advanced convolutional neural networks: Q1 2019

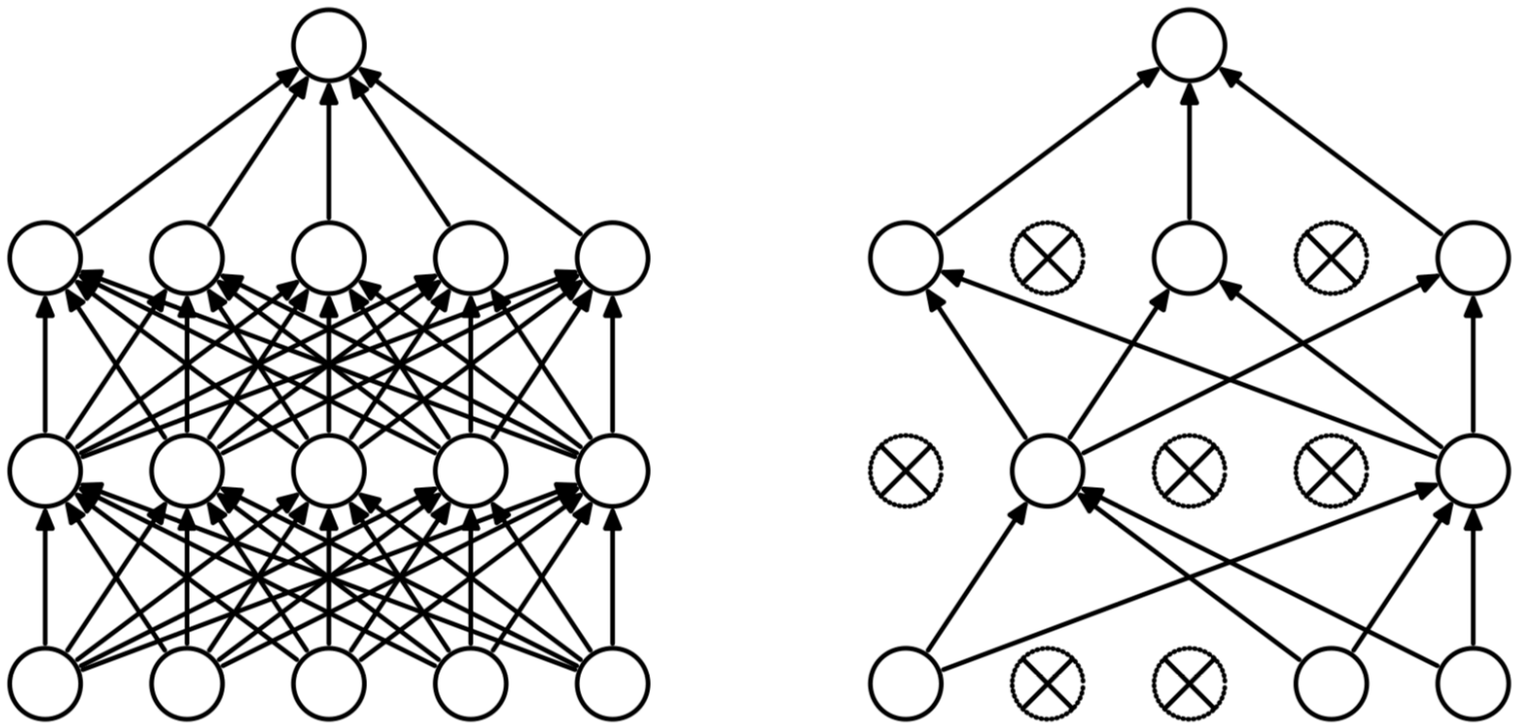
brettkoonce.com/talks

february 23rd, 2019

overview

- **training: dropout, shakeshake, cutout, mixup**
- **models: condensenet, sparsenet, enas, mnasnet**
- **3d: mnist demo, 3d-cnn, voxnet**

dropout



(a) Standard Neural Net

(b) After applying dropout.

Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.

shake-shake

1.3 Training procedure

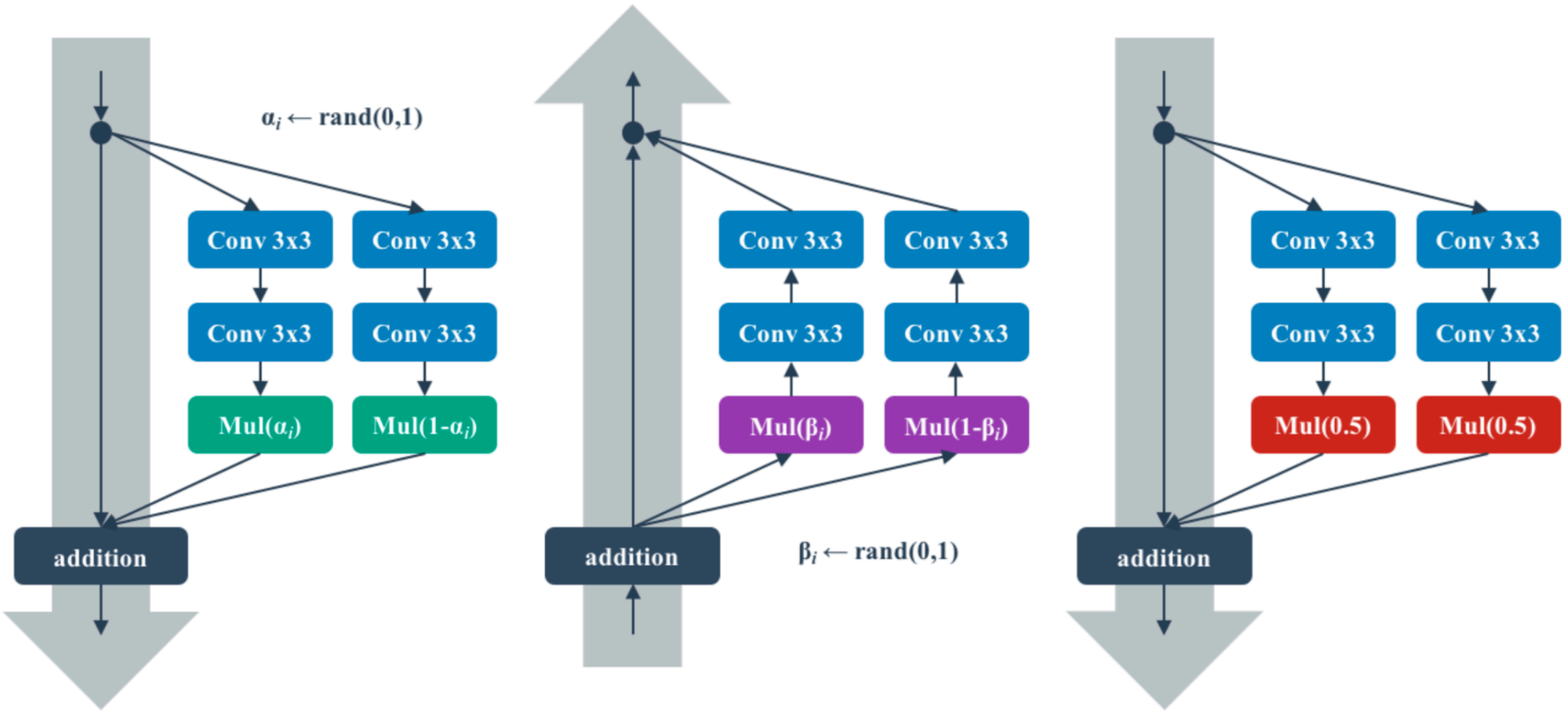


Figure 1: **Left:** Forward training pass. **Center:** Backward training pass. **Right:** At test time.

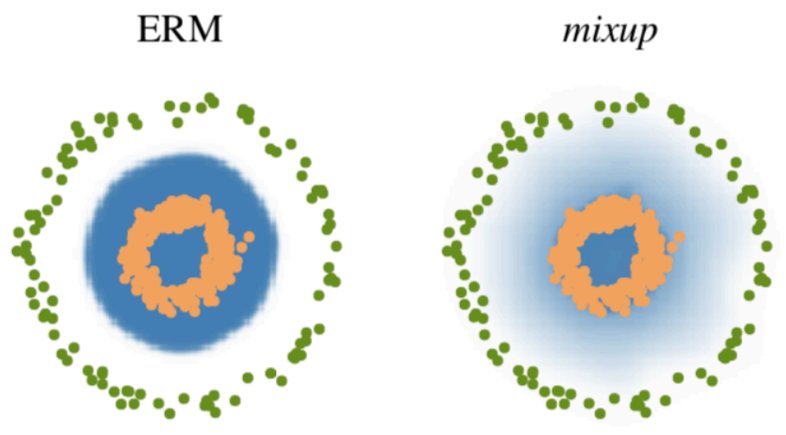
cutout

Figure 1: Cutout applied to images from the CIFAR-10 dataset.

mixup

```
# y1, y2 should be one-hot vectors
for (x1, y1), (x2, y2) in zip(loader1, loader2):
    lam = numpy.random.beta(alpha, alpha)
    x = Variable(lam * x1 + (1. - lam) * x2)
    y = Variable(lam * y1 + (1. - lam) * y2)
    optimizer.zero_grad()
    loss(net(x), y).backward()
    optimizer.step()
```

(a) One epoch of *mixup* training in PyTorch.



