**Handwritten Devanagari Character Recognition using Convolutional Neural Network**

1Jayshree R Pansare, 2Vinit Shingavi, 3Saniraj Desai, 4Dnyaneshwar Algule, 5Aditya Pawar

Modern Education Society’s Wadia College of Engineering, Pune

***Abstract* *-*** Handwritten Devanagari character recognition (HDCR) poses a significant challenge in the field of pattern recognition due to the inherent complexity and variability of handwriting. In this research work, we present a new method for accurate recognition and classification of Devanagari symbols using Convolutional Neural Network (CNN). Using the power of TensorFlow and Keras and various Python libraries, we have developed a powerful CNN specifically designed to capture features of Devanagari text. Technological increase and change in education to make the model resilient to changes in writing style and environmental conditions. We carefully evaluate the effectiveness and efficiency of our proposed HDCR by testing a large number of files containing Devanagari characters. Our results show a significant improvement in recognition accuracy compared to existing methods, highlighting the effectiveness of our approach. Complex interaction between this researches contributes to expanding knowledge about optical behaviour, paving the way for the development of data digitization, natural language processing, and cultural preservation with Devanagari scripts*.*

***Keywords* -** Handwritten Devanagari character recognition, pattern recognition, Convolutional Neural Network (CNN), natural language processing Devanagari script.

1. **Introduction**

Text recognition is a development that has received great attention over the years, especially with the advent of computers. It is necessary to ensure that the machine understands and recognizes characters in different languages ​​and then converts them into computer format. Among the many algorithms developed, deep convolutional networks, which are smart in pre-processing images and provide accurate details, have emerged as a good solution. The difference between online and offline recognition depends on the type of feedback. Online systems track signals over time, while offline systems analyse historical data. This article discusses offline character recognition, focusing on Devanagari, a language known for its spelling and diversity. By carefully collecting and processing preliminary data on Devanagari scripts, this study aims to understand how to solve the problems caused by different alphabet types and finally how to make progress in cognitive behaviour.

Chui En Mook et.al. [1] It conducts a comparative study of six known CNN architectures (VGG16, Numbers and MNIST). While previous studies generally focus on specific data and do not provide a comprehensive comparison of CNN architectures, this study makes a difference by evaluating the performance of the architectures. Experimental results show that the superiority of the InceptionResNetV2 model in data enhancement achieves the highest accuracy among all datasets (93.26%, 97.16%, and 99.71% for English alphabet, and MNIST, respectively). ) is good for the write behaviour and mathematical information performance of different CNN architectures.

Ananya Shukla et.al. [2] Presents an optical character recognition (OCR) system developed to solve problems arising from the complexity and unique characteristics of Devanagari texts. With more than 300 million Hindi speakers in India, the need for Devanagari OCR is huge. This process includes various steps such as data pre-processing, grayscale conversion, binarization, normalization, model training, and mapping analysis. Devanagari's special pronunciation and distinct capital letters make it a challenge that requires careful algorithm design and CNN models.

Prashant S. Kolhe et.al. [3] The traditional use of handwriting in architectural and historical documents causes difficulties in training and achieving accuracy. Recent advances in transcription research have turned to deep learning, which uses convolutional neural networks (CNN) to solve the complexity of automatically retrieving designed unique text. However, with the rapid increase in data collection and computational resources, more research is urgently needed to improve the findings.

Sonia Juneja et.al. [4] Human intelligence is characterized by the ability to learn and explore, which allows people to make quick decisions. The purpose of machine learning is to give machines intelligence and improve their decision-making abilities. Knowing the behaviour of the hand (including numbers and letters) demonstrates the machine's ability to allocate resources. This investigation aims to recognize people's signatures and convert them into digital forms to facilitate banking and healthcare services. Recent research has used a variety of methods, classifications, and procedures to improve knowledge of numbers and letters.

upali Patil et.al. [5] Recognition of Devanagari characters poses a challenge due to the many different characters in Devanagari grammar, especially Barakhadi, which includes Kana, Matra, Ukar, Velanti and Anusvar. To solve this problem, a new method is proposed that combines 454 different detections of Devanagari characters with the closest translation using the k-nearest neighbour (KNN) model to complete sentences. Unlike the traditional method that focuses on 58 characters (Vyanjans), this method tries to complete the 454 recognized characters, thus increasing the sentence recognition accuracy by 86.84%. Additionally, the system achieved 89.52% classification accuracy for 454 characters and a processing speed of 1,464 seconds per word.

Pallavi Patil et.al. [6] Effective classification is important for the good performance of the OCR system because errors in this process can affect the recognition rate. Splitting modifiers and combining symbols pose special challenges because of modifiers and half symbols. In this study, a new method that uses predicted contours and internal character features for accurate segmentation is proposed. The system has been shown to be accurate in both basic and separate/combined segmentation, with segmentation accuracy reaching 91.84% to 99.11%. This new approach is expected to improve the overall performance of OCR systems by providing accurate character separation.

M.Geetha et.al. [7] The degree of blindness in English letters and numbers can vary greatly and vary from person to person. These conditions make it difficult to identify and recognize English letters and numbers. A deep learning-based grammar that can recognize characters and numbers from input images containing English characters and numbers attempts to solve these problems. The proposed model uses Long Short Term Memory (LSTM) method along with Convolutional Neural Network (CNN) for recognition and detection.

Pooja Pathak et.al. [8] In Deep learning, convolutional neural network is a type of deep neural network widely used in image analysis. The concept is called 'Devanagari'. The computer will verify which letter was processed. To accomplish this task, we will train our machine to recognize all Indian letters by teaching it all Devanagri letters. Computers don't just recognize characters, it will also record the English pronunciation of the character on the screen to show the user how to say it.

Ahmed Ayman Mokhtar et.al. [9] A machine learning method is proposed to overcome the problems caused by the imperfections and diversity of written numbers and symbols. Use Kaggle's A to Z handwritten alphabet dataset and multiple MNIST datasets to train the model using convolutional neural network (CNN) technology. Users can draw symbols using a graphical user interface (GUI) and quickly receive numerical representation with % accuracy. After 10 times, the system achieved 98.80% testing accuracy and 99.06% training accuracy using Keras and RMSprop optimizer-based CNN model. The performance of this method is demonstrated by the average macro precision of 0.99, the average macro recall of 0.98, and the average macro F1 score of 0.99.

