Skin Lesion Classification using Convolutional Neural Networks

Tareq Tayeh, *Member, IEEE,* Sulaiman Aburakhia, *Member, IEEE,* Moataz ElKhishen, *Member, IEEE*

***Abstract*—Skin cancer represents a major public health concern as it affects many people in Canada and around the world, where it is the most common of all cancer types. If a patient with a pigmented lesion could be identified as someone with or at a risk of developing skin cancer, then measures could be taken right away to lower their risk or destroy the cancer if developed at an early stage. Thus, a reliable and automatic skin lesion classification system would be essential in detecting a malignancy. In this paper, Convolutional Neural Networks (CNN) were utilized to accurately classify pigmented skin lesions in dermoscopic images to detect the malignant skin lesions as early as possible. CNNs excels in analyzing visual imagery as they are fully connected feed forward neural networks that reduce the number of parameters very efficiently without losing out on the quality of models. The dataset used had a huge class imbalance, which was addressed using an equal train/validation/test split across all classes, as well as direct class loss weighting during the model training. Four different CNN models were employed to analyze and predict the pigmented lesions classes. The models used were InceptionV3, ResNet152V2, VGG16, and our own CNN architecture "TSM12". Evaluation results showed that TSM12 performance was comparable to InceptionV3 and ResNet152V2, and closely competing with our tweaked VGG16 model which topped all our scoring metrics, noticeably scoring an 89% weighted classification accuracy.**

***Index Terms*—Convolutional Neural Networks, Deep Learning, Skin Lesion Classification, Loss Weighting, Dermoscopy.**

I. Introduction

SKIN cancer is defined as the abnormal growth of skin cells that is most often developed on skin exposed to the sun, primarily the areas of the scalp, face, lips, ears, neck, chest, arms, hands, and legs. It affects people of all skin tones, including those with darker complexions. It represents a major public health concern as it affects many people in Canada and around the world, where it is the most common of all cancer types. According to the government of Canada [1], about a third of all new cases in Canada are skin cancers, and the rate keeps growing.

Lesion analysis is an important aspect of medicine. The healthcare industry collects an enormous amount of healthcare data from patients through medical studies, clinic appointments and hospital visits. This data can be analyzed to discover and realize hidden figures and statistics about an illness that could offer a breakthrough in care, treatment, or diagnosis of patients. Accurate analysis of data, such as malignant pigmented lesions, benefits early skin cancer detection, which in turn could lower the number of patients dying, as measures could be taken right away to lower the patient's risk or destroy the cancer. Malignant pigment lesions include melanoma, the deadliest type of skin cancer, basal cell carcinoma, and some vascular lesions. Automated analysis could be done via machine learning (ML), which would create a system and framework in the medical field that would help in providing contextual relevance, improving clinical reliability, helping physicians communicate objectively, reducing errors related to human fatigue, decreasing mortality rates, diminishing medical costs, and identifying diseases more readily [2]. A machine learning algorithm that can classify both malignant and benign pigmented skin lesions is a step in the right direction towards providing these benefits.

In this paper, Convolutional Neural Networks (CNN) were utilized to accurately classify pigmented skin lesions in dermoscopic images to detect the malignant skin lesions as early as possible. A CNN is a class of deep neural networks that use convolution in place of general matrix multiplication in at least one of their layers. It excels in analyzing visual imagery as they are fully connected (FC) feed forward neural networks that reduce the number of parameters very efficiently without losing out on the quality of models. Four different CNN models were employed to analyze and predict the pigmented lesions classes. The models used were our own CNN architecture "TSM12", InceptionV3, ResNet152V2, and VGG16. Information on these models can be found in section II. The models were trained and then compared using the metrics described in section II-F. The data set used in this paper is "The HAM10000 dataset, a large collection of multi-source dermoscopic images of common pigmented skin lesions" [3], and is described in the 'Data Set' section IV. It is very common for data sets in the medical field to have major class imbalances, and this data set was no different. It proved to be a key challenge in this project.

This paper is organized as follows: section II details the evaluation metrics that were used for the models, as well as some basic information on how the models operate; section III contains information on similar research in this field; section IV contains information about the data set that was used; section V discusses the pre-processing methods performed on the data; section VI details the specifics of the models used; section VII discusses all of the model's performances; section VIII contains the final conclusions derived from the research; Lastly, section IX will discuss the future work and next steps for this project.

The authors are with the Department of Electrical and Computer Engineering, Western University, London, ON N6A 3K7, Canada (e-mail: ttayeh@uwo.ca; saburakh@uwo.ca; melkhish@uwo.ca).

Manuscript received April 20, 2020.

II. Background

This section will discuss the required background information about CNNs, the different models used, and the comparison metrics applied to evaluate them.

A. CNN

A CNN is a class of deep neural networks that comprise of FC feed forward neural networks and utilize convolution in place of general matrix multiplication in at least one of their layers. This allows the network to reduce the number of parameters very efficiently without losing out on the quality of models, thriving in analyzing visual imagery. The network keeps on learning new higher dimensionality and more complex features with every layer. It comprises of several kinds of layers:

1) Convolutional Layer: Applies a filter that scans the whole image a couple of pixels at a time, producing a feature map for class probabilities predictions of each feature.

