VNU-HCM University of Science

FACULTY OF INFORMATION TECHNOLOGY

Introduction to Machine Learning

Project: DINOv3

Lecturer:
Bùi Duy Đăng
Huỳnh Lâm Hải Đăng
Trần Trung Kiên

Full name:	*MSSV*	*class ID*
Đặng Vĩnh Tường	**21127720**	**23KHMT2**
Phan Hải Minh	**22127273**	**23KHMT2**
Thái Minh Khoa	**23127394**	**23KHMT2**

Contents

1 Introduction

1.1 Overview of the Project

In recent years, computer vision has become a core research area in artificial intelligence, with applications spanning image classification, object detection, semantic segmentation, and visual understanding. Traditional supervised learning approaches have achieved strong performance but rely heavily on large-scale labeled datasets, which are expensive and time-consuming to collect.

This project focuses on exploring DINOv3, a recent self-supervised vision model, as a general-purpose visual backbone. The objective of the project is to analyze the representation capability of DINOv3 and evaluate its effectiveness on downstream tasks such as image classification using linear probing and object detection. Through experimental analysis and visualization, this project aims to demonstrate how self-supervised visual representations can generalize across different computer vision tasks.

1.2 Overview of DINO and DINOv2

DINO (Distillation with No Labels) is a self-supervised learning framework designed to learn meaningful visual representations without requiring labeled data. It employs a teacher–student paradigm, where the student network is trained to match the output distribution of a teacher network updated using exponential moving average (EMA). This approach enables the model to learn semantic information directly from raw images.

Building upon DINO, DINOv2 introduces improvements in training stability, data scale, and architectural design. By leveraging larger datasets and refined training strategies, DINOv2 significantly enhances the quality and robustness of learned visual features. These representations have been shown to transfer well to various downstream tasks without task-specific fine-tuning.

1.3 Introduction to DINOv3

DINOv3 further advances self-supervised visual representation learning by introducing a more scalable training framework and richer feature representations. Unlike earlier versions, DINOv3 emphasizes dense and structured feature learning, enabling both global and local semantic understanding.

One of the key contributions of DINOv3 is its ability to serve as a vision foundation model, providing high-quality features that can be reused across multiple vision tasks with minimal additional training. This makes DINOv3 particularly suitable for tasks such as linear probing classification, object detection, and dense prediction tasks.

2 Problem Statement

2.1 Computer Vision Tasks Addressed by DINOv3

DINOv3 is designed as a general-purpose vision foundation model capable of supporting a wide range of computer vision tasks through transferable self-supervised representations. By learning rich global and local visual features, DINOv3 can be effectively applied to both high-level and dense prediction tasks without requiring extensive task-specific fine-tuning.

Image Classification is one of the most fundamental vision tasks supported by DINOv3. The model provides discriminative global representations that enable effective classification through linear probing or lightweight classifiers, demonstrating strong semantic understanding of image content.

Object Detection is another important task where DINOv3 can be employed as a backbone to identify and localize objects within an image. The dense patch-level features produced by DINOv3 preserve spatial information, making them suitable for integration with detection frameworks such as DETR and other transformer-based detectors.

Semantic Segmentation and Instance Segmentation benefit from DINOv3's dense feature representations, which allow pixel-level or region-level understanding of images. These features support precise boundary localization and class differentiation across complex scenes.

Image Retrieval and Visual Similarity Learning rely on meaningful feature embeddings to compare visual content. DINOv3 produces high-quality representations that enable effective retrieval, clustering, and similarity search without requiring labeled training data.

Few-shot and Transfer Learning tasks can leverage DINOv3 by adapting pre-trained representations to new datasets with limited labeled samples. The strong generalization capability of self-supervised features allows DINOv3 to perform well even in low-data regimes.

Self-supervised Feature Extraction and Representation Learning itself is a core application of DINOv3. The learned embeddings can be reused across multiple downstream tasks, serving as a foundation for further research and development.

Dense Prediction and Visual Reasoning Tasks, such as keypoint detection and part-based understanding, can also benefit from the spatially-aware transformer features generated by DINOv3, which encode relationships between different regions of the image.

Overall, DINOv3 supports a broad spectrum of computer vision tasks by providing robust, scalable, and transferable visual representations, positioning it as a versatile backbone for modern vision applications.

2.2 Self-Supervised Learning in Computer Vision

Self-supervised learning has emerged as a promising alternative to supervised learning by eliminating the dependence on labeled data. In computer vision, self-supervised methods learn representations by defining pretext tasks or consistency objectives directly from image transformations.

DINO-style self-supervised learning leverages representation alignment between different views of the same image, allowing the model to discover semantic patterns naturally. This paradigm enables scalable training and improves generalization across tasks and datasets.

2.3 Vision Foundation Models

Vision Foundation Models are large-scale models trained on massive datasets to learn general and transferable visual representations. Instead of being optimized for a single task, these models act as reusable backbones that can be adapted to multiple downstream applications with minimal effort.

DINOv3 fits naturally into this category by providing strong visual embeddings that perform well across classification, detection, and dense prediction tasks, making it a versatile foundation for computer vision systems.

2.4 Dense Transformer Features

A key characteristic of DINOv3 is its ability to produce dense transformer features, where meaningful representations are extracted not only from global image tokens but also from local patch tokens. These dense features preserve spatial and relational information across the image, which is crucial for tasks requiring precise localization and structural understanding.

By learning both global context and local relationships, dense transformer features enable DINOv3 to support a wide range of vision tasks, particularly those involving object localization and fine-grained visual reasoning.

3 DINOv3 Architecture & Mechanism

3.1 Overall Architecture

DINOv3 is designed as a large-scale self-supervised vision foundation model with the goal of learning general and transferable visual representations from unlabeled data. Its architecture builds upon a Vision Transformer (ViT) backbone and adopts a teacher–student distillation framework to enable stable and effective self-supervised training. By explicitly modeling both global image-level semantics and dense patch-level representations, DINOv3 provides a unified architectural foundation that supports a wide range of downstream vision tasks, including classification, detection, segmentation, and video understanding.

