Classifying Potential Pneumonia Cases using Convolutional Neural Network

Achyut Kafle

Thinkful Data Science

Final Capstone Presentation

December 21, 2018

Presentation outline

- Background
- Objective
- Summary of results
- Methodology
    - Data
    - Exploratory data analysis
    - CNN architecture
    - Classification and evaluation
- Conclusions and next steps

Background

- **Pneumonia**
  - 15% of all deaths of children under 5 internationally
  - Over 500k visits to emergency departments and over 50k deaths in the US in 2015
  - Review of chest radiographs to locate areas of lung opacities and confirm through clinical history, vital signs and lab exams

- **Deep learning CNN model**
  - Automate the initial detection (image screening) of potential pneumonia cases in order to prioritize and expedite the review

Objective

- To build and evaluate a CNN model to classify potential pneumonia cases using chest radiographs

Summary of results

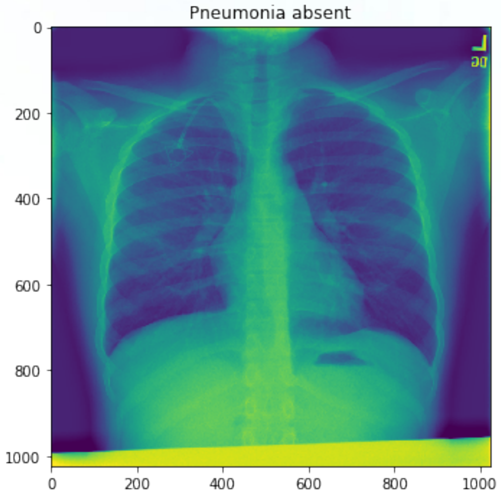
- A relatively simple CNN achieved reasonable performance (accuracy: **~75%**; recall-pneumonia class: **0.82**).
- Evaluated optimizers (Adam, SGD and RMSprop) and batch sizes (100, 200 and 400) performed very consistently.
- Image pre-processing using feature-wise centering and normalization improved the accuracy a little bit (accuracy**~76%**).
- A pre-trained VGG16 model performed very poorly in this case.

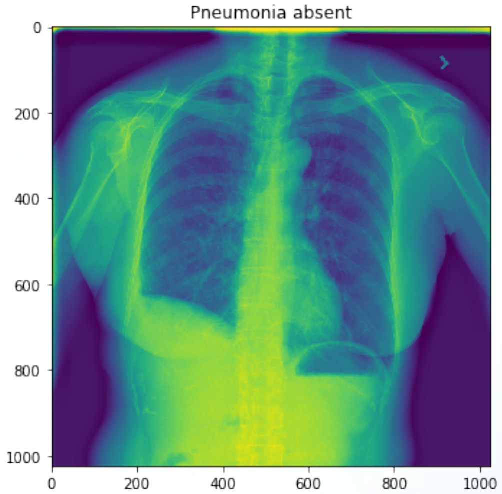
Methodology: Data

- **Kaggle's RSNA Pneumonia Detection Challenge**
  - 26,684 unique chest radiographs
  - 22.5% pneumonia cases
  - DICOM images of the dimension: 1024x1024x3
- **Random under sampling of majority class (i.e., normal case)**
  - 12,024 images
- **Final balanced samples**
  - 5011 train images
  - 1004 validation images
  - 998 test images
  - Image resized to 150x150x3

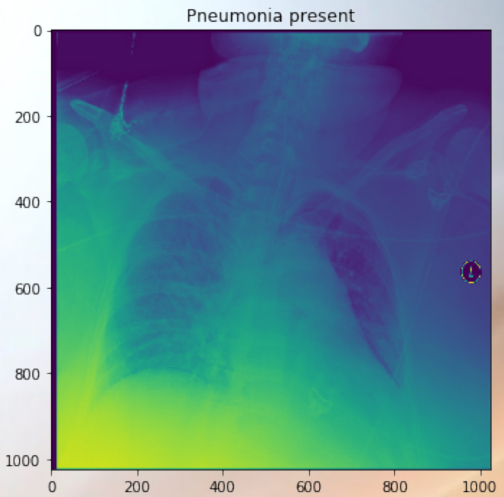
Methodology: Data…

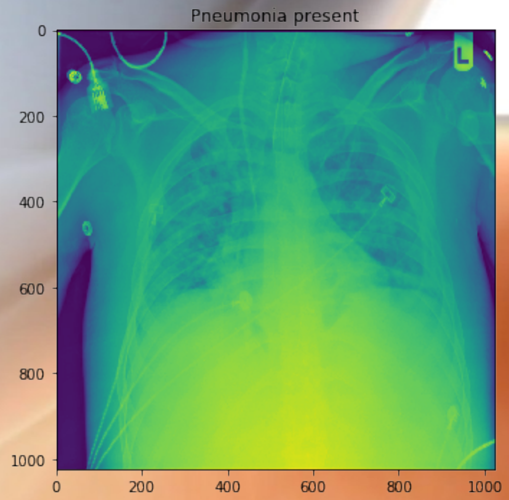
Normal

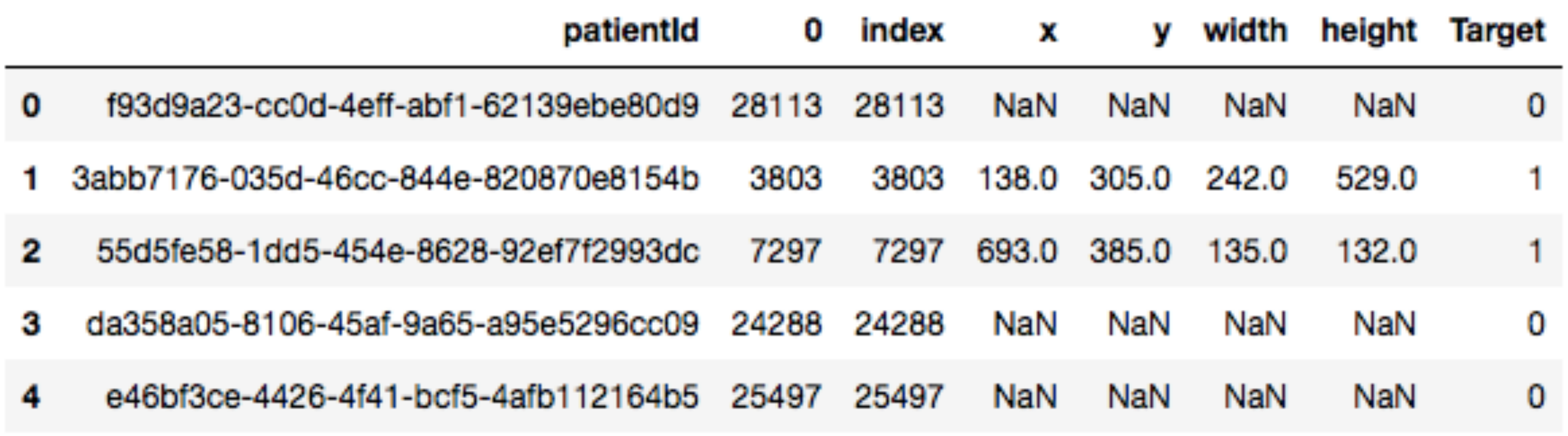




Potential Pneumonia





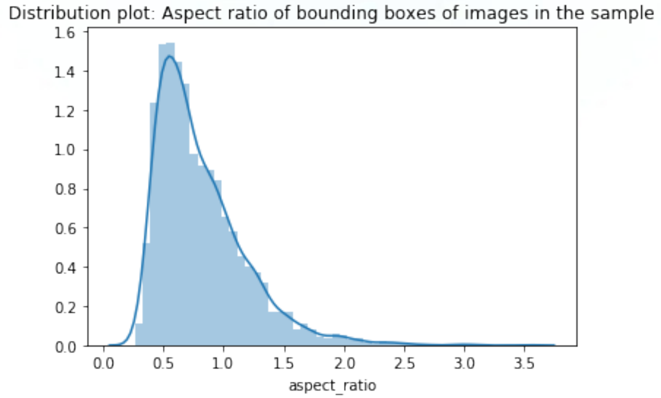


	patientId	0	index	x	y	width	height	Target
0	f93d9a23-cc0d-4eff-abf1-62139ebe80d9	28113	28113	NaN	NaN	NaN	NaN	0
1	3abb7176-035d-46cc-844e-820870e8154b	3803	3803	138.0	305.0	242.0	529.0	1
2	55d5fe58-1dd5-454e-8628-92ef7f2993dc	7297	7297	693.0	385.0	135.0	132.0	1
3	da358a05-8106-45af-9a65-a95e5296cc09	24288	24288	NaN	NaN	NaN	NaN	0
4	e46bf3ce-4426-4f41-bcf5-4afb112164b5	25497	25497	NaN	NaN	NaN	NaN	0

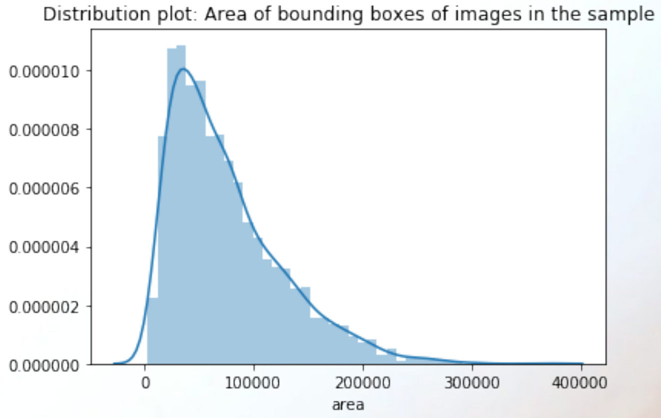
Data frame of image name, Pneumonia locations, target labels

