# Image Classification

Module: Image Processing
&
Deep Learning

May 2020

# Table of Contents.

# 1. Introduction - Objective

My objective for this report is to produce image classifiers which are able to determine the genre associated with the images which have come from film and T.V shows. For the purpose of this report, I have decided to use three genres, these are, Romance, Action and Anime. I feel these are distinctly different genres with different themes associated with the images from these genres. I hope to see whether these models are able to pick up on the patterns which are displayed within the genres from the images provided.

# 2. Data Collection & Cleaning

The collection of images was accomplished through the use of the Bing image search api from Microsoft Azure. This allowed me to search for images related to the genres I was using, for example ‘Action’, I would use search terms such as ‘fighting movie scenes’ or ‘war movie scenes’. This allowed me to amass around 2000 images for each genre. Figure 1 shows examples of the images collected for each genre.


*Figure 1. Image Genre Examples*

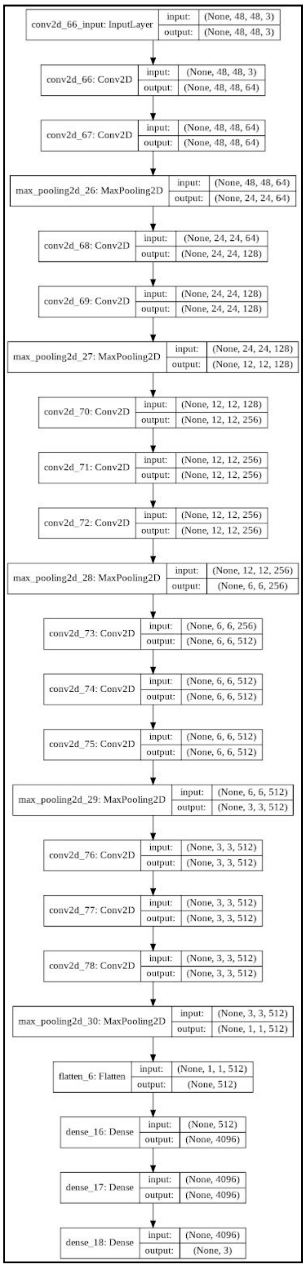
Cleaning the image dataset involved passing my images through a hashing algorithm to detect and remove duplicate images, as our models could become biased and begin overfitting due to learning patterns through the presence of duplicate images, making it harder to generalise the models to unseen data. After this I manually went through all the images, determining their appropriateness for said genre, as some images were not related to that genre. After this left me with 1000 images for Action, 1200 for Romance and 1800 for Anime.

# 3. Data Preprocessing

For each class of images, I cut the amount to 1000 for each class, to form a balanced dataset, which mitigates the effect of biases in the models. All the images were resized to 96x96. This size is small enough where we can still see distinct features and reduce the number of parameters which need training. For the train/val/test split, I've decided to begin with a 70/15/15 split then explore further variations which will be 80/10/10 and 60/20/20. Also normalisation applied to all images putting pixel range between 0 and 1, to improve training.

# 4.Models

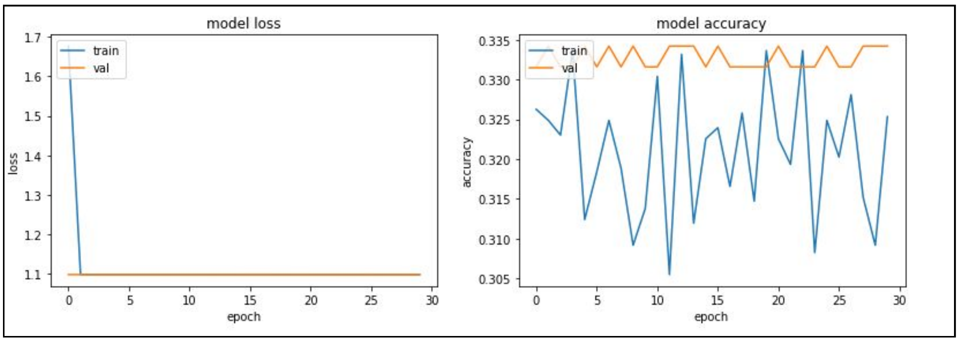
## 4.1 VGG16




The first model, is based up on the VGG16 architecture, which is a deep convolutional neural network, consisting of 12 convolutional layers, 4 max pooling layers followed by four fully connected layers, then in my version theres is a 3 - way softmax classifier as we there are only 3 classes in my dataset. Having a deep network will allow us to extract more features but to a certain extent as it could cause the model to overfit, especially since my dataset is not very big making a smaller version of this network may be required. This would be useful for my dataset as it could discover the romance and action genre are similar as they both have people in the photos, so it's important to key into the details which make them different.

