Image-Guided Object Detection using OWL-ViT and Enhanced Query Embedding Extraction

Melih Serin
Department of Electrical and Electronics Engineering
Boğaziçi University
Istanbul, Turkiye
melihsrnn@gmail.com

Abstract*—Computer vision has been receiving increasing attention with the recent complex Generative AI models released by tech industry giants, such as OpenAI and Google. However, there is a specific subfield that we wanted to focus on, that is, Image-Guided Object Detection. A detailed literature survey directed us towards a successful study called Simple Open-Vocabulary Object Detection with Vision Transformers (OWL-ViT) [1], which is a multifunctional complex model that can also perform image-guided object detection as a side function. In this study, some experiments have been conducted utilizing OWL-ViT architecture as the base model and manipulated the necessary parts to achieve a better one-shot performance. Code and models are available on GitHub[2]

Keywords*— Computer Vision, Open-Vocabulary Object Detection with Vision Transformers (OWL-ViT), Object Detection, Vision Transformers, End-to-End Training, Generalized Intersection over Union (gIoU) Loss, Mean Absolute (L1) Loss, CLIP, DETR, Cosine Similarity, Yolov8, MS-COCO Dataset

I. Introduction

Recently, there have been numerous developments in the use of artificial neural networks, which have expanded the areas of study for Artificial Intelligence. Especially in computer vision, a rapid increase has occurred in a number of distinct real-life applications since researchers discovered a huge range of new utilization areas for complex computer vision models.

These applications include segmentation and object detection. As they constantly work to improve the functionality of these models, some of them now accept multimodal queries such as markers and/or points, masks, and texts. For instance, a new model called Segment Anything (SAM) [2] was developed by Meta-AI to undergo this kind of multi-functionality. These improvements had a significant positive effect on the capability of zero-shot learning in computer vision models. Zero-shot learning is, as the name states, the model's ability to predict unseen data samples. In other words, it shows how generalizable the model is to its problem definition. This term has been used in many studies.

[2]https://github.com/melihsrn/OWL-ViT_with_enhanced_query_embedding_network

After a detailed literature survey, some studies similar to this one are found. One of them is Simple Open-Vocabulary Object Detection with Vision Transformers (OWL-ViT) [1]. In OWL-ViT, they designed an open-vocabulary object detection method based only on transformer encoders (without any decoders). The model's pre-training process is similar to that of CLIP [3] training, which is essentially contrastive image-text pre-training on image-text representations. They then performed end-to-end training that was almost equivalent to DETR [4]. The model is agnostic to the source of the query representations because the image and text components of the model are not fused. Utilizing this aspect, they also used the model for one-shot image-conditioned object detection, which detects certain objects in an image using image query embeddings.

In this study, the aim is to train a model that can detect specific objects in an image defined by image queries that are not of categories fed into the model during training. A model is desired such that it is as generalizable as possible, which means that the model should detect objects not previously seen. To achieve this, several experiments are conducted using the OWL-ViT as the base model.

II. Background

A. DETR [4]

Detection Transformers (DETR) presented a new method that considers object detection as a direct set prediction problem. The approach streamlines the detection pipeline, effectively removing the need for many hand-designed components, such as a non-maximum suppression procedure or anchor generation, that explicitly encode our prior knowledge about the task. The main components of the new framework are a set-based global loss that forces unique predictions via bipartite matching and a transformer encoder-decoder architecture. Given a fixed small set of learned object queries, DETR explains the relations between the objects and the global image context to directly output the final set of predictions in parallel. This model is conceptually simple and does not require a specialized library, unlike many other modern detectors. DETR demonstrates accuracy and run-time performance on par with the well-established and highly optimized Faster R-CNN [5] baseline on the challenging COCO [6] object detection dataset.

Manuscript Received: 22.01.2024 | Manuscript Accepted: 20.03.2024 | DOI: 10.5281/zenodo.10938342

