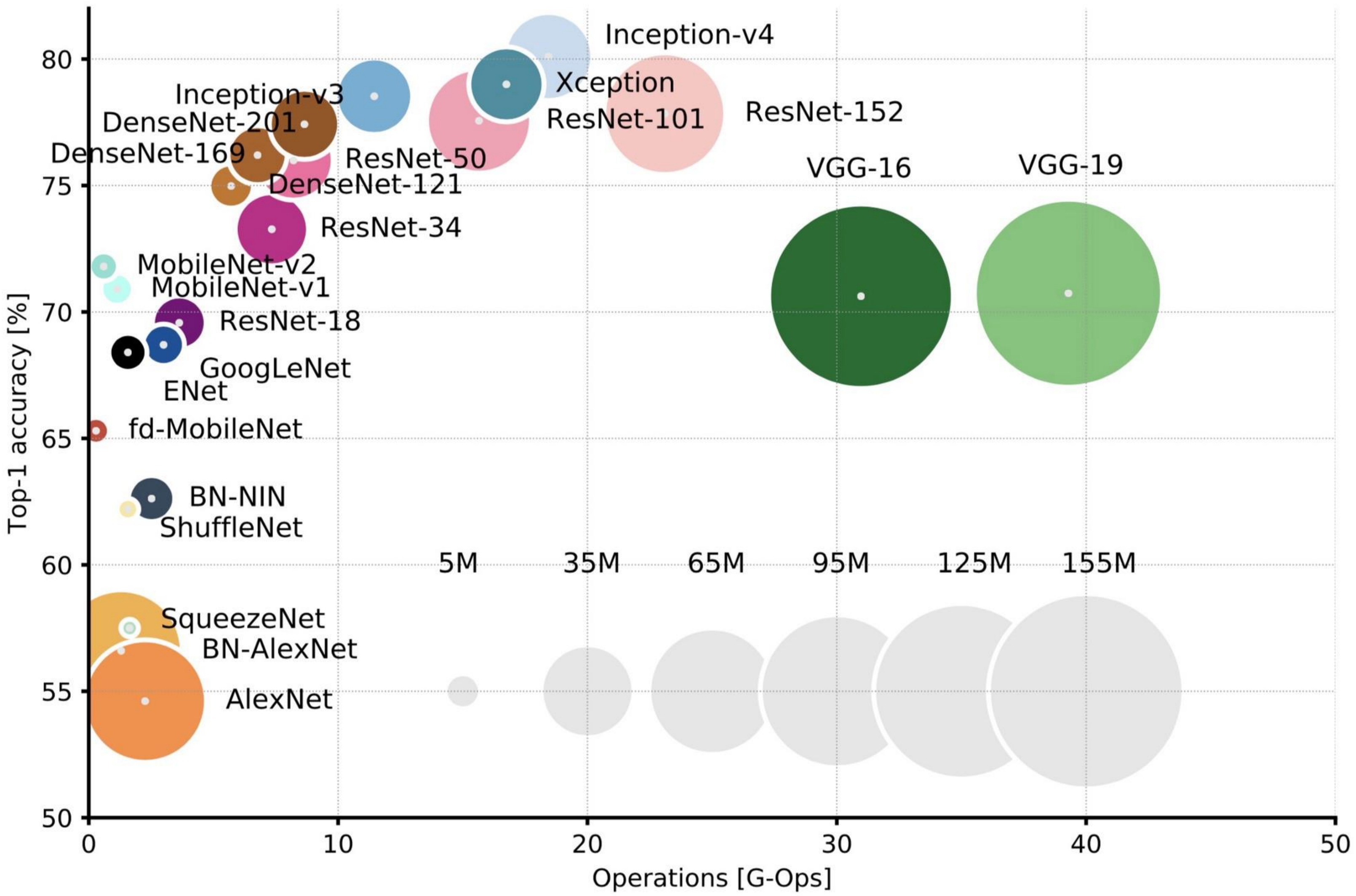
# Transfer Learning

**Industrial AI Lab.**

**Prof. Seungchul Lee**

# ImageNet



Inception-v4
Xception
Inception-v3
DenseNet-201
ResNet-101
ResNet-152
DenseNet-169
ResNet-50
VGG-16
VGG-19
DenseNet-121
ResNet-34
MobileNet-v2
MobileNet-v1
ResNet-18
GoogLeNet
ENet
fd-MobileNet
BN-NIN
ShuffleNet
5M
35M
65M
95M
125M
155M
SqueezeNet
BN-AlexNet
AlexNet
Top-1 accuracy [%]
80
75
70
65
60
55
50
0
10
20
30
40
50
Operations [G-Ops]

# Pre-trained Models

- Training a model on ImageNet from scratch takes days or weeks.

- Many models trained on ImageNet and their weights are publicly available!

- Transfer learning
  - Use pre-trained weights, remove last layers to compute representations of images
  - Train a classification model from these features on a new classification task
  - The network is used as a generic feature extractor
  - Better than handcrafted feature extraction on natural images

# Pre-trained Models

- Training a model on ImageNet from scratch takes days or weeks.

