CSE 5194.01 - Introduction to High-Performance Deep Learning

ImageNet & VGG

Jihyung Kil
ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, University of Toronto
What is ImageNet?

- A large visual dataset designed for use in Computer Vision
  - 15 million labeled high-resolution images on 22,000 categories
  - Collected from the web and labeled by human annotators using Amazon’s Mechanical Turk crowd-sourcing tool

- Variable-resolution images
  - Rescaled the images to a fixed resolution of 256 x 256
  - Centered raw RGB values of the pixels

*Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks*
ILSVRC

- **ImageNet Large-Scale Visual Recognition Challenge**

- **Dataset**
  - A subset of ImageNet with 1000 images in each of 1000 categories
  - 1.2 million training images, 50,000 validation images, 150,000 testing images

- **Performance**
  - Krizhevsky et al – 16.4% error rate
  - ResNet – Best Performance (3.57% error rate)

*Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks*
Introduction – (1)

(ImageNet Classification with Deep Convolutional Neural Networks)

- Machine learning Models in Object recognition
  - To improve the performance?
    - Larger datasets & Larger models
    - Traditional ML models vs Neural Network

- Deep Convolutional Neural Network
  - Why DCNN?
    - Easier to train - Fewer connections and parameters
    - Performance is similar with standard neural network

_Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks_
Feedforward NN vs. Convolutional NN

Feedforward Neural Network

Convolutional Neural Network

*Alex Krizhevsky et al.*, *ImageNet Classification with Deep Convolutional Neural Networks*
Architecture
- DCNN with five convolutional and three fully-connected layers in the ILSVRC-2010 & 2012
- 60 million parameters to be trained
- Multiple technique used to improve its performance
  - GPU Parallelization, ReLU, Normalization, Dropout, etc.

Accelerator
- Two GTX 580 3GB GPUs
- Results can be improved by faster GPUs and bigger datasets
The Dataset

- **1000 ImageNet images in each of 1000 categories**
  - 1.2 million training images, 50,000 validation images, 150,000 testing images

- **ILSVRC-2010**
  - Test set labels available

- **ILSVRC-2012**
  - Test set labels not publicly available

- **Images of a fixed resolution 256 x 256**

- **Report error rates**
  - Top-1 vs Top-5

*Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks*
Important Techniques in DCNN

- **Activation Functions**
  - Step, Sigmoid, tanh, ReLU

- **Cross-GPU Parallelization**
  - Distribution of neurons, GPU communication

- **Pooling**
  - Max vs Average Pooling
  - Overlapping vs Non-Overlapping Pooling

*Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks*
ReLU Nonlinearity – (1)

- **Standard approach**
  - Saturating nonlinearities

- **Rectified Linear Units (ReLU)**
  - Non-Saturating nonlinearities

\[
\sigma(x) = \frac{e^x}{1 + e^x}
\]

\[
\tanh(x) = 2 \cdot \sigma(2x) - 1
\]

\[
f(x) = \text{Max}(0, x)
\]

---

*Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks*
Figure 1: A four-layer convolutional neural network with ReLUs (solid line) reaches a 25% training error rate on CIFAR-10 six times faster than an equivalent network with tanh neurons (dashed line). No regularization of any kind was employed. The magnitude of the effect demonstrated here varies with network architecture, but networks with ReLUs consistently learn several times faster than equivalents with saturating neurons.
Training on Multiple GPUs

- **GTX 580 GPU**
  - 3GB of memory
  - **1.2 million** training examples are too big to fit into one GPU!

