Custom Vision Service

How it works?

Optimization Levels in Machine Learning

All machine learning modeling can be broken down into three phases.

1. **Exploring Hypothesis Space** (Model Selection)**:** always designed manually.
2. **Tuning Hyperparameters:** enabled by computational packages but mostly designed manually.
3. **Training Model Parameters:** always handled by computational packages.

Customvision

Since the customvision does not require the user to select the model and model hyperparameters, it automates all three levels of optimization.

Customvision documentation does not exactly reveal how it searches among top two levels of optimization that we defined. But the documentation states that it uses techniques such as Transfer Learning and Image Augmentation.

Now we will try to understand what these techniques are.

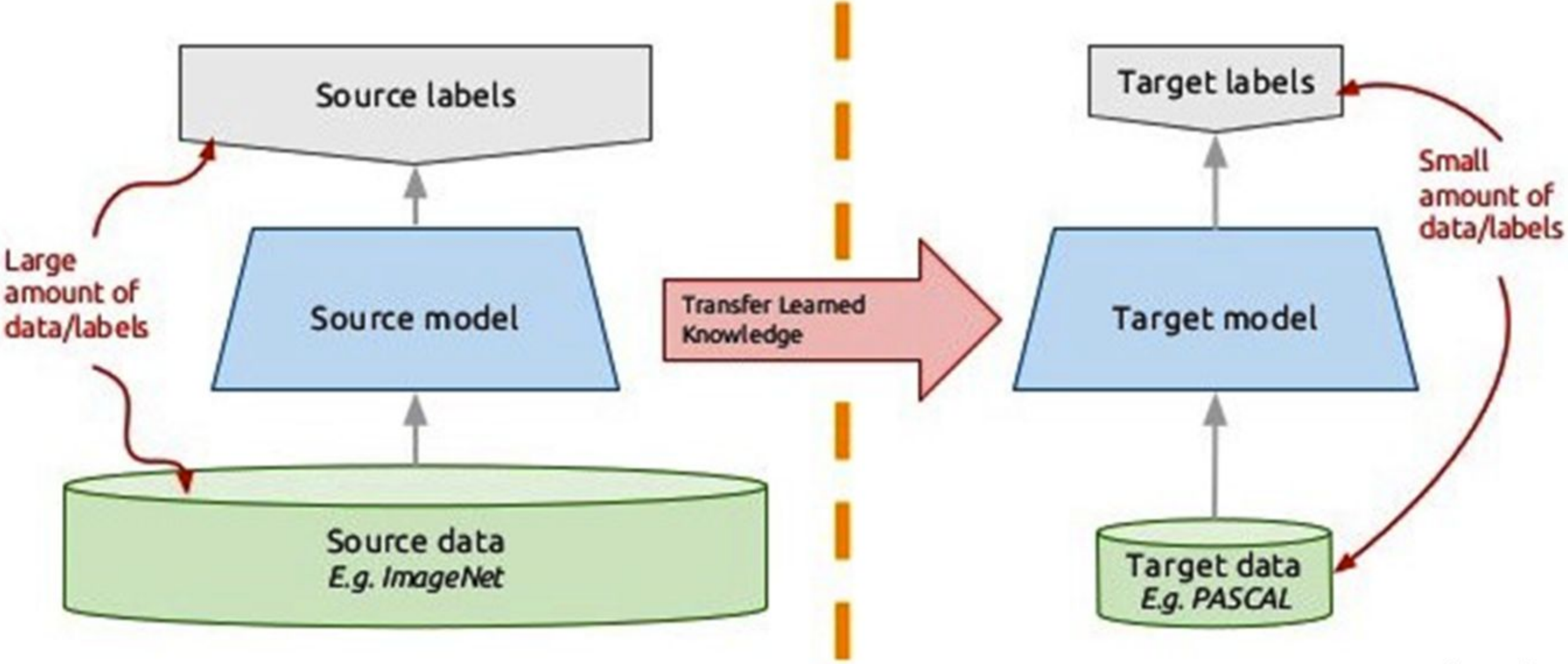
[customvision documentation]

Transfer Learning

To build prior knowledge into the development of the model so it does not learn solely from the data of the problem at hand. Two common ways this is done is through a concept called transfer learning in computer vision and domain adaptation mainly through the use of word embeddings in natural language processing.

