Enhanced Recognition of Handwritten Marathi Compound Characters using CNN-SVM Hybrid Approach

Ashwini Patil*[1], Puneet Dwivedi [2]

[1] Department of Computer Science & Engineering, RKDF University, Bhopal (MP), India
[2] Department of Computer Science and Engineering, RKDF University, Bhopal (MP), India
Emails: ash82patil@gmail.com; puneet86.dwivedi@gmail.com

Abstract

This study presents a hybrid recognition system for multi-class compound Marathi characters, which addresses the problem of handwritten Marathi character recognition. The methodology efficiently bridges the gap between feature extraction and classification by integrating a Convolutional Neural Network (CNN) and Support Vector Machine (SVM). The first step is gathering and preprocessing a wide range of handwritten Marathi compound characters that are written in different styles. Using conventional supervised learning methods, the CNN is trained on this dataset, paying special attention to data augmentation and validation in order to reduce overfitting. High-level features taken from the final fully connected layer of the CNN are fed into an SVM classifier in the next step. By using these features in its training, the SVM improves prediction accuracy. For multi-class classification, the one-vs-all method is used. The hybrid CNN-SVM algorithm demonstrates its effectiveness in the crucial phases of feature extraction and classification by identifying handwritten compound Marathi characters with remarkable accuracy. Evaluation metrics, such as accuracy, precision, recall, F1-score, and confusion matrix analysis, are employed in the process of evaluating the effectiveness of the model. This assessment is carried out on a different testing dataset, offering a thorough examination of the model's functionality. The proposed algorithm demonstrates its superior performance and potential for improved character recognition by achieving training accuracy of 98.60% and validation accuracy of 97.69%. The development of handwriting recognition systems has benefited greatly from this research, especially when it comes to intricate scripts like Marathi. The suggested hybrid algorithm shows encouraging outcomes and has a great deal of potential for use in document processing, natural language comprehension, and character recognition in languages that use the Marathi script. Subsequent efforts will centre on refining the model and investigating ensemble methods to increase the robustness and accuracy of recognition.

Keywords: Character Recognition; Compound Marathi character; CNN-SVM algorithm; Convolutional Neural Networks (CNNs); HCR; OCR.

1. Introduction

The recognition of handwritten characters poses a persistent challenge within the realms of pattern recognition, machine learning, and computer vision. Its applications span various fields, encompassing document analysis, text recognition, and natural language understanding [1]. In multilingual societies like India, characterized by diverse scripts, the complexity of handwritten character recognition is amplified. Within this context, the Marathi script emerges as a prominent writing system, utilized for languages such as Hindi, Marathi, Nepali, and Sanskrit [2].
Distinguishing the Marathi script is its distinctive feature of compound characters, wherein primary consonant and vowel symbols intricately combine, resulting in a visually dynamic character set. Accurate recognition of these compound characters is crucial for applications such as automatic document processing and text-to-speech conversion across these languages [3]. Handwritten character recognition has evolved significantly, largely attributed to advancements in deep learning and Convolutional Neural Networks (CNNs). In the realm of scientific

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024

inquiry, a significant transformation has been ushered in by Convolutional Neural Networks (CNNs), as they have played a pivotal role in automating the extraction of features from unprocessed pixel data. This innovative approach has effectively eliminated the requirement for manual involvement in feature engineering, marking a notable advancement in the field. However, recognizing compound Marathi characters remains challenging due to the script's inherent complexity, diverse writing styles, and potentially confusing character shapes [4].
This study addresses the formidable task of identifying handwritten compound Marathi characters through the introduction of an innovative hybrid algorithm. This approach synergizes the strengths of Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs). CNNs excel at autonomously extracting hierarchical features from images, while SVMs prove particularly effective in tasks with binary outcomes. The integration of these two potent methods into a unified recognition system aims to leverage the CNN's proficiency in recognizing intricate visual patterns and the SVM's precision in fine classification.
The following are the key objectives of this study:

- Dataset Collection and Preprocessing: To collect a complete set of handwritten compound Marathi characters, including different writing styles, variations, and character combinations. Data Preprocessing techniques will be used to ensure the data is consistent and added to the dataset to help the model generalize better.
- CNN-Based Feature Extraction: To create a robust CNN architecture using handwritten images of compound Marathi characters to extract distinguishing features. The CNN will be taught using supervised learning, ensuring it does not overfit by validation and adding more data.
- SVM-Based Classification: To create an SVM classifier that uses the features extracted by the CNN to make more accurate character recognition predictions. The one-vs-all method will deal with the recognition task involving multiple classes.
- Hybrid CNN-SVM Algorithm: To integrate the CNN and SVM components into a hybrid algorithm that offers enhanced accuracy and robustness in recognizing compound Marathi characters.
- Model Evaluation and Application: To rigorously evaluate the performance of the proposed hybrid algorithm using various metrics and apply it to real-world scenarios, such as document processing and text recognition in multiple Indian languages.

The conducted research marks a noteworthy stride in enhancing the precision and dependability of handwritten compound Marathi character recognition. The amalgamation of deep learning and classical machine learning techniques, exemplified by a CNN-SVM hybrid algorithm, emerges as a promising approach to tackle the distinctive challenges posed by the script. The outcomes of this investigation are anticipated to yield extensive implications for applications related to the Marathi script and, more expansively, to propel the domain of handwritten character recognition in intricate scripts.
The paper is structured as follows: Chapter 2 provides a comprehensive background study, offering an in-depth exploration of the existing literature and research relevant to the recognition of handwritten Marathi compound characters. In Chapter 3, the focus shifts to the discussion of deep learning algorithms, laying the foundation for the subsequent proposed system. Chapter 4 introduces the core of the paper, presenting the details of the CNN and CNN-SVM hybrid approach designed for enhanced recognition. Following this, Chapter 5 presents the results obtained from the proposed system and conducts a thorough analysis of the outcomes. Finally, in Chapter 6, the paper concludes by summarizing the key findings, discussing their implications, and suggesting potential avenues for future research in handwritten Marathi character recognition.

2. Background Study

In the current investigation, the utilization of a deep learning algorithm referred to as Convolutional Neural Network (CNN) is evident in the recognition of online handwritten characters within Devanagari and Bangla scripts. The employed model is configured as a fully connected network, encompassing two convolution and pooling layers. This structural design not only highlights the utilization of CNN but also underscores its functionalities in the realms of feature extraction, size reduction, and classification within the confines of the network. The dataset utilized for this investigation comprised 1800 Devanagari characters and 10,000 primary Bangla characters. The application of CNNs facilitated the differentiation between various handwritten Devanagari scripts, as detailed in reference [5]. The recognition of Devanagari characters holds significance across numerous Indian languages, including Hindi, which extensively employs the Devanagari alphabet. The study, delving into the recognition of Devanagari characters through deep learning techniques, specifically CNNs, achieved a noteworthy accuracy rate of 95.6%. This successful recognition of handwritten Devanagari characters not only contributes to advancing the understanding of these scripts but also opens avenues for their rapid translation into other languages.
The utilization of Deep Learning Network Architecture allows for the recognition of handwritten Kannada characters [6]. The Devanagari character set comprises 92,000 images categorized into 46 classes. Conversely, the Kannada character set encompasses 188 classes containing 200 to 500 example images, resulting in 81,654 training

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024

photos and 9,401 testing images. The training of the VGG19 NET employed 123,654 data samples, while testing involved 9,401 samples across 188 classes, each with 40 to 100 samples. This rigorous training process achieved a precision level exceeding 90%. Subsequent verification after 10 tests showed that the VGG19 NET's accuracy stood at 73.51%, reflecting a decrease of 16.18%. Transfer learning was introduced to meet the demand for ample data in deep learning setups. Although deep learning algorithms are dependable and efficient in improving precision, their performance relies heavily on sizable training datasets containing precise labels.

In [7], the author introduces Machine Learning algorithms designed for distinguishing handwritten Odia letters. A dataset was generated using a model capable of discerning between offline handwritten Odia characters, both with and without background noise. The study aims to establish a narrative machine learning methodology by incorporating a blend of Naive Bayes and decision tables for the offline classification of handwritten Odia characters. The categorization process is executed within the framework of the Waikato Environment for Knowledge Analysis (WEKA). Noteworthy findings indicate that characters with clear representation demonstrate enhanced performance in both Naive Bayes and Decision Table classification techniques when compared to their distorted counterparts.

The Devanagari Handwriting Identification Using CNNs paper, authored by [8], discusses the identification of Devanagari handwritten characters. Unlike Latin-based languages, Devanagari characters do not employ capitalization, and they consist of consonants and modifiers. The approach in the article is based on a set of 29 consonants with a single alteration. The writer assembled a set of 29 Devanagari consonants without including a Shirorekha (header line) above them. This dataset consists of 34,604 handwritten images, and advanced machine-learning methods were utilized to recognize letters and extract features from the photographs. Specifically, a Deep CNN (DCNN) was employed to recognize and categorize features within the input images. This method utilizes convolutional layers to extract higher-level information, achieving an impressive model accuracy of 99.65%.

