

Melanoma Detection - Case Study

Author: **Tom Mathews**

Overview

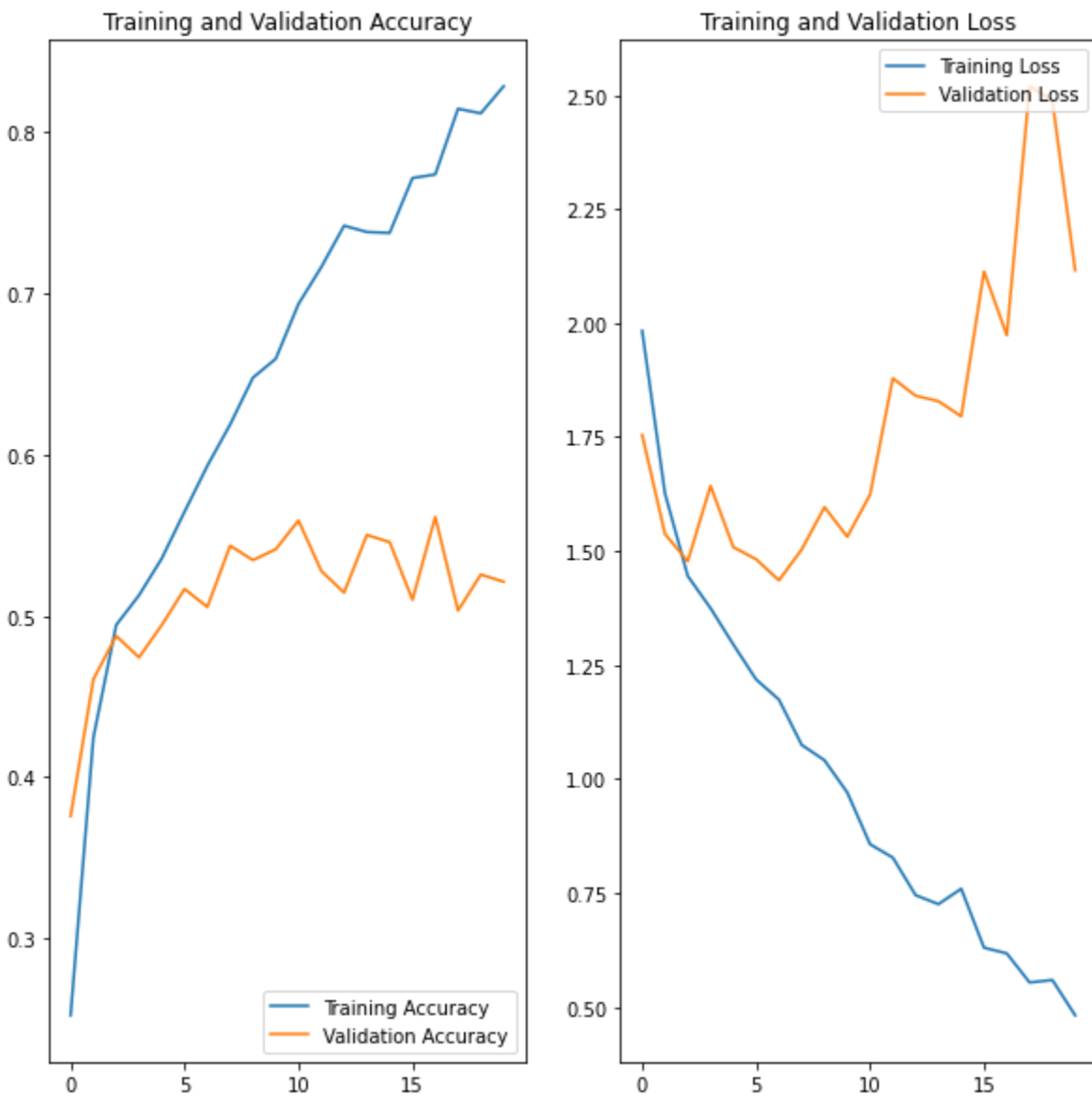
Basic Model

For the basic model, the first layer in the model is a Rescaling layer that rescales the image by 255. We use the original dataset for this model. I am using a 3 CNN unit structure. All CNN Layers have 1 Conv2D layer followed by MaxPooling2D layer. The CNN units have an increasing number of filters going from 32 to 128. After the CNN units, we have a flatten layer and then a Dense layer with 9 (number of classes) layers.

