10.48047/jocaaa.2024.33.07.27

Digitization and classification of Ancient Kannada Handwritten Palm Leaf Manuscript based on Author using VGG19 Model

Parashuram Bannigidad[1], Shivakumar Kalburgi[2] and S. P. Sajjan[3]

[1,2,3]Dept. of Computer Science, Rani Channamma University, Belagavi- 571159, India
parashurambannigidad@gmail.com, Kalburgi_shivakumar@rediffmail.com and sajjanvsl@gmail.com

Abstract: This study examines deep learning approaches for classifying ancient Kannada handwritten palm leaf manuscripts. We classify historical Kannada literature using the VGG19 neural network model, which addresses diverse literary styles, historical events, and ecological impacts. We used a dataset of 1025 historical Kannada images of palm leaf manuscripts and we preprocessed them for training, testing, and validation. The data set contains 807 images for training, 99 for validation, and 119 for testing. The VGG19 model achieved remarkable accuracy and low loss metrics, highlighting its efficacy. The system achieved 99.62% accuracy in training and 98.03% in validation, correctly identifying and sorting ancient manuscripts. These findings are significant for preserving cultural heritage, showing that computers can now automatically sort and identify historical Kannada manuscripts accurately. This research contributes to digital humanities by offering a robust automated classification methodology.

Keywords: *Deep learning, VGG19, ancient Kannada manuscripts, cultural heritage, handwritten classification, convolution neural networks, palm-leaf manuscripts, transfer learning.*

1. Introduction

The main goal of this in-depth study is to find out if it is possible to identify writers in primary Kannada manuscripts by using classification and the VGG19 neural network model. Historical documents from various eras present unique challenges due to their cultural significance, diverse writing styles, and natural deterioration. The goal is to create a complex system that uses deep learning models to automatically figure out who wrote historical Kannada texts, which make up a large body of literature. Parashuram Bannigidad and Chandrashekar Gudada suggested using the Histogram of Oriented Gradients (HOG) feature descriptors to help find and figure out the age of handwritten historical Kannada documents [1]. Khayyat et al. [2] have proposed employing deep learning with multiple fusion levels for image retrieval from texts. The authors classified image networks using DCNNs. Krizhevsky et al. [3] noted recent advancements in neural processors for information. Adnan Khashman [4] introduced the intelligent coin identification system. Demilew, F.A., and B. Sekeroglu [5] recommended using deep learning for ancient Geez script recognition. Danukusumo, Kefin Pudi, and Martinus Maslim [6] reported on the use of CNN models to classify ancient Indonesian temples. Khandagale et al. [7] created the AlexNet model for historical currency coin recognition. Rehman et al. [8] discussed multimedia-handling software in the context of evaluating machine-learning techniques for writer identification. Saddami et al. [9] developed a DNN-based approach for classifying the degradation of ancient text images. Barucci Andrea et al. [10] proposed a deep-learning system for classifying ancient Egyptian hieroglyphs. Parashuram Bannigidad and Chandrashekar Gudada [11] suggested the LBP features to recognize and categorize ancient Kannada handwritten images. Wahdan Ahlam et al. [12] provided a

10.48047/jocaaa.2024.33.07.27

comprehensive review of Arabic text categorization studies using deep learning models. Da, Shaveta, and Munish Kumar [13] demonstrated writer identification through offline handwritten materials in Gurumukhi script. Alsaleh et al. [14] categorized Arabic texts using convolutional neural networks and genetic algorithms. Parashuram Bannigidad and Chandrashekar Gudada [15] have developed a line segmentation technique using GLCM to distinguish ancient Kannada handwritten scripts based on age. Deep learning methods have made significant contributions in other ancient languages such as Arabic, Latin, and Greek. Hence, it is necessary to develop algorithms in this area. This paper aims to categorize handwritten palm leaf images from ancient Kannada manuscripts using deep learning techniques.

2. Proposed Method

The proposed method encompassed multiple stages, including data collection, model training, model evaluation, and performance analysis. The dataset utilized ancient Kannada script inscriptions written on palm leaves and was systematically divided into training, testing, and validation sets. While the training set was employed to train the models, the testing set was used for evaluation, ensuring accurate classification results. This systematic approach facilitated the selection of the most effective models for classifying handwritten palm-leaf manuscripts belonging to ancient Kannada. The VGG19 model served as the primary classifier for categorizing these historical texts. The VGG19 architecture incorporates 5 interconnected layers and 2 fully connected layers, specifically designed to analyze and classify the preprocessed manuscript images. By implementing transfer learning techniques and utilizing manuscripts written in Kannada on palm leaves, we successfully fine-tuned the model, enhancing its performance and classification accuracy. The research used an advanced computer system called VGG19 to identify and classify old Kannada writing with high accuracy. This method worked better than traditional approaches, showing promise for preserving and studying historical documents. The success of this project highlights how combining knowledge from different fields like computer science, language studies, and history can help us better understand and protect our cultural heritage. We could potentially use this approach to study other ancient languages and writings as well. Fig. 1 shows the proposed system architecture of the VGG19 model.

