ISSN 2278-3091
Volume 10, No.3, May - June 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse1391032021.pdf
https://doi.org/10.30534/ijatcse/2021/1401032021

Machine Learning Empowered Urdu Characters Recognition Mechanism

Ayima Zahra[1], **Maneeba Ashraf1**[1], **and Muhammad Sohaib**[1]
[1]Department of Computer Sciences, Lahore Garrison University, Lahore, Pakistan,
ayimazahid6@gmail.com,muneebaashrafbhatti012@gmail.com,md.sohaib@lgu.edu.pk

ABSTRACT

The study was conducted on digitally written Urdu characters. After a little preprocessing (resizing the images and separating different characters based on the classes), we applied different features extraction techniques and artificial intelligence algorithms on the same dataset and compared the results, to successfully predict digitally written Urdu characters. We managed to classify the data with an accuracy of 98%. But of course, these are results from the best combination of algorithms, feature extraction, and hyperparameter combination. There is still a huge issue of recognizing handwritten characters. Much work has been in previous years to successfully recognize characters but most of the research was done in English characters or digits and very less work is done in recognizing Urdu characters. One of the main reason for this is the same characters has different shapes depending on their position in the word. Moreover, some properties of Urdu like calligraphic nature also cause a lot of problems. To cover all these problems, we studied several feature extraction techniques and algorithms to drive the best possible results.

Key words: Character Recognition, Deep Learning Machine Learning, Neural Network, Urdu OCR.

1.INTRODUCTION

Optical character recognition (OCR) could be a branch of pattern recognition that is used to recognize written characters. The main goal is to convert the input characters into data which can be recognized by a computer. A lot of work has been done on OCR in the last several decades. Researchers have developed numerous mechanisms for OCR that corroborate machine learning algorithms to recognize characters written in different languages spoken worldwide. Rapid progress has resulted in very accurate systems that can successfully recognize most of the characters. Unfortunately, much work has not been done for the classification of Urdu characters. The reason is that recognition of character depends on a lot of different factors like wiring style and source script. Machine learning algorithms make use of geometrical aspects to recognize the data, fonts, and writing style, hence the same techniques are not likely to work on different scripts.

In this regard, S. Malik and S.A. Khan [1] developed a system that used to be able to recognize different letters and digits even a few two characters words in Urdu language. They used a rule-based on slant evaluation and conversion. The accuracy of the system was almost 93% for single characters and 78% accuracy for 2 character words. Similarly, S.A. Husain et al. [2] used a little different, segmentation free approach with 20 different structural features and was able to successfully recognize 18000 common words including single, double, and 3 character words. An artificial neural network was used in this study and achieved up to 98% accuracy. Besides, M.I.Razzak and S.A. Hussain [3] used a segmentation free approach that tried to predict online Urdu text. They used a hybrid classifier of hidden Markov model (HMM) and Fuzzy Logic and were able to predict characters up to 87.6% accuracy. K.U. Khan and I. Haider [4] applied many different classifiers like neural networks using backpropagation and probabilistic approaches and also made use of correlation on handwritten characters and were able to achieve up to 98% accuracy using probabilistic neural networks. Further, M. I. Razzak et al. [5] made use of both online and offline techniques for preprocessing to achieve maximum accuracy. Z. Ahmed and J. K. Orakzai [6] used neural networks based on a feed-forward approach to recognize Urdu characters. The recognition rate was 93.4 % keeping the input constant, also used feed-forward neural networks, and was able to achieve 98.3% accuracy on isolated Urdu characters. S. A. Hussain et al. [7] used K-SOM for pre-segmented isolated Urdu characters. Their system was able to predict with an accuracy of 80%. S. Sardar and A. Wahab [8] used 5 feature extraction techniques and KNN algorithms and were able to make classify characters with an accuracy of 97.12%. S.B Ahmed et al. [9] used to decide ligatures from Urdu words is another research region that can be performed utilizing the UNHD database. N.H.Khan and A.Adnan [10] proposed that In hand-engineered features, the structural features deal within the general structure of the character and could appear unseemly for a ligature based recognition system due to being huge in number and gigantic varieties of its character shapes, even have deployed basic connected components (CC), feature extraction as an array of facilitates, and picture comparison for Urdu acknowledgment. S. Sukanya et al. [11] propose two popular gadgets the Tauschek's perusing machine and Fournier Optophone were created from 1870 to 1931 to assist the blind examined. Akhter et al. [12] proposed that in preprocessing of Urdu content, it may be a common hone to expel frequent words but rare words are not expelled. Diverse considers showing that expelling uncommon words in preprocessing includes a perplexing impact in classification. K. Thiruthanigesan et al. [13] proposed that the technologies and research for Arabic OCR have advanced to more cleverly character recognition systems that back written by hand and cursive scripts. By O. Mukhtar et al. [14] Urdu is much closer to Arabic because it is composed in Nastaliq, the calligraphic style of the Persian–Arabic script. S.Naz et al. [15] stated huge complexities associated with Urdu Script make it a scant language to be considered for OCR.16.Muhammad Ismail et al. [16] proposed the new technique taken in this paper is to unite dynamic information got from the speaker mouth occurring during reformist housings of video got during communicated talk. Besides, the sound just, visual just and general media recognizers were considered inside seeing upheaval and exhibit that the expansive media recognizer has progressively unique execution. S. Pandya et al. [17] conducted a survey huge significant learning models with start to finish approach for Natural language handling that makes a more precise supposition on the word arrangement as well as addresses an alternate exploratory investigation for NLP errands as far as text and words with the assistance of CNN, RNN, LSTM, and GRU bidirectional Encode-Decoder. Various NLP undertakings are engaged with semantic execution to comprehend also, create a total sentence where diverse Deep learning models address mindfulness about various NLP errands including consecutive data preparing.

