Image Classification

Nicholas Synovic and Kenneth Hernandez

What is Image Classification?

- Image classification is the task of assigning a label or class to an **entire** image
- Usually supervised learning technique
  - Train on defined image labels to identify what makes up an image class

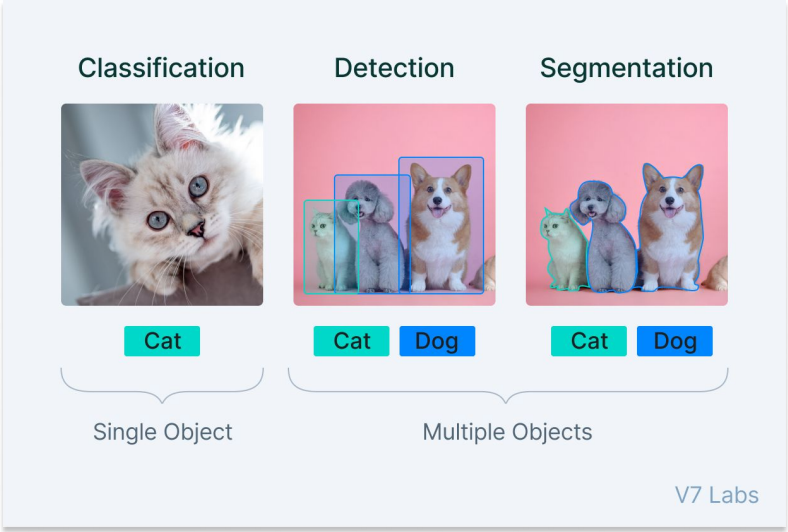
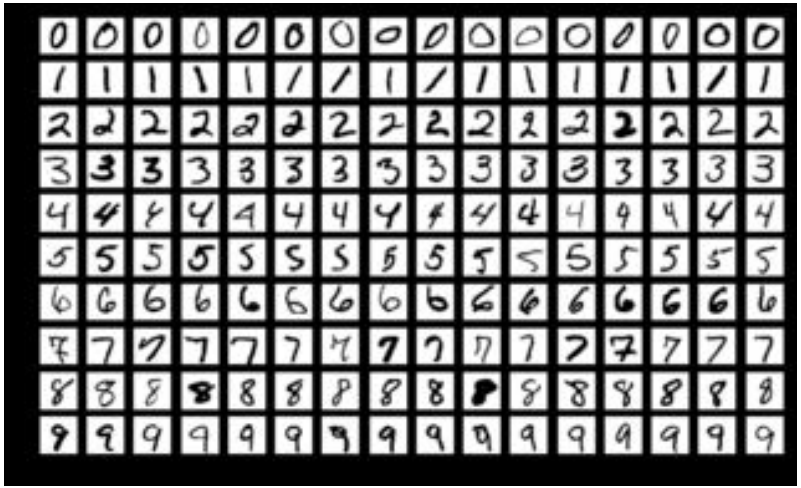


Image Classification DataSets

- Main goal is to have **a lot** of images with associated classes
  - Examples include:
    - MNIST
    - CIFAR 10
    - MS-COCO
    - ImageNet



ImageNet DataSet

- Why is this unique?
  - The first large scale dataset of classified images
  - Done in an attempt to see if CV models were databound or algorithmically bound
- 15 Million labeled high-resolution images
- 22,000 Categories

ImageNet Challenge

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) evaluates algorithms for object detection and image classification at a **large scale**

- Challenge uses a subset 1.2 Million training images
- 50,000 validation images
- 150,000 testing images

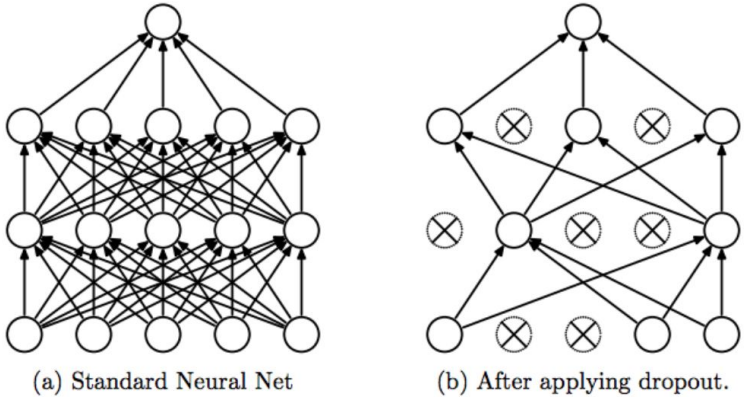
Techniques for Optimizing CNN Models

Image Manipulations for Larger Datasets

- Different crops
- Image expansion
- Color space manipulation
- Image flips
- Sliding view of an image
  - One 256 x 256 px image -> 5 224x224 images
- + more ways to exponentially grow a dataset