Sapna Katoch et.al. [10] Literacy is challenging in computer vision and graphics, as well as in applications ranging from document recognition to bank reading. Optical character recognition (OCR) makes it easy to convert written data into different formats to identify different documents. The main purpose of this research is to propose typing techniques such as touch input and image files from mobile devices. This study evaluated the ability of a convolutional neural network (CNN) to recognize characters in an image dataset, Researchers are pushing the boundaries of typing using extensive data and advances in deep learning and machine learning algorithms.

**2. Various Handwritten Devanagari Character Recognition Techniques**

Text Recognition uses a variety of techniques, including Support Vector Machines (SVM), K-Nearest Neighbours (KNN), and Convolutional Neural Networks (CNN), each of which provides a unique way to judge the difficulty of Devanagari scripts. SVM is a supervised learning algorithm that aims to find the optimal hyper plane that best separates data points of different qualities by using kernel functions to determine the input at a higher location for a good distribution. KNN, on the other hand, is a non-parametric lazy learning algorithm that classifies new events by comparing them to its K neighbours at a given location, providing a list of the most common nearby events. Finally, CNNs are revolutionizing character recognition tasks by extracting hierarchical features from input images, allowing them to identify complex patterns and patterns in Devanagari characters through a combination of techniques and hashes. All of these technologies play an important role in creating a robust and powerful Devanagari alphabet, facilitating applications in data analysis, text conversion reading and natural language rendering.

There is a lot of research on handwriting recognition (HTR) to improve accuracy using deep learning algorithms. This presents a method that uses deep learning (specifically the LSTM model) to improve HTR accuracy by recognizing a word instead of a word. The integration of this method into OCR systems was investigated and the results showed the superiority of the 2DLSTM-based method [11]. Optical character recognition (OCR) plays an important role in pattern recognition by converting text or image into digital format. While many studies have focused on knowledge of English, research on Indian contexts, especially Sanskrit scripts, is limited. To resolve these differences, this presents a comparative study of four classification and two elimination methods [12]. Especially in scripts such as Devanagari, it faces serious problems due to features such as writing style, diversity of script. While it is easy to recognize Roman characters, fully recognizing Devanagari characters requires technology. This presents the development of optical character recognition (OCR) based on convolutional neural networks (CNN) specifically designed for Devanagari text to improve accuracy. Unlike traditional OCR methods, which often suffer from accuracy issues, CNNs show promise using deep learning [13].

The way to recognize Devanagari text, which eliminates the problems caused by complex text. By using convolutional neural networks (CNN) as feature extractors and various classifiers (such as SVM, KNN, RF, DT, MLP, and XGB) for classification, the system achieves high-precision character recognition. Unlike traditional methods based on artificial intelligence, CNNs automatically extract relevant features from images, speeding up the process and improving accuracy [14]. This system developed to recognize Devanagari characters that are important in more than 100 languages ​​spoken in India and Nepal. Consisting of 47 capital letters, 14 vowels, 33 numerals and 10 numerals, Devanagari forms the basis of different languages ​​such as Hindi, Marathi, Sanskrit and Maithili. The data set used by this system consists of 34 characters, 29 numbers and 1 variable, and 34,604 images. Deep learning techniques, specifically deep convolutional neural networks (DCNN), are used for extraction and classification. DCNN architecture consists of a continuous layer that effectively removes higher levels from the image. The training model achieved an accuracy rate of 99.65%, demonstrating the effectiveness of deep learning in Devanagari character recognition [15].

The way to recognize Devanagari characters using transliteration. Transfer learning combines the power of deep convolutional neural networks (Deep CNN) for extraction and support vector machine (SVM) for classification. Adaptive learning improves classification accuracy by using pre-trained models and leveraging new tasks. This study focuses on Devanagari DHCD and DHS datasets containing written characters and sounds. Unlike English, Sanskrit follows the same phonetic rules, making it suitable for this study. The plan is to complete the accuracy of this information. The paper also includes a literature review, a summary of the dataset, a detailed description, experimental results, and conclusions [16]. This system addresses the difficulty of verifying Devanagari text, which is very complex and has different spellings. The specificity of writing emphasizes the importance of verifying authenticity, making it useful in forgery. Mixed signals are particularly challenging due to different segmentation and segmentation complexities. In this study, an analysis method that uses machine learning algorithms such as K-nearest neighbour (K-NN) and support vector machine (SVM) is introduced. Key features such as distribution, eccentricity, equal diameter, and angle are extracted from the image to aid identification [17].

OCR technology facilitates image-to-text conversion through various stages: image capture via mobile camera, pre-processing to enhance readability, segmentation into lines and characters, feature extraction for character recognition based on attributes, and post-processing to store the text in a user-accessible format like .txt [18]. In today's AI-driven era, many data and algorithm advances have supported the training of deep neural networks. The computing power of online education has been greatly improved through GPUs and cloud platforms such as Google Cloud and Amazon Web Services. Our system uses image segmentation for handwriting recognition, OpenCV for image processing, and TensorFlow for neural network training, all developed using Python [19]. This suggests a new approach to characterizing information using machine learning, especially in neural networks (CNN). By training the model on big data, the scheme achieved an accuracy of 98.6%, demonstrating the effectiveness of machine learning in this field. Through applications in data analysis, optical behaviour recognition, and other research, this research will help create automatic writing recognition systems and increase the benefits and accuracy of writing articles [20]. The experiment focuses on using deep neural networks, specifically AlexNet, to recognize Devanagari characters. With a dataset containing approximately 16,870 samples, transformations are used to leverage the AlexNet model before getting good results. Knowing the behaviour of the hand causes problems due to differences in patterns, stroke decisions, and pen types, but deep learning provides good solutions, especially when testing data is limited. Transformational learning increases learning efficiency and ensures accurate classification even with fewer data samples [21].

