KPMG

Active Learning for Computer Vision

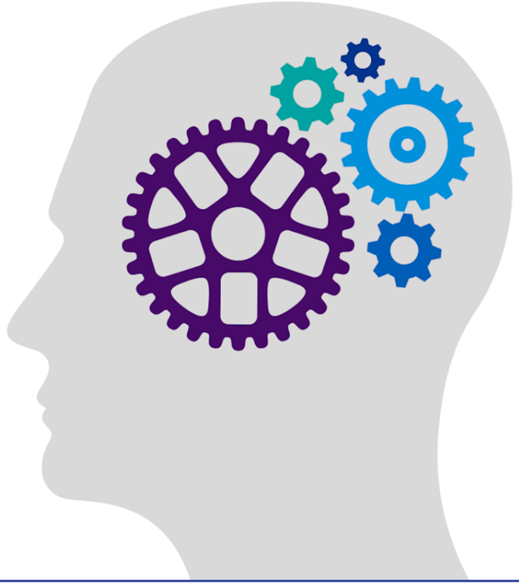
Columbia Capstone Project
& KPMG Digital Lighthouse

Many AI applications rely on "humans in the loop" to function at a high level of accuracy

Human-in-the-loop systems take two forms, both of which contribute to lower gross margins for many AI solutions.

1 State of the art requires manually curated datasets

- This process is laborious, expensive, and the biggest barrier to enterprise adoption of AI
- Maintaining accuracy requires new data to be continuously captured, labeled, and fed back to the system
- In a race to useable data, labeling quality and quality assurance becomes a bottleneck



2 Human reviewers augment AI-based systems

- AI models require human intervention especially as regulations become more stringent
- Issues of safety, fairness, and trust demand human oversight, and is linked to data labeling
- Deciding what needs human evaluation is the key to minimizing costs

Key Objectives

Stage 1

Evaluate key active learning approach given a literature review

Stage 2

Determine performance metrics to compare active learning approaches

Stage 3

Build a modular package to easily use different active learning strategies

Stage 4

Connect pipeline to a GUI

Stage 5

Recommend approach/framework for future application of active learning

Background Research

Active Learning Strategies

Query by uncertainty

Least Confident ✓
Largest Margin ✓
Largest Entropy ✓
Query-By-Committee
Variation Reduction

Query by distribution

Coreset ✓
K-means ++ ✓

Other

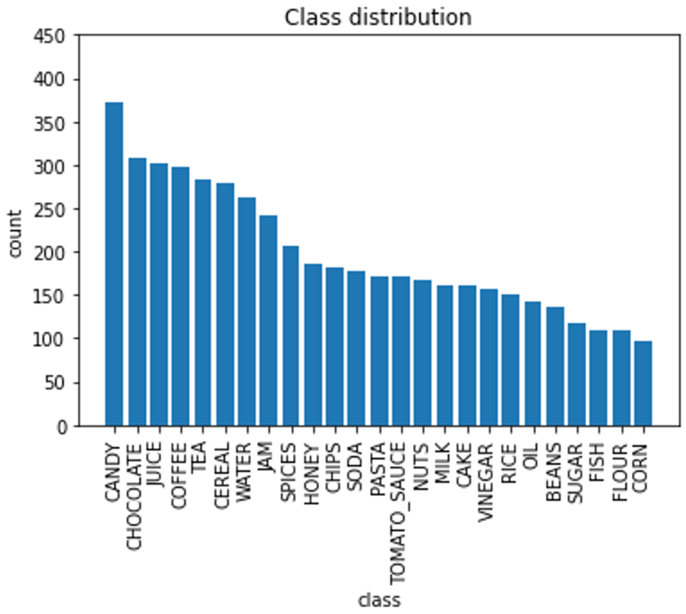
Learning Loss ✓
Confident Coreset ✓
Expected Model Change

Semi-supervised Learning + AL

Pseudo-label
Noisy Student ✓
Temporal Ensembling Model ✓
Mean Teacher ✓
Mixmatch
FIxmatch

- Out of the active learning strategies we reviewed, we chose 7 to experiment since they are suitable to apply to image classification task
- We implemented 3 semi-supervised learning algorithms, and researched the idea combining semi-supervised learning and active learning in our method

Freiburg Groceries Dataset



- Consists 5,000 images from 25 different classes of groceries, with at least 97 images per each class.

- Images were taken from real-world stores and the each image came with different angle, light condition, degree of cluster.

