e-ISSN:2582-7219

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

Volume 7, Issue 7, July 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.521

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707042 |

Deep Learning-based Classification and Recognition of Handwritten Mathematical Special Symbols using Speech Synthesizer

Afsha Kousar, Hamsaveni L, Manasa M R, Thejashwini B L

Department of Studies in Computer Science, Manasagangotri University of Mysore, Mysore, India

ABSTRACT: is particularly beneficial for visually This paper describes a novel approach for the mathematical symbols classification and recognition crucial for applications in education, scientific research, and document analysis. This paper presents a comprehensive study on the usage of deep learning techniques,focusing on enhancing the performance of VGG19. The dataset used in this study contains 5,190 samples divided into 10 classes. The Experiments was conducted evaluated VGG19, ResNet-50, ResNet152, ResNet18, VGG16, InceptionV3, ResNet34, AlexNet, LeNet5, InceptionResNetV2, EfficientNetB0, DenseNet121, and a customized CNN. Initially, performance metrics were recorded for each model. Among the pre-train models DenseNet121 achieved perfect classification with 100% accuracy and VGG19 performed the lowest accuracy of 91.87%. To improvise the VGG19 model, the Histogram of Oriented Gradients (HOG) features, where concatenated with the VGG19 feature, followed by Principal Component Analysis (PCA) for dimensionality reduction then the features were fed to the Support vector Machine (SVM) classifier. This proposed idea performed well in increasing the accuracy of the VGG19 model with 99.4% Furthermore, using a deep ensemble method with VGG19 achieved an accuracy of 99.7%. This study highlights the potential of combining pre-trained models with advanced techniques like ensemble methods to achieve superior classification accuracy. To further enhance the usability and accessibility of our system, we integrated a speech synthesizer that vocalizes recognized symbols. This feature impaired users, providing auditory representation of the symbols and making mathematical notions more accessible.

KEYWORDS: handwritten math symbol recognition, Deep learning, Convolution Neural Network, Speech Synthesis, Deep Ensemble method, HOG, PCA, Image Classification.

I.INTRODUCTION

Mathematical symbols play a pivotal role in conveying mathematical concepts, operations, and relationships within mathematical expression. These symbols encompass a wide range of elements, including special symbols, alpha, beta, sigma, for all, infinity, etc. Each symbol holds a specific meaning and contributes to the overall structure and understanding of mathematical expressions.The accurate recognition and classification of handwritten mathematical symbols are of paramount importance in various domains, including education, scientific research, and document analysis. The recognition of mathematical symbols presents unique challenges due to their diverse appearance and the inconsistencies observed in individual handwritten styles. Handwritten mathematical symbols can exhibit variations in size, shape, slant, and style, making it challenging to develop accurate algorithms.In recent years, deep learning, particularly Convolutional Neural Networks(CNNs), has Revolutionized the felid of image recognition, offering promising solutions to these challenges. This paper presents a comprehensive study on the classification and recognition of handwritten mathematical special symbols using advanced deep learning techniques, complemented by speech synthesis for enhanced accessibility.We conducted an extensive evaluation for thirteen pre-trained models, including VGG19, various versions of ResNet, Inception, AlexNet, LeNet, EfficientNet, DenseNet, and a custom CNN. While DenseNet121 achieved perfect classification accuracy, VGG19 exhibited the lowest initial accuracy 91.87%.To address VGG19's suboptimal performance, we employed a feature extraction strategy that involved concatenating its features with Histogram of Oriented Gradients (HOG) features. This combination was then subjected to Principal Component Analysis (PCA) for dimensionality reduction, followed by classification using a Support Vector Machine (SVM). This approach significantly improved VGG19's accuracy to 99.45%. Moreover, by incorporating VGG19 into the deep ensemble technique, we further boosted the accuracy to an impressive 99.7%.and accurate predictions. aggregation in complex classification tasks. Beyond achieving high classification accuracy, our speech synthesizer that vocalizes the recognized symbols, thereby providing an auditory representation of the mathematical content.

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707042 |

II.PROPOSED SYSTEM

The Proposed system leverages state-of-the-art deep learning techniques to classify handwritten mathematical symbols. Key components of the system include.