In this work, an effort has been made to devise a mechanism that can automatically recognize digitally written Urdu characters with high accuracy. The character recognition mechanism consists of two steps, i.e., 1) features extraction which can adequately represent the original text, 2) Consolidation of machine learning algorithms to classify the data represented by the feature space. Diverse machine learning calculations like support vector machines (SVM), artificial neural arrange (ANN), and decision tree classifiers (DTC) have been used in the experiments to obtain different results so that performance of these algorithms could be compared with each other in term of the classification accuracy. Different feature extraction techniques like pixels greyscale values and HOG features of images, etc., have been utilized in the experiments. The aim was to check which type of feature is best suited to generate satisfactory results provide different types of machine learning classifiers. Also in the end it is concluded that which combination of features and classifiers provides satisfactory recognition results.

2. CHARACTER RECOGNITION OBSTACLES

Different problems might occur in the recognition of Urdu characters. These difficulties can be divided into two main categories.

A. *General Difficulties*

In general, most of the time, one can face the following difficulties throughout the process. Few of the common difficulties are:

1) Presence of lines and characters other than words. (Like emoticons and punctuation).
2) Presence of noise in the data.
3) Linguistic problems might occur. For instance, if a dictionary for checking to spell has been used to improve the accuracy then it might consider names and some other acronyms as false words, therefore, reducing overall accuracy.

B. **Urdu Specific Obstacles**

The following difficulties might occur while working with specifically Urdu text.

1) One of the main problems is that Urdu characters can have more than one shape depending on their position in the word. As shown in Figure 1, the same character in different positions in a word.

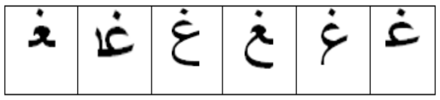


Figure 1: Different positions of the Character

2) Writers often write the same words differently depending on the context.
3) Characters can be misclassified because of the change in shape that occurs due to their position in a word. For example, ‘laam’ can be classified as ‘alif’ because it looks like ‘alif’ when it is at the start of a word.

3. METHODOLOGY

Different steps need to be taken so that a computer may recognize Urdu characters. These include data collection, labeling, preprocessing, classification, and prediction. These steps have further sub-steps which are explained below.

A. *Dataset Description*

The dataset used in this work is composed of in total 5500 images of digitally written isolated Urdu characters as shown in **Figure 2**. These letters are used to developed sentences.