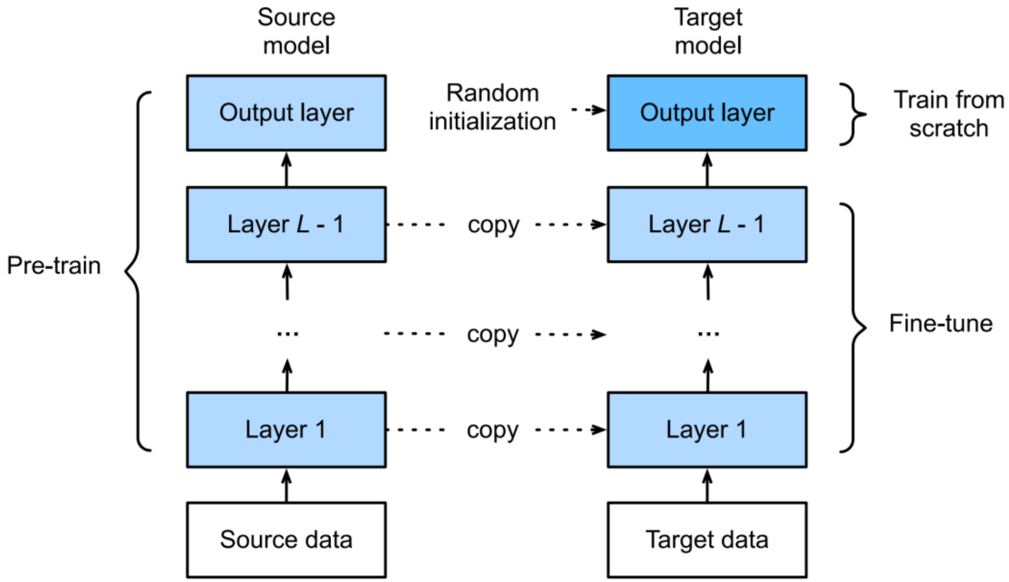
- Many models trained on ImageNet and their weights are publicly available!

- Fine-tuning
  - Retraining the (some) parameters of the network (given enough data)
  - Truncate the last layer(s) of the pre-trained network
  - Freeze the remaining layers weights
  - Add a (linear) classifier on top and train it for a few epochs
  - Then fine-tune the whole network or the few deepest layers
  - Use a smaller learning rate when fine tuning

- Assume we want to identify different kinds of chairs in images and then push the purchase link to the user.
- One possible method is to first find a hundred common chairs, take one thousand different images with different angles for each chair, and then train a classification model on the collected image data set
- Another solution is to apply transfer learning to migrate the knowledge learned from the source data set to the target data set.
- For example, although the images in ImageNet are mostly unrelated to chairs, models trained on this data set can extract more general image features that can help identify edges, textures, shapes, and object composition.
- These similar features may be equally effective for recognizing a chair.

# Transfer Learning

- Pre-train a neural network model on a source data set
- Create a new neural network model. This replicates all model designs and their parameters on the source model, except the output layer. We assume that these model parameters contain the knowledge learned from the source data set and that this knowledge will be equally applicable to the target data set.
- Add an output layer whose output size is the number of target data set categories to the target model, and randomly initialize the model parameters of this layer.
- Train the target model on a target data set, such as a chair data set. We will train the output layer from scratch, while the parameters of all remaining layers are fine tuned based on the parameters of the source model.



# Image Classification with VGG16

- Target data
  - 5 classes

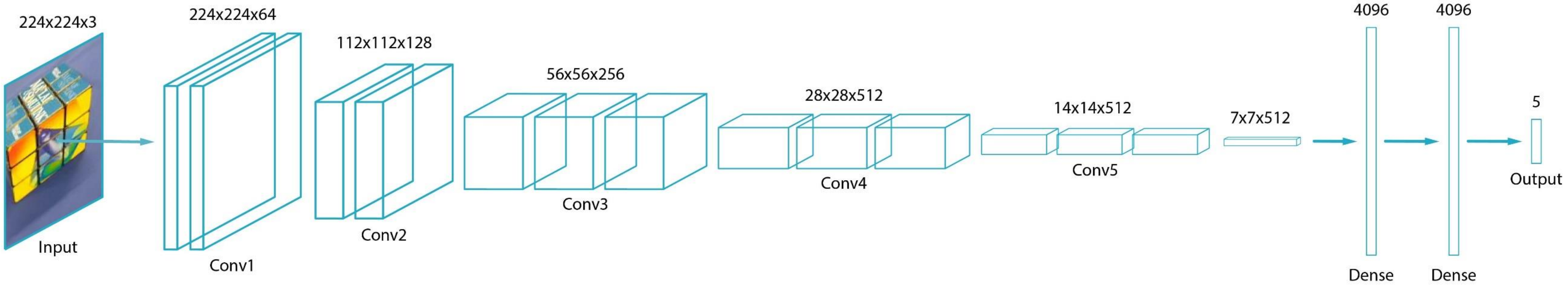
```
Dict = ['Hat','Cube','Card','Torch','screw']
```

- Target data to VGG16
  - Poor performance

```
1. mosquito_net: 4.66%
2. toilet_tissue: 4.00%
3. envelope: 2.29%
4. carton: 2.20%
5. photocopier: 1.86%
```

Label : Cube

# Transfer Learning Structure



# Testing

```
Dict = ['Hat','Cube','Card','Torch','screw']
```

```
Prediction : Cube
Probability : [0.   0.99 0.01 0.   0.  ]
```