Reading Modi Lipi: A Deep Learning Journey in Character Recognition

Kanchan Varpe, Sachin Sakhare *

Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune 411048 (India)

* Corresponding author: Kanchanv2007@gmail.com (K. Varpe), sachin.sakhare@viit.ac.in (S. Sakhare)

Received 23 July 2023 | Accepted 21 April 2024 | Published 9 September 2024

Abstract

Advancements in deep learning methodologies have played a significant role in the success of various character recognition processes. Character recognition refers to the technique of identifying either handwritten or printed characters from documents and their conversion into a form that can be read by machines. MODI script, an ancient Indian script, is categorized under the Devanagari script and holds historical significance. Despite its historical importance, there are only a few MODI translators available. Conversely, there exist a vast number of historical documents written in MODI that are yet to be deciphered. Recognizing characters in Indian language scripts poses many challenges due to the complex nature of the scripts and variations in individuals' writing styles. This paper provides an overview of the newest advancements in the Handwritten Optical Character Recognition (HWCR) methodology specifically designed for the MODI script. Utilization of residual networks and inception in image classification has gained popularity in recent times.

In this paper the authors have implemented three techniques: ResNet9, ResNet50, and InceptionNet V3, trained specifically for handwritten MODI characters and vowels. The dataset used for training the models consists of handwritten MODI script images. The benchmark database from IEEE data port for handwritten MODI script is used to evaluate the performance. The dataset contains 46 classes, including 10 vowel classes and 36 consonant classes. Each class comprises 90 images, resulting in a total of 4140 images. The image size in the dataset is 227×227. The accuracy achieved by the trained models is as follows: 98.92% for ResNet9, 91.91% for ResNet50, and 86% for Inception Net V3. The obtained results have been compared with existing models and it is observed that the proposed model attained improved performance parameters and less training and validation losses in comparison to existing methods. There are several advantages of the proposed model in comparison to state of the art, namely minimal training and validation loss. In addition to this, the proposed approach improved generalization and robustness, and improved model scalability.

Keywords

Classification, Deep Learning, Feature Extraction, Image Processing, InceptionNet V3, ResNet50, ResNet9.

DOI: 10.9781/ijimai.2024.09.002

I. Introduction

The script known as "MODI Lipi" is an old script and has significant importance in medieval Maharashtra. It served as the script for a wide range of historical documents, covering subjects such as land revenue, judiciary, justice, religious matters, property matters, and military orders and strategies during the reign of Chhatrapati Shivaji Maharaj and the Peshwas.

For nearly 700 years, MODI remained the medium for administrative and official documentation in Marathi within Maharashtra, until the late 19th century. This prolonged usage emphasizes the need to prioritize the recognition process for the MODI script, as it contains a wealth of valuable information yet to be uncovered.

Researchers are interested in studying handwritten optical character recognition (HOCR) for MODI Lipi, seeking to understand the writing techniques used during the medieval period of the Maratha Empire. The automation of recognition holds the potential to unveil a trove of historical knowledge and understanding. HOCR holds the most promise for the future in image processing, pattern recognition, natural language processing, and analysis of documents. Due to advancements in various technologies for image processing and pattern recognition, considerable improvements were observed in identifying handwritten characters [1].

HOCR is challenging because different scripts have their own style of writing when it comes to shape and continuity. Table I enumerates challenges faced by handwritten Modi character recognition.

Please cite this article as:
K. Varpe, S. Sakhare. Reading Modi Lipi: A Deep Learning Journey in Character Recognition, International Journal of Interactive Multimedia and Artificial Intelligence, vol. 9, no. 1, pp. 75-83, 2024, http://dx.doi.org/10.9781/ijimai.2024.09.002

A. History

The Modi script has been utilized for writing Marathi language for an estimated duration of 700 years.

The origins of the Modi script are surrounded by various narratives. One theory suggests that Hemadpant, a minister during the rule of Mahadev, is the creator of the Modi script. However, alternative theories propose that Hemadpant did not invent the script from scratch, but rather refined an existing version of Modi before introducing it as the official script for writing Marathi.

According to research conducted by Kulkarni et al. [2], one theory asserts that Hemadpant brought the Modi script from Sri Lanka to India. Another theory suggests that the script was developed during the reign of Chhatrapati Shivaji Maharaj by Baḷaji Avaji, the secretary of state.

These differing accounts contribute to the intriguing history and origins of the Modi script, highlighting the need for further research and exploration to uncover its true genesis and evolution.

TABLE I. Challenges for Handwritten MODI Character Recognition

Sr. No.	Challenges
1	Large number of classes.
2	Open and closed loops, arcs, strokes, straight lines.
3	Various strokes existing in a character may touch each other due to hasty writing.
4	Large intra-class variations due to different writing styles.
5	There are inter-class similarities.
6	Extracting structural features is very difficult due to the complex structure of some of its alphabet letters.
7	MODI is written in cursive type; hence it creates extra branches in letters.

Modi evolved into different forms over the years until the 20th century when the Balbodh style of Devanagari script was adopted as a standard form for writing Marathi. Different styles of Modi are listed according to the eras and centuries in which they emerged in Table II. As all documents before the 20th century were written in the Modi script, this makes Modi historically significant [3].

TABLE II. Modi Script Writing Forms Over Years

Modi style	Time Period
Proto-Modi (आद्यकालीन)	12th Century
Yadav Era (यादवकालीन)	13th Century
Bahamani Era (बहमनीकालीन)	14th – 16th Century
Shiva Era (शविकालीन)	17th Century
Chitnisi (चतिनीसी)	17th Century
Peshwa Era (पेशवेकालीन)	Lasted till 1818
Anglakalin or British Colonial Era (आंग्लकालीन)	1818-1952

B. Properties of MODI Script

During the 12th century, the MODI script gained prominence as a writing system for the Marathi language. This ancient script continued to be extensively utilized starting in the 12th century. The term "MODI" is deemed to have been derived from the Marathi verb "moḍaṇe," which means to "to break or bend" [4].

To write in the MODI script, a writing instrument called "Boru" or "Lekhan" was utilized, with the pen being made from bamboo. Despite the fact that the MODI script is based on the Devanagari script, there are significant variations between them. The MODI script is part of the Nagari family of scripts and is most suitable for continuous writing. These variations show up in rendering behaviors, letter forms and orthography of the characters. Joseph et al. [5] provide a comprehensive exploration of these distinguishing features.