### 4.1.2 Base Model

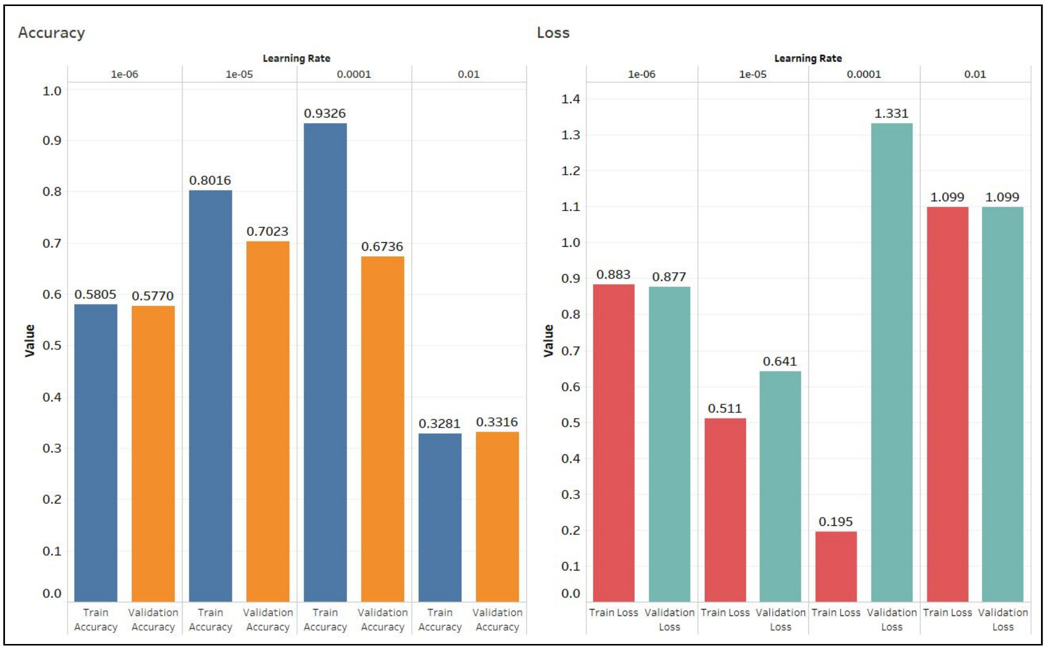
To begin, I have constructed a base model which takes in 96x96 images, which will be fit using the adam optimiser with the default learning rate of 0.01, with a relu activation function in all conv layers, and for the loss will be using the categorical cross entropy. This will be our baseline which we compare our improvements too. The data split initially is 70/15/15.



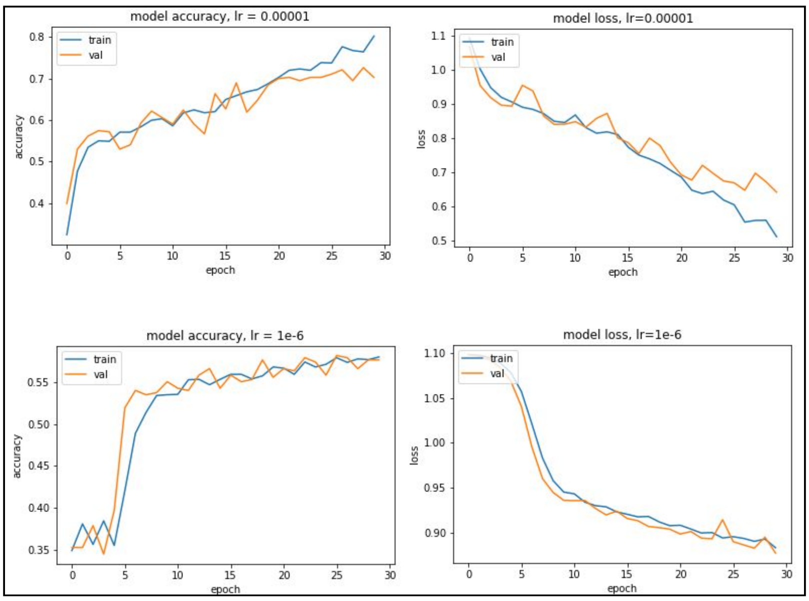

After training the model for 30 epochs, the train and validation loss immediately dropped to a constant loss. The accuracy for the base model is fluctuating up and down, this shows our model is very unstable during our training process, and is not learning,and must be outputting spurious results. To improve this model the next parameter, I should investigate is the learning rate, as the learning rate which could be too large can cause our model to converge to a suboptimal solution, which has been displayed in the graphs above, due to the rapid decrease in loss,hence the updating of weight must be too large.

## 4.1.3 Learning Rate

I investigated the learning rate parameter of my model, as the default parameter for the Adam optimizer 0.001 was possibly too large causing no learning.