Methodology: Exploratory data analysis

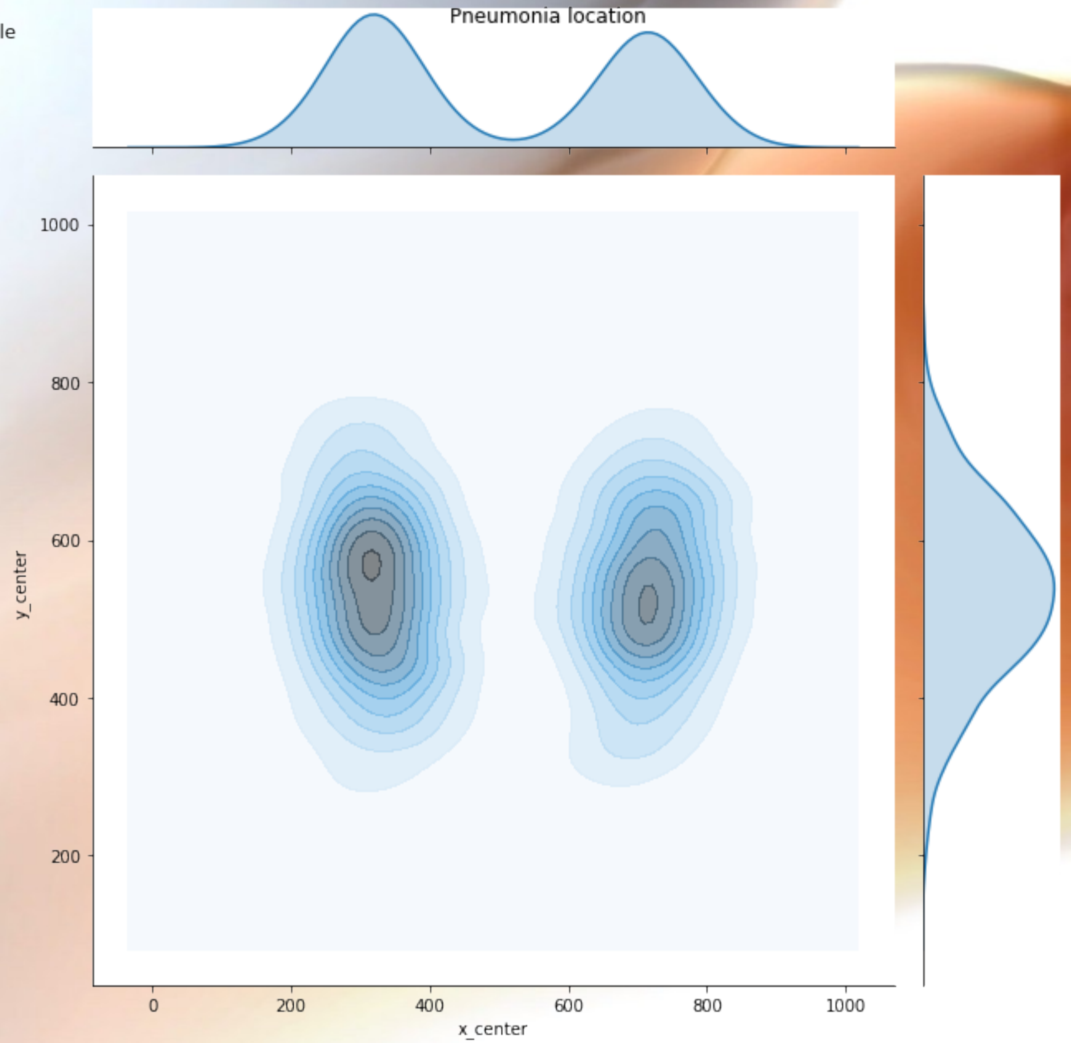
Aspect ratio of bounding boxes

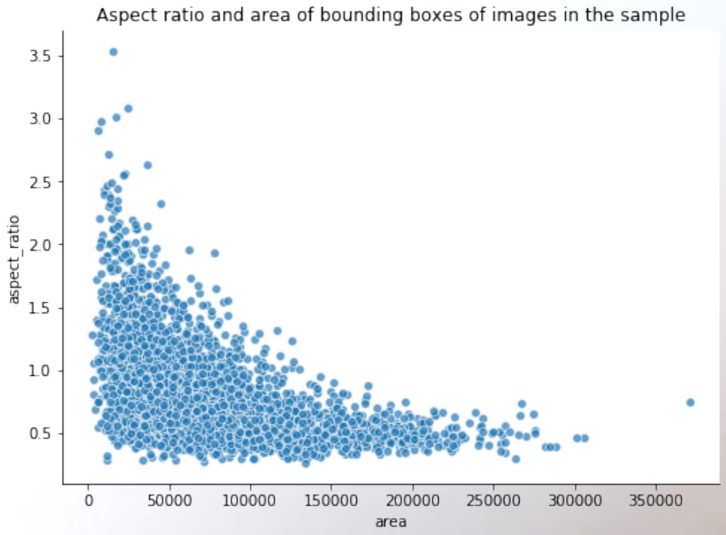


Area of bounding boxes



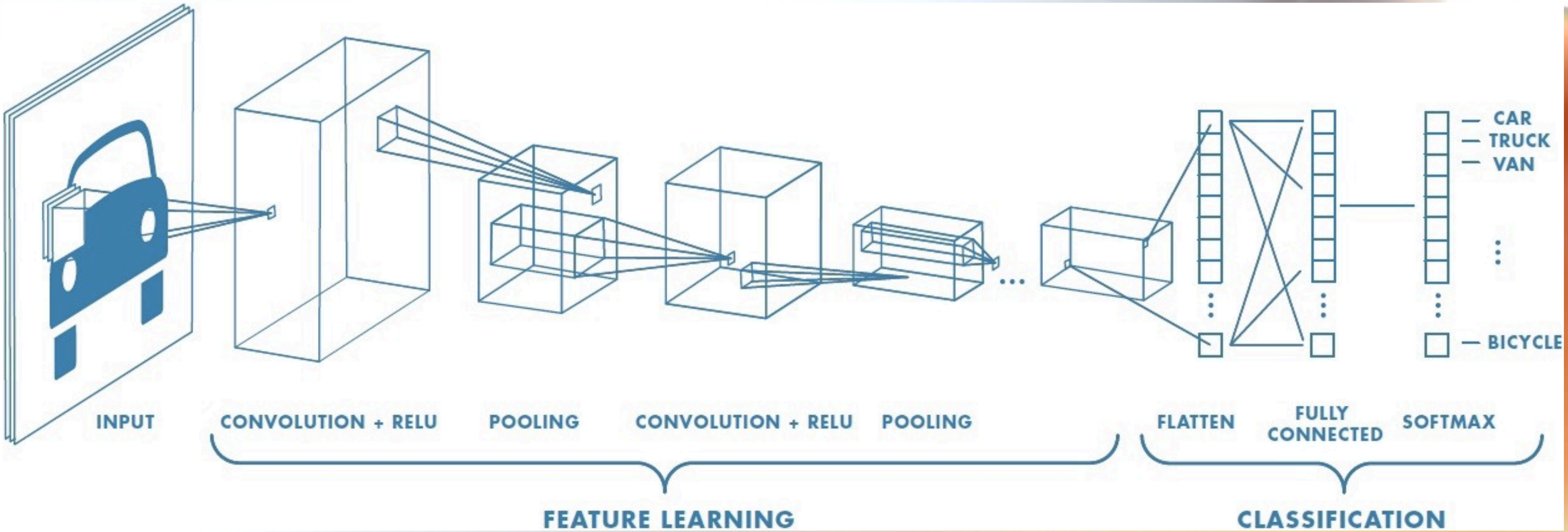
Pneumonia location

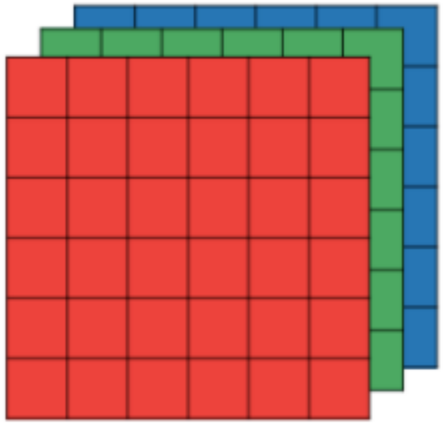




Relationship between aspect ratio and area of bounding boxes

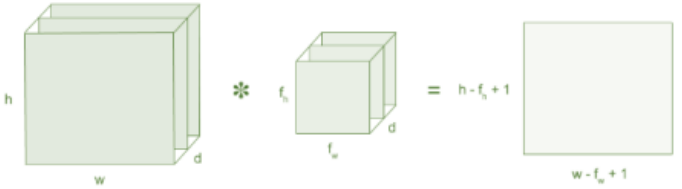
CNN

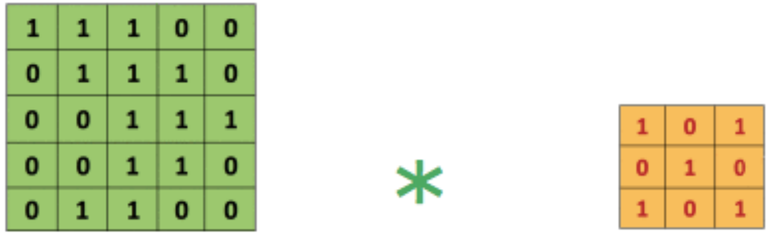

CAR
TRUCK
VAN
BICYCLE
INPUT
CONVOLUTION + RELU
POOLING
CONVOLUTION + RELU
POOLING
FLATTEN
FULLY CONNECTED
SOFTMAX
FEATURE LEARNING
CLASSIFICATION



6 x 6 x 3

- An image matrix (volume) of dimension **(h x w x d)**
- A filter **(f_h x f_w x d)**
- Outputs a volume dimension **(h - f_h + 1) x (w - f_w + 1) x 1**





