International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Approach using CNN for Recognition of Devanagari Handwritten Content

Subhash Rathod[1]*, Mangesh D. Salunke[2], Manisha Bhende[3], Meghna Yashwante[4], Uma B. Karanje[5], Yogesh B. Dongare[6], Vaibhav D. Rewaskar[7]

Submitted: 26/04/2023 **Revised**: 25/06/2023 **Accepted**: 06/07/2023

Abstract: This document discusses the use of Convolutional Neural Networks (CNN) for Devanagari Handwritten Text Recognition (DHTR) tasks. DHTR is a complex task due to the variability and diversity of handwritten characters in the Devanagari script. CNNs are a type of deep learning algorithm that can automatically learn features from images and are widely used in image recognition tasks. This paper presents a CNN-based approach for DHTR the proposed method has shown out performed on a popular dataset commonly used for evaluating image recognition systems. The proposed approach involves a preprocessing step to normalize and segment the input images, followed by a CNN architecture that consists of several convolutional layers and fully connected layers. The network is trained using a large dataset of labeled Devanagari characters. The results show that the proposed approach achieves high accuracy in recognizing Devanagari characters, and can be applied to real-world applications such as document digitization and text-to-speech conversion.

Keywords: *Devanagari, Handwritten text recognition, CNN, Deep learning, Image recognition*

1. Introduction

The term "handwritten character recognition" describes the process of extracting characters from images, documents, and other types of sources. Handwritten characters are, in many ways, optically recognised during the identification of handwritten characters. OCR has several uses across many industries, including text mining and machine translation. In order to recognise objects, mathematical methods, and neural networks are a couple of the currently popular ways. [1] Handwritten text recognition is a challenging problem in the field of computer vision and machine learning. Devanagari Handwritten Text Recognition (DHTR) is even more complex due to the variability and diversity of handwritten characters in the Devanagari script. Devanagari is a commonly used script in India that is utilized for writing several languages, such as Hindi, Marathi, Nepali, and more. Sanskrit. With the increasing use of digital technology, the need for automatic DHTR systems has become more important for applications such as document digitization, text-to-speech conversion, and language translation. Convolutional Neural Networks (CNN) have been shown to be effective for image recognition tasks, and recent studies have demonstrated their potential for DHTR.[2] In this paper, we propose a CNN-based approach for DHTR that achieves great performance on a standard dataset. Neural networks serve as the foundation for deep learning methodologies, including techniques, which are a subset of machine learning. The depth (or quantity of layers) of a neural network is referred to as deep learning. Processing systems for data called neural networks look like the structures of the human brain. Neural Networks, which are analogous to the biological neural network, are built on information-processing units known as neurons. Inputs, weights linked with connection links, bias, and output make up a neural network's main building blocks. A sensor is the name given to each component in a neural network. [2] The proposed approach involves a preprocessing step to normalize and segment the input images, followed by a CNN architecture that consists of fully connected layers and several convolutional layers. A large dataset of labeled Devanagari characters is used to train the network. The proposed method has demonstrated exceptional results on a widely-used dataset that is commonly employed for evaluating image recognition systems, and can be applied to real-world applications such as document digitization and text-to-speech conversion. In

1 Associate Professor, Marathwada Mitra Mandal's Institute of Technology, Pune
ORCID ID : 0000-0003-2528-9202
2 Associate Professor, Marathwada Mitra Mandal's Institute of Technology, Pune
ORCID ID : 0000-0001-8691-8311
3 Professor, Dr. D.Y. Patil Vidyapeeth, Pune, Dr. D.Y. Patil School of Science & Technology, Tathwade, Pune
ORCID ID : 0000-0003-2407-8091
4 Associate Professor, Marathwada Mitra Mandal's Institute of Technology, Pune
ORCID ID : 0000-0002-4081-02655
5 Assistant Professor,Marathwada Mitra Mandal's Institute of Technology, Pune
ORCID ID : 0009-004-4788-156X
6 Marathwada Mitra Mandal's Institute of Technology, Pune
ORCID ID : 0009-0001-8014-4623
7 , Assistant Professor, Marathwada Mitra Mandal's Institute of Technology, Pune
ORCID ID : 0009-0002-5436-2958
** Corresponding Author Email: subhashrathod@gmail.com, salunkemangesh019@gmail.com*

this paper, the neural network approach for character recognition is discussed in detail along with their results. [3]

