

Master Thesis



Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Open Vocabulary Object Detection with Multimodal and Generative Models

Nikita Sokovnin

Supervisor: doc. Ing. Tomáš Pajdla, PhD.

Study program: Open Informatics

Specialisation: Computer Vision and Image Processing

May 2024

I. Personal and study details

Student's name: **Sokovnin Nikita** Personal ID number: **483744**
Faculty / Institute: **Faculty of Electrical Engineering**
Department / Institute: **Department of Cybernetics**
Study program: **Open Informatics**
Specialisation: **Computer Vision and Image Processing**

II. Master's thesis details

Master's thesis title in English:

Open Vocabulary Object Detection with Multimodal and Generative Models

Master's thesis title in Czech:

Detekce objektů s otevřeným slovníkem pomocí multimodálních a generativních modelů

Guidelines:

- 1) Review Open Vocabulary Object Detection [1] and Zero-Shot Image Classification [2] and Image Generation in the context of Open Set Recognition [3].
- 2) Propose a method for (1) identifying unknown objects, (2) retrieving labels for unknown objects using Language-Image models, and (3) training new classes using Multimodal and Generative Models [4].
- 3) Design an experiment to demonstrate and evaluate the learning of new unknown classes for small real-time models using large Language-Image models.

Bibliography / sources:

- [1] Minderer, Matthias, Alexey Gritsenko, and Neil Houlsby. "Scaling Open-Vocabulary Object Detection." arXiv preprint arXiv:2306.09683 (2023).
- [2] Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International conference on machine learning. PMLR, 2021.
- [3] Li, Yuheng, et al. "Gligen: Open-set grounded text-to-image generation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.
- [4] Podell, Dustin, et al. "Sdxl: Improving latent diffusion models for high-resolution image synthesis." arXiv preprint arXiv:2307.01952 (2023).

Name and workplace of master's thesis supervisor:

doc. Ing. Tomáš Pajdla, Ph.D. Applied Algebra and Geometry, CIIRC

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **15.02.2024** Deadline for master's thesis submission: **24.05.2024**

Assignment valid until: **21.09.2025**

doc. Ing. Tomáš Pajdla, Ph.D.
Supervisor's signature

prof. Dr. Ing. Jan Kybic
Head of department's signature

prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature

Acknowledgements

I want to thank my supervisor, doc. Ing. Tomáš Pajdla, Ph.D., for guiding me through engaging projects. Additionally, I am grateful to the Czech Technical University for the opportunity to study and participate in various interesting projects. I am thankful for everyone I have encountered during my academic journey.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Nikita Sokovnin

Prague, 24. May 2024

Abstract

Our study explores open-set classification strategies using various architectures like Transformers and CNNs. We introduce Matrix Entropy for multi-view open-set classification, demonstrating its superior performance and simplicity. Our research underscores the benefits of aggregating multiple views of the same object for classification accuracy. We bridge the knowledge gap for unknown classes through pseudo-annotation with large vision-language models and data generation using Stable Diffusion models, leveraging the DataDreamer library. By combining pseudo-annotated real data and synthetic data, we achieve optimal performance. Additionally, we propose a pipeline that enables a small model to continuously learn under the supervision of larger foundation models. Our findings underscore the effectiveness of these approaches in handling unknown classes and enhancing classification and detection performance.

Keywords: open set recognition, object detection, image classification, image generation, zero-shot classification, computer vision

Supervisor: doc. Ing. Tomáš Pajdla, PhD.


Abstrakt

Naše studie zkoumá strategie klasifikace otevřené množiny pomocí různých architektur, jako jsou Transformátory a CNN. Představujeme metodu Matrix Entropy pro klasifikaci otevřené množiny z více pohledů, která se ukazuje jako nadřazená a jednoduchá. Zjistili jsme, že agregace více pohledů na stejný objekt přináší výhody v klasifikaci. Adresujeme znalostní mezery pro neznámé třídy pomocí pseudo-annotace s využitím velkých jazykovo-obrazových modelů a generací dat s modely Stable Diffusion s využitím knihovny DataDreamer. Kombinace pseudo-annotovaných reálných dat a syntetických dat přináší optimální výsledky. Navíc navrhujeme postup, který umožňuje malému modelu nepřetržitě se učit pod dohledem větších základních modelů. Naše zjištění zdůrazňuje účinnost těchto přístupů při zpracování neznámých tříd a zlepšování výkonnosti klasifikace a detekce.

Klíčová slova: rozpoznávání ve světě bez omezení, detekce objektů, klasifikace obrazů, generování obrázků, klasifikace bez učení, počítačové vidění

Contents

1 Introduction	1	5.4 Open set object detection	35
1.1 Contribution	2	5.4.1 Automatic label retrieval	37
1.2 Overview of chapters	2	5.4.2 Training on novel classes	37
2 Related work	3	5.5 Results discussion	41
2.1 Out of distribution detection	3	5.6 Future work	42
2.2 Open set recognition	3	5.6.1 Multiple Pipeline Iterations	42
2.3 Open world recognition	4	5.6.2 Closing the Synth2Real Gap	42
2.4 Multiview image classification	4	5.6.3 Large Scale	42
2.5 Contrastive Vision-Language Pre-Training	4	6 Conclusion	43
2.6 Open Vocabulary object detection	5	A Bibliography	45
2.7 Synthetic data for Image Recognition	5	B AI Tools Used	51
2.8 Text-to-Image Diffusion Models	6		
3 Approach	7		
3.1 Problem specification	7		
3.2 Approach overview	8		
3.3 Approach details	8		
3.3.1 Multiview open-set object classification	8		
3.3.2 Pseudo-annotation with large Vision-Language models	11		
3.3.3 Synthetic dataset generation	16		
4 Implementation	19		
4.1 Toolkit	19		
4.1.1 YOLOv8	19		
4.1.2 DataDreamer	19		
4.2 Datasets	21		
4.2.1 Imagenette	21		
4.2.2 Tiny ImageNet	21		
4.2.3 Multi-View RGB-D Object Dataset	21		
4.2.4 PASCAL VOC	22		
5 Experiments	23		
5.1 Multi-view Open-Set Classification	23		
5.1.1 Multiview results	25		
5.2 Single view open set classification: simple dataset	26		
5.2.1 Open set performance	26		
5.2.2 Pseudo-annotation of unknown instances with CLIP	28		
5.2.3 Synthetic dataset	29		
5.3 Single view open set classification: complex dataset	33		



Chapter 1

Introduction

Most modern computer vision models used in industry operate with a limited number of classes. The simplest way to handle unknown classes is to reject them. However, with the advancement of models that integrate text and images, the challenge of dealing with an infinite number of real-world classes has become more apparent. Language supervision for vision learning enables models to operate with a much larger set of classes [30][27].

Our work aims to combine both approaches to improve edge real-time models that lack the capacity to learn rich image representations compared to foundation language-image models. When an edge model encounters unknown objects, we can use the supervision of larger, more capable models to close or minimize the knowledge gap.

Another challenge arises when real data is not available, and we need to extract knowledge from large language-image models. For this, we can use text-conditioned generative models [29][22][21]. Realistic synthetic images generated in this way can serve as valuable data sources. These images can be pseudo-annotated by zero-shot or open-vocabulary models to create a dataset to train the edge model.

Theoretically, this loop can run infinitely and can be viewed as follows: the student real-time model operates in the real world and continuously encounters objects it has not seen during training. Images with unknown objects are passed to the teacher network for annotation, and optionally another network generates examples for further annotation. This allows the student network to learn about all the objects it encounters continuously. In this analogy, we use the terms student and teacher, but this approach has little in common with the traditional distillation method, where these terms are commonly used.

To further investigate these concepts, we evaluate various open-set classification strategies using different architectures, including Transformers (Vision Transformer [7]) and CNNs (YOLOv8 [17]). We introduce Matrix Entropy for multi-view open-set classification, which outperforms existing techniques and is easy to implement. Our multi-view and single-view scenario assessments show significant benefits from aggregating multiple views of the same object.

We investigate two methods to close the knowledge gap for unknown classes. The first method involves utilizing large language-image models such as CLIP

for classification and OWLv2 for detection. The second approach involves data generation using Stable Diffusion [33] models. The best performance is achieved by combining pseudo-annotated real data with synthetic data.

In our work, we observe that pseudo-annotation with large language-image models on unknown data nearly matches the quality of training on ground truth data for classification and detection. Synthetic data effectively represent novel classes, although managing distribution shifts is crucial. Combining pseudo-annotated and real images mitigates distribution shifts and enhances data augmentation, improving overall performance. The code is accessible at <https://github.com/sokovnin/open-world-object-detection>.

1.1 Contribution

Our key contributions are:

- We review various Open Vocabulary Detection and zero-shot Image Classification methods and discuss their connection to open set recognition.
- We explore the use of generative models to augment existing datasets with novel classes.
- We test multiple open-set classification methods in a multi-view setting and propose new methods that outperform existing ones.
- We evaluate the open-set performance of Transformer-based and CNN-based networks and report best practices.
- We propose a method that allows small real-time models to learn continuously by detecting unknown instances and leveraging supervision from large language-image models.
- We experiment with the automatic creation of pseudo-labeled datasets using the DataDreamer ¹ library, which we developed for this purpose.

1.2 Overview of chapters

- Chapter 2 reviews and categorizes recent advancements in open set recognition, Open Vocabulary Detection, zero-shot image classification and image generation laying the theoretical groundwork for our approach.
- In Chapter 3, we formalize our problem and propose the method to address the task.
- Chapter 4 contains implementation details of our pipeline.
- Chapter 5 presents the results of conducted experiments.
- Finally, Chapter 6 summarizes our findings.

¹<https://github.com/luxonis/datadreamer>

Chapter 2

Related work

2.1 Out of distribution detection

Detection of anomalous patterns or instances that do not fit the distribution is one of the important tasks in computer vision. It can be applied to different domains, such as image classification, object detection, and even more complex domains, such as semantic segmentation. In [9], the authors introduce a method that allows the identification of novel objects for pixel-level annotations. They combine conventional parametric anomaly scores (max-logit) and the non-parametric Nearest-Neighbor method. This simple approach achieves state-of-the-art results. The study also compares features from different backbone architectures, such as conventional CNN and Vision Transformer [7] and shows that ViT is especially efficient for Out-of-Distribution Detection.

2.2 Open set recognition

OSR [10] is closely related to the OoD problem. In contradiction with OoD, OSR focuses not only on identifying novel data but also on the accurate classification of previously seen instances into multiple classes. While OoD focuses on all types of deviation from the distribution, OSR emphasizes semantic novelty. The assumption that labels during train and test time are drawn from the same feature space does not hold for a real world in which there is an infinite number of labels that appear only during test time. By adding an "unknown" class, we require a classifier to classify seen classes and deal with unseen ones accurately. [39] demonstrates a high correlation between closed-set and open-set performance. The authors of this work significantly improve the baseline (Maximum Softmax Probability or MSP) by leveraging closed-set-related techniques such as longer training, label smoothing, and reach augmentations. Thus, they achieve results comparable to the SOTA methods with their improved baseline. This work highlights that models based on the Vision Transformer have better OSR performance compared to other architectures.

2.3 Open world recognition

In an open-world setting, a recognition system must continually discover new classes and update itself with minimal downtime. Open-world recognition, as formally defined in [1], requires the system to accurately classify known and unknown classes while incrementally learning new ones through labeled instances. This scalable system continuously enhances its knowledge about the open world. The Open World Object Detection problem was formulated in [18], which is the most challenging as it combines object detection, open set recognition, and incremental learning. The proposed solution involves modifying the object detector by explicitly adding an unknown class to the dataset, incorporating contrastive clustering to learn discriminative clusters, integrating an energy-based unknown identifier, and storing a balanced set of class representatives to prevent forgetting. We follow a similar pipeline in our work, introducing new ideas for each step.

2.4 Multiview image classification

Typically, there is one view per object when we talk about image classification. However, a single view is sometimes insufficient to make an accurate decision. It is especially relevant for objects characterized by high inter-class similarity and intra-class variability. Different views of the same object are expected to provide complementary information in such classification problems. In [36], the authors evaluate multiple fusion techniques utilizing features of trained CNNs and show that combining multiple views increases classification accuracy. It was found that the fusion of the late CNN stages results in the most accurate decisions. In our setting, we demonstrate that the information from multiple views is also beneficial in terms of open-set performance.

2.5 Contrastive Vision-Language Pre-Training

Using language supervision for vision training has shown remarkable robustness to natural distribution shifts and impressive zero-shot performance. Zero-shot classification is the task of predicting a class that was not seen by the model during training or was not specifically trained. The main concept of Contrastive Vision-Language Pre-Training involves learning visual perception from natural language supervision. This approach is efficient and scalable, enabling learning image representations from scratch using a dataset of millions or billions of image-text pairs collected from the Internet. CLIP [30] and ALIGN [15] are both models for visual representation learning using natural language supervision in a contrastive learning setting. They differ in their approach to training data and the architectures of their vision and language encoders. CLIP constructs its data set by creating a list of high-frequency visual concepts from English Wikipedia, while ALIGN uses raw alt-text data without expert curation.