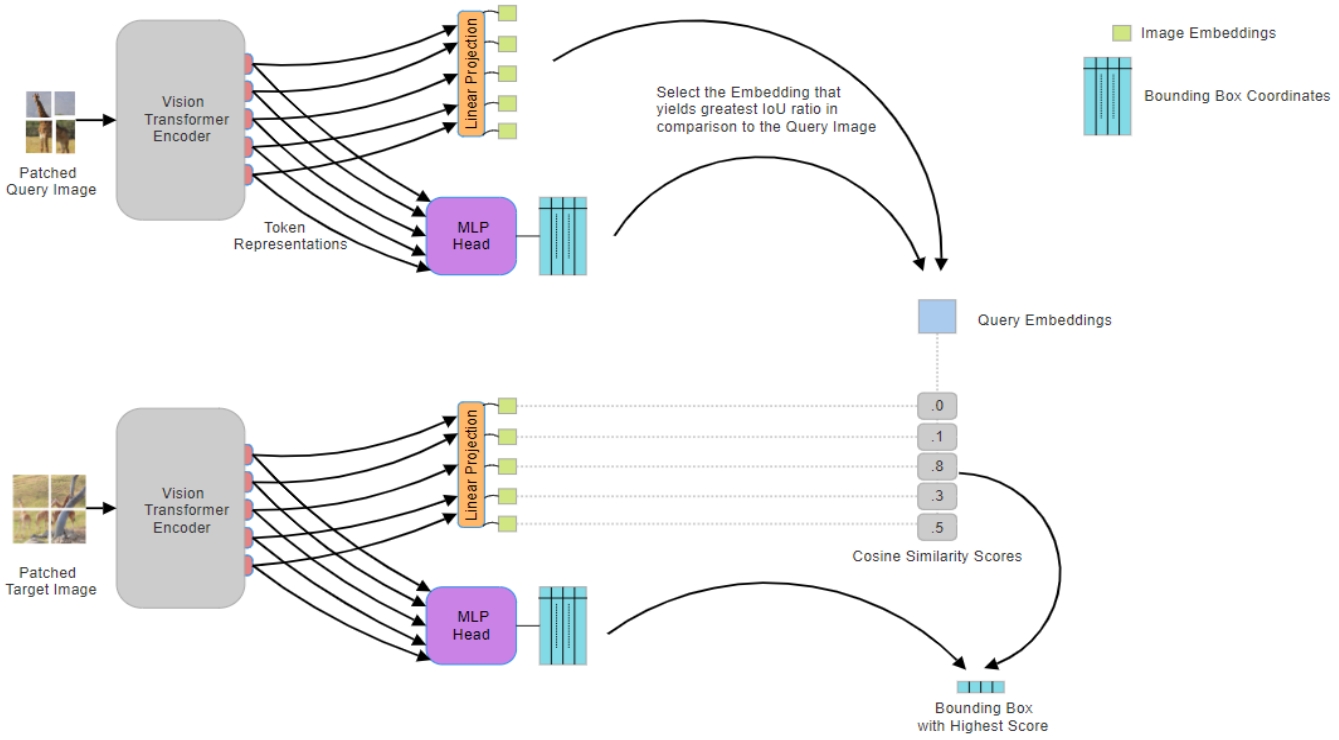


Fig. 1: Architecture of OWL-ViT for Image-Guided Object Detection

B. CLIP [3]

CLIP (Contrastive Language-Image Pre-training) is pre-trained to predict if an image and a text snippet are paired together in its dataset. For each dataset, the names of all classes in the dataset were used as the set of potential text pairings and predicted the most probable (image, text) pair.

The image encoder inside the model is a computer vision backbone that computes a feature representation for the image, and the text encoder is a hypernetwork that generates the weights of a linear classifier based on the text specifying the visual concepts represented by the classes.

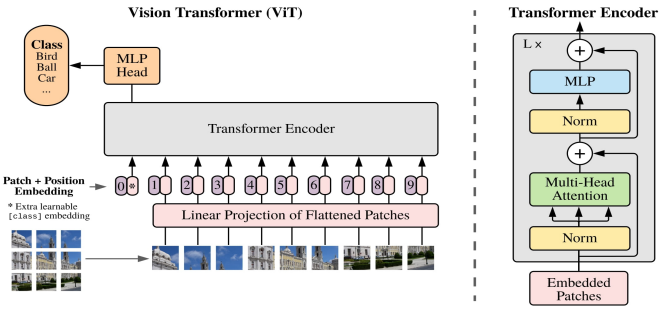


Fig. 2: Architecture of ViT

C. Vision Transformers [7]

While the Transformer [8] architecture has become the de facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while maintaining their overall structure in place. We show that this reliance on CNNs [9] is not necessary, and that a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks as in Figure 2 [10]. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet [11], CIFAR-100 [12], VTAB [13], etc.), the ViT attains excellent results compared to state-of-the-art convolutional networks and requires substantially fewer computational resources to train.

III. Methodology

The objective is to study visual query encoding for deep-object detection models. Nowadays, computer vision models accept multimodal queries, such as images, text, and/or markers, to perform tasks, such as segmentation, as in the case of SAM. In this study, we aim to explore the possibility of training an object detection model with visual queries to train not only the object search/detect module but also the semantic query encoding layers. OWL-ViT is used as the base model to train a one-shot image-guided object detection model on the customized datasets.