- **Cross-GPU Parallelization (Two GPUs)**
  - Half of the kernels on each GPU
  - Communicate only in certain layers
    - E.g. The kernels of layer 3 take input from all kernel maps in layer 2. However, kernels in layer 4 take input only from those kernel maps in layer 3 which reside on the same GPU

*Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks*
Overlapping Max Pooling

- **Traditional Max Pooling**
  - \( S = Z \) (\( S \): Stride, \( Z \): Width of window)

  ![Grid](image1.png)

- **Overlapping Max Pooling**
  - \( S < Z \) (\( S \): Stride, \( Z \): Width of window)
  - E.g. \( S = 2, Z = 3 \)

  ![Grid](image2.png)

  This scheme reduces the top-1 and top-5 error rates by 0.4% and 0.3%, respectively

*Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks*
Overall Architecture – (1)

- **Input Image**: 224×224×3 pixels

- **1st Conv layer**
  - Input: 224 x 224 x 3
  - 96 kernels of size 11×11×3
  - 253,440 neurons
  - Response-normalization & Max-pooling

- **2nd Conv layer**
  - Input: output of 1st Conv layer
  - 256 kernels of size 5 x 5 x 48
  - 186,624 neurons
  - Response-normalization & Max-pooling

- **3rd Conv layer**
  - Input: output of 2nd Conv layer
  - 384 kernels of size 3 x 3 x 256
  - 64,896 neurons

*Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks*
Overall Architecture – (2)

- **4th Conv layer**
  - Input: Output of 3rd Conv layer
  - 384 kernels of size $3 \times 3 \times 192$
  - 64,896 neurons

- **5th Conv layer**
  - Input: output of 4th Conv layer
  - 256 kernels of size $3 \times 3 \times 192$
  - 186,624 neurons
  - Max-pooling

- **6th Fully-connected layer**
  - Input: output of 5th Conv layer
  - 4,096 neurons

*Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks*
Overall Architecture – (3)

- **7th Fully-connected layer**
  - Input: Output of 6th Fully-connected layer
  - 4,096 neurons

- **8th Fully-connected layer**
  - Input: output of 7th Fully-connected layer
  - 1,000 neurons (Categories)

- The kernels of the 2\(^{nd}\), 4\(^{th}\), 5\(^{th}\) Conv layers are connected only to those kernel maps in the previous layer which reside on the same GPU

- The kernels of the 3\(^{rd}\) Conv layer are connected to all kernel maps in the second layer. The neurons in the fully-connected layers are connected to all neurons in the previous layer

*Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks*
Overfitting

- A large number of neurons and parameters to be trained:
  - 253,440 - 186,224 - 64,896 - 64,896 - 43,264 - 4,096 - 4,096 - 1,000 neurons in the network
  - 60 million parameters to be trained!!

- Dropout
  - Ignore some of neurons in layers during training phases, which are chosen randomly
  - Normally, probability of 0.5 used to determine whether to drop or not
  - At test time, use all neurons but multiply their outputs by 0.5

*Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks*
Learning & Weight Update

- Stochastic gradient descent with a batch size of 128 examples, momentum of 0.9, and weight decay of 0.0005

- The Update rule for weight $W$:

\[
v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w} \middle| w_i \right\rangle_{D_i}
\]

\[
w_{i+1} := w_i + v_{i+1}
\]

where $i$ is the iteration index, $v$ is the momentum variable, $\epsilon$ is the learning rate, and $\left\langle \frac{\partial L}{\partial w} \middle| w_i \right\rangle_{D_i}$ is the average over the $i$th batch $D_i$ of the derivative of the objective with respect to $w$, evaluated at $w_i$.

- Initialized the weights with 0.01
- Initialized the biases in 2nd, 4th, 5th, and fully-connected layers with 1. The remaining layers with 0

*Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks*
Results: Performance

- **ILSVRC-2010**

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1</th>
<th>Top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse coding [2]</td>
<td>47.1%</td>
<td>28.2%</td>
</tr>
<tr>
<td>SIFT + FVs [24]</td>
<td>45.7%</td>
<td>25.7%</td>
</tr>
<tr>
<td>CNN</td>
<td>37.5%</td>
<td>17.0%</td>
</tr>
</tbody>
</table>

