

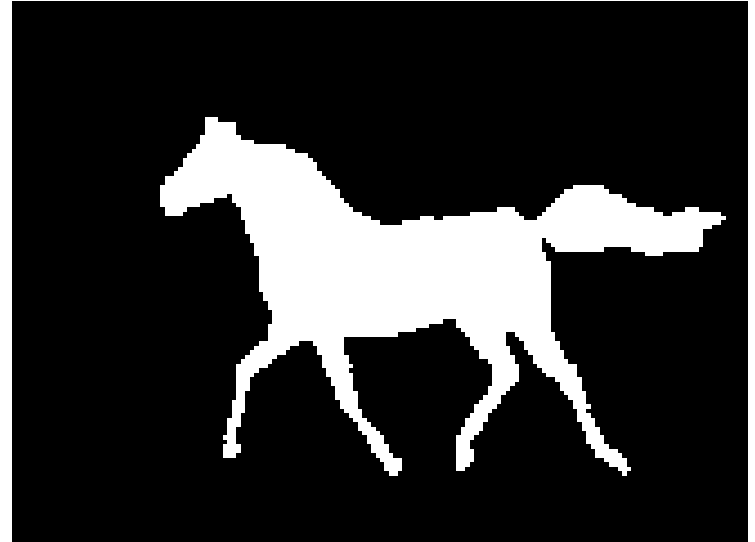
# Learning With Less Data: Active, Semi-Supervised, and Self-Supervised Learning

Rishabh Iyer

University of Texas at Dallas

- We're given lots and lots of labelled examples
  - Goal is to predict the label of unseen examples
  - Observations:
    - We don't necessarily need that many data points to construct a good classifier (think SVMs)
    - In certain applications, labels are *expensive*
      - They can cost time, money, or other resources

# Image Segmentation



Someone (probably a graduate student) had to produce these labels by hand!

- In general, data is easy to come by but labels are expensive
  - Labelled speech
  - Labelled images and video
  - Large corpora of texts
- These tasks are mind numbing and boring
  - Can pay people to do them! (Amazon Mechanical Turk)
  - Can get expensive fast and we need some way to ensure that they are accurately solving the problem or else we are wasting money!

- Given lots of unlabeled examples
  - Learn to predict the label of unseen data points
  - The added feature: we have the ability to ask for the label of any one of the unlabeled inputs (e.g., a labelling oracle/expert)
    - Treat asking the oracle for a label as an expensive operation
    - The performance of the algorithm will be judged by how few queries it can make to learn a good classifier

- Suppose that we want to determine what disease a patient has
  - We can run a series of (possibly expensive) tests in order to determine the correct diagnosis
  - How should we choose the tests so as to minimize cost (dollars and life) while still guaranteeing that we come up with the correct diagnosis?

# A First Attempt



- Could just randomly pick an unlabeled data point
  - Request its label
  - Add it to the training data
  - Retrain the model
  - Repeat
- If labels are really expensive, can be a terrible idea
  - Many unlabeled data points may have very little impact on the predicted labels
  - This is effectively the supervised setting

# A Motivating Example



- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?





# A Motivating Example



- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?



# A Motivating Example



- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?



# A Motivating Example



- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?



# A Motivating Example



- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?



# A Motivating Example



- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?



# A Motivating Example



- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?



Ideal case: number of hypotheses consistent with the labeling is approximately halved at each step

# Types of Active Learning



- Pool based
  - We're given all of the unlabeled data upfront
- Streaming
  - Unlabeled examples come in one at a time and we have to decide whether or not we want to label them as they arrive
  - Also applies to applications in which storing all the data is not possible

- Iteratively build a model
- Use the current model to find “informative” unlabeled examples
- Select the most informative example(s)
  - Label them and add them to the training data
- Retrain the model using the new training data
- Repeat



- Iteratively build a model
- Use the current model to find “informative” unlabeled examples
- Select the most informative example(s)
  - Label them and add them to the training data
- Retrain the model using the new training data
- Repeat

Note: this procedure will result in a biased sampling of the underlying distribution in general (the actively labeled dataset is not reflective of the underlying data generating process)

