CERVIX TYPE CLASSIFIER USING CONVOLUTIONAL NEURAL NETWORK

By: Ivy Huong Nguyen, Ph.D.
Springboard Data Science Intensive Track

Problem Statement

- Cervical cancer is classified as an easy-to-prevent cancer if caught in its precancerous stage
- One of the most problematic issue in treating patients with this type of cancer is the ability to identify an appropriate treatment that works effectively and accordingly to the patient’s physiological needs

→ develop a new algorithm that can effectively identify the type of cervix a patient has based on images.

→ Business Benefits:

  – *Health providers can use this algorithm to have an effective real-time determination to provide an appropriate cervical cancer treatment for their patients.*
  – *Patients don’t have to face high-cost treatment as well as ineffective treatments.*

Approach

- Leverage use of Convolutional Neural Network (CNN) to build an effective classifier that can identify the type of cervix accurately using cervix images
- Why CNNs?
  - *Significant advancements in the field of computer vision*
  - *CNNs are explicitly designed for image data*

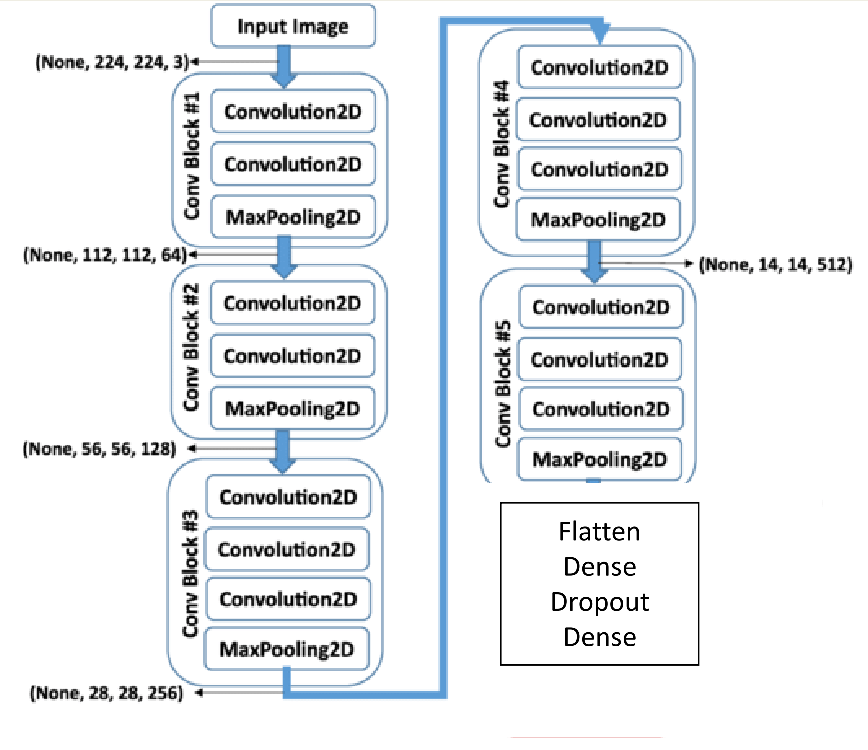
Workflow

- Comparing model performance among three different model types:
  - *Model 1: Simple CNN model that is constructed from scratch with the following architecture:*

Layer (type)	Input Shape	Output Shape
Block1_Conv2D (Conv2D)	(None, 224, 224, 3)	(None, 222, 222, 32)
Block1_Activation (‘relu’)	(None, 222, 222, 32)	(None, 222, 222, 32)
Block1_MaxPooling2D	(None, 222, 222, 32)	(None, 111, 111, 32)
Block1_Dropout	(None, 111, 111, 32)	(None, 111, 111, 32)
Block2_Conv2D (Conv2D)	(None, 111, 111, 32)	(None, 109, 109, 64)
Block2_Activation (‘relu’)	(None, 109, 109, 64)	(None, 109, 109, 64)
Block2_MaxPooling2D	(None, 109, 109, 64)	(None, 54, 54, 64)
Block2_Dropout	(None, 54, 54, 64)	(None, 54, 54, 64)
Block3_Conv2D (Conv2D)	(None, 54, 54, 64)	(None, 52, 52, 128)
Block3_Activation (‘relu’)	(None, 52, 52, 128)	(None, 52, 52, 128)
Block3_MaxPooling2D	(None, 52, 52, 128)	(None, 26, 26, 128)
Block3_Dropout	(None, 26, 26, 128)	(None, 26, 26, 128)
Block4_Flatten	(None, 26, 26, 128)	(None, 86528)
Block4_Dense (‘relu’)	(None, 86528)	(None, 256)
Block4_Dropout	(None, 256)	(None, 256)
Block4_Dense (‘softmax’)	(None, 256)	(None, 3)

Workflow (continued)

- *Model 2: Utilize transfer learning to build a VGG16 model that has the final block of flatten and dense layers. The architecture of this model is as follows:*



- *Model 3: Also utilize transfer learning to build a ResNet50 model that has the final block of flatten and dense layers.*

Workflow (continued)

- Optimizing the best model in the previous step by changing the number of layers, changing the activation method, etc. ….
- Prediction