2.6 Open Vocabulary object detection

Many works aim to transfer the open-vocabulary capabilities of Vision-Language models to object detection. The main challenge is preventing the model from forgetting its existing knowledge while training the detection heads on limited data. GLIP [20] uses a single text query for the entire image and treats detection as a phrase grounding problem, limiting the number of object categories processed per forward pass. OWL-ViT [27] features a simpler and more flexible architecture that avoids image-text fusion and can handle multiple independent text or image-derived queries. OWLv2 [26] improves performance with a more efficient architecture and self-training, scaling up the data used for object detection training. YOLO-World [5] highlights the strong open-vocabulary performance of lightweight detectors like YOLO [31][17], which is crucial for real-world applications. Instead of relying on an online vocabulary, it introduces a prompt-then-detect approach for efficient inference, where users generate prompts as needed. It pre-encodes the prompts or categories to build an offline vocabulary and then seamlessly integrates it into the detector. However, the smallest variant of YOLO-World has poor open vocabulary performance, and the largest cannot run in real-time on edge devices. In our pipeline, we use OWLv2 as it shows the best results to date.

2.7 Synthetic data for Image Recognition

There are two main approaches to generating synthetic data: 1) using traditional renderings of 3D objects in simulation and 2) using generative models.

Synthetic datasets, created using traditional pipelines, involve generating 2D renderings of 3D models from graphics engines [8][32]. However, this method has several drawbacks: 1) a gap often exists between synthetic and real-world data; 2) they require significant storage space and are costly to share; 3) the specific source limits the amount and diversity of data.

Generative models offer a more efficient way to create synthetic data compared to traditional methods. They have several advantages: 1) they produce high-fidelity, photorealistic images closer to real data since they are trained on real-world data; 2) they require much less storage space; 3) they can generate virtually unlimited amounts of data. Few studies have explored generative models for image recognition. Some works used GAN-based generators to produce data for image classification [2], object part segmentation [42], and unsupervised contrastive learning [14]. These works demonstrated promising results, but the scope and image quality were limited. The most relevant work to ours is [12], which extensively studies the usage of text-to-image diffusion models in different settings: 1) there are no or only a few examples of real images per class; 2) large-scale model pre-training using synthetic data. This work demonstrated that synthetic data is beneficial in such scenarios.

2.8 Text-to-Image Diffusion Models

Diffusion models [13] have recently emerged as powerful generative models. Using a likelihood-based approach, they match the data distribution by learning to reverse a noise process. This allows for novel image sampling from a prior Gaussian distribution via the learned reverse path. With high sample quality, good mode coverage, and stable training, diffusion models are quickly becoming a trend in both unconditional and conditional image synthesis. In text-to-image generation, they excel by ensuring that sampled images match given natural language descriptions. Models such as Stable Diffusion [33], DALL-E3 [3] and SDXL [29], based on the formulation of diffusion models, offer an exceptional synthesis quality. Models such as SDXL-Turbo [35], LCM [23], and SDXL-Lightning [22] offer close-to-real-time image synthesis by distilling larger models and reducing the number of steps required to generate an image. GLIGEN [21] extends pre-trained text-to-image diffusion models with grounding capabilities, enabling image generation based on various grounding conditions such as bounding boxes, key points, depth maps, and semantic maps. While text + box conditioned generation accurately produces objects inside bounding boxes, it does not guarantee that target objects are not generated in the background. Therefore, this method will also require pseudo-annotations generated by large vision-language models.

Chapter 3

Approach

In the following chapter, we will describe the method for detecting unknown objects, labeling them, and incorporating them into the dataset. The problem specification is based on our previous work [28].

3.1 Problem specification

We use the following definitions of known and unknown classes:

1. **Known classes** K : classes that have labelled instances in the training dataset. A known object (or simply known) is an instance of a known class.
2. **Unknown classes** U : classes which are not trained, no example of these classes is labelled. Instances of these classes might be present in the background at training time. U is very similar to mixed unknown \mathcal{U}_M . An unknown object (or simply unknown) is an instance of an unknown class.

First of all, we have a set of known classes $K = \{1, \dots, C\}$, where C is the number of classes present during training. Additionally, there is an unlimited set of unknown classes $U = \{C + 1, \dots\}$, which may be encountered during inference. Let 0 be the label associated with an “unknown” class. Each known class is assumed to be represented in the dataset $D = \{X, Y\}$ with cardinality $|D| = M$, where X is a sequence of images $X = \{I_1, \dots, I_M\}$ and Y is a sequence of matching labels $Y = \{L_1, \dots, L_M\}$. L_i encodes k_i instances $\{y_1, \dots, y_k\}$ present in an image I_i . Each label y_j is represented as $y_j = [u, v, w, h, c, s]$, where u, v are the coordinates of the center of the bounding box, w, h are the width and height of the bounding box, c is the class of the object, and s is a segmentation mask (which can be omitted for default object detectors). For the classification problem, $y_j = [c, s]$. The above refers to the object detection model M_d , which produces predictions for a single image at each timestep (or M_c for classification).

A Large Vision-Language model can provide true labels for these objects when the model identifies unknown objects $\{u_1, \dots, u_n\}$. Optionally, based on the retrieved classes, a generation model can produce a dataset with

training examples containing instances of the identified unknown objects to update M_d . Retraining should extend the model’s knowledge about existing classes. After updating itself with new classes, M_d can identify test instances belonging to some class from the updated set $K' = K \cup \{C + 1, \dots, C + n\}$.

The task for the small real-time model is to continuously detect unknown objects and learn by using the supervision of large Vision-Language models.

3.2 Approach overview

Our approach consists of six main steps: 1) training the model on known classes, 2) testing/inferencing the model in the real world and detecting unknown objects, 3) label retrieval, 4) data generation, 5) pseudo-annotation, and 6) model retraining on new data. The approach is illustrated in Figure 3.1.

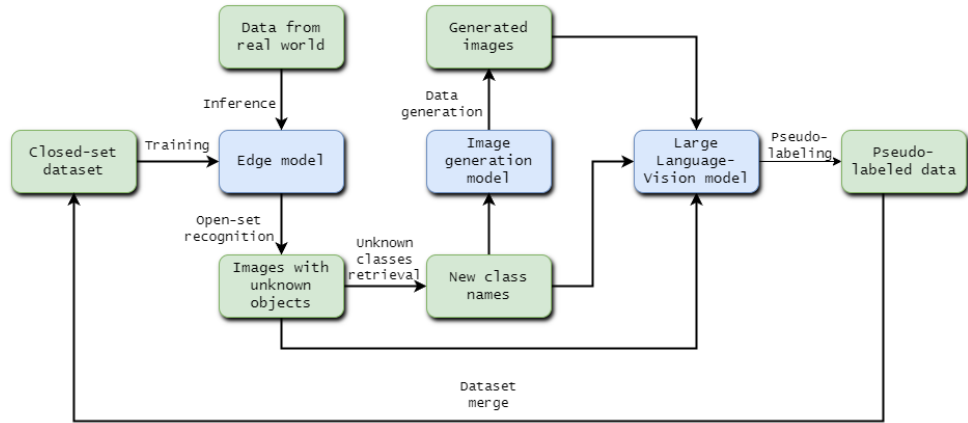


Figure 3.1: Approach scheme. The edge model trained on closed-set data receives data from the real world and classifies some objects as unknown. Labels for new objects are then retrieved from the unlabeled unknown images, either automatically or with human supervision. Image generation models create images containing the new objects. Finally, synthetic and real images are pseudo-annotated using a Language-Image model with zero-shot capability. The resulting dataset is merged with the initial one, and the entire process can be repeated.

3.3 Approach details

In this section, we describe the main components of the proposed pipeline.

3.3.1 Multiview open-set object classification

In Computer Vision, many extensive studies exist on both closed-set and open-set problems (OSR). Usually, they utilize a single view of the object to make a prediction. Most large modern datasets such as ImageNet [34] are single-view. It is much harder to construct a dataset with a high variability of

objects where each object is captured from multiple views. Under the "view," we understand the image of the object captured from the specific position of the camera. While making a decision where only one view is available is a crucial problem for many applications, multiview settings might be beneficial for many problems. Some works, such as [36], show that using multiple views of the same object leads to a better closed-set performance of the classifier, especially for problems with large inter-class visual similarity and low intra-class similarity. Our work shows that utilizing multiple views also benefits Open Set Recognition. Information from different views helps to understand better whether the object is from the novel unseen class. Previous work showed that the Vision Transformer [39][9] outperforms other architectures in OSR tasks. We use ViT to extract features from images and use them to compute anomaly scores, which are used for known/unknown classification.

■ Anomaly score

Here, we introduce simple methods to compute the anomaly score a , which is used to decide whether the object is known or unknown. An anomaly score is computed from the outputs of a Neural Network trained in a closed-set setting. We use the following variables: \bar{s} - average softmax vector over N views, \bar{l} - average logit vector over N views, the output of the last layer before applying softmax, q_i - average output of the penultimate fully connected layer for i^{th} view. In our case, the dimension of \bar{s} and \bar{l} is ten, which is the number of known classes. And the dimension of \bar{q} is 768. For the first two methods, we consider an object as unknown if $a < \theta$. For the other two, we say that the object is unknown if $a > \theta$, where θ is some threshold.

Maximum Softmax Probability (MSP) is often used as a baseline in Open Set Recognition problems. If the confidence of the strongest class is too low, we say that this object is unknown.

$$a_{MSP} = \max(\bar{s})$$

Maximum Logit Score (MLS). The authors of [39] propose to use the MLS instead of the softmax probabilities as an open-set indicator. Logits are outputs of the last layer in the network before softmax is applied, which normalizes them. In [6], it was noticed that the logits of unknown instances have a lower magnitude.

$$a_{MLS} = \max(\bar{l})$$

Average cluster center distance (CCD). As pointed out in [25], unknown objects tend to be farther away in the feature space from the center of clusters consisting of known object's features. Here, we compute an average vector produced by the penultimate layer of the trained network on the training data and call it c_i for the i^{th} class. We compute an average cluster center distance (CCD) to measure novelty.

$$a_{CCD} = \frac{1}{N} \sum_i^N \|q_i - c_{\hat{y}}\| \quad \text{where } \hat{y} = \text{argmax}(\bar{s})$$

Entropy. Another way to reject unknown objects is to measure the epistemic uncertainty of the detector. Instead of using the maximal probability, we can use the full softmax vector with all class probabilities. In [24] \bar{s} is treated as an average vector of class probabilities over a set of score vectors produced by multiple forward passes of the same data with enabled dropout. In other words, it is the average result of the model ensemble. In our case, \bar{s} represents the mean score vector on multiple views of the same 3D object. Therefore, uncertainty may arise when the model is not able to classify the same object from different perspectives identically. To measure uncertainty, we can use the entropy of the probability vector:

$$a_e = H(\bar{s}) = - \sum_{i=0}^C \bar{s}_i * \log(\bar{s}_i)$$

where C is the number of known classes. If \bar{s} has a uniform distribution, the entropy will be large, so the uncertainty is high. If the class probability is concentrated in one class, the entropy will be low, which means that the confidence of the classifier is high.

Matrix entropy. Another way to measure uncertainty, which exploits the consistency between feature vectors obtained from different views, is by computing a matrix (or SVD) entropy. SVD entropy indicates the number of eigenvectors needed for a good data explanation. In other words, it measures the dimensionality of the data. In our case, if the vectors corresponding to different views of the object differ too much, SVD will produce many non-zero singular values, resulting in higher entropy. The SVD entropy is defined as:

$$a_{SVD} = H(Y) = - \sum_{i=1}^M \sigma_i * \log(\sigma_i)$$

where M is the number of singular values of the embedded matrix Y and $\sigma_1, \sigma_2, \dots, \sigma_M$ are the normalized singular values of Y . Y is the matrix that contains feature vectors obtained from different views.

GMM score Instead of using a single cluster center to represent the class, we can use Gaussian Mixture Models (GMM) [?]. We obtain a measure of epistemic uncertainty for each known class by computing the log-likelihood of the data l for every known class model G_i

$$a_{GMM} = P = (\log(p(l; G_1)), \dots, \log(p(l; G_N)))$$

A low log-likelihood represents a high uncertainty that the detected object belongs to the respective known class. To identify and reject potential open-set detections, we can choose a minimum log-likelihood threshold θ_{OSE} and reject detections that do not meet this threshold for at least one known class.

We use the same approach for single-view object detection, but the number of views is 1.

3.3.2 Pseudo-annotation with large Vision-Language models

Contrastive Language-Image Pre-training (CLIP)

As discussed previously, many state-of-the-art computer vision systems are only trained to recognize a fixed set of predefined object categories. This limited form of supervision restricts their versatility and practicality because additional labeled data is required to recognize new visual concepts. CLIP [30] tackles this issue by introducing a pre-training task: predicting which caption matches with which image. The core idea is learning perception from supervision contained in natural language. This method, which is efficient and scalable, allows for learning image representations from scratch using a dataset of 400 million image-text pairs collected from the Internet. Following pre-training, natural language refers to learned visual concepts or describes new ones, enabling the model to transfer seamlessly to new tasks without prior training.

