A New Data Augmented Image Classification Approach in Skin Cancer Detection

Souvik Pramanik
Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24060

***Abstract*—Traditional deep learning frameworks use Convolutional Neural Networks (CNN/ConvNet) to classify images. It can be used extensively in the medical field to determine diseases. The goal of this work is to propose and implement a new deep learning approach to improve the accuracy of the classification of images for disease detection by leveraging useful features of Deep Learning models. Image Preprocessing techniques are applied to enhance specific features and eliminate unwanted information in images as part of data augmentation. Moreover, a new data augmentation image classification approach make the disease detection more accurate. This results in improvement of accuracy of image classification for skin cancer detection. In this approach, the implemented model will be trained and tested using the PAD-UFES-20 and Skin Cancer International Skin Imaging Collaboration (ISIC) datasets.**

***Index Terms*—Machine Learning (ML), Convolutional Neural Networks (CNN/ConvNet), Long-Short-Term Memory (LSTM), Confusion Matrix, Skin Cancer Images, Accuracy, Precision, Recall, F1 Score, International Skin Imaging Collaboration (ISIC), PAD-UFES-20, Data Augmentation**

I. Introduction

Machine Learning (ML) has been adopted across a wide range of industries from finance to marketing. Recently, it has been having an impact in the healthcare sector. Several studies have demonstrated the benefits of detecting diseases at early stages. For example, one study showed the power of image recognition technology based on ML by identifying white blood cells with a 90% accuracy rate [4]. One use of ML related to healthcare is the detection of skin cancer based on classification of images. This technique can be useful for identifying skin abnormalities and cancer like Melanoma. This involves identifying patterns in images of skin lesions that correlate with skin diseases [39].

Different ML methods are adopted for detection of diseases. Deep Learning architectures are used for obtaining results on image classification tasks. These architectures perform better with classification compared to other machine learning methods like Support Vector Machines (SVM) [38]. CNN and its variants are the most common type of deep learning networks that have been found to perform well in image classification and facial recognition [31]. The model in [31] achieved a high testing accuracy for certain conditions but resulted in lower metrics for others. However, performance is sensitive on dataset composition, with variations in accuracy and error rates observed in experiments with different datasets. The ongoing objective is to improve on the accuracy of detection of diseases based on image classification under any conditions as well as similarly structured datasets.

In [5], the authors present how to classify images using LSTM but the significant challenge with this work is network pruning or the elimination of parameters in a network. Furthermore, the authors acknowledges that combination of multiple methods is more likely to obtain higher accuracy scores such as a CNN and a RNN method. [10] utilizes a layered LSTM-CNN structure that does better than a traditional CNN when tested for MNIST and Breast Cancer Datasets. The authors in [10] have implemented the hybrid combination LSTM-CNN for image classification. It is proven that LSTM-CNN performs better than CNN architectures. Because of the characteristics of LSTM, it will remember the long-term dependencies and shape of the input image in a particular pattern. But in [10], there has not been much exploration with respect to reversing the order of CNN and LSTM layer placements. The lack of experimentation of different position of the CNN and LSTM layers with respect to classification of Skin Cancer Images in [10] was the motivation of the CNN-LSTM hybrid approach in this paper.

Other works involve using PAD-UFES-20 and ISIC datasets. Azeem et. al. in [36] utilized SkinLesNet which is a multi-layer deep convolutional network along with pre-treined architectures like VGG16 and ResNet-50 using the PAD-UFES-20 and ISIC datasets. While SkinLesNet performs better compared to the other datasets including a 96 % accuracy for the PAD-UFES-20 dataset. However, it took 100 epochs to reach that accuracy needing significant computational resources. The goal in this paper is to reach high accuracy for image classification with fewer epochs. Similarly, Yin et. al. in [37] prove that DenseNet-169 network model performs better for the ISIC and PAD-UFES-20 datasets. However, the highest accuracy obtained is 81.4 %. In addition, DenseNet-169 is an established neural network and only contains convolutional neural networks. The objective is to reach better accuracy in image classification for disease detection.

An architecture is constructed that will combine CNN and LSTM layers using different order structures for better classification purposes. It is observed that LSTMs can complement the feature extraction ability of CNN when used in a layered order. LSTMs have the capacity to selectively remember patterns for a long duration of time and CNNs can extract the important features out of it. LSTM is meant

to handle sequential data like time series analysis. But its capabilities extend beyond time series data and are applied to data of sequential nature including image data by memorizing and capturing important features in images [33]. This feature makes LSTM good for medical image diagnosis and analysis in which accurate identification of key attributes is important. Furthermore, the authors in [33] were able to prove the capability of memorizing and capturing important features in image data using MedvLSTM for medical image classification. However, using LSTM alone does not always perform well given issues with dimensionality which is why combinations with other architectures as proposed by Bappy et al. [34] are preferred. The above-mentioned attributes of CNN and LSTM along with the dimensionality issues with singular LSTM is the motivation for combining their important features resulting in the hybrid model.This combined approach has been implemented by experimenting with whether the CNN layer acts as an input to the LSTM layer and vice-versa. Moreover, although there is considerable work for image classification in skin disease detection, advance pre-processing techniques like data augmentation have not been applied in the above mentioned research works.

II. Specific Aims

The specific goal of this work is to expand beyond the classification of images using existing architectures for better accuracy. It is important to develop a real-time Skin Cancer detection system that can provide better recommendations about treatments. Pre-processing is important to address the issues of noise in images; inconsistencies in image sizes, formats, quality; and imbalances in categories in data. Computer Vision image pre-processing techniques are applied before feeding into machine learning models to convert data into a reasonable format for training that reduces computational load and help improve model performance. The implemented model is a hybrid approach to the integration of CNN and LSTM architecture in the field of image classification to improve the detection accuracy of skin cancer diseases when data are preprocessed with augmentation techniques. This approach results in higher performance accuracy (96 %) using less computational resources.

III. Background

More information about each deep learning network is discussed in the following. These networks are used to create the architectures that will be used to train and test the Skin Cancer ISIC and PAD-UFES-20 Datasets.

A. Convolutional Neural Networks (CNN)

CNN or ConvNet is a class of neural networks, defined as multilayered neural networks designed for computer vision applications to detect complex features in data [7]. CNN is known for extracting important features from an image by applying filters that scan across an image that help identify patterns and spatial relationships [8]. These patterns are important in classifying objects provided in images. These patterns are derived from concepts like linear algebra specifically matrix multiplication [9]. An illustration of how Convolutional Neural Networks progress through each layer is shown in Fig. 1 below. An input pixel travels through different feature maps which are sub-sampled as part of the feature extraction process [9]. The sub-sampled data passes through the fully connected layers for classification. In addition to CNN architecture, there are variants that have been used for image classification like ResNet architecture which is renowned for the residual connections to extract high-level features from images of skin diseases using transfer learning and has been used for Skin Cancer detection as provided in [24]. The expressions for CNN are presented in [30] where x_t represents the input image pixels instead of the input time series.

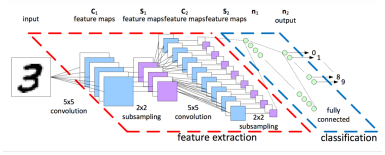

Fig. 1. Diagram of ConvNet presenting how input image navigates through the network.

