# HD-CNN: Hierarchical Deep Convolutional Neural Networks for Large Scale Visual Recognition Supplementary Material

Zhicheng Yan[†], Hao Zhang[‡], Robinson Piramuthu[*], Vignesh Jagadeesh[*],
Dennis DeCoste[*], Wei Di[*], Yizhou Yu[⋆†]
[†]University of Illinois at Urbana-Champaign, [‡]Carnegie Mellon University
[*]eBay Research Lab, [⋆]The University of Hong Kong

## 1. CIFAR100 Dataset

### 1.1. HD-CNN Based on CIFAR100-NIN net

The instance of HD-CNN we use for CIFAR100 dataset is built upon a building block net CIFAR100-NIN. The layer configurations in CIFAR100-NIN are listed in Table 1. The architectures of both CIFAR100-NIN and the corresponding HD-CNN are illustrated in Figure 1. We use the preceding layers from *conv1* to *pool1* as shared layers.

## 2. ImageNet 1000-class Dataset

We experiment with two different building block nets on ImageNet dataset, namely ImageNet-NIN and ImageNet-VGG-16-layer.

### 2.1. HD-CNN based on ImageNet-NIN

The layer configurations of the building block net ImageNet-NIN are listed in Table 2. The architectures of ImageNet-NIN and the corresponding HD-CNN are shown in Figure 2. The preceding layers from *conv1* to *pool3* are shared in HD-CNN.

#### 2.1.1 Category Hierarchy

We learn 89 overlapping coarse categories using the building block net ImageNet-NIN on ImageNet dataset. They are visualized in Figure 3. Fine categories within the same coarse category are more visually similar to each other than those absent in the coarse category. A histogram of the fine category occurrences in the coarse categories is shown in Figure 4. Each fine category can appear in more than one coarse category.

#### 2.1.2 More Case Studies

To better demonstrate HD-CNN can correctly classify the difficult images for which the building block net fails, we include more case studies in Figure 5. For difficult cases, HD-CNN relies on the fine predictions from more than one fine category classifier to make the correct final prediction.

### 2.2. HD-CNN based on ImageNet-VGG-16-layer

The layer configurations of the building block net ImageNet-VGG-16-layer are listed in Table 3. The architectures of ImageNet-VGG-16-layer and the corresponding HD-CNN are shown in Figure 6. Layers from *conv1_1* to *pool4* are shared within HD-CNN.

