# Convolutional Neural Networks: Models, Parameters and the ISIC Dataset

Sophie Bell
*Department of Mathematics and Computing*
*Manchester Metropolitan University*
Manchester, United Kingdom
sophie.bell8@manchester.mmu.ac.uk

***Abstract*—This paper examines the training of multiple convolutional neural networks applied to the ISIC dataset. A literature review will be presented on various parameters within convolutional neural networks and three key network architectures. Models will be created and results compared to identify what parameters are key. Some drawbacks are considered with this method and the issue of class imbalance and small datasets are highlighted in the results of the models.**

***Keywords—convolutional neural network, optimization, parameters, deep learning algorithms, VGG16, residual network, InceptionV3, ResNet50, ISIC, melanoma, skin lesion, image classification***

## I. Introduction

Deep learning algorithms have resulted in breakthroughs in image classification and have been applied to multiple research areas, such as medical, botany, fashion and other general areas of research [1]. Models have been applied to dermatology for some years, particularly in the region of melanoma detection and classification. There are models being used in practice currently that have been found to increase the correct classification of skin cancer, compared to just a medical professional alone [2]

Skin diseases contributed to approximately 1.79% of the global burden of diseases measured in disability-adjusted life years (DALYs). In the United Kingdom, 60% of the population suffer from skin diseases during their lifetime. Skin diseases may be cancerous, inflammatory or infectious and affect people of all ages, especially the elderly and young children. The deep learning approach is powered by computational advances and has shown exceptional performance in object recognition and classification [1]. Given this, it is important to diagnose and treat appropriately all varieties of skin disease. Accurately identifying skin lesions without the need for a counsel of medical professionals could improve health globally and have knock on effects in other areas of healthcare.

This research is conducted on the ISIC 2019 dataset (discussed in detail in section V). There has been a multitude of research conducted on the ISIC dataset and on other skin lesion datasets in the deep learning research field, a summary of these is also included in [3]. As deep learning and transfer learning become more researched, more studies and challenges arise in skin lesion classification. In relation to the ISIC dataset and dermatology, there are numerous approaches that fall into two general types; the first diagnosing melanoma or non-melanoma, the second, classifying images into different types of skin disease. This research will approach the latter and attempt to classify nine types of skin lesions using a deep convolutional network.

This research aims to contribute to the research already conducted on the ISIC dataset and examine how different deep learning models perform on the same dataset. Firstly, an introduction to deep learning will be given, this will be followed by a detailed analysis of two commonly used models (ResNet50 and InceptionV3/VGG-16) and their applications in other research, as well as instances using the same dataset as the current research. Thirdly, there will be a discussion on parameters in deep learning models and how each of these can impact the model and their accuracy in image classification. Following the literature review, the method and results will be discussed from the current papers experimentation, describing the parameters used and the accuracy of each model. Finally, a conclusion will be given on each of the models with comparisons made and recommendations for future models and research.

The primary focus of this paper is to investigate ways of training deep learning models and how to tune these models for increased accuracy. The deep learning experimentation includes a description of how the images were loaded, the pre-processing steps and the deep learning model fine-tuning completed.

Image classification is defined as categorizing images into one of several predefined classes [4]. This ISIC dataset contains nine classes, and each image is assigned to one class only. The aim of the model is to correctly identify the class the image belongs to and be able to classify unknown/test images into the correct class.

The dataset will be first trained using the ResNet50 [5] model using transfer learning and weighted using ImageNet. The same process will then be used with the VGG-16 [6] model and InceptionV3 [7], using the ImageNet dataset to train the model. Finally, a separate model will be presented that has been built specifically for the dataset and not pre-trained on another image set.

## II. Literature review

### A. Convolutional Neural Networks

Image classification is a complex process that may be affected by many factors [8] as deep learning research advances, more advances are made in image classification challenges.

Convolutional neural networks are a type of artificial neural network that use multiple perceptrons [9]. They have evolved in recent years, ranging from LeNet [10] to

AlexNet [11] to GoogleNet [12] and ResNet [13]. Although there are many models, in general they consist of convolutional and pooling layers which are grouped together [4] these are then stacked on top of each other producing deeper networks. The final layer outputs the class label.

In a convolutional layer, a kernel (of user specified size) is passed over an image to create a feature map which is passed to the next layer. Convolution is performed by passing a filter (kernel) across the image, where at each location a matrix multiplication is done, and the sum entered on the feature map. After each convolutional layer is a non-linear activation function, the purpose of which is to introduce non-linearities into the network, helping the model deal with non-linear data (as most data is) [1]. Following the convolutional layer and activation function is a pooling layer. There are many types of pooling (explored in a later section), that all aim to ensure main features of images are not lost when down sampling due to filtering.

Finally, there are fully connected layers where each of the nodes in the first layer is fully connected to each node in the next layer.

In this research, two different models are used (InceptionV3 based on GoogleNet and ResNet50, a residual network), followed by a model built from scratch. Each model is applied to the same dataset to allow a comparison of different numbers of layers and differing parameters. Models are judged on their ability to produce an accuracy score on test data, the higher the accuracy on unseen data, the better the model.

