Character Recognition Using Ensemble Technique

A

Minor Project

Submitted in partial fulfillment for the award of the degree of

Masters

IN

Computer Application

by

Prashant Singh

MCAN1CA22019

Under the guidance of

Dr. Vani Agarwal

Department of Computer Science Engineering

School of Engineering and Technology

ITM UNIVERSITY, GWALIOR - 474026 MP, INDIA

(Dec 2023)

<u>CERTIFICATE</u>

This is to certify that the work titled "**Character Recognition Using Ensemble Technique**" submitted by "Prashant Singh, (*MCAN1CA22019*)" in partial fulfilment for the award of the degree of MCA (CSE), ITM University, Gwalior has been carried out under my/oursupervision.
To the best of my knowledge and belief, the dissertation

(ii) Is an original piece of work of Candidate.

(ii) It has duly been completed.

(iii) It is up to the standard both in respect of contents and language.

(iv) Free from plagiarism

(v) Work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Guide
Dr. Vani Agarwal

Designation: Associate Professor

Date:

DECLARATION

I with this declare that the work entitled "**Character Recognition Using Ensemble Technique**" submitted to the Department of Computer Science Engineering, School of Engineering and Technology, ITM University, Gwalior (M.P.) is our work done under the supervision of Guide name. The dissertation doesn't contain any part which has been submitted for award of any degree either in this University or in any other University.

We further declare that the work is free from any plagiarism.

(Signature of the candidates)

Prashant Singh

MCAN1CA22019

Verified by guide

ACKNOWLEDGEMENT

Firstly, I thank Lord Almighty for making it possible for me to complete this work. The success and outcome of this project required a lot of guidance and assistance from many people, and I am incredibly privileged to have got it all along with the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank Dr. Vani Agarwal for providing me an opportunity to do the project work and giving us all support and guidance, which made me complete the project duly. She took keen interest on my project work and guided me all along and provided all the necessary information for developing a good system.

I would not forget to remember HOD, Mentors, DEAN for their encouragement and more over for his timely support and guidance till the completion of our project work. I heartily thank Dr Vani Agarwal for their guidance and suggestions during this project work.

At last, but not the least I would like to thank our parents who trusted usand helped us through the errand.

Abstract

Handwritten Devanagari Character Recognition (HCR) remains a pivotal area of research with extensive applications across diverse domains, from document analysis to language-specific character recognition systems. This comprehensive literature survey explores recent advancements in HCR, focusing on ensemble techniques, Convolutional Neural Networks (CNNs), and diverse preprocessing methodologies aimed at enhancing recognition accuracy and efficiency. Ensemble methodologies, highlighted in studies such as "Handwritten Digit Recognition Using Ensemble Learning" (Kuppa Venkata Padmanabha Nandan et al., 2020), showcase significant improvements by harnessing CNNs in conjunction with ensemble learning, demonstrating adaptability across various data splits and enhancing recognition accuracy. Additionally, studies emphasizing preprocessing methodologies, like "SVM Based Handwritten Hindi Character Recognition and Summarization" (Sunil Dhankhar et al., 2021), underscore the critical role of preprocessing techniques, employing morphological operations and histograms of oriented gradients (HOG) to elevate character recognition precision. The survey underscores the pivotal role of CNNs, exemplified by "CNN-based ensemble methods to recognize Bangla handwritten characters" (Mir Moynuddin Ahmed Shibly et al., 2021), showcasing their impact on achieving high accuracy in character recognition tasks. Furthermore, the exploration extends to language-specific recognition, exemplified by studies such as "A Novel Weighted SVM Classifier Based on SCA for Handwritten Marathi Character Recognition" (Surendra P. Ramteke et al., 2019), underscoring the importance of recognizing linguistic diversity in character recognition systems. Moreover, investigations into image preprocessing methods and their influence on recognition accuracy, as detailed in "Research on Influence of Image Preprocessing on Handwritten Number Recognition Accuracy" (Tieming Chen et al., 2019), emphasize the significance of preprocessing techniques in various recognition algorithms. Additionally, the integration of transfer learning, exemplified by "Recognition of Handwritten Japanese Characters Using Ensemble of CNNs" (Angel I. Solis et al., 2023), showcases adaptability to diverse datasets, catering to complex, multi-script character recognition. The study explores the omission of conventional preprocessing steps and the reliance on CNN-driven feature extraction methodologies for the Handwritten Devanagari Character Dataset, emphasizing the network's capacity to autonomously learn discriminative features directly from raw image data. Furthermore, individual model performances, including MyModel, VGG16, InceptionV3, Xception, ResNet50, ResNet50V2, ResNet152V2, DenseNet121, and MobileNetV2, were evaluated, with varying accuracies and losses. However, the ensemble approach emerged as the highlight, achieving a remarkable accuracy of 98% and an overall loss of 0.18, surpassing individual models and showcasing the efficacy of collective intelligence in enhancing recognition precision.

Table of Contents

List of Abbreviations

CSE	Computer Science Engineering
HOD	Head of department
CNN	Convolutional Neural Network
SVM	Support Vector Machine
HCR	Handwritten Character Recognition
VGG16	Visual Geometry Group 16
DN121	DesNet121
MN	MobileNetV2
RN50	ResNet50
RN50V2	ResNet50V2
RN15V2	ResNet15V2
IV3	InceptionV3
XC	Xception

List of Figures & Tables (Sample)

1. Introduction

In the intricate landscape of machine learning, few challenges persist as resolutely as the accurate recognition of handwritten letters. Within this domain lie the intricate threads of automation, character recognition, and document processing, each intricately woven into the fabric of diverse applications. Amidst this convergence lies a labyrinthine pursuit: to decipher, with precision, the fluid strokes of handwritten letters, transcending the bounds of human interpretation into the realm of artificial intelligence. The terrain of handwritten letter recognition is beset with multifaceted complexities, where strokes, nuances, and individuality converge to pose a formidable challenge to conventional machine learning models. It is within this intricate tapestry that the quest for innovative methodologies takes root, beckoning researchers and practitioners to forge unique pathways that unravel the mysteries latent within handwritten script.

Traditionally, the trajectory of addressing this challenge often navigates through a prescribed route, marked by meticulous preprocessing methodologies. These methodologies, encompassing a spectrum from normalization techniques to noise reduction and intricate feature extraction, lay the groundwork for subsequent model development. However, within this framework, an unconventional pathway emerges—a divergence from the norm, one that places a distinct emphasis on model construction, ensemble techniques, and a minimized approach to preprocessing. This deviation sparks a journey that navigates through uncharted territories, steering away from the well-trodden path of preprocessing dominance. Here, the spotlight shifts from extensive preprocessing maneuvers to the artistry of model building—a craft honed through the amalgamation of pre-built models and bespoke creations. It's a journey that celebrates the prowess of an ensemble—an orchestra of diverse models orchestrated to harmonize their collective learning capacities, poised to decipher the cryptic strokes of handwritten letters.

The foundation of this unique pathway lies in the deliberate choice to minimize preprocessing intricacies, channeling efforts toward the orchestration of a formidable ensemble. The ensemble comprises a symphony of pre-built models renowned for their prowess in diverse arenas, standing shoulder to shoulder with a bespoke creation meticulously crafted to resonate with the intricacies of handwritten script. This eclectic ensemble is a testament to the vision of convergence—an embodiment of the belief that the amalgamation of diverse learning algorithms can transcend the limitations of individual models. However, amidst this departure from the traditional norm, the lodestar guiding this journey remains unwavering: the rigorous evaluation and validation of performance. Herein lies the crux—a commitment to the K-Fold cross-validation strategy. Despite the deviation from extensive preprocessing methodologies, the adoption of this robust validation technique stands as a testament to the commitment to assess, scrutinize, and refine the ensemble's efficacy in deciphering the complexities of handwritten letters.

2. Literature Survey

HCR is a vital area of research with diverse applications, ranging from document analysis to language-specific character recognition systems. Recent studies have made significant strides in this field, capitalizing on ensemble techniques, CNNs (CNNs), and various pre-processing methods to enhance recognition accuracy and efficiency. This literature survey offers an in-depth overview of the latest developments in HCR, encompassing a wide array of techniques, addressing challenges, and exploring their implications across multiple languages.

Ensemble Techniques for Improved Accuracy

Ensemble methods have emerged as a promising approach to substantially enhance the accuracy of HCR systems. A notable example is the paper "Handwritten Digit Recognition Using Ensemble Learning" by Kuppa Venkata Padmanabha Nandan et al. (2020). This work leverages ensemble learning in conjunction with CNN models and achieves significant accuracy improvements. The ensemble approach proves particularly effective when dealing with various data splits, such as random and class-wise divisions, underscoring its adaptability and potential to boost recognition accuracy.