Recognition of Devanagari text using neural networks (CNN). Thanks to deep learning, feature extraction is automated, reducing the need for operator involvement. The CNN-based technology can recognize Devanagari letters with 91.23% accuracy and numbers with 100% accuracy, proving its ability to perform complex information tasks. The architecture process includes prioritization, elimination, and classification stages that use deep learning to solve problems such as character differences in Devanagari alphabets and complex images [22]. Develop a system to recognize written words in Marathi, a language widely spoken in the Indian state of Maharashtra. The system uses deep learning techniques, specifically convolutional neural network (CNN), to improve its ability to recognize handwritten text in urban data. Unlike previous systems that relied on segmentation, this system provides better accuracy by using a segmentation-free approach. By training the CNN model with various methods on the developed data, the system achieved 94% accuracy in recognizing Marathi text, it was found to be good and reliable in solving the challenge of digitizing urban data [23]. The spelling is difficult to recognize due to the variety of spellings. This work investigates numerical techniques such as support vector machines and multilayer neural networks, focusing on error reduction. The fact that the error is only 0.32% when the classification method is used shows the effectiveness of the method. The aim of the research is to lay the foundation for the development of coding algorithms that will facilitate professional and daily use [24].

Handwritten Character Recognition (HCR) is necessary to convert handwriting into text. This review examines various HCR methods, including their advantages, limitations, and accuracy. Writing can be digitized using technologies such as machine learning and optical character recognition (OCR) for ease of processing and access. The process includes stages such as pre-processing, segmentation, extraction, classification and authentication, and the process is divided into offline authentication and online authentication. This review aims to compare various studies to evaluate accuracy, effectiveness, and other factors that will facilitate the advancement of HCR technology [25].

**3. Methodology**

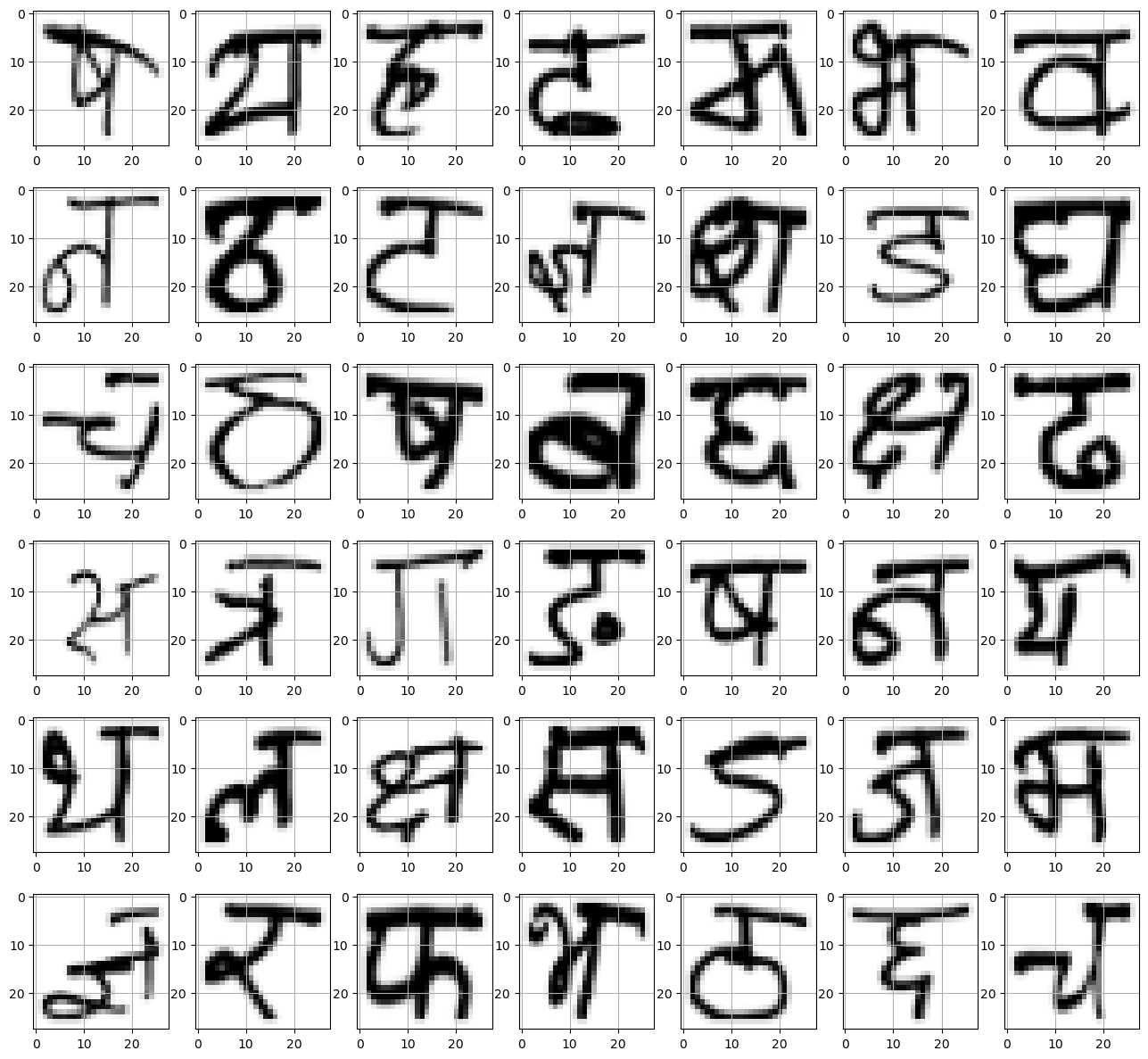
This approach includes data preparation, modelling, training, evaluation and judgment. Initially, data is prepared by organizing Devanagari images into training and test texts. Keras is used to build a CNN model that includes convolutional processes, batch normalization layers, and maximum pooling layers to remove features from input images. The model is trained using training data while monitoring performance metrics such as accuracy and loss. Evaluate the training model on a separate testing dataset to assess its generalizability. Finally, the model used for thinking, processing and classifying invisible images for the purpose of recognizing Devanagari characters. Through this approach, create a useful and effective Devanagari character recognition system that facilitates many applications in data analysis and text transformation.

**A. Data Collection**

A collection of Devanagari character datasets containing approximately 9200 Devanagari character images. Images are divided into 46 categories including Devanagari letters and numbers. The data is divided into a training set (consisting of 85%) and a testing set (consisting of 15%). All images in the file are saved in PNG format with a resolution of 32x32 pixels. To facilitate training the model, the data was generated through a custom catalogue method by splitting the images in their catalogue into training and testing subsets.