## III. Related Work

As previously discussed, skin cancer is a worldwide epidemic that requires an expert medical professional to diagnose which can result in delays in diagnosis and treatment. Due to this, many researchers have proposed methods that utilize deep learning models for the classification of images of skin cancer. There are various approaches to classifying images and a multitude of challenges arising from real-world datasets. The following will evaluate different methods of deep learning applied to different datasets and discuss the approaches used and their outcomes. Firstly, an overview of transfer learning will be given as this is the primary method used in image classification and research has shown one of the most accurate methods.

### *A. Transfer Learning*

Transfer learning is a technique where a model trained for one task is re-trained to perform a second tasl [14]. In deep learning, transfer learning is a technique whereby a neural network model is first trained on a problem similar to the problem that is being solved. One or more layers from the trained model are then used in a new model trained on the problem of interest [1]. The two pre-trained models in this research use transfer learning methods. There are multiple benefits to transfer learning, including faster training times, less computational power required and higher asymptote where the converged skill of the model is better than it would be otherwise [15]. Transfer learning has proved effective for classification tasks.

### *B. ResNet50*

The first model used in this research is ResNet50. Research often states that adding more layers into a neural network improved accuracy and performance, however, this produced large training errors not thought to be caused by overfitting. Residual Networks (ResNet) arose as a response to needing more complex models that did not necessitate more layers which lead to other issues. Initially, the ResNet models were applied to image recognition tasks, but have since been used for non-image tasks.

The key feature of residual networks is Skip Connections, they solve an issue in deep learning of vanishing gradient and allow a model to learn an identity function that ensures the higher layer will perform as good as the lower layer, the benefits of which include no additional parameters and computational time being kept to a reasonable amount. The model contains many repeated steps with kernel sizes varying from 1*1 to 7*7. It is possible to load a pretrained version of the model using the ImageNet dataset. The ImageNet dataset contains over one million images across one thousand categories. The model learns rich feature representation for a wide range of images. The default image size required is 224*224 pixels.

| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer | 152-layer |
|---|---|---|---|---|---|---|
| conv1 | 112×112 | 7×7, 64, stride 2 | | | | |
| | | 3×3 max pool, stride 2 | | | | |
| conv2_x | 56×56 | [3×3, 64; 3×3, 64] ×2 | [3×3, 64; 3×3, 64] ×3 | [1×1, 64; 3×3, 64; 1×1, 256] ×3 | [1×1, 64; 3×3, 64; 1×1, 256] ×3 | [1×1, 64; 3×3, 64; 1×1, 256] ×3 |
| conv3_x | 28×28 | [3×3, 128; 3×3, 128] ×2 | [3×3, 128; 3×3, 128] ×4 | [1×1, 128; 3×3, 128; 1×1, 512] ×4 | [1×1, 128; 3×3, 128; 1×1, 512] ×4 | [1×1, 128; 3×3, 128; 1×1, 512] ×8 |
| conv4_x | 14×14 | [3×3, 256; 3×3, 256] ×2 | [3×3, 256; 3×3, 256] ×6 | [1×1, 256; 3×3, 256; 1×1, 1024] ×6 | [1×1, 256; 3×3, 256; 1×1, 1024] ×23 | [1×1, 256; 3×3, 256; 1×1, 1024] ×36 |
| conv5_x | 7×7 | [3×3, 512; 3×3, 512] ×2 | [3×3, 512; 3×3, 512] ×3 | [1×1, 512; 3×3, 512; 1×1, 2048] ×3 | [1×1, 512; 3×3, 512; 1×1, 2048] ×3 | [1×1, 512; 3×3, 512; 1×1, 2048] ×3 |
| | 1×1 | average pool, 1000-d fc, softmax | | | | |
| FLOPs | | $1.8\times10^9$ | $3.6\times10^9$ | $3.8\times10^9$ | $7.6\times10^9$ | $11.3\times10^9$ |

*Table 1 – ResNet50 Architecture*

The architecture of ResNet50 is shown in the table (Table 1). It shows that the first layers use a larger filter (kernel) size which is reduced through each layer. The model also shows that initially maximum pooling is used which is then changed to average pooling following the fully connected layer, where the filter size is also the smallest. In total there are 50 layers, excluding the activation functions and the maximum/average pooling layers.

In a study using ResNet50 for image classification [16], it was suggested that using ResNet50 to extract features could be followed by a deep forest to better classify images [16]. The ResNet50 model was trained on the ImageNet dataset. The study found that the forest did not give results expected and shown in other studies, the authors attributed this to the ResNet50 model not extracting features that were an accurate representation of the images in the dataset. They advised fine tuning the ResNet50 model by training the last few layers of the model with a lower learning rate.

[17] used a ResNet50 model to classify plant diseases. A stochastic gradient descent (SGD) optimizer was used and SoftMax activation. The study showed that for the same

dataset, other models (VGG16 and AlexNet) required more time to provide target accuracy than ResNet50, they also presented results showing that the ResNet50 model trained in their study was more accurate than AlexNet and GoogleNet models designed in previous studies on the same dataset. The model they presented also used less epochs than other studies, the authors attributed the higher accuracy in part to a larger dataset than previous studies.

### C. InceptionV3

The second transfer learning model in this research is InceptionV3 based on GoogleNet. InceptionV3 is also a transfer learning model. In practice, this means a pre-trained model can be transferred to implement a similar task by learning new data distribution and fine-tuning parameters across all layers of the model [14]. InceptionV3 has been a popular choice for medical image analysis in current research and has been applied successfully to skin lesion image tasks [18].

