Identification and Classification of Skin Cancers using Deep Neural Networks

Kunal Srivastava,[1] Srivatsava Missula,[2] and Arjun Kohli[3]

Henry M. Gunn High School, kunal@masagroup.com,[1] Greenwood High School, missularaghu@gmail.com,[2] The International School Bangalore, arjun0511@hotmail.com[3]

ABSTRACT

Recorded as the most common form of cancer, skin cancer accounts for 40% of total human cancer cases around the globe, with over 5 million cases diagnosed annually. While the final diagnosis is through biopsy and histopathological examination, the first symptoms are usually visual. Hence, disfigurations on the skin (known as lesions) are used as the basis for further diagnosis of patients. As such, there are numerous non-invasive diagnostic methods that rely on photography, dermatoscopy, and spectroscopy along with the processing of visual data, where computer vision has found a niche. Our project looks into the analysis of dermatoscopic images through the implementation of Convolutional Neural Network (CNN) algorithms to consolidate a classification model that detects and distinguishes between different categories of skin cancer. Over the course of this experiment, we look at the differences in performance generated by using different CNN architectures, thus demonstrating which backbone network is best suited for classifying the given dataset. In particular, we will be implementing Resnet, VGG, Densenet, and Google's Inception v3. By further analysing the composition of the results and the networks' performance, we also seek to rationalize variation in results, and suggest improvements. Through the optimization of models and approaches pertaining to skin cancer dermatoscopy, we aim to improve the efficacy of computer-assisted cancer for early diagnosis and treatment for patients of skin cancer.

I. INTRODUCTION

An increase in skin cancer cases has been recorded as a global trend. Of these cases, which can be grouped into melanoma and non-melanoma skin cancer, it was estimated that over 100,000 new cases of melanoma skin cancer emerged in the United States in the year 2019, resulting in over 10,000 deaths [1]. In most cases of melanoma skin cancer, it has been noted that early detection and diagnosis drastically increases one's survival rate. Dermoscopy (also known as dermatoscopy), a method of examining skin lesions, has been reported to improve the sensitivity of skin cancer diagnosis [3], [4], leading to its widespread usage in the medical field for the detection and classification of skin cancer.

Dermoscopic imagery came to present a gateway for computer-aided diagnosis of skin cancer. This was chiefly because unlike manual detection and laboratory based blood tests, well-trained specialists were not required in the case of computer-aided diagnosis, and human bias would not cloud predictions. Further, it would facilitate a near automatic response to the early detection of skin cancer, potentially reducing casualties.

Certain attempts at computer-aided diagnosis were centered around image processing, such as pre-processing digital image data through denoising before analysing it via segmentation [5], [6]. However, a literature review surrounding the subject matter in 2013 [7] reveals that research pertaining to this topic was starting to shift increasingly towards neural networks, such artificial neural networks (ANNs) and k-nearest neighbor algorithms (KNNs) [6]. Specifically, the advent of convolutional neural networks (CNNs) gave rise to an improvement in the reliability and accuracy of machine learning models, to the extent that they were recorded to consistently outperform professionals [8], [9]. We propose a multi-way classification model for skin lesions based on the International Skin Imaging Collaboration (ISIC) Melanoma Project archive. A classification model of dermoscopic images would serve as an early predictor of skin cancer by distinguishing and categorising cancerous and non-cancerous lesions, such that the immediate treatment can be better suited to the patient.

We intend to test the performance of CNNs in satisfying this function by comparing different CNN architectures. Existing literature in the field of computer vision has already proven the efficacy of specific models and architectures. A 2017 study published in Nature [8] used Google's Inception V3 to produce results collated

against the predictions of the Stanford dermatology department to prove that accuracies superior to even professional practitioners could be attained through computer-aided dermoscopic diagnosis. There is also research implementing the AlexNet architecture [14] to similar effect [10]. While such studies highlight the effectiveness of models, we feel that there is much to learn from comparing the performances of different architectures.