2) Pooling Layer: A layer that scales down the information produced from the convolution layer yet still maintaining the most essential information. Different types of pooling include max pooling and average pooling.

3) FC Input Layer: Flattens previous layer's outputs into one vector for use in the next FC layers.

4) FC Output Layer: Determines the image class by generating the final probabilities.

B. TSM12

TSM12 is our own CNN architecture implementation. It consisted of 8 layers and 4 core block layers, hence the number 12 in the name to indicate the total number of layers. The letters TSM indicate the first letter of each author's name: "Tareq", "Sulaiman", "Moataz".

C. InceptionV3

InceptionV3 [5] is the third version roll out of the widely-used Google's Inception CNN. It is a 48-layered model that was initially introduced during the ImageNet recognition challenge in 2015 [6], where it was the 1st runner up. It attained an accuracy of 78.1% and greater in the challenge.

D. ResNet152V2

ResNet152V2 [7] is a 152-layered residual neural network (ResNet) developed by Microsoft's research team. It builds on constructs known from pyramidal cells in the cerebral cortex, by skipping connections or jumping over layers. ResNet won the 1st place in the ImageNet recognition challenge in 2015, having achieved a 3.57% top-5 error. It focused on increasing the network depth rather than width, as well as reducing the effect of the vanishing gradient problem.

E. VGG16

VGG16 [8] is a 16-layered CNN model that utilizes an architecture of increasing depths with very small convolution filters. The model won 1st place in the ImageNet recognition challenge in 2014, having achieved a 92.7% top-5 test accuracy in ImageNet.

F. Comparison Metrics

Comparison metrics were employed to evaluate the models used and compare their performances. Five different metrics were used in this paper.

1) Confusion Matrix: A confusion matrix is a table that is used to describe the performance of a classification model on a set of test data for which the true values are known. It provides insights into the true and false positives and negatives of a model. The goal is to maximize the values of the diagonal of the matrix which encompass the true positives and true negatives, while minimizing the values that are off the diagonal that encompass the false positives and false negatives.

2) Accuracy: The accuracy of a model is defined as the ratio of true predictions to total predictions. The higher the accuracy, the more true predictions the model produced for that class.

3) Recall: The recall of a model is defined as the ratio of true positives to the total true positives and false negatives. The higher the recall, the lower false negative predictions the model produced for that class.

4) Precision: The precision of a model is defined as the ratio of true positives to the total true positives and false positives. The higher the precision, the lower the false positive predictions the model produced for that class.

5) F1-Score: The F1-Score of a model is a balanced measure between precision and recall, utilizing harmonic mean to weigh the lower value heavier.

III. Related Work

Analyzing microscopy pigmented skin lesions using artificial neural networks (ANN) is dated back to the early 1990s [9] [10]. In the recent literature, considerable research was conducted using different machine learning techniques for skin lesion classification. In 2017, N. Codella et al. have combined hand-coded feature extractors, sparse-coding methods and Support-Vector Machines (SVM) with neural networks techniques for melanoma recognition and segmentation in dermoscopy images [11]. The proposed system demonstrated a classification accuracy higher than the average accuracy of 8 expert dermatologists. In 2019, U. Yildiz and V. Kiliç trained the classifiers by extracting various color and texture features from the data set, detecting Melanoma with up to 97% accuracy [12]. N. Kinyanjui et al. collected different skin tones across populations, where individual typology angle (ITA) was used to approximate skin tone in dermatology data sets and compare non-diseased areas of the skin with the affected ones [13]. Furthermore, N. Gessert et al. examined the effectiveness of using high resolution images with pre-trained standard architectures, taking into consideration the high imbalance in multi-class data sets [14].

IV. Data set

As mentioned earlier, the data set used in this paper was "The HAM10000 dataset, a large collection of multi-source dermoscopic images of common pigmented skin lesions". It consisted of 10,015 dermoscopic skin pigmented lesion 600 by 450 pixel images, digitized and stored as JPEG images.

They were initially manually cropped and centered around the lesion, as well as adjusted to enhance visual contrast and color reproduction.

The data set included 7 attributes associated with each image and patient, which were:

- Patient's Age [age]
- Patient's Sex [sex]
- A Lesion ID [lesion_id]
- An Image ID [image_id]
- A Technical Validation Field Type [dx_type]
- The Localization of the Skin Lesion [localization]
- A Diagnostic Skin Lesion Category [dx]

Firstly, the data set was statistically interpreted. Figure 1 shows the main descriptive statistics of the data set contents.

	lesion_id	image_id	dx	dx_type	sex	localization
count	10015	10015	10015	10015	10015	10015
unique	7470	10015	7	4	3	15
top	HAM_0000835	ISIC_0027704	nv	histo	male	back
freq	6	1	6705	5340	5406	2192

	age
count	10015.000000
mean	51.863828
std	16.920252
min	0.000000
25%	40.000000
50%	50.000000
75%	65.000000
max	85.000000

Fig. 1: Main descriptive statistics of the data set contents

We can see how there was a unique image id for each entry, but not a unique lesion id. This indicated that there were duplicate images for the same lesion id but at a different distortion, such as an angle, shear, or zoom distortion. In addition, class nv dominated the skin lesion categories, by having a frequency of 6,705 out of the 10,015 images we got, directly signaling a class imbalance issue in the data set. Before analyzing the skin lesion categories further, let us analyze the rest of the attributes first.

