Skin cancer classification

Using deep learning algorithms

Overview

Skin diseases are more common than other diseases. Skin diseases may be caused by fungal infection, bacteria, allergy, or viruses, etc. The advancement of lasers and Photonics based medical technology has made it possible to diagnose the skin diseases much more quickly and accurately. But the cost of such diagnosis is still limited and very expensive. So, image processing techniques help to build automated screening system for dermatology at an initial stage. The extraction of features plays a key role in helping to classify skin diseases. Computer vision has a role in the detection of skin diseases in a variety of techniques.

Finding out the type of cancer can take months for doctors as it is a very tedious practice and also requires the use of expensive devices and contraptions. This would waste a good lot of time for the patient which could have been utilized by the doctors to treat the patient in time. It is also monetarily very demanding on the patient's part, which could pose a problem if the patient is not financially settled.

Chapter 1

1.1 Introduction

Deep learning is a machine learning branch that fashions high-stage abstractions in facts using many processing layers. In this paper one such model CNN is discussed and used. It was initially designed to recognize cursive numbers and is later proved to be useful in object detection. These model are proved to be powerful classification tools. CNN models inclusive of Google Net, VGG and Resnet have showed better performances in image classification and recognition.

Despite the advancements in technology, however the inefficiency within the clinical dataset has restricted the utility of deep learning in biological data. Melanoma is a most usual skin cancer that showed huge mortality rate. It is estimated that nearly 9,730 deaths have occurred due to melanoma in 2017.

Basal cell carcinoma commonly referred to as BCC is the most usual skin cancer, however is commonly not fatal. So it is very important for both health care services to diagnose the type of cancer and to develop an efficient method to discriminate different types of skin cancer will be useful for initial screening. To improve a classification tool using biological images of 7 different known skin lesions - Dermatofibroma, Melanocytic nevi, Basal cell carcinoma, Benign keratosis, Actinic keratosis, Vascular lesions, Melanoma.

1.2 Related Discussion

Huge efforts should be made to improve a image classification methods for more precise prediction of lesions. In one of the old studies, machine-guided diagnostic methods depending on feature extraction method imaged a considerable diagnostic capability with some types of skin cancer, which includes melanoma. However, an Al algorithm would not create precise diagnoses over a large varieties class of skin cancers. Not very long back, deep CNN architectures became popular in object classification going feature learning especially with image data. Extensive research from ILSVR (ImageNet Large Scale Visual Recognition Challenge) has depicted that object classification abilities of CNNs can exceed over that of human diagnosis abilities.

Many dermatologic studies showed the uses of machine or deep learning. For example, Liao et ai. Used a CNN based model to Classify top level 23 Categories Such as, viral infections, bullous diseases etc. With 23000 images. It showed an accuracy of 73.1% and 91.0% respectively for rates at which a model gives Output of the correct label with top-1 and top-5 predictions for a given image. They have used a binary classification CNN based model which gave an AUC of 0.96 for carcinoma diagnosis using the above mentioned Edinburgh dataset with 707 cases and gave AUC of 0.96 for melanoma diagnosis which has 225 cases.

Chapter 2

Exploratory Data Analysis (EDA)

Here we discuss about the different features of the dataset, their distributions and the count of that types present in the dataset. This is helpful to analysis the nature of our data and helps us in the data processing step. First we will see the number of instances of data present for every possible values of every feature of data feature wise study of the data. Before jumping into analysis part let us look at from where the data is collected.

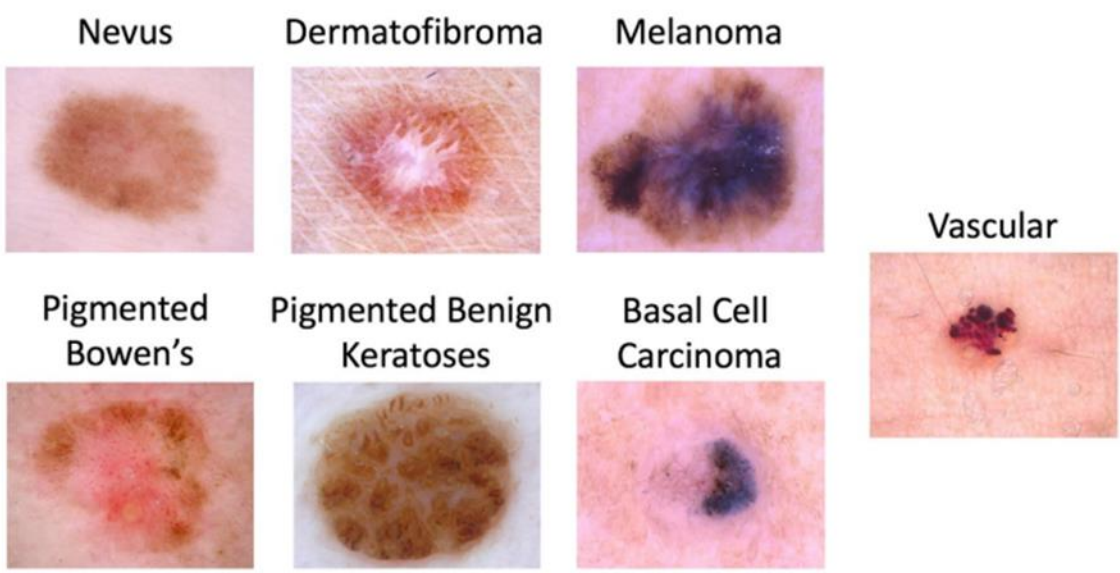
2.1 Data Collection

This the HAM10000 ("Human against machine with 10000 training images") dataset. It consists of 10015 dermatoscopy images which are released as a training set for academic machine learning purposes and are publicly available through the ISIC archive. This benchmark dataset can be used for machine learning and for comparisons with human experts.

https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DBW86T

It has 7 different classes of skin lesion which are listed below:

- Melanocytic Nevi (NV)
- Melanoma (MEL)
- Benign Keratosis-like Lesion (BKL)
- Dermatofibroma (DF)
- Basal Cell Carcinoma (BCC)
- Actinic Keratoses (Akiec)
- Vascular Lesions (Vasc)