We will be implementing ResNet [11], VGG [12], and Google's Inception v3 [13]. By comparing these three architectures and prioritizing specific metrics (such as recall, given diagnosis potentially deals with lives), we intend to analyse which best serves the purpose at hand. This can be taken a step further by investigating the optimization of hyperparameters (such as learning rate, epoch count, etc.) and the effect this has on specific performance metrics to improve the model. This is followed by a deeper analysis of the failings of the classification models, through which we hypothesize why certain architectures can better identify some classes of melanoma over others. Through these insights, we hope to inform future researchers of potential shortcomings that they should take into consideration while designing their own deep learning models, such that their endeavors can be better equipped for the early diagnosis and treatment of cancer victims.

II. METHODOLOGY

Our project seeks to compare the effectiveness of different types of neural networks in classifying disfigurations of the skin dubbed "lesions". We will be using different Convolutional Neural Network (CNN) architectures: namely VGG [12], ResNet [11], and Google's Inception v3 [13], implemented in PyTorch version 1.5.1. CNNs have demonstrated considerable potential in fields of computer vision, including in the medical and oncological streams, so we intend to draw comparisons between the performances of their different architectures under similar conditions.

A Convolutional Neural Network is a deep learning algorithm that deals with image processing. It takes various image inputs and assigns weights to them in order to distinguish between images and patterns in similar images. It relies on the back propagation of error to refine its weights and accuracy, and has seen a rise in popularity in recent years due to being more powerful than traditional classification and regression models, such as Decision Trees and Support Vector Machines, as they require less image pre-processing.

A. Melanoma dataset

The dataset we are using is a subset of the one provided by the Society for Imaging Informatics in Medicine (SIIM) and the International Skin Imaging Collaboration (ISIC) called the HAM10000 ("Human Against Machine with 10000 training images") dataset [16]. It consists of 10,015 dermoscopic images of skin lesions, accompanied by the necessary labels to implement classification. These skin lesions are divided into seven categories, so we will be implementing a 7-class classification model to group images amongst the following: melanocytic nevi, melanoma, benign keratosis-like lesions, basal cell carcinoma, actinic keratoses, vascular lesions, and dermatofibroma.

The classification labels range from benign to potentially cancerous to skin cancer proper. For example, vascular lesions ("vasc") are just birthmarks, while basal cell carcinoma ("bcc") is fully cancerous, necessitating lab tests and radiation therapy. This was done to provide the model with a range of values it can expect to observe if it was employed at a professional capacity (for computer-aided diagnosis), as dermatologists are just as likely to encounter false alarms and semi-developed cases as they are the genuine article.

When looking through the dataset, we found 4501 duplicates we had to remove, leaving us with a remaining 5514. Further, there was serious imbalance in the dataset categories, as the melanocytic nevi class had many times more images than the others, so we had to use equalization sampling to address this and mitigate this imbalance among the categories.

Before feeding the dataset to the network architecture itself, we needed to pre-process the images and to split the dataset up for training and testing. Based on the Pareto principle, we followed the 80-20 rule of splitting between training and testing sets. Depending on the network architecture, appropriate transforms were applied on the images. These included setting the dimensions of the input and normalization. Neural networks process inputs using small weights, and inputs with large integer values can disrupt or slow down the learning process,

so normalizing was necessary, such that each pixel value has a value between 0 and 1. Then, the model is trained against the training set and checked against the testing set. These processes yield the training and testing losses and accuracies, respectively.

B. Neural network architectures

It should be noted that all of the architectures we have implemented have been pre-trained from the ImageNet [15] database, and as such, they are already capable of feature learning. The training process itself doesn't vary for any of the neural network architectures, as pre-processing and splitting the datasets is largely the same across architectures (save for the input size stipulations of different architectures). Similarly, the calculation of accuracies and losses also remains the same across architectures.