- reflecting a true scenario of what people would see everyday at a grocery store so it suited our needs for a dataset that people would use to build machine learning models for image classification tasks

cereal

chips

Statistic Based Query

1 Least Confidence Query

Lease Confidence Query is the simplest and most commonly used query strategy. The idea behind that is get the ground truth of instance that model feel most uncertain.

$$x^*_{Query} = \arg\max_x 1 - P_a(\hat{y}|x)$$

2 Margin Query

Least confidence query only consider the information of highest posterior probability and throws away the information of inference probability of remaining classes. What margin query do is finding the instances that model hesitates between two classes.

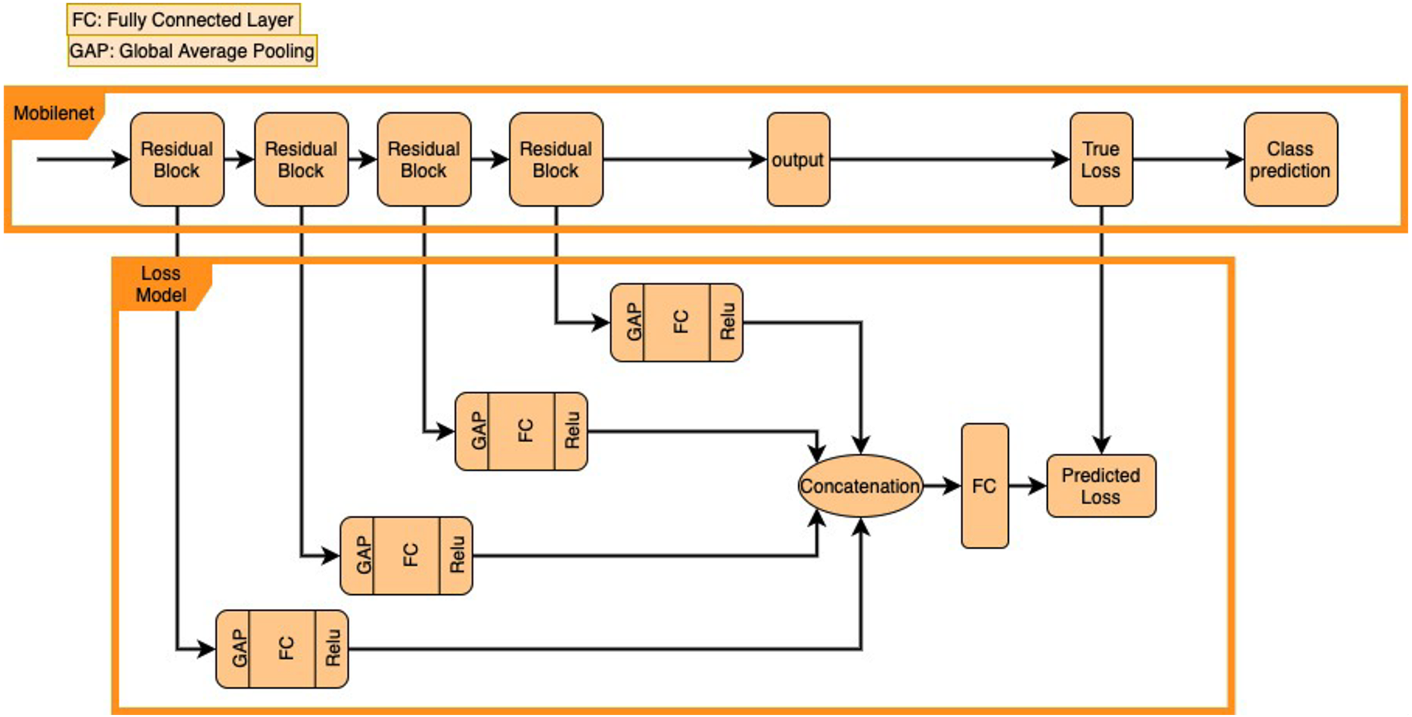
$$x^*_{Query} = \arg\min_x P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x)$$

3 Entropy Query

Entropy is a more comprehensive way to consider the distribution of predicted probability.