In the domain of ancient Devanagari character recognition, a proposed deep learning method known as Deep Net Devanagari has been recommended by researchers [9]. They endorse the use of this deep learning model not only as a feature extractor but also as a classifier, with the objective of distinguishing 33 distinct categories of significant characters found in antique Devanagari manuscripts. An empirical investigation was conducted, leveraging a dataset comprising 5484 characters. Numerous studies in the field affirm that the utilization of Convolutional Neural Networks (CNNs) as feature extractors enhances accuracy when compared to other state-of-the-art methods. The accuracy achieved in the identification of ancient Devanagari characters using the methodology outlined in this study reached a noteworthy 93.73%.

In the domain of density-based semi-automatic labeling, the underpinning of handwritten character recognition relies on multi-feature representations [10]. The evaluation of samples across diverse feature spaces is systematically executed using the nearest neighbor graph in this methodology. At the initiation of each iteration, a pertinent unlabeled sample is randomly chosen by an expert from the region with the most substantial percentage of unlabeled samples. Ensuring safety through sample density and multiview, the manually annotated label is then extended to adjacent samples. The labeling procedure persists until all samples necessitating labeling have been processed. The proposed labeling approach demonstrates an impressive accuracy rate of 93.73% when appraised with datasets from MNIST, Devanagari, Thai, and Lana Dhamma. The findings underscore the superior efficacy of the recommended strategy over prevailing labeling methodologies, attaining the highest accuracy and displaying adeptness in managing unconventional samples and alphabets containing common letters. Furthermore, the classifier's accuracy in categorization remains on par with that of a classifier trained using the semi-automatically generated training dataset when confronted with real-world data.

In the exploration of distinguishing between handwritten and printed Marathi numerals, a stacked ensemble neural network showcases its proficiency [11]. Utilizing a customized meta-learning approach tailored for the recognition of Marathi handwritten digits, the employed method entails the stacking of pre-trained base pipelines to formulate multi-headed meta-learning classifiers that yield the intended target labels. The approach exceeds traditional ensemble techniques by taking into account the maximum and weighted contributions from each pipeline. Diverging from the typical averaging strategy, this methodology consolidates vital pipelines to augment overall effectiveness. The stacked ensemble meta-learning classifiers exhibit noteworthy performance within this particular context. The study includes a comprehensive performance assessment and analysis conducted on handwritten Marathi numerals, consistently illustrating the superior efficacy of this approach over the recommended alternatives.

In the investigation conducted for the SCI-indexed Journal, the utilization of a CNN-RNN network, coupled with the sequence-to-sequence methodology, proved to be effective in the recognition of offline handwritten words within the framework of a successful model [12]. The research implemented the H2TR system, which employs deep neural networks and the sequence-to-sequence (Seq2Seq) approach for the interpretation of handwritten text. This integrated model incorporates crucial components from both CNNs and recurrent neural networks, including an extended short-term memory network (RNN). CNNs play a pivotal role in capturing features from the handwritten image. Subsequently, employing a sequence-to-sequence modeling technique, the extracted features are transmitted to an RNN-LSTM for deciphering the letter sequence and embedding visual details into the

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024

handwritten image. The model's performance is validated through assessments using the IAM and RIMES handwritten datasets, resulting in competitive word and letter accuracy outcomes.

In the study conducted for the SCI indexed Journal, the recognition of manually written Devanagari characters was explored through the utilization of Convolutional Neural Networks (CNNs) and transfer learning [13]. The research focused on a comparative evaluation of VGG16 and DenseNet121's ability to perform transfer learning for the recognition of handwritten Devanagari letters using various training approaches. The models were trained using different methods, and their outcomes were systematically compared against those derived from alternative approaches. Notably, the findings revealed that DenseNet121 exhibited superior performance compared to other trained models and additional learning techniques, particularly when employing a deep fine-tuning strategy. The study also involved the adjustment of hyperparameters, such as batch size and learning rate, to further enhance the learning efficiency of the models.

In the study conducted by [14], an effective approach utilizing Artificial Neural Networks (ANNs) for the recognition of handwritten Devanagari characters is presented. The methodology entails decoding the intricate curves inherent in handwritten Devanagari characters, capturing the evolving form of each character through a segmentation process. This segmentation technique involves extracting Histograms of Oriented Gradient (HOG) features from distinct partitions of the image. These HOG features are then consolidated into a comprehensive feature vector, serving as the training data for the neural network. The reported outcomes of the proposed method indicate significant accomplishments in character recognition. Noteworthy results include an average recognition accuracy of 97.06% and a peak accuracy of 99.2% when identifying diverse handwritten Devanagari characters, contingent upon the availability of sufficient training data. This methodology showcases particular utility in situations where individuals with visual impairments seek to convert handwritten text into audible content.

In the realm of recognizing handwritten Devanagari characters, CNNs have been employed [15]. In pattern recognition tasks, the successful elucidation of patterns is contingent upon the adept execution of categorization and feature extraction steps. The advent of deep learning has notably addressed the challenge of feature extraction, offering a streamlined approach for software engineers engaged in such tasks. Over the course of time, deep learning has progressively superseded traditional pattern recognition methods. Particularly in intricate tasks like character recognition, where substantial datasets prone to errors are imperative, deep learning emerges as the most efficacious method for addressing these complexities.

In the exploration of handwritten word recognition in Devanagari and Bengali scripts, a novel approach is employed, as detailed in [16]. The adopted methodology seamlessly incorporates horizontal zoning while harnessing Recurrent Neural Networks (RNNs), with a specific focus on the utilization of the Bidirectional Long-Short Term Memory (BLSTM) technique in conjunction with the Long-Short Term Memory (LSTM) model. The overarching objective is to discern between cursive and non-cursive words through the strategic horizontal segmentation of individual letters into upper, middle, and lower sections. This approach is meticulously designed to mitigate variations in stroke order within the text, thereby enhancing the system's capability to accurately differentiate between distinct writing styles. Subsequently, the main word segments undergo further dissection into individual components, allowing for a detailed analysis of core letters within each phrase. This analytical process provides valuable insights into the structural and directional properties of each designated zone. The stroke characteristics identified are then scrutinized using variants of RNN, LSTM, and BLSTM. In contrast to prevailing word recognition systems that adopt a word-based approach, the proposed system adopts a more straightforward stroke-based methodology for class labeling. To assess the efficacy of this innovative approach and perform a thorough performance comparison, extensive datasets were rigorously evaluated using RNN and Hidden Markov Models (HMM). Results indicate that the RNN-driven system outperforms previously utilized HMM-based methods, achieving remarkable accuracy levels of 99.50% in Bengali script and 95.24% in Devanagari script.

In the study, a Convolutional Neural Network (CNN) was utilized to transform manually crafted Devanagari script into an editable Word document [17]. A comparative analysis revealed that, in contrast, English text recognition exhibited an accuracy rate of 78.4%, with a perplexity frequency of 4.5% and an 18% likelihood of confusion. Conversely, the Devanagari recognition process achieved a higher accuracy rate of 89%. Extensive testing of the Hierarchical Optical Character Recognition (HOCR) demonstrated a consistent accuracy range of 94% to 97% for most cases. The research specifically employed a CNN for the conversion of handwritten Devanagari script into editable text. This transformative process is particularly crucial for historical materials, such as manuscripts, facilitating their transition from old handwritten characters to a machine-editable format, thereby ensuring preservation and enhanced accessibility.

In the conducted research, a substantial dataset and a diverse set of neural network techniques are utilized to elevate accuracy levels and uphold a comprehensive historical data repository. The study showcases the effectiveness of a specific convolutional neural network (CNN) in the domain of handwriting recognition, as evidenced by a previous work [18]. The computational efficiency and consistent performance of this CNN are emphasized throughout the investigation. The incorporation of machine learning with the CNN proves instrumental in the identification of handwritten numbers. The project's foundation rests on a dataset and resources sourced from Kaggle, with essential libraries such as NumPy, Pandas, TensorFlow, and Keras employed to ensure the efficient

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024

operation of the model. These libraries serve as fundamental components for a significant portion of the tasks undertaken in the study. The Kaggle dataset comprises approximately 70,000 images portraying handwritten digits spanning from 0 to 9, meticulously organized into distinct training and testing sections. Each image is represented as a 28x28 matrix, with grayscale pixel values structured in rows and columns.

In the study conducted by N. Sharma [19], a method based on directional chain code feature extraction was utilized for the recognition of handwritten Devanagari characters. The character samples' bounding box was partitioned into blocks, and from each block, 64-D direction chain code features were computed. Subsequently, a quadratic classifier was employed to identify 11,270 samples, resulting in a reported accuracy of 80.36%.

In the investigation conducted by M. Hanmandlu [20], a recognition approach based on fuzzy modeling was employed. This approach involved the extraction of features using a box methodology, which partitioned characters into a grid of 24 cells (6 × 4). The computation of normalized vector distances for each box was a crucial step, with the exclusion of empty cells from the calculations. The implementation of a reuse policy significantly contributed to accelerating the learning process, ultimately achieving an accuracy rate of 90.65% across 4750 samples.

In a study conducted by Kumar [21], an examination of 25,000 handwritten Devanagari characters involved testing five distinct features, resulting in a reported accuracy of 94.1%. Notably, the gradient feature demonstrated superior performance when paired with the SVM classifier, whereas the Kirsch directional edges feature proved to be the least effective. Additionally, a novel feature was introduced, involving the computation of total distance in four directions.