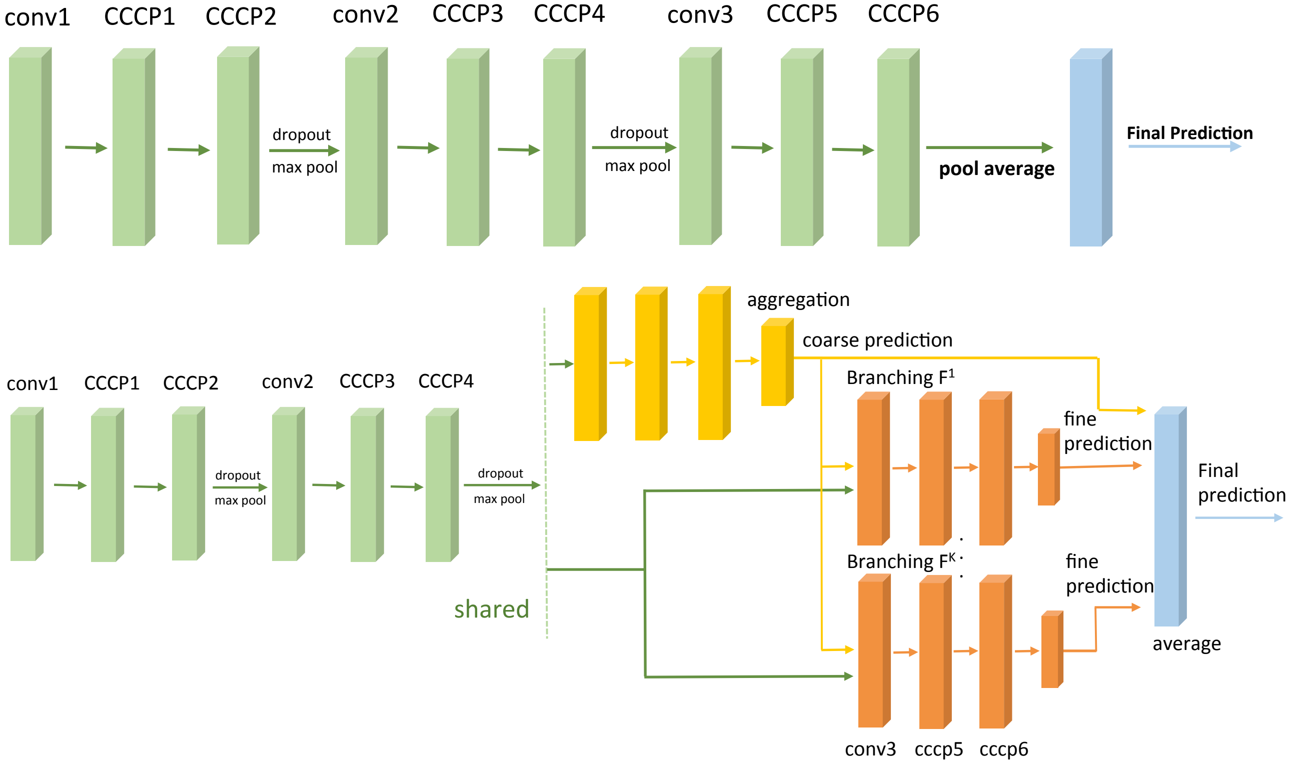

Figure 1: **Top**: CIFAR100-NIN network. **Bottom**: HD-CNN network using CIFAR100-NIN building block.

| **LAY** | conv1 | cccp1 | cccp2 | pool1 | conv2 | cccp3 | cccp4 | pool2 | conv3 | cccp5 | cccp6 | pool3 | prob |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CFG** | 192,5,5 | 160,1,1 | 96,1,1 | 3,3,2 MAX | 192, 5,5 | 192,1,1 | 192,1,1 | 3,3,2 MAX | 192,3,3 | 192,1,1 | 100,1,1 | 6,6,1 AVG | SMAX |
| **ACT** | | ReLU | ReLU | | | ReLU | ReLU | | | ReLU | ReLU | | |
| **PAR #** | 1.4e+4 | 3.1e+4 | 1.5e+4 | | 4.6e+5 | 3.7e+4 | 3.7e+4 | | 3.3e+5 | 3.7e+4 | 1.9e+4 | | |
| **PAR %** | 1.5 | 3.1 | 1.6 | | 46.9 | 3.8 | 3.8 | | 33.8 | 3.8 | 2.0 | | |
| **FLOP #** | 9.7e+6 | 2.1e+7 | 1e+7 | | 7.8e+7 | 6.2e+6 | 6.2e+6 | | 1.2e+7 | 1.3e+6 | 6.9e+5 | | |
| **FLOP %** | 6.7 | 14.3 | 7.2 | | 53.6 | 4.3 | 4.3 | | 8.2 | 0.9 | 0.5 | | |

Table 1: CIFAR100-NIN network. The configuration of convolutional layer is denoted as (filter number, filter height, filter width). The configuration of pooling layer is denoted as (pooling height,pooling width, stride). Notations: **LAY**=Layer. **CFG**=Configuration. **ACT**=Activation. **PAR #**=Parameter number. **PAR %**=Parameter percentage. **FLOP #**=FLoating-point OPerations. **FLOP %**=FLoating-point OPeration percentage. **SMAX**=SOFTMAX.