B. Long Short-Term Memory (LSTM)

LSTM is a type of Recurrent Neural Network (RNN). It is typically used in sequential data analysis, such as analyzing and processing time series datasets because it can remember short- and long-term dependencies. Fig. 2 shows a diagram of an LSTM architecture. Input data passes through a combination of Input, Forget, and Output gates along with a hidden state and memory cell state. These gates selectively determine the data to be retained or discarded for each cell. Data passes through a chain of sequences [18]. Activation functions are used to provide a new memory and hidden state. RNN can be used for image classification; however, few of these implementations have been made [5]. Traditional RNN can be jittery, especially when backpropagating gradients, which can cause gradient explosion and/or vanishing [5]. LSTM mitigates the problems caused by RNN by having each ordinary recurrent node replaced by memory cells. These memory cells can store relevant information so that the model does not have to utilize long-range dependencies in the data. The complete process explaining the input, forget, and output gates and their respective equations can be found in [29] and [30].

IV. Datasets

Two datasets are used for the analysis: International Skin Imaging Collaboration (ISIC) and PAD-UFES-20. Both were originally used for research with the National Institutes of Health (NIH). The dataset taken from the Skin Cancer ISIC organization is known for the collaborations between computer vision and dermatology by using digital skin imaging to detect and mitigate skin cancers [19]. It consists of 2357 images of malignant and benign skin diseases. They are sorted

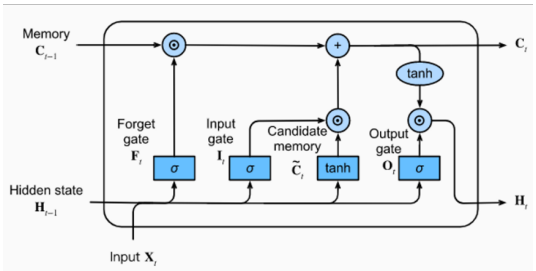

Fig. 2. Diagram of an LSTM Architecture (Source: Taken from [18].)

according to the classifications provided by the ISIC. The PAD-UFES-20 is based on the Dermatological and Surgical Assistance Program, which is a non-profit that provides free skin lesion treatment to low-income Brazilians [25]. This dataset is composed of images of a set of patient clinical data containing up to 21 features for three skin diseases and three skin cancers. It is used to support future research and development of new tools to assist clinicians in detecting skin cancer. This dataset contains 2298 samples of six different types of skin lesions, three cancers and three skin diseases [25] The subsets will be divided into the same number of images using pre-processing methods. Visualizations of the dataset will be shown as provided in Figs. 3 and 4. Both the ISIC and PAD-UFES-20 datasets contain the following diseases: Actinic keratosis, Basal cell carcinoma, Dermatofibroma, Melanoma, Nevus, Seborrheic keratosis, Squamous cell carcinoma, and [2]. The ISIC dataset contains diseases representing Pigmented benign keratosis, and Vascular lesion while PAD-UFES-20 dataset contain images representing Bowen's disease.

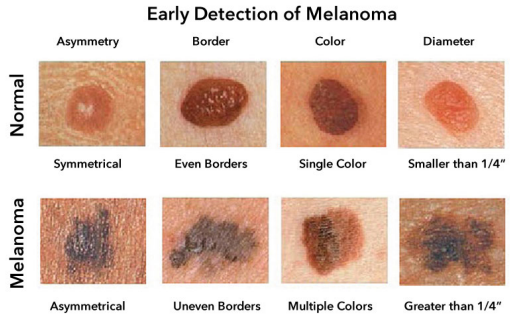

Fig. 3. Images distinguishing between normal skin and melanoma using the ISIC dataset.

V. Research Design

The cancer detection system uses pre-processing, implementation of CNN, LSTM-CNN, and CNN-LSTM models, evaluation using training and testing, and visualization of results. In addition, experiments are also performed to determine the distribution of classes in the training dataset or class imbalances. These research designs and experiments are shown below. An algorithm of the steps for the system is shown in Appendix A.

Fig. 4. Sample Images from PAD-UFES-20 dataset (Source:Taken from [26].

A. Pre-Processing

Details of the images in the datasets are visualized as shown in Fig. 5. Images are resized to a suitable dimension. Normalization is applied to the pixels for each image using Min-Max Scalar Normalization as shown in Equation (1) where x represents the current image pixel value, x_{min} represents the lowest pixel value, x_{max} represents the highest pixel value, x_{new} represents the normalized pixel values [22]. In addition, data augmentation techniques such as rotating images horizontally and vertically are applied to improve performance.

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

B. Data Augmentation

Data Augmentation is a statistical technique that allows maximum likelihood estimation from incomplete data. Datasets are inflated according to their size and quality so that better deep learning models can be trained with them ([27] and Fig. 5). This is useful in Bayesian Analysis and to reduce overfitting on machine learning models. Data Augmentation improves model generalization and performance. Real-life datasets can have a class imbalance in which one class might have a disproportionate number of samples that could affect the final model [28]. For all architectures presented, a distribution check is performed using the data augmentation technique **Augmentor**. **Augmentor** adds more to all classes so that none of the classes has a few samples [37]. In Python, **Augmentor** is instantiated using the pipeline directory containing the initial image data. Data are normalized and augmented using techniques such as zooming, flipping, and rotating. The purpose of augmentation is to increase the size and diversity of the dataset by creating improved versions of existing images. This involved the use of the Python library **Augmentor**. It is used to aid in the augmentation and generation of image data in preprocessing tasks. Other computer vision techniques with data augmentation, such as horizontal and vertical rotation, are also applied. The number of operations are performed in the dataset using the Pipeline object by calling the `sample()` method. 500 samples were added per class for an even distribution.

The augmented images are stored in the output subdirectory of each of the subdirectories of skin cancer types. This balanced data is trained with different architectures. Data Augmentation is helpful in addressing overfitting issues with the dataset distribution.

In addition to applying **Augmentor**, images were randomly flipped, rotated, and zoomed in. This is a common technique used to allow the model to learn the model from different perspectives and angles and ensure the model is robust to variations in real-world data. Performing these operations also prevents overfitting by exposing the image to different image variations and conditions like camera angles or slight changes in position [34].

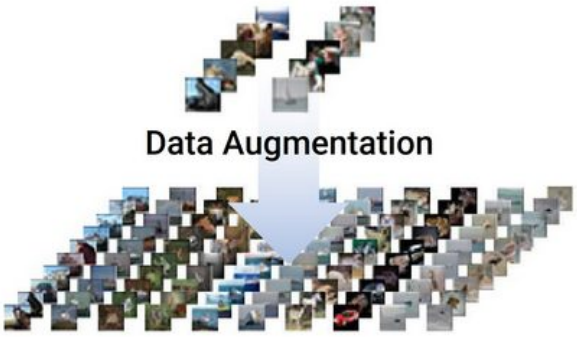

Fig. 5. Data Augmentation visual (Source: Taken from [22]).

C. Training, Testing, and Validation

Data is split into training, testing, and validation sets. The training set is used to train the model. The testing set is used to test the trained data. The system is measured using a validation set to check for model overfitting using accuracy and loss plots (Fig. 6). The test set is used to check the models using evaluation techniques such as accuracy, precision, recall, and F1 scores and to check if they predict the unseen data correctly.

The data is split in a 4:1 Train Test Ratio (80 % Training and 20 % Testing). In addition, the skin cancer images are resized by their original sizes to 120x120 (image height and image width respectively). There is a batch size provided or the number of images to be trained at once. For all architectures, the batch size is set to 32 which is standard in training deep learning models.

