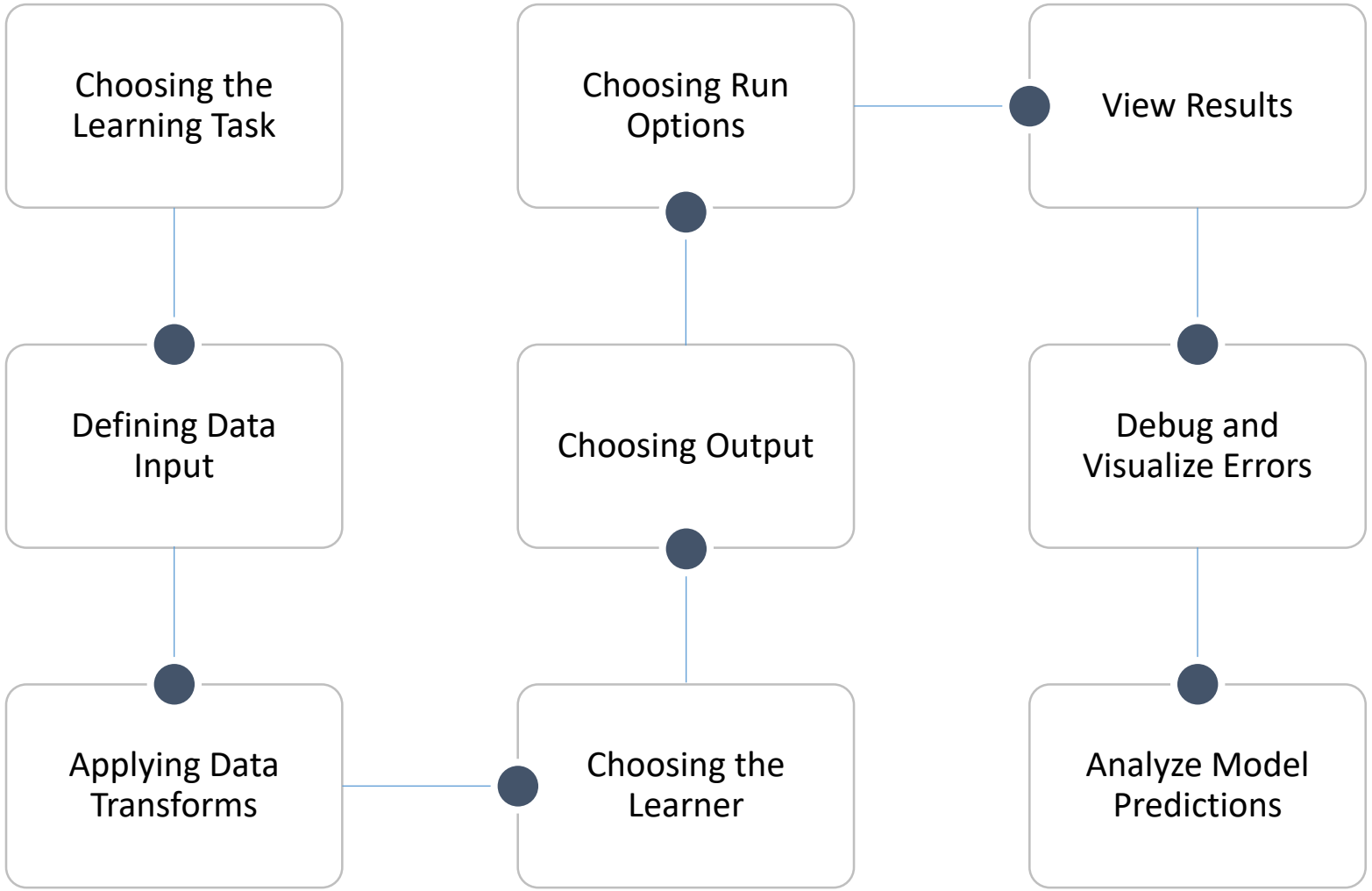
How to use transfer learning to bootstrap image classification and question answering (QA)

Danielle Dean PhD, Wee Hyong Tok PhD
Principal Data Scientist Lead
Microsoft

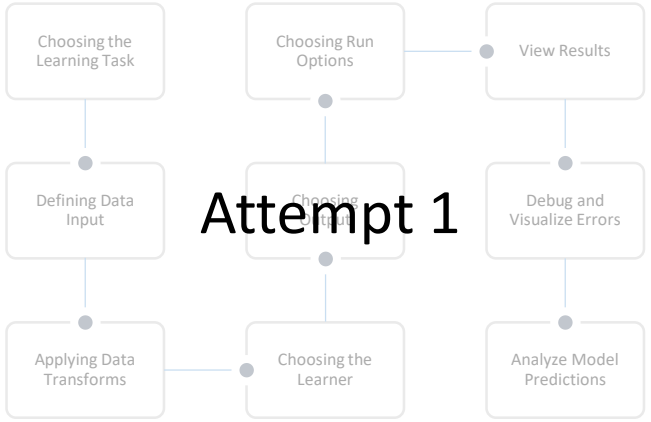
@danielleodean | @weehyong

Inspired by “Transfer Learning: Repurposing ML Algorithms from Different Domains to Cloud Defense” , Mark Russinovich, RSA Conference 2018

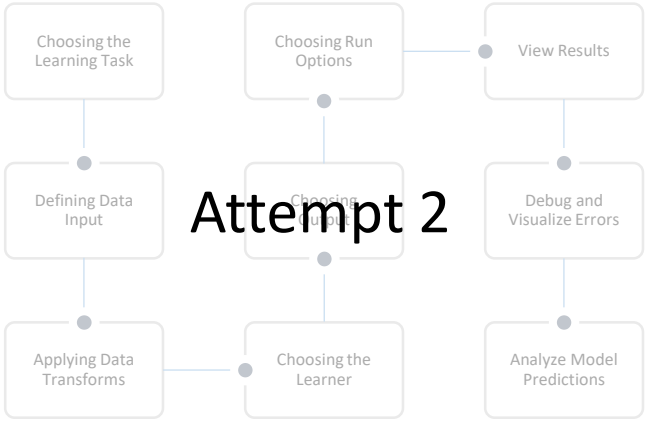
Textbook ML development



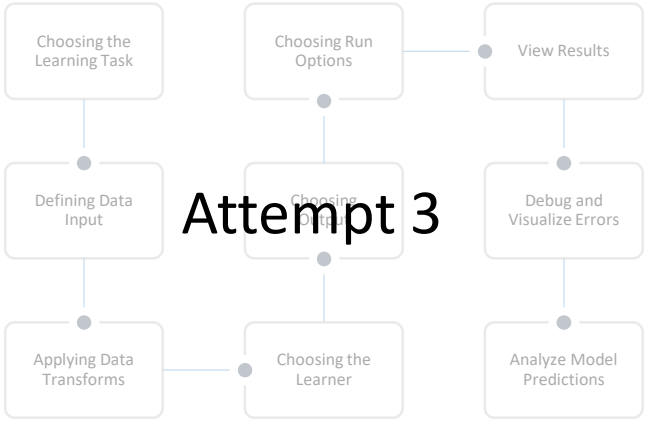
Fact | Industry grade ML solutions are highly exploratory



