National Taiwan University
Computer Science & Information Engineering
Name: Cheng-Yu, Lin
ID: B10902024

113-1 (Fall 2024)
CVPDL

Assignment 2 Report

Generic Object Detection

1. **Transformer architecture**

   DETR uses a Transformer encoder-decoder architecture to achieve end-to-end object detection. Images are first passed through a CNN backbone to extract image features. These features, projected into some latent space, are then passed through the encoder and decoder to generate the decoded outputs. Finally, these outputs will run through some fully-connected feedforward network to get pairs of object classes and corresponding bounding boxes.

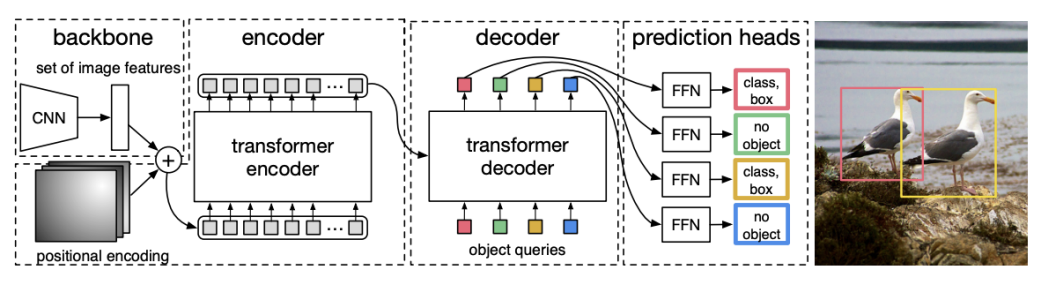

Figure 1: The architecture of DETR

Bipartite matching

After the predictions are generated, a bipartite matching algorithm will be used to find the optimal matching. Since the number of predicted boxes might be different from the number of ground truth boxes, some boxes with class "no object" will be padded to make sure the consistency of the number of boxes. Then, the bipartite algorithm will try different permutation to find the optimal permutation of predictions which yields the lowest cost when matched with the ground truth. After the matches are determined, we can calculate object specific loss of each matching pair and backpropagate the loss.

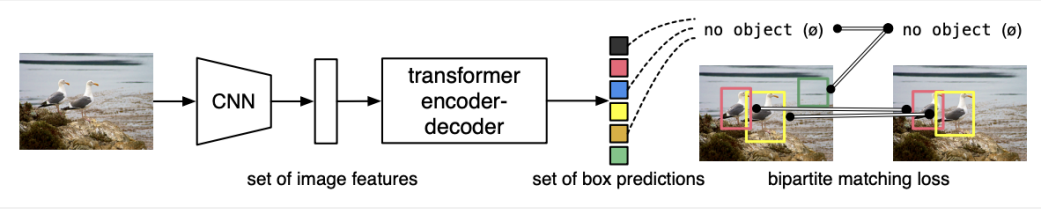

Figure 2: Bipartite matching in DETR

Advantage and limitations

Comparing with traditional CNN based detectors, DETR's end-to-end training without complex post-processing, such as NMS, effectively simplifies the pipeline. In terms of performance, the transformer architecture can better capture long-range dependencies, which improves detection of objects in complex scenes.

However, DETR still faces certain challenges. First of all, DETR has much slower convergence compared to CNN-based model(300 epoch vs. 12 epoch by Fast-RCNN). On top of that, the

interpretability of is another issue. The learned object queries are not human understandable, making it harder to analyze the reasoning behind predictions.

2. DINO improves training efficiency and accuracy over DETR by introducing **Contrastive DeNoising**, **Mixed Query Selection**, and **Look Forward Twice** techniques.