10.48047/jocaaa.2024.33.07.27

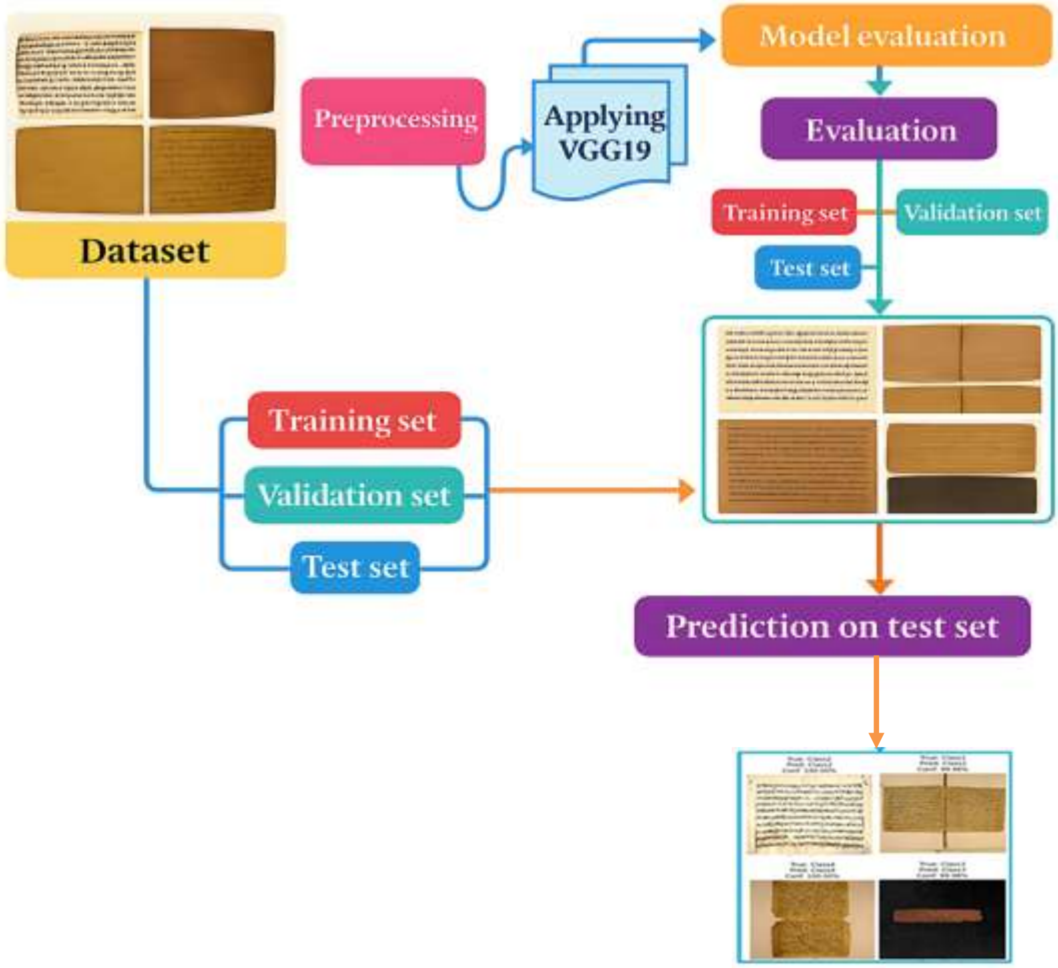


Fig 1. Proposed System Architecture of VGG19 model

2.1 Data Set

The e-Sahithya Documentation Forum in Bangalore has established itself as a significant repository of ancient Kannada manuscripts, meticulously gathered from reliable sources. This comprehensive collection showcases a wide array of manuscripts that offer valuable insights into different historical periods, diverse writing techniques, and various geographical regions within the Kannada-speaking areas. The forum's efforts in preserving these cultural artifacts contribute significantly to the study of Kannada literature, history, and linguistic evolution. To enhance the usability and accessibility of these manuscripts, the forum has implemented advanced preprocessing techniques [16]. These methods aim to improve the overall quality of the manuscript images by enhancing text clarity, which is crucial for accurate transcription and analysis. Additionally, the standardization of image dimensions ensures consistency across the collection, facilitating comparison and study of different manuscripts. Not only do these steps help keep these important historical records safe, but they also make it easier to study and analyze them digitally, which could lead to new areas of research in Kannada literature and historical linguistics.

