Facial Keypoint Detection



Final Presentation

W207 Summer 2019

T. Goter, C. Weyandt, J. Yu

Review the Problem

Competition:

- kaggle challenge to develop a model to predict location of keypoints on unseen images with lowest RMSE across all 15 keypoints.

Facial Keypoint Detection Dataset:

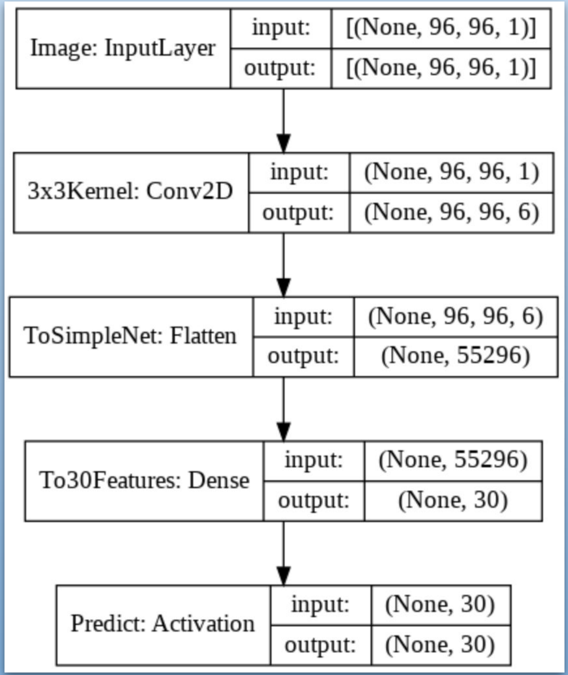
- ~7050 images in total
- 15 (x,y) coordinate pairs per image
- 2140 images with all keypoints labeled

Overview of Convolutional Neural Nets

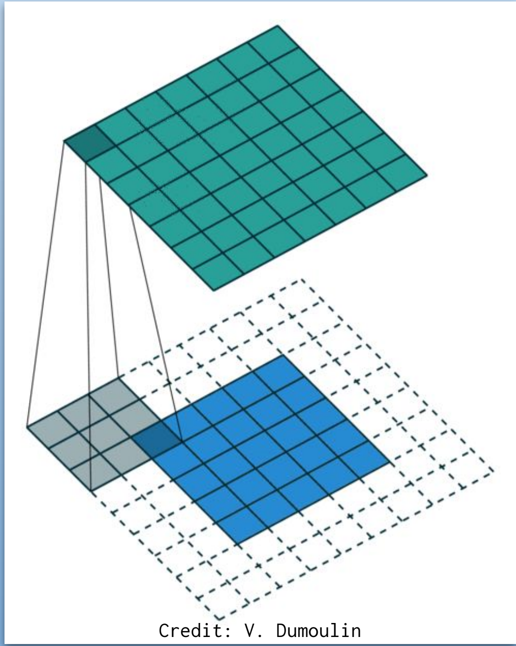
- Benefits:
    - Automate feature extraction along with model training
        - Convolutions → Application of filters to capture importance of regional features
        - Pooling → Dimensionality Reduction
    - Suppress noise and extract important features
    - Reduce required computing power
    - Multiple Layers to combine lower level important features
- Vocabulary?
- Layer types?
    -

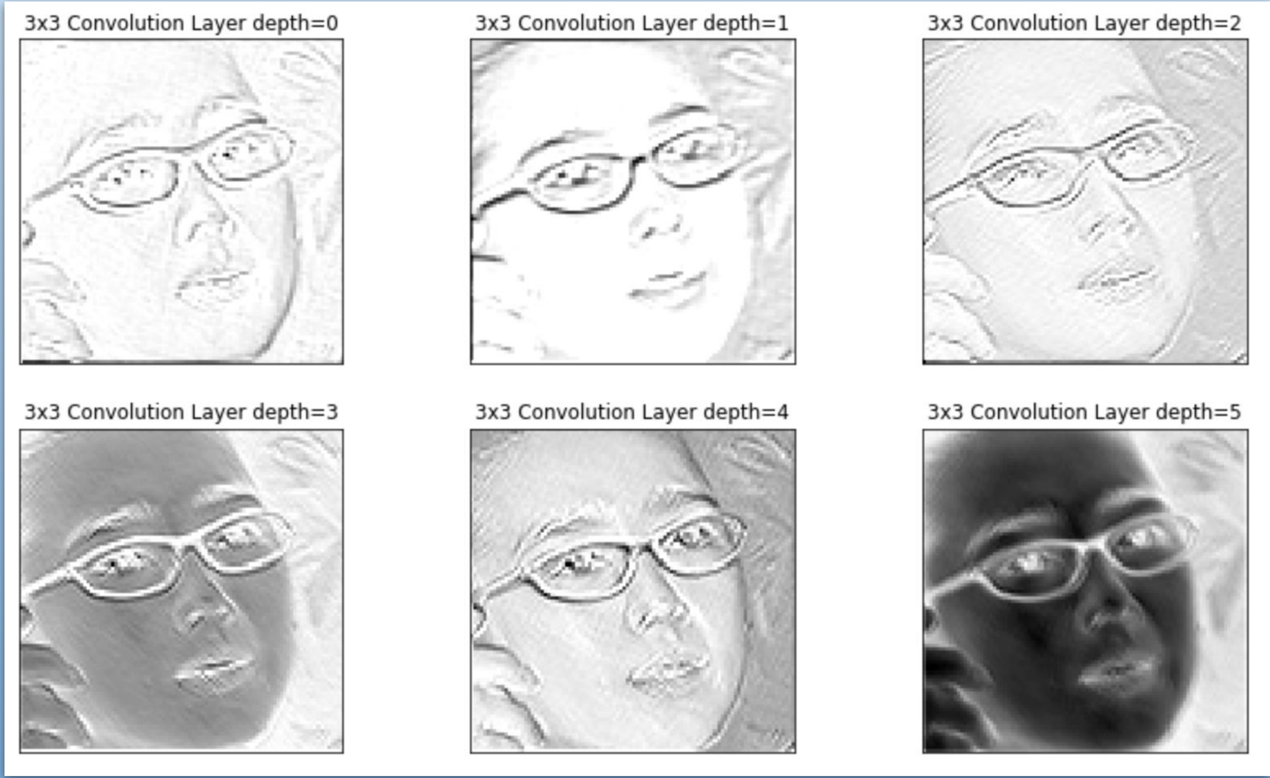
Basic CNN Model

```python
# Create a simple sequential cnn model
simple_cnn = Sequential()

# Take in a 96x96 pixel grayscale image
simple_cnn.add(InputLayer(input_shape=(96,96,1), name='Image'))

# Perform convolution with a 3x3 kernel with a depth of six
simple_cnn.add(Conv2D(6, (3,3), padding='same', activation='relu', name='3x3Kernel'))

# Convert to 1D representation (a simple neural net)
simple_cnn.add(Flatten(name='ToSimpleNet'))

# Convert to feature space representing the 30 keypoint parameters
simple_cnn.add(Dense(30, name='To30Features'))

# Predict outputs
simple_cnn.add(Activation('linear', name='Predict'))
```