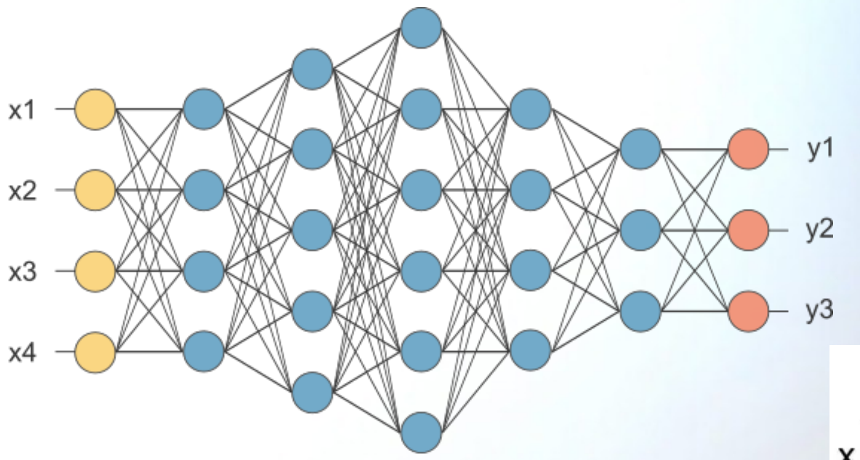
1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

*

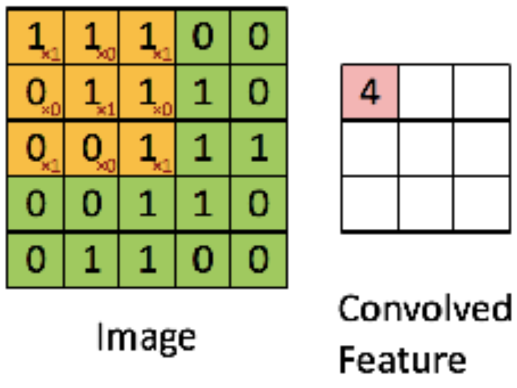
1	0	1
0	1	0
1	0	1

5 x 5 – Image Matrix

3 x 3 – Filter Matrix



Fully connected layer and output layer

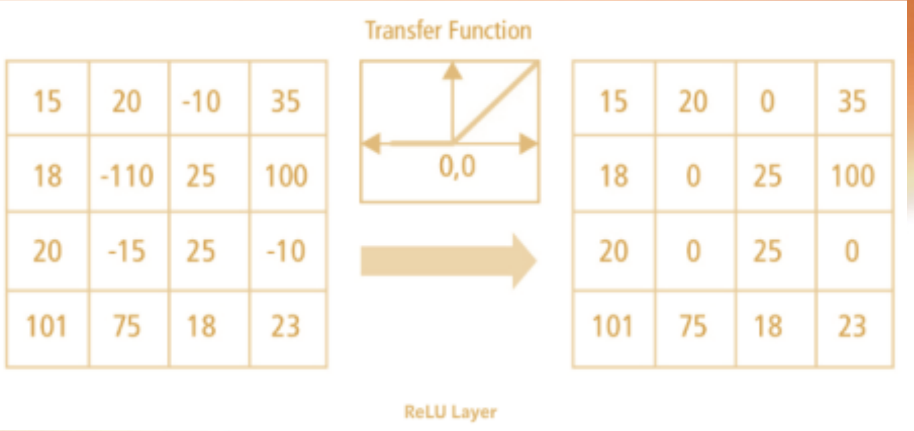


$1_{\times 1}$	$1_{\times 0}$	$1_{\times 1}$	0	0
$0_{\times 0}$	$1_{\times 1}$	$1_{\times 0}$	1	0
$0_{\times 1}$	$0_{\times 0}$	$1_{\times 1}$	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved Feature

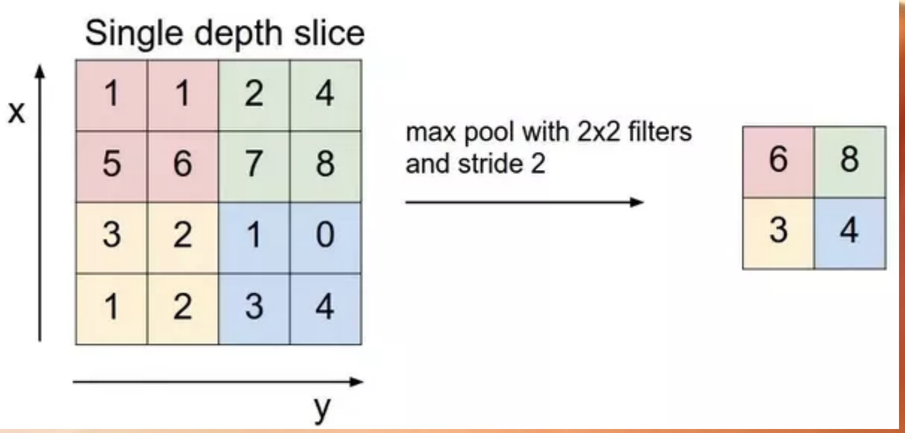


Transfer Function

15	20	-10	35
18	-110	25	100
20	-15	25	-10
101	75	18	23

15	20	0	35
18	0	25	100
20	0	25	0
101	75	18	23

ReLU Layer



Single depth slice

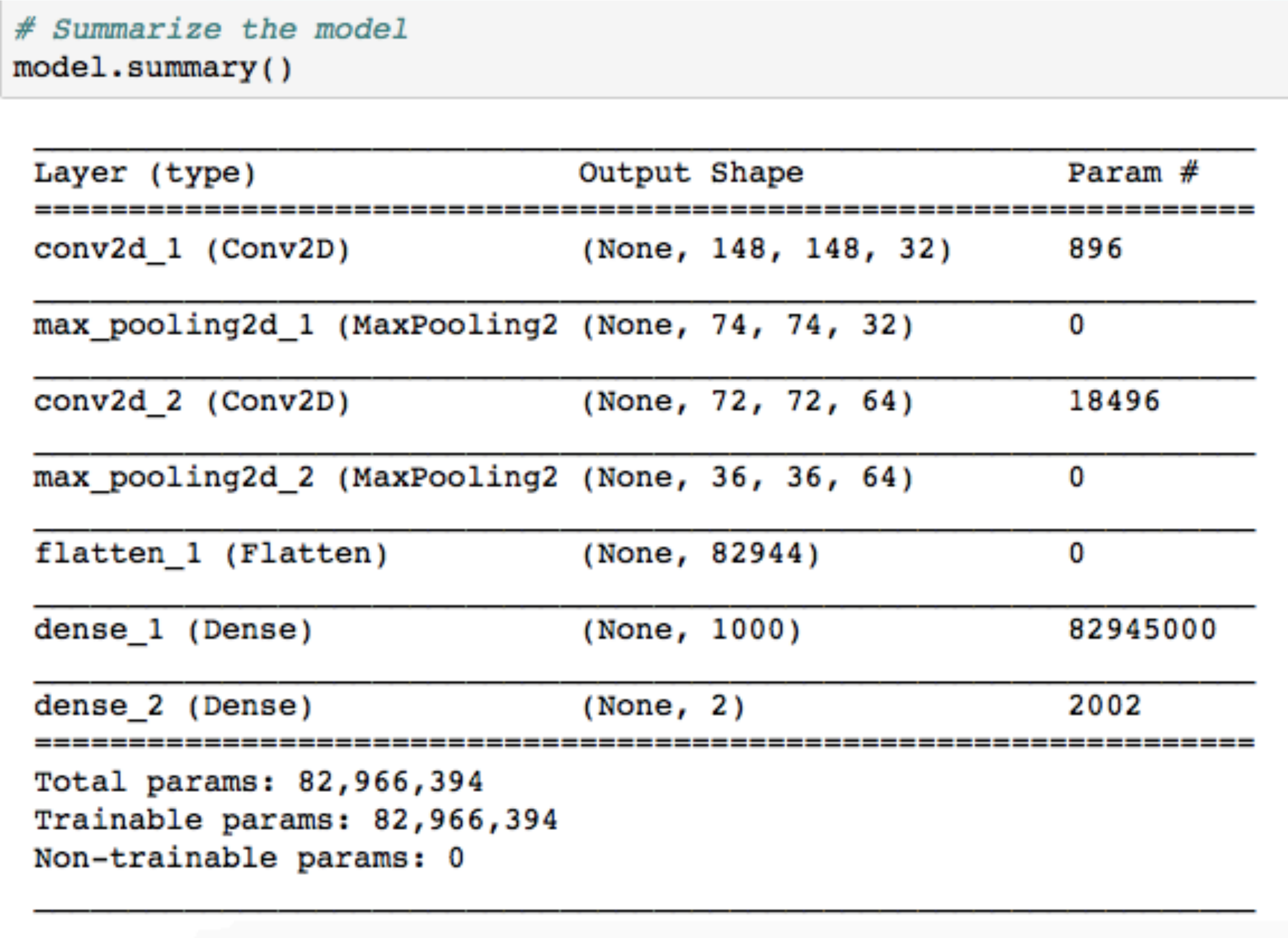
1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