For the study of visual images, deep neural networks like the convolutional neural network (CNN) are employed. Every brain cell in that level is entirely coupled to every neuron in the level below, making it a multi-layered cognitive system that is totally interconnected. A set of neurons examines a section of the graphic referred to as a feature using a three-dimensional framework. It typically operates in three distinct phases. [3]

The convolution layer, which appears initially, scans every pixel of the picture to identify the characteristics before producing the resulting feature pattern. The array of "features" is dimensionally reduced while maintaining important data in the subsequent phase, referred to as pooling. In the final step, the matrix is flattened to a vector as well as sent into an entirely linked layer, which builds an object out of the characteristics and selects the category to whose library the graphic corresponds. The backbone of our infrastructure is a server. [3]

A Python programme can be run on the machine that serves as the server. The necessity for it arises from the fact that a phone with Android lacks the computing capacity necessary to operate neural networks along with carrying out visual processing functions. Additionally, our technology facilitates individuals with previous devices to use it by using infrastructure to carry out operationally taxing activities. In our framework, we employed the convolutional neural network approach. The NIST Dataset, which has been openly accessible and includes examples of handwriting from numerous authors, was utilized. A convolutional neural network is the type of neural network model that has been employed. Modern neural networks called CNNs are widely used in the field of visual analysis. Tensorflow, an open-source framework for artificial intelligence programmes, was employed for developing the model of the neural network. Several image processing tasks, including classification, thresholding, and topological processes, were carried out using OpenCV. An open-source framework employed in image processing is called OpenCV. [4]

2. Literature survey

1) In a recent study, Kharya et al. proposed a CNN-based approach for recognizing handwritten Devanagari characters. In this paper, they proposed a method to recognize the characters and it can be rearranged into right order. The proposed method achieved an accuracy of 96.12% on the Devanagari Handwritten Character Dataset (DHCD). [5]

2) Patil et al. developed a hybrid approach that combined CNN with “Long Short-Term Memory” (LSTM) networks for recognizing Devanagari handwritten text. In this paper, they used neural network for recognizing the multilingual text and handwritten digits into universal language i.e. English. The proposed method achieved an accuracy of 97.56% on the DHCD dataset. [6]

3) In another study, Singh et al. proposed a deep CNN architecture for recognizing Devanagari handwritten characters. In this paper, they used the deep learning algorithms to recognize the digits from 0 to 9 with great accuracy. The proposed method achieved an accuracy of 97.4% on the DHCD dataset. [7]

4) Sharma et al. proposed a deep learning model based on a combination of CNN and “Recurrent Neural Networks” (RNNs) for Devanagari handwritten text recognition. The proposed method achieved an accuracy of 97.67% on the DHCD dataset.[8]

5) A recent study by Kumar et al. proposed a CNN-based approach for recognizing Devanagari handwritten text. In this paper, they compare deep convolutional Neural Network and ResNet algorithm for Handwritten Character Recognition. The proposed method achieved an accuracy of 98.12% on the DHCD dataset.[9]

3. Proposed System

The goal of these study is to create a system capable of recognizing handwritten English characters by processing input images, extracting key features, training a neural network using the Back propagation technique or Scaled conjugate gradient method, classifying the input character, and generating digital output. The system is divided into two primary sections: training a Convolutional Neural Network (CNN) with an image database and testing the CNN with test images.

The training component of the system includes creating a dataset, pre-processing it, extracting features, generating a feature vectors and test vectors, training a CNN, and saving the trained CNN for testing purposes. The testing phase involves several steps, such as determining the number of words and characters in the input image, without the need to train the CNN again. The segmentation step is critical during testing to accurately identify the number of characters.

