LTIMindtree

WHITEPAPER

Address Annotation Issues by

Contextual Training of Feature Extraction Network

Abstract

When we think of machine learning/deep learning models, two techniques come to mind immediately — supervised learning and unsupervised learning. In very simple terms, main difference between two approaches is - availability of labelled data, supervised learning has it, and other, does not. Both approaches have their advantages and shortcomings and have their fair share of relevance based on business use case(s) in question.

Over time, scientists have introduced several techniques that offer the flavors of both worlds. Two most popular techniques are semi-supervised learning and self-supervised learning. These methods are developed, again to create a "data efficient" system. We can say that these are "somewhat" an extension of "unsupervised learning" as pointed out by Yann LeCun – "I Now call it "self-supervised learning", because "unsupervised" is both a loaded and confusing term. " Source – Link

Semi-supervised learning is a machine learning method in which we have input data, and a fraction of input data is labeled i.e. only few input samples of the dataset are provided with target values. It is a mix of supervised and unsupervised learning. This can be useful in training of models with less labelled training data. The training process can use a small chunk of labeled data and pseudo-label rest of the dataset by learning from the feature representation of labeled data.

Self-supervised learning is a machine learning process where a model trains itself to learn one part of input from another part of input. It is also known as predictive or pretext learning. In case of pseudo-labeling, we have some labelled data to learn from but in case of self-supervised learning we don't have any labeled data and thus we train the model using method like contrastive learning. In this process, an unsupervised problem is transformed to a supervised problem by auto generating labels. To make use of huge quantity of unlabeled data, it is crucial to set right learning objectives to get supervision from the data itself. The process of self-supervised learning method is to identify any hidden part of the input from any unhidden part of the input. This work tackles the problems surrounding data availability for CV use cases.

How really these "learnings" pan out? Let's consider a simple example. Consider having a significant number of unlabeled data waiting to be labelled for modelling, such labelling tasks equally require lot of manual labor which further increases the overall resources. There are 2 ways to handle such situations:

1. To label a small amount of data and use it to train the model and pseudo-label remaining data is known as – **Semi-supervised approach**

2. To use techniques such as contrastive learning to extract meaningful information for feature representation of inputs and to use it as a backbone for training model for the objective is known as – **Self-supervised approach**

Using both the approaches - semi and self-supervised learning, we will demonstrate the effectiveness of these methods in comparison to traditional supervised approach. In addition, we will revisit these approaches once more in our further conversation.

This experiments also shows how the current and proposed approach depends on availability of labelled data and its impact on model accuracy.

Motivation

Neural networks have demonstrated their ability to provide remarkable performances on a wide range of supervised learning tasks (e.g., image classification) when trained on extensive collections of labeled data (e.g., ImageNet, ResNet, DenseNet, " etc.). However, in practical business landscape, creating large datasets sometimes may be challenging because of several factors:

1. **Unavailability of trained resources to label vast amount of data.**
2. **Business pressure to take solution to market quickly**
3. **Insufficient labelled data availability or image quality**
4. **Financial constraints to maintain a team for annotation/re-annotation**

Above factors can be a roadblock for deep learning projects, and they continue to plague the industry to move a data science solution from development to production. While it's important that we create awareness, maturity to navigate this through, at the same time, it's also critical that we look at this problem from a different lens and try to explore some rational and logical alternatives.

We are now clear about the challenges regarding data while developing a deep learning application (In terms of Image analytics). Next, let's discuss how to tackle this problem by using semi and self-supervised learning techniques.

Contextualization

Let us recall an easy example from our day-to-day life as analytic practitioners try to explain this further. We were building vision models for a CPG company which entailed exhaustive data labelling activity. The team was struggling in terms of resources, time, and client pressure. That triggered us to relook at this problem and try to evaluate if adoption of these SOTA technologies yield a better (or at least baseline) result in a systematic way without so much annotation needed.

Vision Analytics to Drive Market Share in Retailers

Problem Statement

LTIMindtree built a solution which can identify, measure and track market share of SKUs on-shelf against competition (in stores, for general trade market) which can further prevent lost sales and declining market share because of non-compliance of SKU placement in stores/shelves against set standards/guidelines (planogram). Also, LTIMindtree's proposed solution was aimed to replace the inconsistency in manual updates from individual sales representatives on SKU placement/on-shelf availability (OSA).

