. 9 shows the comparisons of both algorithms on different number of features. From the graph it shows that Lazy.IBK gives much better accuracy than Random Forest algorithm.

TABLE V. EXPERIMENT3 RESULTS

HOG Features			LBP Features
Cell Size	Block Size	Num Bins	No. of Features
[8 8]	[2 2]	12	[28 28]
[8 8]	[2 2]	15	[14 14]
[8 8]	[2 2]	15	No
Extracted Features	Results		
No. of Features	Lazy.IBK	Random Forest	
348	97.37%	95.70%	
477	97.21%	95.94%	
241	97.70%	95.70%	

TABLE VI. EXPERIMENT4 RESULTS

HOG Features			LBP Features
Cell Size	Block Size	Num Bins	No. of Features
[8 8]	[2 2]	25	No
[8 8]	[2 2]	23	No
[8 8]	[2 2]	27	No
[8 8]	[2 2]	29	No
Extracted Features	Results		
No. of Features	Lazy.IBK	Random Forest	
401	98.24%	95.64%	
369	98.14%	95.62%	
433	98.14%	95.58%	
465	98.22%	95.12%	

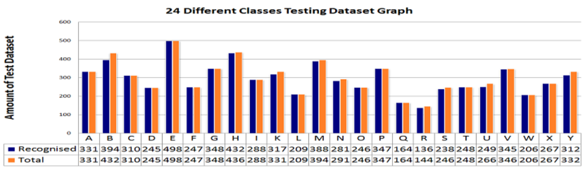


Fig. 8. Training and Testing Data Graph

G. Experiment5

After lot of experiments, we use a different tool “Rapid Miner 9.1.0”. Rapid Miner is very sophisticated tool used for data mining. Algorithms (K-NN, Neural Net, Generalized Linear Model, Deep Learning, Naive Bayes, Random Forest and Decision Tree) are used to train a model and test our data set. Table VII shows the test results. Unfortunately, we could not achieve much better results than Weka. We got highest result 98.03 using K-NN.

Fig. 10, 11 shows the graph of accuracy and time taken of Rapid Miner algorithms. KNN shows top accuracy and time

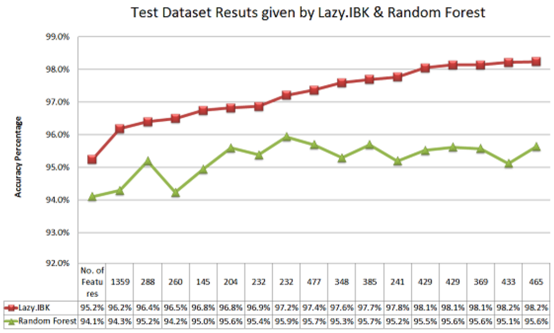


Fig. 9. LazyIBK and Random Forest Result Comparison

TABLE VII. RESULT COMPARISON OF DIFFERENT CLASSIFIERS ON RAPID MINER

Algorithm	Accuracy (%)	Time
KNN	98.03	0:07:41
NN	97.56	0:41:50
GLM	96.83	1:07:43
GL	96.72	5:45:13
NB	74.57	0:03:56
RD	40.63	6:19:52
DT	11.34	4:50:03

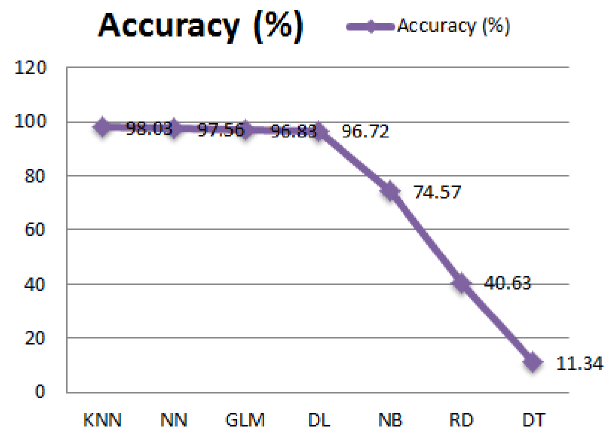


Fig. 10. Algorithms accuracy comparison.