max pool with 2x2 filters and stride 2

6	8
3	4

Flatten matrix into vector

CNN architecture



```
# Summarize the model
model.summary()
```

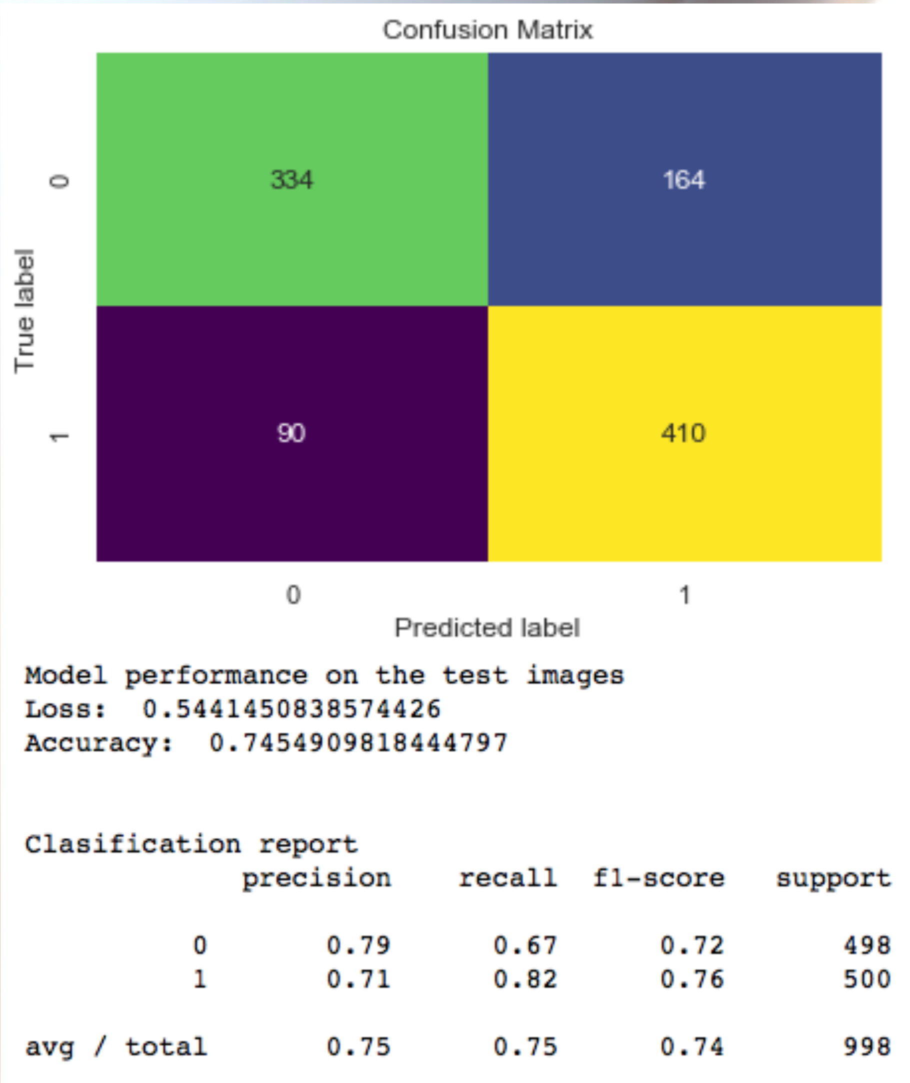
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 148, 148, 32)      896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 74, 74, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 72, 72, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 36, 36, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 82944)             0
_________________________________________________________________
dense_1 (Dense)              (None, 1000)              82945000
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 2002
=================================================================
Total params: 82,966,394
Trainable params: 82,966,394
Non-trainable params: 0

_________________________________________________________________
```

- **Learning rate** =1e-5
- **Loss**: Binary cross-entropy
- **Evaluation metric:** Accuracy
- **Epochs** = 50
- **Number of parameters:** ~83 million

Classification and evaluation

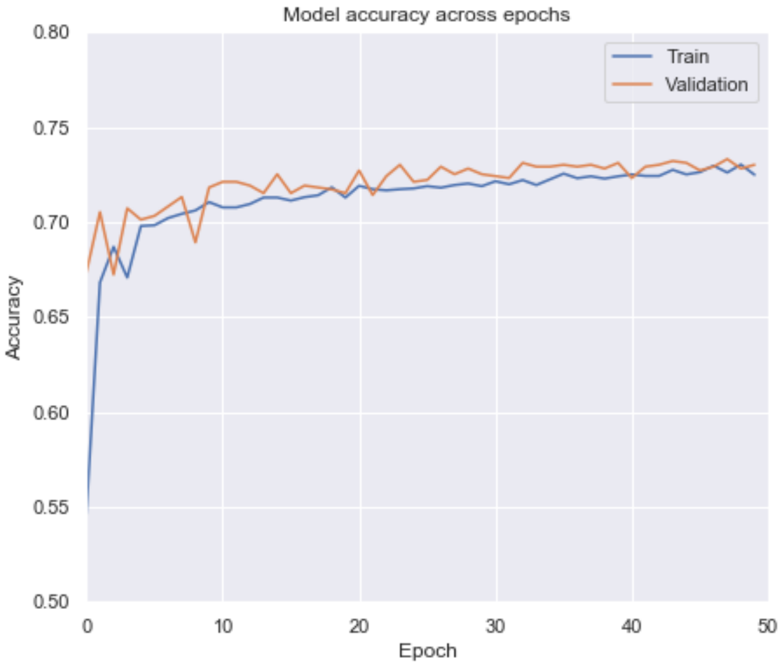
Accuracy on test images			
	Optimizers		
Batch size	Adam	SGD	RMSprop
100	0.74	0.75	0.75
200	0.75	0.75	0.74
400	0.75	0.75	0.75

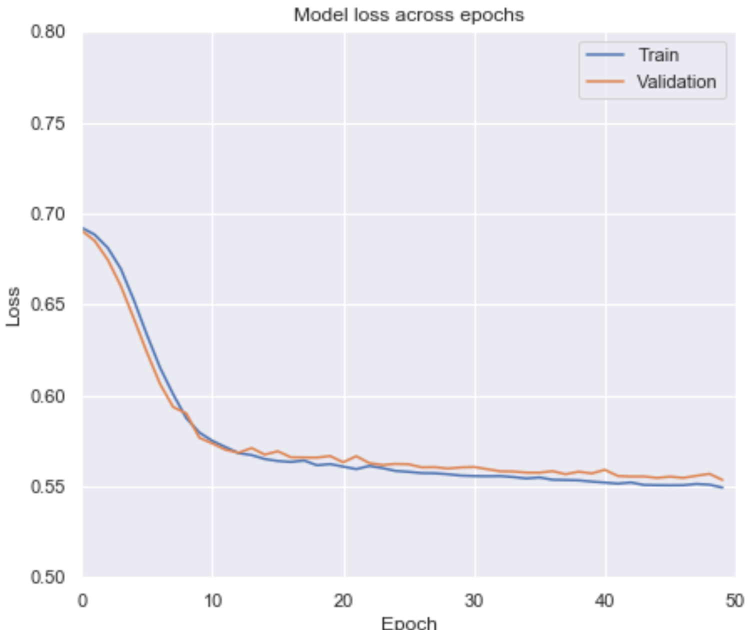
RMSprop; 100 batch size



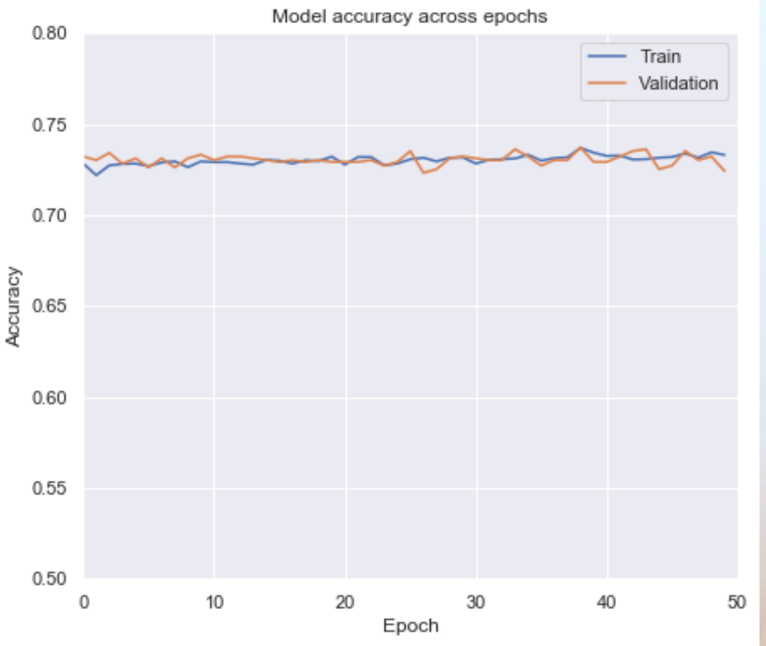
```
Model performance on the test images
Loss:  0.5441450838574426
Accuracy:  0.7454909818444797


Clasification report
             precision    recall  f1-score   support

          0       0.79      0.67      0.72       498
          1       0.71      0.82      0.76       500

