CHOOSING CNN ARCHITECTURE FOR BEST RESULTS, TRANSFER LEARNING, FINE TUNING

How to choose best architecture?

Classic network architectures

- LeNet-5
- AlexNet
- VGG 16

Modern network architectures

- Inception
- ResNet
- DenseNet

WHAT KERAS OFFERS

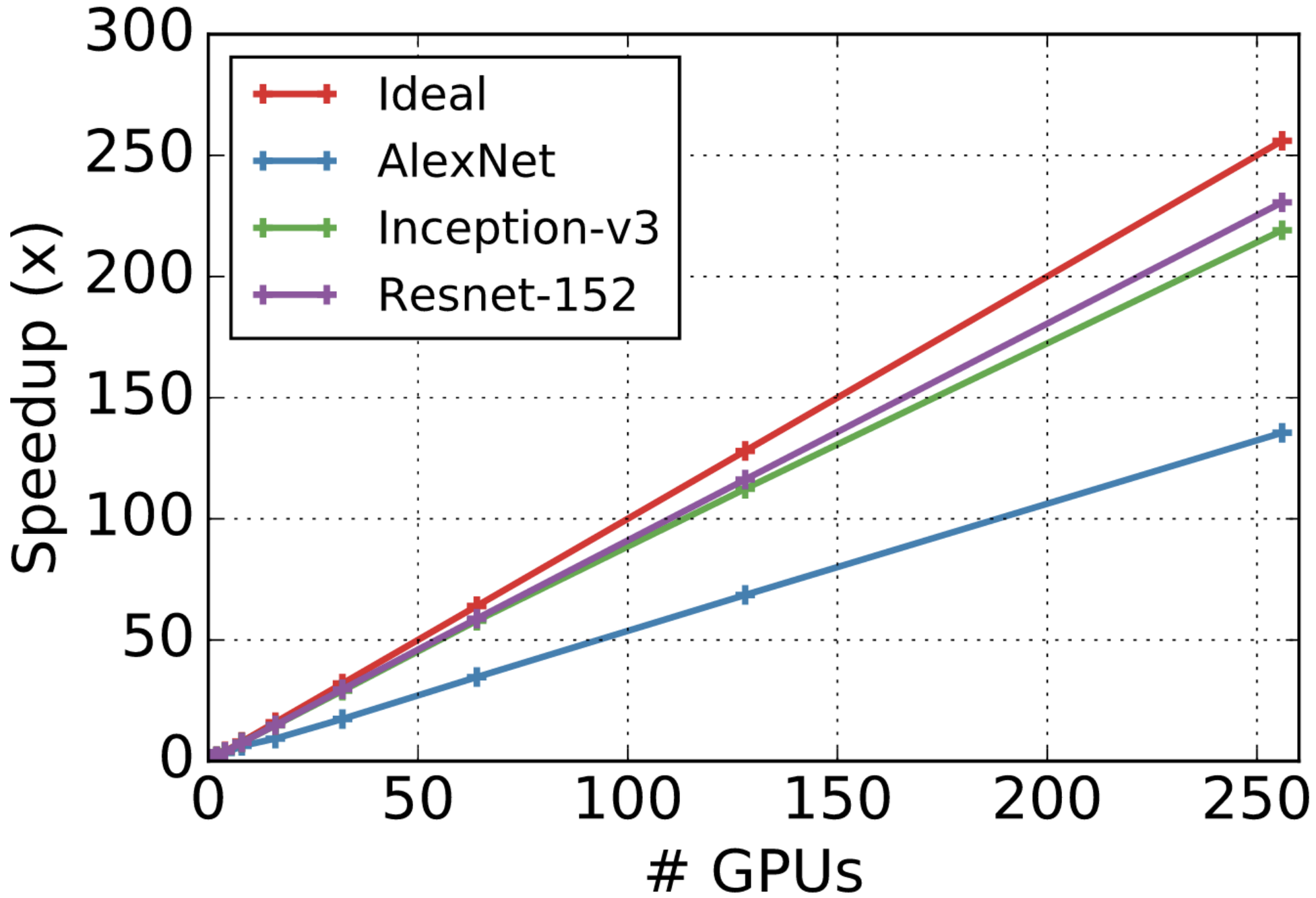
Available models

Models for image classification with weights trained on ImageNet:

