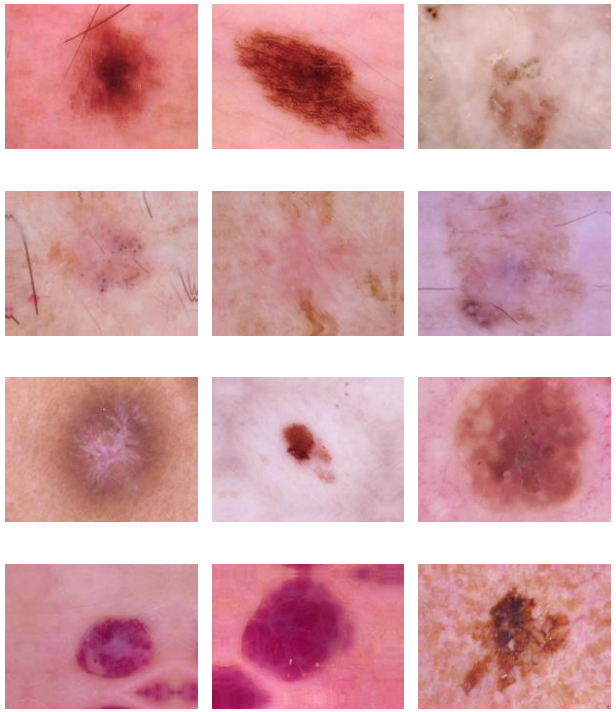
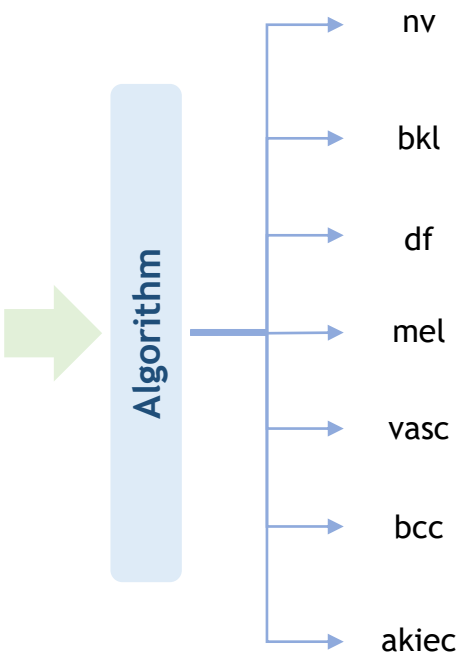
Background

Objective of the algorithm

Skin cancer is one of the most common forms of cancer. Research shows that 1 in 5 Americans will develop skin cancer by the age of 70. Identifying it early and receiving appropriate treatment can stop the spread of a malignant skin lesion.

The algorithm aims to classify images based on 7 categories of skin lesions and help the user identify the severeness of the skin lesion so that they can make an early judgement on the required treatment