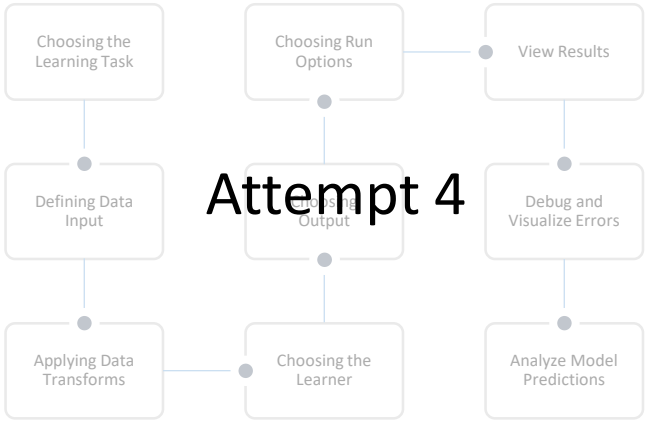
Attempt 1



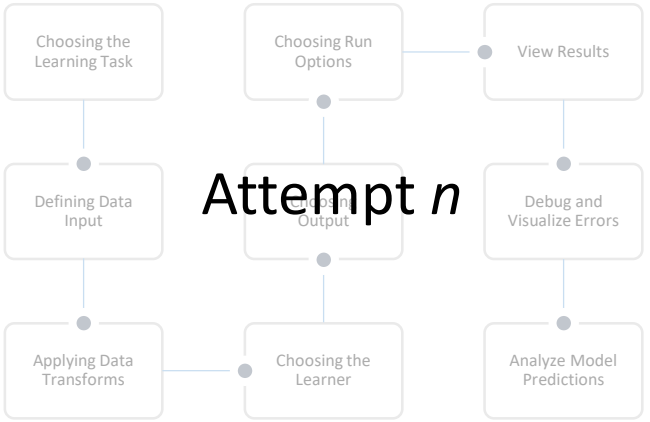
Attempt 2



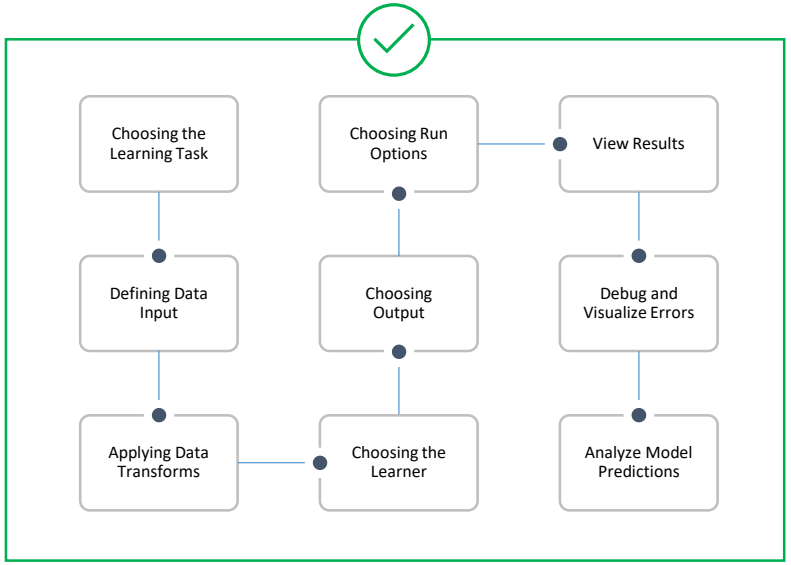
Attempt 3



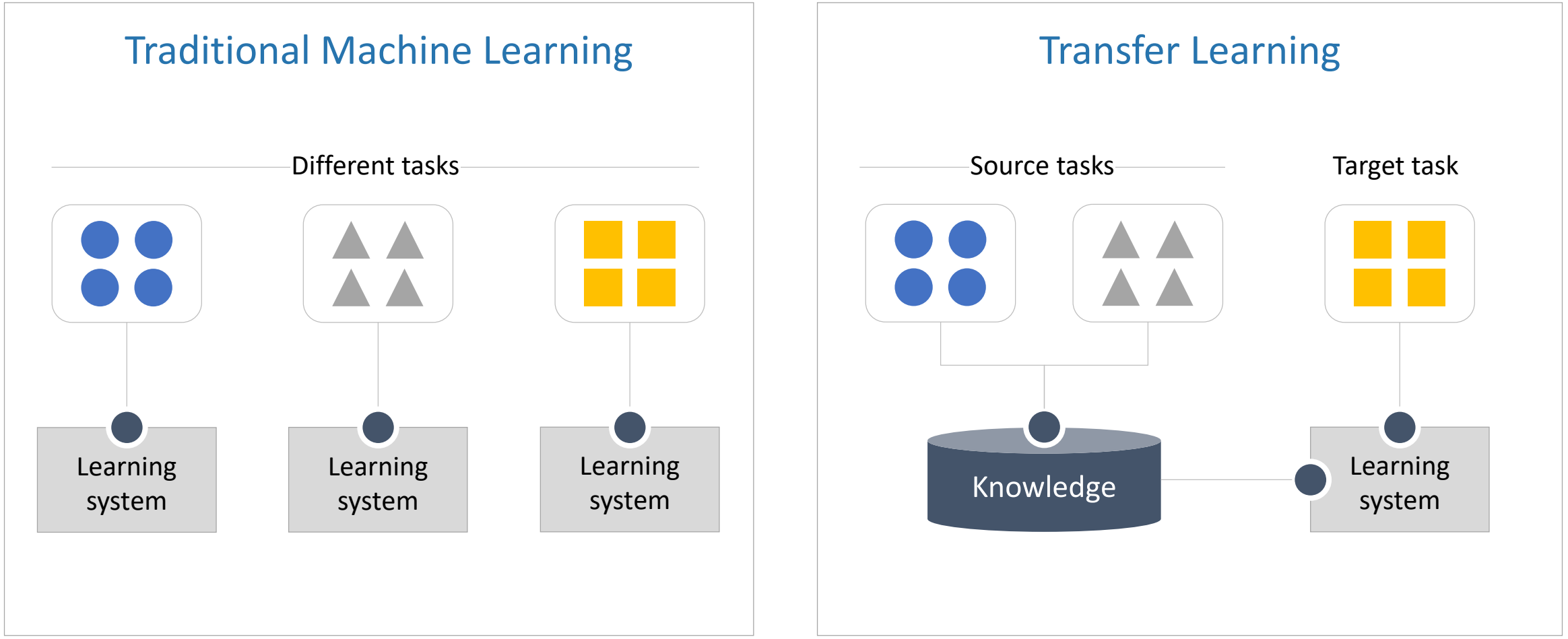
Attempt 4



Attempt *n*

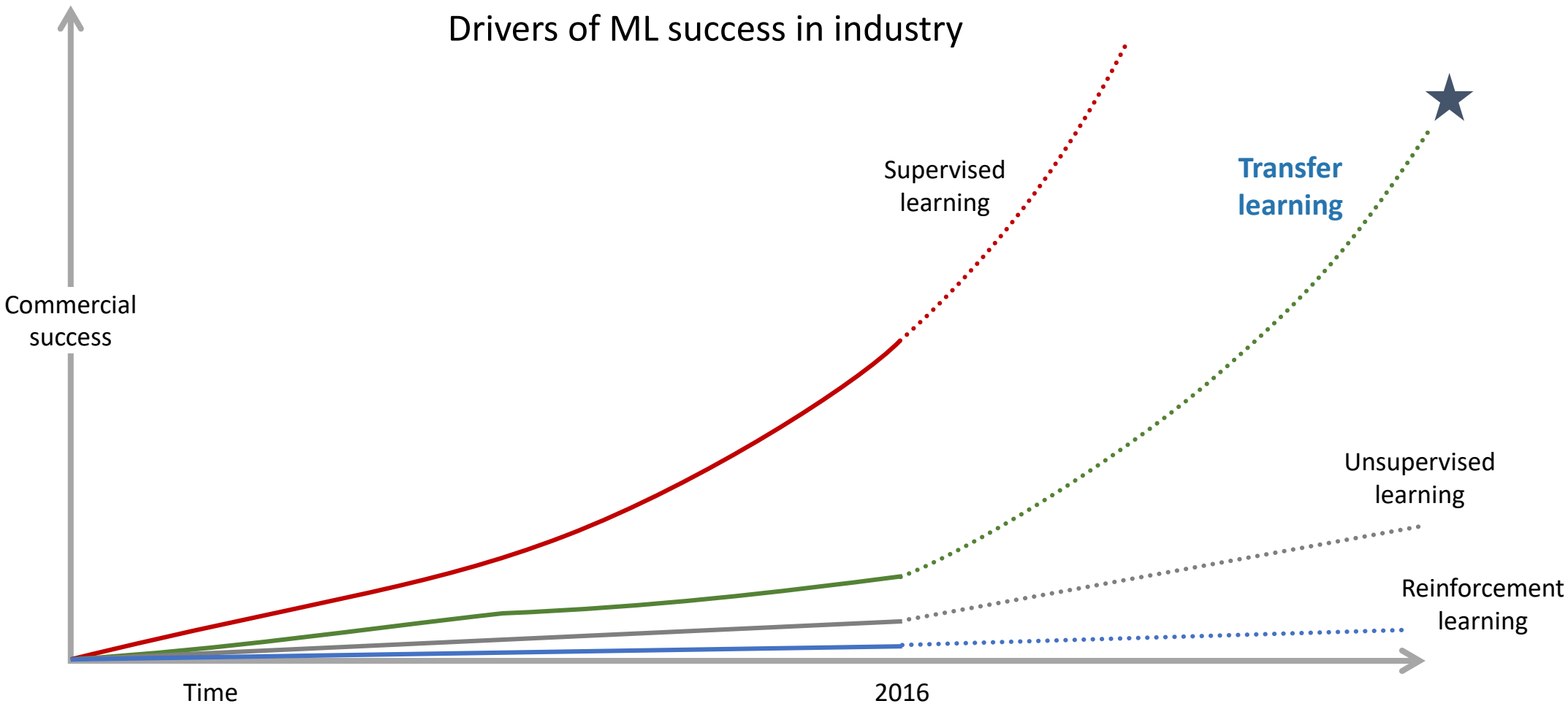


Traditional versus Transfer learning



Source: "A survey on transfer learning." , Pan, Sinno Jialin, and Qiang Yang. *IEEE Transactions on knowledge and data engineering*

Why are we talking about transfer learning ?



Source: "Transfer Learning - Machine Learning's Next Frontier" , Ruder, Sebastian,

Transfer Learning in Computer Vision

Can we leverage knowledge of processing images to help with new tasks?

- What's in the picture?
- Where is the bike located?
- Can you find a similar bike?
- How many bikes are there?

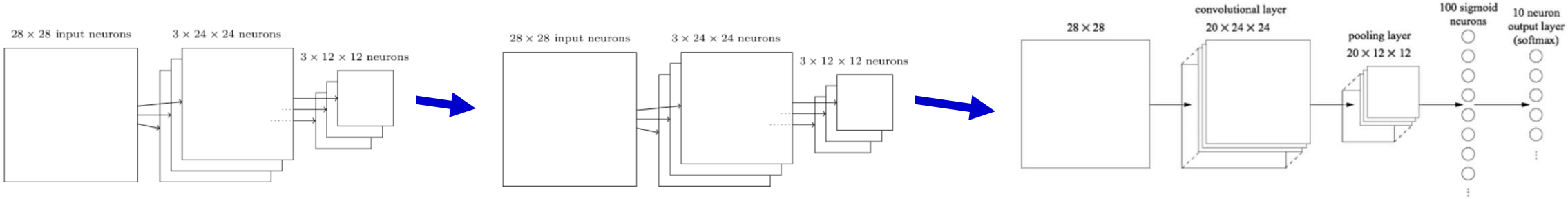
Before Deep Learning

- Researchers took a traditional machine learning approach
  - Manual creation of a variety of different visual feature extractors
  - Followed by traditional ML classifiers

- Example: HoG Detectors
  - Histogram of oriented gradients (HoG) features
  - Sliding window detector
  - SVM Classifier
  - Very fast OpenCV implementation (<100ms)

- Features not very generalizable to other vision tasks – <u>not easy to transfer</u>

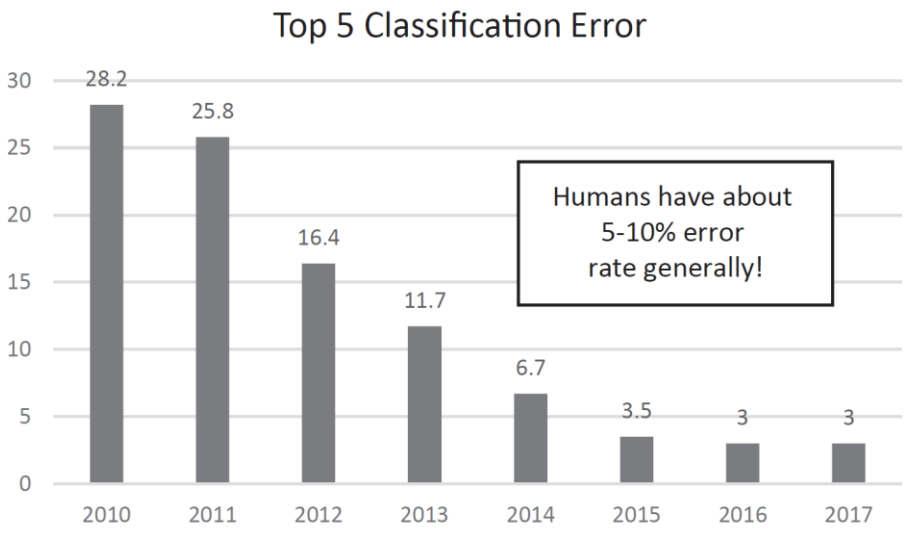
Deep Neural Networks



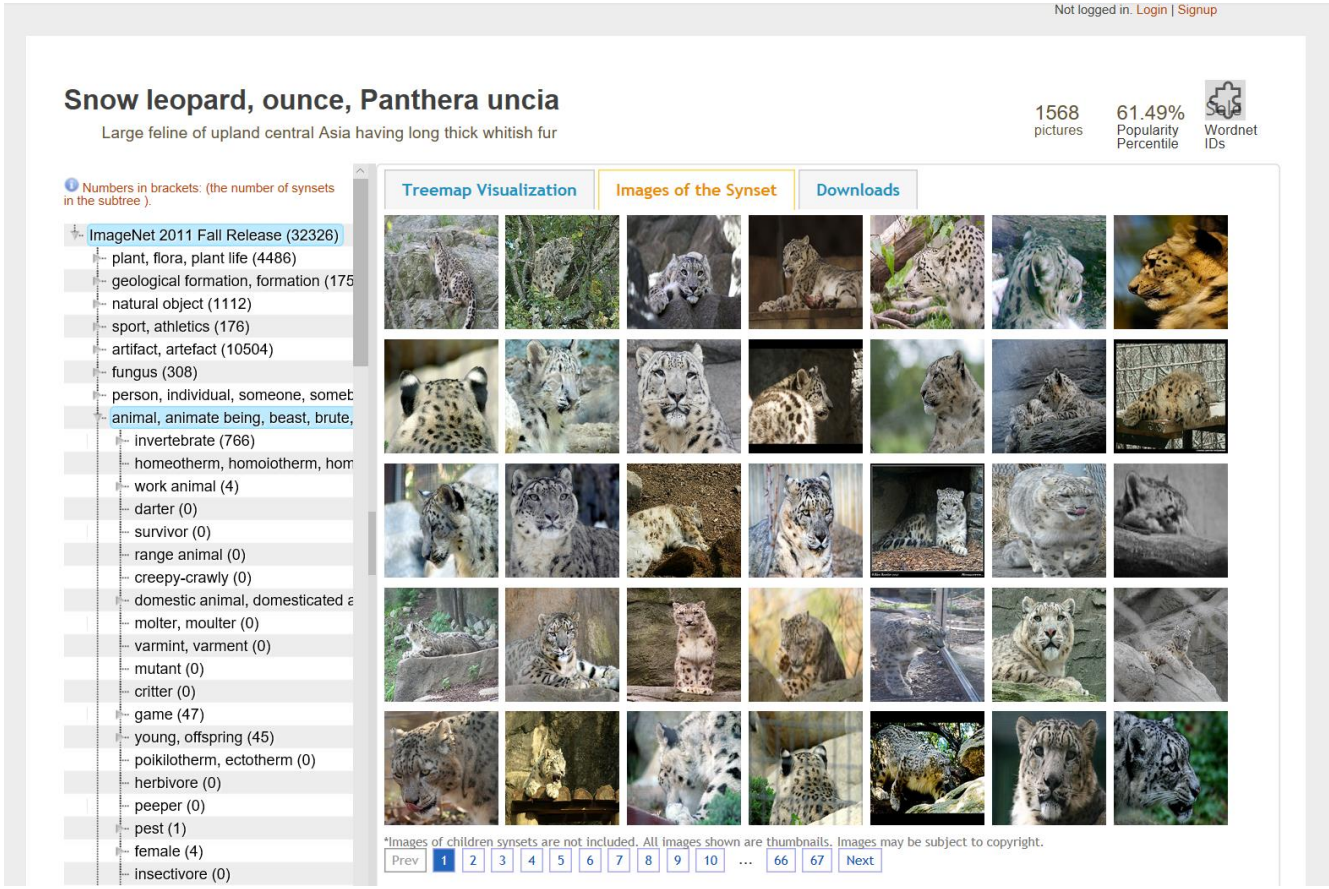
- A convolutional net can have multiple convolution and pooling layers feeding each other
- Shown to perform extremely well at many computer vision tasks
- Need *millions of annotated data points* and *significant compute power*

14,197,122 images
21841 synsets

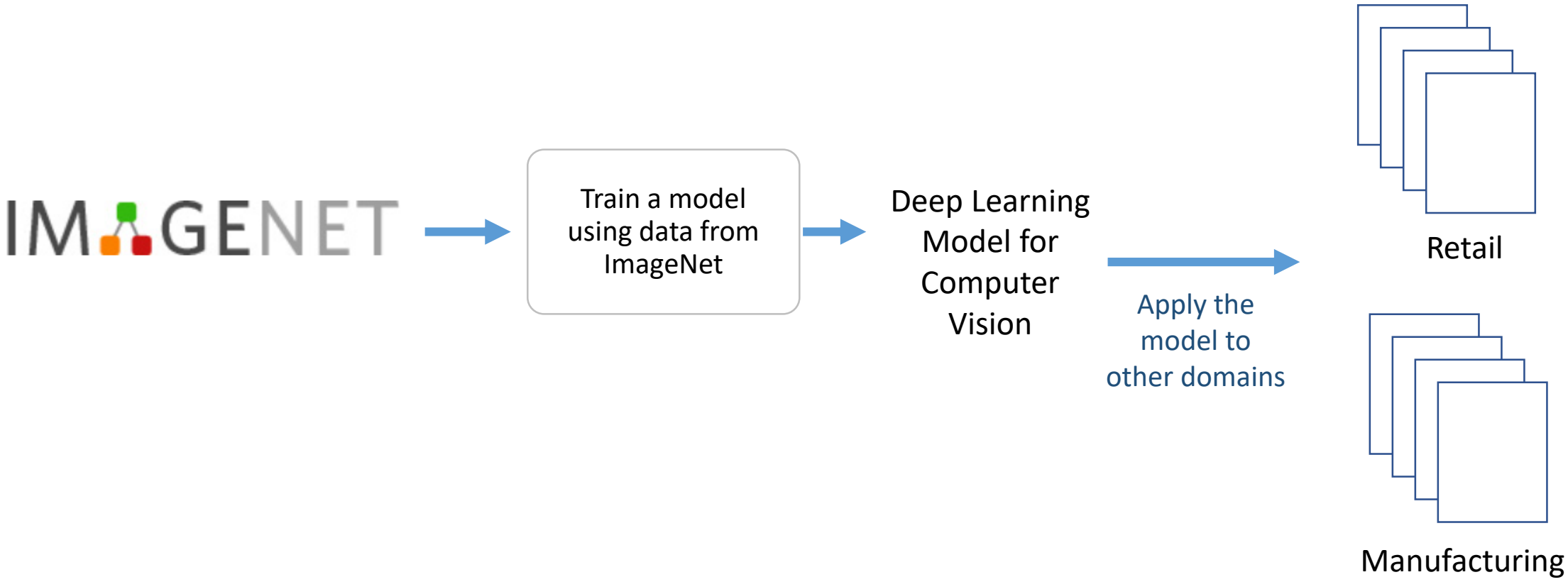
Diverse images, Lots of labels!