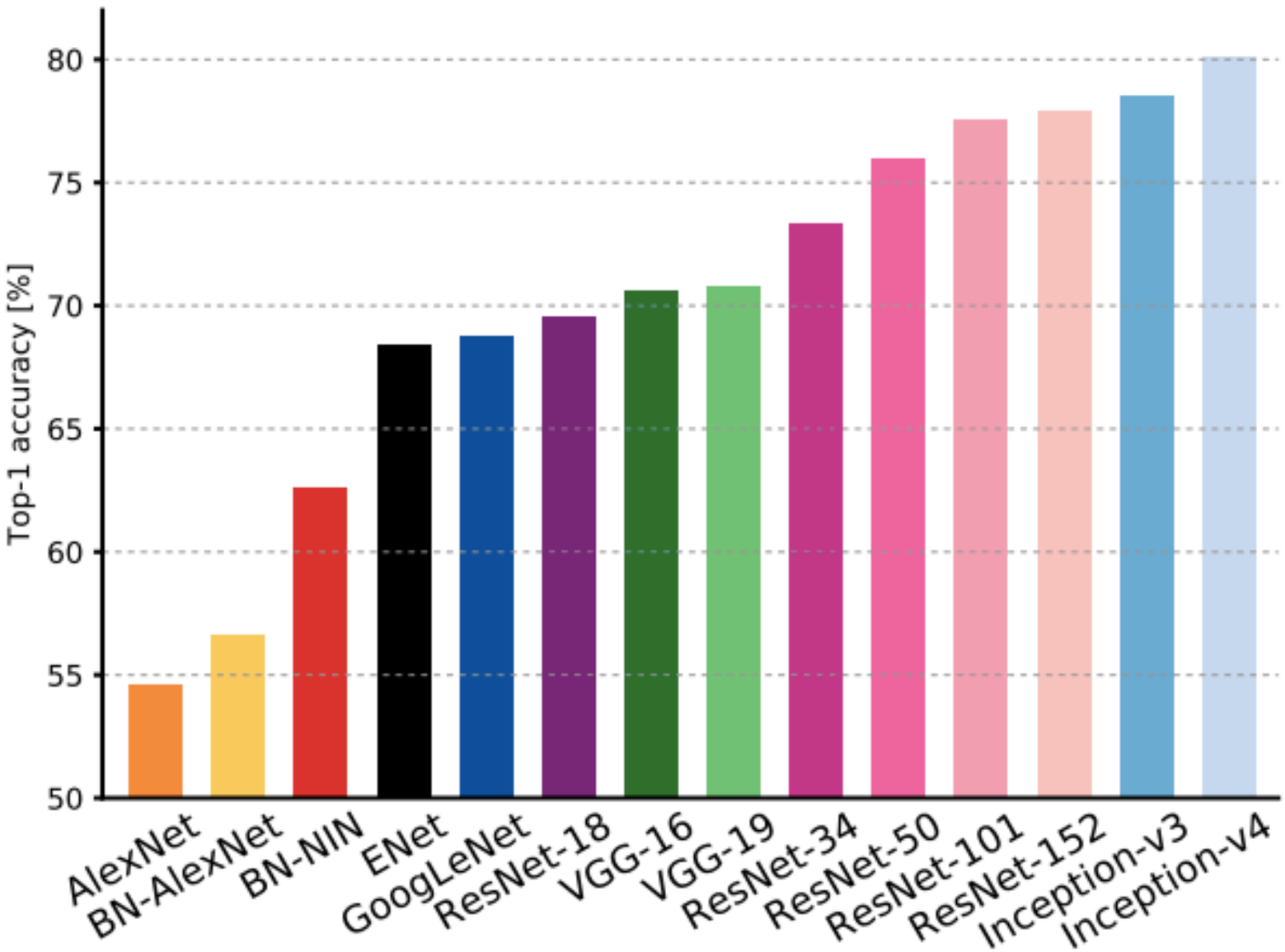
- Xception
- VGG16
- VGG19
- ResNet50
- InceptionV3
- InceptionResNetV2
- MobileNet
- DenseNet
- NASNet
- MobileNetV2