The InceptionV3 model includes three parts: the basic convolution block, improved Inception model (from previous models) and the classifier [14]. The InceptionV3 architecture can be seen in table 2. The first part alternates convolution with max-pooling layers. In InceptionV3 a 1x1 kernel is often used to reduce the number of feature channels and accelerate training speed. The InceptionV3 model has 48 layers in total.

| **type** | **patch size/stride** or remarks | **input size** |
|---|---|---|
| conv | 3×3/2 | 299×299×3 |
| conv | 3×3/1 | 149×149×32 |
| conv padded | 3×3/1 | 147×147×32 |
| pool | 3×3/2 | 147×147×64 |
| conv | 3×3/1 | 73×73×64 |
| conv | 3×3/2 | 71×71×80 |
| conv | 3×3/1 | 35×35×192 |
| 3×Inception | As in figure 5 | 35×35×288 |
| 5×Inception | As in figure 6 | 17×17×768 |
| 2×Inception | As in figure 7 | 8×8×1280 |
| pool | 8 × 8 | 8 × 8 × 2048 |
| linear | logits | 1 × 1 × 2048 |
| softmax | classifier | 1 × 1 × 1000 |

*Table 2 – Inception V3 Architecture*

In a study on skin lesion image classification, InceptionV3 was used [19]. The researchers used the pre-trained InceptionV3 model using ImageNet weights, the final layer was trained using a flower classification dataset. This method was used to decrease the amount of time spent training the model. As described earlier, this method allows the model to use features extracted from general images that are broader and then narrow down the feature selection to be more appropriate to the dataset, in this case flowers [19]. This research shows training accuracy to reach 100% and validation accuracy around 98%, the validation accuracy also improved in an experiment when a larger dataset was used [19], this is higher than research conducted on the same dataset with other models.

Other research has stated that the complexity of Inception models make it difficult to make changes to the network and scaling can result in a loss of computational gains[20], contrary to this [21] stated in their research that Inception V3 worked best.

### D. VGG-16

The final model for discussion is also a transfer learning model – VGG16 [22]. VGG16 was seen as an upgrade to AlexNet and designed in a way that convolutions would be simpler, by replacing large filters (kernels) with a 3*3 filter while padding to maintain the same size as AlexNet and using a maximum pooling layer to downsample the image size. There are 16 layers in total.

The VGG-16 architecture is shown in table 3. The approach for fine tuning is to extract all the layers of the network except the last three. Shift the layers to the new classification task by substituting the last three layers with fc layer, SoftMax layer and classification output layer.

| ConvNet Configuration | | | | | |
|---|---|---|---|---|---|
| A | A-LRN | B | C | D | E |
| 11 weight layers | 11 weight layers | 13 weight layers | 16 weight layers | 16 weight layers | 19 weight layers |
| input (224 × 224 RGB image) | | | | | |
| conv3-64 | conv3-64 **LRN** | conv3-64 **conv3-64** | conv3-64 conv3-64 | conv3-64 conv3-64 | conv3-64 conv3-64 |
| maxpool | | | | | |
| conv3-128 | conv3-128 | conv3-128 **conv3-128** | conv3-128 conv3-128 | conv3-128 conv3-128 | conv3-128 conv3-128 |
| maxpool | | | | | |
| conv3-256 conv3-256 | conv3-256 conv3-256 | conv3-256 conv3-256 | conv3-256 conv3-256 **conv1-256** | conv3-256 conv3-256 **conv3-256** | conv3-256 conv3-256 conv3-256 **conv3-256** |
| maxpool | | | | | |
| conv3-512 conv3-512 | conv3-512 conv3-512 | conv3-512 conv3-512 | conv3-512 conv3-512 **conv1-512** | conv3-512 conv3-512 **conv3-512** | conv3-512 conv3-512 conv3-512 **conv3-512** |
| maxpool | | | | | |
| conv3-512 conv3-512 | conv3-512 conv3-512 | conv3-512 conv3-512 | conv3-512 conv3-512 **conv1-512** | conv3-512 conv3-512 **conv3-512** | conv3-512 conv3-512 conv3-512 **conv3-512** |
| maxpool | | | | | |
| FC-4096 | | | | | |
| FC-4096 | | | | | |
| FC-1000 | | | | | |
| soft-max | | | | | |

*Table 3 – VGG16 Architecture*

The VGG16 model has been applied to a variety of research areas and has been shown to have performed the best in a recent study on brain image classification [23]. VGG16 has been shown to generalise well to datasets other than ImageNet where it performed best in 2014 [24]. In a recent study, a thyroid disease dataset was used to compare the performance of VGG16 and InceptionV3 [25]. VGG-16 outperformed the InceptionV3 model in this research. The researchers state that the VGG16 model is simpler, and this contributes to the generalizability. Although it performed better in this study, there is also research arguing that the computational cost of Inception is much lower than VGGNet or its higher performing successors [20].

In another study [24], aVGG16 model was used for classifying skin lesions, the aim was to classify as benign or malignant. In their study, the researchers described multiple implementations of VGG16, the most similar method to the current research was to train the VGG16 model on the ImageNet dataset and train only the fully connected layers with the ISIC dataset. The model achieved an accuracy of 95.95%, although when all measures were combined, it did not perform as well as a different method cited.

