CaDoD: Phase 4 Cats vs Dogs Detector

Group 25

Martin Breth
mbreth@iu.edu

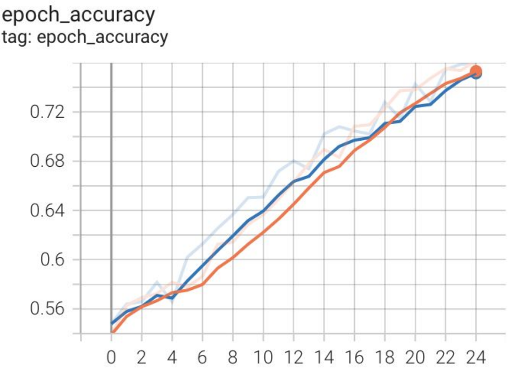
Vicente De Leon
vdeleonw@iu.edu

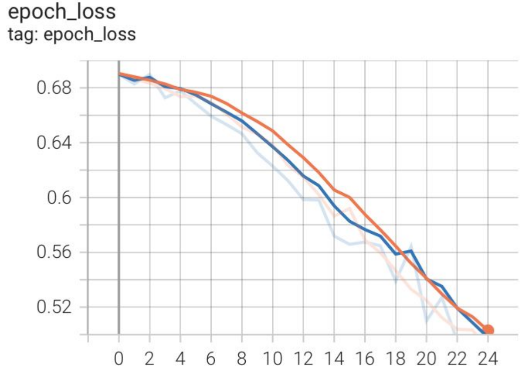
Kelly Craig
craigke@iu.edu

Courtney Payton
coupayto@iu.edu

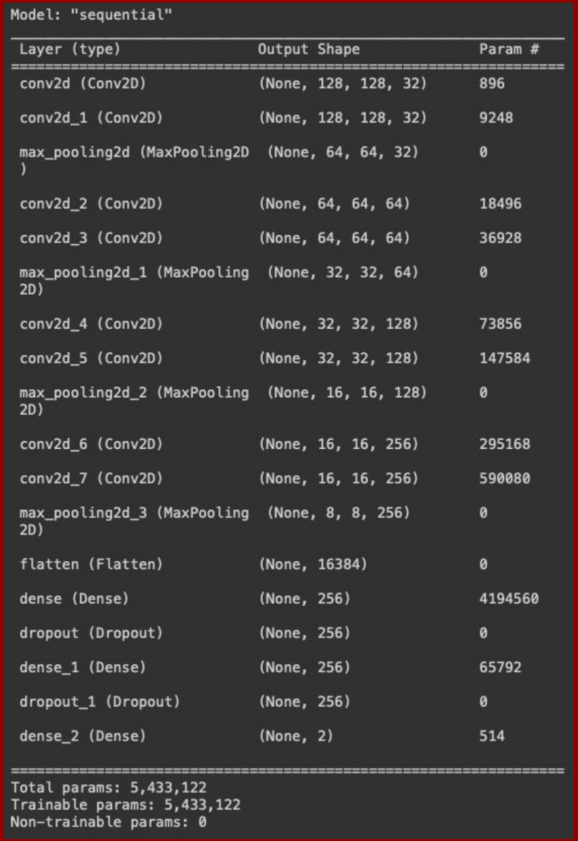
CNN Model

- Dependencies: TensorFlow, Keras, Pandas, NumPy, Sklearn.
- Input dimensions: 128x128x3(Red,Green,Blue).
- ImageDataGenerator for data augmentation/preparation.
- ReLU, Softmax, and Binary Cross Entropy Loss.
- Adam Optimizer vs RMSprop Optimizer = Adam (winner).
- Images are based on Adam Optimizer.
- 25 epochs: 76.2% train acc, 75.9% validation acc, 76% test acc.