➤ https://keras.io/applications/

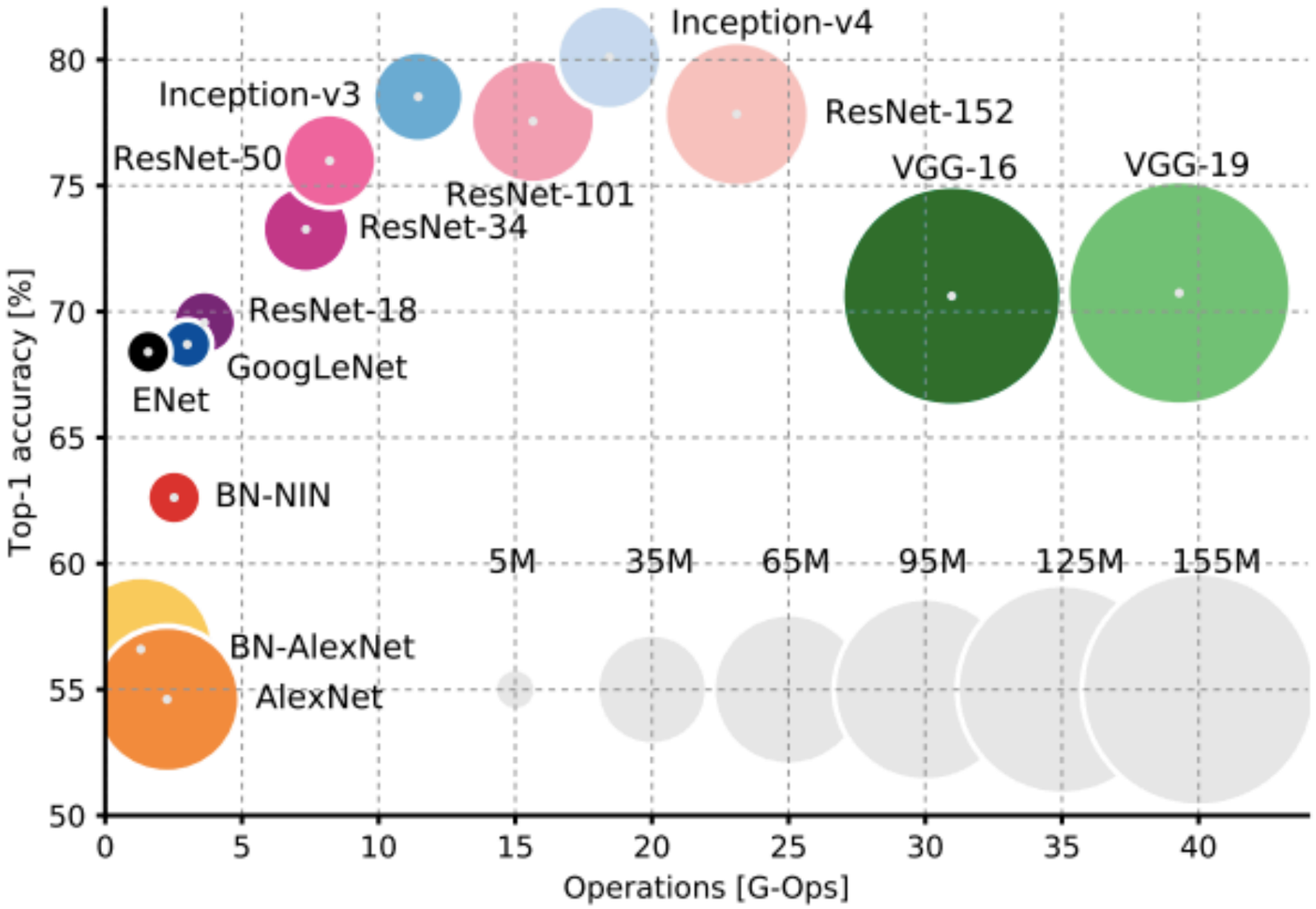
- Why do we have to pick one?
- We don’t have to.
- We can also create an architecture, especially for small datasets.
- However, we can take the advantages of it.