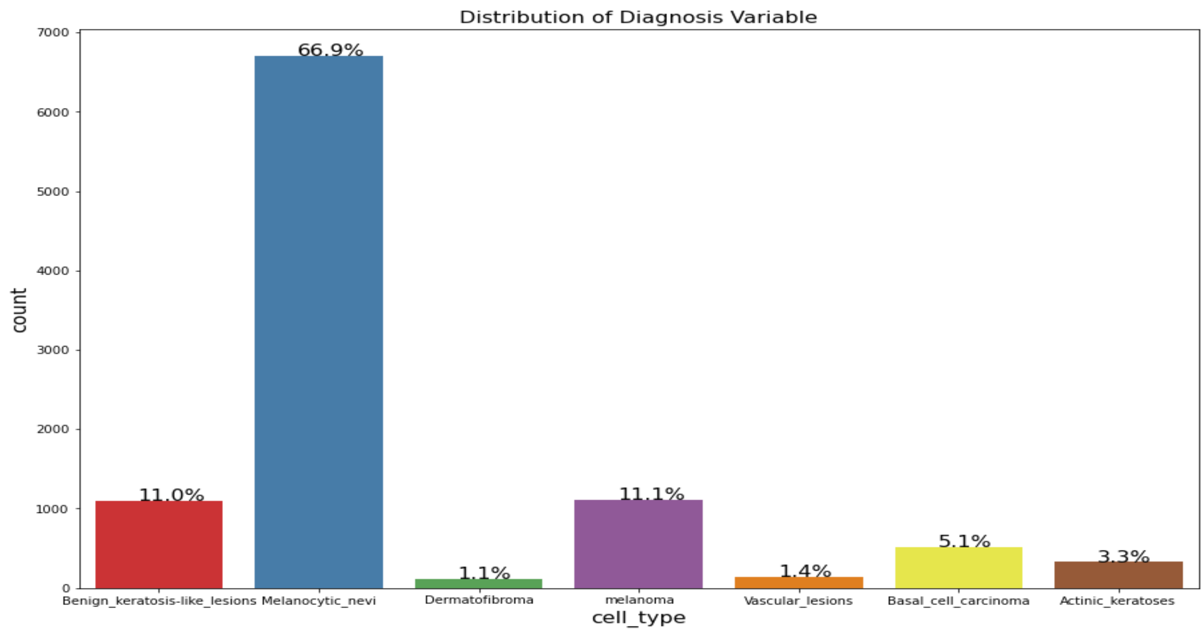


Melanocytic nevi, keratosis, vascular and dermatofibroma are benign lesions

Melanoma, basal cell carcinoma and actinic keratosis are malignant lesions

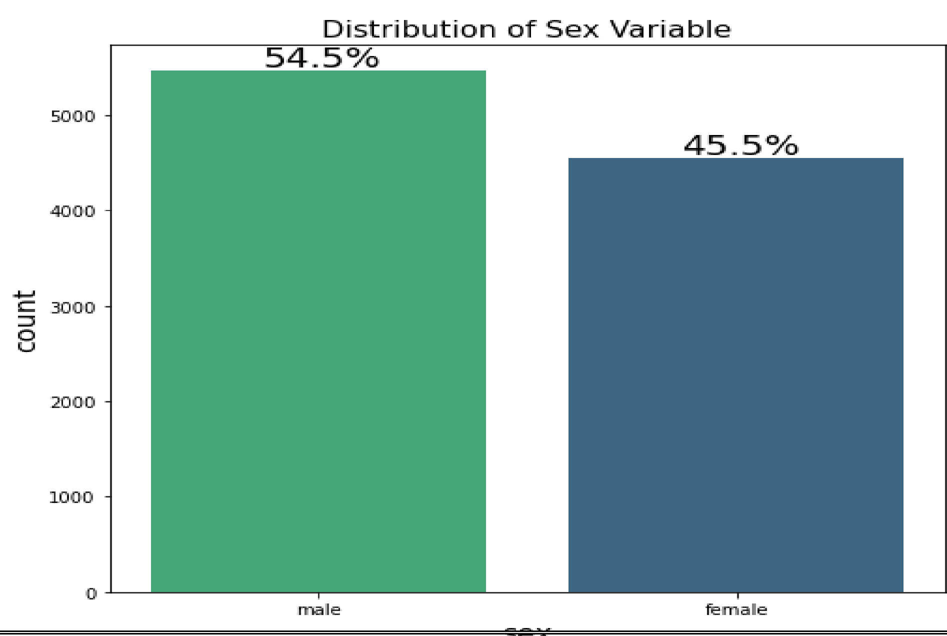
2.2 Feature wise study

1) Diagnosis

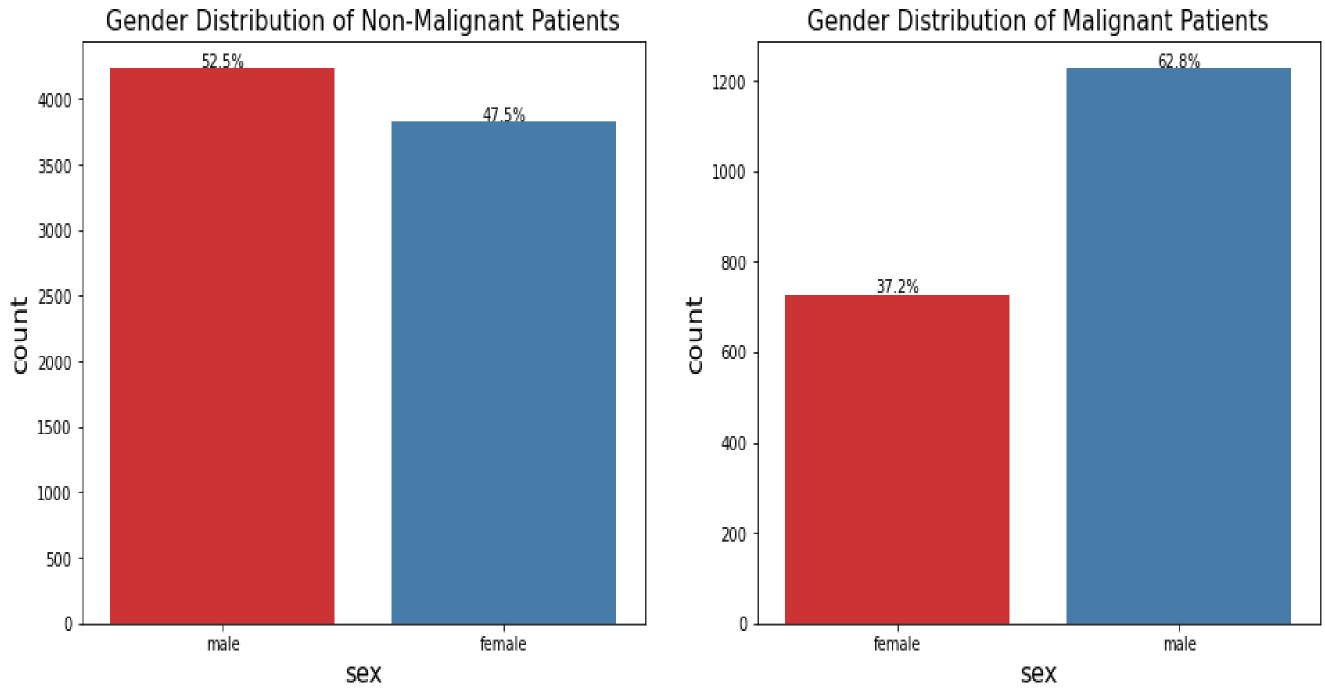


Its seems from the above plot that in this dataset cell type melanocytic nevi has very large number of instances in comparison to other cell types

2) Gender



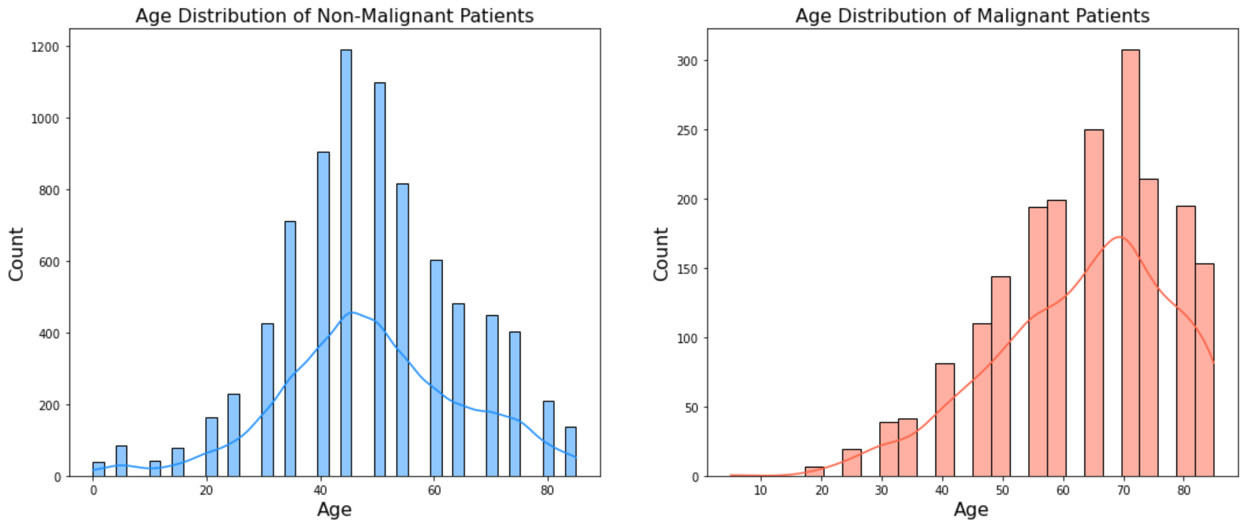
55/45 distribution, dataset matches industry established distribution

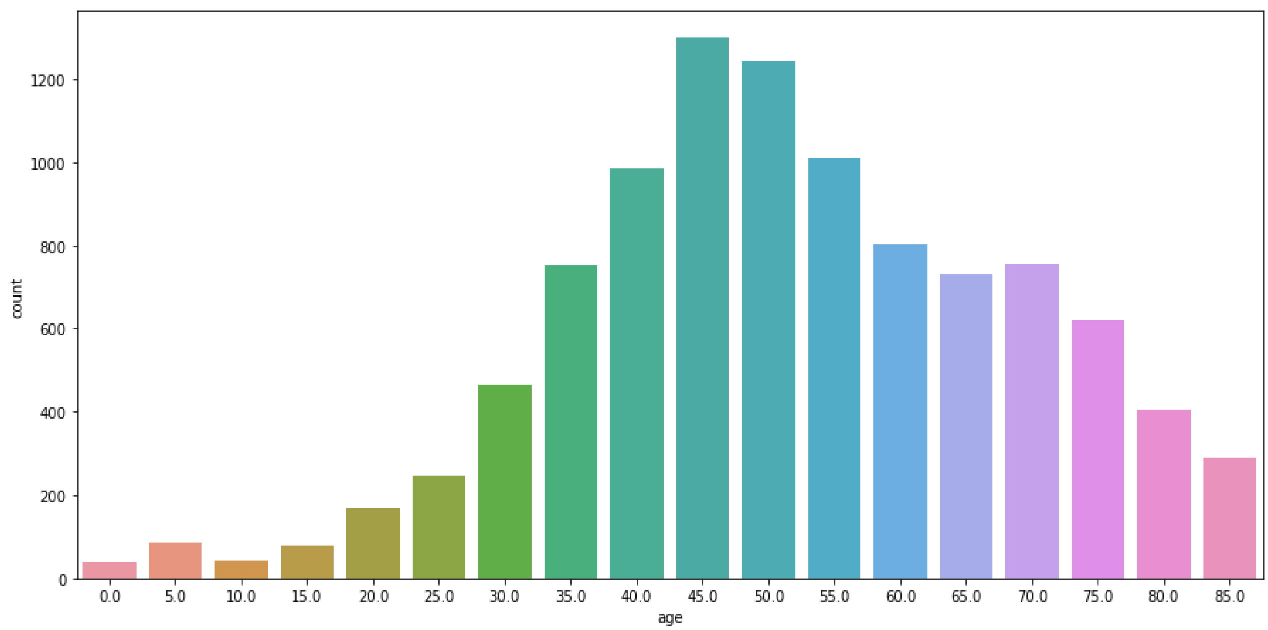


There is a 60/40 split in the Malignant population for males and females due to the fact that under 50, melanoma occurs more frequently in women, while above 50, occurs more often in men and increasingly so moving into 80.

https://www.cancer.net/cancer-types/melanoma/statistics

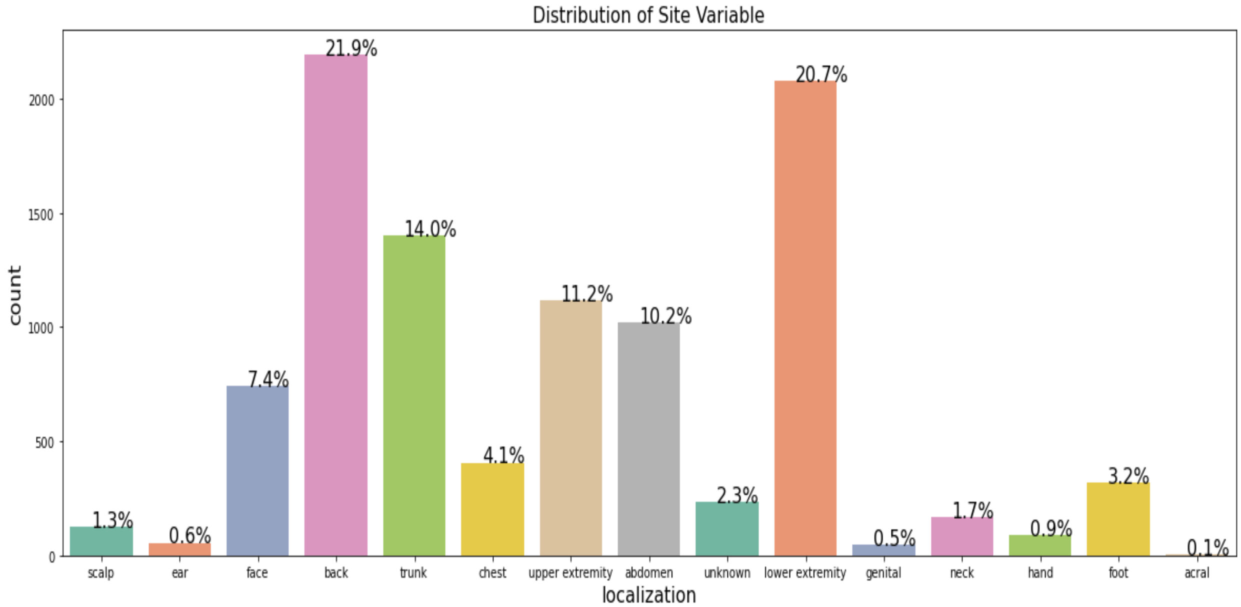
3) Age





It seems that there are larger instances of patients having age from 30 to 60

4) Site



It seems back, lower extremity, trunk and upper extremity are heavily compromised regions of skin cancer