$$x^*_{Query} = \arg\max_x -\sum_i P_\theta(y_i|x) \log P_\theta(y_i|x)$$

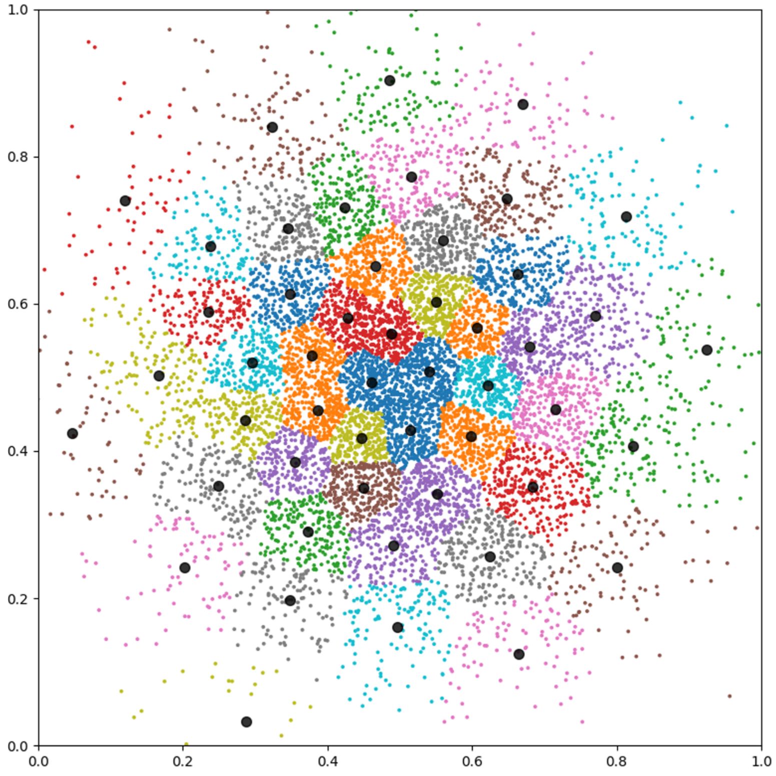
4 Loss Query



- Building a loss model upon original classifier to predict the classification loss of unlabeled data
- Higher loss means more uncertainty to the data

Distribution Based Query

1 K-Means Query



1. Query all centroid in the embedding space
2. Assume a centroid will cover the information of all data in its cluster

2 K-Centers Greedy Query

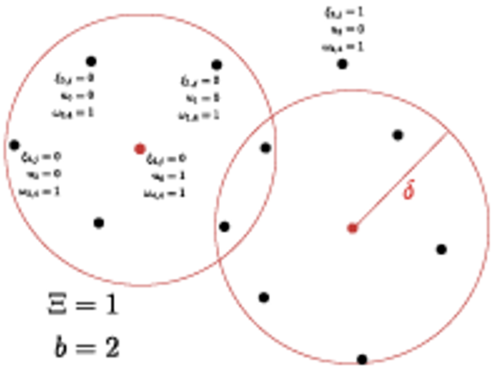
- Map all unlabeled data and labeled data to the embedding space
- Greedily Query the unlabeled data point who is most far away from its nearest labeled data

Algorithm 1 k-Center-Greedy

Input: data $\mathbf{x}_i$, existing pool $\mathbf{s}^0$ and a budget b
Initialize $\mathbf{s} = \mathbf{s}^0$
repeat
  $u = \arg\max_{i \in [n] \setminus \mathbf{s}} \min_{j \in \mathbf{s}} \Delta(\mathbf{x}_i, \mathbf{x}_j)$
  $\mathbf{s} = \mathbf{s} \cup \{u\}$
until $|\mathbf{s}| = b + |\mathbf{s}^0|$
return $\mathbf{s} \setminus \mathbf{s}^0$

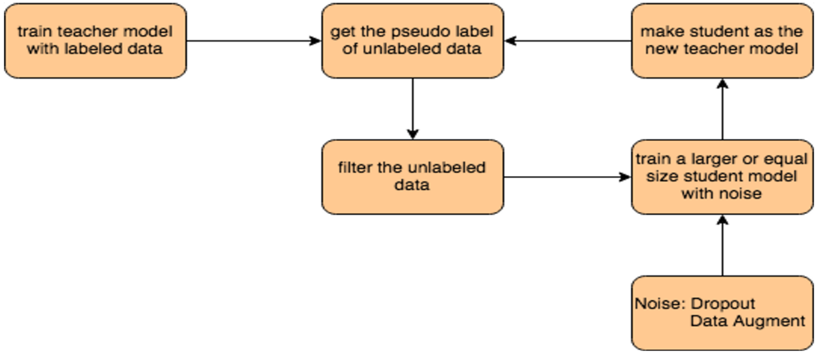
3 Confident-Coreset Query

- Same query logic as K Center Greedy
- Adding predicted loss into account when calculate the distance from nearest labeled data