D. CNN-LSTM Architecture

Fig. 7 presents the CNN-LSTM architecture. This model involves using CNN layers for feature extraction in input data combined with LSTMs to perform sequence prediction on feature vectors. The images are pre-processed using normalization and filtering techniques. The output shape from the normalization and filtering is the input shape for the CNN or Convolutional Layer. A Convolutional kernel is created that produces a tensor by convolving the input of the layer over a single spatial dimension and extracting important features

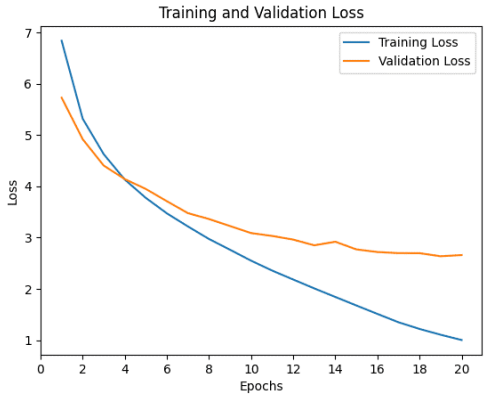

Fig. 6. Training and Validation Loss Plot vs. Epochs.

[20]. The tensor produced by the convolutional layer will be fed into a max pooling layer to reduce spatial dimension of feature maps [21]. The output of the max-pooling layer is then transferred into the LSTM Architectures to achieve the output shape. For the implementation of the CNN-LSTM model, the training and testing of data is performed in Google Colab Notebook.

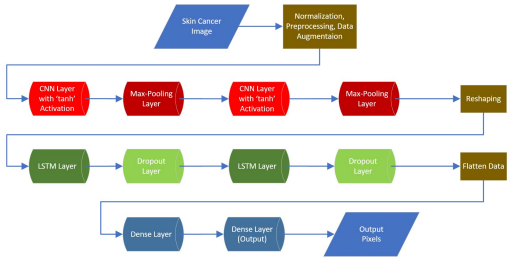

Fig. 7. CNN-LSTM Network Architecture.

For this CNN-LSTM architecture, input values are rescaled from values in the [0, 255] range to the [0, 1] range. The rescaled input values are passed into the CNN layer with 64 filters and a pooling size of with a hyperbolic tangent (tanh) activation function. The output of this CNN layer is passed into a MaxPooling 2D Layer. The output of the Max-Pooling 2D layer is passed through two additional CNN layers with the same filters, pooling sizes, and activation functions. The output of the third CNN layer is reshaped on the basis of the number of features in the image. This is passed into the LSTM layer with 32 units with a return sequence of True. A return sequence of True means that the output of the hidden state is relied upon for a time-step [39]. The output of the LSTM layer is inputted to the Dropout layer with a value of 0.2. The output of the dropout layer is flattened and passed into two dense layers, one with 128 units and a ReLu activation function and another specifying the number of classes to be identified.

E. Model Performance Comparisons

The CNN-LSTM is evaluated using the accuracy, precision, recall and F1 scores and confusion matrices. The CNN-LSTM implementation is compared with other state-of-the-art implementations such as those provided in [5], [7], [33], [10] utilizing accuracy scores. These implementations were selected as they were tested using images such as Fashion-MNIST [5], Pneumonia Detection [7], [33], MNIST, and Breast Cancer IDC [10]. All architectures utilize the data augmentation techniques described above.

F. Evaluation

Several metrics are used to evaluate the Skin Cancer Detection Models. They are confusion matrices, accuracy, precision, recall, and F1 scores. All of the implementations correct the issues with imbalanced data using Data Augmentation. These experiments with accuracy and data augmentation are helpful in determining the architecture improvements.

1) Confusion Matrix: Confusion Matrix is generally provided as a 2D matrix in classification problems to check the performance of a system by showing the number of correctly and wrongly classified data helping to identify classes of data, which might be misplaced (Fig. 8). Confusion Matrices are used to determine which images for cancer detection are classified correctly.

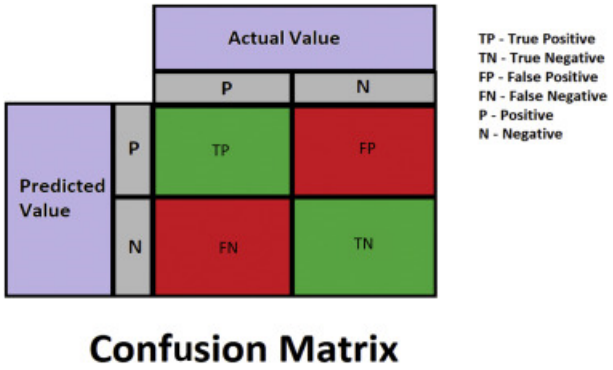

Fig. 8. Confusion Matrix showing Actual and Predicted Values.

In a confusion matrix, there are four characteristics which represent the measurement metrics of the classifier (Fig. 8 and [12]). These combinations are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP and TN mean that the classification correctly identified the positive and negative classes respectively. FP and FN mean that the classification predicted a positive or negative class respectively, but actually represents the opposite class([12]- [13]).

2) Accuracy, Precision, Recall, and F1 Scores: Accuracy, Precision, Recall, and F1 scores are used for classification of the CNN-LSTM model for the Skin Cancer images. They provide assessments of a model's predictive capabilities, each focusing on different aspects of classification performance [15].

$Accuracy$ is used to calculate percentage of the images that are correctly classified (2). This will be used for the testing images.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

However, when there are imbalanced datasets, accuracy may not always be a good metric. Precision and Recall scores are usefulfor these situations ([14] - [15]). $Precision$ determines how correct are the predictions by taking the number of true positive predictions TP and dividing by all of the relevant positive predictions (TP and FP) as provided in Equation (3).

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$Recall$ checks for instances that are actually correct among instances that might have been missed (Eq. (4)).

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$F1Score$ combines both precision and recall providing an overall view of the model's accuracy as shown in Eq. (5). This is especially useful with respect to false positives (FP) and false negatives (FN) [15].

$$F1Score = \frac{2 \times Precision \times \ Recall}{Precision + Recall} \tag{5}$$

VI. Results

The Accuracy, Precision, Recall, and F1 scores, and Confusion Matrices are taken after retraining datasets with **Augmentor** with the implemented models. These results provided an outlook on which architecture performs better with Skin Disease Classification. Training deep learning architectures can take time due to calculations across multiple layers, backpropagation, forward propagation, hardware component usage, and the amount of data needed for training. These have been factors when obtaining the following results below.

The Accuracy, Precision, Recall, and F1 Scores for the results of the model performances are shown in Appendix B. Tables I and II presents the accuracy scores for 10, 20, 50, and 100 epochs for CNN, CNN-LSTM, and LSTM-CNN models using the ISIC and PAD-UFES-20 datasets. In the Skin-Cancer ISIC dataset, CNN performs better than CNN-LSTM and LSTM-CNN for 10 epochs while CNN-LSTM performs better than the CNN and LSTM-CNN models as the number of epochs used for training increases with an accuracy of 87.28 % for 100 epochs. This shows how well the CNN-LSTM architecture performs for the ISIC dataset. For the PAD-UFES-20 dataset in Table II, the CNN model performs better for all epochs. However, once image augmentations using horizontal and vertical rotations were applied, the CNN-LSTM implementation performed with a 96.34 % accuracy for 50 epochs for the PAD-UFES-20 dataset. The same goes true for the Skin-Cancer ISIC dataset with 91.23 % at 50 epochs. However, an outlier exists in the ISIC dataset at 30

epochs with an accuracy of 28.49 %. Accuracy scores are presented for 10, 20, 30, and 50 epochs as shown in Tables III and IV for both the ISIC and PAD-UFES-20 datasets. Note that Table III represents the training and testing accuracies with only horizontal rotations applied as data augmentation while Table IV represents the accuracies after applying vertical and horizontal rotation. Overall, the results demonstrate that higher accuracy is attainable with a CNN-LSTM model using modified data augmentations.