5) Technical Validation field (ground truth)

The distribution of its 4 categories which are listed below:

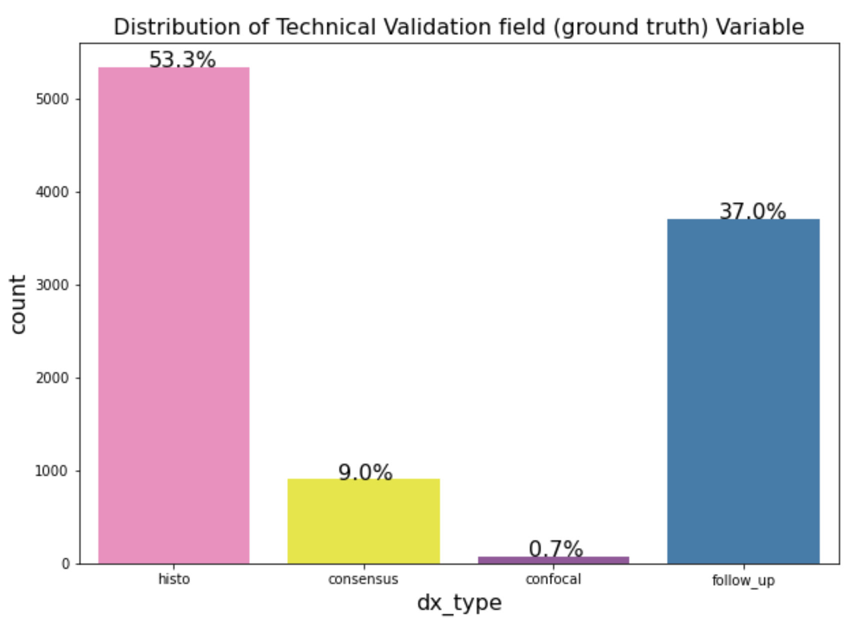
1) Histopathology (Histo): Histopathologic diagnoses of excised lesions have been performed by specialized dermatopathologists.

2) Confocal: Reflectance confocal microscopy is an in-vivo imaging technique with a resolution at near-cellular level , and some facial benign with a grey-world assumption of all

training-set images in Lab-color space before and after manual histogram changes.

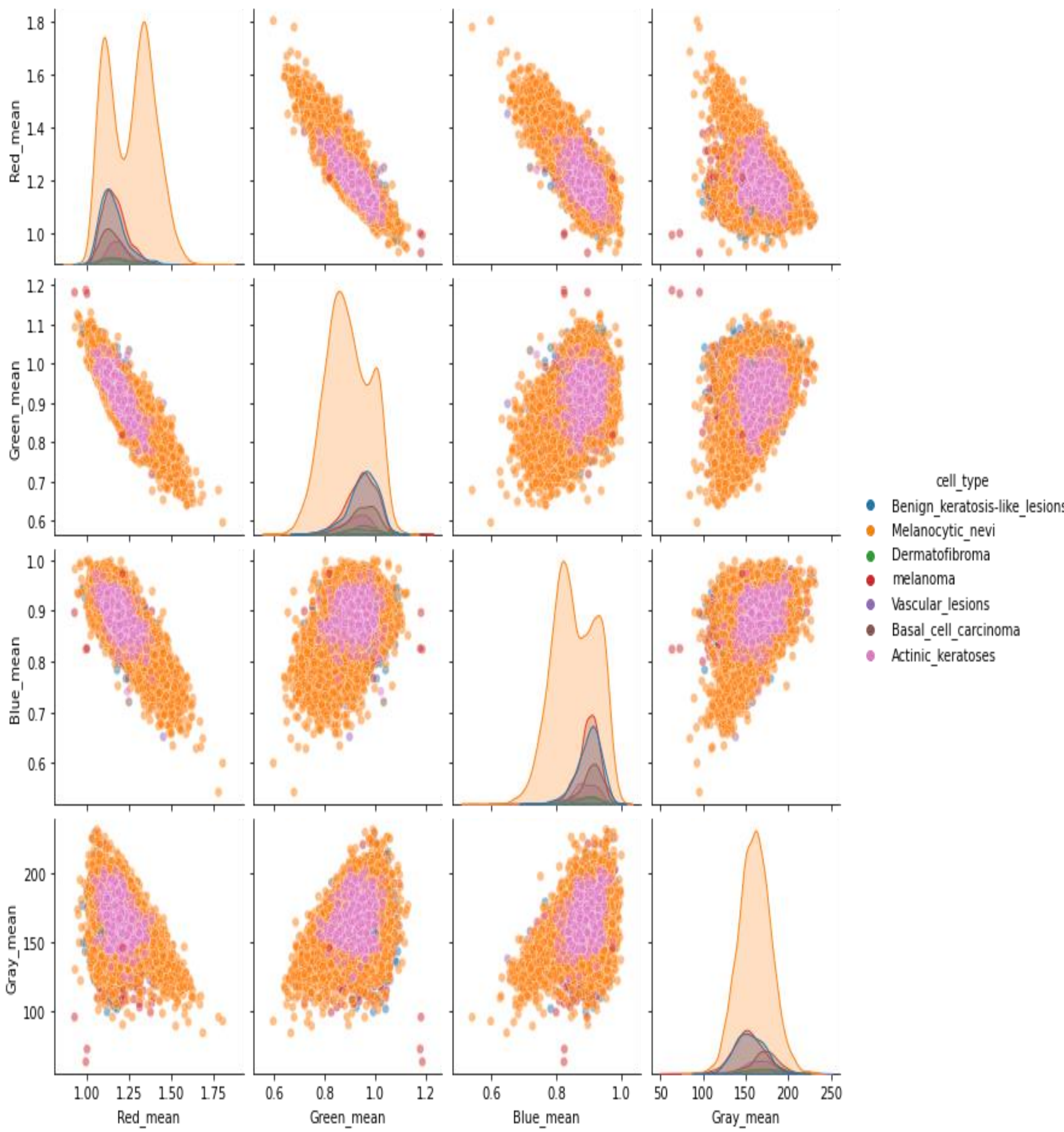
3) Follow-up: If nevi monitored by digital dermatoscopy did not show any changes during 3 follow-up visits or 1.5 years biologists accepted this as evidence of biologic benignity. Only nevi, but no other benign diagnoses were labeled with this type of ground-truth because dermatologists usually do not monitor dermatofibromas, seborrheic keratoses, or vascular lesions.

4) Consensus: For typical benign cases without histopathology or follow up biologists provide an expert-consensus rating of authors PT and HK. They applied the consensus label only if both authors independently gave the same unequivocal benign diagnosis. Lesions with this type of ground truth were usually photographed for educational reasons and did not need further follow-up or biopsy for confirmation.



2.3 Average Color Information

Here we get and normalize all of the color channel information, the shape of the image array is (450, 600, 3), 3 are the 3 channels: Red, Blue and Green! Taking the mean across axis= (0, 1) gives the mean for each 3 channels.



Chapter 3

Data Processing

- **Data Cleaning**

Data cleaning is an important step in machine learning. Data cleaning plays an important role in building a proper model. Proper data cleaning can make or break the project. There is a popular belief that "Better data beats fancier algorithms".

A different steps in data cleaning are,

1) Unwanted observations removal

This means deleting duplicate, irrelevant and redundant values from dataset. Duplicate data arise mostly during data collection. Irrelevant observations are those observations that does not fit the certain problem that we are trying to solve. Our data does not contain any unwanted or duplicate, we removed 57 unknown values in gender field

2) Handling missing data

This type of data poses tricky problems in machine learning. This data cannot be ignored or removed from dataset because this data may contain important features specific to the corresponding class other than that of missing feature. In this step we find all the null values in the data and replace them with mean and mode values of that field. In our data only ‘age’ field has null data as shown in and they are replaced by the median values.

- **Resizing Images**

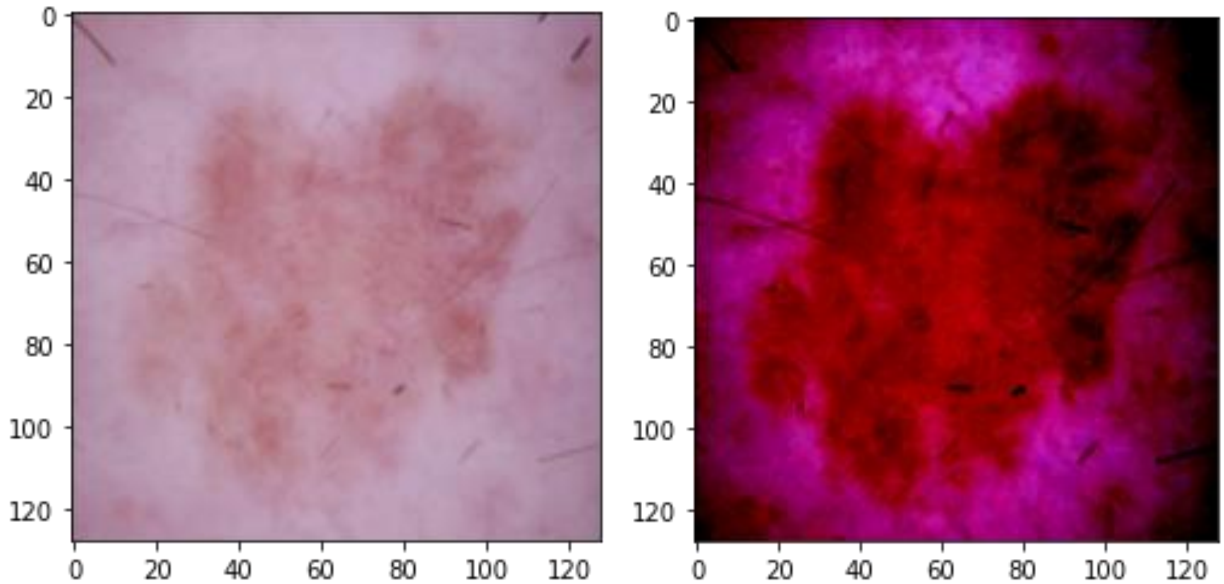
Original images in the dataset are all of same size but the size is very large. The images are of dimension (450 x 600 x 3) as seen this will have a huge computation time while training and TensorFlow cannot handle this.

