Convolutional Neural Network Architectures

Hello!

I am Wesley Osborne

Head
Research
Data
Scientist

1.

Problem Statement

What are we researching?

Problem Statement

Osborne Research Labs is a recently started research lab. Our lab is conducting research on computer vision systems for drones and robotics. Given we are just entering the field and we have a basic understanding of convolutional neural networks (CNNs), we wanted to research popular complex CNN architectures. The goal of this project is to learn about the pros and cons of three popular CNN architectures and see how they perform on an indoor scene dataset. We're hoping for accuracy scores above the baseline of 9%.

2.

About the Data

Data Collection

- Used MIT Indoor Scene Dataset
  - 15,000 images across 67 classes
- Used 11 Classes
  - Bathroom, Bedroom, Pantry, Corridor, Staircase, Gym, Living Room, Kitchen, Office, Closet, and elevator
- Scraped ~500 new images for each class from google
- Total dataset of approx. 7,000 images

3.

Basic Building Blocks of CNNs

CNN Basics

- Convolutional Layers (Conv)
- Pooling Layer (POOL)
- Fully Connected Layer (FC)

a.k.a Dense Layer

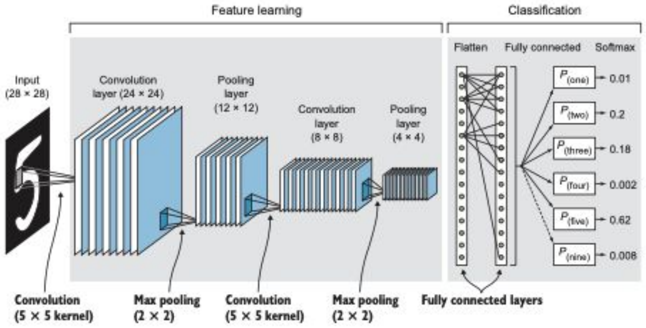


Figure 3.12 The basic components of convolutional networks are convolutional layers and pooling layers to perform feature extraction, and fully connected layers for classification.

4.

CNN Architectures

VGGNet

- Developed in 2014 by The Visual Geometry Group at Oxford University (Keren Simonyan and Andrew Zisserman)
- Most popular configuration is VGG16
- Popular because it’s simple to understand and has a uniform architecture
- Inspired by AlexNet and LeNet
- Top-5 error rate of 8.1% on ImageNet

VGGNet Architecture

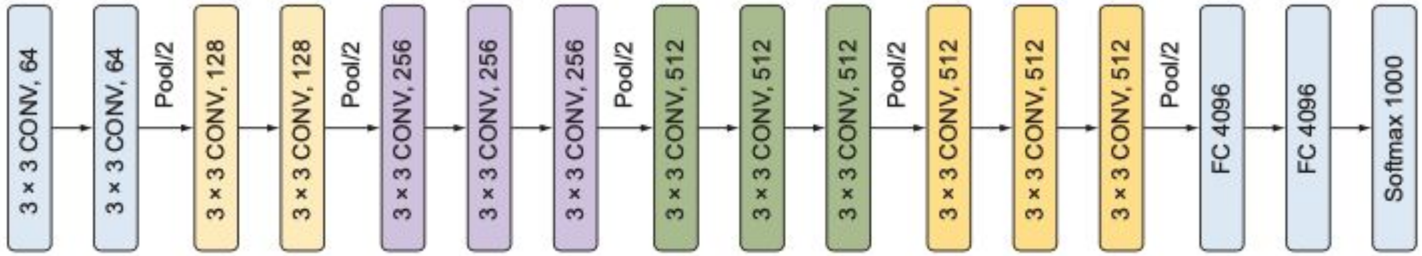
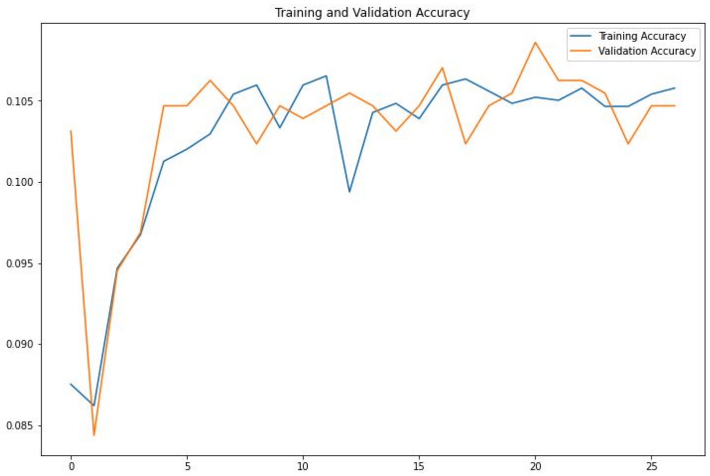
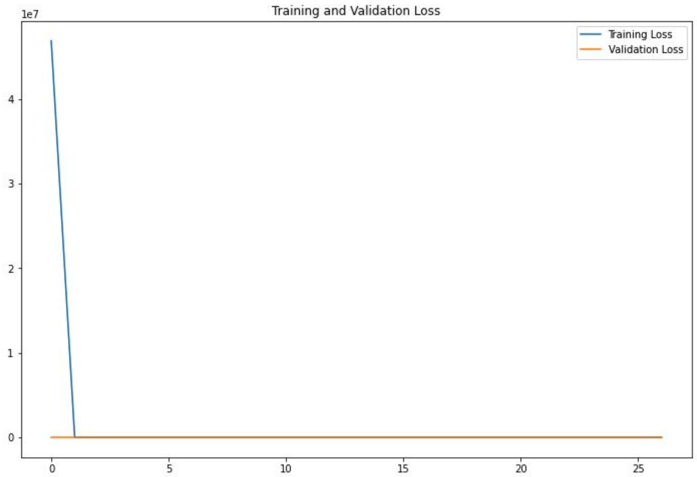