A. Model Architecture

In the architecture of OWL-ViT, two fundamental parts undergo the same procedure. These parts extract query and image

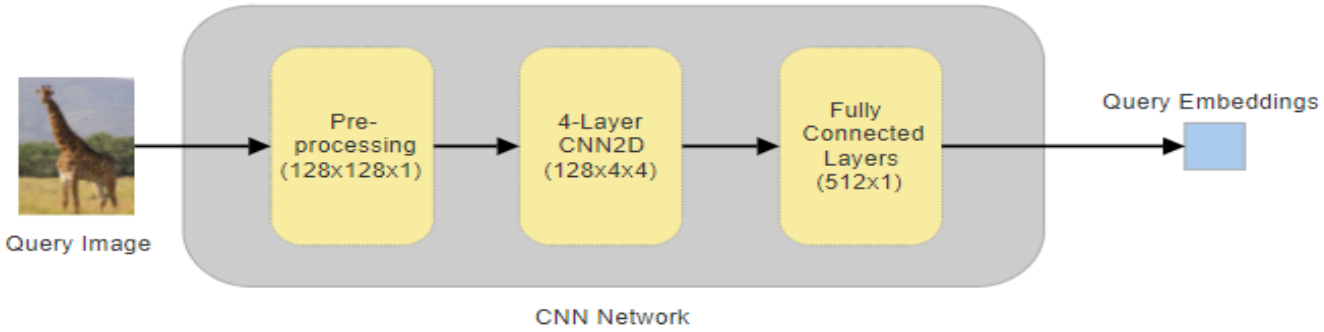


Fig. 3: Proposed Query Embedding Extraction

TABLE I: Customized Datasets used in this study.

Name	Train	Seen Test	Unseen Test	Categories
Digit Detection	280	120	30	10
Object Detection	400	100	25	5

embeddings from the query and target images successively.

To extract query embeddings, a vision transformer is used to obtain the feature vectors (tokens). The tokens contain the information for each image patch. These tokens are then fed into three different processes. First, by linearly projecting tokens onto the embedding dimension, class-embedding vectors for each patch are obtained. Consecutively, tokens go through two different Multilayer Perceptrons (MLPs) [14]. While one calculates a score showing the probability that each token is an object (objectness score), the other finds the corresponding bounding box coordinates for each token. A bias is added to the state location of each patch for the second MLP to return meaningful location coordinates. Finally, the top k class embeddings were chosen based on their objectness scores and used as query embeddings for the target image.

In the other part, the target image passes through the same pre-trained vision transformer. After the composition of image feature vectors, tokens are fed only into the linear projection layer and coordinate calculating MLP because we want to select the necessary image embeddings based on their similarity to query embeddings, not on objectness scores. To match the image and query embeddings, Cosine Similarity is used to obtain output logit scores for each image embedding. Subsequently, the output object(s) are defined as those with the highest scores.

Although the architecture of OWL-ViT is conceptually simple, it may lead to an overkill if the objective is object detection with image queries because the image queries are allowed to be small in size and chosen as basic as possible in order them to visually represent the corresponding object in a better way. Therefore, in one of the experiments, the part where the extraction of query embeddings occurs, is simplified to output only one representative embedding from the object query. This experiment uses a CNN network that is analogous to the one in [15] and has been pre-trained to get a representative embedding for each object query image. Further information about this network is provided in the Model Details section.

B. Training

The OWL-ViT method involves a contrastive pre-training of images and texts, which is accomplished using Vision Transformers (ViT) with Multihead Attention Pooling (MAP) [16]. In this work, we made use of one of the pre-trained OWL-ViT models.

To ensure good performance, a series of training experiments are conducted with parameters close to those reported in the OWL-ViT paper. As the focus of this study is not on the object classification component, the loss used is tailored so that only the object detection function is optimized, for which the bipartite matching loss of DETR was employed with the necessary task-specific normalizations.

IV. Experiments

A. Model Details

ViT's working process begins with a given image being divided into many patches, the size of which is specified by the ViT-B/32 model. To suit this model, the input image must first be preprocessed. Assuming we have a 650x900 image, it must be resized to 768x768 pixels. The encoder then divides this image into 576 patches of 32x32, with each patch's feature representation vector (token) being of size 1x1x1024.