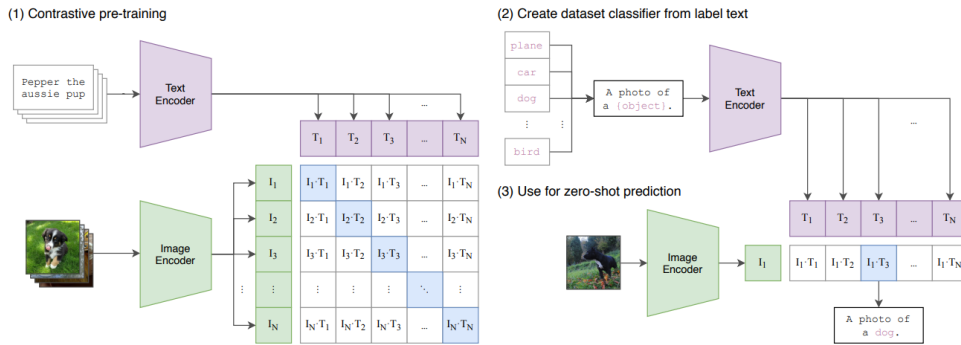


Figure 3.2: CLIP operates differently from traditional image models. Instead of training an image feature extractor and a linear classifier together to predict a label, CLIP simultaneously trains an image encoder and a text encoder to identify the correct matches in a set of (image, text) training pairs. During testing, the trained text encoder creates a zero-shot linear classifier by embedding class names or descriptions from the target dataset. Figure from [30]

Architecture and training

CLIP learns to match image text pairs in the batch of size N ($N \times N$ possible image-text pairings) by joint training of image and text encoders. The task is to maximize the cosine similarity between image and text embeddings for the N actual pairs in the batch and minimize the similarity between $N^2 - N$ incorrect pairs. In order to achieve this, a symmetric cross-entropy loss is optimized.

For the image encoder, Vision Transformer (ViT) [7] architecture is used, which outperforms CNN-based ResNet in both accuracy and compute efficiency. The text encoder is a regular Transformer [38] with slight modifica-

	Dataset Examples	ImageNet ResNet101	Zero-Shot CLIP	Δ Score
ImageNet		76.2	76.2	0%
ImageNetV2		64.3	70.1	+5.8%
ImageNet-R		37.7	88.9	+51.2%
ObjectNet		32.6	72.3	+39.7%
ImageNet Sketch		25.2	60.2	+35.0%
ImageNet-A		2.7	77.1	+74.4%

Figure 3.3: CLIP is more robust than dataset-specific ResNet-101. Figure from [30]

tions.

Given a single image and a set of texts (classes), first image embeddings and all possible matching text embeddings are computed during inference. Then, cosine similarity between all possible image-text pairs is calculated and scaled by a temperature parameter. Finally, the cosine similarities are normalized using softmax.

Strengths

Learning from natural language offers several advantages over other training methods. Scaling natural language supervision is much simpler than traditional crowd-sourced image classification labelling. Methods utilizing natural language can passively learn from the abundance of textual data available on the Internet. Learning from natural language also provides a crucial advantage over most unsupervised or self-supervised learning approaches. Although these approaches primarily learn representations, learning natural language connects these representations with language, enabling flexible zero-shot transfer. Moreover, the performance of the task-agnostic CLIP is competitive with fully-supervised task-specific models.

Another essential feature of CLIP is its robustness to distribution shift, which is superior to standard ImageNet models. Because the training dataset contains a large amount of data, there are minimal distribution-specific images. Figure 3.3 illustrates the effectiveness of this approach.

Weaknesses

While CLIP learns rich image representations which are helpful for common tasks (such as food or car model classification), its zero-shot performance on more specialized, abstract, and complex tasks (such as satellite image classification or counting objects in synthetic scenes) is poor.

CLIP generalizes well to image distribution shift, but it still generalizes poorly to truly out-of-distribution data. For example, simple logistic regression exceeds CLIP on the classification of hand-written digits in MNIST because digitally rendered text is much more common in the training dataset, as shown in [30]. So, its generalization capability is limited. The main naive assumption of CLIP is training on such a large and diverse dataset that all data encountered in the future will be in-distribution. It is limited to choosing from concepts and patterns that were encountered during training.

Open-set and zero-shot performance discussion

CLIP does not really solve the problem of having an infinite number of classes of visual categories in the real world and does not work well on truly out-of-distribution data. Instead, it tries to cover as many visual concepts as possible by seeing a vast amount of data. There might still be unknown classes not encountered in the 400M images dataset, but identifying such objects might be a complex problem.

For our approach, it's important that CLIP allows us to create our own classes without retraining by combining relatively simple known concepts such as "man in a red shirt".

Deep learning models are good at spotting correlations and patterns in the data they are trained on. This helps them perform well within that same kind of data. However, sometimes, these correlations are actually spurious and useless, which can result in a significant performance drop on other datasets.

When we talk about zero-shot models, they should not be affected by these useless correlations because they are not trained on that specific data. The robustness to image distribution shift is crucial since we will work with synthetic images that look different from real ones.

■ Vision Transformer for Open-World Localization

Vision Transformer for Open-World Localization (OWL-ViT) [27] leverages large-scale image-level pre-training similar to CLIP but takes another step to enable zero-shot localization of objects in images. It achieves competitive zero-shot performance with much more complex approaches. For object detection, data labeling is much more time-consuming, so making large-scale datasets with objects localized by humans seems infeasible. OWL-ViT shows that it is possible to effectively transfer the representations obtained by contrastive learning with image-text pairs to detection tasks by fine-tuning on smaller closed-set datasets. It has been found that increasing the size of the pre-training datasets and the size of the model leads to better performance on the downstream object detection task. This suggests that pre-training on billions of image-text examples provides strong generalization ability, which can be transferred to detection tasks even with relatively limited object-level data. OWL-ViT focuses on standard transformer-based models because of their scalability and good close-set detection performance.

Architecture

OWL-ViT uses standard Transformer-based encoders similar to CLIP. To adapt the image encoder for detection, the token pooling and the final

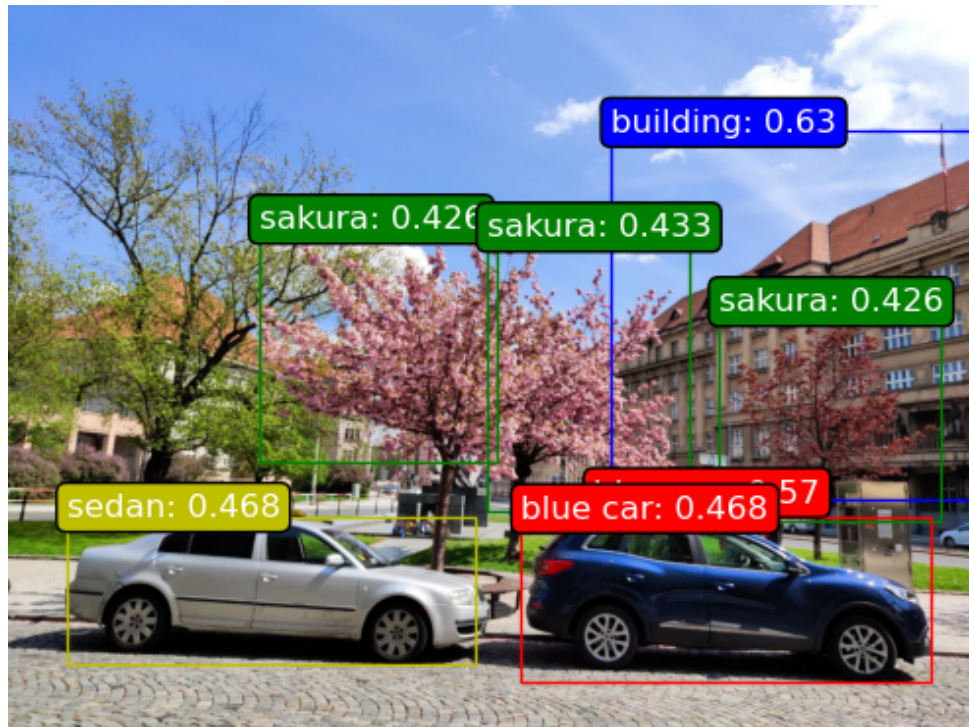


Figure 3.4: OWLv2 demonstrates impressive open vocabulary performance. Any text can be used as target classes for detection, enabling the model to correctly detect specific kinds of trees, such as sakura, and to distinguish car types and colours. OWLv2 can detect objects and properties not represented in standard object detection datasets. The knowledge acquired from weak supervision during contrastive text-image pre-training is effectively transferred to the detection task during fine-tuning.

projection layer are removed, and instead, each output token representation is linearly projected to obtain image embeddings per object for classification (Figure 3.5). The overall setup resembles DETR [4] but is simplified by removing the decoder.

For open-vocabulary classification of detected objects, OWL-ViT uses text embeddings instead of learned class embeddings in the classification head. These text embeddings, called queries, are generated by passing category names or object descriptions through a text encoder. The model’s task is to predict a bounding box and the probability of each query applying to each object. This means each image has its own set of text-based labels. This method includes traditional closed-vocabulary object detection as a special case where all object category names are used as queries for each image. Each query is a separate token sequence processed individually by the text encoder. It is also possible to use image-derived embeddings as queries in the classification head without modifying the model. The model can perform image-conditioned one-shot object detection by using embeddings of prototypical object images as queries. This method enables the detection of objects that are difficult to describe in text.

Training

Image and text models are initially pre-trained on the image level and then fine-tuned on object-level annotations. Pre-training is performed from scratch using a dataset of 3.6 billion image-text pairs similar to LiT [40]. After pre-training, detection heads are added and fine-tuned on medium-sized detection data. All components of the fine-tuning approach aim to reduce overfitting on the relatively small number of available detection annotations and the limited semantic label space they cover. For example, the learning rate for the text encoder is 100x smaller than for the image encoder. This may reduce overfitting by preventing the text encoder from "forgetting" the semantics learned during pre-training while fine-tuning on a small set of detection labels. Interestingly, completely freezing the text encoder yields poor results.

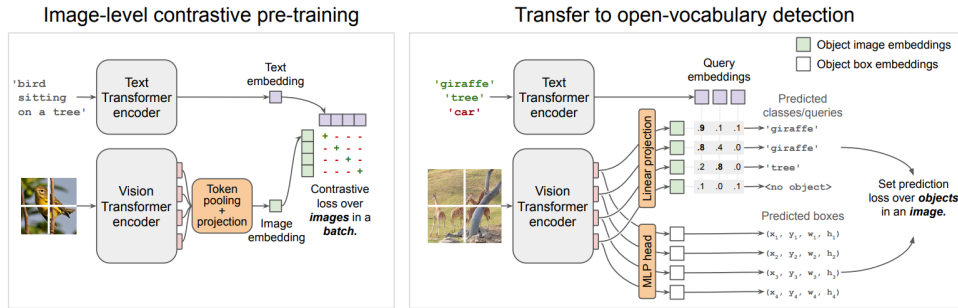


Figure 3.5: Open-Vocabulary Object Detection with ViT [27] Overview. Left: Initially, the image and text encoder are pre-trained using image-text pairs in a contrastive manner similar to CLIP. Right: Subsequently, the pre-trained encoders are adapted for open-vocabulary object detection by eliminating token pooling and integrating lightweight object classification and localization heads directly onto the output tokens of the image encoder. For open-vocabulary detection, query strings are processed through the text encoder for classification purposes. The model undergoes fine-tuning using conventional detection datasets. During inference, the model utilizes embeddings derived from text for open-vocabulary detection or from images for few-shot image-conditioned detection.

Scaling Open-Vocabulary Object Detection

The scarcity of detection data can be addressed with self-training. In self-training, an existing detector (OWL-ViT) predicts bounding boxes on unlabeled images to generate data for training better detectors. Combining open-vocabulary detectors with web image-text data allows this pseudo-labeling to create nearly unlimited amounts of open-vocabulary detection training data using image-associated text for semantic supervision. Scaling Open-Vocabulary Object Detection [26] introduces the OWLv2 model, which has improved training efficiency, and a self-training recipe called OWL-ST, which together largely improve open vocabulary and zero-shot performance.

The self-training approach consists of three steps:

1. Utilize an existing open-vocabulary detector to predict bounding boxes for a large Web image-text dataset.

2. Self-train a new detector using the pseudo-annotations.
3. Optionally, the self-trained model can be briefly fine-tuned on human-annotated detection data.

All 10 billion images from the pre-training phase are annotated with bounding box pseudo-annotations using OWL-ViT CLIP-L/14.

Open-Vocabulary and zero-shot detection performance discussion

To measure performance on unseen categories, the authors of OWL-ViT remove all box annotations for rare categories. The zero-shot performance for the model is measured in the sense that the model has not seen localized annotations for these categories. However, these categories might have occurred during contrastive image-text pre-training and during self-training. Also, during fine-tuning the removal of bounding boxes for some categories, it might lead to the model being learned to ignore or classify objects from these categories as background. So, the performance on this rare category does not represent true zero-shot performance when target objects are not present during training. The results presented in [27] also suggest that Transformers may be more biased toward learning semantic generalization than CNN or Transformer-CNN hybrids, which is crucial for high zero-shot performance and could be advantageous with large-scale pre-training. So, similar to CLIP, good open vocabulary and zero-shot performance are achieved due to the huge amount of images seen.

3.3.3 Synthetic dataset generation

To generate images for novel categories, we use image generation models. In the following text, we briefly describe the chosen models.