The unique characteristics and historical significance of the MODI script have attracted considerable scholarly attention, prompting in-depth investigations into its structural intricacies and functional aspects.

The behaviors exhibited by characters in specific contexts, such as combinations of consonants and vowels and consonant conjuncts, are intrinsic features of MODI orthography and differentiate it from the Devanagari script. The MODI script comprises 46 distinct letters, with 10 vowels and 36 consonants. The task of word segmentation for the MODI script is quite challenging as no termination symbol is used to designate the end of a sentence. In the MODI script, all the characters are written as if they are hanging from a horizontal line that is drawn across the page. This distinct style sets MODI apart. Notably, research conducted by Kulkarni et al. [6] highlights that the elimination of terminating symbols in the MODI script significantly increased the writing speed.

This removal reduced the need to lift the "Boru" pen frequently. The unique characteristics of the MODI script play a crucial role in defining its identity and require specialized approaches for precise interpretation and analysis. Continual research endeavors to further enhance our comprehension of MODI orthography and foster the development of effective methods for working with MODI script documents.

By delving into the intricacies of the MODI script, researchers strive to uncover its nuances, intricacies, and underlying patterns. This deeper understanding helps in refining techniques for accurate recognition, transcription, and translation of MODI script content. Additionally, ongoing research contributes to the advancement of tools and methodologies that facilitate efficient handling and processing of MODI script documents.

The ultimate goal is to bridge the gap between the historical significance of the MODI script and its effective utilization in contemporary contexts. By harnessing insights gained from ongoing research, scholars, language enthusiasts, and technological advancements can collaborate to preserve, analyze, and leverage the rich cultural heritage contained within MODI script documents.

Modi is a beautifully flowing cursive script known for its elegance. The fundamental graphical structure of Modi letters, as shown in Fig. 1, Fig. 2, and Fig. 3, follows the traditional bārākhadi format.

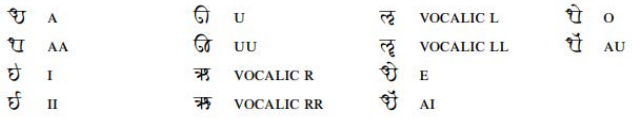

Fig. 1. Vowels used in Modi Script.

There are 13 combining vowel signs as shown in Fig. 2.

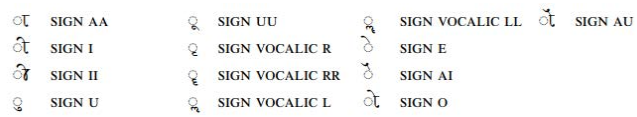

Fig. 2. Vowel Signs in Modi Script.

There are 34 consonant letters as shown in Fig. 3.

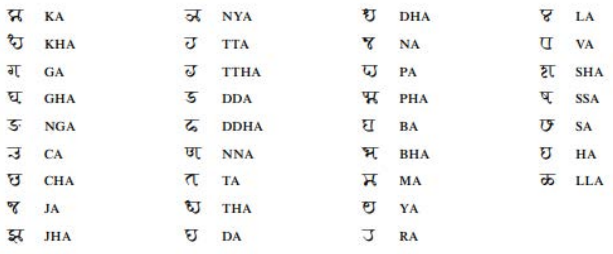

Fig. 3. Consonants in Modi Script.

The organization of the remainder of the paper is set out here. Section II contains an in-depth and organized summary of a literature review of the innovative methodologies that researchers have proposed along with challenges from state of the art. Section III contains an outline of research techniques of the proposed Convolutional Neural Network (CNN)-based deep learning models. Section IV describes the database. Section V includes a discussion of the initial outcome procured from the experimentation utilizing the deep learning architectures. Finally, Section VI concludes and highlights future research possibilities.

II. Literature Review

Table III presents various research works that focus on MODI character or numeral recognition. Each researcher has employed distinct methodologies in their studies.

In the research conducted by Besekar et al. [7], classification and recognition of MODI characters were achieved using mathematical morphology and decision tree algorithms. Features extracted from vertical and horizontal lines, blobs and concavities were utilized, resulting in an accuracy of 75%.

Besekar et al. [8] employed a two-layer feed-forward neural network with scaled conjugate gradient for MODI vowel classification. Different accuracies were obtained by using different directional chain codes. For instance, 65.3% accuracy was achieved using the 4-directional chain code histogram (CCH4) and the 4-directional normalized chain code histogram (CH4D). Furthermore, using the 4-directional CCH, the 4-directional NCCH, and a centroid, the accuracy improved to 67.9%. Similarly, the 8-directional CCH and the 8-directional NCCH resulted in 65.9% accuracy, while utilizing the 8-directional CCH, the 8-directional NCCH, and a centroid increased the accuracy to 73.5%.

Ramteke et al. [9] introduced the use of a Variance table to categorize MODI numerals, achieving an accuracy of 93.5%. This approach involved dividing the numeral into four identical 15 × 15 square zones and analyzing their polar coordinates, variance, theta, and rh distance to generate a feature set for recognition.

These research studies demonstrate various techniques and algorithms employed for MODI character recognition, highlighting the progress and achievements in this field.

In the study conducted by Katkar et al. [10], the utilization of the Kohonen Neural Network for classification, coupled with a measured structural similarity approach for feature extraction, resulted in a performance rate ranging from 91% to 97%.

The Euclidean distance classifier is generally employed for MODI character identification. In the research by Kulkarni et al. [5], a correct recognition rate of 94.92% was obtained using Zernike moments, while an accuracy of 94.78% was obtained using Zernike complex moments with an integrated approach for heterogeneous zones in offline character recognition.

Kulkarni et al. [11] employed Hu's seven invariant moments as the feature vector for training samples, and recognition of test sample features was performed using the Euclidean distance classifier. The accuracy achieved with Hu's invariant seven moments was 70%. Zernike Moments, on the other hand, provide statistical measures of pixel distribution around the character's center of gravity and capture information at a single boundary point. When using Zernike moments, an accuracy of 86.66% was obtained.

In the research work of Sidra Anam et al. [12], the Kohonen Neural Network was employed for classification, preceded by Otsu's Binarization algorithm, resulting in a performance rate of 72.6%. Otsu's threshold algorithm was employed for the Modi Script Character Recognizer System (MSCR).