CNN Summary

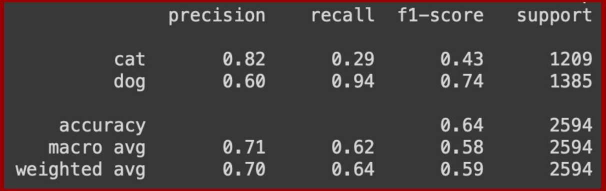


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 128, 128, 32)      896

 conv2d_1 (Conv2D)           (None, 128, 128, 32)      9248

 max_pooling2d (MaxPooling2D  (None, 64, 64, 32)       0
 )

 conv2d_2 (Conv2D)           (None, 64, 64, 64)        18496

 conv2d_3 (Conv2D)           (None, 64, 64, 64)        36928

 max_pooling2d_1 (MaxPooling  (None, 32, 32, 64)       0
 2D)

 conv2d_4 (Conv2D)           (None, 32, 32, 128)       73856

 conv2d_5 (Conv2D)           (None, 32, 32, 128)       147584

 max_pooling2d_2 (MaxPooling  (None, 16, 16, 128)      0
 2D)

 conv2d_6 (Conv2D)           (None, 16, 16, 256)       295168

 conv2d_7 (Conv2D)           (None, 16, 16, 256)       590080

 max_pooling2d_3 (MaxPooling  (None, 8, 8, 256)        0
 2D)

 flatten (Flatten)           (None, 16384)             0

 dense (Dense)               (None, 256)               4194560

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 256)               65792

 dropout_1 (Dropout)         (None, 256)               0

 dense_2 (Dense)             (None, 2)                 514

=================================================================
Total params: 5,433,122
Trainable params: 5,433,122
Non-trainable params: 0
```

CNN Phase 4 vs MLP Phase 3

CNN Model (Adam Optimizer) Highest Accuracy Phase 4:

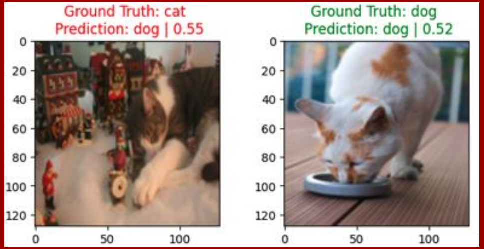
- Using 25 epochs.
- 76.2% train acc, 69% validation acc, 76% test acc.
- Major improvement since Phase 3.
- Actually classifies both cat and dog images.
- Section 4.5.5 of the notebook compares both Phases.
- It compares classification report and conf. matrix.

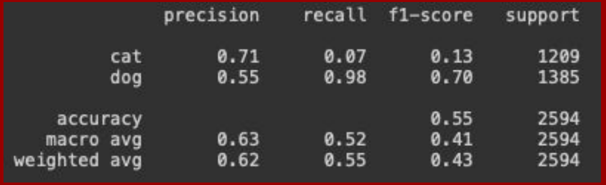


```
              precision    recall  f1-score   support

         cat       0.82      0.29      0.43      1209
         dog       0.60      0.94      0.74      1385

    accuracy                           0.64      2594
   macro avg       0.71      0.62      0.58      2594
weighted avg       0.70      0.64      0.59      2594
```

MLP Model (Adam Optimizer) Highest Accuracy Phase 3:

- Using 20 epochs.
- 55.14% train acc, 56.62% validation acc, 55.28% test acc.
- Having more trouble classifying almost all cat images.
- See section 4.5.5 of the notebook.



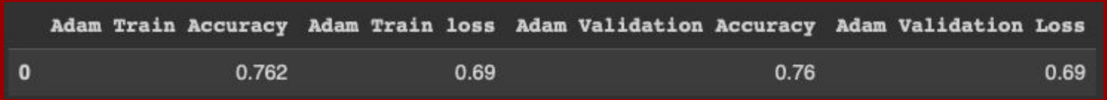


```
              precision    recall  f1-score   support

         cat       0.71      0.07      0.13      1209
         dog       0.55      0.98      0.70      1385