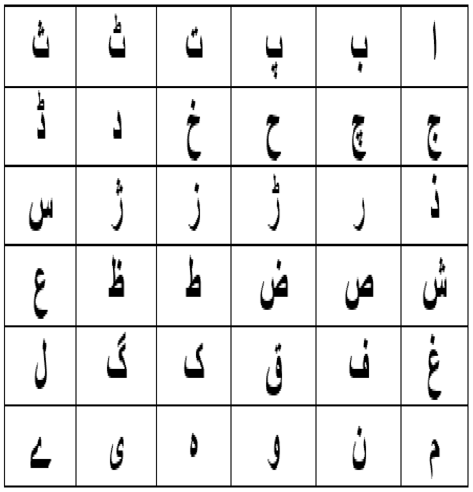


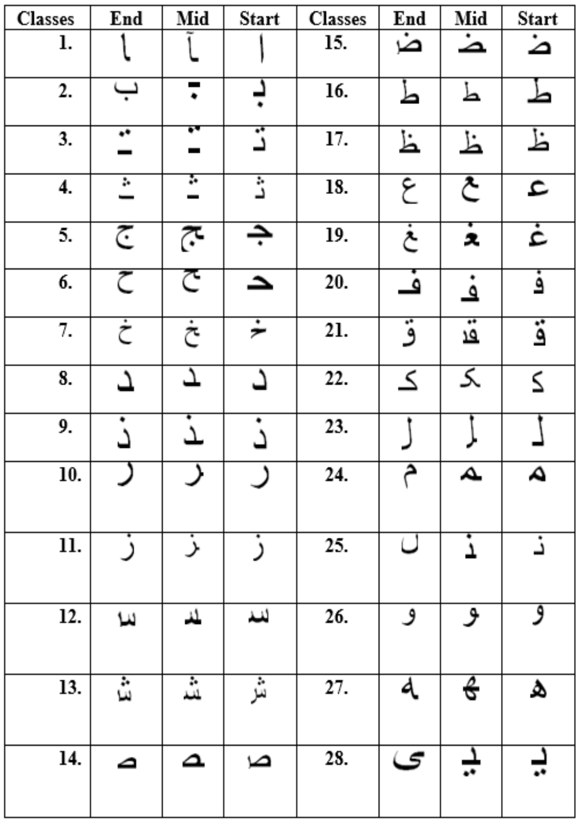
Figure 2: Letters of Urdu Script

Table 1, shows the distribution of the dataset i.e. 80% training and 20% testing. The dataset is further divided into 28 different classes based on the number of letters in the Urdu language.

Table 1: Dataset Distribution

Dataset Description	**Total**
No. of Images in the Dataset	5500
Training Dataset	4400
Testing Dataset	1100
No. of Classes	28

Figure 3, shows the images of cropped characters from digitally written Urdu words. Each image has a white background and with the black character on it. Each class contained every possible shape of the given character. For example, ‘baa’ was separated into 3 further classes ‘baa-start’, ‘baa-mid’, and ‘baa-end’ to achieve more accuracy.



Classes	End	Mid	Start	Classes	End	Mid	Start
1.	ﺎ	آ	ا	15.	ﺾ	ﻀ	ﺿ
2.	ﺐ	ﺒ	ﺑ	16.	ﻂ	ﻄ	ﻃ
3.	ﺖ	ﺘ	ﺗ	17.	ﻆ	ﻈ	ﻇ
4.	ﺚ	ﺜ	ﺛ	18.	ﻊ	ﻌ	ﻋ
5.	ﺞ	ﺠ	ﺟ	19.	ﻎ	ﻐ	ﻏ
6.	ﺢ	ﺤ	ﺣ	20.	ﻒ	ﻔ	ﻓ
7.	ﺦ	ﺨ	ﺧ	21.	ﻖ	ﻘ	ﻗ
8.	ﺪ	ﺪ	ﺩ	22.	ﻚ	ﻜ	ﻛ
9.	ﺬ	ﺬ	ﺫ	23.	ﻞ	ﻠ	ﻟ
10.	ﺮ	ﺮ	ﺭ	24.	ﻢ	ﻤ	ﻣ
11.	ﺰ	ﺰ	ﺯ	25.	ﻦ	ﻨ	ﻧ
12.	ﺲ	ﺴ	ﺳ	26.	ﻮ	ﻮ	ﻭ
13.	ﺶ	ﺸ	ﺷ	27.	ﻪ	ﻬ	ﻫ
14.	ﺺ	ﺼ	ﺻ	28.	ﻲ	ﻴ	ﻳ