3.1. System architecture

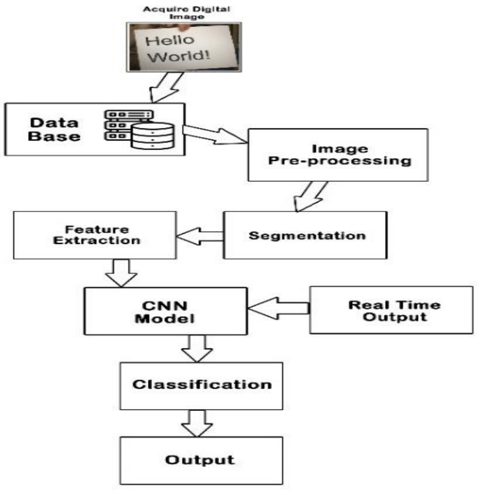

Fig. 1: System Architecture

The above figure 1 System architecture shows how the system will work. First we accept the image of handwritten Devanagari sentence or words. Then we store it in the database and we can use that image or data as a training data to further improve the accuracy and performance of the model. Next the size of image is scaled down to 28x28 pixels and then the image pre-processing is done. After image processing we segment the sentence into the words and words into the characters, note that the character with the kana-"matra" and with "velanties" are considered as one character. Then we extract the features using the 3x3 kernel at every time, the extracted features are also stored in the database to use it in future for training the model and at next step we create the neural network using the feature maps we got from feature extraction of image, after building the convolutional neural network, we give the real time input direct to the network and CNN predicts the characters and save the output in the multi-dimensional array, then finally that array is converted into the words again and displayed as an digital sentence.

Character Segmentation

Input : Segmented words from lines Output : Segmented characters from words

Step 1: Start

Step 2: Select Document Image.

Step 3: Repeat for x = 0 to Height of Segmented word.

Step 4: Repeat for y = 0 to Width of Segmented word.

Step 5: Scan Each pixel from the Vertical pixel column.

Step 6: Extract RGB value for each pixels inPixels[x][y]

Step 7: If pixel with no Black pixel is found then
{
Segment characters from words
}

Step 8: Stop

Fig. 2: Algorithm Used for Proposed system

3.2. Image pre-processing

During both the testing and training stages, the input image undergoes a series of operations for pre-processing to improve image quality and preparing image for segmenting. The primary aim of pre-processing step is to eliminate background noises, enhancing the region of interest in the image, and differentiate between the foreground and background more clearly. To achieve these objectives, the input image is subjected to noise filtering, binary conversion, and smoothing operations. An example of image pre-processing is illustrated in the figure below.[11][12][13]

Once pre-processing is complete, the input image undergoes a compressed representation as part of the process. Additionally, edge detection is carried out to identify the region of interest, while binary conversion guarantees a clear distinction between foreground and background. In the pre-processing stage, the edges are also dilated.[14][15][16]

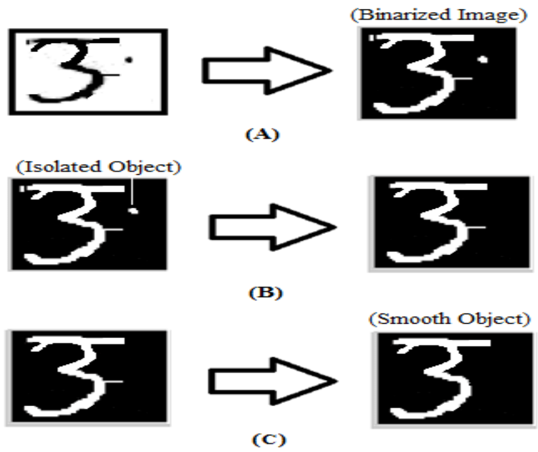

Fig. 3: Image processing

3.3. Segmentation

The entire image is disintegrated into a series of distinct text/sub-images using a method that relies on edge detection and identifying the gaps between characters. Following segmentation, each subdivided part is labeled and processed independently to determine the number of characters present in the input image. To ensure consistency, every sub-image segmented is resized to 70×50 and normalized relative to itself.

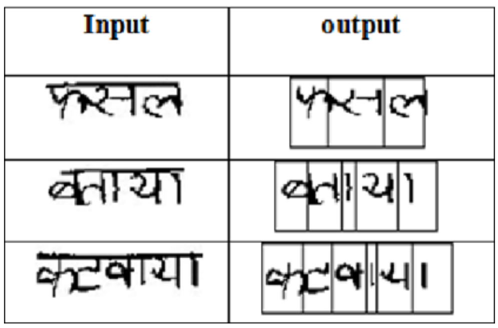

Fig. 4: Segmentation

Extracting high-quality features from the image is facilitated by this approach. In handwritten texts, identifying valid segmentation points is straightforward because minima or arcs locations between characters are used. These to ensure accurate segmentation in an incomplete image, the segmentation points undergo a thorough inspection process.