Deshmukh et al. [13], utilized the chain code feature extraction technique with a non-overlapping blocking strategy, followed by the use of correlation coefficient. The correlation feature (r or R) is a measure of the strength and direction of the linear relationship between two variables. It is calculated by dividing the covariance of the variables by the product of their standard deviations. The maximum recognition rate procured on a database of 30,000 images was 85.21%. The results of the recognition showed an improved performance when using 5 × 5 grid divisions. These research studies highlight various approaches and techniques employed for MODI character recognition, showcasing the advancements and achievements in this field.

Chandure et al. [14], conducted a study in which the entire character set of the MODI script was recognized using the chain code in combination with BPNN, KNN, and SVM. The respective accuracies obtained were 37.5%, 60%, and 65%. Feature extraction was performed using these techniques while combining the BPNN, KNN, and SVM techniques, resulting in accuracies of 15%, 40%, and 47.5% respectively. There are multifarious reasons of low accuracy namely inadequate training of model such as imbalanced dataset and limited discriminative power of algorithm. etc.

The output layer consisted of 13 neurons for Devanagari characters and 12 neurons for MODI characters. When the chain code feature extraction technique was used, an accuracy of 37.5% was achieved, while an accuracy of 15% was obtained when the insertion junction feature extraction technique was employed.

Since there are 13 classes of vowels, 13 SVM classifiers were required for their separation. When the chain code was used as a feature extraction technique, an accuracy of 65% was achieved. Similarly, when the insertion junction was used as a feature extraction technique, an accuracy of 47.5% was obtained.

The k-nearest neighbor algorithm (KNN) yielded an accuracy of 60% when the chain code feature extraction technique was employed, and an accuracy of 40% when the insertion junction feature extraction technique was utilized.

In research conducted by Gharde et al. [15], a combination of moment invariant and affine moment invariant approaches was used. This hybrid approach extracted 18 features from each number or character. SVM was employed as the classifier, resulting in an accuracy of 89.72%. An accuracy of 89.72% was obtained when two feature extraction approaches, namely, moment invariant and affine moment invariant were utilized along with SVM being used as the classifier.

Maurya et al. [16] came up with a proposal for a framework for recognizing handwritten MODI characters digitally. The authors used heuristics that were empirically determined to figure out the contribution of features from a hybrid feature space for character recognition. The pre-processing methods included noise removal, binarization, skeletonization, character segmentation and smoothing. The characters that were segmented underwent post-processing and recognition once the pre-processing was done. The average accuracy obtained was 91.20%, with a claimed best accuracy of 99.10%.

TABLE III. Overview of Research Work Related to Modi Character Recognition

Sr. No.	Author	Feature Extraction	Classification	Data type			Accuracy%
				Characters	Vowels	Numerals	
(1)	Besekar D.N. et al., 2011 [7]	Blobs, vertical & horizontal lines, concavities	Mathematical morphology, decision tree.	-	-	✓	75
(2)	Besekar D.N. et al., 2012 [8]	Chain code, image, and centroid method	Two-layer feed forward network with scaled conjugate gradient.	-	✓	-	65.3, 67.9, 65.9, 73.5
(3)	Ramteke R.J. et al., 2012 [9]	Polar coordinate of zone, Variance, theta angle and Rh distance	Comparing the variance table.	-	-	✓	93.5
(4)	Katkar G. S. et al., 2013 [10]	Structural similarity	Kohonen neural network, BPNN	✓	-	-	91-97
(5)	Solley Joseph et al., 2014 [5]	Zoning, (1) Zernike Moments and (2) Zernike complex moments	Euclidean Distance	✓	-	-	1) 94.92 2) 94.78
(6)	Kulkarni S.A et al., 2015 [11]	(1) Hu's invariant features (2) Zernike moments	Euclidean distance	-	-	✓	1) 70 2) 86.66
(7)	Sidra Anam et al., 2015 [12]	-	Kohonen neural network	✓	-	-	72.6
(8)	Manisha S. et al., 2015 [13]	Chain code feature extraction and non-overlapping blocking strategy	Correlation coefficient	-	-	✓	85.21
(9)	Chandure S.L. et al., 2016 [14]	Chain Code Histogram Features, Intersection / Junc. Features	1)BPNN 2)KNN 3)SVM	✓	-	-	Chain Code: 60, 37.5, 65 InsertionJunc 40, 15, 47.5
(10)	Gharde S.S. et al., 2016 [15]	Using Moment Invariant and Affine Moment Invariant	SVM	✓	-	-	89.72%
(11)	Maurya R.K. et al., 2018 [16]	Chain code	Empirically determined heuristics	✓	-	-	Average: 91.20% Best: 99.10%
(12)	Joseph S. et al., 2020 [17]	-	1)Euclidean distance classifier, 2)Manhattan distance classifier	✓	-	-	1) 99.28% 2) 94%
(13)	S. Joseph et al., 2020 [18]	CNN autoencoder	SVM	✓	-	-	99.3%
(14)	Shruti Sawant et al., 2020 [19]	-	CNN	✓	-	-	95.44%, 95.97%
(15)	Solly Joesph et al., 2021[20]	WT-SVD	Euclidean distance	✓	-	-	99.5%
(16)	Joseph S. et al., 2021 [21]	-	ACNN	✓	-	-	99.78%
(17)	Tamhankar et al., 2021 [22]	-	DCNN	✓	-	-	64%
(18)	Kirti et al., 2021 [23]	-	AlexNet	✓	-	-	89.72 %
(19)	Chandure et al., 2021 [24]	-	1)Supervised TL 2)SVM	✓	-	-	92.32%
(20)	Kulkarni S.A. et al., 2021 [25]	Zoning, 1.Zernike moments 2.Zernike Complex moments 3.Ensemble Bagging	Euclidean distance classifier	✓	-	-	1. 94.92% 2. 94.78% 3. 97.68%
(21)	Jidnyasa Kondhare et al., 2022 [26]	-	1)CNN 2)VGG16	✓	-	✓	1)76.46% 2)92.48%
(22)	Maitreyi Ekbote et al., 2022 [27]	-	1)Random Forest 2)XGBoost	✓	-	✓	1)92% 2)93.3%
(23)	Vishal Pawar et al., 2022 [28]	-	CNN	✓	-	-	91.62%
(24)	Chaitali Chandankhede et al., 2023 [29]	-	1)Inception V3 2)ResNet 50	✓	✓	-	93.923% 94.552%

In the study conducted by Joseph et al. [17], two algorithms utilizing distance classifiers were implemented for the classification of handwritten Modi script. The data underwent vectorization, followed by noise reduction techniques. The first experiment utilized Euclidean distance classifiers, while the second experiment used the Manhattan distance classifier. The procured accuracies were 99.28% and 94% respectively. The Manhattan distance method demonstrated better performance in terms of time complexity, although it was less accurate in comparison to the second method.