## IV. Parameters and Tuning

### A. Activation Functions

As previously outlined, activation functions are included in all convolutional neural networks. According to the research, there is no clear-cut solution for which activation function to apply in each setting, trial and error is the most effective solution [4].

The ReLU [26] is a piece wise linear function. ReLU retains only the positive part of the activation by reducing the negative part to zero. ReLU is commonly used in image classification systems [4]. In some research, the activation functions are tested, [27] found ReLU to perform the best and recommended it over Tanh and Sigmoid when training deep neural networks.

An alternative to ReLU and often utilized in the fully connected layer is SoftMax. SoftMax is largely used because of its simplicity and probabilistic interpretation.

### B. Epochs and Batch Size

Epochs are the number times that the learning algorithm will work through the entire training dataset. Given that this is the number of times the model goes through the data, increasing this increases the model running time.

It is generally suggested that the number of epochs is decided based on the complexity of the dataset. If the model is still improving after the epochs have completed, the number of epochs should be increased. Overfitting is a risk if too many epochs are used as the model becomes too sensitive to the training data. A measure to counteract this is Early Stopping which stops training when an increase loss is detected or a decrease in accuracy values.

Batch size describes the number of images to work through before updating the internal model parameters. [28] conducted research on batch size and learning rate, they found instead of decreasing learning rate, the batch size should be increased during training. They surmised that this strategy achieves near-identical performance on the test set with the same number of epochs, but with significantly fewer parameter updates.

### C. Optmisers and Learning Rate

Optimizers are algorithms or methods that minimize an error function (loss) or maximize efficiency of production. There are a number of optimizers that can be used in research, for brevity this section will describe the two used in the experimental stage.

Stochastic Gradient Descent (SGD) is one of the foremost approximation techniques. The SGD randomly selects the next set of examples that will update the trainable parameters thereby improving the training speed. The advantage of SGD is that it performs better than the adaptive optimizers at very prolonged training time for effective hyperparameter tuning. Disadvantages are that there is no adaptive way for finding the optimal learning rate for the training process and the SGD has the gradients tending to zero at some point. It does not scale well with large datasets [29]. The other commonly used and cited optimizer is Adam. It is a method that computes adaptive learning rates for each parameter. It stores both the decaying average of the past gradients, similar to momentum and also the decaying average of the past squared gradients. Adam is well suited for large scale parameters and data as well as offering improved computational efficiency, improved memory requirement and invariant to the diagonal rescaling of gradients. Most used optimizer in recent DL model developments and has been used across diverse applications [29].

Learning rate is a parameter that provides the model a scale of how much model weights should be updated. There is no way to choose a learning rate based on a calculation, it is done by systematic experimentation. In recent research [30] argued that a good learning rate could be estimated by training the model with a very low learning rate and increasing at each iteration.

### D. Pooling

A pooling function replaces the output of the net with a summary statistic of nearby outputs [1].

Pooling is an important step in convolutional networks, it is used to reduce dimensionality [4], which enables the number of parameters to be reduce, in turn reducing training time and combatting overfitting. Research [31] also stated that if the coding step has not been designed so the pooling operation preserves as much information as possible, then the pooling step itself should be more selective. The two most used types of pooling are max and average. Max pooling produces the maximum pixel value over the non-overlapping region of the weighted window [14]. The performance of either max or average pooling is dependent on the data and its features, and that for a classification problem using either strategy alone may not be optimal [4].

### E. Regularisation

Dropout is a form of regularization used in deep learning models. when each training case was presented to the network during the training phase, each hidden neutron was randomly omitted from the network with a probability of 0.5. Therefore, hidden neutrons could not rely on other hidden neutrons being present. This helps tackle the issue of overfitting. The primary benefit of Dropout is its proven ability to significantly reduce overfitting by effectively preventing feature coadaptation. According to [27] dropout should be applied in hidden and input layers. Dropout drops nodes during forward propagation and reduces the interdependent learning among the nodes, forcing the network to learn better robust features independently [32] Other forms of regularization include label smoothing regularization (LSR) and data augmentation.

### F. Other Factors

Unbalanced datasets exist naturally in real world data and have become one of the greatest challenges in deep learning [33]. There are many cited ways in which to handle a class imbalance in a dataset. One commonly used method is oversampling, which involves generating extra images in the smaller classes using data augmentation. Other research has used data augmentation to increase the dataset and found the results were significantly worse [18]. The other commonly cited method is undersampling which is removing images from the larger classes to balance them with the smallest. Other instances of using the ISIC data set [34] used random

downsampling, taking images from the largest class as did [35]. A study [36] compared over sampling, balanced batch sampling and class-specific loss weighting. The research showed that class balancing significantly improves the mean sensitivity. Generally, there is no preferred way to handle imbalance and authors has argued that there is not enough research on class imbalance [37].

In summary, there are many hyperparameters that need to be tuned during the training of a deep learning model, the majority of which rely on experience and knowledge to train and have little in the way of theoretical guidance. In the current research, hyperparameters were tested and the most successful/accurate were kept and reported on.