Figure 2 shows the distribution of the patients' age. It can be seen that the majority of patients resided between the ages of 35 and 70.

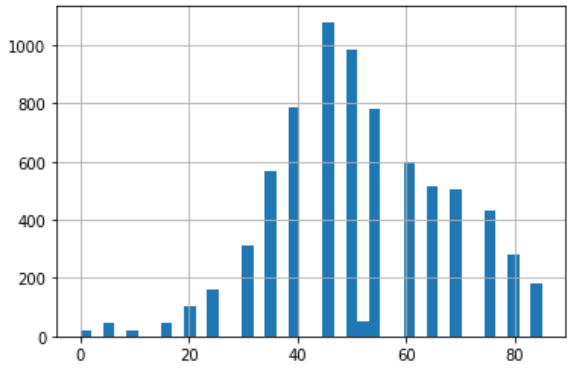

Fig. 2: Patients' Age Distribution

Figure 3 shows the distribution of the patients' sex. It can be seen that there was almost an equal amount of male to female patients.

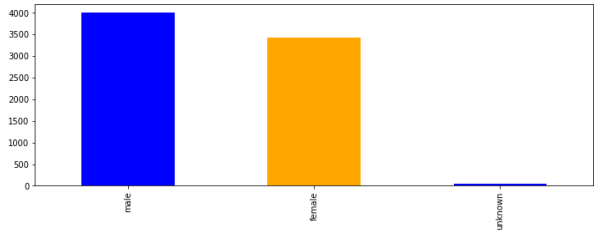

Fig. 3: Patients' Sex Distribution

The technical validation field category represented the ground truth of the data set and indicated how the skin lesion diagnosis was made. The publishers defined four different types of ground truths which were:

- Histopathology: Diagnoses of excised lesions have been performed by specialized dermatopathologists.
- Confocal: Diagnoses of excised lesions have been based on reflectance confocal microscopy.
- Follow-up: This type is limited to nevi class only, where digital dermatoscopy did not show any changes during 3 follow-up visits or 1.5 years.
- Consensus: Based on experts consensus. This type is defined for typical benign cases without histopathology or follow-up, and where two experts have provided same unequivocal benign diagnosis.

Figure 4 shows the distribution of the technical validations. As shown in the figure, more than 50% of the skin lesion diagnosis were based on histopathology.

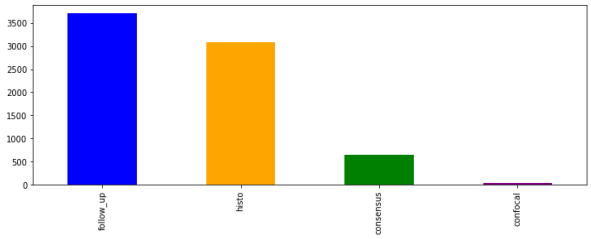

Fig. 4: Technical Validation Distribution

Figure 5 shows the localization distribution of the data set. It can be seen that the back, lower extremity, and trunk are heavily compromised skin cancer regions.

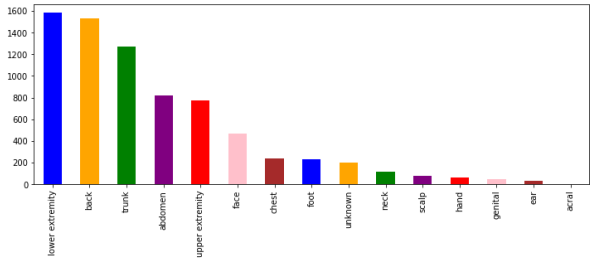

Fig. 5: Localization Distribution

As for the diagnostic skin lesion categories, seven different classes were present in the data set. Figure 6 shows sample

images from the data set for each class. The seven categories were:

- Melanocytic Nevi [nv]: Benign neoplasms of melanocytes and appear in a myriad of variants. The variants may differ significantly from a dermatoscopic point of view. [6705 images]
- Melanoma [mel]: Malignant neoplasm derived from melanocytes that may appear in different variants. If excised in an early stage, it can be cured by simple surgical excision. [1113 images]
- Benign Keratosis-like Lesions [bkl]: Can be regarded as a flat variant of seborrheic keratosis and lichen-planus like keratoses (LPLK), which corresponds to a seborrheic keratosis or a solar lentigo with inflammation and regression. [1099 images]
- Basal Cell Carcinoma [bcc]: A common variant of epithelial skin cancer that rarely metastasizes but grows destructively if untreated. [514 images]
- Actinic Keratoses [akiec]: Common non-invasive, variants of squamous cell carcinoma that can be treated locally without surgery. [327 images]
- Vascular Lesions [vasc]: These range from cherry angiomas to angiokeratomas and pyogenic granulomas, meaning they could be benign or malignant. [142 images]
- Dermatofibroma [df]: Benign skin lesions regarded as either a benign proliferation or an inflammatory reaction to minimal trauma. [115 images]