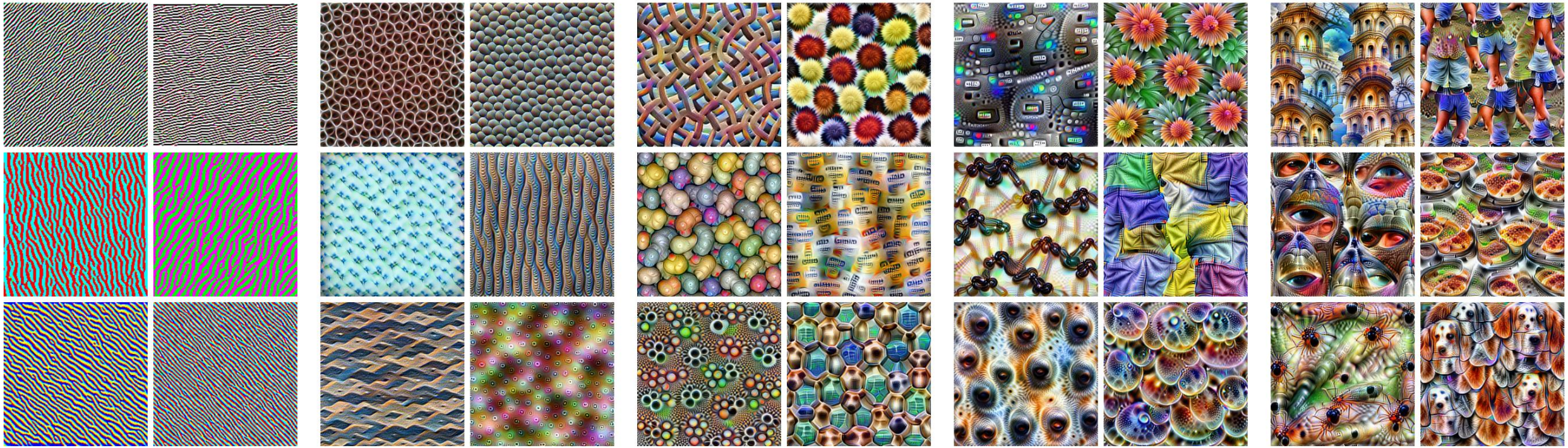
IMAGENET



Transfer Learning for **Computer Vision**



Example – Visualizing the different layers



Edges (layer conv2d0) **Textures** (layer mixed3a) **Patterns** (layer mixed4a) **Parts** (layers mixed4b & mixed4c) **Objects** (layers mixed4d & mixed4e)

Source: Olah, et al., "Feature Visualization", Distill, 2017
https://distill.pub/2017/feature-visualization/

Another fun site:
https://deepart.io/nips/submissions/random/

http://cs231n.stanford.edu/

Example – Visualizing the different layers

Simple Optimization

Optimization with diversity reveals four different, curvy facets. *Layer mixed4a, Unit 97*

Dataset examples

Check out these sites -

https://deepart.io/nips/submissions/random/

http://cs231n.stanford.edu/

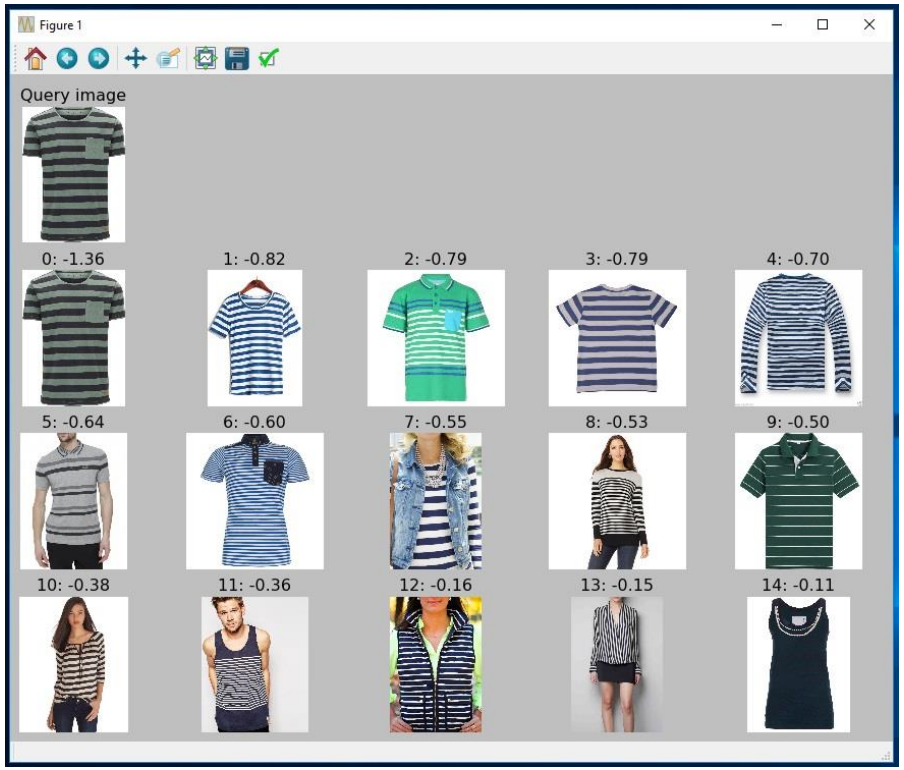
Source: Olah, et al., "Feature Visualization", Distill, 2017

https://distill.pub/2017/feature-visualization/

Retail Website

Clothing texture dataset:

- 1716 images from Bing which were manually annotated

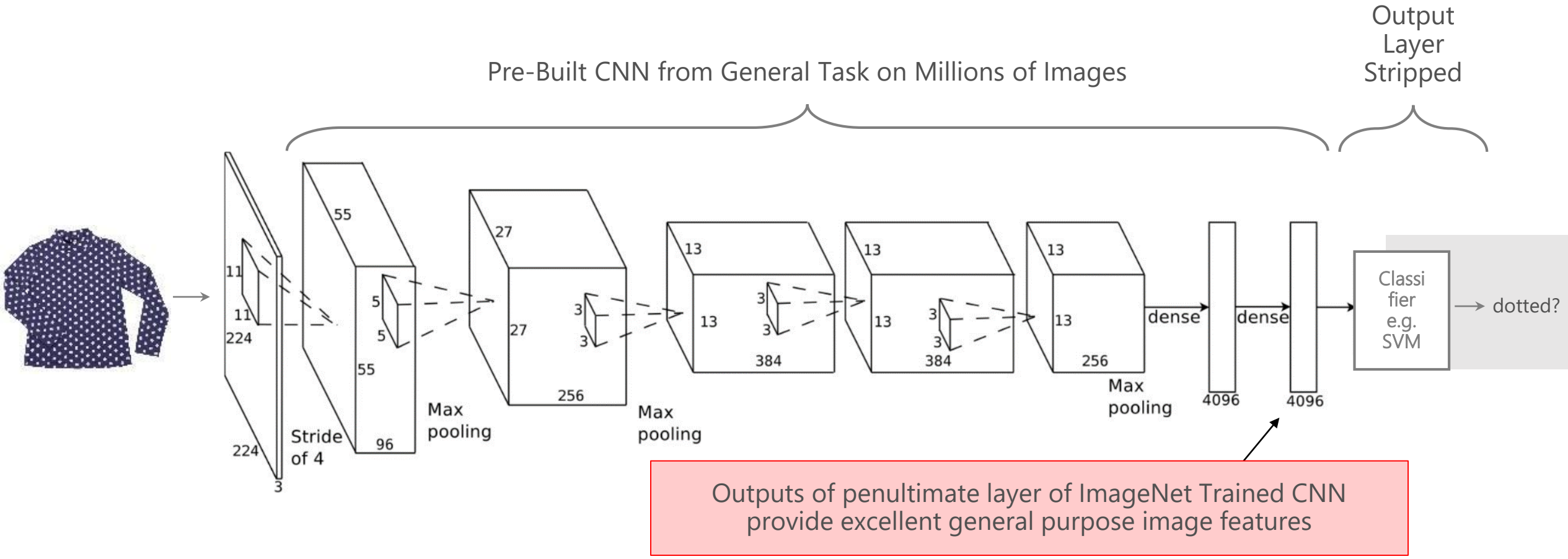


*... But that's **not enough data** to train a deep learning model!*

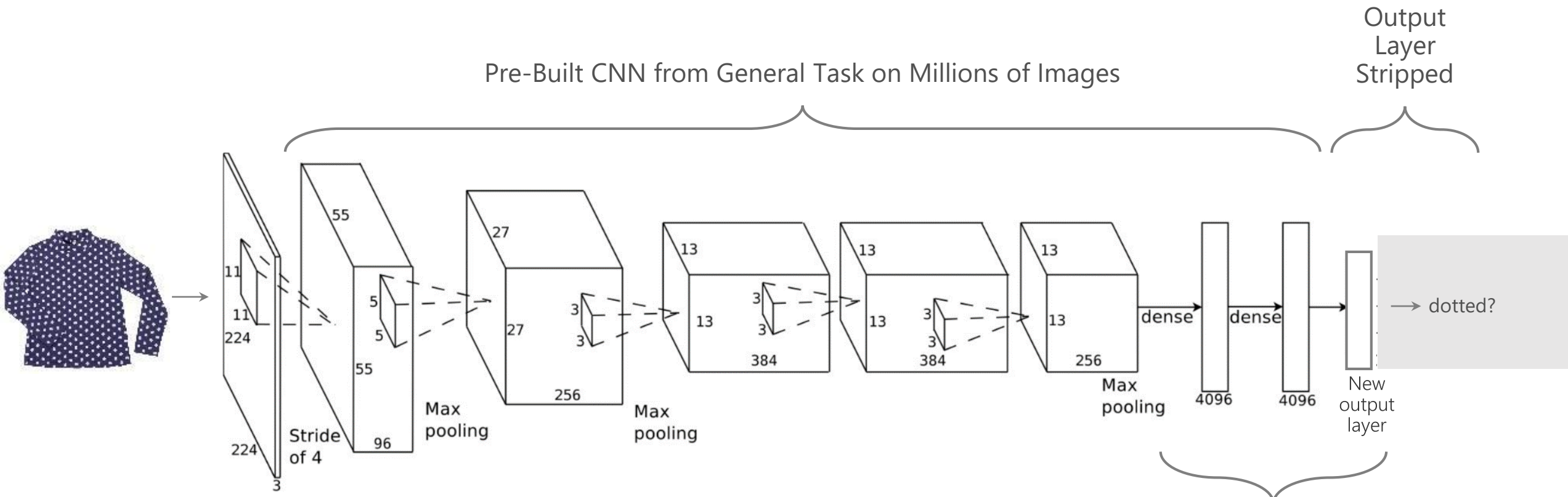
Transfer Learning – How to get started?