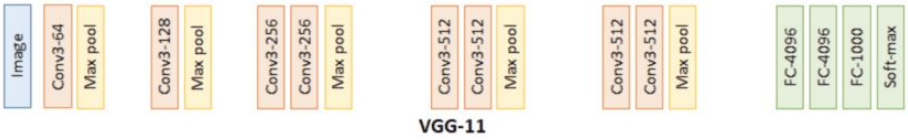

FIG. 1. Representation of the VGG-11 CNN architecture (Source: medium.com)

The first of our CNN architectures is VGG-11 (see Figure 1). Proposed in 2014 [12], it consists of eleven weighted layers, of which three are fully connected (the green "FC" layers in the figure), and it leverages the max pooling and Softmax activation functions which lower computational costs, thus optimizing the model for training with larger datasets.

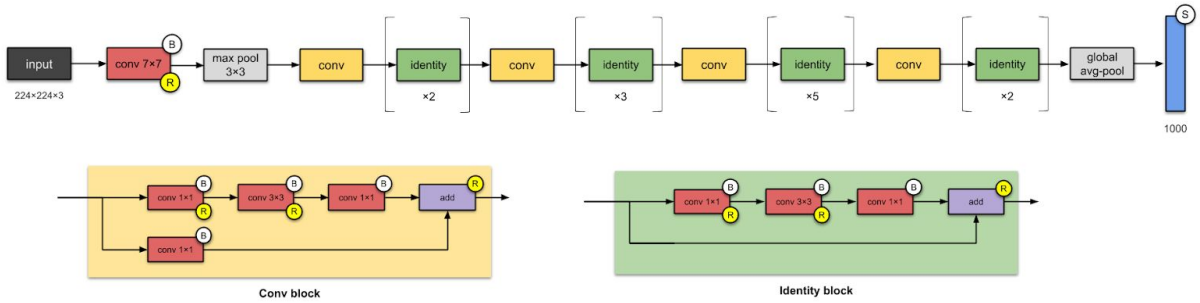

FIG. 2. Representation of the ResNet CNN architecture (Source: towardsdatascience.com)

The second is ResNet [11] (see Figure 2), an architecture that lessens the problem of "vanishing gradients": the inability to fully propagate error all the way back to the initial layers, so learning speed increases. The model we implemented, ResNet-50, consists of 50 weighted layers.

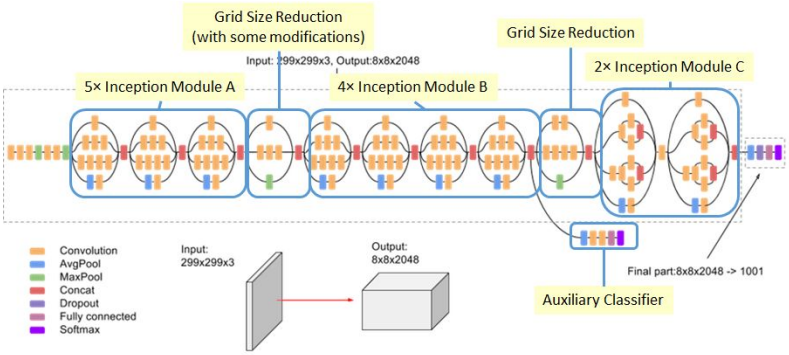

FIG. 3. Representation of Google's Inception v3 CNN architecture (Source: https://cloud.google.com/tpu/docs/inception-v3-advanced)

Finally, we implemented Google's Inception v3 [13]. As a CNN architecture, it stands out in its use of "Inception blocks", which can be identified in Figure 3 as the parts where the layers split and branch off and then rejoin (the circular portions in the figure). These inception blocks convolve the same input tensor with multiple filters and concatenate the results; the various filter sizes help the model generalise for objects of different sizes. By contrast, traditional CNNs perform only one operation on each tensor.

C. Hyperparameter tuning

	Default	Alternative
Batch Size	32	64
Learning Rate	1×10^{-3}	1×10^{-1}
Optimizer	Adam	SGD
Momentum	None	0.9

TABLE 1. Our proposed changes to each architecture

As opposed to the architecture, which deals with how the model is trained and improved, the hyperparameters represent constant aspects of the program that aren't affected by which architecture is being used. Rather than leaving them as controlled variables, we wanted to see how variations in the architecture would interact with variations with the hyperparameters to influence the performance itself. Note that the loss function (another potential hyperparameter) was kept constant at cross entropy loss.