Let us resize the images by keeping the same size ratio so that no information is lost. We have resized the images to dimension (128 x 128x 3).

- **Data Normalization**

Original images are represented in color code format with 3 values of Red, Blue and Green for each pixel and each value ranging from 0 to 255. So the normalization is applied as follows, normalized image = (original Image – mean (all original images) / standard deviation (all original images)

After normalization, each value of color code format changed to a range of -2 to 2 which is preferred by neural networks.



- **Data Augmentation**

Deep neural networks perform better with large amount of data. Aim of this step is to create images that depict the features of its class in every possible angle. This makes sure that at whatever angle the image may be taken, our trained model can predict it with more precision.

Different techniques used for this are,

1) Randomly rotate the images in the range from 0 to 180 degrees

2) Randomly zoom images

3) Randomly flip images horizontally

4) Randomly flip images vertically

5) Randomly shift the images horizontally

6) Randomly shift the images vertically

Augmentation of images is done to deal with the problem of skewed classes, overfitting, and training image scarcity. As can be seen from the frequency table of classes, the NV class dominates with approximately 67% of images in training data. Hence, to balance the distribution various augmentation techniques are implemented to increase the size of each class

Chapter 4

Modeling

Convolutional Neural Networks are currently the most popular and effective technique for image classification. In this project, we aim to develop a CNN model for classifying Skin Lesion images into 7 different classes of lesions. There are various CNN models developed by companies like Google and researchers based on ImageNet dataset. We have explored some of these models by training them on ISIC 2018 Skin Lesion Image dataset to verify whether they provide expected accuracy. Moreover, we have developed custom models based on study of various research papers. In this report, we have described all the attempts made to develop a robust CNN model for classification.

We did extensive research to understand the intuition behind making a robust CNN model especially for Skin Lesion classification. Most of the research work we reviewed were performing binary classification to detect whether a Lesion is Melanoma or not. These models achieved good accuracy due to binary nature of classification. Some of the research work was for all the 7 classes of ISIC dataset however, the dataset used to train was a small subset of 10k images that are available in ISIC archives. Although the models developed in these papers achieved good accuracy, they were not generalized model since the amount of data used to train was less.

As we developed an idea about the purpose of different convolution layers, number of filters, pooling and dropout, we were able to tweak proposed and pre-implemented models to train them for our ISIC 2018 dataset.

The following is a detailed description of different attempts made to develop an efficient model along with limitations of each model.

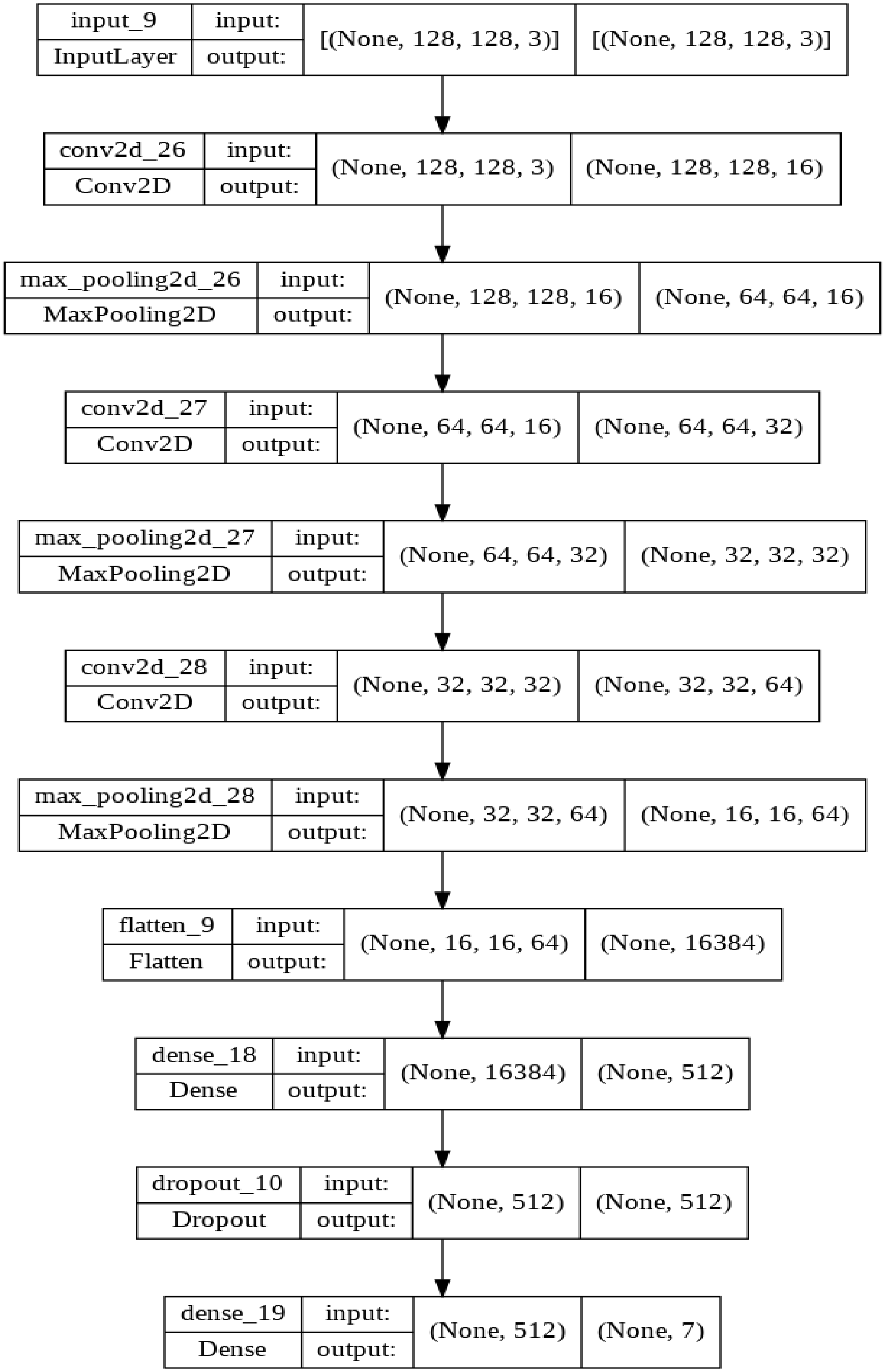
Baseline model (CNN)

CNN model proposed by researchers at Telkom University was first utilized to develop a CNN model for ISIC 2018 images. The researchers developed model for 4 different classes of images while for our dataset, we tweaked the model to work for 7 classes of output layer. We used this model as a base reference because it was used for skin lesion classification & achieved good accuracy. Also, it is a simpler model for implementation and understanding.

The architecture of this model is heuristically based we follow the convention in famous DCNNs: using the smallest (3x3) convolutional layers; and double the number of filters in the output whenever the spatial activation size is halved to maintain roughly constant hidden dimensions.

To train this model, data augmentation is employed. The intuition of this method is to transform the training dataset a bit in each epoch to produce variation and to guarantee that the model will never see the same image twice.

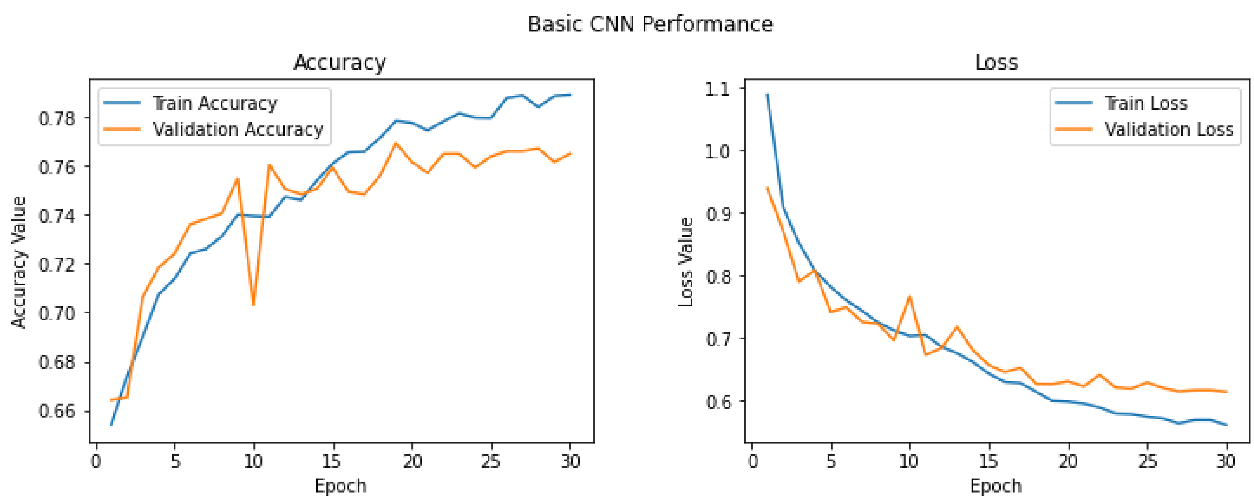
Learning rate is initialized at 0.01 and Adam optimizer is used. Learning rate decay is also used so that the learning rate will halve whenever the validation accuracy plateaus for 3 epochs. Baseline model is trained for a total of 30 epochs


input_9
InputLayer
input:
output:
[(None, 128, 128, 3)]
[(None, 128, 128, 3)]
conv2d_26
Conv2D
input:
output:
(None, 128, 128, 3)
(None, 128, 128, 16)
max_pooling2d_26
MaxPooling2D
input:
output:
(None, 128, 128, 16)
(None, 64, 64, 16)
conv2d_27
Conv2D
input:
output:
(None, 64, 64, 16)
(None, 64, 64, 32)
max_pooling2d_27
MaxPooling2D
input:
output:
(None, 64, 64, 32)
(None, 32, 32, 32)
conv2d_28
Conv2D
input:
output:
(None, 32, 32, 32)
(None, 32, 32, 64)
max_pooling2d_28
MaxPooling2D
input:
output:
(None, 32, 32, 64)
(None, 16, 16, 64)
flatten_9
Flatten
input:
output:
(None, 16, 16, 64)
(None, 16384)
dense_18
Dense
input:
output:
(None, 16384)
(None, 512)
dropout_10
Dropout
input:
output:
(None, 512)
(None, 512)
dense_19
Dense
input:
output:
(None, 512)
(None, 7)