The implementation is conducted on a Windows system utilizing the Anaconda3 Distribution, Spyder, and Python 3.7. This project is executed on a Windows system prepared with a 2.30 GHz Intel i5 processor, 8 GB of RAM, and a 4 GB GPU (NVIDIA GeForce GTX 1050

10.48047/jocaaa.2024.33.07.27

Ti). Furthermore, the machine is furnished with 8 GB of graphics memory. Fig. 2 illustrates the sample image dataset comprising various classes.

Fig. 2 illustrates the sample image dataset comprising various classes.

2.2 Preprocessing

Various techniques, including rescaling, rotation, shifting, zooming, and flipping, enhance the training data to improve model robustness and mitigate overfitting. The validation and test data were rescaled to maintain an unbiased evaluation. We source data from the directories, resize it to 224 x 224 pixels, and organize it into batches for optimal processing efficiency. Loading test data disables shuffling to ensure consistency in the prediction mapping. This pre-processing pipeline ensures a diverse range of training data and reliable assessments. Data augmentation techniques are very important for making machine learning models work better and be able to generalize, especially when it comes to image classification tasks. By changing the training data in ways like resizing, rotating, shifting, zooming, and flipping, the dataset is made to look bigger than it really is. This gives the model more examples of different situations. This process makes the model more stable when it comes to different input situations and lowers the risk of overfitting, which happens when the model becomes too dependent on the training data and can't adapt well to new examples.

The preprocessing pipeline described ensures that the data is prepared optimally for model training and evaluation. Resizing images to a uniform 224 × 224-pixel format standardizes the input, allowing for consistent processing across all samples. Batch organization improves computational efficiency during training. The decision to disable shuffling for test data is particularly important, as it maintains the order of predictions, facilitating interpretation and analysis of results. By applying rescaling to both validation and test data, we maintain consistency with the training data while preserving the integrity of the evaluation process. This comprehensive approach to data preparation and augmentation sets the foundation for developing a high-performing and reliable image classification model.

2.3 VGG19 Model:

VGG19 is a sophisticated computer program designed to analyze and understand images. Oxford University researchers developed it, incorporating 19 layers of artificial intelligence.

10.48047/jocaaa.2024.33.07.27

The program uses special filters to look at small parts of an image and gradually builds up an understanding of the whole picture, from simple shapes to complex patterns. What makes VGG19 powerful is its depth and the fact that it has been trained on millions of images. This extensive training allows it to recognize a wide variety of objects and features in new images it hasn't seen before. While VGG19 is excellent at classifying images accurately, it does need a lot of computer power to run. Despite this drawback, many people still use it because of its reliability and its ability to be applied to many different types of image recognition tasks. The VGG19 model architecture is shown in Fig. 3.