(b) Effect of *mixup* ($\alpha = 1$) on a toy problem. Green: Class 0. Orange: Class 1. Blue shading indicates $p(y = 1|x)$.

Figure 1: Illustration of *mixup*, which converges to ERM as $\alpha \to 0$.

condensenet

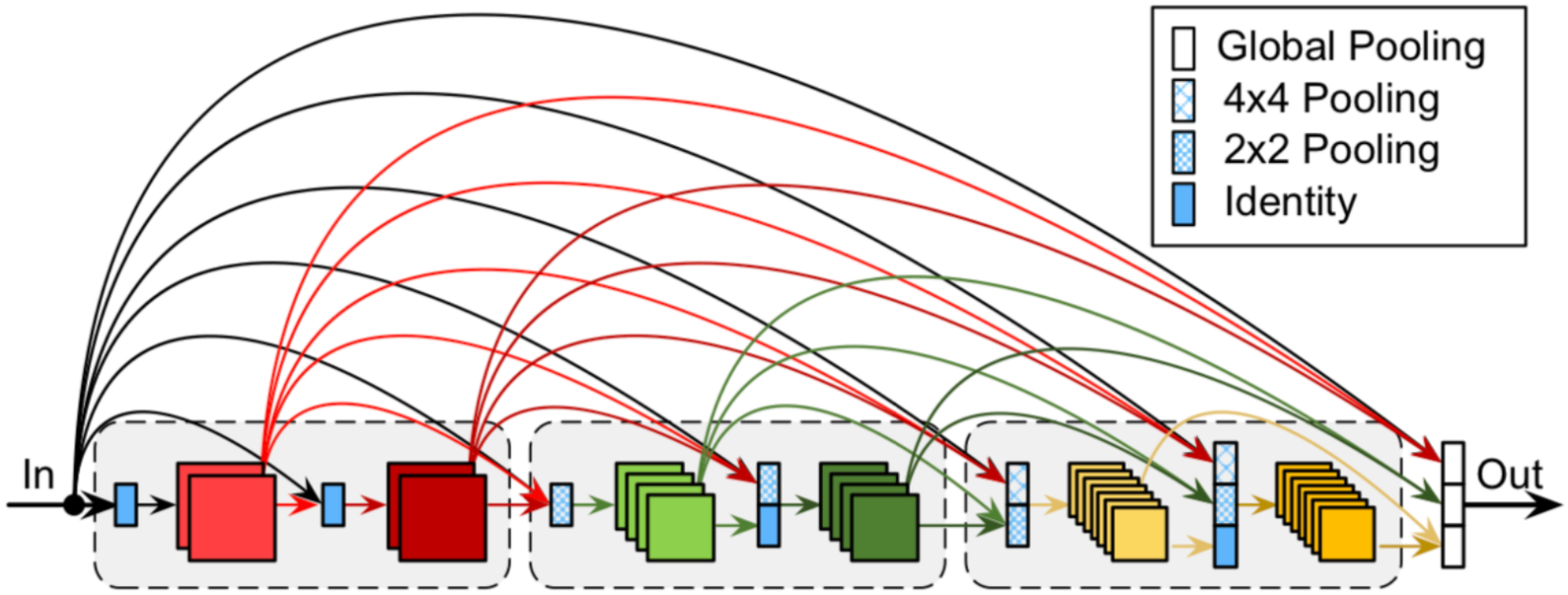


Figure 5. The proposed DenseNet variant. It differs from the original DenseNet in two ways: (1) layers with different resolution feature maps are also directly connected; (2) the growth rate doubles whenever the feature map size shrinks (far more features are generated in the third, yellow, dense block than in the first).