Stable Diffusion XL (SDXL)

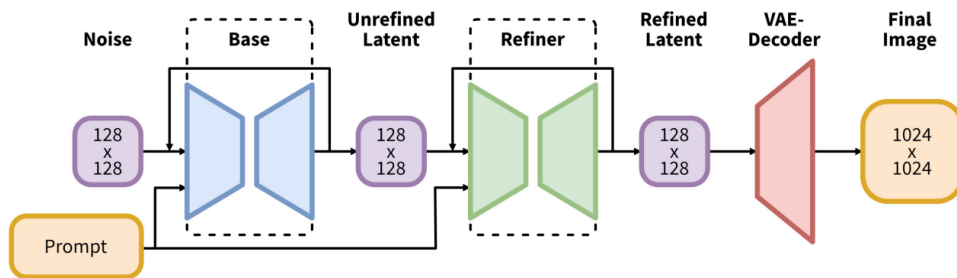


Figure 3.6: SDXL architecture. Figure from [29]

Stable Diffusion XL [29] is a Latent Diffusion Model (LDM) for text-to-image synthesis. Based on the given text/prompt, the model generates an image that follows this prompt, which can be used as a caption for the image. Figure 3.6 shows the architecture of the model.

Latent Diffusion Models (LDM) are probabilistic models that learn data distributions by iteratively removing noise from a normally distributed variable. This approach involves a sequence of denoising autoencoders trained to predict a cleaned-up version of their input. By doing so, LDM abstracts away subtle details, creating a compact, efficient latent space suitable for generative modeling. This space focuses on essential semantic features and enables computationally efficient training. The generation can be conditioned with semantic maps, text, representations, or images. In our work, we only use text conditioning.

■ SDXL-Lightning

SDXL-Lightning [22] proposes an effective way to distil SDXL that results in much faster generation due to a lower number of required generation steps with a slight loss in generated image quality. Figure 3.7 shows the images generated with both models. We can see that SDXL-Lightning generates slightly less realistic images, but it is more than ten times faster than the undistilled model.



Figure 3.7: Visual comparison of images generated by SDXL and SDXL-Lightning using the prompt "A photo of a dog walking in the park." Left: SDXL with 65 steps. Right: SDXL-Lightning with 4 steps.

Chapter 4

Implementation

This chapter briefly describes tools, datasets, and implementation details used in our work.

4.1 Toolkit

4.1.1 YOLOv8

Ultralytics YOLOv8 [17] represents the latest advancement in object detection models, leveraging the achievements of previous YOLO iterations while integrating novel features and enhancements to enhance both performance and adaptability. YOLOv8 prioritizes speed, precision, and user-friendliness, making it the top pick for various tasks, including object detection and image classification.

In our experiments, we use its smallest and fastest version nano with 3.2M parameters.

4.1.2 DataDreamer

In modern AI, data plays a crucial role. Collecting data usually takes the most time when training a model for a particular task. However, consider the possibility of skipping this step altogether and creating a Computer Vision model without using real-world data. The only requirement is to provide the names of the objects to be identified or classified.

Consider a scenario requiring an application to identify robots in videos and images. DataDreamer ¹ simplifies this process, allowing users to generate thousands of annotated images with just one command. This innovative approach saves time and expands the possibilities for AI development, freeing it from the constraints of traditional data collection methods.

DataDreamer is a library that facilitates the creation of custom datasets covering virtually any imaginable class, starting from scratch. This process is broken down into three key steps.

¹<https://github.com/luxonis/datadreamer>

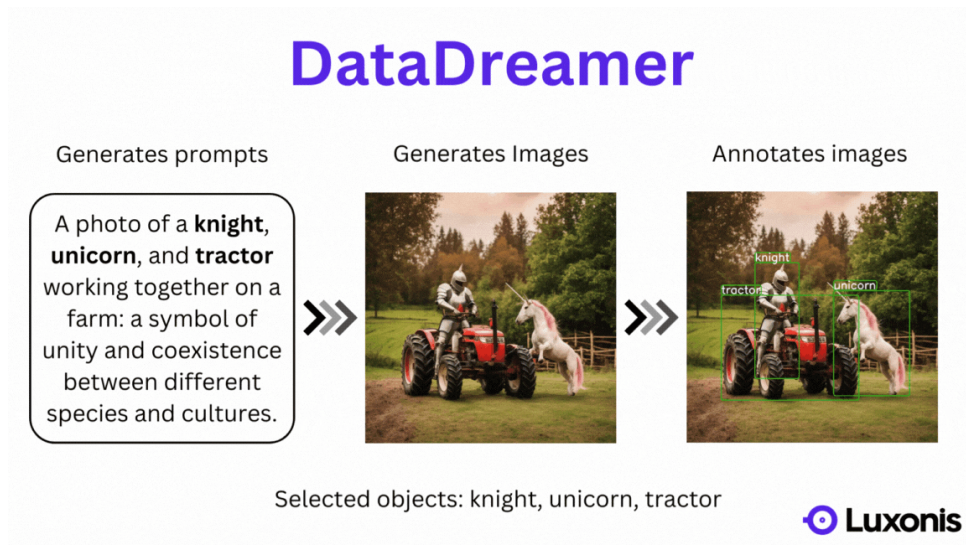


Figure 4.1: DataDreamer pipeline

■ Prompt Generation

During this step, DataDreamer automatically generates prompts for the image generation models based on a list of class names. While the most straightforward approach involves concatenating target objects in the prompt, such as "A photo of a knight, unicorn, and tractor," research suggests that richer prompts, which also describe the interaction between objects and the scene, lead to a better dataset quality [12].

DataDreamer leverages language models like Mistral-7B-Instruct [16] and TinyLLama [41] to generate semantically rich prompts, thereby enhancing the diversity of the dataset. The prompt format used for a language model is as follows:

```
f"[INST] Generate a short and concise caption for an image.
Follow this template: 'A photo of {' , '.join(selected_objects)}',
where the objects interact in a meaningful way within a scene,
complete with a short scene description. [/INST]"
```

Consequently, generated prompts may look like this: "A photo of a knight, unicorn, and tractor working together on a farm."

■ Image Generation

DataDreamer provides a solution to reduce the reliance on large datasets for AI training. It allows users to create synthetic datasets from scratch using advanced generative models. These models can generate diverse, high-quality images customized to specific requirements.

Users can choose between three image generators within DataDreamer. The first option is Stable Diffusion XL [29], which is known for its fidelity to

prompts and ability to produce images of exceptional quality. However, this comes at the cost of slower generation speed. Alternatively, users can opt for SDXL-Turbo [35] or SDXL-Lightning [22], which offers a faster generation time albeit with a slight trade-off in image fidelity compared to Stable Diffusion XL.

To enhance image quality, the prompts are refined by appending ", hd, 8k, highly detailed" at the end.

■ Image Annotation

In the final stage, the framework employs foundation models such as CLIP and OWLv2 to label the generated images. This process depends on the initial class names provided, ensuring accurate labeling of each image according to the specifications.

The preceding two steps can be skipped to annotate unlabeled real images.

Several strategies are employed to improve the annotations, such as Test Time Augmentation (TTA), usage of synonyms for class names, and careful selection of the confidence/IOU thresholds.

■ 4.2 Datasets

■ 4.2.1 Imagenette

Imagenette is a small dataset that contains 10 simple-to-identify classes from Imagenet. These include objects like tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, and parachute. The images in Imagenette are kept at their original size from the Imagenet dataset.

■ 4.2.2 Tiny ImageNet

Tiny ImageNet is a subset of ImageNet [34], which has 200 classes and 500 images per class in the training set. The validation and test datasets have 50 images per class. Usually, it is considered the most challenging case for open-set recognition due to the large number of classes and their complexity. The dimensions of each image are 64×64 .

■ 4.2.3 Multi-View RGB-D Object Dataset

To evaluate the multi-view performance of the proposed model, we use A Large-Scale Hierarchical Multi-View RGB-D Object Dataset [19]. It contains 300 objects common in office and home environments organized into 51 categories. There are 3-10 distinct object instances within each category. Each object is spinning on a turntable at a constant speed. Three cameras mounted at three different angles relative to the turntable (approximately 30° , 45° and 60°) record a video sequence at 20 Hz, which results in around

250 RGB-D images per camera and around 250,000 RGB-D images in total. In our work, we use only RGB data, and depth images are ignored.

■ 4.2.4 PASCAL VOC

The PASCAL VOC dataset is a widely used benchmark for object detection and image segmentation tasks. It contains approximately 17,000 training and 5,000 validation images, with annotations for 20 different object classes. These classes include everyday objects like person, car, dog, and bicycle. Each image in the dataset is annotated with bounding boxes, segmentation masks, and class labels, making it valuable for training and evaluating machine learning models for object detection and image segmentation.

Chapter 5

Experiments

In this section, we conduct multiple experiments to demonstrate and evaluate each part of our approach and test how the entire pipeline works on different datasets.

5.1 Multi-view Open-Set Classification

The authors of [39] showed that there is a roughly linear relationship between the top-1 accuracy used for closed-set performance evaluation and the Area Under the Receiver Operating Characteristic (AUROC) used for open-set performance evaluation. In the following experiment, we use the most prominent architecture for image classification, the Vision Transformer (ViT) [7], specifically its ViT-B/16 variant. We take the pre-trained model on ImageNet and train it on selected classes from Tiny ImageNet for ten epochs, with an input size of 128x128.

The number of common classes between Tiny ImageNet and Multi-View RGB-D Object Dataset is very low. We choose ten classes (keyboard, water bottle, flashlight, pitcher, plate, bell pepper, orange, lemon, banana, coffee mug) that have a one-to-one correspondence and consider them as known. We train ViT only on Tiny ImageNet images from these 10 selected classes; the other 190 classes are not used. Objects from 41 classes in the RGB-D Object Dataset are labeled unknown.

To measure closed-set performance, we use the top-1 accuracy. For open-set performance, we use the Area Under the Receiver Operating Characteristic Curve (AUROC).

In the following text, we will show how multiple views affect the closed-set and open-set performance of the classifier. We run the model multiple times using a different number of views. Each result is averaged over ten runs in which random views are selected.

Feature Embeddings Visualization

To ensure that our approach of measuring the distance to the center of the cluster in the feature space (CCD) is valid, we visualize the features of known and unknown objects in 2D (Figure 5.2). We plot 2D T-SNE latent

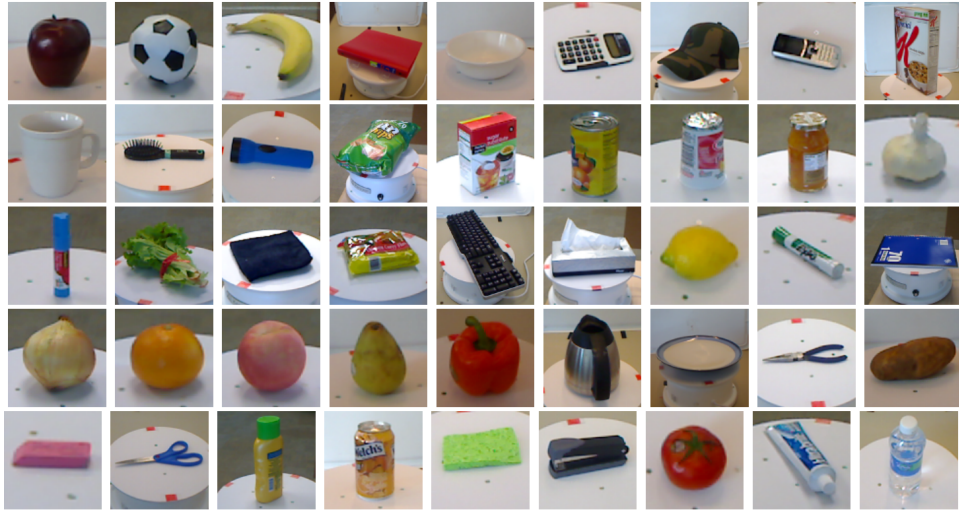


Figure 5.1: Objects from the RGB-D Object Dataset [19]. Each object shown here belongs to a different category. Objects are placed on a turntable, which allows them to be captured from different sides.

representations of 300 objects from the multi-view dataset. As input for 2D T-SNE, we provide outputs of the penultimate layer (with dimension 768) averaged over five different views. We see that known objects form tight clusters and are close to the cluster centers computed from the single-view training dataset. Interestingly, semantically similar objects like "orange" and "lemon" or "pitcher" and "water bottle" are placed close to each other. Figure 5.2 demonstrates that unknown objects are usually farther away from the cluster centers than known ones.

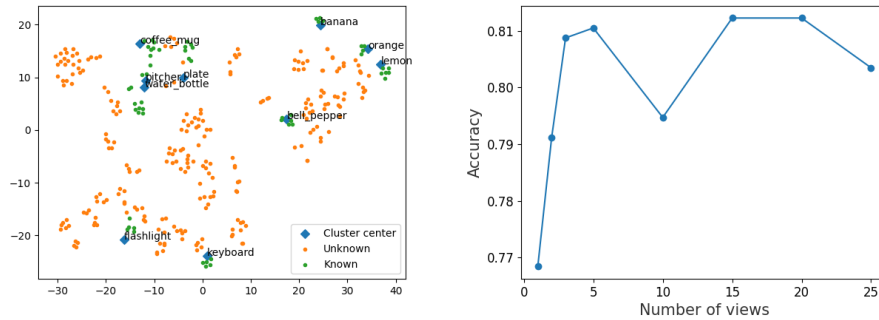


Figure 5.2: The left plot shows 2D T-SNE latent representations of cluster centers, known and unknown objects. The right figure visualizes classification accuracy for a different number of available object views. Each result is averaged over ten runs where random views are selected.

