Convolutional Neural Network Comparison using Human Against Machine 10,000 Dataset

Alexandro Ayala

Department of Electrical and Computer Engineering

California State University, Chico

Chico, United States

Aayala4@csuchico.edu

***Abstract*—This report serves as an exploration and comparison of different convolutional neural network architectures for the use of detecting various skin cancers and lesions of the skin. Performance metrics are gathered and weighed to examine the drawbacks and benefits of each chosen architecture. The neural networks used for this report are GoogLeNet, ResNet50, VGG19, and MobileNetV2.**

I. INTRODUCTION

With non-malignant skin cancer being so prevalent in the Western world [1], it is imperative that we develop techniques to prevent or identify early stages to have a fighting chance. Even with our advancements in imaging technology for medical applications such as optical coherence tomography [2], there still needs to be someone or something to process and recognize the data in the image. That is where a Convolutional Neural Network would be utilized. Convolutional Neural Networks can imitate the mechanisms in our visual cortex to recognize images [3] which prove to be vital in the ongoing effort of medical image processing. This report does not serve to create a breakthrough in this ever-growing field of biomedical technology; instead, it serves as a comprehensive analysis between current popular convolutional neural network architectures. The architectures chosen for this report are GoogLeNet, ResNet50, VGG19, and MobileNetV2.

II. LITERATURE REVIEW

A. GoogLeNet

The GoogLeNet architecture is derived from another architecture called Inception which was showcased in 2014 at the ImageNet Large-Scale Visual Recognition Challenge [4][5]. GoogLeNet itself consists of 22 layers that have parameters with another 5 pooling layers [6]. If you count all of the blocks that make up the architecture, it is around 100 blocks. GoogLeNet is referred to as a convolutional neural network developed, as the name suggests, at Google. As Szegedy et al [6] puts it “the network was designed with computational efficiency and practicality in mind, so that inference can be run on individual devices including even those with limited computational resources, especially with low-memory footprint”.

B. ResNet50

The Residual Network architecture [7] was originally based off the VGG neural network architecture but with a greater depth without an increase in complexity. This architecture would go on to win 1st place in the ILSVRC 2015 classification competition. There are variations of the Residual Network such as ResNet18 and ResNet34. What ResNet50 did was replace every 2-layer block with a 3-layer bottleneck block [7]. This block reduces the number of parameters and matrix multiplications which allows for faster training of each layer.

C. VGG19

VGG-19 is a convolutional neural network made by the Visual Geometry Group. It consists of 16 convolution layers, 3 fully connected layers, 5 maxpool layers, and 1 softmax layer [8]. The VGG networks were to serve as a successor to the AlexNet neural network [9]. Just like the previous networks mentioned, it used the ImageNet image database which consists of 14,197,122 images [5].

D. MobileNetV2

This architecture aimed to improve neural networks on a mobile platform capacity. It is based on an inverted residual structure where the residual connections are between the bottleneck layers. The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. The architecture contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers [10].

E. Dataset

The dataset the neural networks are going to be training on comes from Harvard and is called the Human Against Machine 10,000 dataset [11]. It has 7 classifications of various skin cancers and lesions. These classifications are Melanocytic Nevi (nv), Melanoma (mel), Benign Keratosis-like Lesions (bkl), Basal Cell Carcinoma (bcc), Actinic Keratoses (akiec), Vascular Lesions (vas), and Dermatofibroma (df). The files are named with an identification number, and this is used in conjunction with a metadata file to organize each image into their respective classification folder. This was done through a small script in python, but the rest of the work is going to be done using the MATLAB Deep Network Designer Toolbox. The images need to be resized from 600x450 to 224x224 to be able to work with the chosen neural networks. This was done with a simple script in MATLAB.

III. METHODS AND MATERIALS

A. Parameters

The training parameters chosen were picked through trial and error while also keeping in mind the time it would take to train each neural network. An RTX 2070 Super graphics card was used for the training which considerably reduced the trained time compared to a processor, but even so, as the results will show, there was a deadline to meet so only a few parameters were tweaked. The first thing that had to be done was a replacement of the final fully connected layer and classification layer. The output size of the fully connected layer was changed to 7 and the Weight Learn Rate Factor and Bias Learn Rate Factor was changed to 10. The other settings were kept at the default of 1 and 0. The learning rate chosen that gave the best accuracy was 1e-3. The validation frequency was 5, batch size 64, and the max epochs was 30. Everything else was again kept default. In terms of augmentation, the x-axis was randomly reflected with a random rotation from -90 to 90 degrees, and a random rescaling from 1 to 2.