S. Joseph et al. [18], came up with a proposal to make use of a CNN autoencoder as a feature extractor in order to recognize characters in the MODI script. Making use of the CNN autoencoder, the feature set size was decreased from 3600 to 300. SVM was then used as a classifier for the features that were extracted and an accuracy of 99.3% was achieved.

Two architectures were considered in the study conducted by Shruti Sawant et al. [19]. The first framework was made up of two convolution layers that came first. Next came max pooling and fully connected layers. The second architecture comprised three sets of convolution and max pooling layers. Three fully connected layers followed. The accuracy obtained for the first architecture was 95.44%, and for the second architecture, it was 95.97%.

Joseph et al. [21], utilized an Augmented CNN (ACNN) model by incorporating data augmentation techniques such as 45-degree horizontal and vertical flips. This combined approach with a CNN resulted in an accuracy of 99.78%.

Tamhankar P.A. et al. [22], implemented a Deep CNN (DCNN) with Rectified Linear Unit neural activation in the convolutional layers, which improved the performance and reduced computational requirements. Their work achieved an accuracy of 64%.

Mahajan Kirti et al. [23], employed a pre-trained Conventional Neural Network called AlexNet for training their model. AlexNet has been trained on a vast dataset of 15 million labeled high-resolution images from 22,000 categories. The experimental arrangement utilizing the AlexNet model in MATLAB yielded an accuracy of 89.72%.

Savitri Chandure et al. [24], utilized a DCNN, namely, AlexNet as a pre-trained network, transferring its weights for retraining. They trained a SVM classifier on activation features to procure classifier models, obtaining an accuracy of 92.32%.

Kulkarni S.A et al. [25] used Zernike moments to achieve an accuracy of 94.92% for an integrated approach, while Zernike Complex moments yielded 94.78% for the integrated approach.

Jidnyasa Kondhare et al. [26] employed CNN for Modi character recognition, achieving a training accuracy of 73.93%, a validation accuracy of 76.46%, and a training time of 9866 seconds. They also tried VGG16, which resulted in an accuracy of 99.73% for training, an accuracy of 92.48% for validation, and a training time of 7267 seconds.

Maitreyi Ekbote et al. [27] proposed a character recognition model on the basis of a CNN for effectively identifying MODI numerals and alphabets. They enhanced the model by utilizing a classifier like Random Forest or XGBoost, achieving recognition accuracies of 92% for characters and 93.3% for numerals.

Vishal Pawar et al. [28] developed a CNN model to recognize characters in the MODI script. Due to the limited dataset of 4140 images, they applied data augmentation methodologies such as flipping, rotation, noising, and blurring to expand the dataset. The trained model was approximately 91.62% accurate in recognizing handwritten MODI characters.

Chaitali Chandankhede et al. [29], experimented with character recognition using ResNet50 and InceptionNet V3. Their dataset consisted of about 8000 photos, and they employed the Global Otsu threshold approach for binarization. The processed images recognized using ResNet50 achieved a testing accuracy of 94.552% with a model precision of 0.86, while InceptionV3 provided a testing accuracy of 93.923% with a model precision of 0.843. The article suggests further research in different binarization strategies, varied CNN models, regularization treatment configurations, and the automation of a powerful word recognition model.

III. Methodology

The primary focus of this work is to leverage deep learning models, specifically Residual Networks and Inception V3, for character recognition in the MODI script. To evaluate performance, the benchmark database for handwritten MODI script from IEEE DataPort [30] is utilized.

The training involves handwritten MODI script images from this IEEE DataPort dataset, which serves as the benchmark for model performance evaluation. This dataset includes 46 distinct classes: 10 vowel classes and 36 consonant classes. Each class contains 90 images, leading to a total of 4140 images. These images are standardized to a size of 227×227 pixels.

The hyperparameters are as follows: an initial learning rate of 0.001, a batch size of 32, activation functions including Sigmoid and ReLU, and the Adam optimizer.

A. ResNet 9-Residual Networks

ResNet is short for Residual Networks, and in our model, we pass our data through 9 layers. In Residual Networks, skip connections are employed for connecting the activations of each of the layers to the other layers, omitting a few layers in between. This forms a residual block, and multiple residual blocks are stacked on top of each other to create the entire network. In our character recognition model, we have 8 convolutional blocks. Each block performs a convolution operation on the image using a 3x3 kernel size, followed by batch normalization. The results of each block are then passed through the ReLU activation function and sent to the next block. The skip connections are utilized to form a residual block, which consists of two consecutive convolutional blocks. Additionally, max pooling is performed to down sample or pool the feature map before passing it to the next layers. The flow of this process is illustrated in Fig. 4.

We will utilize the ResNet-9 architecture in our character recognition model. In our model, we have 8 convolutional blocks, each performing a convolution operation on the image using a 3x3 kernel size, followed by batch normalization. The output of each block is normalized and then passed through the ReLU activation function before being sent to the next block. A residual block is formed by stacking two consecutive convolutional blocks, and skip connections are used to connect the activations.

Additionally, max pooling is performed after each block to down sample or pool the feature map before passing it to the next layers, creating a down sampled or pooled feature map. This flow is illustrated in Fig.4.

To classify images based on predictions, our ResNet model utilizes max pooling to reduce the spatial dimensions, followed by flattening the feature vector into a linear vector. This linear vector is then passed through a dropout layer to consider only relevant features before being fed into the linear layer. The linear layer performs a linear transformation to obtain the class probabilities vector. The predicted class is determined by selecting the class with the highest probability. This process is depicted in the classifier block shown in Fig. 4.