- For learning algorithms that model the data generating process...
  - A data point is informative if the current model is not confident in its prediction for this example
  - Least confident labeling (binary label case):

$$\arg \max_{x \text{ unlabeled}} 1 - \max_y p(y|x, \theta)$$

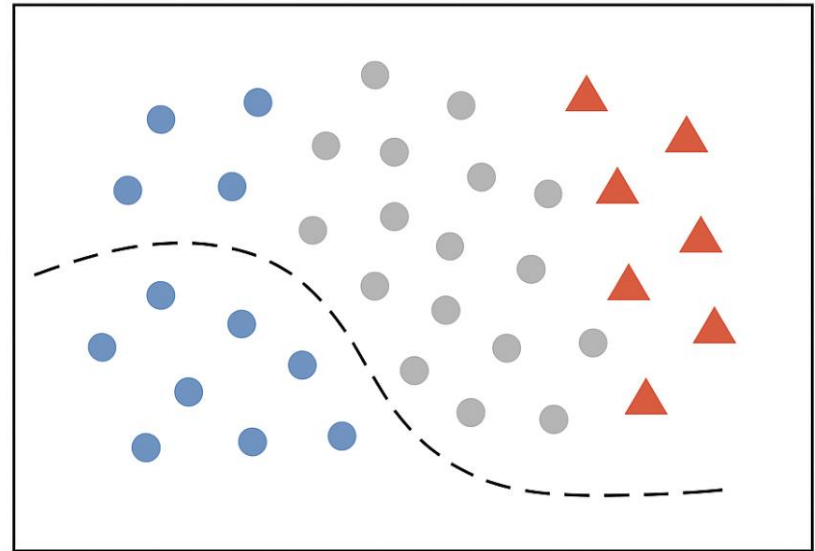
- For learning algorithms, like SVMs, that are simply selecting among a collection of hypotheses...
  - Unlabeled data points that are far from the current decision boundary are unlikely to provide useful information

- Select a committee of  $T$  consistent classifiers using the labeled data
- Find examples for which the committee has the largest disagreement
  - For example, in a binary labeling problem, find the examples for which the committee's votes are split as close to 50/50 as possible between +1 and -1
- Request the label for these examples

Goal: reduce the version space as much as possible by selecting points whose label will eliminate the most hypotheses

- Key Idea:
  - Maintain a **committee of diverse models** (e.g., via different initializations, subsets of data, or architectures).
  - For each unlabeled example, evaluate how much the models **disagree** in their predictions.
  - **Select examples with the highest disagreement** to query for labels from the oracle (e.g., a human annotator).
- Why it Works?
  - High disagreement implies **high model uncertainty**.
  - Labeling such samples **reduces hypothesis space** more efficiently.
  - Leads to faster learning with **fewer labeled samples**.

- Given a collection of labeled and unlabeled data, use it to build a model to predict the labels of unseen data points
  - We never get to see the labels of the unlabeled data
  - However, if we assume something about the data generating process, the unlabeled data can still be useful...



# Semi-Supervised Learning



## Core Idea:

- Semi-supervised learning sits between **supervised** and **unsupervised learning**, leveraging a **small set of labeled data** + a **large set of unlabeled data** to improve learning.

## Why It Matters:


- Labeling data is **expensive and slow**
- Unlabeled data is **abundant and cheap**
- SSL bridges the gap by exploiting structure in the data distribution

# Key Assumptions in SSL



- **Smoothness Assumption:** Close points likely share the same label
- **Cluster Assumption:** Data forms clusters; points in the same cluster likely share a label
- **Manifold Assumption:** Data lies on a lower-dimensional manifold

## 1. Pseudo-Labeling:

- Use the model to assign “pseudo-labels” to unlabeled data
- Retrain model using both true + confident pseudo-labels
-  Repeat iteratively

## 2. Consistency Regularization:

- Add a loss that encourages predictions to be **stable** under small input perturbations
- Ex: If an image is flipped or augmented, the model should still predict the same label



## 3. Graph-Based SSL:

- Represent data as a graph (nodes = samples, edges = similarity)
- Propagate labels from labeled to unlabeled nodes

## 4. Entropy Minimization & Confidence-Based Filtering:

- Prefer confident predictions; penalize uncertain outputs
- Can be combined with pseudo-labeling or consistency

## What is Self-Supervised Learning?

**Self-Supervised Learning** is a machine learning paradigm where the model **learns useful representations from unlabeled data** by solving **pretext tasks** — tasks created from the data itself — without requiring manual labels.

---

## Key Idea:

Use the data's **inherent structure** to generate supervisory signals.

This allows the model to "**supervise itself**" using clever surrogate tasks.