1. Dataset Preparation: The dataset consists of 5190 samples divided into 10 classes, sourced for the NC State University analytics website. Each image size is resized to 32x32 pixels and normalized.
2. Model Evaluation:13 pre-trained models, including VGG19, ResNet-50, ResNet-152, VGG16, InceptionV3, and DenseNet121, were evaluated for their Performance on the dataset.
3. Performance Enhancement: VGG19's initial accuracy of 91.87% was improved to 99.4% using feature extraction techniques such as Histogram of Gradients (HOG), Principal Component Analysis (PCA) for dimensionality reduction, and an SVM classifier. Further enhancement using a deep ensemble method achieved an accuracy of 99.7%.
4. Speech Synthesizer Integration: A speech synthesizer is integrated into the system to provide auditory feedback for recognized symbols, improving accessibility for visually impaired users.

A. DATASET INFORMATION

The dataset used in this study is crucial for training and evaluating the proposed system. It comprises 5190 samples of handwritten mathematical symbols, categorized into 10 distinct classes. These symbols include alpha, beta, sigma, infinity, belongs to, for all, exists, empty set, Delta, and Integral. Each image in the dataset is resized to 32x32 pixels to ensure uniformity and facilitate model training. The dataset was collected from the http://analytics.ncsu.edu/ website and underwent preprocessing, including manual relabeling to correct mislabels and data augmentation to enhance model generalization. The dataset was split into training and testing sets, with 20% reserved for validation to monitor and prevent overfitting.

III.DEEP LEARNING PRE-TRAIN MODELS

A. VGG 19

This study investigates the application of a pre-trained VGG19 model for the task of symbol classification. To prepare the dataset, we collected images of various symbols and resized them to a consistent size of 32x32 pixels to maintain uniformity. The images were normalized by dividing pixel values by 255.0, converting them to a range of [0,1], which is suitable for training deep learning models. We utilized a batch size of 32 to manage the computational load during training. The dataset was split into training and testing sets with 20% of the data reserved for validation purposes to monitor the model's performance and prevent overfitting. We employed the VGG19 model, pre-trained on the ImageNet dataset, as our base model. The VGG19 architecture, known for its depth an ability to capture intricate features, consists of 19 layers including a convolutional layer, pooling layers, and fully connected layers. For our specific task, we excluded the top fully connected layers including a flattened layer to convert the 3D feature maps into a 1D feature vector, a dense layer with 512 units and ReLU activation to introduce non-linearity, a dropout layer with a 0.5.rate to mitigate overfitting, and an output dense layer with 10 units and softmax activation to produce probabilities for each class. After training, the fine-tuned VGG19 model achieved a test accuracy rate of 0.918%.

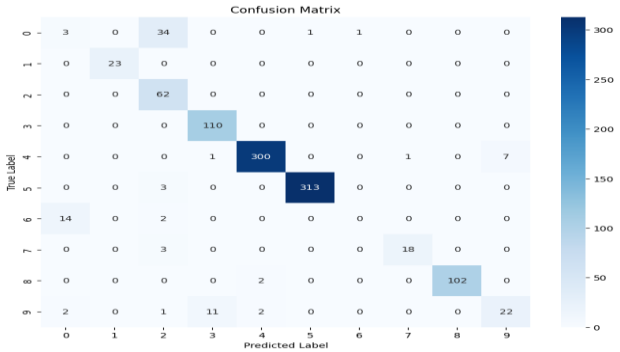

Fig.1: VGG19

B.Resnet-50

Deep learning in symbol classification using a pre-trained ResNet-50 model. Symbol classification plays a crucial role in various fields, including automated systems for interpreting and processing visual data. The above VGG -19 contains the