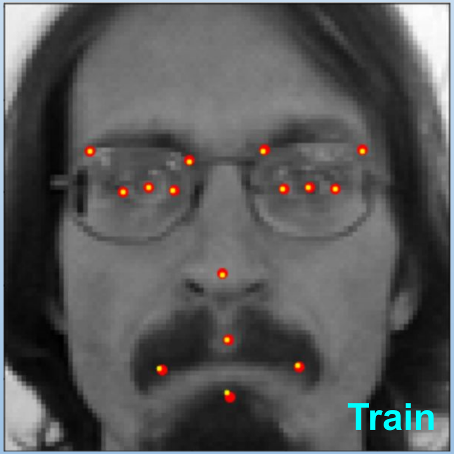


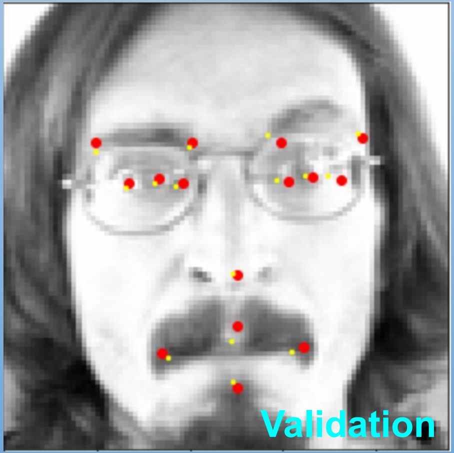
Convolution Layers





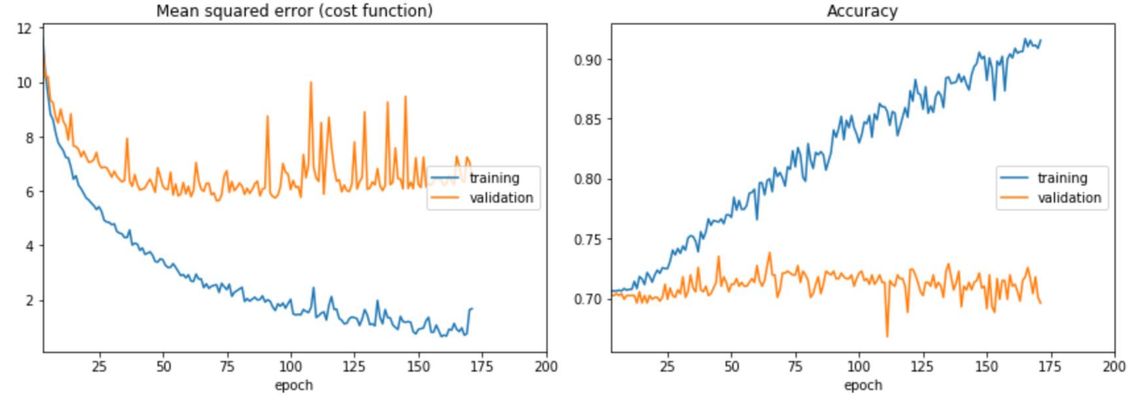
3x3 Conv Results





Prediction
Actual

```
Layer (type)                 Output Shape              Param #
=================================================================
3x3Kernel (Conv2D)           (None, 96, 96, 6)         60
_________________________________________________________________
To1Dimensional (Flatten)     (None, 55296)             0
_________________________________________________________________
To30Features (Dense)         (None, 30)                1658910
_________________________________________________________________
Predict (Activation)         (None, 30)                0
=================================================================
Total params: 1,658,970
Trainable params: 1,658,970
Non-trainable params: 0
_________________________________________________________________
```

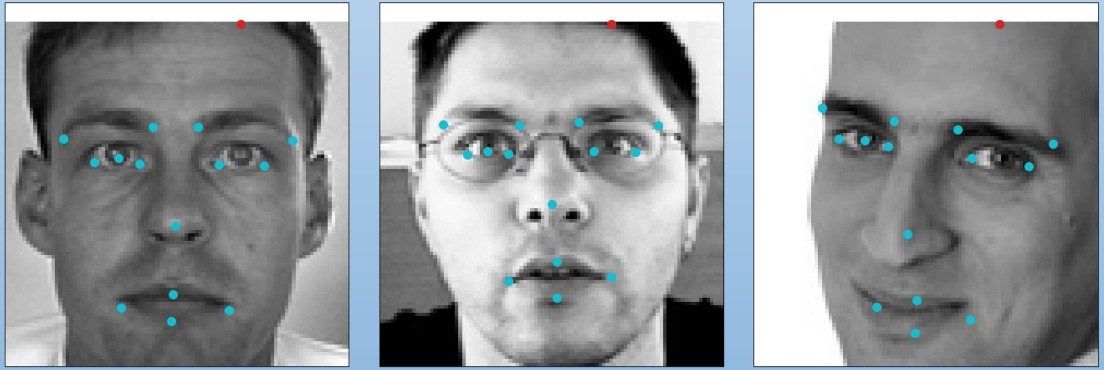



```
Mean squared error (cost function):
training   (min:    0.646, max:  769.241, cur:    1.672)
validation (min:    5.642, max:   93.179, cur:    6.579)

Accuracy:
training   (min:    0.678, max:    0.917, cur:    0.915)
validation (min:    0.668, max:    0.738, cur:    0.696)
```

Overfit!

Basic Neural Nets

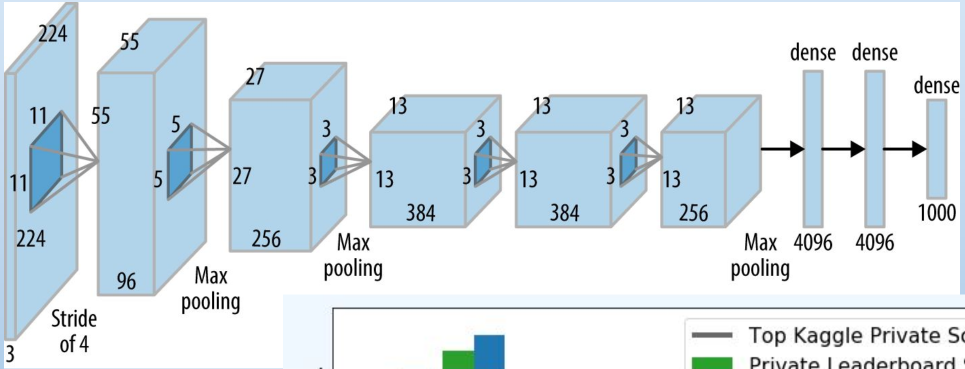
- User familiarity with new tools
  - TensorFlow with Keras
  - Colab Environment
- Sensitivities
  - Gradient Descent Optimizers
  - Activation Functions
  - Hidden Units
- Did not optimize



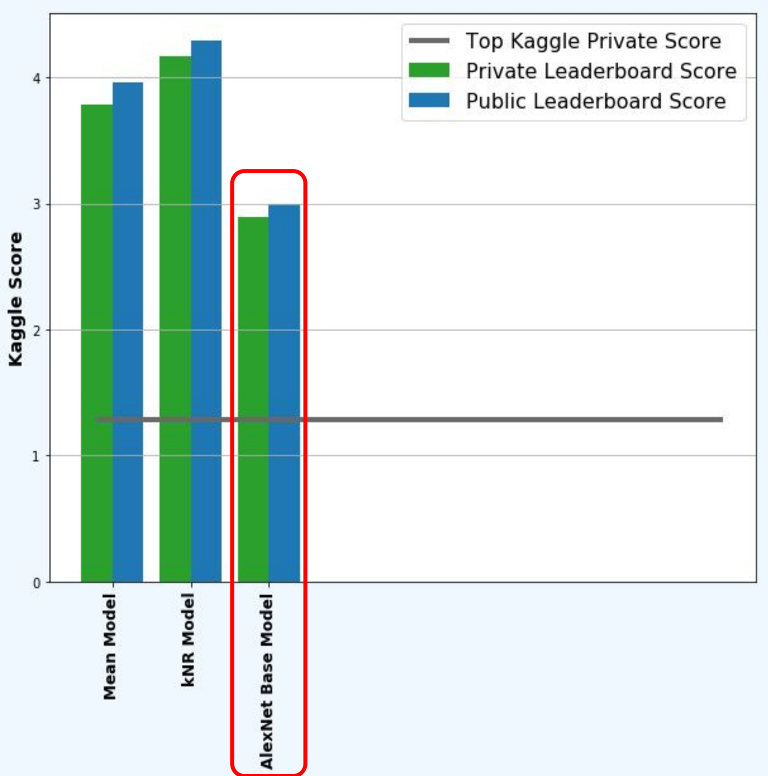
Valuable Lesson Learned:

For regression analysis, don’t use a sigmoid or RELU activation on output layer!