# Self Supervised Learning

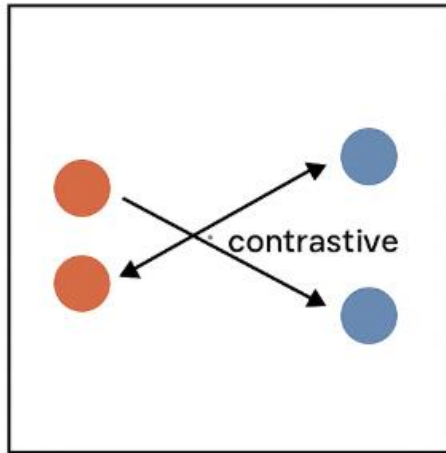


## Examples of Pretext Tasks

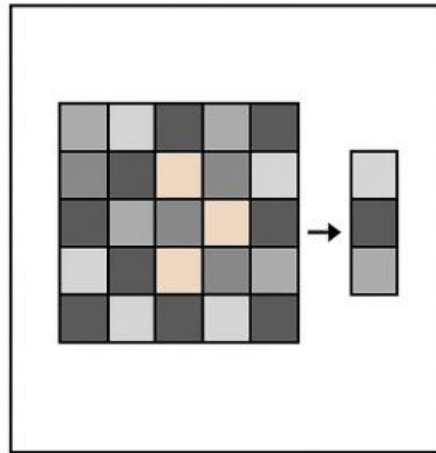
Modality	Pretext Task	Goal
Images	Predict rotation angle (0°, 90°, 180°, 270°)	Learn spatial understanding
Images	Solve jigsaw puzzle	Learn part-whole relationships
Images	Masked patch prediction (like MAE or SimMIM)	Learn local-global context
Text (NLP)	Masked Language Modeling (e.g., BERT)	Predict missing words
Text (NLP)	Next Sentence Prediction	Understand sentence relationships
Video	Predict future frames	Learn temporal consistency
Audio	Contrastive learning between audio segments	Learn speaker or content invariance

- **Contrastive Learning:** Learn representations by pulling similar examples together and pushing dissimilar ones apart  
e.g., SimCLR, MoCo, BYOL, DINO
- **Masked Modeling:** Learn to reconstruct missing parts  
e.g., BERT (text), MAE (vision)
- **Clustering-Based:** Learn to group similar instances  
e.g., DeepCluster, SwAV

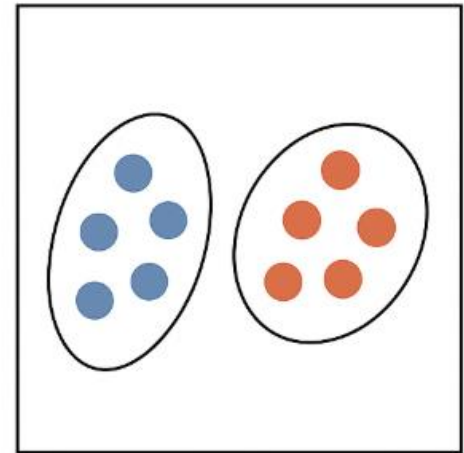
# Self-Supervised Approaches



**Contrastive  
Learning**



**Masked  
Modeling**



**Clustering-  
Based**

**Core Idea:** Learn a **shared embedding space** for multiple modalities (like images and text) so they can be compared **directly**.

*“Connect images and their descriptions without ever needing labeled categories.”*

# Contrastive Image Language Pretraining

Pre-trained on 400M (image, text) pairs from the web

Learns to **align images with corresponding text** using contrastive loss

Each modality has its own encoder (ViT/CNN for image, Transformer for text)

- Pulls together correct (image, caption) pairs, pushes apart incorrect ones

# Contrastive Image Language Pretraining

Pre-trained on 400M (image, text) pairs from the web

Learns to **align images with corresponding text** using contrastive loss

Each modality has its own encoder (ViT/CNN for image, Transformer for text)

- Pulls together correct (image, caption) pairs, pushes apart incorrect ones

## Why it Works:

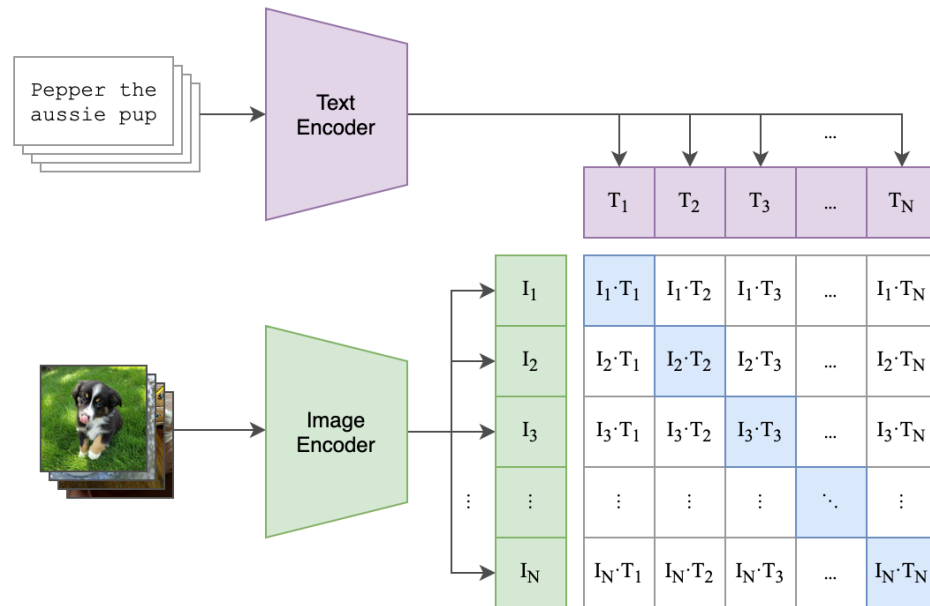
- **No need for labeled classes** (works zero-shot!)
- Learns **semantic understanding** across modalities
- Generalizes to new tasks via **prompting**