- https://github.com/ROCmSoftwarePlatform/mxnet/tree/master/example/image-classification



- Alfredo Canziani & Eugenio Culurciello, An Analysis of Deep Neural Network Models for Practical Applications, arXiv 2016
- https://arxiv.org/pdf/1605.07678.pdf



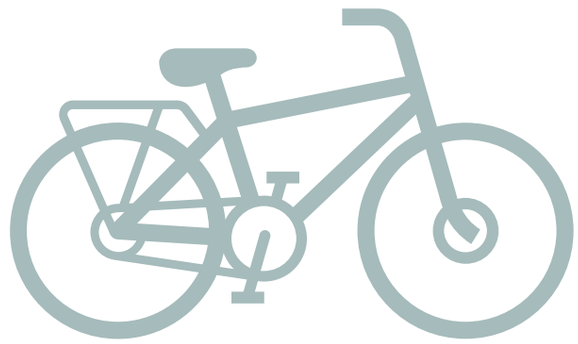
- Alfredo Canziani & Eugenio Culurciello, An Analysis of Deep Neural Network Models for Practical Applications, arXiv 2016
- https://arxiv.org/pdf/1605.07678.pdf

- You can freely use CNN architectures that work well on ImageNet Dataset. On every data.
- ImageNet is a metric of classification.
- How to choose best architecture?
- Just find the best model for your problem.

CHOOSING CNN ARCHITECTURE FOR BEST RESULTS, TRANSFER LEARNING, FINE TUNING

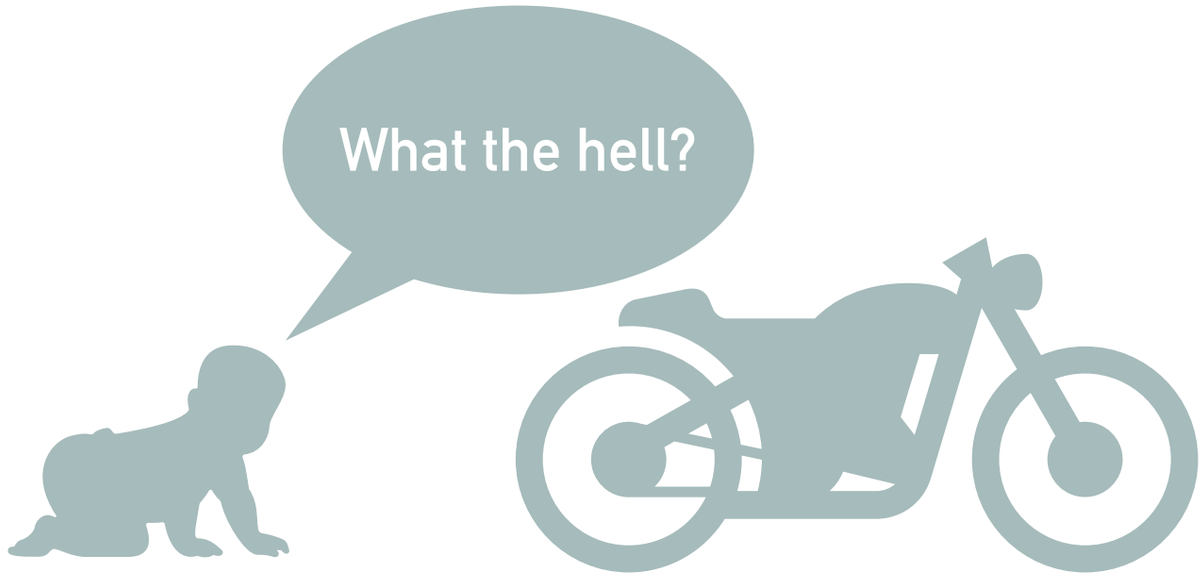
What is Fine Tuning and Transfer Learning?

➤ Suppose you’re new to riding bicycle, you don’t know yet.