In a separate investigation, Pal [22] employed a mean filter iteratively four times, followed by the extraction of direction gradient features, which were further reduced using a Gaussian filter. The application of a modified quadratic classifier on a dataset comprising 36,172 samples yielded a noteworthy accuracy of 94.24%, as confirmed through cross-validation. Furthermore, Pal [26] expanded upon this work by incorporating SVM and MIL classifiers, achieving recognition accuracies of 95.13% and 95.19%, respectively.

Pandey et al. [23] employed Convolutional Neural Networks for recognizing Devanagari characters, achieving 95.6% accuracy with a model consisting of two convolutional layers, two max-pooling layers, and two fully-connected layers. A dataset of 3,78,951 images was used, potentially leading to overfitting.

Narang et al. [24] achieved 93.73% accuracy with a model using 3 convolutional layers on a dataset of 5,484 characters. The authors suggested three-phase segmentation for improved accuracy.

Dokare et al. [25] compared four CNN architectures on Devanagari and Nepali Handwritten Character Datasets, achieving 97.56% accuracy. Bisht et al. (2021) proposed a double CNN architecture with 95.97% accuracy, also comparing SVM+HOG and single CNN architecture.

Shalaka Deore [26] proposed a fine-tuned Deep CNN for Devanagari Handwritten Character Recognition, achieving 96.55% testing accuracy in a two-stage approach with a dataset of 5800 isolated images.

Bhagat et al. [27] used deep convolutional neural networks to recognize Devanagari characters, achieving 99.65% accuracy with an offline-based model consisting of four convolutional layers and two max-pooling layers. Using a dropout layer aimed to prevent overfitting, but challenges remained due to close character resemblance.

The literature survey underscores the extensive exploration of methodologies for recognizing handwritten characters in Indian scripts, particularly Devanagari. Various approaches, including Convolutional Neural Networks (CNNs), fuzzy models, and directional chain codes, have been adeptly applied for feature extraction and classification. Diverse datasets, ranging in size and complexity, have been utilized to train and evaluate these models, resulting in reported accuracies from 80.36% to an impressive 99.65%. The studies not only focus on contemporary Devanagari characters but also delve into ancient scripts, showcasing the versatility of deep learning approaches in preserving and deciphering historical manuscripts.

In the identified research domain, a discernible research gap is evident concerning the specific recognition of compound Devanagari characters—those intricately formed through the combination of multiple primary characters. The inherent complexity of compound characters poses a unique challenge, characterized by intricate structures and diverse combinations. While existing studies have touched upon the recognition of characters in isolation, there exists a limited exploration of their recognition in compound forms. A comprehensive investigation into the accuracy and challenges associated with recognizing compound characters is deemed imperative for real-world applications, particularly in text processing for natural language understanding. Future research endeavors are encouraged to prioritize the development of models adept at accurately deciphering compound Devanagari characters, thereby fostering a more practical and holistic approach to handwritten script recognition. Such a focused extension in research emphasis is anticipated to significantly enhance the capabilities of these models in navigating the intricacies of real-world data, thereby contributing to advancements in the broader field of character recognition.

3. Deep Learning Algorithm

In the presented methodology, the researchers utilized two distinct deep learning algorithms, specifically CNN and CNN-SVM, to categorize the 40 classes of Marathi compound characters. This section delivers an exhaustive

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024

explanation of the architectures of CNN and the hybrid SVM-CNN, elucidating their respective contributions to the character classification procedure.

3.1 CNN

CNNs are a specific neural network with exceptional performance in various image processing and classification tasks [28]. CNNs are multi-layered feed-forward neural networks with convolutional layers. CNNs are built using programmable weights, parameters, and biases for their filters, kernels, or neurons, which comprise the CNN [20]. Every filter receives specific inputs, conducts convolution, and may or may not perform non-linearity. Convolutional layers, Rectified Linear Unit (ReLU), Fully Connected, Pooling, and makeup CNN's structure.
The architecture of the CNN algorithm is shown in Fig.1.

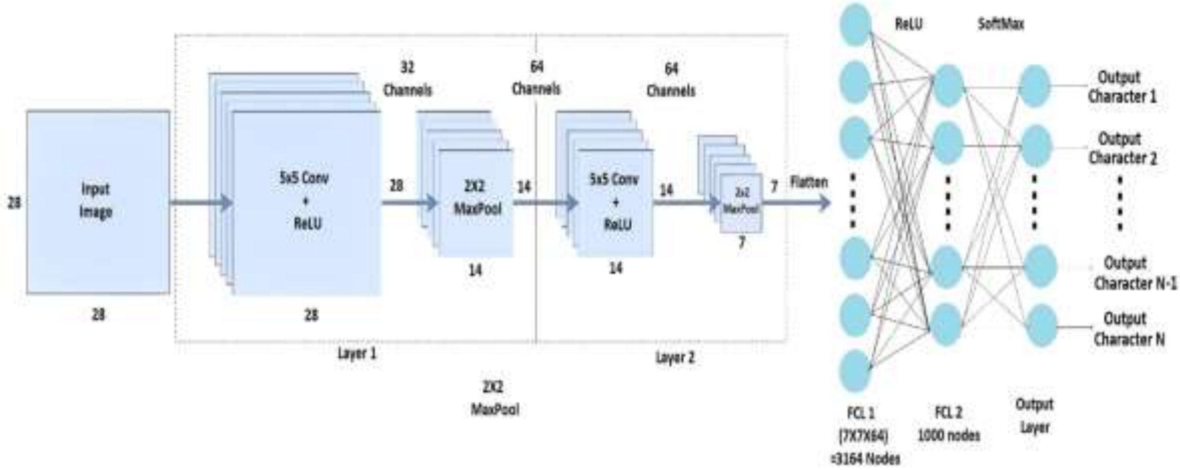


Figure 1: Architecture of CNN for Marathi compound character

Each block of CNN architecture is explained below.

- Convolutional Layers: These layers are in charge of learning from the input data and extracting its features. They apply a set of learnable filters or kernels to the input image, which performs convolution operations by sliding over the image and computing dot products at each location. The result is a feature map set that captures different patterns and structures in the image [30].
- Activation Function: After each convolutional operation, a layer-by-layer activation function is applied in the network to give it a non-linear quality. In CNNs, the activation function known as Rectified Linear Unit (ReLU) is utilized most of the time. This activation function keeps all positive values unaltered while putting all negative values to zero [29].
- Pooling Layers: The layering of the feature maps can be used to minimize their spatial dimensions. These maps still have their essential information content. In a typical pooling process called max pooling, the feature map is divided into non-overlapping areas, and the most significant value inside each zone is kept while the other values are discarded. This facilitates a decrease in parameters and limits overfitting [31].
- Fully Connected Layers: After many convolutional and pooling layers, the spatial dimensions of the feature maps are turned into a vector and fed into one or more fully connected layers. Like those in a typical neural network, these layers use the learned characteristics to make predictions [29].
- Softmax Layer: A softmax layer is frequently added at the network's end for classification tasks to generate probability scores for each class. The outputs of the preceding layer are transformed into a probability distribution over the classes using the softmax function [32].
- Loss Function: The loss function quantifies the disparity between the expected results and the actual labels provided as ground truth. It measures the network's performance during training and guides the learning process. Cross-entropy loss is one of the standard loss functions for classification problems.
- Optimization Algorithm: Optimization techniques are used to train CNNs, updating the network's parameters depending on the gradients of the loss function. Adam and RMSprop optimization techniques and stochastic gradient descent (SGD) are frequently employed [33].

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024

3.2 Hybrid CNN-SVM

The CNN-SVM algorithm combines CNNs and SVMs commonly used for image classification tasks. The CNN portion of the algorithm extracts features from the photos, and the SVM portion uses those characteristics to categorize the images. Fig. 2 depicts the architecture of the CNN-SVM algorithm.