sparsenet

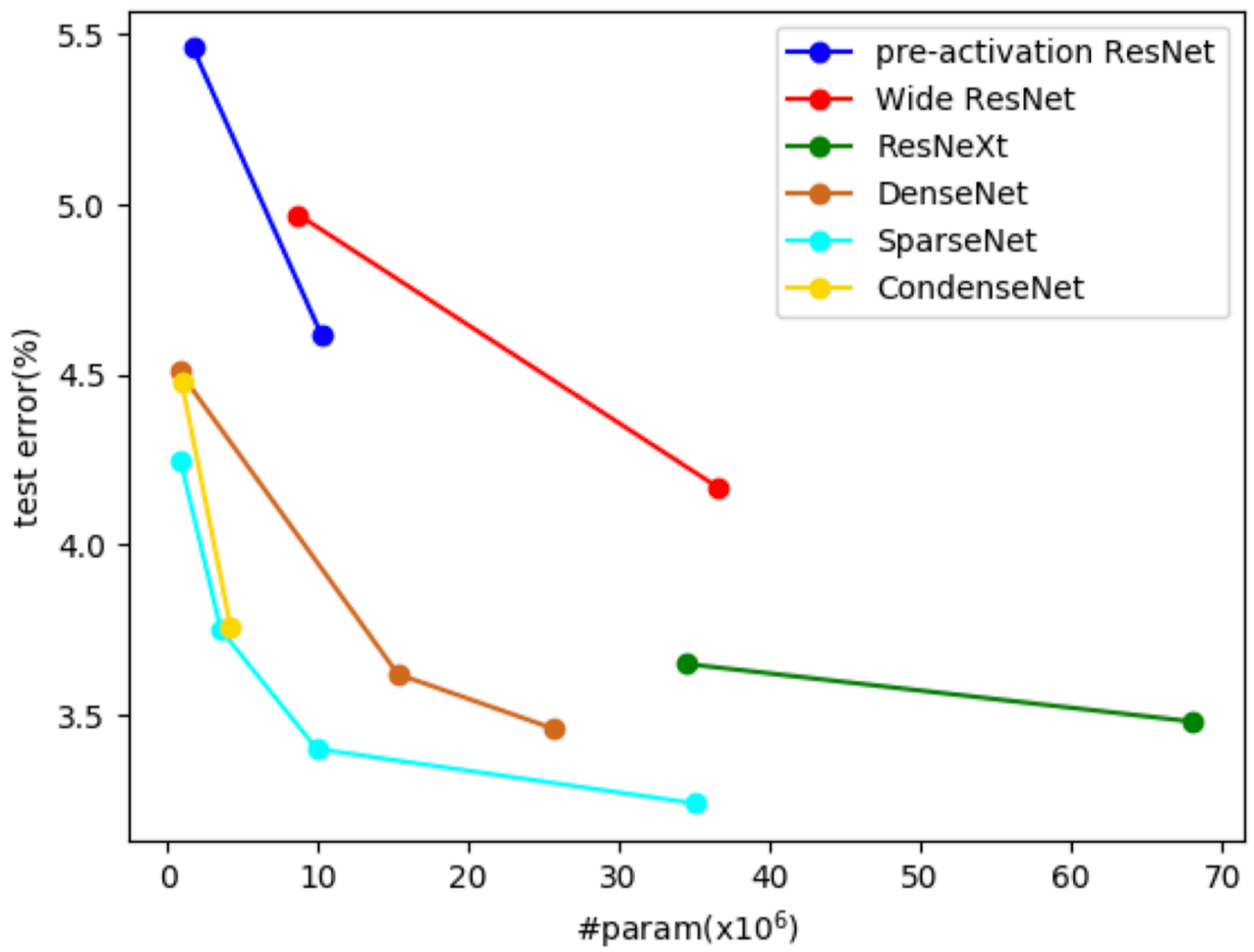


Fig. 8: Comparison parameter-efficiency on CIFAR10 of