avg / total       0.75      0.75      0.74       998
```

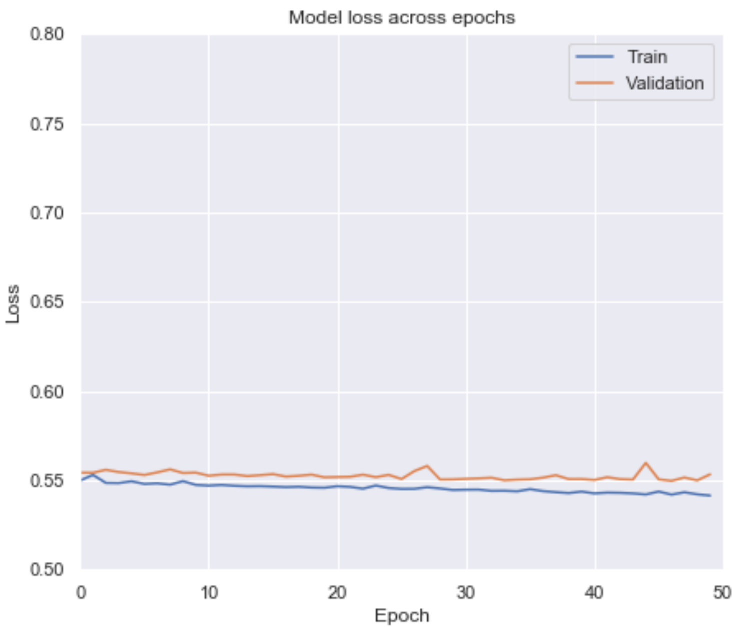
Adam, batch size= 100



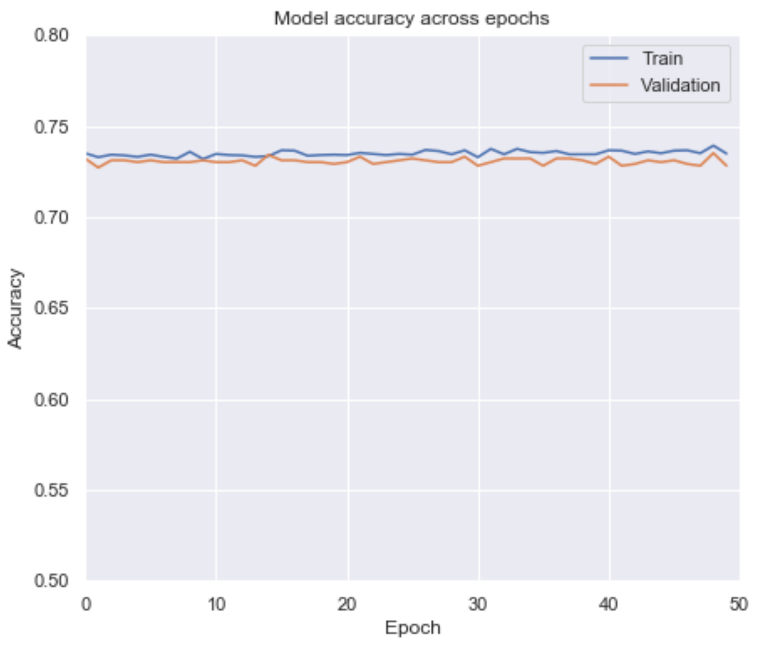


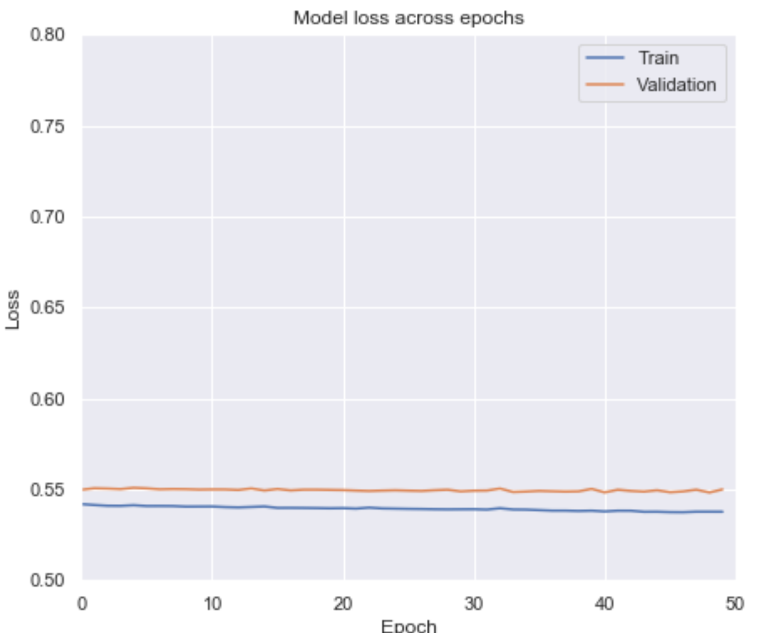
Adam, batch size= 200



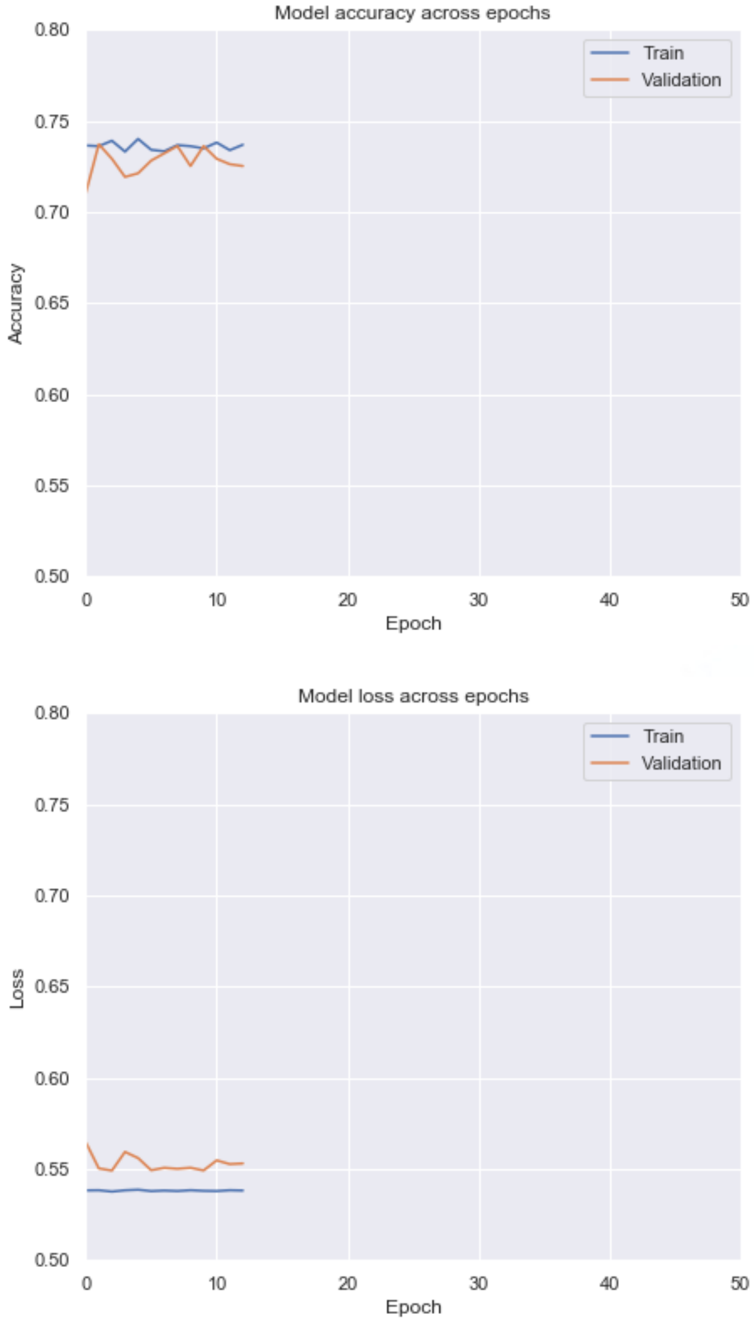


Adam, batch size= 400

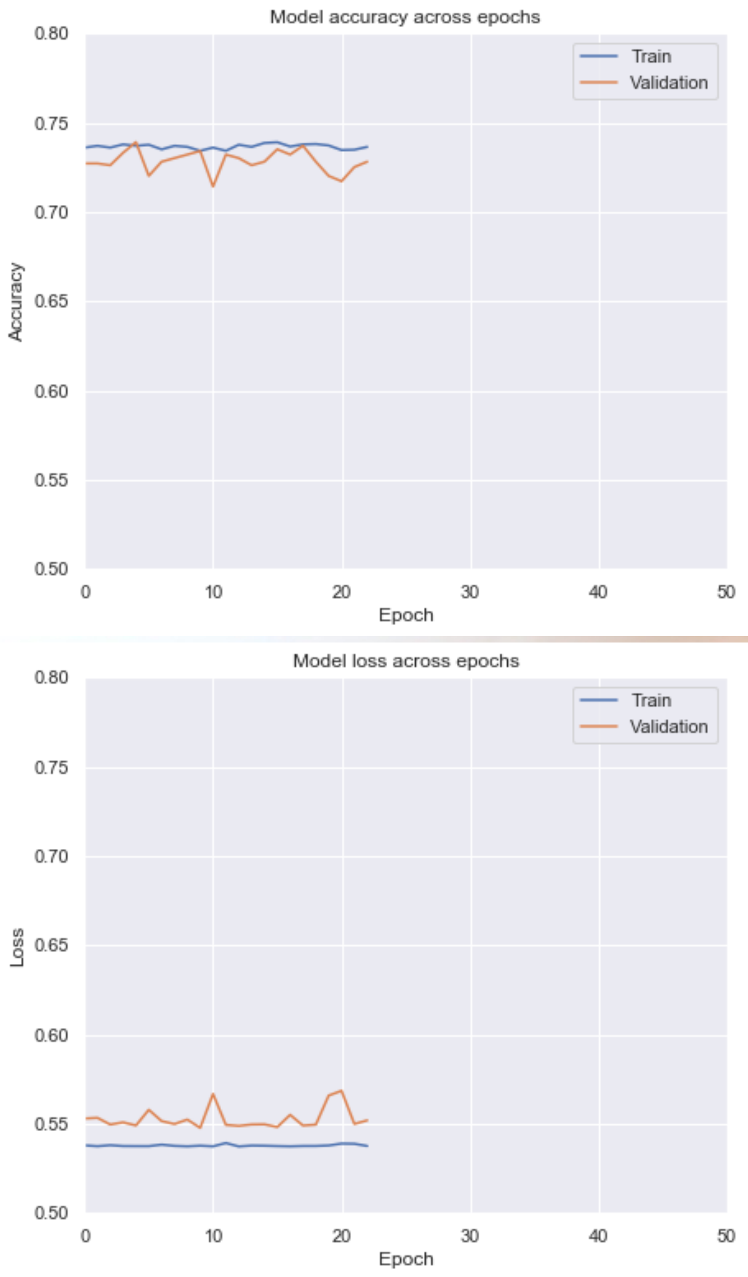




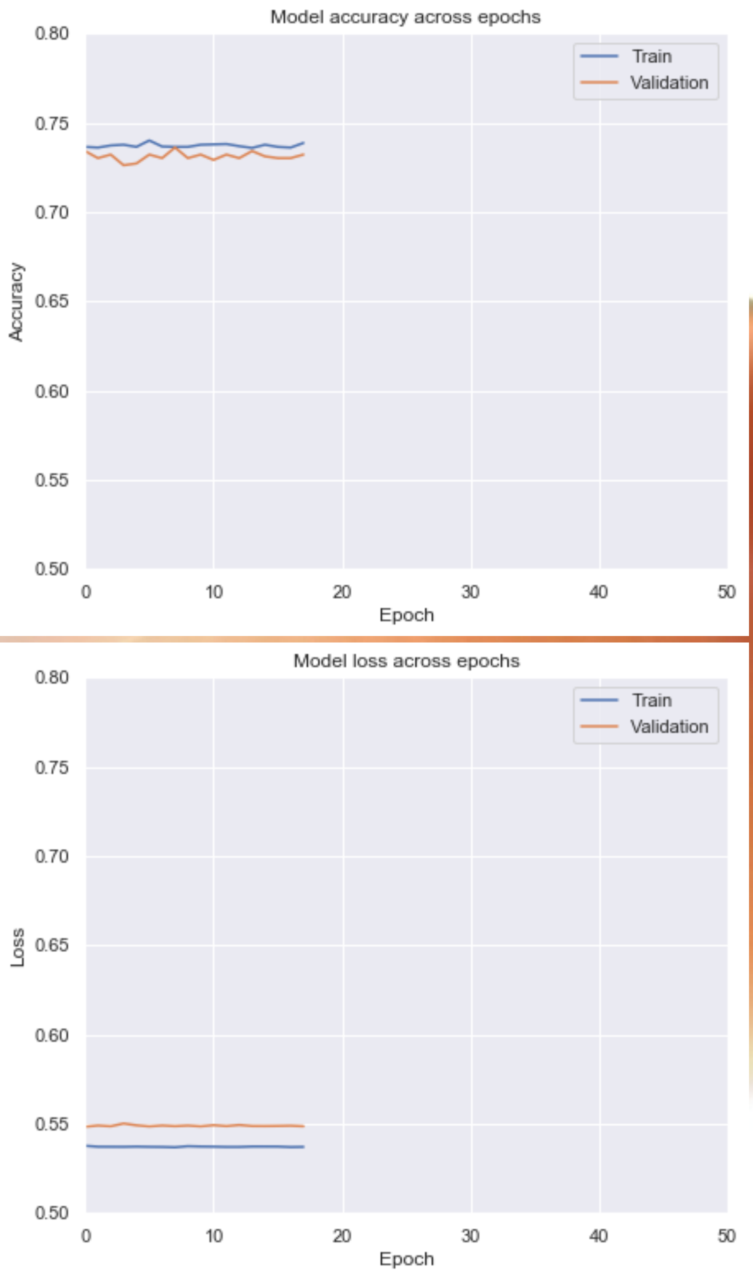
SGD, batch size= 100



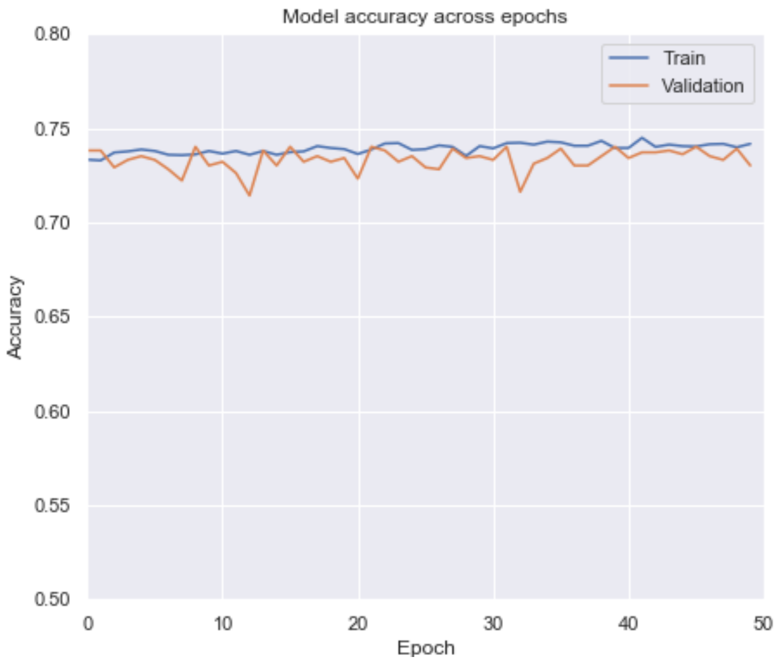
SGD, batch size= 200

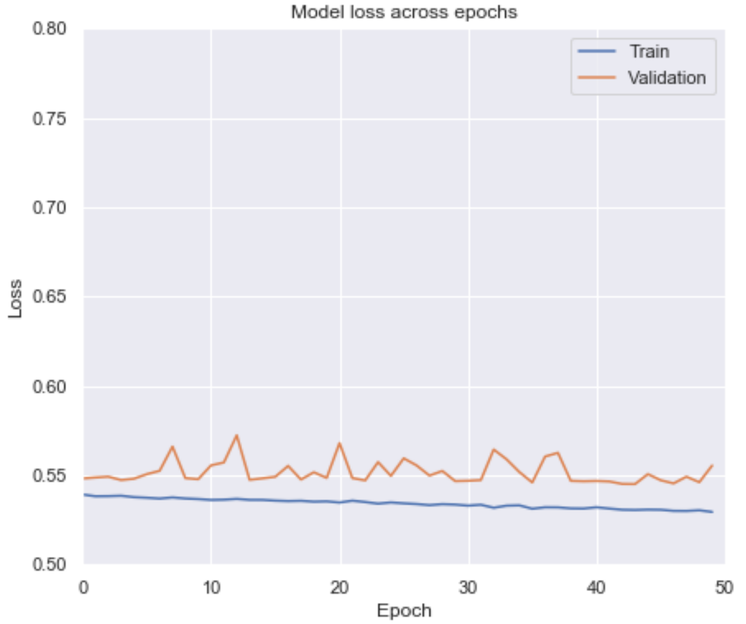


SGD, batch size= 400

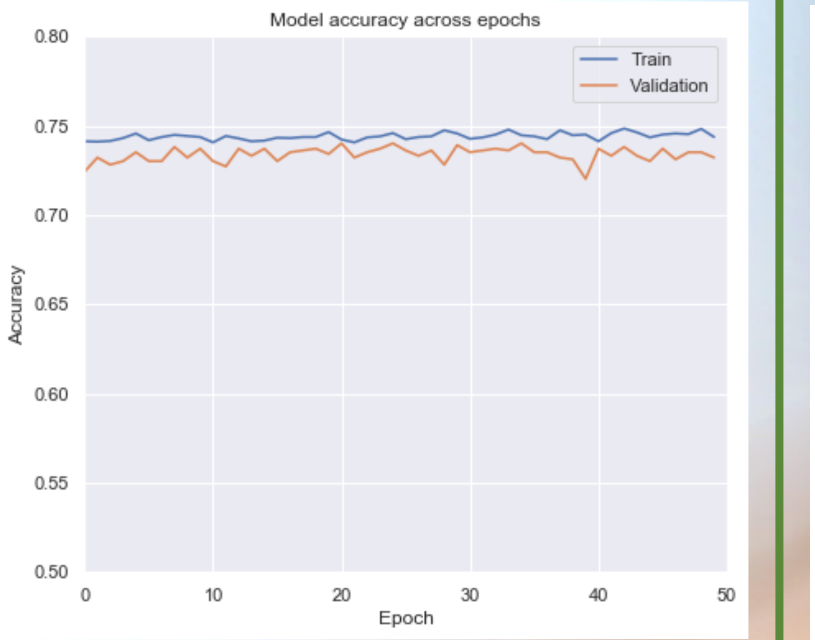