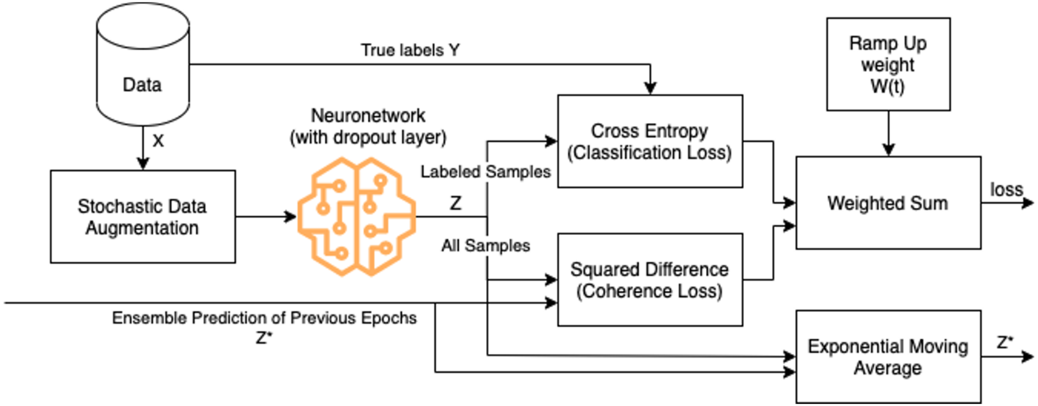
Semi-Supervised Learning

1 Noisy Student



- Teacher model create pseudo label for student model to learn
- Student become teacher model for next round