Figure 5.8 **VGGNet-16 architecture**

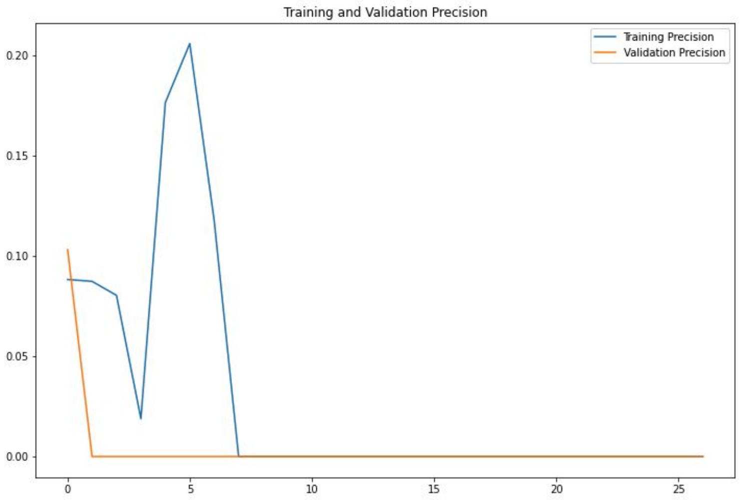
VGG16 Performance on Indoor Scenes

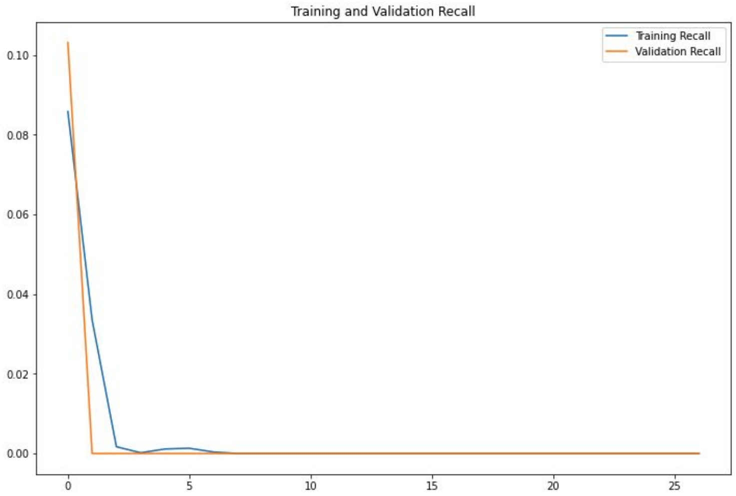
- Top accuracy of around 10%



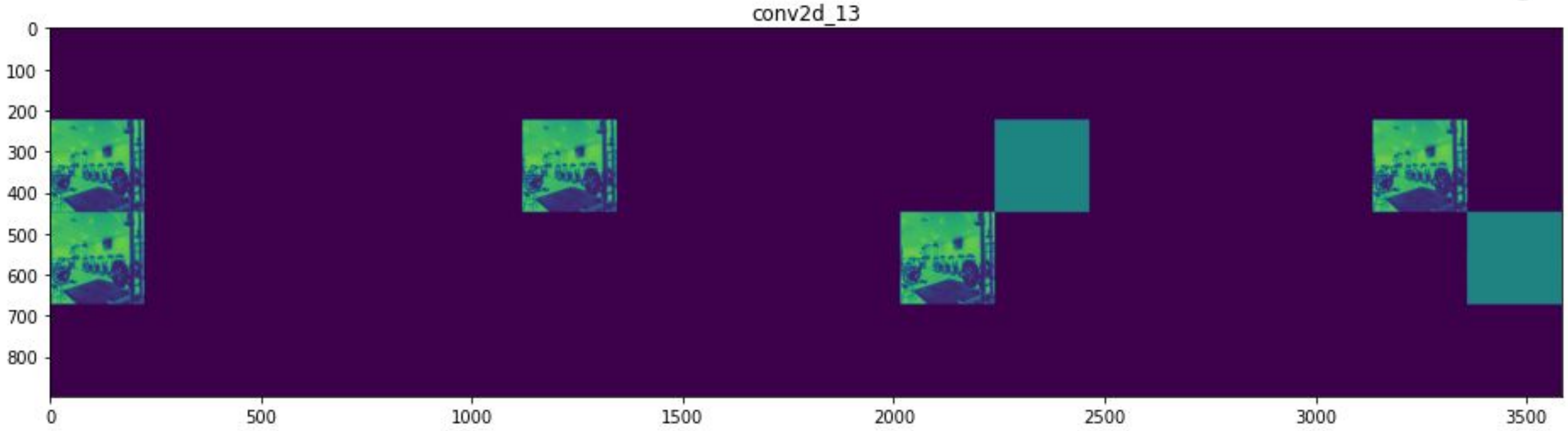


VGG16 Performance on Indoor Scenes

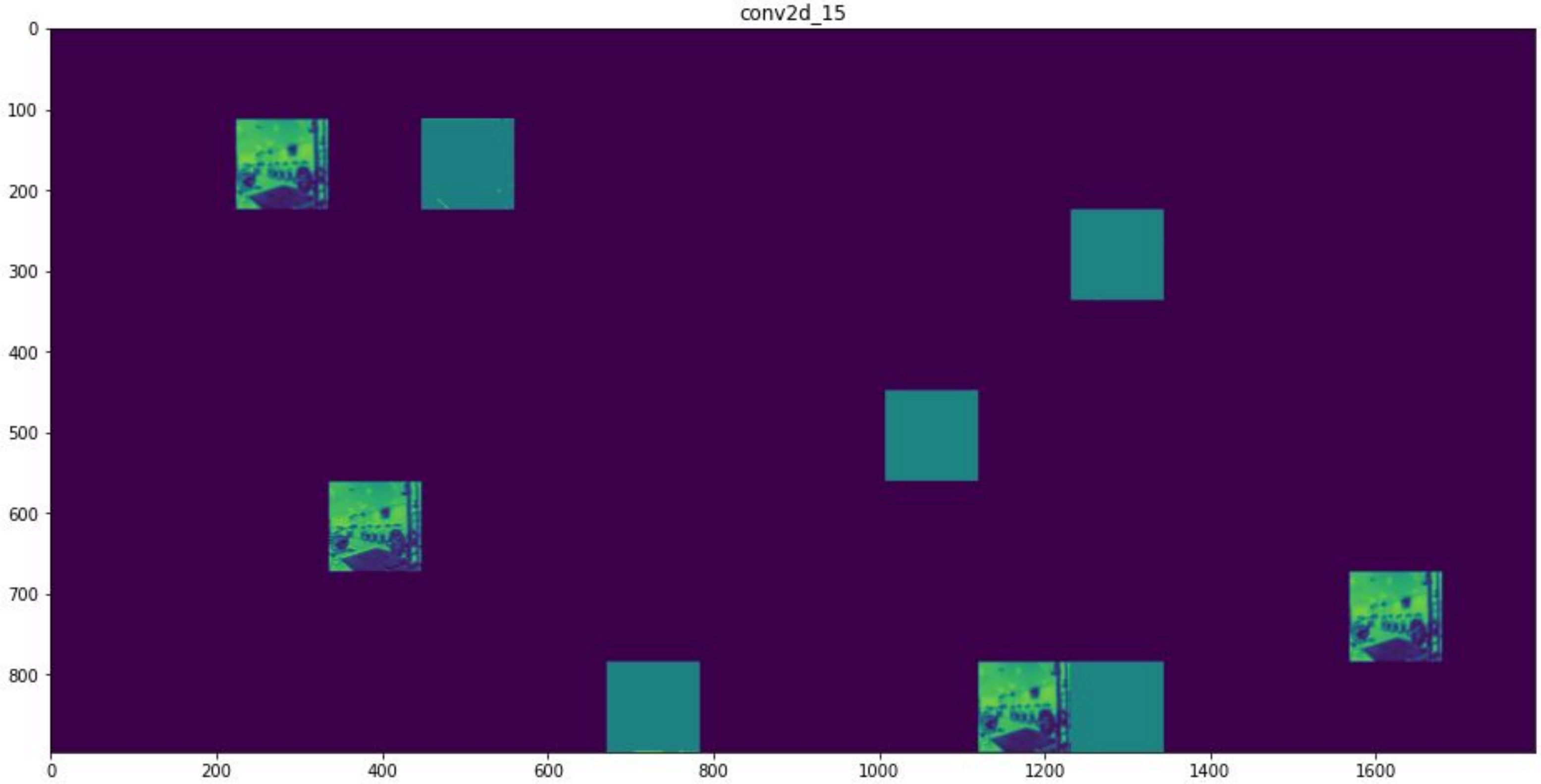




VGG16 Performance on Indoor Scenes



VGG16 Performance on Indoor Scenes



Inception and GoogleNet

- Developed in 2014 by a group of researchers at Google
- GoogleNet is a configuration of the Inception Network
- Created a deeper network than VGGNet while reducing the number of parameters.
- Introduced the Inception Module
- Top-5 error rate of 6.67% on ImageNet

Inception Module

- Researchers at Google decided to implement different sized convolutional layers and pooling layer all together in one block.
- The network architecture is developed stacking series of inception modules together