Preprocessing: A Cornerstone of Character Recognition

Effective preprocessing techniques play a pivotal role in refining the quality and consistency of input data. "SVM Based Handwritten Hindi Character Recognition and Summarization" by Sunil Dhankhar et al. (2021) is a prime illustration of the importance of preprocessing. This study employs a range of methods, including morphological operations, edge detection, and histograms of oriented gradients (HOG), to enhance the recognition of handwritten Hindi characters. These preprocessing steps significantly contribute to achieving remarkable precision in character recognition, emphasizing their critical role in the overall process.

CNNs: The Backbone of Recognition Systems

CNNs (CNNs) continue to be a cornerstone in recent research endeavors to attain state-of-the-art results in HCR. "CNN-based ensemble methods to recognize Bangla handwritten characters" by Mir Moynuddin Ahmed Shibly et al. (2021) exemplifies the powerful impact of CNNs when trained on a Bangla handwritten character dataset. This research explores various CNN architectures and shallow machine learning algorithms, demonstrating the pivotal role of CNNs in character recognition. The high accuracy achieved underscores the proficiency of CNNs in this domain.

Language-Specific Character Recognition

Recent research ventures extend their reach to encompass a wide range of languages, showcasing the extensive applicability of HCR. "A Novel Weighted SVM Classifier Based on SCA for Handwritten Marathi Character Recognition" by Surendra P. Ramteke et al. (2019) exemplifies the recognition of Marathi characters. The study introduces a unique approach with a high degree of accuracy. By addressing the challenge of language-specific character recognition, this research paves the way for similar studies focused on under-resourced languages. It underscores the importance of recognizing linguistic diversity in character recognition systems.

Image Preprocessing and Recognition Accuracy

Handling variations in handwritten characters is a substantial challenge in character recognition. "Research on Influence of Image Preprocessing on Handwritten Number Recognition Accuracy" by Tieming Chen et al. (2019) underscores the significance of preprocessing techniques. The study emphasizes the importance of character segmentation, tilt correction, offset correction, and size normalization in elevating recognition accuracy. These techniques yield positive results across various recognition algorithms, including SVMs (SVMs), hidden Markov models (HMMs), CNNs, and extreme learning machines (ELMs).

Transfer Learning and Robust Recognition

Transfer learning emerges as a potent strategy in the development of character recognition systems. "Recognition of Handwritten Japanese Characters Using Ensemble of CNNs" by Angel I. Solis et al. (2023) introduces a CNN-ensemble model employing transfer learning to recognize handwritten Kanji characters. This research showcases the model's adaptability to diverse datasets, underlining the importance of recognizing complex, multi-script characters. Transfer learning stands out as a formidable tool for constructing robust recognition systems capable of handling intricate and diverse scripts effectively.

Comparative Study

Comparative Summarization

Table 1 summarizes some of the recent research on HWR. The table includes information on the authors, year of publication, journal, techniques used, and summary of the research. The research covers a variety of topics, including ensemble learning, feature extraction, and pre-processing.

Techniques Used	**Summary**	**Reference**
CNN with dropouts	Ensemble learning for handwritten digit recognition using a simple CNN model	[1]
Morphology, HOG, SVM, RSVM, Modified Pihu	Two-stage approach for recognizing handwritten Hindi characters and summarizing text	
CNNs, Ensemble methods (Bagging, Boosting, Random Forests, etc.)	CNNs with different architectures, shallow machine learning algorithms, and ensemble techniques to recognize Bangla handwritten characters	4
Weighted One-Against-Rest SVMs (WOAR-SVM), Sine Cosine Algorithm (SCA), Morphological operations, Modified Pihu, Statistical, Global, Geometrical, and Topological feature extraction	Weighted SVM classifier with SCA and various feature extraction methods for recognizing handwritten Marathi characters	5
Background modeling, Fingertip detection, Virtual character reconstruction, Normalization, DTW-based classifier	Vision-based finger-writing character recognition system	10
Preprocessing (Tilt correction, Offset correction, Size normalization, Thinning)	Preprocessing algorithm for hand-written character recognition	3
Grayscale, normalization, Statistical SVM (SVM)	Statistical SVM-based framework for HCR	8
CNN, MLP, Various preprocessing and post-processing techniques	Supervised classifier approach for recognizing handwritten Gujarati characters	6
tilt correction, offset correction, size normalization, and thinning. SVMs, NNs,HMMs,RFs	Impact of image preprocessing techniques on recognition accuracy for handwritten numbers	7
CNNs, Transfer learning	CNN-ensemble model for recognizing handwritten Japanese characters (Kanji)	9

Table 1: Summarization of Recent Literature

Comparative accuracy evaluation

Table 2 summarizes the accuracies of various HCR (HWR) techniques and algorithms presented in recent research papers.

Paper Title	**Accuracy**
Handwritten Digit Recognition Using Ensemble Learning	99% (Random split), 95% (Class-wise split)
SVM Based Handwritten Hindi Character Recognition and Summarization	96.97%
CNN based Ensemble Methods to Recognize Bangla Handwritten Characters	98.68% (Ekush dataset), 98.69% (BanglaLekha-Isolated dataset)
A Novel Weighted SVM Classifier Based on SCA for Handwritten Marathi Character Recognition	95.14%
A Novel Vision-Based Finger-Writing Character Recognition System	95.3% (Uppercase), 98.7% (Lowercase English alphabets)
Construction of Statistical SVM based Recognition Model for HCR	99.70%
Gujarati HCR from Text Images	97.21%
Recognition of Handwritten Japanese Characters Using Ensemble of CNNs	Ranging from 95% to 99.35% on various datasets
Research on Influence of Image Preprocessing on Handwritten Number Recognition Accuracy	Significant improvements for various recognition algorithms
A Preprocessing Algorithm for Hand-Written Character Recognition	Up to 5% Imporvements

Table 2: Summarization of Accuracy and Improvements

3. Existing Gaps

In the expansive domain of HCR, the current landscape reveals an amalgamation of diverse methodologies, each striving to decode the intricacies of handwritten script across multiple languages and scripts. Yet, within this realm of innovation and progress, discernible gaps persist, presenting ripe opportunities for further exploration and advancement.

Integration of Ensemble Techniques and Preprocessing Strategies

While the literature showcases the effectiveness of ensemble techniques in augmenting recognition accuracy, a pivotal area for exploration lies in the intricate interplay between ensemble methods and preprocessing strategies. The presented studies illustrate the independent effectiveness of ensemble learning and preprocessing methods. However, a notable gap emerges in understanding how these methodologies synergize and complement each other. Exploring the symbiotic relationship between ensemble techniques and preprocessing steps could unveil novel pathways to further enhance recognition accuracy and robustness.

Unexplored Frontiers in Language-Specific Recognition

The literature review underscores the significance of recognizing characters from specific languages such as Hindi, Marathi, and Bangla. Despite these commendable endeavors, there exists a niche awaiting exploration—the extension of language-specific recognition to under-resourced or lesser-studied languages. The studies predominantly focus on languages with available datasets or substantial research attention. Addressing this gap could involve exploring strategies to adapt recognition systems to languages lacking extensive resources, thereby broadening the applicability of character recognition methodologies.

Transfer Learning in Multilingual Recognition

The utilization of transfer learning emerges as a potent tool in recognizing complex, multi-script characters, as highlighted in the context of recognizing handwritten Kanji characters in Japanese. Nevertheless, the literature predominantly focuses on individual languages or scripts, leaving an uncharted domain—the exploration of transfer learning frameworks capable of seamlessly transitioning across multiple languages and scripts. Investigating transfer learning models that adeptly adapt to diverse scripts without losing efficiency or accuracy could bridge the gap in creating versatile recognition systems for multilingual contexts.

Robustness to Variations in Handwriting Styles

The literature accentuates the challenge of handling variations in handwritten characters and the subsequent impact on recognition accuracy. However, a discernible gap exists in addressing the robustness of recognition systems to adapt and generalize across diverse handwriting styles, especially when faced with limited training data or unconventional writing styles. Exploring methodologies that bolster recognition systems' adaptability to variations in handwriting styles could significantly enhance their practical utility across diverse real-world scenarios.

4. Proposed Methodology

HCR stands as a challenging frontier in machine learning, characterized by intricate strokes, varying styles, and diverse writing patterns. Existing models often rely on singular classification algorithms, potentially constraining their ability to comprehensively capture the nuances within handwritten data. This research aims to construct an ensemble model, strategically merging the strengths of multiple classification algorithms. This ensemble framework seeks to address the limitations of individual models, offering a more robust and comprehensive approach to HCR.