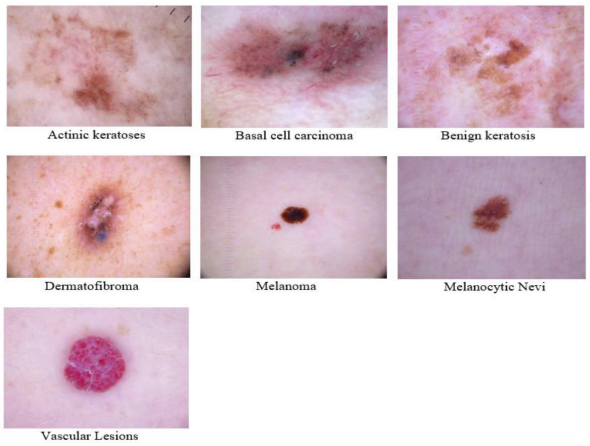

Fig. 6: Sample images for the seven skin lesion categories

Figure 7 visualizes the skin lesion categories distributions. As mentioned earlier, it can be clearly seen that melanocytic nevi [nv] represented the majority of the data set entries, and dermatofibroma had the fewest representations. This illustrated a key class imbalance challenge which was addressed in the next section in this paper.

V. Data Pre-Processing

A. Data Editing and Cleansing

1) Created two dictionaries for the images and their labels. The first dictionary included image names that were retrieved from the various image folders of the downloaded data set. Afterwards, a second dictionary was created to map the diagnostic skin lesion categories code to the category full name.
2) Data cleansing. This involved dropping lesion_id's with duplicate images, keeping only one image for each lesion_id. These images are of the same exact lesion but at a different angle, zoom etc. In addition, there were fifty-two NA values in the age entries. These were replaced by the age mean of the data set. That ensured reasonable estimation of what the missing values could be rather than ignoring the entire patient's record.
3) Unique numeric codes were created for each skin lesion category to assist with the predictions to be made later on, as integers were easier to handle than strings.
4) Images were resized and processed. Width and height sizes would have be an issue when it came to training our CNN models, due to the huge amount of images on hand. To speed up the process and ensure our CNN models worked smoothly, images were resized by a factor of 0.25. The new images were 150 by 112.5 pixels. Afterwards, the image was flattened and stored as a numeric image list.
5) Applied one-hot encoding to the skin lesion categories unique numeric codes that were created. This is where the integer encoded variable was removed and a new binary variable was added for each unique integer value, producing the (1,7) row vector in our case as we had seven unique label integer values. This was needed as there was no ordinal relationship between the label's integer values. It eliminated any natural integer ordering assumption the CNN algorithm may conclude.

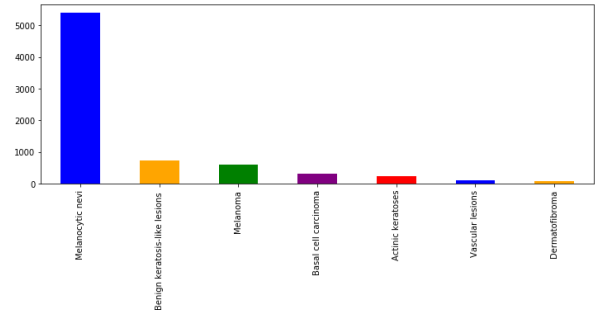

Fig. 7: Skin Lesion Categories Distribution

B. Data Splitting

1) Feature and target split. The feature used in this project was the flattened numeric image lists created in the previous subsection. The target was the one-hot encoding created for the skin lesion categories.
2) Training 70 / Validation 10 / Test 20 split. The data was split 70:10:20 respectively across each class individually to ensure there was enough samples from each class in each split for accurate modeling. This was necessary due to the huge class imbalance demonstrated by the data set. A 70:10:20 is a split commonly used when the data is less than 100,000 entries.

C. Feature Normalization

The image feature was then normalized. Normalization is a scaling technique in which values are shifted and re-scaled so that they end up ranging between 0 and 1. The normalization of each image was done by subtracting it's values from the training's mean value and then dividing by the training's standard deviation. The normalization formula is equation 1.

$$x\_new = \frac{x - \mu}{\sigma} \tag{1}$$

D. Data Augmentation

All the original images were transformed and augmented every epoch and then used for training to avoid overfitting. This allowed the model to be more robust and accurate, as it was trained on different variations of the same image. The number of images in each epoch was equal to the number of original images. The images were:

- Randomly rotated by 20%
- Randomly shifted horizontally by 20%
- Randomly shifted vertically by 20%
- Randomly sheared by 10%
- Randomly zoomed by 10%
- Randomly channel shifted by 10%

This method was preferred over random oversampling, which was one way of dealing with class imbalances. Random oversampling consisted of re-sampling less frequent samples to adjust their amount in comparison with predominant samples. However, the distribution of classes would change significantly, where the smaller classes would be much less variable, and the larger classes would have richer variations.

E. Further Data Editing

Flattened images were reshaped back to [width x height x depth] to be fed into our models.

VI. Models and Implementations

The models were built and implemented in Python 3.7.4, using the Tensorflow and Keras libraries.