Dropout

- Set the output of each hidden neuron with a probability .5 to zero
  - Reduces co-adaptations, forcing the network to learn more robust features

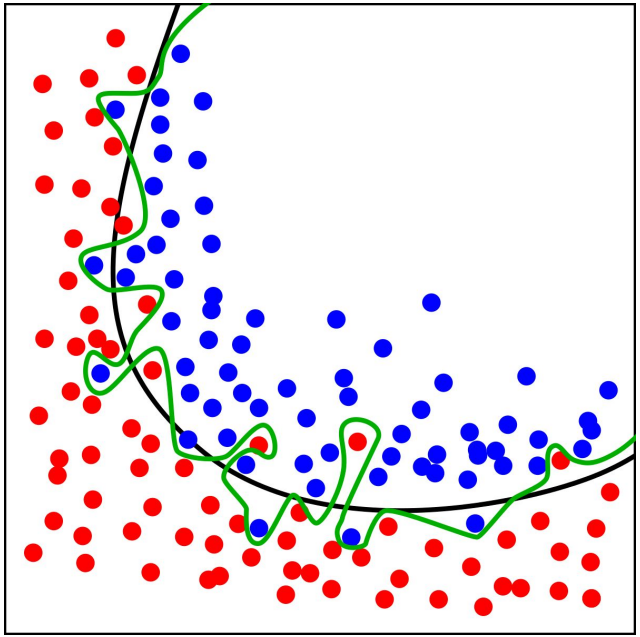


(a) Standard Neural Net

(b) After applying dropout.

Overfitting

- Solve overfitting via image manipulation + dropout

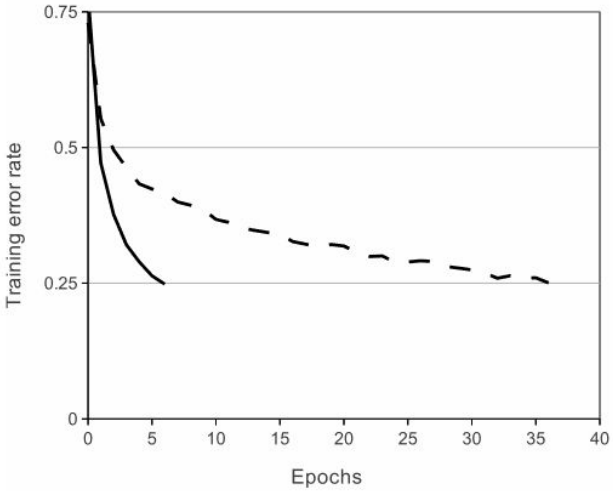


Rectified Linear Units (ReLU)

- Default activation function for many neural networks.
- Solves the issue of vanishing gradients