The model give training accuracy of 78% with loss 0.56 and test accuracy around 76% with loss 0.61

	loss	accuracy	val_loss	val_accuracy	lr
29	0.5613	0.7891	0.6142	0.7650	3.1250e
26	0.5639	0.7889	0.6149	0.7661	6.2500e
28	0.5694	0.7886	0.6170	0.7616	3.1250e
25	0.5718	0.7878	0.6206	0.7661	6.2500e
27	0.5694	0.7841	0.6169	0.7672	6.2500e
22	0.5800	0.7815	0.6212	0.7650	1.2500e
23	0.5790	0.7798	0.6193	0.7594	1.2500e
24	0.5747	0.7797	0.6291	0.7639	1.2500e
18	0.6002	0.7786	0.6265	0.7694	2.5000e
21	0.5896	0.7783	0.6414	0.7650	2.5000e

Show the changes in accuracy and loss training data and validation data during training process as the epochs progress



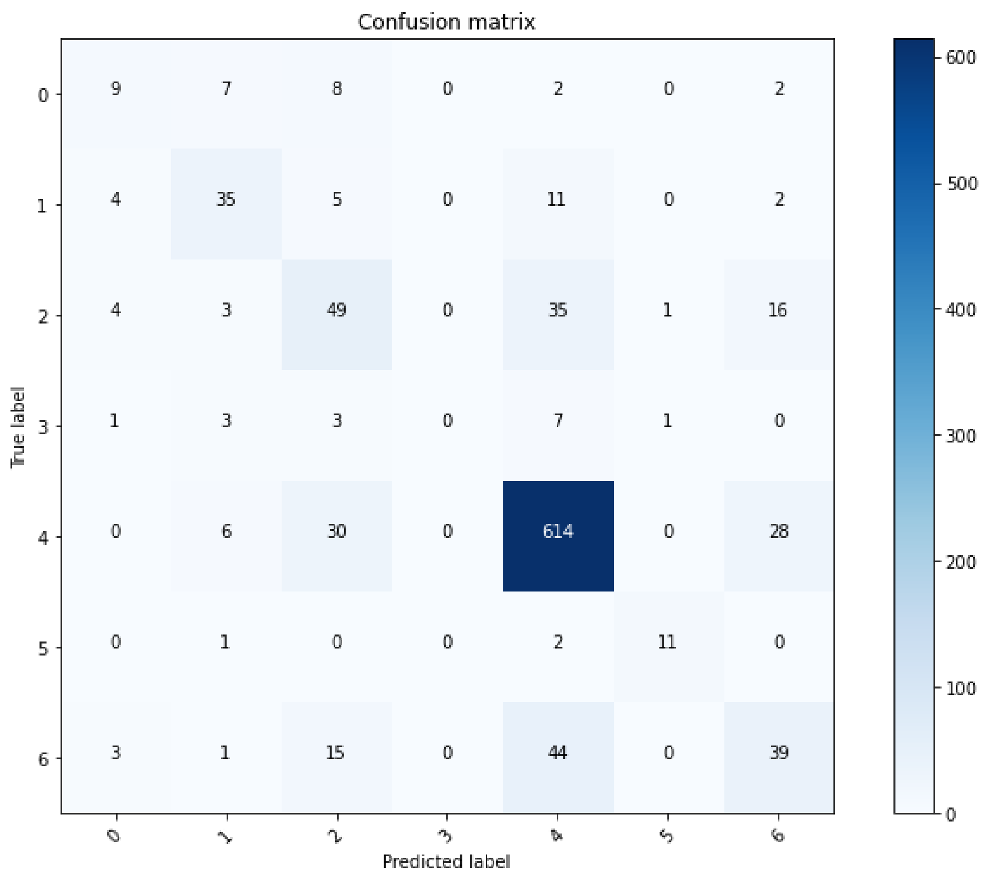
Even though there are some spikes in both the validation accuracy and loss, we can still say that this particular model with image augmentation is doing well

Training and validation accuracies are much similar now, which proves that the model is not overfitting

Showing the main classification metrics using classification report

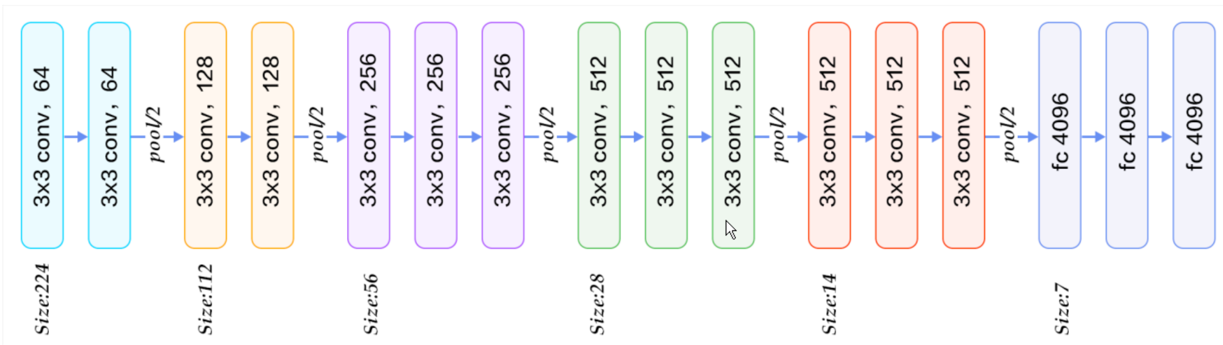
	precision	recall	f1-score	support
0	0.50	0.11	0.18	28
1	0.69	0.42	0.52	57
2	0.55	0.38	0.45	108
3	0.00	0.00	0.00	15
4	0.88	0.88	0.88	678
5	0.91	0.71	0.80	14
6	0.54	0.34	0.42	102
micro avg	0.82	0.71	0.76	1002
macro avg	0.58	0.41	0.46	1002
weighted avg	0.78	0.71	0.73	1002
samples avg	0.71	0.71	0.71	1002

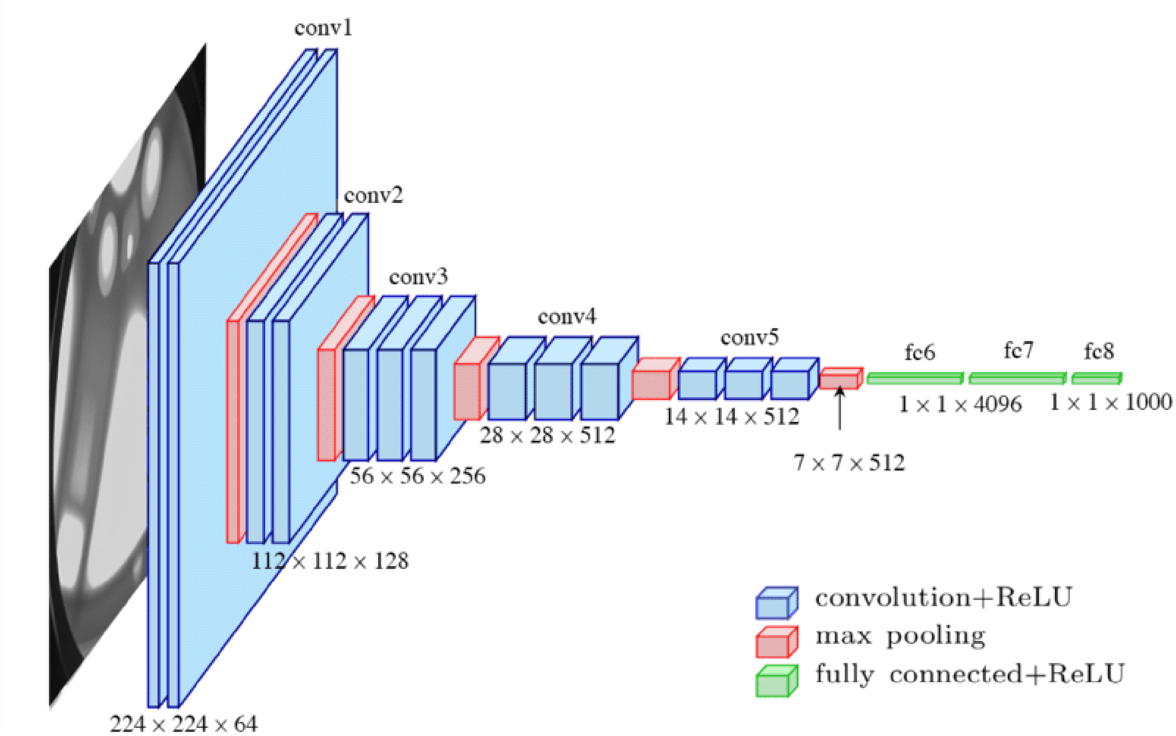
Computing confusion matrix to evaluate the accuracy of a classification.



VGG16

Even though there are many DCNNs model achieving better result on ImageNet than VGG16, we choose to fine-tune VGG16 given its simplicity. The best performing VGG16 net is similar except that the third, fourth and fifth convolutional block has 4 convolutional layers.