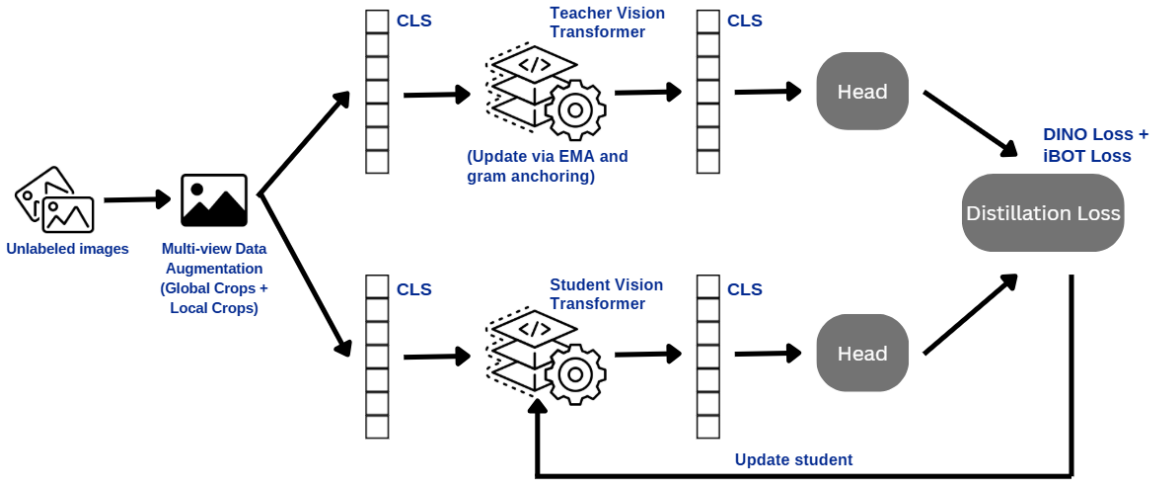

Figure 1: DINOv3 architecture

Key Components of the DINOv3 Architecture:

- Multi-view Data Augmentation: Each input image is transformed into multiple global and local views, which serve as the source of self-supervised signals by enforcing consistency across different perspectives of the same image.
- Vision Transformer (ViT) Backbone: DINOv3 employs a Vision Transformer to encode images into a global representation (CLS token) and dense patch-level representations, enabling both semantic understanding and spatial reasoning.
- Teacher–Student Framework: The architecture consists of two identical networks. The student is trained via backpropagation, while the teacher is updated using an exponential moving average (EMA) of the student parameters to provide stable pseudo-targets.

- Projection Heads: Lightweight projection heads are attached during training to map features into the space where self-supervised objectives are applied. These heads are discarded after pretraining.
- Self-Supervised Losses (DINO + iBOT): DINO loss supervises global image representations, while iBOT loss operates at the patch level to learn dense features and preserve spatial structure.
- Gram Anchoring: A regularization mechanism that stabilizes dense patch representations by constraining the relationships between patch features, preventing degradation during long training.
- Frozen Backbone for Downstream Tasks: After pretraining, the DINOv3 backbone can remain frozen and be reused across multiple task-specific heads, simplifying downstream training and improving efficiency.

3.2 Data Preparation

DINOv3 is trained on large-scale unlabeled image collections, making data preparation a critical component of the overall training pipeline. Since no human annotations are available, the quality and diversity of the training data directly influence the generalization ability of the learned representations.

To address this, DINOv3 adopts a careful data curation strategy to reduce redundancy and balance the distribution of visual content. Images are selected to maximize diversity across different visual concepts and scenes, preventing over-representation of common patterns that could bias the learned features.

During training, each image is transformed into multiple augmented views through a multi-crop data augmentation strategy. This includes both global and local crops, combined with color jittering, blurring, and random flipping. These augmented views serve as the foundation for self-supervised learning by enforcing representation consistency across different perspectives of the same image.

Figure 2: DINOv3 multi-view data augmentation

All images are resized to a fixed resolution and partitioned into patches before being processed by the Vision Transformer backbone. The resulting patch-level tokens

enable DINOv3 to learn dense representations that are crucial for spatially sensitive downstream tasks such as object detection and semantic segmentation.

3.3 Self-Supervised Learning Mechanism

DINOv3 adopts a self-supervised learning paradigm to learn visual representations from large-scale unlabeled image data. Instead of relying on human-annotated labels, the model derives supervisory signals directly from the data by enforcing consistency between different augmented views of the same image. This design enables DINOv3 to scale effectively with data size while avoiding the cost and limitations associated with manual annotation.

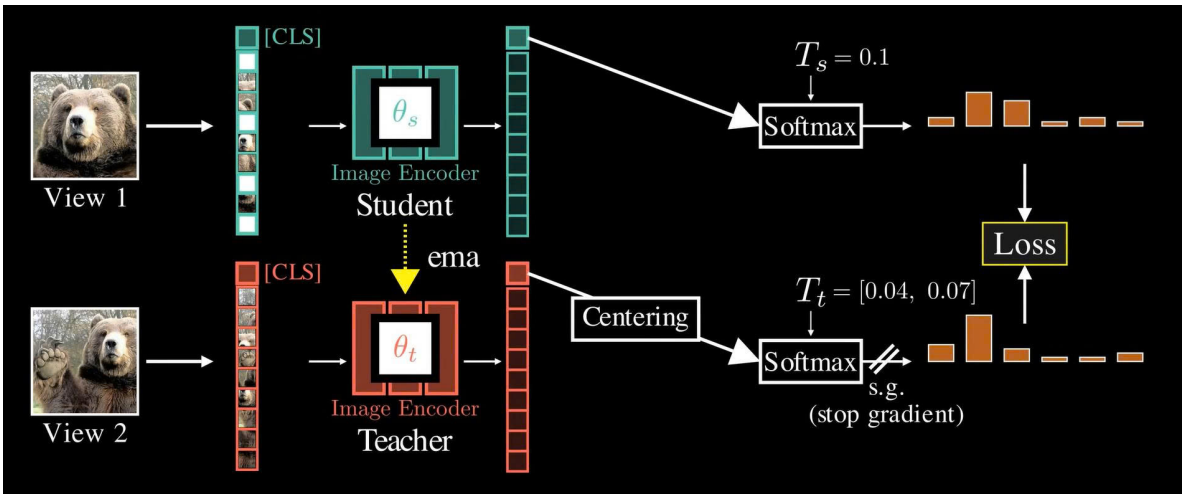

Figure 3: DINOv3 self-supervised learning mechanism

The self-supervised framework of DINOv3 is built upon a teacher-student architecture, where both networks share the same Vision Transformer backbone but follow different update mechanisms. The student network is trained via backpropagation, while the teacher network provides stable pseudo-targets and is updated as an exponential moving average of the student parameters. This asymmetric design allows the model to learn meaningful representations without explicit supervision and prevents representation collapse during training.