    accuracy                           0.55      2594
   macro avg       0.63      0.52      0.41      2594
weighted avg       0.62      0.55      0.43      2594
```

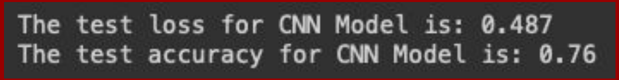
Results CNN

CNN Adam Optimizer highest scores:

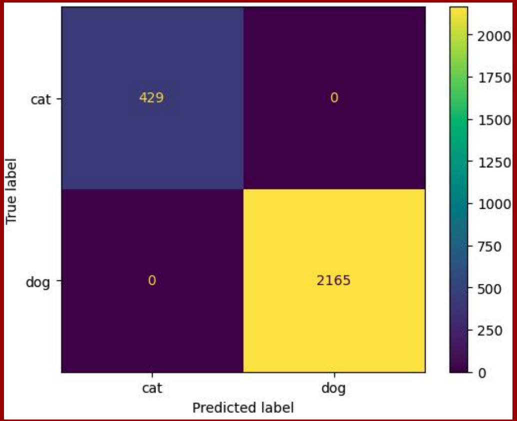
- 25 epochs.
- We can clearly see Phase 4 CNN has higher scores.



	Adam Train Accuracy	Adam Train loss	Adam Validation Accuracy	Adam Validation Loss
0	0.762	0.69	0.76	0.69

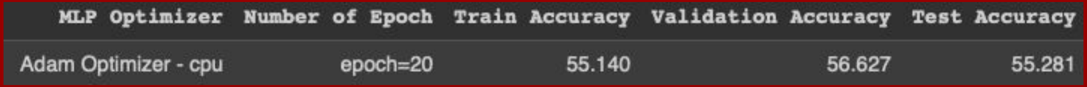