AlexNet* Studies



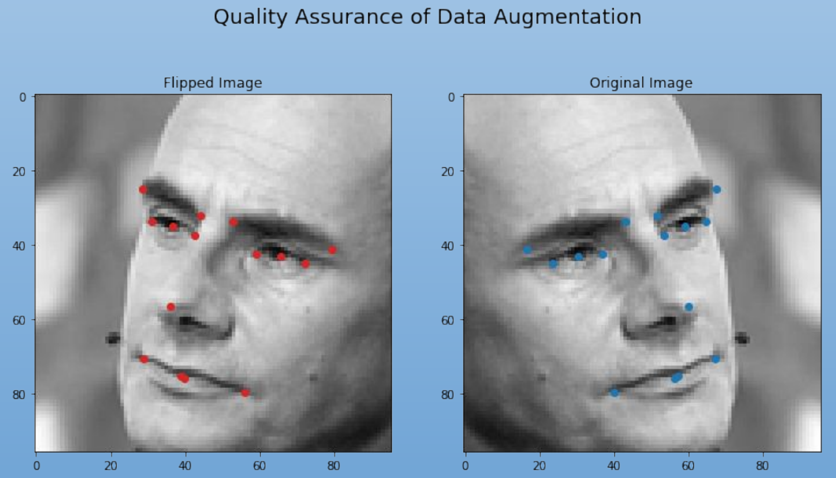
- Taken as inspiration
  - Simple to understand and implement
  - Alternate convolutions with pooling (feature reduction)
- Key additional features
  - Batch normalization for increased learning rate
  - **Data augmentation** for increase training set size
- Sensitivity Studies for Optimization
  - Starting Filter Depth
  - Gradient Descent Optimizer
  - Dropout Rate Strategy

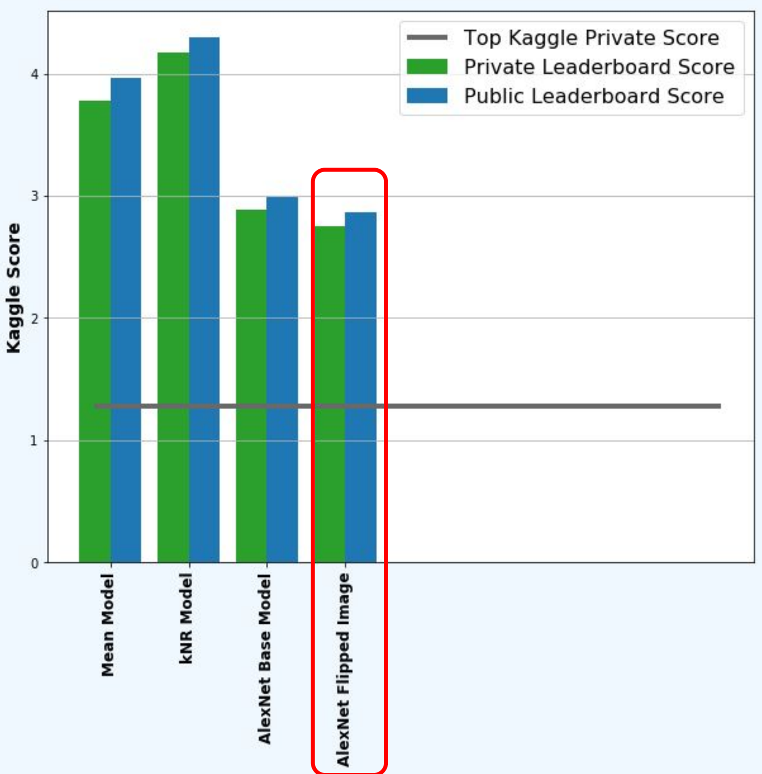


*Alex Net Paper

Data Augmentation

- Lack of complete training examples
  - Lots of images without all keypoints identified
  - ~2140 complete training examples (of >7000 images)
  - Increase generalization through data set enhancement
  - Modify training images to represent “new” images to the model





Results Visualization with Bokeh

FACIAL KEYPOINT DETECTION RESULTS EXPLORATION

*(Kaggle Website **Kaggle**)*

*(GitHub Website **Project Repo**)*

Net Architecture

alex

Gradient Descent Optimizer

adam

Maximum Validation RMSE: **30**

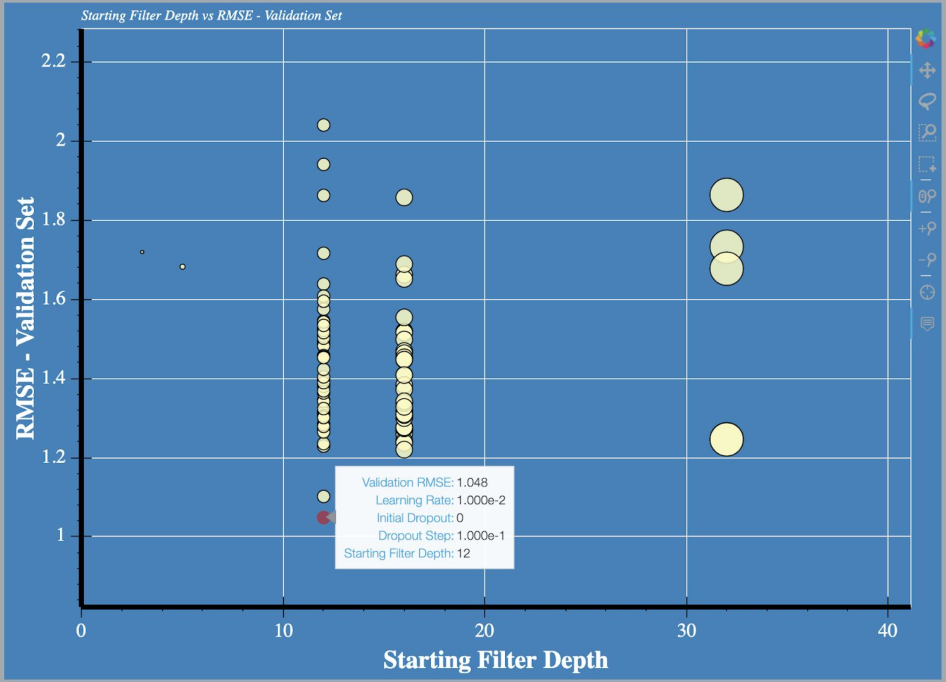
X Axis

Starting Filter Depth

Y Axis

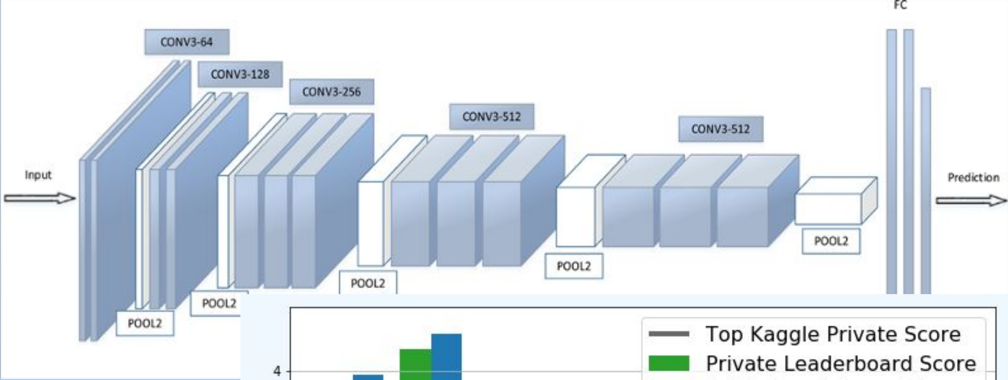
RMSE - Validation Set

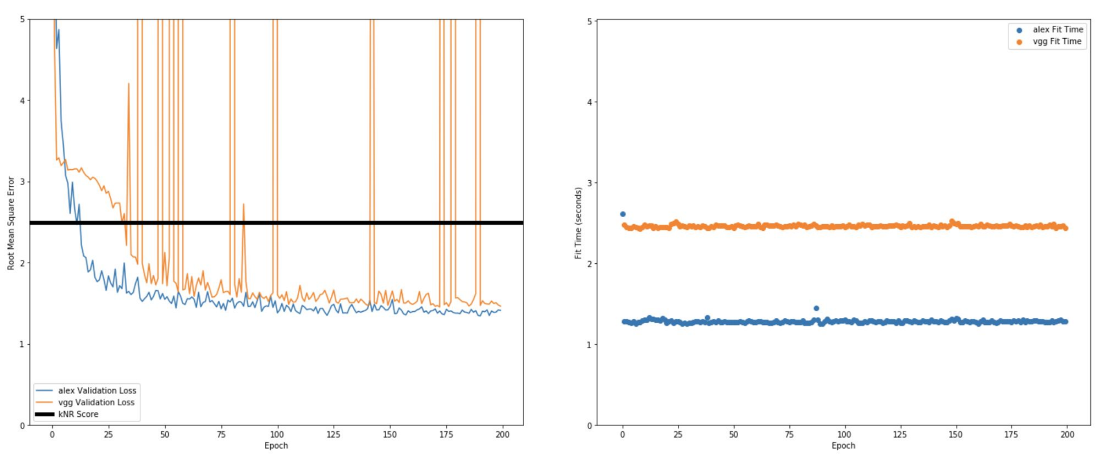
#	Validation RMSE	Learning Rate	Starting Filter Depth	Initial Dropout	Dropout Step
0	1.72	0.001	3	0.00	0.00
1	1.68	0.001	5	0.00	0.00
2	1.59	0.001	12	0.00	0.00
3	1.55	0.002	12	0.00	0.00
4	1.49	0.005	12	0.00	0.00
5	1.45	0.010	12	0.00	0.00
6	1.34	0.010	12	0.00	0.00
7	1.34	0.010	12	0.00	0.00
8	1.32	0.010	12	0.00	0.00
9	1.34	0.015	12	0.00	0.00