A. TSM12

Figure 8 shows a high level overview of the model's architecture.

Table I highlights the layers and hyperparameters used in the network. The following provides a thorough walk-through of the network architecture:

- The first 2 layers consisted of 2 Convolutional 2D layers. Two layers were used rather than 1 as the model had more flexibility in expressing non-linear transformations without losing information, making it easier to learn. MaxPool (next layer) removed information from the signal, and Dropout (the layer afterwards) forced distributed representation, both effectively made it harder to propagate information. 32 filters were used to extract the simple features. ReLu activation function was used here and in the rest of the network layers (except in the last layer) as it solved the vanishing gradient problem and was computationally light. Same padding was used so the output size was the same as the input size.
- Layer 3 consisted of a MaxPool 2D layer, which acted as a down-sampling filter. The 2x2 pool size looked at 2 neighboring pixels and picked the maximal value.
- Layer 4 consisted of a Dropout layer. It is a regularization technique where randomly selected neurons are ignored during training, forcing the network to learn features in a distributed way while improving generalization and reducing overfitting. A quarter of the neurons were ignored in this layer.
- Layers 5-8 are duplicates of layers 1-4 but with hyperparameter value differences. The Convolutional 2D layers used 64 filters rather than 32 filters, to be able to extract more complex features from the images, as typically simple features are extracted first in a deep neural network before extracting the complex features. The Dropout layer had a higher neurons value due to the weight parameter matrices being bigger from the extra filters used in the Convolutional 2D layers.
- Layer 9 consisted of a Flatten layer, which flattened the input onto a single 1D vector.
- Layer 10 consisted of a FC input Dense layer with 128 units.
- Layer 11 consisted of a Dropout layer. Half the neurons were ignored during training to minimize any potential overfitting.
- Layer 12 consisted of a FC output Dense layer with 7 units, the same number of lesion categories in the data set. A softmax activation function was used here rather than ReLu as it mapped a vector to probability of a given output in binary classification, working well with our one-hot encoding of our 7 target skin lesion categories.

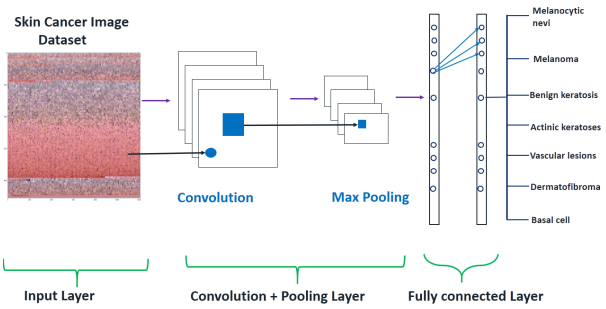

Fig. 8: High level overview of TSM12's Architecture

B. InceptionV3, ResNet152V2, VGG16

These models were pre-built and loaded via the keras.applications library package. Extra layers were added to each model's base network to align and flatten the model's inputs, the number of parameters, and the output.

C. Models Hyperparameters

All the CNN models employed had so many parameters, meaning there were so many possible changes in the archi-

TABLE I: TSM12 Layers and Hyperparameters

Layer #	Layer	Hyperparameters
1	Conv2D	32 Filters, 3x3 Filter Size ReLU Activation, Same Padding
2	Conv2D	32 Filters, 3x3 Filter Size ReLU Activation, Same Padding
3	MaxPool2D	2x2 Pool Size
4	Dropout (Core Layer)	0.25 Neurons
5	Conv2D	64 Filters, 3x3 Filter Size ReLU Activation, Same Padding
6	Conv2D	64 Filters, 3x3 Filter Size ReLU Activation, Same Padding
7	MaxPool2D	2x2 Pool Size
8	Dropout (Core Layer)	0.4 Neurons
9	Flatten (Core Layer)	-
10	Dense	128 Units, ReLU Activation
11	Dropout (Core Layer)	0.5 Neurons
12	Dense	7 Units, Softmax Activation

TABLE II: Models Hyperparameters

Hyperparameter	Value
Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Epochs	50
Batch Size	10
Learning Rate	0.001 - 0.00001 (Reduces on Plateau)

tecture. In addition, training them with the huge data set in hand took quite a long time. Performing hyperparameter optimization in our case would have been a total overkill, specially with the computing resources we had available to us. Therefore, we utilized some common hyperparameter values and looked further into techniques to achieve a better model evaluation. Table II highlights the hyperparameter values used across all four CNN models.

The following explains the reasoning behind the hyperparameters values selected:

- Optimizer: Adam [15] is the most common optimization algorithm used today for training deep neural networks, as it is straightforward to implement, computationally very efficient, and is very effective in dealing with large data and parameters. Adam can be looked as a combination of RMSprop and Stochastic Gradient Descent with momentum.
- Loss Function: Categorical cross-entropy is a loss function that is used for single label categorization. This is when only one category is applicable for each data point. This worked perfectly here as one example could only belong to one of the seven skin lesion categories.
- Epochs: Upon multiple initial trials with values of 20, 50, 100, 150 and 200, 50 epochs was sufficient to get to the most optimum results.
- Batch Size: Upon multiple initial trials with values of 5, 10, 20, and 40, a batch size of 10 produced the most optimum results.
- Learning Rate: A learning rate annealer was utilized here. A decreasing learning rate during training enabled the global minimum of a loss function to be reached efficiently. Learning rate started at 0.001 and decreased by factor of 0.5 if the validation accuracy was not improved after 3 epochs (patience).