```
The test loss for CNN Model is: 0.487
The test accuracy for CNN Model is: 0.76
```

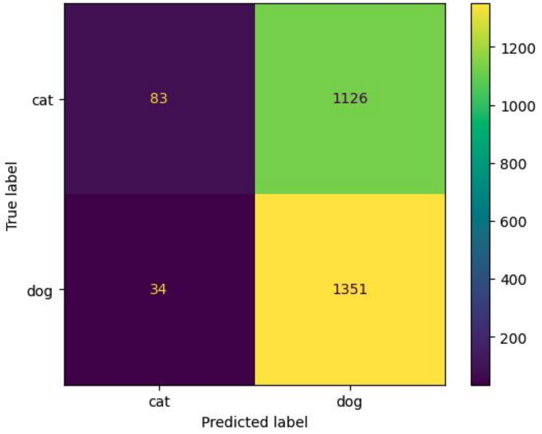


MLP Adam Optimizer highest scores:

- 20 epochs.
- Even though MLP had a 55% test accuracy, the model wasn't performing correctly.
- Possibly the most inaccurate model of all classifiers.



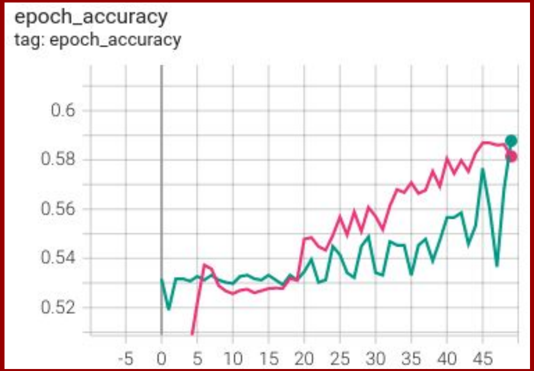
MLP Optimizer	Number of Epoch	Train Accuracy	Validation Accuracy	Test Accuracy
Adam Optimizer - cpu	epoch=20	55.140	56.627	55.281

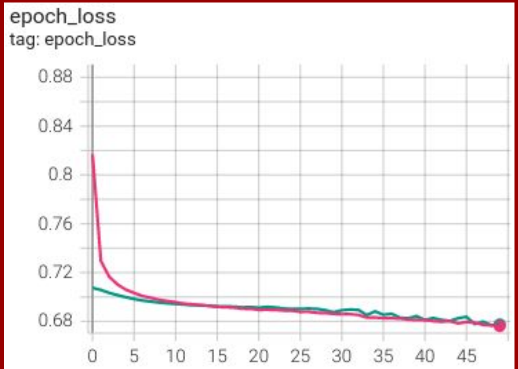


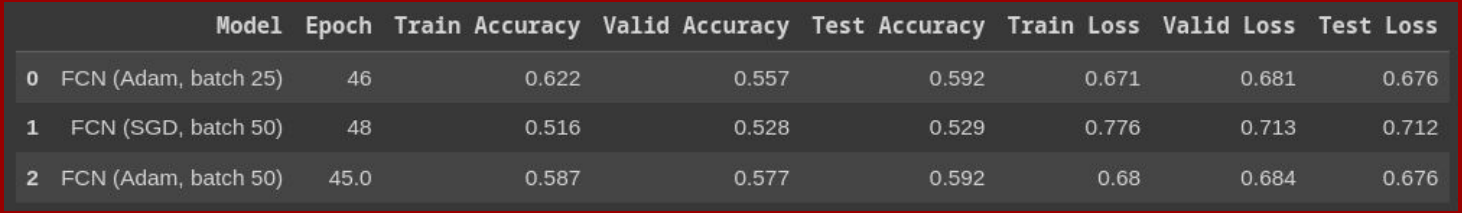
Fully Convolutional Networks (FCN)

- Similar to CNN
- Comparing Adam and SGD optimizers with different batch sizes
- More epochs needed

- TensorBoard was used for real-time results



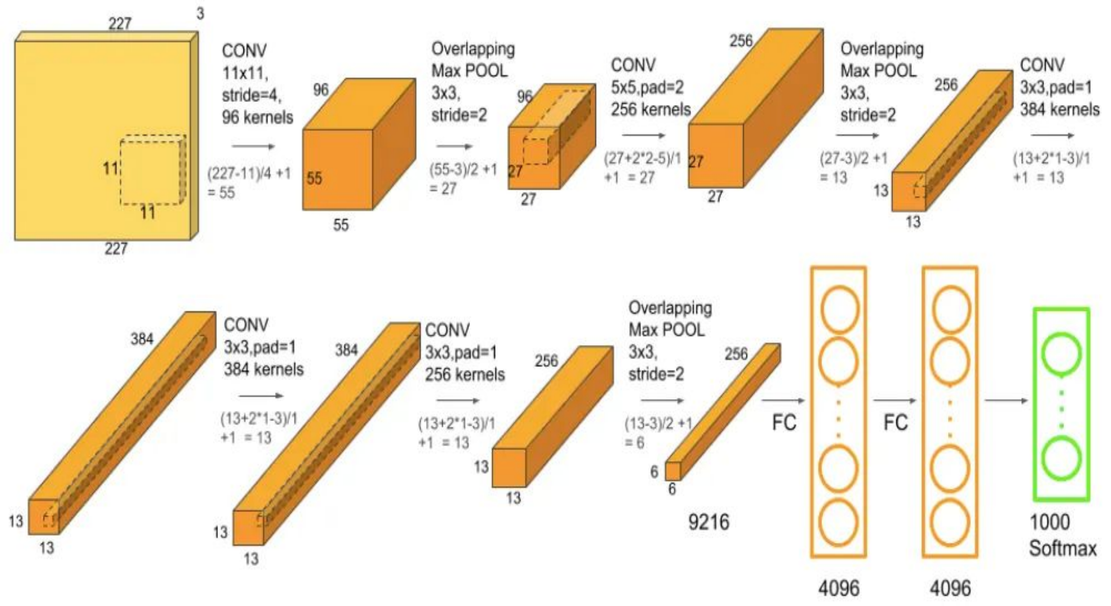




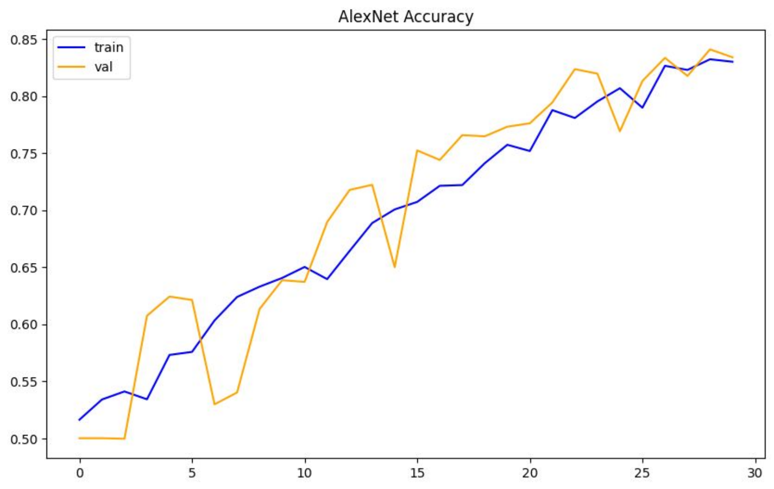
	Model	Epoch	Train Accuracy	Valid Accuracy	Test Accuracy	Train Loss	Valid Loss	Test Loss
0	FCN (Adam, batch 25)	46	0.622	0.557	0.592	0.671	0.681	0.676
1	FCN (SGD, batch 50)	48	0.516	0.528	0.529	0.776	0.713	0.712
2	FCN (Adam, batch 50)	45.0	0.587	0.577	0.592	0.68	0.684	0.676

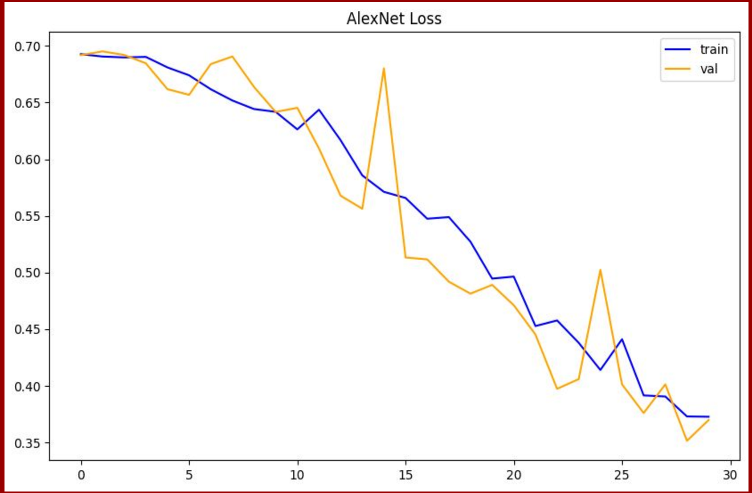