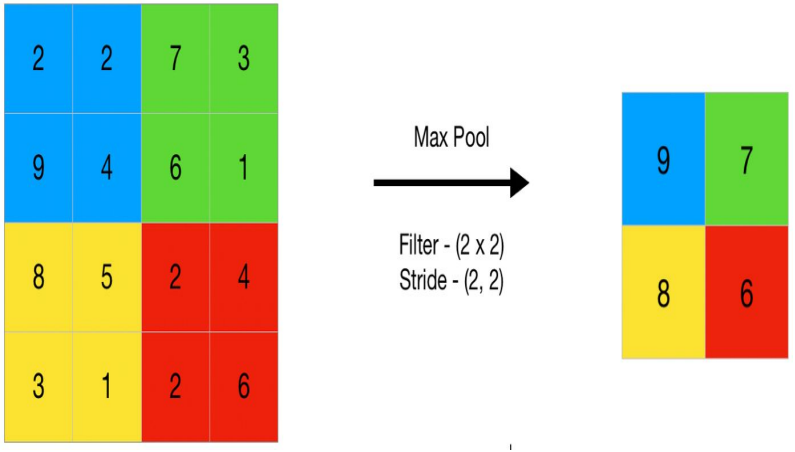
$$f(x) = \max(0, x)$$

Das



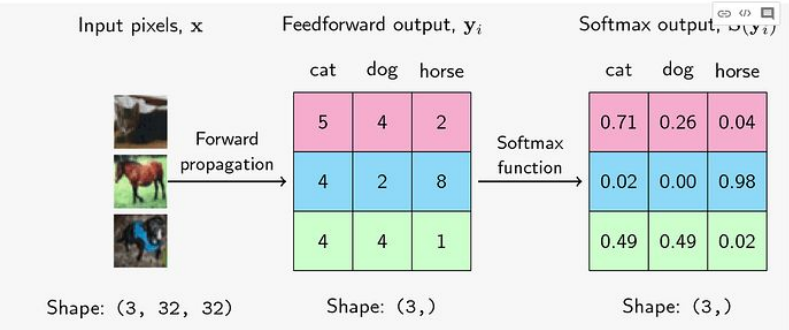
Pooling Layers

- Pooling is an approach to down sample feature maps
  - Average Pooling
    - Calculates the average for each patch of the value map
  - Max Pooling
    - Calculates the max value of each patch on the feature map

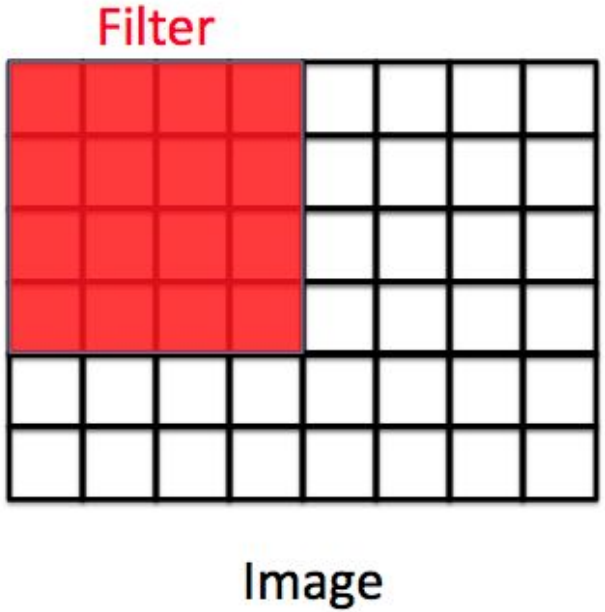


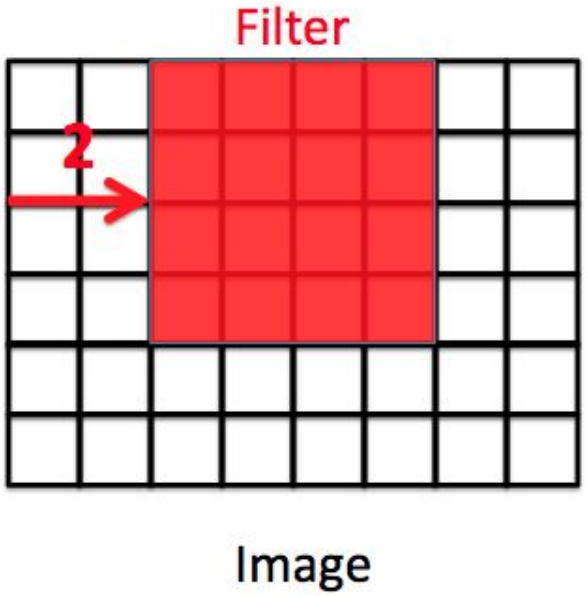
Softmax

- Softmax is a generalized form of logistic regression applied for multiple (more than two) classes
  - We need to figure out if a dog is different from a horse or a cat, how do we activate a neuron based on the distinction?



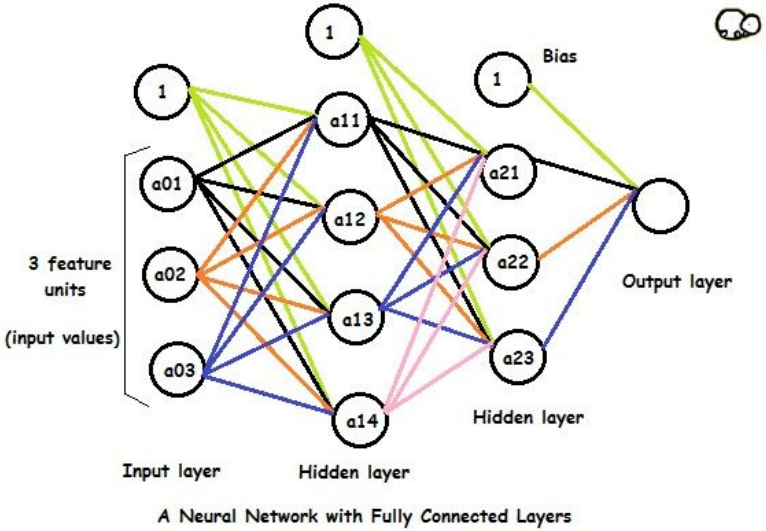
Strided Convolutions





Fully Connected Layers

- Every node in this layer is connected to every node in the previous layer
  - Is used to analyze all of the data generated from the previous layers
  - Typically used for classification purposes



What is Top 1% and Top 5% Error Image Classification Metrics?