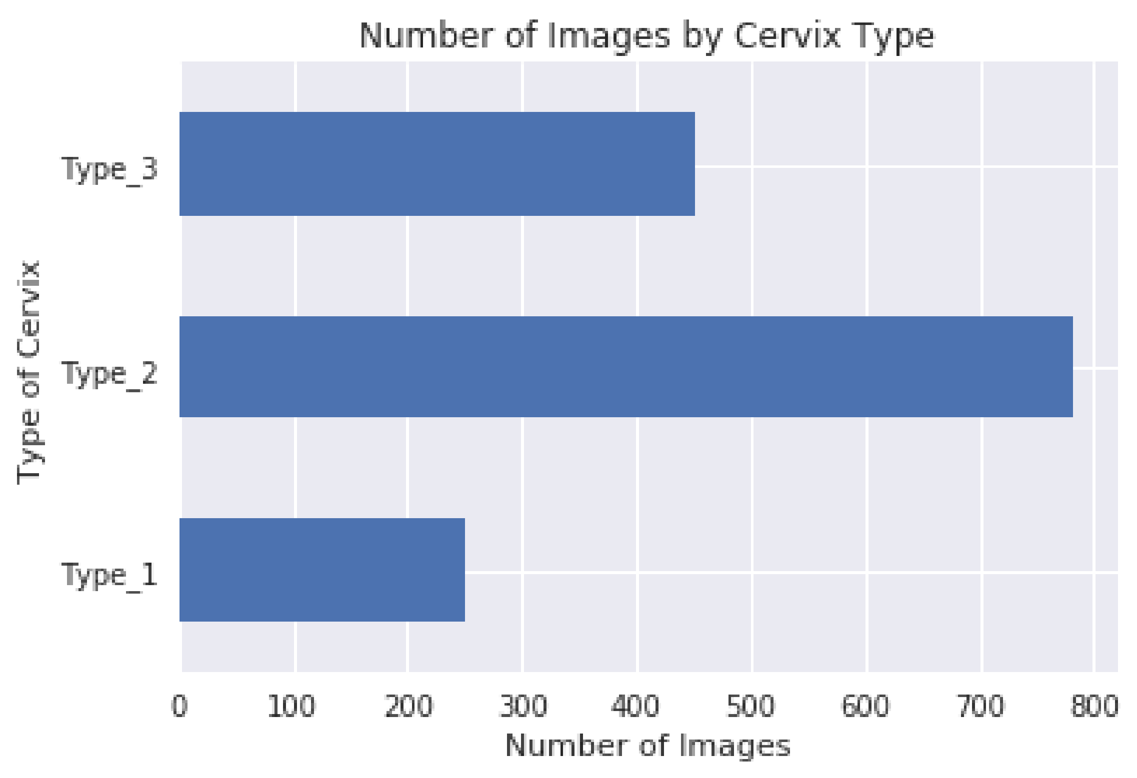
Introduction to the data

- Both the training and the testing dataset were obtained from Intel and Mobile ODT Kaggle's competition webpage: https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening
- Data processing:
  - *The training data set was split into two parts: 80% of the training dataset was kept for training and 20% of the training dataset was used for the validation portion*
  - *Cervix type is the label or the target variable. There are 3 different cervix types of this dataset and the target variable was converted into one-hot-encoding*
    - [1, 0, 0] represents cervix Type 1
    - [0, 1, 0] represents cervix Type 2
    - [0, 0, 1] represents cervix Type 3

Exploratory Data Analysis (EDA)

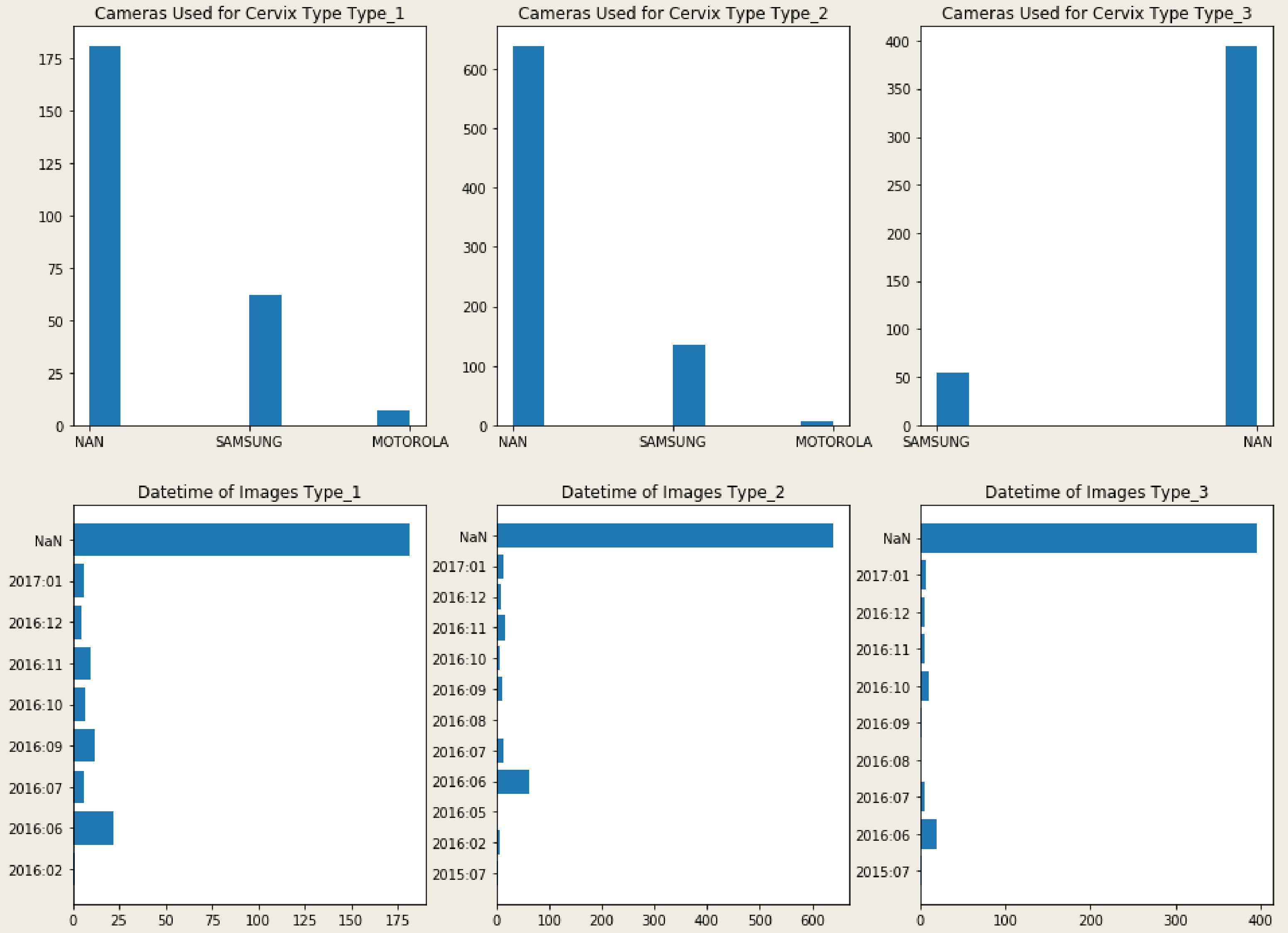
- The EDA portion was split into different categories including:
  - *EDA based on image sizes*
  - *EDA based on the available Exchangable Image File Format (EXIF data) of ll images*
  - *Inferential statistics to determine if there is any image resolution difference between different cameras.*
  - *Descriptive statistics for both the training the testing dataset*
  - *Structural Similarity Index (SSIM) of a cervix type*

EDA based on image sizes



- There are 3 different cervix type in the training dataset: 250 type 1, 781 of type 2, 450 of type 3.
- All images have very high resolution → reduce image sizes to (224,224,3).
- (224,224,3) was chosen to accommodate the default input size of the VGG16 and the ResNet 50 models.