This process involves comparing all points to the average distance between two segmentation points. By doing so, the algorithm can detect any potential errors in point inclusion and adjust accordingly. This methodology will be further explained and demonstrated later in this study.[17][18][19]

3.4. Feature extraction

To obtain the feature set, the pre-processed image is transformed into a 7×5 bit-mapped version. Figure 4 depicts several instances of bit map versions of different characters developed in the proposed system. [20][21][22]

The bit-mapped version retains the main characteristics of the input image while taking up less space/data, resulting in reduced time during training of neural network without sacrificing the accuracy of correct input identification. Subsequently, a single array of size 35×1 is created from the bit-mapped images, which functions as the input vector for the CNN [23][24][25]

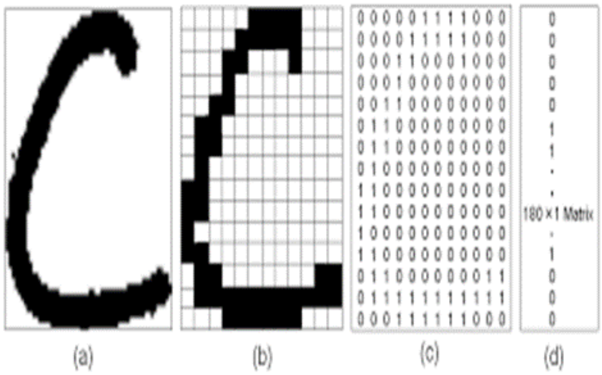

Fig. 5: Feature Extraction

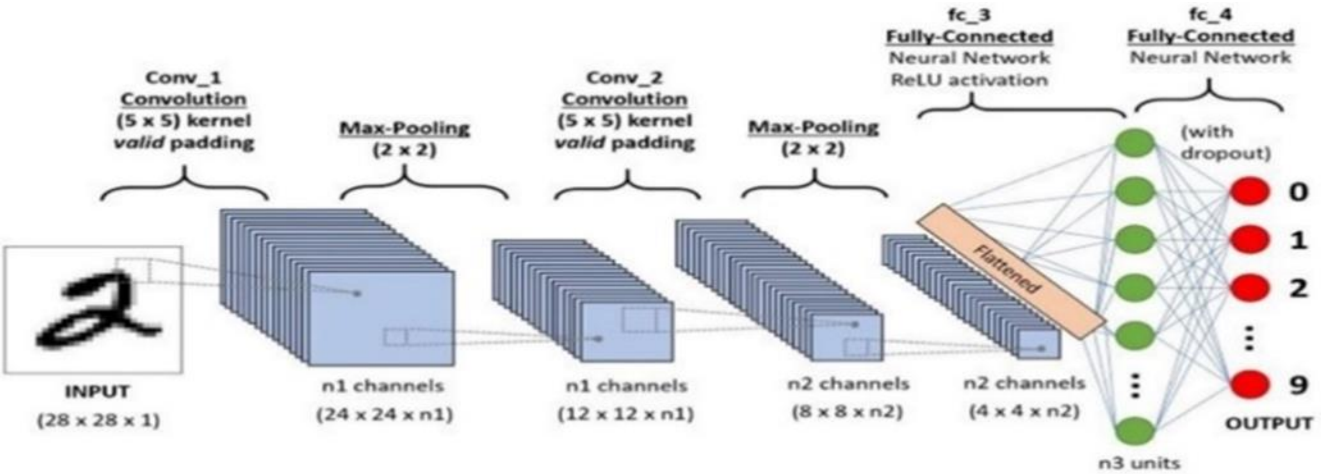

Fig. 6: CNN mode

3.5. CNN model

To train the CNN, it is necessary to define the number of hidden layers in the neural network and select the appropriate learning algorithm. Normalizing the input set and target set between the range of [-1, 1] aids in efficient training. The gradient is set to -10, and the maximum number of iterations is limited to 1000 during the training phase. A training dataset consisting of 55 samples of each character is created for training. Finally, to evaluate the effectiveness of the designed system, new test images are generated, and a sample handwritten document is presented in the figure 6.

3.6. Devnagari character dataset

The Brahmic family of scripts, which is used in Nepal, India, Tibet, and Southeast Asia, includes Devanagari, which is used to write Hindi, Marathi, Nepal like other similar languages. The Marathi variant of Devanagari script includes 12 vowel characters, 36 base consonant characters, 10 number characters, and some of the special characters. The vowel characters are depicted in Figure 1, while the consonant characters are illustrated in Figure 2, and the number characters show in Figure 3. Additionally, for each of the 36 consonant characters, 12 derived forms can be created by combining them with the vowels. So in total there are 12 x 36 i.e. 432 characters in Marathi.