The batch size of a neural network is the number of training examples used in one iteration. We explore switching between a large (64) and small (32) batch size to increase performance during training. By observing the effect of this change on our dataset and specific use case, we aim to discern what sort of batch size is better suited for the task.

Next, we suggest altering the learning rate. The learning rate is a parameter in the network that determines the step size when working to find the minimum of its loss function. By experimenting with the learning rate in the training of our model(s), we want to see how learning rate can optimize overall performance.

Finally, we switched the optimizer. An optimizer is an algorithm that simply adjusts attributes of the network, such as weights, to achieve a higher accuracy. Optimizers can take a momentum parameter. Momentum is a technique used to accelerate the process of gradient vectors converging. The reason for swapping between Adam and Stochastic Gradient Descent (SGD) is the nature of their momentum parameter. SGD requires a declaration of its momentum, which we provided as 0.9 to speed up the convergence of gradients, but Adam dynamically calculates and updates its momentum as the training proceeds. This is because Adam uses the moving average of the gradient instead of the gradient itself, like SGD does. This results in a gradient that constantly changes in Adam, but is stuck at one value in SGD. Therefore, by comparing these two optimizers, we are also able to see to what extent the performance of a static momentum value varies from a dynamic one.

III. RESULTS

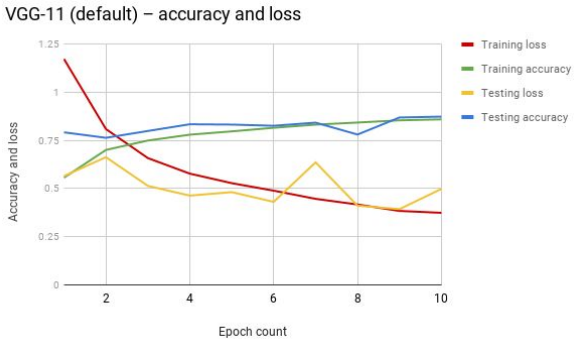

FIG. 4. Accuracy and loss values of the VGG architecture, without any hyperparameter changes

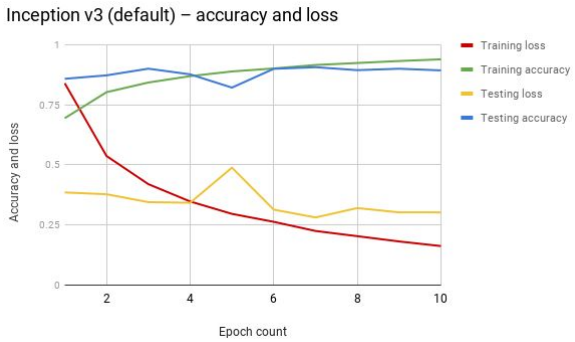

FIG. 5. Accuracy and loss values of Google's Inception v3 architecture, without any hyperparameter changes

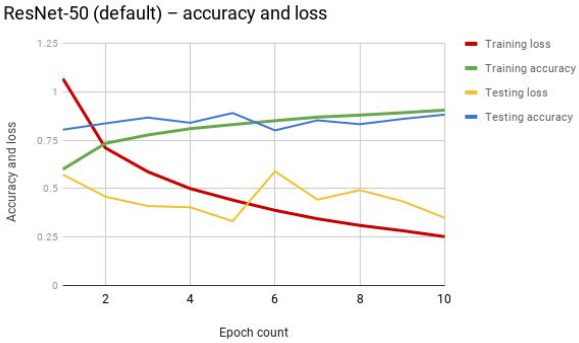

FIG. 6. Accuracy and loss values of the VGG architecture, without any hyperparameter changes

The experimental simulations were implemented in the Google Colaboratory environment on the GPU runtime. The GPU ensured a lower processing time and quicker results. These proposed simulations were implemented on the HAM10000 skin cancer dataset (a subset of the ISIC-SIIM archive) to assess the performance of various architectures and optimize them by tuning their hyperparameters.