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707042 |

methodology that was followed for all other pre-trained models.ResNet-50, a convolutional neural network (CNN) architecture, is chosen for its deep layers and residual connections, which facilitate better gradient flow during training. The pre-trained ResNet-50 model, initially trained on ImageNet, is used as a feature extractor. The top layers are removed, and a custom layer is added, including a fattening layer to convert 3D feature maps into 1D vectors, dense layers with ReLU activation for non-linearity, dropout layers for regularization, and an output layer with softmax activation for multi-class classification.The model is compiled with the Adam optimizer and sparse categorical cross-entropy loss function, suitable for mut-class classification tasks. Training is conducted over 10 epochs, where the model learns to minimize loss and optimizer accuracy. During training, validation accuracy and loss are monitored to prevent overfitting and ensure generalization.After training, the fine-tuned ResNet-50 model achieves an impressive test accuracy rate of 0.971%. Evaluation metrics such as accuracy, precision, recall, and F1 score provide a comprehensive assessment of the model's performance. A confusion matrix is employed to visualize class-wise prediction, highlighting areas of strong and weak classification.

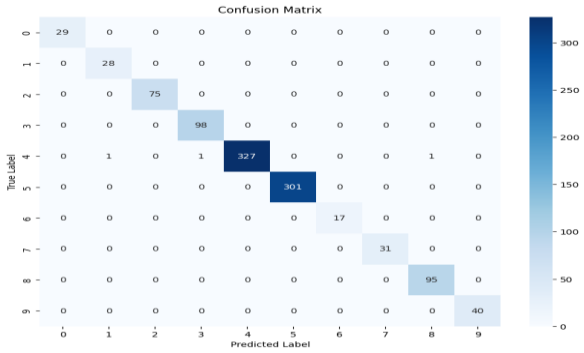

Fig.2: ResNet-50

C.ResNet-152

The ResNet-152 model architecture consisted of convolutional layers with batch Normalization and ReLU activation. The architecture began with an initial convolutional layer with a 7x7 kernel and a stride of 2, followed by a Max pooling layer with a 3x3 pool size and a stride and stride 2. This was followed by multiple residual blocks and convolutional blocks across different stages:

- Conv2_x stage:3 identity blocks with 64 filters each.
- Conv3_x stage: 1 convolutional block with 128 filters and a stride of 2, followed by 7 identity blocks.
- Conv4_x stage: 1 convolutional block with 256 filters and a stride of 2 followed by 35 identity blocks.
- Conv5_x stage: 1 convolutional block with 512 filters and a stride of 2, followed by 3 identity blocks.

The model was compiled using the Adam optimizer and sparse categorical cross-entropy loss function. The training was conducted for 10 epochs with a batch size of 32, and the training process was monitored to ensure that both training and validation accuracy and loss improved over time. Upon evaluation, the model achieved a test accuracy of 0.9808%.

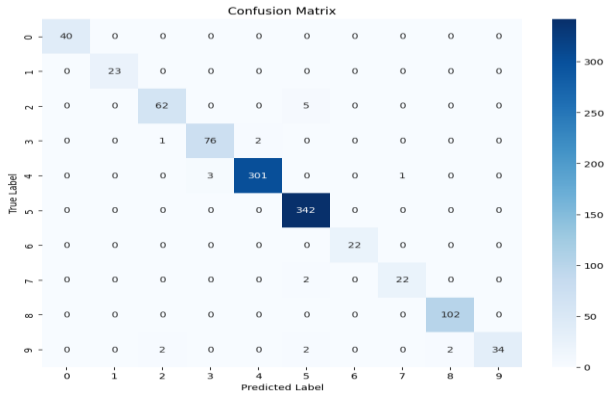

Fig.3: ResNet-152

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707042 |

D.ResNet-18

The model architecture followed the ResNet-18 design, which features convolutional layers with Batch Normalization and ReLU activation. Residual blocks were employed to help mitigate the vanishing gradient problem, enabling the training of deep networks. Specifically, the architecture began with an initial convolutional layer with 7x7 Kernal and a stride of 2, followed by a Max pooling layer with a 3x3 pool size and a stride of 2. This was followed by two identity blocks with 64 filters each, maintaining the dimensions. The network then incorporated a convolutional block with 128 filters and a stride of 2, succeeded by an identity block. This pattern was repeated with convolutional blocks of 256 and 512 filters, each followed by an identity block. The model concluded with a Global Average Pooling layer and a Dense layer with 10 units, corresponding to the 10 symbol classes, using SoftMax activation for the final classification.The model was compiled using the Adam optimizer and categorical cross-entropy loss function. The training was conducted for 10 epochs with a batch size of 32, and the training process was monitored to ensure that both training and validation accuracy and loss improved over time. Upon evaluation, the model achieved a test accuracy of 0.983%.