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

- **ILSVRC-2012**

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1 (val)</th>
<th>Top-5 (val)</th>
<th>Top-5 (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT + FVs [7]</td>
<td>—</td>
<td>—</td>
<td>26.2%</td>
</tr>
<tr>
<td>1 CNN</td>
<td>40.7%</td>
<td>18.2%</td>
<td>—</td>
</tr>
<tr>
<td>5 CNNs</td>
<td>38.1%</td>
<td>16.4%</td>
<td><strong>16.4%</strong></td>
</tr>
<tr>
<td>1 CNN*</td>
<td>39.0%</td>
<td>16.6%</td>
<td>—</td>
</tr>
<tr>
<td>7 CNNs*</td>
<td>36.7%</td>
<td>15.4%</td>
<td><strong>15.3%</strong></td>
</tr>
</tbody>
</table>

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were “pre-trained” to classify the entire ImageNet 2011 Fall release. See Section 6 for details.
Results: What network Learned - (1)

- **Top-5 Predictions**
  - Having a correct label for most of images
  - Successfully identified an object regardless of its position (E.g. mite)
  - Most of top-5 predictions appear reasonable (E.g. leopard)

Eight ILSVRC-2010 test images and the five labels considered most probable by our model. The correct label is written under each image, and the probability assigned to the correct label is also shown with a red bar (if it happens to be in the top 5).

Alex Krizhevsky et al., *ImageNet Classification with Deep Convolutional Neural Networks*
Results: What network Learned - (2)

- Feature vectors with Euclidean distance
  - Feature activation vectors in the last hidden layer (4,096)
- Five images from training set similar with a image from testing set
- Correctly identified an object even if different poses

Five ILSVRC-2010 test images in the first column. The remaining columns show the six training images that produce feature vectors in the last hidden layer with the smallest Euclidean distance from the feature vector for the test image.
Conclusion

- A large, deep convolutional neural network is capable of achieving record-breaking results on image dataset only using supervised learning.

- The network’s performance decreases if a single convolutional layer is removed. E.g. Removing any of the middle layers results in a loss of about 2% for the top-1 performance of the network (Depth of the network important!!)

- Results have improved as the network have become larger and trained longer.
  - Big Dataset & High Performance Computing required!!

*Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks*
VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Visual Geometry Group (VGG),
Department of Engineering Science, University of Oxford
Introduction

(VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION)

◦ Large Image Dataset & high-performance computing systems
  ◦ ImageNet, GPUs, large-scale distributed clusters

◦ Deep Convolutional neural network
  ◦ Bring a great success in large-scale image recognition
  ◦ Krizhevsky et al., 2012; Zeiler & Fergus, 2013; Sermanet et al., 2014; Simonyan & Zisserman, 2014

◦ In this paper..
  ◦ Modified the architecture of CNN
    ◦ Depth & Filter size

Karen Simonyan and Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition
Configuration – (1)

**Architecture**
- Designed a neural network based on the principles used in Krizhevsky et al. (2012)
- Input: 224 x 224 RGB Images
- 3 x 3 filters & 1 x 1 filters
- Stride 1 & Padding 1
- Max pooling over 2 x 2 pixel window with stride 2
- Different Depth & number of channels in different architectures
  - 11, 13, 16, 19 weight layers
  - 64, 128, 256, 512 channels
  - 3 fully-connected layers
Configuration – (2)

- **Architecture (Continued..)**
  - 3 fully-connected layers
    - 1st, 2nd layers: 4096 neurons each
    - 3rd layer: 1,000 neurons (one for each class)
    - Last layer: Softmax layer
  - ReLU activation function
  - **No** Local Response Normalization layer in networks except for one
  - Number of channels start from 64 to 512, increasing by a factor of 2
Configuration – (3)

Table 1: ConvNet configurations (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as “conv<receptive field size> - number of channels”. The ReLU activation function is not shown for brevity.