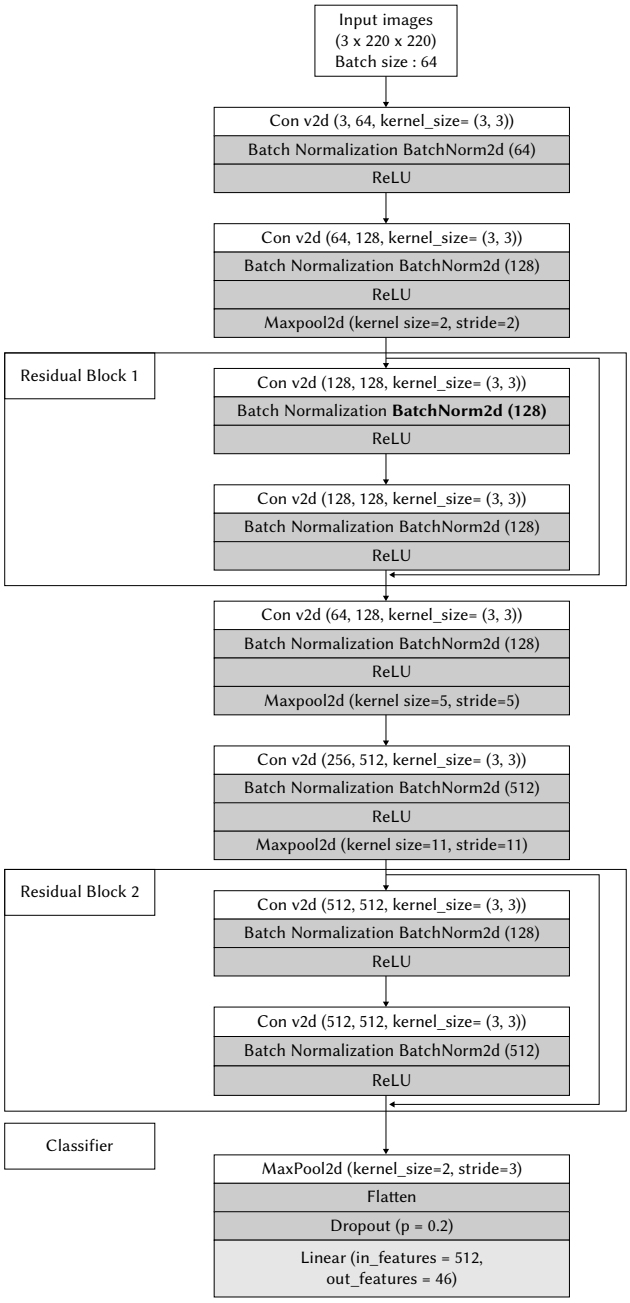

Fig. 4. Resnet 9's architecture.

B. Resnet 50

ResNet-50 [31] is a CNN with 50 layers, and it is a widely used architecture in computer vision applications. ResNet, short for Residual Networks, is a common neural network that has enabled the training of deep networks with hundreds of layers. In the ResNet architecture, denoted as the Residual Neural Network, the labels "50 layers," "101 layers," and "152 layers" correspond to the collective count of convolutional layers integrated into the network. These numerical designations, encapsulated within square brackets like [3, 4, 5, 6], [3, 4, 23, 8], and [3, 8, 36, 3], signify the arrangement of convolutional layers within distinct segments of the ResNet structure.

One of the challenges faced by CNNs is the "Vanishing Gradient Issue," where gradients significantly diminish during back propagation, leading to minimal weight updates. To address this problem, ResNet introduces a solution known as "SKIP CONNECTION."

A skip connection is a direct connection that bypasses certain layers in the model. This skip connection alters the output. When a skip connection is not used, the input 'X' is multiplied by the layer weights, and a bias term is added, followed by the activation function F () to obtain the output as:

$$F\,(w^{*}x + b \text{ (equivalent to } F(X)) \tag{1}$$

However, with the skip connection technique, the output becomes:

$$F(X) + X \tag{2}$$

Before training the model, preprocessing steps are applied to the images. This involves padding and cropping the images to a size of (224,224) to ensure uniformity.

The images are also adjusted through normalization. The standard deviation of each channel is divided by the means of the image tensors before subtracting them.

This normalization ensures that values from any one channel do not disproportionately influence losses and gradients during training.

To classify images based on predictions, our ResNet model utilizes max pooling to downs ample the feature map, followed by flattening the resulting feature vector into a linear vector. A dropout layer is applied to retain only relevant features before passing them to a linear layer, which performs a linear transformation to obtain the class probabilities vector. The predicted class is determined based on the maximum probability.

C. InceptionNet V3

Inception-v3 is a CNN consisting of 48 layers. It is known for its effective architecture design and parameter reduction techniques. The network incorporates three different kinds of Inception modules: Inception A, Inception B, and Inception C. These modules combine both convolutional layers and pooling layers in a parallel fashion to capture distinctive features while reducing the number of parameters. To achieve parameter reduction, small convolutional layers such as 3×3, 1×3, 3×1, and 1×1 are employed within the Inception modules. The model also includes symmetric and asymmetric components like convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers. Batch normalization is utilized to a great extent, ensuring improved training stability. Softmax is employed to compute the loss during training.

InceptionNet V1 serves as the foundation for subsequent versions, including Inception-v3. Each version builds upon the previous one, introducing iterative improvements. In the case of Inception-v3, notable modifications include factorization into smaller convolutions, spatial factorization using asymmetric convolutions, the incorporation of auxiliary classifiers, and efficient grid size reduction. For our specific implementation, we adjusted the number of output channels in the final layer from 1,000 to 98, as our network had fewer classes compared to the original Inception-v3 model. The original model was designed for a dataset with 1,000 classes. InceptionNet performs exceptionally well on large datasets, and expanding our dataset can enhance the model's accuracy in comparison to other models [32].

IV. Database Description

This section of the paper includes a description of datasets that are available publically for the handwritten MODI script [30]. System performance is analyzed on the benchmark database from the IEEE data port (https://ieeedataport.org/documents/handwritten-modi-characters). The dataset for handwritten MODI script comprises 46 classes, with 10 classes for vowels and 36 classes for consonants. Each class contains 90 images, resulting in a total of 4140 images. To

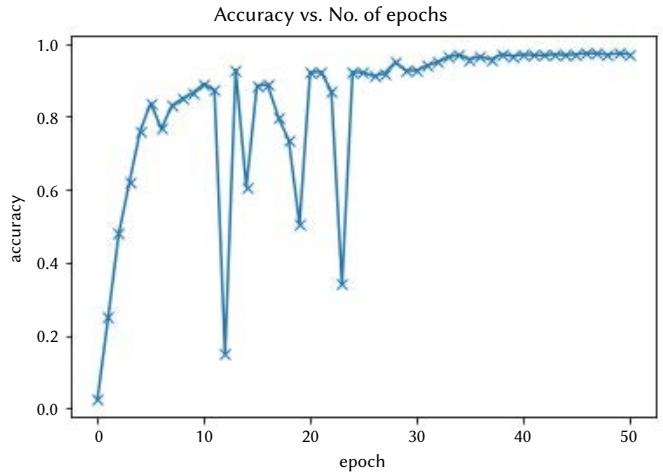

Fig. 5. ResNet 9 Accuracy Plot.