AlexNet

- Reshaped data input size to (227,227,3)
- Rotated images and applied a flip to avoid overfitting
- Fine-tuned model through testing different learning rates and optimizers(SGD, and Adam)



AlexNet Results - Best Model with SGD and Learning rate of 0.1

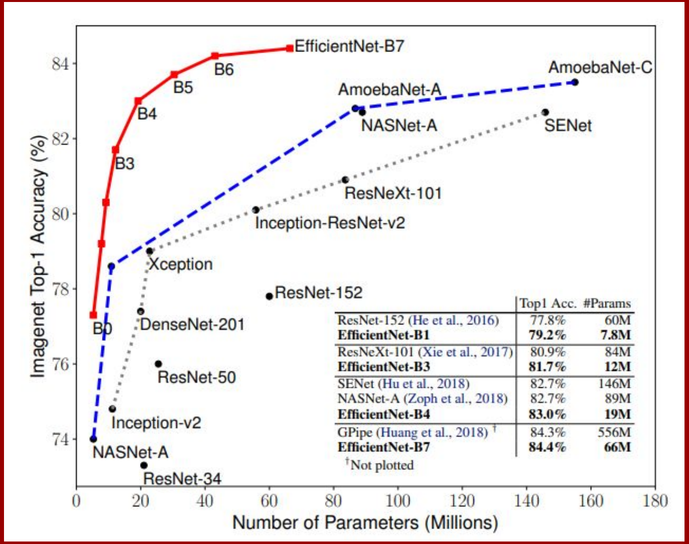




30 epochs	Accuracy	Loss
Train	81.57%	0.3866
Validation	84.73%	0.3413

Transfer Learning with EfficientNet

- Image Classification Family of Models
- EfficientNet is pre-trained on ImageNet data.
- Froze EfficientNet Model and added a top layer
- Experiment with Regularization
- Optimizer = RMSprop
- Loss function = Binary Cross Entropy

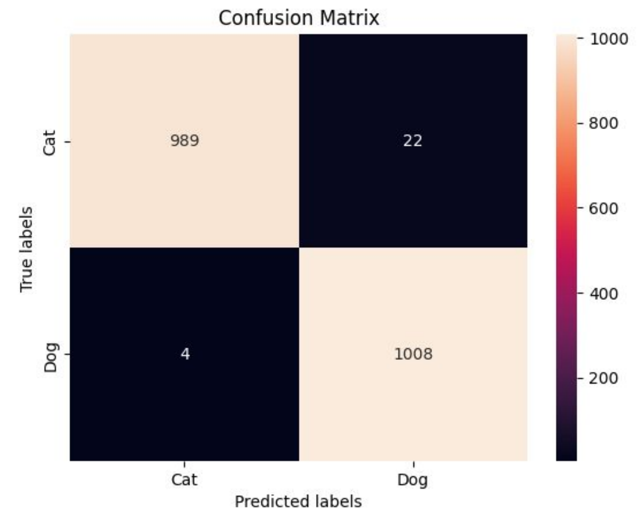


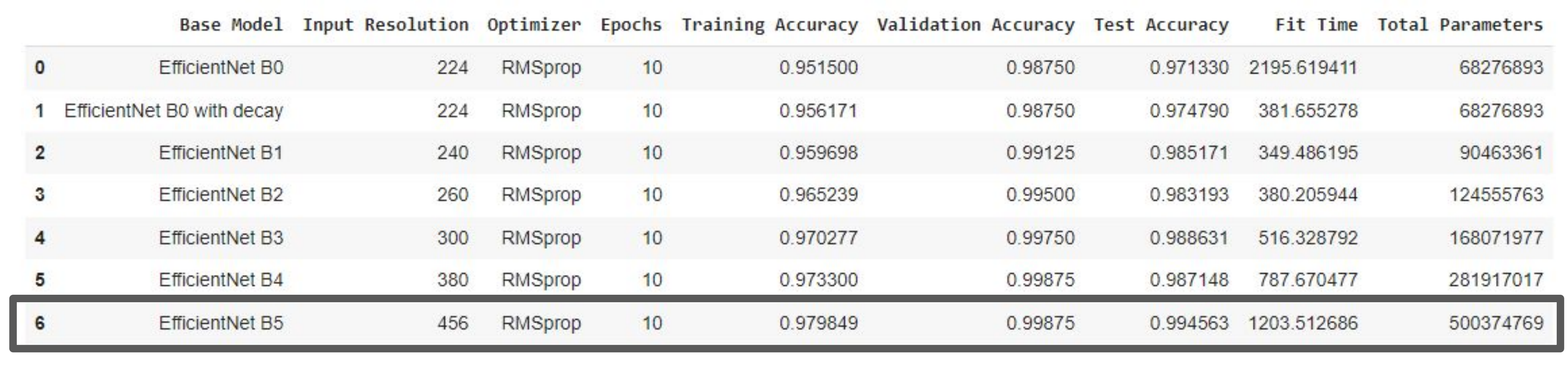
	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.2%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.7%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**83.0%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted

EfficientNet

- Best Model:
  - EfficientNet B5
  - Regularization - weight decay of 1e-6.
- Problems:
  - Memory limitations for EfficientNet B6 and EfficientNet B7