different models

enas

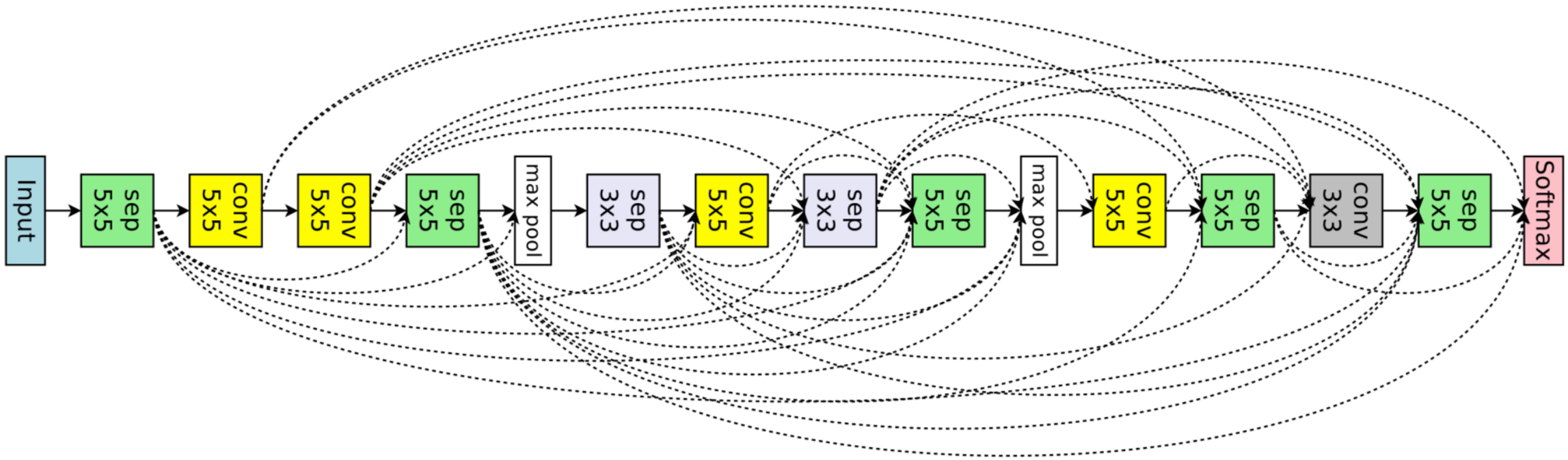
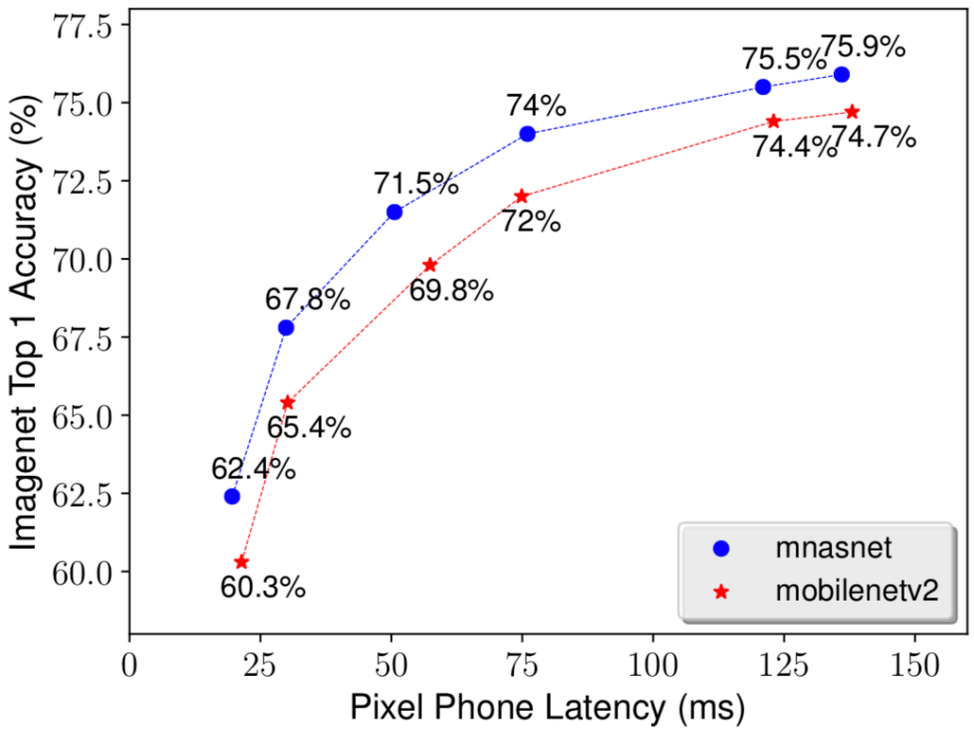