Transfer learning enables data scientists to quickly adapt existing pretrained models (e.g., AlexNet, ResNet-50, InceptionV3, etc.) to new domains.

Transfer learning: idea



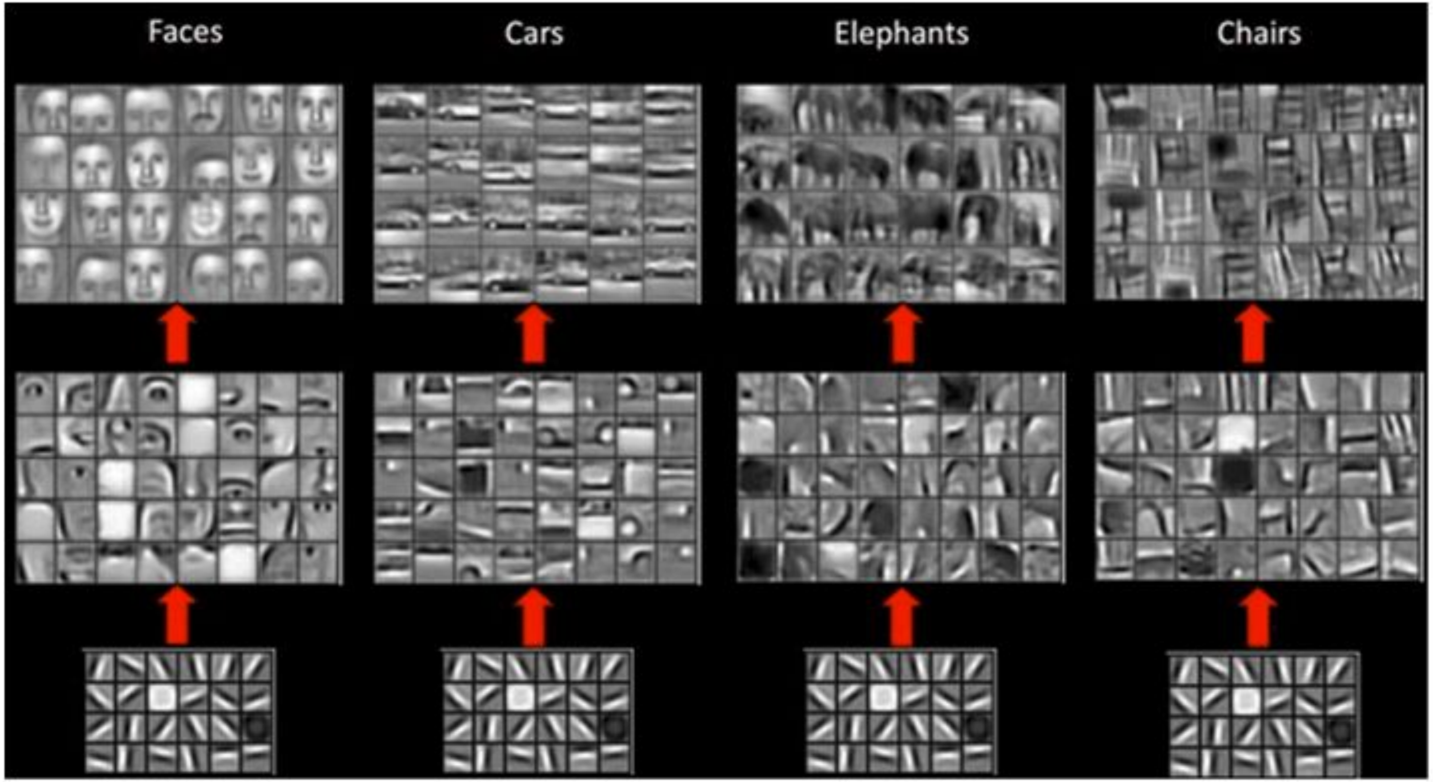
James Le

Two conceptual parts of CNN

CNNs can be conceptually split into two main pieces, both of which are optimized together:

1. The automatic feature extractor creates the hidden feature state—features that represent aspects of image that are relevant for classification—and is made up of layers such as convolutional and pooling layers.