Accuracy and loss plots for 10 Epochs are shown for CNN-LSTM in Fig. 9. The Confusion matrix for the CNN-LSTM architecture for ISIC is provided in Fig. 10. These plots were provided with a data augmentation of horizontal flips only. The Accuracy and loss plots and confusion matrices for the PAD-UFES-20 are provided in Figs. 11 and 12. These plots and matrices are for the data augmentation of horizontal rotation. The plots and matrices for the vertical and horizontal rotation are shown in Figs. 13 and 14. Additional Plots for 20, 30 and 50 epochs before (horizontal flip) and after (horizontal and vertical flip) data augmentation for the ISIC and PAD-UFES-20 datasets are provided in Appendix VIII-C. The accuracy plots for the training and validation datasets in Figs. 13, 11, and Appendix VIII-C show an increase in accuracy over time. However, it is the loss plots that are important in the CNN-LSTM models. The loss plots show that with the CNN-LSTM data, the training and validation data hold well, indicating that the CNN-LSTM data is better at tracking unseen data.

Fig. 9. Training and Validation Accuracy and Loss Plots for 10 Epochs for CNN-LSTM Architecture.

In addition to Accuracy, Precision, Recall, F1 Scores, and Confusion Matrices, the trained model and the images tested were deployed in an app where an image of skin disease is selected along with their skin disease prediction. Fig. 15 presents this screenshot of the app.

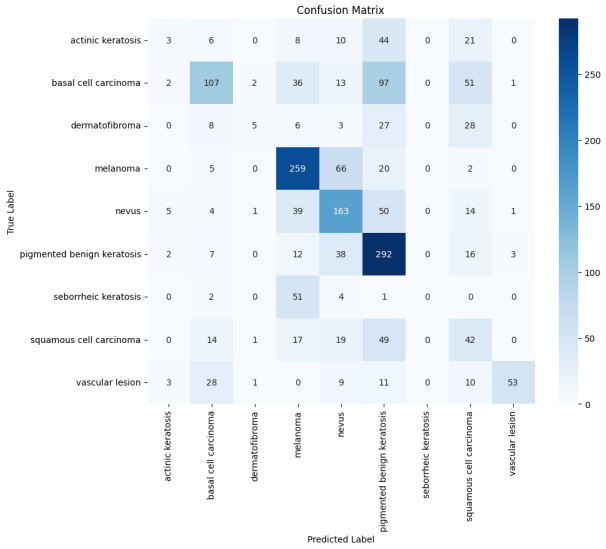

Fig. 10. Confusion Matrix for CNN-LSTM implementation trained for 10 Epochs using Training Data.

Fig. 11. Training and Validation Accuracy and Loss Plots for 10 Epochs for CNN-LSTM Architecture using PAD-UFES-20.

VII. Conclusion and Future Works

In this work, a novel approach has been proposed for detection and classification of skin-cancer disease images. The results show that CNN-LSTM outperforms in terms of accuracy compared to other architectures. The proposed model along with Augmented data outperforms the other implemented architectures for both datasets. The augmentation techniques using horizontal and vertical rotations in the pre-processing phase have improved the performance of the models by increasing the size and diversity of the data for training. The proposed approach using Augmentation techniques resulting in

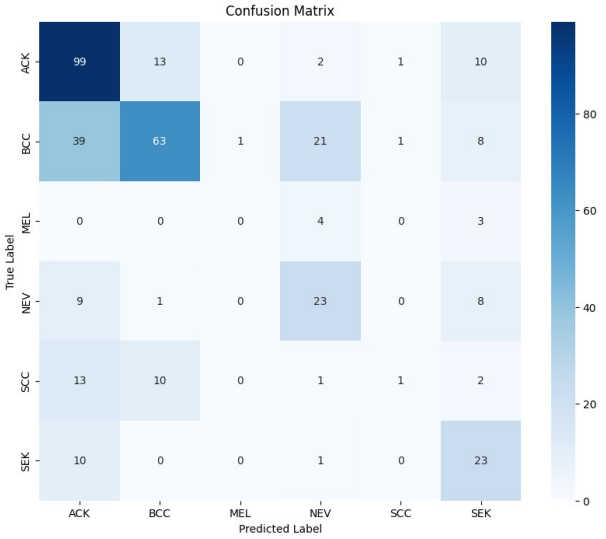

Fig. 12. Confusion Matrix for CNN-LSTM implementation trained for 10 Epochs using PAD-UFES-20 dataset.

Fig. 13. Training and Validation Accuracy and Loss Plots for 10 Epochs for CNN-LSTM Architecture using PAD-UFES-20 and modified Data Augmentation.

better performance of the models is also part of this research work.

For future work, improvements can be made by training models with larger image sizes, upgrading hardware components such as the GPU and TPU to expedite training. To expand on this work on data augmentation for skin cancer concepts like Generative Adversarial Networks (GANs), Transfer Learning & Pretrained Models with augmented datasets and adaptive augmentation techniques will be considered.

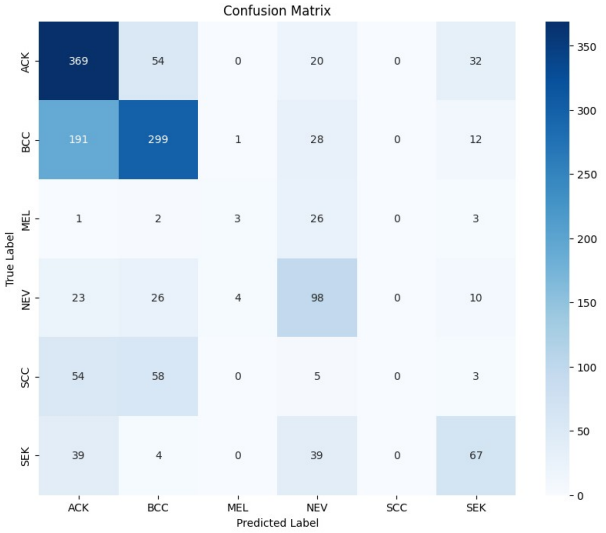

Fig. 14. Confusion Matrix for CNN-LSTM implementation trained for 10 Epochs with modified Data Augmentation using PAD-UFES-20 dataset.

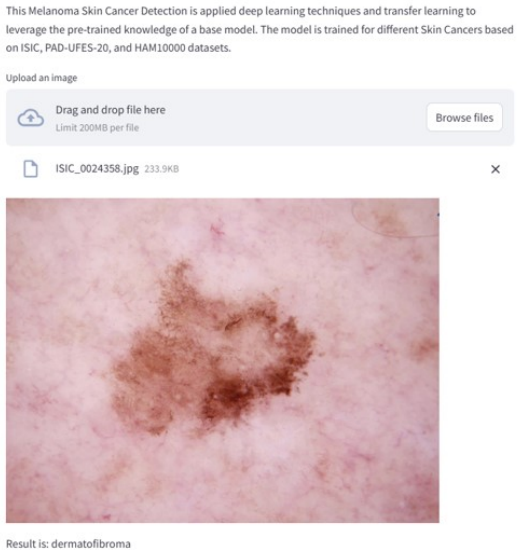

Fig. 15. Melanoma Skin Cancer Detection App Screenshot.

