Facial Keypoint Detection

Our experiments to beat Kaggle Leaderboard

Final Project Presentation - W207 Fall 2018

Agenda

- Facial Keypoint Dataset - Exploration
- Image Augmentation
- Deep Learning Approach
- Experiments / Hyper-parameter Tuning
- Popular CNN Architectures
- Transfer Learning
- Ensemble Methods
- Results & Future Enhancements

Overview: Facial Keypoint Prediction (FKP)

- Predict keypoint positions on face images
- A challenging problem in the field of Computer Vision
- Facial features vary greatly between individuals
- Dependent on angle, size, illumination conditions etc.
- Facial keypoint detection - critical to face recognition

Data (Kaggle Dataset)

Dataset: https://www.kaggle.com/c/facial-keypoints-detection/data

01 Training data: Facial image
Labels: 15 pairs (x & y coordinates) of facial keypoints

02 training.csv: list of training 7049 images. Each row contains the (x,y) coordinates for 15 keypoints and image data as row-ordered list of pixels.

test.csv: list of 1783 test images. Each row contains ImageId and image data as row-ordered list of pixels

Data Cleaning & Exploration

Steps involved:

- Load .csv file; extract data and labels
- Image data: 96x96 pixels
- Normalize pixel values to be between 0 and 1
- Perform 80/20 (training / validation) split
- Records with missing labels found

Dealing with Missing Values

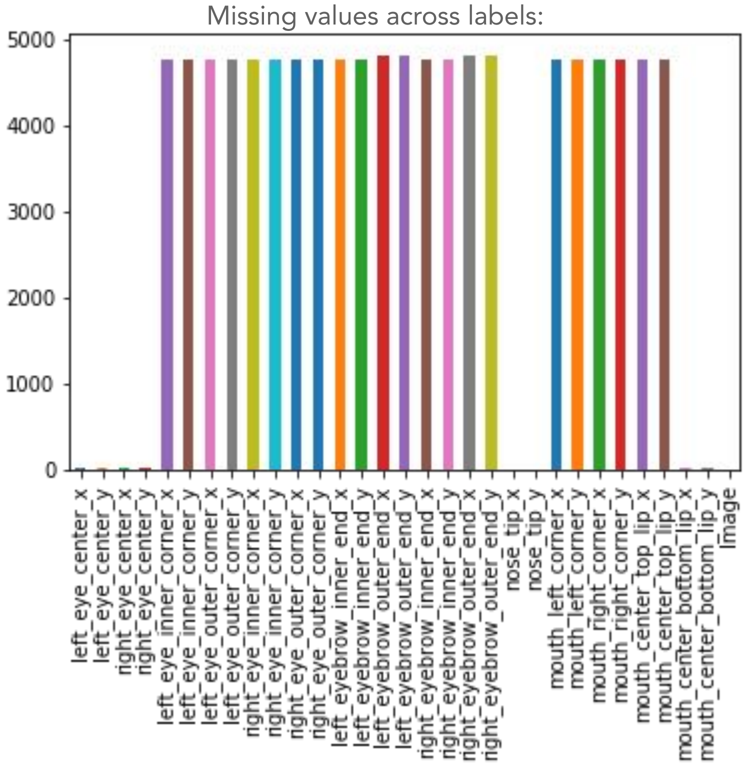
Two approaches for modeling:

#1 Eliminate training data where labels are missing (across 30 data points); build model using the available training data (total records: 1712 for training)

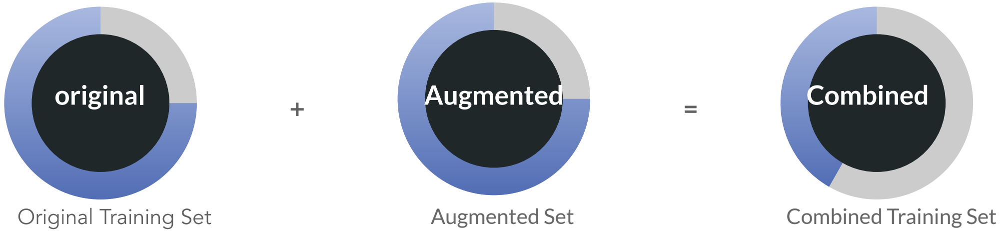
#2 Build 2 sets of datasets:

1. Training data, where 8 out of 30 labels are available across all records (7000 records)
2. Training data where remaining 22 labels are available across all records (2155 records)

Advantage of #2: More training data available

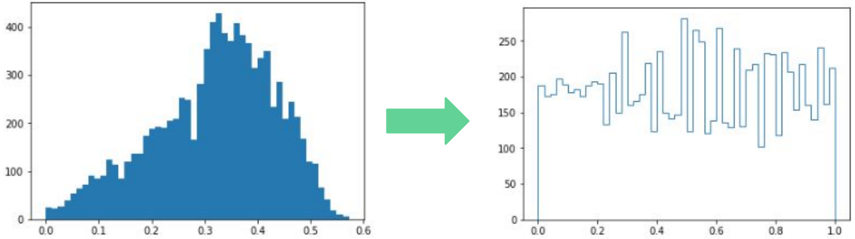


Augmentation



A common practice for image processing and involves one or more of the following techniques:

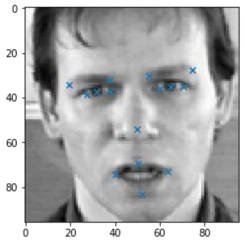
- Scale / normalization
- Rotation
- Flip - horizontal / vertical
- Histogram Equalization
- Blurring (Gaussian / Median)
- Shift (positional change)
- Contrast Reduction

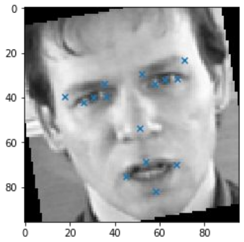


Augmentation - a few examples

Experiment 1: Rotation

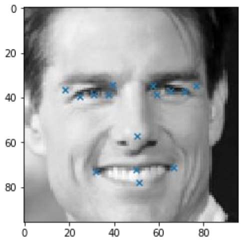
- Rotation of image
- Transformation applied on labels

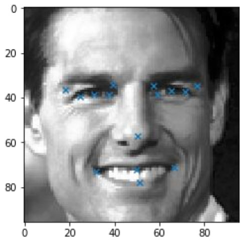




Experiment 2: Histogram Equalization

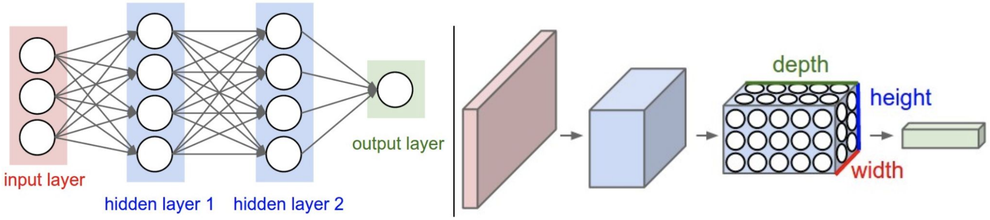
- Equalize pixel values
- No transformation applied on labels