Inception Module

Inception module with dimensionality reduction

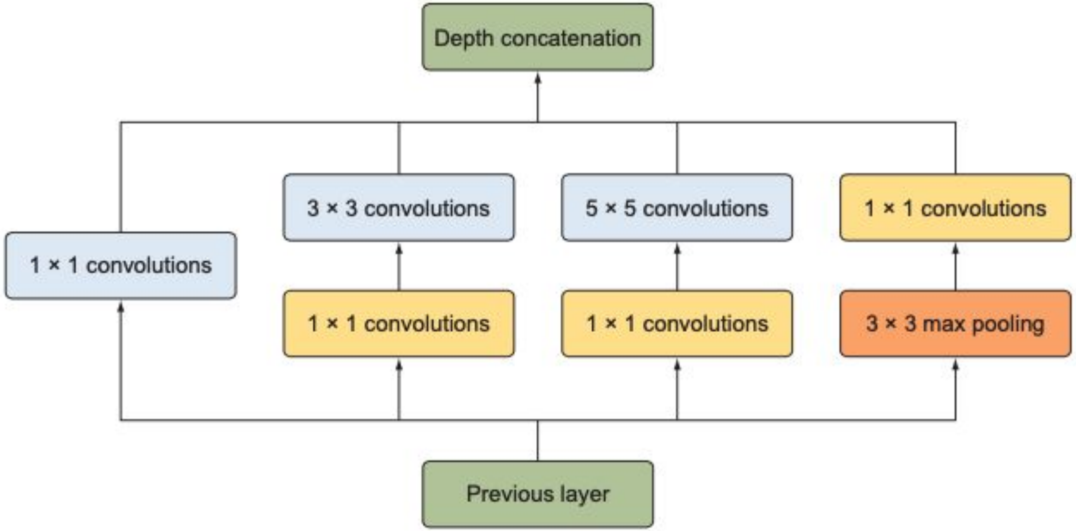


Figure 5.13 Building an inception module with dimensionality reduction

Inception Architecture

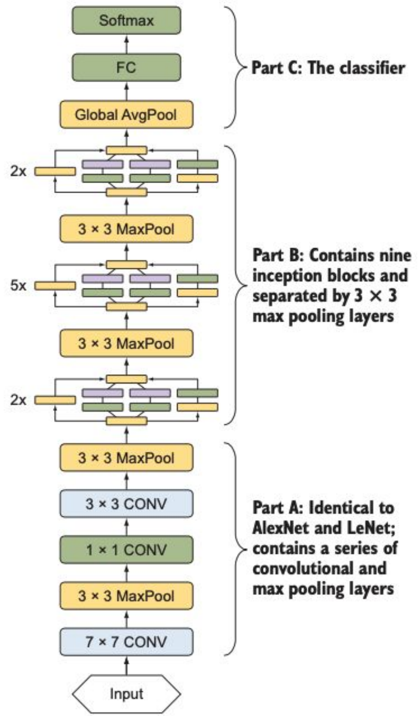
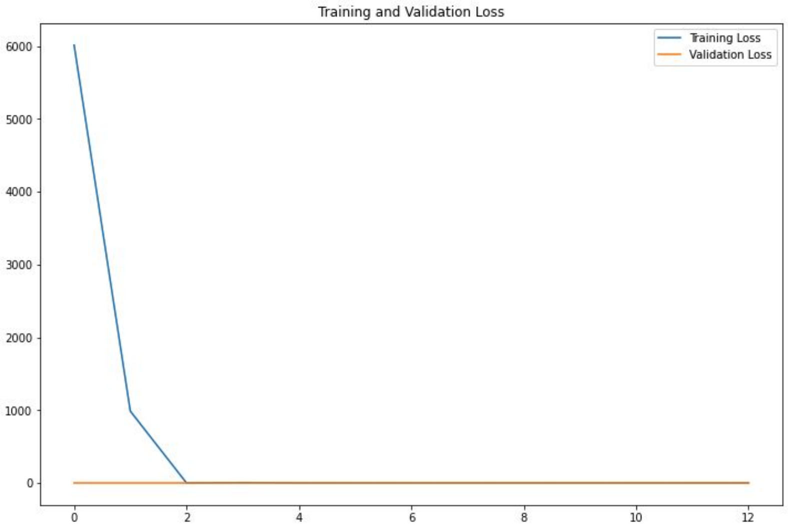


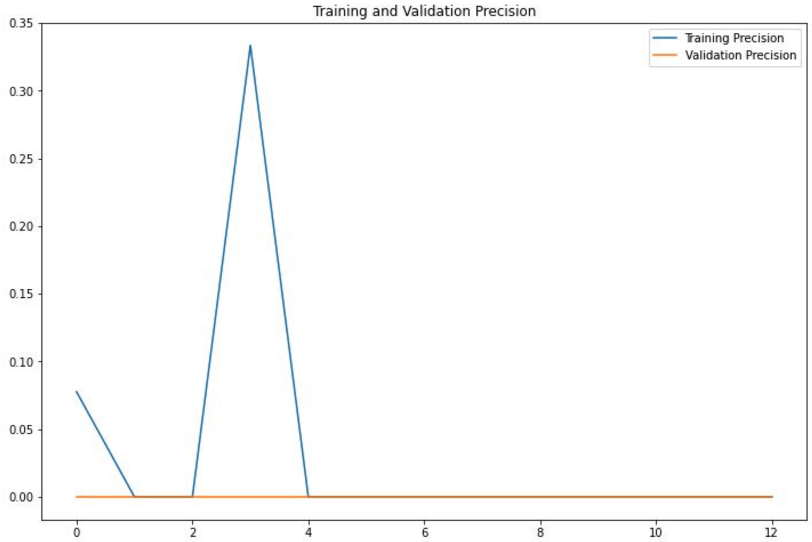
Figure 5.15 The full GoogLeNet model consists of three parts: the first part has the classical CNN architecture like AlexNet and LeNet, the second part is a stack of inceptions modules and pooling layers, and the third part is the traditional fully connected classifiers.