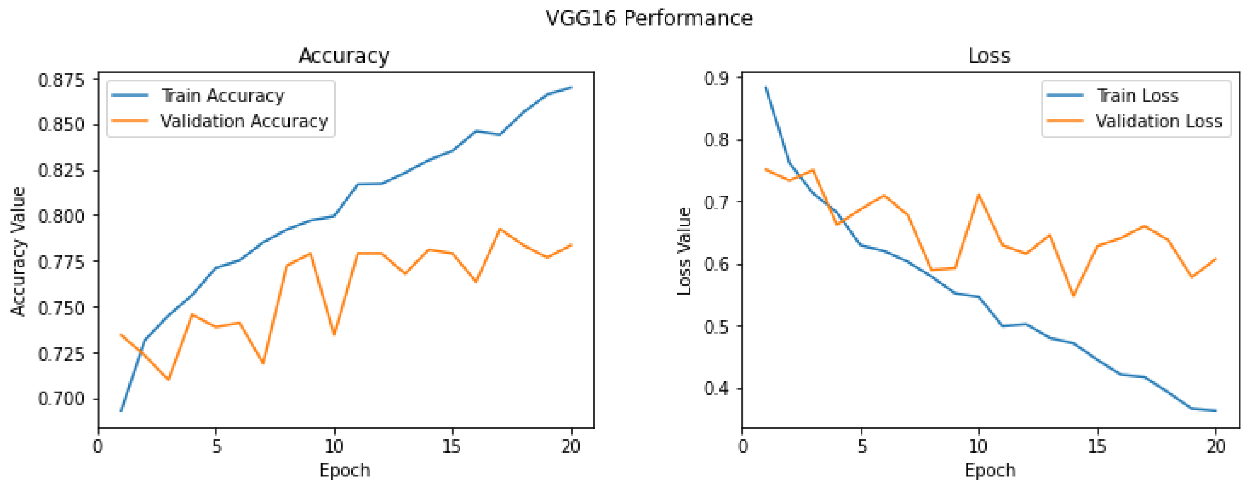


To fine-tune VGG16, the top fully-connected layers are removed, and new fully-connected layers (consisting of: one global max pooling layers, one fully connected layer with 512 units, one dropout layer with 0.5 rate, one softmax activation layer for 7 types of skin lesions) for our classification tasks are added, First, freeze all layers in VGG16, and perform feature extraction for the newly added FC layers so that the weights for these layers aren't completely random and the gradient wouldn't be too large when we start fine-tuning. After 3 epochs of feature extraction, we unfreeze the final convolutional block of VGG16 and start fine-tune the model for 20 epochs. Throughout the training process, learning rate of0.001 and Adam optimizer are used. The same data augmentation and learning rate decay strategy as in baseline model is used.

The model give training accuracy of 86% with loss 0.36 and test accuracy around 60% with loss 0.78

	loss	accuracy	val_loss	val_accuracy	lr
19	0.3622	0.8698	0.6061	0.7835	1.0000e
18	0.3656	0.8660	0.5770	0.7768	1.0000e
17	0.3919	0.8563	0.6369	0.7835	1.0000e
15	0.4205	0.8460	0.6402	0.7634	1.0000e
16	0.4164	0.8440	0.6593	0.7924	1.0000e
14	0.4443	0.8352	0.6270	0.7790	1.0000e
13	0.4709	0.8301	0.5469	0.7812	1.0000e
12	0.4794	0.8232	0.6449	0.7679	1.0000e
11	0.5018	0.8171	0.6150	0.7790	1.0000e
10	0.4986	0.8168	0.6287	0.7790	1.0000e

Show the changes in accuracy and loss training data and validation data during training process as the epochs progress

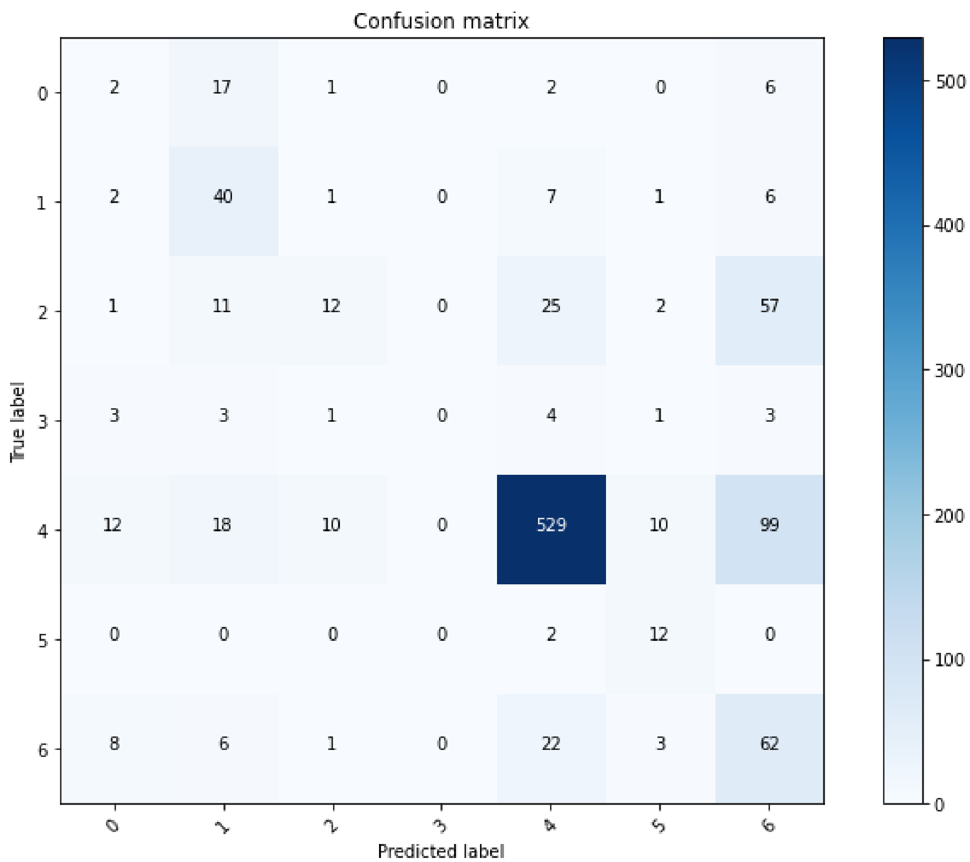


We find that the model with image augmentation is doing well with Training data having high accuracy but we have overfitting with validation data.

Showing the main classification metrics using classification report

	precision	recall	f1-score	support
0	0.08	0.07	0.08	28
1	0.42	0.68	0.52	57
2	0.44	0.10	0.17	108
3	0.00	0.00	0.00	15
4	0.90	0.77	0.83	678
5	0.41	0.86	0.56	14
6	0.27	0.61	0.37	102
micro avg	0.66	0.65	0.65	1002
macro avg	0.36	0.44	0.36	1002
weighted avg	0.71	0.65	0.66	1002
samples avg	0.65	0.65	0.65	1002

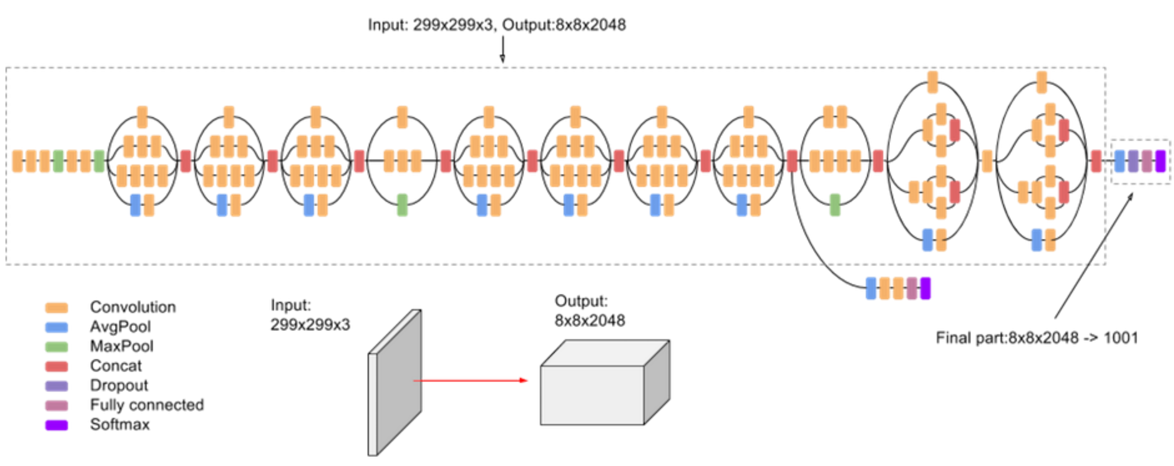
Computing confusion matrix to evaluate the accuracy of a classification.

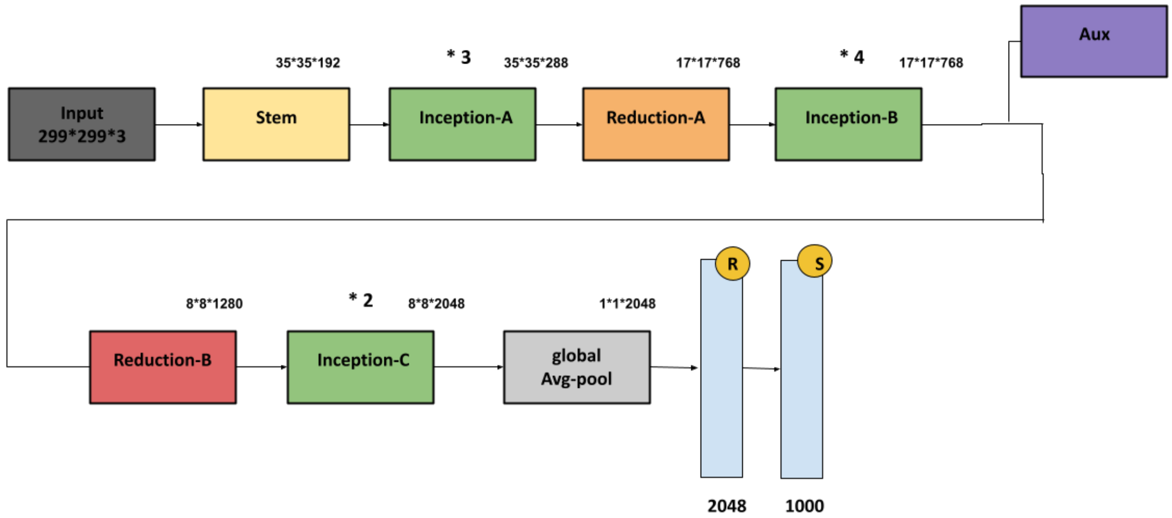


Inception V3

The namesake of Inception v3 is the Inception modules it uses, which are basically mini models inside the bigger model. The inspiration comes from the idea that you need to make a decision as to what type of convolution you want to make at each layer: Do you want a 3×3? Or a 5×5? The idea is that you don't need to know ahead of time if it was better to do, for example, a 3×3 then a 5×5. Instead, just do all the convolutions and let the model pick what's best. Additionally, this architecture allows the model to recover both local feature via smaller convolutions and high abstracted features with larger convolutions.