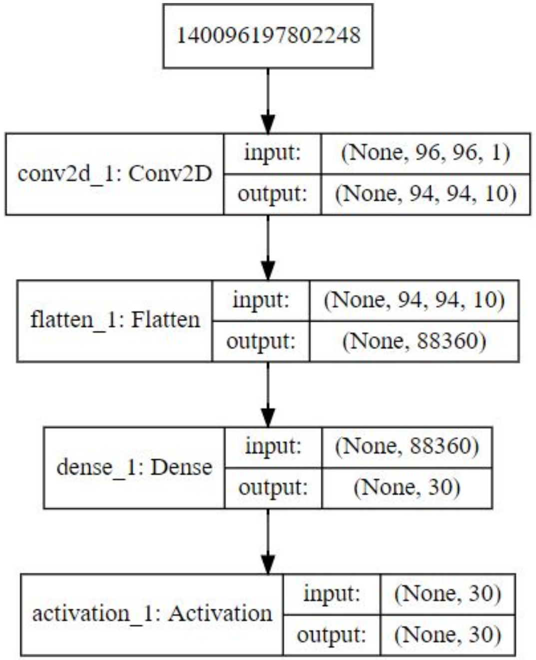


Approach - Deep Learning

- Regular Neural Networks vs. Convolutional Neural Networks



- Baseline Model
  - No data augmentation, removed missing data
  - 1 Conv Layer + 1 Fully Connected Layer
  - Baseline validation RMSE ~8



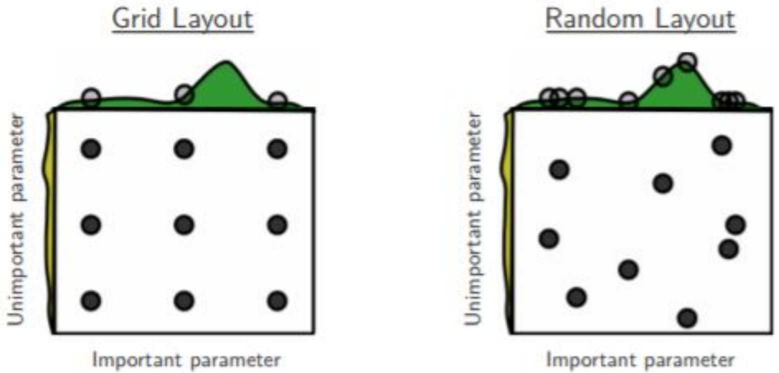
Experiments - Grid Search vs. Random Search

Grid Search

- Exhaustively search all parameter combinations

Random Search

- Sample a given number of candidates from a parameter space with a specified distribution

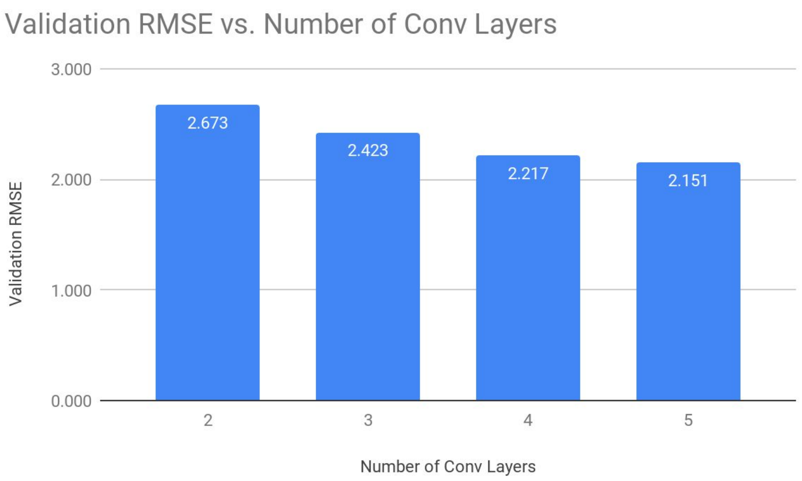


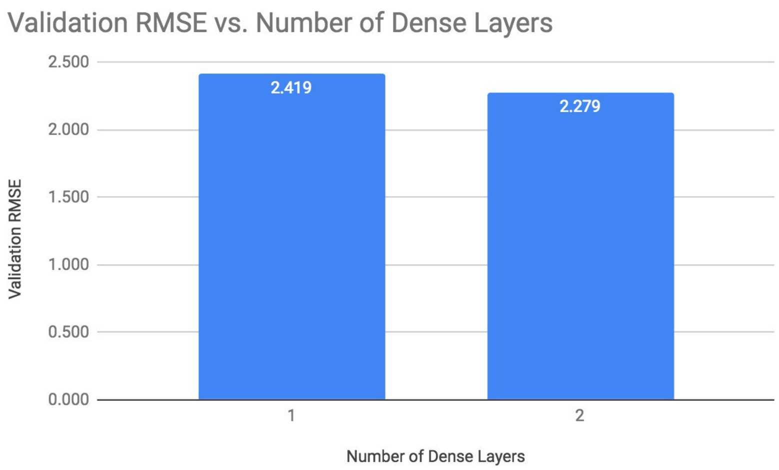
Our Approach

- Grid search of 16 combinations
- Random search of 200 combinations
    - Basic architecture setup from initial model testing (LeNet-5, AlexNet, VGG-16)
    - Varying number of layers, filter size, number of neurons, activation functions, dropout layers, etc.
- Selected top models to continue tuning