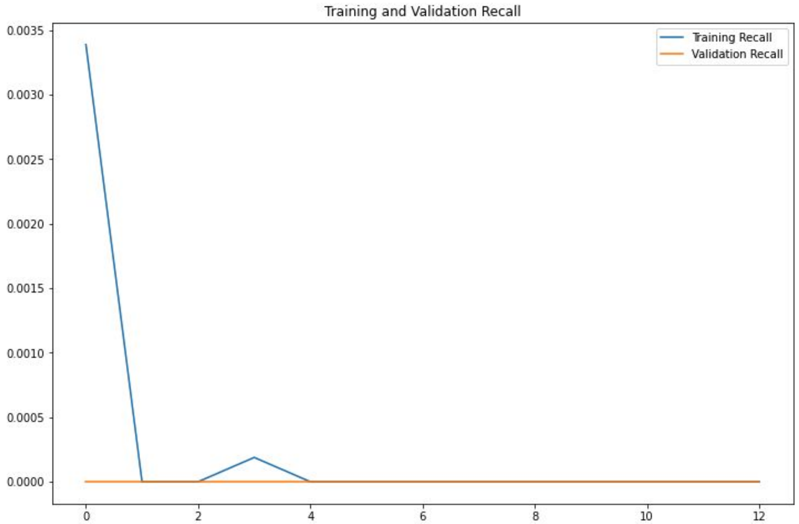
GoogleNet Performance on Indoor Scenes

- Top accuracy of around 10.8%

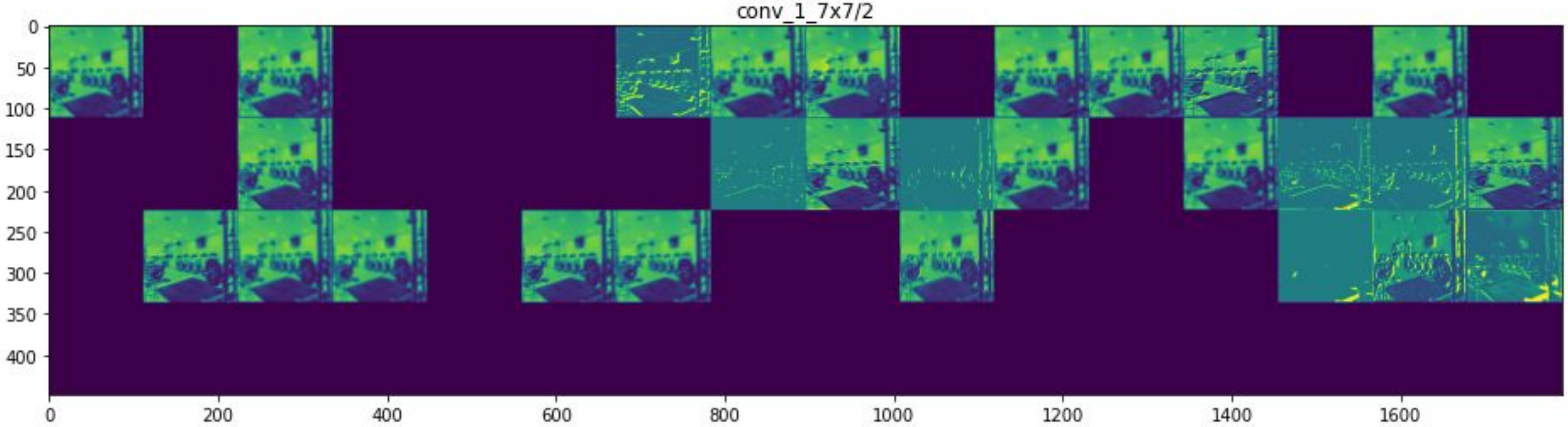


GoogleNet Performance on Indoor Scenes

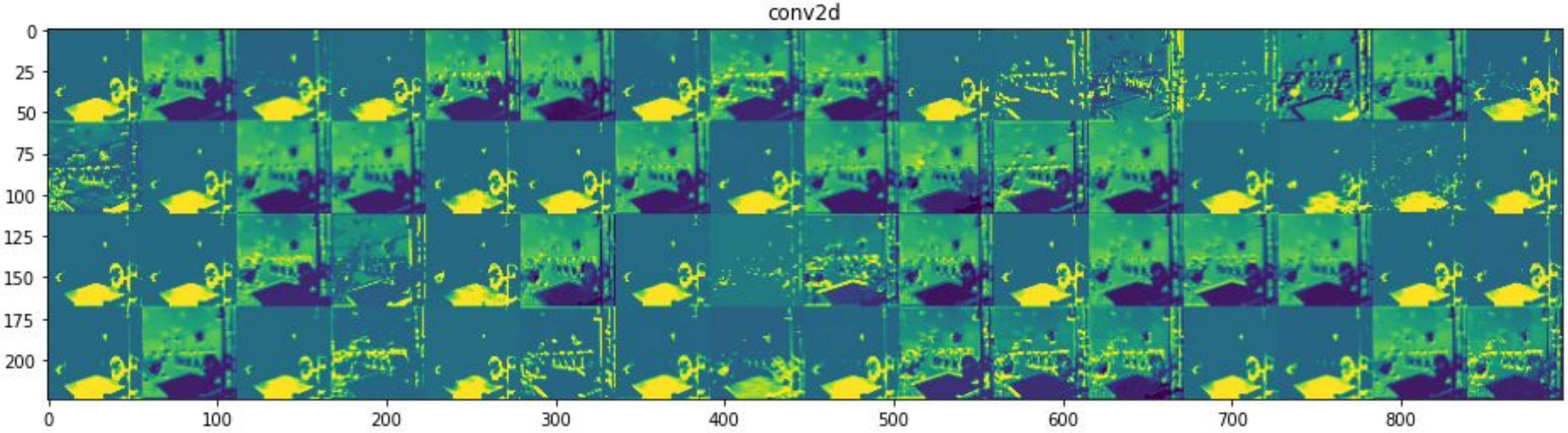




GoogeNet Performance on Indoor Scenes



GoogleNet Performance on Indoor Scenes



ResNet

- Residual Neural Network (ResNet) developed in 2015 by a group of researchers from the Microsoft Research team.
- Introduced the Residual Module with skip connections
- Uses Batch Normalization heavily for hidden layers
- Able to achieve really deep neural networks with 50, 101, and 152 weight layers with lower complexity than smaller networks like VGGNet.
- Top-5 error rate of 3.57% on ImageNet

ResNet Skip Connections

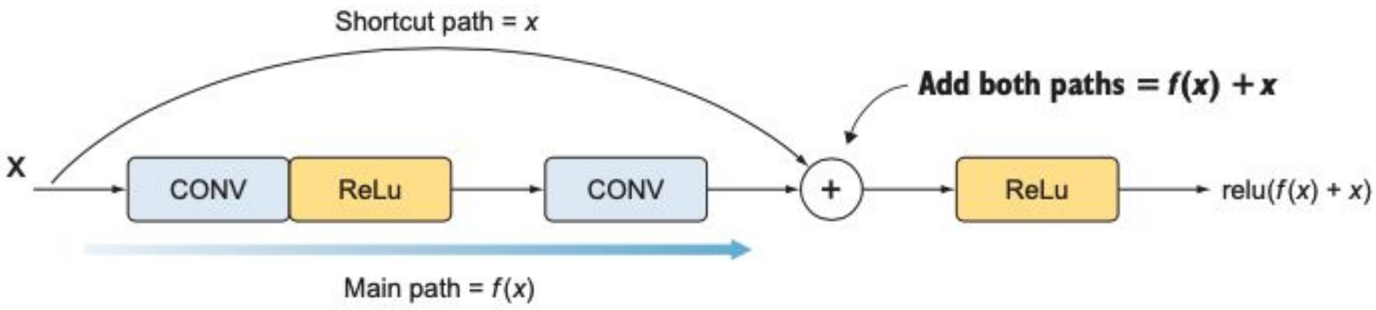


Figure 5.20 **Adding the paths and applying the ReLU activation function to solve the vanishing gradient problem that usually comes with very deep networks**