<table>
<thead>
<tr>
<th>Column</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>weight layers</strong></td>
<td>11</td>
<td>11</td>
<td>13</td>
<td>16</td>
<td>19</td>
</tr>
<tr>
<td><strong>input (224 × 224 RGB image)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>conv2-64</strong></td>
<td><strong>conv3-64</strong></td>
<td>conv3-64</td>
<td>conv3-64</td>
<td>conv3-64</td>
<td>conv3-64</td>
</tr>
<tr>
<td><strong>conv3-128</strong></td>
<td>conv3-128</td>
<td>conv3-128</td>
<td>conv3-128</td>
<td>conv3-128</td>
<td>conv3-128</td>
</tr>
<tr>
<td><strong>conv3-256</strong></td>
<td>conv3-256</td>
<td>conv3-256</td>
<td>conv3-256</td>
<td>conv3-256</td>
<td>conv3-256</td>
</tr>
<tr>
<td><strong>conv3-512</strong></td>
<td>conv3-512</td>
<td>conv3-512</td>
<td>conv3-512</td>
<td>conv3-512</td>
<td>conv3-512</td>
</tr>
<tr>
<td><strong>conv3-1024</strong></td>
<td>conv3-1024</td>
<td>conv3-1024</td>
<td>conv3-1024</td>
<td>conv3-1024</td>
<td>conv3-1024</td>
</tr>
<tr>
<td><strong>maxpool</strong></td>
<td><strong>maxpool</strong></td>
<td><strong>maxpool</strong></td>
<td><strong>maxpool</strong></td>
<td><strong>maxpool</strong></td>
<td><strong>maxpool</strong></td>
</tr>
</tbody>
</table>

Table 2: Number of parameters (in millions).

<table>
<thead>
<tr>
<th>Network</th>
<th>A-LRN</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of parameters</td>
<td>133</td>
<td>133</td>
<td>134</td>
<td>138</td>
<td>144</td>
</tr>
</tbody>
</table>
Discussion

- **Why small receptive fields (filter)?**
  - Incorporating three non-linear rectification layers instead of a single one makes the decision function more discriminative
  - Decreasing the number of parameters:
    - 3 x 3 Conv layers VS. 11 x 11 Conv layers in Krizhevsy et al., 2012
    - Assuming that both the input and the output of a three-layer 3 x 3 convolutional stack has C channels parametrized by $3(3^2C^2) = 27C^2$ weights; at the same time, a single 11 x 11 conv. layer would require $11^2C^2 = 121C^2$ parameters, i.e. ~348% more
  - Multiple other networks with small filters
    - GoogleLeNet & Goodfellow et al. (2014)

- **Linear transformation of the input channels**
  - 1 x 1 conv. Layers
    - Increase the non-linearity of the decision function without affecting the receptive fields of the conv. Layers

Karen Simonyan and Andrew Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*
Learning & Weight Update – (1)

- Stochastic gradient descent with a batch size of 256 examples, momentum of 0.9, and weight decay of 0.0005

- The Update rule for weight $W$:

  $$
  v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w_i} \right\rangle_{D_i}
  $$

  $$
  w_{i+1} := w_i + v_{i+1}
  $$

  where $i$ is the iteration index, $v$ is the momentum variable, $\epsilon$ is the learning rate, and $\left\langle \frac{\partial L}{\partial w_i} \right\rangle_{D_i}$ is the average over the $i$th batch $D_i$ of the derivative of the objective with respect to $w$, evaluated at $w_i$.

- Initialized the learning rate with 0.01

Karen Simonyan and Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition
Learning & Weight Update – (2)

- Learning rate decreased by a factor of 10 when the validation set accuracy stopped improving.
- In total, the learning rate was decreased 3 times, and the learning was stopped after 370K iterations (74 epochs).
- Random initialization based on a normal distribution with the zero mean and 0.01 variance for net A. Initialized deeper models with the weights from A.
- Less epochs required to converge compared to (Krizhevsky et al., 2012)

The Update rule for weight W:

\[ v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w} |_{w_i} \right\rangle_{D_i} \]

\[ w_{i+1} := w_i + v_{i+1} \]

where \( i \) is the iteration index, \( v \) is the momentum variable, \( \epsilon \) is the learning rate, and \( \left\langle \frac{\partial L}{\partial w} |_{w_i} \right\rangle_{D_i} \) is the average over the \( i \)th batch \( D_i \) of the derivative of the objective with respect to \( w \), evaluated at \( w_i \).