EDA based on EXIF data



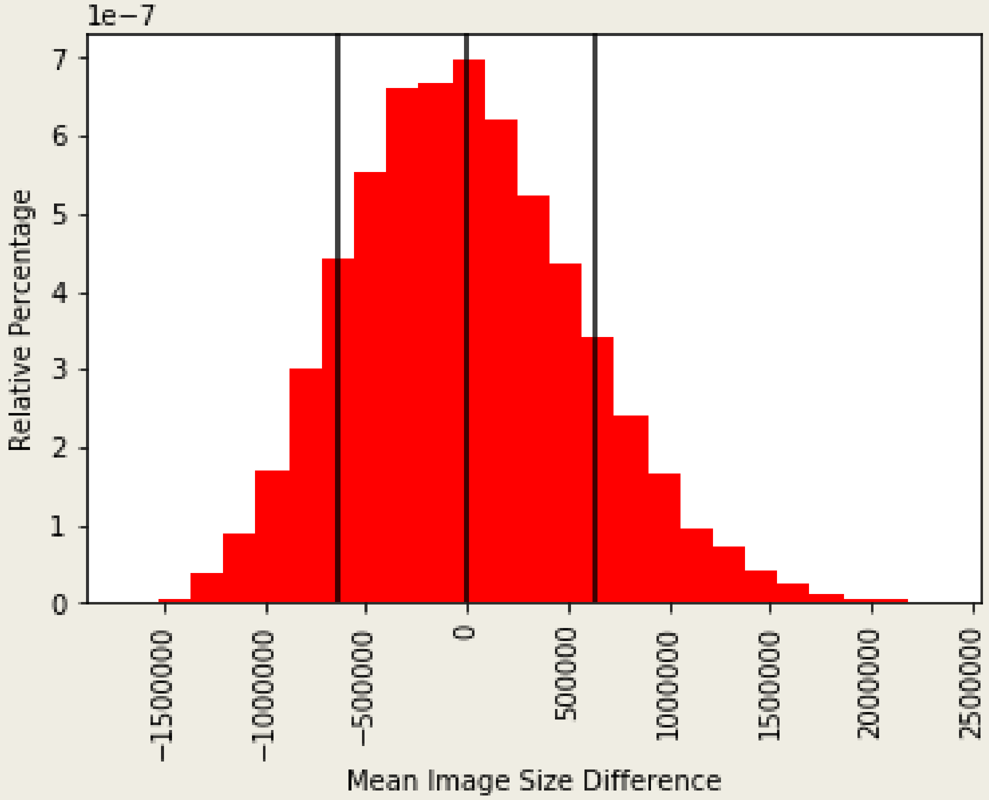
- Two information of the EXIF data were explore: camera models and image datetime
- Majority of images were taken by Samsung and Motorola.
- There is a large portion of images that have no EXIF information
- If we disregard the NaN portion, June of 2016 is the time that most images were taken.

Inferential Statistics

- Since there were at least 2 different types of cameras used to take images, I decided to test whether there is a difference in image resolution taken by Samsung versus Motorola.
- Null hypothesis: there is no difference in the resolution mean between the images taken by Samsung and those taken by Motorola
- Alternative hypothesis: there is a significant difference in the resolution mean between the images taken Samsung and those taken by Motorola.
- The inferential Statistics were conducted using two different approaches:
  - *Frequentist approach: two-tailed t test because the sample size of Motorola sub-dataset is less than 30 → calculate the statistical t-value (two-tailed test) and compare the corresponding p-value to alpha=0.05*
  - *Bootstrap hypothesis testing: calculate the statistical p-value by computing 10,000 bootstrap replicates*

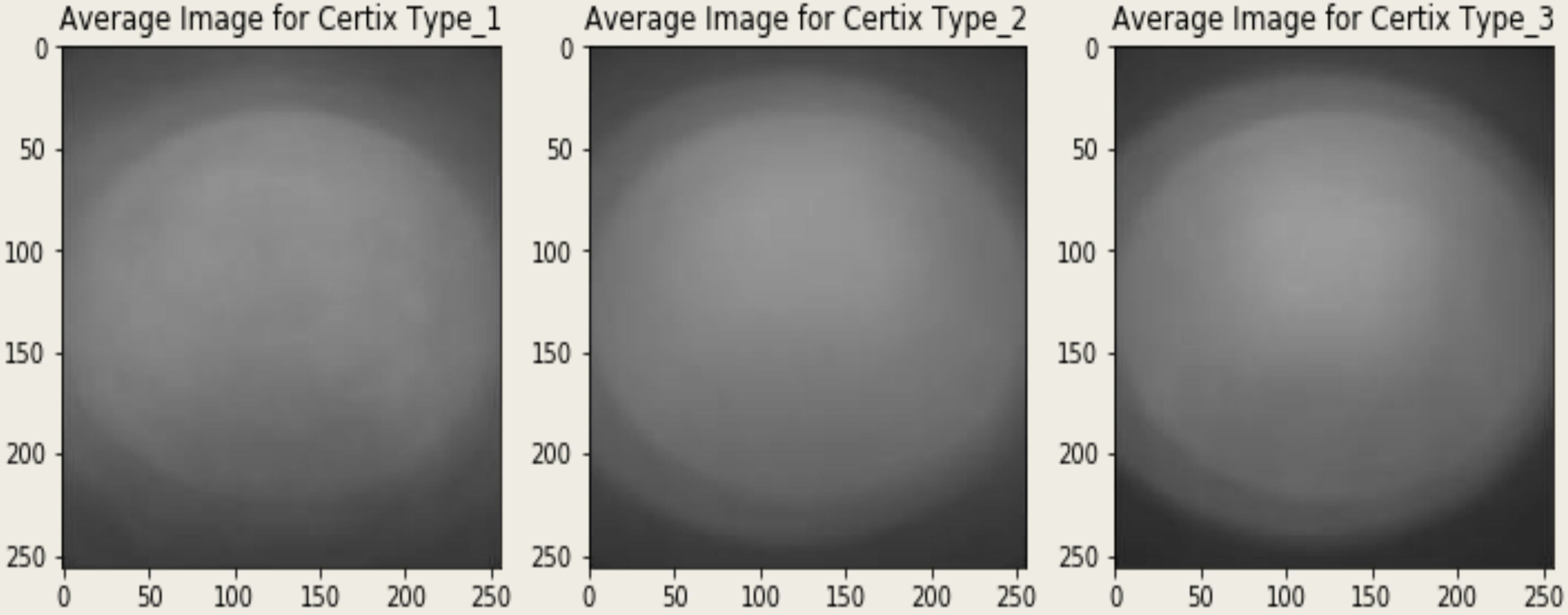
Inferential Statistics (continued)