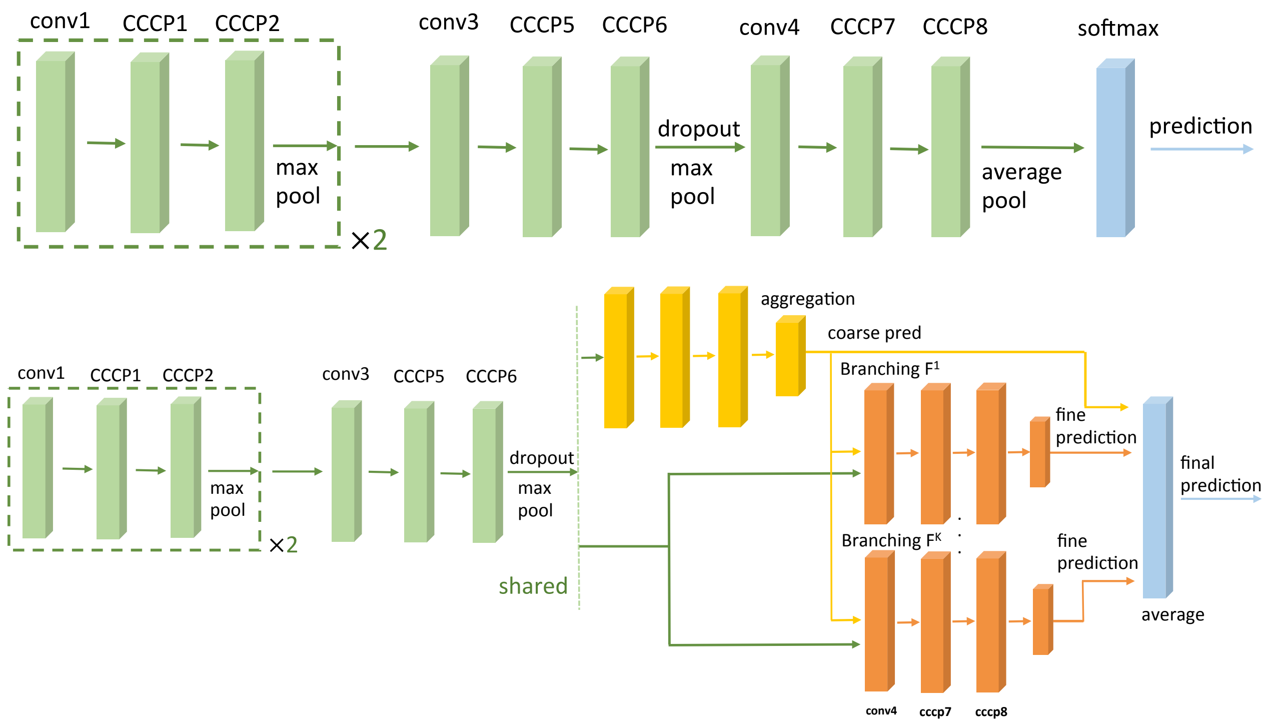

Figure 2: **Top**: ImageNet-NIN network. **Bottom**: HD-CNN network using ImageNet-NIN building block.

| **LAY** | conv1 | cccp1 | cccp2 | pool0 | conv2 | cccp3 | cccp4 | pool2 | conv3 | cccp5 | cccp6 | pool3 | conv4 | cccp7 | cccp8 | pool4 | prob |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CFG** | 96,11,11 | 96,1,1 | 96,1,1 | 3,3,2<br>MAX | 256,5,5 | 256,1,1 | 256,1,1 | 3,3,2<br>MAX | 384,3,3 | 384,1,1 | 384,1,1 | 3,3,2<br>MAX | 1024,3,3 | 1024,1,1 | 1000,1,1 | 6,6,1<br>AVG | SMAX |
| **ACT** | ReLU | ReLU | ReLU | | ReLU | ReLU | ReLU | | ReLU | ReLU | ReLU | | ReLU | ReLU | ReLU | | |
| **PAR #** | 3.5e+4 | 9.2e+3 | 9.2e+3 | | 6.1e+5 | 6.6e+4 | 6.6e+4 | | 8.9e+5 | 1.5e+5 | 1.5e+5 | | 3.5e+6 | 1.1e+6 | 1.1e+6 | | |
| **PAR %** | 0.5 | 0.1 | 0.1 | | 8.1 | 0.9 | 0.9 | | 11.7 | 1.9 | 1.9 | | 46.6 | 13.8 | 13.5 | | |
| **FLOP #** | 1e+8 | 2.7e+7 | 2.7e+7 | | 4.5e+8 | 4.8e+7 | 4.8e+7 | | 1.5e+8 | 2.5e+7 | 2.5e+7 | | 1.3e+8 | 3.8e+7 | 3.8e+7 | | |
| **FLOP %** | 9.2 | 2.4 | 2.4 | | 40.7 | 4.3 | 4.3 | | 13.6 | 2.3 | 2.3 | | 11.6 | 3.4 | 3.4 | | |