Figure 3: Letter Variations

B. Preprocessing

There was some inconsistency in the dataset, hence, it could not be used directly in the experiments. The main problem was that images were having different dimensions. The maximum width and height were recorded to be 43*52 pixels respectively. The steps of the preprocessing phase are given below.

Step 1: All the images were resized to have the same size. To do so, we created a blank image of 60*60 pixel resolution and pasted the smaller image in the center of the 60*60 frame. We did this for all the images in our dataset.

Step 2: We separated our characters further into two, three, or four classes depending on their shape variation inside a word.

C. Feature Extraction

Different feature extraction techniques were adopted in this work to get better results.

1. Custom Features

Along with all other feature extraction techniques we used something of our own. We extracted some useful information which could be used as features and also used this information to train our model. We extracted the information as shown in **table 2**.

Table 2: Custom Feature Description

Name	Formulation
Height and Width	Pillow (PIL) library in Python is used to read the image in which img.size function is used to get the height and width of the image.
Height*Width Width/Height	If the dataset isn't adjusted appropriately in one shape, preprocessing will not be precisely done, comes about in less exact results. So we ought to align the dataset in one shape. For this, we utilized to deskew.
Black Pixels	This is to read the characters from the image having a white background. Adjust the image using ‘imadjust’ function. Afterward, do image segmentation by thresholding by setting a specific level. In the last perform image morphological operation to get the black pixels.
Black Pixels/Height Black Pixels/Width	This is to calculate the black pixels lying in a horizontal/vertical position. We will divide the black pixels calculated with the above-mentioned way with height and width to calculate the black pixels in the horizontal/vertical position.
Black Pixels/ Total Pixels	We have used PIL library to load the image. The total pixels will be its width * height. Earlier we extracted black pixels. Now divide black pixels/total pixels.
Average Black Pixel Density	In this function, we will pass the row/column parameter which contains all row and column array values to calculate the average black pixel density.
Number of Corner Points	Use OpenCV to load images using ‘imread’ function. Convert the image into grayscale and Perform threshold. Perform dilation on the threshold image. Cropping the text block based on black pixels calculated on the x-axis and y-axis using image.

2. Histogram of Gradients

After custom information retrieval about the image's pixels, we computed the histogram of gradients of each image. Each cell contained 9 bins for angles from 0 to 160. Each cell either contained an edge passing through it or not. If an edge passes through the cell it creates a perpendicular vector whose length was equal to the magnitude of the curve and assigns the value to that specific bin where it belonged. The features extracted in this way were 225, 9 bins for 25 cells each. All the data was stored in a CSV to be used later.

3. Crossing

Generally, it is some lines throughout the image i.e. from background to the foreground or vice versa. In simpler words, we can say that it counts the number of strokes throughout the image.

4. Moments

This technique is extensively used in image processing. In the image, processing moment is, we can say, a weighted average of pixel intensities; these are useful in describing the object after segmentation. We can define two-dimension moments of a greyscale image as shown in equation (1).

$$m_{pq} = \sum_{x}\sum_{y} x^{p}\, y^{q} f_{(x,y)} \qquad (1)$$

5. Projection Histogram

Projection histograms were first used in the 1950s for character recognition. In this technique, the image is scanned from one side to the other side. The numbers of pixels scanned in the foreground are calculated along with the pixels in the background thus giving the technique its name "Histogram projection"

$$\boldsymbol{H_i} = \sum_{j} \boldsymbol{f}\,(\boldsymbol{i}, \boldsymbol{j}) \qquad (2)$$

Equation (2) shows the basic definition of projection histogram, where i and j are rows and columns.