Methodology:

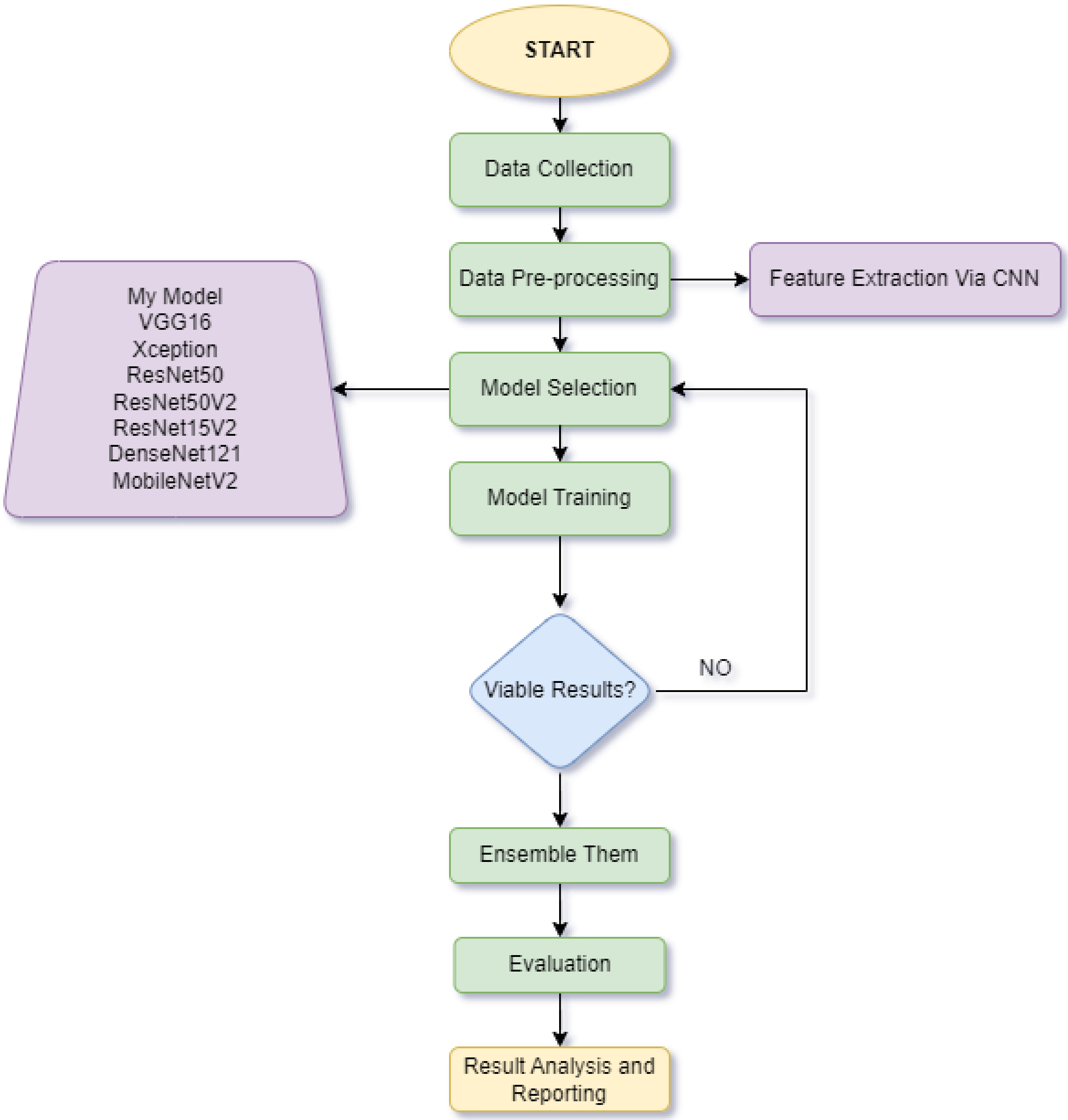


Fig 1: Process Flow Diagram

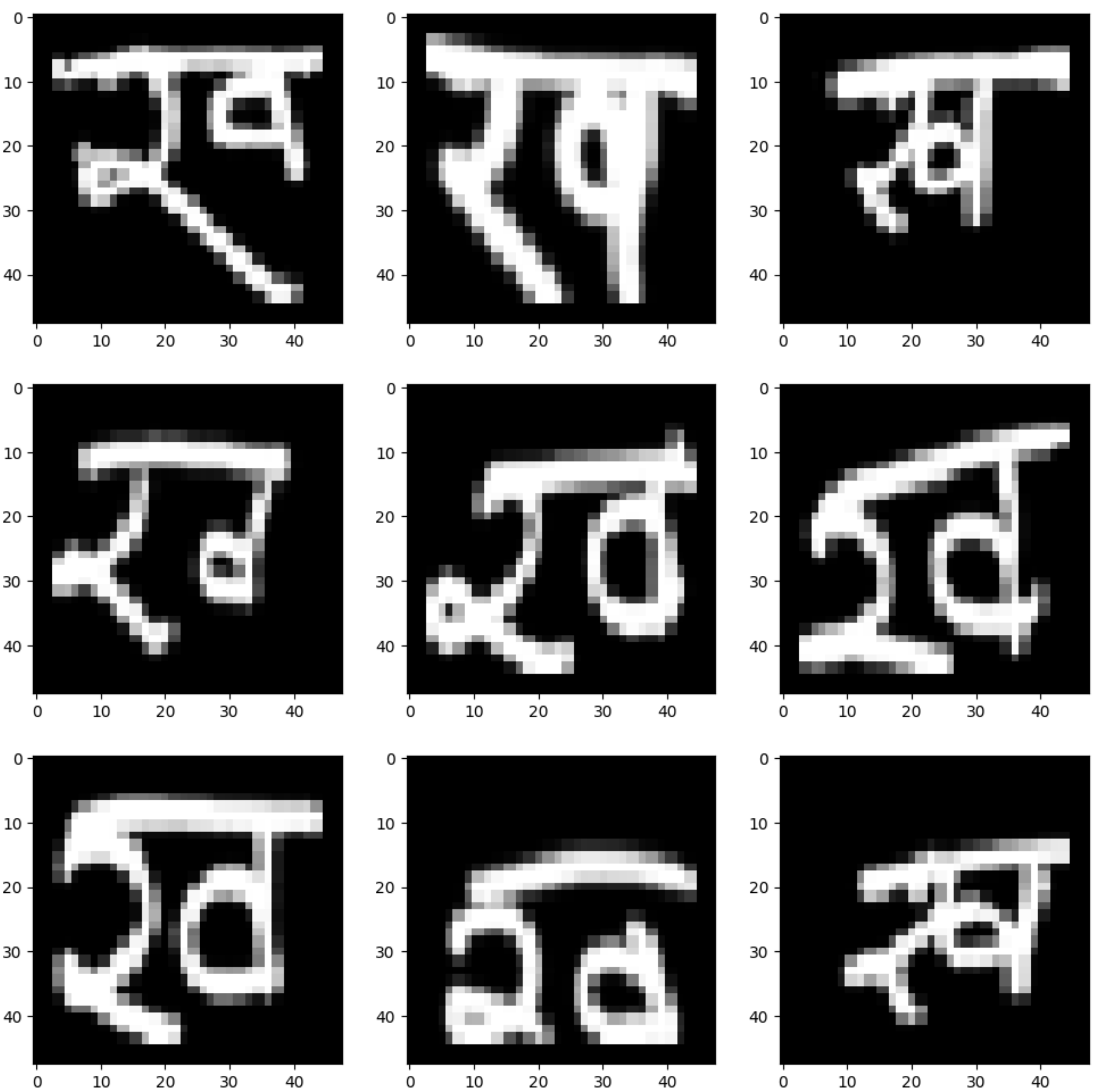
a. Data Collection

The cornerstone of this endeavor involves sourcing and preparing the dataset crucial for training the HCR models. The dataset selected for this research is the Devanagari Handwritten Character Dataset obtained from Kaggle. This comprehensive dataset comprises 46 distinct classes of Devanagari characters, each class containing 2000 examples.

The dataset has been meticulously partitioned into a training set, accounting for 85% of the data, and a testing set comprising the remaining 15%. This partitioning, sourced from the UCI Machine Learning Repository, ensures a robust division for training and evaluating the models' performance.

The Devanagari Handwritten Character Dataset serves as the foundational bedrock for this research, providing a rich and diverse array of characters, meticulously split into training and testing subsets, thereby facilitating the construction and validation of the ensemble-based HCR models.

Fig 2: An Instance from the dataset



b. Data Preprocessing

Within the realm of preprocessing for the Handwritten Devanagari Character Dataset, the primary focus has been on feature extraction facilitated through CNNs (CNNs), omitting conventional preprocessing steps to harness the CNN's inherent capabilities in learning discriminative features directly from the raw image data.

CNN-based Feature Extraction

The preprocessing pipeline initiates with the utilization of CNNs for feature extraction directly from the raw image data. CNNs, renowned for their proficiency in image-related tasks, serve as a powerful tool for automatically learning and extracting relevant features from the handwritten character images.

Through the utilization of convolutional layers, pooling operations, and activation functions, the CNNs discern intricate patterns, edges, shapes, and textures inherent in the Devanagari character images. This process enables the network to automatically extract hierarchical and discriminative features directly from the pixel-level data, without explicit preprocessing steps like noise reduction or normalization.