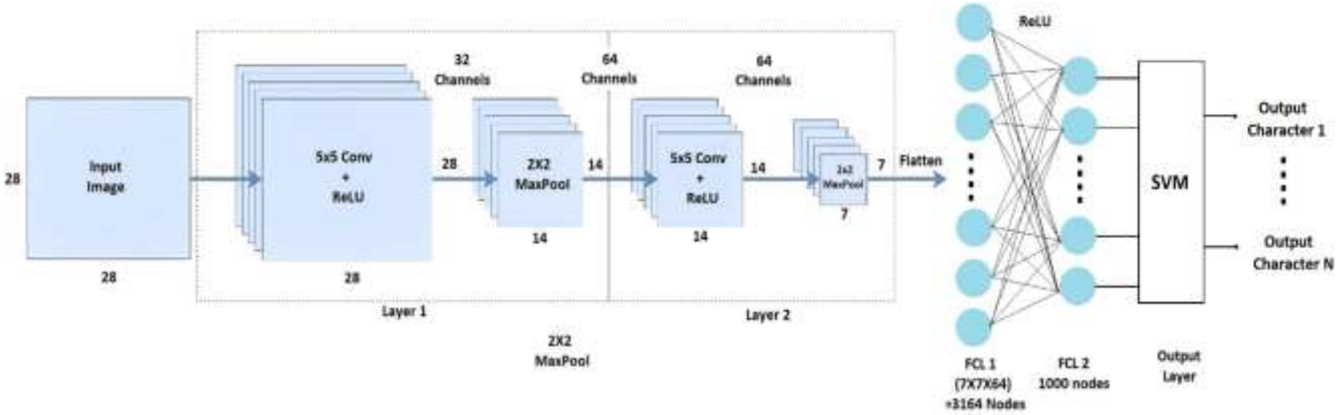


Figure 2: CNN-SVM architecture for Marathi compound character classification

Step-by-step guide on how to implement the CNN-SVM algorithm for image classification:

- Data preparation: Collect a dataset of images and divide it into training and testing sets. Ensure that the images are in a suitable format, such as JPEG or PNG, and are the same size.
- Feature extraction using CNNs: Train a CNN on the training set to extract features from the images. This involves feeding the images through the CNN and extracting the output from one of the intermediate layers of the network, typically just before the fully connected layers. The output of this layer is a feature vector that represents each image.
- SVM classification: Use the feature vectors from the CNN to train an SVM on the training set. The SVM can be trained using the sci-kit-learn library in Python, with the feature vectors as input and the corresponding labels as output.
- Testing and evaluation: Utilize the SVM to make predictions about the labels of the images included in the testing set and then assess the algorithm's effectiveness based on measures like accuracy, precision, recall, and F1 score.

Here are some additional considerations when using the CNN-SVM algorithm for Marathi compound character image classification:

- Preprocessing: It is often beneficial to preprocess the images before feeding them into the CNN, such as by normalizing the pixel values or applying data augmentation techniques to increase the size of the training set.
- CNN architecture: The choice of CNN architecture can significantly impact the algorithm's performance.
- SVM hyperparameters: The performance of the SVM can be influenced by its hyperparameters, such as the kernel type, regularization parameter, and gamma parameter. These should be tuned using cross-validation on the training set to obtain the best performance.
- Class imbalance: If the dataset has a class imbalance, with some classes having fewer examples than others, it may be necessary to use techniques such as oversampling or undersampling to balance the classes. Alternatively, one can use techniques such as weighted loss or focal loss during training to give more weight to the underrepresented classes.

4. Proposed System

The block diagram of the proposed system is shown in Fig. 3. It consists of an Input dataset, Preprocessing, dataset splitting, training and classification.

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024

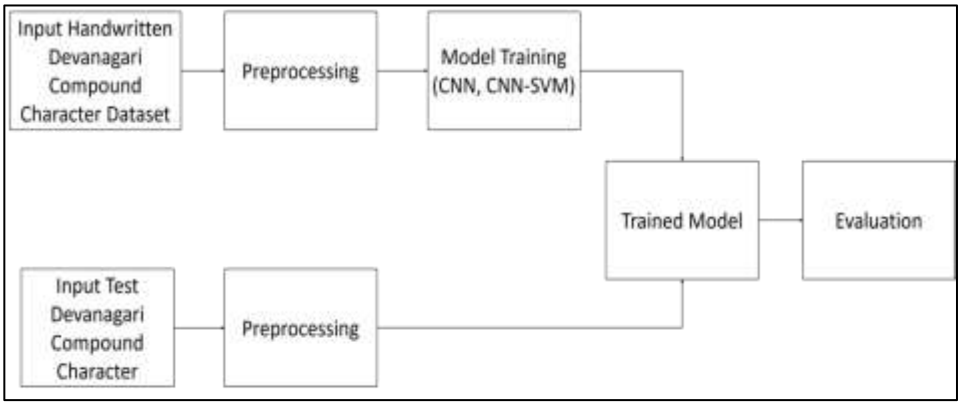


Figure 3: Block diagram of the proposed system

4.1 Dataset Preparation

To achieve the variation in writing styles, the Marathi handwritten multilevel compound characters are collected from 21 writers ages 13 to 66, including males and females. The 40 Marathi handwritten multilevel compound characters are considered for this work, as shown in Fig.4.

Figure 4: The 40 classes of Marathi handwritten multilevel compound characters

The data collection sheet of A4 size with 10 rows and 7 columns, i.e., 77 cells on each page, is designed and used to collect Marathi multilevel compound characters. Each writer should write one character on the entire page 77 times. The dataset of 1617 characters each is collected for 40 such selected characters, i.e., the total number of collected characters is 64680. Finally, a database of 64680 isolated handwritten Marathi multilevel compound characters of 40 classes, written by 21 people of different age groups, is ready for experiments. Figure 4.9 shows the two sample A4 sheets for Handwritten multilevel compound characters.

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024

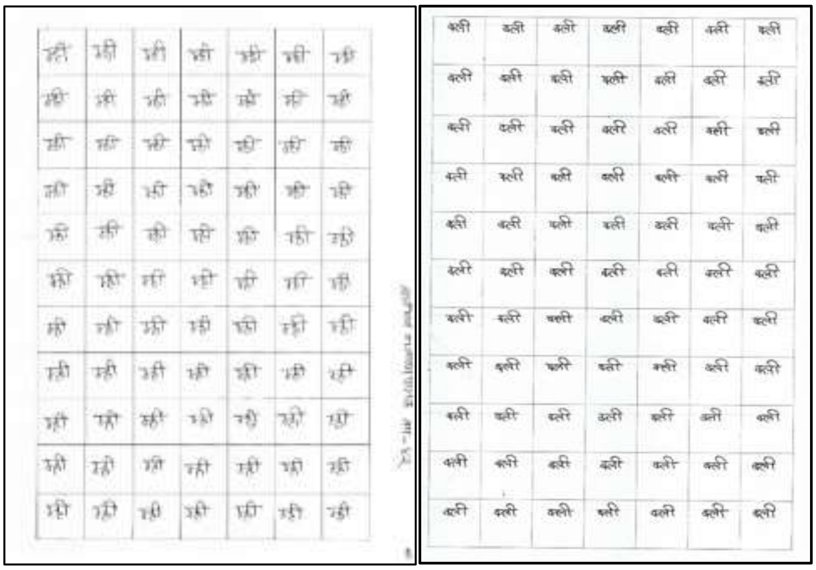


Figure 5: Sample A4 sheets for Handwritten multilevel compound characters

4.2 Data preprocessing

The Samsung M12 smartphone camera has digitally captured all 840 A4-sized papers collected from 21 writers. Each digitized sample sheet picture is 1024 by 786 pixels in size. Digitization has made it conceivable and straightforward to recognize delicate handwritten characters. The digitized dataset was converted from colour to grayscale images in the preparation stage. The data quantity and computing costs were lowered by converting coloured images to grayscale, positively impacting learning and recognition rates. All greyscale images have undergone erosion as part of the dataset preparation. Eroding images makes it easier to extract meaningful information from digital data. The collection of eroded images has been cut into character images, and 40 composite character images have been obtained from each sample eroded image. The dataset may be sorted classwise with the aid of this technique. Class names dataset sorting on cropped images has been done as the next stage in dataset preparation. Cropped images for each of the 40 classes of Marathi compound characters have been organized into the corresponding folders for that class. The dataset as a whole has 64680 images as a consequence. With the help of this process, differentiation has been added between each image of the several classes of Marathi compound characters.

4.3 Training and Classification

This approach uses two algorithms to train and classify the Marathi compound characters. The architecture of CNN and CNN-SVM algorithms are present in Fig.1 and Fig.2. The model summary of the CNN model is presented in Table 1.

Table 1: Model Summary of CNN Algorithm

Layer (type)	**Output Shape**	**Number of Parameters**
conv2d (Conv2D)	(None, 223, 223, 256)	3,328
activation (Activation)	(None, 223, 223, 256)	0
max_pooling2d (MaxPooling2D)	(None, 111, 111, 256)	0
conv2d_1 (Conv2D)	(None, 110, 110, 128)	131,200
activation_1 (Activation)	(None, 110, 110, 128)	0
max_pooling2d_1 (MaxPooling2D)	(None, 55, 55, 128)	0
flatten (Flatten)	(None, 387,200)	0
dense (Dense)	(None, 64)	24,780,864
activation_2 (Activation)	(None, 64)	0
Dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 40)	2,600

The provided model summary outlines a CNN architecture designed for a classification task with 40 output classes. The model starts with a Conv2D layer with 256 filters, which processes input data of spatial dimensions 223x223 pixels. An activation layer follows to introduce non-linearity, and then a MaxPooling2D layer reduces the spatial dimensions to 111x111 while preserving important features.

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024

The architecture continues with another Conv2D layer employing 128 filters and another activation layer. Subsequently, a second MaxPooling2D layer reduces the spatial dimensions further to 55x55. After that, the data is prepared for the fully connected layers by flattening the 3D tensor into a 1D vector.

In the initial Dense (fully connected) layer, there are 64 neurons, contributing significantly to the model with an extensive parameter count of 24,780,864. Following this, an activation layer is introduced to introduce non-linearity, and a Dropout layer is implemented in the training process to mitigate overfitting by selectively deactivating certain neurons. Subsequently, a second Dense layer is incorporated with 40 neurons, presumably corresponding to the number of output classes in the classification task.