The query image is run through three processes - a linear projection layer to create embeddings of size 1x1x512, a MLP for objectness score and another MLP to return the bounding box coordinates for each token. For the bounding box MLP, center location information of each patch is fed into the MLP as biases to get more accurate bounding boxes. In this case, as the query image contains one object only, the objectness score

TABLE II: AP50 Performance of Image-Conditioned Detection Models

Object Detection		
Method	Seen Test	Unseen Test
OWL-ViT(ViT-B/32)	73.5	58.3
OWL-ViT(ViT-B/32)(ours)	**86.2**	**80.0**
OWL-ViT+CNN(ours)	79.5	68.2
Digit Detection		
OWL-ViT(ViT-B/32)	10.8	19.2
OWL-ViT(ViT-B/32)(ours)	**19.6**	15.0
OWL-ViT+CNN(ours)	13.5	**20.0**

MLP is omitted and the embedding with the biggest bounding box is chosen as the query embedding.

The target image is also put through the linear projection layer and the MLP for box coordinates. As in Equation 1 [17], the Cosine Similarity is then used to compare the target image embeddings to the query embedding and the highest scoring bounding box is the output object in the target image. Target and query image embeddings are expressed as A and B, successively.

$$\cos(\theta) = \frac{\sum_i^n A_i B_i}{\sqrt{\sum_i^n A_i^2}\sqrt{\sum_i^n B_i^2}} \tag{1}$$

Moreover, in order to prevent an overabundance of query embedding extraction, it was opted to utilize a pre-trained CNN Network architecture which only works on one patch at a time instead of 576 different patches just like in Figure 3. One embedding that is created to describe an entire digit or another kind of object query image is generated by this process. The embedding is pre-trained on digit and object detection query images independently for approximately 50 epochs.

B. Dataset

In order to carry out this study, we generated two datasets tailored for this purpose. It should be noted that the pre-trained model applied as the base model was pre-trained on the LiT [18], O365 [19], and VG [20] datasets.

The first dataset was intended for digit detection and consisted of 10 labeled door number images and five digit images for each digit class, with classes 1 and 9 only used in the test set. The training set was composed of 280 pairs of target images and objects, with the test set on seen classes being 120 pairs and test set on unseen classes being 30 pairs as shown in the Table I.

The other dataset contained some of the MS-COCO classes, such as Horse, Car, Person, Airplane, and Elephant, with Elephant chosen as the unseen object class. There were five labeled target images and objects for each class, making the training set 400 pairs and unseen and seen test sets 25 and 100 pairs respectively as shown in the Table I.

Additionally, target images in the customized MS-COCO dataset were labeled using Yolov8 [21], a model that is successful for object detection.

C. Hyperparameters

Hyperparameter settings used in this study are as following

- Learning rate = 10^{-6}
- Weight Decay = 10^{-4}
- Equal weights for bounding box and gIoU losses

D. Loss

The bipartite matching loss introduced by the DETR is used, which is composed of L1 loss and gIoU loss. As in Equation 2 [22], the L1 loss is utilized to reduce the discrepancy between the target and the predicted bounding box by summing up all the absolute differences. To compute the Generalized Intersection over Union, two boxes A and B are firstly enclosed in the smallest convex shape C in S in R^n. As in Equation 3 [23], the ratio of C excluding A and B is then divided by the area of C.

$$L1 = \sum_i^n |y_{true} - y_{predicted}| \tag{2}$$

$$gIoU = \frac{|A \cap B|}{|A \cup B|} - \frac{|C \backslash (A \cup B)|}{|C|} \tag{3}$$

E. Model Performances

As described above, the purpose is to identify corresponding objects in a target image to a query image. Table I shows the distribution of our custom datasets that contain target and query images as pairs. Evaluation is done with the AP50 metric as they did it in [1] and [24]. To compare our results, the results of the pre-trained model we used are also presented in Table II. The train and test losses are shown in Figure 4. Both Digit and Object Detection results clearly demonstrate the improvement of the performance of our image-guided object detection model on both seen and unseen test sets. Additionally, as seen in Table II, the OWL-ViT+CNN model has an edge when tested with unseen categories after it has been trained on the Digit Detection dataset. In contrast, the original OWL-ViT model is more adept at working with the COCO categories in the Object Detection dataset.

V. Discussion

The primary obstacle we face is technical in nature. Developing intricate models necessitates GPUs and improved computers, which we don't have access to. We have been utilizing Google Colab during our work, though it is not the pro version so it has its own restrictions. We did our best to tackle this difficulty.

A. Social, Environmental and Economic Impact

We hope that this study is a positive step towards achieving one-shot image-guided object detection.

B. Cost Analysis

For the project, the duration and probable expenses for an online processor were essential.

C. Standards

The project will be held with special emphasis to IEEE, IET, EU and Turkish standards. Engineering code of conduct will be followed.