By leveraging multi-view data augmentation and distillation-based self-supervision, DINOv3 is able to capture both semantic and structural information from images. This general self-supervised training strategy forms the foundation upon which more specific learning objectives are applied, as detailed in the following sections.

3.4 Loss Function & Mathematical Formulation

1. **DINO Loss**
   The DINO loss in DINOv3 is an image-level objective that encourages a Vision Transformer (ViT) to learn a consistent global representation across different views of the same image. The method feeds multiple crops of an image to a student and a teacher ViT: the teacher sees only large *global* crops, while the student sees both global and smaller *local* crops. For each view, the learnable `[CLS]` token embedding is passed through a DINO head to produce a probability distribution, and the loss enforces the student's distribution to match the teacher's across all non-identical crop pairs. The teacher is not trained independently but is an exponential moving average (EMA) of the student, providing a slowly evolving target that stabilizes training and prevents collapse. By matching outputs across diverse crops, the DINO loss forces the model to capture the underlying semantic content of the image rather than memorizing specific views, yielding strong global representations in DINOv3.

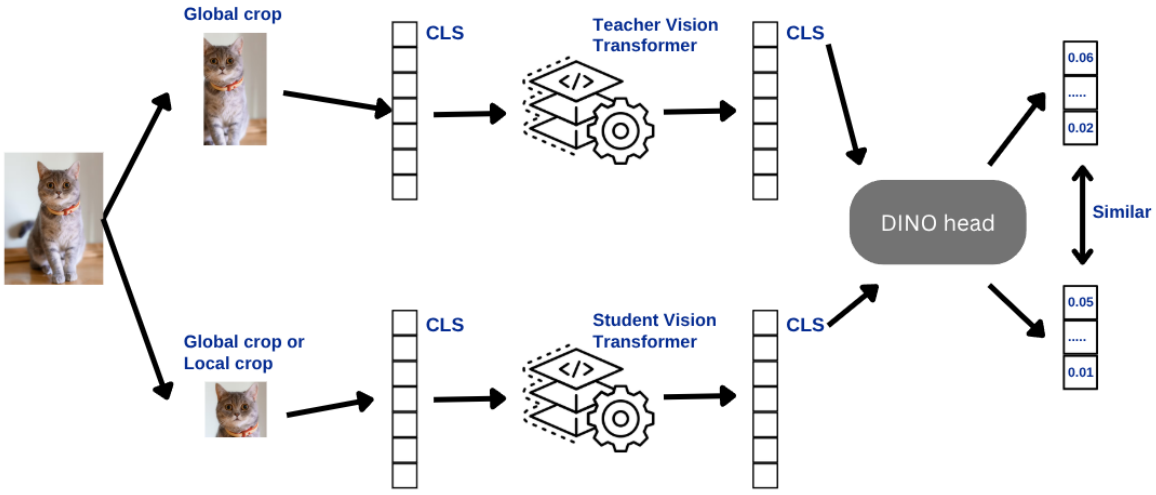

Figure 4: DINO Loss illustration

2. **iBOT Loss**
   The iBOT loss complements DINO by enforcing patch-level consistency, enabling the model to capture fine-grained local details that are crucial for dense prediction tasks such as segmentation. Using the same multi-crop strategy, the teacher processes only global crops while the student processes both global and local crops. Unlike DINO, which operates on class-level outputs, iBOT works on patch tokens. A subset of the student's output patch tokens is randomly masked, and an iBOT head is trained to predict the corresponding teacher output tokens at the same image locations. The loss is applied only to patches where the student and teacher crops spatially overlap, ensuring meaningful comparisons. Since the objective reconstructs latent teacher features rather than raw pixels, iBOT en-

courages the student to learn semantically rich local representations instead of low-level pixel details, resulting in strong dense features in models such as DINOv3.

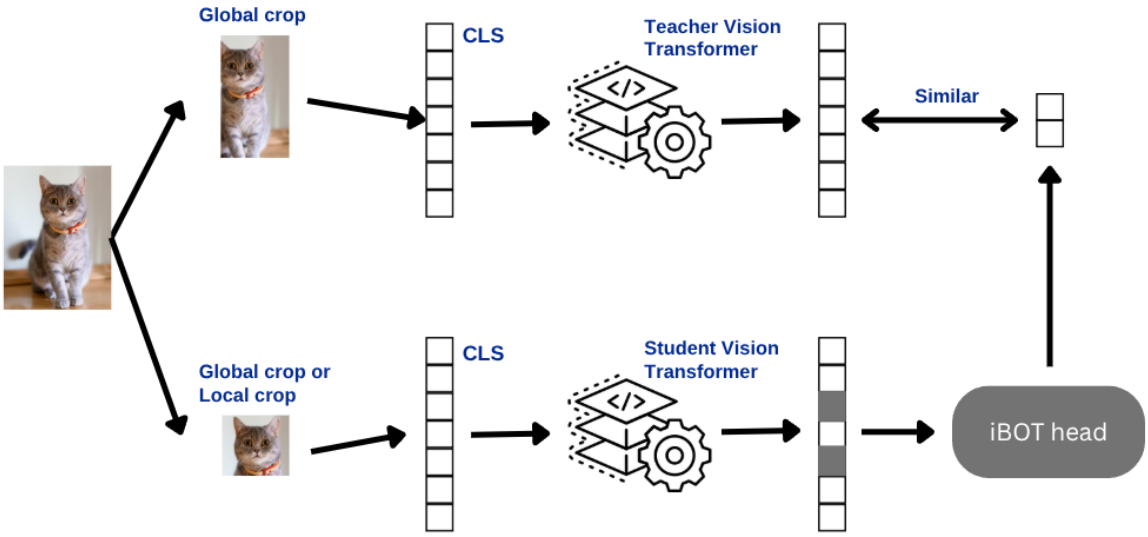

Figure 5: iBot Loss illustration

3.5 Gram Anchoring

Problem. When DINOv2 is scaled to very large models and trained for a long time, a clear imbalance appears between global and local learning. Image-level tasks like classification keep improving, showing that the model's global understanding is getting stronger. However, dense tasks such as segmentation start to degrade. Visualizations of patch similarities reveal the issue: patch-level features slowly lose their structure, and unrelated patches begin to look similar. This loss of local consistency explains why dense prediction performance drops and becomes unstable, even though global performance continues to improve.