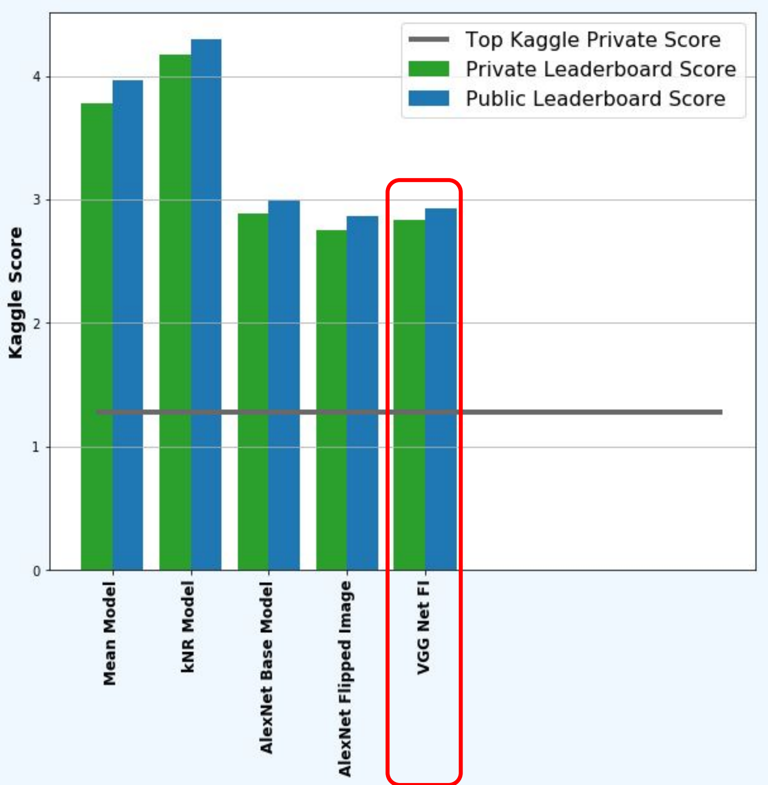


VGGNet* Studies

- Very similar to AlexNet
- Multiple Convolution layers prior to pooling → More parameters to train
- Similar accuracy - double training time







*VGG Net Paper

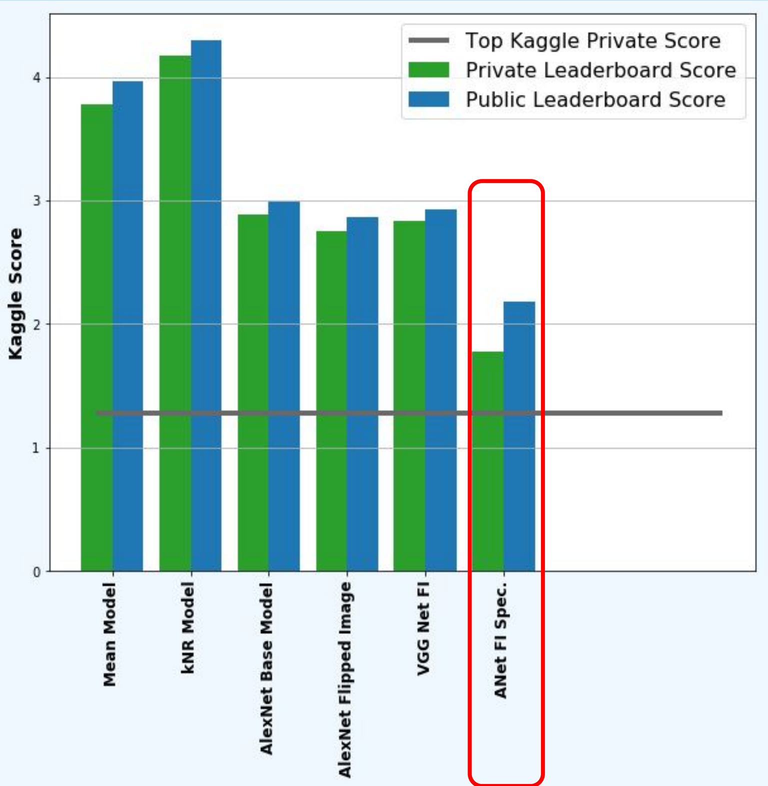
Specialist Models

- Initial models used to predict all x, y coordinates for all 15 keypoints
- Alternative: *keypoint specific models*
  - Pros
    - Increase relevance of the network
    - Use all training data
  - Cons
    - 15 Models to train!
    - 47.5 MILLION PARAMETERS
- First determine worth:
  - Train 15 models with same architecture as base AlexNet (with flipped training data)
  - Transfer learned weights from base model
  - Early stopping to reduce runtime

Specialist Models

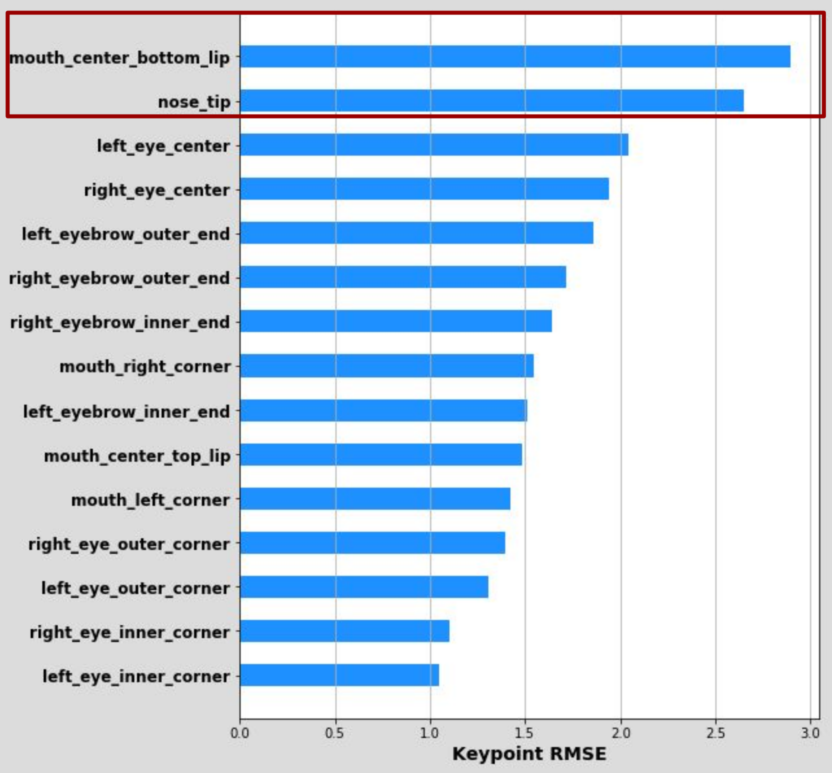
- Initial models used to predict all x, y coordinates for all 15 keypoint
- Alternative: *keypoint spe* s
  - Pros
    - Increase rel
    - Use all
  - Cons
    - ARAMETERS
- e worth:
    - o models with same architecture as
    - se AlexNet (with flipped training data)
    - Transfer learned weights from base model
  - Early stopping to reduce runtime

Totally Worth It!!!