VI. Conclusion

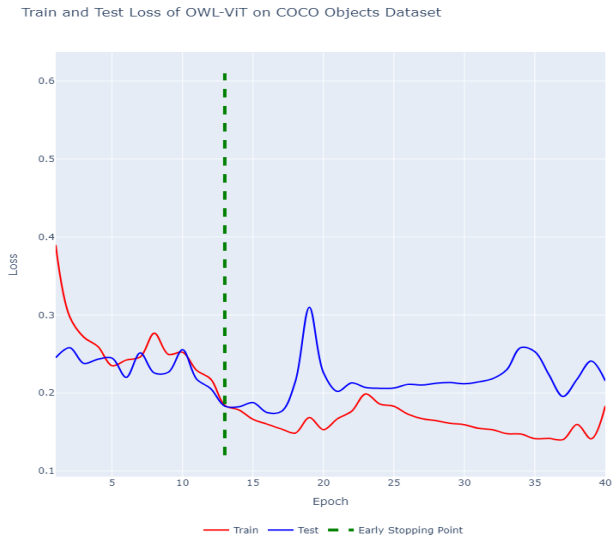
In summary, we employed OWL-ViT as the basis to construct a model of image-aided object detection which is also capable of detecting objects it has not encountered yet. Several experiments were conducted in which certain modifications were made to the original OWL-ViT model design to boost its performance. Based on the results, we are confident that we have taken a significant step forward.

Acknowledgement

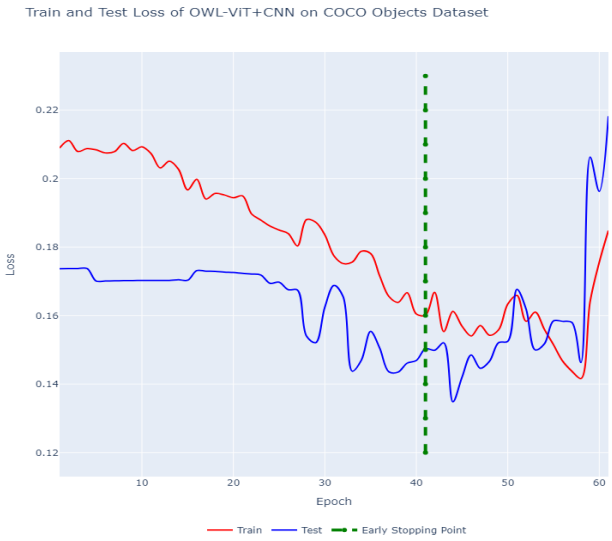
Over the course of the study, I had a great experience working closely with Burak ACAR and Zafer DOĞAN. I owe them my gratitude for guiding me and providing me with assistance during this process. I would like to express my appreciation for their help.