**B. Pre-processing**

The Date pre-processing phase ensures that the image is in the correct format for input into the convolutional neural network (CNN). First, if the image does not fit the size required by the CNN model, it is converted to the specified format (32x32 pixels in this case). Additionally, data enhancement techniques were used to increase the diversity of the data set. These techniques include rotating, transforming, retouching, cropping, and scaling and are designed to create changes to the original image. By increasing the data set, the neural network can be expanded to more data, thus improving its overall ability and achieving a higher testing performance. In general, the first step plays an important role in preparing training materials to ensure that the CNN model receives the necessary and diverse information for effective learning.



**Fig.1** Data Pre-processing

**C. Convolutional Neural Network Model**

The proposed convolutional neural network (CNN) architecture for typing Devanagari character is specifically designed to obtain the principal value of the input image and classify it into one of 46 different characters. The design has been carefully designed to solve problems with character recognition such as changes in handwriting, different line thicknesses and different characters.

i) Feature Extraction Layers:

CNN architecture has various layers to extract features from the input image:

* Layer 1: The first layer in CNN is the convolutional layer, which uses 32 large filters (3, 3) for the input image. Perform a convolution function on input images to capture local patterns and features. Normalization, which helps stabilize and accelerate the educational process.
* Layer 2: Similarly to the first layer, this layer uses another layer of 32 filters with the same configuration to improve feature extraction. The model can capture many patterns and details in the input image.
* Layer 3: This layer provides 64 filters for feature removal, allowing the model to recognize a higher level of the input image. Written characters have different possibilities and nuances.
* Layer 4: The last convolutional layer in the design uses 64 filters to remove the most distinctive features in the input image.

ii) Fully connected process:

After removal the flattened working map will be separated from the fully connected process:

* Layer 1: (Thick layer) this layer contains 128 neurons with ReLU activation function. Personality category.
* Layer 2: (Dense layer) A second dense layer with 64 neurons and ReLU optimization to improve learning agents. It can be flexible and difficult.
* Layer 3 (Output Layer) The last layer of the CNN architecture has 46 neurons, each neuron represents a group of characters in the Sanskrit alphabet. The classification has more than 46 classes showing the probability of each character class.

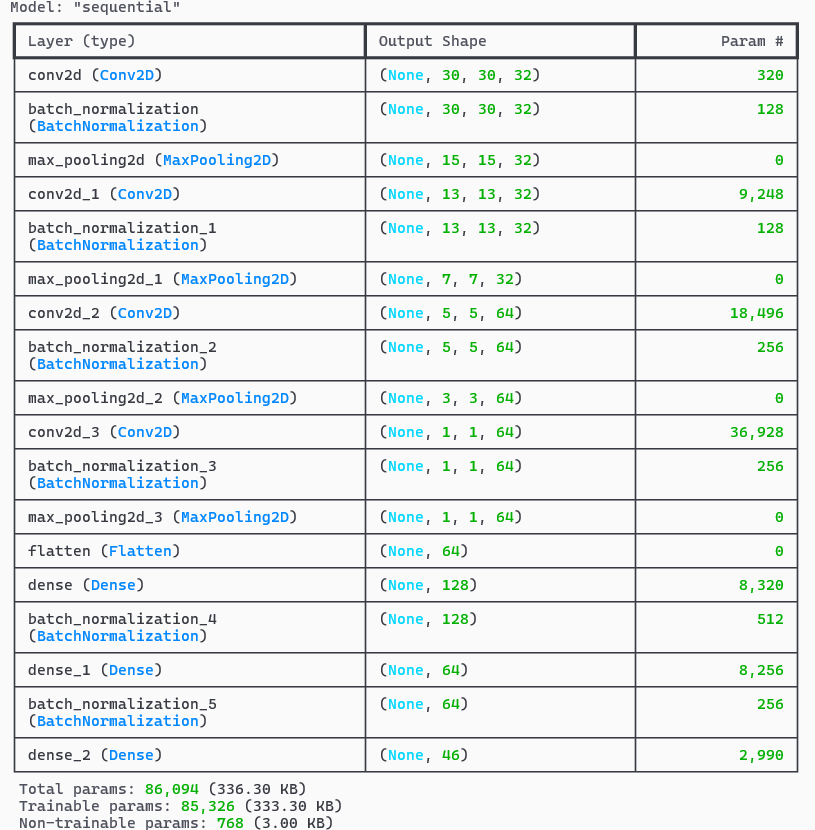
iii) Model compilation:

The model is compiled using the Adam optimizer, which effectively modifies the model parameters in training to reduce the categorical cross-entropy loss function. Difference between distribution and actual class distribution.

**D. CNN Model Training**

To train a convolutional neural network (CNN) model for writing Devanagari characters, the process usually follows several important steps. First, the data set is divided into training set and testing set, where approximately 85% of the data is reserved for training purposes and the remaining data is used to evaluate the effectiveness of the model. For "period". Each time there is a complete back and forth pass through all the training data in the network. The model learns from training data by adjusting its parameters to minimize the loss, of the prior function, which is usually measured as the difference between the predicted and the actual. A few steps from the model. First, analyze the unemployment rate, cost of education, and improvement during the education period, which play an important role in determining the direction of the new evaluation model. Then write the model, show the fault, the optimizer, and other parameters to monitor during training. While batch size refers to the number of samples processed in a forward/backward transfer, it also determines the number of processing on the entire data set. Exposure and loss, and evaluating the model's performance on valid data. Use the optimization method to change the parameters of the model based on the calculated slope from unemployment. The training process continues until preliminary stopping criteria are met, such as reaching the required level or completing the specified time. To evaluate its performance. The output shows the accuracy of the model after training 25 times, this shows that the model reaches 98.96% of the test data.

The configuration of a Convolutional Neural Network (CNN) model that is learning recognizing handwritten Devanagari characters. The model, sort of comprises convolutional layers followed by incredibly important batch normalization and max-pooling layers to extract and enhance features from the much input images. The final layers, which consist of fully connected dense layers with batch normalization; culminating in interesting output layers for classification with a total of approximately actually confusing 86,094 parameters, the model visibly demonstrates a robust capacity for accidentally learning and surprisingly distinguishing various Devanagari characters. Through intentionally training, it overly aims to accurately and inaccurately classify handwritten characters, contributing to the advancement of Devanagari character recognition and somehow technology.