B. Performance Metrics

Various performance metrics will be extrapolated using the data from a generated confusion matrix for each of the neural networks. The confusion matrix generates true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values which are then used in each performance metric equation. The tabulated metrics include accuracy (1), precision (2), specificity (3), and sensitivity (4) [12].

$$Accuracy = \frac{TP + TN}{Total} \tag{1}$$

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

$$Specificity = \frac{TN}{FN+FP} \tag{3}$$

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

After these metrics are gathered, the overall accuracy (5), macro-precision (6), and macro-sensitivity (7) can be found. These metrics are an average of all the classifications where n is the number of classifications, P is precision, and S is sensitivity.

$$Overall\ Accuracy = \left(\frac{1}{total}\right) \sum_{i=1}^{n} TP_i \tag{5}$$

$$Macro - Precision = \left(\frac{1}{n}\right) \sum_{i=1}^{n} P_i \tag{6}$$

$$Macro - Sensitivity = \left(\frac{1}{n}\right) \sum_{i=1}^{n} S_i \tag{7}$$

IV. EXPERIMENT AND RESULTS

A. GoogLeNet Training

The GoogLeNet architecture took 60 minutes and 51 seconds at 30 epochs with a validation accuracy of 76.85%. Using the generated confusion matrix in Fig. 1 and the equations mentioned previously Table 1 was formed.

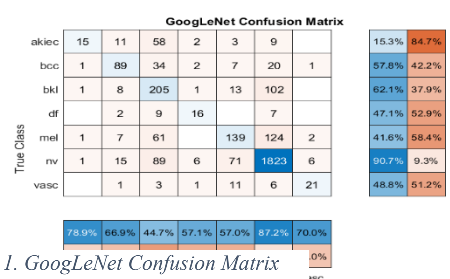

Fig. 1. GoogLeNet Confusion Matrix

TABLE I. GOOGLENET PERFORMANCE METRICS

GoogLeNet	TP	TN	FP	FN	Accuracy	Precision	Specificity	Sensitivity
akiec	15	2902	4	83	0.971	0.789	0.998	0.153
bcc	89	2806	44	65	0.963	0.669	0.984	0.577
bkl	205	2420	254	125	0.873	0.446	0.905	0.621
df	16	2958	12	18	0.990	0.571	0.995	0.470
mel	139	2565	105	195	0.900	0.569	0.960	0.416
nv	1823	725	268	188	0.848	0.871	0.730	0.906
vasc	21	2952	9	22	0.989	0.7	0.996	0.488

B. ResNet50 Training

ResNet50 achieved a validation accuracy of 76.25% with 30 epochs. The elapsed time was 156 minutes and 6 seconds. Fig. 2 was then generated, and the data was tabulated in Table 2.

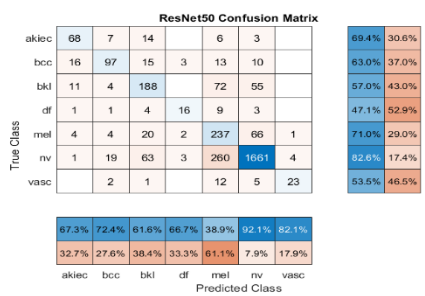

Fig. 2 ResNet50 Confusion Matrix

TABLE II. RESNET50 PERFORMANCE METRICS

ResNet50	TP	TN	FP	FN	Accuracy	Precision	Specificity	Sensitivity
akiec	68	2873	33	30	0.979	0.673	0.988	0.693
bcc	97	2813	37	57	0.968	0.72	0.98	0.629
bkl	188	2557	117	142	0.913	0.616	0.956	0.569
df	16	2962	8	18	0.991	0.666	0.997	0.470
mel	237	2298	372	97	0.843	0.389	0.860	0.709
nv	1661	851	142	350	0.836	0.921	0.856	0.825
vasc	23	2952	9	22	0.989	0.7	0.996	0.534

C. VGG19 Training

VGG19 achieved a 72.4% validation accuracy at 30 epochs with an elapsed time of 802 minutes and 40 seconds. This model took considerably longer compared to all the other architectures. Fig. 3 shows the confusion matrix and like the others, the performance metrics are calculated off of it in Table 3.

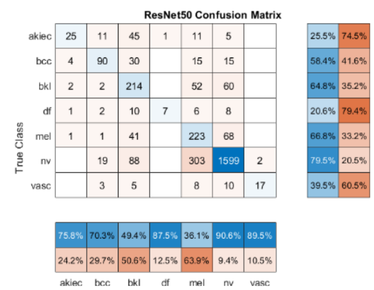

Fig. 3 VGG19 Confusion Matrix

TABLE III. VGG19 PERFORMANCE METRICS

VGG19	TP	TN	FP	FN	Accuracy	Precision	Specificity	Sensitivity
akiec	25	2898	8	73	0.973	0.757	0.997	0.255
bcc	90	2812	38	64	0.966	0.703	0.986	0.584
bkl	214	2455	219	116	0.888	0.494	0.918	0.648
df	7	2969	1	27	0.990	0.875	0.999	0.205
mel	223	2275	395	111	0.831	0.360	0.852	0.667
nv	1599	827	166	412	0.807	0.905	0.832	0.795
vasc	17	2959	2	26	0.990	0.894	0.999	0.395

D. MobileNetV2

MobileNetV2 achieved a validation accuracy of 74.5% during the 30 epochs which took 106 minutes and 58 seconds which is more in line with the first two architectures. Fig. 4 shows us the confusion matrix and Table 4 shows us the performance metrics.

