

Introduction

Melanoma is a type of skin cancer, develops in the cells (melanocytes) that produce melanin — the pigment that gives your skin its color, The exact cause of all melanomas isn't clear, but exposure to ultraviolet (UV) radiation from sunlight increases your risk of developing melanoma. (1)

melanoma is more dangerous because of its ability to spread to other organs more rapidly if it is not treated at an early stage. (2)

At present, CNN has achieved very good performance in the field of computer vision, such as object detection, image recognition, classification, etc. (3)

Convolutional Neural Network (CNN) is a type of deep learning model for processing data that has a grid pattern, such as images, which is designed to automatically and adaptively learn spatial hierarchies of features. CNN is a mathematical construct that is typically composed of three types of layers (or building blocks): convolution, pooling, and fully connected layers. The first two, convolution and pooling layers, perform feature extraction, whereas the third, a fully connected layer, maps the extracted features into final output, such as classification. A convolution layer plays a key role in CNN, which is composed of a stack of mathematical operations, such as convolution, a specialized type of linear operation. (4)

Because of the difficulty of detecting melanoma cancer in an ordinary way CNN is used to classify melanoma skin cancer.

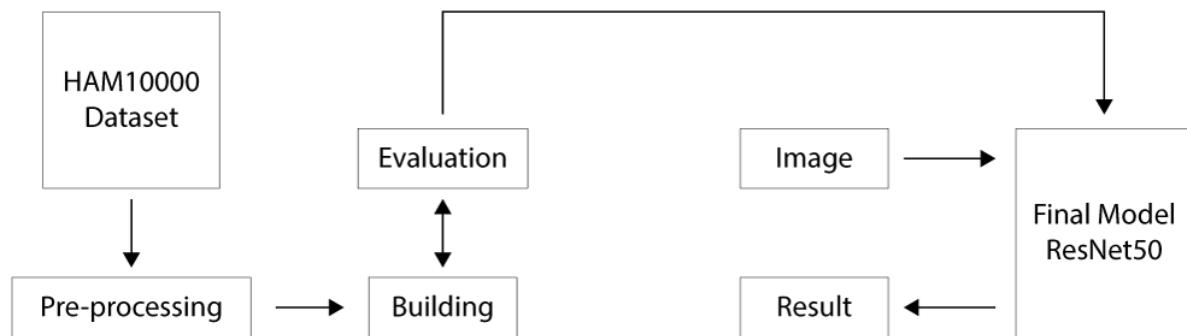
Research on the classification and detection of melanoma cancer by various methods has been carried out. In 2016 there was a paper entitled "Deep Residual Learning for Image Recognition" [9] using the ResNet architecture. The paper was a winner at the 2015 ILSVRC (Imagenet competition). (5)

Proposed Convolutional Neural Network Model

The main aim of this implementation is to detect melanoma skin cancer through RGB images, to achieve this, we build a deep learning model that is capable of extracting features from the given dataset.(6)

After delving into many articles and studies, we have found that the best convolutional neural network model we can suggest in this case is resnet50 and so we are going to implement it from scratch.

Figure 1 shows the proposed architecture which we have used for melanoma recognition.



Dataset ([MNIST- HAM10000](#))

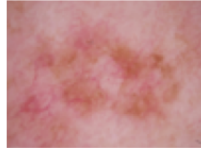
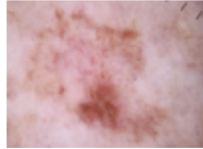
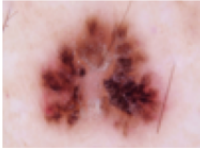
The ISIC archive is the largest public database for dermatoscopic image analysis research, and where the original HAM10000 was made available.(7)

The HAM10000 dataset is composed of 10.015 dermatoscopic images of pigmented skin lesions. The data was collected from Australian and Austrian patients. Two institutions participated in providing the images: Cliff Rosendahl in Queensland, Australia, and Medical University of Vienna, Austria. According to the authors, seven classes are defined on this dataset where some diagnoses were unified into one class for simplicity. Information regarding patient age, sex, lesion location and diagnosis is also provided with each image. (7)