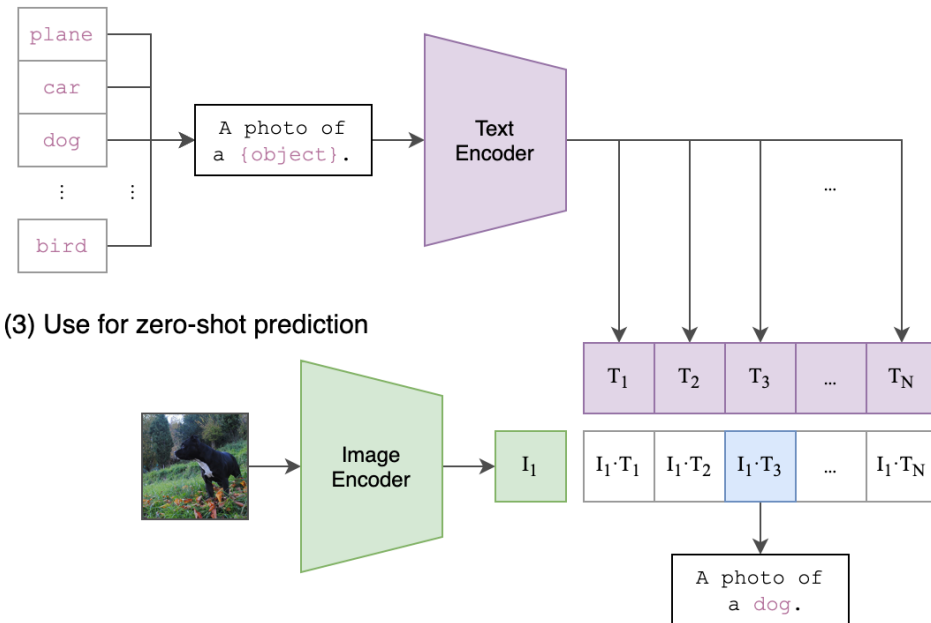
# CLIP Architecture



(1) Contrastive pre-training



(2) Create dataset classifier from label text



# CLIP in Action



## Food101

**guacamole** (90.1%) Ranked 1 out of 101 labels



- ✓ a photo of **guacamole**, a type of food.
- ✗ a photo of **ceviche**, a type of food.
- ✗ a photo of **edamame**, a type of food.
- ✗ a photo of **tuna tartare**, a type of food.
- ✗ a photo of **hummus**, a type of food.

## SUN397

**television studio** (90.2%) Ranked 1 out of 397 labels



- ✓ a photo of a **television studio**.
- ✗ a photo of a **podium indoor**.
- ✗ a photo of a **conference room**.
- ✗ a photo of a **lecture room**.
- ✗ a photo of a **control room**.

## Youtube-BB

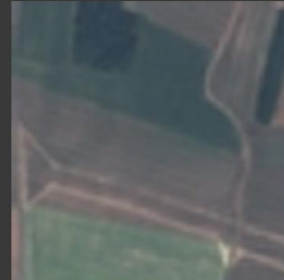
**airplane, person** (89.0%) Ranked 1 out of 23 labels



- ✓ a photo of a **airplane**.
- ✗ a photo of a **bird**.
- ✗ a photo of a **bear**.
- ✗ a photo of a **giraffe**.
- ✗ a photo of a **car**.

## EuroSAT

**annual crop land** (46.5%) Ranked 4 out of 10 labels



- ✗ a centered satellite photo of **permanent crop land**.
- ✗ a centered satellite photo of **pasture land**.
- ✗ a centered satellite photo of **highway or road**.
- ✓ a centered satellite photo of **annual crop land**.
- ✗ a centered satellite photo of **brushland or shrubland**.

Please evaluate the course!

[eval.utdallas.edu](https://eval.utdallas.edu)