	Base Model	Input Resolution	Optimizer	Epochs	Training Accuracy	Validation Accuracy	Test Accuracy	Fit Time	Total Parameters
0	EfficientNet B0	224	RMSprop	10	0.951500	0.98750	0.971330	2195.619411	68276893
1	EfficientNet B0 with decay	224	RMSprop	10	0.956171	0.98750	0.974790	381.655278	68276893
2	EfficientNet B1	240	RMSprop	10	0.959698	0.99125	0.985171	349.486195	90463361
3	EfficientNet B2	260	RMSprop	10	0.965239	0.99500	0.983193	380.205944	124555763
4	EfficientNet B3	300	RMSprop	10	0.970277	0.99750	0.988631	516.328792	168071977
5	EfficientNet B4	380	RMSprop	10	0.973300	0.99875	0.987148	787.670477	281917017
6	EfficientNet B5	456	RMSprop	10	0.979849	0.99875	0.994563	1203.512686	500374769

Results and Discussion

CNN:

- 76.2% train acc, 69% validation acc, 76% test acc.
- Major improvement since Phase 3.

FCN:

- Best model: Adam Optimizer gave best results
- Accuracy: 58.7% train 57.7% valid and 59.2% test

AlexNet:

- Best Model: Stochastic Gradient Descent (SGD), and learning rate: 0.1
- 81.6% train acc, 84.7% validation acc
- Major improvement since Phase 3

EfficientNet

- Best Model: Rmsprop optimizer, EfficientNetB5
- 97.9% train acc, 99.9 val acc, and 99.5 test accuracy