In the alternative approach, a hybrid CNN-SVM algorithm is employed and trained. This hybrid classifier amalgamates the capabilities of both CNNs and SVMs, enhancing the overall performance of the machine learning model, particularly in tasks involving image classification. The initial phase of the hybrid classifier utilizes a CNN for feature extraction. CNNs prove advantageous in processing image data, as they autonomously learn hierarchical and discriminative features from the raw pixel values. The CNN consists of convolutional layers for feature extraction and pooling layers for spatial reduction, collectively contributing to the generation of high-level feature representations derived from the input images.

After feature extraction, the CNN typically ends with one or more fully connected layers, followed by an output layer. In the hybrid approach, the output of the last layer or a specific intermediate layer is flattened and vectorized. This step transforms the 2D or 3D feature maps into a 1D vector representation.

Instead of using a fully connected layer followed by softmax activation for classification, the flattened feature vectors are passed to an SVM classifier. SVMs are effective for jobs requiring binary and multiple-class classification. They try to determine which hyperplane in the feature space best divides the various classes. In the case of multi-class classification, multiple SVMs (one per class) can be trained, or a multi-class SVM can be used. The model summary of the CNN-SVM algorithm used for the proposed Marathi compound character recognition is shown in Table 2.

Table 2: Model Summary of CNN-SVM Algorithm

Layer (type)	**Output Shape**	**Number of Parameters**
conv2d (Conv2D)	(None, 223, 223, 256)	3328
activation (Activation)	(None, 223, 223, 256)	0
max_pooling2d (MaxPooling2D)	(None, 111, 111, 256)	0
conv2d_1 (Conv2D)	(None, 110, 110, 128)	131200
activation_1 (Activation)	(None, 110, 110, 128)	0
max_pooling2d_1 (MaxPooling2D)	(None, 55, 55, 128)	0
flatten (Flatten)	(None, 387200)	0
dense (Dense)	(None, 64)	24780864
dense_1 (Dense)	(None, 1)	65

Table 2 outlines the architecture of a CNN-SVM classifier, forming a CNN-SVM hybrid algorithm for a Marathi compound character recognition task. The CNN, a fundamental architecture component, begins with a Convolutional Layer (conv2d) that applies 2D convolution to the input image. This operation generates an output tensor with dimensions (None, 223, 223, 256), indicating that it detects 256 different features or patterns in the image. The Activation Layer (activation) follows, applying the Rectified Linear Unit (ReLU) activation function element-wise to the previous layer's output without introducing any new parameters. Subsequently, the MaxPooling Layer (max_pooling2d) performs 2D max-pooling, reducing the spatial dimensions to (None, 111, 111, 256) while retaining essential features. The network continues with another Convolutional Layer (conv2d_1), which identifies 128 distinct features, and an Activation Layer (activation_1). Another MaxPooling Layer (max_pooling2d_1) reduces the dimensions to (None, 55, 55, 128). Afterwards, a Flatten Layer (flatten) transforms the data into a 1D vector for further processing. The architecture concludes with two Fully Connected Layers (dense) where the first has 64 neurons, connecting densely with the previous layer and contributing significantly to the model's 24,780,864 parameters. The final Dense Layer (dense_1) produces a single output. The CNN-SVM hybrid approach leverages the CNN for feature extraction and the SVM for classification. The total number of parameters impacts the model's complexity and computational requirements, making this architecture well-suited for intricate recognition tasks but necessitating substantial computational resources for training.

4.4 Experimental Setup

In the pursuit of the research outcomes, the investigation involved simulating Marathi compound characters utilizing a Tesla K80 GPU, which was virtually accessible on Google Colab. The generation of a bespoke dataset comprised handwritten Marathi compound characters. The experiment utilized hardware with memory specifications, boasting RAM and ROM sizes of 12.69 GB and 107.72 GB, respectively. Moreover, the construction of the proposed CNN and CNN-SVM model architectures was undertaken using Keras with a TensorFlow backend, leveraging the advantageous tools available in the Python programming language.

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024

In the exploration of the proposed CNN-SVM model on the Marathi compound handwritten character dataset, the investigation involves a meticulous examination of trade-offs in performance among various optimizers, including stochastic gradient descent (SGD) and Adam. The selection of an appropriate deep optimizer plays a crucial role in achieving optimal accuracy rates in the simulated scenarios. Notably, higher accuracy rates have been observed with the utilization of the Adam optimizer compared to alternative optimization methods.

In the evaluation of the model's performance, a rigorous assessment is conducted through precision (P), recall (R), F1 score, and accuracy (A). Within the domain of binary classification, the samples are stratified into four categories: true positive (TP), false positive (FP), true negative (TN), and false negative (FN), based on their actual and expected classifications. The classification outcomes, as formulated in Equations (1), (2), (3), and (4), provide meaningful insights into Precision, Recall, F1 Score, and Accuracy. Together, these metrics contribute to a comprehensive evaluation of the model's efficacy.

$$Precision\ (P) = \frac{TP}{TP+FP} \tag{1}$$

$$Recall\ (R) = \frac{TP}{TP+FN} \tag{2}$$

$$F1Score = 2 * \frac{P*R}{P+R} \tag{3}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

In this context, TP stands for instances accurately classified as belonging to a specific class, while TN denotes instances correctly identified as not part of any class. FP refers to instances mistakenly categorized as belonging to a different class when they do not belong to that class. FN indicates instances that genuinely belong to a particular class but were erroneously classified as not belonging.

5. Result and Analysis

An in-depth discussion of the findings of the proposed CNN and CNN-SVM models will be provided in this part. In order to comprehensively evaluate the effectiveness of the CNN model, this section will include a comparison of its precision, recall, F score, and accuracy against those of the CNN-SVM model.

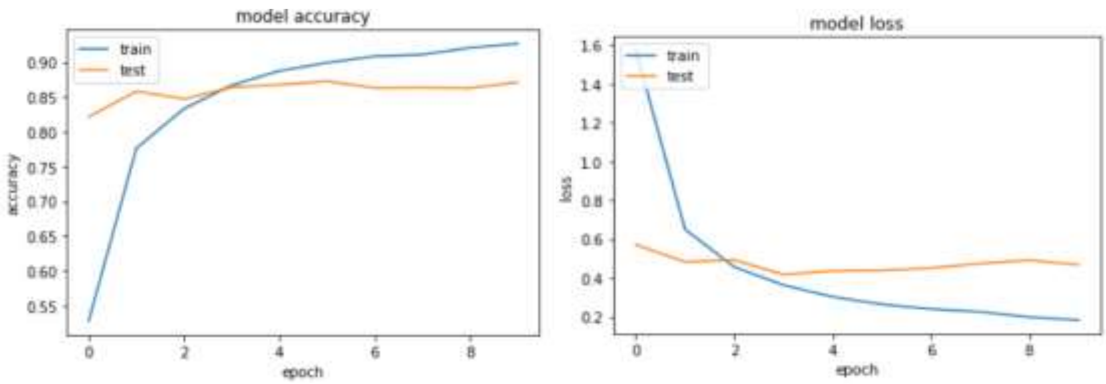
5.1 Results of the CNN algorithm

The results of the CNN algorithm for Marathi compound character recognition are presented In Table 4, and the training progress graph is shown in Fig.6.

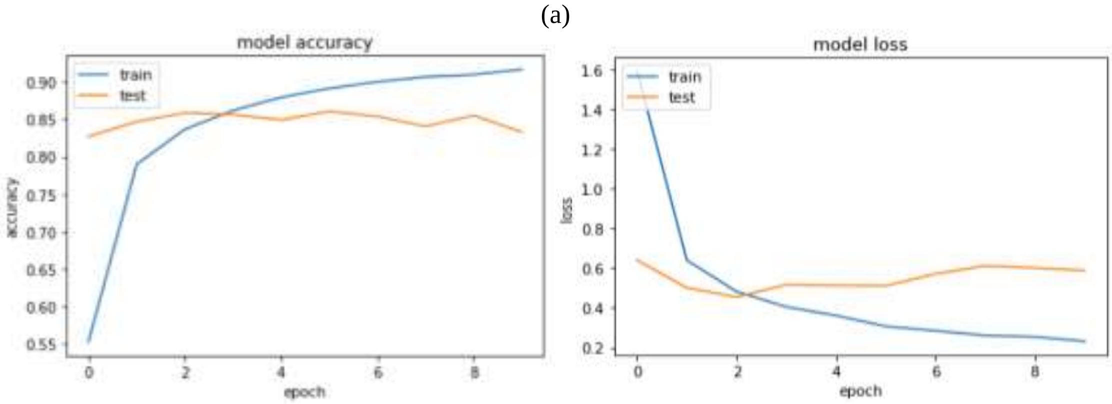
Table 3: Performance of CNN algorithm for Marathi compound character recognition with different optimizers and number of epochs