## V. ISIC DATASET

For this paper, the ISIC 2019 dataset was used. The data was downloaded from Kaggle [38]. The dataset contains a mix of images showing 9 different classes of skin cancer (shown below). The data is pre-partitioned into 2239 test images and 118 training images. The height and width of each image was set as a resolution of 224x224 for analysis. In other deep learning research, this dataset has been used in demonstrations of multi-class classification and adapted to binary classification in some instances (melanoma or non-melanoma). The dataset is widely known due to competitions being based on the dataset using deep learning models to classify skin lesions. In this paper, the dataset will be used to classify types of skin cancer (9 classes), however the primary focus of this paper is the different types of deep learning model and how each performs with different parameters. This data set has been chosen for this task as it is widely available for replication, well researched due to various competitions and has a pre-defined set of training and test images. It would also be possible to use additional

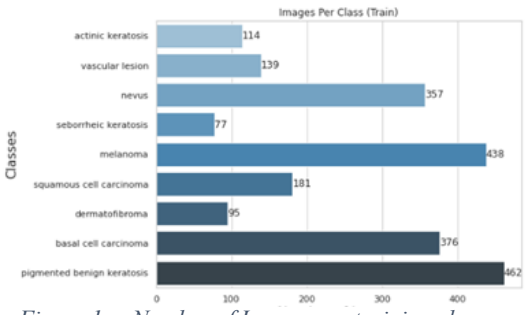


*Figure 1 – Number of Images per training class*

ISIC data from different years to build upon this research. There are some drawbacks to this dataset when creating convolutional neural networks, such as the number of images in the dataset. There are few images in the data set (2239 test, 118 training) which can create challenges with overfitting in models, however, real-world datasets are often lacking in numbers, particularly those in medical fields as some diseases are considered rare and are therefore less photographed.

The number of images in each class is not consistent, creating an imbalance issue, this is an important observation as this imbalance can affect the deep learning models [39] [40]. This imbalance is shown in the training data set in figure 1. Within the test dataset there is also an imbalance (shown in figure 2), there are 16 images for 7 of the classes,

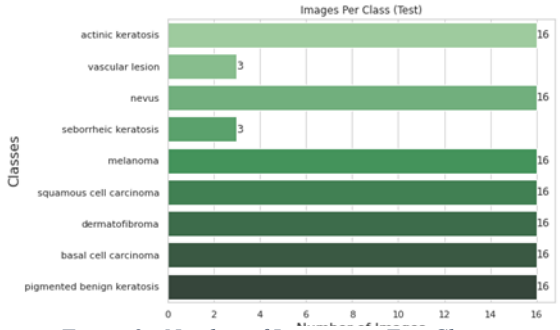


*Figure 2 - Number of Images per Test Class*

but only 3 images for the remaining 2 classes. The implication class imbalance can have on the deep learning model has been discussed earlier in this paper, recommendations for how to deal with this imbalance have also been discussed and will be reviewed further in the method and conclusion sections.

## VI. METHODOLOGY

The aim of this research was to test pre-trained models and design one convolutional neural network to classify images from the ISIC dataset into 9 images. The method used for each will be detailed first followed by a description of each method. All research was conducted on Google Colab using a Tesla T4. Keras [41] a deep learning framework for Python was utilized to implement architectures alongside TensorFlow. The models were trained using 2239 training images on 50 epochs. Following training, the model was tested on 118 images. The validation split used when creating the image generator was 0.3. Table 3 shows the general settings used in all 4 methods. Based on research, transfer learning with weights from ImageNet produce the highest accuracy in classification models, the epochs were selected after trying lower numbers, 50 was the most realistic given timeframe and computational power.

**Table 3: Hyperparameters Used Throughout**

| *Parameter* | *Number* |
|---|---|
| Epochs | 50 |
| Batch Size | 32 |
| Learning Rate | 0.0001 |

### *A. Data Augmentation*

Although there were few images in the dataset with some major imbalance, no data augmentation was completed. The experiments show how each of the models handles imbalanced data. Class weights were created, although when implemented made the models worse, each method will show the results with and without class weights. Class

weights were created using the compute class weight method and implemented when the model was compiled.

### B. ResNet50

The aim of this method was to use the transfer learning methodology applied to the ResNet50 architecture and tune parameters to achieve the best accuracy for the ISIC dataset. The architecture of ResNet50 was used, trained on the ImageNet dataset. To begin, only the first layer was frozen. The loss was measured using categorical crossentropy and the metric used was accuracy. The Adam optimizer was always used for ResNet50.

### C. VGG16

The aim of this method was also to use transfer learning with a different architecture to evaluate the difference in the number of training parameters and how different parameter changes affect the accuracy of the model. The architecture of VGG16 was used, trained on the ImageNet dataset. Initially, the first 15 layers were frozen. Following the last layer, a Flatten layer was added, followed by a Dense layer, then a Dropout and the final dense layer with SoftMax activation. The VGG16 was compiled using a SGD optimizer, the metric was accuracy and the loss was categorical crossentropy.

### D. InceptionV3

Further to the above, an additional transfer learning model was included to further evaluate the impact of the number of layers in a model and the parameter changes have on the accuracy of a model. The InceptionV3 model was used, trained on the ImageNet dataset. All layers were made not trainable. Two dropout (0.6, 0.8) layers were added alongside a flatten layer and an additional layer. The fully connected layer used SoftMax activation. The model was compiled with an SGD optimizer, metric of accuracy and loss of categorical crossentropy.