- Top-1% indicates how many times the network has predicted the correct label with the highest probability
- Top-5% indicates how many times the correct label appears in the network's top five predicted classes

ImageNet Classification with Deep Convolutional Neural Networks (2012)

Overview

- Largest CNN model to date (2012)
  - Five CNN layers, 3 fully connected layers
- First to train on GPUs for faster training times and better performance
- Implemented early solutions to overcome overfitting in large DNNs
- SOTA performance in the 2012 ILSVRC competition

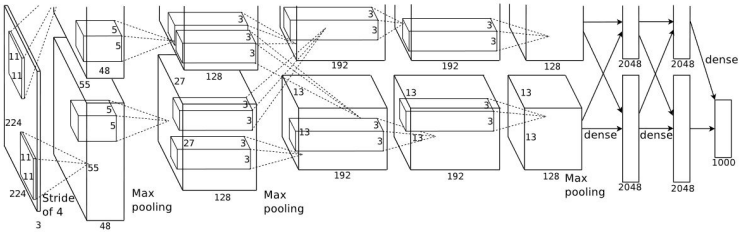


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

Background

- Previous CV models get near human performance on different tasks
- CV models didn't scale well to larger datasets
- Used a fixed resolution of 256 x 256
- One of the first to use GPUs with CNNs

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

Table 1: Comparison of results on ILSVRC-2010 test set. In *italics* are best results achieved by others.

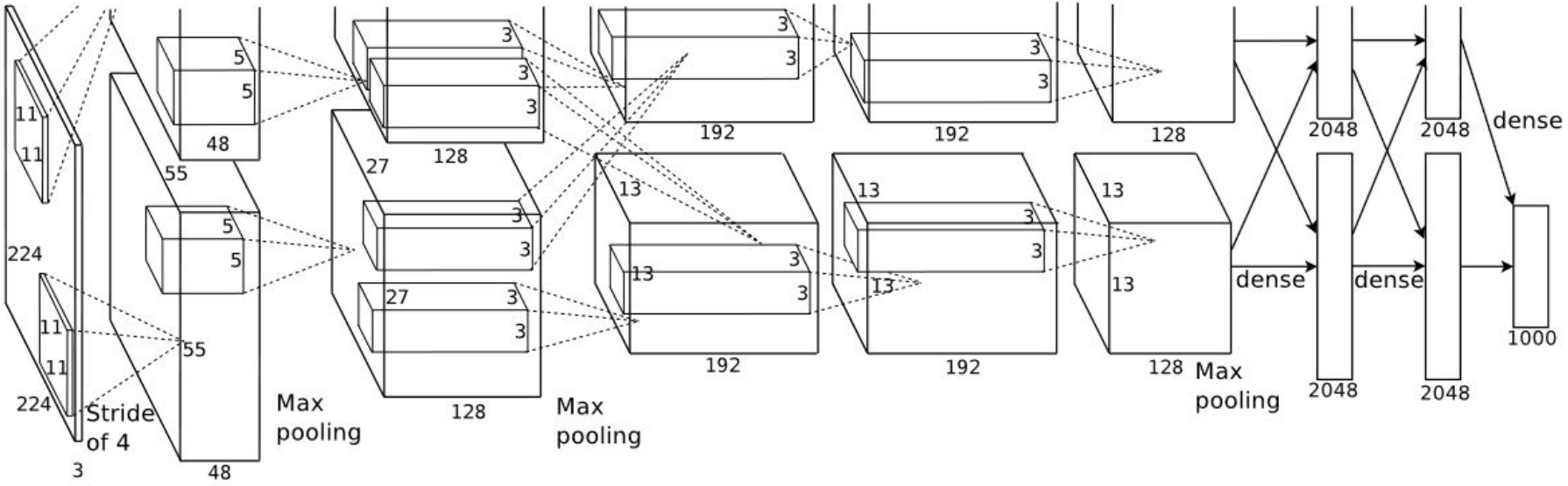


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

Results

- First to train CNN on GPUs
- Developed AlexNet as a reference implementation
  - Had 5 layers + 3 fully connected layers
  - Had 60 million parameters

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

Table 1: Comparison of results on ILSVRC-2010 test set. In *italics* are best results achieved by others.

Discussion Questions

- What prevented people from using GPUs previously?