■ Closed Set Performance

After training the classifier on ten Tiny ImageNet classes with a validation accuracy of 95.7%, we tested it on the multi-view dataset. Figure 5.2 shows that the accuracy increases when multiple views are used for classification. Only two additional views result in more than a 3% increase. However, more additional views do not affect performance much. The maximum classification accuracy achieved on the test multi-view dataset is 81.2% (15 and 20 views).

■ Open Set Performance

In a similar setting, we evaluate the open-set classification performance. Figure 5.3 shows that all methods benefit from using multiple views. The greatest boost in performance is observed when only one view is added. Also, we can observe that using more than 20 views is not very beneficial. We can see that using raw output (MLS) instead of softmax probabilities (MSP) results in a large increase in AUROC. CCD, MLS, and entropy methods perform very similarly. However, using the Matrix (SVD) Entropy of probability scores as an uncertainty measure shows the best performance. The maximum AUROC achieved is 0.919 (50 views). GMM score shows the best performance on a low number of views (1-3).

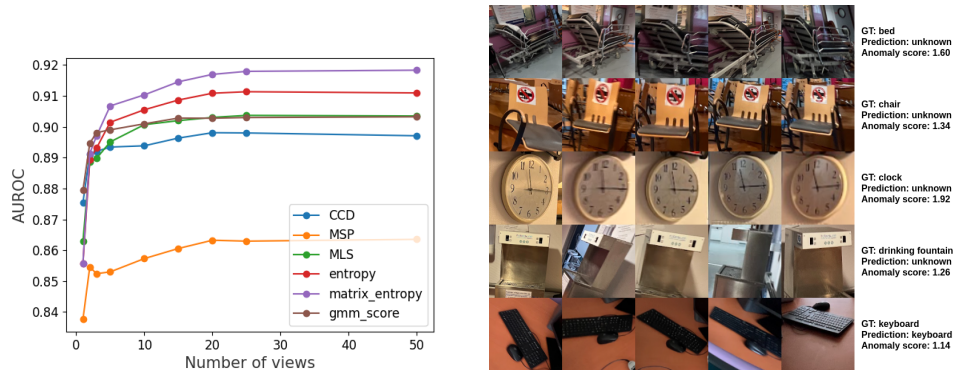


Figure 5.3: The left plot shows the AUROC for different numbers of views. CCD stands for Cluster Center Distance, MSP for Maximum Softmax Probability, and MLS for Maximum Logit Score. These are evaluated on the dataset visualized in Figure 5.1. The right plot shows the evaluation of the best-performing anomaly score, i.e., Matrix entropy (ViT-B trained on 10 classes), on the Broca dataset. Bed, chair, clock, and drinking fountain classes were not used during training.

■ 5.1.1 Multiview results

Table 5.1 shows that the use of multiple views is beneficial for both closed-set and open-set classification. Observing an object from different angles provides richer information about it. If the classifier response from different points of view is similar, then the object can be considered known. On the other

hand, if the object looks different from different sides and the model outputs different predictions, then this object can be classified as unknown. From Table 5.1, we can conclude that 20 views are the best choice in terms of accuracy and AUROC, but 5 views seem optimal since the performance drop is not significant and they are easier/faster to obtain.

Metric/N views	1	2	3	5	10	15	20	25
Accuracy	0.768	0.791	0.808	0.811	0.795	0.812	0.812	0.803
AUROC	0.856	0.891	0.893	0.907	0.910	0.914	0.916	0.918

Table 5.1: Classification results on RGB-D Object Dataset (10 classes are known, 41 are unknown). For AUROC results, the best open-set method (matrix entropy) is used.

We demonstrated that combining features from multiple object views is beneficial for closed-set tasks and open-set recognition. A few additional views of the same object can significantly enhance classification performance. We integrated successful approaches from previous similar studies. We evaluated multiple easy-to-implement methods for computing anomaly scores, finding that Matrix Entropy performs the best in the multi-view setting. This method, also known as matrix entropy, aggregates features from multiple views to determine whether an object belongs to the unknown category. Matrix entropy serves as an indicator of the number of eigenvectors required to explain the dataset adequately. As a result, objects with substantially distinct feature vectors from different views will exhibit higher Matrix Entropy. This method has proven to outperform all the previously tested approaches.

■ 5.2 Single view open set classification: simple dataset

To evaluate our pipeline in a single-view setting, we initially utilize a small dataset called Imagenette, which contains 10 easily distinguishable classes.

■ 5.2.1 Open set performance

We begin by training the model on the entire Imagenette dataset, achieving a 97.7% accuracy. This high accuracy indicates that the classes are visually distinct and easily separable. Next, we remove images from three randomly chosen classes: English springer, tench, and chainsaw. We then train the model on the remaining seven classes and again achieve a 97.7% accuracy.

We use the following commands to train the model:

```

1 # Full dataset
2 yolo classify train model=yolov8n.pth data=datasets/imagenette320
  imgsz=256 epochs=20 batch=8 label_smoothing=0.1 workers=8
  amp=False
3 # Dataset with 7 classes
4 yolo classify train model=yolov8n.pth data=datasets/
  imagenette320_70 imgsz=256 epochs=10 batch=8 label_smoothing
  =0.1 workers=8 amp=False

```

We can now use the same methods for detecting unknown objects as in a multi-view setting, except for matrix entropy. Figure 5.4 shows that most methods perform equally well when classes are easily separable. Surprisingly, thresholding on the maximum logit score (MLS) yields the best AUROC (0.968), while thresholding on entropy performs slightly worse. These results align with observations from previous studies: good closed-set performance leads to good open-set performance, and even simple rules for detecting unknown instances are highly effective in such scenarios.

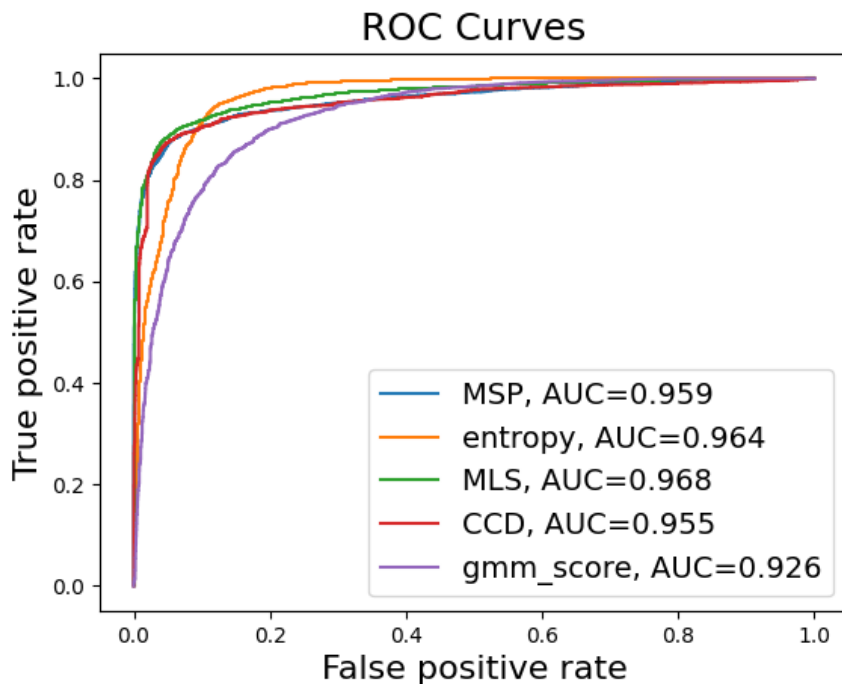


Figure 5.4: ROC curves for different methods on Imagenette with 7 known and 3 unknown classes.

We visualize the logits and softmax vectors for the seven classes in the training set to investigate the open-set performance further. Figure 5.5 shows that the embeddings can be easily projected into 2D space, with images from different classes forming distinct clusters. We also visualize the logit and softmax vectors on the validation set, including unknown objects. Figure

5.6 reveals that unknown objects form a large, separate cluster from known objects, explaining why simple anomaly scores are effective for this dataset. The placement of the unknown cluster in the center of the 2D T-SNE latent representation space indicates that both logits and softmax scores for unknown objects typically have lower magnitudes.

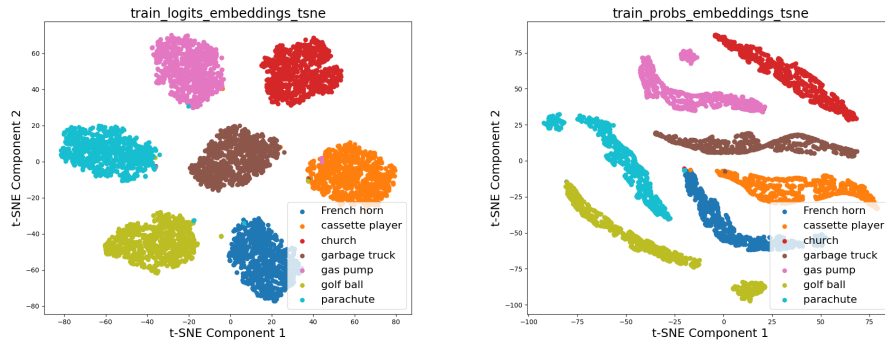


Figure 5.5: Left plot shows 2D T-SNE latent representations of logit embeddings, right plot shows 2D T-SNE latent representations of softmax embeddings. Retrieved from the train dataset.

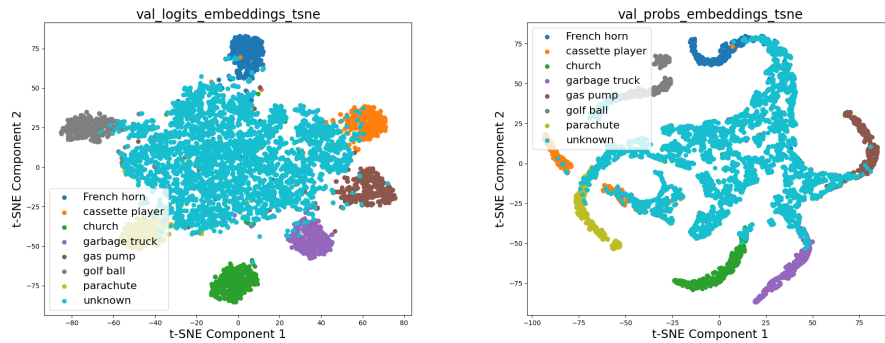


Figure 5.6: Left plot shows 2D T-SNE latent representations of logit embeddings, right plot shows 2D T-SNE latent representations of softmax embeddings. Retrieved from the validation dataset.

5.2.2 Pseudo-annotation of unknown instances with CLIP

The next step in our pipeline is to train the model on the detected unknown objects. We run open-set detection using the MLS score and save the images identified as unknown. Next, we annotate these images with CLIP using DataDreamer:

```

1  datadreamer --save_dir predicted_unknown_imagenette \
2      --class_names "tench" "English_springer" "chain_saw" \
3      --use_tta \
4      --task classification \
5      --image_annotator clip \
6      --annotator_size large \
7      --annotate_only \
8      --dataset_format cls-single \
9      --split_ratios 0.7 0.3 0.0

```

Fine-tuning the model trained on the seven known classes using the merged initial and newly obtained dataset results in an accuracy of 96.8%. This performance closely matches the accuracy achieved with human annotations for all classes in the initial dataset, demonstrating the success of our proposed approach.

```

1  yolo classify train model=imagenette320_70_best_yolov8n.pth data=
    datasets/imagenette320_70_unk_clip imgsz=256 epochs=10 batch
    =8 label_smoothing=0.1 workers=8 amp=False

```

5.2.3 Synthetic dataset

We also investigate the model's performance when synthetic data are used to train the model on previously unknown classes. Data generation is conducted using DataDreamer, with TinyLLama enriching prompts for the image generation model to enhance the diversity of object appearances and scenes. Subsequently, SDXL-Lightning generates images based on these prompts, and CLIP pseudo-annotates the images. Figure 5.7 illustrates the resulting dataset containing both real and synthetic images. In this and the following experiments, we generate approximately the same number of images for each class as in the real dataset. For comparison, Figure 5.8 shows the full Imagenette dataset with real images. It is evident that while synthetic images may appear realistic, they often lack the imperfections and nuances found in real-world photography, appearing as though they were captured under ideal conditions with a high-quality camera and without noise.