2 Temporal Ensembling

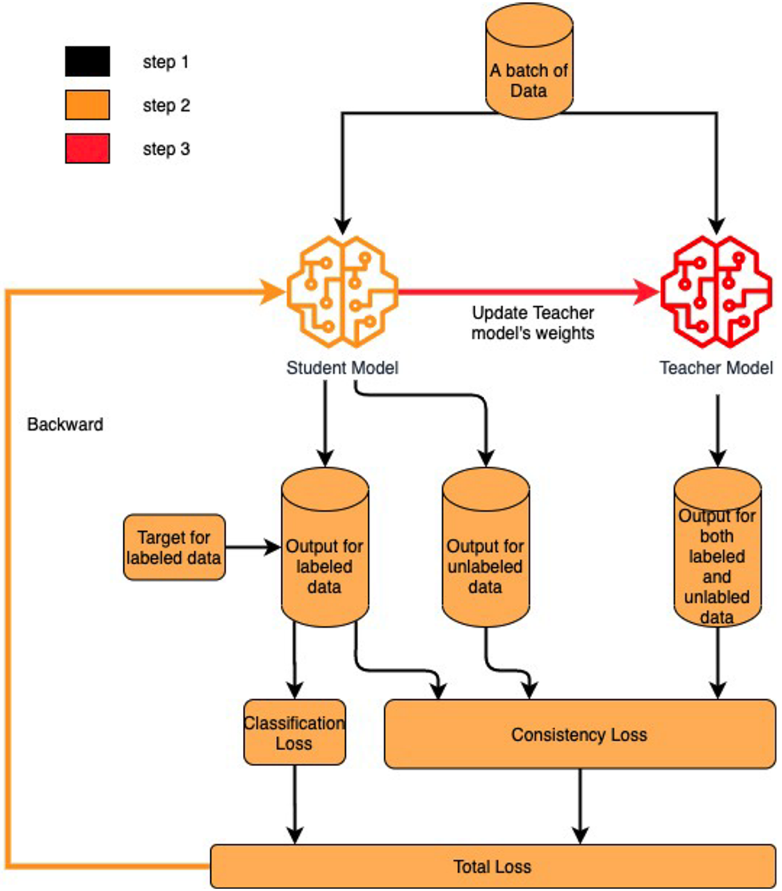


- Pseudo Label is the moving average prediction of all previous teacher models

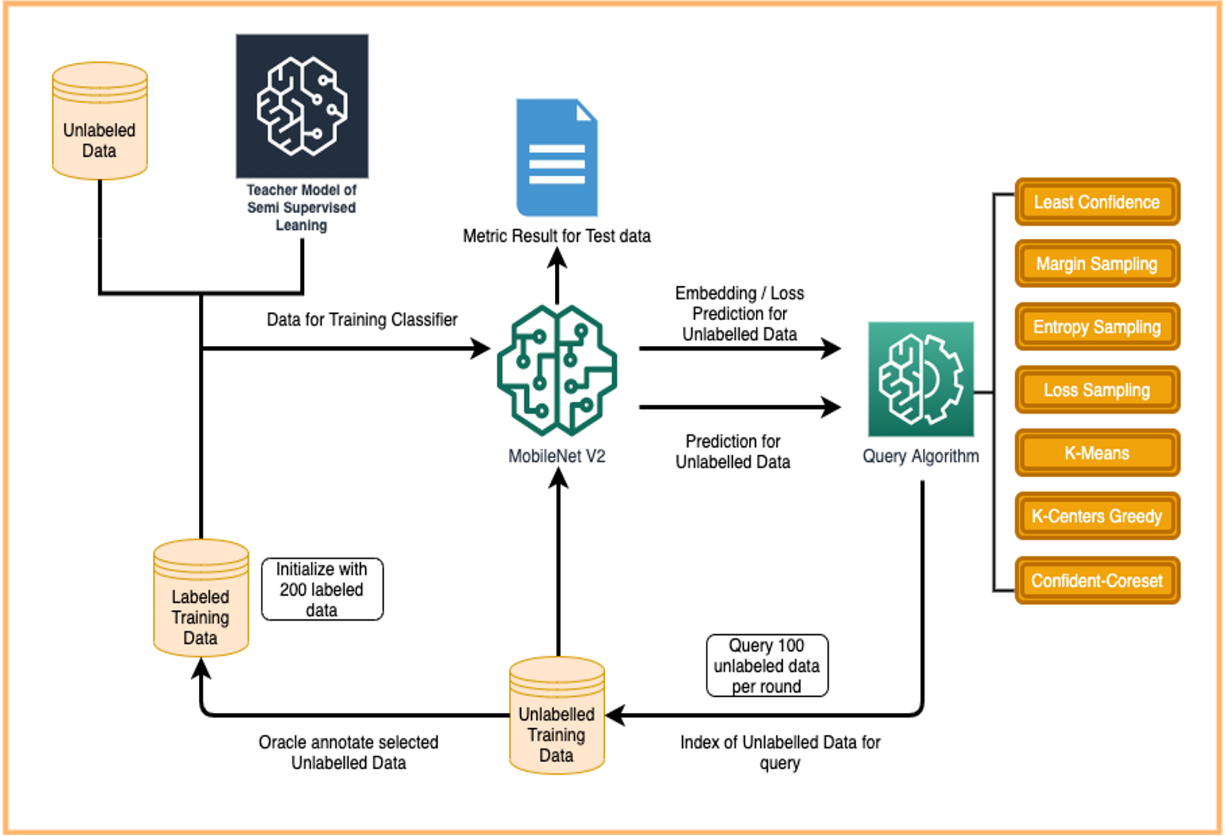
Semi-Supervised Learning

3 Mean Teacher

- Teacher Model is not the student model of last round
- The Weights of Teacher Model are the moving average of all previous student model's weights



Frameworks

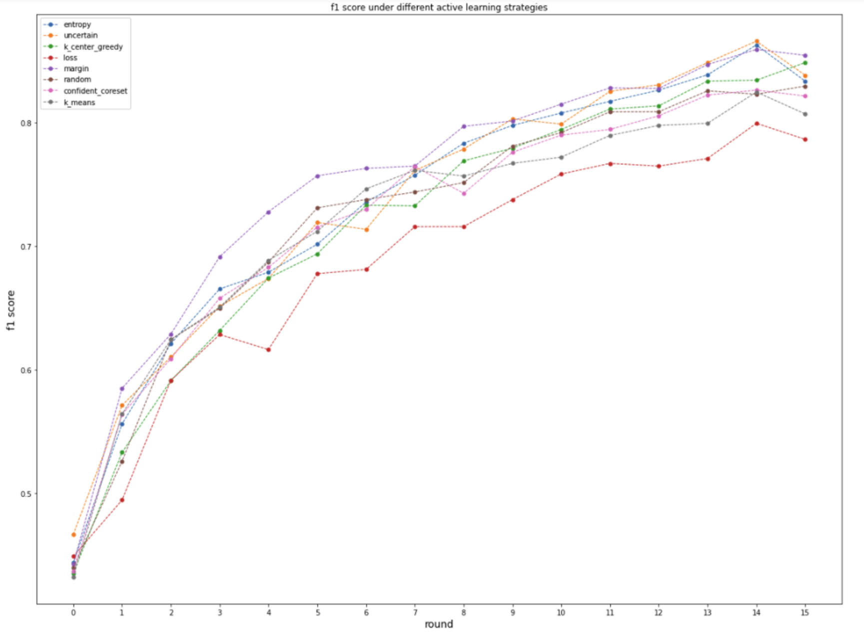


Task for Each Round:

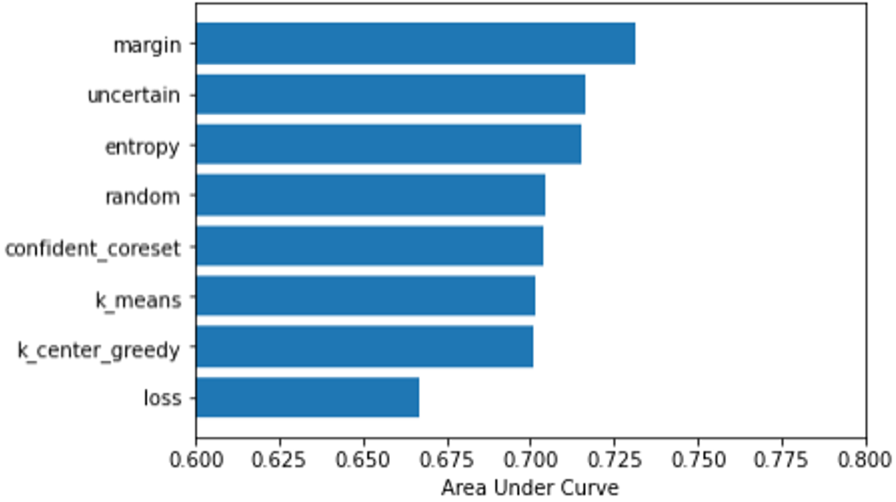
1. Train on current labelled dataset (also unlabelled data if using semi-supervised learning
2. Inference on test data and report metrics
3. Predict and Encode unlabelled data
4. Query the unlabelled data and add them to labelled dataset

Experiments: Active Learning

Performance Measure



Learning Curve(weighted F-1 score) for AL Strategies

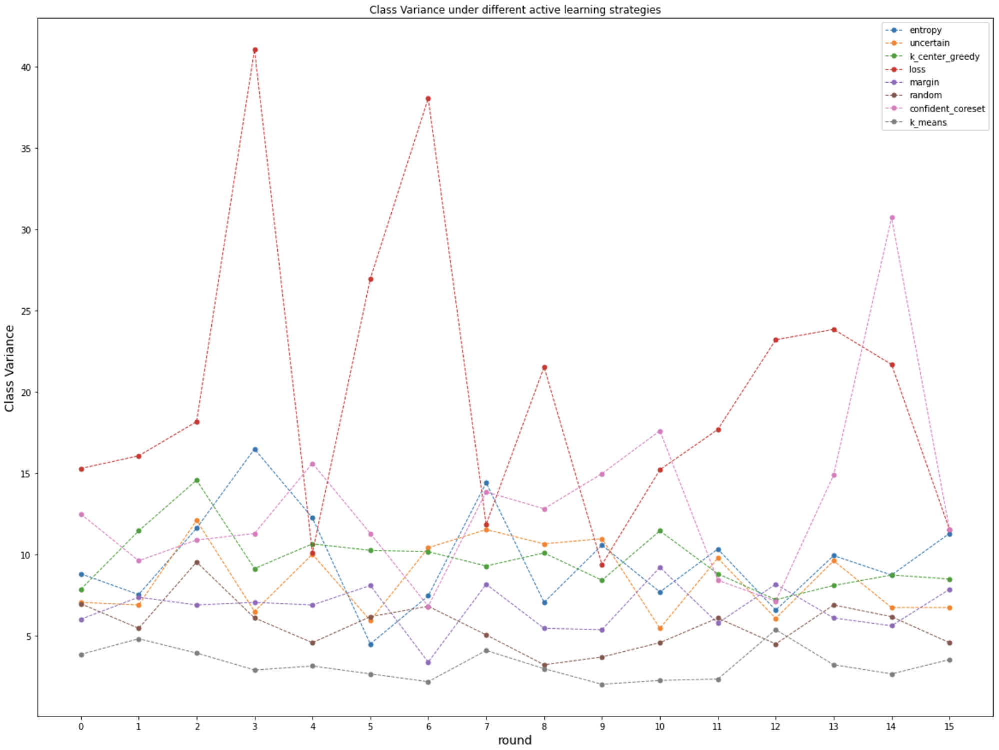


Area Under Curve

- Out of the 7 active learning strategies we tested, **margin, least confidence(uncertain), entropy sampling** achieved higher AUC than uniform random sampling
- **K-center greedy** consistently outperforms random sampling when there are more labeled data available

Experiments: Active Learning

Diversity Measure - I



- Higher variance in number queried for each class means lower diversity
- Loss sampling, confident coreset tend to query images from a few classes
- **K-means** sampling queries most diverse images batches
- **Margin** sampling queries diverse batches while achieves best performance

t-SNE visualization

- We used the t-SNE technique to project the final embedding produced by the last layer of the network onto 2D space

- Gave us insights on what each strategy is doing, especially by using the plot on the early iterations