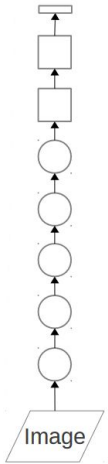
Going deeper with convolutions (2014)

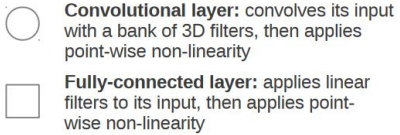
Overview

- Proposed the *Inception* image classification architecture
  - GoogLeNet is the name of the reference implementation
- GoogLeNet won the 2014 ImageNet Large-Scale Visual Recognition Challenge

Background

- LeNet-5 utilized CNNs for SOTA performance on MNIST and CIFAR
  - Larger datasets (ImageNet) requires increasing layer count and layer size while using dropout to address overfitting
- 2012's SOTA model was SuperVision
  - 7 deep layers
    - 5 CNN layers
    - 2 Fully connected layers
  - Trained on two NVIDIA GPUs
- 2013 's SOTA model was Clarifai
  - Current documentation points to CV as a Service…





SuperVision Architecture

Motivation

- To build an architecture that can find an optimal local sparse structure that can be approximated and covered with dense components
- To create a model with a *limit to inference operations*
  - 1.5 **billion** multiply-add operations
  - Utilizes dimensionality reduction in Inception modules to reduce computational cost

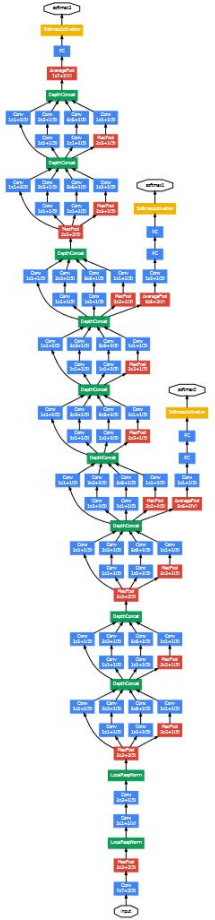
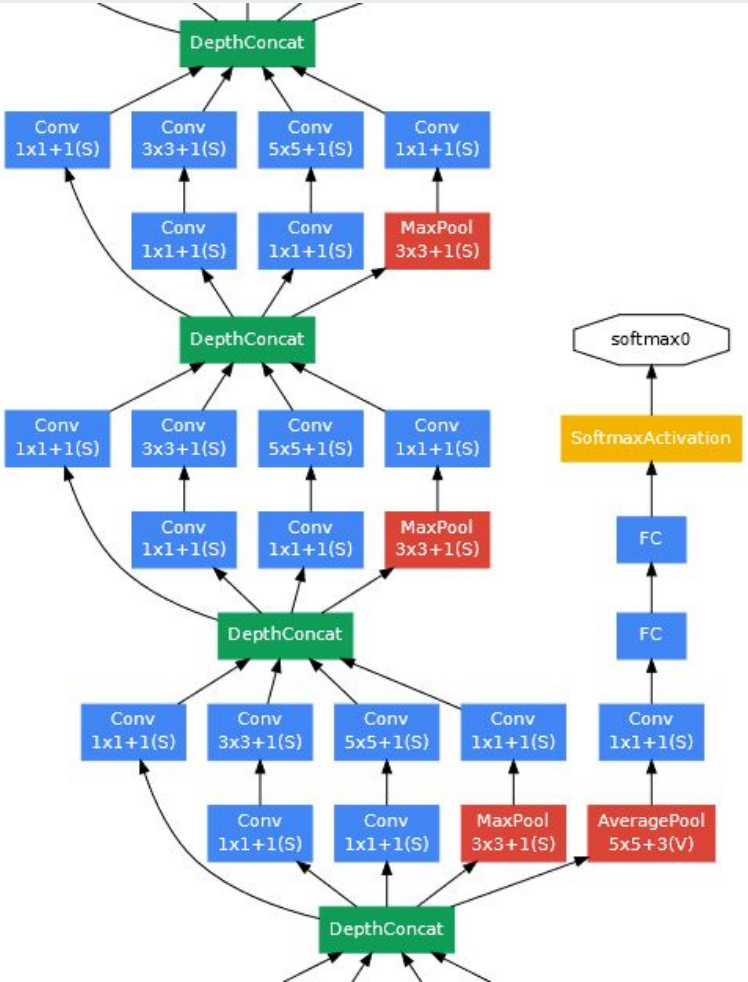


Figure 3: GoogLeNet network with all the bells and whistles


DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
softmax0
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
SoftmaxActivation
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
FC
DepthConcat
FC
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
AveragePool 5x5+3(V)
DepthConcat