http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

Experiments - Number of Layers



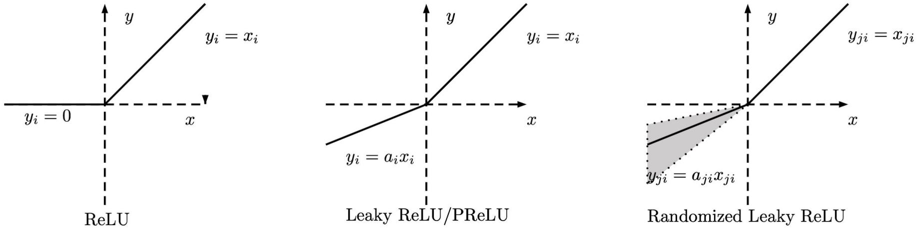


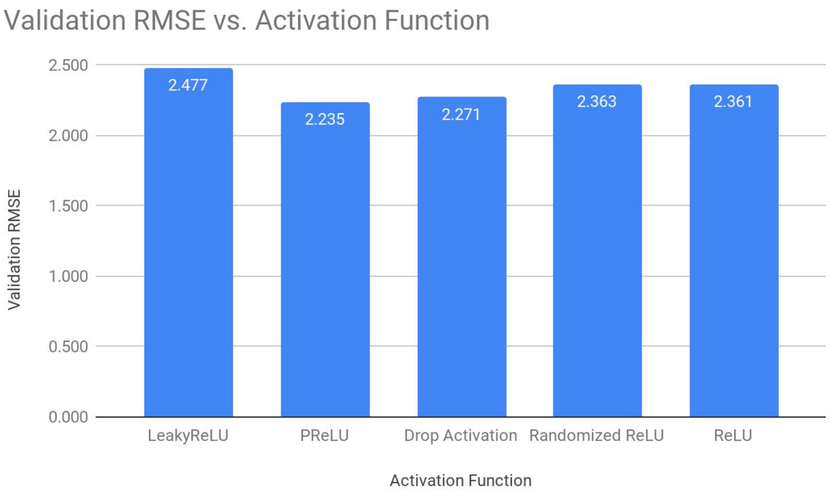
Go deep or go home -

Performance significantly improves with more layers

Experiments - Activation Function

- ReLU
- LeakyReLU
- Parametric ReLU
- Randomized ReLU
- Drop-Activation





https://arxiv.org/abs/1811.05850
https://arxiv.org/pdf/1505.00853v2.pdf

Experiments - Batch Size

"Don't Decay the Learning Rate, Increase the Batch Size"

Final Model Architecture

Conv Layer @8 + BatchNormalization
MaxPooling
Conv Layer @32 + BatchNormalization
MaxPooling
Conv Layer @64 + BatchNormalization
Conv Layer @96 + BatchNormalization
Conv Layer @128 + BatchNormalization
MaxPooling
Conv Layer @256 + BatchNormalization
MaxPooling
Dense @96
Dense @96
Dense @32
Dense @number of output points

20 Epochs @64

20 Epochs @128

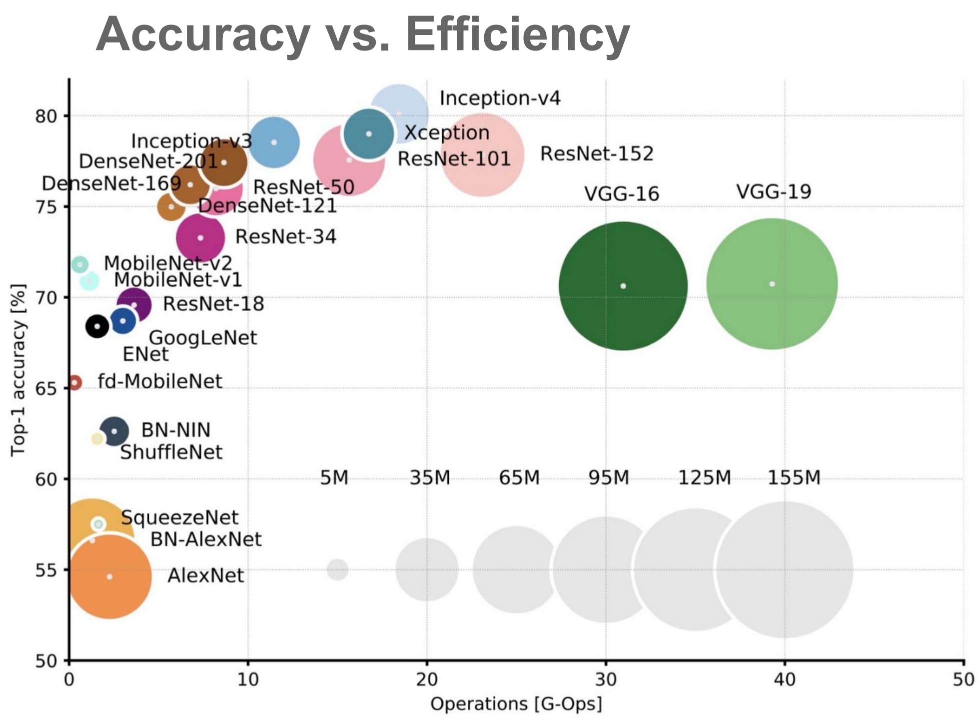
20 Epochs @256

60 Epochs @512

Final Model

https://openreview.net/pdf?id=B1Yy1BxCZ

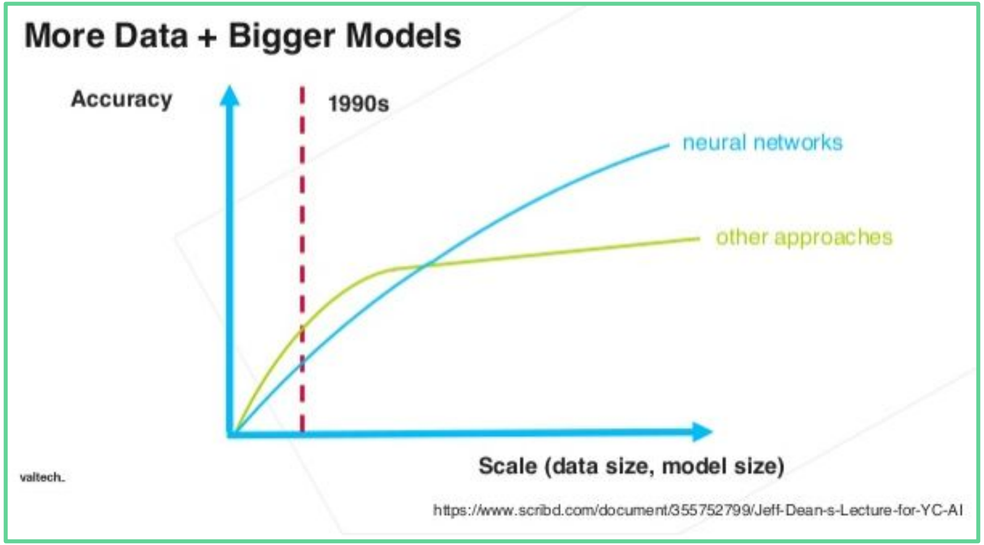
Popular CNN Architectures



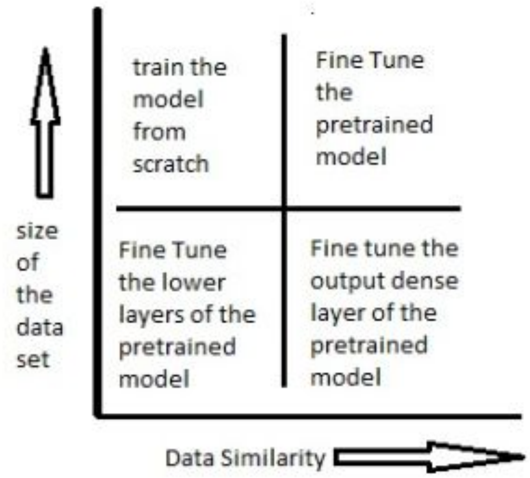
Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	99 MB	0.749	0.921	25,636,712	168
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-

The top-1 and top-5 accuracy refers to the model's performance on the ImageNet validation dataset.