Number of Epochs	Optimizer	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Execution Time
50	Adam	61.76	1.1701	75.71	0.7915	10645.9120
50	RmsProp	71.41	0.9618	83.37	0.5700	10848.8925
40	Adam	96.09	0.0791	87.48	0.5919	4960.1768
40	RmsProp	95.22	0.1186	83.65	0.7429	9472.1504
30	Adam	95.75	0.0866	89.04	0.4776	7664.0690
30	RmsProp	95.87	0.0925	88.38	0.5160	3948.5040
20	Adam	92.61	0.1793	84.50	0.7353	22297.0046
20	RmsProp	93.35	0.1723	84.73	0.6775	2638.1340
10	Adam	92.68	0.1832	87.12	0.4679	1236.9991
10	RmsProp	91.65	0.2323	83.29	0.5883	1321.6109

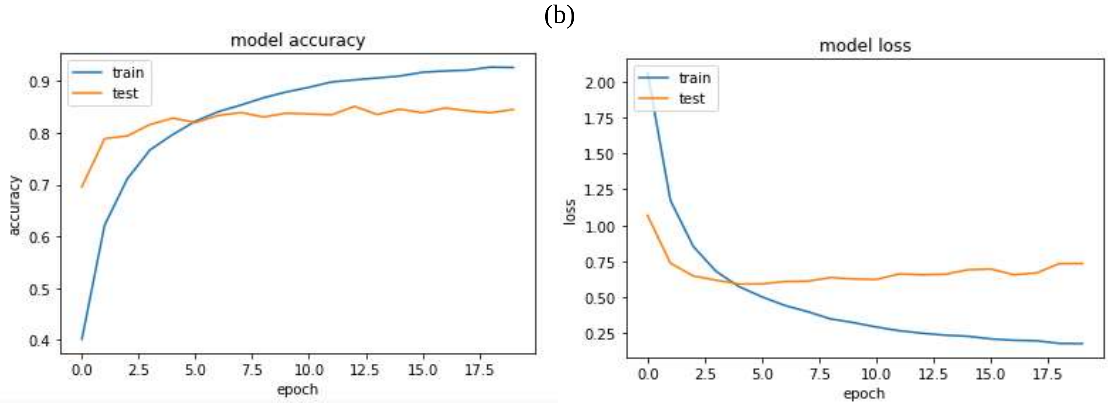
Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024



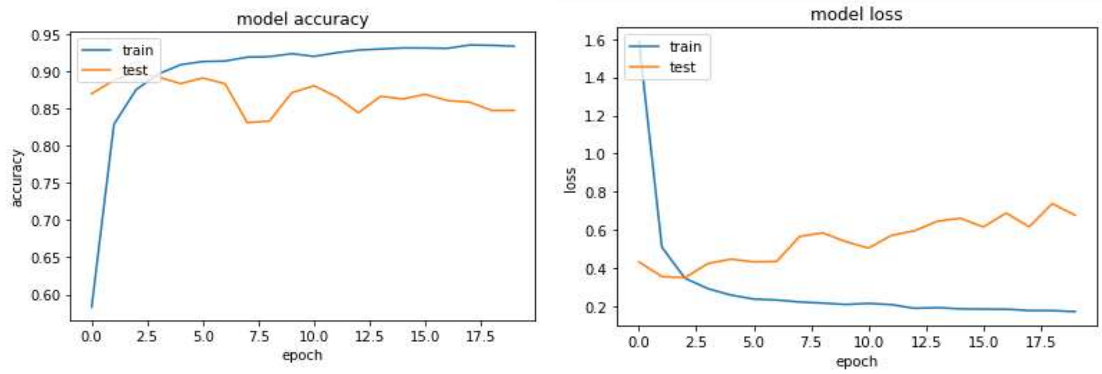
(a)



(b)

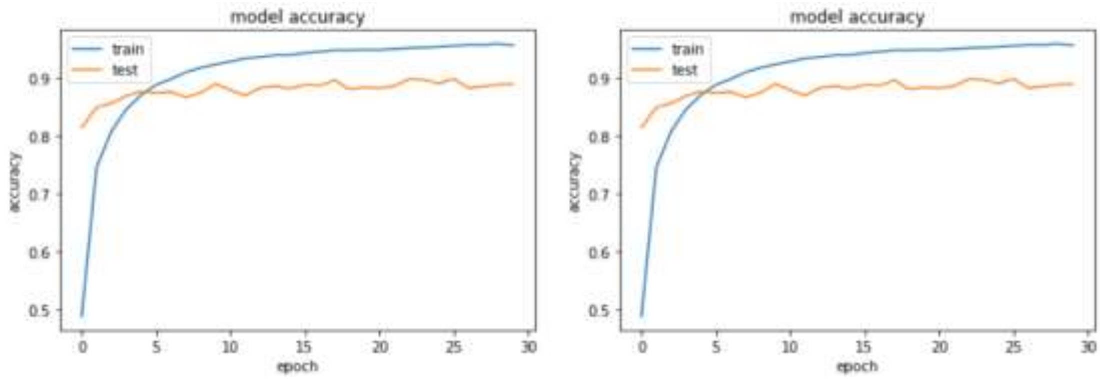


(c)

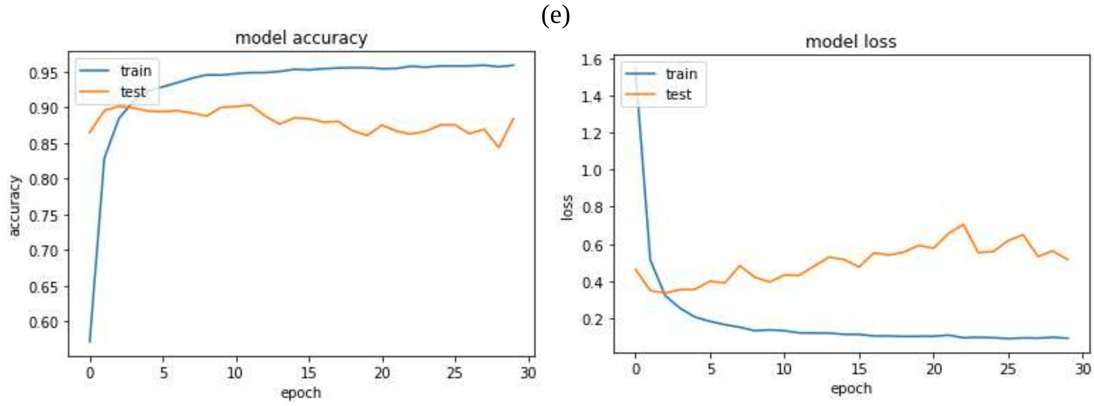


(d)

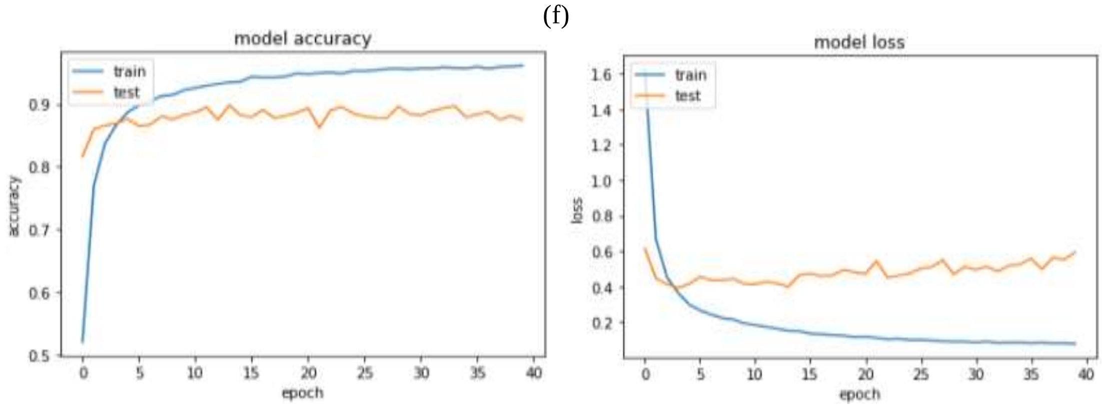
Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024



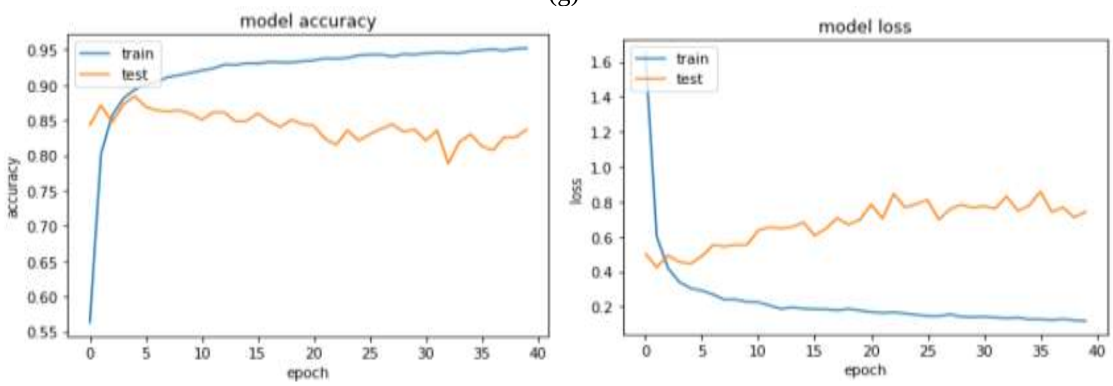
(e)