Solution Description

Images are ingested and processed by an AI based solution to detect attributes such as SKU unit, category, sub-category, quantity, estimated shelf occupancy, shelf compliance (derived) for both customer and competition. To reduce the overall efforts in data labelling, active learning and auto-annotation techniques were employed, model output was delivered in excel format on a weekly basis. In addition, an end-to-end solution was developed, deployed, maintained in Azure further adhering to the Azure well-architecture framework guidelines. This solution was adopted across 2000+ stores, for 15 + Brands across 50+ products with 1000+ Images being processed per week and with the accuracy of >85% for all vision modules.

Current Approach on Model Building – Dependent on Enough Annotated Data(supervised)

The above given example is a typical textbook example, starting from problem statement identification, data collection, annotation, augmentation, model evaluation, finalization, hyperparameter tuning, deployment, maintenance, and so on. This is "the" right approach for a data science exercise with supervised learning. However, this method is completely dependent on the availability of enough labelled and quality data, which was a challenge!

LTIMindtree's Experiment Premise

The dataset that is being considered for our experiment is a retail data of ~150K images having 5 classes generated from open source SKU110K dataset. The labelled dataset constitutes about ~35K images among which training set size was varied for different approaches with testing set size fixed at ~5K (15%) images across different experiment.

Our experiments will deeply focus on three approaches with each differing on how much labelled data is fed to it during model training.

1. Supervised Learning – Fully Annotated Dataset (100%)
2. Semi-Supervised Learning – Partial Annotated Dataset (10% - 50%)
3. Self-Supervised Learning – Huge Dataset with minimal Annotation to train classification head (<10%)

Let us now deep dive into these 3 methods respectively.

Supervised Learning – Fully Annotated Dataset

The Supervised learning approach requires dataset to contain 100% labelled data using which, the model will be trained to find relation between data points and label to able to predict for new datapoints. For this experiment, SOTA ResNet50 feature extraction network with ImageNet weights and ResNet50 backbone network trained using SWAV algorithm by Meta, with pretrained weights obtained by trained on 1 billion images are considered for comparative analysis. Let us try to understand this statement by breaking it in multiple pieces.

Why SOTA pretrained ResNet50 model with ImageNet weights and pretrained ResNet50 backbone network trained using SWAV method?

We know ResNet50 models are powerful SOTA models for image classification problem as they use skip connections to pass information to deeper layers of network. SWAV algorithm proposed by Meta utilizes ResNet50 as a backbone network trained on 1 billion unlabeled images. We used same method to perform our experiments by using network with pretrained weights as feature extraction network and training a feature extraction network on custom dataset for comparative analysis.

We will use this for comparison against non-traditional approaches, Semi and Self-Supervised learning.

Semi-Supervised Learning – Partial Annotated Dataset (10% - 50%)

In general, the core idea of semi-supervision is to treat a datapoint differently based on whether it has a label or not. For labeled points, semi-supervised learning algorithm will use traditional supervision to update model weights (for increasing confidence/ prediction capability of the model), and for unlabeled points, semi-supervised learning algorithm minimizes the difference in predictions between other similar training examples.

Our labeled points act as a sanity check; they ground our model predictions and add a structure to the learning problem by establishing how many classes are present, and which clusters correspond to which class. Unlabeled datapoints provide context; by exposing our model to as much data as possible, so that we can accurately estimate shape of entire distribution. With both parts, labeled and unlabeled data, we can train more accurate and resilient models.

Self-Supervised Learning – Large unlabeled data

Self-supervised learning is a neural network training technique that can be regarded as a mix between supervised and unsupervised learning methods. Like unsupervised learning models, it gains knowledge from unlabeled sample data. The neural network learns in two steps.

1. Training the feature extraction network with unlabeled data utilizing the above mentioned SWAV algorithm.

2. Employ self-supervision to enhance and fine-tune the model. A self-supervised model often predicts the hidden component or property of an item from the observed part, repeating this activity multiple times until it can identify the object from any viewpoint. This is what self-supervised learning aims to achieve. It operates on the structure of data to identify patterns deeper than just similarities between objects as done by clustering or grouping.

In essence, self-supervised learning algorithms are designed to get all information they require directly from the data itself. Self-supervised learning systems need to be efficient in terms of runtime and memory because they require a lot of data and operate with billions of parameters.