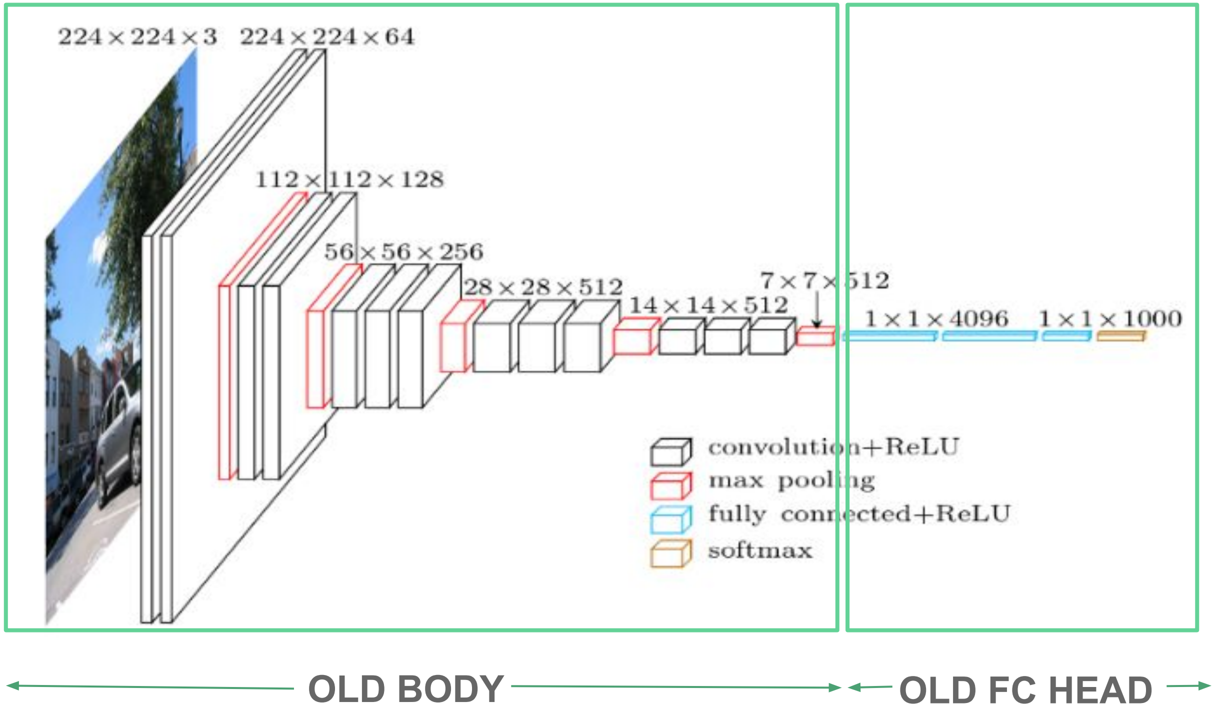
Transfer Learning

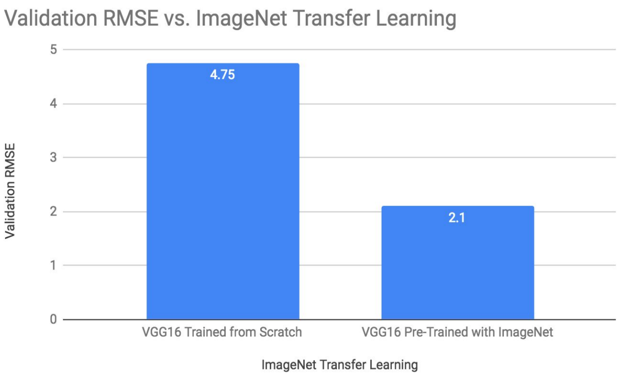


Feature Extraction vs. Fine Tuning



VGG 16 Transfer Learning Results





Ensembling

Model 1	6 conv layer (8-32-64-96-128-256) + 3 dense layer (96-96-32)
Model 2	6 conv layer (8-64-96-128-256-324) + 3 dense layer (256-96-64)
Model 3	6 conv layer (28-64-64-64-128-128) + 3 dense layer (256-96-64)
Model 4	Transfer learning

Jensen's Inequality

The convex combined ensemble will have error less than or equal to the average of all models.

Kaggle Submissions	Private Score	Public Score
Model 1 (Best)	1.91591	2.19943
Model 2	2.11234	2.39057
Model 3	2.13715	2.39698
Model 4 - Transfer Learning	2.10209	2.3543
Ensemble 1+2	1.87646	2.17739
Ensemble 1+2+3	1.87597	2.1763
Ensemble 1+2+3+4	1.82864	2.13138

Results - Experiments Recap

Data Pre-Processing

Data Augmentation:

- Mirroring
- Histogram Equalization
- Rotation
- Contrast Reduction

Missing Data Schemes

- Drop Data
- Augment Data with Image Processing
- 2 Models

CNN Architectures

Popular Architectures

Hyper Parameter Tuning

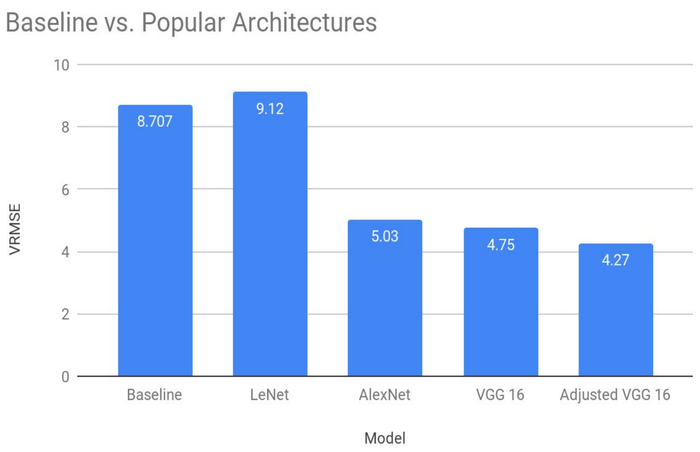
- Conv, Dense Layers
- Dropout Rates
- Activation Functions