TABLE III: Class Weights

Skin Lesion Category	Weight
Actinic keratoses [akiec]	1.0
Basal cell carcinoma [bcc]	1.0
Benign keratosis-like lesions [bkl]	1.0
Melanoma [mel]	1.0
Melanocytic nevi [nv]	3.0
Vascular lesions [vasc]	1.0
Dermatofibroma [df]	1.0

D. Class Weights

Weights were added to make the models more sensitive to the Melanocytic nevi [nv] skin lesion category due to the classes imbalance, as it represented around two-thirds of the data. The weights were the factors by which the loss value was multiplied internally for use in the backpropagation algorithm. Equation 2 demonstrates how it worked.

$$weighed\_loss\_class[i] = loss[i] * class\_weights[i] \quad (2)$$

Table III illustrates the weights associated with each skin lesion category. In reference to table III and equation 2, this made the model penalize any Melanocytic nevi [nv] classification mistake by a factor of three.

VII. Results

All four CNN models were run 5 times and the average results from each run was taken. They were run on a machine which comprises of Nvidia GeForce GTX 960 - 4GB, 2x 8GB DDR3-1600 Memory RAM, and Intel Core i7-6700HQ 2.6 GHz quad-core processor.

A. Models Accuracy, Loss, Errors, and Time Taken for the Training and Validation Sets

Table IV shows the weighted training and validation sets accuracy, loss, errors and the total training time associated with each model. All models had a low bias, as the training set errors are all below 1% and close to the optimal Bayes error of 0%. In addition, all models had a low variance, as training and validation accuracies were within 1% of each other. This meant all models were neither underfitting or overfitting. Furthermore, TSM12 took the shortest to train in around 1,200 seconds, and ResNet152V2 took the longest to train in around 20,000 seconds. That was expected due to the huge number of layers difference between the models.

B. Models History

Figure 9 shows TSM12's training history against the number of epochs. That was an important visualization to ensure the model kept learning with every epoch, improving its accuracy and reducing its losses and errors, while trying to optimize the objective function. The other three models' improvements with every epoch were verified as well through this visualization. The figures can be viewed in the code.

TABLE IV: Models Training and Validation

		TSM12	InceptionV3	ResNet152V2	VGG16
Accuracy (%)	**Train**	78.62	76.17	82.15	89.16
	Val	77.84	75.67	82.69	89.80
Loss	**Train**	0.676	0.789	0.695	0.553
	Val	0.688	0.781	0.679	0.477
RMSE	**Train**	0.119	0.119	0.096	0.063
	Val	0.124	0.119	0.095	0.060
MSE	**Train**	0.0426	0.0495	0.0392	0.025
	Val	0.0437	0.0492	0.0380	0.024
Total Train Time (s)		1,224	6,631	20,264	7,853

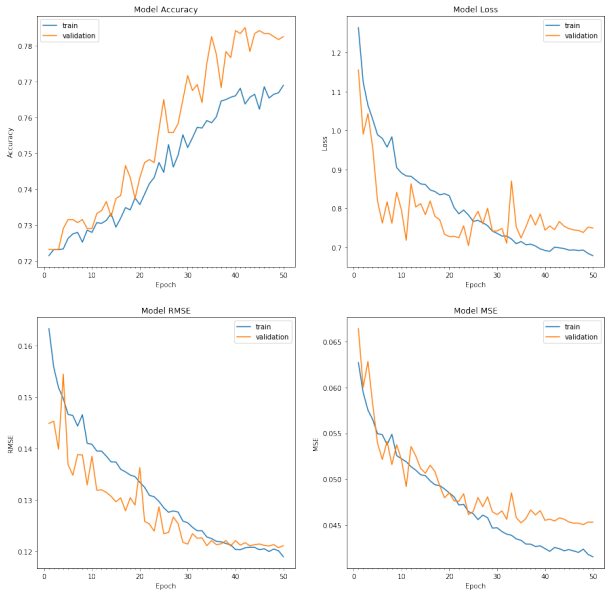

Fig. 9: TSM12 training history against epochs

C. Models Performance Comparisons for the Testing Set

Figure 10 shows the confusion matrices for true vs predicted labels for each model when run on the test set. The goal was to maximize the diagonal, which encompassed the true positives. To fully capture the values of the true and false positives and negatives for each model, the metrics discussed in section II were utilized. Table V shows the weighted testing sets accuracy, recall, precision, and f1-score associated with each model. Firstly, all of the testing accuracies were comparable to the validation accuracies, further solidating the fact that all models were neither underfitting or overfitting. VGG16 topped each metric score, scoring 88.54% accuracy, 87.76% precision, 88.74% recall, and 87.69% f1-score. VGG16 performance scores were followed by ResNet152V2, then followed by our TSM12, and lastly, by InceptionV3. Taking total training time into account however, TSM12's performance was very comparable to ResNet152V2. TSM12 trained in under 25 minutes, whereas ResNet152V2 took around 5 hours and 40 minutes, yet only achieving a 5% improvement in the performance metrics. Not only was TSM12 more time efficient than ResNet152V2, but the model's memory consumption was also much smaller, due to a more compact network architecture.