Karen Simonyan and Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition
Training Image size

- **Image size**
  - $S = \text{smallest side of training image}$
  - Original image: 224 x 224
  - If $S = 224$, use an original image
  - If $S > 224$, use the part of the original image

- **Training images**
  - Single-scale training
    - Fix scales on every image: 256, 384
  - Multi-scale training
    - Scales vary on images: $[S_{\text{min}}, S_{\text{max}}]$ ($S_{\text{min}} = 256, S_{\text{max}} = 512$)

*Karen Simonyan and Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition*
Testing Image size

- **Image size**
  - Q = smallest side of testing image
  - Original image: 224 x 224
  - If Q = 224, use an original image
  - If Q > 224, use the part of the original image

- **Test images**
  - Q is not necessarily equal to the training scale S
  - Several values of Q for each S improve the performance

- **Dense evaluation vs Multi-crop evaluation**
  - Dense: an uncropped image as input
  - Multi-crop: multiple crops of the test image as input
Implementation Details

- Derived from C++ Caffe toolbox (Jia, 2013)
  - Multi-GPUs on a single system
  - Full-size (uncropped) images at multiple scales

- Data parallelism
  - Split each batch of training images into several GPU batches
  - GPU batch gradients are computed & averaged to obtain the gradient of the full batch
  - Synchronous across the GPUs

- Four NVIDIA Titan Black GPUs, 2-3 weeks to train
The Dataset

- **ILSVRC-2012 dataset**
  - Images of 1,000 classes
  - 1.3 million training images, 50,000 validation images, 100,000 testing images
  - Used the validation set as the test set for the majority of experiments

- **Report error rates**
  - Top-1 & Top-5
Single Scale Evaluation

- **Single Test Scale**
  - $Q = S$ for fixed $S$
  - $Q = 0.5(S_{\text{min}} + S_{\text{max}})$ for jittered $S \in [S_{\text{min}}, S_{\text{max}}]$

<table>
<thead>
<tr>
<th>ConvNet config. (Table 1)</th>
<th>smallest image side</th>
<th>top-1 val. error (%)</th>
<th>top-5 val. error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>train ($S$)</td>
<td>test ($Q$)</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>256</td>
<td>256</td>
<td>29.6</td>
</tr>
<tr>
<td>A-LRN</td>
<td>256</td>
<td>256</td>
<td>29.7</td>
</tr>
<tr>
<td>B</td>
<td>256</td>
<td>256</td>
<td>28.7</td>
</tr>
<tr>
<td>C</td>
<td>256</td>
<td>256</td>
<td>28.1</td>
</tr>
<tr>
<td></td>
<td>384</td>
<td>384</td>
<td>28.1</td>
</tr>
<tr>
<td></td>
<td>[256;512]</td>
<td>384</td>
<td>27.3</td>
</tr>
<tr>
<td>D</td>
<td>256</td>
<td>256</td>
<td>27.0</td>
</tr>
<tr>
<td></td>
<td>384</td>
<td>384</td>
<td>26.8</td>
</tr>
<tr>
<td></td>
<td>[256;512]</td>
<td>384</td>
<td>25.6</td>
</tr>
<tr>
<td>E</td>
<td>256</td>
<td>256</td>
<td>27.3</td>
</tr>
<tr>
<td></td>
<td>384</td>
<td>384</td>
<td>26.9</td>
</tr>
<tr>
<td></td>
<td>[256;512]</td>
<td>384</td>
<td><strong>25.5</strong></td>
</tr>
</tbody>
</table>

*Table 3: ConvNet performance at a single test scale.*

*Karen Simonyan and Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition*
Multi-Scale Evaluation

- **Multi-Test Scale**
  - $Q = \{S - 32, S, S + 32\}$ for fixed $S$
  - $Q = \{S_{min}, 0.5(S_{min} + S_{max}), S_{max}\}$ for jittered $S \in [S_{min}, S_{max}]$