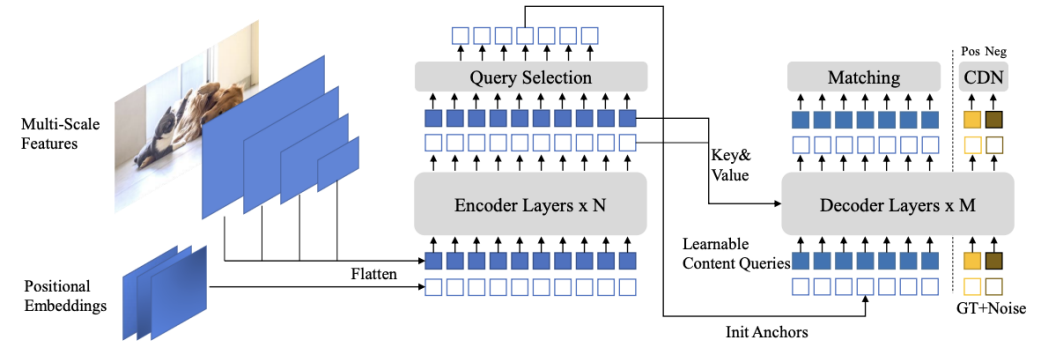

Figure 3: Model architecture of DINO

Contrastive DeNoising(CDN) is motivated by DeNoising queries in DN-DETR. DN queries add noise to the object queries during training, making the model more robust and better able to distinguish between similar objects. However, it lacks a capability of predicting "no object" for anchors with no object nearby. CDN is proposed to address this issue. For each ground truth box in an image, a positive and a negative query will be generated. Both of them are moderately noised with hyper-parameters λ_1 and λ_2. For positive queries, model is expected to predict correctly; for negative queries, model is expected to predict "no object" instead. This technique serves as a form of data augmentation within model's latent space, helping improve accuracy on distinguishing true objects from noisy backgrounds.

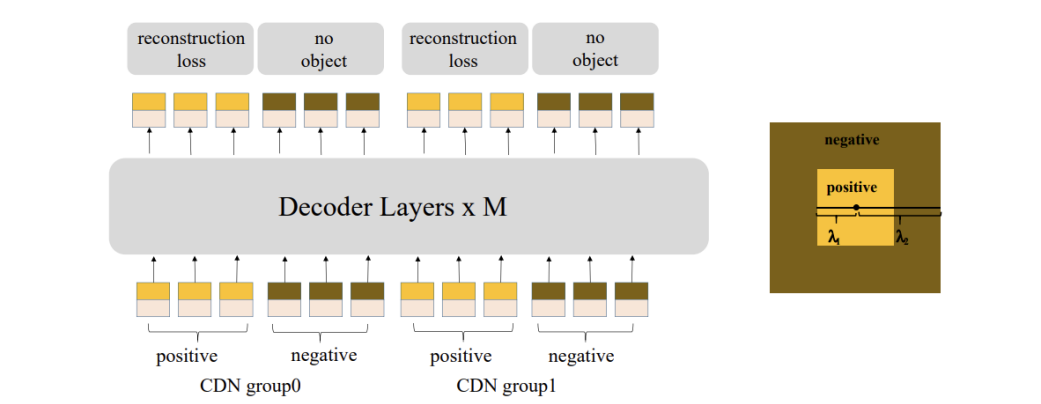

Figure 4: Contrastive DeNoising

Queries in DETR related model are formed by two parts: a positional part and a content part. In DETR and DN-DETR, both positional queries and content queries are directly learned from training data. Deformable DETR, on the other hand, uses a query selection technique to select top K encoder features as priors to serve as queries for decoder. DINO mixes both ideas by selecting top K features as positional queries and leaving content queries static the same as DETR. It helps the model to use better positional information to pool more comprehensive content features from the encoder.

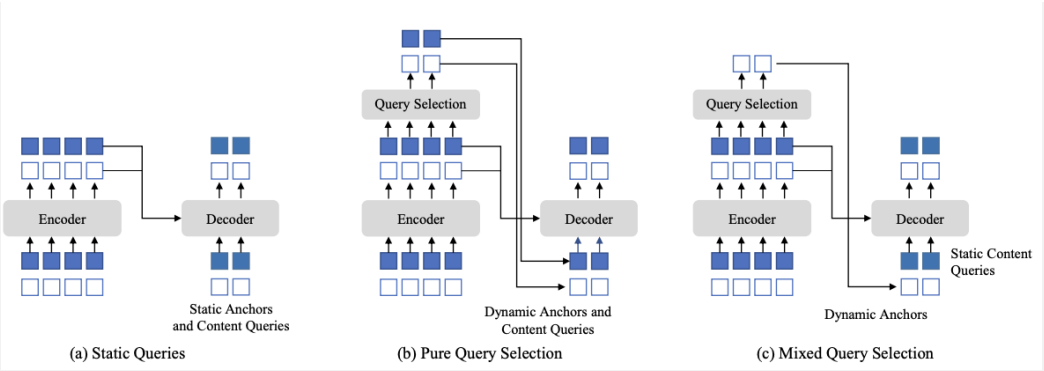

Figure 5: Different query selection techniques

The Look Forward Twice technique optimizes the idea from Deformable DETR. DINO propose a new way to box prediction. The predicted box is determined by two factors: the quality of the initial box and the predicted offset of the box. The idea is based on the author's conjecture that the improved box information from a later layer could be more helpful to correct the box prediction in its adjacent early layer.