REFERENCES

[1] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification. John Wiley & Sons, 2012

[2] J. Ahuja, "Skin Cancer Detection using CNN," Kaggle.com, 2024. https://www.kaggle.com/datasets/jaiahuja/skin-cancer-detection

[3] "GPU vs CPU - Difference Between Processing Units - AWS," Amazon Web Services, Inc. https://aws.amazon.com/compare/the-difference-between-gpus-cpus/

[4] "How to Build a Disease Detection System Using ML," Akkio. https://www.akkio.com/post/disease-detection-using-machine-learning

[5] K. Zhang, "LSTM: An Image Classification Model Based on Fashion-MNIST Dataset." Available: https://users.cecs.anu.edu.au/Tom.Gedeon/conf-/ABCs2018/paper/ABCs2018_paper_92.pdf

[6] S. Pangre, "Hardware for Machine Learning and Neural Network," Medium, Feb. 19, 2020. https://medium.com/sayalipangre123/hardware-for-machine-learning-and-neural-network-864544e54e4f

[7] "Pneumonia Detection using CNN with Implementation in Python," Analytics Vidhya, 18-Sep-2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/09/pneumonia-detection-using-cnn-with-implementation-in-python/.

[8] H. Shi, A. Wei, X. Xu, Y. Zhu, H. Hu, and S. Tang, "A CNN-LSTM based deep learning model with high accuracy and robustness for carbon price forecasting: A case of Shenzhen's carbon market in China," Journal of Environmental Management, vol. 352, p. 120131, Feb. 2024, doi: https://doi.org/10.1016/j.jenvman.2024.120131.

[9] Piotr Skalski, "Gentle Dive into Math Behind Convolutional Neural Networks," Medium, Apr. 12, 2019. https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9

[10] Aditi, & Nagda, Mayank & Eswaran, Poovammal. (2019). Image Classification using a Hybrid LSTM-CNN Deep Neural Network. International Journal of Engineering and Advanced Technology. 8. 1342-1348. 10.35940/ijeat.F8602.088619.

[11] "Slides26," Cornell.edu, 2020. https://www.cs.cornell.edu/courses/cs4787/2021sp/notebooks/Slides26.html (accessed Oct. 11, 2024).

[12] A. Kulkarni, "Confusion Matrix - an overview — ScienceDirect Topics," Sciencedirect.com, 2019. https://www.sciencedirect.com/topics/engineering/confusion-matrix

[13] Tonatiuh Hernández-Del-Toro, Fernando Martínez-Santiago, Arturo Montejo-Ráez, Chapter 7 - Assessing classifier's performance, Editor(s): Alejandro A. Torres-García, Carlos A. Reyes-García, Luis Villaseñor-Pineda, Omar Mendoza-Montoya, Biosignal Processing and Classification Using Computational Learning and Intelligence, Academic Press, 2022, Pages 131-149, ISBN 9780128201251, https://doi.org/10.1016/B978-0-12-820125-1.00018-X.

[14] Is accuracy a good measure of model performance? — Fiddler AI," www.fiddler.ai. https://www.fiddler.ai/model-accuracy-vs-model-performance/is-accuracy-a-good-measure-of-model-performance

[15] S. Walker, "F-Score: What are Accuracy, Precision, Recall, and F1 Score? — Klu," klu.ai, Jul. 04, 2023. https://klu.ai/glossary/accuracy-precision-recall-f1

[16] M. Pharr, W. Jakob, and G. Humphreys, "Sampling and Reconstruction," Elsevier eBooks, pp. 401–504, Oct. 2016, doi: https://doi.org/10.1016/b978-0-12-800645-0.50007-5.

[17] Matt Pharr, Wenzel Jakob, Greg Humphreys, 07 - Sampling and Reconstruction, Editor(s): Matt Pharr, Wenzel Jakob, Greg Humphreys, Physically Based Rendering (Third Edition), Morgan Kaufmann, 2017, Pages 401-504, ISBN 9780128006450, https://doi.org/10.1016/B978-0-12-800645-0.50007-5.

[18] S. N. Qasem and S. M. Alzanin, "An Effective Forecasting Approach of Temperature Enabling Climate Change Analysis in Saudi Arabia," International Journal of Advanced Computer Science and Applications, vol. 15, no. 3, 2024, doi: https://doi.org/10.14569/ijacsa.2024.0150372.

[19] "ISIC Archive," www.isic-archive.com. https://www.isic-archive.com/

[20] Ying Chen, Yiqi Huang, Zizhao Zhang, Zhen Wang, Bo Liu, Conghui Liu, Cong Huang, Shuangyu Dong, Xuejiao Pu, Fanghao Wan, Xi Qiao, Wanqiang Qian, Plant image recognition with deep learning: A review, Computers and Electronics in Agriculture, Volume 212, 2023, 108072, ISSN 0168-1699, https://doi.org/10.1016/j.compag.2023.108072.

[21] L. Zhao and Z. Zhang, "A improved pooling method for convolutional neural networks," Scientific Reports, vol. 14, no. 1, p. 1589, Jan. 2024, doi: https://doi.org/10.1038/s41598-024-51258-6.

[22] T. A. Team, "How, When, and Why Should You Normalize / Standardize / Rescale Your Data? – Towards AI — The Best of Tech, Science, and Engineering," Towards AI, May 16, 2019. https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff

[23] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," Scientific Data, vol. 5, no. 1, Aug. 2018, doi: https://doi.org/10.1038/sdata.2018.161.

[24] S. Aladhadh, M. Alsanea, M. Aloraini, T. Khan, S. Habib, and M. Islam, "An Effective Skin Cancer Classification Mechanism via Medical Vision Transformer," Sensors, vol. 22, no. 11, p. 4008, May 2022, doi: https://doi.org/10.3390/s22114008.

[25] A. G. C. Pacheco et al., "PAD-UFES-20: A skin lesion dataset composed of patient data and clinical images collected from smartphones," Data in Brief, vol. 32, Aug. 2020, doi: https://doi.org/10.1016/j.dib.2020.106221.

[26] W. Yin, J. Huang, J. Chen, and Y. Ji, "A study on skin tumor classification based on dense convolutional networks with fused metadata," Frontiers in Oncology, vol. 12, Dec. 2022, doi: https://doi.org/10.3389/fonc.2022.989894.

[27] T. A. Team, "How, When, and Why Should You Normalize/Standardize/Rescale Your Data? – Towards AI — The Best of Tech, Science, and Engineering," Towards AI, May 16, 2019. https://towardsai.net/p/data-science/how-when-and-whyshould-you normalize-standardize-rescale-your-data-3f083def38ff

[28] G. O. Assunc¸ao, R. Izbicki, and M. O. Prates, "Is Augmentation˜ Effective in Improving Prediction in Imbalanced Datasets?," Journal of Data Science, pp. 1–16, Jan. 2024, doi: https://doi.org/10.6339/24jds1154.

[29] B. Marcus, Augmentor Documentation Release 0.2.12. 2023. Accessed: Dec. 13, 2024. [Online]. Available: https://augmentor.readthedocs.io/˙/downloads/en/stable/pdf/

[30] A.A. Awan, "A Complete Guide to Data Augmentation," www.datacamp.com, Nov. 2022. https://www.datacamp.com/tutorial/complete-guide-data-augmentation

[31] N. Gaffoor and S. Soomro, "Skin Disease Detection and Classification Using ResNet-50 and Support Vector Machine: An Effective Approach for Dermatological Diagnosis," 2023 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS), Bali, Indonesia, 2023, pp. 140-145, doi: 10.1109/IoTaIS60147.2023.10346059. keywords: Support vector machines;Sensitivity;Scalability;Eczema;Melanoma;Skin;Medical diagnosis;Skin Diseases;CNN;Classify;Feature Extraction;SVM algorithm;Accuracy,

[32] P. Szepesi and L. Szilagyi, "Detection of pneumonia using con- ´ volutional neural networks and deep learning," Biocybernetics and Biomedical Engineering, vol. 42, no. 3, Aug. 2022, doi: https://doi.org/10.1016/j.bbe.2022.08.001.