### E. Model 4

Finally, a model was created from scratch with no weights, designed specifically for the ISIC challenge presented earlier (multi-class classification). The model was compiled using an Adam optimizer, loss of categorical cross entropy and a metric of accuracy. This model will be referred to as Model 4 henceforth. Model 4 started with a larger filter size (5*5), followed by a 1*1 filter, this then changed to 3*3 filter for the rest of the convolutional layers. The number of filters began at 32 and increased through to 256 before the final dense layer where the number of filters was 512. ReLU was used as the activation function for all layers, excluding the fully connected layer which utilized SoftMax.

## VII. Results

The highest performing network in terms of accuracy was InceptionV3 with an accuracy of 27.12% on 118 test images across 9 classes. This was around 6% better than the second-best, Model 4 that had 21.19% accuracy.

Table 4 summarizes the results of all experiments and shows the difference in the models when using class weights.

| Model | Accuracy from Experimentation | | |
|---|---|---|---|
| | ***Weights Accuracy*** | ***No Weights Accuracy*** | ***Total Number of Parameters*** |
| VGG16 | 17.80% | 20.34% | 27,564,873 |
| InceptionV3 | 13.56% | 27.12% | 21,802,784 |
| Model 4 | 16.95% | 21.19% | 3,696,649 |

The table above shows the performances of all models. Interestingly, some performed better than others with class weights but not better overall when weights were removed. InceptionV3 performed the worst of all three when weights were included and the best when weights were not included. Similarly, VGG16 performed the best with weights but the worst when weights were removed.

The number of parameters is defined by the number of layers in a network, number of units in every layer and dimensionality of the input and output. A noticeable difference between the models is the number of parameters, VGG16 has the most, showed no signs of overfitting and performed the worst overall, which is unexpected. Model 4 has the least parameters and was only 6% less accurate than InceptionV3 that had almost seven times more parameters.

### A. Inception V3 Results

Following editing parameters, a chart was created based on the InceptionV3 model.

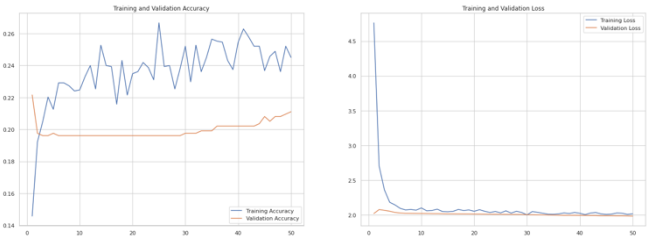


*Chart 1 - Training and Validation Accuracy/Loss – InceptionV3*

Chart 1 shows the training and validation accuracy and loss. Accuracy gives the percentage of instances that are correctly classified.

The left chart shows the accuracy, the blue line shows that during training there was peaks and drops across all epochs whereas the validation accuracy is lower, yet around the same level. As seen in the chart, the lines do not follow the same general pattern, this could be a sign of underfitting.

Loss is the distance between the ground truth and the predictions. Predicted classes are based on probability. The loss is therefore also based on probability. In classification, the neural network minimizes the likelihood to assign a low probability to the actual class.

The loss chart (right side) shows that there was a large loss in first five epochs on the training data, however the loss remained constant throughout, without a drop on the validation data. The loss chart would suggest a good fit to the data as the lines are extremely close together and follow the same pattern until the end of training and validation.

### B. *Model 4 Results*

As shown in table 4, Model 4 had the second highest accuracy when weights were used and the second highest when weights were included. Although the accuracy was higher, it is suspected there is some underfitting on the model after examining the charts.

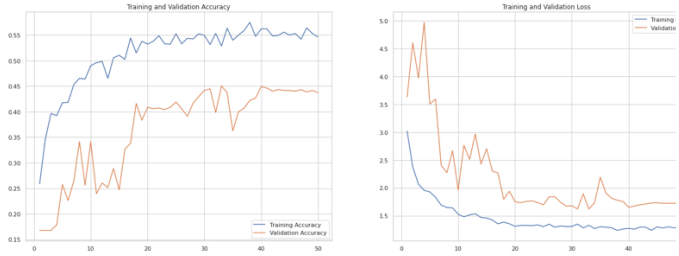


*Chart 2 - Accuracy and Loss - Model 4*

Chart 2 shows the training and validation accuracy and loss. The accuracy shows a steady incline on training data and a plateau towards the end around 55% accuracy, whereas there are a lot of steep changes on the validation accuracy line (orange). A similar but opposite pattern can be observed on the loss chart, which shows a sharp decline on both lines before a plateau after around 40 epochs.

It is suspected that this is because the dataset was too complicated for Model 4 to learn anything meaningful about. Given the large gap between the lines on the loss chart, it could be that the dataset was unrepresentative of the problem. This may be attributed to the imbalance in the dataset with no weights to compensate.

A confusion matrix was created to further show where predictions are being made and how many are correct. Figure 4 shows that while some correct predictions were made there was an issue with predicting two classes (squamous cell carcinoma and vascular lesion). Interestingly, these are two classes with the lowest amount of training images and vascular lesion had only three test images.