Table 2: ImageNet-NIN network.

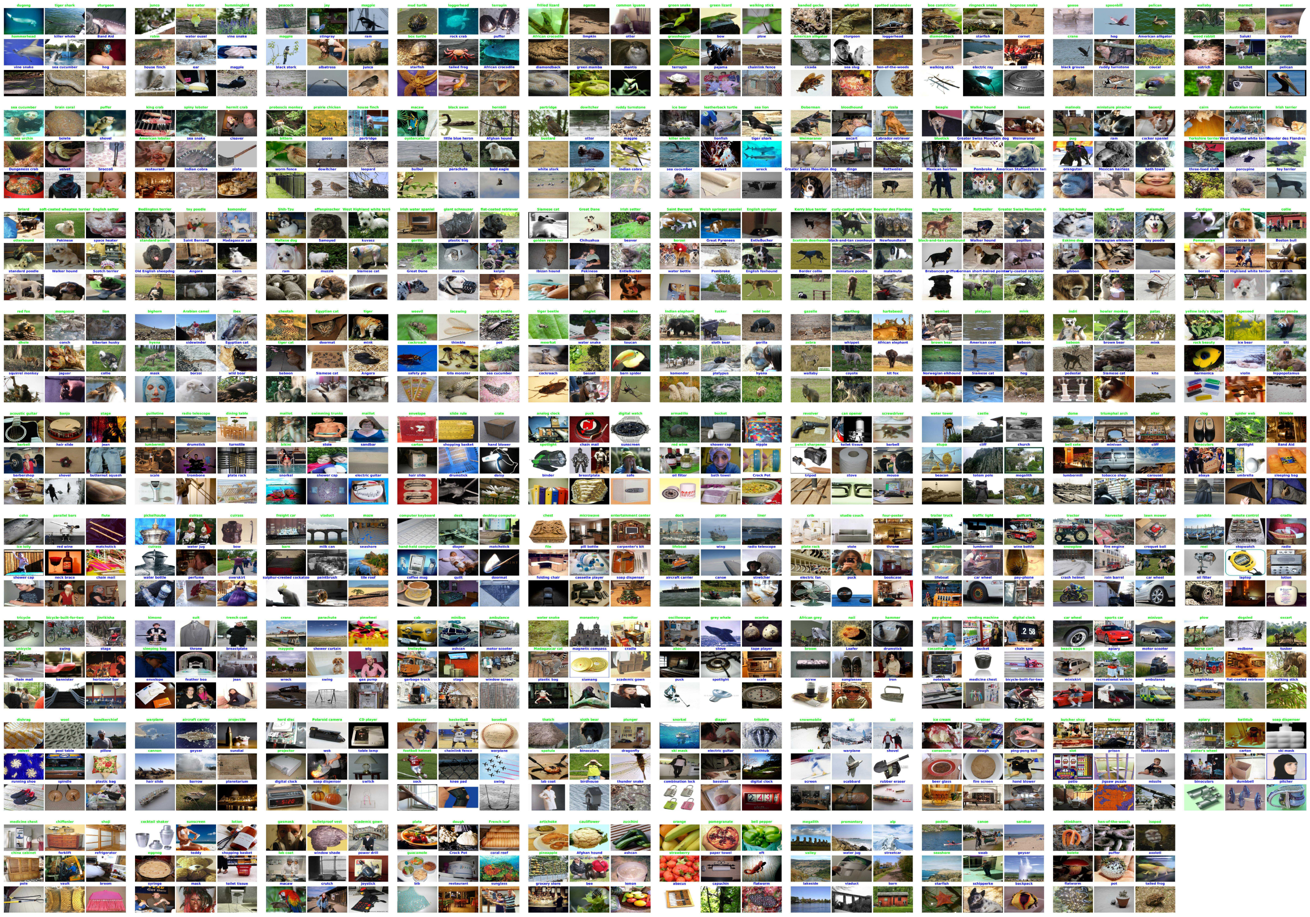

Figure 3: The learnt 89 overlapping coarse categories, each of which is represented by a grid of size $3 \times 3$. For each coarse category, we randomly choose 9 fine categories within it. An example image for each fine category is shown. Among the 9 fine categories, 4 of them are found by spectral clustering on the confusion matrix and their category labels are in green. The remaining 5 fine categories are added subsequently to remove the separability constraint between coarse categories. Their category labels are in blue.

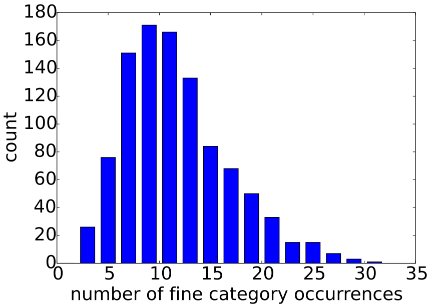

Figure 4: Histogram of fine category occurrences in 89 overlapping coarse categories. The category hierarchy is learnt using the building block net ImageNet-NIN.