Type	How to Initialize Featurization Layers	Output Layer Initialization	How is Transfer Learning used?	How to Train?
Standard DNN	Random	Random	None	Train featurization and output jointly
Headless DNN	Learn using another task	Separate ML algorithm	Use the features learned on a related task	Use the features to train a separate classifier
Fine Tune DNN	Learn using another task	Random	Use and fine tune features learned on a related task	Train featurization and output jointly with a small learning rate
Multi-Task DNN	Random	Random	Learned features need to solve many related tasks	Share a featurization network across both tasks. Train all networks jointly with a loss function (sum of individual task loss function)

Using a Pre-Trained CNN as a **Featurizer**



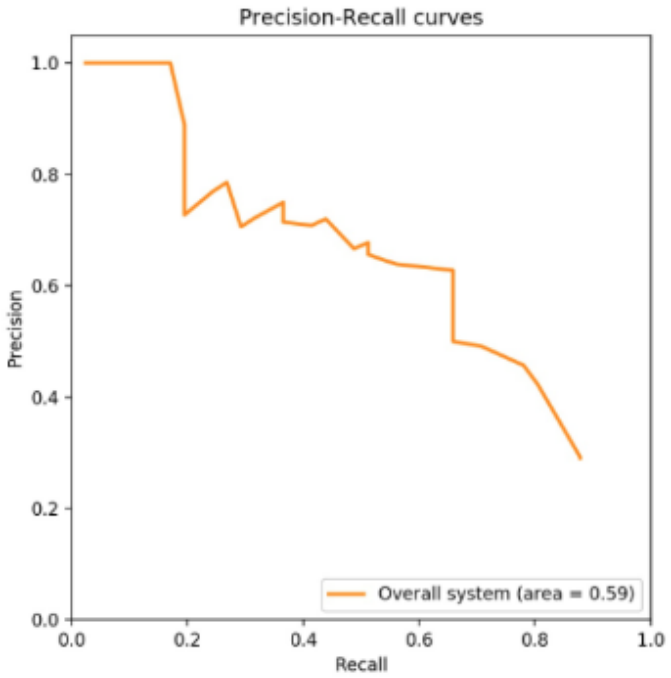
Using a Pre-Trained CNN and **Finetune**



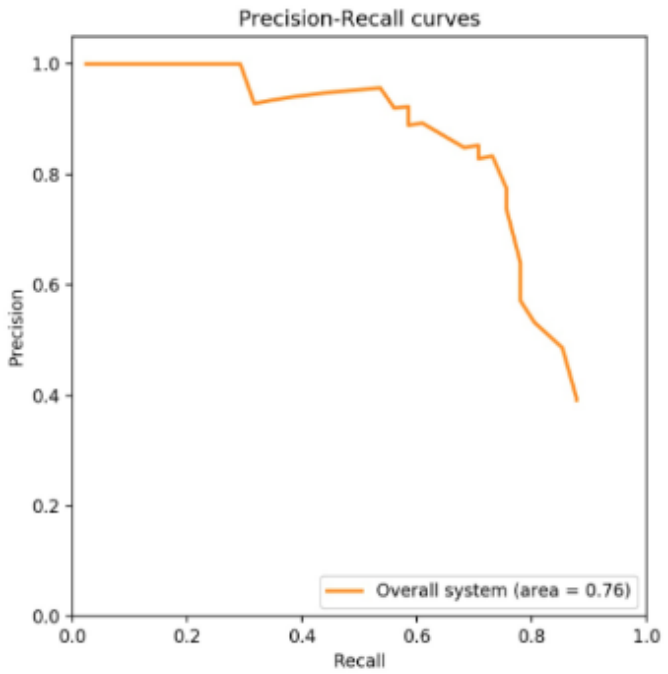
Using a pre-trained DNN, an accurate model can be achieved with thousands (or less) of labeled examples instead of millions

Transfer Learning Results - Texture Dataset

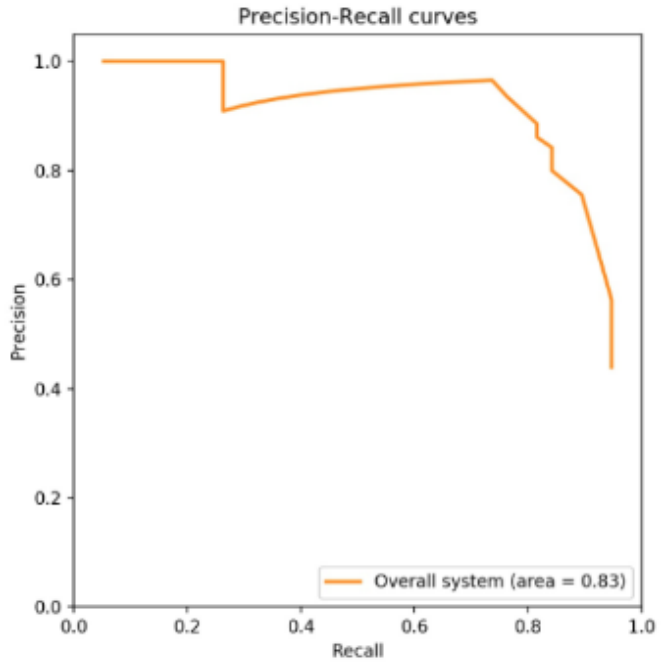
DNN featurization
Input Image Size: 224x224 pixels
Area Under Curve: 0.59
Classification Accuracy: **69.0%**



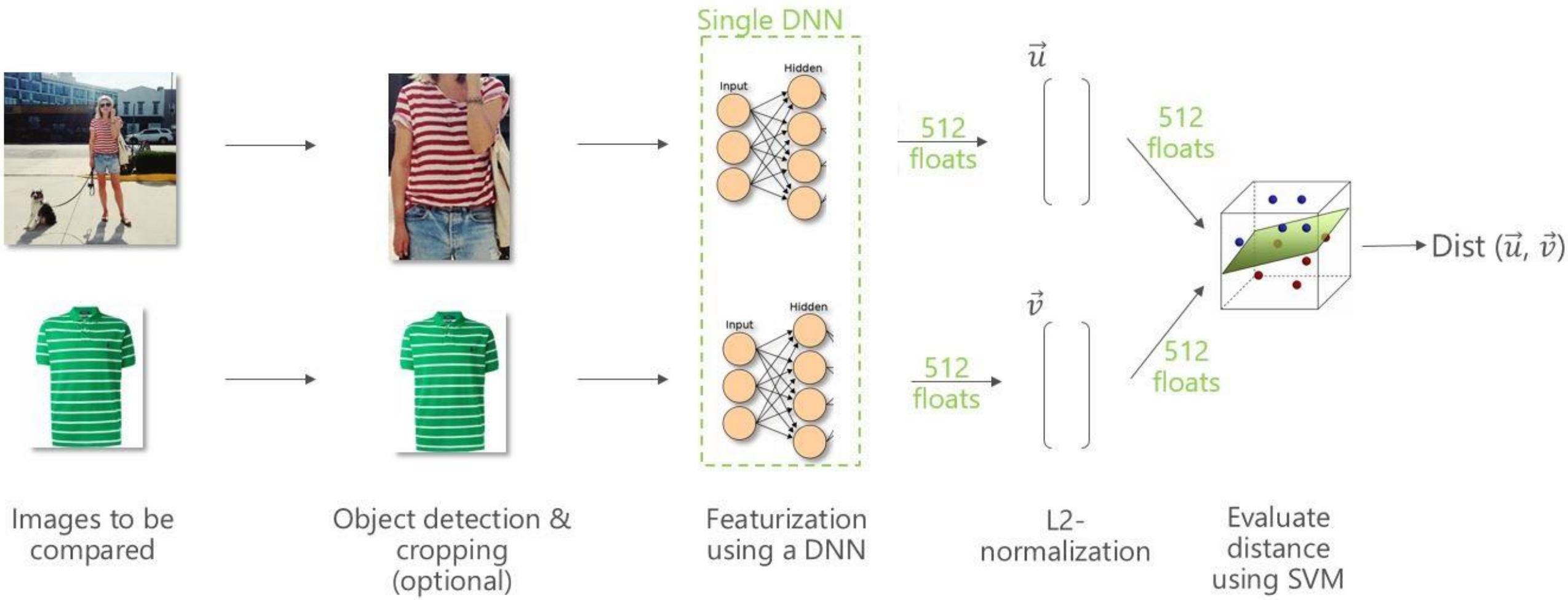
Fine-tuning (full CNN)
Input Image Size: 224x224 pixels
Area Under Curve: 0.76
Classification Accuracy: **77.4%**



Fine-tuning (full CNN)
Input Image Size: 896x886 pixels
Area Under Curve: 0.83
Classification Accuracy: **88.2%**



Transfer Learning for Similarity



Even with significant labeled data, pre-trained weights can be useful for initializing the network

Comparing Transfer Learning Approaches

- Hymenoptera, 2 classes and 397 images.
- Simpsons, 20 classes (subset of total) and 19548 images.
- Dogs vs Cats, 2 classes and 25000 images.
- Caltech 256, 257 classes and 30607 images.

['bees', 'ants', 'ants', 'bees']

['moe_szyslak', 'apu_nahasapeemapetilon', 'krusty_the_clown', 'homer_simpson']

['dogs', 'cats', 'cats', 'dogs']

['257.clutter', '246.wine-bottle', '122.kayak', '026.cake']

Full code:
https://github.com/miguelgfierro/sciblog_support/blob/master/A_Gentle_Introduction_to_Transfer_Learning/Intro_Transfer_Learning.ipynb

Finetune

```
def finetune(dataloaders, model_name, sets, num_epochs, num_gpus, lr, momentum, lr_step, lr_epochs, verbos
e=True):
    #Class adaptation
    num_class = len(dataloaders[sets[0]].dataset.class_to_idx)
    model_ft = models.__dict__[model_name](pretrained=True)
    num_ftrs = model_ft.fc.in_features
    model_ft.fc = nn.Linear(num_ftrs, num_class)

    #gpus
    if num_gpus > 1:
        model_ft = nn.DataParallel(model_ft)
    model_ft = model_ft.cuda()

    #loss
    criterion = nn.CrossEntropyLoss()

    # All parameters are being optimized
    optimizer = SGD(model_ft.parameters(), lr=lr, momentum=momentum)

    # Decay LR by a factor of lr_step every lr_epochs epochs
    exp_lr_scheduler = lr_scheduler.StepLR(optimizer, step_size=lr_epochs, gamma=lr_step)
    model_ft = train_model(dataloaders, model_ft, sets, criterion, optimizer, exp_lr_scheduler,
                           num_epochs=num_epochs, verbose=verbose)
    return model_ft
```

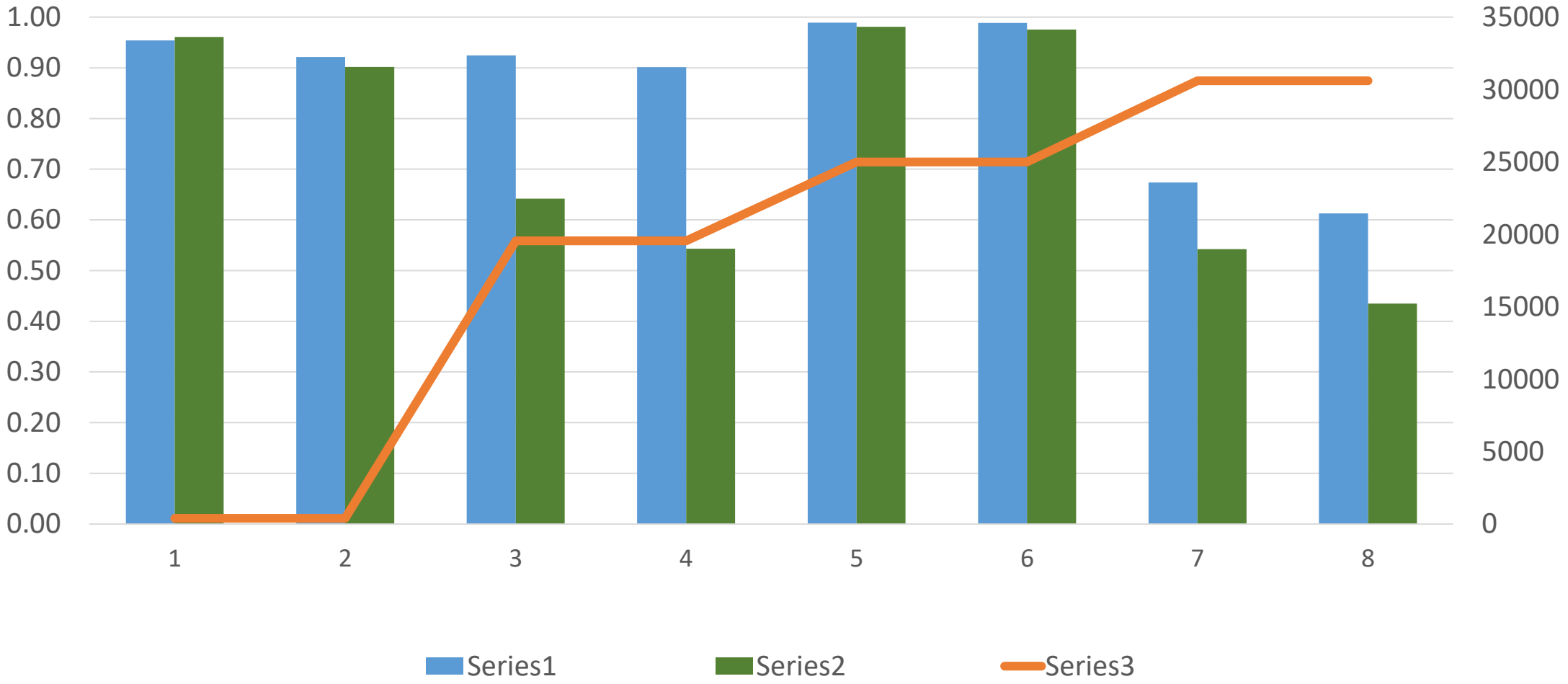
Full code:
https://github.com/miguelgfierro/sciblog_support/blob/master/A_Gentle_Introduction_to_Transfer_Learning/Intro_Transfer_Learning.ipynb

Freeze and Train

```
def freeze_and_train(dataloaders, model_name, sets, num_epochs, num_gpus, lr, momentum, lr_step, lr_epochs
, verbose=True):
    #Class adaptation
    num_class = len(dataloaders[sets[0]].dataset.class_to_idx)
    model_conv = models.__dict__[model_name](pretrained=True)
    for param in model_conv.parameters(): #params have requires_grad=True by default
        param.requires_grad = False
    num_ftrs = model_conv.fc.in_features
    model_conv.fc = nn.Linear(num_ftrs, num_class)

    #gpus
    if num_gpus > 1:
        model_conv = nn.DataParallel(model_conv)
    model_conv = model_conv.cuda()

    #loss
    criterion = nn.CrossEntropyLoss()

    # Only parameters of final layer are being optimized
    if num_gpus > 1:
        params = model_conv.module.fc.parameters()
    else:
        params = model_conv.fc.parameters()
    optimizer = SGD(params, lr=lr, momentum=momentum)

    # Decay LR by a factor of lr_step every lr_epochs epochs
    exp_lr_scheduler = lr_scheduler.StepLR(optimizer, step_size=lr_epochs, gamma=lr_step)
    model_conv = train_model(dataloaders, model_conv, sets, criterion, optimizer, exp_lr_scheduler,
                             num_epochs=num_epochs, verbose=verbose)
    return model_conv
```