2. The classifier is a fully connected neural network made up of at least one layer that classifies the hidden feature state.



Visualizing increasingly complex filters learned in CNN layers.

Table 2-2. *Different Types of Transfer Learning*

Type	How Is Transfer Learning Used?	How to Train?
Standard DNN	None	Train featurization and output jointly
Headless DNN	Use the features learned on a related task	Use the features to train a separate classifier
Fine-tune DNN	Use and fine-tune features learned on a related task	Retrain featurization and output jointly with a small learning rate
Multitask DNN	Learned features need to solve many related tasks	Share a featurization network across both tasks

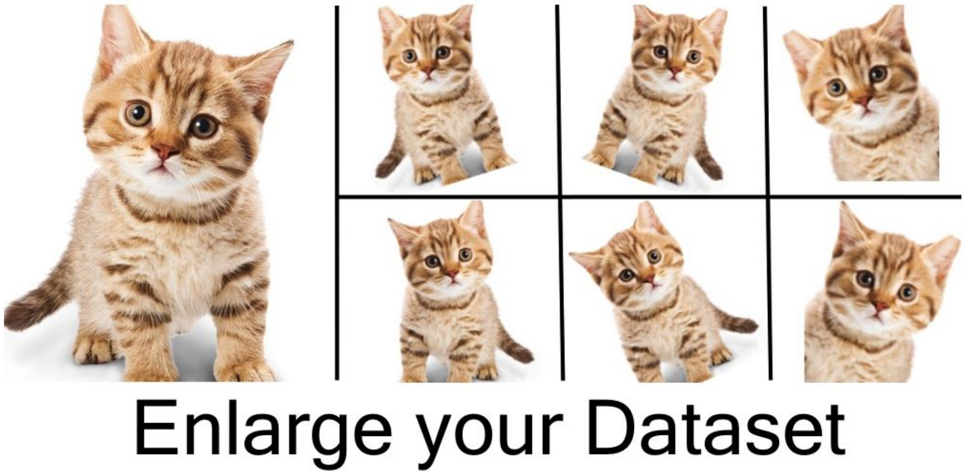
Pre-trained Model Approach

1. **Select Source Model**. A pre-trained source model is chosen from available models. Many research institutions release models on large and challenging datasets that may be included in the pool of candidate models from which to choose from.
2. **Reuse Model**. The model pre-trained model can then be used as the starting point for a model on the second task of interest. This may involve using all or parts of the model, depending on the modeling technique used.
3. **Tune Model**. Optionally, the model may need to be adapted or refined on the input-output pair data available for the task of interest.

Read more here

Image Augmentation and translation Invariance

Translation invariance is a concept that an object should be detected irrespective of its location and orientation. So dataset can be enhanced by creating new images by transformations of the existing images. [Read More]



Custom Vision Training

So based on these ideas that is what I suppose happens when you Train a model.

1. Image Augmentation
2. Selects a suitable trained CNN based on domain you choose
3. Initializes parameters for feature extraction layers
4. Trains fully connected Layers (and might fine tuned inner layers)

Benefits and Drawbacks

- Small Training Data (50 to 100 for a classifier)
- Faster Training
- Model might not be globally optimum in all three levels of optimization.

Data Centric Model Improvement

Since everything else is taken care of, the only thing that the model developer needs to worry about is the quality, diversity and quantity of data provided to the model. Following are important considerations in data preparation.

1. Class Imbalance Treatment (read more)
2. Ensuring variation in light condition, background, angles, styles
3. Ensuring that non characteristic features are uniformly distributed among label classes. (So that model does not associate e.g daylight with a certain class)

A valuable resource

Book: Deep Learning with Azure

I have the pdf copy of this book, you can reach out to me if you want it.