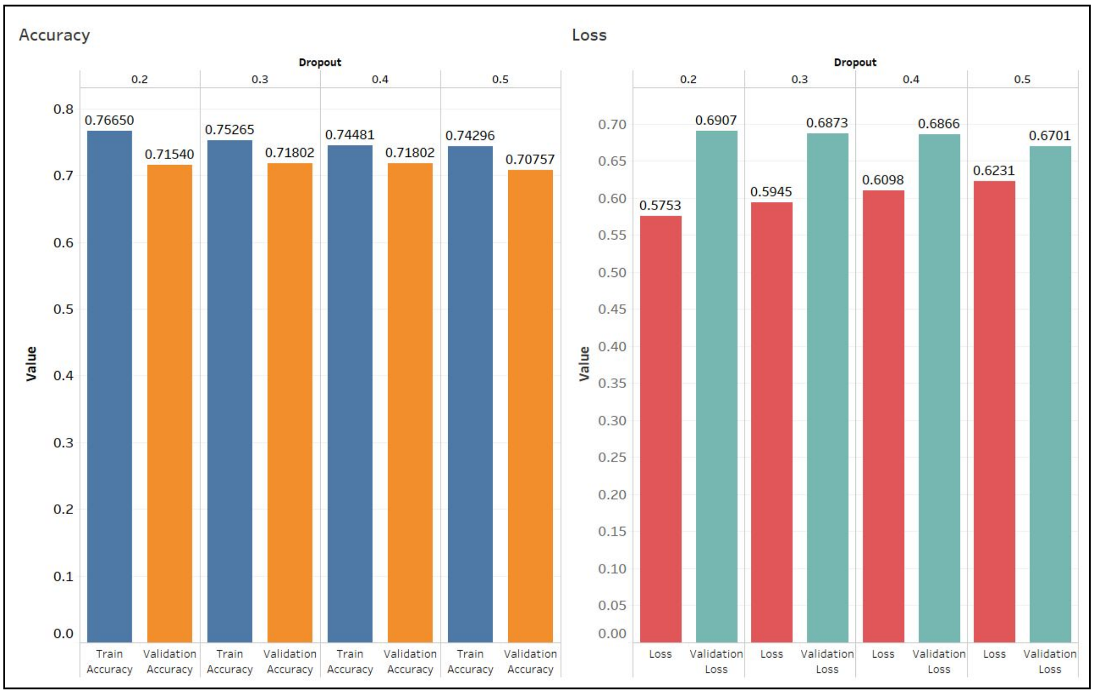

To investigate whether having large learning rates may have affected model performance, I increased the LR (learning rate) to 0.01. Model performance still suffered, similar performance to the default LR of 0.001. Decreasing the LR to 0.0001 massively improved performance as the model was consistently learning, at the end, it achieved a very high accuracy and very low loss compared to the validation accuracy and loss but the model at this learning rate is massively overfitting if you refer to the graph above we can see how large this difference is. The most interesting results were when the learning rate was adjusted to 1e-06 and 1e-05.



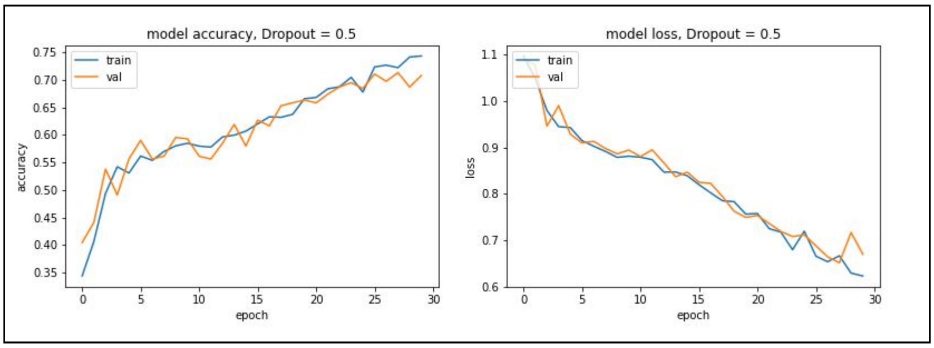

Decreasing the learning rate to 1e-5, produced positive results for accuracy and loss, because the model was only slightly overfitting compared to the validation set hence with further improvements we could fix this overfitting. The model achieved an 80 percent accuracy and a 70 percent validation accuracy. On the other hand, decreasing the LR even further  to 1e-6 produced a model with a very similar accuracy and loss with the validation set which show very little sign of under/over fitting, but on closer look it appears to plateau towards the end of training time, showing little improvement compared to the LR of 1e-5 also producing a lower accuracy score. After comparison,there is room for improvement with the learning rate of 1e-5, so I will carry on forward with this value as other improvements to the model are made. As this is quite a large model there is a higher chance of overfitting occurring with a small dataset, so the next step is to explore the use of the dropout layer to curb the effect of overfitting.

### 4.1.4 Dropout

Using a large model gives it larger capacity to learn and memorise a smaller dataset (3000 images), therefore the model is more prone to overfitting. So by using dropout more individual nodes to learn more about the characteristics of the network we are training. So in the model I have developed thus far, I have added two dropout layers, each after the dense layers in the final part of the model. I have decided to iterate through dropout values between 0.2 and 0.5 to decide the optimal value for these layers.