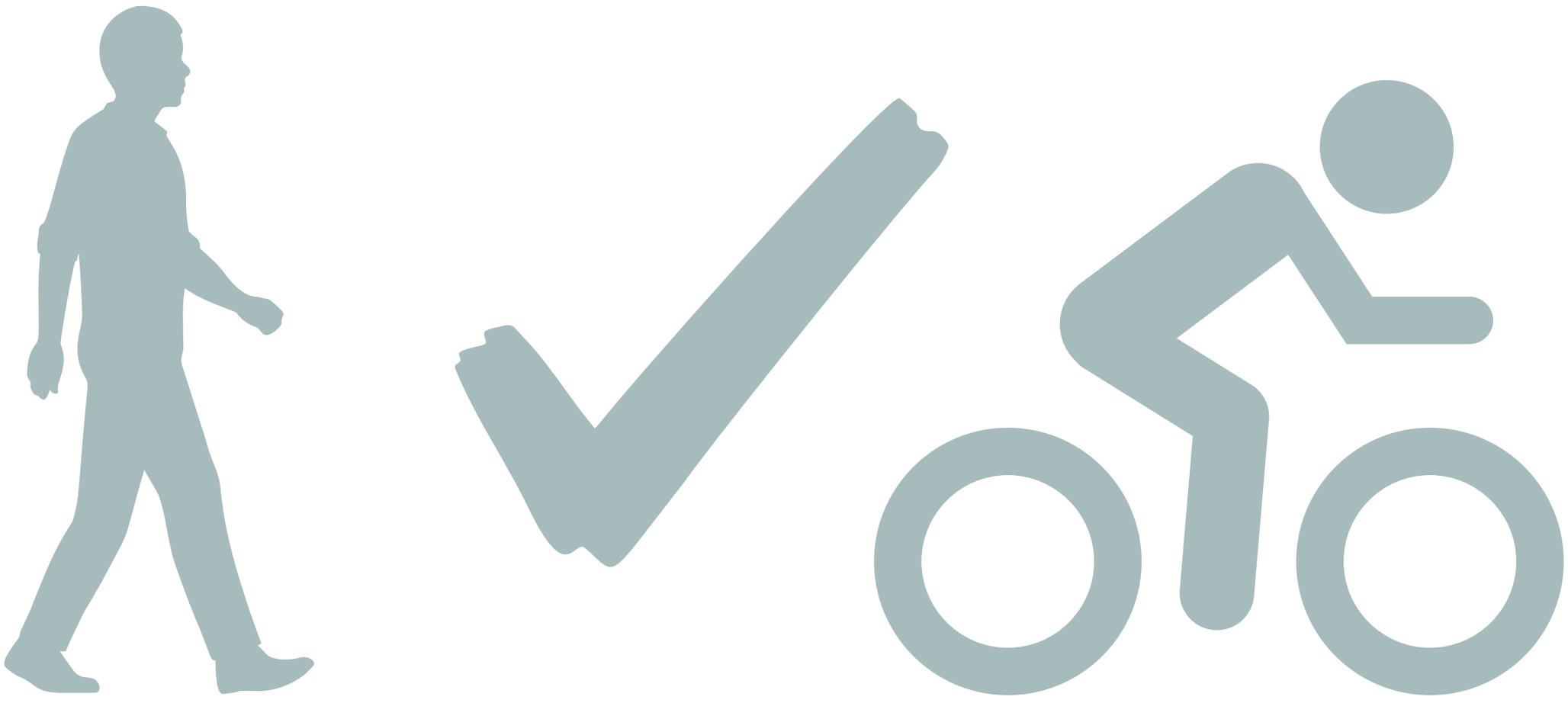


➤ Since you can’t cycle,

➤ You are not able to ride a motorbike.



➤ But if you learn cycling,



- You are more likely to ride a bike.
- Conditions are not the same with cycling.
- But the concepts are the same.
- So what are those?

- Transfer Learning and Fine Tuning are the methods that you can adjust the pre-trained model(ResNet trained on ImageNet) for your dataset.
- There are already CNN architectures that trained on ImageNet data and has very accurate predictions, has high accuracy.
- If you apply transfer learning/finetuning, you are more likely to get a good accuracy on your problem.

- What do you transfer?
- Transferring the knowledge of the high accuracy CNN architecture.
- You transfer the filters that have success on big datasets.
- So you have these filters that can distinguish images, but you should change the weights of filter, or weights, little bit. You should fine tune them.

- These sentences are not wrong, but this is ugly way to teach.

- Let's check the awesome work by Fabien Tence.

- http://ankivil.com/visualizing-deep-neural-networks-classes-and-features/

- What are the advantages?