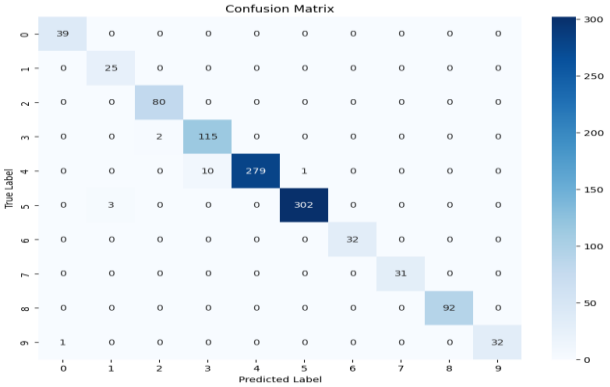

Fig.4: ResNet-18

E. VGG-16

The VGG16 model is loaded with pre-trained weights, excluding the top fully connected layers. Custom layers are added on top of the base model, including a flattened layer, a Dense layer with 512units and ReLU activation, a Dropout layer with a rate of 0.5, and a final Dense layer with 10 units using SoftMax activation for classification.The model is compiled using the Adam optimizer and sparse categorical cross-entropy loss function. Training is conducted for 10 epochs with a batch size of 32, and the training process is monitored to ensure both training and validation accuracy and improve over time. Upon evaluation, the model achieves a test accuracy of 0.995%.

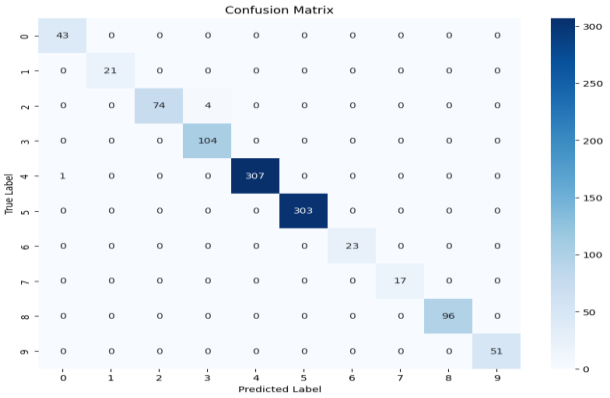

Fig.5: VGG-16

F. Inception-V3

This demonstrates a sophisticated approach to symbol classification using the powerful InceptionV3 model, pre-trained on ImageNet. The workflow starts by loading and preprocessing the dataset, resizing images to 75x75 pixels to match the InceptionV3 input requirements, and normalizing pixel values for optimal performance. Build upon the InceptionV3

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707042 |

model, known for its superior image classification capabilities, by adding custom layers tailored to our specific classification task. The architecture includes GlobalAveragePooling2D layers, a 512-unit Dense layer, a 512-unit Dense layer with ReLU activation, a Dropout layer to prevent overfitting, and a final softmax layer for classification into 10 distinct classes. The model is trained using the Adam optimizer and sparse categorical cross-entropy loss, achieving impressive results with a test accuracy of 0.996%. Comprehensive evaluation metrics such as accuracy, precision, recall, and F1 score are calculated, showcasing balanced and high performance and areas for improvement.

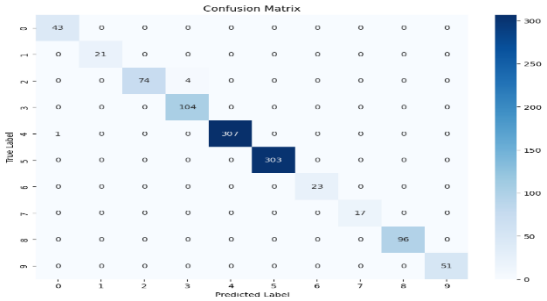

Fig.6: InceptionV3

G. ResNet-34

This paper focuses on advanced symbol classification using a custom ResNet-34 deep-learning architecture. The task begins with preparing the dataset, constructing and training the Resnet-34 model, and evaluating its performance through various metrics.The model is compiled using the Adam optimizer, known for its efficiency and effectiveness in training deep neural networks. The loss function used is sparse categorical cross-entropy, which is suitable for multi-class classification tasks. The model is then trained for a set number of epochs, during which it learns to classify the symbols accurately. The training process is monitored using validation using data to prevent overfitting and ensure that the model generalizes well to unseen data.After training, the model's performance is evaluated using the test set. Several performance metrics can calculated, accuracy, precision, recall, and F1 score, providing a comprehensive understanding of the model's classification capabilities.