We utilize the following command to generate the synthetic dataset:

5. Experiments

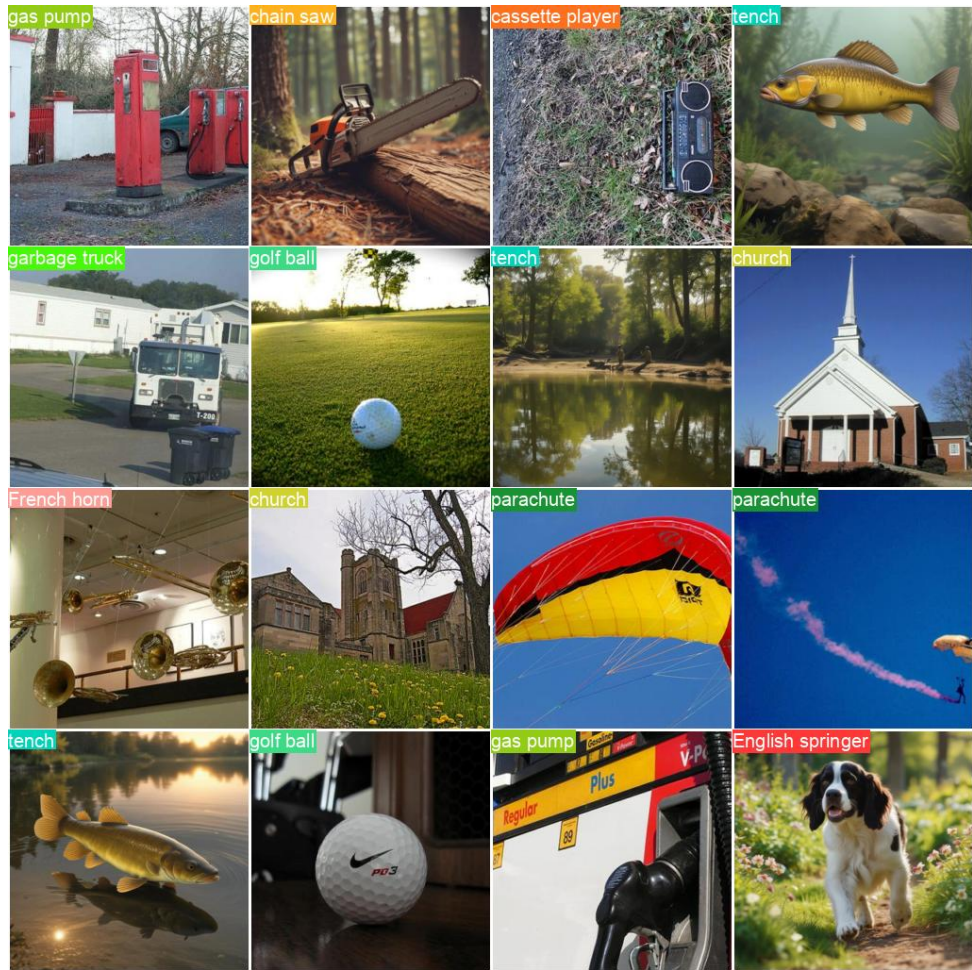


Figure 5.7: The dataset with seven real and three synthetic classes (chain saw, tench, English springer). While the images generally exhibit a high degree of realism, occasional artefacts are present, such as flying fish above the water.

```
1  datadreamer --save_dir imagenette_3classes_4500 \  
2      --class_names "tench" "English_springer" "chain_saw" \  
3      --prompts_number 4500 \  
4      --prompt_generator tiny \  
5      --num_objects_range 1 1 \  
6      --image_generator sdxl-lightning \  
7      --use_tta \  
8      --batch_size_prompt 256 \  
9      --batch_size_image 4 \  
10     --task classification \  
11     --image_annotator clip \  
12     --annotator_size large
```

We find that to achieve the best performance when training on the merged real and synthetic datasets, we need to freeze all layers except the last and

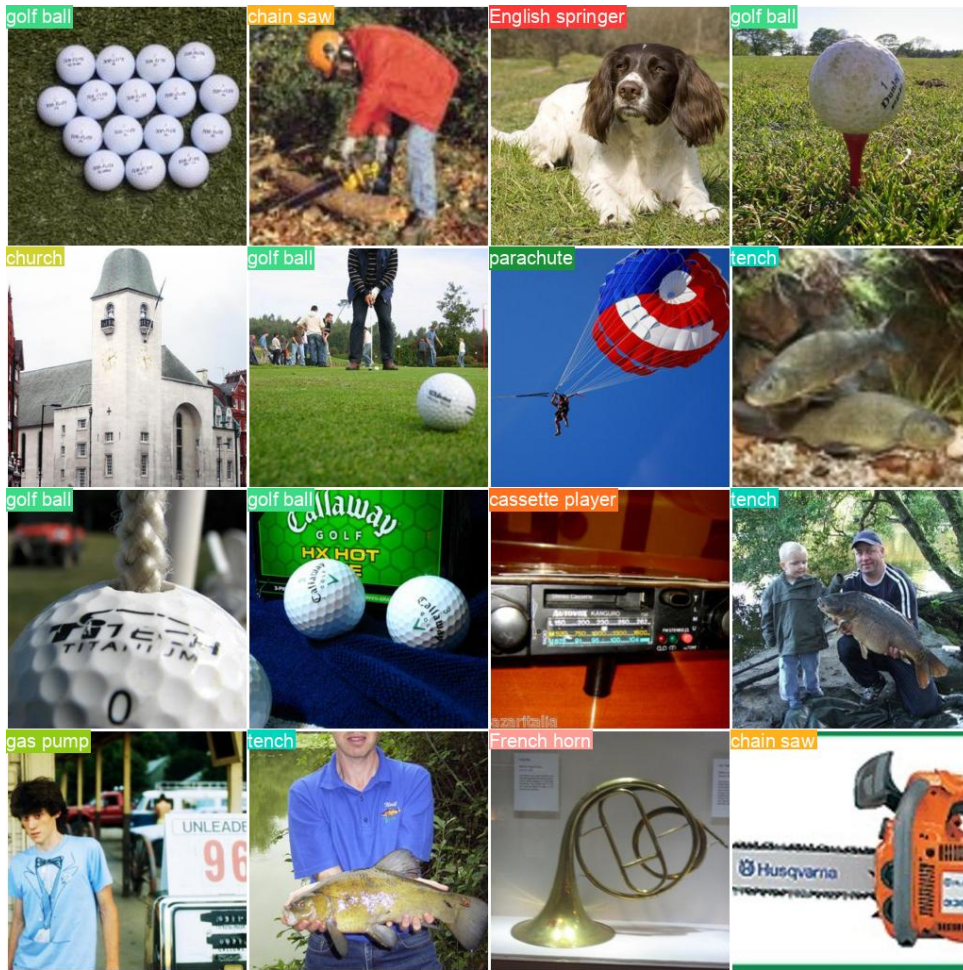


Figure 5.8: Images from the original Imagenette dataset.

fine-tune the model for only 1 epoch, which yields the best results. We obtain an accuracy of 84.1% on the full real dataset when training on the dataset with 4500 images in total and an accuracy of 86.4% when training on a dataset three times larger, with 13500 images. Figure 5.9 illustrates the performance on individual classes, revealing a lower accuracy for synthetic classes. This discrepancy could be attributed to a domain shift between real and synthetic images. A domain shift refers to differences in the distributions of data between different domains. When training models on synthetic data and then applying them to real-world data, the model may struggle due to these distribution discrepancies, leading to a decrease in performance, as observed in this experiment.

```

1 yolo classify train model=imagenette320_70_best_yolov8n.pth data=
  datasets/imagenette320_70_synth_merged imgsz=256 epochs=1
  batch=8 label_smoothing=0.1 workers=8 amp=False freeze=9

```

In the following experiments, we omit the bash commands used for training models, as they mainly differ in the dataset used and the number of epochs.

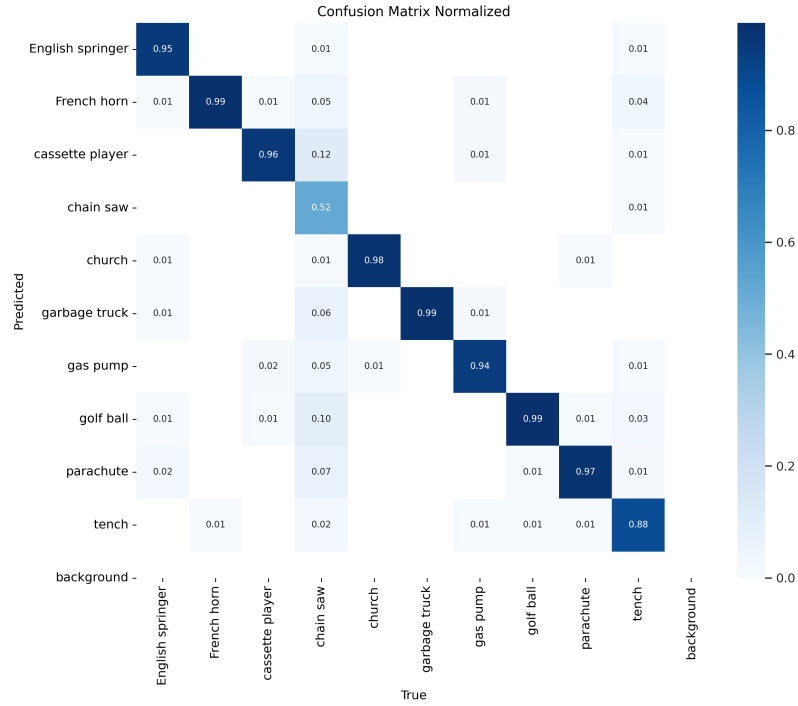


Figure 5.9: The model trained on the dataset comprising 7 real and 3 synthetic classes exhibits lower accuracy for the English springer, chain saw, and tench classes, which consist only of synthetic images.

Figure 5.10 illustrates the comparison of accuracy when the model is trained on different datasets. We observe that the pseudo-annotation of predicted unknown images yields the largest boost, achieving performance close to that of the model trained on the original dataset with 10 classes. Considering the absence of real images of unknown classes, adding synthetic images works quite well, particularly when enlarging the synthetic dataset three times, resulting in a small boost. However, the main challenge for the synthetic dataset lies in its distribution difference compared to real images. Combining pseudo-annotated and synthetic data results in a slight decrease in the examined setting.

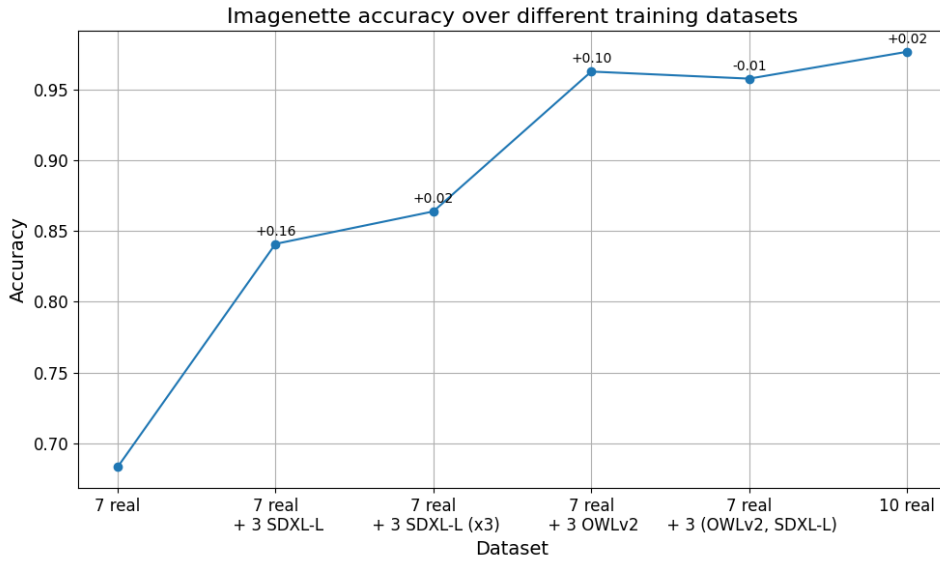


Figure 5.10: Accuracy on the original Imagenette dataset validation set.

5.3 Single view open set classification: complex dataset

Similarly to the previous experiment, we follow the same steps for Tiny ImageNet. We remove 20 random classes and run the open set classification. From Figure 5.11, we observe that the entropy and the MSP score perform the best on more complex data. The complexity of the data is evident from Figure 5.12, where known and unknown classes are no longer easily separable in a 2D space. Since curves have different shapes, the choice of score to use depends on the task requirements (for example, how many false positives we can allow). We choose entropy for the next step, where we detect unknown objects.

We generate pseudo-annotate datasets using commands similar to those for Imagenette but with different classes. We achieve 70.6% accuracy on the full dataset, 69.4% with pseudo-annotated unknown classes by CLIP, again highlighting the power of pseudo-annotated data. However, performance drops significantly to only 64.1% when trained in 180 real + 20 synthetic classes (Figure 5.14). It is evident that the model almost did not learn to correctly classify the real images for 20 new classes, indicating that it learned a synthetic-specific distribution for unknown classes, which resulted in failure on real classes. To further investigate this issue, we train the model only on the 20 synthetic classes and evaluate it on the 20 real classes, resulting in 59.8% accuracy on real data (subset with 20 classes). The per-class accuracies are visualized in Figure 5.13. This suggests that while achieving good performance with synthetic data is possible, the difference in real and synthetic data distribution should be addressed to improve performance on real-world tasks.