Advantages of CNN-based Feature Extraction

By adopting a CNN-driven feature extraction strategy, this approach capitalizes on the network's ability to inherently learn abstract and discriminative representations from the raw image data. The hierarchical nature of CNN architectures enables the extraction of increasingly complex features at deeper layers, potentially capturing intricate nuances crucial for distinguishing between different Devanagari characters.

Moreover, the exclusion of conventional preprocessing steps streamlines the workflow, potentially preserving fine-grained details and preventing information loss that could occur during traditional preprocessing stages. This strategy also aligns with the philosophy of end-to-end learning, allowing the model to autonomously discern relevant features for the HCR task.

c. Model Selection for Ensemble

The selection process for constructing the ensemble in Handwritten Devanagari Character Recognition involves a comprehensive exploration and training of diverse classification models, each renowned for its efficacy in image-related tasks. The ensemble comprises the following meticulously chosen models, each trained individually to harness their unique architectural strengths:

My Model:

The architecture of the custom-built model integrates convolutional and pooling layers followed by fully connected layers. This model, while tailored for the specific requirements of the Handwritten Devanagari Character Dataset, offers flexibility in design and feature extraction. However, its performance might vary depending on the complexity and diversity of the dataset, potentially lacking the depth and inherent generalization capabilities of pre-trained architectures.

VGG16:

VGG16 comprises 16 convolutional and fully connected layers, characterized by its simplicity and uniform architecture with small (3x3) convolutional filters. Its straightforward architecture allows ease of understanding and implementation. However, the depth of the network might lead to increased computational requirements and susceptibility to overfitting on smaller datasets due to a large number of parameters.

Xception:

XC presents an innovative architecture featuring depth-wise separable convolutions. Its depth-wise separable convolutions replace the standard convolutional layers, reducing computational complexity and increasing efficiency. However, while offering promising results in larger datasets, XC might struggle with smaller datasets due to its extensive depth and complexity.

ResNet50:

RN50 introduces residual connections to tackle the vanishing gradient problem in deeper networks. Its skip connections facilitate the flow of gradients, enabling the training of remarkably deep models. Nevertheless, the increased depth might lead to more complex training and potential issues with overfitting on smaller datasets.

ResNet50V2:

RN50V2, an improved version of RN50, refines the architecture by introducing identity shortcut connections in a revised order. This modification aims to improve training convergence and computational efficiency. However, its added complexity might require more computational resources for training.

ResNet152V2:

RN50V2 extends the ResNet architecture by employing deeper layers (152 in total), fostering better feature extraction and representation learning. However, the increased depth might lead to higher computational demands, especially during training, and might suffer from overfitting on smaller datasets.

DenseNet121:

DN121 employs dense connectivity patterns where each layer receives direct inputs from all preceding layers, promoting feature reuse and enabling efficient information flow. This architecture facilitates better parameter efficiency and alleviates the vanishing gradient problem. Nevertheless, its increased parameter sharing might make the model more sensitive to noise and outliers in smaller datasets.

MobileNetV2:

MN introduces depth-wise separable convolutions and inverted residuals, optimizing for mobile and resource-constrained devices. Its lightweight architecture offers high efficiency with reduced computational demands. However, its design optimized for efficiency might compromise its performance in tasks requiring more complex feature extraction or on datasets with intricate patterns.

Architecture of My Custom Model:

The architecture commences with an input layer, accepting 32x32x3-dimensional data, denoting the dimensions of the input images with three colour channels (RGB). Sequentially, it incorporates convolutional layers, where the first convolutional layer comprises 32 filters followed by batch normalization and activation functions, promoting feature extraction and non-linearity. Subsequent max-pooling layers halve the spatial dimensions, enhancing computational efficiency while preserving essential features.

Further, the architecture deepens with additional convolutional layers, increasing the number of filters to 64, continuing with batch normalization and activation functions, and followed by max-pooling operations, enabling the extraction of higher-level features. The subsequent layers transition to fully connected dense layers, densely interconnecting the extracted features for more intricate pattern recognition. The architecture unfolds into multiple dense layers, progressively reducing the dimensions, culminating in a final dense layer with 46 units, signifying the output classes representing the distinct Devanagari characters. The model concludes with a SoftMax activation layer, producing probabilities across the 46 output classes, allowing for the classification of Devanagari characters.

Architecture of Custom Model:

Fig 3: Model Summary

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input (InputLayer)          [(None, 32, 32, 3)]       0

 conv1 (Conv2D)              (None, 32, 32, 32)        896

 batch_normalization (Batch  (None, 32, 32, 32)        128
 Normalization)

 activation (Activation)     (None, 32, 32, 32)        0

 max1 (MaxPooling2D)         (None, 16, 16, 32)        0

 conv2 (Conv2D)              (None, 16, 16, 64)        18496

 batch_normalization_1 (Bat  (None, 16, 16, 64)        256
 chNormalization)