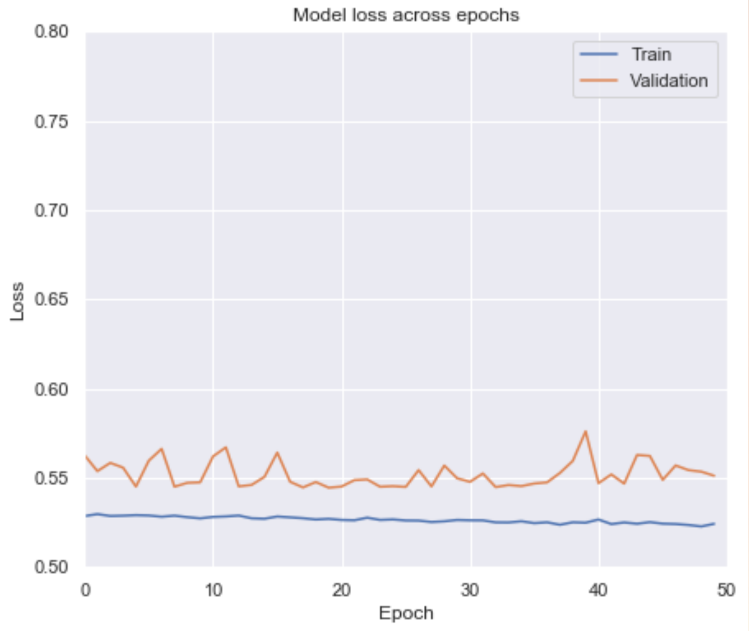
RMSprop, batch size= 100


Model accuracy across epochs

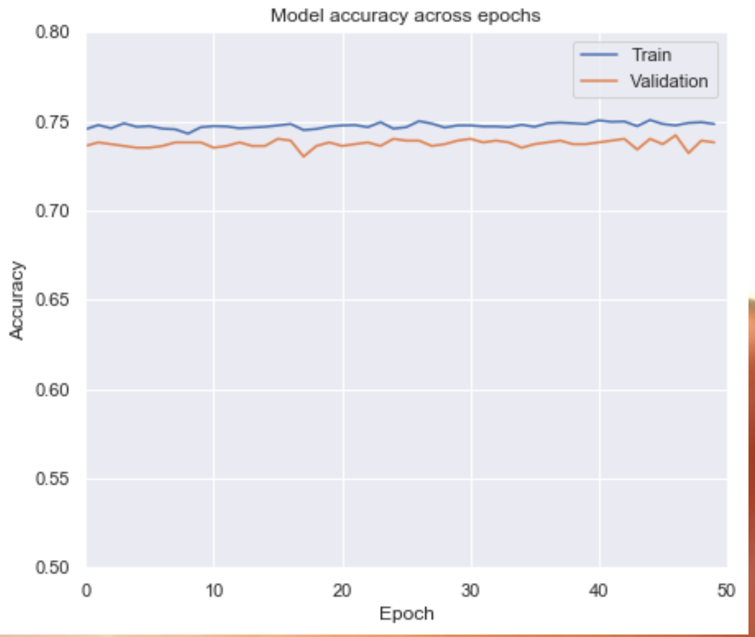

Model loss across epochs

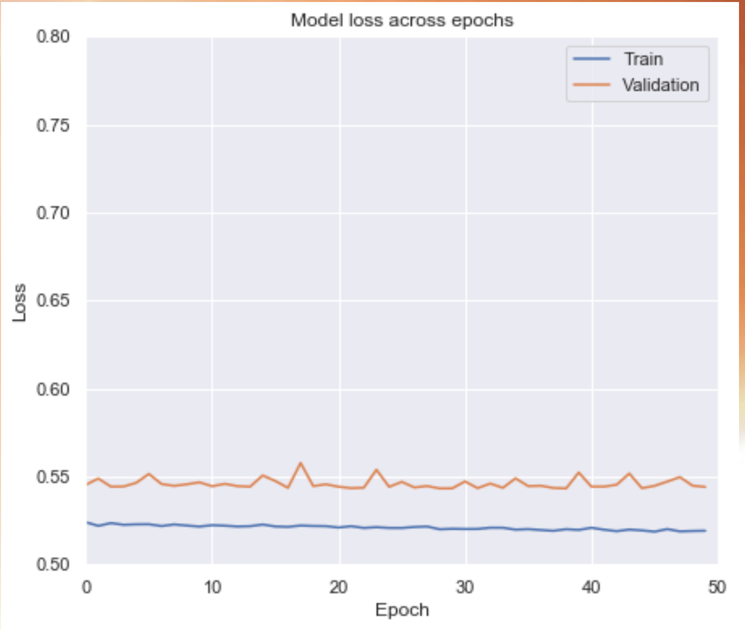
RMSprop, batch size= 200


Model accuracy across epochs

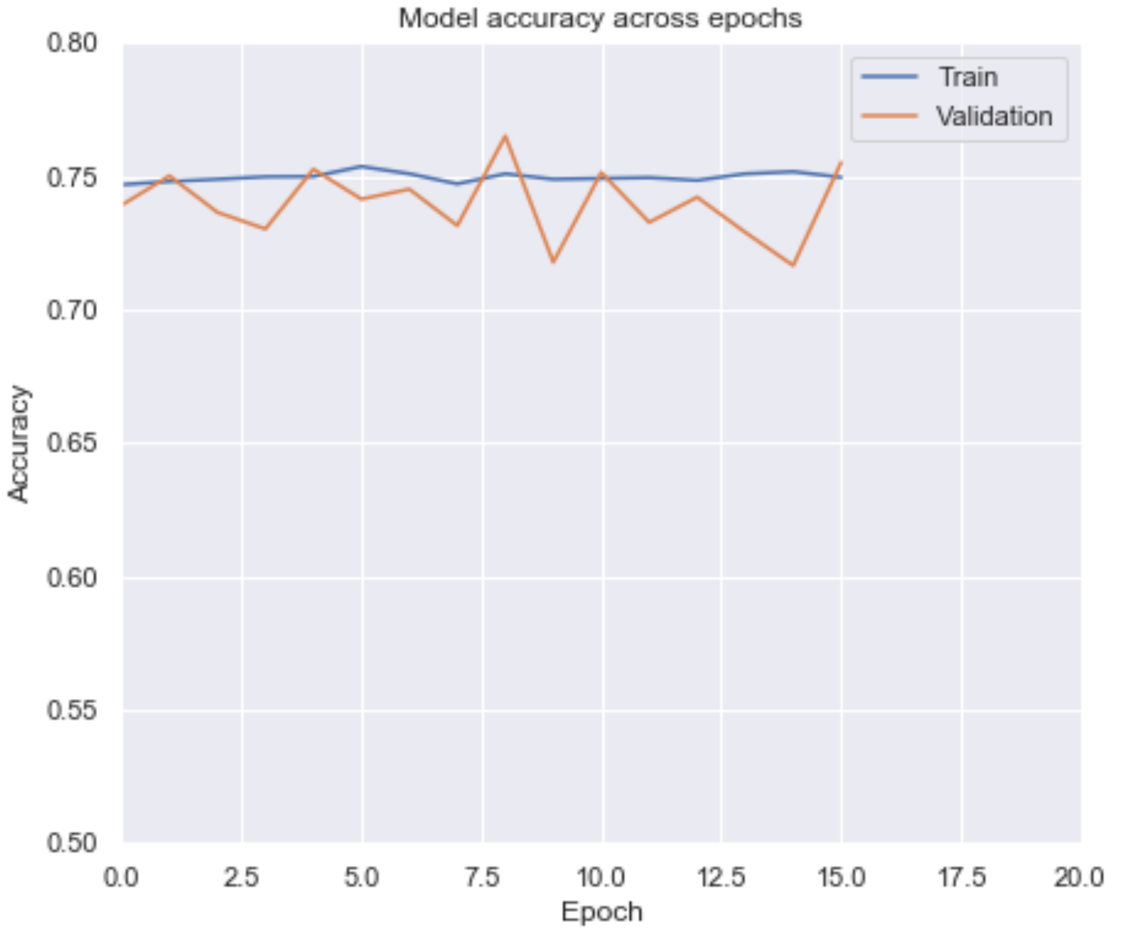

Model loss across epochs

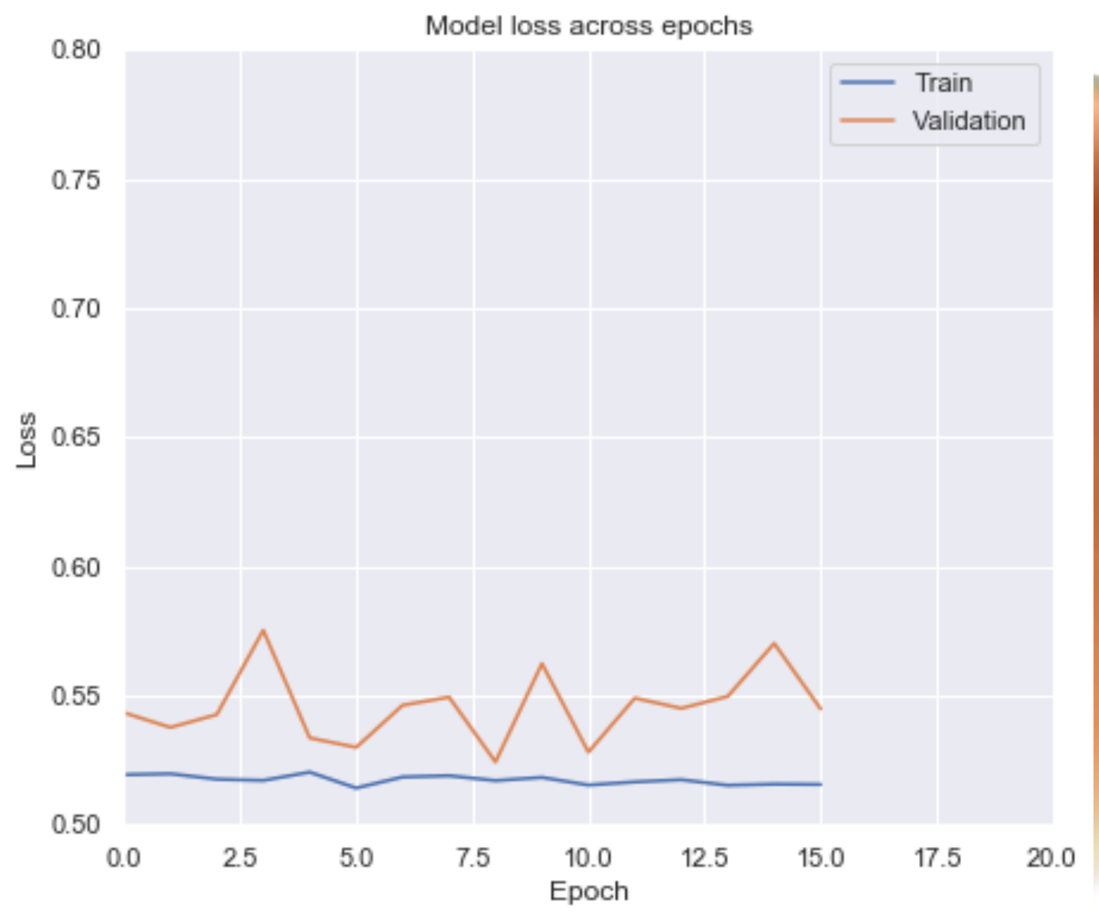
RMSprop, batch size= 400


Model accuracy across epochs

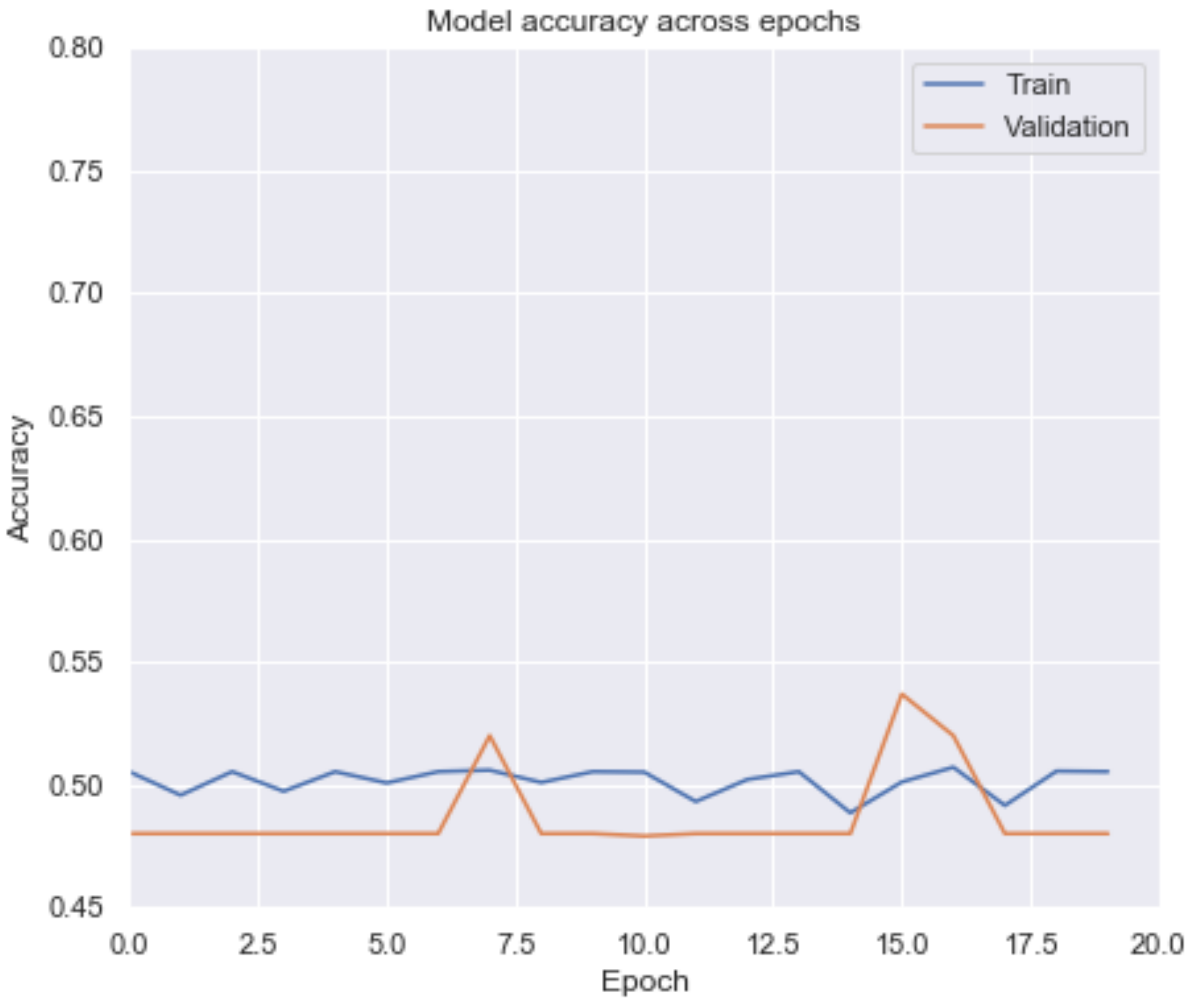

Model loss across epochs

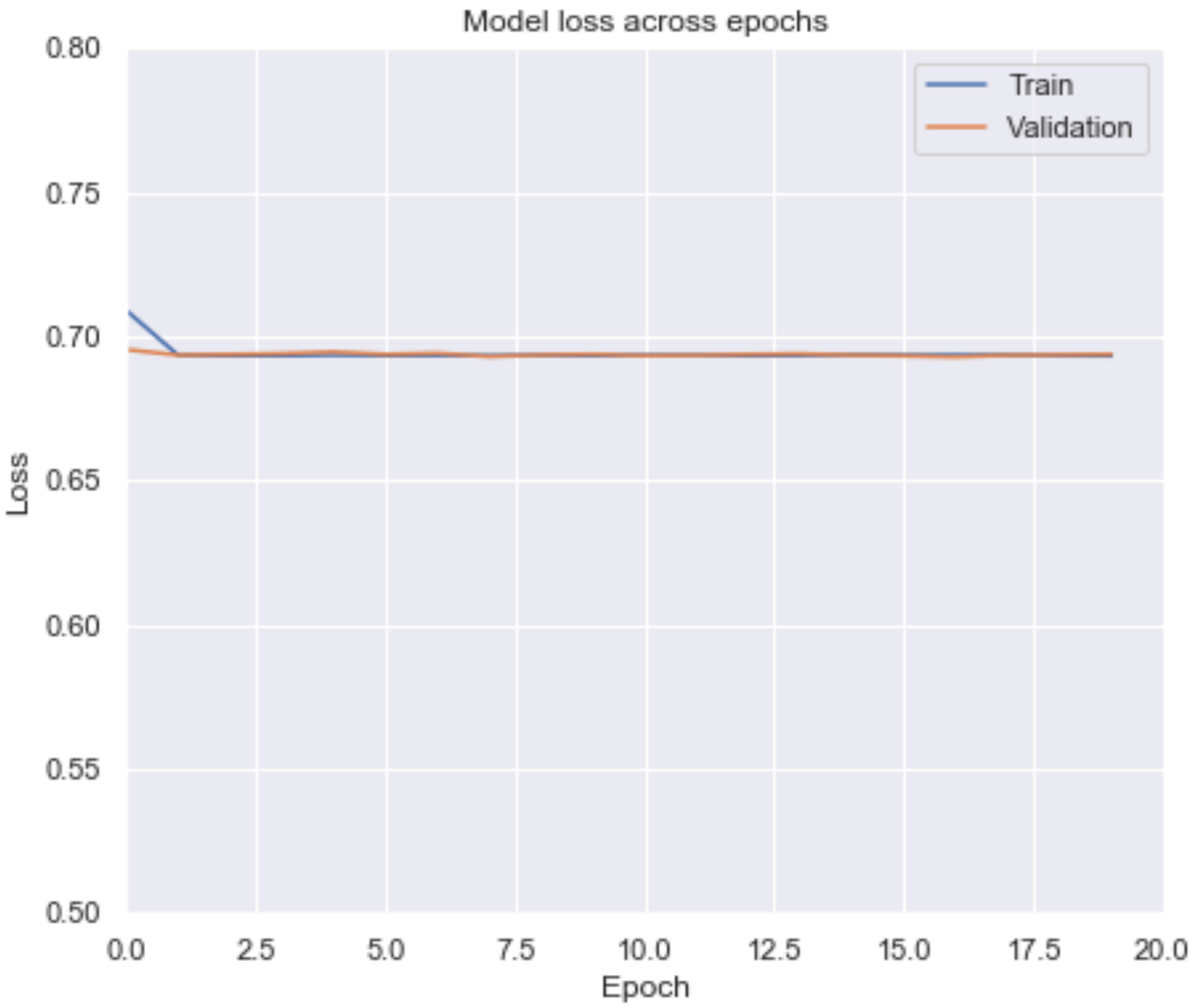
Adam, batch size=200, feature-wise center and normalization





Pre-trained VGG16

