Deep Learning Methods in Melanoma Cancer Detection

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

***Abstract*—Traditional deep learning framework utilize Convolutional Neural Networks (CNN/ConvNet) to detect and classify images. It has been used extensively in the medical fields for classifying images to determine diseases. The goal of this work is to propose and implement a new deep learning method to improve the accuracy of classification of diseases. The proposal is to utilize a hybrid approach by leveraging useful features of Deep Learning Models. In addition, different hardware components will be examined to determine additional improvements in performance.**

***Index Terms*—Machine Learning (ML), Convolutional Neural Networks (CNN), Long-Short-Term Memory (LSTM), Central Processing Unit (CPU), Graphics Processing Unit (GPU), Recurrent Neural Network (RNN), Confusion Matrix, Skin Cancer Images, Accuracy, Precision, Recall, F1 Score**

I. Background

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. Numerous 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 detecting skin cancer based on classification of images. This technique can be useful for identifying skin abnormalities and cancer like Melanoma. There are several architectures of neural networks. Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) Networks will be examined for this approach. For this project, the CNN and LSTM architectures will be combined to introduce a hybrid framework. In addition to implementing a CNN-LSTM architecture for classification of images, different hardware components will be experimented with such as Graphics Processing Units (GPU), Central Processing Unit (CPU), and Tensor Processing Unit (TPU). More Information about each deep network and the hardware components used is discussed below.

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 which helps in identifying 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. The sub-sampled data passes through the fully connected layers for classification.

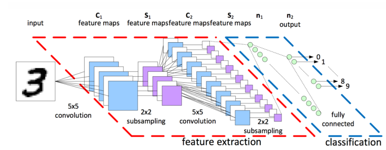

Fig. 1. Diagram of ConvNet presenting how input image navigates through the network.

B. Long Short-Term Memory (LSTM)

LSTM is a type of Recurrent Neural Network (RNN). It is typically used in sequential data analysis like analyzing and processing time series datasets because it can remember short-term 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 done [5]. Traditional RNN can be sometimes unstable in practice especially when back-propagating gradients which may cause gradient explosion and vanishing. LSTM mitigates the issues 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.

C. Hardware Components

Recent developments in Deep Learning show that neural networks are getting larger and more complex. This is due to growing amounts of data and more powerful computer

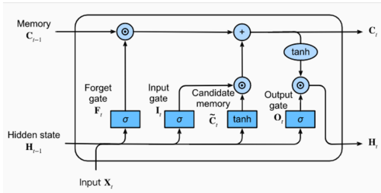

Fig. 2. Diagram of an LSTM Architecture (Source: Taken from [18].)

processing, data storage, and large-value predictions that can guide better decisions in real time without needing human intervention. ML plays a role in developing models that can predict in real-time. However, with big data comes memory issues especially when running at a large scale or clusters. The issue present is that the workload for training the data is higher due to factors such as dense matrix multiplication, convolution operation, and recurrent operation [6]. This hardware experimentation will be used to compare performance by examining the speed of the building blocks like matrix multiplication, convolution, and data/memory paths [6]. Fig. 3 provides a plot of the different hardware used for training compared with throughput and energy consumption.

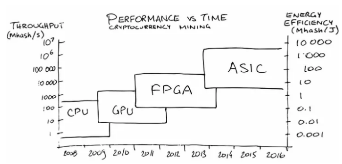

Fig. 3. Performance vs. Time graph for different hardware comparing throughput and energy efficiency (Source: Taken from [6]).

For this implementation, the hardware to be used to train and test the dataset with our CNN-LSTM hybrid implementation are CPU, GPU, and TPU. CPU is flexible in terms of programmability and handling workloads. GPU is widely used for machine learning due to its high memory bandwidth to perform large computations faster and is the best option for training. TPU is known for its fast inference through high throughput 8-bit arithmetic [11]. It performs dense-matrix multiplication and uses a 32-bit accumulator to sum the result [11].

II. Specific Aims

The specific goal of this project is to expand beyond classifying images using existing architectures. Considering the dataset that will be used for the classification of cancer cells, it is important to develop a real-time cancer detection system that can give better recommendations about treatments. In [5], the authors present how to classify images using LSTM and [7] uses CNN model for Pneumonia Detection using X-Ray images. [10] utilizes a layered LSTM-CNN structure that does better than a traditional CNN when tested for MNIST and Breast Cancer Datasets. The goal of this work is to construct a new architecture that will combine CNN and LSTM layers to classify images for cancer detection with better performance than existing architectures.

III. Dataset

The Dataset used is taken from the International Skin Imaging Collaboration (ISIC). The organization is known for collaborations between computer vision and dermatology by using digital skin imaging to detect and mitigate skin cancers [19]. The Set consists of 2357 images of malignant and benign skin diseases. They are sorted according to the classifications provided by the ISIC. The subsets will be divided into the same number of images using preprocessing methods. Visualizations of the dataset will be shown as provided in Fig. 4. The dataset will contain the following diseases: Actinic keratosis, Basal cell carcinoma, Dermatofibroma, Melanoma, Nevus, Pigmented benign keratosis, Seborrheic keratosis, Squamous cell carcinoma, and Vascular lesion [2].