- Also shows the reason why some strategies are not performing well. For example, we observed Loss strategy sampled a lot of images that are close in the embedding space within each cluster. These findings also agree with the class variance metric

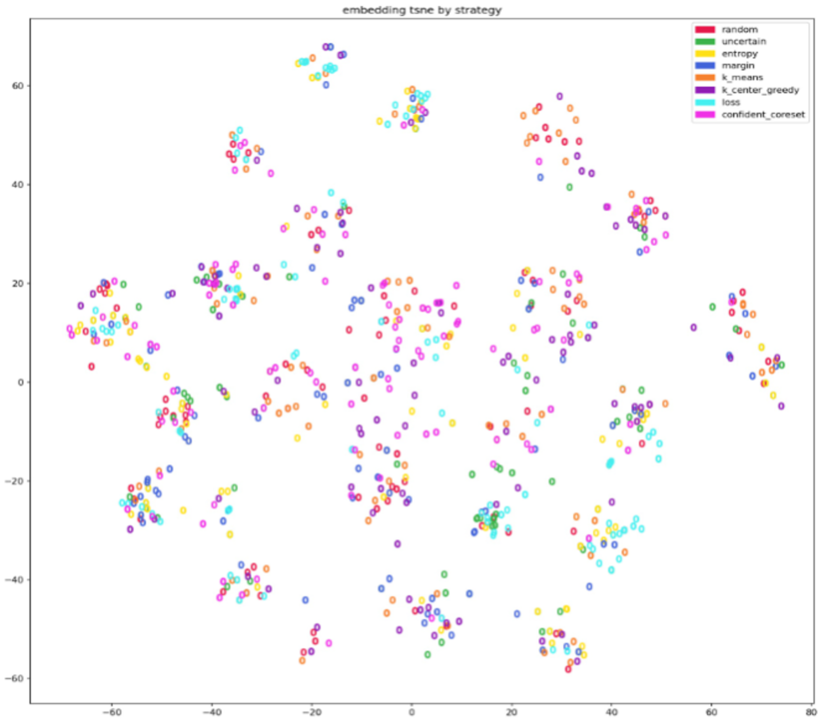


Figure 8: embeddings of samples chosen from 1st iteration for all the strategies

Experiments: Semi-supervised Learning

1 Noisy Student

Table 1: Classification Performance for Noisy Student with 800 labeled data

Training Method	Resnet34	Resnet50	Resnet101
Supervised Training	0.731	0.739	**0.763**
Noisy Student	**0.744**	**0.755**	0.756

NoisyStudent doesn’t outperform the common supervised training method.

2 Temporal Ensembling

Table 4: Classification Performance for Pre-Trained Mobilenet with Temporal Ensembling

Model	Highest F1 Score	Epoch to Achieve Highest F1	Epoch to Achieve 0.65 F1 Score
Supervised MobileNet	**0.717**	**40**	**18**
Semi-Supervied MobileNet	0.716	72	37

Table 5: Classification Performance for Non-pretrained Mobilenet with Temporal Ensembling

Model	Highest F1 Score	Epoch to Achieve Highest F1	Epoch to Achieve 0.25 F1 Score
Supervised MobileNet	**0.294**	**140**	**98**
Semi-Supervied MobileNet	0.211	172	Never

Experiments: Semi-supervised Learning

3 Mean Teacher

Table 2: Classification Performance for Pre-Trained Mobilenet with Mean Teather

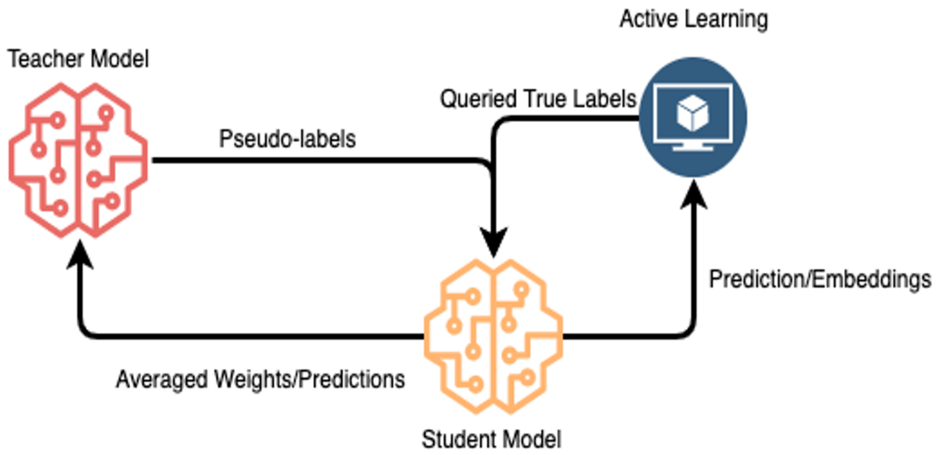
Model	Highest F1 Score	Epoch to Achieve Highest F1	Epoch to Achieve 0.65 F1 Score
Supervised MobileNet	0.716	25	22
Semi-Supervied MobileNet	**0.739**	**14**	**7**