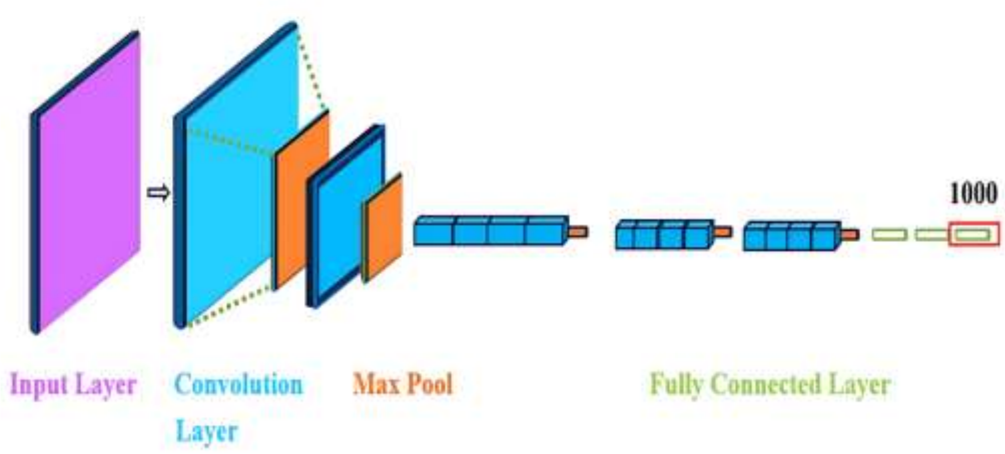


Fig 3. VGG19 model Architecture

The proposed model employs a sequential architecture and a pre-trained VGG19 network to extract features. We employed the VGG19 model, which has 20,024,384 non-trainable parameters, to generate feature maps with dimensions of (7, 7, 512). The flattened layer transforms the features into a vector of size 25,088. For the feature transformation, a dense layer with 256 neurons was used. This was followed by a dropout layer to prevent overfitting. We implemented a dense output layer with four neurons for classification. The model comprised 26,448,196 parameters, with 6,423,812 designated as trainable. This configuration facilitates effective fine-tuning while utilizing the pre-trained weights of VGG19. The text describes a machine learning model for image classification. The procedures and steps of the VGG19 model are given below:

1. The model uses a pre-existing network called VGG19 as its foundation.

2. VGG19 acts as a feature extractor, identifying important aspects of the input images.

3. The extracted features are then flattened into a long list of numbers.

4. This list goes through a layer that further processes the features.

5. A dropout layer is added to prevent the model from becoming too specialized to the training data.

10.48047/jocaaa.2024.33.07.27

6. The final layer classifies the image into one of four categories.

7. The model has over 26 million parameters in total, but only about 6.4 million can be adjusted during training.

8. This setup allows the model to benefit from VGG19's pre-learned knowledge while still being able to adapt to the specific task at hand.
Here, is a list of parameters that comprise this model. The list of parameters extracted and used is given in the Table 1 and the basic summary of VGG19 layered design is given in the Table 2.

Table 1 the list of parameters

The total parameters	26448196
The total trainable parameters	6423812
The total non-trainable parameters	20024384

Table 2 Provides a basic summary of VGG19 layered design.

Layer (type)	Output Shape	Param #
vgg19 (Functional)	(None, 7, 7, 512)	20024384
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 256)	6422784
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 4)	1028

3. Experimental Results and Discussion

The evaluation of the proposed model using the KHDD dataset yielded comprehensive insights into its performance. Fig. 4 provides a visual class distribution of the dataset, offering a clear understanding of the data composition used in the study. The evaluation results of the VGG19 model presented are given in Table 3, and a detailed breakdown of its performance metrics across various parameters is provided in Table 4, which presents a classification report detailing the algorithm's performance, likely including metrics such as precision, recall, and F1-score for each class, given in Table 5.

Visual representations of the model's efficacy are provided through Figures 5, 6, and 7. These pictures show different aspects of the model's performance. They include a confusion matrix for the VGG19 model, graphs that show the accuracy of training and validation over time, and the loss curves that go with them. Additionally, the figures also showcase predictions made on the test set, providing concrete examples of the model's real-world application..

10.48047/jocaaa.2024.33.07.27

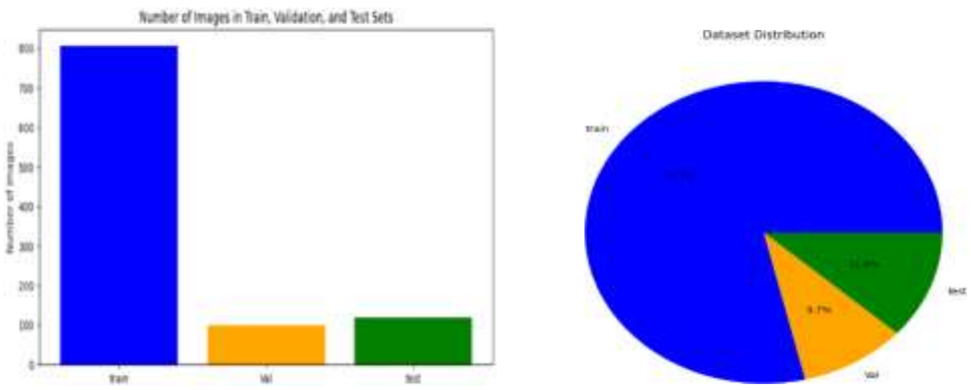


Fig 4. Class Distribution in Data Set

Table 3 Evaluation Results of the VGG19 Model

Epoch	Train Accuracy	Train Loss	Validation Accuracy	Validation Loss	Time per Epoch
5	0.9809	0.0455	0.9899	0.0395	38s
10	0.9922	0.0275	0.9899	0.0290	35s
15	0.9924	0.0275	0.9899	0.0139	41s
20	0.9833	0.0315	0.9899	0.0157	41s
25	0.9959	0.0121	0.9899	0.0160	41s

Table 4 The total and average values for the relevant columns across all epochs:

Metric	Total	Average
Train Accuracy	24.9059	0.9962
Train Loss	1.0603	0.0424
Validation Accuracy	24.8023	0.9921
Validation Loss	0.7118	0.0285
Tamper Epoch(sec)	939s	37.56s

These values represent the sum and the mean of overall 25 epochs.