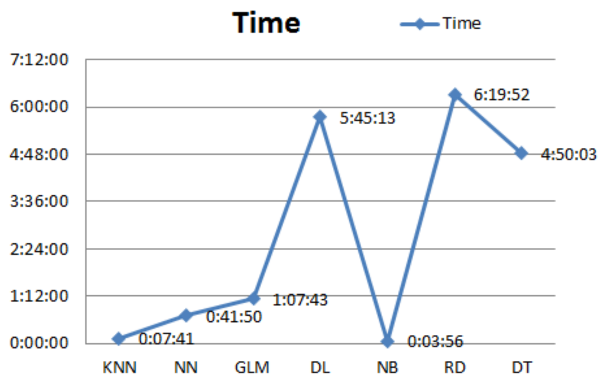


Fig. 11. Algorithms time comparison.

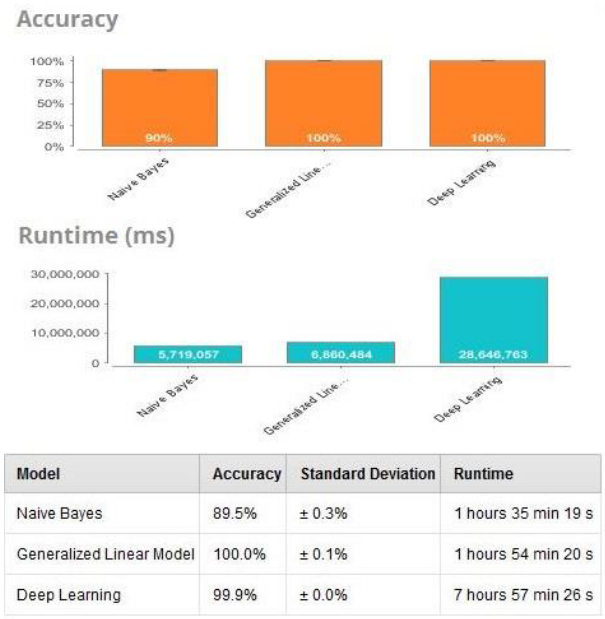
in the graph.

H. Experiment6

After lot of experiments 8 in Rapid Miner, we decided to use auto model facility in Rapid Miner to build a model for training data. After building model, we achieved impressive results. Using HOG features with Cell [8 8] and NumBin 25, we got the success to build a model with 100% accuracy result using 'Generalized Linear Model' in auto model, in Rapid Miner. Using 'Deep Learning' we achieve 99.9% results whereas "Naïve Bayes" achieved 89.5% our results. Fig. 12 shows the detail of our achieved results on given test data set.

V. Conclusion and Result Analysis

From the above experiments, numbers of well-known algorithms are test on training and testing data provided by kaggle. Their results are clearly giving a message that Rapid Miner using auto model gives 100% accuracy. Whereas building up model using Lazy.IBK in Weka 3-9-3 gives 98.24% accuracy. Naive Bayes and Decision Tree did not achieve much better results and we did not add their results in this paper. On the other hand in Rapid Miner we use the following algorithms: KNN (K-Nearest Neighbour), Neural Net, Generalized Leaner Model, Deep Learning, Naïve Bayes, Random Forest and Decision Trees. Auto model is also used in Rapid Miner with following algorithms: "Nave Bayes", "Generalized Leaner Model" and "Deep Learning" in Rapid miner. In Rapid Miner 9.1.0, using different algorithms, we achieved highest accuracy 98.03% from K-NN (K Nearest Neighbor) algorithm.



Model	Accuracy	Standard Deviation	Runtime
Naive Bayes	89.5%	± 0.3%	1 hours 35 min 19 s
Generalized Linear Model	100.0%	± 0.1%	1 hours 54 min 20 s
Deep Learning	99.9%	± 0.0%	7 hours 57 min 26 s

Fig. 12. Results on Rapid Miner using Auto Model Algorithm.

In Rapid Miner using auto model, "Generalized Linear Model" produced 100% results whereas "Deep Learning" also produced 99.9%results and "Naive Bayes" achieved 89.5% results.

Rapid Miner performs extra ordinary performance on test data. Rapid Miner achieved 100% results as compare to Weka tool.

Acknowledgment

We would like to thank University of Lahore research initiative, HOD of CS&IT and Directer Academics.

References

[1] M. J. Cheok, Z. Omar, and M. H. Jaward, "A review of hand gesture and sign language recognition techniques," Int. J. Mach. Learn. Cybern., vol. 10, no. 1, pp. 131–153, 2019.