[33] I. Salehin et al., "Real-Time Medical Image Classification with ML Framework and Dedicated CNN–LSTM Architecture," Journal of Sensors, vol. 2023, p. e3717035, Dec. 2023, doi: https://doi.org/10.1155/2023/3717035.

[34] Jawadul H. Bappy, Cody Simons, Lakshmanan Nataraj, B. S. Manjunath, and Amit K. Roy-Chowdhury. Hybrid lstm and encoder–decoder architecture for detection of image forgeries. IEEE Transactions on Image Processing, 28(7):3286–3300, 2019.

[35] S. P. Sone, J. J. Lehtomaki and Z. Khan, "Wireless Traffic Usage¨ Forecasting Using Real Enterprise Network Data: Analysis and Methods," in IEEE Open Journal of the Communications Society, vol. 1, pp. 777-797, 2020, doi: 10.1109/OJCOMS.2020.3000059. keywords: Forecasting;Time series analysis;Wireless networks;Data analysis;Machine learning;Resource management;5G;CNN;CNNGRU;CNN-LSTM;forecasting;GRU;holt winters;LSTM;neural network;real network data;SARIMA;spatio temporal;temporal;time series analysis;WLAN,

[36] M. Azeem, K. Kiani, T. Mansouri, and N. Topping, "SkinLesNet: Classification of Skin Lesions and Detection of Melanoma Cancer Using a Novel Multi-Layer Deep Convolutional Neural Network," Cancers, vol. 16, no. 1, p. 108, Dec. 2023, doi: https://doi.org/10.3390/cancers16010108.

[37] W. Yin, J. Huang, J. Chen, and Y. Ji, "A study on skin tumor classification based on dense convolutional networks with fused metadata," Frontiers in Oncology, vol. 12, Dec. 2022, doi: https://doi.org/10.3389/fonc.2022.989894.

[38] E. Shevchenko, "Why deep learning may be preferred over support vector machines (SVMs)," Medium, Jan. 05, 2023. https://medium.com/@eugenesh4work/why-deep-learning-may-be-preferred-over-support-vector-machines-svms-9d2d3021aa29 (accessed Dec. 14, 2024).

[39] K. Conger, "AI improves accuracy of skin cancer diagnoses in Stanford Medicine-led study," News Center, Apr. 10, 2024. https://med.stanford.edu/news/all-news/2024/04/ai-skin-diagnosis.html

VIII. Appendix

A. Algorithm of CNN-LSTM implementation

Algorithm 1 Training Architecture using Skin Cancer ISIC Dataset

1: Load Data
2: Split training data to training and validation data (80% Training and 20% Testing)
3: Set up `batch_size` and resize images (`img_height` and `img_width`)
4: Normalization: Using `MinMax`
5: Visualize images
6: Model Implementation (CNN, CNN-LSTM, LSTM-CNN)
7: Train Model (`batch_size` and `epochs`)
8: Evaluation: `acc`, `val_acc`, `loss`, `val_loss`
9: Data Augmentation: Image rotation and zoom
10: Model Implementation with Data Augmentation
11: Train Model with Augmented Data (`batch_size` and `epochs`)
12: Evaluation using Re-Trained Data: `acc`, `val_acc`, `loss`, `val_loss`
13: Check Class Imbalance and Re-balance distribution of images in Training Data (**Augmentor**)
14: Model Implemtation using **Augmentor**
15: Retrain using model implemented with **Augmentor** (`batch_size` and `epochs`)
16: Evaluation using Re-Trained Data: `acc`, `val_acc`, `loss`, `val_loss`
17: Precision, Recall, F1 Scores and Confusion Matrix

B. Accuracy, Precision, Recall, and F1 Scores.

TABLE I
Accuracy Scores for 10, 20, 50, and 100 Epochs for Skin-Cancer ISIC dataset.

Architecture	10 Epochs (%)	20 Epochs (%)	50 Epochs (%)	100 Epochs (%)
CNN-LSTM	48.31	**59.26**	**74.94**	**87.28**
LSTM-CNN [10]	46.32	54.35	60.44	65.74
CNN [2]	**50.95**	52.96	66.07	78.35

TABLE II
Accuracy Scores for 10, 20, 50, and 100 Epochs for PAD-UFES-20 dataset.

Architecture	10 Epochs (%)	20 Epochs (%)	50 Epochs (%)	100 Epochs (%)
CNN-LSTM	56.42	62.61	66.21	71.99
LSTM-CNN [10]	42.09	52.68	62.07	67.57
CNN [2]	**58.60**	**64.92**	**68.66**	**73.56**

TABLE III
Accuracy Scores for the CNN-LSTM architecture 10, 20, 30, and 50 Epochs before Data Augmentation Techniques.

Dataset	10 Epochs (%)	20 Epochs (%)	30 Epochs (%)	50 Epochs (%)
Skin Cancer ISIC	48.31	59.26	64.84	**74.94**
PAD-UFES-20	**56.42**	**62.61**	**66.21**	71.99

TABLE IV
Accuracy Scores for the CNN-LSTM architecture 10, 20, 30, and 50 Epochs after Data Augmentation Techniques.

Dataset	10 Epochs (%)	20 Epochs (%)	30 Epochs (%)	50 Epochs (%)
PAD-UFES-20	**57.04**	**66.96**	**83.48**	**96.34**
Skin Cancer ISIC	52.62	59.71	28.49	**91.23**

C. Training, Validation Accuracy Loss Plots and Confusion Matrices.

Fig. 16. Training and Validation Accuracy and Loss Plots for 20 Epochs for CNN-LSTM Architecture.

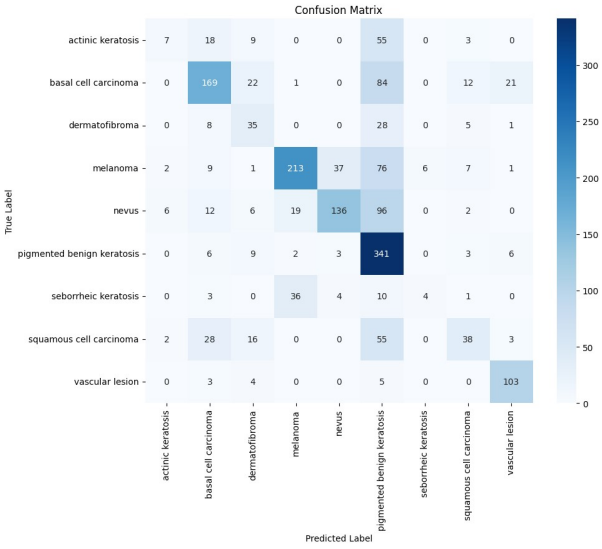

Fig. 17. Confusion Matrix for CNN-LSTM implementation trained for 20 Epochs using Training Data.

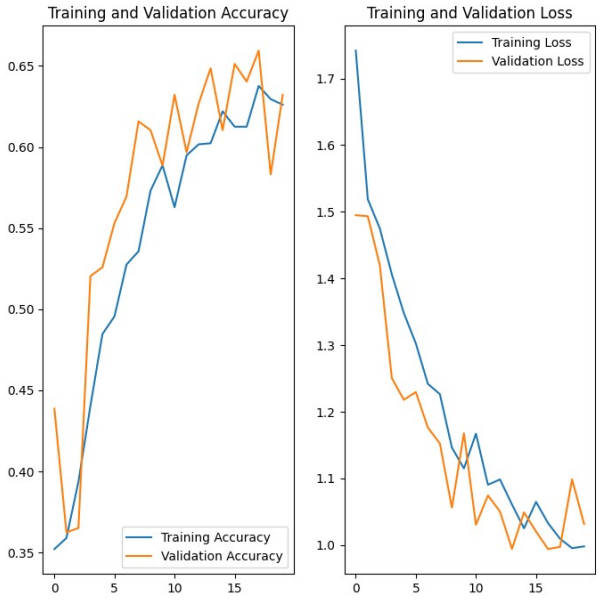

Fig. 18. Training and Validation Accuracy and Loss Plots for 20 Epochs for CNN-LSTM Architecture using PAD-UFES-20.