The training outcomes presented in Table 2 provide valuable insights into the model's performance and learning progression over 25 epochs. The consistent improvements in accuracy and reduction in loss in the training data indicate that the model is effectively learning from the provided examples. The initial training accuracy of 98.37% is already quite high, suggesting that the model had a strong starting point. The final training accuracy of 99.03% represents a notable improvement, demonstrating the model's ability to refine its predictions further.

The validation results offer additional perspective on the model's generalization capabilities. At epoch 20, getting 100% validation accuracy is especially impressive because it means the model can perfectly classify data it hasn't seen yet. However, the slight increase in validation loss toward the end of training (reaching 0.0740 by epoch 25) warrants attention. Such

10.48047/jocaaa.2024.33.07.27

an increase could potentially indicate the onset of overfitting, where the model begins to memorize training data rather than learning generalizable patterns. The decreasing duration of each training step, from 1 second to approximately 935 milliseconds, indicates improved computational efficiency as training progresses, which is a positive aspect of the training process

.

Table 5. Algorithms classification report performance

Models	Classes	Precision	Recall	F1-score	Support
VGG19	Class 1	1.00	1.00	1.00	24
	Class 2	1.00	1.00	1.00	25
	Class 3	1.00	1.00	1.00	25
	Class 4	1.00	1.00	1.00	45

The VGG19 model's performance, as depicted in Table 5, showcases exceptional classification capabilities across all four categories. With precision, recall, and an F1-score consistently achieving the maximum value of 1.00, the model demonstrates perfect accuracy in identifying and categorizing instances within each class. This flawless performance indicates that the VGG19 model successfully avoided any false positives or false negatives, correctly classifying every sample in the dataset.

The support values provide insight into the distribution of instances across the four classes, with Class 1 containing 24 samples, Class 2 having 45 samples, Class 3 comprising 25 samples, and Class 4 consisting of 25 samples. Despite the varying number of instances in each category, the model maintained its perfect performance, suggesting robust generalization capabilities. This extraordinary consistency across different class sizes shows that the VGG19 model is very good at and reliable at the classification task at hand, making it a great choice for this dataset and problem domain.

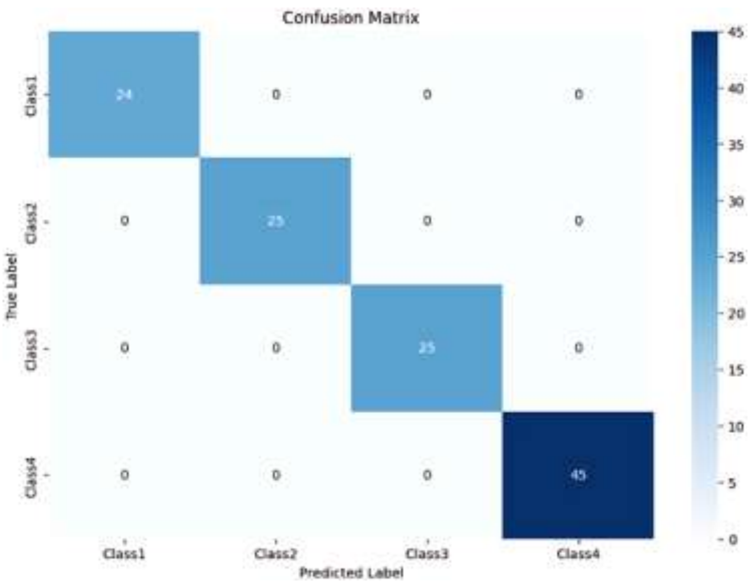


Fig 5. Visualizing VGG19 Model Efficacy with a Confusion Matrix.

The VGG19 model's performance, as depicted in Fig. 5, demonstrates exceptional accuracy in classifying the given dataset. The confusion matrix reveals a perfect classification scenario,

10.48047/jocaaa.2024.33.07.27

where each sample was correctly assigned to its respective class without any errors. This remarkable outcome underscores the model's robustness and its ability to discern distinctive features among the four classes effectively.

The flawless classification across all classes suggests that the VGG19 model has successfully captured the intricate patterns and characteristics unique to each category. This level of precision is particularly noteworthy given the varying sample sizes among the classes, ranging from 4 to 45 instances. The model has learned to recognize the most important features and has also done a good job of adapting to the subtleties that come up in smaller sample sets, as shown by its consistent performance across different class sizes. This result speaks to the model's generalization capabilities and its potential applicability in real-world scenarios where class imbalances are common.