Training data	1000s to millions of labeled images
Computation	Compute intensive
Training Time	Days to Weeks for real problems
Model accuracy	High (can over fit to small datasets)

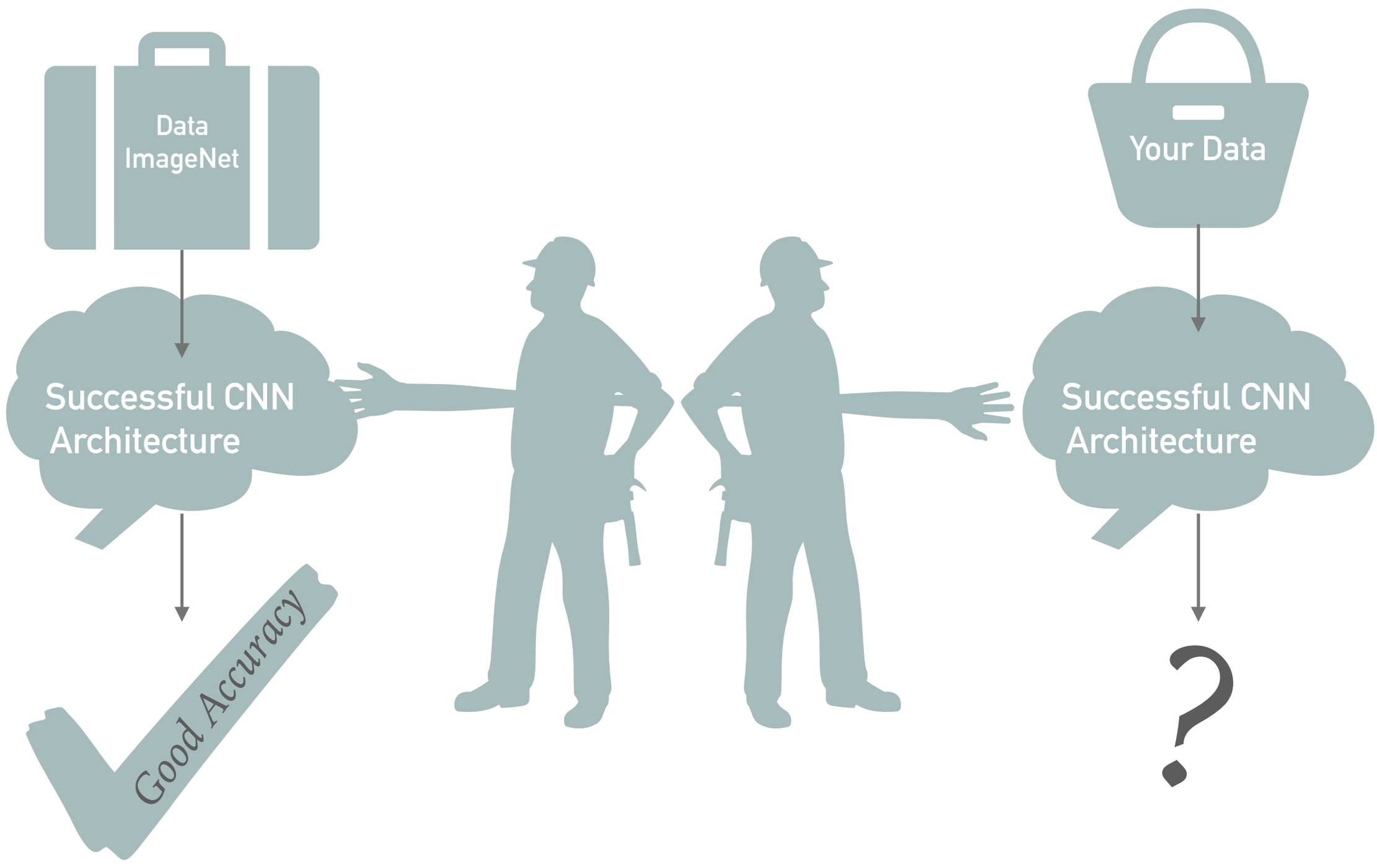
Training data	100s to 1000s of labeled images (small)
Computation	Moderate computation
Training Time	Seconds to minutes
Model accuracy	Good, depends on the pre-trained CNN model