अ	आ	इ	ई	उ	ऊ	ए	ऐ	ओ	औ	अं	अः

Fig. 7: Vowels

क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	ट	ठ
ड	ढ	ण	त	थ	द	ध	न	प	फ	ब	भ
म	य	र	ल	व	स	ष	श	ह	क्ष	त्र	ज्ञ

Fig. 8: Consonants

The HDC (Handwritten Devanagari Character) Dataset comprises in total 92,000 images, with 72,000 images in the consonant dataset and 20,000 images in the numeral dataset. Table

I shows the statistics for the Handwritten Devanagari consonant character dataset, the statistics for the Handwritten Devanagari numeral character dataset. [26][27]

4. Cnn Train and Test Model:

Choosing the learning technique and the hidden layer configuration are required for teaching CNN. To facilitate effective training, the input vector and the target vector are both normalised in the [-1 to 1] range. The gradient is set at -10 during training, and no more than 1,000 repetitions are allowed. The creation of the learning data requires fifty-five instances of every single character. For the purpose of verifying the accuracy of our planned system, additional test photos are currently being developed. An example of a handwritten manuscript is shown in Figure 9.

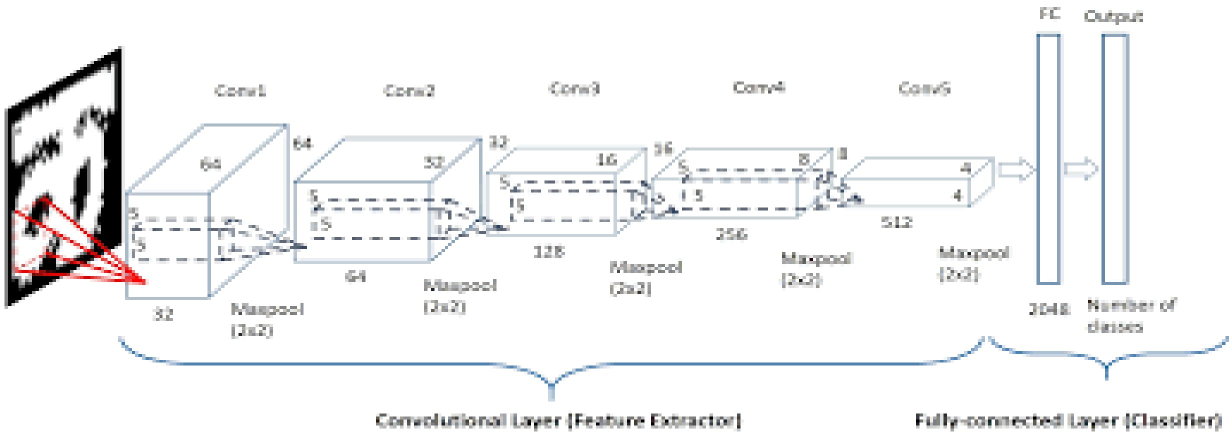

Fig. 9: Training and Testing CNN Model

4.1. Training Model

Once we have split the appropriate data set into training and testing sets, we will import the one from MNIST and use a portion of it. Initially, we want to create a framework for models utilising Python's deep learning packages and a combination of CNN, RNN, and any other relevant components. In order to provide the information for the learning framework, we must initially perform the necessary pre-processing procedures on the collection of data. Once the algorithm has been trained, we are going to employ it to generate numerous models by just modifying its variables, thereby giving us a larger pool of simulations to evaluate against.

4.2. Testing the Model:

The earlier-trained classifiers won't be put to the test using novel information. We will receive varying degrees of accuracy for several simulations, after which we are going to be able to choose the one that is the most correct. As soon as we have that idea, we can create and launch our HTR solution for those who use it.

5. Result and Discussion

The overall result emphasizes the potential applications of the proposed approach, such as document "digitization", "text-to-speech" conversion, and "language translation", which showcase the practical value of HTR systems based on CNNs. It acknowledges the need for further research to overcome the limitations and enhance the overall effectiveness and efficiency of the approach.

Such visualisations let us track how the framework changes as it learns, allowing us to identify issues and improve the precision of predictions. [20][21]

In illustrations, we frequently find both of these categories associated with learning curves: Optimising Learning Curves: Learning curves are based on the measure used to optimise the settings of the model, such as loss or Mean Squared error**.** Performance Learning Curves: These curves are based on the metrics that will be used to assess and choose the framework, such as "accuracy, precision, recall, or F1 score 48".