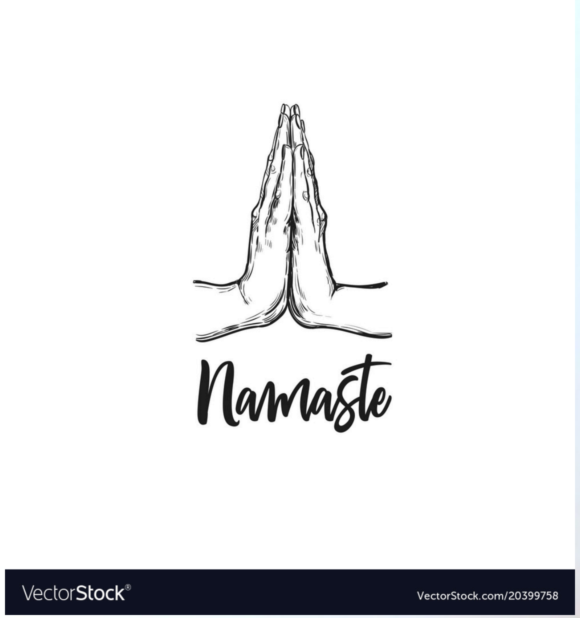




Conclusions and next steps

- **Conclusions**
  - A simple CNN achieved **~75%** accuracy, **0.82** positive recall.
  - Optimizers and batch sizes evaluated performed very consistently.
  - Feature-wise normalization improved accuracy a little bit (accuracy**~76%**)
  - VGG16 pre-trained model performed very poorly.

- **Next steps**
  - Use more images and the original size
  - Add few more convolutional and pooling layers with batch normalization and dropping input units
  - Explore few other image pre-processing techniques such as flipping, rotation, transforming images, etc.
  - Explore other pre-trained models



Thank you!

Questions, comments or suggestions are welcome!

References

1. Rui P, Kang K. National Ambulatory Medical Care Survey: 2015 Emergency Department Summary Tables. Table 27. Available from: www.cdc.gov/nchs/data/nhamcs/web_tables/2015_ed_web_tables.pdf
2. Deaths: Final Data for 2015. Supplemental Tables. Tables I-21, I-22. Available from: www.cdc.gov/nchs/data/nvsr/nvsr66/nvsr66_06_tables.pdf
3. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148
4. https://www.kaggle.com/c/rsna-pneumonia-detection-challenge