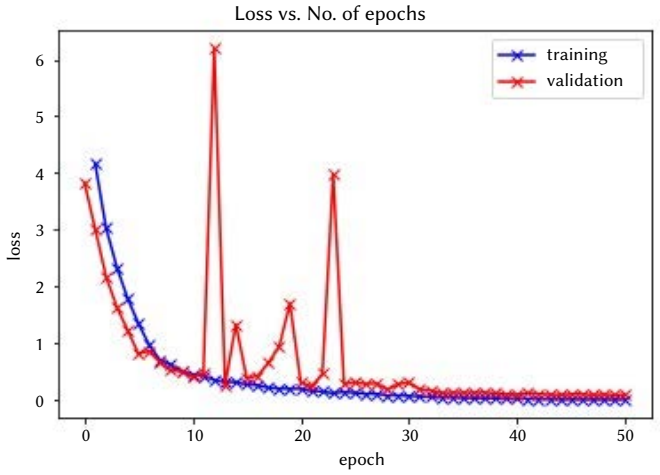

Fig. 6. ResNet 9 Loss Plot.

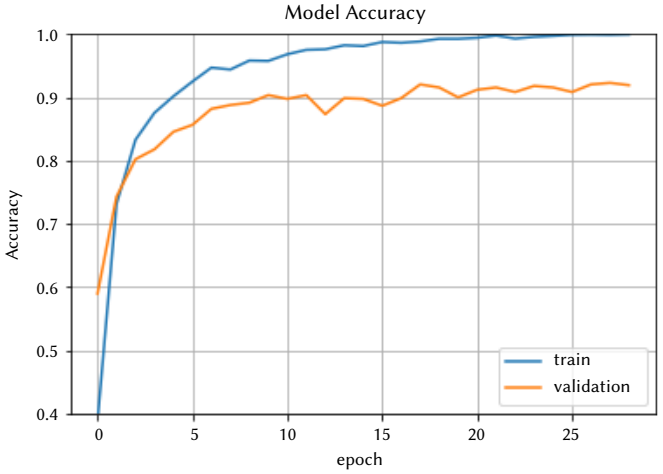

Fig. 7. ResNet 50 Accuracy Plot.

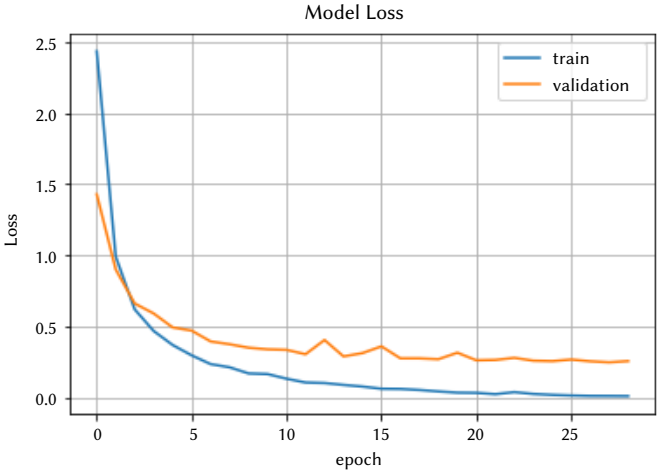

Fig. 8. ResNet 50 Loss Plot.

prepare the dataset for training and testing, it was split into a ratio of 80:20, generating 3336 training images and 834 testing. images. During training, a batch size of 32 was utilized, along with other hyper parameters including weight decay, gradient clipping, and the Adam optimizer as the optimization function with a specified learning rate.

In this research, there was employed multifarious data augmentation methods to improve the diversity as well as robustness of the dataset utilized for model training. These techniques include but not limited to, image rotation, vertical and horizontal flipping, random cropping and zooming, brightness and contrast adjustment. Arbitrary rotation of pictures by a certain degree is also done to simulate alterations in image perspective.

Images were mirrored vertically or horizontally to augment the dataset with additional variations. Arbitrary zooming and cropping were applied to introduce changes in image scaling and composition. Brightness and contrast levels were adjusted to simulate different lighting conditions.

V. Experimental Results and Discussion

All the mentioned CNN models were trained with different values of hyper parameters to get optimal results. The performances of these CNN models are d iscussed in this section.

A. ResNet9

Data augmentation techniques were utilized to enhance the generalization and training effectiveness of the model. The images were converted into tensors using the PyTorch framework for training purposes.

The model underwent training for 50 epochs, resulting in a training accuracy of 97.32%. The accuracy progression throughout the epochs is illustrated in Fig. 5. Additionally, the model's loss consistently decreased over time.

Subsequently, the trained model was evaluated using the test dataset images, yielding a test accuracy of 98.92% shown in Fig. 6.

B. ResNet50

The ResNet-50 model underwent training for 29 epochs to achieve optimal accuracy on the dataset. The training accuracy obtained was 99.94%, while the testing accuracy reached 91.91%.

Fig. 7 illustrates the relationship between the model accuracy and the number of epochs, showcasing how the accuracy improves over time. On the other hand, Fig. 8 showcases the model loss throughout the epochs, demonstrating a gradual decrease in loss.

C. InceptionNet V3

The Inception V3 model underwent training for 50 epochs. The training accuracy achieved was 78.6%, while the testing accuracy reached 86%.

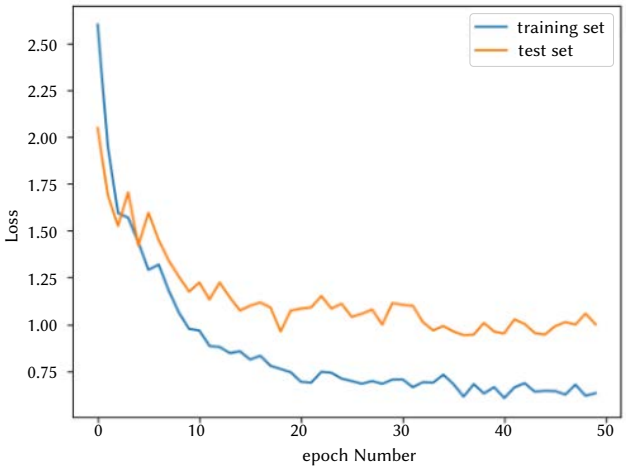

Fig. 9. Inception V3 Loss Plot.