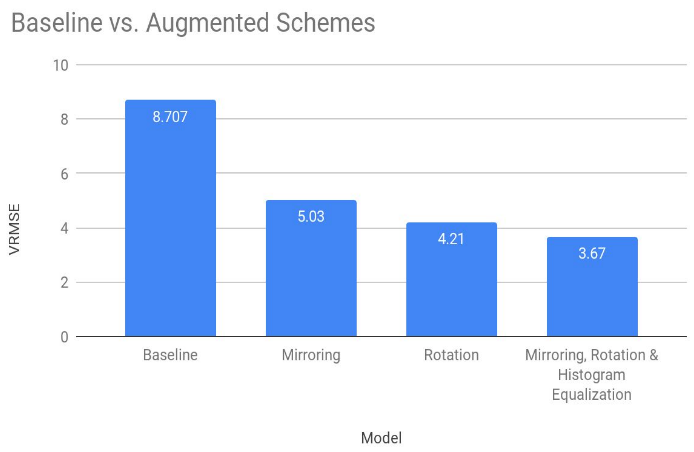
Transfer Learning

- VGG16
- Xception

Ensemble Methods

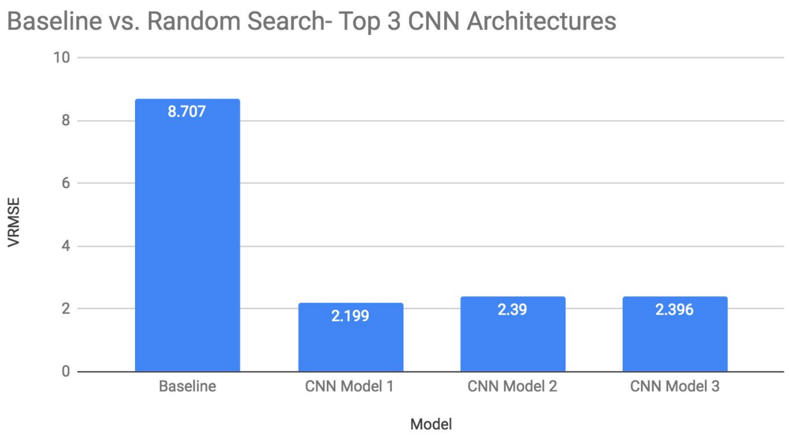
Results - Popular Architectures & Augmentation

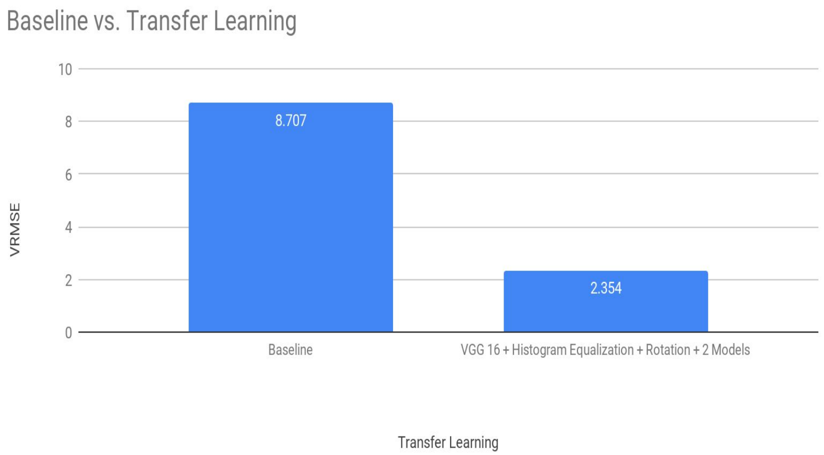




- VGG & Adjusted VGG worked very well
- Rotation & Histogram Equalization

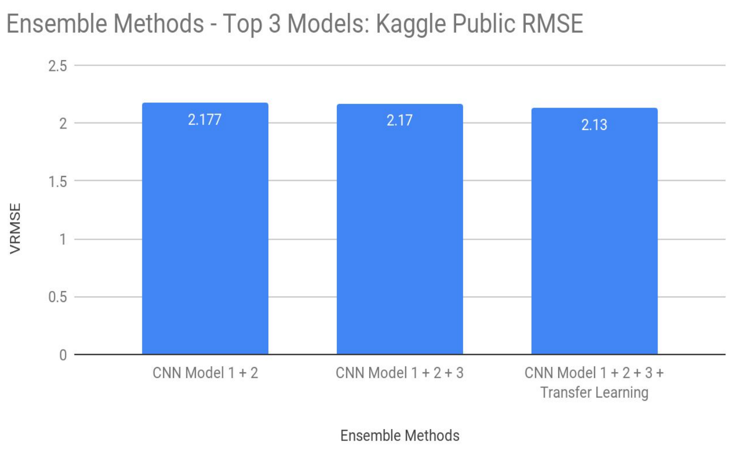
Results - Random Search & Transfer Learning

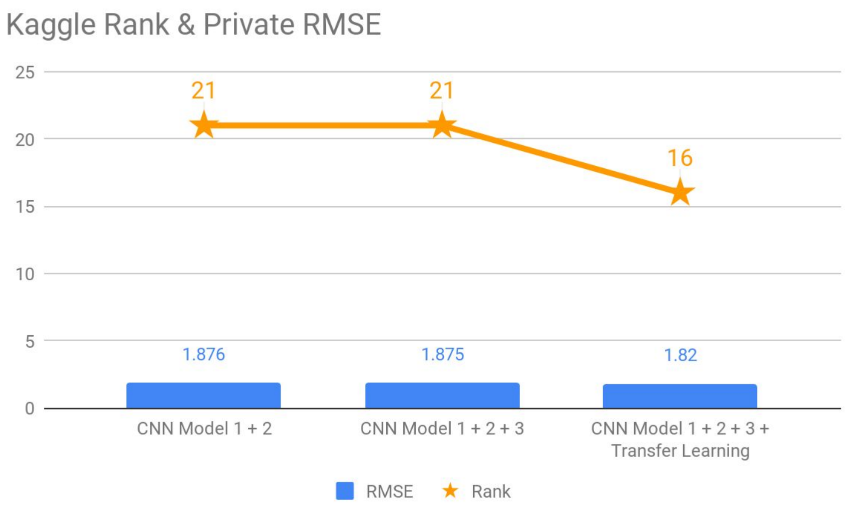




- Random Search:
  - CNN Model 1: 6 conv layer (8-32-64-96-128-256) + 3 dense layer (96-96-32)
  - CNN Model 2: 6 conv layer (8-64-96-128-256-324) + 3 dense layer (256-96-64)
  - CNN Model 3: 6 conv layer (28-64-64-64-128-128) + 3 dense layer (256-96-64)
- Transfer Learning:
  - VGG 16 + HE + Rotation + 2 Models

Results - Ensemble Methods





Final Placement: 16 out of 175 - We beat previous teams!