For our experiments, we have used self-supervised learning in training the backbone network using SWAV. Let us quickly summarize this in the below table -

Key Difference Between Semi and Self Supervised Learning

Semi Supervised Learning	Self-Supervised Learning
Data is partially labeled	Entire data is unlabeled
Can be used to train an End-to-End network	Can only be used to train feature extraction networks which later serves as backbone for final model
Works with smaller subsets of data as well	Requires huge amount of data
Uses partially labelled data to provide pseudo-labels to unlabeled set	Uses unlabeled data to train feature extraction networks
Require less training time and resources	Require more training time and resources
Trained for a particular dataset	Reusable for similar datasets

Modelling approach – For all experiments

The modelling approach considered for experimentation consists of two main components, backbone/feature extraction network and a classification head. Classification head is a linear layer with 5 nodes (for our experiment of retail dataset) which gives probability scores across classes for an input image. This head remains the same for all experiments which was trained on custom retail dataset with training set ranging from 10 % - 50 % and 85% of all labelled data for all three experiments.

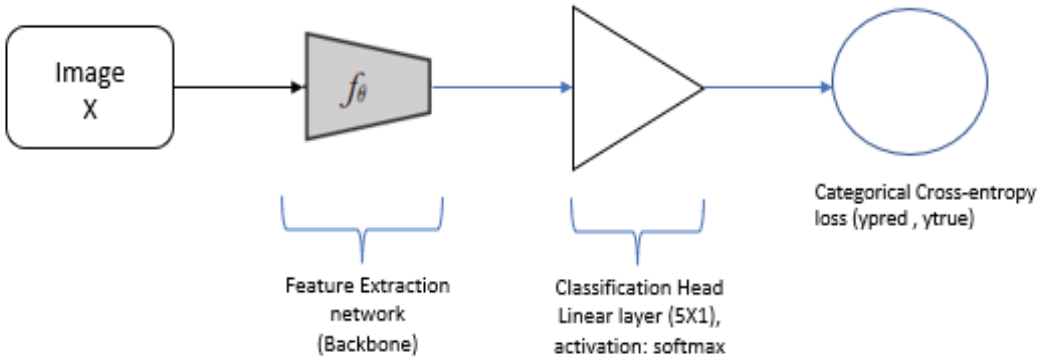

Fig 1: Modelling approach overview

For feature extraction network, they are varied for 3 different experiments (Supervised, Semi-supervised, and Self-supervised) which we'll see in a while.

We kept the head same (a linear layer - 5 output nodes, activation = softmax) for all experiments so that we can compare the feature extraction network's performance in different scenarios. We are doing these iterations to understand how a trained feature extraction network on huge unlabeled data with self-supervised learning can help the network converge in fewer epochs and give better accuracy.

Experiment 1

Traditional supervised learning approach where a pretrained ResNet-50 model with ImageNet weights as backbone network for extracting features of input images which are utilized to train the classification head.

Experiment 2

Feature extraction network is a pretrained model trained on 1 billion unlabeled images in self-supervised training approach – SWAV. The features extracted from backbone network are used to train the classification head.

Experiment 3

This is like Experiment2 but instead of pretrained backbone network we have trained this network on 100K unlabeled custom retail dataset (using SKUs from SKU110K dataset). Here, we'll see how training feature extraction network on similar unlabeled dataset using self-supervised approach can generate better feature representation of an input and give higher accuracy with comparatively less labeled data. Self-supervised training approach (SWAV) was used as custom training approach. Below is a generalized representation for training backbone network (ResNet-50) using self-supervised approach.

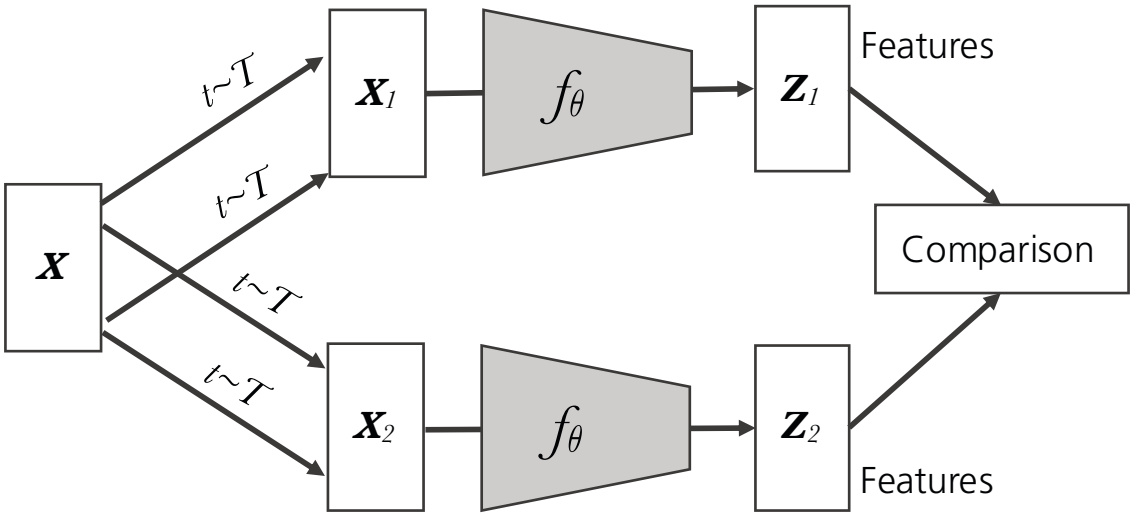

Fig 2: Contrastive Instance Learning (Ref link)