6. Zoning

In this technique, images are first divided into some overlapping and non- overlapping zones. After that, the number of pixels in the foreground is calculated and the densities of each zone are compared with the rest.

7. Celled projection

The image is first partitioned into different regions. And then its projection is taken from every region. The foreground pixels are considered to be 1 and background pixels are considered 0.

D. K-Fold Cross-Validation

A simple one-time train-test split does not guarantee the reliability of the developed model as it is prone to overfitting. To properly train the model and generate stable results we used the k-fold cross-validation criterion. The experiment was repeated multiple times while setting K value to 5, 10, and 15, and results were recorded after the completion of each experiment. By analyzing the results for different k values it was figured out the setting the value of k higher than 10 doses not enhances the classification performance of the network. Therefore, in the result analysis section, all the results presented are according to k = 10.

E. Result Analysis

For classification, none of the algorithms of ML is prevalent on all benchmark datasets since of two reasons imbalance dataset have varieties of their imbalance ratio K. Mehmood et al. [22], and during learning, the classifier performance is different on different datasets R. Duwairi et al. [23]. Machine learning includes anticipating and classifying the information and to do so, we employ various machine learning algorithms according to the dataset. A good review of imbalance-class dataset processing, methods, and applications can be found in G. Haixiang et al. [24]. We tried to classify character using all the possible combinations of algorithms and features. After feature extraction, we implemented 4 different machine learning algorithms including neural networks, KNN, decision tree classifier, and SVM on it. Moreover, in total seven different feature extraction techniques were used in this work which is already discussed in section 3. The results obtained from different combinations of classifiers and features set varied a lot. The details of the obtained results are presented in the proceeding sub-sections.

1. SVM

Table 3: Classification Result Obtained By SVM Classifier Using Feature Extraction Technique

Features	Accuracy	Parameters
Celled Projection	85%	C=10, Gamma=0.1, Kernel=rbf
Crossing	96%	C=1, Gamma=0.1, Kernel=poly
Custom Features	70%	C=30, Gamma=0.01, Kernel=rbf
HOG	96%	C=20, Gamma=0.1
Moments	28%	C = 1, Gamma = 0.1, Kernel = rbf
Project Histogram	42%	C=2, Gamma=0.1, Kernel=rbf
Zoning	89%	C=10, Gamma=15

SVM didn't give good results on moments and projection histogram feature extraction techniques as shown in **table 3**. On the other side, others gave better results on almost all the other features, HOG and Crossing being the best at 96%.

2. ANN

Table 4: Classification Result Obtained By ANN Classifier Using Feature Extraction Technique

Features	Accuracy	Parameters
Celled Projection	80%	Activation=logistic, Hidden layers=120 Solver=lbfgs
Crossing	94%	Activation=relu, Hidden layers=120,60 Solver=adam
Custom Features	59%	Activation=logistic, Solver=adam
HOG	97%	Activation=logistic, Solver=adam
Moments	1%	Activation=tanh, Hidden layers=14,7 Solver=lbfgs
Project Histogram	93%	Activation=relu, Hidden layers=120,60,30 Solver=adam
Zoning	85%	Activation=logistic, Hidden layers=25 Solver=lbfgs

ANN predicted the characters most accurately when used with the HOG feature with an accuracy of up to 97% as shown in **table 4**. While performed very badly when used with moments feature extraction.

3. KNN

Table 5: Classification Result Obtained By KNN Classifier Using Feature Extraction Technique

Features	Accuracy	Parameters
Celled Projection	84%	Neighbors =3
Crossing	98%	Neighbors =1
Custom Features	80%	Neighbors =3
HOG	93%	Neighbors =1
Moments	65%	Neighbors =3
Project Histogram	94%	Neighbors =1
Zoning	87%	Neighbors =3

Table 5, shows KNN gave the best results in our experiment at crossing features at 98% accuracy and performed very well on almost all the features.