To begin with, there are the three base architectures: ResNet-50, VGG-11 and Google's Inception V3. For these, refer to figures 4, 5, and 6. When they were trained on the dataset, all of them yielded similar trends in training losses and accuracies, as the graphs look very alike. In table 2, note that their marginally different highest testing accuracies (Inception > Resnet > VGG) are reminiscent of their comparative performances in the ImageNet challenge [15], which may in part be influenced by the models being pre-trained by the ImageNet database. The training loss decreased quickly, while the training accuracy increased at a steady, if slow, rate. Moreover, the testing loss seemed to fluctuate at some points as the number of epochs progressed but decreased overall, while the testing accuracy fluctuated slightly once, but otherwise consistently increased over the range of epochs.

Model	Change Made	Highest Testing Accuracy	Weighted Average of Precision	Weighted Average of Recall	Weighted Average of F1-Score
ResNet-50	Default parameters	89.0%	.91	.89	.90
	Learning rate: 1e-1	79.1%	.86	.79	.82
	Batch size: 64	88.5%	.91	.88	.89
	SGD (momentum=0.9)	**89.4%**	**.91**	**.90**	**.91**
VGG-11	Default parameters	87.3%	.89	.80	.83
	Learning rate: 1e-1	80.3%	.00	.04	.00
	Batch size: 64	86.8%	.89	.78	.82
	SGD (momentum=0.9)	**89.3%**	**.90**	**.86**	**.88**
Inception v3	Default parameters	90.6%	.92	.89	.90
	Learning rate: 1e-1	58.6%	.72	.60	.65
	Batch size: 64	**90.7%**	.92	.90	.91
	SGD (momentum=0.9)	90.2%	**.92**	**.90**	**.91**

TABLE 2. Performance metrics of each architecture (before and after hyperparameter tuning). Values in bold represent the highest value for each metric per model

A. Hyperparameter tuning

The changes to the models' hyperparameters started with the learning rate. The increase in the learning rate from 1e-3 (0.001) to 1e-1 (0.1) resulted in a decrease in the performance metrics of all the models respectively. Table 2 illustrates how the Inception V3 suffers a massive drop in all metrics (accuracy, precision, recall, and F1-score). By comparison, ResNet-50 faced a much smaller decrease in metrics, but the precision, recall and F1 score of VGG-11 dropped to nearly 0. This consequently led to a major anomaly (when VGG-11 was run with learning rate 1e-1), where the model predicted the same class for all the images (see the confusion matrix in figure 7). These results, we hypothesize, are due to the step size being too large, such that the model constantly overshoots and misses the local minima. Hence, we can conclude that a smaller step size yields greater performance results, even if it causes the model to take longer to train.

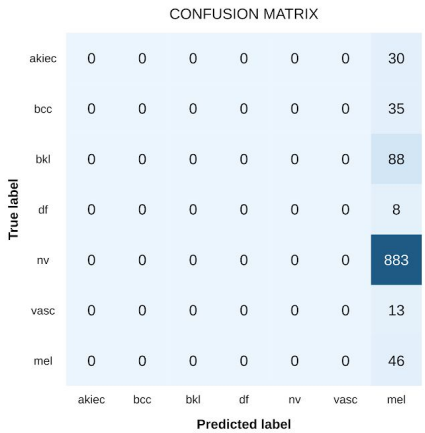

FIG. 7. Confusion matrix for VGG-11 (learning rate 0.1)

Afterwards we experimented with the batch size as we doubled it to 64 for all the models. This had a very minor effect on the performance metrics: they were largely the same, with only minimal changes from the metrics of the control (the default versions for each architecture). However, the larger batch sizes quickly depleted GPU memory, making them difficult to run at times.

The final change was swapping the optimizer from Adam to Stochastic Gradient Descent (SGD), with a momentum of 0.9; in the case of Inception alone this degraded the maximum testing accuracy, decreasing it by 0.4%, but in the other two architectures, changing the architecture resulted in increases in the performance metrics. While these increases were extremely minute (<0.5%) for ResNet, VGG's improvements were significant in terms of testing accuracy and recall: 2% and 0.06 respectively.