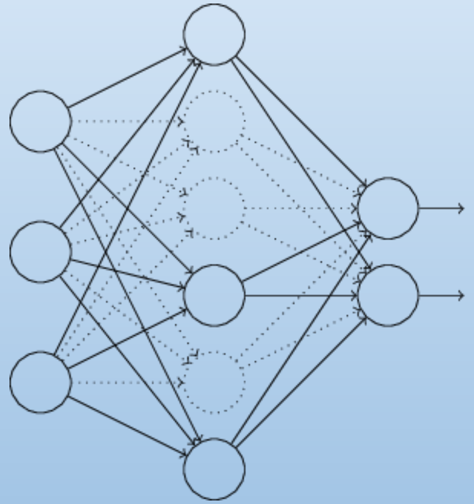


Improved Specialist Models

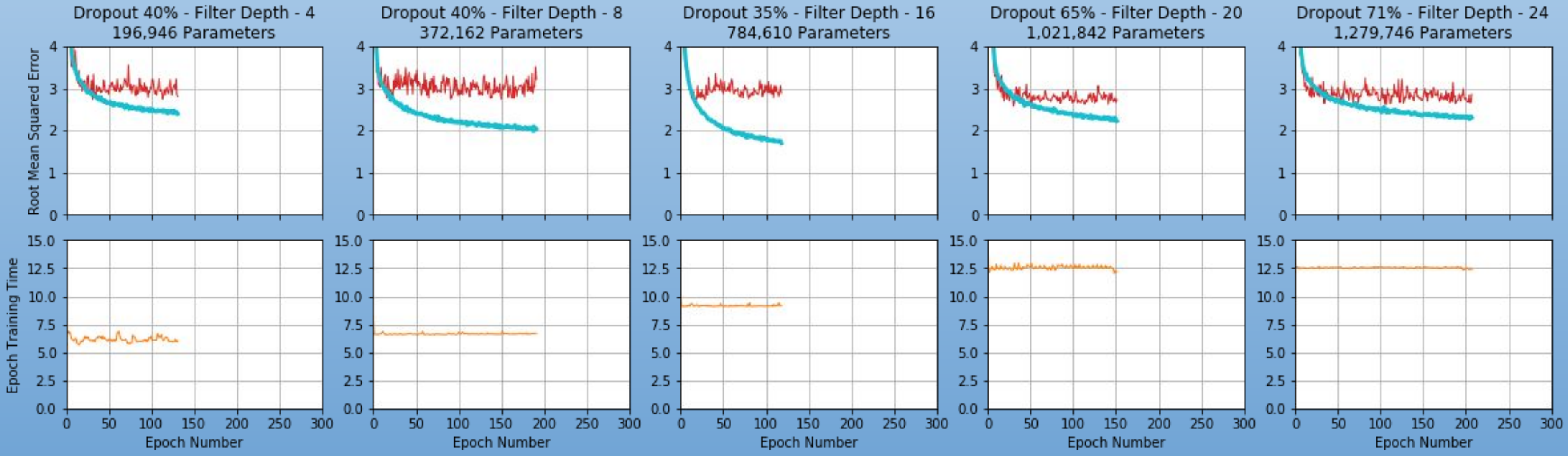
- No optimization of the 15 original specialists
- Focused optimization on two worst performers
- Bad news: both > 7000 training images



Dropout Rate Optimization



- Applied after every conv/pool layer at an increasing rate
- Use instead of regularization for reducing overfitting and improving model generalization



Final Specialist Models - Some Improvements

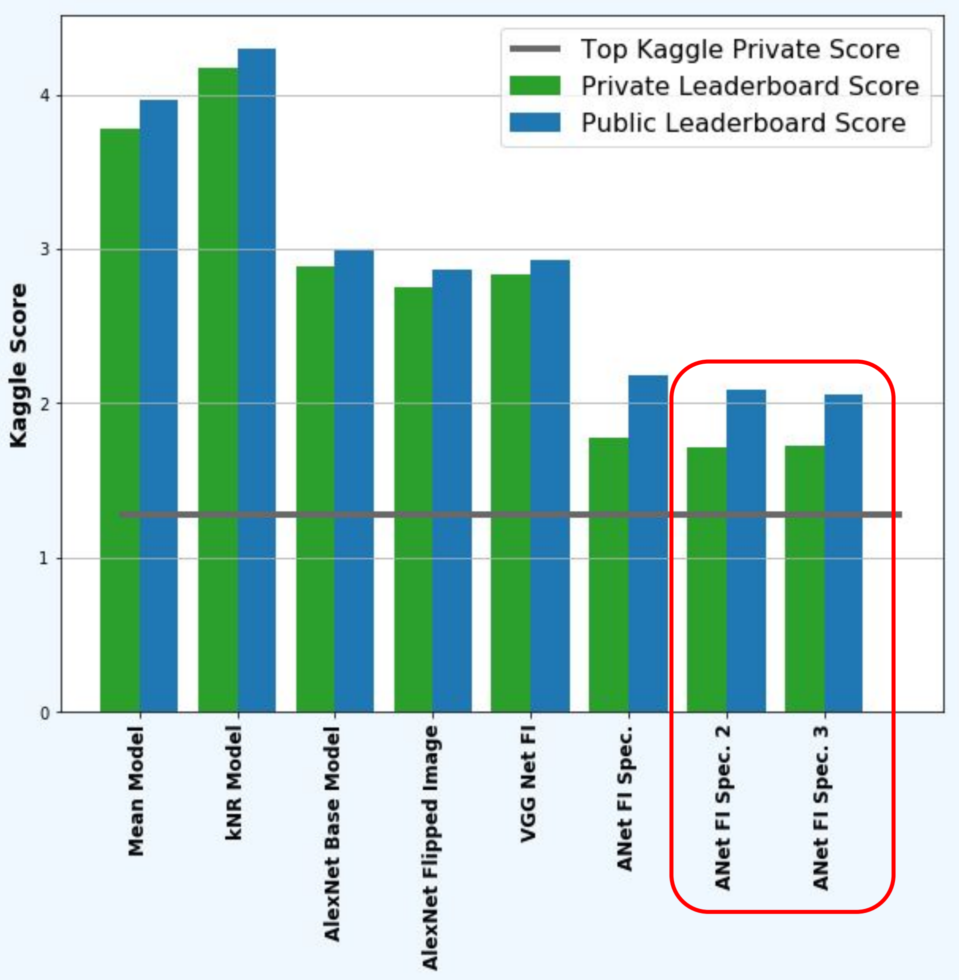
Nose Tips

Final Specialist Models - Still Not Perfect

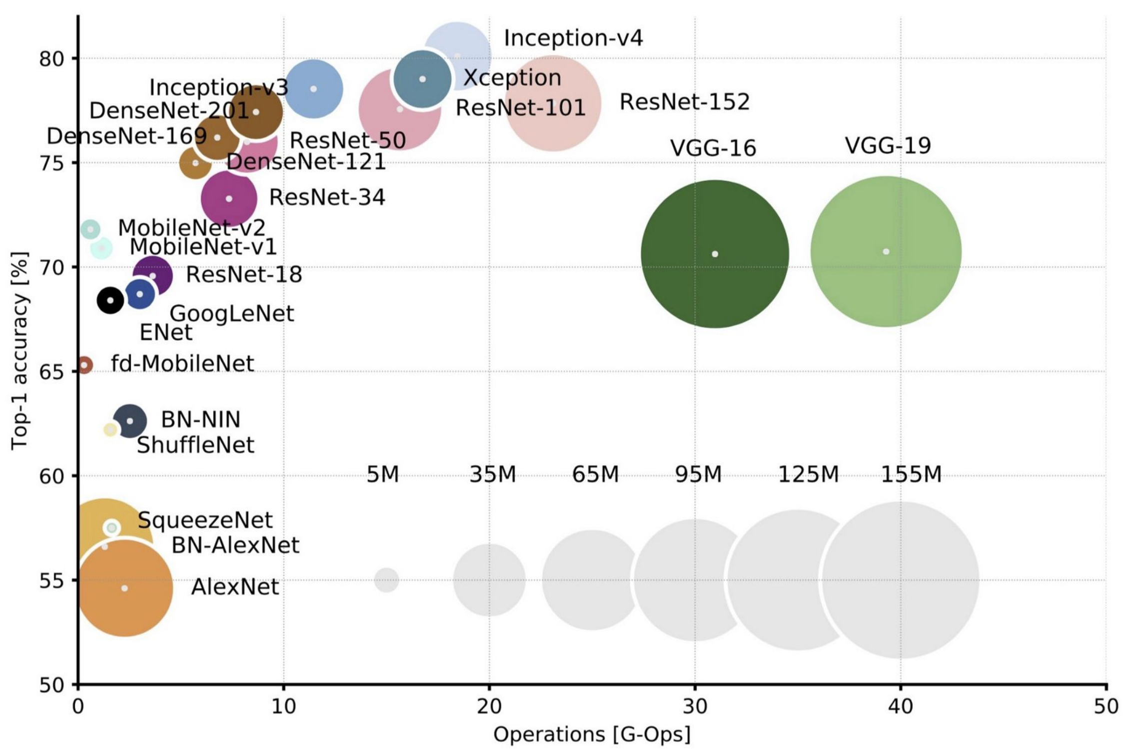
Nose Tips

Final Specialist Models

- Modified architectures and dropout strategy for nose and bottom lip
  - Small improvements in overall RMSE on test data
- Adjustments to eye models not entirely successful
- 9th place overall on Kaggle Leaderboard (frozen)
- Still room for improvement
  - Very blurry images
  - Images with faces rotated



