Project 1 - Improving Self Supervised Riemannian Manifold

Self supervised learning is proposed for utilizing unlabeled data with the success of supervised learning. Producing a dataset with good labels is expensive, while unlabeled data is being generated all the time. The motivation of Self-Supervised Learning is to make use of a large amount of unlabeled data. The main idea of Self-Supervised Learning is to generate the labels from unlabeled data, according to the structure or characteristics of the data itself, and then train on this unsupervised data in a supervised manner. Self-Supervised Learning is now widely used in representation learning to make a model learn features of the data. This technique is often employed in computer vision, video processing, and robot control. In this project, you will experiment with the latest SSL-based methods and try to improve this paper, using the idea of a Riemannian Manifold.

Papers to read :

- Unsupervised Representation Learning by Predicting Image Rotation
- A Simple Framework for Contrastive Learning of Visual Representations
- Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning
- SELF-LABELLING VIA SIMULTANEOUS CLUSTERING AND REPRESENTATION LEARNING
- Barlow Twins: Self-Supervised Learning via Redundancy Reduction

Project Phases:

Phase 0: (10 Points)

- Read the 5 papers,
- Meet with the instructor and confirm your understanding
- Discuss the dataset and benchmark
- Prepare for presentation in class

Phase 1: (30 Points)

- Setup open source Self-Supervised learning benchmark repo
- Setup data loaders for all the dataset
- Reproduce the benchmark of Barlow twins on one of the benchmark datasets

Phase 2: (40 Points)

- Implement the Riemannian manifold based loss
- Explore sample mining

Phase 3: (20 Points)

- Hyperparameter search
- Evaluate the code on the benchmark test sets

Project 2 - Improving Deep Metric Learning with Proxy Synthesis

The goal of **Metric Learning** is to learn a representation function that maps objects into an embedded space. The distance in the embedded space should preserve the objects' similarity. Various loss functions have been developed for Metric Learning. For example, the contrastive loss guides the objects from the same class to be mapped to the same point and those from different classes to be mapped to different points whose distances are larger than a margin. Triplet loss is also popular, which requires the distance between the anchor sample and the positive sample to be smaller than the distance between the anchor sample and the negative sample. In this project, you will experiment with the latest metric learning methods and try to improve this paper by incorporating inter batch memory and weighted sampling

Papers to read:

- No Fuss Distance Metric Learning using Proxies
- ProxyNCA++: Revisiting and Revitalizing Proxy Neighborhood Component Analysis
- Proxy Anchor Loss for Deep Metric Learning
- Cross-Batch Memory for Embedding Learning
- Proxy Synthesis: Learning with Synthetic Classes for Deep Metric Learning

Project Phases:

Phase 0: (10 Points)

- Read the 5 papers,
- Meet with the instructor and confirm your understanding
- Discuss the dataset and benchmark (CUB, VegFru)
- Prepare for presentation in class

Phase 1: (20 Points)

- Setup open source DML benchmark repo
- Setup data loaders for all the dataset
- Reproduce the benchmark of Proxy Synthesis

Phase 2: (40 Points)

- Augment the synthesis process using incorporating a memory
- Explore center-based proxy location calculation
- Implement inter-class distance-based weighted sampling

Phase 3: (30 Points)

- Hyperparameter search
- Evaluate the code on the benchmark test sets

Project 3 - Improving Cross-Domain Detection Using Object Colocation

The objective of **Cross-Domain object detection** is to detect similar objects in images belonging to a different domain. This is a challenging task as the standard detector is trained using images from only one domain (source). Cross-domain detectors find applicability when target domain annotations are not available to perform transfer learning. In the area of documents, object detectors are trained to detect different elements of the documents like Heading, figures, Paragraphs, tables, etc. In this project, you will explore training object detectors for document object detection and try to improve this paper using an object colocation loss

Papers to read:

- Cross-Domain Document Object Detection: Benchmark Suite and Method
- Domain Adaptive Faster R-CNN for Object Detection in the Wild
- Data Augmentations for Document Images
- Decoupled Adaptation for Cross-Domain Object Detection
- Adapting Object Detectors via Selective Cross-Domain Alignment

Project Phases:

Phase 0: (10 Points)
- Read the 5 papers,
- Meet with the instructor and confirm your understanding
- Discuss the dataset and benchmark (RICO, CDOD benchmark)
- Prepare for presentation in class

Phase 1: (20 Points)
- Setup open source CDOD benchmark repo
- Setup data loaders for all the dataset
- Reproduce Cross-domain document detection benchmark

Phase 2: (40 Points)
- Perform object co-location analysis
- Incorporating this information into the loss formulation

Phase 3: (30 Points)
- Hyperparameter search
- Evaluate the code on the benchmark test sets

Chart Analysis

The main task of chart analysis is given a chart image, extract the raw data that was used to create the chart image. We have done some of the leading research work in this domain. Our recently concluded challenge divides the overall task into several smaller sub-tasks that can be solved in isolation. Two teams participated in the final **Task-6.** We have also curated a novel dataset that provides dense annotations on real-world chart data as compared to any other dataset which is generated synthetically. Our recent survey provides a comprehensive overview of a chart analysis pipeline from the inception at the document stage to the data extraction and further involved applications like Question Answering, Retrieval, etc.

Project 4 - Box/Point detector based on Cascade-RCNN and FPN

The winning solution in the challenge uses two models: a box detector and a point detector to detect and extract data from chart images. The architecture itself is based upon a Cascade-RCNN and a Feature Pyramid(FPN) Backbone. Your objective is to do a reproducibility study for Task-6 based on this submission and provide an improvement on the same if possible.