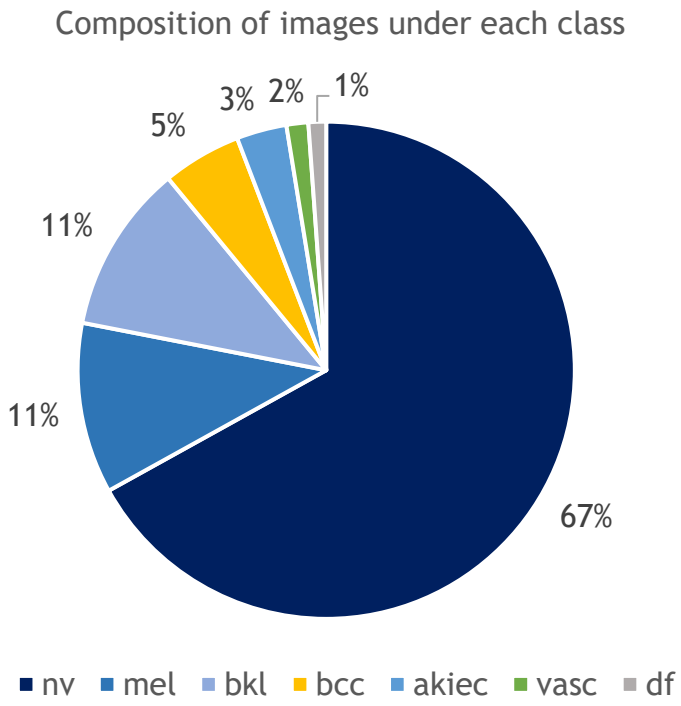
Understanding the types of skin lesions

Type	Description
nv	A common benign skin lesion due to a local proliferation of pigment cells. Patients with multiple nv appear to have an increased lifetime risk of melanoma, where risk increases with the size and number of nv
bkl	Non-cancerous skin growths that some people develop as they age. It often appear on the back or chest but can occur on any part of the body. It grows slowly, in groups or singly. Most people will develop at least one during their lifetime
df	Common cutaneous nodule of unknown etiology that occurs more often in women. Frequently develops on the extremities (mostly the lower legs) and is usually asymptomatic, although pruritus and tenderness can be present
mel	A form of skin cancer that begins in the cells (melanocytes) that control the pigment in your skin. It is the most dangerous form of skin cancer.
vasc	Relatively common abnormalities of the skin and underlying tissues, more commonly known as birthmarks.
bcc	A form of skin cancer in the basal cells — a type of cell within the skin that produces new skin cells as old ones die off. Although rare, it can spread and metastasize to other parts of the body, such as the bones and lungs and even cause death.
akiec	Dry scaly patches of skin that have been damaged by the sun. The patches are not usually serious. But there's a small chance they could become skin cancer, so it's important to avoid further damage to your skin

Sources: Skin Cancer Foundation

Preparing the underlying dataset

Issue: The original dataset was imbalanced



An imbalanced dataset can render the learnings of the algorithm useless, since it will most probably learn to predict the modal class. In that case the algorithm's prediction accuracy will not be a useful measure to judge its performance, and in fact can be a misleading measure. Hence this issue will need to be addressed prior to training the algorithm to achieve its stated objective

Solution: Creating an artificially balanced dataset

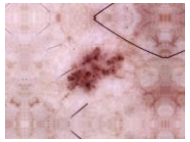
1 **Identifying the imbalance: Shows the number of images that need to be created under each imbalanced class**

Class	Number of images needed to match the majority class ('nv')
bkl	4951
df	4464
mel	5088
vasc	4453
bcc	5268
akiec	5242

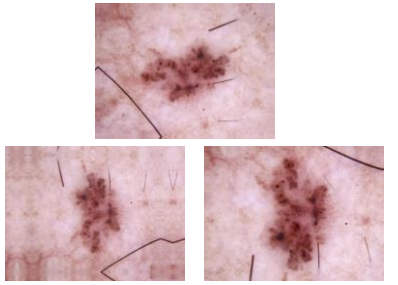
2 **Populating the imbalanced classes by artificially creating augmented images using the existing images under each class**

Original Image

New images added to the class



Artificial augmented image generation



The Custom Convolutional Neural Net algorithm

Stage 01: Setting up the underlying model architecture (base model)

Input Image

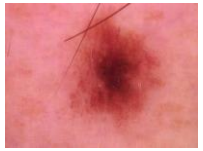
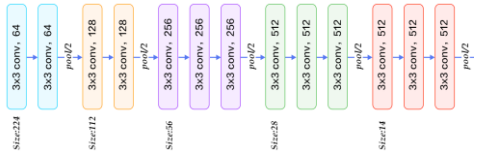


Image Augmentation

VGG-16: Excluding FC[1] Layers (Transfer learning)



Max Pool Layer → **Flatten Layer** → **Dense Layer (128 units, ReLu)** → **Dropout Layer (20%)** → **Dense Layer (7 units, Softmax)** →