Full code:
https://github.com/miguelgfierro/sciblog_support/blob/master/A_Gentle_Introduction_to_Transfer_Learning/Intro_Transfer_Learning.ipynb

Accuracy – Finetuning vs Freezing

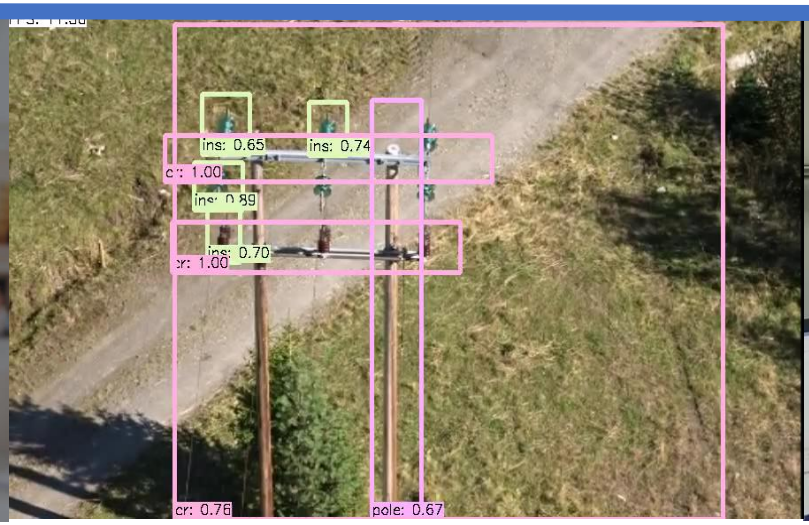


Full code:
https://github.com/miguelgfierro/sciblog_support/blob/master/A_Gentle_Introduction_to_Transfer_Learning/Intro_Transfer_Learning.ipynb

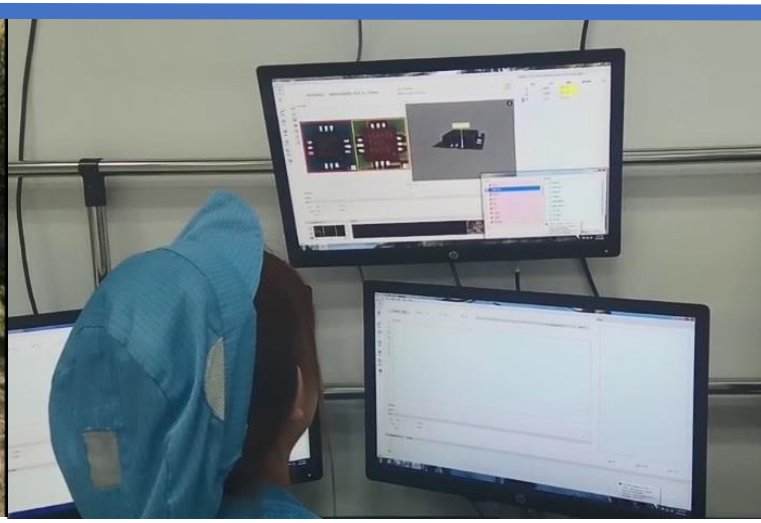
Example Applications in Computer Vision

Aerial Use Classification

Lung Cancer Detection



ESmart – Connected Drone



Jabil – Defect Inspection

Distributed deep domain adaptation for automated poacher detection

Mark Hamilton (Microsoft), Anand Raman (Microsoft)

11:05am-11:45am Friday, September 7, 2018

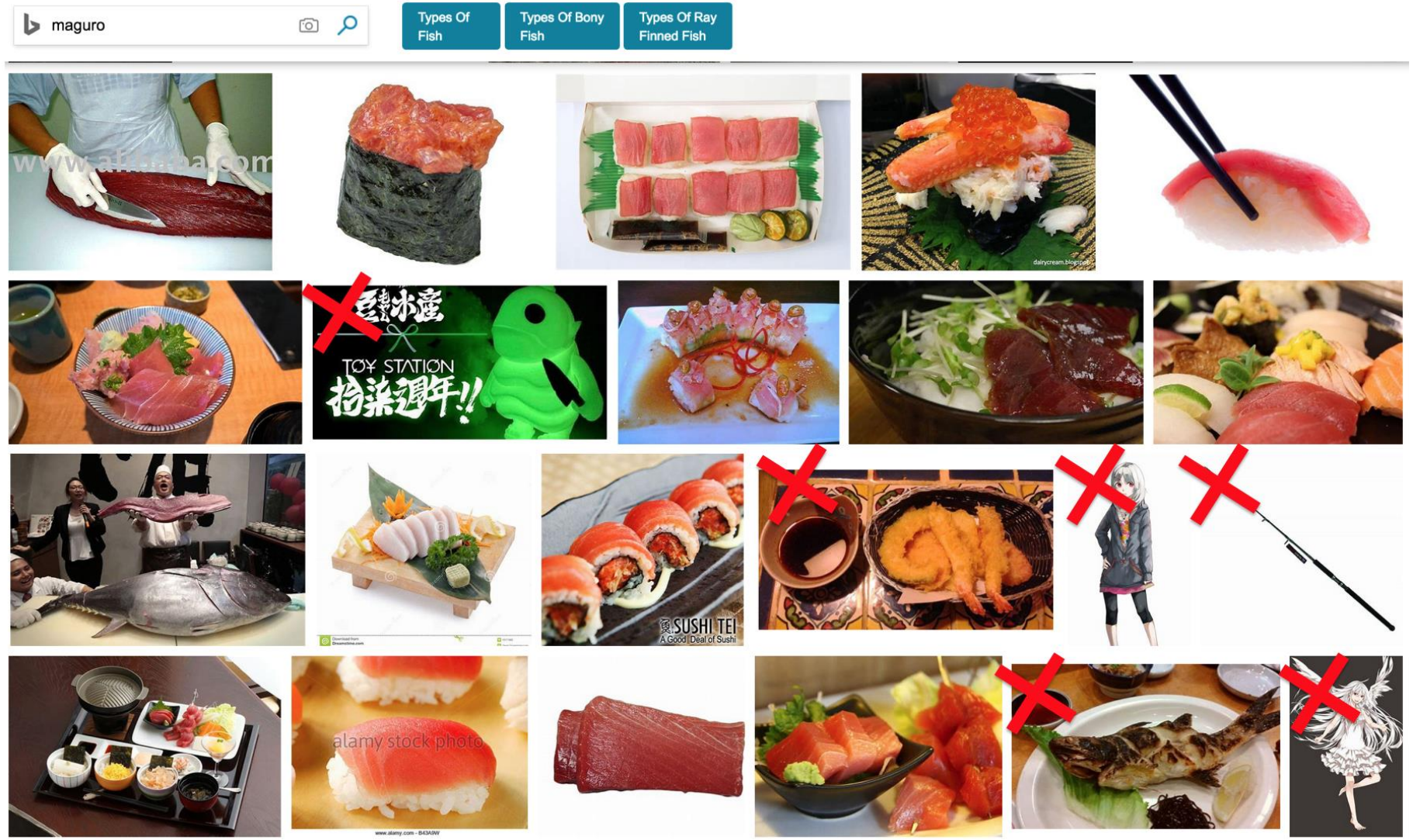
Location: Yosemite A

There are many pre-trained models available in community to leverage!

https://github.com/MattKleinsmith/void-detector

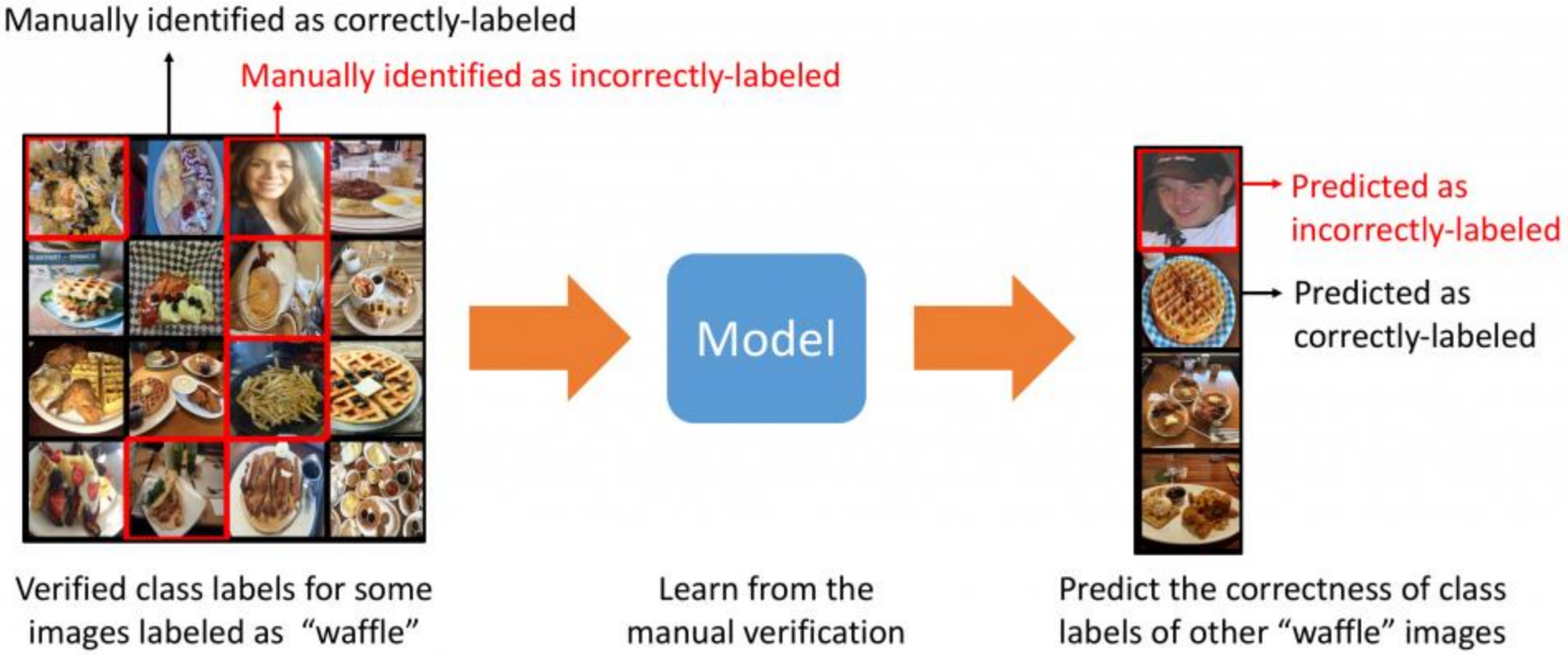
Using pre-trained models to seed annotations or improve annotation quality

Label Noise



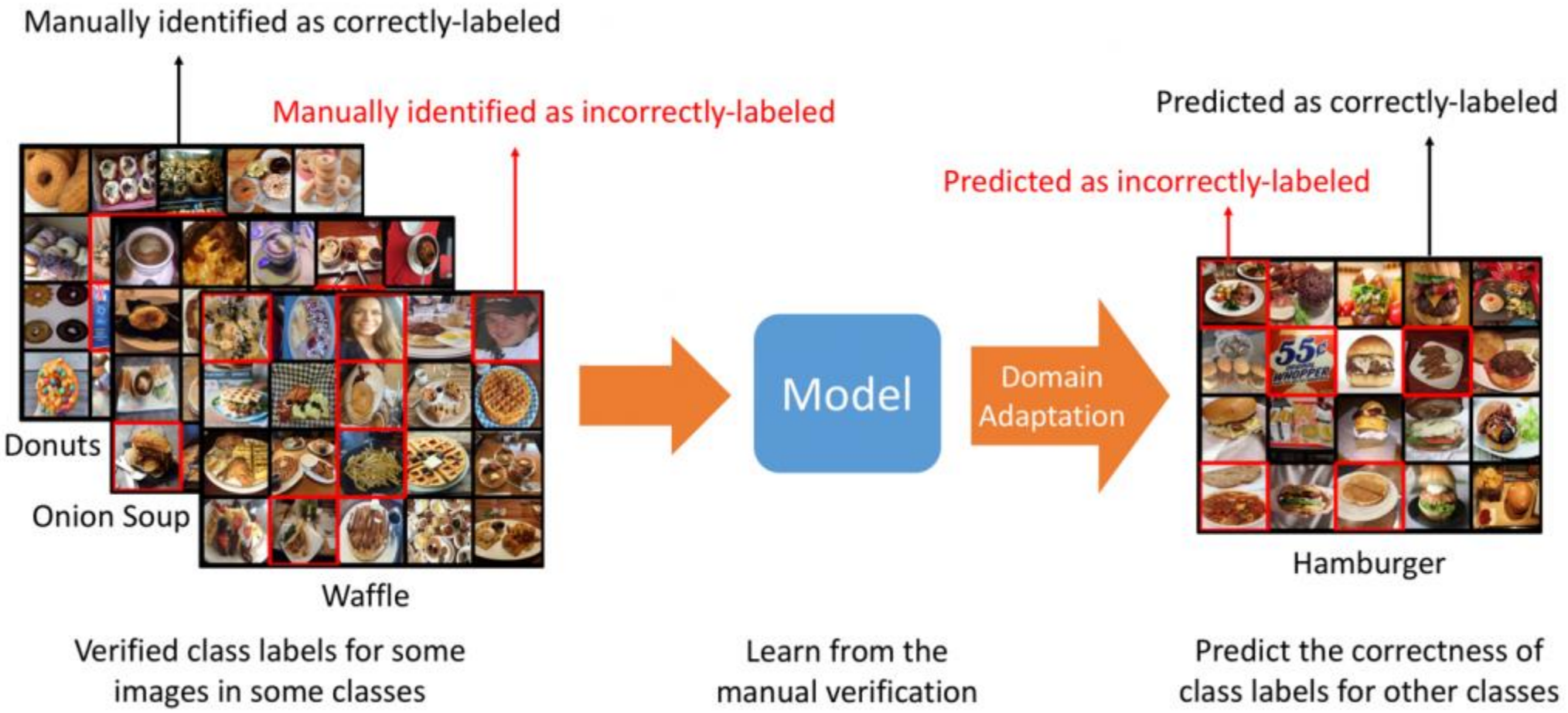
Read more details: https://www.microsoft.com/en-us/research/blog/using-transfer-learning-to-address-label-noise-for-large-scale-image-classification/