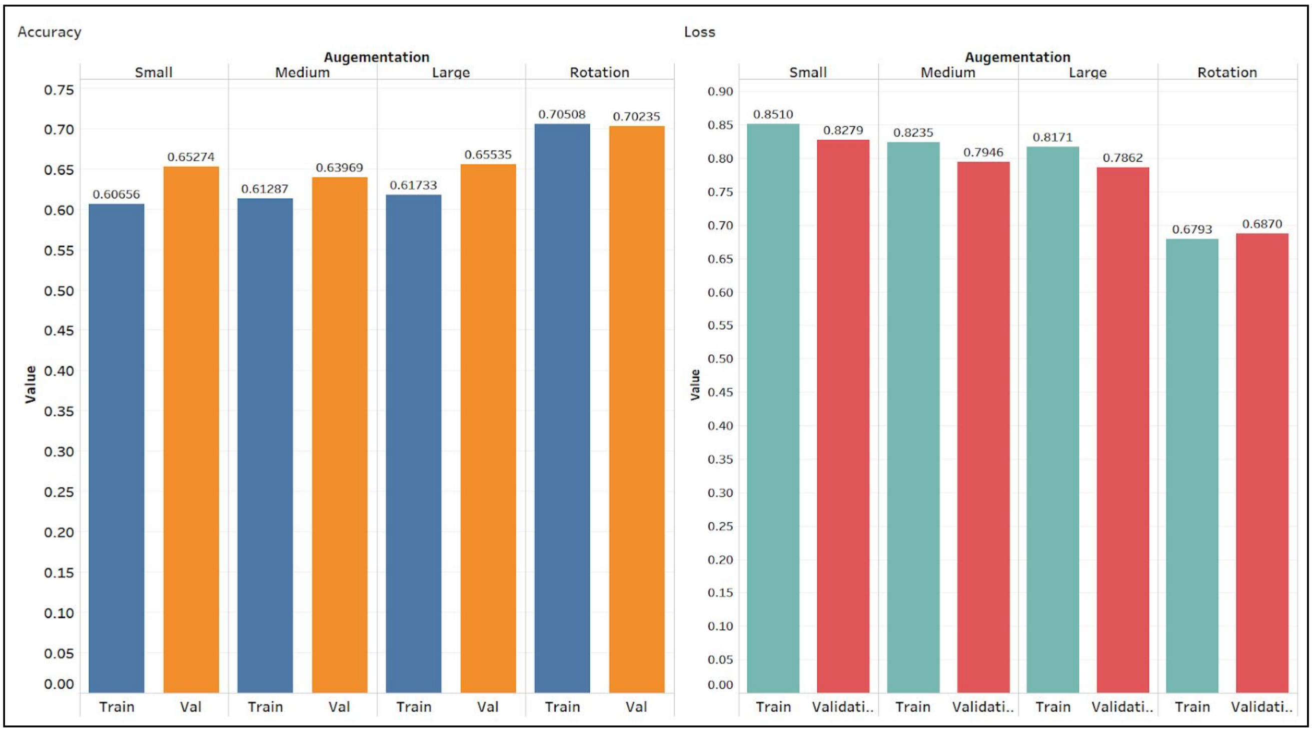



Applying dropout at all values specified all showed a positive improvement in regards to overfitting, and the maintained a validation accuracy of over 70 percent throughout. Particularly at the 0.5dropout achieved a minimal difference in terms loss and validation loss, and second smallest difference in accuracy, proving to be the most valid value we should use as we try to decrease the validation as much as possible. So 0.5 will be the value used going forward. Looking at the graph below the model showed continued performance and no signs of plateuning.

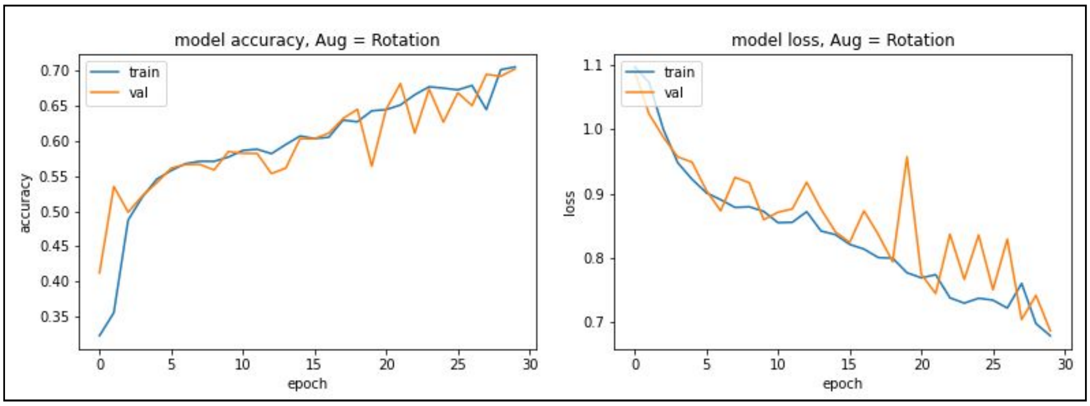



## 4.15 Data Augmentation

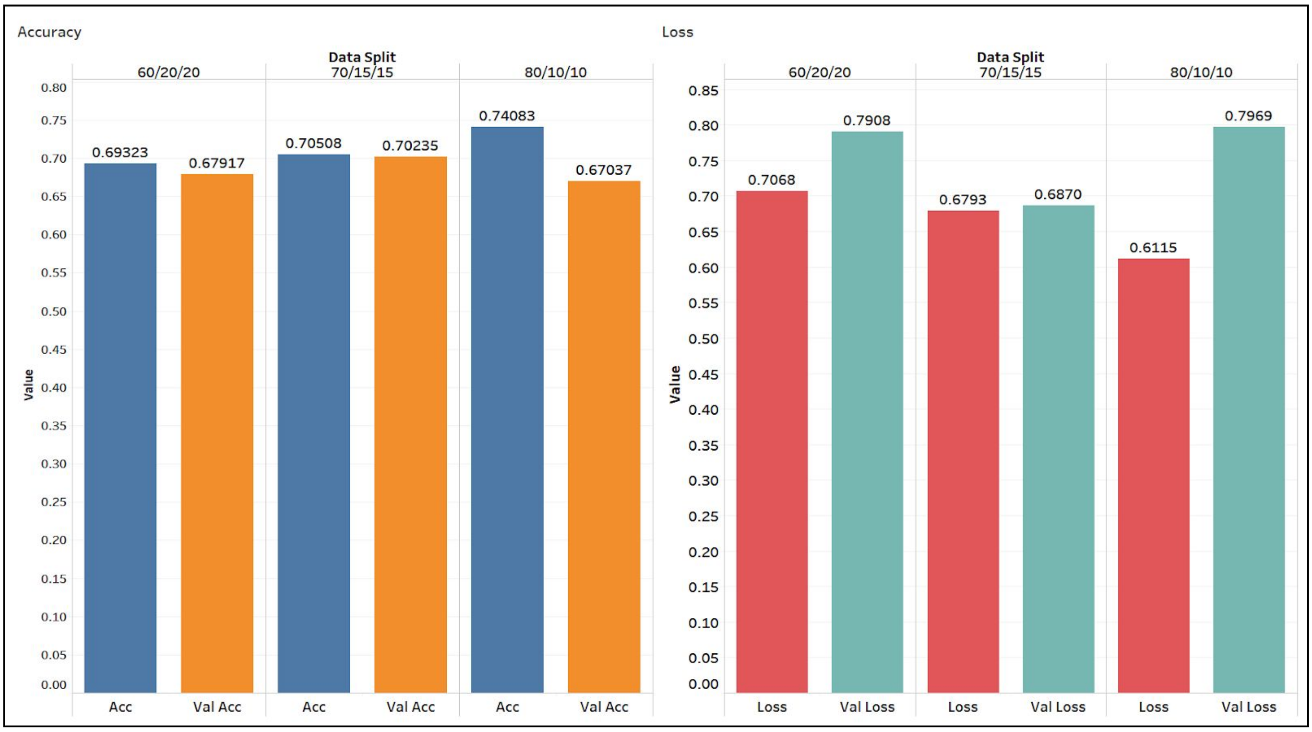
Data augmenting is one way of training the model so that it is able to generalise better to new incoming data, especially when dealing with a smaller dataset, as it doesnt give the model a chance to just memorise our training dataset. For this model I augmented the images in 3 different times with varying quantities of change occurring.  Augmentations such as shift, rotate,shear and zoom were used to augment the images.