Solution. Gram Anchoring is introduced to keep patch-level representations stable during long training runs. Instead of forcing individual patch features to match a teacher exactly, it focuses on preserving the *relationships* between patches. This is done using a Gram matrix: the matrix of all pairwise dot products of patch features in an image. The student model is encouraged to match the Gram matrix of a *Gram teacher*, which is an earlier checkpoint of the teacher network known to have strong dense-task performance. By anchoring the similarity structure rather than the features themselves, the model is free to keep learning globally while avoiding patch-level collapse. In practice, this stabilizes dense features, restores segmentation performance, and does not harm classification accuracy. Let us denote by X_S (respectively X_G) the $P \times d$ matrix of ℓ_2-normalized local features produced by the student (respectively the Gram teacher). The loss $\mathscr{L}_{\text{Gram}}$ as follows:

$$\mathscr{L}_{\text{Gram}} = \left\| X_S X_S^T - X_G X_G^T \right\|_F^2$$

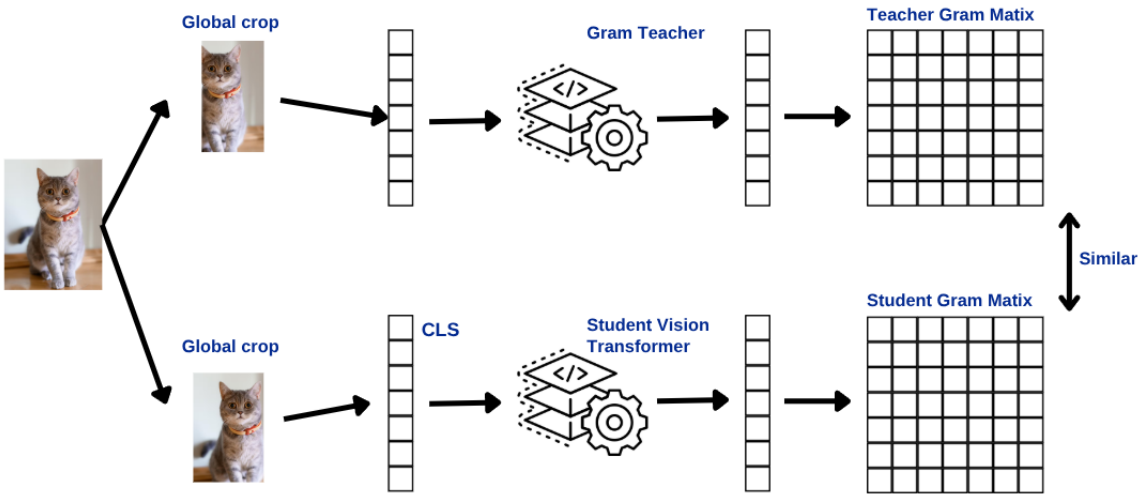

Figure 6: Gram Anchoring illustration

4 Applications of DINOv3

4.1 Object Detection

Task Description:

In this implementation, we build an object detection pipeline based on the DETR framework combined with a pre-trained DINOv3 visual backbone. The backbone is kept frozen and used as a feature extractor, while a transformer encoder–decoder and learnable object queries are trained to predict object classes and bounding boxes in an end-to-end manner. The model is trained and evaluated on a COCO-style dataset, and performance is reported using mean Average Precision (mAP) at IoU thresholds from 0.5 to 0.95 for demonstration purposes.

Implementation:

We implement an end-to-end object detection pipeline based on the DETR framework with a pre-trained DINOv3 model as a frozen visual backbone. The backbone is used to extract feature representations from input images, which are then projected to a fixed hidden dimension and processed by a Transformer encoder–decoder with learnable object queries to predict object classes and bounding boxes. The model is trained on a COCO-style dataset, where bounding boxes are converted to the DETR format and class labels are mapped to contiguous indices with a dedicated background class. Hungarian matching is applied during training using a combination of classification loss and bounding box regression loss, and optimization is performed with AdamW

while keeping the backbone frozen. This implementation focuses on demonstrating the feasibility of integrating a self-supervised visual backbone with a transformer-based detection head under limited training conditions.

Experimental Protocol:

Table 1: Hyperparameter settings for the object detection experiment

Hyperparameter	**Value**
Backbone model	DINOv3 (pre-trained, frozen)
Detection framework	DETR
Hidden dimension	256
Number of object queries	100
Number of encoder layers	6
Number of decoder layers	6
Optimizer	AdamW
Learning rate	5×10^{-5}
Weight decay	0.05
Batch size	64
Number of training epochs	15
Bounding box loss	L1 loss
Classification loss	Cross-entropy loss
Matching algorithm	Hungarian matching
Evaluation metric	mAP@[0.5:0.05:0.95]

Object Detection Evaluation Metrics

Total Loss

Total loss represents the overall training objective of the object detection model. It is computed as a combination of multiple loss components and reflects how well the model jointly learns object classification and localization during training.

Classification Loss

Classification loss measures the error between the predicted class probabilities and the ground-truth object labels. This metric evaluates the model's ability to correctly recognize and distinguish different object categories.

Bounding Box Loss

Bounding box loss quantifies the discrepancy between the predicted bounding box

coordinates and the ground-truth bounding boxes. It reflects the accuracy of object localization and spatial alignment.

mAP@[0.5:0.95]
Mean Average Precision (mAP) averaged over multiple Intersection over Union (IoU) thresholds from 0.5 to 0.95 is the primary evaluation metric for object detection. It provides a comprehensive measure of detection performance by jointly considering localization accuracy and classification confidence across varying overlap criteria.

