Supplementary Materials

In this supplementary material, Sec. 6 provides the training details of the proposed DDI framework; Sec. 7 shows the structure of the gating network of channel skipping in DenseNet. Sec. 8 introduces the structure of branch classifiers of RADI; In Sec. 9 we provide additional ImageNet results of the proposed IADI and DDI framework; We show further visualization of feature maps under channel/ layer skipping, and the skipping ratio distributions of the two schemes at Sec. 10

6. Training Details of DDI

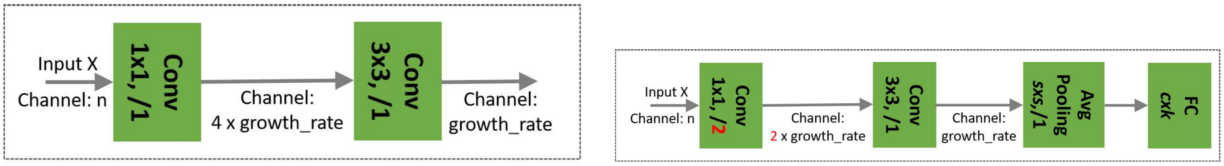
IADI Stage The training of DDI framework has two stages. For the first IADI, we train it from a pre-trained base CNN model. If we directly start the training with randomly initialized gate networks using Gaussian distribution, the skipping ratio will be around 50% from the beginning, which causes the mismatch between pre-trained batch normalization layers and unseen feature maps statistical distribution. To this end, we propose a warm-up process at the beginning to reduce the training instability caused by the mismatch. During the warm-up process, parameters in the base networks including batch normalization layers are fixed and only gate networks are trained to have zero skipping ratio. Then we use SGD as the optimizer to train the base network and gate networks end to end as described in the paper.

DDI Stage After the first stage is finished, we add intermediate branch classifiers to the well-trained IADI model (structure of the branch classifiers will be introduced in Sec. 8), then train the whole DDI framework end to end as described in 3.3, the hyperparameters used for this stage is the same for IADI training.

Hyperparameter Settings For both CIFAR-10 and ImageNet datasets, We set momentum to 0.9, weight-decay to 1e-4. For CIFAR-10 experiments, we set learning rate to 5e-2, batch size to 128, α to 2e-4, for the IADI stage, we train a total of 50k iterations, for DDI stage, we train another 50k iterations. For ImageNet dataset, we set initial learning rate to 5e-2, batch size to 512, α to 4e-6, and similarly, we train IADI and DDI in sequential order, each with 90 epochs.

7. Gating Design for DenseNet

As we discussed in Sec. 3.1.3, to reduce the prohibitive computational cost of gating network G^C in DenseNet, we design a light weight gating network. Specifically, we construct the first two layers of the gating network in similar structure to a denselayer in DenseNet. By adjusting the stride and output channel number of the first 1x1 convolution layer, we can control the FLOPs of the gating network, and we experimentally found out that when the FLOPs of the gating network are around 11 % of that of the denselayer, the channel skipping method in DenseNet achieves the best computation saving/accuracy tradeoff.

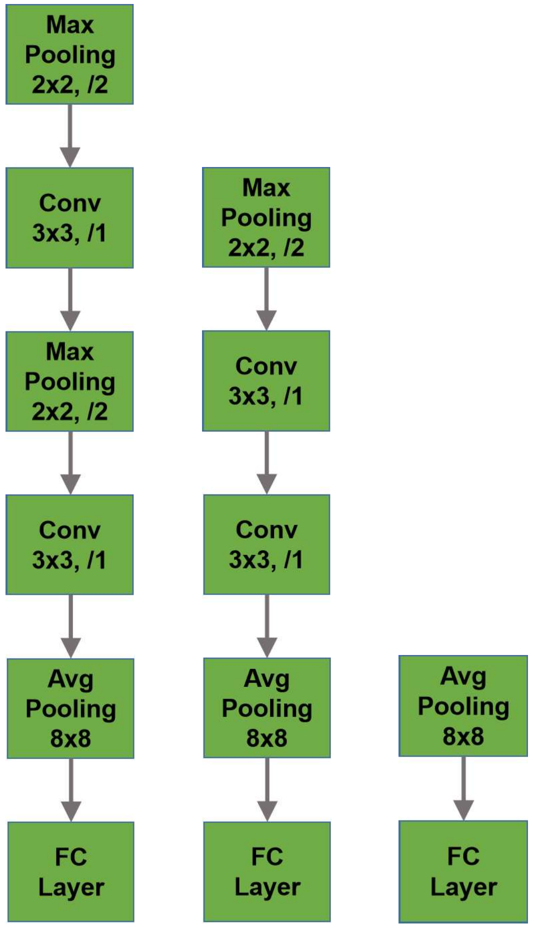


(a) Structure of a Denselayer (b) Structure of the proposed gating network

Figure 12: (a) Canonical denselayer structure in DenseNet. (b) The corresponding channel skipping gating network for (a). The first two layers of the gating network have similar structure of the denselayer in (a), the differences are that we change the stride of the first 1x1 convolution layer to 2 and decrease its output channel numbers by half. Note that by applying these modifications, the total FLOPs of the gating network is roughly 11 % of that of a denselayer.

8. RADI Branch Structure

Since in both ResNet and DenseNet architectures, the feature map size differs only between stages, we design the branch classifiers accordingly. Specifically, as shown in Fig. 13(a), in order to get coarse-grained features that is amenable for classification at early stage, the branch classifier has two max pooling layer at the first stage of the network. After that, the number of max pooling layer is decreased by one at second stage (Fig. 13(b)), and for the third stage, the branch classifier consists of average pooling layer and fully connected layer only (Fig. 13(c)).