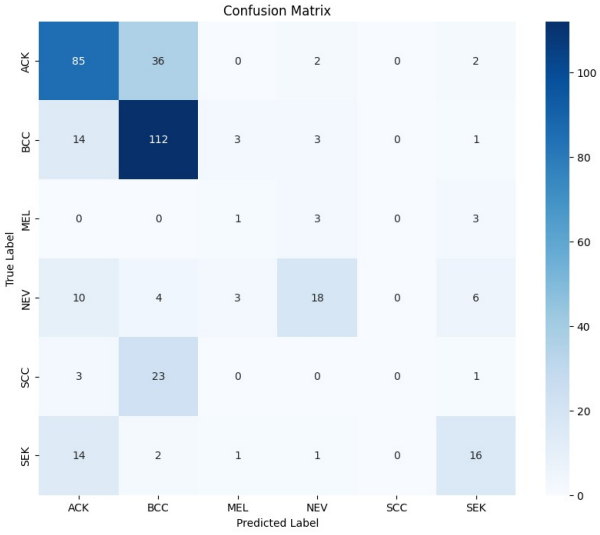

Fig. 19. Confusion Matrix for CNN-LSTM implementation trained for 20 Epochs using PAD-UFES-20 dataset.

Fig. 20. Training and Validation Accuracy and Loss Plots for 20 Epochs for CNN-LSTM Architecture using PAD-UFES-20 and modified Data Augmentation.

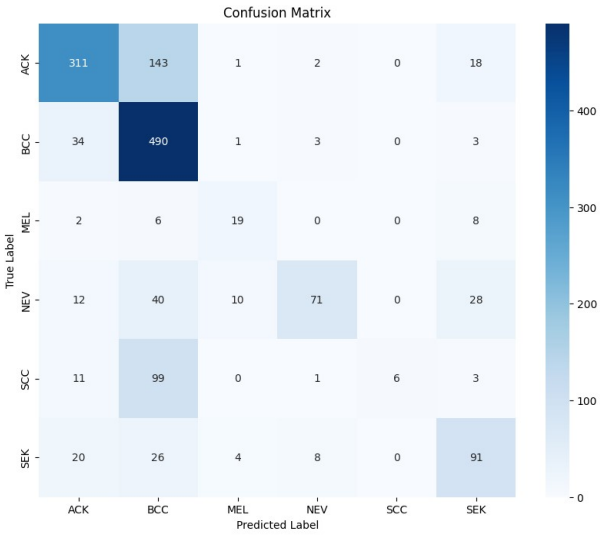

Fig. 21. Confusion Matrix for CNN-LSTM implementation trained for 20 Epochs with modified Data Augmentation using PAD-UFES-20 dataset.

Fig. 22. Training and Validation Accuracy and Loss Plots for 30 Epochs for CNN-LSTM Architecture.

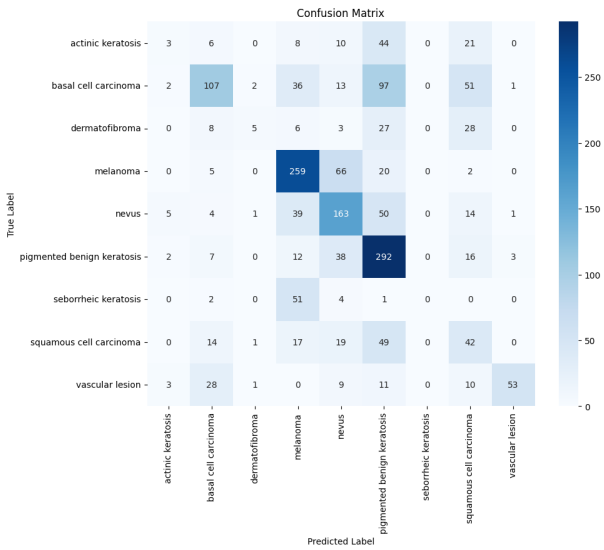

Fig. 23. Confusion Matrix for CNN-LSTM implementation trained for 30 Epochs using Training Data.

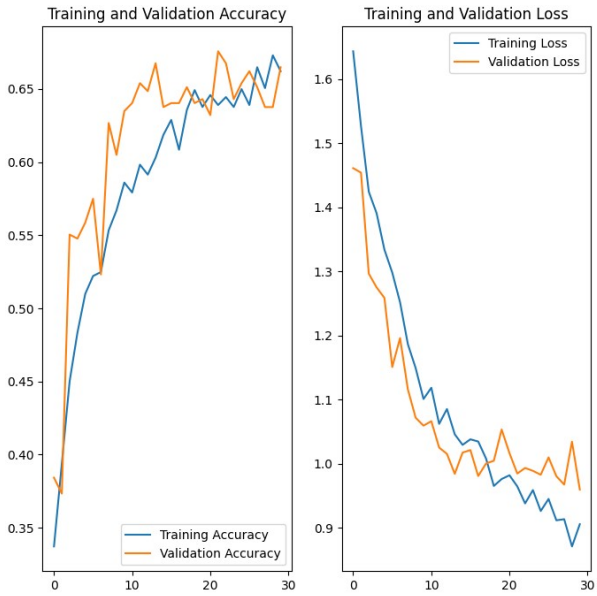

Fig. 24. Training and Validation Accuracy and Loss Plots for 30 Epochs for CNN-LSTM Architecture using PAD-UFES-20.

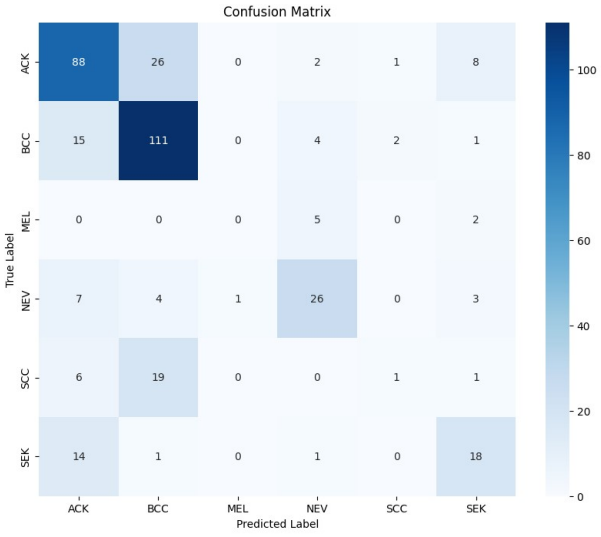

Fig. 25. Confusion Matrix for CNN-LSTM implementation trained for 30 Epochs using PAD-UFES-20 dataset.

Fig. 26. Training and Validation Accuracy and Loss Plots for 30 Epochs for CNN-LSTM Architecture using PAD-UFES-20 and modified Data Augmentation.