- Using frequentist approach, I obtained a statistical t-value of -1.106 which has the corresponding p-value of 0.27. This p-value > alpha → failed to reject the null hypothesis
- Using bootstrap hypothesis testing, p-value >>> alpha → failed to reject the null hypothesis

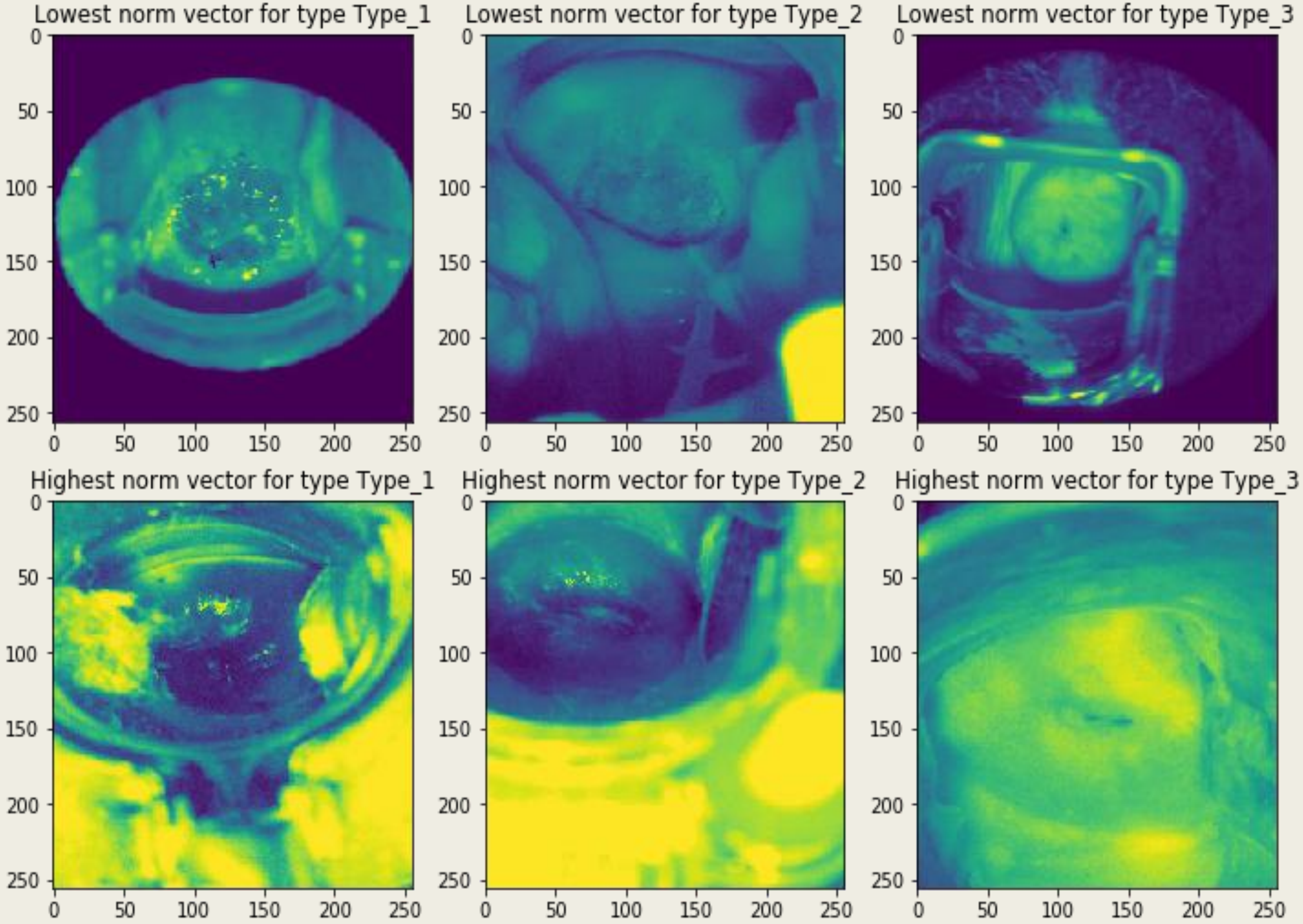


Descriptive Statistics

- Average image of both testing and training dataset
- Min and max images based on vector norms