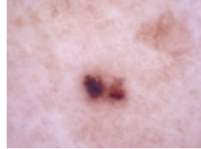
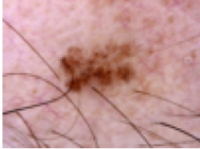
The dataset has been collated and published by [Tschandl, P., Rosendahl, C. & Kittler, H.](#)(7)

A sample of each type of skin lesion present in the dataset is demonstrated in the chart below.(7)

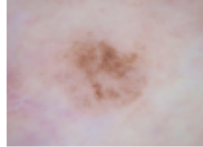
Actinic keratoses



Basal cell carcinoma



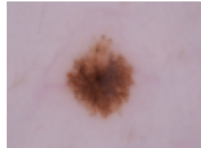
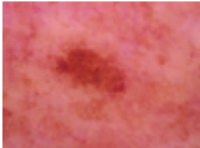
Benign keratosis-like lesions



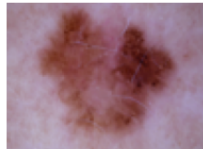
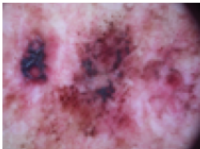
Dermatofibroma



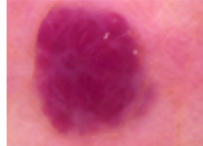
Melanocytic nevi



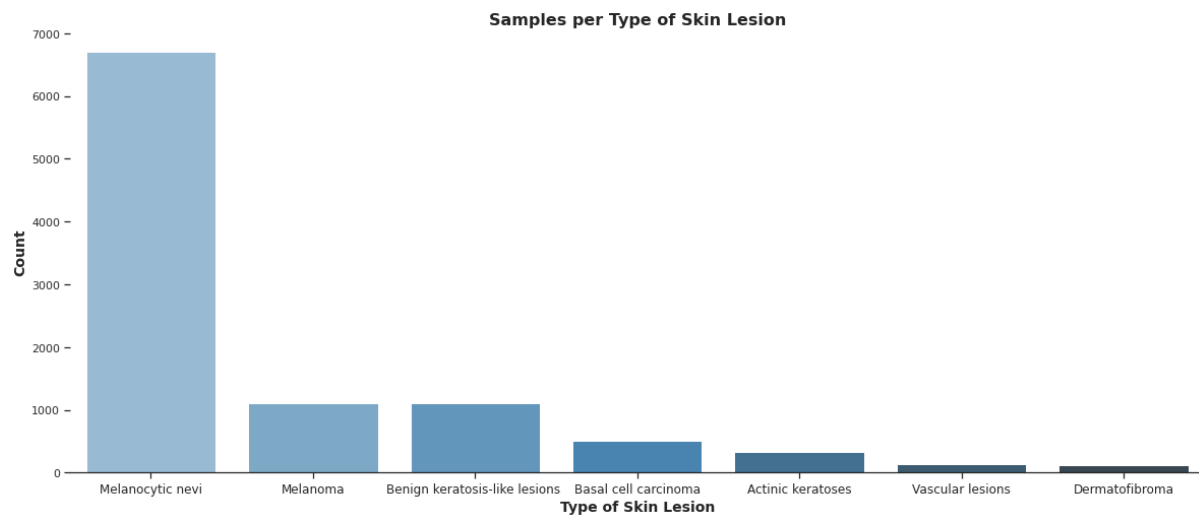
Melanoma



Vascular lesions



The count plot below helps to understand the distribution of the data.