AP50
AP50 represents the average precision computed at an IoU threshold of 0.5, which evaluates detection performance under a relatively lenient overlap requirement.

AP75
AP75 measures average precision at a stricter IoU threshold of 0.75, emphasizing more precise object localization.

Hyperparameter Benchmark:

To analyze the training behavior of the proposed object detection model under different configurations, we conduct a small benchmark using three hyperparameter settings. The comparison focuses on training stability and loss convergence rather than absolute detection performance.

Standard configuration:

Metric	**Value**
Total loss	2.23
Classification loss	1.20
Bounding box loss	0.21
mAP@[0.5:0.95]	0.00
AP50	0.00
AP75	–

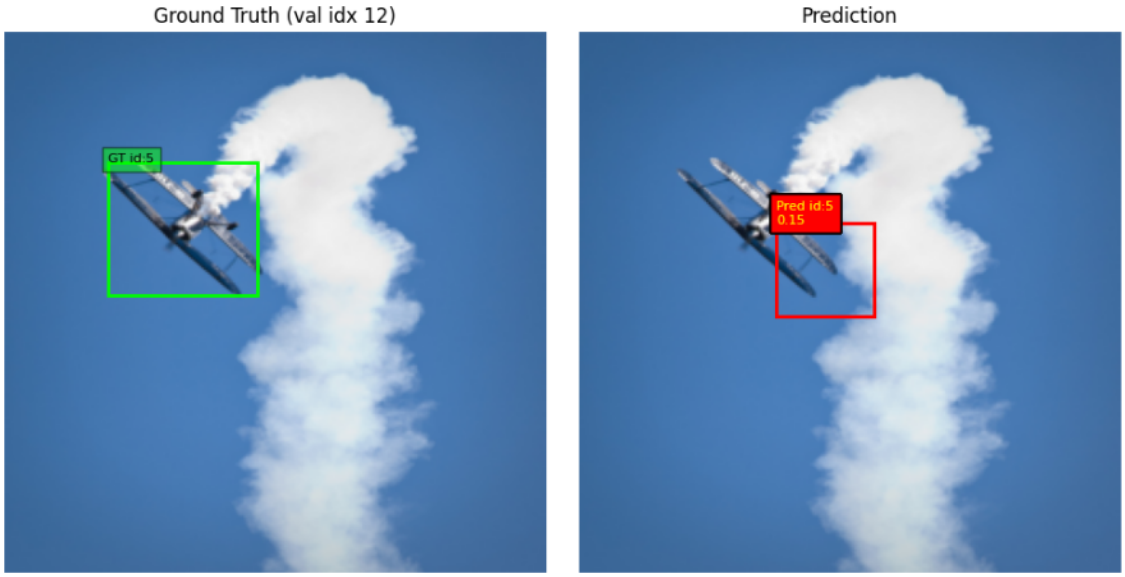

Figure 7: object detection visualization

Extended training configuration (Epoch = 5):

Metric	**Value**
Total loss	3.28
Classification loss	2.00
Bounding box loss	0.26
mAP@[0.5:0.95]	0.00
AP50	0.00
AP75	–

Figure 8: object detection visualization epoch = 5

Extended training configuration (Epoch = 50):

Metric	**Value**
Total loss	1.3581
Classification loss	0.6730
Bounding box loss	0.1370
mAP@[0.5:0.95]	0.0004
AP50	0.0011
AP75	–

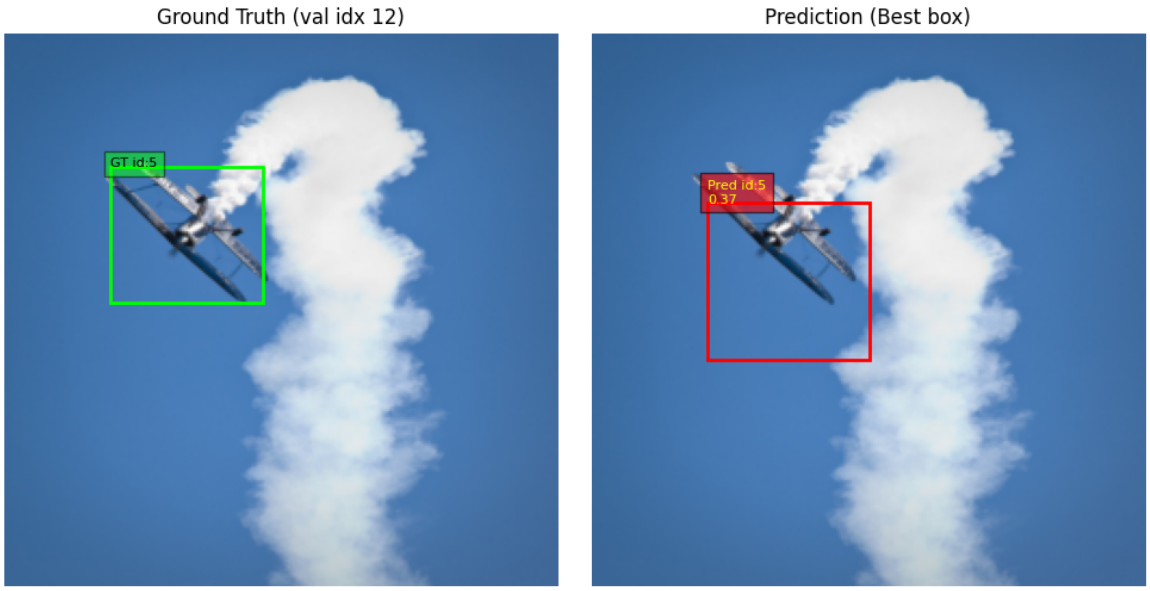

Figure 9: object detection visualization epoch = 50

Alternative optimization configuration (lr = 10^{-4}, weight decay = 10^{-4}):

Metric	**Value**
Total loss	2.38
Classification loss	1.30
Bounding box loss	0.22
mAP@[0.5:0.95]	0.00
AP50	0.00
AP75	–