Architecture Comparison

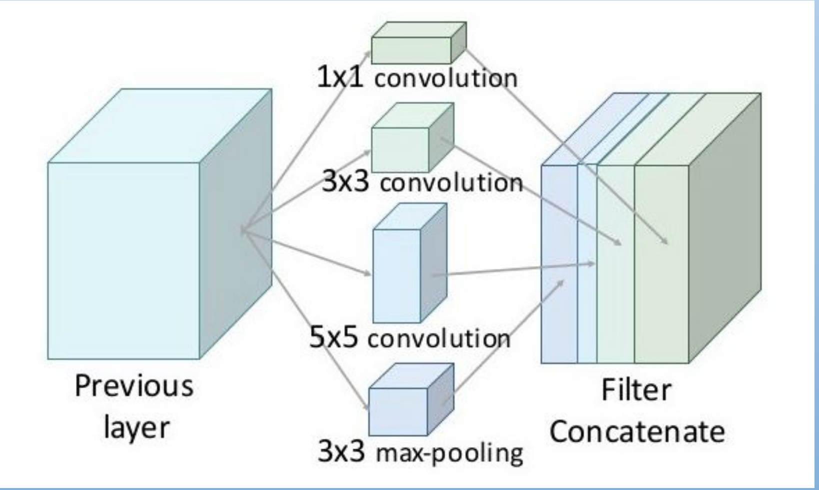


Credit: E. Culurciello

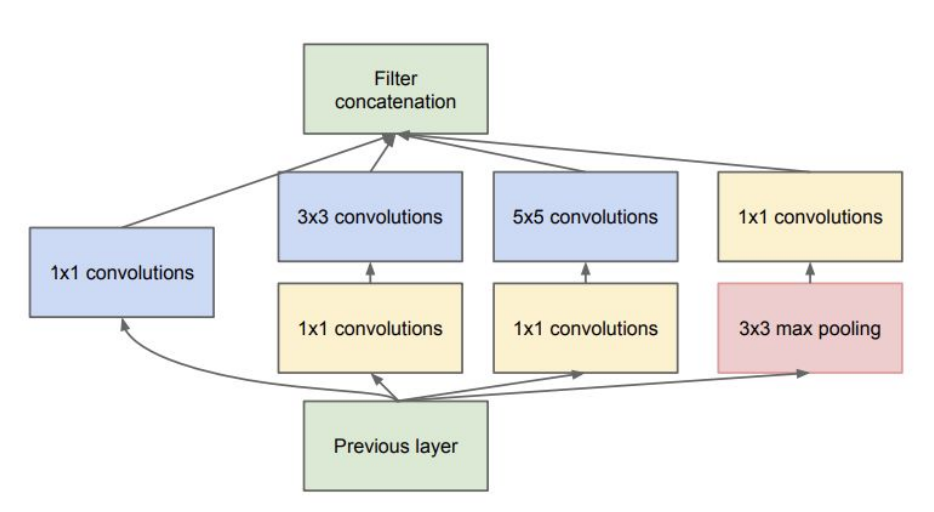
- Top-1 Accuracy vs. Operations
- Size of the blobs indicates parameter count
- Inception and ResNets in general have high accuracy
- MobileNet is highly efficient
- VGG requires high operation costs and parameter count

GoogleNet - Inception Module

Why not let the model choose what transformation provides the most useful information?

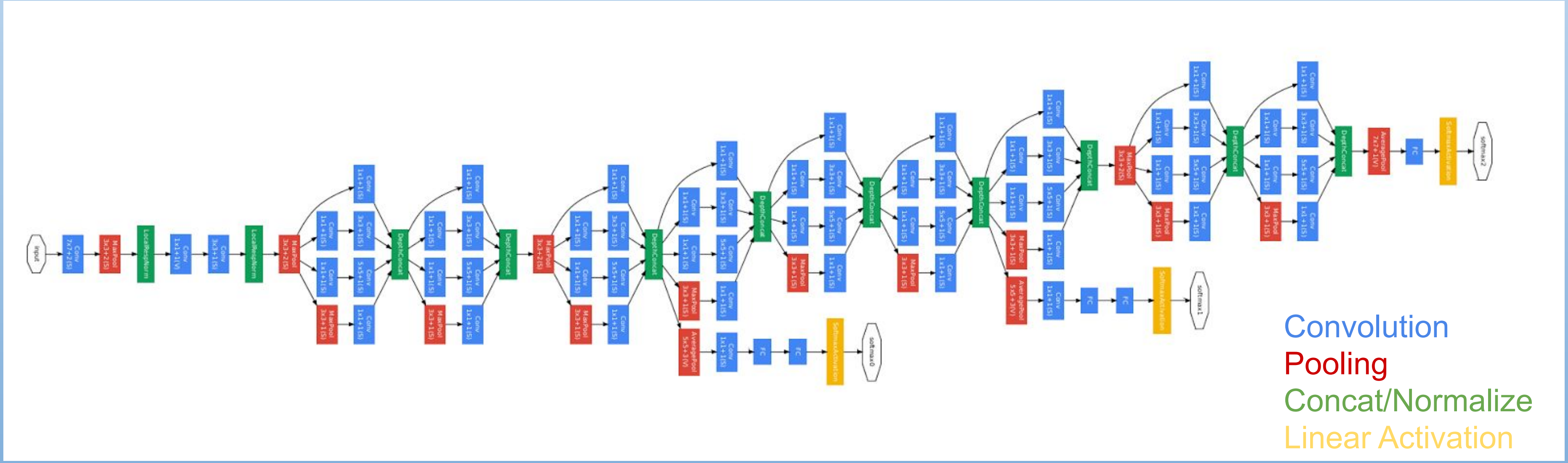


Credit: J. Xu

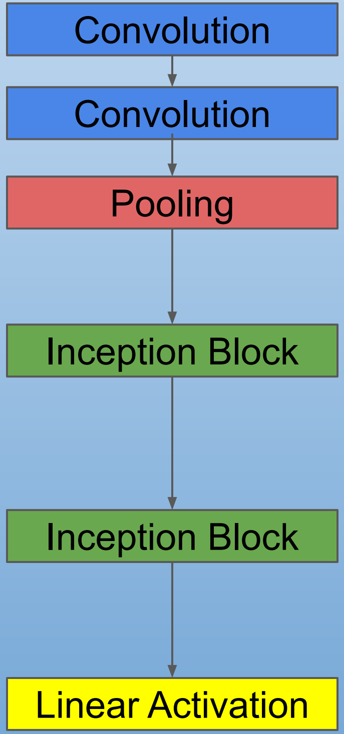


Credit: J. Brownlee

GoogleNet: 22-Layer Inception Network v1



Our Best Inception Model



ResNet - Residual Module

Problems:

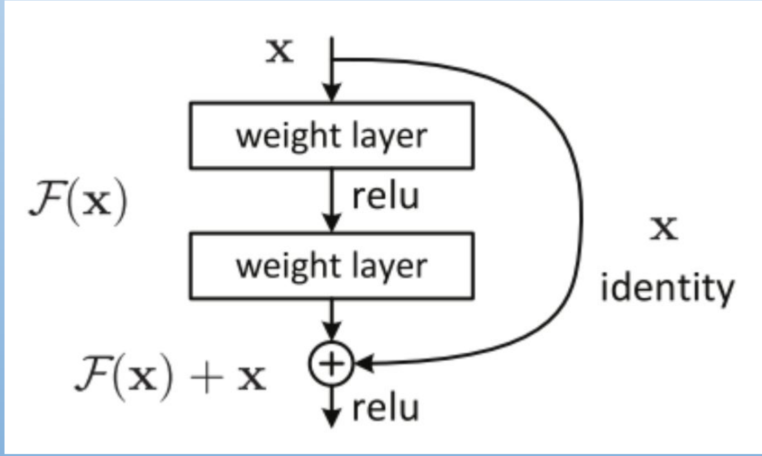
- Deeper nets perform worse as layers increase.
- Direct mappings are hard to learn.