Traditional Method: Manual Verification



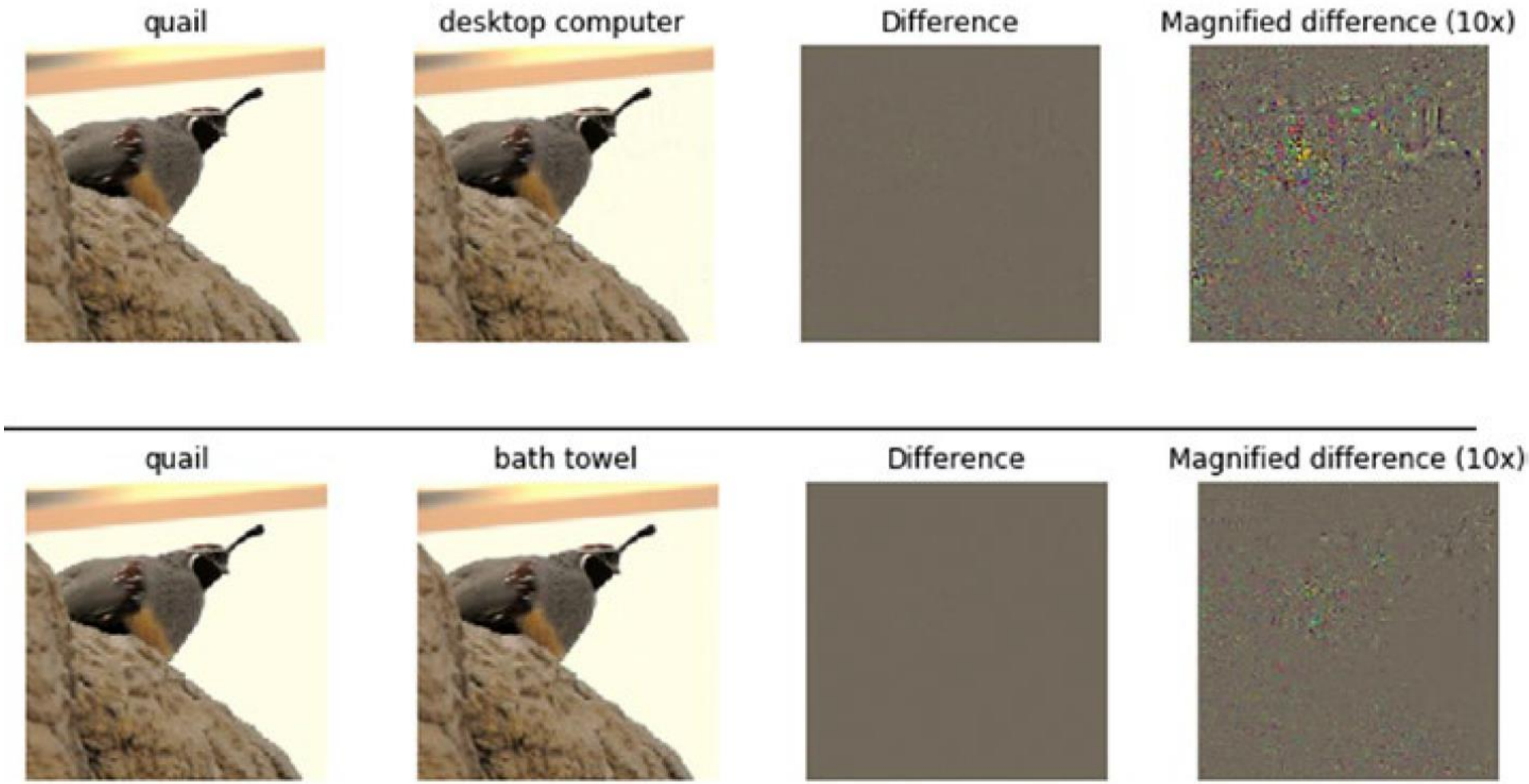
Read more details: https://www.microsoft.com/en-us/research/blog/using-transfer-learning-to-address-label-noise-for-large-scale-image-classification/

Applying Transfer Learning



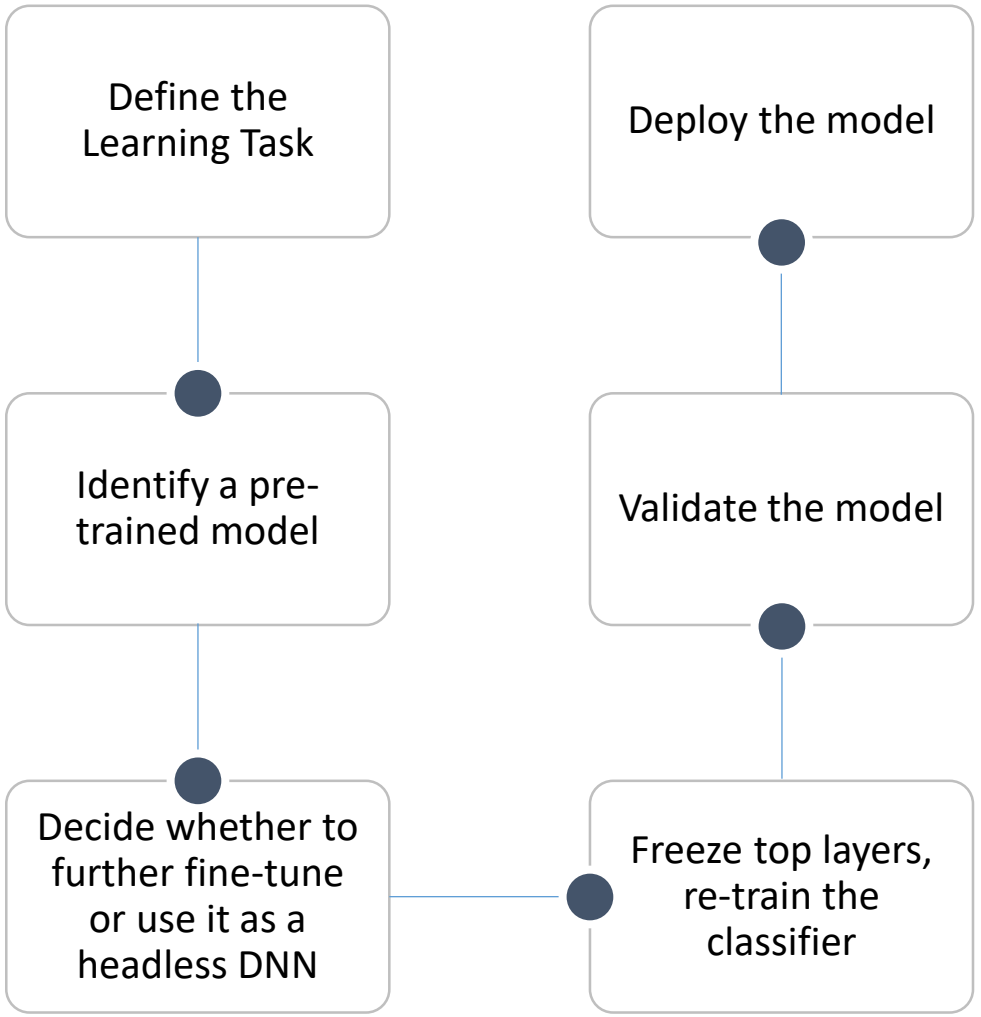
Read more details: https://www.microsoft.com/en-us/research/blog/using-transfer-learning-to-address-label-noise-for-large-scale-image-classification/

*Computer Vision is **not** a "solved problem"*

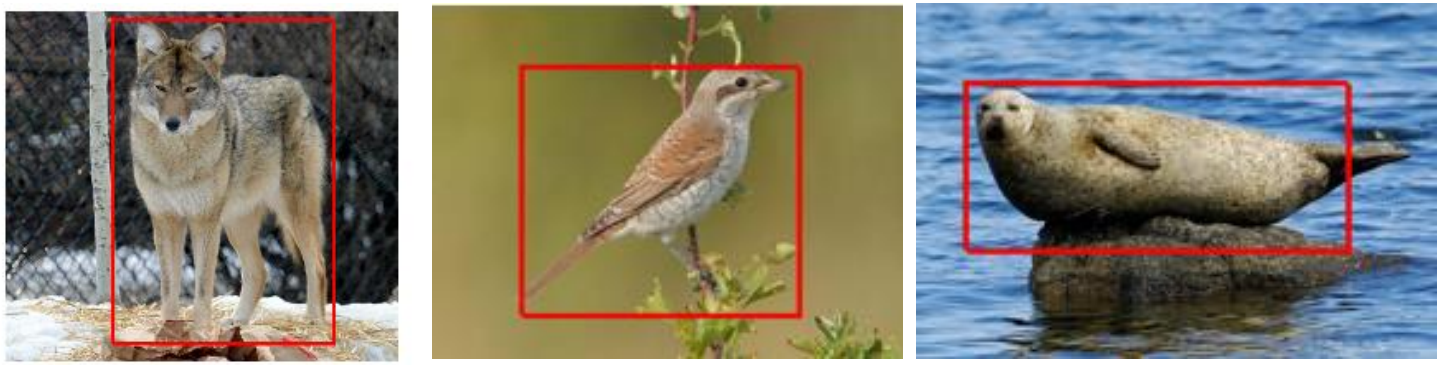


The knowledge being "transferred" can be very useful but not the same as how humans learn to see