Model: "sequential"

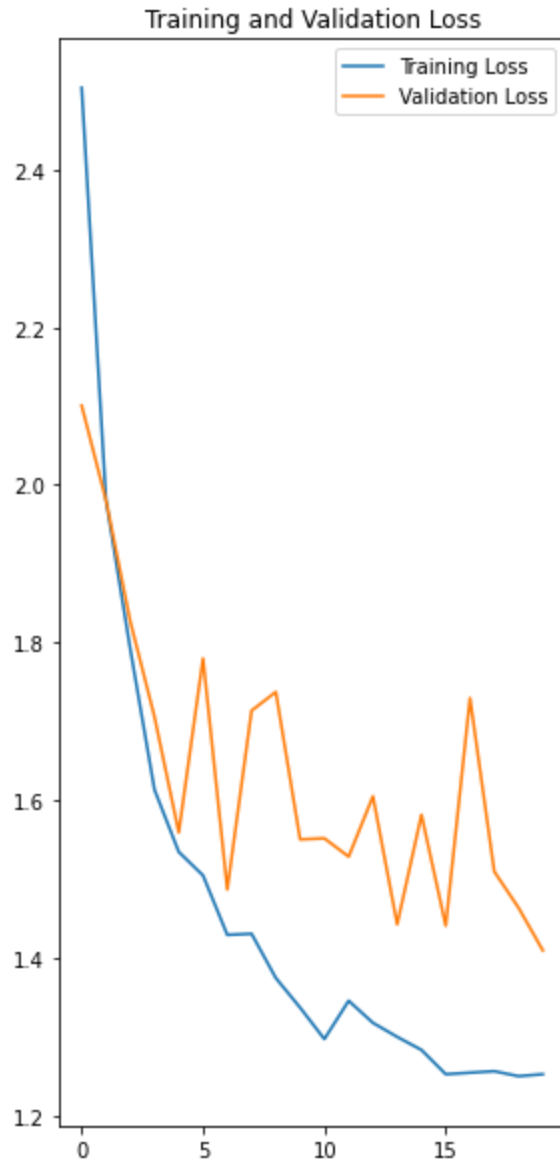
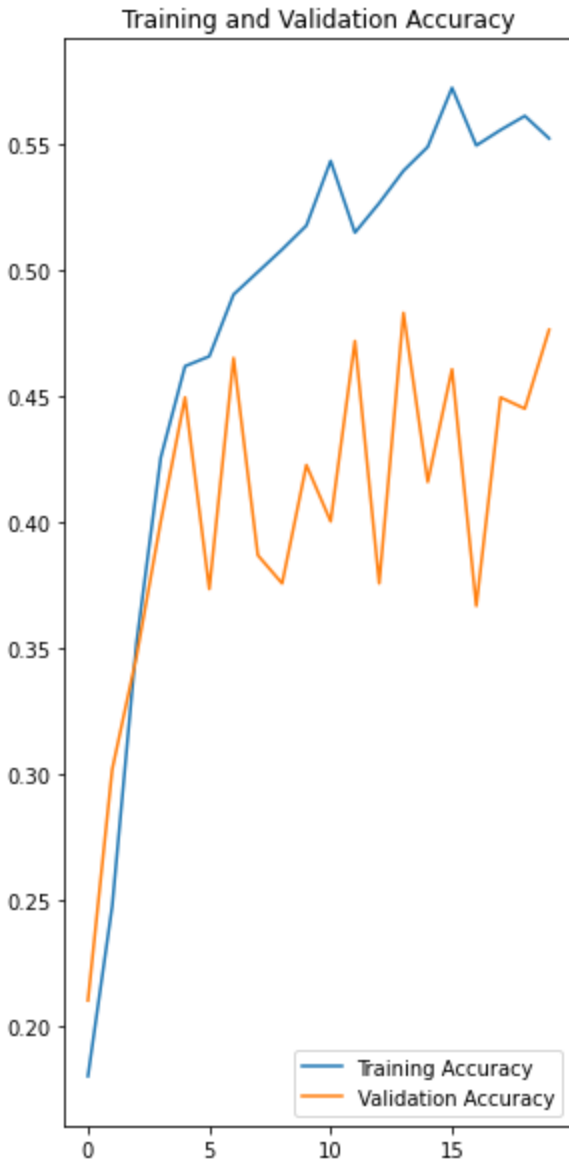
Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 180, 180, 3)	0
conv2d (Conv2D)	(None, 180, 180, 32)	896
max_pooling2d (MaxPooling2D)	(None, 90, 90, 32)	0
conv2d_1 (Conv2D)	(None, 90, 90, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 45, 45, 64)	0
conv2d_2 (Conv2D)	(None, 45, 45, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 22, 22, 128)	0
flatten (Flatten)	(None, 61952)	0
dense (Dense)	(None, 9)	557577
Total params: 650,825		
Trainable params: 650,825		