Future Enhancements

Data Pre-Processing

Data Augmentation:

- Mirroring
- Histogram Equalization
- Rotation
- Contrast Reduction

Missing Data Schemes

- Drop Data
- Augment Data with Image Processing
- 2 Models

Enhancements:

- Keras Preprocessing

CNN Architectures

Popular Architectures

Hyper Parameter Tuning

- Conv, Dense Layers
- Dropout Rates
- Activation Functions

Transfer Learning

- VGG16
- Xception

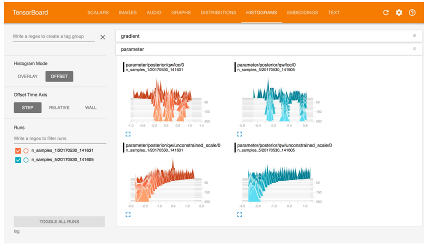
Ensemble Methods

Enhancements:

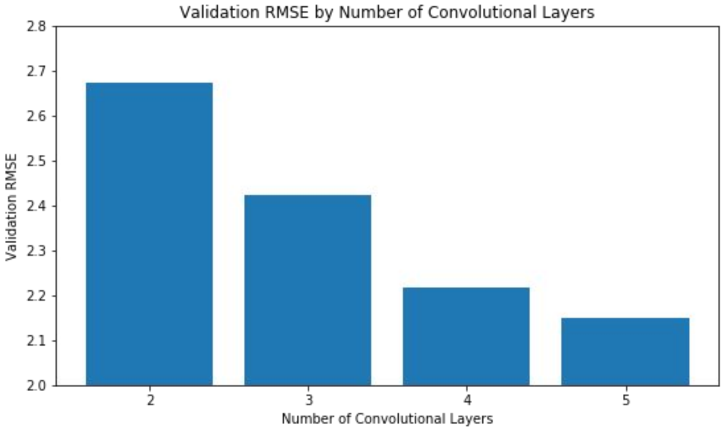
- Ensemble Methods
- Advanced Tuning (Random, Optimizers)

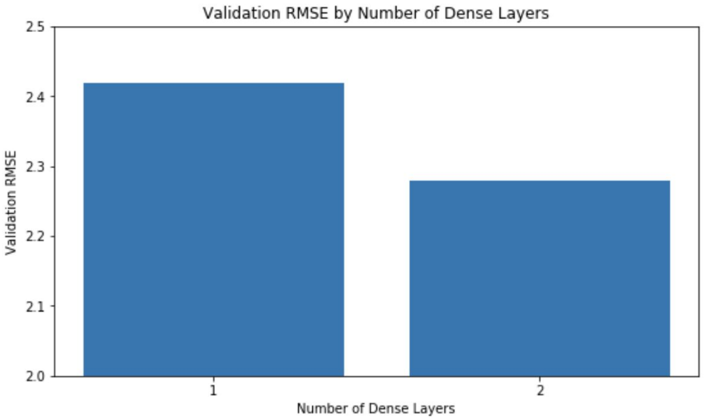
Other Enhancements

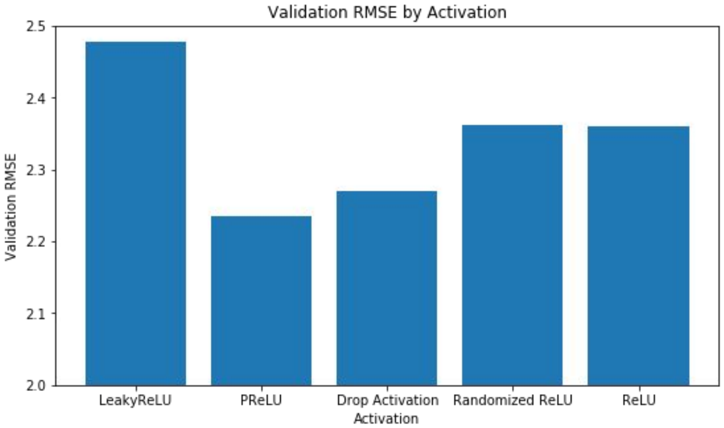
OpenCV

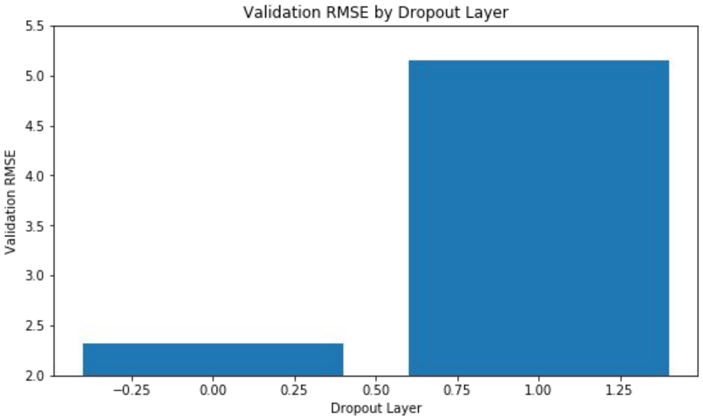


Appendix

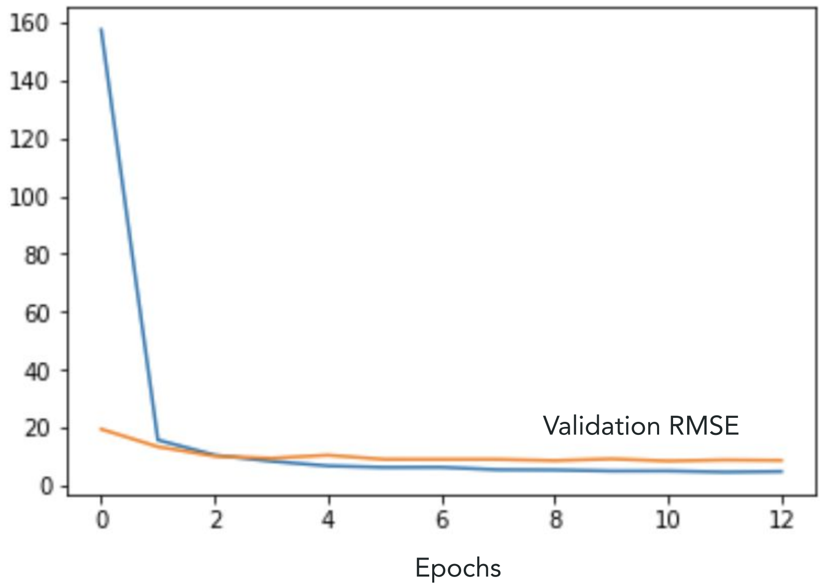

Validation RMSE by Number of Convolutional Layers
Validation RMSE
2.0
2.1
2.2
2.3
2.4
2.5
2.6
2.7
2.8
2
3
4
5
Number of Convolutional Layers