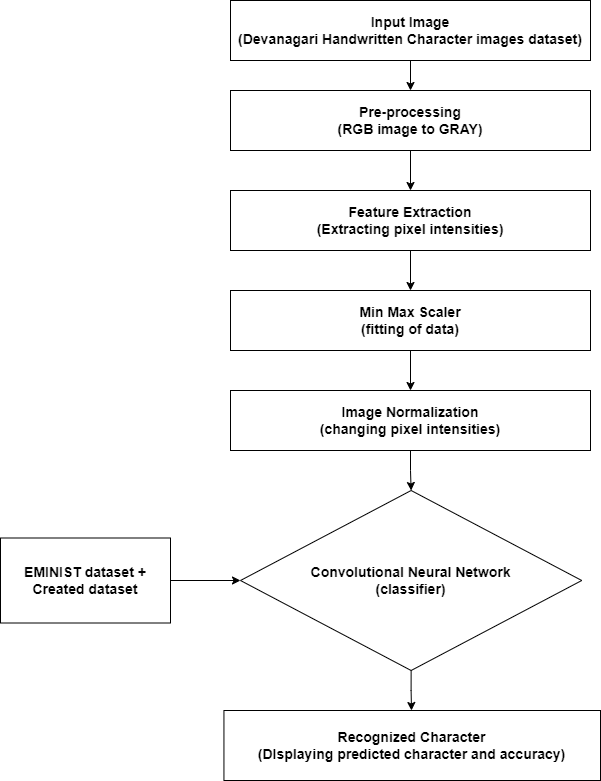
**Fig.2** CNN Model Architecture

**E. Model Evaluation and Saving**

After training a convolutional neural network (CNN) model to type Devanagari characters, the next important step involves evaluating the model's performance and saving it for future use. To evaluate a model, test data is used to evaluate its accuracy and generalizability. This evaluation often includes metrics such as accuracy and loss of test data, which can provide insight into how the model performs on invisible components. Additionally, visualization techniques such as training plans and verification/validation curves can help determine the effectiveness of the learning model and identify potential problems such as over fitting or under fitting. It preserves the saved format architecture, weight and layout. In the implementation, the Model Checkpoint callback is used to save the model according to certain conditions during training, such as achieving maximum validation accuracy. This ensures that the best results of the sample are saved for later use. Additionally, the model is saved using the Keras “save\_model” function, which saves the model, weights, parameters, and other settings in a format (such as HDF5) that can be easily loaded eight times and used for work without the need for training. Overall, accurate evaluation and registration of the training model is an important step in machine learning to ensure efficient and effective deployment in real-world applications.

**3.1 System Architecture**

The system architecture of Devanagari character writing starts from the input image, and the input image is pre-processed first. This preliminary step involves converting RGB images to grayscale images, since colour data is not required for character recognition and grayscale images are easy to calculate. Next, feature extraction is performed on the pre-processed image to extract the pixel density of the image. This step captures the basic features of written characters that are important for recognition. This parameter ensures that the pixel density of image data remains within a certain range, usually between 0 and 1. Scaling helps standardize the data and improve its processing by the next layer of the neural network. Convolutional Neural Network (CNN) classifier. CNN architecture is specifically designed for image classification tasks and is suitable for identifying patterns in image data. CNN consists of several layers followed by artefact removal and spatial down sampling layers. A batch normalization layer is placed after each layer to improve the stability and performance of the network. These layers use functions to make predictions based on extracted features. In this case, the softmax function is used for multi-class classification and one result is given for each character class. The feature with the highest probability is considered the accepted feature. Each step plays an important role in converting the input image into recognizable characters and shows the correct way to write Devanagari characters.

****

**Fig 3.** System Architecture of HDCR

**3.2 Algorithm**

The algorithm implements the Convolutional Neural Network (CNN) model for handwritten Devanagari character recognition using TensorFlow and Keras. It begins by preparing training and validation data using ImageDataGenerator, which facilitates data augmentation and rescaling. Too much, the CNN model architecture is then defined, consisting of multiple convolutional layers followed by batch normalization and max-pooling layers. The model is compiled with the Adam optimizer and categorical cross-entropy loss function. Training is performed using the fit() function, monitored by a ModelCheckpoint callback to save the best-performing model based on validation accuracy. After training the model's performance is visualized using matplotlib by plotting accuracy and loss over epochs. Finally, the trained model is used to predict the labels of test images, demonstrating its efficacy in recognizing handwritten Devanagari characters.

Here's the stepwise algorithm for the provided code:

Step 1: Data Preparation

* Import the necessary and must-required libraries.
* Define image data generators for both training and testing data sets.
* Specify augmentation parameters such as rotation, width shifting, and shifting in height.
* Load training data and the validation data by using the flow\_from\_directory method.

Step 2: Model Definition

* Create a Sequential, step-by-step model.
* Add some convolutional layers with ReLU activation along with BatchNormalization.
* Add some maximum-pooling layers to downscale the maps of features.
* Flatten the result and add some dense layers for classification of outcomes.
* Compile the model using the highly performing Adam optimizer plus categorical cross-entropy loss function.

Step 3: Model Training

* Define a ModelCheckpoint callback that can store or save the best model based on the validation accuracy obtained.
* Train the model with the training data and validate its performance using the validation data provided for evaluation purposes.
* Specify the number of epochs that need to be covered while training the model.

Step 4: Visualization

* Plot the graph to show the progression of both accuracy and loss during epochs for training and validation purposes.

Step 5: Prediction

* Load a random test image into the system.
* Preprocess the image, ensuring it is rescaled and converted to grayscale, thus enhancing the accuracy of predictions.
* Expand some dimensions to align with the input shape of the model.
* Load the model that has been previously trained.
* Utilize the model to determine or predict the exact label for the test image supplied.
* Calculate the percentage accuracy of the prediction process.

Step 6: Model Output

* Print out the predicted letter in addition to its corresponding accuracy percentage.