4. DTC

Table 6: Classification Result Obtained By DTC Classifier Using Feature Extraction Technique

Features	Accuracy	Parameters
Celled Projection	78%	max depth=26, random state= 1
Crossing	87%	max depth=24, random state =1
Custom Features	82%	max depth=22, random state=4
HOG	88%	max depth=26, random state=2
Moments	73%	max depth=26, random state=2
Project Histogram	87%	max depth=28, random state=6
Zoning	82%	max depth=18, random state=1

With an accuracy of 88% DTC gives good results as shown in **table 6**. But as compare to other results obtained DTC is less accurate. Figure 4, Show the accuracy results. If we combine all the features and apply ML algorithms, we can have optimized accuracy.

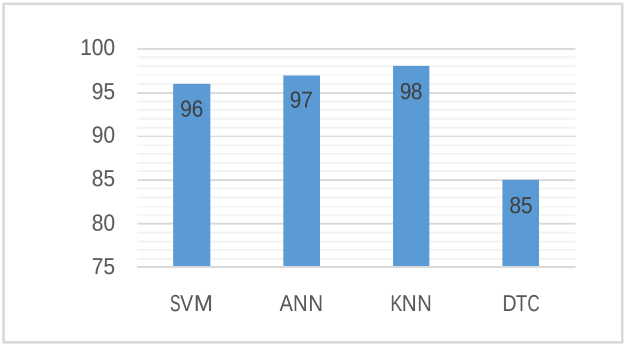


Figure 4: Accuracy Result

A drawback of ML classifiers is that they perform well with limited numbers of features. Classification of long text documents, the comparative study of shows that the maximum performance of SVM and KNN was achieved using 1000 and 2000 features respectively. The performance started to decrease when the numbers of features were increased from a specific value.

4. CONCLUSION

We tried to work on digitally written Urdu characters by first applying text preprocessing, then different feature extraction techniques, and machine learning algorithms. The efficacy of different machine learning algorithms tested on each feature separately as well as on the whole pool of features collectively. It was observed that by using the pool of all the seven features with machine learning algorithms overall 98% Urdu character recognition accuracy was achieved. After comparing the results of all the possible combinations of features and the machine learning algorithms, it can be concluded that KNN along with crossing features yielded a maximum accuracy of 98%. Relatively we can see that classification using the moment's features doesn't give the best results, and the results of HOG features were best in all the algorithms. Further, the possible improvement of this work can be the exploration of salient features from the visual data and the implication of suitable deep learning algorithms to achieve enhanced Urdu character recognition accuracy.

REFERENCES

1. Malik, S., & Khan, S.A. **Urdu online handwriting recognition**, *in Proceedings of the IEEE Symposium on Emerging Technologies, 27-31*,2005.
2. Husain, S., Sajjad, A., and Anwar, F. **Online Urdu character recognition system**, *In MVA2007 IAPR Conference on Machine Vision Applications*, 2007.
3. Razzak, Muhammad & Anwar, Fareeha & Husain, Afaq & Belaid, Abdel & Ramzan, Muhammad Sher**. HMM and Fuzzy Logic: A Hybrid Approach for Online Urdu Script Based Languages Character Recognition**, *Knowledge-Based Systems 23 (2010) 914–923*. 2010.
4. Khan, K.U., & Haider, I. **Online recognition of multi-stroke handwritten Urdu characters**, *2010 International Conference on Image Analysis and Signal Processing, 284-290*, 2010.
5. Razzak, Muhammad & Husain, Afaq & Muhammad, Sher & Khan, Zeeshan. **Combining Offline and Online Preprocessing for Online Urdu Character Recognition**, *Lecture Notes in Engineering and Computer Science. 2174*, 2009.
6. Ahmad, Z., Orakzai, J.K., Shamsher, I., & Adnan, A. (2007). **Urdu Nastaleeq Optical Character Recognition**, *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 1*, 2374-2377, 2007.
7. Hussain, S.A., Zaman, S., & Ayub, M.S. **A Self Organizing Map based Urdu Nasakh character recognition**, *2009 International Conference on Emerging Technologies*, 267-273, 2009.
8. Sardar, S., & Wahab, A.J. **Optical character recognition system for Urdu**, *2010 International Conference on Information and Emerging Technologies*, 1-5, 2010.
9. Ahmed, S.B., Naz, S., Swati, S., & Razzak, M.I., **Handwritten Urdu character 'recognition using one- dimensional BLSTM classifier**, *Neural Computing and Applications, 31*, 1143-1151, 2017.
10. Khan, N.H., & Adnan, A. **Urdu Optical Character Recognition Systems: Present Contributions and Future Directions**, *IEEE Access, 6*, 46019-46046, 2018.
11. Sukanya, S. & Gladwin, Joseph & Kumar, C. **A Tool for Extracting Text from Scanned Documents and Convert it into Editable Format**, *Vision Towards Emerging Trends in Communication and Networking (ViTECoN) 2019 International Conference on*, 2019.
12. M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, A. Mehmood and M. T. Sadiq. **Document-Level Text Classification Using Single-Layer Multisize Filters Convolutional Neural Network**, *in IEEE Access, vol. 8, pp. 42689-42707*, 2020.
13. Kanagasabai, Thiruthanigesan & Ragel, Roshan. **Optical Character Translation Using Spectacles (OCTS)**, *in IEEE 14th Conference on Industrial and Information Systems (ICIIS),186-191*, 2019.
14. Mukhtar, O., Setlur, S., & Govindaraju, V. **Experiments on Urdu Text Recognition**, 2009.
15.