B. Analysis of models

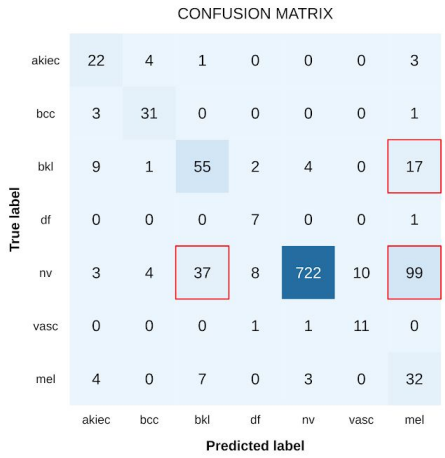

FIG 8. Confusion matrix for VGG-11 with default parameters

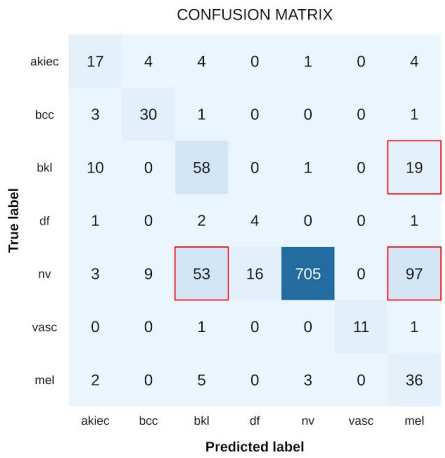

FIG 9. Confusion matrix for VGG-11 with batch size 64

Through the analysis of generated confusion matrices (see Figures 8, and 9), we were able to compare between predictions and ground truths for classifications between the 7 categories. Among the categories that were classified, on average, actinic keratoses ("akiec"), basal cell carcinoma ("bcc"), and melanoma ("mel") had relatively high percentages of correctly predicted labels. The same can be said for melanocytic nevi ("nv"), but it should be noted that this is only because the distribution of the dataset meant that the "nv" label had an overwhelmingly large subset (around 800, as compared to the two digit sizes of the other subsets). This skewed the relevant accuracies for the "nv" class to extremely large numbers, even when the testing accuracies weren't particularly high.

On the other hand, the predictions made some mistakes that were common across all models. The first was mistaking melanocytic nevi for melanoma and the second was mistaking melanocytic nevi for benign keratosis-like lesions ("bkl"). In figures 7 and 9, these are seen by the cells with red outlines. This illustrates the earlier evaluation of the "nv" label: there were high accuracies for predicting "nv", but only because the "nv" subset was massive; a closer look shows that in spite of these high accuracies, the "nv" label also had some of the most common instances of misclassification.

The other common error across models was "bkl"-labelled images being mistaken for the "mel" class. Note that this common misclassification (along with all of the others discussed prior) is on *average*, after considering all the calculated matrices. This means that all of the models are colloquially poorest at correctly recognising melanocytic nevi and benign keratosis-like lesions.

To see how these images may have been mistaken for each other, we took a closer look at their visual appearance in figure 10, which shows us the trends over which the classification model has been recorded to frequently err.