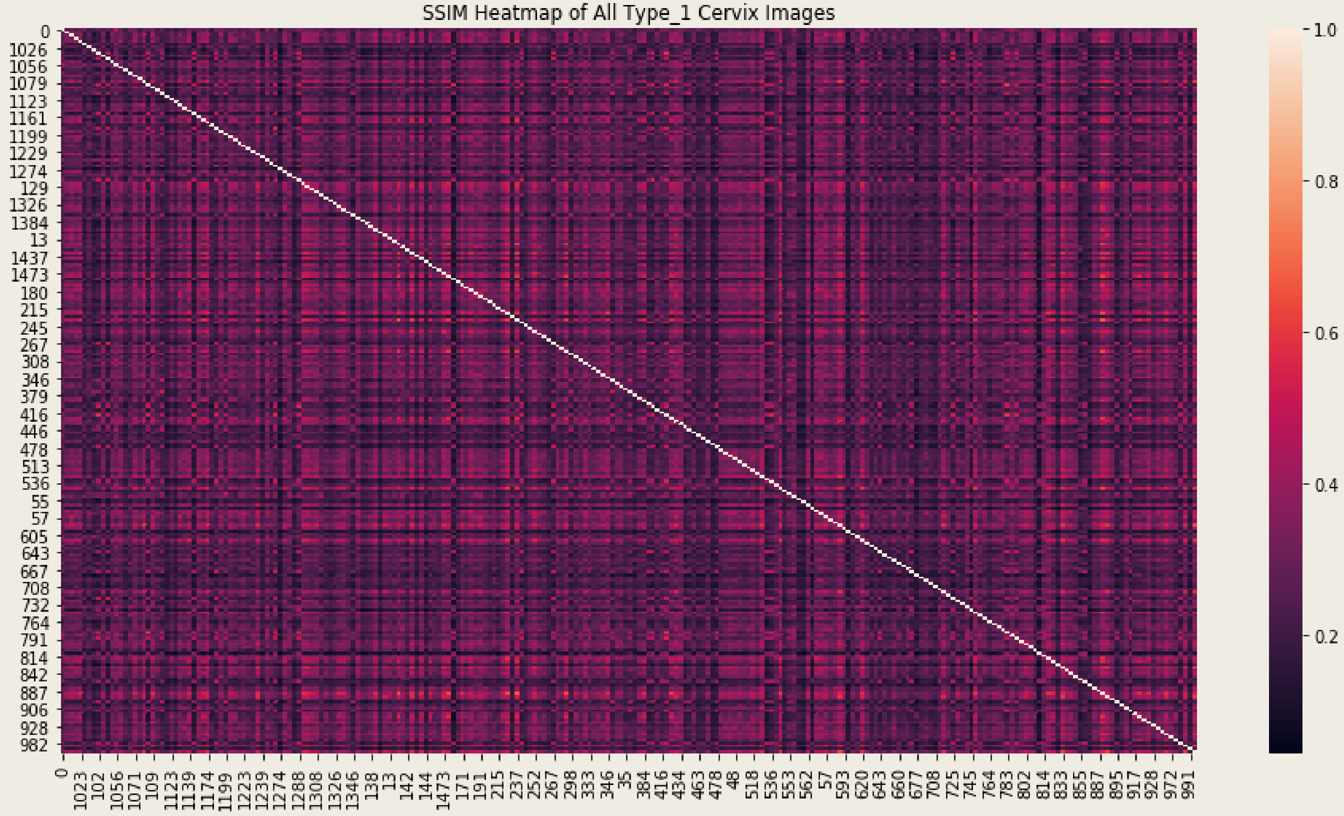
Descriptive Statistics (continued…)



- Images with the higher norms have higher color contrast than that of the lower norms.

Structural Similarity Index (SSIM)

- SSIM keeps track of the structural information of the images and thus can be used to compare the similarity in structures of a given set of images data.
- SSIM value can vary between -1 and 1, where 1 indicates perfect similarity.
- Only the SSIM of cervix type 1 from the training dataset was determined due to the time limit.



Model Selection

- Step 1: Comparing model performance
  - *Model 1: simple CNN with 3 blocks of layers consisting of a convolutional layer followed by an activation, a maxpooling2D layer and a dropout layer. The last block of layers of this model includes a flatten layer followed by a dense layer, a dropout layer, and a dense layer.*
  - *Model 2: built up from the trained model VGG16 (transfer learning). This model was completed by adding the last block of layers of model 1 including flatten, dense, dropout, and dense*
  - *Model 3: built up from the trained model ResNet50 (transfer learning). This model was completed by adding the last block of layers of model 1 including flatten, dense, dropout, and dense*
- Weights: used pre-trained weights obtained from training the model using ImageNet dataset

Model Performance

Model	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	Training Time/Epoch	Early_Stopping was used?
Model 1	0.5169	0.5118	1.0503	1.0171	206s	No
Model 2	0.5270	0.5118	7.4928	7.8691	1432s	Yes
Model 3	0.5211	0.5118	7.5800	7.8691	837s	Yes

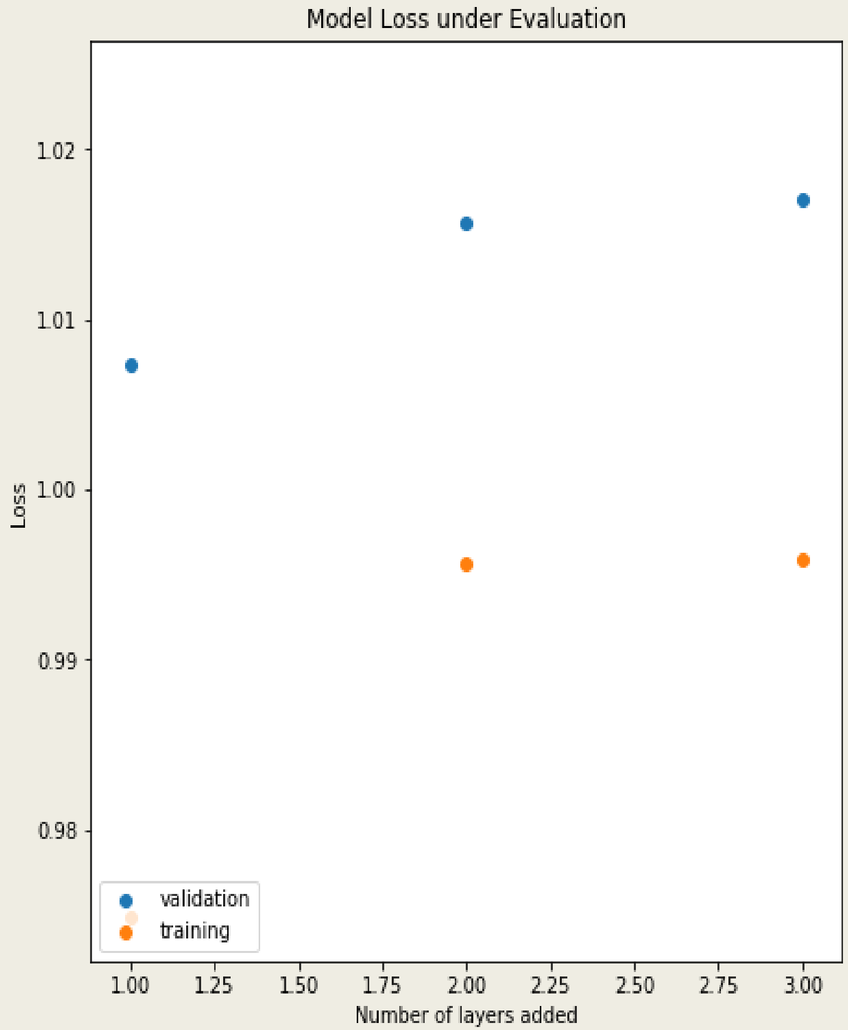
- Model 2 and model 3 took much longer time to train with the same or slightly better accuracy in comparison to model 1.