Non-trainable params: 0



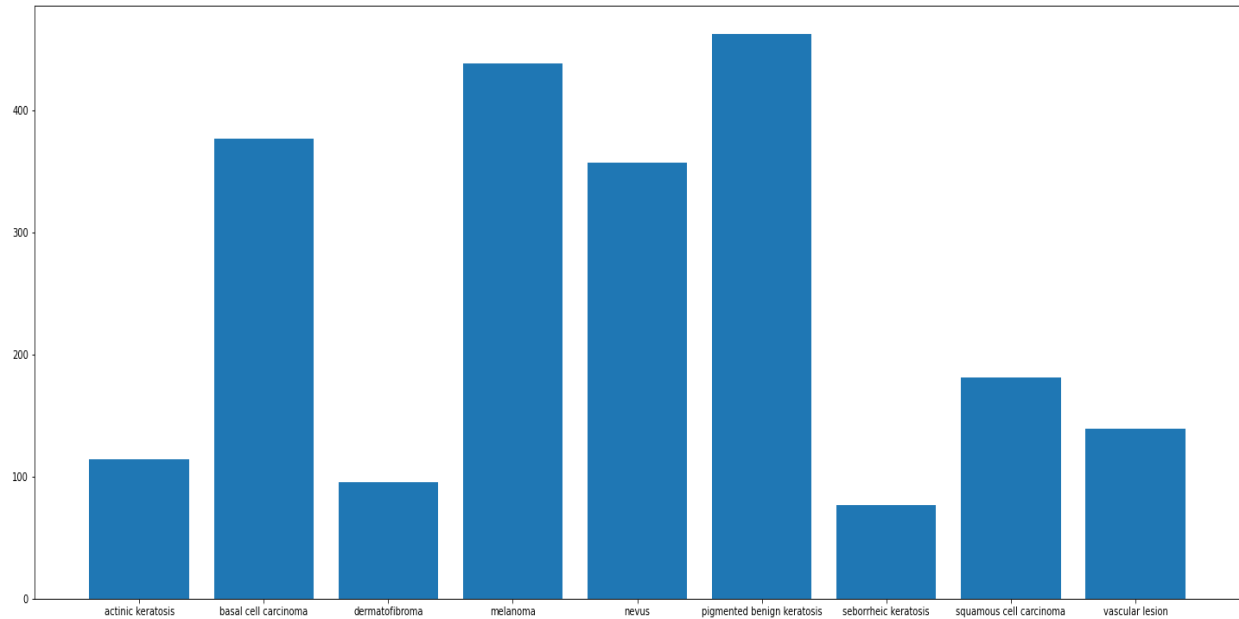
Data Augmentation and Drop Outs

In this case, we update the basic model and add dropouts after each CNN Unit. We can see adding dropout layers reduces the over fitting issue.



Distribution in the Dataset

We can visualise the initial distribution of the dataset and observe that there is data imbalance between the classes. It would be a good idea to augment the data so that we reduce the sparsity of the dataset.



Data Augmentation

We add 500 augmented images to each class to reduce the sparsity of the classes with lower numbers of data. The updated number of images in each class is as shown below:

pigmented benign keratosis	962
melanoma	938
basal cell carcinoma	876
nevus	857
squamous cell carcinoma	681
vascular lesion	639
actinic keratosis	614
dermatofibroma	595
seborrheic keratosis	577

BatchNormalisation

We add a batch normalisation layer to improve the model performance.

Model: "sequential_3"

Layer (type)	Output Shape	Param #
rescaling_2 (Rescaling)	(None, 180, 180, 3)	0

batch_normalization (Batch Normalization)	(None, 180, 180, 3)	12
conv2d_6 (Conv2D)	(None, 180, 180, 32)	896
batch_normalization_1 (Batch Normalization)	(None, 180, 180, 32)	128
max_pooling2d_6 (MaxPooling2D)	(None, 90, 90, 32)	0
dropout_3 (Dropout)	(None, 90, 90, 32)	0
conv2d_7 (Conv2D)	(None, 90, 90, 64)	18496
max_pooling2d_7 (MaxPooling2D)	(None, 45, 45, 64)	0
dropout_4 (Dropout)	(None, 45, 45, 64)	0
conv2d_8 (Conv2D)	(None, 45, 45, 128)	73856
max_pooling2d_8 (MaxPooling2D)	(None, 22, 22, 128)	0
dropout_5 (Dropout)	(None, 22, 22, 128)	0
flatten_2 (Flatten)	(None, 61952)	0
dense_2 (Dense)	(None, 9)	557577

```

=====
Total params: 650,965
Trainable params: 650,895
Non-trainable params: 70

```