Step 7: End

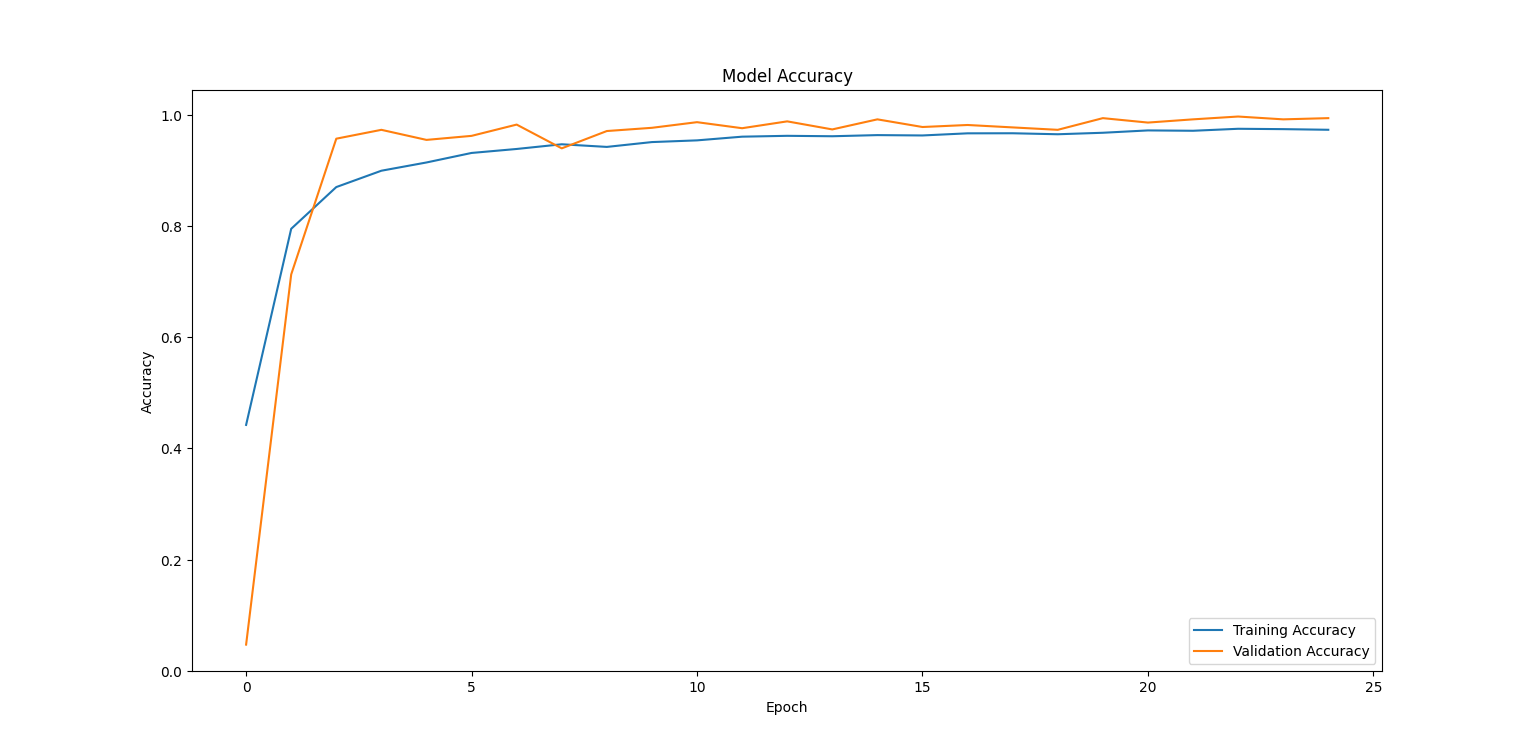
**4. Experimental Result**

Experimental results obtained by training a convolutional neural network (CNN) model for the classification of Devanagari characters are promising and demonstrate the effectiveness of the model. The model was trained on a file containing images of handwritten Devanagari characters, totalling 46 groups representing various characters and numbers. Data optimization techniques such as rotation, transformation, rearrangement, and translation are used during training to increase the power of the model and improve its general capabilities. There is a pooling layer and then a thick layer for distribution. Each convolution layer is followed by batch normalization to stabilize and speed up the training process. Additionally, the model uses the softmax activation function in the output layer to estimate the probability of each cluster. It is 98.97%. Among them, characters such as "ka", "ma", "ra" and "la" have the highest accuracy of over 99%. However, some characters, especially ‘ठ’ and ‘ड’, are somewhat obscure due to their differences and similarities with other characters. The model's performance was further evaluated on set, and the average accuracy was approximately 97.88%. More importantly, the model showed robustness in classifying Indian alphabets even

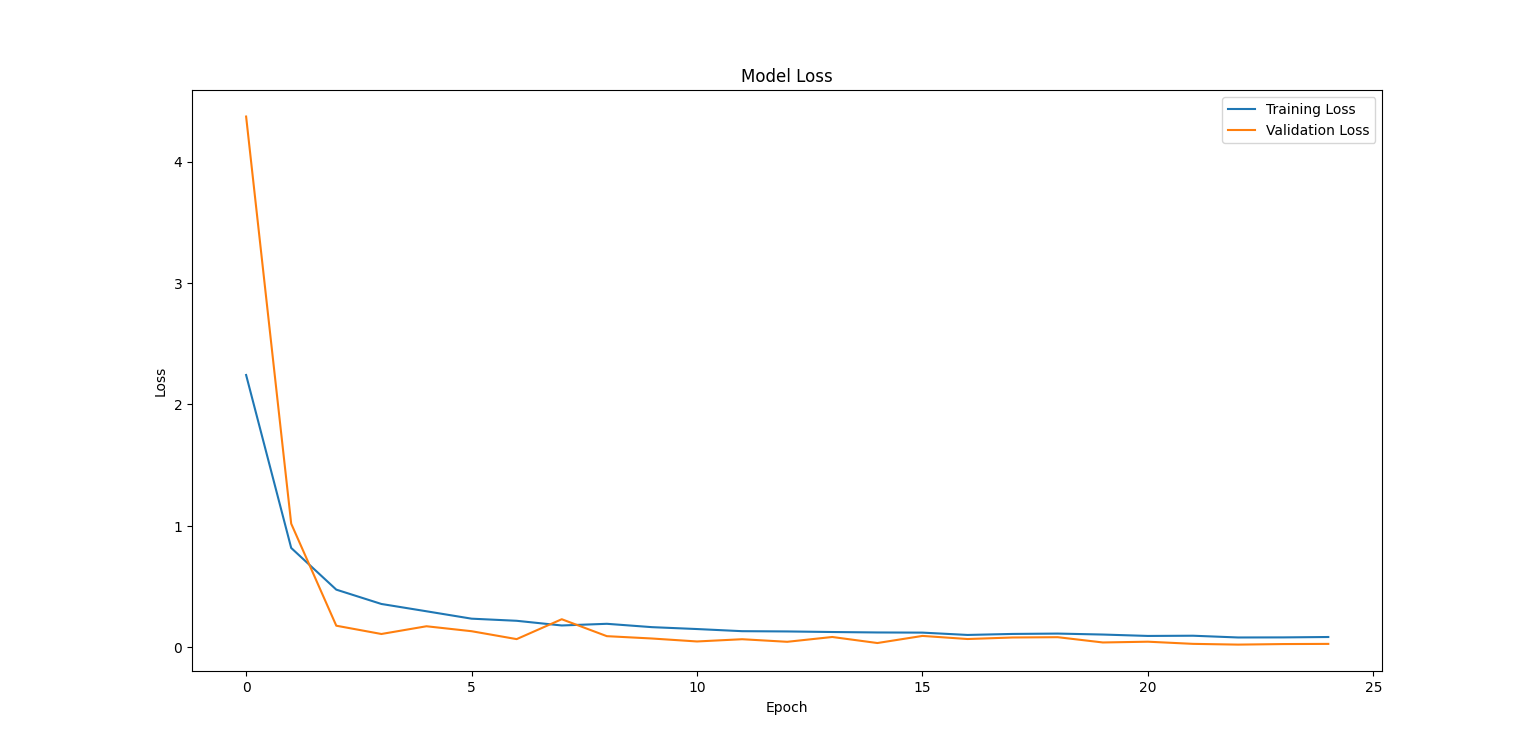
When accurate for numbers.