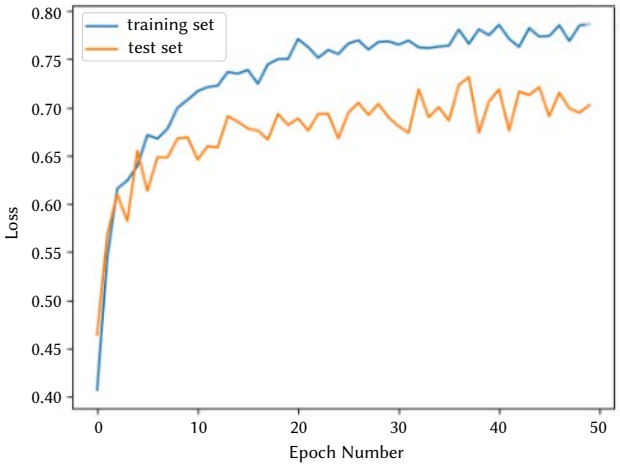

Fig. 10. Inception V3 Accuracy Plot.

Fig. 9 illustrates how training and validation accuracy and the number of epochs are related, providing insights into how accuracy evolves during training. Additionally, Fig. 10 showcases the training and validation loss throughout the epochs, demonstrating the gradual decrease in loss.

TABLE IV. Training and Testing Accuracy

Technique used	Number of Epochs	Training Accuracy	Testing Accuracy
ResNet-9	50	97.32%	98.92%
ResNet-50	29	99.94%	91.91%
InceptionNet V3	50	78.6%	86%

The models demonstrated significant performance levels. ResNet-50, trained over 29 epochs, achieved a training accuracy of 99.94% and a testing accuracy of 91.91%. Inception V3, after its training process, recorded a training accuracy of 78.6% and a testing accuracy of 86%, as presented in Table IV. Meanwhile, in our experiments, ResNet-9 yielded a training accuracy of 97.32% and a testing accuracy of 98.92%

VI. Conclusion and Future Scope

The work in this paper focuses on utilizing deep learning models, specifically Residual Networks and Inception V3, for MODI script character recognition. The training process involved fine-tuning various hyperparameters to obtain the desired outcomes. However, challenges such as a limited image dataset, similarities between classes, word continuity, and the cursive nature of the MODI script remain significant obstacles in handwritten MODI character recognition. To address these challenges, future work can focus on augmenting the dataset to avoid inter-class misclassification and improve CNN invariance qualities. Additionally, there is potential for enhancing recognition accuracy and expanding the scope to include word and line recognition. Segmenting text in the MODI script poses a considerable challenge due to the absence of word or sentence stopping symbols, making it an area for future investigation.

References

[1] V. Kanchan and S. Sakhare, "Review of Character Recognition Techniques for MODI Script," *Indian Journal of Science and Technology*, vol. 16, no. 26, pp. 1935-1946, 2023. doi: 10.17485/IJST/v16i26.485.

[2] K. Sadanand, L. Prashant, R. Ramesh and L. Pravin, "Impact of zoning on Zernike moments for handwritten MODI character recognition," *2015 International Conference on Computer, Communication and Control (IC4), Indore*, India, 2015, pp. 1-6, doi: 10.1109/IC4.2015.7375516.

[3] S. Joseph and J. George, "Handwritten Character Recognition of MODI Script using Convolutional Neural Network Based Feature Extraction Method and Support Vector Machine Classifier," *2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP), Nanjing, China*, 2020, pp. 32-36, doi: 10.1109/ICSIP49896.2020.9339435

[4] A. Pandey, "Proposal to Encode North Indic Number Forms in ISO/IEC 10646," 2007. [Online]. Available: https://api.semanticscholar.org/CorpusID:215863507

[5] S. Joseph and J. George, "Feature Extraction and Classification Techniques of MODI Script Character Recognition," *Pertanika Journal of Science and Technology*, vol. 27, pp. 1649-1669, 2019. Available: https://api.semanticscholar.org/CorpusID:260481222.

[6] K. Sadanand, P. Borde, R. Ramesh, and P. Yannawar, "Offline MODI Character Recognition Using Complex Moments," *Procedia Computer Science*, vol. 58, pp. 516-523, 2015, doi: 10.1016/j.procs.2015.08.067.

[7] D. Besekar, "Recognition of numerals of MODI script using morphological approach," *International Referred Research Journal*, ISSN 0974-2832, pp. 0974-2832, 2011

[8] D. Besekar, "Special Approach for Recognition of Handwritten MODI Script's Vowels," *in National Conference "MEDHA 2012"*, vol. 1, no. 1, pp. 48-52, September 2012. [Online]. Available: /proceedings/medha/number1/8679-1023/.

[9] D. Besekar and R. Ramteke, "Feature extraction algorithm for handwritten numerals recognition of MODI script using zoning-based approach," *International Journal of Systems, Algorithms and Applications*, vol. 2, pp. 1-4, 2012

[10] S. Ramteke and G. Katkar, "Recognition of Off-line Modi Script: A Structure Similarity Approach," *International Journal of Research in Engineering, IT and Social Science (IJREISS)*, vol. 2, no. 11, pp. 2250-0588 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:16802396

[11] S. Kulkarni and P. Borde, "Analysis of orthogonal moments for recognition of handwritten MODI numerals," *VNSGU Journal of Science and Technology*, vol. 4, pp. 36-43, Jul. 2015.

[12] S. Anam and S. Gupta, "An approach for recognizing Modi Lipi using Ostu's binarization algorithm and Kohenen neural network," *International Journal of Computer Applications*, vol. 111, no. 2, pp. 29–34, Feb. 2015, doi: 10.5120/19511-1128.

[13] M. Deshmukh, M. Patil, and S. Kolhe, "Off-line handwritten Modi numerals recognition using chain code," *in Proceedings of the Third International Symposium on Women in Computing and Informatics, Kochi, India*, 2015, pp. 388–393, doi: 10.1145/2791405.2791419.

[14] S. Chandure and V. Inamdar, "Performance analysis of handwritten Devnagari and MODI Character Recognition system," *2016 International Conference on Computing, Analytics and Security Trends (CAST), Pune, India*, 2016, pp. 513-516, doi: 10.1109/CAST.2016.7915022.