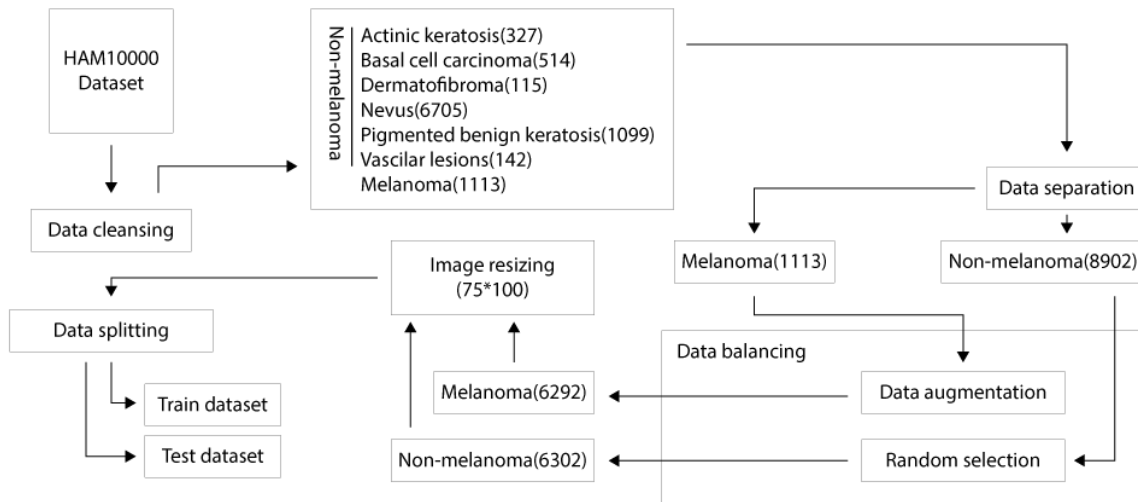


Pre-processing

Before starting the model training process we need to process the dataset, as we learned earlier the dataset consists of around 10015 labeled images for 7 different types of skin lesions, but in our case, we want to get images classified on only two types of skin lesions (Melanoma and Not melanoma). We do this in several steps:

- Data cleansing : In this step, we remove unused and damaged data, also repair data that is incorrectly formatted.
- Data separation : After cleansing the data set, we separate the data set into two types of skin lesions by changing the data label for the non-melanoma types to non-melanoma and we keep the data label for the type of melanoma as it is.
- Data balancing : When reclassifying the data set, we notice that the data set is numerically unbalanced. To solve this problem, we increase the number of images of the melanoma type by rotating, cropping and scaling. As for the non-melanoma type, we reduce the number of images by randomly selecting a specified number of images.
- Image resizing : In this step, we reduce the image size to 75*100 to speed up the training process of the deep learning model.
- data splitting : Before the data set becomes usable, we divide it into two parts, the first part is the training set with 80 percent, and the second part is the test set with 20 percent

The diagram below helps to understand these steps



The Problem of Very Deep Neural Networks

(all of these from coursera)

In recent years, neural networks have become much deeper, with state-of-the-art networks evolving from having just a few layers (e.g AlexNet) to over a hundred layers.

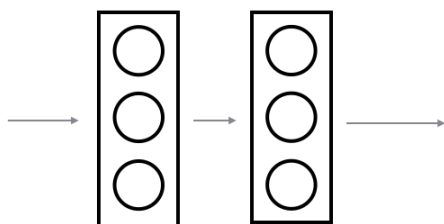
- The main benefit of a very deep network is that it can represent very complex functions. It can also learn features at many different levels of abstraction, from edges (at the shallower layers, closer to the input) to very complex features (at the deeper layers, closer to the output).
- However, using a deeper network doesn't always help. A huge barrier to training them is vanishing gradients: very deep networks often have a gradient signal that goes to zero quickly, thus making gradient descent prohibitively slow.
- More specifically, during gradient descent, as you backpropagate from the final layer back to the first layer, you are multiplying by the weight matrix on each step, and thus the gradient can decrease exponentially quickly to zero (or, in rare cases, grow exponentially quickly and "explode," from gaining very large values).
- During training, you might therefore see the magnitude (or norm) of the gradient for the shallower layers decrease to zero very rapidly as training proceeds.