 activation_1 (Activation)   (None, 16, 16, 64)        0

 max2 (MaxPooling2D)         (None, 8, 8, 64)          0

 flatten (Flatten)           (None, 4096)              0

 dense1 (Dense)              (None, 256)               1048832

 dense2 (Dense)              (None, 128)               32896

 dense3 (Dense)              (None, 64)                8256

 dense8 (Dense)              (None, 32)                2080

 dense4 (Dense)              (None, 46)                1518

 softmax (Activation)        (None, 46)                0

=================================================================
Total params: 1113358 (4.25 MB)
Trainable params: 1113166 (4.25 MB)
Non-trainable params: 192 (768.00 Byte)
```

MODEL	ARCHITECTURE	SIZE (MB)	TOP-1 ACCURACY (%)	TOP-5 ACCURACY (%)	PARAMETERS (MILLIONS)	DEPTH
VGG16	Stacked convolutional layers	528	71.3	90.1	138.4	16
XC	Inception blocks with depthwise separable convolutions	88	79	94.5	22.9	81
RN50	Residual connections with bottleneck design	98	74.9	92.1	25.6	107
RN50V2	Improved RN50 with attention mechanism and inverted residual blocks	98	76	93	25.6	103
RN50V2	Deeper RN50V2 with more residual connections	232	78	94.2	60.4	307
DN121	Densely connected convolutional layers with shortcut connections	33	75	92.3	8.1	242
MN	Lightweight architecture with depthwise separable convolutions and inverted residual blocks	14	71.3	90.1	3.5	105

Table 3: Comparison of Pre-Trained Model

d. Model Training

Each model underwent individual training procedures, tailored to their specific architectures and input image sizes, optimizing their learning for the Handwritten Devanagari Character Dataset.

My Model and VGG16:

For My Model and VGG16, the input image size was standardized to 32x32 pixels. This smaller input size facilitated faster training and reduced computational overhead while retaining essential features for character recognition tasks. The training procedures for these models prioritized feature extraction and learning representations pertinent to the dataset's characteristics within the constraints of the smaller image size.

Models with Larger Input Size (RN50, RN50V2, RN50V2, DN121, MN, XC):

Conversely, RN50, RN50V2, RN50V2, DN121, MN, and XC were trained with a larger input image size of 75x75 pixels. This larger input size allowed these models to capture more intricate details and nuanced patterns within the Devanagari characters. The training strategies for these models revolved around leveraging their deeper architectures and larger input dimensions to extract richer and more complex features crucial for accurate character recognition.

e. Model Evaluation

Comparing validation accuracy and validation loss during model evaluations provides crucial insights into the performance and behavior of each model trained on the Handwritten Devanagari Character Dataset.

Validation Accuracy:

Validation accuracy serves as a metric to gauge the model's ability to correctly classify and recognize Devanagari characters within the validation dataset. It measures the proportion of correctly predicted character classes among all validation samples. Higher validation accuracy indicates the model's proficiency in accurately identifying characters, showcasing its overall effectiveness in character recognition tasks.

Validation Loss:

Validation loss, on the other hand, represents the measure of the model's predictive error on the validation dataset. It quantifies the disparity between the predicted class probabilities and the actual ground truth labels. Lower validation loss signifies better alignment between predicted and actual values, indicating a more precise and confident model in character classification.

Model	**Validation Accuracy**	**Validation Loss**
My Model	0.94	0.21
VGG16	0.789	0.82
IV3	0.65	1.19
XC	0.66	1.2
RN50	0.23	2.78
RN50V2	0.736	0.92
RN50V2	0.74	1
DN121	0.71	0.96
MN	0.345	2.39

Table 4: Performance Comparison

My Model:

MyModel demonstrates impressive performance on the validation dataset, achieving a validation accuracy of 94% with a low validation loss of 0.21. This custom-designed model exhibits a robust capability to accurately classify Handwritten Devanagari characters. The high accuracy coupled with a minimal loss signifies the model's efficacy in learning intricate patterns and features within the dataset, showcasing its potential for reliable character recognition.

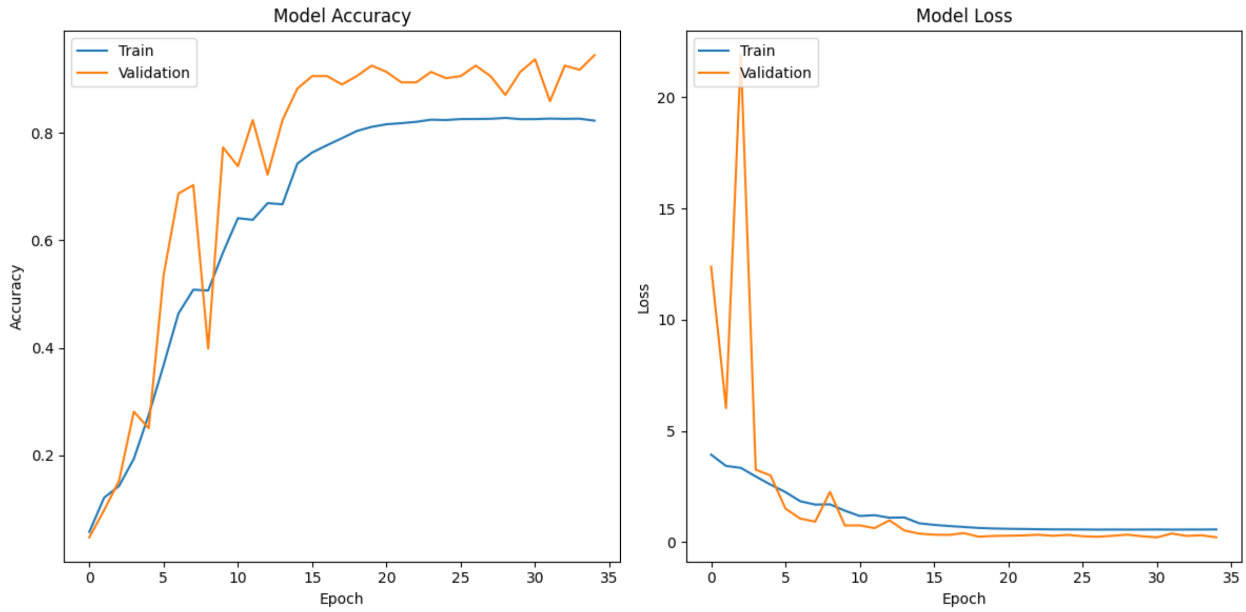


Fig 4: Performance of Custom Model

VGG16:

VGG16, while displaying a moderate validation accuracy of 78.9%, demonstrates a relatively high validation loss of 0.82. Despite its simplicity and uniform architecture, VGG16 appears to struggle in capturing the intricate nuances of the Handwritten Devanagari Character Dataset, leading to a comparatively higher predictive error and reduced accuracy.

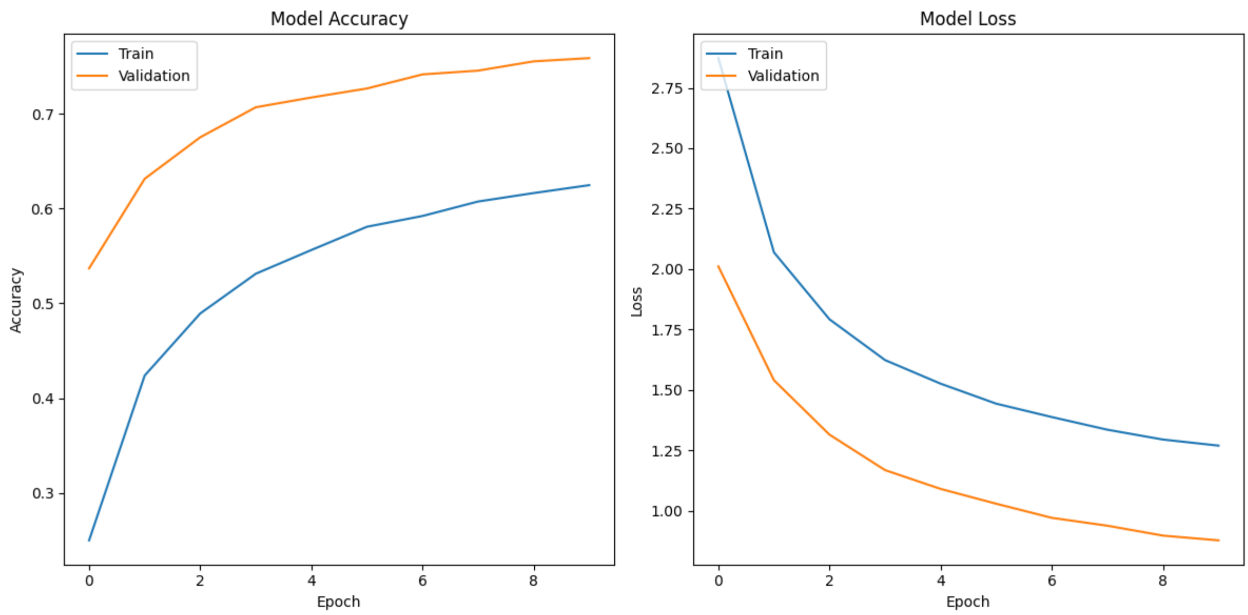


Fig 5: Performance of VGG16 Model

InceptionV3:

IV3 exhibits a validation accuracy of 65% with a relatively high validation loss of 1.19. This model, characterized by its depth-wise separable convolutions, appears to face challenges in accurately recognizing the diverse Devanagari characters. The lower accuracy and higher loss suggest limitations in learning the intricate patterns inherent in the dataset.

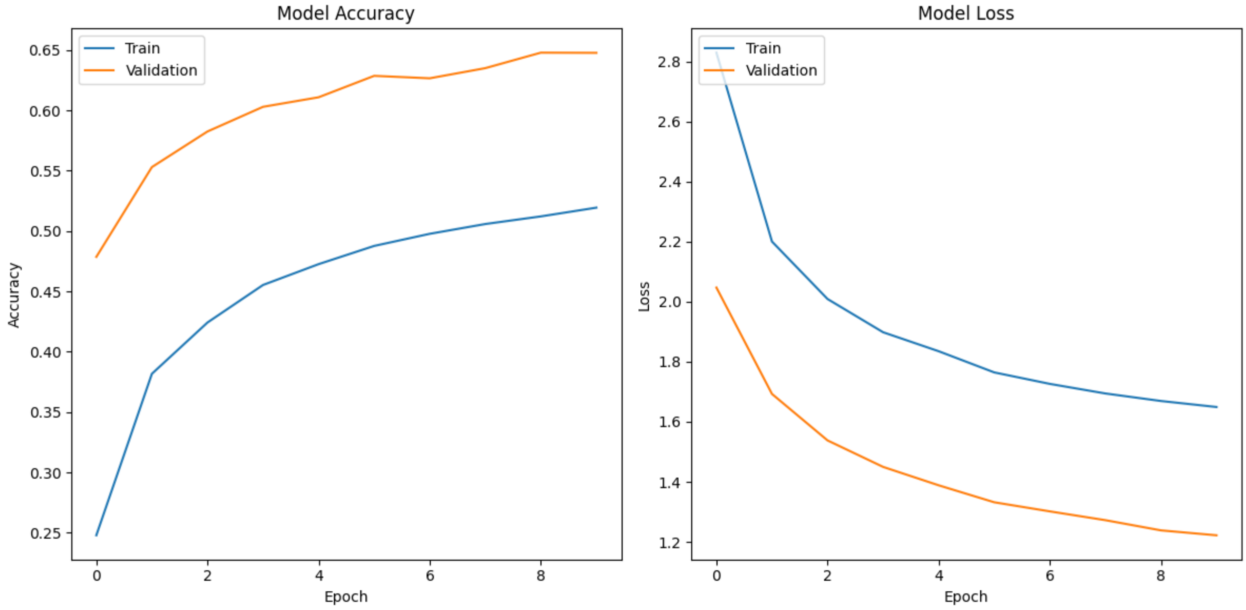


Fig 6: Performance of InceptionV3 Model

XC:

Similar to IV3, XC presents a validation accuracy of 66% and a validation loss of 1.2. Despite its innovative architecture, leveraging depth-wise separable convolutions, XC encounters difficulties in effectively discerning and classifying the diverse Devanagari characters, resulting in a moderate accuracy coupled with a relatively high loss.

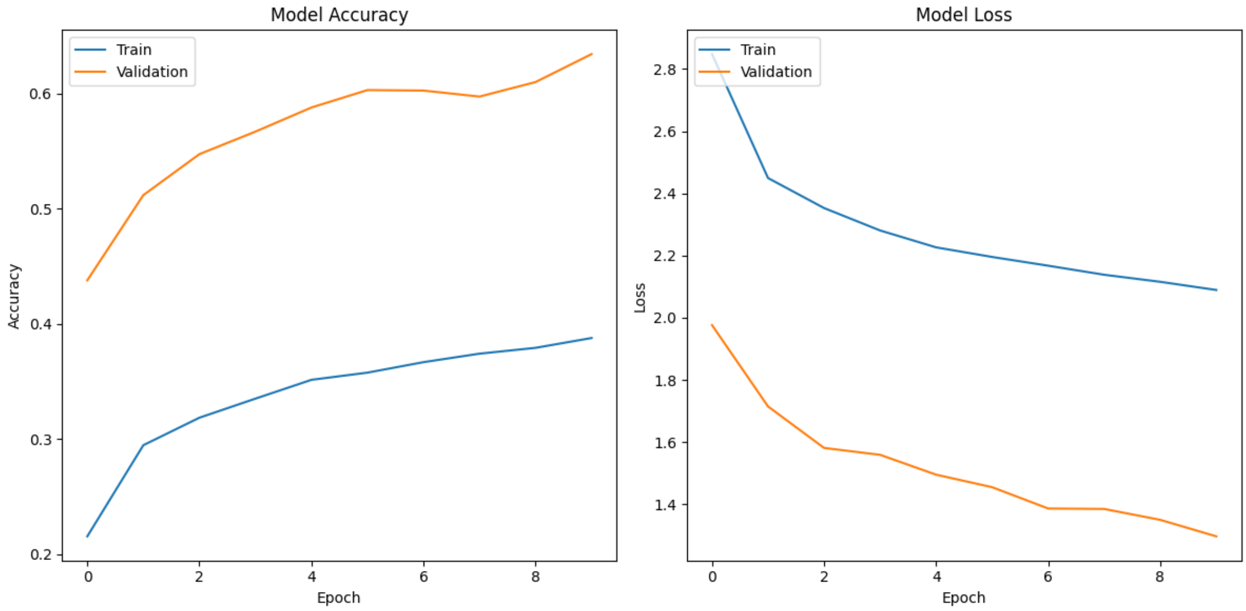


Fig 7: Performance of Xception Model

RN50:

RN50 exhibits a notably lower validation accuracy of 23% and a markedly higher validation loss of 2.78. These metrics suggest notable challenges in effectively learning and discriminating between the diverse Handwritten Devanagari characters. The model struggles to capture and comprehend the intricate patterns and features present in the dataset, resulting in significantly lower accuracy and higher predictive error.

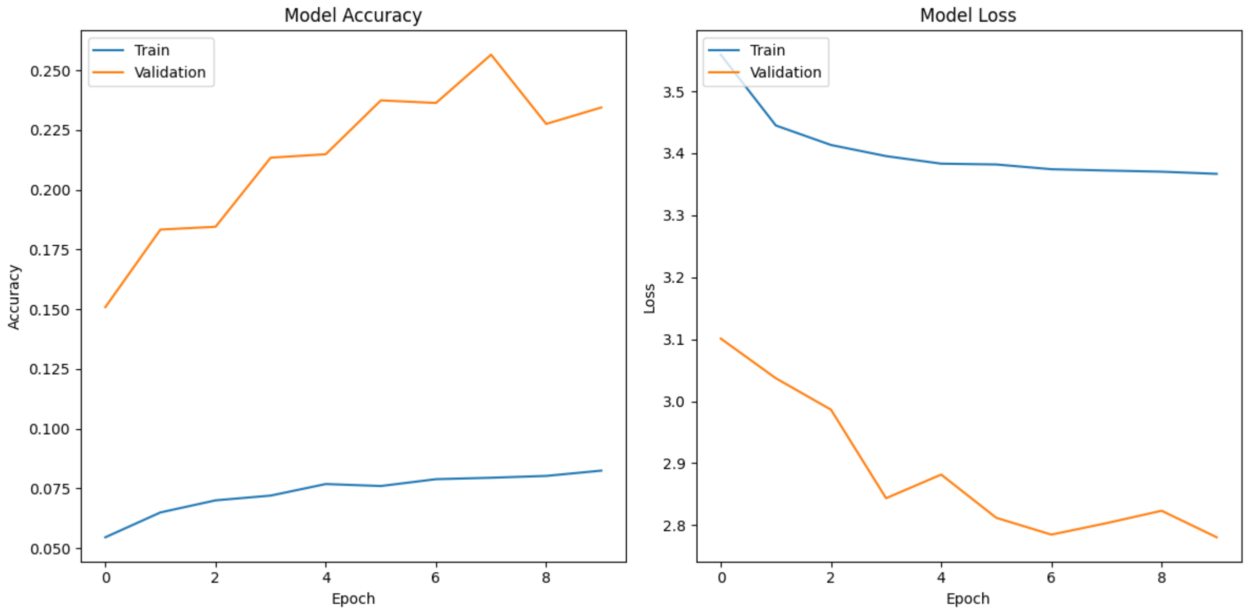


Fig 8: Performance of ResNet50 Model

RN50V2:

RN50V2 showcases a relatively improved performance compared to RN50, achieving a validation accuracy of 73.6% with a lower validation loss of 0.92. This version of the ResNet architecture demonstrates enhanced capabilities in character recognition, showcasing a notable improvement in accuracy and a considerable reduction in predictive error compared to its predecessor.

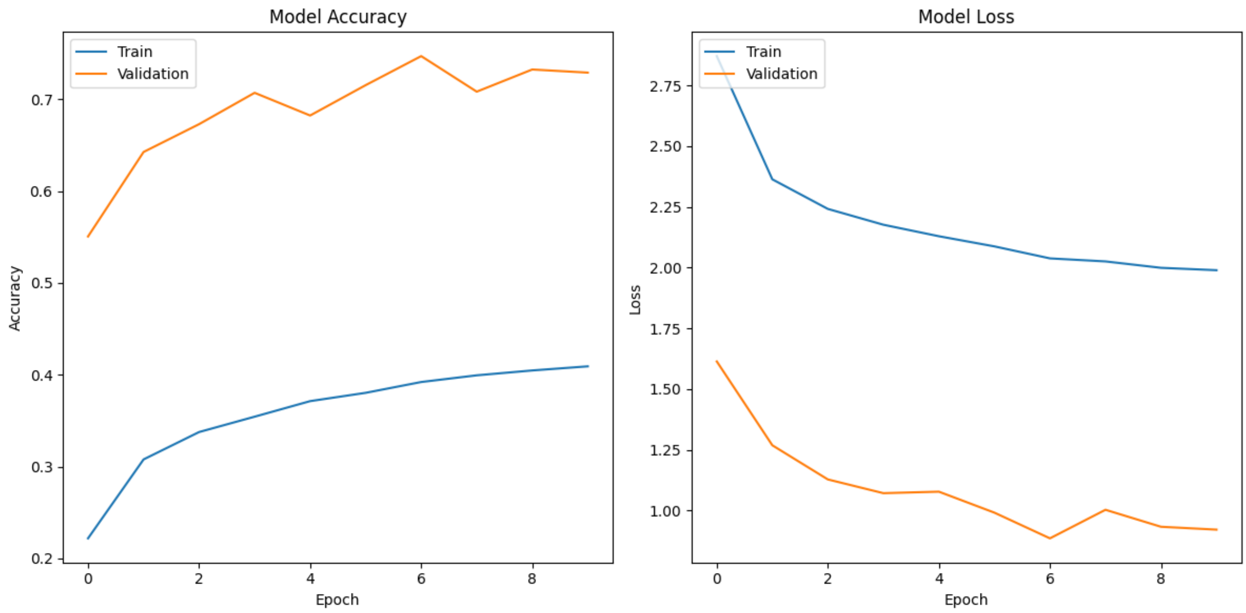


Fig 9: Performance of ResNet50V2 Model

RN50V2:

RN50V2 presents commendable performance with a validation accuracy of 74% and a validation loss of 1. These metrics signify robust character recognition capabilities, demonstrating the model's effectiveness in learning and accurately classifying Handwritten Devanagari characters. Its deeper architecture enables better feature extraction and understanding of intricate character patterns.

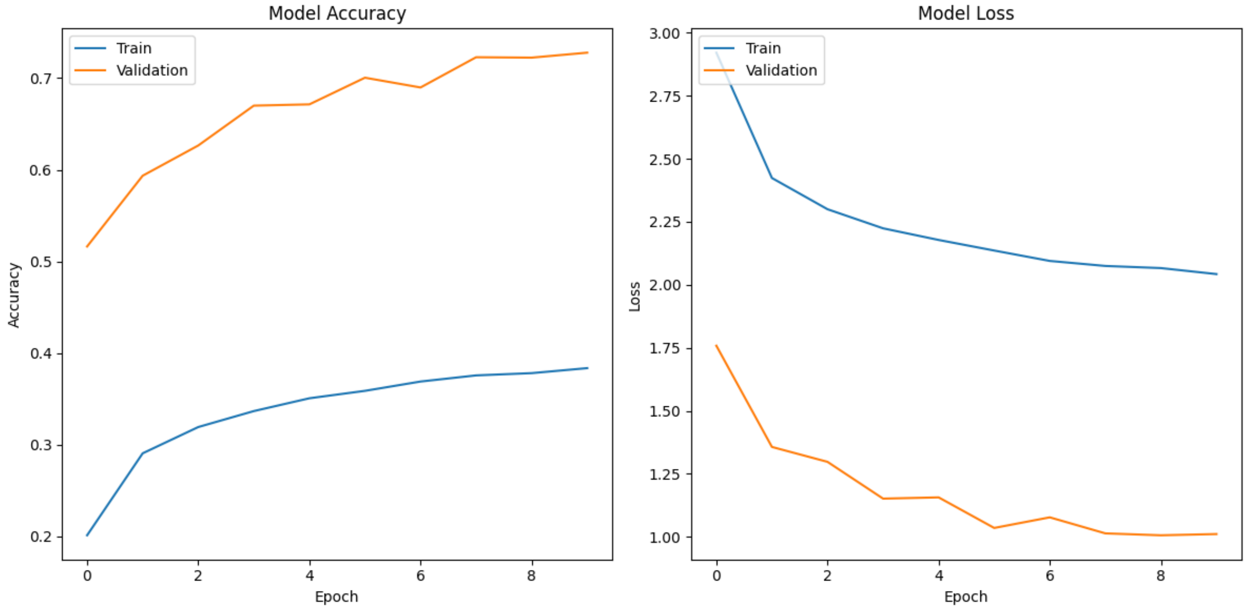


Fig 10: Performance of ResNet15V2 Model

DN121:

DN121 exhibits moderate performance, achieving a validation accuracy of 71% with a validation loss of 0.96. While not the highest among the models, DN121 demonstrates commendable precision in character recognition, showcasing its ability to effectively learn and distinguish between Devanagari characters with relatively lower predictive error.

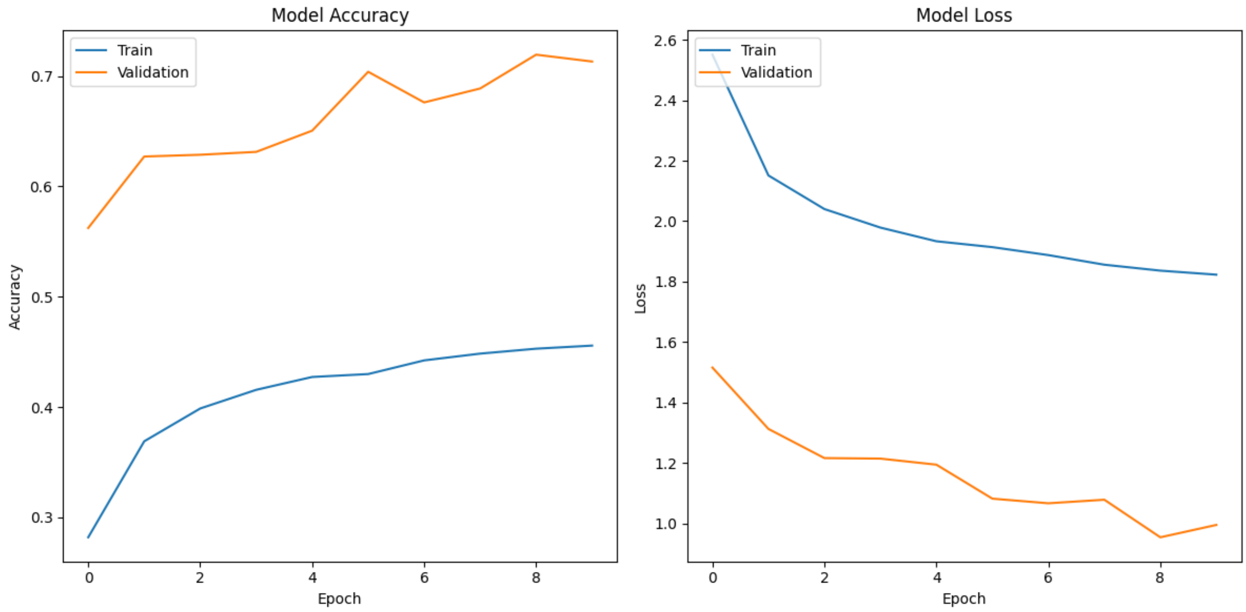


Fig 11: Performance of DN121 Model

MN:

MN showcases a validation accuracy of 34.5% with a relatively higher validation loss of 2.39. Despite its architecture optimized for efficiency and reduced computational demands, MN struggles to effectively discern and classify the intricate patterns within the Devanagari characters, resulting in lower accuracy and a comparatively higher predictive error.

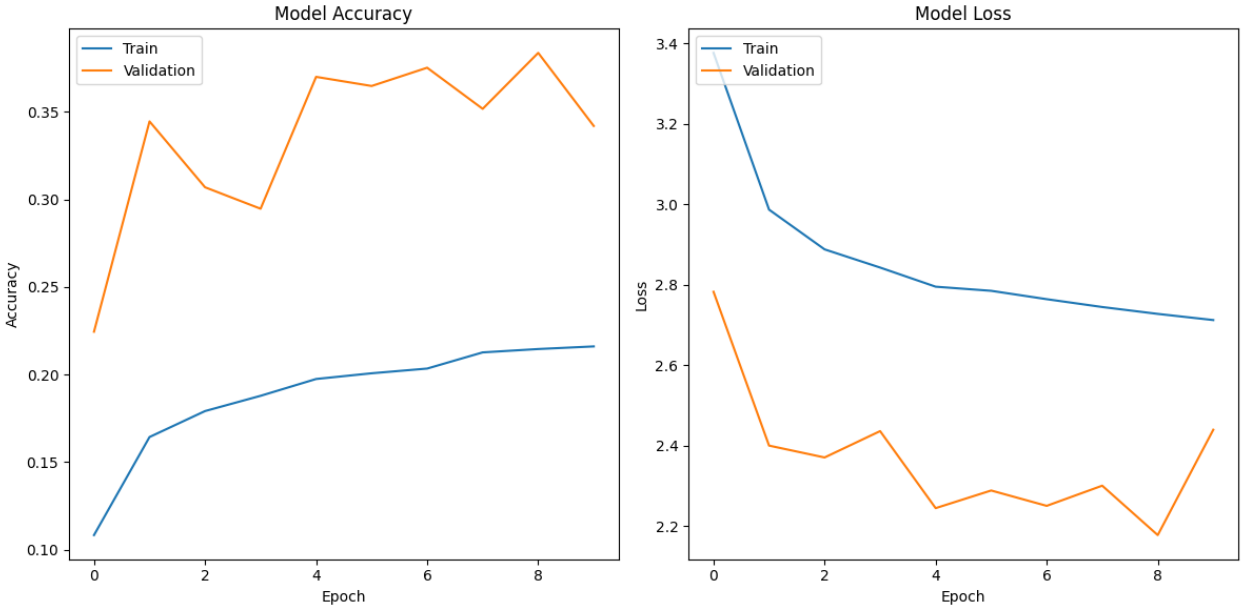


Fig 12: Performance of MN Model