Recap: Transfer Learning for Image Classification

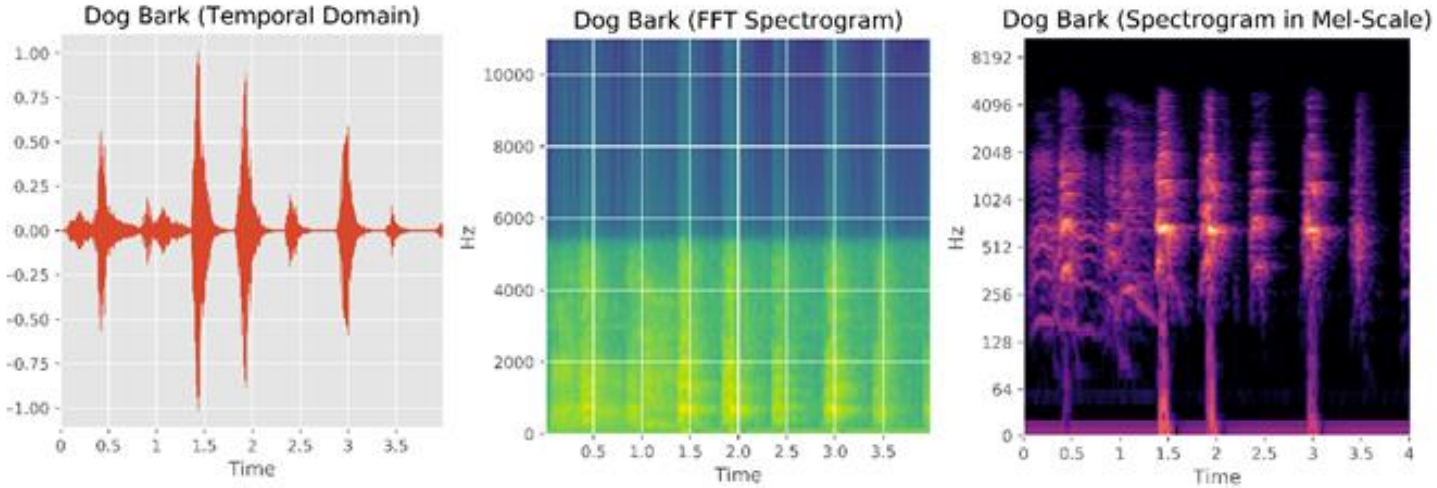


Deep Learning on Different Types of Data



Images

Rich, high-dimensional datasets



Audio Spectrograms

Rich, high-dimensional datasets

I		s	e	e		a		b	l	g		c	a	t	

Text

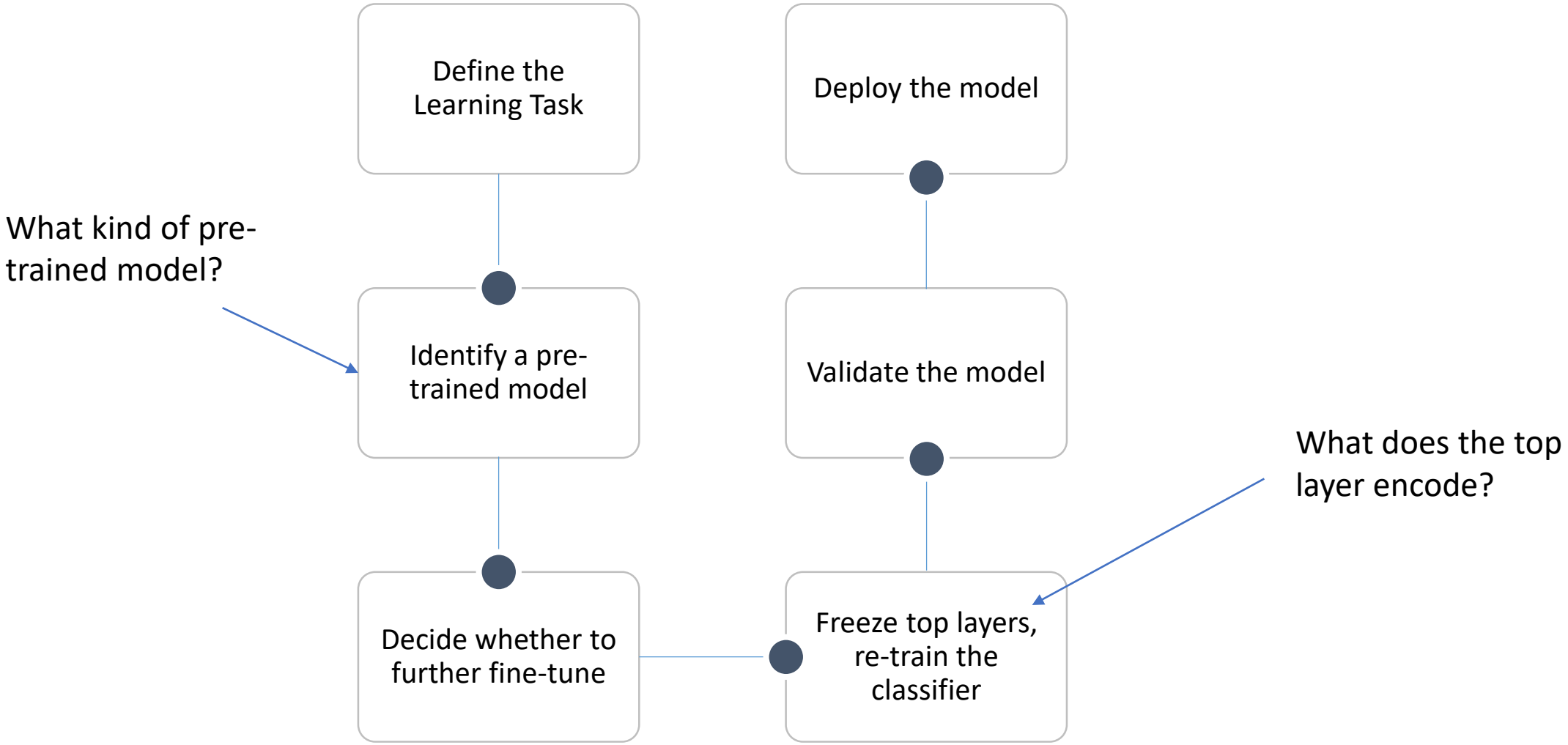
Spare data (depends on the encoding)

How do we apply Transfer Learning to NLP?

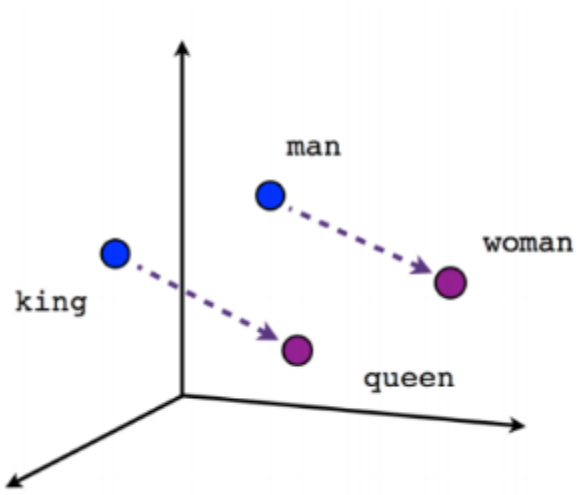
Different Type of NLP Tasks

And many more….

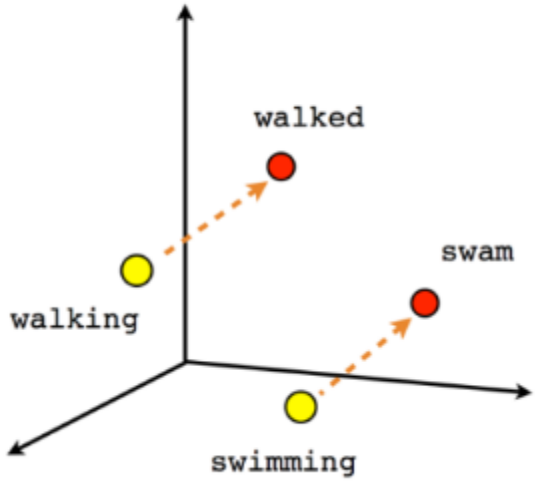
Transfer Learning for Text



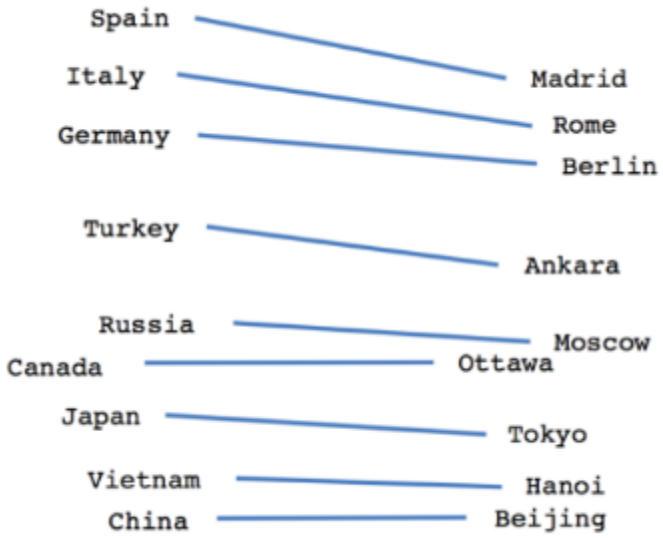
Word Embeddings



Male - Female



Verb Tense



Country - Capital

Source: Tensorflow Tutorial - https://www.tensorflow.org/tutorials/representation/word2vec

Word Embeddings

2013 2014-2015 2017

Using Pre-trained Embeddings

Text Classification using 20 Newsgroup dataset

```
embeddings_index = {}
f = open(os.path.join(GLOVE_DIR, 'glove.6B.100d.txt'))
for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    embeddings_index[word] = coefs
f.close()
```

Compute an index mapping words to known embeddings

```
embedding_matrix = np.zeros((len(word_index) + 1, EMBEDDING_DIM))
for word, i in word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        # words not found in embedding index will be all-zeros.
        embedding_matrix[i] = embedding_vector
```

Compute Embedding Matrix

Source: https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model.html

Using Pre-trained Embeddings

Text Classification using 20 Newsgroup dataset

```
from keras.layers import Embedding

embedding_layer = Embedding(len(word_index) + 1,
                            EMBEDDING_DIM,
                            weights=[embedding_matrix],
                            input_length=MAX_SEQUENCE_LENGTH,
                            trainable=False)
```

Load the Embedding Matrix into an **Embedding Layer**

Prevent weights from being updated during training

Source: https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model.html

Using Pre-trained Embeddings

Text Classification using 20 Newsgroup dataset

```
sequence_input = Input(shape=(MAX_SEQUENCE_LENGTH,), dtype='int32')
embedded_sequences = embedding_layer(sequence_input)
x = Conv1D(128, 5, activation='relu')(embedded_sequences)
x = MaxPooling1D(5)(x)
x = Conv1D(128, 5, activation='relu')(x)
x = MaxPooling1D(5)(x)
x = Conv1D(128, 5, activation='relu')(x)
x = MaxPooling1D(35)(x)  # global max pooling
x = Flatten()(x)
x = Dense(128, activation='relu')(x)
preds = Dense(len(labels_index), activation='softmax')(x)

model = Model(sequence_input, preds)
model.compile(loss='categorical_crossentropy',
              optimizer='rmsprop',
              metrics=['acc'])
model.fit(x_train, y_train, validation_data=(x_val, y_val),
          epochs=2, batch_size=128)
```

Build a small 1D convnet to solve the classification problem

Source: https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model.html

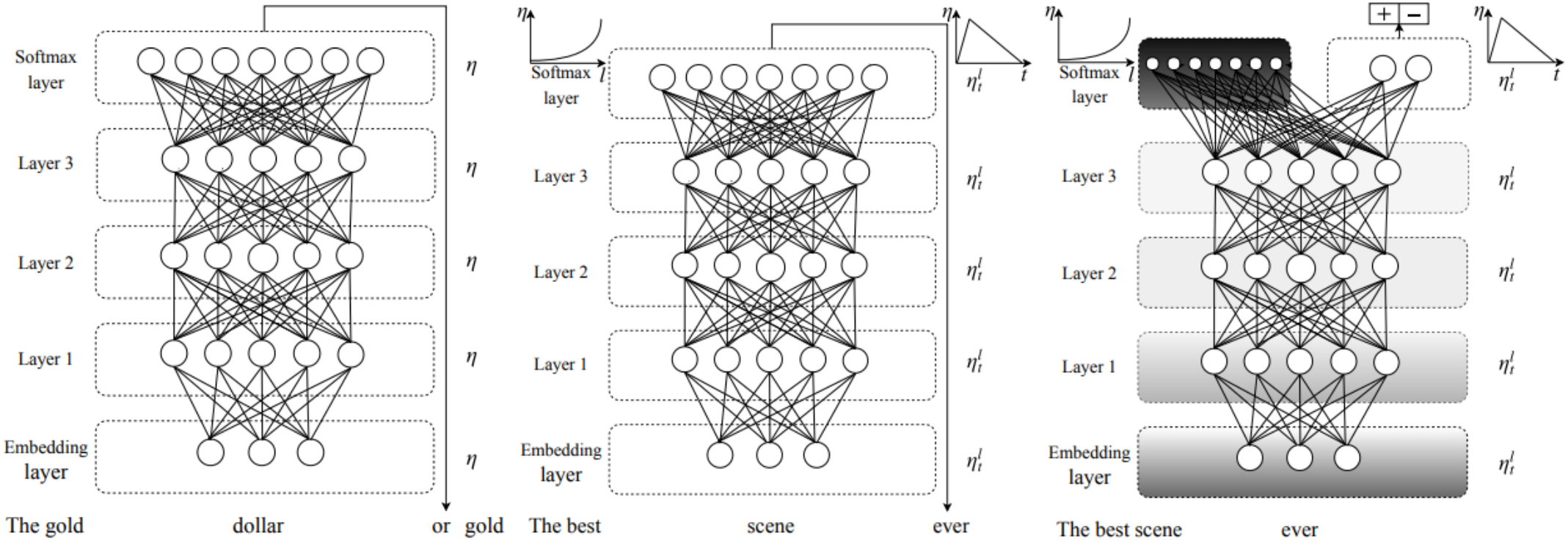
From **initializing the first layers** to **pre-training the entire model**
(and learning higher level semantic concepts)