→ Chose to use model 1 for further optimization

Model 1 Optimization

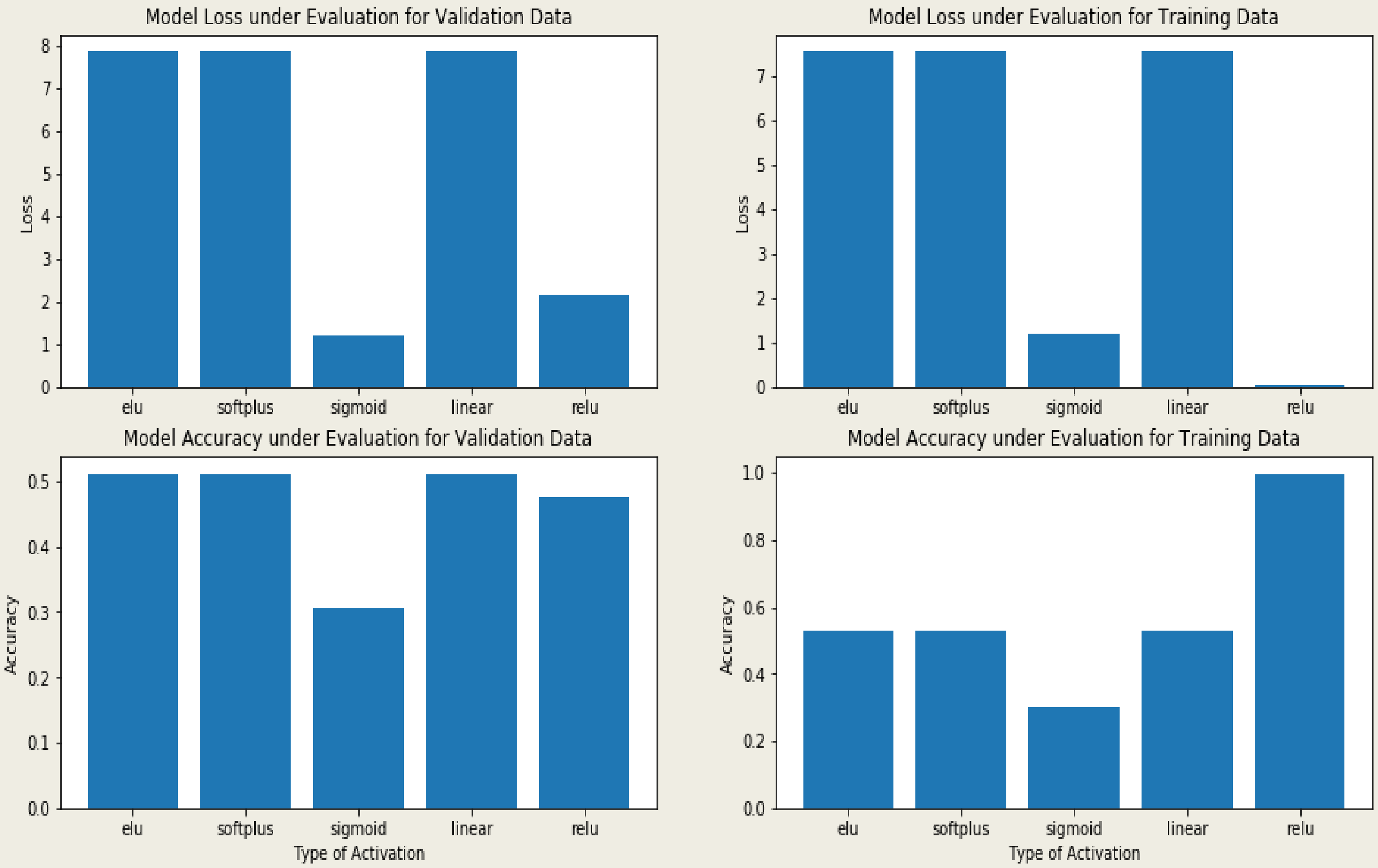
- Model 1 was optimized by tuning several factors:
  - *Number of layers*
  - *Activation method*
  - *Number of filters*
  - *Optimizer/loss function combination*

Tuning the number of layers of Model 1



- It appears that adding more layers into model 1 did not help improving its performance in both training and testing dataset

Tuning model 1 using different activation method



- ‘relu’ appears to the best activation method out of the 5 tested activation methods.

→ seen through ‘relu’ accuracy in both the training and the validation dataset.