**Table 4: ConvNet performance at multiple test scales.**

<table>
<thead>
<tr>
<th>ConvNet config. (Table II)</th>
<th>smallest image side</th>
<th>top-1 val. error (%)</th>
<th>top-5 val. error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>train ($S$)</td>
<td>test ($Q$)</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>256</td>
<td>224,256,288</td>
<td>28.2</td>
</tr>
<tr>
<td>C</td>
<td>256</td>
<td>224,256,288</td>
<td>27.7</td>
</tr>
<tr>
<td></td>
<td>384</td>
<td>352,384,416</td>
<td>27.8</td>
</tr>
<tr>
<td></td>
<td>[256; 512]</td>
<td>256,384,512</td>
<td>26.3</td>
</tr>
<tr>
<td>D</td>
<td>256</td>
<td>224,256,288</td>
<td>26.6</td>
</tr>
<tr>
<td></td>
<td>384</td>
<td>352,384,416</td>
<td>26.5</td>
</tr>
<tr>
<td></td>
<td>[256; 512]</td>
<td>256,384,512</td>
<td><strong>24.8</strong></td>
</tr>
<tr>
<td>E</td>
<td>256</td>
<td>224,256,288</td>
<td>26.9</td>
</tr>
<tr>
<td></td>
<td>384</td>
<td>352,384,416</td>
<td>26.7</td>
</tr>
<tr>
<td></td>
<td>[256; 512]</td>
<td>256,384,512</td>
<td><strong>24.8</strong></td>
</tr>
</tbody>
</table>

Karen Simonyan and Andrew Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*
Dense VS. Multi-Crop Evaluation

- Multiple crops performs slightly better than dense evaluation
- Two approaches are complementary

Table 5: ConvNet evaluation techniques comparison. In all experiments the training scale $S$ was sampled from $[256; 512]$, and three test scales $Q$ were considered: $\{256, 384, 512\}$.

<table>
<thead>
<tr>
<th>ConvNet config. (Table II)</th>
<th>Evaluation method</th>
<th>top-1 val. error (%)</th>
<th>top-5 val. error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>dense</td>
<td>24.8</td>
<td>7.5</td>
</tr>
<tr>
<td></td>
<td>multi-crop</td>
<td>24.6</td>
<td>7.5</td>
</tr>
<tr>
<td></td>
<td>multi-crop &amp; dense</td>
<td>24.4</td>
<td>7.2</td>
</tr>
<tr>
<td>E</td>
<td>dense</td>
<td>24.8</td>
<td>7.5</td>
</tr>
<tr>
<td></td>
<td>multi-crop</td>
<td>24.6</td>
<td>7.4</td>
</tr>
<tr>
<td></td>
<td>multi-crop &amp; dense</td>
<td>24.4</td>
<td>7.1</td>
</tr>
</tbody>
</table>

Karen Simonyan and Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition
ConvNet Fusion

- Combined the outputs of several models by averaging their soft-max class posteriors
- Improves the performance due to complementarity of the models

### Table 6: Multiple ConvNet fusion results.

<table>
<thead>
<tr>
<th>Combined ConvNet models</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>top-1</td>
</tr>
<tr>
<td>ILSVRC submission</td>
<td></td>
</tr>
<tr>
<td>(D/256/224,256,288), (D/384/352,384,416), (D/[256;512]/256,384,512)</td>
<td></td>
</tr>
<tr>
<td>(C/256/224,256,288), (C/384/352,384,416)</td>
<td></td>
</tr>
<tr>
<td>(E/256/224,256,288), (E/384/352,384,416)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24.7</td>
</tr>
<tr>
<td></td>
<td>7.5</td>
</tr>
<tr>
<td></td>
<td>7.3</td>
</tr>
<tr>
<td>post-submission</td>
<td></td>
</tr>
<tr>
<td>(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), dense eval.</td>
<td></td>
</tr>
<tr>
<td>(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop</td>
<td></td>
</tr>
<tr>
<td>(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop &amp; dense eval.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24.0</td>
</tr>
<tr>
<td></td>
<td>7.1</td>
</tr>
<tr>
<td></td>
<td>7.0</td>
</tr>
<tr>
<td></td>
<td>23.9</td>
</tr>
<tr>
<td></td>
<td>7.2</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>23.7</td>
</tr>
<tr>
<td></td>
<td>6.8</td>
</tr>
<tr>
<td></td>
<td>6.8</td>
</tr>
</tbody>
</table>

*Karen Simonyan and Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition*
Comparison with the state of the art

- Outperformed other neural networks in ILSVRC classification challenges in 2012 & 2013
- Achieved the best results!