Accuracy besides Loss are the two most well-known and discussed metrics in machine learning.

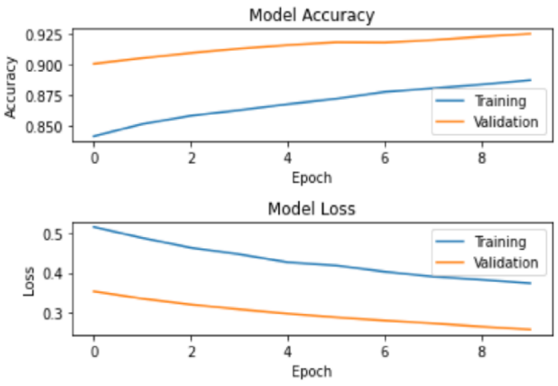

Fig 10: Accuracy Curve

From the above curve we can say that accuracy during training and validation has increased with increase in number of epochs and loss has been subsequently decreases during training and validation.

6. Conclusion

In conclusion, the proposed CNN-based approach for Devanagari Handwritten Text Recognition (DHTR) has several advantages over other DHTR methods, including high accuracy, automation, scalability, flexibility, and state-of-the-art performance. However, there are also some limitations to consider, such as dataset dependency, preprocessing errors, computational intensity, limited interpretability, and contextual constraints. Despite these limitations, the proposed approach has the potential to be a valuable tool for applications such as document digitization, text-to-speech conversion, and language translation. Further research is needed to address the limitations and improve the effectiveness and efficiency of the approach.

References

[1] K. Malhotra, Y. Kumar, and P. Bhattacherjee, "A CNN-based approach for Devanagari HCR," Journal of Advanced CS and App. vol. 8, no. 6, pp. 303-309, 2017.

[2] S. Mittal, S. Kumar, and S. Bhatia's, Paper "Devanagari script recognition using deep CNN," International Journal of Computer Applications, vol. 156, no. 4, pp. 28-34, 2017.

[3] N. Aggarwal, S. Kumar, and A. Jain, "A review on Devanagari Handwritten Character Recognition using Deep Learning," International Journal of Computer Science and Information Security, vol. 16, no. 3, pp. 80-88, 2018.

[4] A. Kumar and A. Kumar, "Devanagari Handwritten Character Recognition using Convolutional Neural Networks," in Proceedings of the International Conference on Computing, Communication and Automation, 2016, pp. 480-485.

[5] S. Phadikar and In P.Roy's article "A Complete Review of Handwritten Devanagari Character Recognition Methods," published in International Journal of Machine Learning and Computing, vol.3, no. 1, pp. 9-14, 2013.

[6] Joshi, D., Saini, P., & Sharma, D. (2021). A Comprehensive Study on Devanagari Handwritten Character Recognition. International Journal of Recent Technology and Engineering, 9(1), 5311-5316.

[7] Prasad, S., & Subramanian, K. (2017). Devanagari HCR Using CNN. International paper of Computer software and Applications, 176(4), 22-26.

[8] Kumar, S., & Gupta, S. K. (2019). DL based Handwritten Devanagari Character Recognition. International Journal of Computer software and applications, 182(17), 1-6.

[9] Singh, N., & Vyas, S. (2019). HDCR using CNN. Journal of Research in CSE and Information Technology, 5(1), 453-458.

[10] Bajaj, N., Singh, N., & Garg, R. (2021). Handwritten Devanagari Character Recognition using Convolutional Neural Network with Augmented Dataset. International Journal of Artificial Intelligence and Machine Learning, 13(1), 58-66.

[11] Kumar, P. (2021). A Proposed Methodology to Mitigate the Ransomware Attack. In Recent Trends in Intensive Computing (pp. 16-21). IOS Press.

[12] Salunke, M., Kabra, R., & Kumar, A. (2015). Layered architecture for DoS attack detection system by combined approach of Naive Bayes and Improved K-means Clustering Algorithm. International Research Journal of Engineering And Technology, 2(3), 372-377.

[13] Salunke, M. D., & Kabra, R. (2014). Denial-of-service attack detection. International Journal of Innovative Research in Advanced Engineering, 1(11), 16-20.

[14] Salunke, M. D., Kumbharkar, P. B., & Kumar, S. (2020). Proposed Methodology to Prevent a Ransomware Attack. International Journal of Recent Technology and Engineering (IJRTE), 9(1), 2723-2725.