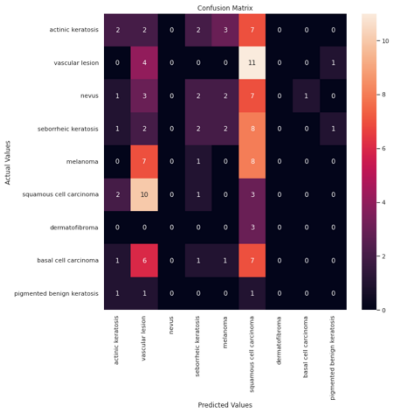


*Figure 4 – Confusion Matrix - Model 4*

Overall the results show that the models did predict some categories and make some predictions, but the accuracy and charts suggest the models were not a good fit for the data. A potential remedy would be further fine tuning of the models, increasing complexity (layers) on Model 4 and increasing the amount of data used for training and validation.

## VIII. Conclusion and Research Recommendations

Even though the accuracy of the tested models was not as high as demonstrated in other research, this could be due to multiple factors. Firstly, no data augmentation was completed to create more test or training data which is often conducted in other research to solve class imbalance issues and increase the number of images in training and test datasets. The algorithms tested here are generally suited to this dataset, as well as multi-class classification and have been shown to have good accuracy in other research, but usually when there is more data or some data augmentation. Therefore, it can be concluded that the algorithms are suited to this dataset but with some caveats. The research has shown that attention should be paid to class imbalance and steps taken to balance classes before classification. In this research, the classes in the dataset were imbalanced and the accuracy low. This is to be expected with a small training and test set as well as no steps taken to resolve the imbalance. Where efforts were made to improve the imbalance problem (class weighting) this had a negative effect on the models, this could be due to class weights being implemented at the model level, rather than data level. Or because the layers were frozen at an inappropriate level when class weights were being used.

Transfer learning has been cited as the best way to train models with lower computational power and shorter training times. In this research, three transfer learning models were trained on ImageNet. The transfer learning model InceptionV3 did perform the best overall (with no class weights), however it's accuracy was not much larger than a model with significantly less layers, parameters and no pre-training on ImageNet. The main reason for this is suspected to be the training data used as it was a small set and imbalanced. Overall, this research has contributed to the learning on parameters in convolutional neural networks. Although the models did not show great performance, the study has highlighted the importance of parameters and fine tuning, as well as dataset selection.

Recommendations for future research have been identified throughout this paper, these will now be outlined. When training the first three models, the weights used were ImageNet, although this is a large dataset where some information can be transferred for learning, it is also not the most appropriate dataset for the ISIC data. Researchers have identified a dataset "DermNet", which is similarly used to ImageNet but contains a database of skin lesion images [24]. Similarly, it is recommended that research focuses on building a large dataset of skin lesion images. For a classification system to be implement in the real world to assist diagnosis it is important that the models have enough training and test data to improve accuracy without overfitting. The current models presented in this research would not be appropriate due to reasons outlined in the results section and the risk of misdiagnosis or images not being classified correctly would be too great.

The literature review of this paper discussed some pertinent parameters that are often changed during experimentation, although researchers often state these changes in their publications there is currently no guidance or research that highlights best practice with different types of datasets. Also, a recommendation would be to

identify the interaction between different activations and the effect these have on regularization methods such as dropout [4].

The research presented in this paper highlights further the issue of using an imbalanced dataset. Further information needs to be gained on class balancing and the most effective way to do this without overfitting or underfitting models. Although some research has been conducted there is also evidence to the contrary for each of the methods [36]. Another solution could be to combine [previous year ISIC data to create a larger dataset, although some modifications would have to be completed to ensure the datasets join properly.

When diagnosing skin lesions multiple different factors are considered when deciding on the type of disease (multi-label classification) or its severity (malignant or benign). This information would be held as metadata, such as location on the body, total number of moles, history of skin cancer (familial or individual) amongst other factors. When designing a classification system adding this information for additional weighting could be useful and could also aide with the natural imbalance between different types of skin lesion.

## IX. Prototype

The prototype for the models can be found at the following link as a Google Colab file.
https://colab.research.google.com/drive/1ZciN8yNE8pIxZlRT6iljQj_hvEbmaucz?usp=sharing

## X. References


[1] Goodfellow, I., Bengio, Y. and Courville, A., 2009. *Deep learning*.

[2] Hameed, Z., Zahia, S., Garcia-Zapirain, B., Javier Aguirre, J., & María Vanegas, A. (2020). Breast cancer histopathology image classification using an ensemble of deep learning models. *Sensors*, *20*(16), 4373.

[3] Cassidy, B., Kendrick, C., Brodzicki, A., Jaworek-Korjakowska, J. and Yap, M.H., 2022. Analysis of the ISIC image datasets: usage, benchmarks and recommendations. *Medical image analysis*, *75*, p.102305.

[4] Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, *29*(9), 2352-2449.

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[6] Theckedath, D., & Sedamkar, R. R. (2020). Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks. *SN Computer Science*, *1*(2), 1-7.

[7] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

[8] Lu, D., & Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, *28*(5), 823-870.

[9] Tammina, S. (2019). Transfer learning using VGG-16 with deep convolutional neural network for classifying images. *International Journal of Scientific and Research Publications (IJSRP)*, *9*(10), 143-150.