Validation RMSE by Number of Dense Layers
Validation RMSE
2.0
2.1
2.2
2.3
2.4
2.5
1
2
Number of Dense Layers

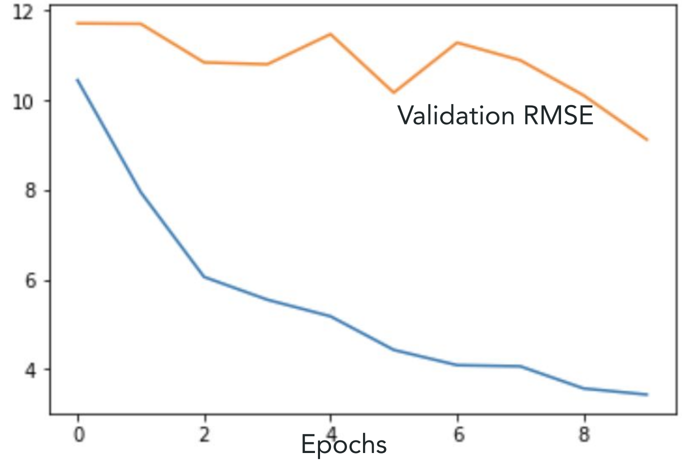

Validation RMSE by Activation
Validation RMSE
2.0
2.1
2.2
2.3
2.4
2.5
LeakyReLU
PReLU
Drop Activation
Randomized ReLU
ReLU
Activation


Validation RMSE by Dropout Layer
Validation RMSE
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
Dropout Layer