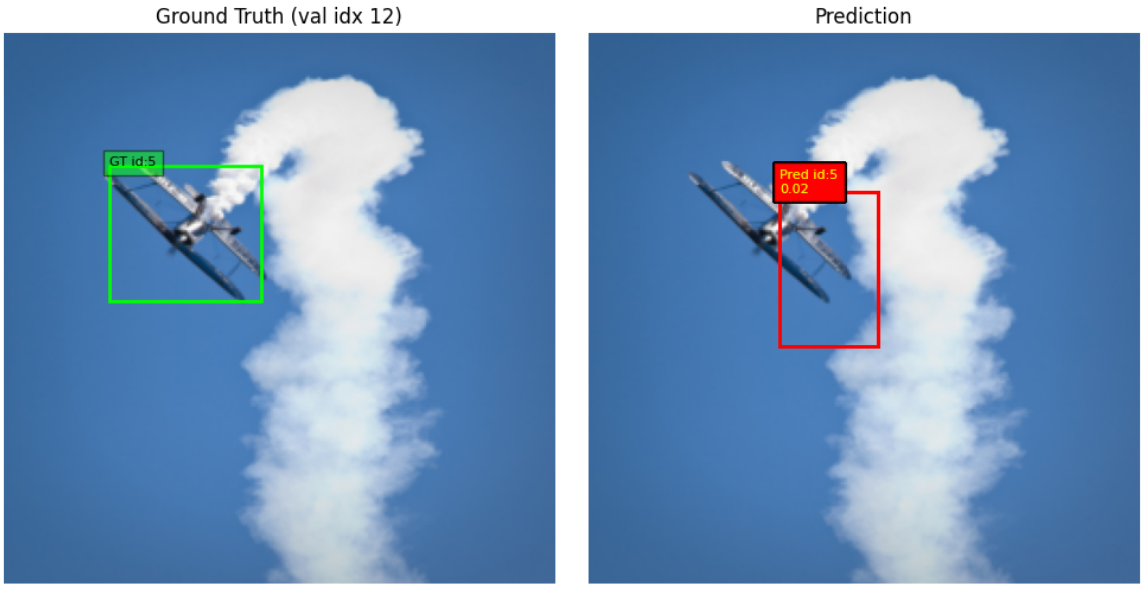

Figure 10: object detection visualization lr=1e-4, weight decay=1e-4

Evaluation:

Overall, the experimental results indicate that the standard configuration produces the most favorable predictions compared to both the extended training setup with five epochs and the alternative optimization configuration using a higher learning rate and lower weight decay.

The visualization results indicate that the object detection performance of the Standard configuration and the Alternative optimization configuration is noticeably better than that of the Extended training configuration (Epoch = 5). The predicted bounding boxes are more closely aligned with the aircraft objects, and the best qualitative detection performance is observed when the model is trained for 50 epochs.

Nevertheless, the quantitative detection metrics remain low across all configurations, with mAP@[0.5:0.95] and AP50 equal to zero and AP75 being undefined. This outcome is primarily attributed to the limited number of training epochs, the use of a frozen backbone, and the demonstration-oriented experimental setup, which is not intended to fully optimize the detection performance of the DETR-based model. When the number of training epochs is increased to 50, a slight improvement in detection performance can be observed, with mAP@[0.5:0.95] increasing to 0.0004 and AP50 to 0.0011, indicating that longer training is necessary for the model to begin learning meaningful object localization patterns.

4.2 Image Classification - Linear Probing

Task Description:

This task focuses on evaluating the image classification capability of a pre-trained DINOv3 visual backbone using a linear probing approach. The backbone is kept frozen and used to extract image-level feature representations, which are then fed into a logistic regression classifier trained on labeled data. The classification performance is evaluated on a validation set using Top-1 and Top-5 accuracy, along with additional metrics such as precision, recall, and F1-score, to analyze the effectiveness of the learned representations under limited training conditions.

Implementation:

The proposed system is implemented in Python using PyTorch and the HuggingFace Transformers library. A pre-trained DINOv3 model is employed as a frozen visual backbone to extract high-level feature representations from input images. The ImageNet100 dataset is organized in an ImageFolder-style directory structure, and class labels are mapped from a predefined JSON file to ensure consistent label indexing. Input images are preprocessed using the official AutoImageProcessor associated with the DINOv3 backbone. During feature extraction, images are forwarded through the backbone in evaluation mode, and the CLS token embeddings from the final hidden layer are used as global image representations. These features are then used to train a multinomial logistic regression classifier in a linear probing setting, where only the classifier parameters are optimized while the backbone remains fixed. Model performance is evaluated on a held-out validation set using top-1 accuracy, along with confusion matrix and per-class accuracy analysis to identify misclassified categories. This implementation enables an efficient and reliable assessment of the representation quality learned by DINOv3 on the ImageNet100 classification task.

Experimental Protocol:

Table 2: Hyperparameter settings for the DINOv3 linear probing experiment

Hyperparameter	**Value**
Backbone model	DINOv3 ViT-S/16 (pre-trained, frozen)
Pre-training method	Self-supervised learning
Input image size	224×224
Feature representation	CLS token embedding
Feature dimension	384
Classifier type	Multinomial Logistic Regression
Training paradigm	Linear probing
Optimizer (classifier)	LBFGS (scikit-learn default)
Regularization type	L2
Regularization strength (C)	1.0
Maximum iterations	1000
Batch size (feature extraction)	64
Gradient computation	Disabled for backbone
Training dataset	ImageNet100
Training split	train.X1 – train.X4
Validation split	val.X
Evaluation metric	Top-1 accuracy
Additional analysis	Confusion matrix, per-class accuracy

Classification - Linear Probing Evaluation Metrics

Top-1 Accuracy: Measures the ability of the model to correctly predict the most likely class for each input sample.

Top-5 Accuracy: Evaluates whether the ground-truth label appears among the five most confident predictions, providing insight into near-correct classifications.

Balanced Accuracy: Computes the average recall across all classes, ensuring that evaluation is not biased toward classes with larger sample sizes.

Precision: Indicates how many of the predicted samples for a given class are correctly classified.

Recall: Measures the proportion of true samples of a class that are correctly identified by the model.

F1-score: Represents the harmonic mean of precision and recall, balancing the trade-off between these two metrics.