[10] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

[11] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

[12] Ballester, P., & Araujo, R. M. (2016, February). On the performance of GoogLeNet and AlexNet applied to sketches. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[13] He, K., Zhang, X., Ren, S., & Sun, J. (2016, October). Identity mappings in deep residual networks. In *European conference on computer vision* (pp. 630-645). Springer, Cham.

[14] Zhang Y, Allem JP, Unger JB, Cruz TB. Automated identification of hookahs (waterpipes) on Instagram: an application in feature extraction using convolutional neural network and support vector machine classification. *J Med Internet Res.* 2018 Dec 21;20(11):e10513. doi: 10.2196/10513.

[15] Géron, A. and Demarest, R., 2019. *Hands-on machine learning with Scikit-Learn and TensorFlow*. Sebastopol (Clif.) [etc.]: O'Reilly. [16] Ray, S. (2018). Disease classification within dermascopic images using features extracted by resnet50 and classification through deep forest. *arXiv preprint arXiv:1807.05711*.

[17] Mukti, I. Z., & Biswas, D. (2019, December). Transfer learning based plant diseases detection using ResNet50. In *2019 4th International Conference on Electrical Information and Communication Technology (EICT)* (pp. 1-6). IEEE.

[18] Bissoto, A., Perez, F., Ribeiro, V., Fornaciali, M., Avila, S., & Valle, E. (2018). Deep-learning ensembles for skin-lesion segmentation, analysis, classification: RECOD titans at ISIC challenge 2018. *arXiv preprint arXiv:1808.08480*.

[19] Xia, X., Xu, C., & Nan, B. (2017). Facial Expression Recognition Based on TensorFlow Platform.20 (Szegedy et al., 2015)

[21] Cui, Y., Chen, R., Chu, W., Chen, L., Tian, D., Li, Y., & Cao, D. (2021). Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems*.

[22] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[23] Kaur, T., & Gandhi, T. K. (2019, December). Automated brain image classification based on VGG-16 and transfer learning. In *2019 International Conference on Information Technology (ICIT)* (pp. 94-98). IEEE.

[24] Lopez, A. R., Giro-i-Nieto, X., Burdick, J., & Marques, O. (2017, February). Skin lesion classification from dermoscopic images using deep learning techniques. In *2017 13th IASTED international conference on biomedical engineering (BioMed)* (pp. 49-54). IEEE.

[25] Guan, Y., Wei, Q., & Chen, G. (2019). Deep learning based personalized recommendation with multi-view information integration. *Decision Support Systems*, *118*, 58-69.26 (Nair and Hinton, 2010)

[27] Koutsoukas, A., Monaghan, K. J., Li, X., & Huan, J. (2017). Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *Journal of cheminformatics*, *9*(1), 1-13.28 Smith et al., 2018

[29] Nwankpa, C. E. (2020). Advances in optimisation algorithms and techniques for deep learning. *Advances in Science, Technology and Engineering Systems Journal*, *5*(5), 563-577.

[30] Codella, N. C., Nguyen, Q. B., Pankanti, S., Gutman, D. A., Helba, B., Halpern, A. C., & Smith, J. R. (2017). Deep learning ensembles for melanoma recognition in dermoscopy images. *IBM Journal of Research and Development*, *61*(4/5), 5-1.

[31] Boureau, Y. L., Le Roux, N., Bach, F., Ponce, J., & LeCun, Y. (2011, November). Ask the locals: multi-way local pooling for image recognition. In *2011 International Conference on Computer Vision* (pp. 2651-2658). IEEE.

[32] Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, *6*(1), 1-54.

[33] Zhao, S., Liu, P., Tang, G., Guo, Y., & Li, G. (2022, January). External validation of a deep learning prediction model for in-hospital mortality among ICU patients. In *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)* (pp. 329-334). IEEE.

[34] Burdick H , Pino E , Gabel-Comeau D , *et al* Validation of a machine learning algorithm for early severe sepsis prediction: a retrospective study predicting

severe sepsis up to 48 h in advance using a diverse dataset from 461 US hospitals. BMC Med Inform Decis Mak 2020;**20**:276.
[35] Xia, X., Xu, C., & Nan, B. (2017, June). Inception-v3 for flower classification. In *2017 2nd International Conference on Image, Vision and Computing (ICIVC)* (pp. 783-787). IEEE.
[36] Gessert, N., Nielsen, M., Shaikh, M., Werner, R., & Schlaefer, A. (2020). Skin lesion classification using ensembles of multi-resolution EfficientNets with meta data. *MethodsX*, *7*, 100864.
[37] Wang, Y., Widrow, B., Zadeh, L. A., Howard, N., Wood, S., Bhavsar, V. C., ... & Shell, D. F. (2016). Cognitive intelligence: Deep learning, thinking, and reasoning by brain-inspired systems. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, *10*(4), 1-20.
[38] https://www.kaggle.com/datasets/andrewmvd/isic-2019
[39] Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent data analysis*, *6*(5), 429-449.
[40] Oliveira, D. A. B., Pereira, L. G. R., Bresolin, T., Ferreira, R. E. P., & Dorea, J. R. R. (2021). A review of deep learning algorithms for computer vision systems in livestock. *Livestock Science*, *253*, 104700.
[41] https://keras.io/