[15] S. Gharde and R. Ramteke, "Recognition of characters in Indian MODI script," *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), Jalgaon, India*, 2016, pp. 236-240, doi: 10.1109/ICGTSPICC.2016.7955304

[16] R. Maurya and S. Maurya, "Recognition of a Medieval Indic-Modi Script

using Empirically Determined Heuristics in Hybrid Feature Space," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 2, pp. 136-142, Feb. 2018, doi: 10.26438/ijcse/v6i2.136142.

[17] S. Joseph, J. George, and S. Gaikwad, "Character Recognition of MODI Script Using Distance Classifier Algorithms," *in ICT Analysis and Applications*, S. Fong, N. Dey, and A. Joshi, Eds. Singapore: Springer Singapore, 2020, pp. 105–113.

[18] S. Joseph and J. George, "Handwritten Character Recognition of MODI Script using Convolutional Neural Network Based Feature Extraction Method and Support Vector Machine Classifier," 2020 *IEEE 5th International Conference on Signal and Image Processing (ICSIP), Nanjing, China*, 2020, pp. 32-36, doi: 10.1109/ICSIP49896.2020.9339435

[19] S. Sawant, A. Sharma, G. Suvarna, T. Tanna and S. Kulkarni, "Word Transcription of MODI Script to Devanagari Using Deep Neural Network," *2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA), Mumbai, India*, 2020, pp. 18-22, doi: 10.1109/CSCITA47329.2020.9137781

[20] J. Solley and J. George, "Efficient Handwritten Character Recognition of MODI Script Using Wavelet Transform and SVD," *in Data Science and Security*, D. S. Jat, S. Shukla, A. Unal, and D. K. Mishra, Eds. Singapore: Springer Singapore, 2021, pp. 227–233, doi: 10.1007/978-981-15-5309-7_24

[21] S. Joseph, A. Datta, O. Anto, S. Philip, and J. George, "OCR System Framework for MODI Scripts using Data Augmentation and Convolutional Neural Network," *in Data Science and Security*, D. S. Jat, S. Shukla, A. Unal, and D. K. Mishra, Eds. Singapore: Springer Singapore, 2021, pp. 201-209, doi: 10.1007/978-981-15-5309-7_21

[22] P. Tamhankar, K. Masalkar, and S. R. Kolhe, "Character Recognition of Offline Handwritten Marathi Documents Written in MODI Script Using Deep Learning Convolutional Neural Network Model," *in Recent Trends in Image Processing and Pattern Recognition, K. C. Santosh and B. Gawali, Eds. Singapore: Springer Singapore*, 2021, pp. 478–487, doi: 10.1007/978-981-16-0507-9_40

[23] K. Mahajan and N. Tajne, "An Ancient Indian Handwritten Script Character Recognition by Using Deep Learning Algorithm," *EFFLATOUNIA: Multidisciplinary Journal*, vol. 5, no. 2, pp. 123-134, Oct. 2021.

[24] S. Chandure and V. Inamdar, "Handwritten MODI Character Recognition Using Transfer Learning with Discriminant Feature Analysis," *IETE Journal of Research*, vol. 69, no. 5, pp. 2584-2594, 2023. doi: 10.1080/03772063.2021.1902867

[25] S. Kulkarni and P. Yannawar, "Recognition of Partial Handwritten MODI Characters Using Zoning," *in Recent Trends in Image Processing and Pattern Recognition, K. C. Santosh and B. Gawali, Eds. Singapore: Springer Singapore*, 2021, pp. 407–430, doi: 10.1007/978-981-16-0507-9_35.

[26] J. Kondhare, V. Yaduvanshi, R. Patil, and R. Kaldate, "Recognition of Handwritten Modi Digits and Characters by Using Deep Learning Algorithm," *International Journal of Emerging Technologies and Innovative Research*, vol. 9, no. 8, pp. 563-572, Aug. 2022. [Online]. Available: http://www.jetir.org/papers/JETIRFP06101.pdf.

[27] M. Ekbote, A. Jadhav, and D. Ambawade, "Implementing a Hybrid Deep Learning Approach to Achieve Classic Handwritten Alphanumeric MODI Recognition," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 12, no. 1, pp. 1-8, Oct. 2022, doi: 10.35940/ijeat.A3846.1012122.

[28] V. Pawar, D. Wadkar, S. Kashid, P. Prakare, V. More, and Prof. S. A. Babar, "MODI Lipi Handwritten Character Recognition Using CNN and Data Augmentation Techniques," *International Research Journal of Engineering and Technology (IRJET)*, vol. 09, no. 06, pp. 2345-2351, Jun. 2022.

[29] C. Chandankhede and R. Sachdeo, "Character Recognition using MODI script: Facts, Challenges and its future," *TEST Engineering and Management*, vol. 83, pp. 25389–25395, 2020.

[30] S. Jadhav, V. Inamdar, October 12, 2021, "Handwritten MODI Characters ", *IEEE Dataport*, doi: https://dx.doi.org/10.21227/z3gg-8b29.

[31] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA*, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas,* NV, USA, 2016, pp. 2818-2826, doi: 10.1109/CVPR.2016.308.

Kanchan Varpe

Kanchan Varpe is Research scholar in the Computer Engineering Department at Vishwakarma Institute of Information Technology, Pune, India. Awarded the Master of Engineering Degree from Savitribai Phule Pune University, Sinhgad College of Engineering, Wadgaon, Pune. Research interests include Natural Language Processing (NLP), Image Processing, Computer Networks and Machine Learning.

Dr. Sachin Sakhare

Dr. Sachin R. Sakhare is working as a Professor and Head of the Computer Engineering Department at Vishwakarma Institute of Information Technology, Pune, India. He has 27 Years of experience in engineering education. He is recognized as PhD guide by Savitribai Phule Pune University and currently guiding 8 PhD scholars. He is a life member of CSI, ISTE and IAEngg. He has Published 51 research communications in national, international journals and conferences, with around 393 citations and H-index 7. He has authored 6 books which is published by Springer Nature, CRC Press and IGI Global. He worked as a reviewer of journals published by Elsevier, Wiley, Hindawi, Springer, Inder science, and IETE. He worked as a reviewer for various conferences organized by IEEE, Springer, and ACM. He worked as a member of the Technical and Advisory Committees for various international conferences. Dr. Sachin has Delivered invited talks at various Conferences, FDP's and STTP's as well as to PG and PhD students. He has guided 26 PG students. He has filed and published 07 patents out of which 01 Indian, 03 Australian and 02 south African patents are granted.