Applying augmentation to the model at varying amounts produced several models which were underfitting, augmentation is used to prevent overfitting and model before was slightly overfitting. This led to the use of only using rotation as the augmentation on the training set which produced a model with a loss and accuracy with a very small absolute difference compared to the validation set hence, showing little sign of under/overfitting.




## 4.1.6 Train Validation Test Split




Testing the data splits I had earlier proposed, it can be seen that the current split which I was using is the most optimal one as it achieved minimal loss, with little signs of under/over fitting. Second best was the 60/20/20 split which also received similar accuracy but from the loss we can tell there is sign of overfitting occuring same with the 80/10/10 split there is even more sign of overfitting.

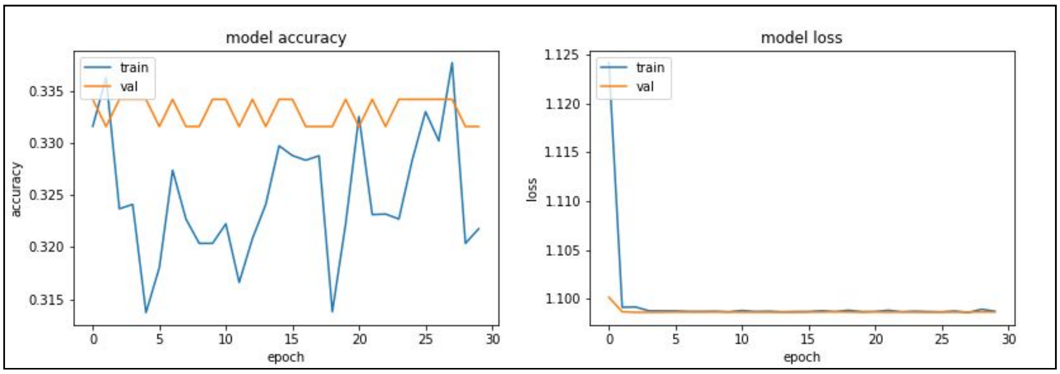
## 4.1.7 Test Results


Applying our final model to our test dataset with a structure of 12 convolution layers, 4 max pooling layers, 2 dropout (0.5) layers, 2 fully connected layers plus a softmax layer, learning rate = 1e-5, optimiser = Adam and a data split of 70/15/15. We achieved a test accuracy of 66 percent, given the nature of the dataset, and how abstract the images are this is a fair performance. Looking above brightly coloured images were likely to cause more confusion for the model to differentiate between the Romance and Anime genre. Further Improvements would be to use higher res images as this would allow more features to be extracted, maybe allowing an increase in performance.

## 4.2 AlexNet

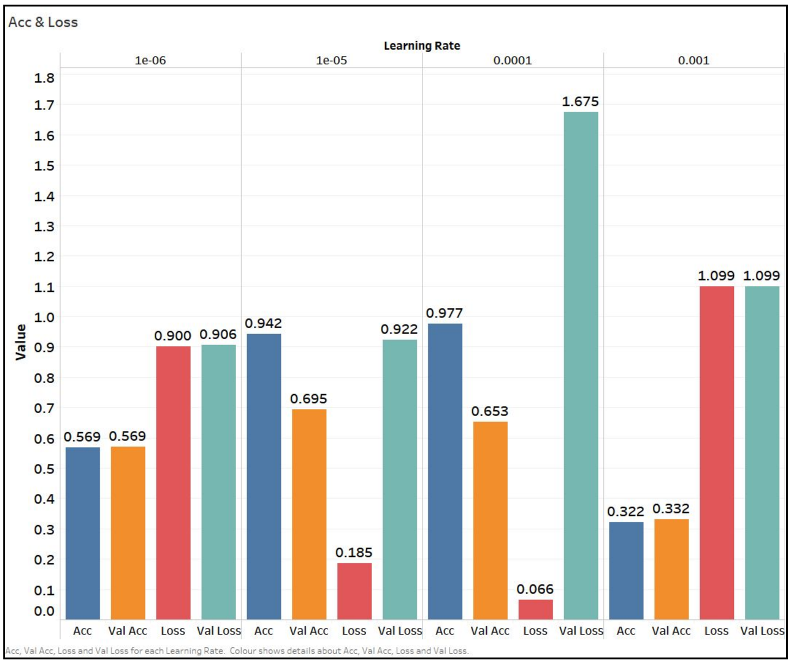
The alexnet consists of 5 convolutional layers and 3 full connected layers and 3 maxpool layers. It is a smaller network which may be more appropriate for the small sized dataset used.