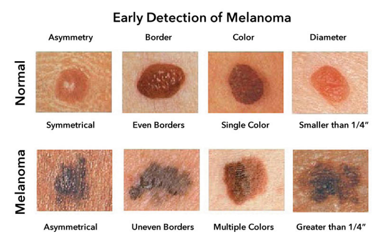

Fig. 4. Images distinguishing between normal skin and melanoma.

IV. Research Design, Methods, and Experiments

The cancer detection system will utilize the steps of preprocessing, implementing the CNN-LSTM model, and training, testing and visualizing results. There will also be experiments with finding distribution of classes in the training dataset or class imbalances, confusion matrices, accuracy, precision and recall scores. These experiments and methods are highlighted below.

A. Preprocessing

Images in the dataset will be visualized as shown in Fig. 4. In addition, the images will be resized to a reasonable dimension which will be used for training. Furthermore, there will be normalization applied to the pixels for each image optimal comparisons across data acquisition methods and texture instances as provided 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.

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

In addition to normalization of the pixels, there will be image filtering involved (Fig. 5). This is used as a form of image denoising or reducing noise in an image. The approach is that for each pixel in an input image, compute one value at the corresponding pixel location in an output image within the local neighborhood using a function. The result is that the output and input images are typically the same size. This function may involve averaging all of the values within its neighborhood. Another function is to apply box filtering or computing an output image in which each original pixel has been replaced by the average of its local neighborhood which helps in achieving a smoothing effect in the output.

Fig. 5. Image Filtering function applied to a local image data.

B. Training, Testing, and Validation

Data will be split into a training, testing, and validation sets. The training set will be used for training the model. The testing sets will be used to test the implemented models. The system will also be measured using validation sets to check for model overfitting using accuracy and loss plots (Fig. 6). The Testing Set will be used to check the CNN-LSTM model using Accuracy, Precision, Recall, and F1 scores. Furthermore, the testing set will be used to check if the CNN-LSTM model predicts the unseen data correctly.

Different training, testing, and validation splits will be explored. Some of those splits might be 80 % Training, 10 % Testing, 10 % Validation to 4:1 Train Test Ratio (80 % Training and 20 % Testing) where the Training Data is further subdivided into 4:1 Training Validation Ratio making for a 64 % Training, 16 % Validation, 20 % Testing Data.

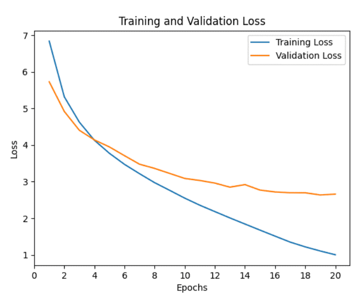

Fig. 6. Training and Validation Loss Plot vs. Epochs.

C. Evaluation

Several Metrics will be used to evaluate the Skin Cancer Detection Model. These include confusion matrices, accuracy, precision, recall, and F1 scores. In addition, the skin cancer detection model will handle imbalanced datasets. These metrics and experiments will help in determining what improvements are needed in the model.

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. 7). Confusion Matrices will be used to determine which images for cancer detection are classified correctly.

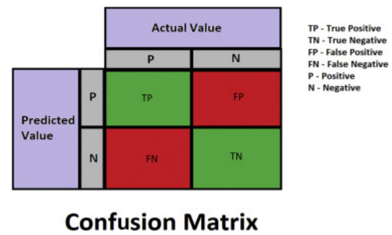

Fig. 7. Confusion Matrix showing Actual and Predicted Values.

In a confusion matrix, there are four characteristics which represent the measurement metrics of the classifier (Fig. **??** 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 will be 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].

$Accuracy$ will be 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 useful in this aspect ([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 positive and false negatives [15].

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

D. Hardware Components and State of the Art Implementations

During the experiments with the CNN-LSTM implementation, the model will be experimented with other hardware components like CPU, GPU, and TPU. Furthermore, this CNN-LSTM implementation will be compared with other state-of-the-art implementations such as those provided in [5], [7], [10] utilizing the same accuracy, precision, recall, F1 scores, and confusion matrices. These implementations were selected as they were tested using images such as Fashion-MNIST [5], Pneumonia Detection [7], MNIST, and Breast Cancer IDC [10]. These comparisons will help in determining improvements to the CNN-LSTM implementation for skin cancer detection.

V. Timelines

This is a timeline of the stages involved in creating the Skin Cancer Detection System. There are six stages as presented below.

1) Gather Dataset and Preprocessing Data including Visualizations.
2) Create CNN-LSTM Model, Train and Test the Dataset.
3) Evaluate CNN-LSTM model for Accuracy, Precision, Recall, F1, and Confusion Matrix.
4) Compare results to other state of the art implementations.
5) Repeat Steps 3 and 4 using GPU and TPU.
6) Report results, draw conclusions, and formulate plan for future work in Final Paper.

These are the stages used to progress through the project. Stages 1-3 are expected to take around a day. Around two to four weeks might be expected for Stages 4 and 5. Finally the remainder of the project will be focused on Stage 6 which will be reporting results, drawing conclusions, and formulating future plans. Throughout this work, certain stages may progress sooner or may take longer than expected. In addition, Stage 6 can begin during the implementation and evaluation of the CNN-LSTM model (Stages 1-3).

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/T̃om.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/