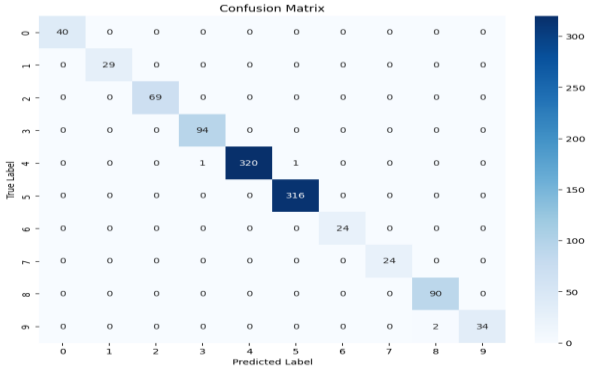

Fig.7: ResNet-34

H. AlexNet

The AlexNet model Architecture is built by stacking several layers sequentially. It begins with a Convolutional Layer using 96 filters of size 11x11 with a stride of 4x4, followed by a max pooling layer to reduce the spatial dimensions. This is followed by a second convolutional layer with 256 filters of size 5x5, another max-pooling layer, then three more convolutional layers with384,384, and 256 filters respectively, all with 3x3 filters. These layers extract various levels of features from the input images. After these convolutional layers, a series of fully connected dense layers are added: two layers with 4096 neurons each and a dropout rate of 0.5 to prevent overfitting, followed by a final dense layer with the number of neurons equal to the number of class, using a softmax activation function to produce probability distribution over the classes. The model is compiled using Adam optimizer and sparse categorical cross-entropy loss function, and then trained on the training data for 10 epochs, with validation on the validation set.

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707042 |

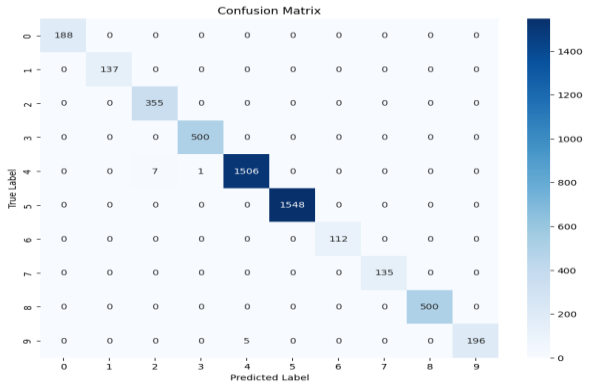

Fig.8: AlexNet

I. LeNet-5

LeNet-5 architecture for the classification of grayscale symbol images. The data, consisting of images resized to 32x32 pixels, Is loaded and processed testing subset to evaluate the model's performance.The LeNet-5 model is built using the Keras Sequential API, starting with a convolutional layer (C1) that applies 6 filters of size 5x5 starting with a tanh activation, followed by an average pooling layer (S2). This is followed by another convolutional layer (C3) with 16 filtered of size 5x5 and a second average pooling layer using softmax activation for classification into 10 classes. The model is compiled with the Adam optimizer and trained using sparse categorical cross-entropy loss for 10 epochs, with both training and validation data provided.