### 4.2.1 Base Model



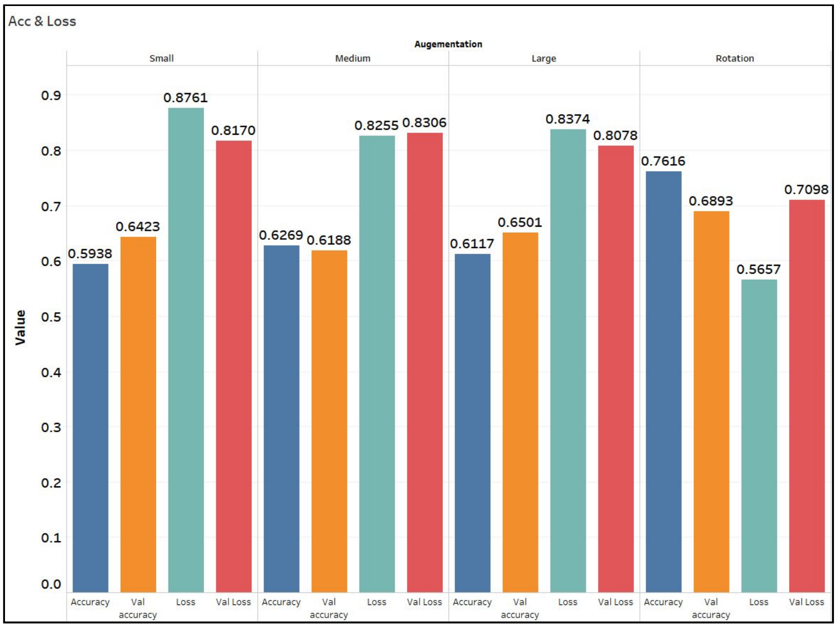

The model here is not learning just like the previous model as the loss drops to a constant value, so investigating the learning rate will be required.

## 4.2.2 Learning rate

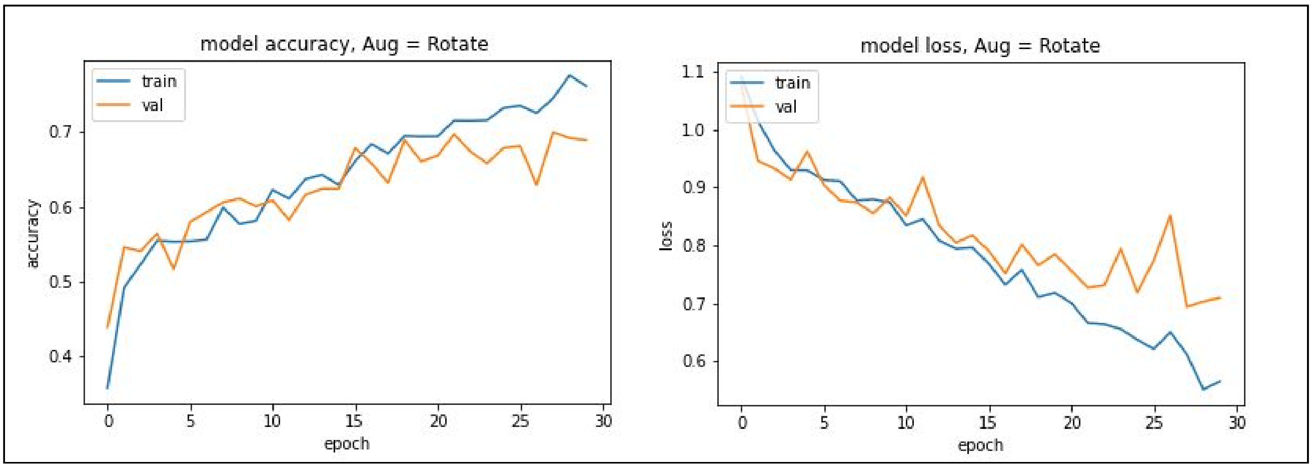



Decreasing the LR of the model produced allowed it to learn, rapidly improving accuracy and loss. But there is a large amount of overfitting occuring at LR = 1e-4, and LR= 1e-5. LR = 1e-5 is the value I will proceed with as it reduces the val loss the most, and with further overfitting techniques can produce a more robust model.

## 4.2.3 Data Augmentation

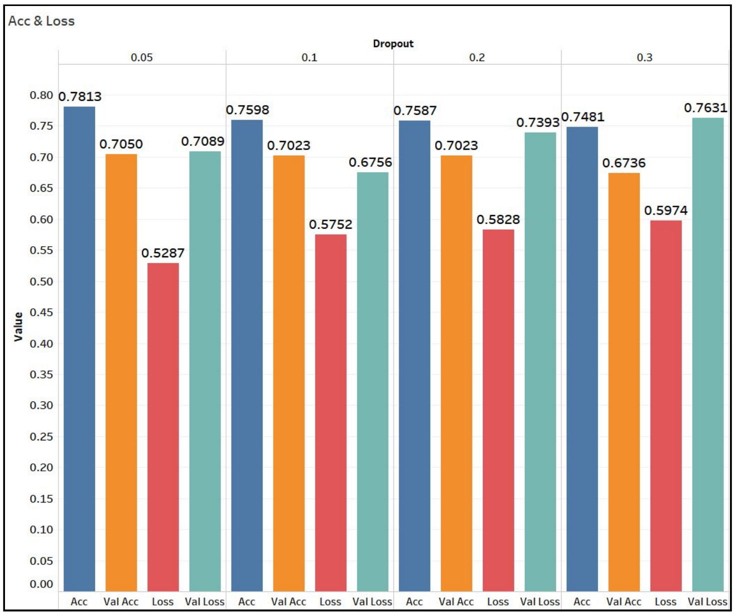



Applying the various augmentations, at different strengths, produced two models which underfit those were small and large. Medium augmentations produced a model which did not overfi, but the model as whole had a reduction in performance. Applying only rotational augmentations allowed the model to reduce loss the most although still overfitting we can apply additional techniques to curb this, so once again continue using the train set with rotational augmentations.