Table 3: Classification Performance for Non-Pretrained Mobilenet with Mean Teather

Model	Highest F1 Score	Epoch to Achieve Highest F1	Epoch to Achieve 0.25 F1 Score
Supervised MobileNet	0.302	**106**	69
Semi-Supervied MobileNet	**0.321**	115	**52**

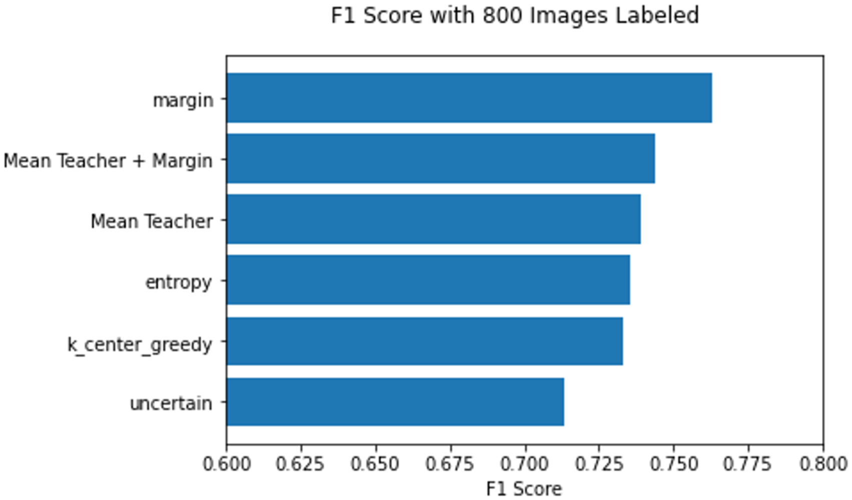
- Compared to the normal supervised learning methods, Mean-teacher achieves higher F1 score on the pretrained mobilenet.
- Based on this positive feedback, we explore further if we can adopt this method of improving active learning queries

Experiments: Semi-supervised Learning + Active Learning



Experiment Setup:

- Train 100 epochs of Mean Teacher(as in experiment of pure Mean Teacher)
- Query 200 images with margin sampling strategies every 14 epoch, so that we have 800 images queried in the end



Result:

- With 800 images queried in total, Mean Teacher + Margin outperformed pure semi-supervised learning but did not outperform margin sampling

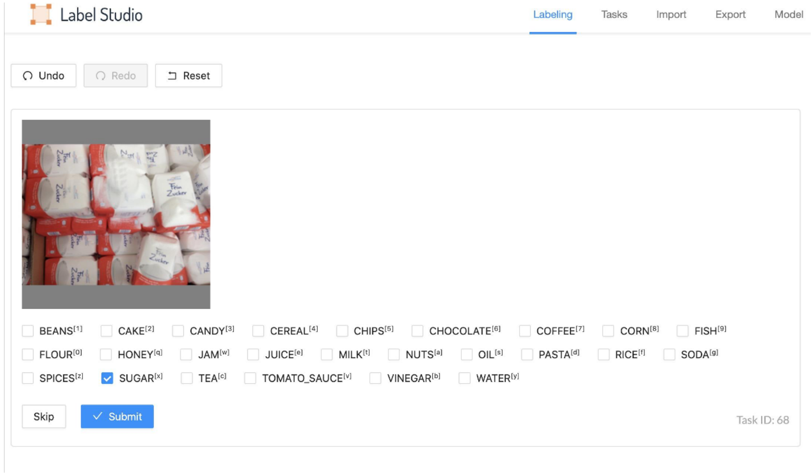
An Easy-to-use AL Package Connected with GUI

Code for Your Active Learning Training Loop:

```
model.fit()
query_time, queried_index = query('margin',model,20)
update_json(json_path, queried_index,idx2base, base2idx, model,
dataset, class_name_map)
index_list, target_list = read_from_oracle(path, idx2base,
base2idx)
dataset.update_target(index_list, target_list)
model.update()
```

Queried Images

Labels from Human Annotator



Future Works

1 Better Structure to combine active learning and semi-supervised learning

2 Try other encoders to improve the quality of embedding

2 Develop more comprehensive metric to measure the diversity of query data which can be useful for the selection of query strategy