Appendix

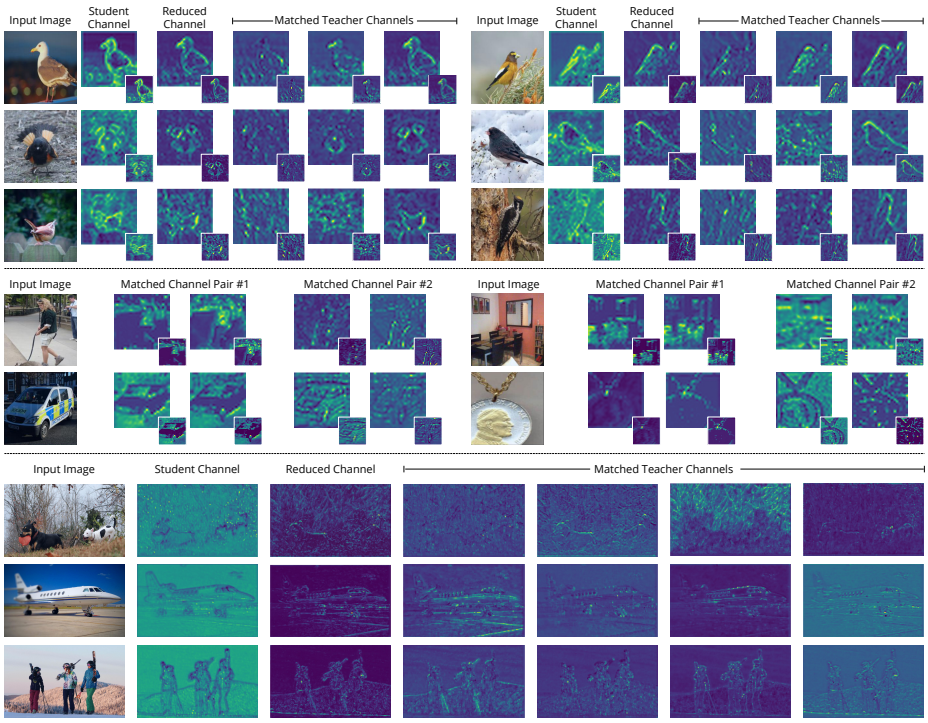

Fig. A1. Channels matching with reduction. The visualization has three parts separated by two dash lines. The first part (top) shows the matching results of *stage-2* in MobileNet-V2 on CUB-200. The channel tensors are visualized in two square patches: small one is in original size of 28×28, the large one is generated by resizing small patch into the input size of 224×224. Each student channel matches three teacher channels. The second part (middle) shows the intermediate matching results in distilling ResNet-50 on IN-1K. Here we find the one-to-one match pair because student has the same channel number with teacher. We randomly select two pairs to visualize. The last part (bottom) shows the results in distilling ResNet-18 on COCO `train2017` set. Each student channel matches four teacher channels. According to this whole visualization, we can easily conclude that the semantic features activations are same between student channels and reduced channels generated by AMP operation.

A1.1 Analysis of Pooling Operations

In order to figure out why the absolute max pooling (AMP) stably works better than average pooling (AvgP) and max pooling (MP) when performing features reduction, we do a fundamental experiment in this part. In Fig. A2, there are two input images (first column from left). First, we build a very simple convolutional

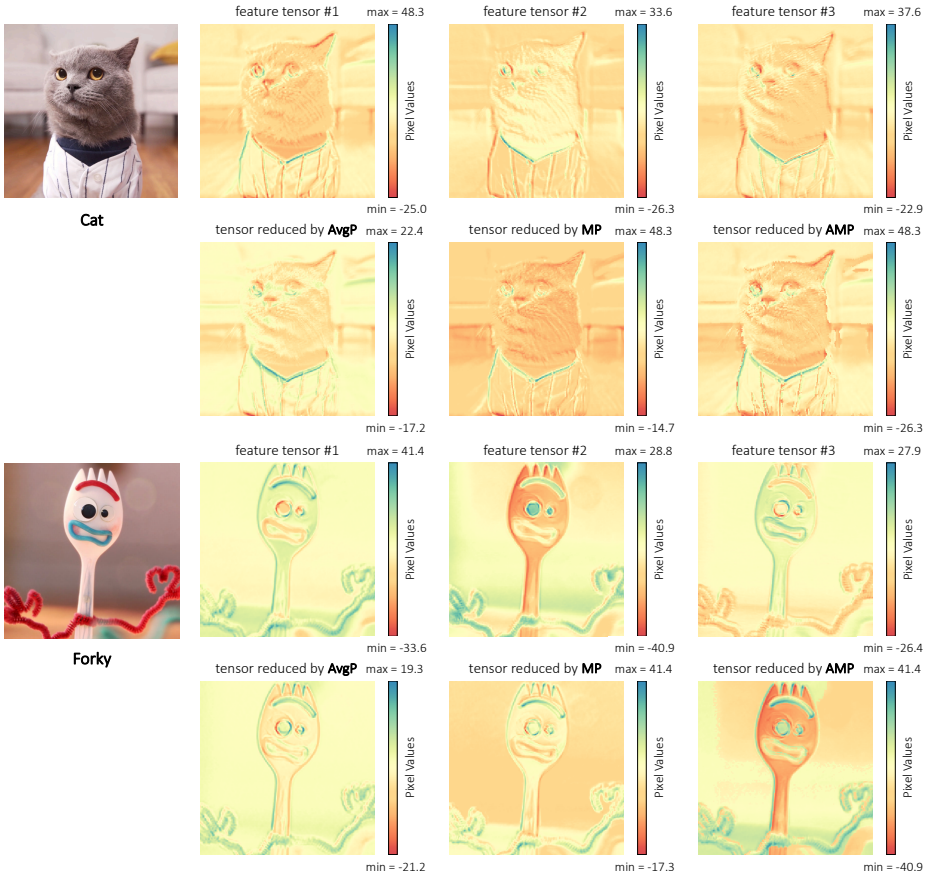

Fig. A2. Comparison of pooling operations. All the feature tensors are normalized into $[0, 1]$ for visualization in order to clearly compare their textures in pixel level and degree. But their min and max values in the color bars use original pixel values without normalization.

network (e.g. LeNet-7 [21]) with random initialization to extract features. Then we select[3] three high-related tensors (in the same row with input images), which have similar semantic feature structures[4] with each other. After using AvgP, MP and AMP operations to perform reduction, we achieve three reduced tensors of each example.

In the case of `Cat`, although AvgP keeps the responses of collar and eyes, it loses the edge activations of right shoulder. MP works well, but its responses of eyes are too weak and also its responses of head texture (including background) are stronger than those of three original features.

[3] This behavior imitates that three teacher channels have been matched with one student channel.

[4] The definition of similar feature structures is made according to their high responses in feature maps.

In the case of `Forky`, AvgP erases the face-body responses from 2^{th} feature. MP not only shades the negative face-body pixels, but also loses the activations of mouth.

Overall, AMP works stably on keeping all the negative and positive texture responses. Moreover, it has ability to hold a good balance between objective and background. This result concludes that AMP works better than both of AvgP and MP for aggregating features. It's possible to use AMP as an alternative general operation for other tasks. For example, in the video classification, AMP can be used to aggregate/pool features along the temporal dimension.