[2] A. Akoum and N. Al Mawla, "Hand Gesture Recognition Approach for ASL Language Using Hand Extraction Algorithm," J. Softw. Eng. Appl., vol. 08, no. 08, pp. 419–430, 2015.

[3] J. (Junta D. de la S. E. de N. (SEN)) Sancho Rieger et al., "Vision Based Gesture Recognition Using Neural Networks Approaches: A Review," Int. J. Hum. Comput. Interact., vol. 32, no. 1, pp. 480–94, 2008.

[4] H. Zhong, J. Shi, and M. Visontai, "Detecting Unusual Events in Video," IEEE Conf. Comput. Vis. Pattern Recognit., vol. 2, no. June, p. II–819, 2004.

[5] R. Yang and S. Sarkar, "Gesture recognition using hidden Markov models from fragmented observations," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 1, pp. 766–773, 2006.

[6] T. Starner, J. Weaver, and A. Pentland, “Real-time American sign language recognition using desk and wearable computer based video,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 12, pp. 1371–1375, 1998.

[7] K. Grobel and M. Assan, “Isolated sign language recognition using hidden Markov models,” pp. 162–167, 2002.

[8] C. Vogler and D. Metaxas, “ASL recognition based on a coupling between HMMs and 3D motion analysis,” pp. 363–369, 2002.

[9] B. K. Ko and H. S. Yang, “Finger mouse and gesture recognition system as a new human computer interface,” Comput. Graph., vol. 21, no. 5, pp. 555–561, Sep. 1997.

[10] T. Khan and H. Pathan, “Hand Gesture Recognition based on Digital Image Processing using MATLAB,” Int. J. Sci. Eng. Res., vol. 6, no. 9, pp. 338–346, 2015.

[11] K. M. Lim, A. W. C. Tan, and S. C. Tan, “A feature covariance matrix with serial particle filter for isolated sign language recognition,” Expert Syst. Appl., vol. 54, pp. 208–218, 2016.

[12] M. Maraqa and R. Abu-Zaiter, “Recognition of Arabic Sign Language (ArSL) using recurrent neural networks,” 1st Int. Conf. Appl. Digit. Inf. Web Technol. ICADIWT 2008, pp. 478–481, 2008.

[13] V. Adithya, P. R. Vinod, and U. Gopalakrishnan, “Artificial neural network based method for Indian sign language recognition,” 2013 IEEE Conf. Inf. Commun. Technol. ICT 2013, no. Ict, pp. 1080–1085, 2013.

[14] N. M, “Alphabet Recognition of American Sign Language : A Hand Gesture Recognition Approach Using Sift Algorithm,” Int. J. Artif. Intell. Appl., vol. 4, no. 1, pp. 105–115, 2013.

[15] M. HafizurRahman and J. Afrin, “Hand Gesture Recognition using Multiclass Support Vector Machine,” Int. J. Comput. Appl., vol. 74, no. 1, pp. 39–43, 2013.

[16] S. Suharjito, M. C. Ariesta, F. Wiryana, and G. P. Kusuma, “A Survey of Hand Gesture Recognition Methods in Sign Language Recognition,” Pertanika J. Sicence Technol., vol. 26, no. 4, pp. 1659–1675, 2018.

[17] C. F. F. Costa Filho, R. S. De Souza, J. R. Dos Santos, B. L. Dos Santos, and M. G. Fernandes Costa, “A fully automatic method for recognizing hand configurations of Brazilian sign language,” Rev. Bras. Eng. Biomed., vol. 33, no. 1, pp. 78–89, 2017.

[18] M. Mohandes, S. Aliyu, and M. Deriche, “Arabic sign language recognition using the leap motion controller,” IEEE Int. Symp. Ind. Electron., no. June, pp. 960–965, 2014.

[19] H. Bhavsar and J. Trivedi, “Review on Feature Extraction methods of Image based Sign Language Recognition system,” Indian J. Comput. Sci. Eng., vol. 8, no. 3, pp. 249–259, 2017.

[20] L. Quesada, G. López, and L. A. Guerrero, “Sign Language Recognition Using Leap Motion,” pp. 277–288, 2015.

[21] “Sign Language MNIST — Kaggle.” [Online]. Available: https://www.kaggle.com/datamunge/sign-language-mnist. [Accessed: 26-Mar-2019].

[22] Hofmann, Markus and K. Ralf,RapidMiner: Data mining use cases and business analytics applications, CRC Press, 2013