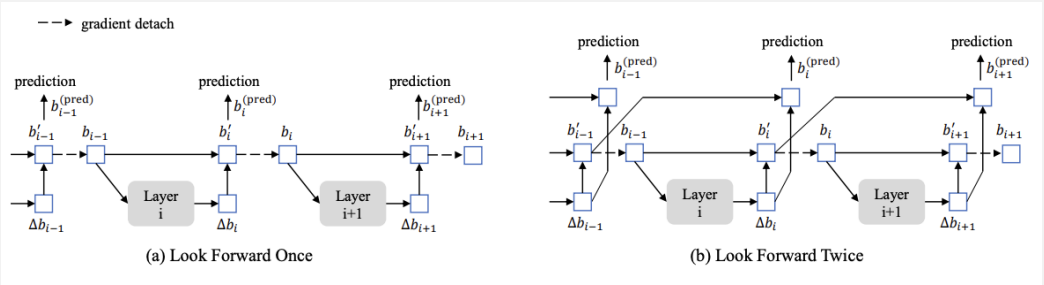

Figure 6: Comparison between Look Forward Once and Look Forward Twice

The experimental result demonstrates that the combination of these mechanisms leads to higher AP, particularly beneficial on the COCO dataset, where small and densely packed objects are common. Through ablation test, the author further proves that all these methods alone can also improve the performance.

Practical Issue 1: Lightweight Computer Vision

3. The EdgeViT achieve efficient performance by adding three key components into traditional ViT block, including **Local Aggregation**, **Local Propagation**, **and Sparse Attention**.

   For Local Aggregation, it uses the convolutional idea to aggregate information in local window with size of k*k. This reduces the number of tokens and significantly fasten the computation without sacrificing important information.

   Afterward, the Global Sparse Attention is applied on these selected tokens only. The idea is basically the same as traditional attention but with fewer tokens each represents an k*k area.

   The calculated information is propagated back to local region in final step. A Technique called "Transposed convolution" is used to upsample the information within each token.

   With these techniques, EdgeViT save lots of computations while preserving decent performance on mobile devices.

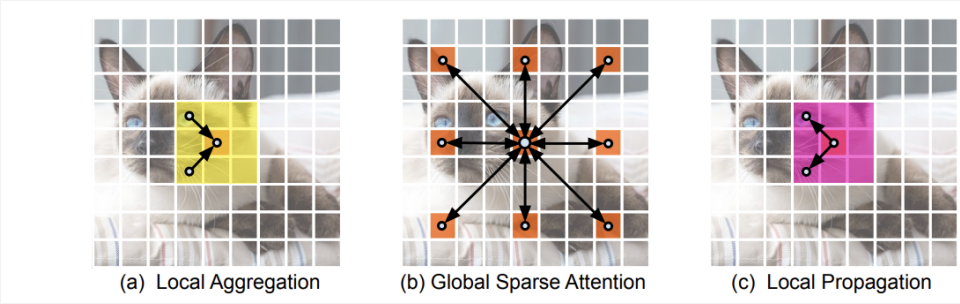

Figure 7: Key components in EdgeViT

4. Lite DETR focus on speeding up the encoder block of the transformer. It adopts an idea similar with gaussian pyramid, trying to combine different resolutions of the same image as input passing into encoder block. Different level resolution of images are not acquired by simply downsampling. Instead, they are feature maps in different stages within the backbone model(such as ResNet-50 or Swin-Tiny).

   Lite DETR replaces the dense attention in standard DETR with a sparse attention mechanism, which only attends to a subset of important spatial regions rather than processing every region equally. Since the high-level resolution might contain more information while consuming fewer tokens(approximately 25%), we can mainly focus on these tokens and reduce lots of redundant calculations.

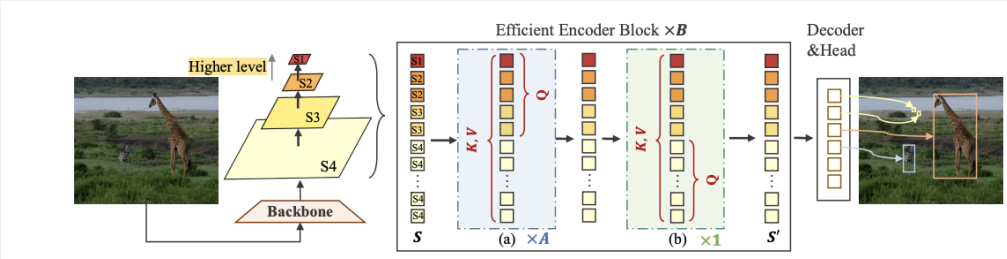

Figure 8: Different scales in Lite DETR

Practical Issue 2: Data Imbalance and Domain Adaption

5. **Effective number of samples**

   The effective number of samples for a class represents the number of independent, unique pieces of information that the samples of that class contribute to learning. Mathematically, it is defined as:

$$E_n = \frac{1-\beta^n}{1-\beta}, \beta = \frac{N-1}{N}$$

   The above equation implicates that $E_n = 1$ when $N = 1$ and $E_n = n$ when $N \to \infty$. The idea is to take the diminishing marginal benefits into account, and requires more samples of the same class to increase the effective number as n grows larger.

   How it helps address challenges associated with long-tailed data distributions?

   In the Class-Balanced Loss, the loss function is multiplied with the inverse of effective number. By re-weighting classes based on their effective number of samples, the loss function reduces the contribution of majority classes and increases the importance of minority classes. This helps the model learn more balanced representations across classes, improving performance on underrepresented classes.

6. Adversarial domain alignment tries to minimizes the domain shift between source and target distribution. We can pre-train a source encoder CNN using labeled source images, then perform adversarial domain adaptation by learning a target encoder CNN such that a discriminator that sees both encoded source and target examples can't reliably predict their domain labels. By doing so, the target CNN can learn important image features and also work well under target domain.

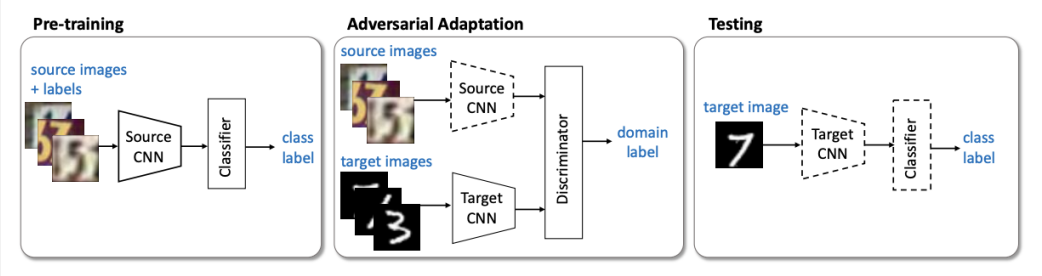

Figure 9: Adversarial domain alignment

Pracitcal Issue 3: Weakly-/Semi-supervised Learning

7. Data annotations for weakly supervised learning has lower degree at train time than the required output at test time. For example, in object detection task, training data in weakly supervised learning might only provide image-level labels without corresponding bounding boxes. Therefore, weakly supervised learning requires some form of guidance to indicate relevant features in the data, such as multiple instance learning.

   Data annotations in semi-supervised learning are complete, but some of the data is unlabeled. Semi-supervised learning often uses labeled data to train a model to predict pseudo-labels for unlabeled data, then recursively update the model. Self-training and consistency regularization are the main methods.

8. **Purpose**

   The standard assumption of MIL using positive and negative bags allows the model to train without precise annotations by using the presence or absence of the object in the image-level labels.

   Challenges

   (a) Instance ambiguity

   Determining which instance in a positive bag is responsible for the label can be difficult.

   (b) False positive

   Due to the design of positive bag, the model may identify irrelevant regions as positive instances, leading to noisy prediction.

   Considering the above factors, MIL algorithms are not optimal for solving instance classification.