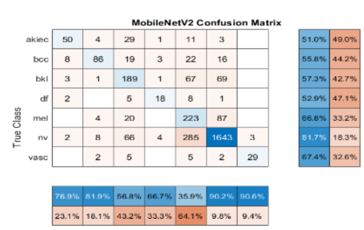

Fig. 4. MobileNetV2 Confusion Matrix

TABLE IV. MOBILENETV2 PERFORMANCE METRICS

MobileNetV2	TP	TN	FP	FN	Accuracy	Precision	Specificity	Sensitivity
akiec	50	2891	15	48	0.979	0.769	0.994	0.510
bcc	86	2831	19	68	0.971	0.819	0.993	0.558
bkl	189	2530	144	141	0.905	0.567	0.946	0.572
df	18	2961	9	16	0.991	0.666	0.996	0.529
mel	223	2272	398	111	0.830	0.359	0.850	0.667
nv	1643	815	178	368	0.818	0.902	0.820	0.817
vasc	29	2958	3	14	0.994	0.906	0.998	0.674

E. Overall Performance Comparison

TABLE V. PERFORMANCE METRICS RESULTS

CNN	Overall Accuracy	Macro-Precision	Macro-Sensitivity
GoogLeNet	0.76830	0.65974	0.51911
ResNet50	0.76231	0.68743	0.63349
VGG19	0.72403	0.71306	0.50743
MobileNetV2	0.74500	0.71287	0.61855

V. DISCUSSION

Overall, this was a success with training various convolutional neural network architectures and extrapolating their data so that a thorough analysis of their respective performance metrics can be measured and tabulated. These pre-trained architectures are normally used for large datasets which may cause issues when training from datasets comparatively smaller such as the one used for this report. Parameters could have been further tweaked and tested to increase the overall performance, but the close deadline of the project prevent me from taking too much time in that regard. From the data we see GoogLeNet had better accuracy at the cost of some precision and sensitivity.

REFERENCES

[1] Neville JA, Welch E, Leffell DJ. Management of nonmelanoma skin cancer in 2007. *Nat Clin Pract Oncol* 2007;**4**:462–9.

[2] MOGENSEN, METTE, et al. "Assessment of Optical Coherence Tomography Imaging in the Diagnosis of Non-Melanoma Skin Cancer and Benign Lesions Versus Normal Skin: Observer-Blinded Evaluation by Dermatologists and Pathologists." *Dermatologic Surgery*, vol. 35, no. 6, 2009, pp. 965–72, https://doi.org/10.1111/j.1524-4725.2009.01164.x.

[3] Kim, Phil. *MATLAB Deep Learning : With Machine Learning, Neural Networks and Artificial Intelligence*, Apress L. P., 2017. *ProQuest Ebook Central*, https://ebookcentral.proquest.com/lib/csuchico/detail.action?docID=4878025.

[4] Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. *IJCV,* 2015.

[5] *ImageNet*. http://www.image-net.org

[6] Szegedy, Christian, et al. Going Deeper with Convolutions. 2014, https://doi.org/10.48550/arxiv.1409.4842.

[7] He, Kaiming, et al. "Deep Residual Learning for Image Recognition." arXiv.org, 2015.

[8] Simonyan, Karen, and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014, https://doi.org/10.48550/arxiv.1409.1556.

[9] Krizhevsky, Alex, et al. "ImageNet Classification with Deep Convolutional Neural Networks." Communications of the ACM, vol. 60, no. 6, 2017, pp. 84–90, https://doi.org/10.1145/3065386.

TABLE VI. Performance Comparison Results

[10] Sandler, Mark, et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018, https://doi.org/10.48550/arxiv.1801.04381.

[11] Tschandl, Philipp, 2018, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions", https://doi.org/10.7910/DVN/DBW86T, Harvard Dataverse, V3, UNF:6:/APKSsDGVDhwPBWzsStU5A== [fileUNF]

[12] Chen, Qingguang, et al. "Hierarchical CNN-Based Occlusal Surface Morphology Analysis for Classifying Posterior Tooth Type Using Augmented Images from 3D Dental Surface Models." Computer Methods and Programs in Biomedicine, vol. 208, 2021, pp. 106295–106295, https://doi.org/10.1016/j.cmpb.2021.106295.