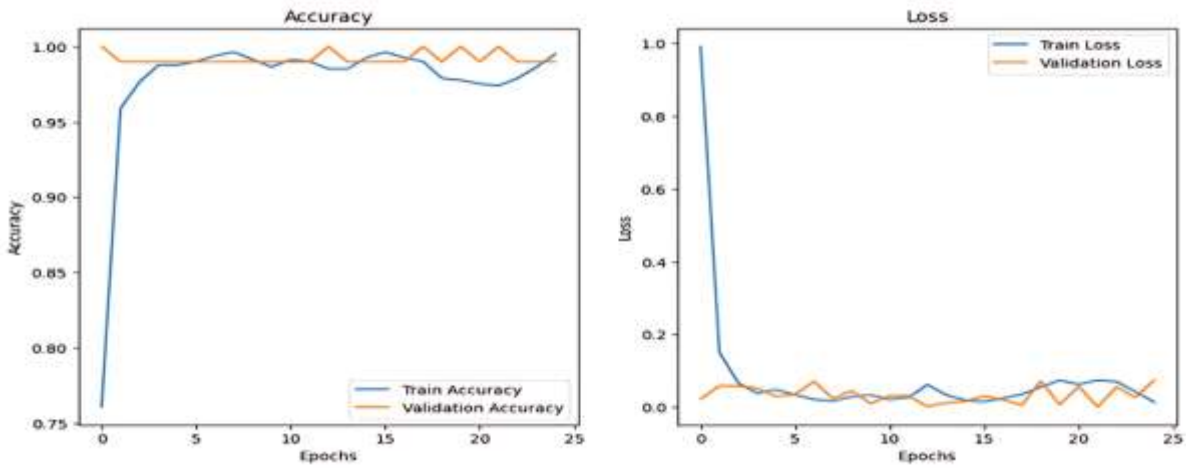


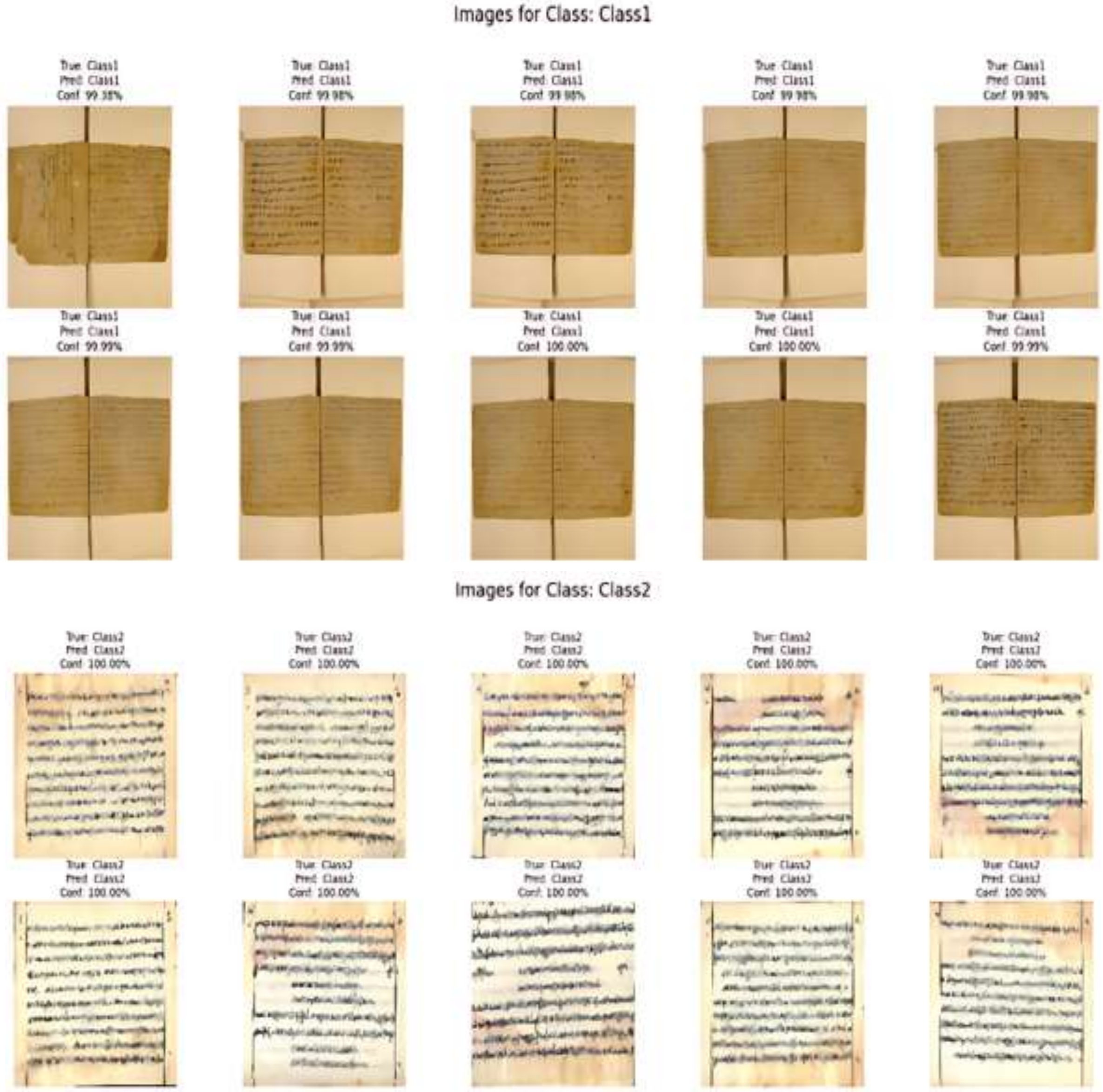
Fig 6**.** Train and Validation accuracy and Train Loss and Validation Loss