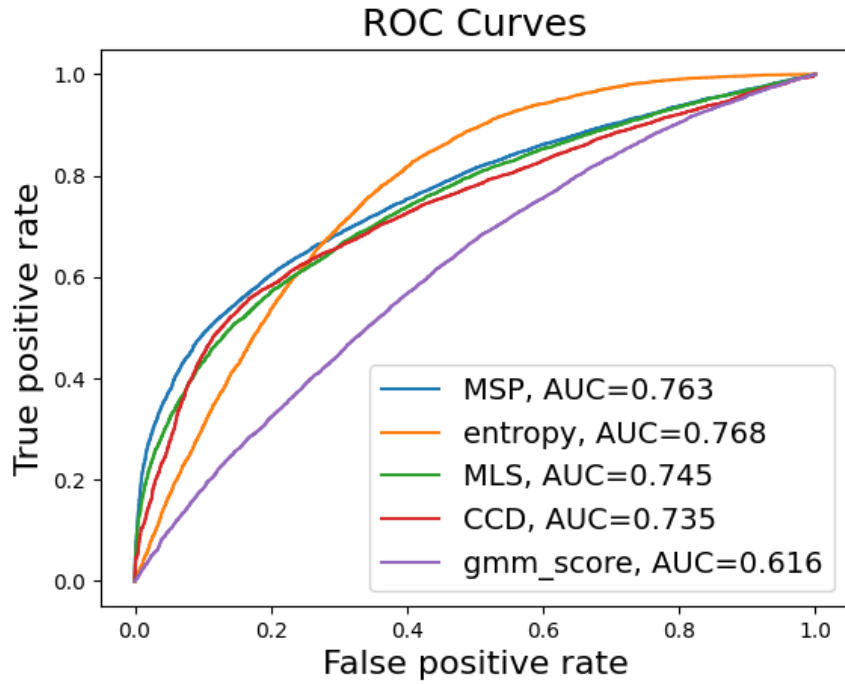


Figure 5.11: ROC curves for TinyImageNet with 180 known and 20 unknown classes.

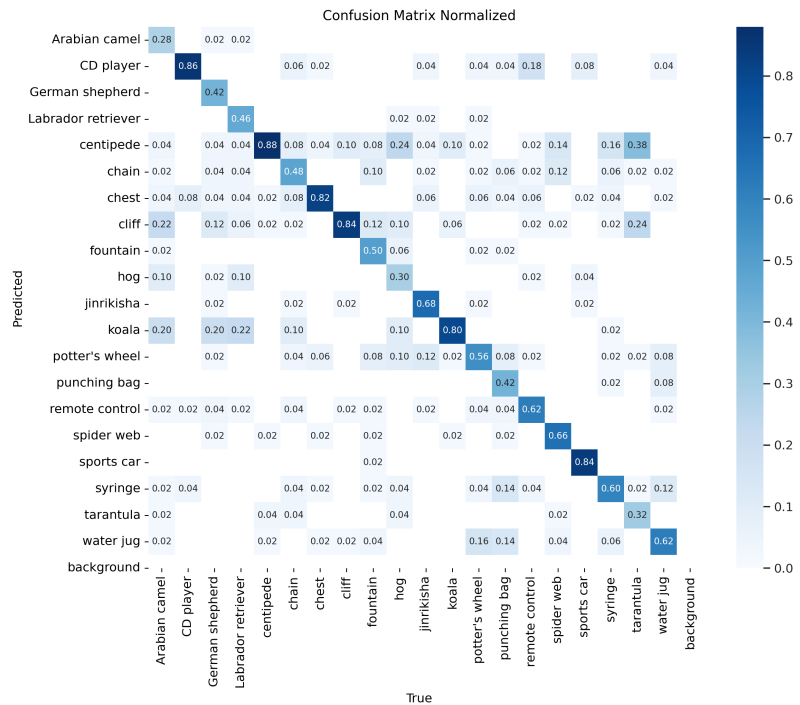


Figure 5.13: Confusion matrix for the model trained on 20 synthetic classes and evaluated on 20 real classes from Tiny ImageNet.

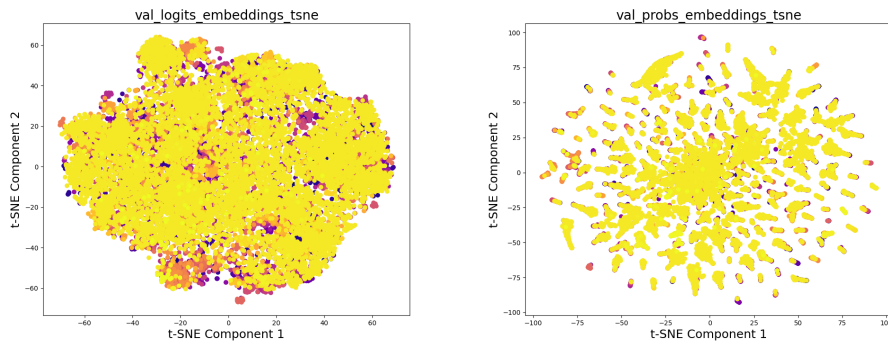


Figure 5.12: The left plot displays 2D T-SNE latent representations of logit embeddings, while the right plot shows 2D T-SNE latent representations of softmax embeddings, both retrieved from the Tiny ImageNet validation dataset. In both plots, yellow indicates unknown classes.

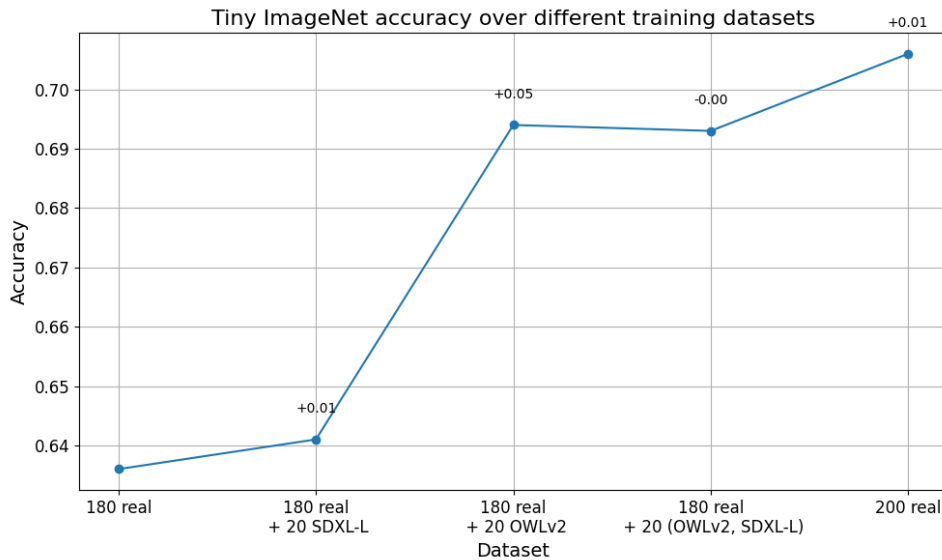


Figure 5.14: Accuracy on the original Tiny ImageNet dataset validation set.

5.4 Open set object detection

In the last experiment, we evaluate the performance of our proposed pipeline in the object detection setting. First, we train the YOLOv8n model on the entire PASCAL VOC dataset, then remove the 5 most underrepresented classes (bus, cow, dining table, sofa, and train) and train the model on the remaining 15 classes. The removed classes are treated as unknown. We then investigate the open-set performance using an entropy threshold of 0.2. The results of the open-set detection are shown in Figure 5.15. We observe that the model fails to detect half of the unknown objects, and many unknown objects with over-confident predictions are misclassified as known objects.

5. Experiments

This occurs because some unknown objects have a high degree of similarity to the known ones. For example, buses share many common visual features with cars, and some chairs closely resemble sofas. While it is challenging to avoid such errors, they do not pose a problem for the next step in the pipeline (pseudo-labeling) since the known objects predicted as unknown will also be pseudo-annotated.

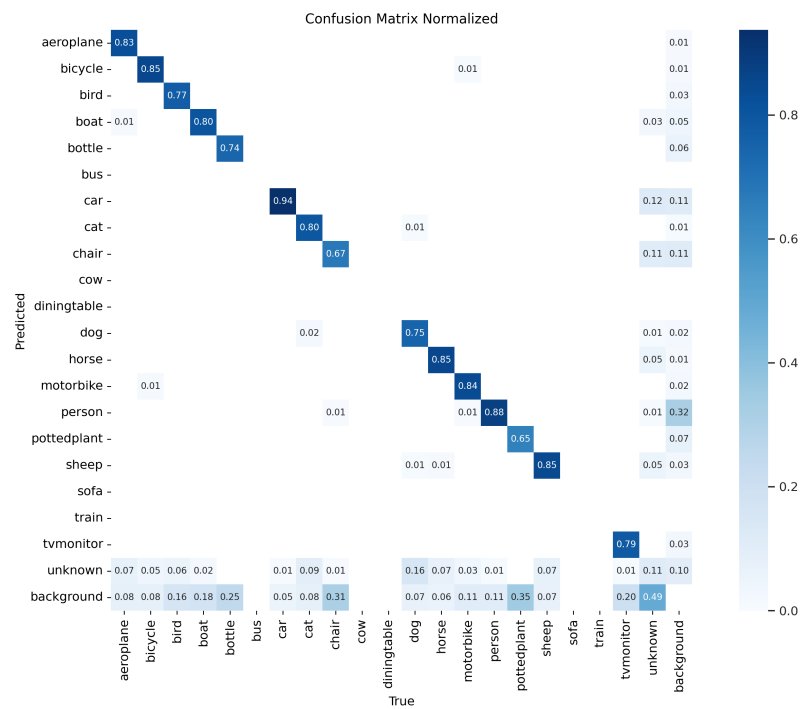


Figure 5.15: Open set performance on VOC PASCAL dataset with 15 known and 5 unknown classes. 80% of known objects are correctly detected.

5.4.1 Automatic label retrieval

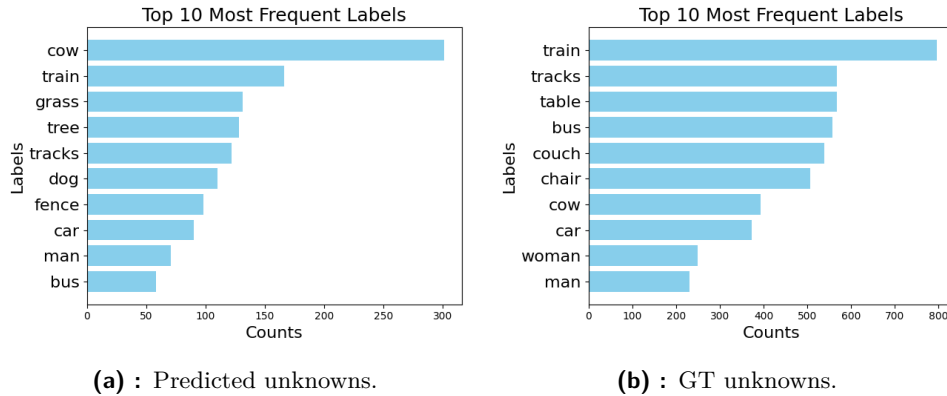


Figure 5.16: Retrieved labels. Deleted (gt) unknown labels: cow, train, sofa, bus, dinning table.

Labels can be automatically retrieved from the unlabeled image set using a multimodal model for visual question answering (VQA) purposes. We employ TinyLLaVA [43] to run inference on detected unknown images and ground truth unknown images. The model receives images along with the following prompt: "USER: <image> Name 3 objects in the image, separated by commas. ASSISTANT:". We then post-process the outputs to extract object names and count the occurrences of each label. Figure 5.16 displays the top 10 labels for both predicted and ground truth unknowns. We observe that labels retrieved from ground truth contain all 5 unknown classes, while those from predictions include only 3. This approach can be used without human intervention, allowing the top-k labels to be directly provided to annotation and generation models. Alternatively, humans can select relevant objects for specific applications.

5.4.2 Training on novel classes

After obtaining images with unknown labels, we can train the model to detect new classes. First, we generate a synthetic dataset with 5000 images using SDXL-Lightning. Next, we generate a dataset with the same properties but using SDXL, which requires more denoising steps for generation, to see if there is a significant gap in quality that affects performance on real data. From Figure 5.19, it is evident that a stronger model generates much better data, resulting in a +0.03 mAP-50 increase in absolute values. The visual quality comparison can be seen in Figure 5.17.

5. Experiments



Figure 5.17: Synthetic object detection dataset created using SDXL-Lightning (up) and SDXL (down).

Dataset generation command:

```

1  datadreamer --save_dir voc_5_classes_sdxl_5k \
2              --class_names "bus" "cow" "dining_table" "sofa" "
           train" \
3              --prompts_number 5000 \
4              --prompt_generator tiny \
5              --num_objects_range 1 1 \
6              --image_generator sdxl \
7              --use_tta \
8              --batch_size_prompt 256 \
9              --batch_size_image 4 \
10             --task detection \
11             --image_annotator owlv2 \
12             --annotator_size base

```

Then we reannotate the resulting dataset with all classes present in PASCAL VOC, as some previously known objects can occur in the background:

```

1  datadreamer --save_dir voc_5_classes_sdxl_5k \
2              --class_names "aeroplane" "bicycle" "bird" "boat" "
           bottle" "bus" "car" "cat" "chair" "cow" "dining_
           table" "dog" "horse" "motorbike" "person" "potted_
           plant" "sheep" "sofa" "train" "tv_monitor" \
3              --use_tta \
4              --task detection \
5              --image_annotator owlv2 \
6              --annotator_size base \
7              --annotate_only

```

We use five ground truth class names to pseudo-annotate real images using OWLv2. With just 1000 images labeled as unknown, we observe that real images are much more data-effective than synthetic ones, resulting in a +0.05 increase in mAP-50 compared to the best synthetic dataset with 5x more images (Figure 5.19).