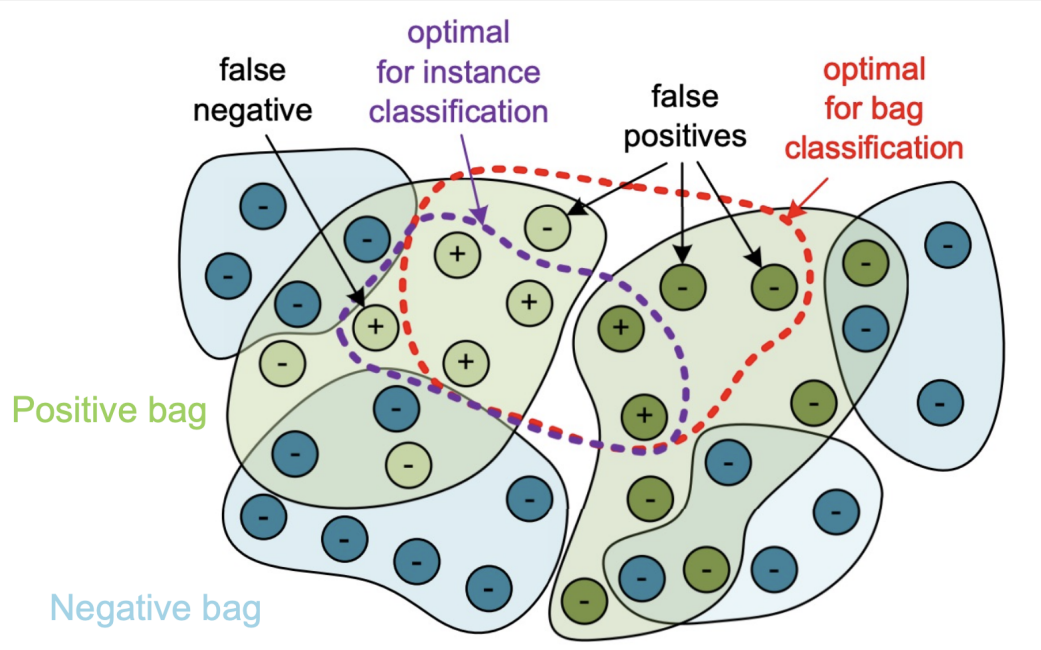

Figure 10: Challenges in WSOD

9. MixMatch applies a strong data augmentation technique. For labeled data, it creates an augmentation of it. For unlabeled data, it creates K augmentations and get the pseudo label by averaging the model prediction. MixMatch further mix up the label data with the unlabeled data in both image and label level, which encourages the model to learn invariant features and

improves its generalization ability. These design increases the confidence of pseudo-label and turns unlabeled data into valuable data, which in turn reduces the reliance on labeled data.

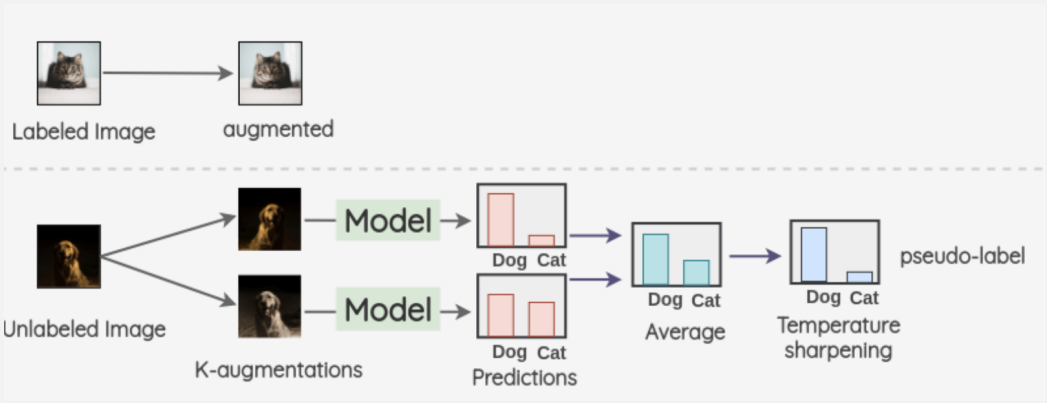

Figure 11: Data augmentation in MixMatch

Pracitcal Issue 4: Self-supervised Learning

10. **Mechanism**

    Using positive pair and multiple negative pairs to maximize the similarity between representation of positive pair and minimize the similarity between negative pairs. The design is to attract the distance within positive pair and repel the distance with rest of the images. Cosine similarity is used as similarity measure. Temparature-scaled technique is also used to make the predicted confidence score align with true correctness likelihood.

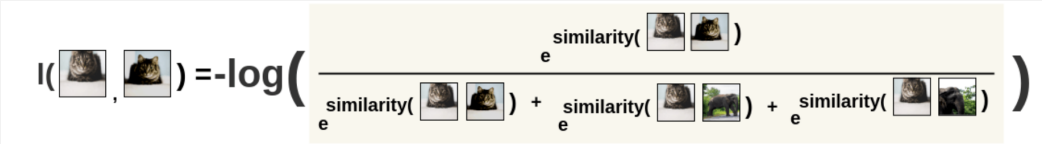

Figure 12: Loss calculation

Why it is particularly effective for self-supervised learning? The NT-Xent loss takes full advantage of contrastive learning by leveraging both positive and negative samples within a batch. By forcing the model to produce similar embeddings for different augmentations of the same image, NT-Xent helps the model learn invariant features, which makes the model robust.