The training and validation metrics illustrated in Fig. 6 provide valuable insights into the model's learning process and generalization capabilities over 25 epochs. The training accuracy exhibits a gradual but consistent improvement, initiating from a lower value and progressively increasing until it plateaus at approximately 99%. This trend shows that the model does a good job of taking in information from the training data and finding the patterns and features that are needed to make accurate predictions. Conversely, the validation accuracy demonstrates remarkably high performance from the early stages of training, ultimately achieving a perfect score of 100% near epoch 20. This high level of validation accuracy shows that the model works well with data it hasn't seen before, which means it doesn't over fit.

The loss curves offer additional information regarding the model's optimization process. In the first few epochs, both training and validation losses drop sharply. This shows that the model is learning quickly and making better predictions. The training loss continues to decrease gradually, eventually stabilizing at a low value, which aligns with the high training accuracy observed. Notably, the validation loss remains slightly higher than the training loss throughout the training process, but it also stabilizes, further supporting the model's ability to generalize effectively. The combination of high accuracy and low loss values for both training and validation sets suggests that the model has achieved an optimal balance between fitting the training data and maintaining its predictive power on unseen samples.

10.48047/jocaaa.2024.33.07.27

The VGG19 model does a great job of putting samples into all four groups, as shown in Fig. 7. For Classes 1, 2, and 3, the model achieves perfect classification accuracy, with confidence levels ranging from 99.98% to 100%. This achievement indicates that the model has successfully learned the distinguishing features of these classes and can identify them with near-absolute certainty. The fact that these three classes all have high confidence levels suggests that the model has done a good job of capturing the patterns and traits that are unique to each class. Class 4 presents a slightly more nuanced picture, although it still showcases impressive performances. While the majority of predictions for this class maintain the same high confidence levels observed in the other classes, a few instances exhibit marginally lower confidence scores, with the lowest being 91.72%. This slight variation in confidence levels for Class 4 could be attributed to several factors, such as increased complexity or subtle similarities with other classes. Nevertheless, even these lower confidence scores remain notably high, demonstrating the model's robust ability to discern and classify samples across all four classes with remarkable accuracy and reliability.



10.48047/jocaaa.2024.33.07.27

Fig 7. VGG19 model prediction on test set

Conclusions:

The VGG19 model-based system for classifying historical Kannada handwritten palm leaf manuscripts has shown remarkable promise in the field of cultural heritage preservation. By achieving high accuracy rates across multiple epochs, this deep learning approach demonstrates its robustness and reliability in handling complex historical texts. Using pre-trained models and custom preprocessing methods together effectively solves the specific issues that come up when trying to read old manuscripts. This approach preserves valuable cultural artifacts and paves the way for more comprehensive studies in linguistics, paleography, and historical research.

We used a dataset of 1025 historical Kannada palm leaf manuscript images and we preprocessed them for training, testing, and validation. The data set contains 807 images for training, 99 for validation, and 119 for testing. VGG19 model achieved remarkable accuracy and low loss metrics, highlighting its efficacy. The system achieved 99.62% accuracy in training and 98.03% in validation, correctly identifying and sorting ancient manuscripts. These find-

10.48047/jocaaa.2024.33.07.27

ings are significant for preserving cultural heritage, showing that computers can now automatically sort and identify historical Kannada manuscripts accurately. This research contributes to digital humanities by offering a robust automated classification methodology.