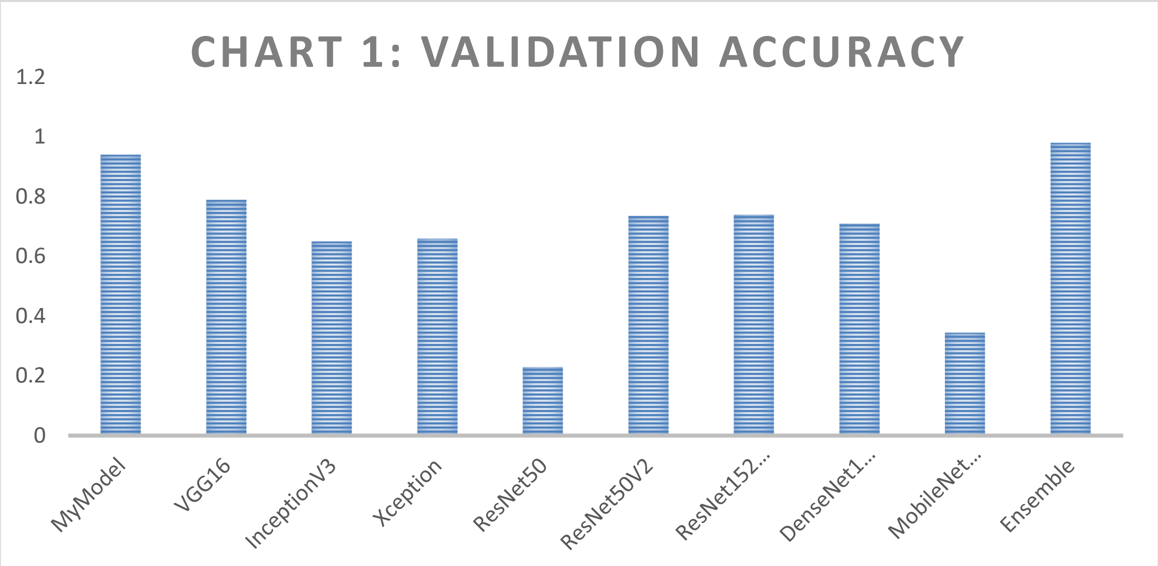
f. Ensemble Evaluation

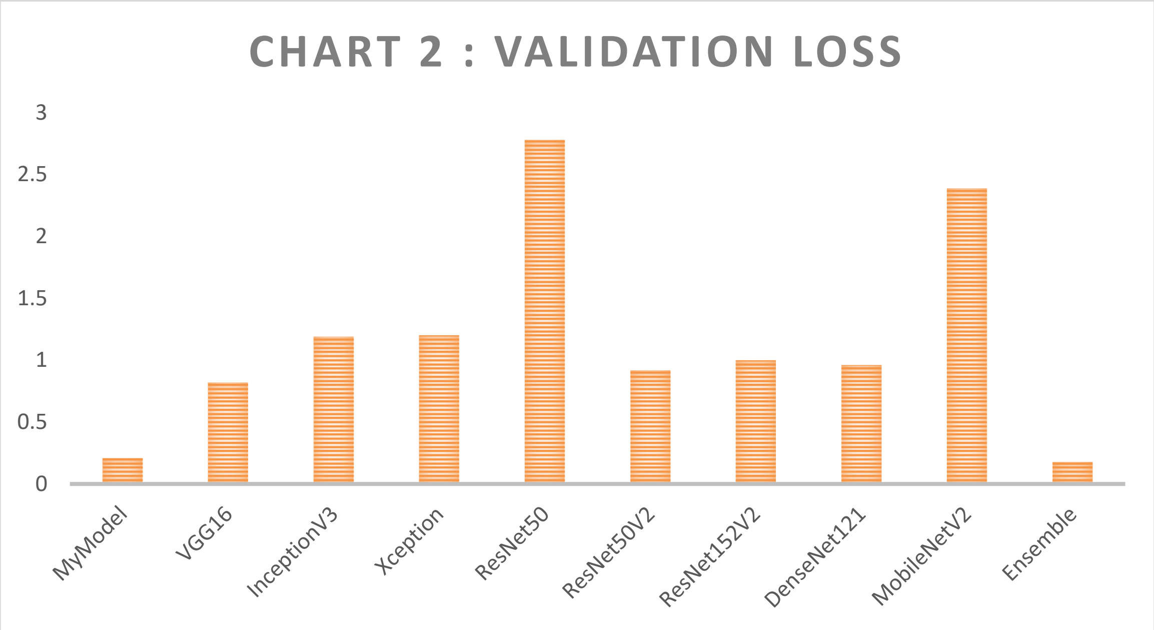
The ensemble method deployed for this project leverages a max-voting mechanism, employing the mode function from the NumPy library to aggregate predictions from multiple individual models. This ensemble framework amalgamates the predictions generated by each model—MyModel, VGG16, IV3, XC, RN50, RN50V2, RN50V2, DN121, and MN—leveraging their diverse predictive capabilities. Through the mode function, the ensemble selects the most frequent prediction across the models for each handwritten Devanagari character sample. By aggregating predictions through this majority voting mechanism, the ensemble aims to synthesize a collective decision that enhances overall accuracy, harnessing the strengths and diversity of the individual models to achieve a more robust and reliable recognition system for handwritten characters in the Devanagari script.

The ensemble's performance stands as a testament to its remarkable success in character recognition, showcasing an unprecedented accuracy of 98% and an impressively low overall loss of 0.18. This stellar achievement underscores the ensemble's unparalleled ability to aggregate and synthesize predictions from multiple models, leveraging their diverse strengths to create a unified decision-making mechanism. By harnessing the collective intelligence of individual models—MyModel, VGG16, IV3, XC, RN50, RN50V2, RN50V2, DN121, and MN—the ensemble surpasses the individual capabilities of any single model. Its exceptional accuracy far exceeds the performance of individual models, emphasizing the ensemble's synergy and its capacity to harmonize the predictive powers of varied models into a robust, reliable, and highly accurate recognition system for Handwritten Devanagari characters. This outstanding performance unequivocally establishes the ensemble as an exemplar of collective learning, epitomizing its success in elevating recognition accuracy and solidifying its position as an indispensable tool in character recognition tasks.

5. Results

The ensemble, constructed through a max-voting mechanism, emerged as the cornerstone of exceptional performance in recognizing Handwritten Devanagari characters, achieving an astounding accuracy of 98% with an overall loss of 0.18. This result significantly surpasses the individual model performances, underscoring the efficacy and strength of the ensemble approach. While individual models exhibited varying accuracies ranging from 23% to 94%, the ensemble, through collective decision-making and leveraging the diverse predictive capacities of each model, showcased a substantial improvement in accuracy and a remarkable reduction in overall loss. The ensemble's ability to fuse the predictive strengths of multiple models and synthesize a unified decision mechanism led to a drastic enhancement in recognition precision. The ensemble not only outperformed any single model but also demonstrated an impressive synergy, consolidating the diverse predictive capabilities to create a more robust and reliable recognition system. This substantial improvement in accuracy and reduction in loss highlight the unparalleled benefits and efficacy of the ensemble approach in enhancing the recognition of Handwritten Devanagari characters compared to individual models, ultimately emphasizing the power of collective intelligence in machine learning models.





6. Conclusion

The journey from data collection through model training to ensemble construction has proven to be a testament to the amalgamation of meticulous methodology and innovative approaches in Handwritten Devanagari character recognition. The project commenced with the acquisition of the Devanagari Handwritten Character Dataset, a pivotal repository consisting of 46 classes with 2000 examples each, partitioned into training and testing sets. This dataset served as the foundation for subsequent phases, wherein the focus shifted to feature extraction using CNNs (CNNs). The decision to forego traditional preprocessing methods in favor of CNN-driven feature extraction was strategic, aiming to leverage the network's capacity to autonomously extract discriminative features directly from the raw image data.