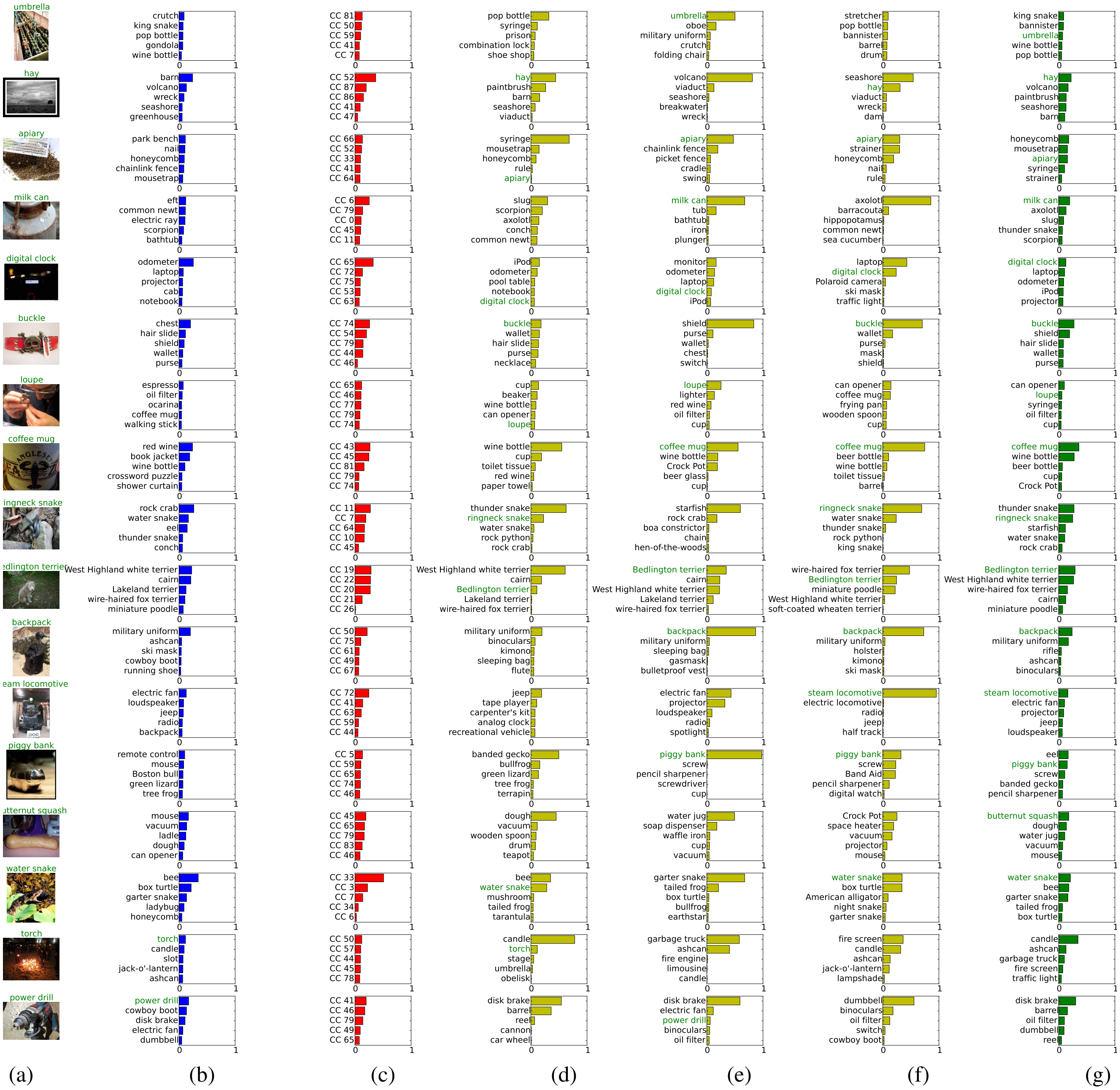

Figure 5: More case studies. **Column (a)**: test image with ground truth label. **Column (b)**: top 5 guesses from the building block net ImageNet-NIN. **Column (c)**: top 5 Coarse Category (**CC**) probabilities. **Column (d)-(f)**: top 5 guesses made by the top 3 fine category CNN components. **Column (g)**: final top 5 guesses made by the HD-CNN. All but the the last two are positive cases where HD-CNN predicts the ground truth label in the top 5 guesses while the building block net fails.