Table 7: Comparison with the state of the art in ILSVRC classification. Our method is denoted as “VGG”. Only the results obtained without outside training data are reported.

<table>
<thead>
<tr>
<th>Method</th>
<th>top-1 val. error (%)</th>
<th>top-5 val. error (%)</th>
<th>top-5 test error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG (2 nets, multi-crop &amp; dense eval.)</td>
<td>23.7</td>
<td>6.8</td>
<td>6.8</td>
</tr>
<tr>
<td>VGG (1 net, multi-crop &amp; dense eval.)</td>
<td>24.4</td>
<td>7.1</td>
<td>7.0</td>
</tr>
<tr>
<td>VGG (ILSVRC submission, 7 nets, dense eval.)</td>
<td>24.7</td>
<td>7.5</td>
<td>7.3</td>
</tr>
<tr>
<td>GoogLeNet (Szegedy et al., 2014) (1 net)</td>
<td>-</td>
<td>7.9</td>
<td></td>
</tr>
<tr>
<td>GoogLeNet (Szegedy et al., 2014) (7 nets)</td>
<td>-</td>
<td>6.7</td>
<td></td>
</tr>
<tr>
<td>MSRA (He et al., 2014) (11 nets)</td>
<td>-</td>
<td>-</td>
<td>8.1</td>
</tr>
<tr>
<td>MSRA (He et al., 2014) (1 net)</td>
<td>27.9</td>
<td>9.1</td>
<td>9.1</td>
</tr>
<tr>
<td>Clarifai (Russakovsky et al., 2014) (multiple nets)</td>
<td>-</td>
<td>-</td>
<td>11.7</td>
</tr>
<tr>
<td>Clarifai (Russakovsky et al., 2014) (1 net)</td>
<td>-</td>
<td>-</td>
<td>12.5</td>
</tr>
<tr>
<td>Zeiler &amp; Fergus (Zeiler &amp; Fergus, 2013) (6 nets)</td>
<td>36.0</td>
<td>14.7</td>
<td>14.8</td>
</tr>
<tr>
<td>Zeiler &amp; Fergus (Zeiler &amp; Fergus, 2013) (1 net)</td>
<td>37.5</td>
<td>16.0</td>
<td>16.1</td>
</tr>
<tr>
<td>OverFeat (Sermanet et al., 2014) (7 nets)</td>
<td>34.0</td>
<td>13.2</td>
<td>13.6</td>
</tr>
<tr>
<td>OverFeat (Sermanet et al., 2014) (1 net)</td>
<td>35.7</td>
<td>14.2</td>
<td>-</td>
</tr>
<tr>
<td>Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)</td>
<td>38.1</td>
<td>16.4</td>
<td>16.4</td>
</tr>
<tr>
<td>Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)</td>
<td>40.7</td>
<td>18.2</td>
<td>-</td>
</tr>
</tbody>
</table>

Karen Simonyan and Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition
Conclusion

- Very Deep convolutional neural network (up to 19 weight layers)
- Increasing depth -> Improving the performance
- State-of-the-art performance on the ImageNet challenge dataset can be achieved using a conventional ConvNet architecture (LeCun et al., 1989; Krizhevsky et al., 2012) with substantially increased depth
References

- http://www.image-net.org/challenges/LSVRC
- https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2
- https://slidesplayer.com/slide/11579438/
- https://pprivulet.gitbooks.io/use-deep-learning-to-solve-real-life-problems/content/chapter2.html
- https://shafeentejani.github.io/2016-12-20/convolutional-neural-nets/