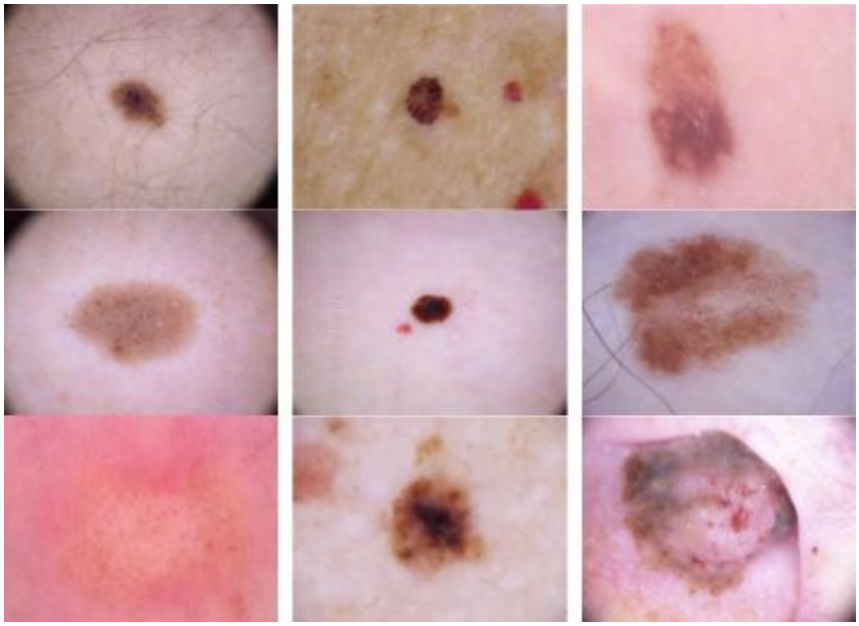

FIG 10. Multi-image comparison of melanocytic nevi (left column), melanoma (center column), and benign keratosis-like lesions (right column).

C. Optimal implementation

Upon analyzing the effects that these changes in the hyperparameter tuning had on the model and the final metrics, we saw that the Google Inception V3 model was the best in all the four metrics: testing accuracy, weighted precision, weighted recall and weighted F1 score. Based on our results, we tested to find the optimal model: learning rate 1e-3, batch size 64, and SGD with momentum optimizer. While evaluating the performances of different models, we prioritized the recall (calculated for each individual class) over all other metrics given the medical use case (we would rather misclassify a negative cancer case than miss a positive case). Hence, we chose the SGD optimizer, which had a lower testing accuracy but a higher recall, for the optimal combination of parameters.

The results were promising, as we were able to raise our previous highest training accuracy of 94% to 99%, as well as a new highest testing accuracy of 92% (figure 12 shows the training and testing accuracies of the model across ten epochs of training). Furthermore, we achieved new highs in the weighted precision, recall and F1 scores: 0.93, 0.92, and 0.92 respectively. The model importantly caused a reduction in the aforementioned misclassification of categories (such as melanocytic nevi and melanoma), with misclassification counts following from the high 90s to single digits (see figure 11).

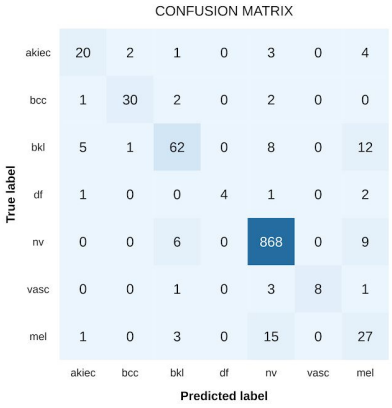

FIG. 11. Confusion matrix for the optimal model

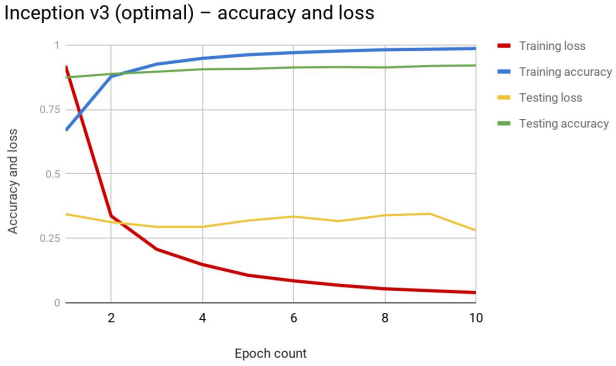

FIG. 12. Accuracy and loss values of the optimal model

IV. CONCLUSION

Over the course of our research, we explored the application of different neural networks in the classification and detection of skin cancer through visually processing dermoscopic images of skin lesions. By conducting such experiments across different architectures and hyperparameters, we demonstrated their relative performances, thus establishing Google's Inception v3 neural network as the optimal choice, while also providing insights into what manner of hyperparameters further improve the model's performance.