The larger convolutions are more computationally expensive, so [4] suggests first doing a 1×1 convolution reducing the dimensionality of its feature map, passing the resulting feature map through a Relu, and then doing the larger convolution (in this case, 5×5 or 3×3). The 1×1 convolution is key because it will be used to reduce the dimensionality of its feature map.

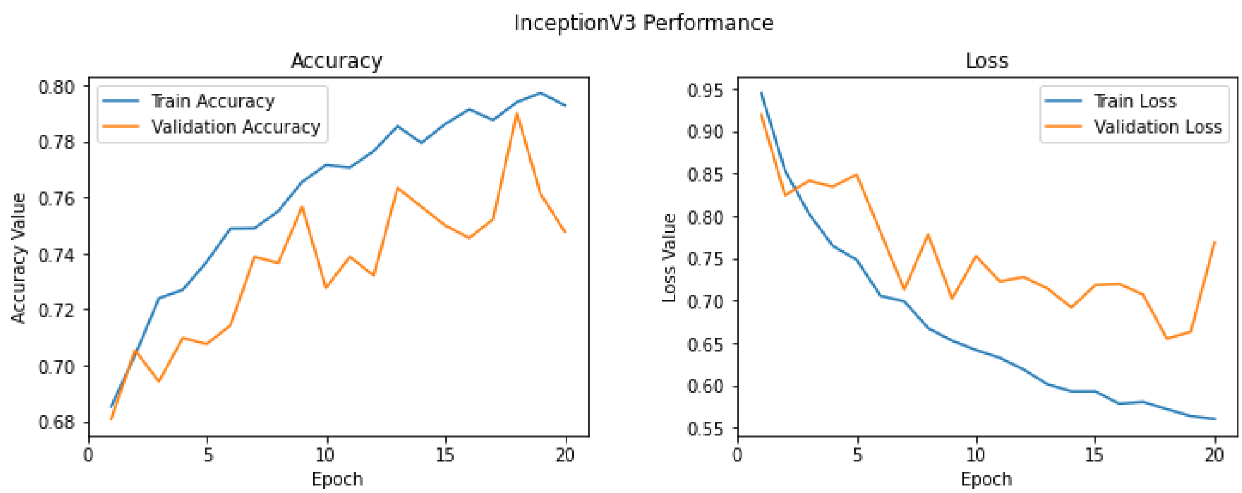




The model give training accuracy of 80% with loss 0.56and test accuracy around 76% with loss 0.66

	loss	accuracy	val_loss	val_accuracy	lr
18	0.5634	0.7974	0.6630	0.7612	2.5000e
17	0.5716	0.7941	0.6549	0.7902	2.5000e
19	0.5599	0.7930	0.7685	0.7478	2.5000e
15	0.5780	0.7915	0.7196	0.7455	2.5000e
16	0.5802	0.7877	0.7070	0.7522	2.5000e
14	0.5927	0.7863	0.7182	0.7500	5.0000e
12	0.6009	0.7855	0.7141	0.7634	5.0000e
13	0.5927	0.7795	0.6917	0.7567	5.0000e
11	0.6184	0.7766	0.7276	0.7321	5.0000e
9	0.6414	0.7716	0.7524	0.7277	1.0000e

Show the changes in accuracy and loss training data and validation data during training process as the epochs progress

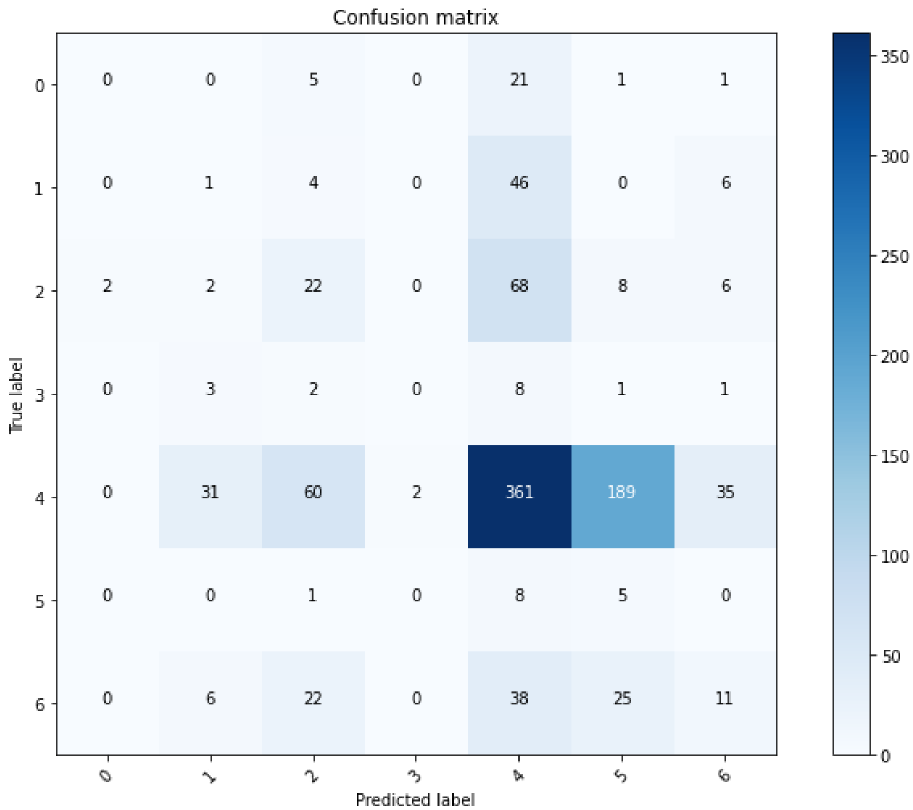


We find that the model with image augmentation is doing well with Training data having high accuracy but we have some overfitting with validation data.

Showing the main classification metrics using classification report

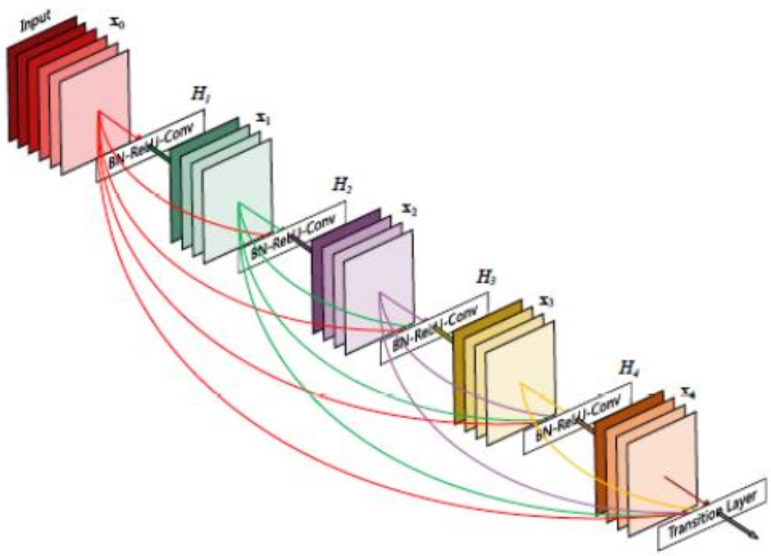
	precision	recall	f1-score	support
0	0.00	0.00	0.00	28
1	0.04	0.02	0.02	57
2	0.20	0.15	0.17	108
3	0.00	0.00	0.00	15
4	0.65	0.45	0.53	678
5	0.02	0.36	0.05	14
6	0.19	0.08	0.11	102
micro avg	0.41	0.33	0.37	1002
macro avg	0.16	0.15	0.13	1002
weighted avg	0.48	0.33	0.39	1002
samples avg	0.33	0.33	0.33	1002

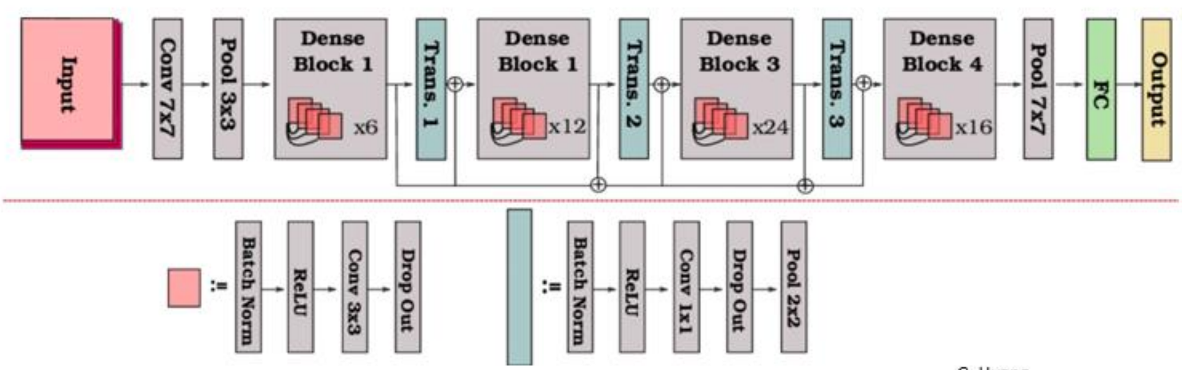
Computing confusion matrix to evaluate the accuracy of a classification.



Dense Net

Dense Net is competitive to Inception V3, but Dense Net has less parameters (approximately 20M compare with approximate 23M of Inception V3). Dense Net 201 has 4 dense blocks. In a dense block, the lth layer has l inputs, consisting of the feature-maps of all preceding convolutional blocks, and its own feature-maps are passed on to all subsequent layers L – l. Each layer reads the state from its preceding layers and writes to the subsequent layer. It changes the state but also passes on information that needs to be preserved. Dense Net architecture explicitly differentiates between information that is added to the network and information that is preserved by concatenating features instead of summing features as in Res Net.