Transfer Learning for NLP - **ULMFiT**

Train a Language Model using Large General Domain Corpus

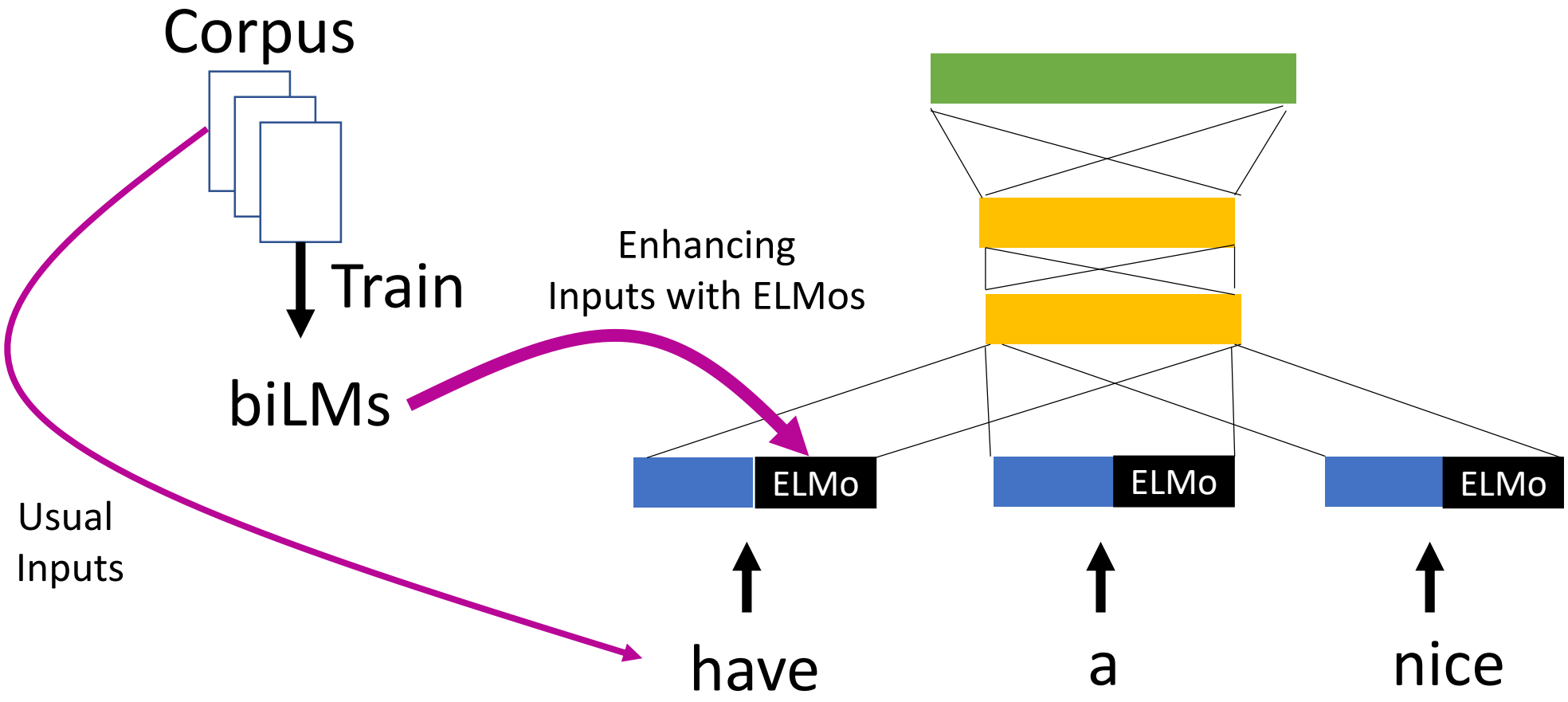
Fine-tune the Language Model

Fine-tune Classifier



Source: Universal Language Model Fine-tuning for Text Classification, Jeremy Howard, Sebastian Ruder, ACL 2018

Transfer Learning for NLP - **ELMo**



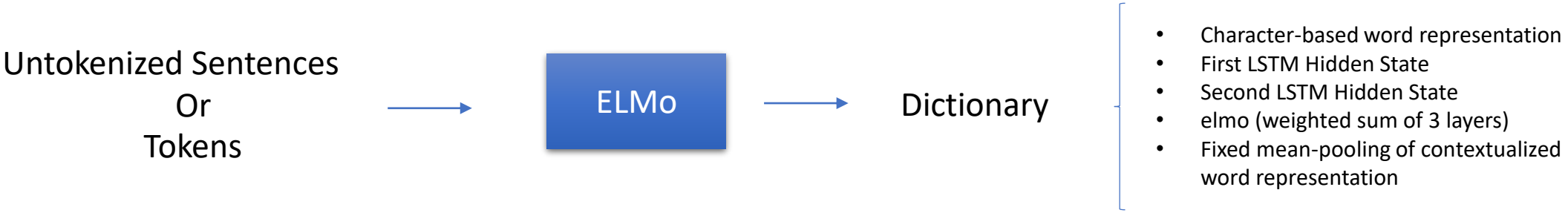
Source: Deep contextualized word representations, Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer., NAACL 2018

ELMo Pre-trained Models

Model	#Parameters (Millions)	LSTM Hidden Size	Output Size	# Highway Layers	SRL F1	Constituency F1
Small	13.6	1024	128	1	83.62	93.12
Medium	28.0	2048	256	1	84.04	93.60
Original	93.6	4096	512	2	84.63	93.85
Original (5.5B)	93.6	4096	512	2	84.93	94.01

Source: https://allennlp.org/elmo

Using ELMo with TensorFlow Hub




```
elmo = hub.Module("https://tfhub.dev/google/elmo/2",
trainable=True)
embeddings = elmo(
    ["the cat is on the mat", "dogs are in the fog"],
    signature="default",
    as_dict=True)["elmo"]
```

```
elmo = hub.Module("https://tfhub.dev/google/elmo/2", trainable=True)
tokens_input = [["the", "cat", "is", "on", "the", "mat"],
                ["dogs", "are", "in", "the", "fog", ""]]
tokens_length = [6, 5]
embeddings = elmo(
    inputs={
        "tokens": tokens_input,
        "sequence_len": tokens_length
    },
    signature="tokens",
    as_dict=True)["elmo"]
```

Source: https://www.tensorflow.org/hub/modules/google/elmo/2

Transfer Learning for MRC tasks

Source:

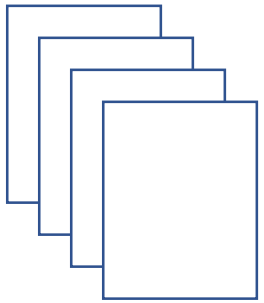
Transfer Learning for Machine Reading Comprehension - https://bit.ly/2Cmiffy

Transfer Learning for **MRC**

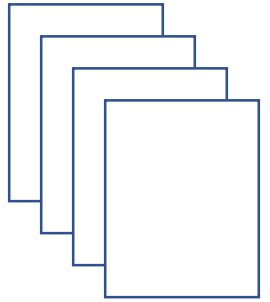
Train a model using data from WikiPedia

MRC Model

Apply the model to other domains



News Articles



Customer Support Data

SQUAD

Stanford Question Answering Dataset (SQuAD)
Reading comprehension dataset

Based on Wikipedia articles
Crowdsource questions

Answer is Text Segment, or span, from the corresponding reading passage, or the no answers found.

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

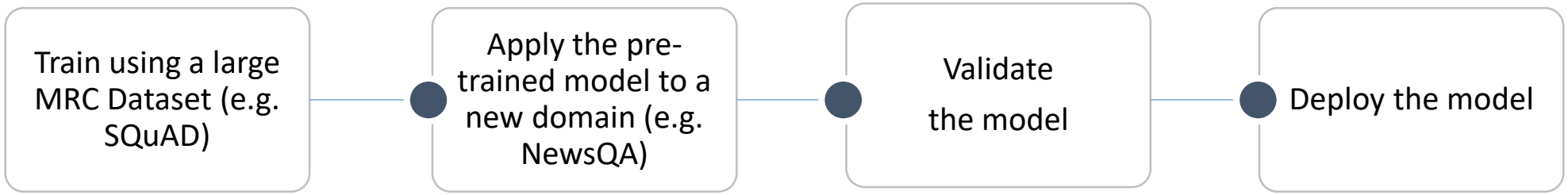
Where do water droplets collide with ice crystals to form precipitation?
within a cloud

Question Answer Pairs

MRC Datasets

Dataset	Question source	Formulation	Size
SQuAD	**crowdsourced**	**RC, spans in passage**	**100K**
MCTest (Richardson et al., 2013)	crowdsourced	RC, multiple choice	2640
Algebra (Kushman et al., 2014)	standardized tests	computation	514
Science (Clark and Etzioni, 2016)	standardized tests	reasoning, multiple choice	855
WikiQA (Yang et al., 2015)	query logs	IR, sentence selection	3047
TREC-QA (Voorhees and Tice, 2000)	query logs + human editor	IR, free form	1479
CNN/Daily Mail (Hermann et al., 2015)	summary + cloze	RC, fill in single entity	1.4M
CBT (Hill et al., 2015)	cloze	RC, fill in single word	688K

Transfer Learning for MRC using SynNet



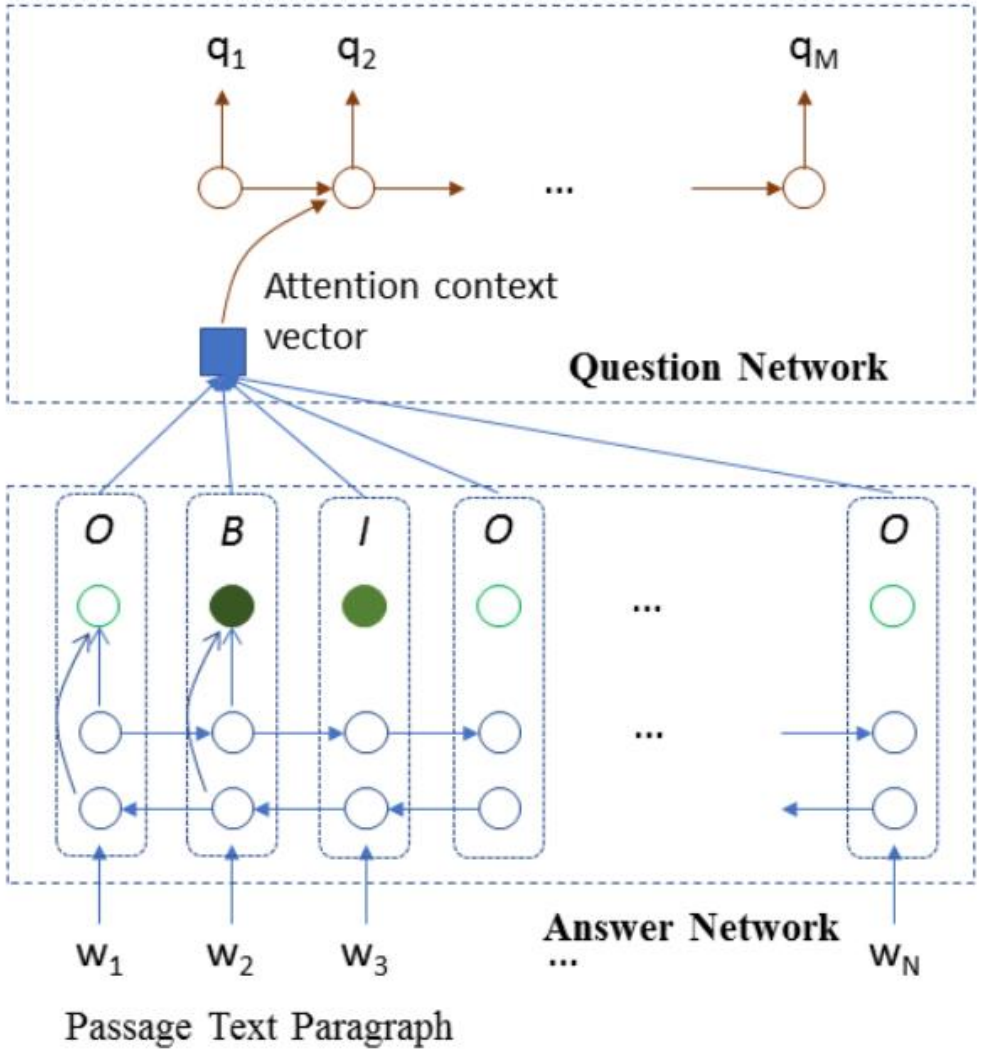
More comparisons between different MRC Approaches

Transfer Learning for MRC –Survey - https://bit.ly/2JAt1h0

SynNet

Stage 1- Answer Synthesis module uses a bi-directional LSTM to predict IOB tags on the input paragraph. Marks out semantic concept that are likely answer

Stage 2 – Question Synthesis module uses a uni-directional LSTM to generate the questions



Source: ACL 2017, https://www.microsoft.com/en-us/research/publication/two-stage-synthesis-networks-transfer-learning-machine-comprehension/

SynNet – Question/Answer Generation Example

Article: San Francisco , California (CNN} -- Repair work on the San Francisco-Oakland Bay Bridge will continue nonstop into the weekend and the bridge may reopen Monday, but officials were making no promises Friday. "Commuters are going to need to check back with us over the weekend," said Bart Ney, a spokesman for the California Department of Transportation. "We're going to do everything we can to get the bridge open for the Monday morning commute, but safety is the priority for us right now." Repair work has not stopped since it began Tuesday night when two steel rods and a steel crossbeam plummeted from the bridge, landing on the roadway and forcing the span 's closure. The same section had been the site of repairs over Labor Day weekend, when crews fixed a crack.

Synthesized question: What is the San Francisco Rapid Transit open for ?

Synthesized answer: the Monday morning commute

Article: JAKARTA, Indonesia (CNN) -- Five Europeans rescued Saturday after an Indonesia diving trip went wrong had to fight off a Komodo dragon while they were waiting to be found, according to reports. Rescued diver Kath Mitchinso embraces fellow diver Ernest Lewandowsky as they arrive on Flores island. The group was found at Mantaolan, on the island of Rinca off the Komodo National Park, after going missing Thursday. The divers -- three Britons, a Frenchman and a Swede -- spent two nights on the deserted island, which is home to the large Komodo dragon, before rangers found them Saturday.

Synthesized question: How many Britons were rescued?

Synthesized answer: three

How to use transfer learning to bootstrap image classification and question answering (QA)

Summary

1. Transfer Learning and Applications
2. How to use Transfer Learning for Image Classification
3. How to use Transfer Learning for NLP tasks

Thank You!

How to use transfer learning to bootstrap image classification and question answering (QA)

Danielle Dean PhD, Wee Hyong Tok PhD
Principal Data Scientist Lead
Microsoft

@danielleodean | @weehyong