Macro Average: Computes metrics independently for each class and then averages them, treating all classes equally regardless of their size.

Weighted Average: Computes metrics by weighting each class according to its number of samples, reflecting the overall dataset distribution.

Hyperparameter Benchmark:

We evaluate the impact of training-related hyperparameters under a linear probing setting using two configurations. The standard configuration follows the default optimization setup for the logistic regression classifier, while the extended training configuration restricts the maximum number of iterations to 100 to assess training efficiency.
Standard configuration:

Table 3: Overall classification performance on the validation set

Metric	**Value**
Top-1 Accuracy	85.64%
Top-5 Accuracy	98.00%
Balanced Accuracy	85.64%
Number of samples	5000

Table 4: Classification report on the validation set

Averaging	**Precision**	**Recall**	**F1-score**	**Support**
Macro average	0.8587	0.8564	0.8561	5000
Weighted average	0.8587	0.8564	0.8561	5000
Overall accuracy	–	–	0.8564	5000

Table 5: Worst 10 classes by classification accuracy

Class ID	Accuracy (%)	Support
64	36.00	50
78	40.00	50
9	48.00	50
48	52.00	50
88	60.00	50
50	60.00	50
43	62.00	50
34	64.00	50
42	68.00	50
62	68.00	50

	path	true	pred
4970	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4971	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4972	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4973	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4974	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4975	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4976	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	5
4977	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	87
4978	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4979	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4980	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4981	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	55
4982	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75

Figure 11: classification predict result

Overall, the results obtained with the standard configuration for the Image Classifica-

tion linear probing task are quite strong, with misclassifications occurring in only a small number of samples. A more detailed analysis of the results is provided through the following visualizations.

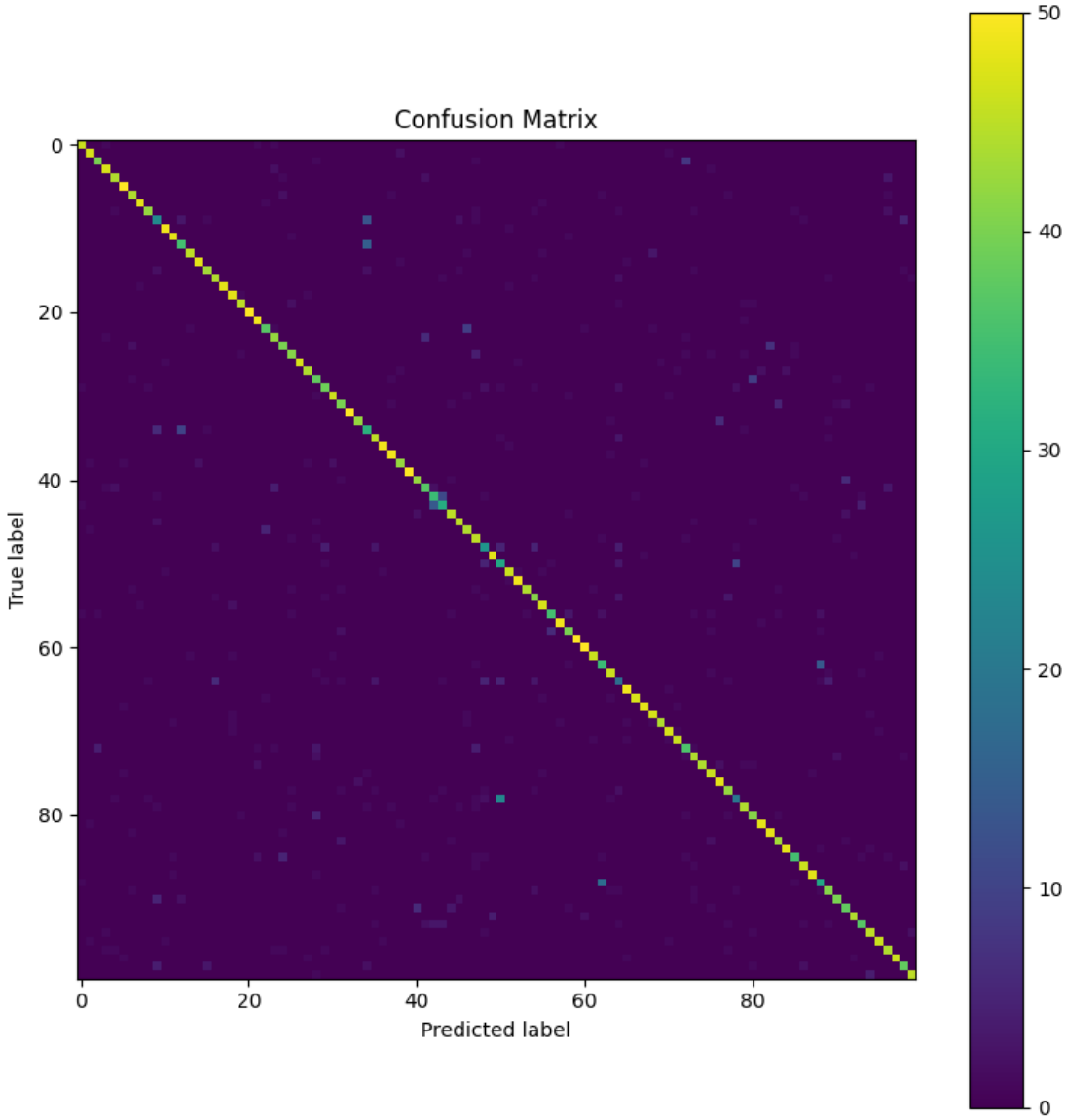

Figure 12: classification diagram result

The results shown in the figure indicate that most labeled samples are correctly predicted, as reflected by the prominent yellow diagonal in the chart.