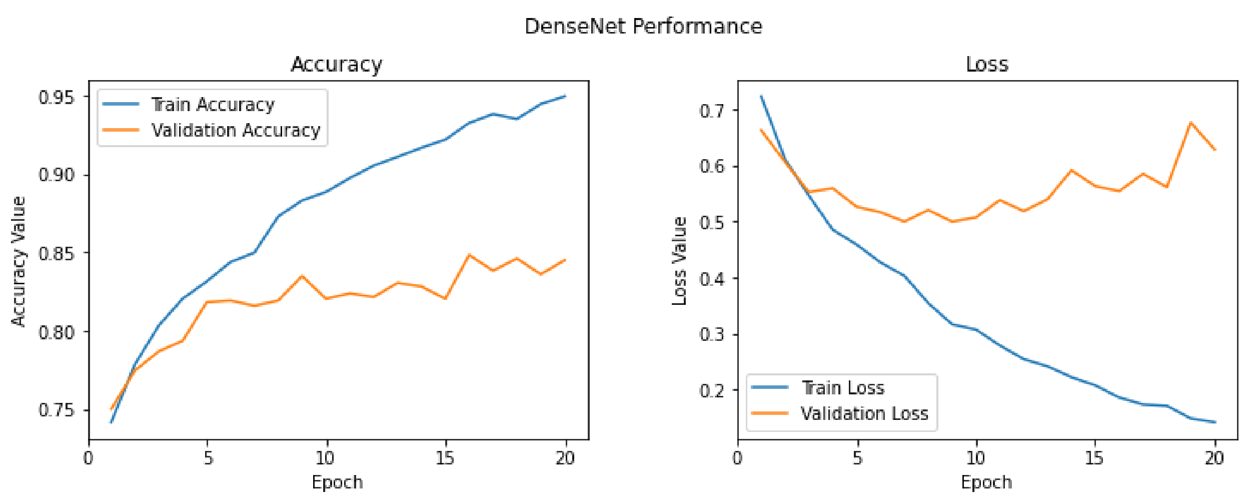


G. Huang

The model give training accuracy of 95% with loss 0.1 and test accuracy around 85% with loss 0.6

	loss	accuracy	val_loss	val_accuracy	lr
19	0.1422	0.9494	0.6285	0.8449	1.0000e
18	0.1490	0.9444	0.6759	0.8359	1.0000e
16	0.1739	0.9380	0.5847	0.8382	1.0000e
17	0.1714	0.9349	0.5612	0.8460	1.0000e
15	0.1862	0.9324	0.5539	0.8482	1.0000e
14	0.2079	0.9218	0.5629	0.8203	1.0000e
13	0.2224	0.9166	0.5911	0.8281	1.0000e
12	0.2417	0.9108	0.5394	0.8304	1.0000e
11	0.2545	0.9052	0.5183	0.8214	1.0000e
10	0.2789	0.8973	0.5379	0.8237	1.0000e

Show the changes in accuracy and loss training data and validation data during training process as the epochs progress

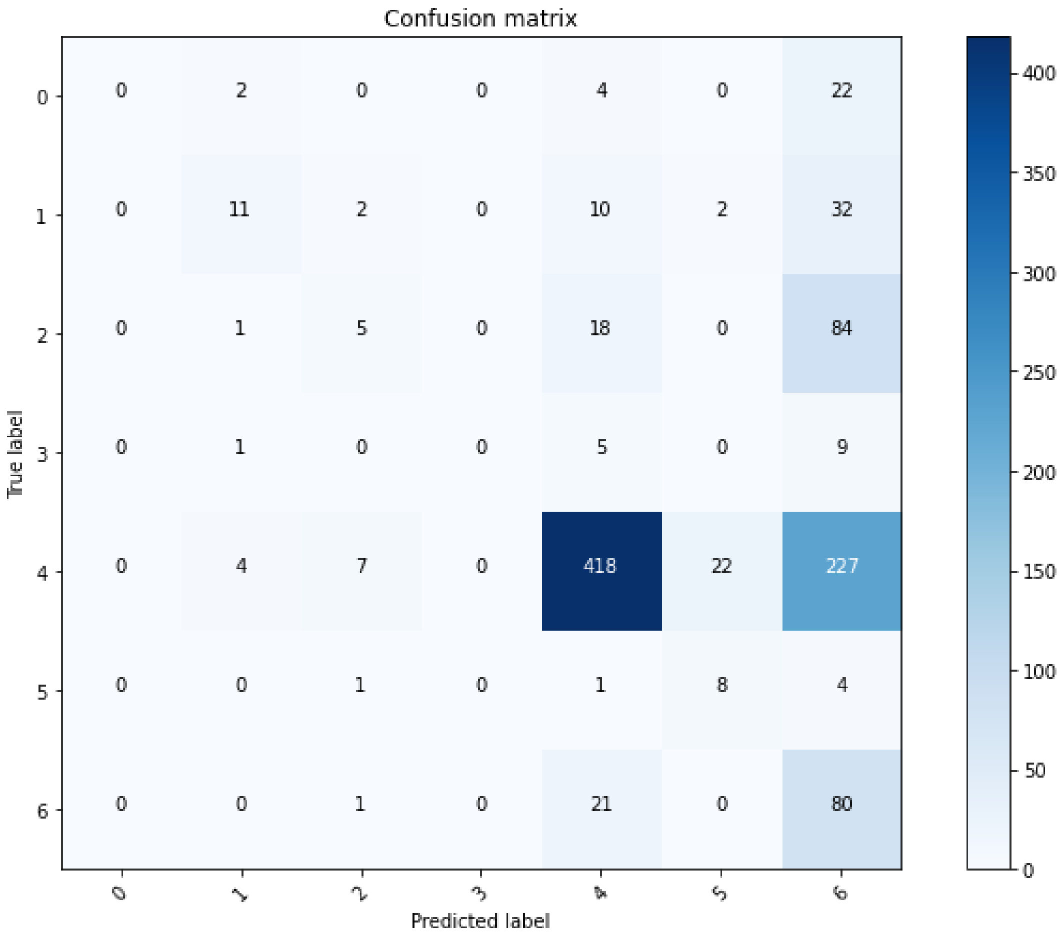


We find that the model with image augmentation is doing well with Training data having high accuracy but we have overfitting with validation data.

Showing the main classification metrics using classification report

	precision	recall	f1-score	support
0	0.00	0.00	0.00	28
1	0.58	0.19	0.29	57
2	0.36	0.05	0.08	108
3	0.00	0.00	0.00	15
4	0.88	0.61	0.72	678
5	0.27	0.57	0.36	14
6	0.18	0.78	0.29	102
micro avg	0.52	0.51	0.52	1002
macro avg	0.32	0.31	0.25	1002
weighted avg	0.69	0.51	0.55	1002
samples avg	0.51	0.51	0.51	1002

Computing confusion matrix to evaluate the accuracy of a classification.


Confusion matrix
True label
Predicted label
0
1
2
3
4
5
6
0 2 0 0 4 0 22
0 11 2 0 10 2 32
0 1 5 0 18 0 84
0 1 0 0 5 0 9
0 4 7 0 418 22 227
0 0 1 0 1 8 4
0 0 1 0 21 0 80
400
350
300
250
200
150
100
50
0

Chapter 5

Results

Model	Accuracy	Loss	Test accuracy	test loss	Depth	Params
Baseline Model	78%	0.56	76%	0.61	11 layers	2,124,839
VGG16	87%	0.36	78%	0.60	23 layers	14,980,935
Inception V3	80%	0.56	76%	0.66	315 layers	22,855,463
Dense Net 201	95%	0.14	85%	0.62	711 layers	19,309,127

When a model is underfit, we are oversimplifying the problem. If you have pegs with the following shape: square, circle, triangle, star, and hexagon, making the hole too big such that all of them fit through would be an example of underfitting. This is also considered error due to bias. Underfit learning curves are often indicated by a flat or decreasing training loss until the end of training, as well as a high training and validation loss.

When a model is overfit, we are overcomplicating the problem. If you have a hole that only allows for the star shape to pass through, that would be one way of thinking about overfitting. Overfit learning curves are often indicated by training loss continuing to decrease with epochs and validation loss decreases and increasing again, where there is an increasing difference between the training and validation loss. In other words, high validation and low training error.

Noisy movements around training loss can be indicative of an unrepresentative validation dataset or one that has too few examples compared to the training dataset.

Best models were decided by lowest validation loss, which ensures that the model is not already overfit such that the training and validation loss start diverging.

Some of the models looked like they could still be further trained as the training loss was still continuing to decrease through the end of the epochs.

Chapter 6

Future work and Conclusion

6.1 Future work and other ideas

In many of the image classification projects correct results depends mostly on dataset. Our dataset contains 10,015 images, but most of those images are of only one class Melanocytic Nevi. It is important to collect a balanced data for better predictions.

Also our dataset has only seven different lesion types whereas in practical there are many more different types of skin cancer. One of the future work is to collect a better dataset with more balanced data with more lesion types.

Also our data only contains cancer images of patients. This model can only tell different types of cancer, but cannot tell whether the lesion itself is malignant or not. Along with the 7 classes if we can get images of non-malignant lesions and add them to our data and then train the model, we can also detect cancer and also tell the type of cancer.

6.2 Conclusion

This work involved a novel method involving deep learning algorithm for skin lesion type identification based on image data. This introduces a new method to identify lesion type before medical diagnosis and helps the medical staff to diagnose the type predicted by the model first. This reduces a lot of time and effort. This model uses a Convolution Neural Network (CNN) within a TensorFlow framework. For the specific cancer types the accuracy ranges from 88% to 100%. And the overall accuracy of the model is 95%.

Compared to traditional medical diagnosis this method is better in two ways. First. This method will not be influenced by diagnostic medical instruments that is error in the instruments can occur during medical diagnosis and this will not have any effect in this method if the images are of good quality. With more and more cancer cases our model becomes more and more powerful unlike medical instruments which has wear and tear.

Second, this method does not depend on stage of cancer and the cancer can be detected even in an early stage. But some medical diagnosis can detect some types of skin cancer only when it reaches advanced stage and this does not occur with this model. If among seven types, we add one more class which is non-malignant cancer type and then train the model, our model can even detect whether it is malignant or not and also tells the type of cancer.

This model may have some limitations like the prediction probability cannot be 100%. But it provides a good basis for further diagnosis and will be very helpful.

References

https://github.com/datascisteven/Melanoma-Image-Classification

https://github.com/ashishpatel26/Skin-Lesions-Detection-Deep-learning

https://www.kaggle.com/code/jagdmir/siim-melanoma-classification-modelling/notebook#Some-insights-about-the-training-and-testing-data

https://www.kaggle.com/code/sid321axn/step-wise-approach-cnn-model-77-0344-accuracy#Step-11:-Model-Building

https://www.kaggle.com/code/jagdmir/all-you-need-to-know-about-cnns

https://github.com/Geeky-star/Skin-Cancer-Detection-App

https://github.com/MoatazMaher1998/Skin-Cancer-Detection-Mobile-App

https://github.com/DefineDan/ml-skin-cancer-classification-app

Vatican 2022