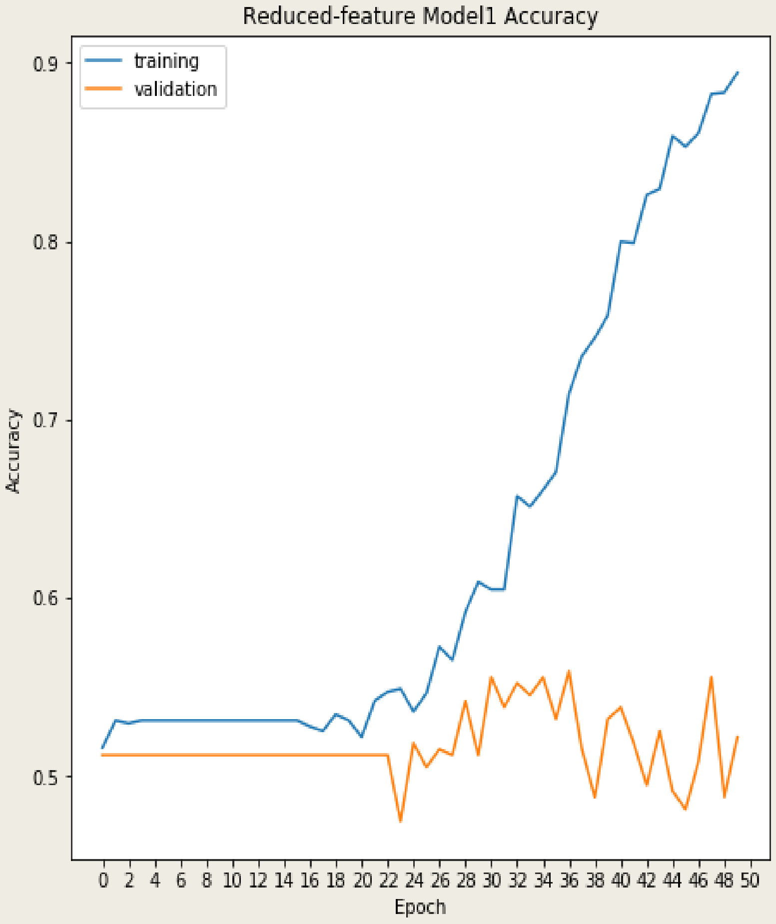
- While ‘elu’, ‘softplus’ and ‘linear’ also give us very high accuracy in the validation dataset, their losses are much greater than ‘relu’, which make them not as ideal as ‘relu’ activation method.

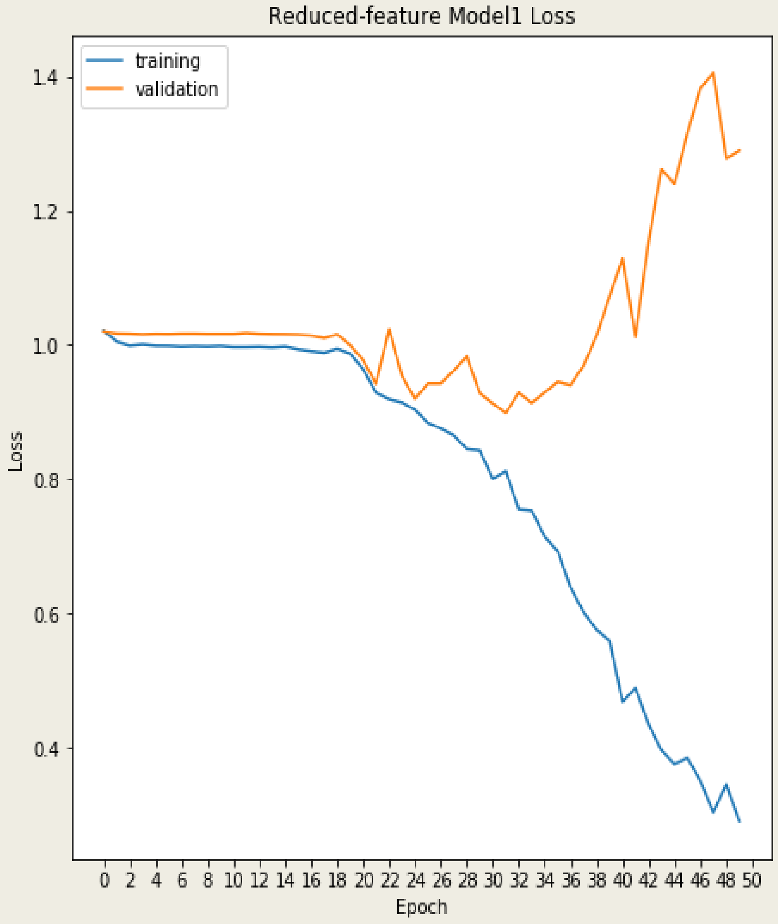
Reducing the number of filters

Layer (type)	Input Shape	Output Shape
Block1_Conv2D (Conv2D)	(None, 224, 224, 3)	(None, 222, 222, 8)
Block1_Activation (‘relu’)	(None, 222, 222, 8)	(None, 222, 222, 8)
Block1_MaxPooling2D	(None, 222, 222, 8)	(None, 111, 111, 8)
Block1_Dropout	(None, 111, 111, 8)	(None, 111, 111, 8)
Block2_Conv2D (Conv2D)	(None, 111, 111, 8)	(None, 109, 109, 16)
Block2_Activation (‘relu’)	(None, 109, 109, 16)	(None, 109, 109, 16)
Block2_MaxPooling2D	(None, 109, 109, 16)	(None, 54, 54, 16)
Block2_Dropout	(None, 54, 54, 16)	(None, 54, 54, 16)
Block3_Conv2D (Conv2D)	(None, 54, 54, 16)	(None, 52, 52, 32)
Block3_Activation (‘relu’)	(None, 52, 52, 32)	(None, 52, 52, 32)
Block3_MaxPooling2D	(None, 52, 52, 32)	(None, 26, 26, 32)
Block3_Dropout	(None, 26, 26, 32)	(None, 26, 26, 32)
Block4_Flatten	(None, 26, 26, 32)	(None, 21632)
Block4_Dense (‘relu’)	(None, 21632)	(None, 256)
Block4_Dropout	(None, 256)	(None, 256)
Block4_Dense (‘softmax’)	(None, 256)	(None, 3)

- The architecture of model after reducing the number of filters

Reduced-filters model 1 (version 1)





- reducing the number of filters seems to help the performance of model 1 slightly.
- Reducing the number of filter helps to reduce the training time significantly → makes this new model 1 a better candidate for the next optimization step.