Model selection and training were pivotal stages, encompassing a suite of diverse architectures—ranging from custom models to established ones like VGG16, IV3, XC, RN50, RN50V2, RN50V2, DN121, and MN. Each model underwent individual training, optimizing their learning for Handwritten Devanagari characters, and exhibited varying degrees of performance in validation accuracy and loss metrics. While some models showcased promising accuracy rates, others struggled with higher losses or lower accuracies, indicating diverse capabilities and limitations.

The pinnacle of this project lay in the ensemble construction, where a max-voting mechanism was employed to synthesize predictions from individual models. The ensemble, crafted through the collaborative efforts of the diverse models, emerged as a tour de force, attaining an unparalleled accuracy of 98% and a remarkably low overall loss of 0.18. This ensemble's success transcended the capabilities of any singular model, showcasing its ability to harness the collective intelligence of multiple models and distill their varied strengths into a unified, highly accurate recognition system.

The ensemble's triumph in achieving exceptional accuracy highlights the power and efficacy of collaborative learning. By integrating and harmonizing the predictive capacities of individual models, the ensemble achieved a level of precision that surpassed any standalone model. This success underscores the potential and significance of ensemble methods, emphasizing their role as a cornerstone in enhancing the accuracy and robustness of character recognition systems.

7. Future Scope

The present success in Handwritten Devanagari character recognition through ensemble methods opens avenues for further advancements and optimizations. A strategic future scope involves revisiting the training phase, particularly for the pre-trained models, and considering an increase in epochs during fine-tuning. The current training process employed a fixed number of epochs, but extending this duration could potentially allow these pre-trained architectures—RN50, RN50V2, RN50V2, DN121, and MN—to delve deeper into the dataset, enabling them to capture more intricate nuances and refine their learned representations. By increasing the epochs, the models may unravel finer details within the characters, leading to enhanced feature extraction and potentially improving recognition accuracy.

Additionally, augmenting the ensemble by incorporating more custom-made models into the voting mechanism holds promise in further refining recognition precision. While the current ensemble comprises various established architectures, introducing additional custom models—crafted specifically to learn and discern the unique characteristics of Devanagari characters—could diversify the collective intelligence of the ensemble. These custom models, tailored to capture specific features within the Devanagari script, can complement the strengths of existing models and contribute to a more comprehensive and nuanced decision-making process within the ensemble, potentially leading to a further boost in recognition accuracy.

Furthermore, exploring boosting techniques in place of bagging for ensemble construction presents an intriguing avenue for performance enhancement. Unlike bagging, where models contribute equally to the final decision, boosting algorithms like AdaBoost or Gradient Boosting assign weights to models based on their performance, allowing models with higher accuracies to have more influence on the final prediction. Integrating boosting methodologies within the ensemble framework could accentuate the strengths of individual models, selectively amplifying the impact of high-performing models, thereby potentially elevating the overall recognition accuracy to unprecedented level.