(f)

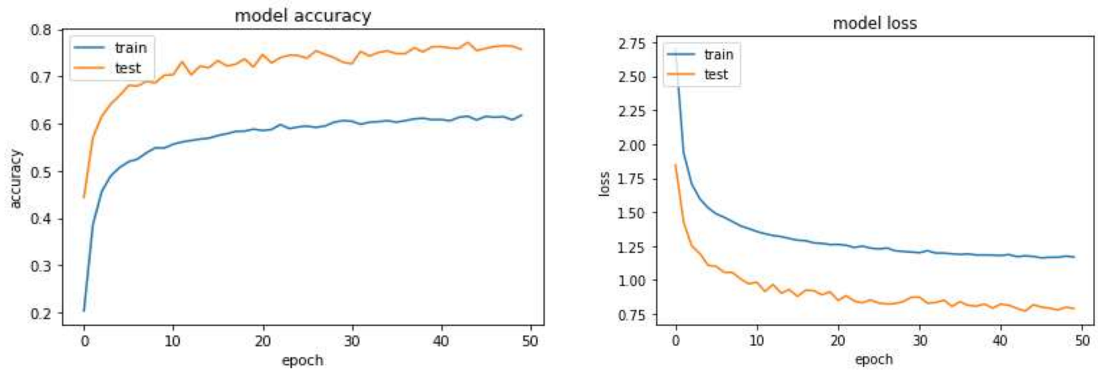


(g)

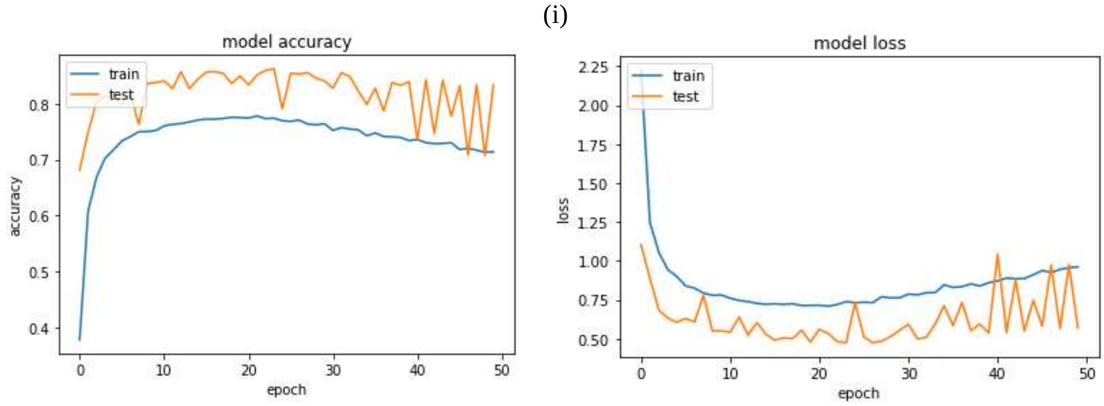


(h)

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024



(i)



(j)

Figure 6: Accuracy and Loss of CNN Model with (a) 10 Epochs and Adam Optimizer (b) 10 Epochs and RMSprop Optimizer (c) 20 Epochs and Adam Optimizer (d) 20 Epochs and RMSprop Optimizer (e) 30 Epochs and Adam Optimizer (f) 30 Epochs and RMSprop Optimizer (g) 40 Epochs and Adam Optimizer (h) 40 Epochs and RMSprop Optimizer (i) 50 Epochs and Adam Optimizer (j) 50 Epochs and RMSprop Optimizer

Table 3 and Fig.6 present the performance of a CNN algorithm for Marathi compound character recognition, where different configurations are tested with varying numbers of training epochs and optimizers. The results demonstrate the impact of varying the number of training epochs and the optimizer choice on the CNN's performance. For instance, with 40 epochs and the "Adam" optimizer, the model achieved impressive training accuracy (96.09%) and relatively low training loss (0.0791). However, performance on the validation dataset (87.48% accuracy) suggests some potential for overfitting. On the other hand, the "RmsProp" optimizer with 30 epochs yielded high validation accuracy (88.38%) while maintaining a competitive training accuracy (95.87%) and a lower validation loss (0.5160). Interestingly, execution times vary significantly between configurations, highlighting the trade-off between computational resources and model performance. These results provide valuable insights into selecting the optimal combination of epochs and optimizers for Marathi compound character recognition, balancing training efficiency and recognition accuracy.

5.2 Results of CNN-SVM algorithm

Table 4: Performance analysis of the Hybrid approach of CNN_SVM

Model	**Number of Epochs**	**Training Accuracy**	**Training Loss**	**Validation Accuracy**	**Validation Loss**	**Execution Time**
CNN-SVM	50	0.9860	0.9769	0.9782	0.9775	6214.1724

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024

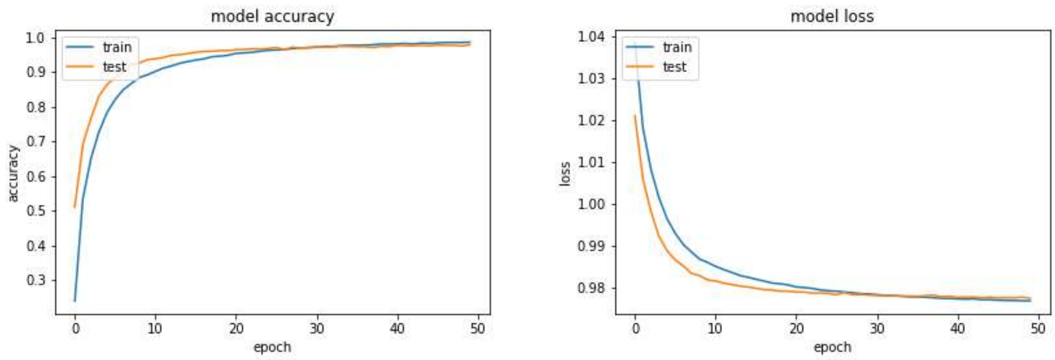


Figure 7: Accuracy and Loss of CNN-SVM Model with 50 Epochs and Adam Optimizer

Table 4 and Fig.7 present a performance analysis of a hybrid approach called "CNN-SVM," which combines both CNNs and SVM for Marathi compound character recognition. The "Number of Epochs" column indicates that the hybrid model was trained for 50 epochs, signifying that the entire training dataset was used 50 times to update the model's parameters. This iterative process helps the model learn and adapt over time. Regarding training performance, the "Training Accuracy" metric is notably high, measuring 98.60%. This means the hybrid model correctly classified the data points in the training dataset during training.

Similarly, the "Training Loss" of 0.9769 indicates that the model's predictions during training closely align with the actual training labels, reflecting a solid fit for the training data. Moving to the validation results, the "Validation Accuracy" stands at 97.82%, showcasing the model's capacity to generalize effectively to unseen data. This high validation accuracy suggests the model performs well on data it has never encountered. Furthermore, the "Validation Loss" of 97.75% indicates that the model's predictions on the validation dataset are very close to the true labels, demonstrating its robustness. Lastly, the "Execution Time" column records the total time to train and evaluate the hybrid model, which amounts to approximately 6214.1724 seconds (or about 1 hour and 43 minutes). This metric provides insights into the computational resources required for running this particular configuration.

The comparative analysis of the performance of the proposed CNN and CNN-SVM algorithm for Marathi compound character recognition is presented in Table 5.

Table 5: Comparative analysis of proposed CNN and CNN-SVM

Model	**Number of Epochs**	**Training Accuracy**	**Training Loss**	**Validation Accuracy**	**Validation Loss**
Proposed CNN	40	Adam	96.09	0.0791	87.48
Proposed CNN_SVM	50	0.9860	0.9769	0.9782	0.9775

Table 5 presents a comparative analysis of the two approaches. The model was trained using the Adam optimizer for 40 epochs in the "Proposed CNN" method. The training accuracy of 96.09% indicates that this CNN-based model achieved impressive accuracy on the training dataset, correctly classifying approximately 96% of the data points. The associated training loss of 0.0791 suggests that the model's predictions during training closely matched the actual training labels, indicating an excellent fit to the training data. The model exhibited a validation accuracy of 87.48%, signifying its ability to generalize reasonably well to previously unseen data. While the validation loss is not provided in the table, it typically measures how close the model's predictions were to the true labels on the validation dataset.

On the other hand, the "Proposed CNN-SVM" approach utilized a more extended training duration of 50 epochs. This hybrid CNN-SVM model achieved an exceptionally high training accuracy of 98.60%, indicating that it correctly classified nearly 99% of the training dataset. The training loss of 0.9769 reflects a strong alignment between the model's training predictions and the actual labels. During the validation phase, the model demonstrated an impressive validation accuracy of 97.82%, highlighting its robust generalization capabilities to previously unseen data. The validation loss, which was 0.9775, indicates that the model's predictions on the validation dataset were remarkably consistent with the true labels.

The "Proposed CNN-SVM" approach outperformed the "Proposed CNN" in terms of both training and validation accuracy. However, it is essential to note that the CNN-SVM approach used a more extended training period, potentially requiring more computational resources. This comparative analysis aids in evaluating the trade-offs between model complexity and computational resources, assisting practitioners in making informed decisions about which model to select based on their specific application requirements.