X is the input image from which X1, X2 are generated using multicrop augmentation(t~T). f θ is the Resnet-50 network which is set to be trainable. Features from both the f θ networks (Z1, Z2) are used to perform comparison and fit the network on data using contrastive loss. SWAV network were considered which is an improved version of above architecture where in swapping assignments between views are performed to train the network. For further study please refer [1]

Results

In all three experiments, the feature extraction network was not trained on our target labeled data as shown in below table.

Training Data	Classification Head	Feature extraction network
Data is partially labeled	Yes	No
Unlabeled Data/ Pre-trained	No	Yes

Table -1

The below tabulation illustrates results from experiments. Interesting inferences can be drawn which may act as a guide in determining which model training technique will be appropriate for different scenarios.

Experiment	Backbone network	Note	Classficatio n Head	% Labelled Data	Validation Accuracy % - 15% Test Data	Epoch	Train Time
Exp 1	Supervised	Resent trained on imagenet dataset	Semi - Supervised	10	65.27	13	1hr 17min
				20	61.86	8	1hr 1min
				30	75.2	11	2hr 26min
				40	76.1	10	1h 57min
				50	77.4	8	2hr 4min
			Supervised	100	84.18	16	6h 5min
Exp 2	Self-Supervised	SWAV Trained on 1Billion Unlabelled Instagram Images	Semi - Supervised	10	75.9	6	1hr 1min
				20	85	9	1hr 21min
				30	85.5	7	1hr 35min
				40	81.8	4	1h 21min
				50	84.4	6	1hr 35min
			Supervised	100	86.79	6	2hr 18min
Exp 3	Self-Supervised	SWAV trained on ~100k unlabelled product images and trained for 50 epochs	Semi - Supervised	10	75.9	27	2hr 36min
				20	80.4	24	3hr 51min
				30	78.7	13	2hr 55min
				40	82.4	23	5h 32min
				50	81.4	19	5hr 18min
			Supervised	100	83.85	15	5hr 23min