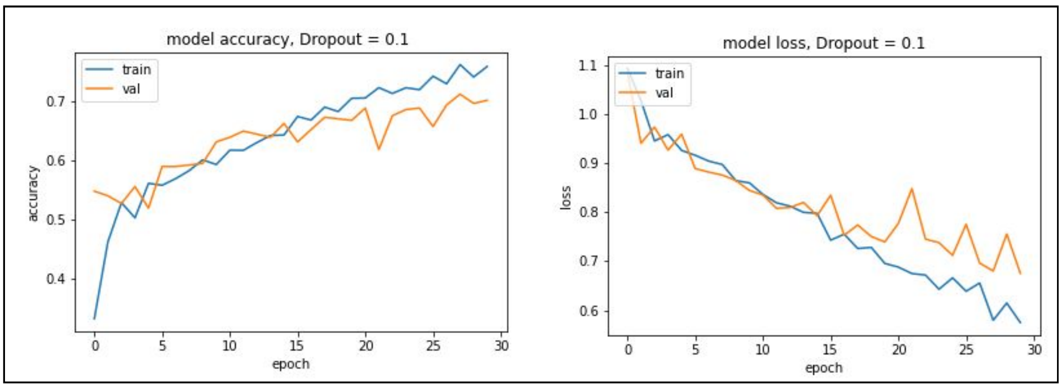

## 4.2.4 Dropout

To curb the overfitting, a dropout layer was appended to the end of the 3 fully connected layers in the final part of the model.

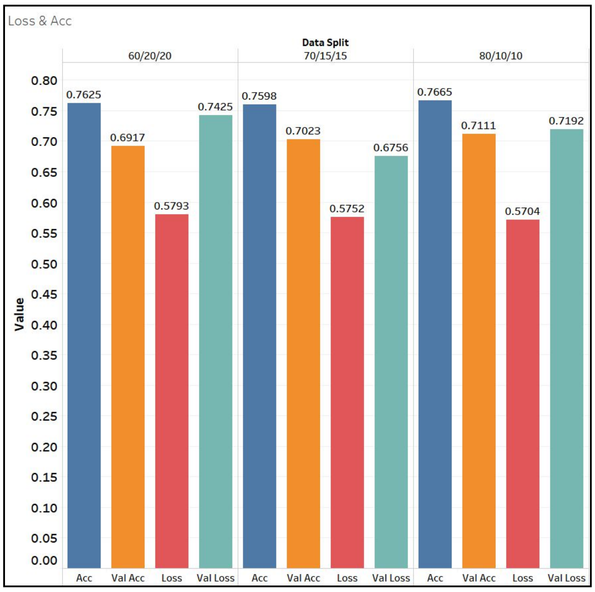



Applying Dropout to curb the overfitting occurring, when the only transformation applied to the images was rotation improved the most when dropout was set to the value of 0.1. Validation Loss decreased by 2.5% and Improved the Validation accuracy by 1.2%.

Overall there was an improvement using this value of dropout but there is still some overfitting which could be improved by using a different data split. Looking at the graph below there is still some room for improvement, as was showing any signs of plateauing yet.



### 4.2.5 Data Split




On Investigation of using a different data split. 70/15/15 still provided the best results. The other splits were producing models with a loss above 70 percent which is not good as that meant it was showing a decline in improvement which would not be beneficial. Although the accuracy and loss on the training did show slight improvement but this is more of a case of the models overfitting to the training set.

### 4.2.6 Test Results

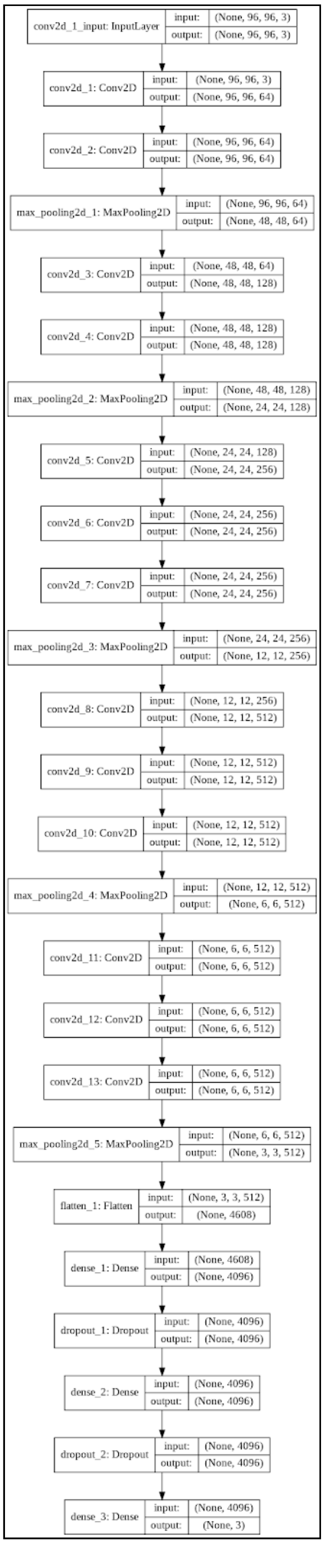

Applying the final model on our test set which based on AlexNet architecture, produced an accuracy score of 69.6% this is a rather impressive improvement over the vgg16 final model, I produced. This could be due to the fact that the modified vgg16 model is a larger model which is more prone to overfitting on the training set than our smaller modified AlexNeT hence producing a higher accuracy on our test set as it was able to generalise much better. With more parameters a model is more likely to suffer from overfitting, so more regularisation techniques will be needed but they can only do so much. Final model Implementation included 3 dropout layers at 0.1, learning rate = 1e-5, rotation augmentation only, using the 70/15/15 split.