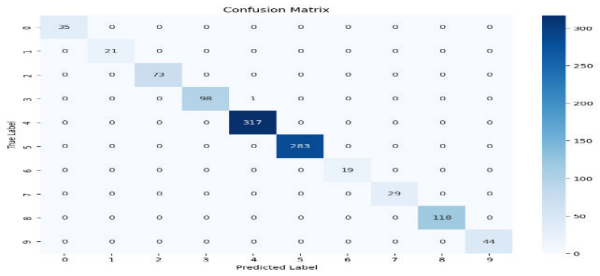

Fig.9: LeNet-5

J. EfficientNet-B0

The images are resized to 224x224 pixels and normalized, ensuring compatibility with the pre-trained model's input specifications. The EfficientB0 model, known for its efficiency and accuracy, is employed with weights pre-trained on the ImageNet dataset. The base model, excluding its top classification layer, is integrated with a custom layer: a global pooling layer followed by a dense layer with 10 units and a softmax activation function, aligning with the number of symbol classes in the dataset. The data preprocessing pipeline involves TensorFlow's 'ImageDataGenrator' for real-time data augmentation and normalization. The dataset is split into training and validation sets with an entropy loss function, and trained for 10 epochs. Training and validation accuracy, along with loss, are tracked and visualized to monitor the model's performance. Following the base model, a Global Average Pooling 2d layer is employed to condense the spatial dimensions of the feature map into a vector representation. This is then connected to the Dense layer with 10 units.

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707042 |

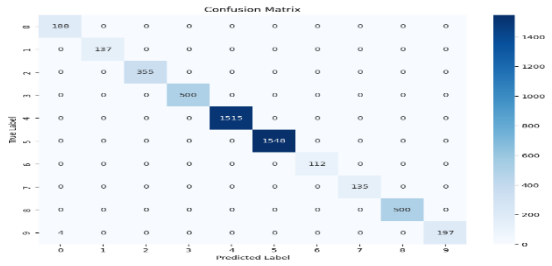

Fig.10: EfficientNetB0

K. DenseNet121

This paper presents an implementation of the DenseNet121 convolutional neural network architecture for The dataset is loaded using TensorFlow's 'ImageDataGenerator', with images resized to (224,224) pixels and normalized to a range of [0,1]. A validation split of 20% is applied to ensure model generalization and prevent overfitting. DenseNet12, pre-trained on ImageNet, is employed as the base model using transfer learning. A global average pooling layer aggregates spatial information followed by a dense layer with softmax activation to output class probabilities across 10 classes.The model has complied with the Adam optimizer and sparse categorical cross-entropy loss function, ideal for multi-class classification tasks. Training is conducted for 10 epochs with batch sizes . Performance metrics including accuracy, precision, recall, and F1-score are computed using clear metrics. for testing, a generator is created similarly to training and validation. the model achieves an exceptional test accuracy of 100 %, reflecting its capability to generalize well to unseen data.

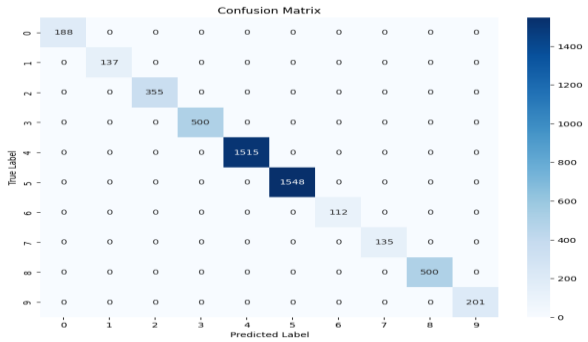

Fig.11: DensNet121

Table 1. Results Obtained from Pre-trained models

Pre-trained models	Precision	recall	F1-score	Test accuracy
VGG-19	0.9032	0.9181	0.906	0.9187
ResNet-50	0.9717	0.9717	0.9717	0.9808
ResNet-152	0.9809	0.9808	0.9805	0.9837
ResNet-18	0.9847	0.9837	0.9838	0.9952
VGG-16	0.9953	0.9952	0.9952	0.9961
ResNet-34	0.9971	0.9971	0.9971	0.9971
AlexNet	0.9975	0.9974	0.9974	0.9974
LenNet-5	0.9981	0.9984	0.9980	0.9980
Inception-ResNet-v2	0.999	0.999	0.999	0.999
EfficientNet-B0	0.999	0.999	0.999	0.999
DensNet121	1.0	1.0	1.0	1.0

L.Achieving High Accuracy in Mathematical Symbol Classification Using Hybrid VGG19 and HOG Features with SVM