How about learning the <u>difference</u>?

Example:

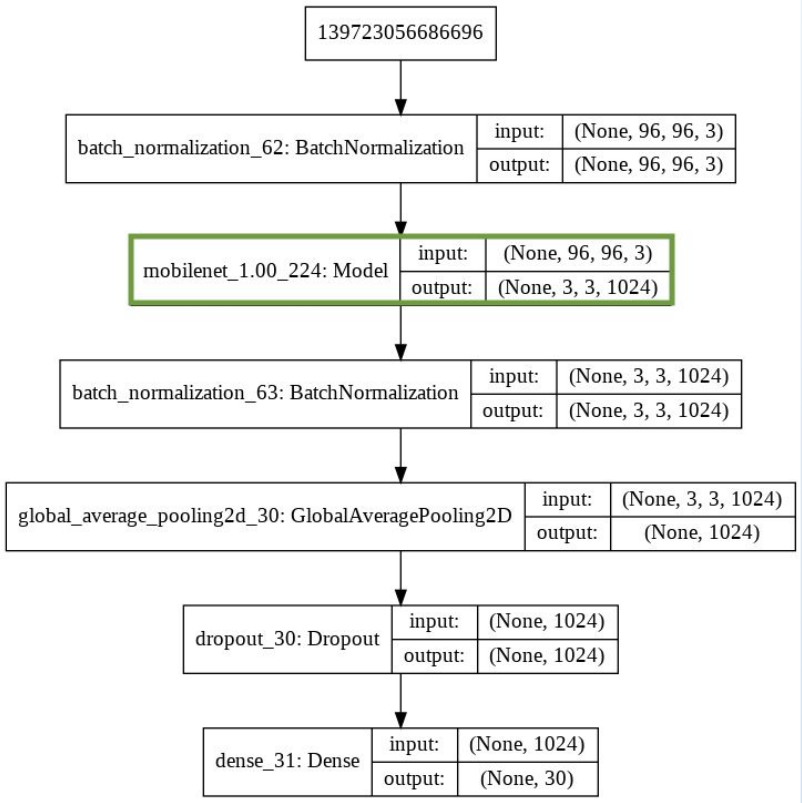
If residual F(x) = H(x) - x

Instead of trying to learn an underlying mapping from x to H(x), the model will try to learn F(x)+x.



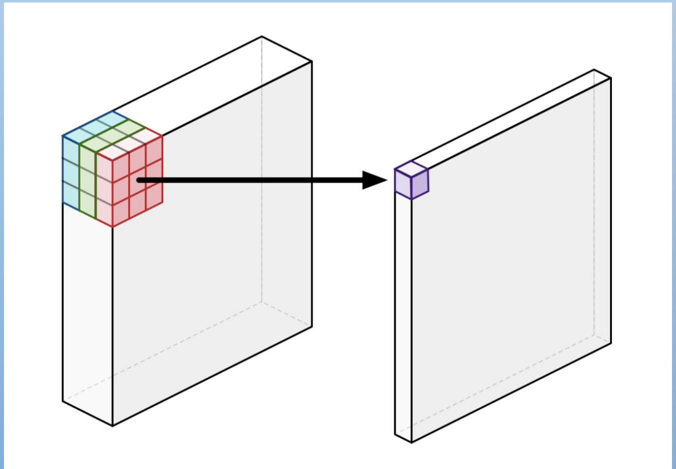
MobileNet

- Designed for on-device or embedded applications.
- Small, low-latency, low-power models.
- Can be built upon for classification, detection, embeddings, and segmentation.



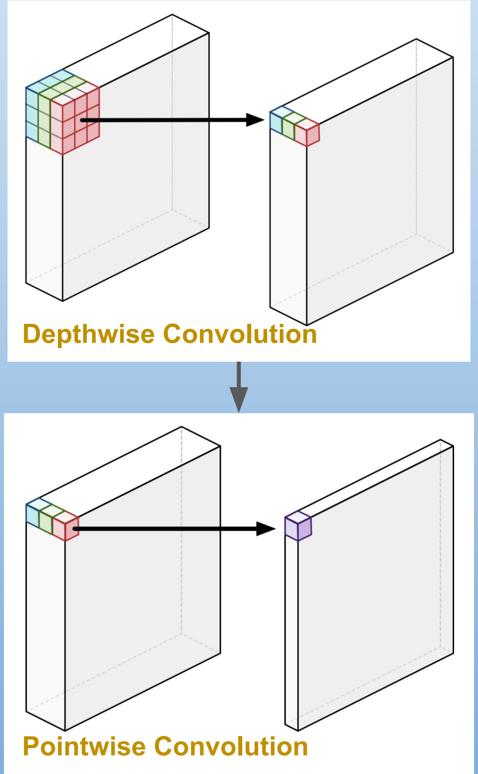
MobileNet

The introduction of ***Depthwise Separable Convolution*** *means a lot less computation.*



Regular Convolution Layer

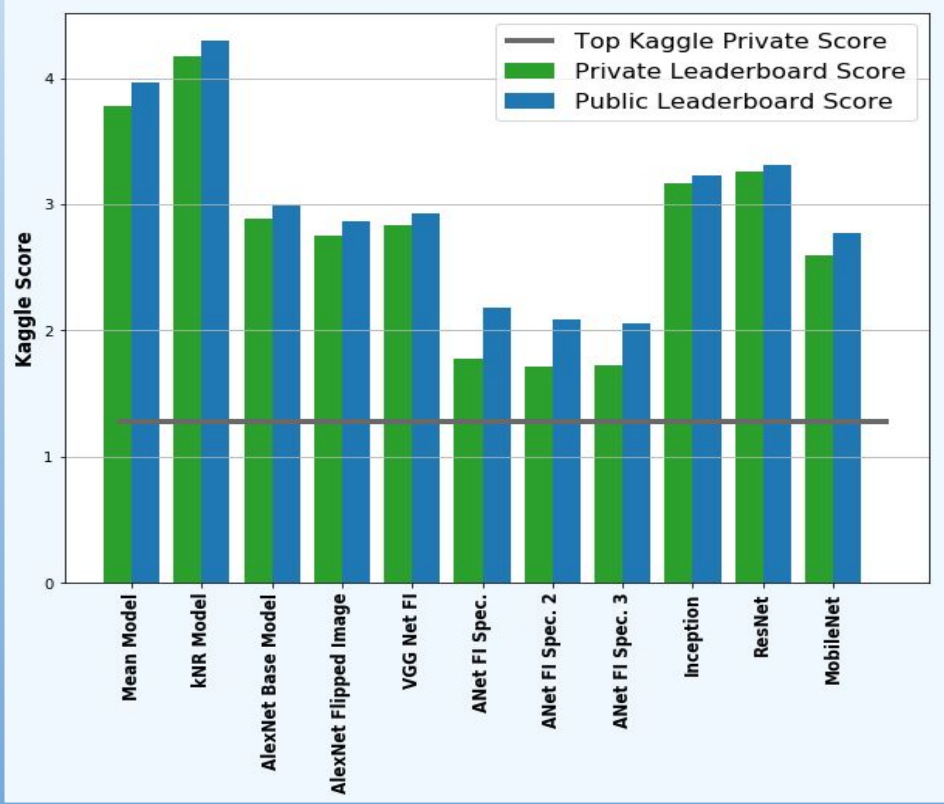
VS



MobileNet Depthwise Separable Convolution

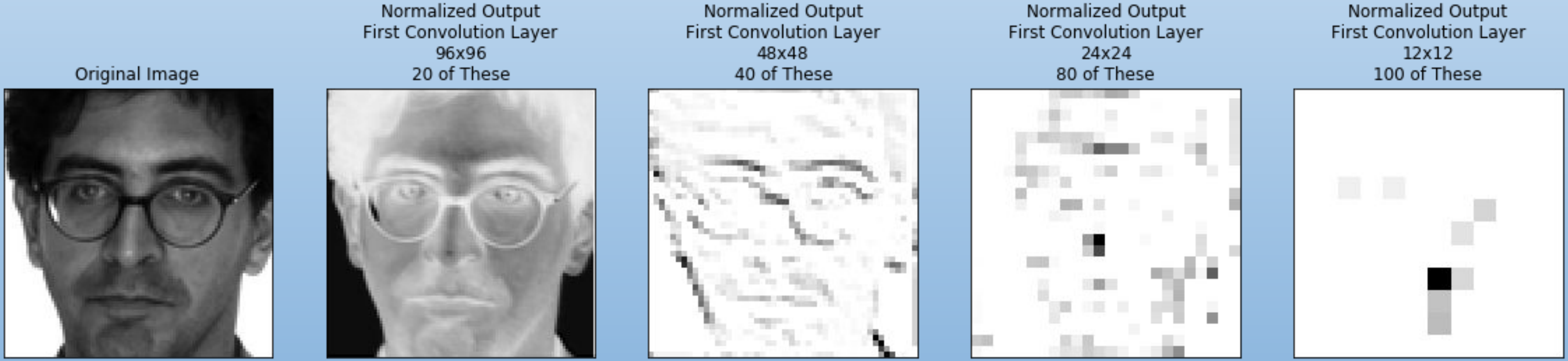
Summary

- Lots of opportunity for further study
  - Continued improvement of specialists
  - Architecture investigations - focused on MobileNet
  - Ensemble models
- Apply model to external test data