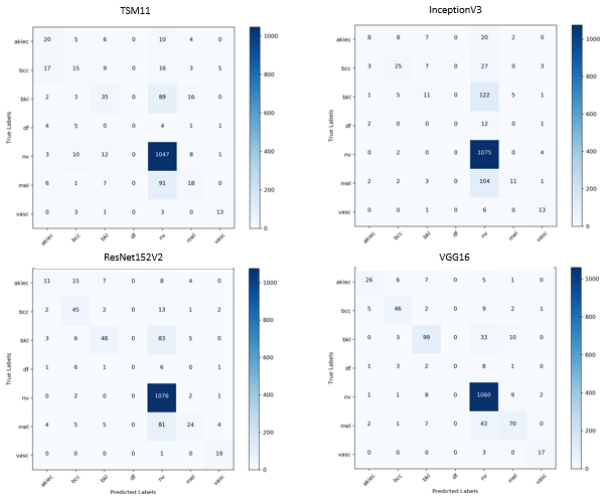

Fig. 10: Models Confusion Matrices

TABLE V: Models Performance on the Testing Set

Metrics	Model			
	TSM12	**InceptionV3**	**ResNet152V2**	**VGG16**
Accuracy (%)	77.86	77.53	82.14	88.54
Precision (%)	72.30	71.06	79.75	87.76
Recall (%)	77.91	77.44	82.61	88.74
F1-score (%)	73.72	70.58	78.16	87.69

D. Models Skin Lesion Classification Accuracy Plot

Figure 11 shows the weighted skin lesion classification accuracy plots for all seven skin lesion classes associated with each model. All four models excelled in melanocytic nevi [nv] classification, each achieving a class accuracy of over 85%. However, classification accuracy results for all the other classes varied. For example, all four models could not classify a single dermatofibroma [df] label correctly, achieving an accuracy of 0%. As for the malignant skin lesion categories melanoma [mel], basal cell carcinoma [bcc] and vascular lesions [vasc], the highest accuracy achieved for each class was around 55%, 70%, and 85% respectively, all by the VGG16 model. TSM12 performed poorly in classifying melanoma [mel] and basal cell carcinoma [bcc] classes, but relatively well with vascular lesions [vasc], producing a comparable accuracy of 70%. In general, nevertheless, it could be seen that the more training data available for the skin lesion category, the better the models were at predicting them during test time.

VIII. Conclusion

This work investigated the use of different CNN architectures to predict skin lesion categories based on skin lesion images. The data set was first pre-processed via data editing and cleaning, then split into the feature and target values, before applying feature normalization and data augmentation. We built our own basic CNN architecture called "TSM12" to be employed against the processed data set images to classify their respective skin lesion category, alongside InceptionV3, ResNet152V2, and VGG16, which were pre-trained CNN

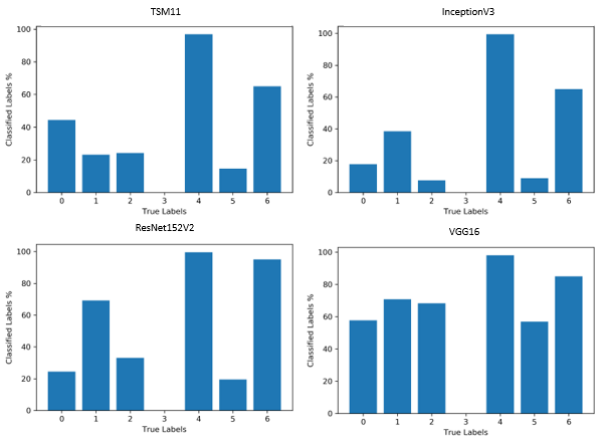

Fig. 11: Models Classification Accuracy Plots

models on the ImageNet data set. Hyperparameters were pre-selected and kept constant across all models. Weights were added to address the major class imbalance in the data set, by making the models more sensitive to Melanocytic nevi [nv] as it represented around two-thirds of the data. As for the results, VGG16 topped each metric score, scoring 88.54% accuracy, 87.76% precision, 88.74% recall, and 87.69% f1-score. Results also indicated a promising future for TSM12, as its performance was comparable to InceptionV3 and ResNet152V2 performance, without any hyperparameter tuning. However, according to the confusion matrices and classification accuracy plots in section VII, it could be seen that the more training data available for the skin lesion category, the better the models were at predicting them during test time. More malignant skin lesion images had to be provided for more accurate results and better malignancy classification, as the main aim in the medical field is to reduce the number of patients dying. Overall, this project demonstrated a huge potential for TSM12, as well as CNN's potential in the context of disease classification to lead further advances in the medical field with its image recognition abilities.