Table -2

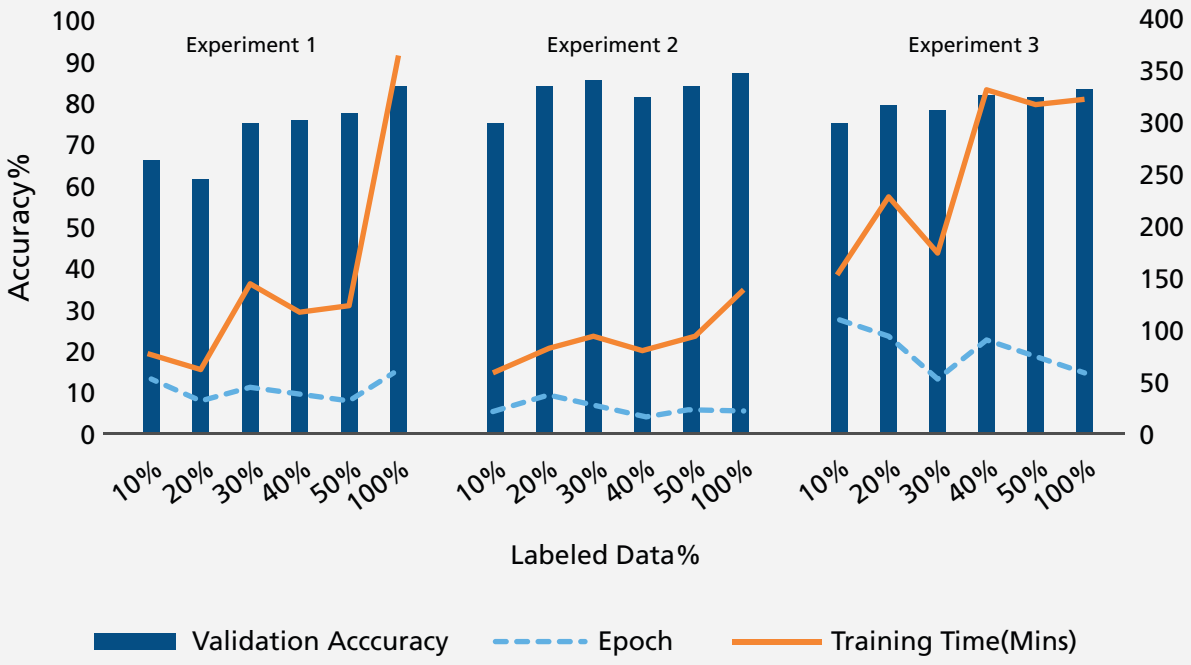

Fig 3: Experiment Observations

Based on the various factors related to availability of resources, labeled/unlabeled data we propose different approaches as below(these are some of the proposed combinations and not an exhaustive list):

Image Data Availability in terms of numbers	Infrastructure availability	Annotation time, resource Availability	Data science skill in the organization	Business criticality (in terms of prediction accuracy)	Proposed Approach
High	High	High	High	High	1/2
High	High	Low	High	Low	2/3
Low	High	High	Low	Low	1
High	Low	High	High	Low	1
Low	Low	High	Low	Low	1
Very High	High	Low	High	High	3

Table -3

Apart from this, a few general observations that we noted are as follows -

1. More the relevant data to train feature extraction network (backbone), more informative will be the features.

2. Since labelling huge amount of data requires a lot of efforts and resources, a self-supervised learning approach can be leveraged to train the backbone network and the classification head can be finetuned further using small, labelled data, as clear from Table -2, only 50% of labeled data was required in experiment 2 to achieve similar accuracy in experiment 1 which used 85% labeled data.

3. Ideally, experiment 3 should have outperformed experiment 2 as all the train data threshold levels given the backbone network was trained on relevant unlabeled data. This shortcoming can be attributed due to the dataset size (1B vs 0.1M) used to train the backbone network.

Next Steps

These are preliminary results which we derived by using a mix of predefined images and a set of custom images, only relevant to a specific project. So, in some sense, we can say that these results are generic and not derived from a typical lab setup. However, to validate these conclusions we can:

1. Train the backbone network on much larger dataset of relevant images to outperform the pre trained SWAV model.

2. Experiment with datasets from different domains.

3. Experiment with other feature extraction networks such as InceptionNet, DenseNet, etc.

4. Experiment with variables such as image size, projection prototype vector dimension, multi-crop algorithm cropping techniques, etc.

5. Compare with other contrastive learning algorithms such as SimCLR, MoCo.

Business Benefits:

Below are few of the benefits of our proposed approach:

Reusability:

Businesses can build a master backbone model for each domain (E.g. CPG - Retail) and reuse it to develop customized vision-based solutions.

Time:

We can considerably reduce time to market for vision-based solutions which require a lot of data annotation and training time thus increasing the timeline.

Reduced effort:

Data annotation is an iterative task and require a lot of effort from SMEs, we can use the proposed approach to reduce development effort and cost of solution.

References:

1. [2006.09882] Unsupervised Learning of Visual Features by Contrasting Cluster Assignments (arxiv.org)
2. Self-supervised learning: The dark matter of intelligence
3. Supervised, Semi-Supervised, Unsupervised and Self-Supervised Learning
4. Self-Supervised Learning for Image Classification
5. High-performance self-supervised image classification with contrastive clustering
6. Understanding Contrastive Learning
7. The SWAV method
8. Compare SimCLR, BYOL, and SwAV for Self-Supervised Learning
9. Weights & Biases (wandb.ai)
10. SWAV Explained
11. GitHub - facebookresearch/swav: PyTorch implementation of SwAV https//arxiv.org/abs/2006.09882

Authors

Anirban Pramanik
Principal Data Scientist

Santhosh Kumar R C
Module Lead

Surya Chauhan
Senior Software Engineer

About LTIMindtree

LTIMindtree is a global technology consulting and digital solutions company that enables enterprises across industries to reimagine business models, accelerate innovation, and maximize growth by harnessing digital technologies. As a digital transformation partner to more than 750 clients, LTIMindtree brings extensive domain and technology expertise to help drive superior competitive differentiation, customer experiences, and business outcomes in a converging world. Powered by nearly 90,000 talented and entrepreneurial professionals across more than 30 countries, LTIMindtree — a Larsen & Toubro Group company — combines the industry-acclaimed strengths of erstwhile Larsen and Toubro Infotech and Mindtree in solving the most complex business challenges and delivering transformation at scale. For more information, please visit www.ltimindtree.com.