Output vector

nv	bkl	df	mel	vasc	bcc	akiec
1	0	0	0	0	0	0

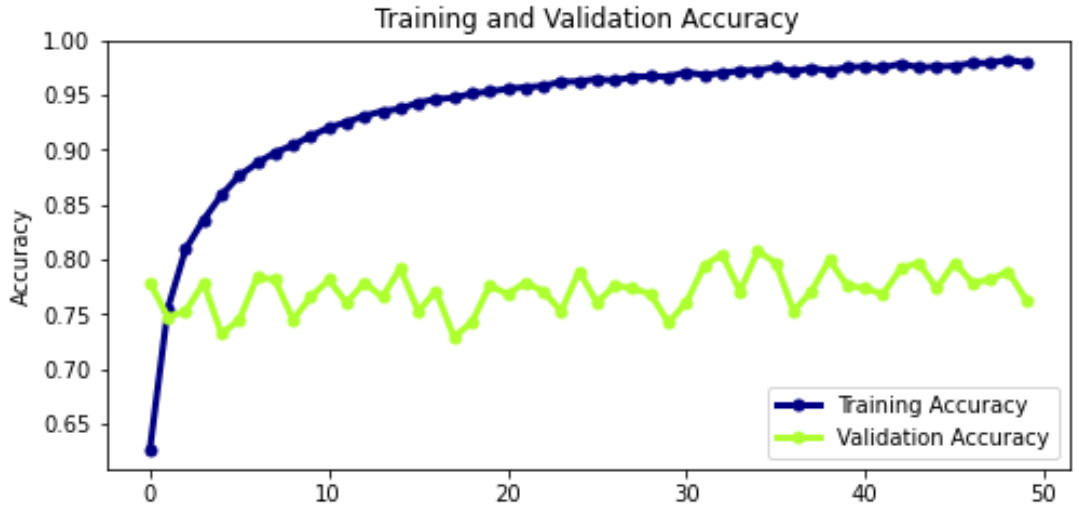
1 Assuming the algorithm works well, an input image which belongs to the 'nv' class will be classified by the algorithm as 'nv' and the output will be represented as a vector with 7 dimensions. Each dimension represents a class and in this case the dimension corresponding to 'nv' will take the value 1 and the other dimensions will take the value 0

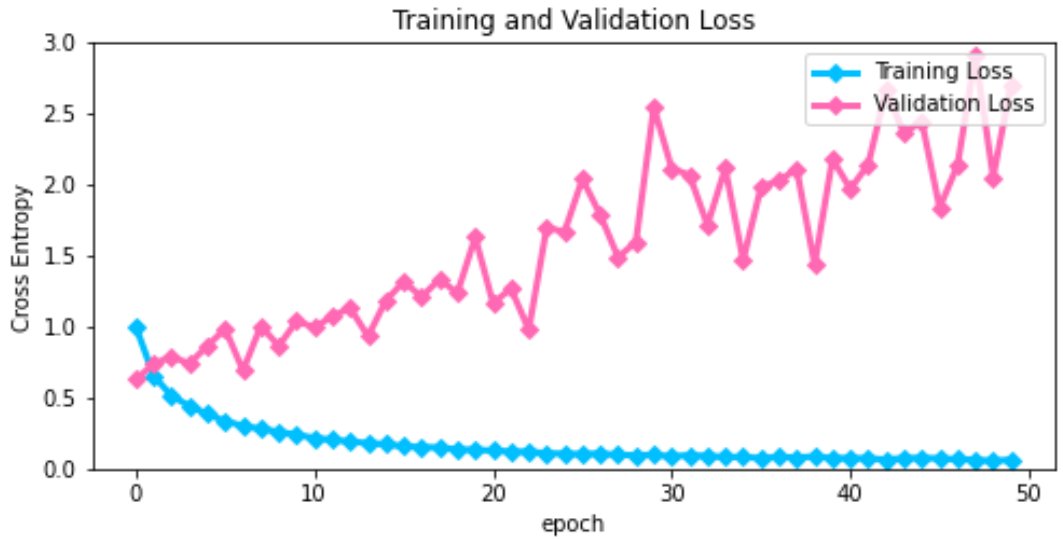
2 A pre-trained version of the VGG-16 model was used as a building block of the custom CNN algorithm. This is generally known as 'Transfer Learning' and can be useful in image classification problems. The first seven trainable layers were frozen, and the last 6 trainable layers were re-trained to fit the dataset.

Footnotes: 1) FC – Fully Connected

The Custom Convolutional Neural Net algorithm

Stage 02: Training the algorithm on the Training set and validating performance on the Validation set





Stage 03: Converting the output of the trained base model into a vector with probabilities

Input Image belonging to the 'nv' class

Trained base model

Output

nv	bkl	df	mel	vasc	bcc	akiec
1	0	0	0	0	0	0

Softmax Layer

New Output

nv	bkl	df	mel	vasc	bcc	akiec
0.55	0.4	0.3	0.3	0.33	0.41	0.42

Vector with the probability of the image belonging to a given class

The Custom Convolutional Neural Net algorithm

Stage 04: Evaluating the output (Top 2 Accuracy)