Results

- GoogLeNet had SOTA performance on classification and detection in ImageNet Large-Scale Visual Recognition Challenge 2014
  - Had 22 layers
  - Had 6,797,700 (~ 6.8 million) parameters
- Is a DNN
- Showed that better algorithms can get better results
  - Better results are *always* not determined by the breadth and depth of a CNN, or the size of the dataset

Discussion Questions

- As the saying goes, “If it works, don’t fix it.” So what is the benefit of having competing CV architectures?
- Where else has the Inception architecture been utilized?
- The authors argue that better algorithms are the way forward for CV. But others (i.e AlexNet, VGG) argue that better datasets and deeper models are the way forward. Is there a compromise to be made here?

Very Deep Convolutional Networks For Large-Scale Image Recognition (2014; paper released in 2015)

Overview

- Discusses the usage of deep CNN
  - Reference model is VGG
- VGG won the 2014 ImageNet Classification and Object Detection challenges
  - Had 19 layers
  - Had 144 million parameters
- Tested a variety of CNN configurations to find the optimal depth

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Background

- CNNs are popular, but not very deep
  - AlexNet was only 5 layers
- CNN configurations are not well understood
- Better datasets and libraries exist now to train CNN models
  - GPU usage
  - Larger labelled datasets

Motivation

- To find the optimal size of a CNN architecture measured in layers deep
- GPUs are being utilized to train CNN
- Better labelled datasets are available to train CNN
- Optimizations for training CNNs
  - Smaller filter sizes
  - Input size is smaller than the size of the image
    - Allows for a single image (256 x 256) to be turned into five (top/bottom left/right and center) images for analysis

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Results

- Found that a CNN model architecture of 19 layers deep can achieve SOTA results on the 2014 ImageNet Classification and Object Detection challenges
- Is a DNN
- Optimized CNN configurations lead to better performing CNNs

Discussion Questions

- VGG is simpler to implement than GoogLeNet, but has a higher computational cost (measured in number of parameters). In your opinion, is implementation simplicity or model efficiency driving the development of image classification research?

Deep Residual Learning for Image Recognition (2015)

Overview

- Introduce new learning method for CNNs which is deeper and less complex than other CNNs
- ResNet-152 is the reference implementation
- Found that really deep CNNs can achieve SOTA performance
    - Developed solutions for overfitting and reducing training errors
- SOTA Performance on 2015 ImageNet detection, localization
- SOTA Performance on 2015 COCO detection and segmentation