Residual Block

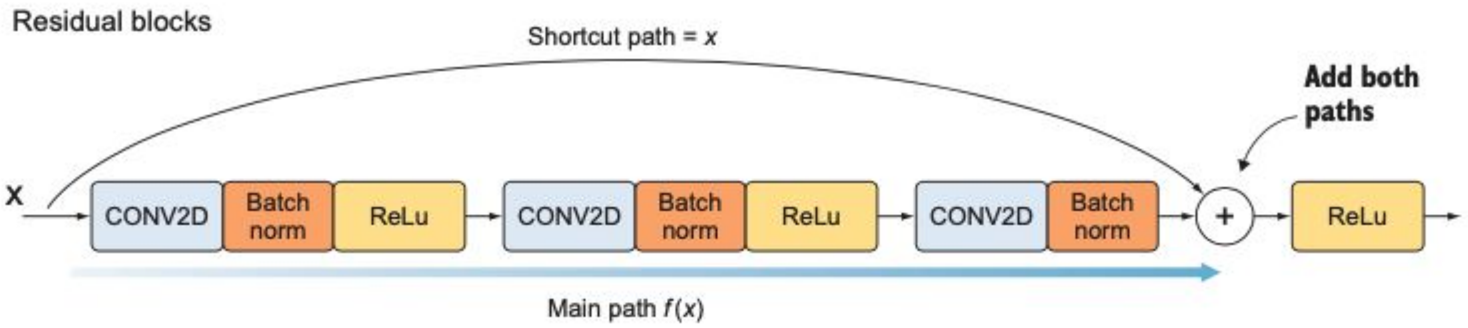


Figure 5.22 The output of the main path is added to the input value through the shortcut before they are fed to the ReLU function.

Residual Block

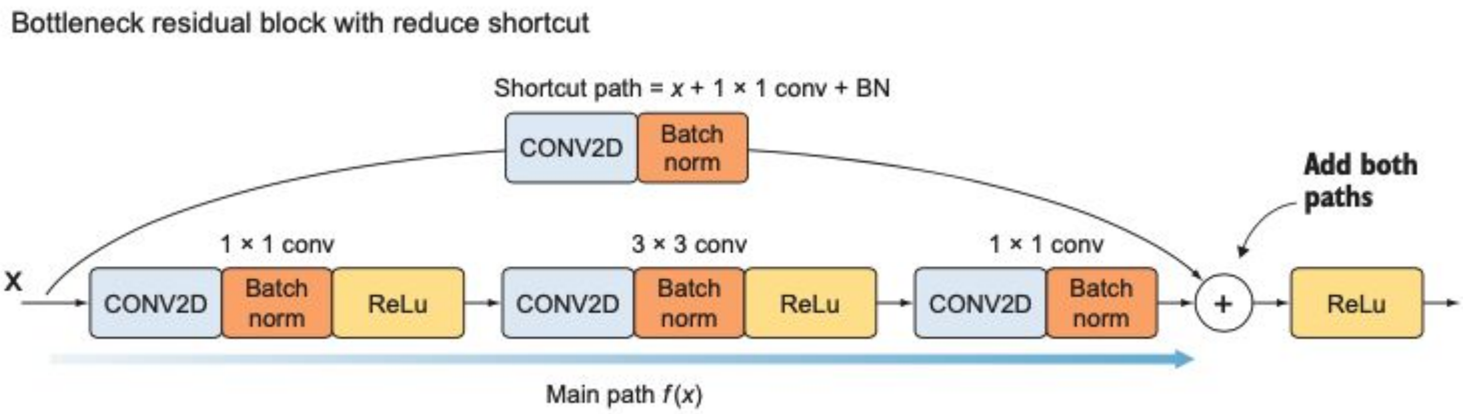


Figure 5.23 **To reduce the input dimensionality, we add a bottleneck layer (1 × 1 convolutional layer + batch normalization) to the shortcut path. This is called the *reduce shortcut*.**