Input Image belonging to the ‘nv’ class

CNN

Output probability vector

nv	bkl	df	mel	vasc	bcc	akiec
0.55	0.4	0.3	0.3	0.33	0.41	0.42

Select the top two probability classes

nv	akiec
0.55	0.42

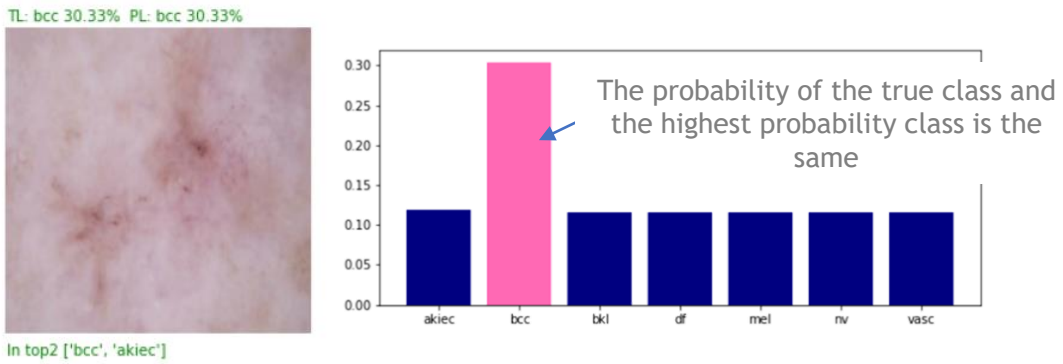
True Label in Top 2 classes = Classified correctly

True Label not in Top 2 classes = Classified incorrectly

Understanding the output and Top 2 Accuracy computation

Output Type 01

The True label and predicted label are the same

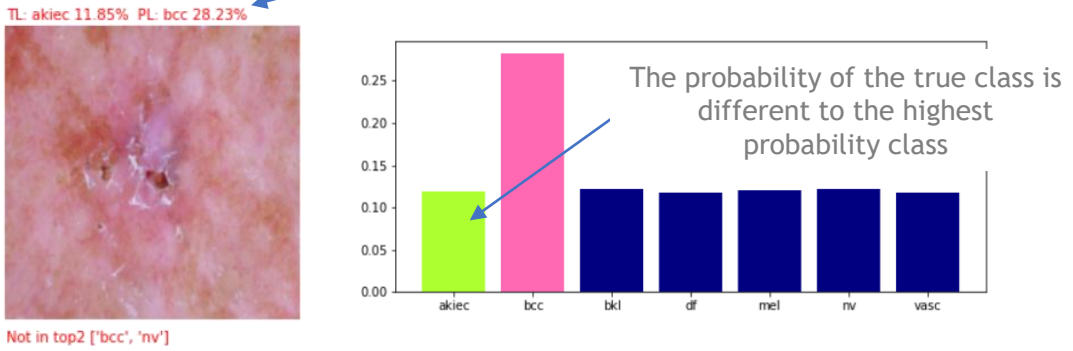


The probability of the true class and the highest probability class is the same

True label in Top 2 classes

Output Type 03

The True label and predicted label are different

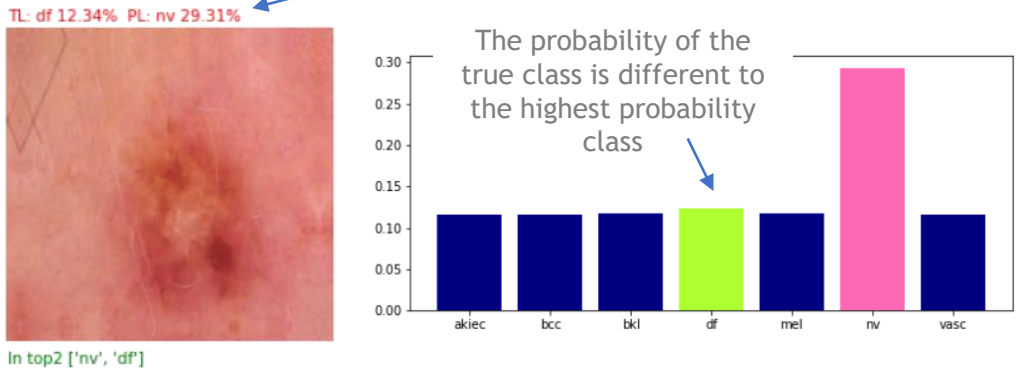


The probability of the true class is different to the highest probability class

True label not in Top 2 classes

Output Type 02

The True label and predicted label are different



The probability of the true class is different to the highest probability class

True label in Top 2 classes

Top 2 Accuracy computation

- **Output Type 01 and 02 are considered to be correctly classified since the true label is in the Top 2 classes**
- **Output Type 03 is considered to be incorrectly classified**

Top 2 Accuracy

The Algorithm was able to achieve a c. 89% Top 2 Accuracy when evaluated on the Test Set