APPENDIX

A. *Losses*

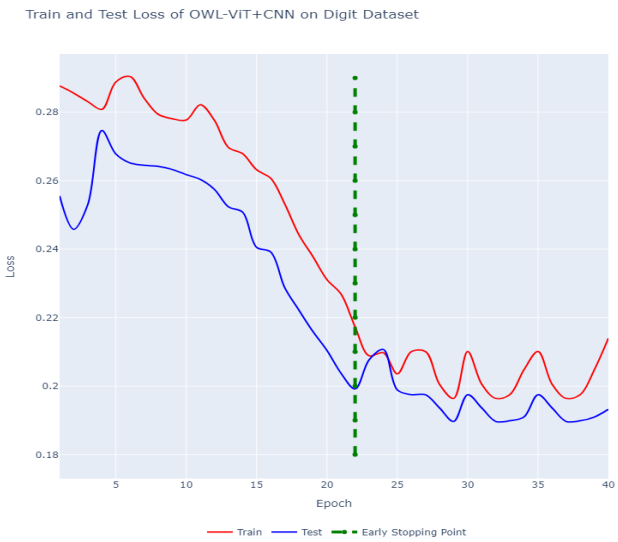


OWL-ViT on Object Detection Dataset



OWL-ViT+CNN on Object Detection Dataset

OWL-ViT on Digit Detection Dataset

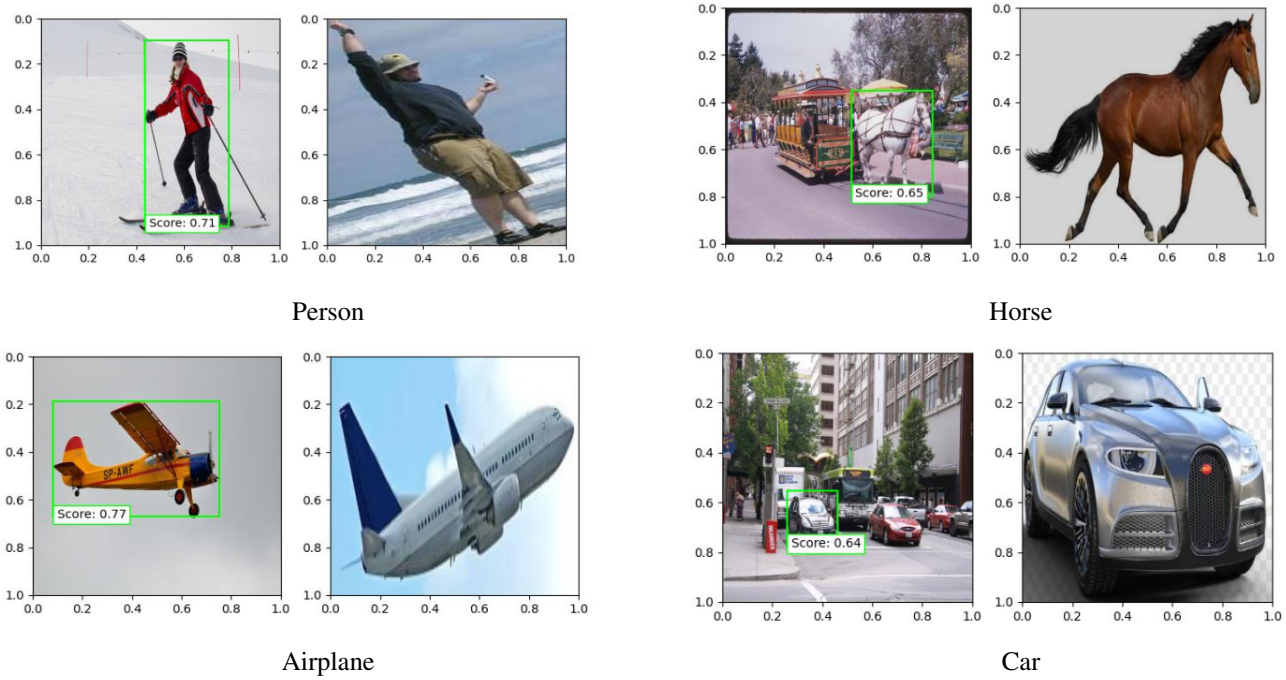


OWL-ViT+CNN on Digit Detection Dataset

Fig. 4: Losses

B. *Qualitative Examples*

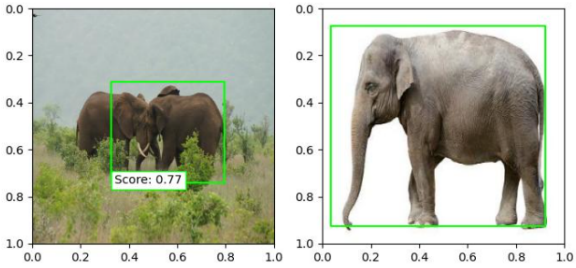
The figures below shows query object images on the right while target images are situated on the left.



Person Horse

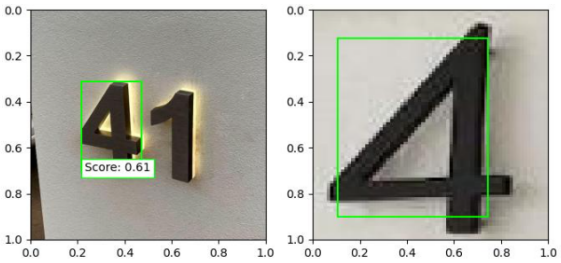
Airplane Car

Fig. 5: Previously Seen Query Examples for Object Detection

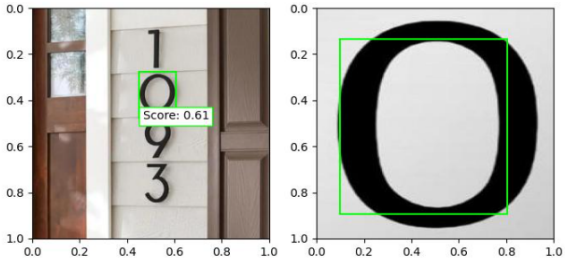


Elephant

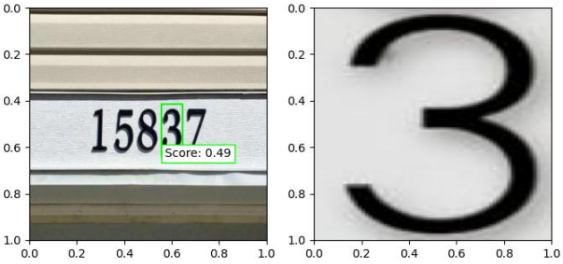
Fig. 6: Unseen Query Example for Object Detection