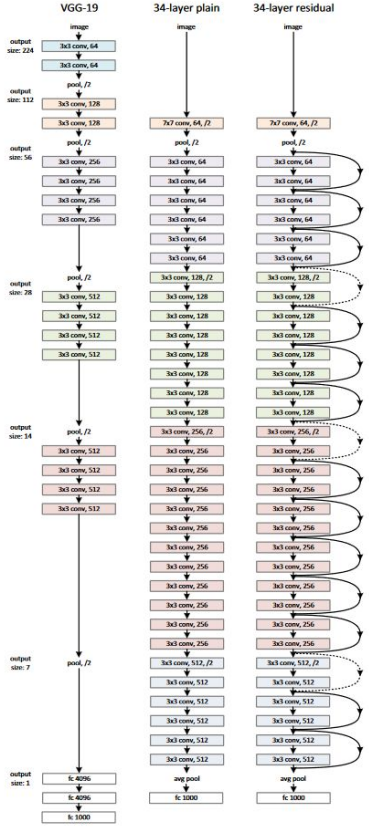


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

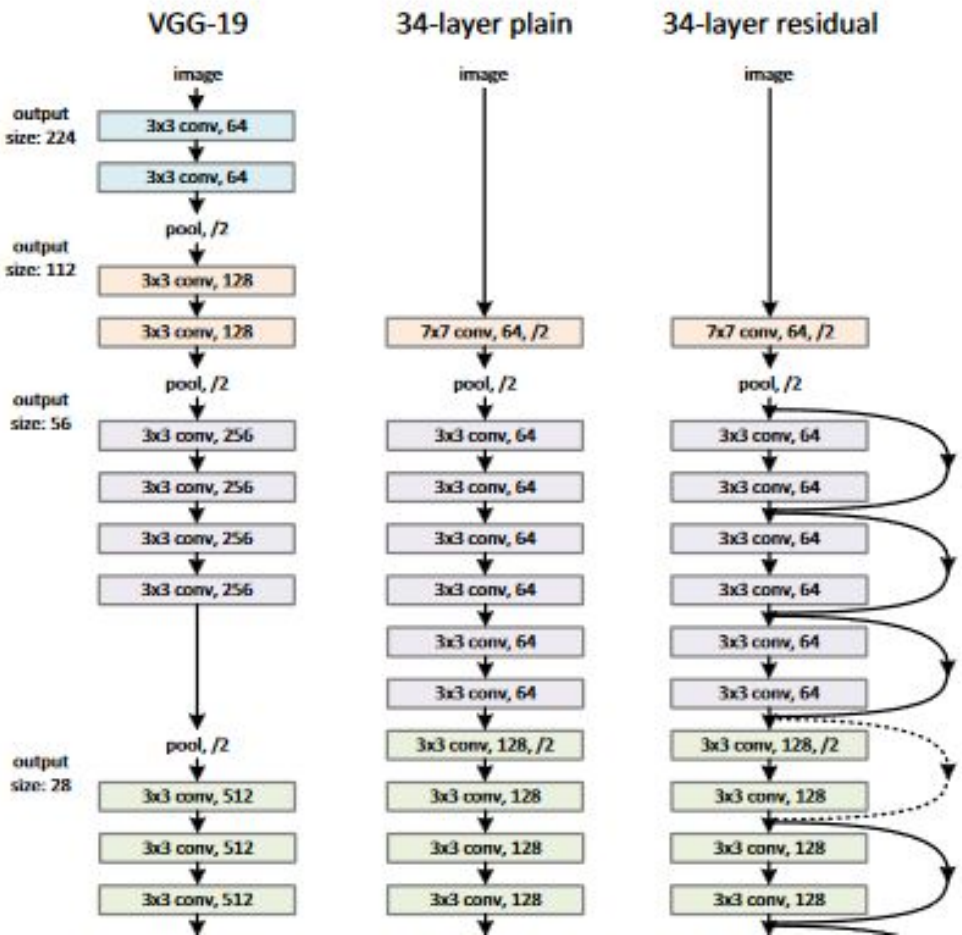


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

Residual Learning

- **Problem**: Gradients vanish or explode in DNNs

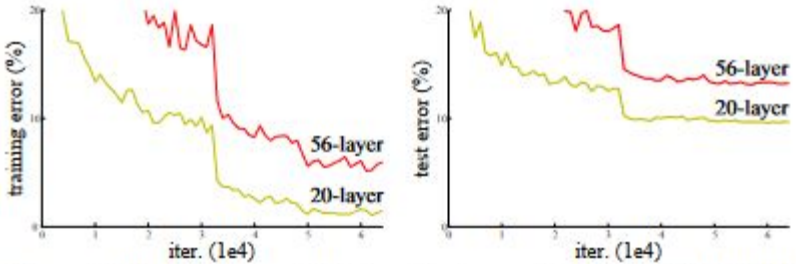


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

- **Solution:** Utilize a network that fits to a mapping rather than a network that learns a mapping
  - Implementations called residual blocks. Multiple residual blocks form a residual network

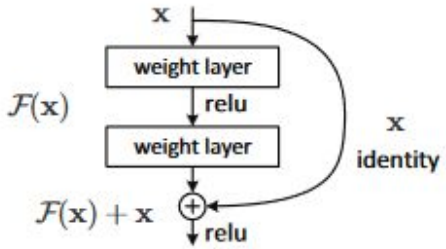


Figure 2. Residual learning: a building block.

Results

- Introduced Residual Learning and Identity Mapping by Shortcuts as solutions for developing really deep CNNs
- SOTA performance on detection, localization, and segmentation
- Explored models > 1000 layers and found that they weren't optimal
- Developed ResNet-152 as the reference implementation
    - Had 152 layers
    - Had 1.7 Million Parameters
- Improved upon VGG

Discussion Questions

- Let's talk about residual learning!

MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications (2017?)

Overview

- Introduced a new class of models targeting mobile devices called MobileNets
  - MobileNets aim to be small and efficient
- Utilize two hyperparameters to control the *width* and *resolution* of the model respectively
- Not benchmarked on the ImageNet Challenge
  - We will not be discussing SOTA performance for this model

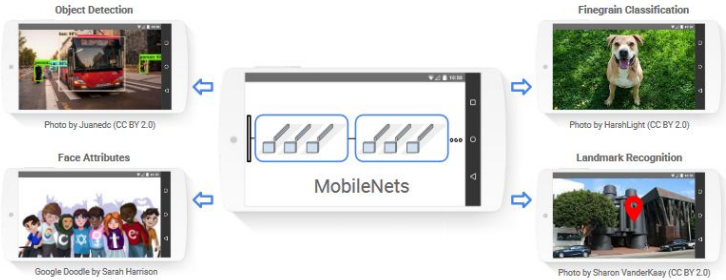


Figure 1. MobileNet models can be applied to various recognition tasks for efficient on device intelligence.