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707042 |

By combining VGG-19 derived deep features and Histogram of Oriented Gradients (HOG) features, and subsequently applying Principal Component Analysis (PCA) and Support Vector Machine (SVM) classification, the proposed approach achieves high accuracy in classifying symbol images. By integrating deep learning techniques with traditional feature extraction methods, we aim to overcome the limitations of using VGG19 alone, which yielded a test accuracy of 0.908%.The goal was to significantly enhance the accuracy of mathematical symbol classification, leading to the adoption of a hybrid approach combining deep learning and traditional feature extraction methods. Each image is converted to grayscale and resized to 224x224 pixels, the input size required for the VGG19 model. Two complementary methods are used to extract features;

1.VGG19 for Deep Feature Extraction: VGG19 is selected for its deep architecture and proven effectiveness in capturing intricate features from images. By fine-tuning VGG19 for symbol recognition, we leverage its ability to extract high-level features crucial for accurately identifying mathematical symbols. This choice is motivated by VGG19's established performance in image classification tasks and its capability to handle complex visual patterns.

2. Complementary HOG Features: In conjunction with VGG19, HOG is utilized to capture fine-grained texture and edge information from grayscale images. This method c

3. Dimensionality Reduction with PCA: To manage the high-dimensional feature space derived from VGG19 and HOG, Principal Component Analysis (PCA) is employed. PCA reduces dimensionality while preserving the variance necessary for accurate classification.

4. SVM for Effective Classification: The Support Vector Machine (SVM) with a linear kernel is chosen for its ability to handle high-dimensional data and provide effective separation between classes.Each image was resized to 224x224 pixels, normalized to a range of [0,1], and processed batches of 32 to manage the computational load. The dataset was split into training and testing sets, with 20% reversed for validation to monitor the model's performance and prevent overfitting pre-trained VGG19 model, excluding the top fully connected layers, was used to extract deep features from the images. The modified architecture included a flattened layer to convert the 3D feature map into 1D feature vectors, a dense layer with 512 units and ReLU activation, a dropout layer with 0.5 rate, and an output dense layer with 10 units and softmax activation.The deep features from VGG19 and Hog features were flattened and concatenated, creating a comprehensive feature set. This comprehensive evaluation demonstrates the model's robustness and reliability in symbol classification tasks Notably, the combination of VGG19 and HOG features significantly improved the classification accuracy, resulting in a test accuracy rate of 99.41%.

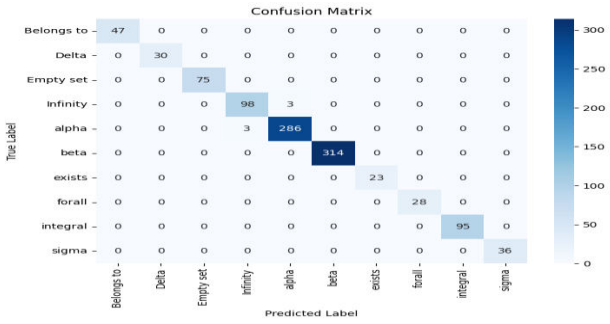

Fig.12: VGG-19 with HOG

Table 2. Result Obtained From Integration Of HOG with VGG-19

VGG-19 without HOG	VGG-19 with HOG
Test accuracy	Test accuracy
0.908%	0.9941%

M. Enhancing Mathematical Symbol Classification With Deep Ensemble Learning Using VGG16, VGG19, ResNet-50, and ResNet-152