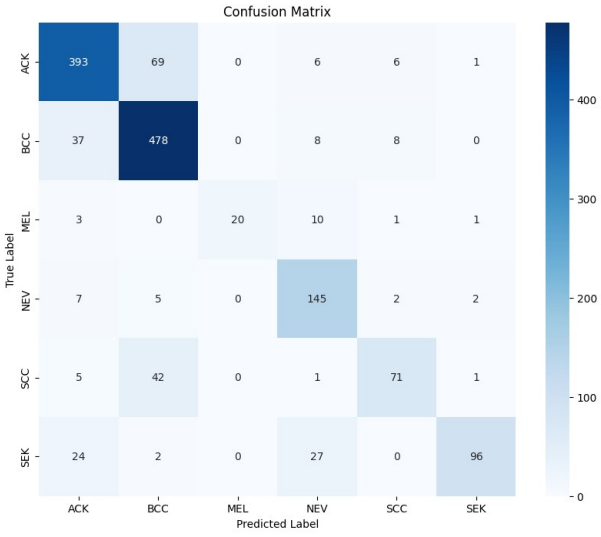

Fig. 27. Confusion Matrix for CNN-LSTM implementation trained for 30 Epochs with modified Data Augmentation using PAD-UFES-20 dataset.

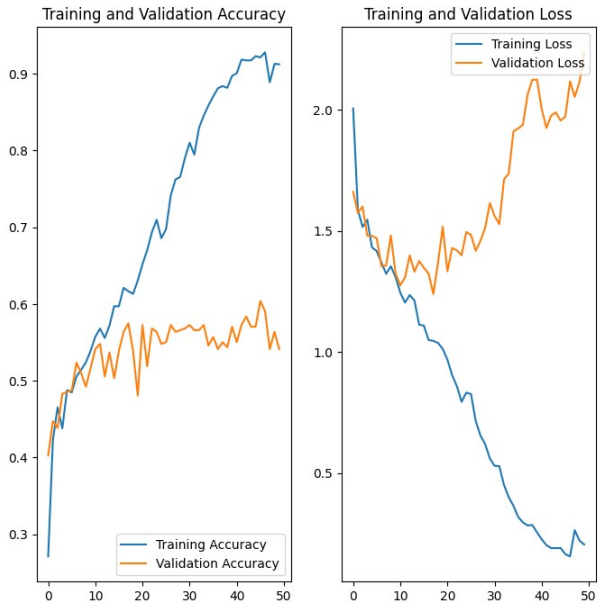

Fig. 28. Training and Validation Accuracy and Loss Plots for 50 Epochs for CNN-LSTM Architecture.

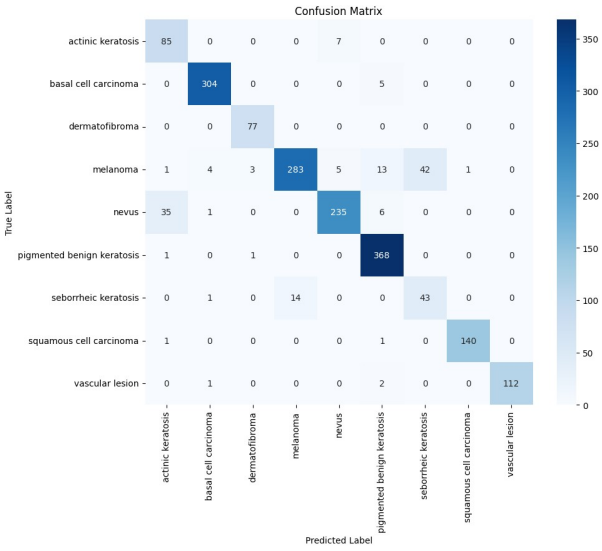

Fig. 29. Confusion Matrix for CNN-LSTM implementation trained for 50 Epochs using Training Data.

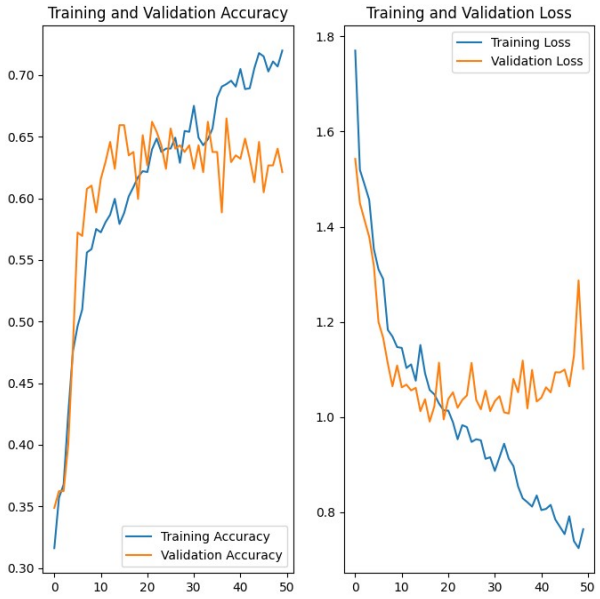

Fig. 30. Training and Validation Accuracy and Loss Plots for 50 Epochs for CNN-LSTM Architecture using PAD-UFES-20.

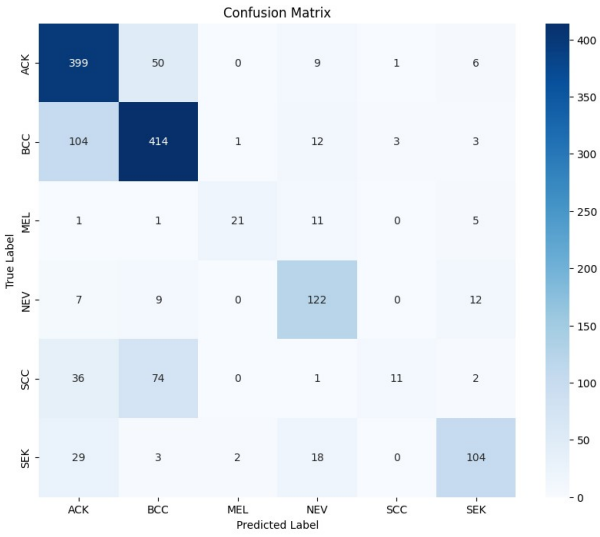

Fig. 31. Confusion Matrix for CNN-LSTM implementation trained for 50 Epochs using PAD-UFES-20 dataset.

Fig. 32. Training and Validation Accuracy and Loss Plots for 50 Epochs for CNN-LSTM Architecture using PAD-UFES-20 and modified Data Augmentation.

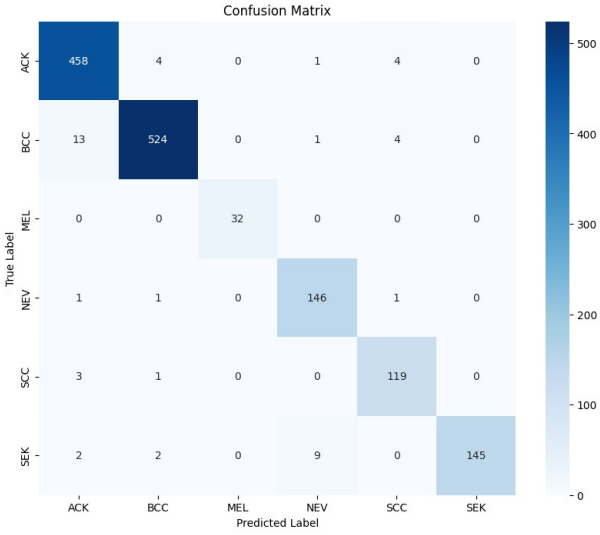

Fig. 33. Confusion Matrix for CNN-LSTM implementation trained for 50 Epochs with modified Data Augmentation using PAD-UFES-20 dataset.