# Conclusion

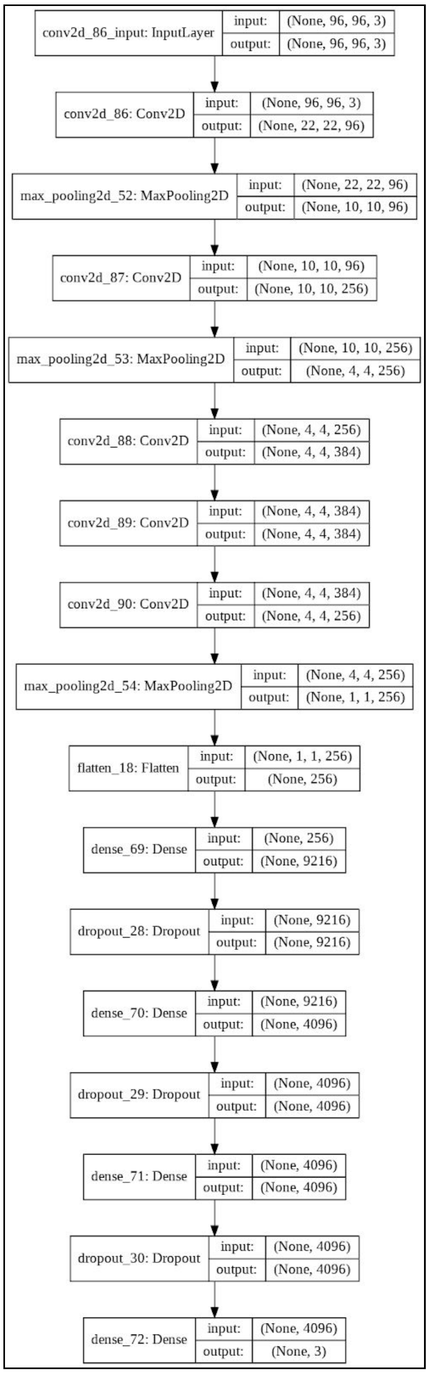
The Objective was to classify images into their respective genres. This Goal has been reasonably met as there is still room for improvement from the selection of images to model architecture and hyperparameter fine tuning. Further work would involve curating a bigger dataset and deploying a larger model on this dataset, and using higher res images for feature detection. Final Model structures can be found in the appendix below, and more examples of image classification as well.

# Appendix

### VGG16 Final Model Structure

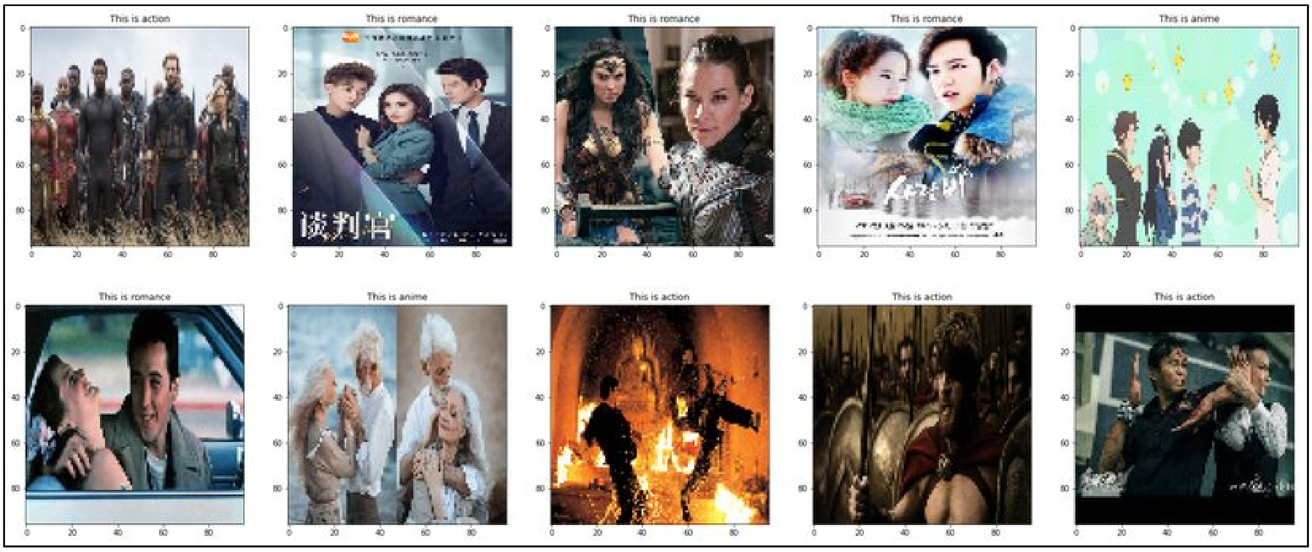



### AlexNet Final Model Structure



## More Image Classification Examples

### AlexNet Modified




### VGG16 Modified