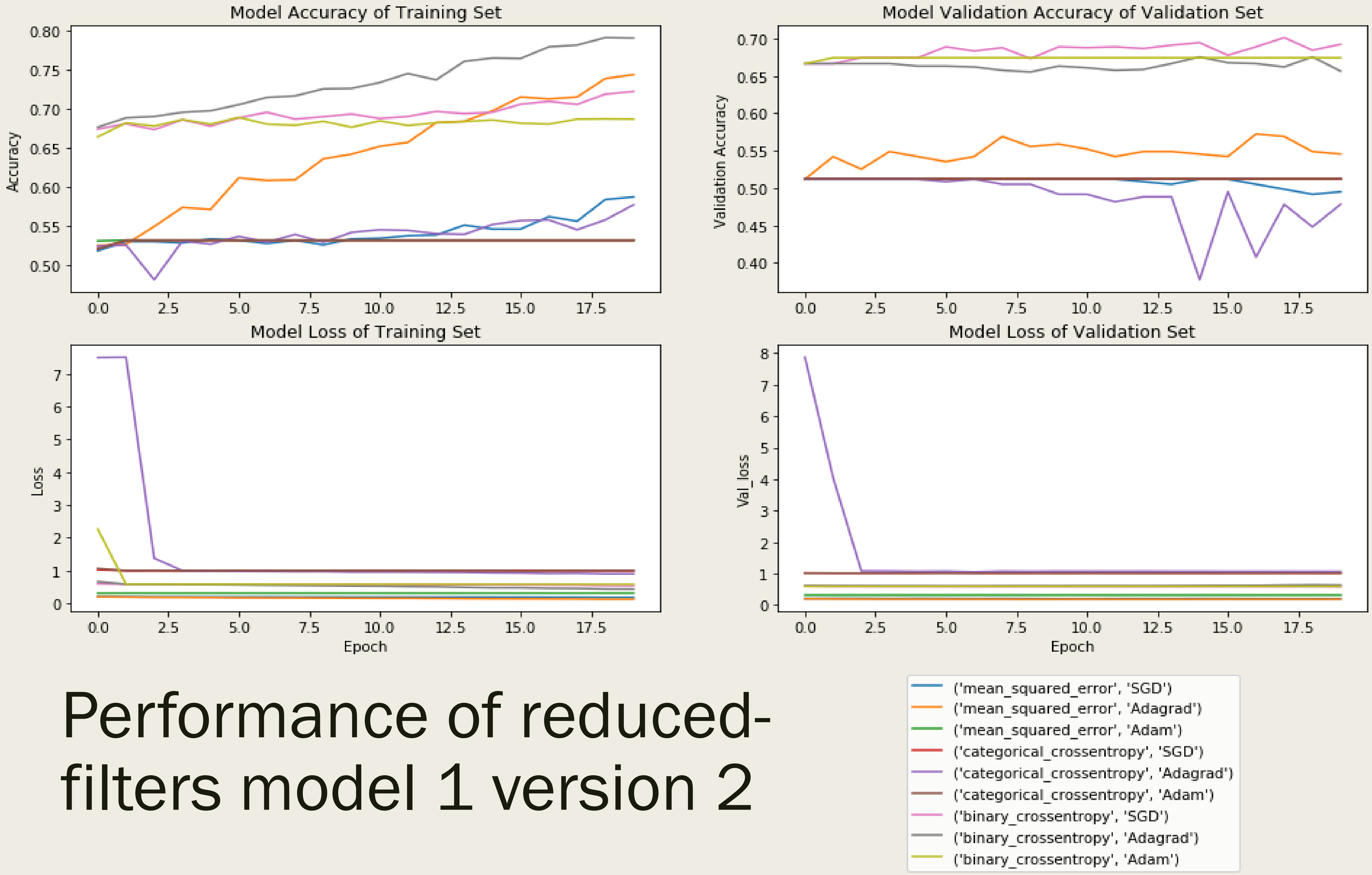
Tuning the performance of model 1 by using different optimizer/loss function combination

- Evaluate the performance of the reduced-filters model 1 version under 9 different optimizer/loss function combinations
- The best combination is: stochastic gradient descent for optimizer and binary_crossentropy for loss function.

Further reducing the number of filters in reduced-filters model 1 version 1

Layer (type)	Input Shape	Output Shape
Block1_Conv2D (Conv2D)	(None, 224, 224, 3)	(None, 222, 222, 7)
Block1_Activation (‘relu’)	(None, 222, 222, 7)	(None, 222, 222, 7)
Block1_MaxPooling2D	(None, 222, 222, 7)	(None, 111, 111, 7)
Block1_Dropout	(None, 111, 111, 7)	(None, 111, 111, 7)
Block2_Conv2D (Conv2D)	(None, 111, 111, 7)	(None, 109, 109, 7)
Block2_Activation (‘relu’)	(None, 109, 109, 4)	(None, 109, 109, 4)
Block2_MaxPooling2D	(None, 109, 109, 4)	(None, 54, 54, 4)
Block2_Dropout	(None, 54, 54, 4)	(None, 54, 54, 4)
Block3_Conv2D (Conv2D)	(None, 54, 54, 4)	(None, 52, 52, 4)
Block3_Activation (‘relu’)	(None, 52, 52, 5)	(None, 52, 52, 5)
Block3_MaxPooling2D	(None, 52, 52, 5)	(None, 26, 26, 5)
Block3_Dropout	(None, 26, 26, 5)	(None, 26, 26, 5)
Block4_Flatten	(None, 26, 26, 5)	(None, 3380)
Block4_Dense (‘relu’)	(None, 3380)	(None, 256)
Block4_Dropout	(None, 256)	(None, 256)
Block4_Dense (‘softmax’)	(None, 256)	(None, 3)

- Activation method used in block 1, block 2, and block 3: ‘relu’
- Best optimizer: SGD
- Best loss function: binary_crossentropy



Performance of reduced-filters model 1 version 2

Prediction and Conclusions

Model 1	Accuracy score
Reduced-filtered model 1 version 1	24.9%
Reduced-filtered model 1 version 2	47.7%

- The best model found for this project is reduced-filtered model 1 version 2
- The best optimizer and loss function that works well for this model is binary_crossentropy and stochastic gradient descent. Rectified Linear Unit (relu) shows to be the best activation method to be used for block 1, block 2, and block 3 of model 1. This model was able to predict accurately up to 47.7% of the testing dataset.
- The process in training a model in deep learning requires a lot of time and thus in order to improve or optimize any model performance, we would need a better GPU and larger storage memory capacity.