We believe that there is still great scope for improvement in this topic. Extending the range and values of hyperparameters tested can lead to even better models; in particular, we wanted to test the impact of different loss functions (all our training scenarios used only cross entropy loss), and a wider range of learning rates and optimizers, if time permitted. However, the chief shortcoming of our research was definitely the maldistribution of our dataset: a majority of the images were of melanocytic nevi (see Figure 7), which meant that high accuracy could be achieved simply by predicting more of that particular label, instead of consistently predicting the *correct* label. As such, we definitely aspire to work with a more balanced dataset in the future to get more viable results. Although the overall efficacy of our models show promise, there are still ways to go before it can be applied in the real world. Performance metrics, especially recall, need to be near perfect (through more rigorous training and datasets), as the intrinsic threat of skin cancer as a disease raises serious issues of liability. However, steady progress in overcoming these issues presents great opportunities for the application of neural networks for the early detection of skin cancer.

REFERENCES

[1] R. L. Siegel, K. D. Miller and A. Jemal, "Cancer statistics, 2019", *ACS Journal*, January 2019

[2] A. F. Jerant, J. T. Johnson, C. D. Sheridan, and T. J. Caffrey, "Early Detection and Treatment of Skin Cancer", American Family Physician, July 2000

[3] D. Piccolo, A. Ferrari, K. Peris, R. Daidone, B. Ruggeri and S. Chimenti, "Dermoscopic diagnosis by a trained clinician vs. a clinician with minimal dermoscopy training vs. computer-aided diagnosis of 341 pigmented skin lesions: a comparative study", Wiley Online Library, September 2002

[4] J. Kato, K. Horimoto, S. Sato, T. Minowa and H. Uhara, "Dermoscopy of Melanoma and Non-melanoma Skin Cancers", National Center for Biotechnology Information (NCBI), August 2019

[5] Y. Fujisawa, S. Inoue and Y. Nakamura, "The Possibility of Deep Learning-Based, Computer-Aided Skin Tumor Classifiers", National Center for Biotechnology Information (NCBI), August 2019

[6] T. D. Srividya and V. Arulmozhi, "A Review of Threshold based Segmentation for Skin Cancer with Image Processing", *International Journal of Recent Technology and Engineering (IJRTE)*, February 2019

[7] A. Masood and A. A. Al-Jumaily, "Computer Aided Diagnostic Support System for Skin Cancer: A Review of Techniques and Algorithms", *International Journal of Biomedical Imaging*, December 2013

[8] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H.M. Blau and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks", *Nature*, January 2017

[9] A. Hekler, J. S. Utikal, A. H. Enk, A. Hauschild, M. Weichenthal, R. C. Maron, C. Berking, S. Haferkamp, J. Klode, D. Schadendorf, B. Schilling, T. Holland-Letz, B. Izar, C. V. Kalle, S. Fröhling and T. J. Brinker, "Superior skin cancer classification by the combination of human and artificial intelligence", *European Journal of Cancer*, October 2019

[10] A. Nylund, "To be, or not to be Melanoma: Convolutional neural networks in skin lesion classification", Kth Royal Institute Of Technology School Of Technology And Health, Stockholm, Sweden, 2016

[11] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", arxiv, December 2015

[12] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for large-scale image recognition", in *International Conference on Learning Representations (ICLR) 2015*, April 2015

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision”, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, December 2015

[14] A. Krizhevsky, I. Sutskever and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, in *Advances in neural information processing systems*, January 2012, pages 1097–1105

[15] J. Deng, W. Dong, R. Socher, L. Li, K. Li and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database”, *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009

[16] P. Tschandl, C. Rosendahl and H. Kittler, “The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions”, *Nature*, January 2017

ACKNOWLEDGEMENTS

We give acknowledgement to Jedrzej Kozerawski and Aiwen Xu for their mentorship and guidance throughout this project and the program, in addition to the Summer Research Academies and the University of California, Santa Barbara for this unique opportunity.

AUTHOR CONTRIBUTION STATEMENT

All authors conceived the experiments, K.S. devised the code for the experiments, S.M. and A.K. conducted the experiments, A.K and S.M. analysed the results. All authors wrote and reviewed the manuscript.