|  |  |  |
| --- | --- | --- |
| **Characters** | **Accuracy in %** | **Times in msec** |
| क | 99.98 | 226 |
| ख | 99.57 | 219 |
| ग | 99.71 | 356 |
| घ | 99.84 | 201 |
| ङ | 99.94 | 224 |
| च | 99.65 | 233 |
| छ | 99.90 | 240 |
| ज | 99.96 | 309 |
| झ | 99.87 | 254 |
| ञ | 99.93 | 250 |
| ट | 99.94 | 230 |
| ठ | 98.02 | 232 |
| ड | 82.67 | 393 |
| ढ | 99.56 | 213 |
| ण | 99.09 | 250 |
| त | 99.85 | 239 |
| थ | 99.80 | 245 |
| द | 98.55 | 231 |
| ध | 99.85 | 223 |
| न | 99.19 | 319 |
| प | 99.22 | 242 |
| फ | 99.65 | 210 |
| ब | 97.58 | 233 |
| भ | 99.95 | 252 |
| म | 99.99 | 225 |
| य | 99.91 | 247 |
| र | 99.95 | 254 |
| ल | 99.98 | 250 |
| व | 99.88 | 256 |
| श | 99.83 | 225 |
| ष | 98.55 | 228 |
| स | 99.97 | 226 |
| ह | 99.49 | 285 |
| क्ष | 99.02 | 225 |
| त्र | 93.40 | 258 |
| ज्ञ | 99.63 | 200 |
| ० | 93.04 | 233 |
| १ | 99.38 | 210 |
| २ | 99.98 | 279 |
| ३ | 99.97 | 207 |
| ४ | 99.58 | 263 |
| ५ | 99.95 | 230 |
| ६ | 99.93 | 242 |
| ७ | 99.96 | 234 |
| ८ | 99.96 | 222 |
| ९ | 99.97 | 234 |
| **MEAN** | 98.96934 | 244.71739 |

**Fig 4.** Accuracy Measure Table



**Fig .3** Model Accuracy vs. epoch



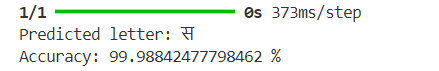
**Fig .4** Model Loss vs. epoch

The chart shows the machine learning model's performance measurement over time during training and validation. In the first sub-figure, which represents accuracy over time, the accuracy of the training and testing data model can be seen. In general, as the number increases, the model learns to generalize its predictions better, thus increasing accuracy. However, training and validation need to be monitored to ensure that the model does not match the training material. Over fitting occurs when a model learns training material well and fails on unseen material, this results in high training but low accuracy. The figure shows the loss versus time, loss model, or explains the error of the training and testing data. The loss represents the performance of the model; lower values ​​indicate better performance. Similar to accuracy, the goal is to minimize the loss of training and testing data. During training, the model adjusts its parameters to reduce loss and ultimately improve its predictive ability. However, it is important to watch for signs of over-accommodation, the learning loss continues to decrease as the acceptance rate increases, and indicating that the model is not optimized for fabric text Overall, analysis of these images provides a good insight into the training process and working model. By carefully monitoring exposure and failure over time, researchers and practitioners can make informed decisions about training models and optimization and ensure that they are effective and reliable in real use.

Let's we placed the image of the character "स" into the model and after the prediction model, we can guess the picture of each letter, let's identify the letter in the prediction picture.



Result:



**Conclusion**

This Paper is a major advancement in handwritten Devanagari character recognition using convolutional neural networks (CNN). In the age of artificial intelligence, where smart machines are becoming popular, this project is a pioneering work on recognizing Devanagari handwritten characters. While optical character recognition (OCR) systems for English characters are already well developed, this project focuses on the unique nuances of Devanagari character recognition. Although it currently relies on machine interfaces, future integration of the system into mobile applications could facilitate extensive language exchange. Core skills are developed throughout the career development process, including time management, optimization and teamwork. The results show encouraging accuracy, which lays the foundation for further development of Devanagari-specific character recognition technology**.**

**References**

[1] C. E. Mook, C. Poo Lee, K. M. Lim and J. Yan Lim, "Handwritten Character and Digit Recognition with Deep Convolutional Neural Networks: A Comparative Study," 2023 11th International Conference on Information and Communication Technology (ICoICT), Melaka, Malaysia, 2023, pp. 137-141, doi: 10.1109/ICoICT58202.2023.10262721.

[2] Shukla, Ananya & Sharma, Ayush & Aggarwal, Akriti & Jain, Shikha. (2023). Devnagari Character Recognition using Optical Character Recognition (OCR). 373-379. 10.1109/CICTN57981.2023.10141358.

