# Supplementary Material:

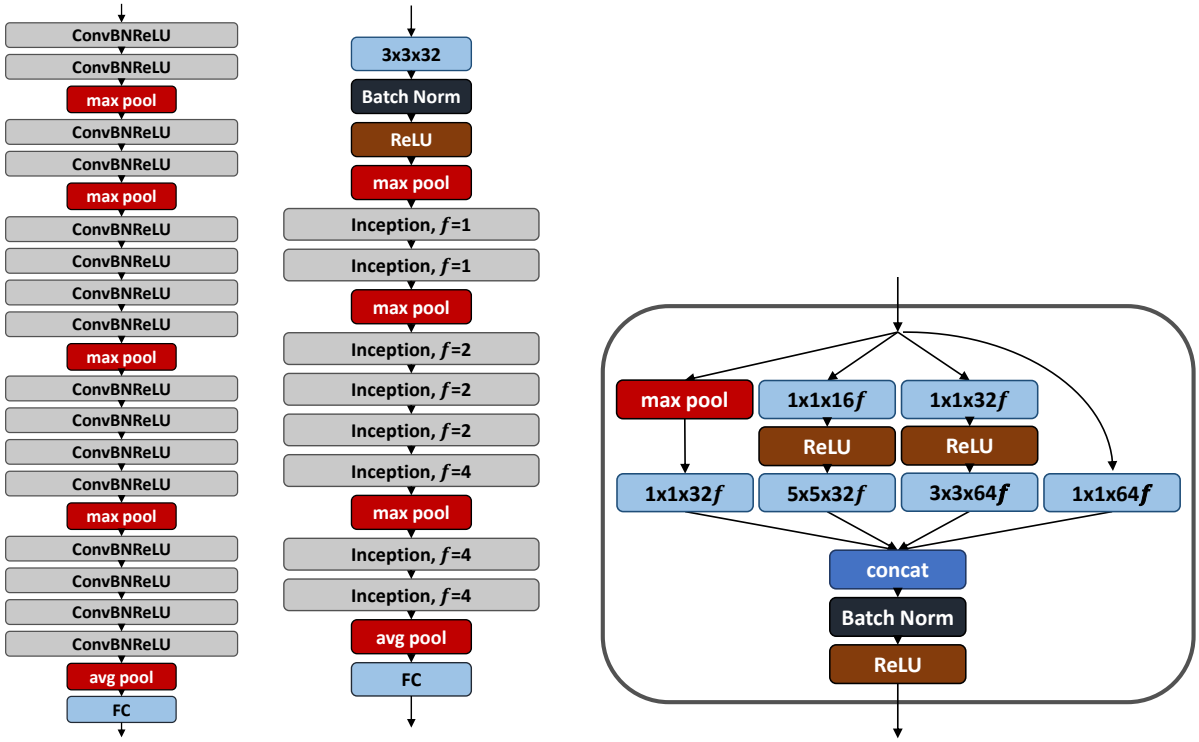
## Confident Multiple Choice Learning

## A. Experimental Setups for Image Classification

In this section, we describe detailed explanation about all the experiments described in Section 5.1.

**Detailed CNN structure and training.**

The CNN we use for evaluations in Table 1 is consist of two convolutional layers followed by one fully-connected layer. Convolutional layers have 128 and 256 filters respectively. Each convolutional layer has a $5 \times 5$ receptive field applied with a stride of 1 pixel. Each max pooling layer pools $2 \times 2$ regions at strides of 2 pixels. Dropout was applied to all layers in the network with drop probability $0.5$. Similar to (Zagoruyko & Komodakis, 2016), the softmax classifier is used, and each model is trained by minimizing the cross-entropy loss using SGD with Nesterov momentum. The initial learning rate is set to 0.01, weight decay to 0.0005, dampening to 0, momentum to 0.9 and minibatch size to 64. We drop the learning rate by 0.2 at 60, 120 and 160 epochs and we train for total 200 epochs. We report the mean of the test error rates produced by repeating each test 5 times.




(a) VGGNet-17 overview (b) GoogLeNet-18 overview (c) Structure of an inception module with width factor $f$

*Figure 5.* Detailed structure of large-scale CNNs used in Section 5.1.

**Detailed large-scale CNN models.**

In case of residual networks, we use ResNet-20 model suggested by the author, which has 19 $3 \times 3$ convolutional layers. We also follow the author's descriptions to train the model: minibatch size is set to 128, weight decay to 0.0001, momentum to 0.9, and initial learning rate to 0.1 and drop by 0.1 after 82 and 123 epochs with 164 epochs in total. Figure 5(a) shows the detailed structure of VGGNet-17 with one fully-connected layer and 16 convolutional layers. Each ConvBNReLU box in the figure indicates a $3 \times 3$ convolutional layer followed by batch normalization (Ioffe & Szegedy, 2015) and ReLU activation. Figure 5(b) shows the detailed structure of GoogLeNet-18 with one fully-connected layer and 8 inception

modules consist of 17 convolutional layers in total, where $1 \times 1$ convolutional layers are not considered as weighted layers. To simply increase the number of convolutional filters as layers stacked on, we introduce width factor $f$ which controls the overall size of an inception module as shown in Figure 5(c). For both VGGNet-17 and GoogLeNet-18, all convolutional layers have stride 1 and use padding to keep the feature map size equal. Also, all max pooling layers have $3 \times 3$ receptive fields with stride 1 and all average pooling layers indicate the global average pooling (Lin et al., 2014a). We use initial learning rate 0.1 and drop it by 0.2 at 25, 50 and 75 epochs with total 100 epochs for both networks. We use Nesterov momentum 0.9 for SGD, minibatch size is set to 128, and weight decay is set to 0.0005. We report the mean of the test error rates produced by repeating each test 5 times.

## B. Experimental Setups for Background-Foreground Segmentation

In this section, we describe detailed explanation about all the experiments described in Section 5.2. It consists of three convolutional layers followed by a fully convolutional layer. The convolutional layers have 128, 256 and 1 filters respectively. Each convolutional layer has a $4 \times 4$ receptive field applied with a stride of 2 pixel. For feature sharing, the 2-th activation of FCNs is used. The softmax classifier is used, and each model is trained by minimizing the cross-entropy loss using Adam learning rule (Kingma & Ba, 2015) with a mini-batch size of 20. The initial learning rate is chosen from $\{0.001, 0.0005, 0.0001\}$ and we used an exponentially decaying learning rate. We train every model for total 300 epochs. Similar to (Guzman-Rivera et al., 2012; Lee et al., 2016), we initialize the parameter of FCNs using that of FCNs trained by IE for 20 epochs in case of MCL and CMCL. The best test result is reported for each method.