(a) (b) (c)

Figure 13: Branch classifier of: (a) stage one of the network, (b) stage two of the network, (c) stage three of the network

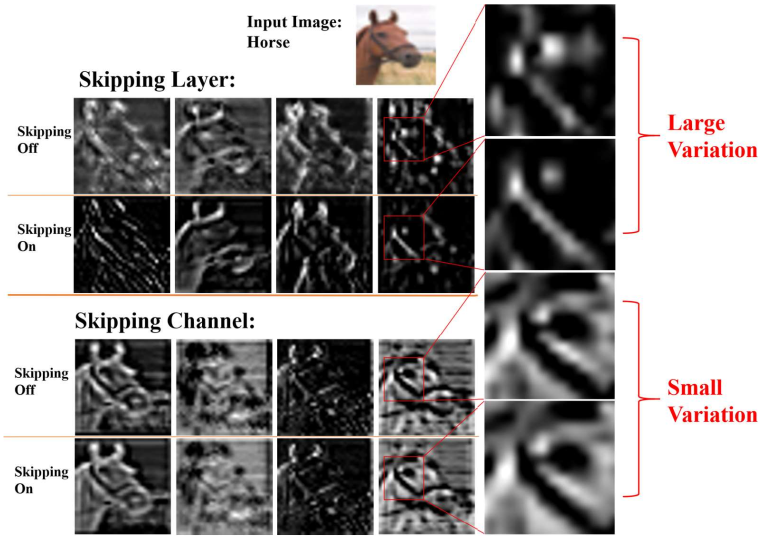


Figure 14: Visualizing feature degradation of layer/channel skipping (i.e., "skipping on") over the original model (i.e., "skipping off"), where the features are obtained from the 6th residual block when using ResNet38.

9. Experimental results on ImageNet

For ImageNet results, we compare our DenseNet-IADI model with skipnet [9] architectures under the same computational cost (measured in millions of FLOPs). As shown in Tbl. 3, under the same 262M FLOPs, DenseNet201-IADI achieves 0.8 % higher accuracy than SkipNet101, and DenseNet121 surpasses SkipNet34 by a margin of 0.6 %, while having around 25 % less FLOPs than it. This demonstrates the effectiveness of our proposed IADI method.

	DenseNet201-IADI	SkipNet101	DenseNet121-IADI	Skipnet34
Computation Cost	262M	262M	217M	285M
Accuracy (%)	**73.8**	73	**74.6**	74

Table 3: Comparison of computational cost and accuracy tradeoff between DenseNet-IADI models and SkipNet models.

10. Detailed Skipping Behavior Visualization and Analysis

Feature Degradation of Layer/Channel Skipping. We here visualize the feature degradation of layer/channel skipping compared with that of the original model. As shown in the example of Fig. 14, layer skipping can cause feature change and large variation in terms of illumination sharpness and clarity, as compared with that of the original one, whereas the feature change and variation is marginal for channel skipping. This justifies the more gradual accuracy loss (see Fig. 7) offered by channel skipping than layer skipping under the same computational cost.

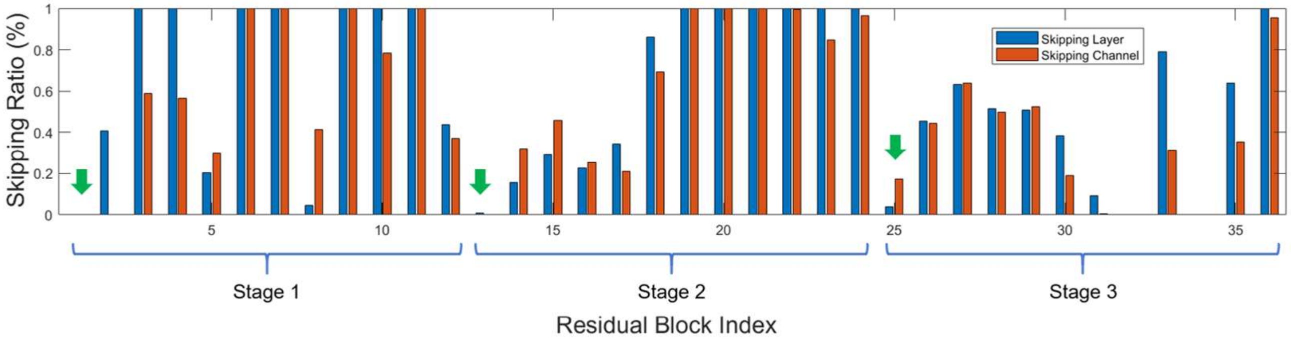


Figure 15: Visualizing the skipping patterns when applying layer/channel skipping to ResNet74 with both having an accuracy of about 92% and 50% computational saving, where there are 36 residual blocks divided into 3 stages uniformly. The first layer of each stage is marked with a green arrow.

Skipping Patterns of Layer/Channel Skipping. To visualize the effectiveness of layer/channel skipping, we show in Fig. 15 the skipping ratio of all the layers in ResNet74 when applying layer and channel skipping with both having a 92% accuracy and 50% computational savings. [43] has shown that while it is possible to skip most of the residual blocks except the first one at each stage for maintaining the accuracy. It is interesting to observe in Fig. 15 that both layer and channel skipping automatically learn the importance of the first residual block at each stage and avoid skipping them.