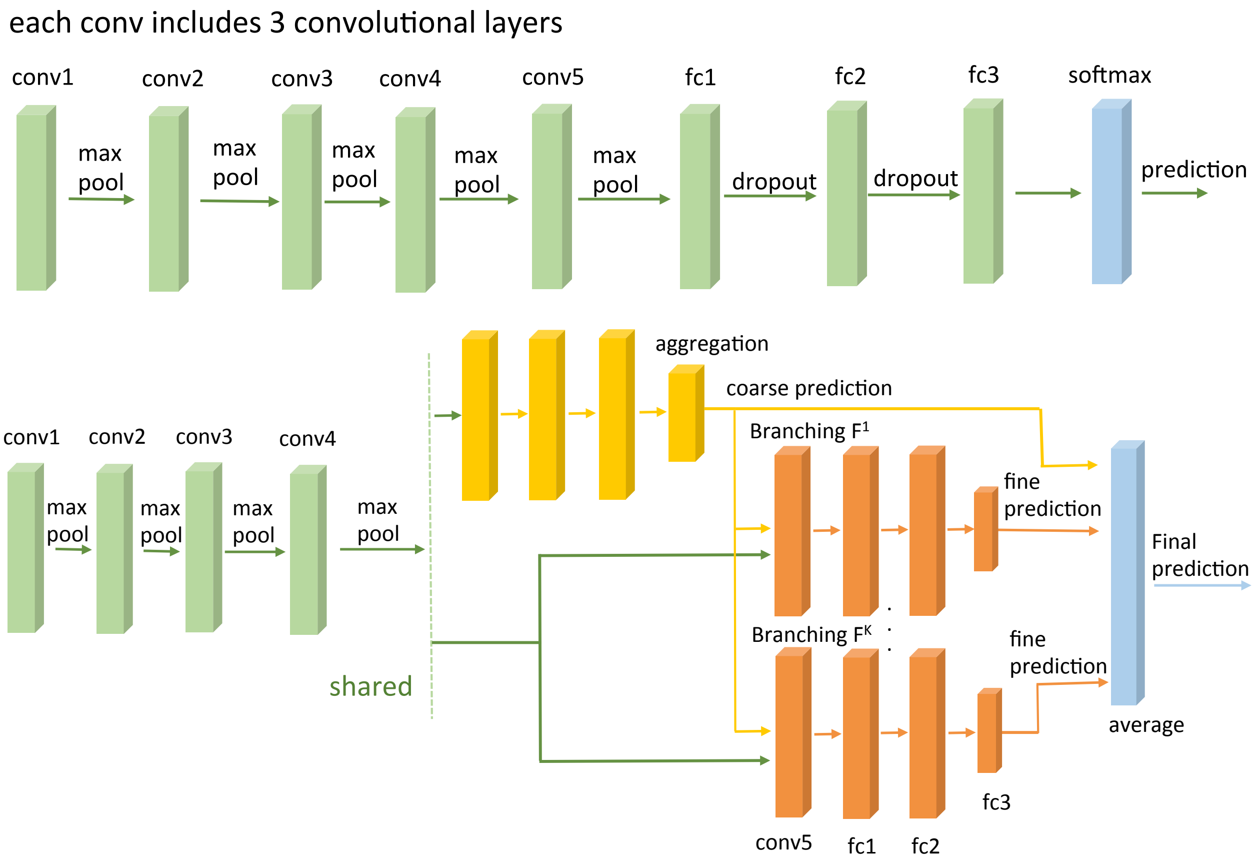

Figure 6: **Top**: ImageNet-VGG-16-layer network. **Bottom**: HD-CNN network using ImageNet-VGG-16-layer building block.

| **LAY** | conv 1_1 | conv 1_2 | pool1 | conv 2_1 | conv2_2 | pool2 | conv 3_1 | conv 3_{2,3} | pool3 | conv 4_1 | conv 4_{2,3} | pool4 | conv 5_{1,2,3} | pool5 | fc6 | fc7 | fc8 | prob |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CFG** | 64, 3,3 | 64, 3,3 | 2,2,2 MAX | 128, 3,3 | 128, 3,3 | 2,2,2 MAX | 256,3,3 | 256, 3,3 | 2,2,2 MAX | 512, 3,3 | 512, 3,3 | 2,2,2 MAX | 512,3,3 | 2,2,2 MAX | 4096 | 4096 | 1000 | SMAX |
| **ACT** | ReLU | ReLU | | ReLU | ReLU | | ReLU | ReLU | | ReLU | ReLU | | ReLU | | ReLU | ReLU | | |
| **PAR #** | 1.7e3 | 3.7e4 | | 7.4e4 | 1.5e5 | | 3.0e5 | 5.9e5 | | 1.2e6 | 2.4e6 | | 2.4e6 | | 1.0e8 | 1.7e7 | 4.1e6 | |
| **PAR %** | 0.01 | 0.03 | | 0.1 | 0.1 | | 0.2 | 0.4 | | 0.9 | 1.7 | | 1.7 | | 74.3 | 12.1 | 3.0 | |
| **FLOP #** | 8.7e7 | 1.9e9 | | 9.3e8 | 1.9e9 | | 9.3e8 | 1.9e9 | | 9.3e8 | 1.9e9 | | 4.6e8 | | 1.0e8 | 1.7e7 | 4.1e6 | |
| **FLOP %** | 0.6 | 12.0 | | 6.0 | 12.0 | | 6.0 | 12.0 | | 6.0 | 12.0 | | 3.0 | | 0.7 | 0.11 | 0.1 | |

Table 3: ImageNet-VGG-16-layer network. For clarity, adjacent layers with the same configuration are merged, such as layers *conv3_2* and *conv3_3*.