Background

- Efficiency is the main concern, not SOTA performance
- Expands upon the Inception V3 architecture in order to be efficient

Table 4. Depthwise Separable vs Full Convolution MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7%	4866	29.3
MobileNet	70.6%	569	4.2

Table 5. Narrow vs Shallow MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
0.75 MobileNet	68.4%	325	2.6
Shallow MobileNet	65.3%	307	2.9

Table 6. MobileNet Width Multiplier

Width Multiplier	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
0.75 MobileNet-224	68.4%	325	2.6
0.5 MobileNet-224	63.7%	149	1.3
0.25 MobileNet-224	50.6%	41	0.5

Table 7. MobileNet Resolution

Resolution	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
1.0 MobileNet-192	69.1%	418	4.2
1.0 MobileNet-160	67.2%	290	4.2
1.0 MobileNet-128	64.4%	186	4.2

Motivation

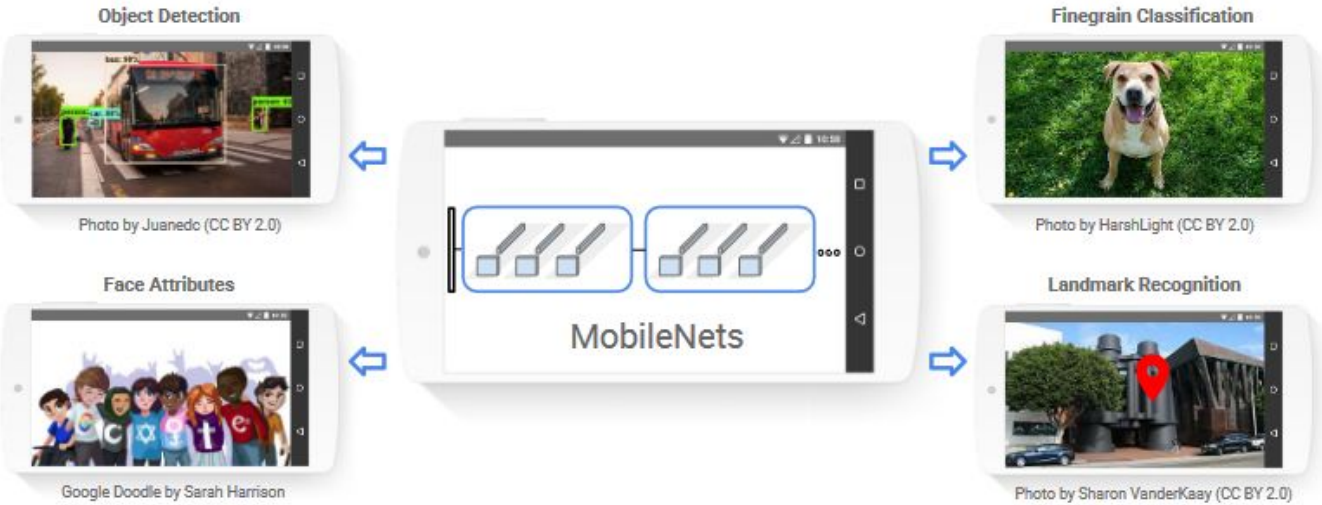


Figure 1. MobileNet models can be applied to various recognition tasks for efficient on device intelligence.

Results

- Developed the MobileNet architecture
- Developed several reference models
  - Models can be created by adjusting both the **width** and **resolution** of the model
  - All models have a depth of 28 layers

Table 4. Depthwise Separable vs Full Convolution MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7%	4866	29.3
MobileNet	70.6%	569	4.2

Table 5. Narrow vs Shallow MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
0.75 MobileNet	68.4%	325	2.6
Shallow MobileNet	65.3%	307	2.9

Table 6. MobileNet Width Multiplier

Width Multiplier	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
0.75 MobileNet-224	68.4%	325	2.6
0.5 MobileNet-224	63.7%	149	1.3
0.25 MobileNet-224	50.6%	41	0.5

Table 7. MobileNet Resolution

Resolution	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
1.0 MobileNet-192	69.1%	418	4.2
1.0 MobileNet-160	67.2%	290	4.2
1.0 MobileNet-128	64.4%	186	4.2

Discussion Questions

- Where can we take MobileNets further?
  - Think towards the future, not what we have accomplished