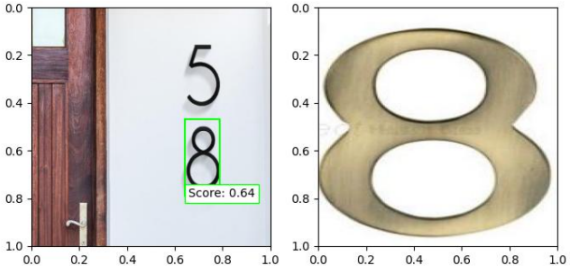
Person



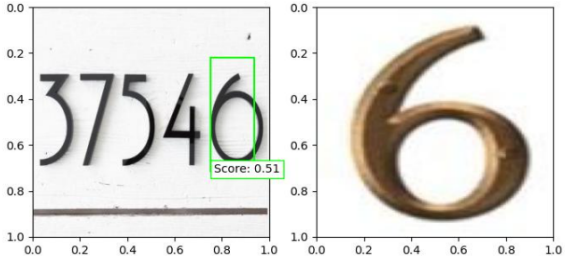
Horse



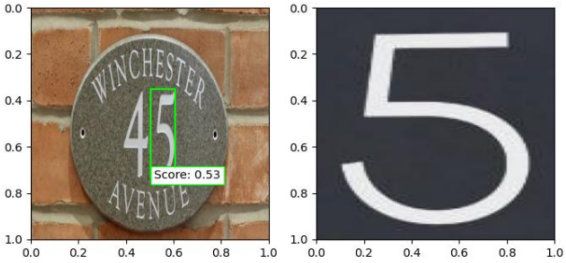
Airplane



Car

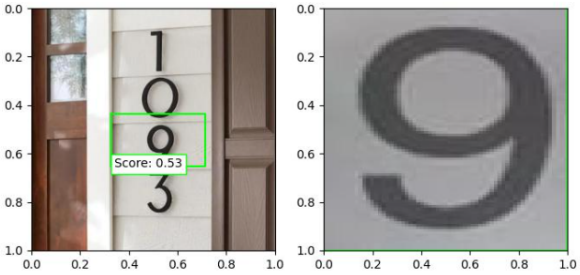


Car



Car

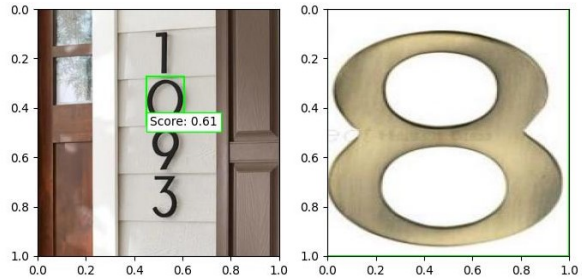
Fig. 7: Previously Seen Query Examples for Digit Detection