[15] Salunke M. D., Dr. Kumbharkar P. B., Dr. Yogesh Kumar Sharma. (2020). A Proposed Methodology to Detect a Ransomware Attack using Deception Technique. International Journal of Advanced Science and Technology, 29(3), 10532 -. Retrieved from http://sersc.org/journals/index.php/IJAST/article/view/27134

[16] Telshinge, M. P. R., & Salunke, M. M. D. Un-Compromised Credibility: Social Media based Multi-Class Hate Speech Classification for Text: A Review.

[17] Deshpande, P., Joshi, P., Madekar, D., Pawar, P., & Salunke, M. (2020). A Survey On: Classification of Twitter data Using Sentiment Analysis. Asian Journal For Convergence In Technology (AJCT) ISSN -2350-1146, 5(3), 34-37. Retrieved from https://asianssr.org/index.php/ajct/article/view/911

[18] Rathod, S. ., Salunke, M. D. ., Yashwante , M. ., Bhende , M. ., Rangari, S. R. ., & Rewaskar, V. D. . (2023).

Ensuring Optimized Storage with Data Confidentiality and Privacy-Preserving for Secure Data Sharing Model Over Cloud. International Journal of Intelligent Systems and Applications in Engineering, 11(3), 35–44. Retrieved from https://www.ijisae.org/index.php/IJISAE/article/view/3140

[19] Rathod, S. G., Salunke, M. D., Jadhav, H. B., Ajalkar, D. A., Banurao, D., & Satre, D. B. Role Based Secure Data Access Control for Cost Optimized Cloud Storage Using Data Fragmentation While Maintaining Data Confidentiality. computing, 4(5), 38.

[20] Sonawane, P., Kardile, A., Lagad, K., Pati, V., & Salunke, M. D. (2021). A Survey on Machine Learning Approach for Classification of Breast Cancer. INTERNATIONAL JOURNAL, 5(12).

[21] Deshapnde, P., Joshi, P., Pawar, P., Madekar, D., & Salunke, M. D. (2019). A Survey On: Sentiment Analysis framework of Twitter data Using Classification. Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146, 5(3), 118-122.

[22] Rathod, S., Khobragade, R. N., Thakare, V. M., Walse, K. H., & Pawar, S. (2022, September). Model for Efficient Data Storage on Public Cloud. In 2022 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS) (pp. 1-5). IEEE.

[23] S. Rathod, R. N. Khobragade, V. M. Thakare, K. H. Walse and S. Pawar, "Lightweight Auditable Secure Cloud Storage With Privacy Enabled Data Storage Optimization," 2022 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS), Pune, India, 2022, pp. 1-6, doi: 10.1109/ICBDS53701.2022.9935980.

[24] Rathod, S., Bagal, P., Gujar, R., Hannure, A., & Waghulde, D. (2021). Smart Helmet for Detection of Unsafe Events in Mining Industy Based on IoT (No. 5799). EasyChair.

[25] Pawar, M. S., & Rathod, S. (2020). Text summarization using cosine similarity and clustering approach. Int J Curr Eng Technol, 8, 669-673.

[26] Rathod, S., Khobragade, R. N., Thakare, V. M., Walse, K. H., & Pawar, S. (2022, September). Model for Efficient Data Storage on Public Cloud. In 2022 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS) (pp. 1-5). IEEE.

[27] Rathod, S. G. (2022). Security for Shared Data Over Public Cloud for Maintaining Privacy. Mathematical Statistician and Engineering Applications, 71(4), 7167-7173.

[28] Rana, P. ., Sharma, V. ., & Kumar Gupta, P. . (2023). Lung Disease Classification using Dense Alex Net Framework with Contrast Normalisation and Five-Fold Geometric Transformation. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2), 94–105. https://doi.org/10.17762/ijritcc.v11i2.6133

[29] María, K., Järvinen, M., Dijk, A. van, Huber, K., & Weber, S. Machine Learning Approaches for Curriculum Design in Engineering Education. Kuwait Journal of Machine Learning, 1(1). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/111

[30] Mehraj, H., Jayadevappa, D., Haleem, S. L. A., Parveen, R., Madduri, A., Ayyagari, M. R., & Dhabliya, D. (2021). Protection motivation theory using multi-factor authentication for providing security over social networking sites. Pattern Recognition Letters, 152, 218-224. doi:10.1016/j.patrec.2021.10.002