From the ImageNet benchmarking information, we were able to select some models that seem to strike a good balance between performance and lower to moderate parameter count. The general architectures these models cover are visual transformers, EfficientNet, ResNet, and Convolutional Neural Networks (CNN).

ConvNeXt-L (CNN)

The CNN architecture we specifically looked at was ConvNeXt, which is just a modernization of the standard ResNet with design visions towards a Transformer. From there, further architectural changes were made like using depthwise convolutional layers, which is something that reduces parameter count and overall computational overhead. It has a moderate number of parameters (198M) and moderate performance (85.5% top-1).

ResNet

ResNet (Residual Networks) is a model architecture where residual blocks are stacked on top of each other, each of which contain a main path and a shortcut path. The output from each residual block is the sum of these two paths, which allows for better learning of the residual functions. This allows ResNet to work well with very deep networks because the gradients don't vanish.

Global average pooling reduces the spatial dimension and can lead to relatively good performance in terms of memory usage. Although ResNet is fit well for deep networks, the fact that these networks are deep (50-200 layers) can lead to computational cost. The residual connections and global average pooling can also lead to greater computational cost.

The ResNet200_vd_26w_4s_ssld model had significantly lower parameter counts (76M) and moderate performance (85.1%), exhibiting similar performance to ConvNeXt-L with less than half the parameter counts.

EfficientNet (Meta Pseudo Labels)

Meta Pseudo Labels model seems to produce very high top-1 accuracy (90.2%) with moderate parameter counts (480M). The model works by having two networks: one student and one teacher network. The networks are trained in parallel and an instance of EfficientNet-L2 is used for both of the networks.

These EfficientNet models are known for their ability to balance depth, width, and resolution and are suited for lower parameter counts and lower computational

resources. Note that although the model may have a relatively large number of parameters compared with the other four models, it outperforms other models that utilize EfficientNet-L2 with the same number of parameters significantly. This is important to the user because if they have the resources to utilize an EfficientNet-L2, this would be the ideal model to use.

Visual Transformers

We looked at two models that utilize the general visual transformer architecture.

The first model is the TokenLearner model. Because visual transformers tend to create large numbers of tokens to process, the model focuses on “adaptively generating a small number of tokens” in order to save memory and computation without damaging classification performance. It does this through a learnable module that selects the most informative tokens from the input. The number of tokens per layer are reduced to either 8 or 16 per layer instead of thousands, and the TokenLearner can be inserted between transformer layers. It has relatively high performance (88.87% top-1) and moderate parameter count (460M), however it’s more memory efficient as a result of the token reduction.

The second model we read about that utilizes a ViT architecture are Dual Attention Visual Transformers. This model specializes in capturing a global context of the image through the use of two tokens: spatial and channel tokens. The tokens are in essence an inverse of each other. Spatial tokens tend to relate to specific spatial arrangements or locations of features on the image. Channel tokens tend to be the opposite, capturing “an abstract representation of the entire image”. This model seamlessly integrates these two self-attention mechanisms and reduces computational complexity by performing interactions with the channel groups. This model tied Meta Pseudo Labels for the highest accuracy (90.2% top-1) but had a significantly less parameter count (362M).