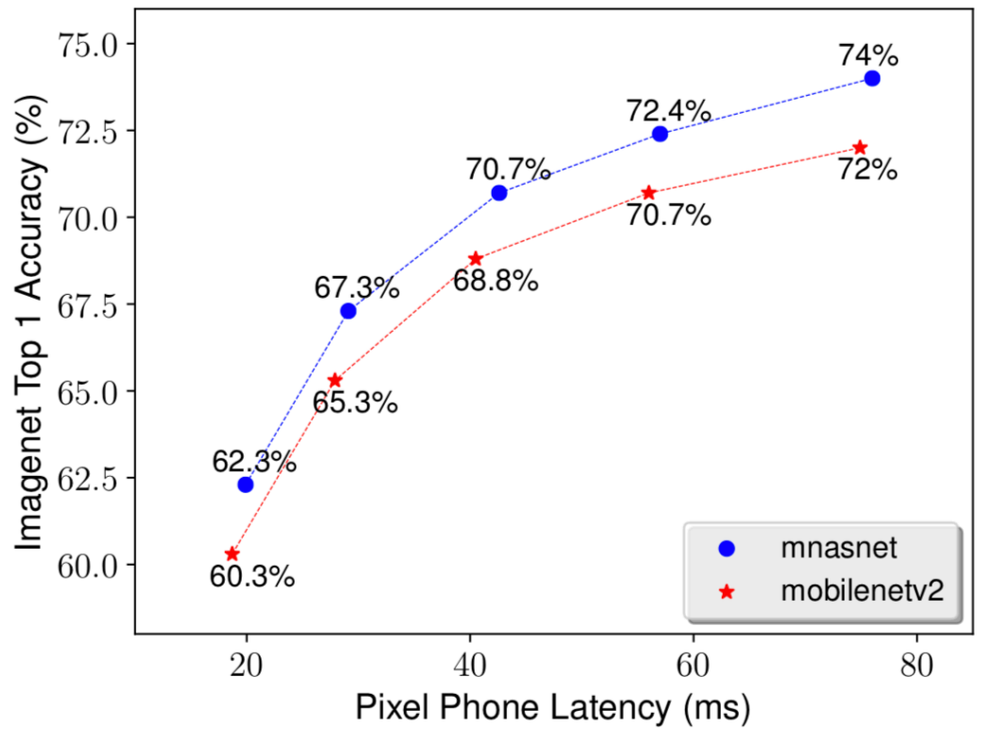
Figure 7. ENAS's discovered network from the macro search space for image classification.

- **lstm + nas search**

mnasnet



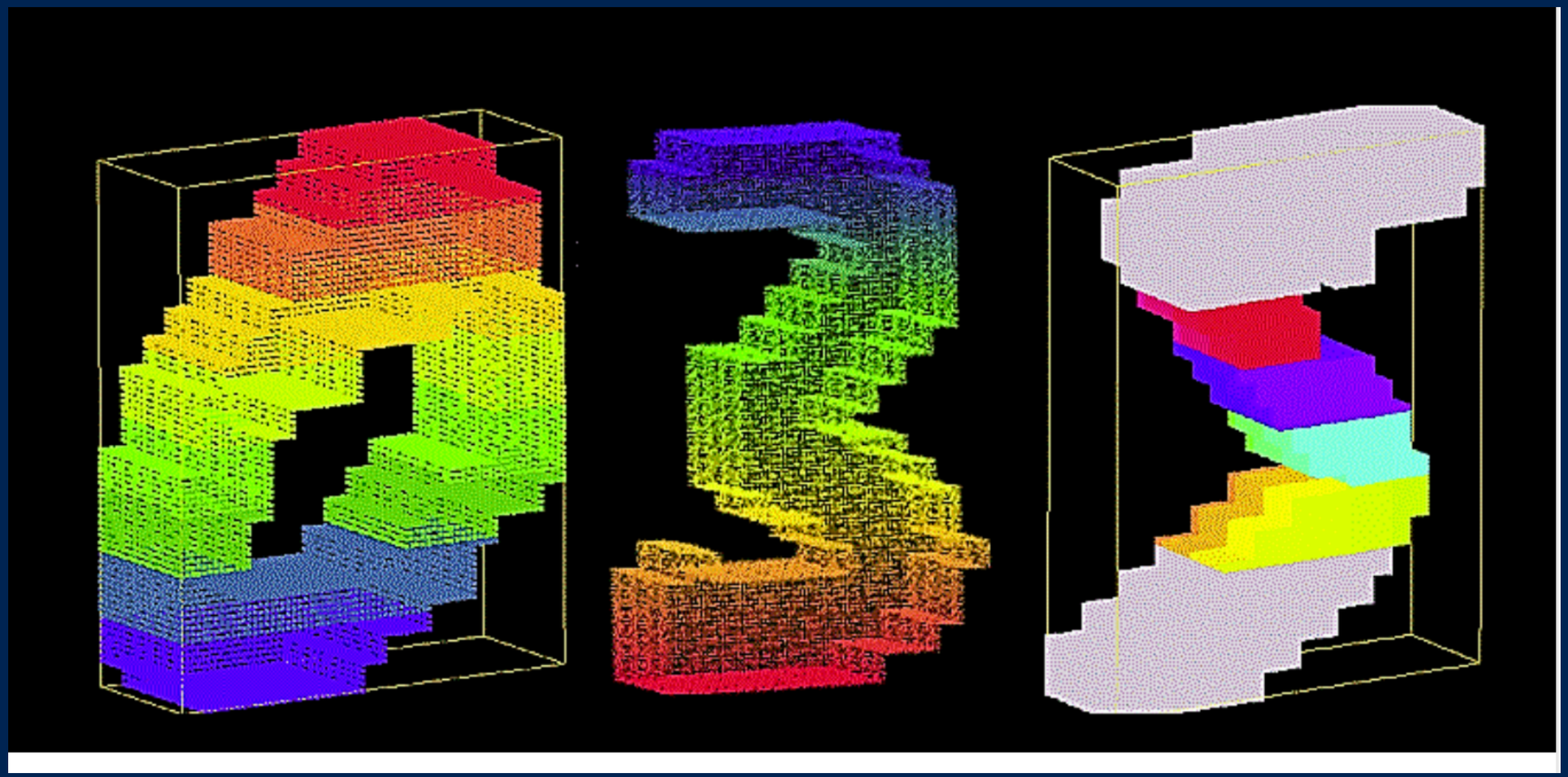
(a) Depth multiplier = 0.35, 0.5, 0.75, 1.0, 1.3, 1.4, corresponding to points from left to right.