The implications of this study extend beyond the immediate classification of Kannada manuscripts. By establishing a solid methodological foundation, this research opens up new avenues for exploring other ancient languages and scripts. Future research directions could include expanding the dataset to encompass a wider range of historical documents, refining the model architecture to better capture the nuances of different writing styles and times, and developing transfer learning techniques to adapt the system for use with other ancient scripts. Adding this technology to other digital humanities tools could also make it easier to study and protect cultural heritage as a whole. This could completely change how we think about historical texts and the situations they were written in.

Acknowledgement

The authors are grateful to Rani Channamma University, Belagavi for sanctioning the minor research project and providing financial assistance for this research work vide sanction order No RCUB/N0/24-25/6828/11 Dated 22.02.2025.

References

1. Bannigidad Parashuram and Chandrashekar Gudada. "Age-type identification and recognition of historical kannada handwritten document images using HOG feature descriptors." In Computing, Communication and Signal Processing: Proceedings of ICCASP 2018, pp. 1001-1010. Springer Singapore, 2019.
2. Khayyat, Manal M. and Elrefaei. "Towards author recognition of ancient Arabic manuscripts using deep learning: A transfer learning approach." International Journal of Computing and Digital Systems, 90,No 5 (2020).
3. Krizhevsky, Alex, Sutskever, I., and Hinton, Geoffrey E. "Imagenet classification with deep convolutional neural networks." Advances in Neural Information Processing Systems, 25 (2012).pp
4. Khashman, Adnan, Sekeroglu, Boran, and Dimililer Kamil. "Intelligent coin identification system." In 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, pp. 1226-1230. IEEE, 2006.
5. Demilew, Fitehalew Ashagrie and Boran Sekeroglu. "Ancient Geez script recognition using deep learning." SN Applied Sciences, 1, No. 11 (2019): 1315.
6. Danukusumo, Kefin Pudi, and Martinus Maslim. "Indonesia ancient temple classification using convolutional neural network." In 2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC), pp. 50-54. IEEE, 2017.
7. Bannigidad, Parashuram, and S. P. Sajjan. "Automatic Writer Identification of Historical Kannada Handwritten Palm Leaf Manuscripts using AlexNet Deep Learning Approach."vol issue
8. Rehman, Arshia, Saeeda Naz, Muhammad Imran Razzak. "Writer identification using machine learning approaches: a comprehensive review." Multimedia Tools and Applications 78 (2019): 10889-10931.
9. Saddami, Khairun, Khairul Munadi, and Fitri Arnia. "Degradation classification on ancient document image based on deep neural networks." In 2020, 3rd International Confer-

10.48047/jocaaa.2024.33.07.27

ence on Information and Communications Technology (ICOIACT), pp. 405-410. IEEE, 2020.
10. Barucci, Andrea, Costanza Cucci, Massimiliano Franci, Marco Loschiavo, and Fabrizio Argenti. "A deep learning approach to ancient egyptian hieroglyphs classification." Ieee Access 9 (2021): 123438-123447.
11. Bannigidad, Parashuram, and Chandrashekar Gudada. "Identification and classification of historical Kannada handwritten document images using LBP features." International Journal of Intelligent Systems Design and Computing 2, No. 2 (2018): pp 176-188.
12. Wahdan, Ahlam, Sendeyah Hantoobi, Said A. Salloum, and Khaled Shaalan. "A systematic review of text classification research based on deep learning models in Arabic language." Int. J. Electr. Comput. Eng 10, No. 6 (2020):pp 6629-6643.
13. Dargan, Shaveta, and Kumar, "Writer identification system based on offline handwritten text in Gurumukhi script." In 2020, the Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), pp. 544-549. IEEE, 2020.
14. Alsaleh, Deem, and Souad Larabi-Marie-Sainte. "Arabic text classification using convolutional neural network and genetic algorithms." IEEE Access 9 (2021): 91670-91685.
15. Bannigidad, Parashuram, and Chandrashekar Gudada. "Identification and Classification of Historical Kannada Handwritten Scripts based on their Age-Type using Line Segmentation with GLCM." (2019).
16. Sajjan, S P; Bannigidad, Parashuram; Domlur, Ashok (2024), "Ancient Kannada Handwritten Palm Leaf Dataset", Mendeley Data, V1, doi: 10.17632/w5px7czbn9.1.