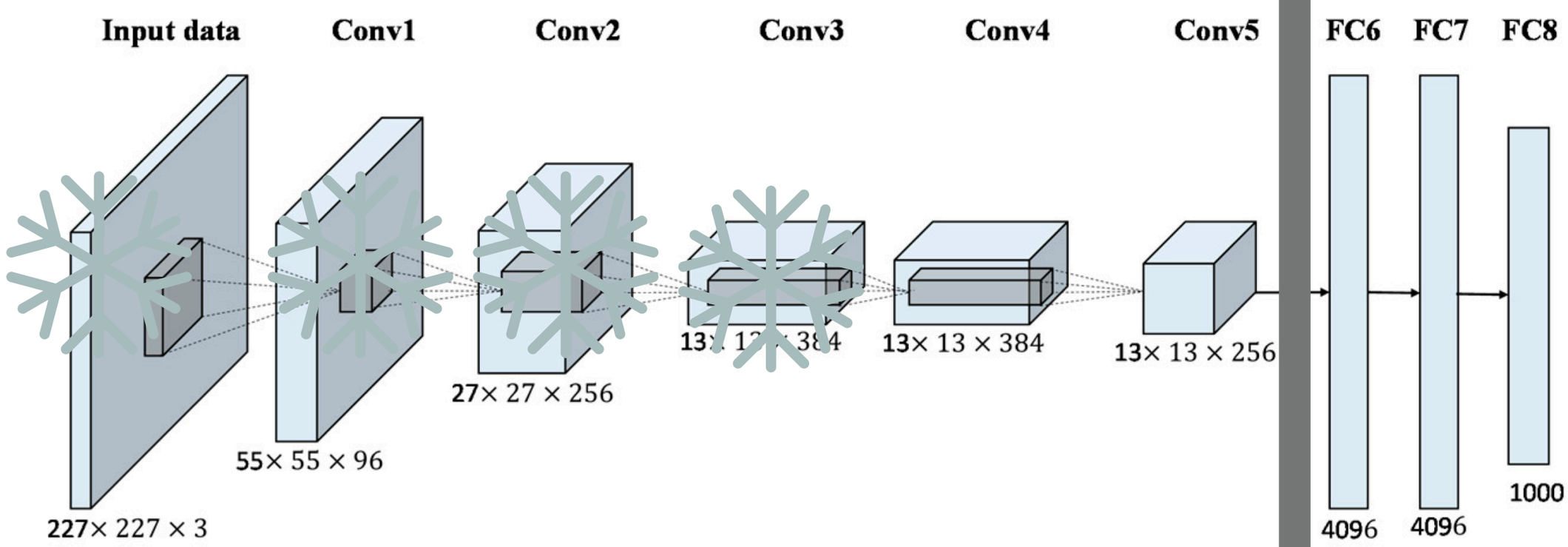
- Anusua Trivedi
- https://www.slideshare.net/PyData/py-datasf

When and how to fine-tune? How do you decide what type of transfer learning you should perform on a new dataset? This is a function of several factors, but the two most important ones are the size of the new dataset (small or big), and its similarity to the original dataset (e.g. ImageNet-like in terms of the content of images and the classes, or very different, such as microscope images). Keeping in mind that ConvNet features are more generic in early layers and more original-dataset-specific in later layers, here are some common rules of thumb for navigating the 4 major scenarios:

1. *New dataset is small and similar to original dataset*. Since the data is small, it is not a good idea to fine-tune the ConvNet due to overfitting concerns. Since the data is similar to the original data, we expect higher-level features in the ConvNet to be relevant to this dataset as well. Hence, the best idea might be to train a linear classifier on the CNN codes.
2. *New dataset is large and similar to the original dataset*. Since we have more data, we can have more confidence that we won't overfit if we were to try to fine-tune through the full network.
3. *New dataset is small but very different from the original dataset*. Since the data is small, it is likely best to only train a linear classifier. Since the dataset is very different, it might not be best to train the classifier form the top of the network, which contains more dataset-specific features. Instead, it might work better to train the SVM classifier from activations somewhere earlier in the network.
4. *New dataset is large and very different from the original dataset*. Since the dataset is very large, we may expect that we can afford to train a ConvNet from scratch. However, in practice it is very often still beneficial to initialize with weights from a pretrained model. In this case, we would have enough data and confidence to fine-tune through the entire network.

➤ http://cs231n.github.io/transfer-learning/


Data
ImageNet
Successful CNN
Architecture
Good Accuracy
Your Data
Successful CNN
Architecture
?



- 1.We will only take the convolutional neural
- network.
- Freeze the majority of the convolutional layer starting from beginning.
- Write your own Top(Dense, Fully Connected Layer) according to your classes.
- Choose a small learning rate. lr=1e-5