(b) Input size = 96, 128, 160, 192, 224, corresponding to points from left to right.

Figure 4: **Performance Comparison with Different Model Scaling Techniques**. MnasNet is our baseline model shown in Table 1. We scale it with the same depth multipliers and input sizes as MobileNetV2.

3d mnist



- medium.com/shashwats-blog/3d-mnist-b922a3d07334

3d-cnn

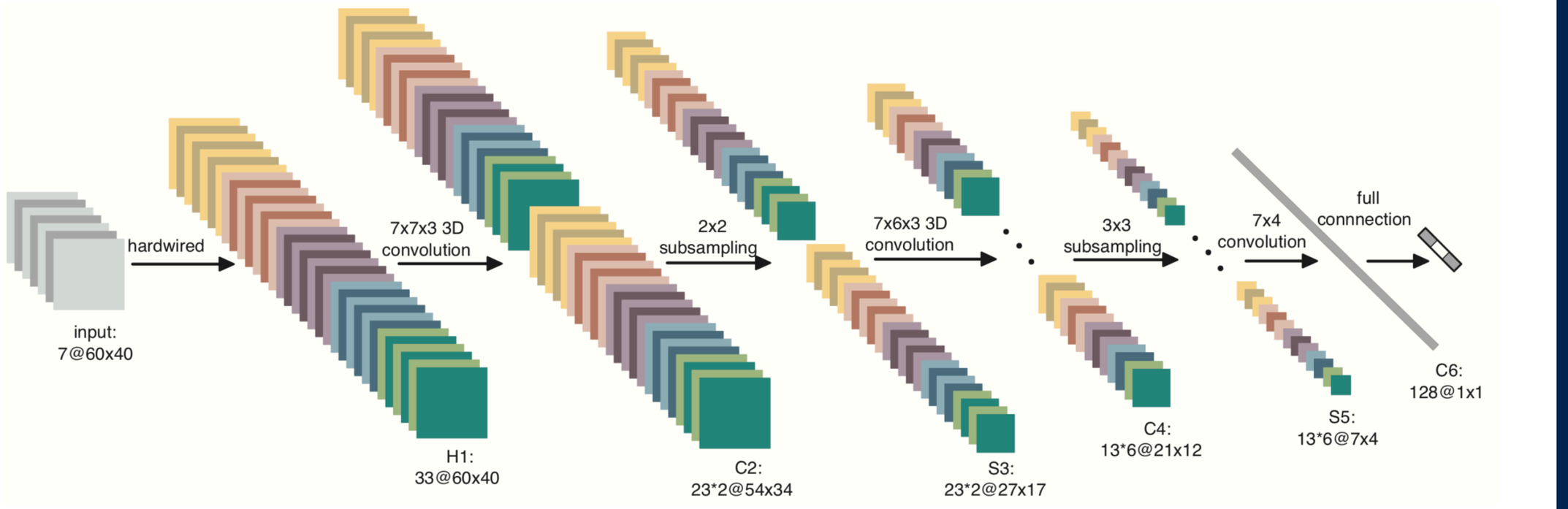


Figure 3. A 3D CNN architecture for human action recognition. This architecture consists of 1 hardwired layer, 3 convolution layers, 2 subsampling layers, and 1 full connection layer. Detailed descriptions are given in the text.