[4] Juneja, Sonia & Srivastva, Shagun & Tomer, Sonia & Gangwar, Pallavi & Tyagi, Yashika. (2023). Analysis of Machine Learning Algorithms for Digit and Alphabet Recognition: A Review. 1-6. 10.1109/ICICAT57735.2023.10263678.

[5] R. Patil, B. Narkhede, S. Gaonkar and T. Dave, "Deep Learning Based Marathi Sentence Recognition using Devnagari Character Identification," 2023 International Conference on Communication System, Computing and IT Applications (CSCITA), Mumbai, India, 2023, pp. 10-15, doi: 10.1109/CSCITA55725.2023.10104985.

[6] P. Patil and K. Noronha, "Character Segmentation of Devnagari Script in Printed Document Images using Projection Profiles," 2023 International Conference on Communication System, Computing and IT Applications (CSCITA), Mumbai, India, 2023, pp. 158-163, doi: 10.1109/CSCITA55725.2023.10104858.

[7] M. Geetha, R. C. Suganthe, S. K. Nivetha, S. Hariprasath, S. Gowtham and C. S. Deepak, "A Hybrid Deep Learning Based Character Identification Model Using CNN, LSTM, And CTC To Recognize Handwritten English Characters And Numerals," 2022 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2022, pp. 1-6, doi: 10.1109/ICCCI54379.2022.9740746.

[8] P. Pathak, N. Pachauri and P. Tiwari, "Devanagari Lipi Recognition Using Deep Learning Techniques," 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2023, pp. 1-5, doi: 10.1109/ISCON57294.2023.10111956.

[9] A. A. Mokhtar et al., "An Intelligent Handwritten Digits and Characters Recognition System," 2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakheer, Bahrain, 2022, pp. 670-673, doi: 10.1109/3ICT56508.2022.9990669.

[10] S. Katoch, M. Rakhra and D. Singh, "Recognition Of Handwritten English Character Using Convolutional Neural Network," 2022 4th International Conference on Artificial Intelligence and Speech Technology (AIST), Delhi, India, 2022, pp. 1-6, doi: 10.1109/AIST55798.2022.10064860.

[11] A. Nikitha, J. Geetha and D. S. JayaLakshmi, "Handwritten Text Recognition using Deep Learning," 2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, 2020, pp. 388-392, doi: 10.1109/RTEICT49044.2020.9315679.

[12] R. Dey, P. Gawade, R. Sigtia, S. Naikare, A. Gadre and D. Chikmurge, "A comparative Study of Handwritten Devanagari Script Character Recognition Techniques," 2022 IEEE World Conference on Applied Intelligence and Computing (AIC), Sonbhadra, India, 2022, pp. 431-436, doi: 10.1109/AIC55036.2022.9848911.

[13] B. Dessai and A. Patil, "A Deep Learning Approach for Optical Character Recognition of Handwritten Devanagari Script," 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, India, 2019, pp. 1160-1165, doi: 10.1109/ICICICT46008.2019.8993342.

[14] S. K. Manocha and P. Tewari, "Devanagari Handwritten Character Recognition using CNN as Feature Extractor," 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Pune, India, 2021, pp. 1-5, doi: 10.1109/SMARTGENCON51891.2021.9645786.

[15] Y. Gurav, P. Bhagat, R. Jadhav and S. Sinha, "Devanagari Handwritten Character Recognition using Convolutional Neural Networks," 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Istanbul, Turkey, 2020, pp. 1-6, doi: 10.1109/ICECCE49384.2020.9179193.

[16] M. S. Ansari, M. Wasid and S. A. Rahman, "Devanagari Handwritten Character Recognition using Transfer Learning with Deep CNN and SVM," 2022 5th International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT), Aligarh, India, 2022, pp. 1-5, doi: 10.1109/IMPACT55510.2022.10029268.

[17] S. P. Deore, "Devanagari Handwritten Compound Character Recognition Using Various Machine Learning Algorithms," 2021 2nd International Conference on Communication, Computing and Industry 4.0 (C2I4), Bangalore, India, 2021, pp. 1-6, doi: 10.1109/C2I454156.2021.9689425.

[18] V. V. Mainkar, J. A. Katkar, A. B. Upade and P. R. Pednekar, "Handwritten Character Recognition to Obtain Editable Text," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 599-602, doi: 10.1109/ICESC48915.2020.9155786.

[19] R. Vaidya, D. Trivedi, S. Satra and P. M. Pimpale, "Handwritten Character Recognition Using Deep-Learning," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2018, pp. 772-775, doi: 10.1109/ICICCT.2018.8473291.

[20] A. Kumar, K. Srivastava, A. Saraswat, A. Mishra, I. Masand and F. Fhadil, "Handwritten Character Recognition Using Machine Learning," 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2023, pp. 1150-1153, doi: 10.1109/ICACITE57410.2023.10182858.

[21] P. K. Sonawane and S. Shelke, "Handwritten Devanagari Character Classification using Deep Learning," 2018 International Conference on Information , Communication, Engineering and Technology (ICICET), Pune, India, 2018, pp. 1-4, doi: 10.1109/ICICET.2018.8533703.

[22] A. Mohite and S. Shelke, "Handwritten Devanagari Character Recognition using Convolutional Neural Network," 2018 4th International Conference for Convergence in Technology (I2CT), Mangalore, India, 2018, pp. 1-4, doi: 10.1109/I2CT42659.2018.9057991.

[23] D. Salunke, P. Sabne, H. Saini, V. Shivanagi and P. Jadhav, "Handwritten Devanagari Word Recognition using Customized Convolution Neural Network," 2021 International Conference on Computing, Communication and Green Engineering (CCGE), Pune, India, 2021, pp. 1-5, doi: 10.1109/CCGE50943.2021.9776351

[24] A. Shrivastava, I. Jaggi, S. Gupta and D. Gupta, "Handwritten Digit Recognition Using Machine Learning: A Review," 2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC), Greater Noida, India, 2019, pp. 322-326, doi: 10.1109/PEEIC47157.2019.8976601.

[25] B. M. Vinjit, M. K. Bhojak, S. Kumar and G. Chalak, "A Review on Handwritten Character Recognition Methods and Techniques," 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2020, pp. 1224-1228, doi: 10.1109/ICCSP48568.2020.9182129.

.