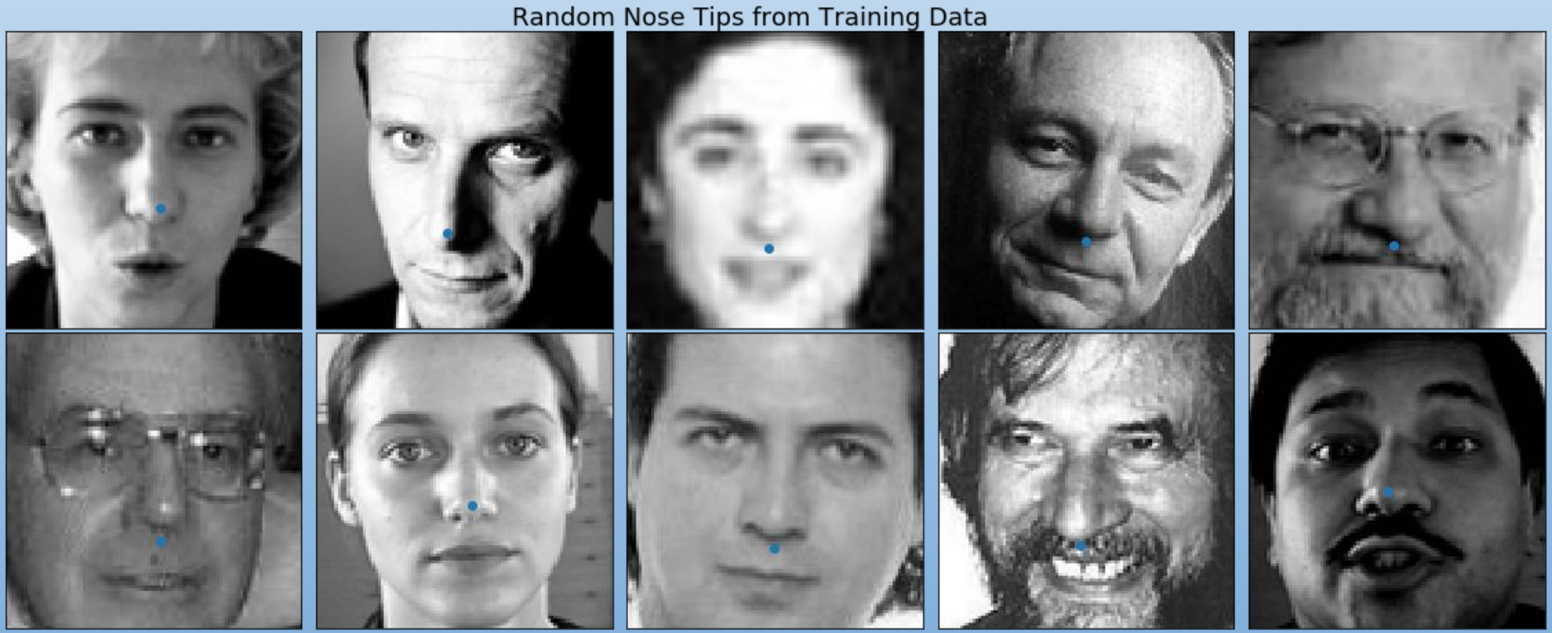
Back-up Slides

Convolutions → What is going on?



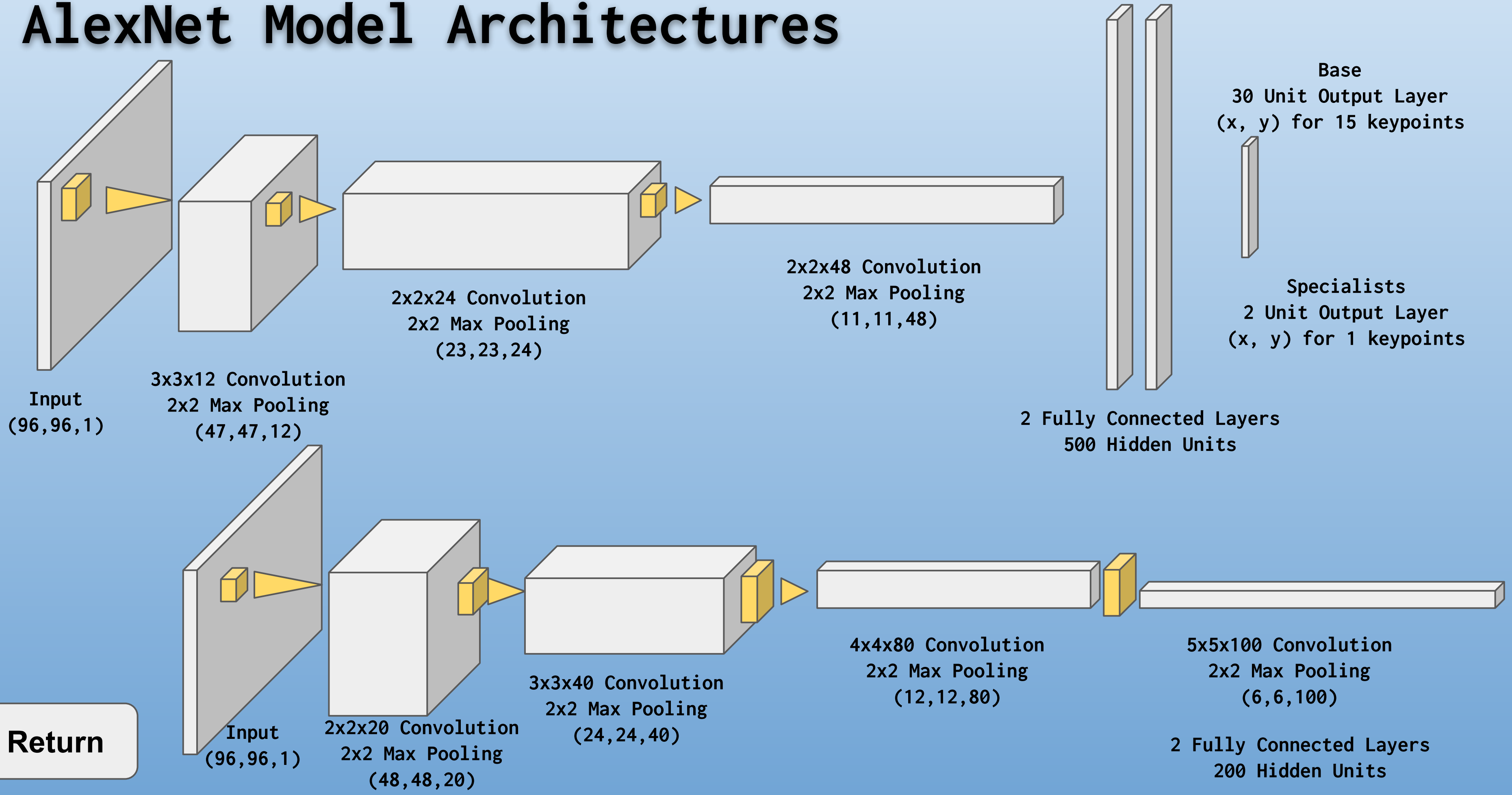
Return

Nosetip Label Locations - Inconsistent



Return

AlexNet Model Architectures



Return