- ICPR 2020 - Competition on Harvesting Raw Tables from Infographics
- Towards an Efficient Framework for Data Extraction from Chart Images
- Improving Machine Understanding of Human Intent in Charts
- Towards an Efficient Framework for Data Extraction from Chart Images
- Deep Layer Aggregation

Project Phases:

Phase 0: (10 Points)
- Read the 5 papers,
- Meet with the instructor and confirm your understanding
- Discuss the dataset and benchmark
- Prepare for presentation in class

Phase 1: (20 Points)
- Code up basic variations of box/point detector models based on established architectures.
- Setup data loaders for all the datasets (Ub-PMC, AdobeSynth)

Phase 2: (40 Points)
- Reproduce the Chart-Infographic results on UB-PMC(task6)

Phase 3: (20 Points):

- Implement Iterative-Closest-Point(ICP) objective
- Implement Graph-Matching (GM) objective

Phase 3: (10 Points)

- Hyperparameter search
- Single code repository with a modular structure to plug-n-play different detectors, on different datasets.

Project 5 - Box detector and point detector using CenterNet:

The team who finished second in the competition also employed two models using established architectures for box detection and point detection, using CenterNet with DLA-34 as the backbone (link). Your objective is to do a reproducibility study for Task-6 based on this submission and provide an improvement on the same if possible.

- ICPR 2020 - Competition on Harvesting Raw Tables from Infographics
- Towards an Efficient Framework for Data Extraction from Chart Images
- Improving Machine Understanding of Human Intent in Charts
- CenterNet: Keypoint Triplets for Object Detection
- A Benchmark For Analyzing Chart images

Project Phases:

Phase 0: (10 Points)

- Read the 5 papers,
- Meet with the instructor and confirm your understanding
- Discuss the dataset and benchmark
- Prepare for presentation in class

Phase 1: (20 Points)

- Code up basic variations of box/point detector models based on established architectures.
- Setup data loaders for all the datasets (Ub-PMC, AdobeSynth)

Phase 2: (40 Points)

- Reproduce the Chart-Infographic results on UB-PMC (task6)

Phase 3: (20 Points)

- Implement Iterative-Closest-Point(ICP) objective
- Implement Graph-Matching (GM) objective

Phase 3: (10 Points)

- Hyperparameter search
- Single code repository with a modular structure to plug-n-play different detectors, on different datasets.

Project 6 - ChartOCR:

Recently a team from Microsoft Research has also independently shown some traction on a new large-scale dataset. They focus only on three types of charts and propose a point-detector-based model having a backbone based on CornerNet with Hourglass Network. Your objective is to do a reproducibility study for Task-6 based on their method and provide an improvement on the same.

Papers to Read :

- ICPR 2020 - Competition on Harvesting Raw Tables from Infographics
- Chart Mining: A Survey of Methods for Automated Chart Analysis
- ChartOCR: Data Extraction from Charts Images via a Deep Hybrid Framework
- CornerNet: Detecting Objects as Paired Keypoints
- Stacked Hourglass Networks for Human Pose Estimation

Project Phases:

Phase 0: (10 Points)

- Read the 5 papers
- Meet with the instructor and confirm your understanding
- Discuss the dataset and benchmark
- Prepare for presentation in class

Phase 1: (20 Points)

- Setup open source Chart-OCR benchmark repo (Viability study - has multiple open issues)
- Benchmark pre-trained models(provided) on UB-PMC

Phase 2: (40 Points)Explore leveraging new data :

- Develop a training code from scratch based on the CornerNet repo.
- Train and Evaluate on UB-PMC(task6)

Phase 3: (20 Points)

- Can we train an ensemble of 5 models on disjoint splits of Excel400k to match/improve on results?

Phase 4: (10 Points)

- Hyperparameter search

- Single code repository with a modular structure to plug-n-play different detectors, on different datasets.

Project 7 - Key Point Detection and Graph Clustering - Multi-Human Pose Estimation

Multi-person pose estimation aims at localizing 2d key points of an unknown number of people in an image. It has attracted much research interest because of its significance in various real-world applications, such as human behavior understanding, human-computer interaction, and action recognition. This can either be done by first detecting each human and then predicting their key points (top-down) or predicting all key points and then grouping them into separate objects(bottom-up). We are interested in the latter approach and its application of graph clustering.

Papers to Read :

- Simple Baselines for Human Pose Estimation and Tracking
- Differentiable Hierarchical Graph Grouping for Multi-Person Pose Estimation
- Deep High-Resolution Representation Learning for Human Pose Estimation
- Dynamic Graph CNN for Learning on Point Clouds
- Graph Clustering

Project Phases:

Phase 0: (10 Points)

- Read the 5 papers,
- Meet with the instructor and confirm your understanding
- Discuss the dataset and benchmark
- Prepare for presentation in class

Phase 1: (20 Points)

- Setup train and evaluation scripts for coco and UB-PMC

Phase 2: (50 Points)

- Implement Differentiable Hierarchical Graph Grouping
- Train and Evaluate on coco keypoints 2017

Phase 2: (20 Points)

- Explore OHGC for chart analysis :
- Train and Evaluate on UB PMC

Phase 4: (10 Points)

- Hyperparameter search

- Single code repository with a modular structure to plug-n-play different detectors, on different datasets.