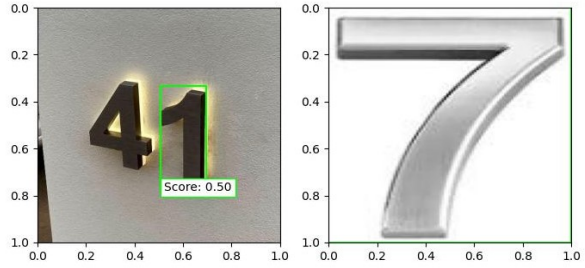
Elephant

Fig. 8: Unseen Query Example for Digit Detection

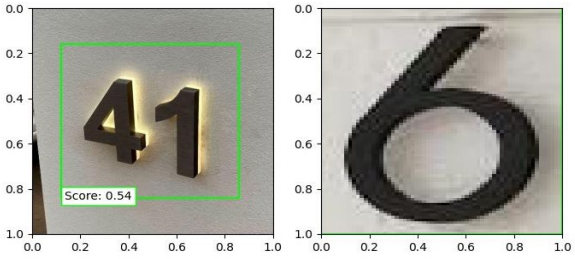
C. Failures



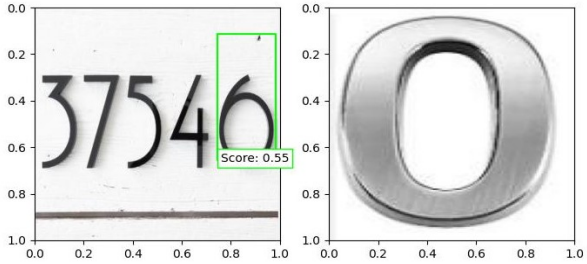
Eight



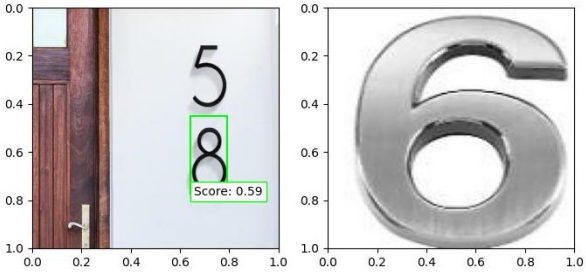
Seven



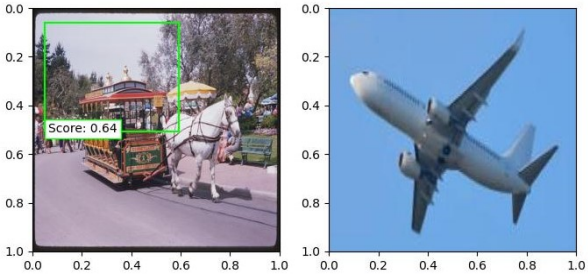
Six



Zero

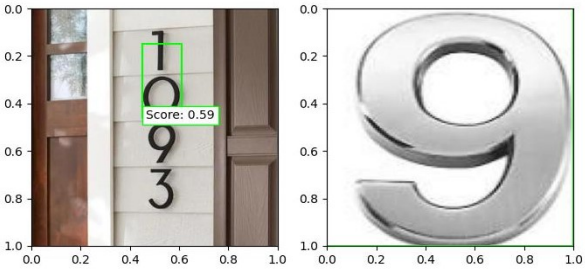


Six

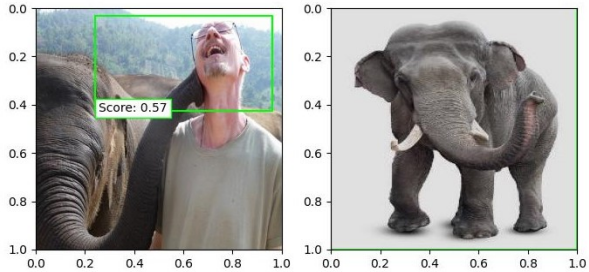


Airplane

Fig. 9: Previously Seen Query Failures



Nine



Elephant

Fig. 10: Unseen Query Failures

REFERENCES

[1] Minderer, M., "Simple Open-Vocabulary Object Detection with Vision Transformers," , pp. 1–28.

[2] Kirillov, A., "Segment Anything," April 2023, pp. 1–30.

[3] Radford, A., "Learning Transferable Visual Models From Natural Language Supervision," February 2021, pp. 1–48.

[4] Carion, N., "End-to-End Object Detection with Transformers," .

[5] Ren, S., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," January 2016, pp. 1–114.

[6] "COCO," https://cocodataset.org/#home.

[7] Dosovitskiy, A., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," October 2020.

[8] Vaswani, A., "Attention Is All You Need," August 2023, pp. 1–15.

[9] O'Shea, K., "An Introduction to Convolutional Neural Networks," December 2015, pp. 1–11.

[10] "ViT," https://huggingface.co/docs/transformers/model_doc/vit.

[11] Russakovsky, O., "ImageNet Large Scale Visual Recognition Challenge," January 2015, pp. 1–43.

[12] Krizhevsky, A., "Learning Multiple Layers of Features from Tiny Images," April 2009, pp. 1–60.

[13] Zhai, X., "A Large-scale Study of Representation Learning with the Visual Task Adaptation Benchmark," February 2020, pp. 1–33.

[14] Tolstikhin, I., "MLP-Mixer: An all-MLP Architecture for Vision," June 2021, pp. 1–16.

[15] "CNN Embedding Network Example," https://www.kaggle.com/code/timurrashitov/metric-learning-embedding-generation-on-mnist.

[16] "MAP," https://d2l.ai/chapter_attention-mechanisms-and-transformers/multihead-attention.html.

[17] "Cosine Similarity," https://en.wikipedia.org/wiki/Cosine_similarity.

[18] "LiT Dataset," https://paperswithcode.com/dataset/lit-pcba-aldh1.

[19] "O365 Dataset," https://www.objects365.org/overview.html.

[20] "VG Dataset," https://paperswithcode.com/dataset/visual-genome.

[21] "Yolov8," https://docs.ultralytics.com/.

[22] "L1 Loss," https://amitshekhar.me/blog/l1-and-l2-loss-functions.

[23] "gIoU Loss," https://giou.stanford.edu/.

[24] Hsieh, T.-I., "One-Shot Object Detection with Co-Attention and Co-Excitation," November 2019, pp. 1–10.