VOXNET

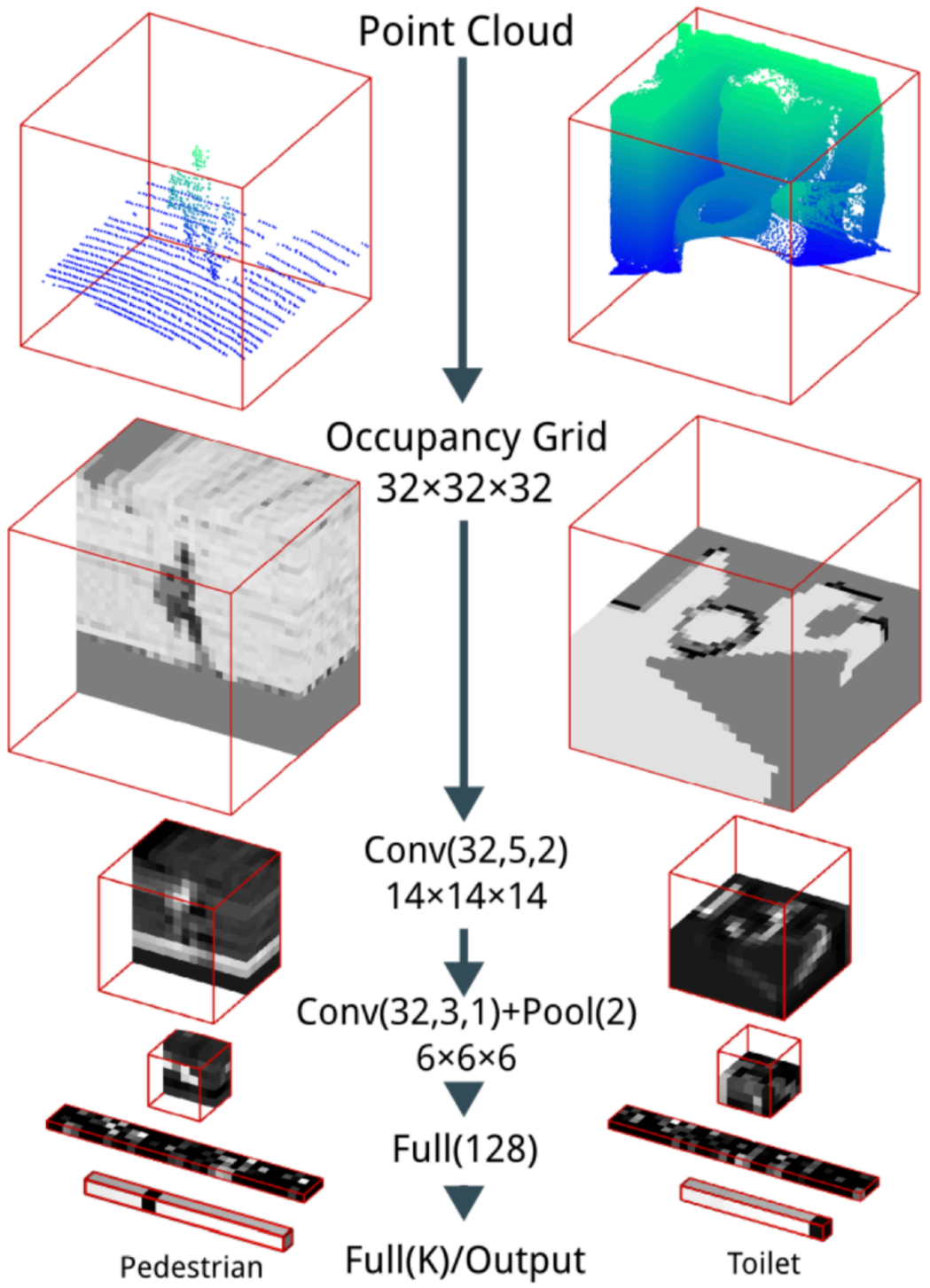


Fig. 1. The VoxNet Architecture. $Conv(f, d, s)$ indicates f filters of size d and at stride s, $Pool(m)$ indicates pooling with area m, and $Full(n)$ indicates fully connected layer with n outputs. We show inputs, example feature maps, and predicted outputs for two instances from our experiments. The point cloud on the left is from LiDAR and is part of the Sydney Urban Objects dataset [4]. The point cloud on the right is from RGBD and is part of NYUv2 [5]. We use cross sections for visualization purposes.