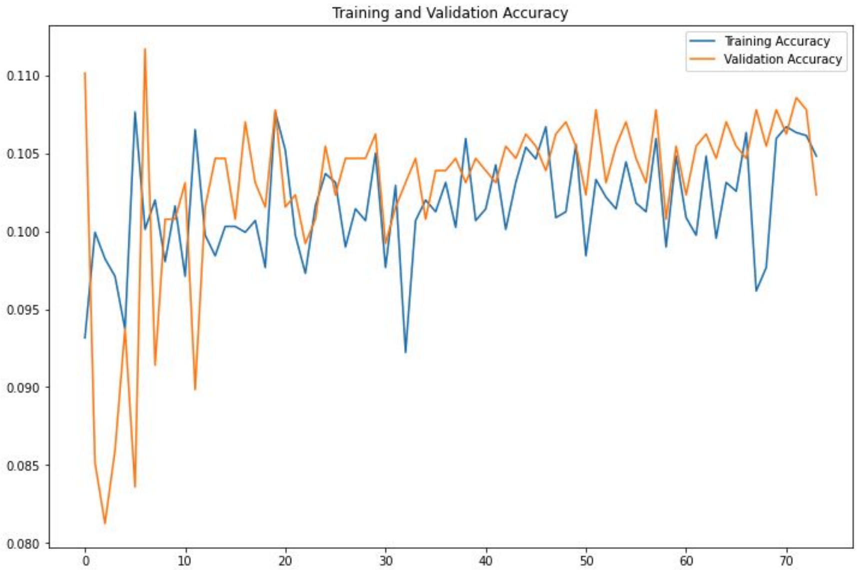
ResNet Architecture

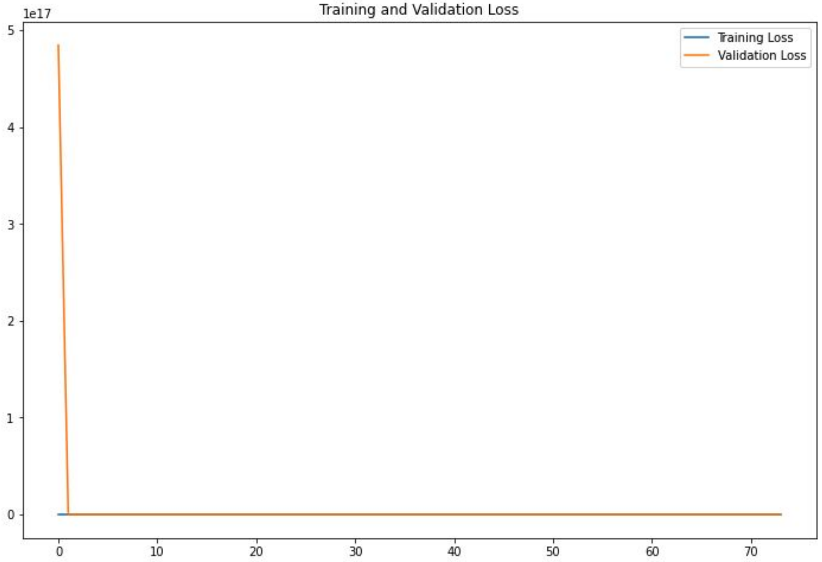
Layer name	Output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112x112	7x7, 64, stride 2				
conv2_x	56x56	3x3, maxpool, stride 2				
		[3x3, 64; 3x3, 64] x2	[3x3, 64; 3x3, 64] x3	[1x1, 64; 3x3, 64; 1x1, 256] x3	[1x1, 64; 3x3, 64; 1x1, 256] x3	[1x1, 64; 3x3, 64; 1x1, 256] x3
conv3_x	28x28	[3x3, 128; 3x3, 128] x2	[3x3, 128; 3x3, 128] x4	[1x1, 128; 3x3, 128; 1x1, 512] x3	[1x1, 128; 3x3, 128; 1x1, 512] x4	[1x1, 128; 3x3, 128; 1x1, 512] x8
conv4_x	14x14	[3x3, 256; 3x3, 256] x2	[3x3, 256; 3x3, 256] x6	[1x1, 256; 3x3, 256; 1x1, 1024] x3	[1x1, 256; 3x3, 256; 1x1, 1024] x23	[1x1, 256; 3x3, 256; 1x1, 1024] x36
conv5_x	7x7	[3x3, 512; 3x3, 512] x2	[3x3, 512; 3x3, 512] x3	[1x1, 512; 3x3, 512; 1x1, 2048] x3	[1x1, 512; 3x3, 512; 1x1, 2048] x3	[1x1, 512; 3x3, 512; 1x1, 2048] x3
	1x1	Average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 5.24 Architecture of several ResNet variations from the original paper