IX. Future Work

To improve upon these results, a couple of next steps are required:

- Explore other class imbalance techniques to improve the skin lesion category classifications for classes that are underrepresented.
- Utilize all the data in this data set, such as age and sex, via other machine learning techniques to extract useful information that can be combined with the work demonstrated in this paper.
- Explore other skin lesion data sets to obtain more training data to feed into our models, specially for the underrepresented and malignant skin lesion classes.
- Explore other pre-trained CNN models.
- Enhance TSM12's architecture and implementation, by further tuning hyperparameters, such as the number of layers, the type of layers and the layers' hyperparameter values.
- Potentially, with the correct computational resources, optimize all the possible hyperparameters in each network.

Appendix A
Group Members Contribution

Each member contributed equally to all parts of the project. This includes the literature research and survey, data set exploration, data pre-processing, models building and implementations, and result comparisons. Meetings were conducted every week via zoom to discuss our project progress.

Acknowledgment

The authors would like to thank Dr. Abdallah Shami, and TA's Dimitrios Manias and Anas Saci for providing us the platform to conduct a project of such magnitude and for all their guidance and support.

References

[1] P. Canada, "Skin cancer - Canada.ca", Canada.ca, 2020. [Online]. Available: https://www.canada.ca/en/public-health/services/sun-safety/skin-cancer.html.

[2] Artificial intelligence in medicine:Machine learning." [Online]. Available: https://www.ibm.com/watson-health/learn/artificial-intelligence-medicine

[3] 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, 2018. Available: 10.1038/sdata.2018.161.

[4] "A Comprehensive Guide to Convolutional Neural Networks—the ELI5 way", Medium, 2020. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[5] 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), Las Vegas, NV, 2016, pp. 2818-2826.

[6] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge", International Journal of Computer Vision, vol. 115, no. 3, pp. 211-252, 2015. Available: 10.1007/s11263-015-0816-y.

[7] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.

[8] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," CoRR, vol. abs/1409.1556, 2014.

[9] R. White, D. Rigel and R. Friedman, "Computer Applications in the Diagnosis and Prognosis of Malignant Melanoma", Dermatologic Clinics, vol. 9, no. 4, pp. 695-702, 1991. Available: 10.1016/s0733-8635(18)30374-7.

[10] M. Binder, A. Steiner, M. Schwarz, S. Knollmayer, K. Wolff and H. Pehamberger, "Application of an artificial neural network in epiluminescence microscopy pattern analysis of pigmented skin lesions: a pilot study", British Journal of Dermatology, vol. 130, no. 4, pp. 460-465, 1994. Available: 10.1111/j.1365-2133.1994.tb03378.x.

[11] N. Codella et al., "Deep learning ensembles for melanoma recognition in dermoscopy images", IBM Journal of Research and Development, vol. 61, no. 45, pp. 5:1-5:15, 2017. Available: 10.1147/jrd.2017.2708299.

[12] U. E. Yildiz and V. Kiliç, "Detection of Melanoma with Multiple Machine Learning Classifiers in Dermoscopy Images," 2019 Medical Technologies Congress (TIPTEKNO), Izmir, Turkey, 2019, pp. 1-4.

[13] N. Kinyanjui et al., "Estimating Skin Tone and Effects on Classification Performance in Dermatology Datasets," NeurIPS 2019 Workshop on Fair ML for Health, 2019. Available: arXiv:1910.13268.

[14] N. Gessert et al., "Skin Lesion Classification Using CNNs With Patch-Based Attention and Diagnosis-Guided Loss Weighting," in IEEE Transactions on Biomedical Engineering, vol. 67, no. 2, pp. 495-503, Feb. 2020.

[15] D. P. Kingma and J. Ba, "Adam: A Method For Stochastic Optimization". 3rd International Conference for Learning Representations, San Diego, 2015. Available: arxiv.org/abs/1412.6980.

Tareq Tayeh received the B.E.Sc degree in Software Engineering from Western University, London, Canada, in 2018. He is currently pursuing the M.E.Sc degree in Software Engineering, with the Vector Institute Accredited collaborative specialization in Artificial Intelligence, with the Department of Electrical and Computer Engineering, Western University, London, Canada. He worked for IBM, Markham, Canada as a QA student intern between May 2016 and August 2017, and as a full time DevOps Developer between May 2018 and September 2019. His current research interests are in the areas of machine learning, automated AI, data analysis, and cloud computing.

Sulaiman A. Aburakhia received the B.S. degree with honors in Electrical Engineering from October 6 University, Egypt, in 2003, and M.S. degree in Electronics and Communications Engineering from the Arab Academy for Science and Technology and Maritime Transport, Alexandria, Egypt, in 2007. Following his M.S., he worked at several consulting firms and vendors of radio networks for more than 10 years. His expertise lies in the fields of network rollout, performance analysis, and radio access optimization. He is currently pursuing a PhD degree in Software Engineering in the Department of Electrical and Computer Engineering, Western University, London, Canada. His research interests include machine learning, data analysis, mobile radio, and digital communications.

Moataz ElKhishen received his B.Eng Honours degree in Computer and Hardware and Software Engineering from Coventry University, Coventry, England in 2019. He is currently working towards the M.Eng Degree in Software Engineering at Western University, Ontario, Canada. His research interests include Processor Design, Artificial Intelligence and Machine Learning.