16. Naz, Saeeda & Umar, Arif & Ahmad, Riaz & Ahmed, Saad & Shirazi, Syed & Siddiqi, Imran & Razzak, Muhammad. **Offline Cursive Urdu-Nastaliq Script Recognition using Multidimensional Recurrent Neural Networks**, *Neurocomputing, 177*, 2016.
17. Muhammad Ismail Mohmand, Amiya Bhaumik, Muhammad Humayun, Qayyum Shah. **The Performance and Classifications of Audio-Visual Speech Recognition byUsing the Dynamic Visual Features Extractions**, *in International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No.5,* September - October 2019.
18. Sheetal S. Pandya, Nilesh B. Kalani. **Review on Text Sequence Processing with use of different Deep Neural Network Model**, *in International Journal of Advanced Trends in Computer Science and Engineering,Volume 8, No.5*, September - October 2019.
19. Vapnik, V., "**The Nature of Statistical Learning Theory**," *NY: Springer-Verlag*. 1995
20. April, April & Basu, Jayanta & Bhattacharyya, Debnath & Kim, Tai-Hoon. **Use of Artificial Neural Network in Pattern Recognition**,". *In International Journal of Software Engineering and Its Application, Vol. 4,* 2020.
21. Lee, J. A., and Almond, D. P. **A neural-network approach to fatigue-life prediction**, *Fatigue in Composites 2003, Pages 569-589,* 2003.
22. Novakovic, Jasmina & Veljovic, Alempije & Ilic, Sinisa & Papic, Milos. **Experimental Study Of Using The K-Nearest Neighbour Classifier With Filter Methods**, *in Computer Science And Technology*, 2016.
23. Patel, Harsh & Prajapati, Purvi., **Study and Analysis of Decision Tree Based Classification Algorithms**, *International Journal of Computer Sciences and Engineering, 6, 74-78,* 2018.
24. Patel, Bhaskar. **Efficient Classification of Data Using Decision Tree**, *Bonfring International Journal of Data Mining, 2,* 2012.
25. Mehmood, K., Essam, D., & Shafi, K. **Sentiment analysis system for roman urdu**, *in Intelligent Computing (Advances in Intelligent Systems and Computing). Cham, Switzerland: Springer, pp. 29–42,* 2019
26. Duwairi, Rehab. **A Study of the Effects of Preprocessing Strategies on Sentiment Analysis for Arabic Text**, *Journal of Information Science. 40. 501-513.* 2014.
27. Guo, H., Li, Y., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. **Learning from class-imbalanced data: Review of methods and applications**, *Expert Syst. Appl., 73, 220-239,* 2017.