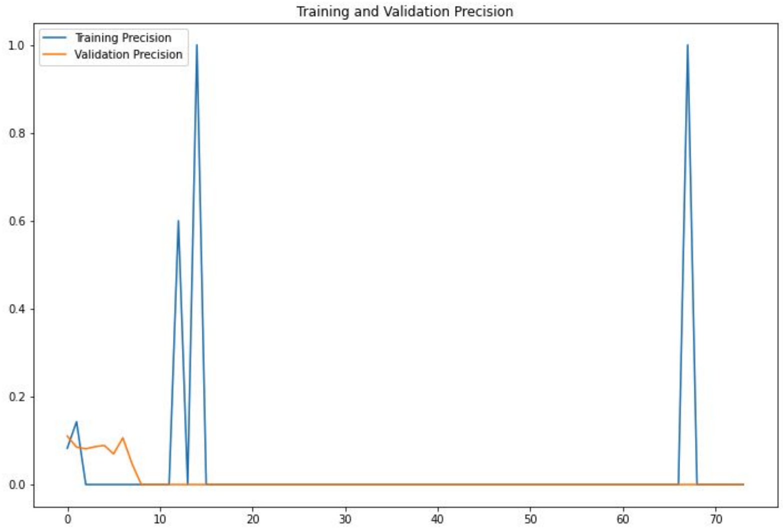
GoogleNet Performance on Indoor Scenes

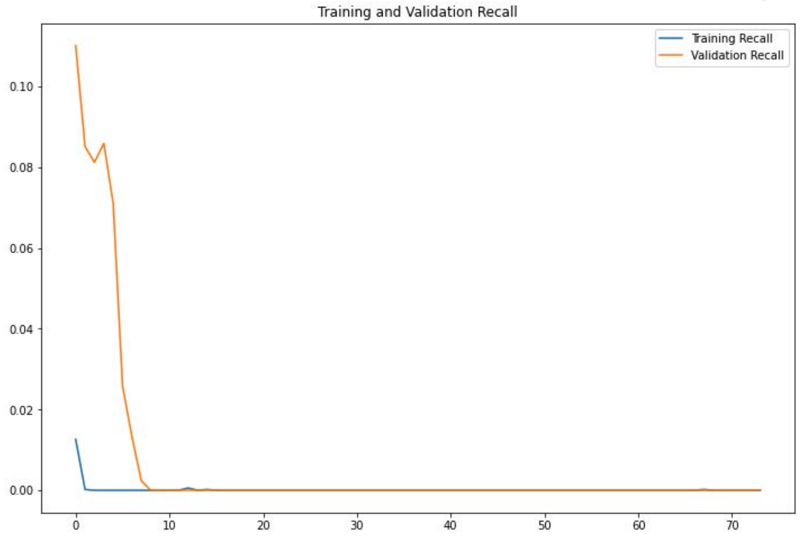
- Top accuracy of around 11%



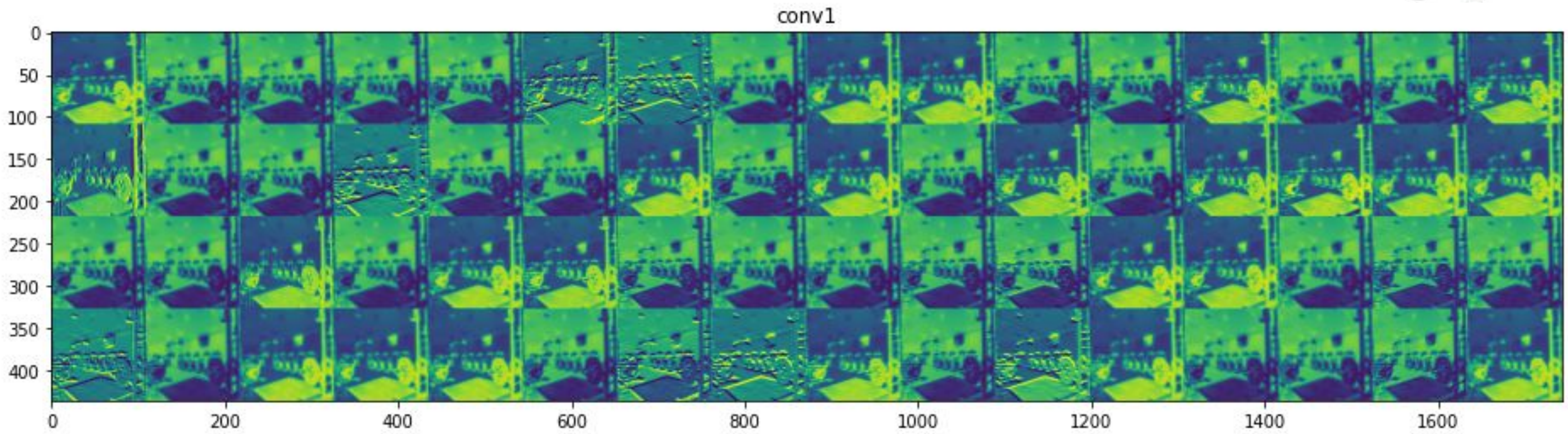


GoogleNet Performance on Indoor Scenes

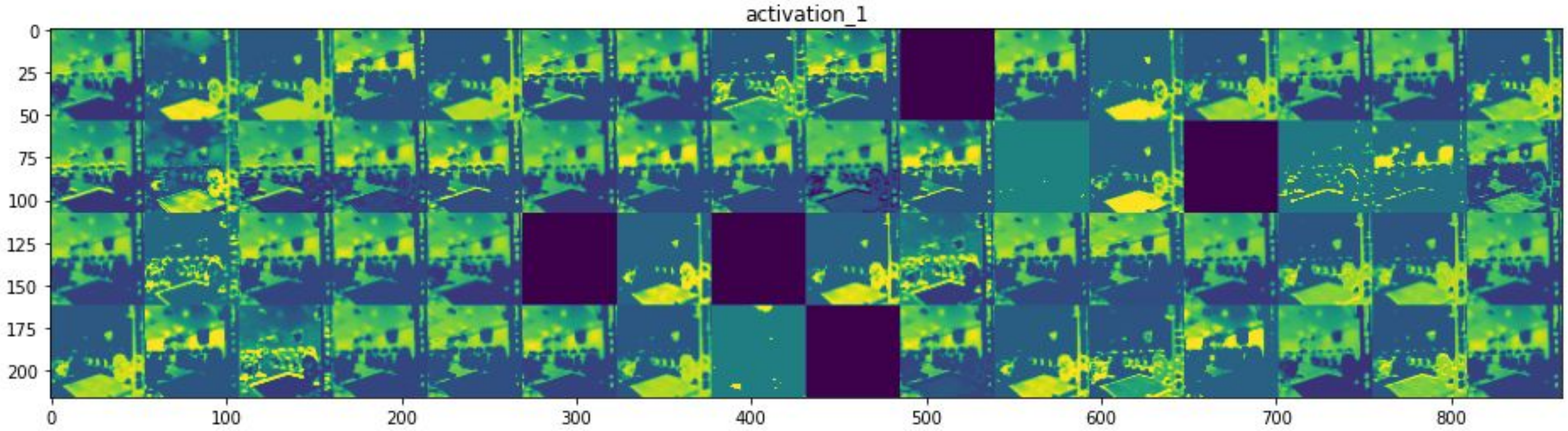




GoogeNet Performance on Indoor Scenes



GoogleNet Performance on Indoor Scenes



5.

Demo

Conclusion and Further Steps

- All 3 architectures performed poorly on the Indoor Scene dataset
- Each one performed the same on the test data
  - 20 out of 200 correctly classified
  - Correctly classified all 20 bathrooms
- These are great architectures to pull inspiration from
- Each dataset needs its own custom architecture
- Add More Data for training

Thanks!

Any questions?