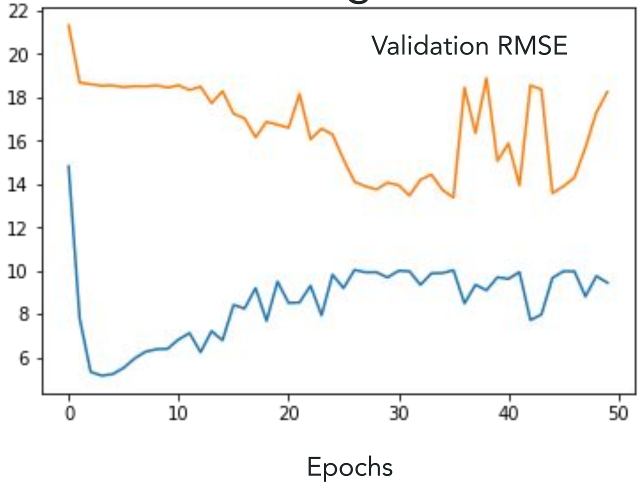
Baseline Model - 8.707



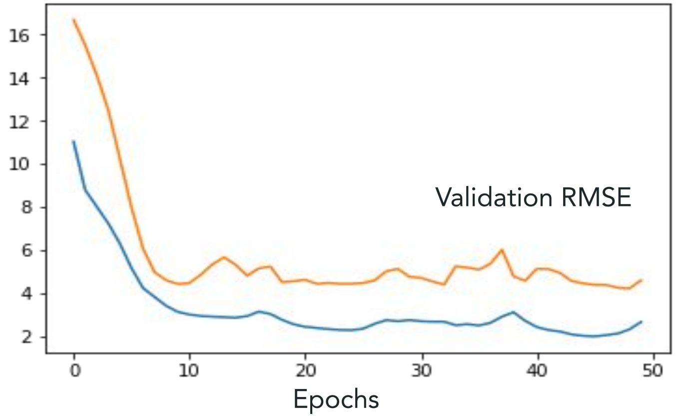
Simple CNN - 4.03



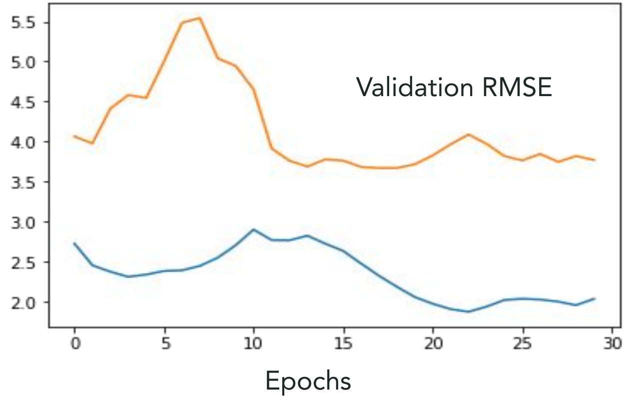
Mirroring - 5.03



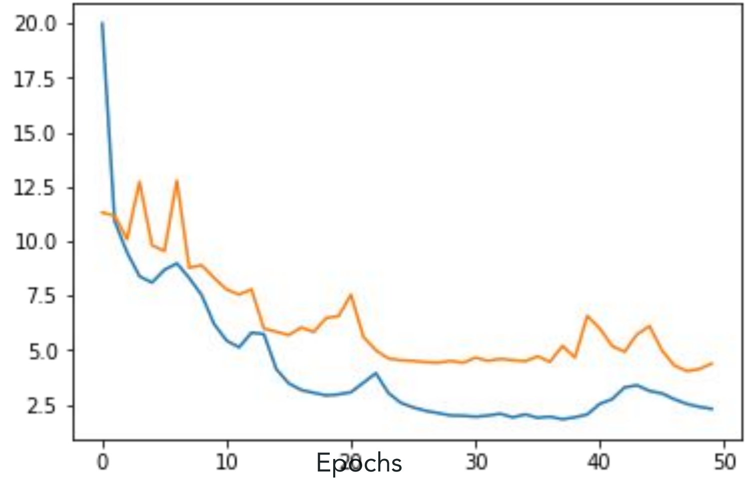
Rotation - 4.21



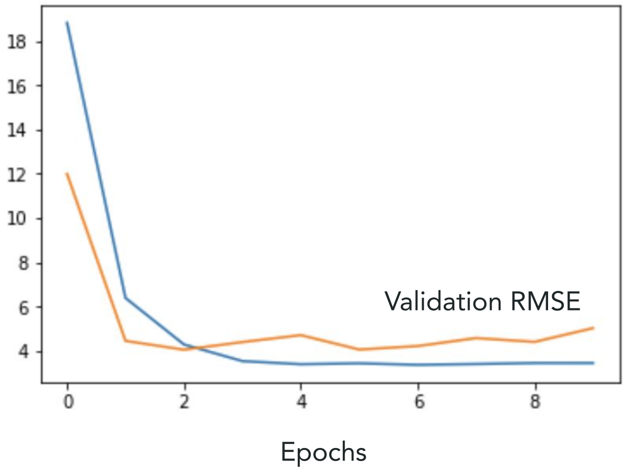
All Transformations - 3.67



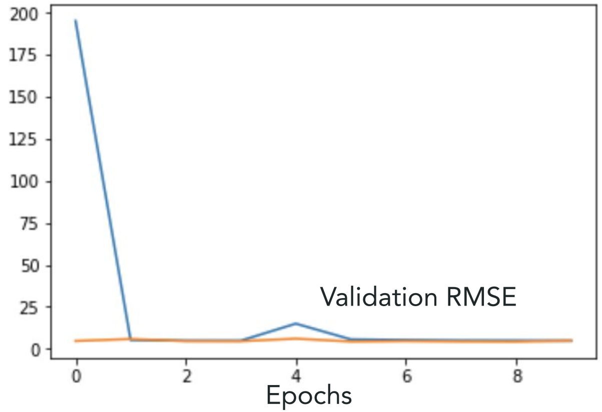
LeNet - 9.12



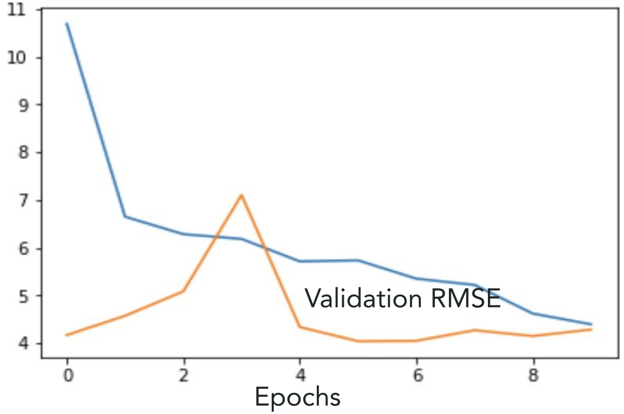
AlexNet - 5.03



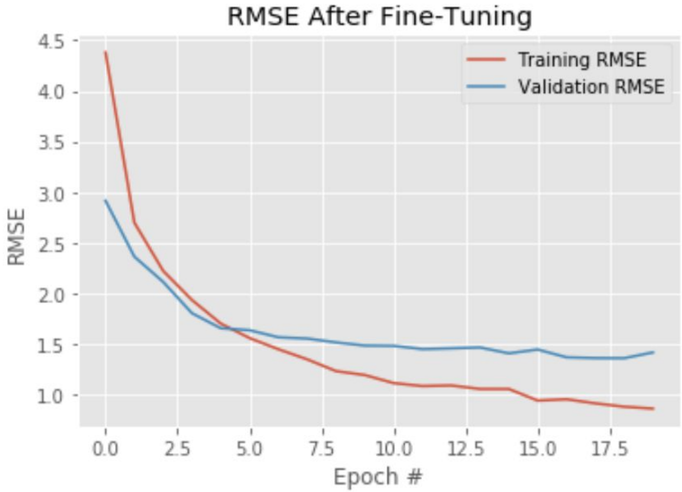
VGG 16 - 4.75



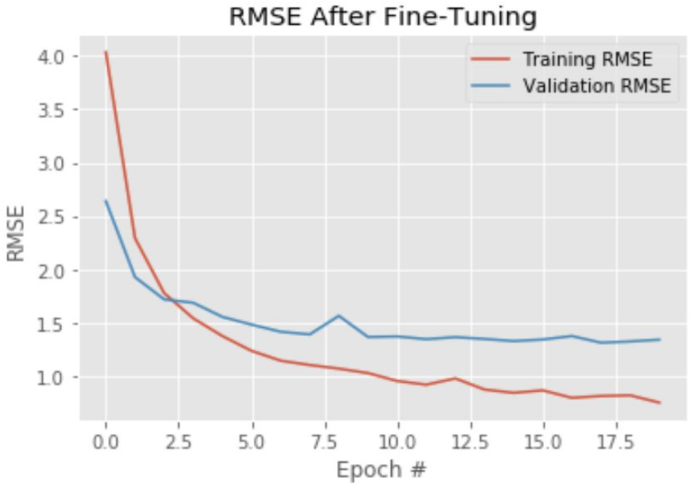
Adjusted VGG 16 - 4.27



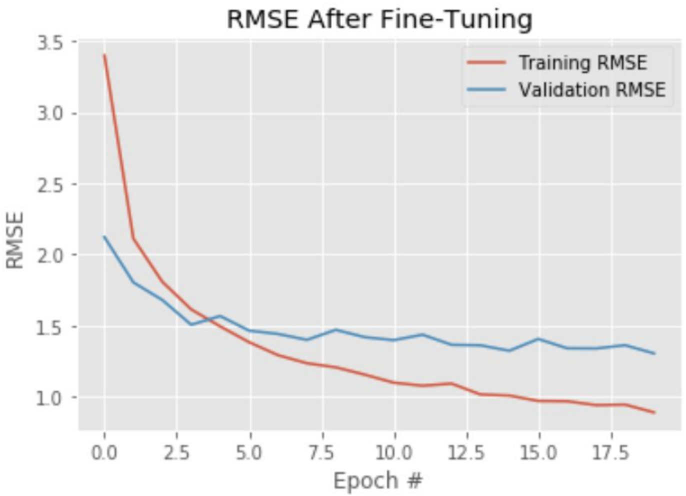
VGG 16 - 1.436



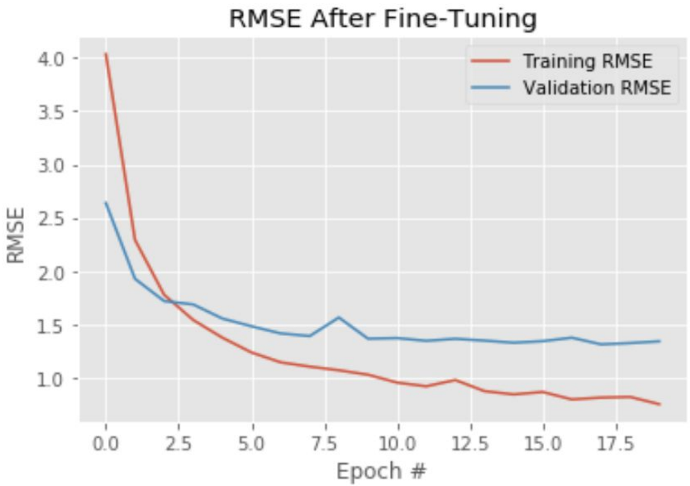
VGG 16 with Histogram Equalization - 1.36



VGG 16 with Histogram Equalization and 10 degree rotation - 1.33



VGG 16 with Histogram Equalization, 10 degree rotation and 2 models for missing data - 1.358



Xception with Histogram Equalization, 10 degree rotation and 2 models for missing data - 1.335