Figure 5.18 shows the class distributions for different datasets. Interestingly, despite being explicitly stated to contain target objects, the synthetic dataset (Figure 5.18b) generates images with many other objects. For example, synthetic data contain more people and many more potted plants. This may occur for two reasons. First, we use a language model (TinyLlama) to generate prompts that should include the names of target objects, but it can also generate other names, such as "A photo of a person in the living room with a sofa and potted plant." Second, the image generation model (SDXL) might not strictly follow the given text prompt and may generate some unwanted objects in the background. From Figure 5.18c, we can also observe that our method managed to detect almost all unknown cows and many trains, but only a few instances of other classes (bus, sofa, dining table).

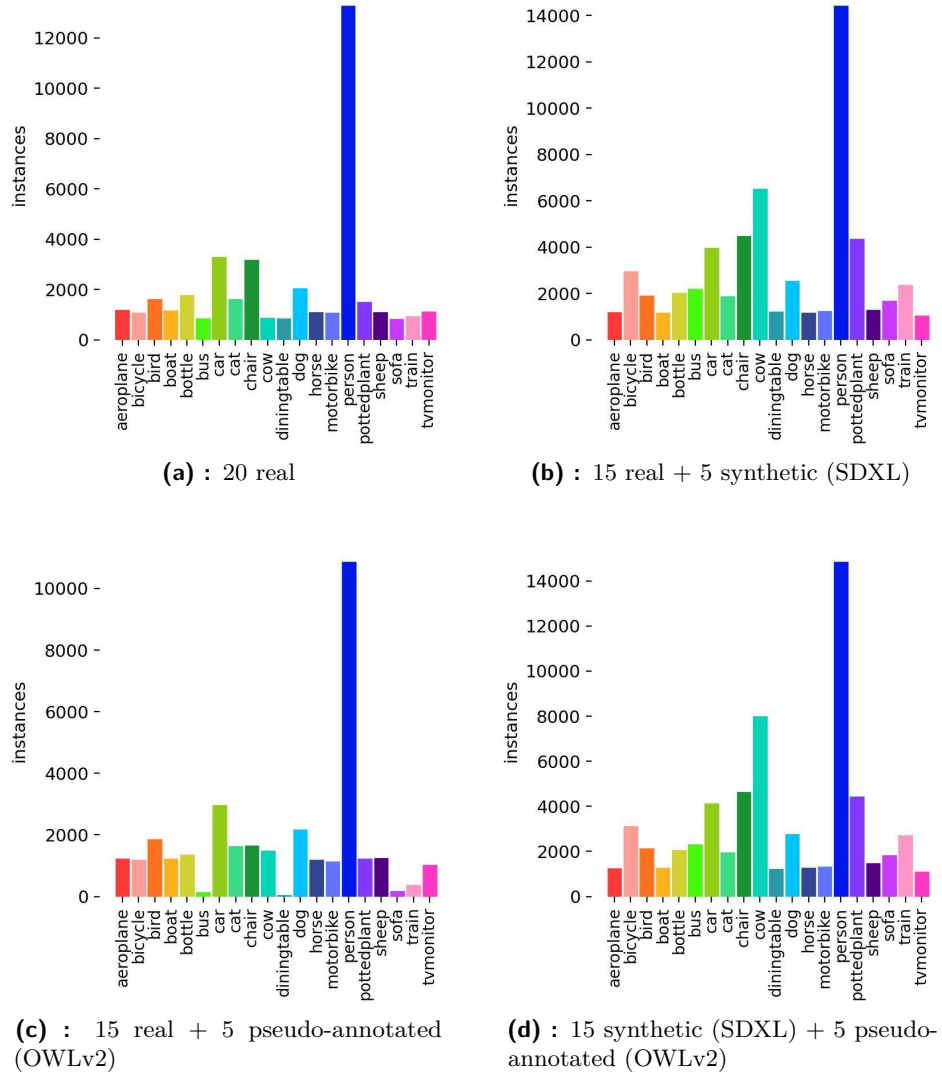


Figure 5.18: Label distributions.

Finally, we combine both pseudo-annotated and synthetic data, which results in a further performance increase. All results are shown in Figure 5.19. This demonstrates that, in this setting, synthetic data provide a significant boost. Furthermore, we train the model on the merged real + synthetic dataset generated using SDXL and achieve a new best mAP-50 of 0.827, which is better than training only on real data (0.816). This highlights that the use of synthetic data is highly effective for object detection. One possible reason it works here and not as well for classification is mosaic augmentation, which combines images from the batch into a 2x2 grid, thereby mixing synthetic and real images so the model does not overfit on the synthetic data distribution.

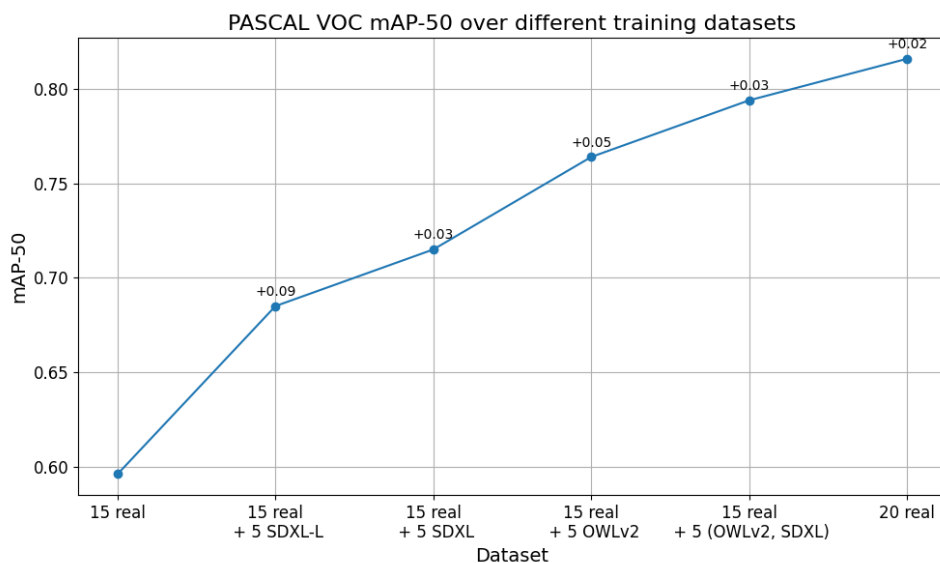


Figure 5.19: YOLOv8n mAP-50 on the validation split of VOC PASCAL for different training datasets. SDXL-L stands for SDXL-Lightning.

5.5 Results discussion

We evaluated multiple simple open-set detection approaches in different settings, multi-view and single-view, and for different architectures, Transformer-based and CNN-based. We demonstrated that using multiple views of the same objects is highly beneficial for both closed-set and open-set recognition. Our introduced method, called Matrix Entropy, achieved the best performance for multi-view classification. Almost all methods perform equally well for single-view classification with simple data, while entropy thresholding is the optimal choice for more complex data.

We showed that pseudo-annotation with large language-image models on detected unknown data can achieve almost the same quality as training on ground truth data for both classification and detection.

We demonstrated that synthetic data can be relatively effective in representing novel classes. Synthetic data was revealed to work better for object detection than for image classification. We highlight that for effective usage, the distribution shift should be handled.

Finally, we illustrated that combining pseudo-annotated and synthetic images is very beneficial and can achieve excellent performance. This approach seems to mitigate distribution shift because real instances of novel classes are present, and synthetic images can be considered as advanced data augmentation.

In summary, our findings are:

1. **Multi-view Benefits:** Using multiple views of the same object enhances both closed-set and open-set recognition performance. Matrix Entropy was the most effective method for multi-view classification.

2. **Open-Set Detection:** In simpler datasets, various methods performed well, but for complex datasets, entropy thresholding was the best approach.
3. **Pseudo-Annotation Effectiveness:** Leveraging large language-image models for pseudo-annotation of unknown classes can yield results comparable to ground truth training, benefiting both classification and detection tasks.
4. **Synthetic Data Usage:** While synthetic data is useful, especially for object detection, handling the distribution shift between synthetic and real data is crucial for maximizing effectiveness.
5. **Combining Data Types:** Integrating pseudo-annotated and synthetic images leads to significant performance improvements, addressing distribution shifts and enhancing overall model robustness.

These findings suggest a robust pipeline for improving open-set recognition and object detection tasks through the strategic use of multi-view data, pseudo-annotation with large language-vision models, and synthetic data generation with Stable Diffusion models.

5.6 Future work

5.6.1 Multiple Pipeline Iterations

We considered and experimented with only a single iteration of our pipeline: 1) unknown object detection, 2) dataset production for novel classes using foundational models, and 3) retraining with new classes. In theory, this could be iterated multiple times, allowing the edge model to continuously learn until it can detect/classify every object in the specific environment in which it operates.

5.6.2 Closing the Synth2Real Gap

We observed that using synthetic data with real data could be challenging due to the difference in data distribution. Therefore, some advanced techniques should be applied to mitigate this and minimize the Synth2Real gap.

5.6.3 Large Scale

To properly verify our proposed pipeline’s efficiency, the experiments should be run on a larger scale. By using a larger foundational model, it is possible to obtain even better quality data and better results. Additionally, our approach can be verified on larger open vocabulary datasets such as LVIS [11] and Object365 [37], which contain significantly more categories and images.



Chapter 6

Conclusion

In conclusion, we conducted an evaluation of various open-set classification strategies using different architectures, including Transformers (represented by Vision Transformer) and CNNs (represented by YOLOv8). Our novel method, Matrix Entropy, for multi-view open-set classification, outperformed existing techniques and is straightforward to implement. We assessed open-set classification in both multi-view and single-view scenarios, highlighting the clear benefits of aggregating multiple views of the same object.

For single-view classification, our experiments across several datasets confirmed that performance in closed-set scenarios significantly impacts open-set outcomes. We explored two approaches to address the knowledge gap and learn about unknown classes. The first approach used large language-image models (CLIP for classification and OWLv2 for detection). The second approach involved data generation using Stable Diffusion models. Combining pseudo-annotated real data with synthetic data yielded the highest performance.

Our work demonstrated that pseudo-annotation with large language-image models on detected unknown data can achieve nearly the same quality as training on ground truth data for both classification and detection tasks. Furthermore, we showed that synthetic data could effectively represent novel classes, though addressing distribution shift is necessary. Finally, our approach of combining pseudo-annotated and real images mitigates distribution shift and serves as advanced data augmentation, achieving excellent overall performance for object detection.

Appendix A

Bibliography

- [1] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [2] Victor Besnier, Himalaya Jain, Andrei Bursuc, Matthieu Cord, and Patrick Pérez. This dataset does not exist: training models from generated images. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2020.
- [3] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3):8, 2023.
- [4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.
- [5] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. *arXiv preprint arXiv:2401.17270*, 2024.
- [6] Akshay Raj Dhamija, Manuel Günther, and Terrance Boult. Reducing network agnostophobia. *Advances in Neural Information Processing Systems*, 31, 2018.
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and

- Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.
- [9] Silvio Galesso, Max Argus, and Thomas Brox. Far away in the deep space: Nearest-neighbor-based dense out-of-distribution detection. *arXiv preprint arXiv:2211.06660*, 2022.
- [10] Chuanxing Geng, Sheng-Jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2020.
- [11] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.
- [12] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022.
- [13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [14] Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. *arXiv preprint arXiv:2106.05258*, 2021.
- [15] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.
- [16] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.
- [17] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, January 2023.
- [18] K J Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection, 2021.
- [19] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *2011 IEEE international conference on robotics and automation*, pages 1817–1824. IEEE, 2011.

- [20] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022.
- [21] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521, 2023.
- [22] Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.
- [23] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.
- [24] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3243–3249, 2018.
- [25] Dimity Miller, Niko Sunderhauf, Michael Milford, and Feras Dayoub. Class anchor clustering: A loss for distance-based open set recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3570–3578, 2021.
- [26] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection, 2023.
- [27] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers, 2022.
- [28] Sokovnin Nikita. Rozpoznávání neznámých objekt pro detekci 3d objektu ve světě bez omezení. B.S. thesis, České vysoké učení technické v Praze. Vypočetní a informační centrum., 2021.
- [29] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.
- [30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

- [31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [32] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*, pages 102–118. Springer, 2016.
- [33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- [35] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.
- [36] Marco Seeland and Patrick Mäder. Multi-view classification with convolutional neural networks. *Plos one*, 16(1):e0245230, 2021.
- [37] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019.
- [38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [39] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. *arXiv preprint arXiv:2110.06207*, 2021.
- [40] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022.
- [41] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

- [42] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10145–10155, 2021.
- [43] Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. Tinyllava: A framework of small-scale large multimodal models. *arXiv preprint arXiv:2402.14289*, 2024.



Appendix B

AI Tools Used

In the course of this diploma thesis, the following AI tools were utilized:

- **GitHub Copilot**¹ - Used for code suggestions and completions.
- **ChatGPT**²: Offered suggestions for writing style and rephrasing.
- **Writefull**³: Used for improving academic writing, checking grammar, and suggesting rephrasing.
- **Grammarly**⁴: Assisted with grammar checking, punctuation, and style suggestions to enhance the quality of the text.

¹<https://github.com/features/copilot>

²<https://chat.openai.com>

³<https://writefull.com>

⁴<https://www.grammarly.com>