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024

6. Conclusion

This research comprehensively investigates Handwritten Compound Devnagari Character Recognition, leveraging CNN and a novel CNN-SVM Hybrid Algorithm. The comparative analysis of these two approaches sheds light on their strengths and potential character recognition applications.

Following 40 epochs of training with the Adam optimizer, the Proposed CNN model achieved a noteworthy training accuracy of 96.09%, accompanied by a training loss of 0.0791. The validation accuracy, albeit slightly lower at 87.48%, signifies the model's resilience in identifying handwritten Devnagari characters. This observation underscores the practical viability of CNNs for recognition tasks, emphasizing their proficiency in capturing intricate features from the input data.

In contrast, the innovative CNN-SVM Hybrid Algorithm demonstrated notable outcomes, achieving a training accuracy of 98.60% with a corresponding training loss of 0.9769. The validation metrics yielded an accuracy of 97.82% and a loss of 0.9775 after 50 training epochs. These results underscore the superior performance of the hybrid methodology, affirming its potential for enhanced accuracy and resilience in character recognition tasks.

For future research, several promising avenues are worth exploring. Integrating more advanced deep learning architectures and techniques, such as recurrent neural networks (RNNs) or attention mechanisms, could enhance recognition accuracy. Secondly, increasing the diversity of the dataset by incorporating more handwriting styles and variations can help improve the model's generalization capabilities. Additionally, exploring incorporating transfer learning from other character recognition tasks may expedite training and boost performance. Furthermore, developing a real-time recognition system for practical applications, such as document digitization or text-to-speech conversion, could be a valuable direction for future research. Finally, optimizing hyperparameters and investigating ensemble methods to combine multiple models could lead to even more robust and accurate character recognition systems.

This study represents a significant step forward in Handwritten Compound Marathi Character Recognition, demonstrating the effectiveness of both CNNs and the CNN-SVM Hybrid Algorithm. The future holds exciting opportunities for further advancements in this domain, paving the way for practical applications that can benefit from accurate and efficient character recognition systems.

References

[1] Momeni, Saleh & Baba Ali, Bagher. (2023). A Transformer-based Approach for Arabic Offline Handwritten Text Recognition.

[2] Ajmire, Prafulla & Warkhede, Shankar. (2018). An Analytical Study of Devanagari Script Recognition.

[3] Sachdeva, Juhee & Mittal, Sonu & Yean, Len. (2022). Handwritten Offline Devanagari Compound Character Recognition Using CNN. 10.1007/978-981-16-6289-8_18.

[4] Kale, Karbhari & Chavan, Shriniwas & Kazi, Majharoddin & Rode, Yogesh. (2013). Handwritten Devanagari Compound Character Recognition Using Legendre Moment: An Artificial Neural Network Approach. 10.1109/ISCBI.2013.62.

[5] Gupta, Piyush, et al. "Convolutional Neural Network based Handwritten Devanagari Character Recognition." 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE). IEEE, 2020.

[6] Rani, N. Shobha, et al. "Deep Learning Network Architecture based Kannada Handwritten Character Recognition." 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE, 2020.

[7] Prof. Parvaneh Basaligheh. (2017). Design and Implementation of High Speed Vedic Multiplier in SPARTAN 3 FPGA Device. International Journal of New Practices in Management and Engineering, 6(01), 14 - 19. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/51

[8] Sahu, Anupama, and S. N. Mishra. "Odia Handwritten Character Recognition with Noise using Machine Learning." 2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC). IEEE, 2020.

[9] Gurav, Yash, et al. "Devanagari Handwritten Character Recognition using Convolutional Neural Networks." 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE). IEEE, 2020.

[10] Narang, Sonika Rani, Munish Kumar, and Manish Kumar Jindal."DeepNetDevanagari: a deep learning model for Devanagari ancient character recognition." Multimedia Tools and Applications 80.13 (2021): 20671-20686.

[11] Sarangi, D. P. K. (2022). Malicious Attacks Detection Using Trust Node Centric Weight Management Algorithm in Vehicular Platoon. Research Journal of Computer Systems and Engineering, 3(1), 56–61. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/42

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024

[12] Inkeaw, Papangkorn, Piyachat Udomwong, and Jeerayut Chaijaruwanich. "Density based semi-automatic labeling on multi-feature representations for ground truth generation: Application to handwritten character recognition." Knowledge-Based Systems 220 (2021): 106953.
[13] Mane, Deepak T., Rushikesh Tapdiya, and Swati V. Shinde."Handwritten Marathi numeral recognition using stacked ensemble neural network." International Journal of Information Technology (2021): 1-7.
[14] Geetha, R., T. Thilagam, and T. Padmavathy. "Effective offline handwritten text recognition model based on a sequence-to-sequence approach with CNN–RNN networks." Neural Computing and Applications (2021): 1-12.
[15] Bhati, Gaurav Singh, and Akhil Ranjan Garg. "Handwritten Devanagari Character Recognition Using CNN with Transfer Learning." Congress on Intelligent Systems. Springer, Singapore, 2020.
[16] Jain, Mayank, et al. "Handwritten Digit Recognition Using CNN." 2021 International Conference on Innovative Practices in Technology and Management (ICIPTM). IEEE, 2021.
[17] Moore, B., Clark, R., Martinez, J., Rodriguez, A., & Rodriguez, L. Anomaly Detection in Internet of Things (IoT) Data Streams. Kuwait Journal of Machine Learning, 1(4). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/151
[18] Dokare, Indu, et al. "Recognition of Handwritten Devanagari Character using Convolutional Neural Network." 2021 3rd International Conference on Signal Processing and Communication (ICPSC). IEEE, 2021.
[19] Sharma, N.; Pal, U.; Kimura, F.; Pal, S. Recognition of offline handwritten Devanagari characters using quadratic classifier. In Computer Vision, Graphics and Image Processing; Springer: Berlin/Heidelberg, Germany, 2006; pp. 805–816.
[20] Hanmandlu, M.; Murthy, O.V.R.; Madasu, V.K. Fuzzy Model based recognition of handwritten Hindi characters. In Proceedings of the 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications, Glenelg, Australia, 3–5 December 2007; pp. 454–461.
[21] Kumar, S. Performance comparison of features on Devanagari handprinted dataset. Int. J. Recent Trends 2009, 1, 33–37.
[22] Pal, U.; Sharma, N.; Wakabayashi, T.; Kimura, F. Offline handwritten character recognition of Devanagari script. In Proceedings of the Ninth International Conference on Document Analysis and Recognition, Curitiba, Parana, Brazil, 23–26 September 2007; Volume 1, pp. 496–500.
[23] P. Gupta, S. Deshmukh, S. Pandey, K. Tonge, V. Urkunde and S. Kide (2020), "Convolutional Neural Network based Handwritten Devanagari Character Recognition," 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), pp. 322-326, doi: 10.1109/ICSTCEE49637.2020.9277222.
[24] Narang, Sonika Rani, Munish Kumar, and Manish Kumar Jindal.(2021) "DeepNetDevanagari: a deep learning model for Devanagari ancient character recognition." Multimedia Tools and Applications 80.13:20671-20686.
[25] Dokare, Indu, et al.(2021) "Recognition of Handwritten Devanagari Character using Convolutional Neural Network." 2021 3rd International Conference on Signal Processing and Communication (ICPSC). IEEE.
[26] Deore, Shalaka Prasad, and Albert Pravin.(2020) "Devanagari Handwritten Character Recognition using fine-tuned Deep Convolutional Neural Network on trivial dataset." Sadhana 45.1: 1-13.
[27] Bhagat et al.(2020) "Devanagari Handwritten Character Recognition using Convolutional Neural Networks." 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE). IEEE.
[28] Chen L, Li S, Bai Q, Yang J, Jiang S, Miao Y. Review of Image Classification Algorithms Based on Convolutional Neural Networks. Remote Sensing. 2021; 13(22):4712.
[29] Yamashita, R., Nishio, M., Do, RKG et al. Convolutional neural networks: an overview and application in radiology. Insights Imaging 9, (2018): 611–629.
[30] KC Santosh, Nibaran Das, Swarnendu Ghosh, "Chapter 2 - Deep learning: a review," In Primers in Biomedical Imaging Devices and Systems, Deep Learning Models for Medical Imaging, Academic Press, 2022, 29-63, ISBN 9780128235041
[31] Gholamalinezhad, H., & Khosravi, H. (2020). Pooling Methods in Deep Neural Networks, a Review. *ArXiv, abs/2009.07485*.
[32] Montesinos López, O.A., Montesinos López, A., Crossa, J. (2022). Fundamentals of Artificial Neural Networks and Deep Learning. In: Multivariate Statistical Machine Learning Methods for Genomic Prediction. Springer, Cham.
[33] Hassan, E., Shams, M.Y., Hikal, N.A. et al. The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study. Multimed Tools Appl 82, 16591–16633 (2023)

Doi: https://doi.org/10.54216/FPA.140202
Received: August 23, 2023 Revised: October 29, 2023 Accepted: January 06, 2024