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707042 |

This study focuses on the classification of mathematical symbols using a deep ensemble learning approach. By integrating the strengths of multiple pre-trained models, including VGG16, VGG19, ResNet-50, and ResNet-152, The goal was to improve upon the performance achieved with individual models and traditional feature extraction methods. Initially, VGG19 was dine-tuned for the task of mathematical symbol classification, achieving a test accuracy of 0.997%. To further enhance the accuracy, Histogram of Oriented Gradients (HOG) features were combined with VGG19, resulting in a significant improvement with an accuracy of 0.9941%..Despite the high accuracy achieved with VGG19 and HOG the complexity and diversity of mathematical symbols warranted more robust solutions. Consequently, The ensemble aimed to address the limitations of individual models and ensure higher accuracy and generalization. Images of mathematical symbols were collected and preprocessed. Each image was resized to 224x224 pixels to fit the input requirements of the pre-trained models. The dataset was split into training and testing sets, with 20% reserved for validation to monitor performance and prevent overfitting. Labels were encoded to facilitate a classification ensemble comprised of pre-trained models: VGG16, VGG19, ResNet50, and ResNEt152.Each model was modified by excluding the top fully connected layers and incorporating a GlobalAveragePooling2D layer to convert the 3D feature maps into 1D feature vectors. The output of the base models was concatenated to form a comprehensive feature representation. Dense layers were added to this concatenated output for classification, including a dense layer with 512 units and ReLU activation, followed by a softmax output layer corresponding to the number of classes. This model was compiled with an Adam optimizer and sparse categorical cross-entropy loss. It was trained for 10 epochs with a batch size of 32, utilizing the training set.

Table.3 Results obtained from the Deep Ensemble model

VGG19 Without deep ensemble	VGG-19 with deep ensemble
Test accuracy	Test accuracy
0.9941%	0.997%

After Studying the efficacy of the classifiers the proposed model was further tested for the Recognition Phase. In this Phase, the data has been trained to the CNN model for Recognition of symbols. These symbols are given to the speech synthesizer for conversion from Predicted text output to audio output.

IV.CONCLUSION

This study presents a novel approach to the classification and recognition of handwritten mathematical symbols, utilizing deep learning techniques enhanced with ensemble methods. By integrating VGG19 with HOG features and employing PCA for dimensionality reduction, we significantly improved the classification accuracy of the VGG19 model to 99.4% Further enhancement was achieved through a deep ensemble method, incorporating VGG16, VGG19, ResNEt50, and ResNet152, leading to an impressive accuracy of 99.7%.Our comprehensive evaluation demonstrated the potential of combining pre-trained models with advanced techniques to achieve superior classification accuracy. Additionally integrating a speech synthesizer provided an auditory representation of recognized symbols, enhancing accessibility for visually impaired users.Overall, this study underscores the feasibility and effectiveness of hybrid deep learning methods advancing the field of handwritten mathematical symbol recognition.

REFERENCES

[1] L. Dong and H. Liu, “Recognition of offline handwritten mathematical symbol using convolutional neural networks,” International Conference on Image and Graphics,2017, pp. 149-161.
[2] L. D’souza and M. Mascsrenhas,” offline handwritten mathe mathematical expression recognition using convolutional neural network,” in 2018 International Conference on Information, Communication, Engineering, and Technology(ICICET), August 2018, pp. 1-3.
[3] K.K Ayeb, Y.Meguebli, and A.K.Echi,” Deep learning Architecture for Off-line Recognition of Handwritten Math symbols, in pattern recognition and Artificial Intelligence: 4th Mediterranean Conference, MedPRAI 2020, Hammamet, Tunisia, December 20-22, 2020, pp.200-214, Springer International Publishing.
[4]P.Forcha and T.O. Adewumi, “Hand-written Mathematical Symbols Classification Using Convolutional Neural Network Model, “Experiment Findings, May 2024.DOI:10.13140/RG.2.2.24161.52320.
[5] V.Kukreja and Sakshi,” Machine Learning models for mathematical symbol recognition: A stem to stern literature analysis,” Multimedia Tools and Application, vol.81, pp. 2865128687, Mar. 2022.DOI:10.1007/s11042-022-12644-2.

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707042 |

[6]Y.Zhang," Investigation on handwritten mathematical symbol recognition based on the combining of CNN and KNN method", in Proceedings of the 2023 International Conference on Image, Algorithm and Artificial Intelligence (ICIAAI 2023), Advances in Computer Science Resarch108,2023, pp.639-645.DOI:10.2991/978-94-6463-300-9_66.
[7]P.Forcha and T.O.Adewumi, "Hand-written mathematical symbol classification using Convolutional neural network model," Experiment Finding, May 2024, pp.1-5.DoI:10.13140/RG.2.2.24161.52320.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com