Residual Network

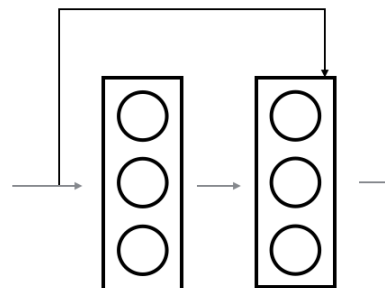
(all of these from coursera)

In ResNets, a "shortcut" or a "skip connection" allows the model to skip layers.

without skip connection



with skip connection



The image on the left shows the "main path" through the network. The image on the right adds a shortcut to the main path. By stacking these ResNet blocks on top of each other, you can form a very deep network.

Having ResNet blocks with the shortcut also makes it very easy for one of the blocks to learn an identity function. This means that you can stack on additional ResNet blocks with little risk of harming training set performance.

There is also some evidence that the ease of learning an identity function accounts for ResNets' remarkable performance even more than skip connections help with vanishing gradients.

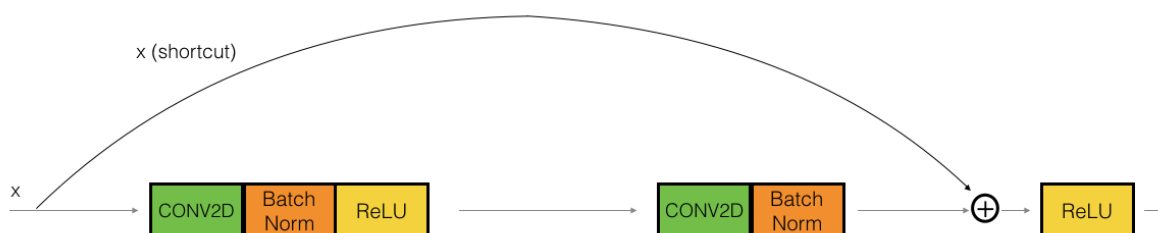
Two main types of blocks are used in a ResNet, depending mainly on whether the input/output dimensions are the same or different. the "identity block" and the "convolutional block"

The Identity Block

(all of these from coursera)

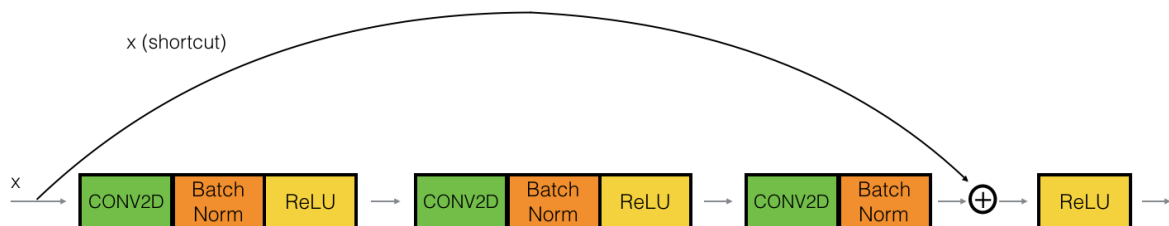
The identity block is the standard block used in ResNets, and corresponds to the case where the input activation (say $a[l]$) has the same dimension as the output activation (say $a[l+2]$).

To flesh out the different steps of what happens in a ResNet's identity block, here is an alternative diagram showing the individual steps:



The upper path is the "shortcut path." The lower path is the "main path." In this diagram, notice the CONV2D and ReLU steps in each layer. To speed up training, a BatchNorm step has been added.

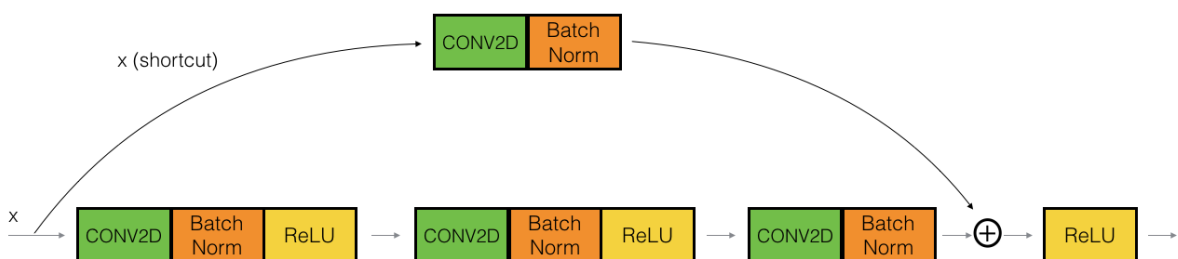
But in this study, i'll actually implement a slightly more powerful version of this identity block, in which the skip connection "skips over" 3 hidden layers rather than 2 layers. It looks like this:



The Convolutional Block

(all of these from coursera)

The ResNet "convolutional block" is the second block type. You can use this type of block when the input and output dimensions don't match up. The difference with the identity block is that there is a CONV2D layer in the shortcut path:



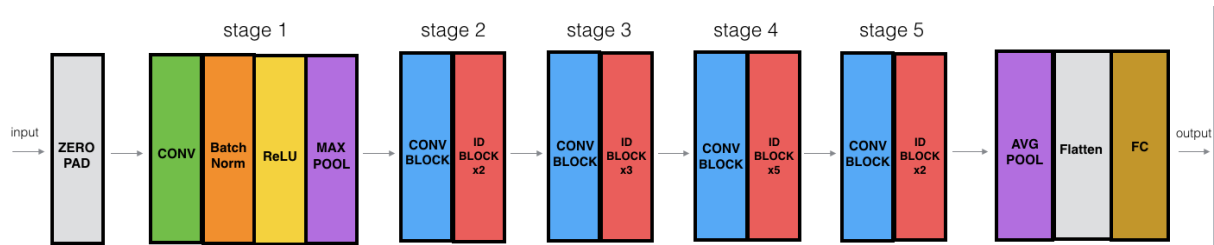
The CONV2D layer in the shortcut path is used to resize the input x to a different dimension, so that the dimensions match up in the final addition needed to add the shortcut value back to the main path.

The CONV2D layer on the shortcut path does not use any non-linear activation function. Its main role is to just apply a (learned) linear function that reduces the dimension of the input, so that the dimensions match up for the later addition step.

Residual Network Model with 50 layers (ResNet50)

(all of these from coursera)

The following figure describes in detail the architecture of this neural network. "ID BLOCK" in the diagram stands for "Identity block," and "ID BLOCK x3" means you should stack 3 identity blocks together.



The details of this ResNet-50 model are:

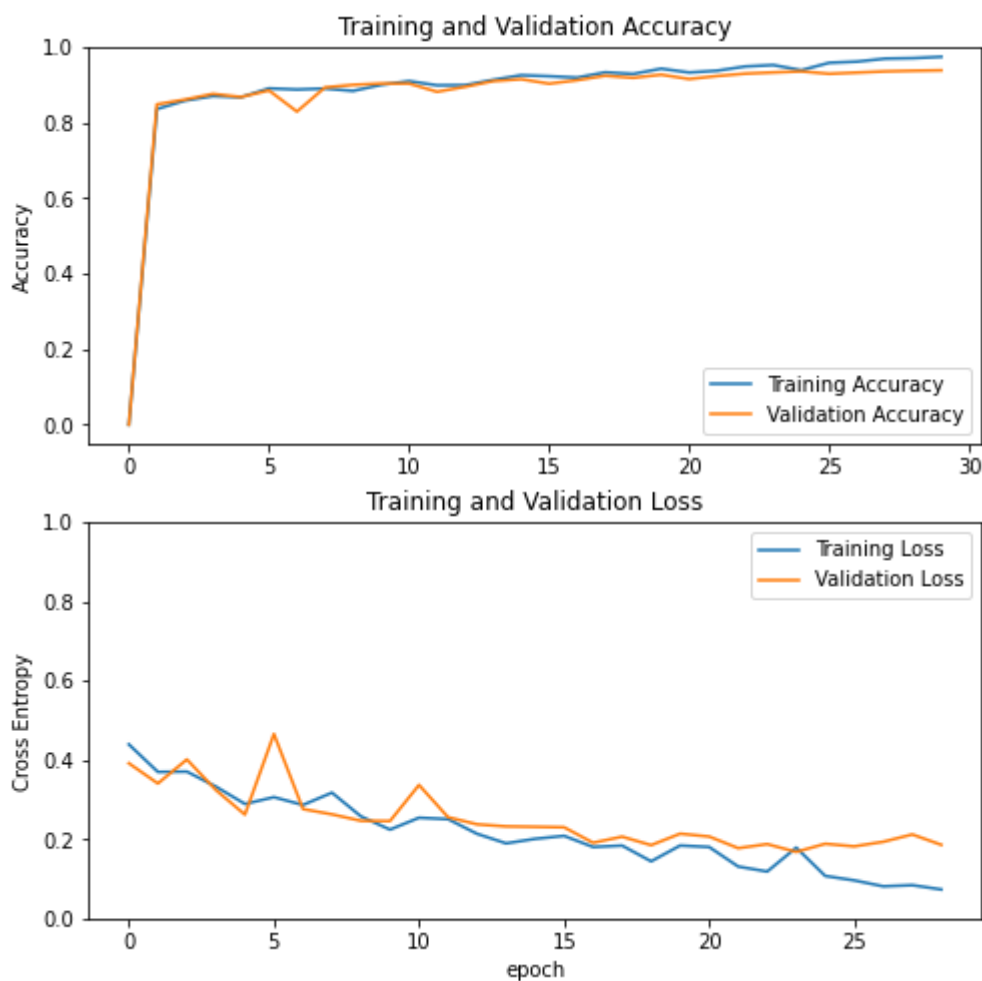
- Zero-padding pads the input with a pad of (3,3)
- Stage 1:
 - o The 2D Convolution has 64 filters of shape (7,7) and uses a stride of (2,2).
 - o BatchNorm is applied to the 'channels' axis of the input.
 - o ReLU activation is applied.
 - o MaxPooling uses a (3,3) window and a (2,2) stride.
- Stage 2:
 - o The convolutional block uses three sets of filters of size [64,64,256], "f" is 3, and "s" is 1.
 - o The 2 identity blocks use three sets of filters of size [64,64,256], and "f" is 3.
- Stage 3:
 - o The convolutional block uses three sets of filters of size [128,128,512], "f" is 3 and "s" is 2.
 - o The 3 identity blocks use three sets of filters of size [128,128,512] and "f" is 3.
- Stage 4:
 - o The convolutional block uses three sets of filters of size [256, 256, 1024], "f" is 3 and "s" is 2.
 - o The 5 identity blocks use three sets of filters of size [256, 256, 1024] and "f" is 3.
- Stage 5:
 - o The convolutional block uses three sets of filters of size [512, 512, 2048], "f" is 3 and "s" is 2.
 - o The 2 identity blocks use three sets of filters of size [512, 512, 2048] and "f" is 3.
- The 2D Average Pooling uses a window of shape (2,2).
- The 'flatten' layer doesn't have any hyperparameters.
- The Fully Connected (Dense) layer reduces its input to the number of classes using a softmax activation.

Experimental results

To judge the performance of the model for the task of predicting skin lesions, we use several evaluation metrics to evaluate our model. This is because the model may perform well using one measurement from one evaluation metric, but may perform poorly using another measurement from another evaluation metric. Using evaluation metrics are critical in ensuring that our model is operating correctly and optimally.

When the model was trained for 30 epochs, it was observed that the accuracy for both the training and test data started with rather large values and continued to increase small from epoch 4 until it reached its peak in epoch 30, where the test accuracy reached 93 percent and the training accuracy was 97 percent.

The plot for the accuracy and loss obtained during the training and testing process is shown in Fig 14



The table below also includes several other measurements that we used in evaluating our model

Classes	Precision	Recall	F1-score	Support
Non-melanoma	0.95	0.93	0.94	1293
Melanoma	0.93	0.95	0.94	1226