Extended training configuration (max iter = 100):

Table 6: Overall classification performance on the validation set (Configuration 2)

Metric	**Value**
Top-1 Accuracy	85.88%
Top-5 Accuracy	98.08%
Balanced Accuracy	85.88%
Number of samples	5000

Table 7: Classification report on the validation set (Configuration 2)

Averaging	**Precision**	**Recall**	**F1-score**	**Support**
Macro average	0.8610	0.8588	0.8587	5000
Weighted average	0.8610	0.8588	0.8587	5000
Overall accuracy	–	–	0.8588	5000

Table 8: Worst 10 classes by classification accuracy (Configuration 2)

Class ID	**Accuracy (%)**	**Support**
64	38.00	50
78	40.00	50
48	48.00	50
9	52.00	50
34	60.00	50
50	60.00	50
88	62.00	50
12	64.00	50
43	64.00	50
42	66.00	50

	path	true	pred
4970	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4971	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4972	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4973	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4974	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4975	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4976	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4977	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	87
4978	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4979	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4980	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75
4981	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	44
4982	/kaggle/input/imagenet100/val.X/n02077923/ILSV...	75	75

Figure 13: classification predict result

Overall, the results obtained with the Extended training configuration (max iter = 100) for the Image Classification linear probing task are quite strong, with misclassifications occurring in only a small number of samples. A more detailed analysis of the results is provided through the following visualizations.

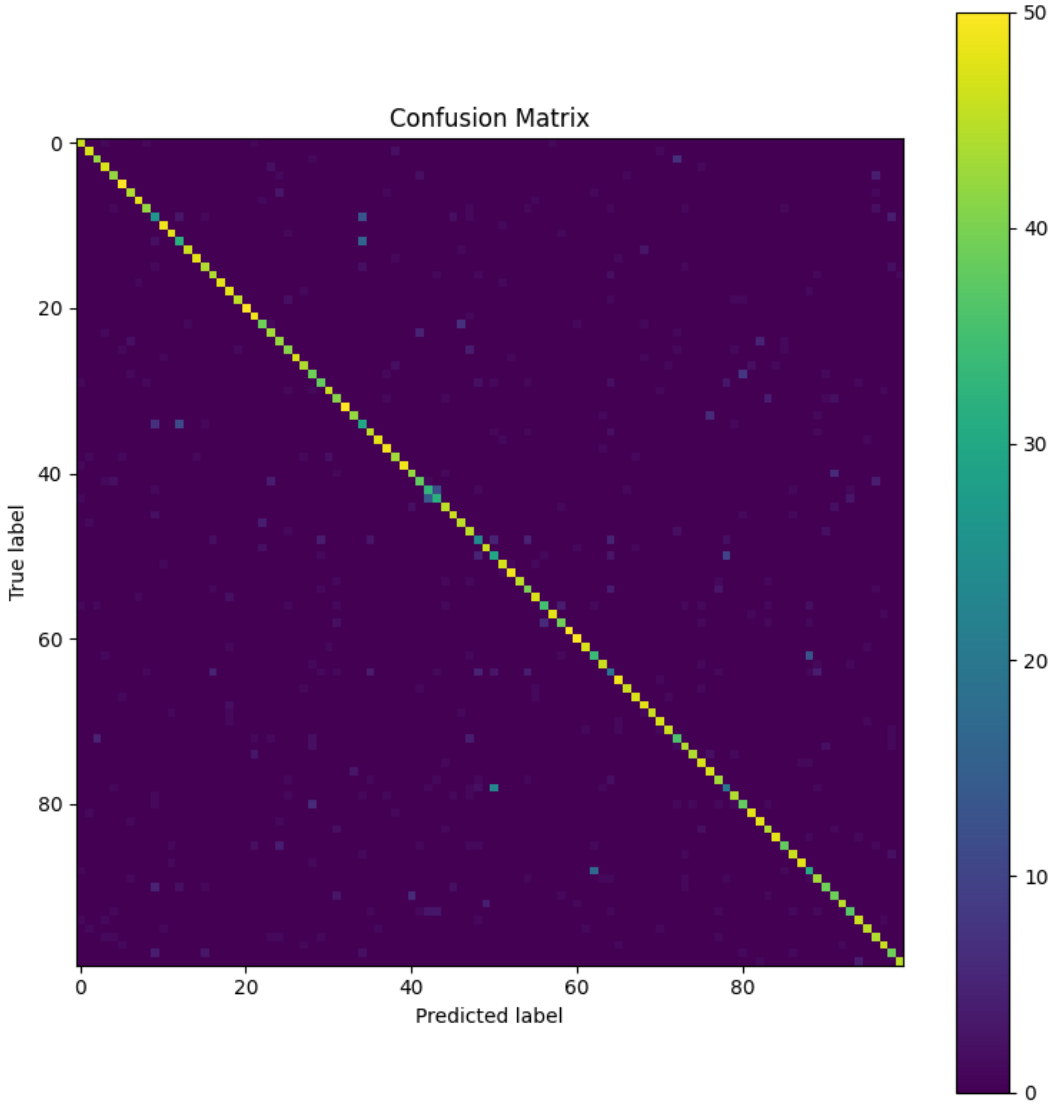

Figure 14: classification diagram result

The results shown in the figure indicate that most labeled samples are correctly predicted, as reflected by the prominent yellow diagonal in the chart.

Evaluation:

For the Image Classification linear probing task, DINOv3 is used as the backbone and produces good results, with most images being classified correctly. This shows that the features extracted by DINOv3 are effective for image classification.

4.3 Semantic Segmentation

Task Description:

This task focuses on evaluating the semantic segmentation capability of a pre-trained DINOv3 visual backbone using a frozen-feature evaluation protocol. The backbone is kept fixed and used to extract dense, pixel-level feature representations, which are then fed into a lightweight decoder network trained to predict per-pixel semantic labels. Segmentation performance is evaluated on a validation set using mean Intersection over Union (mIoU), providing a clear measure of how effectively the learned representations support dense prediction under limited training conditions.

Implementation:

This implementation evaluates semantic segmentation using a frozen DINOv3 ViT backbone combined with a lightweight convolutional decoder. Images from the VOC 2012 segmentation dataset are resized to a fixed resolution and passed through the pre-trained DINOv3 model to extract patch-level token embeddings, while all backbone parameters remain frozen. The patch tokens are reshaped into a 2D feature map and processed by a simple decoder consisting of stacked convolution, normalization, and activation layers to produce per-pixel class logits, which are then upsampled to the original image resolution. Training is performed using cross-entropy loss, optimizing only the decoder parameters, and model performance is assessed on a validation set using mean Intersection over Union (mIoU). The best model checkpoint is selected based on validation mIoU.

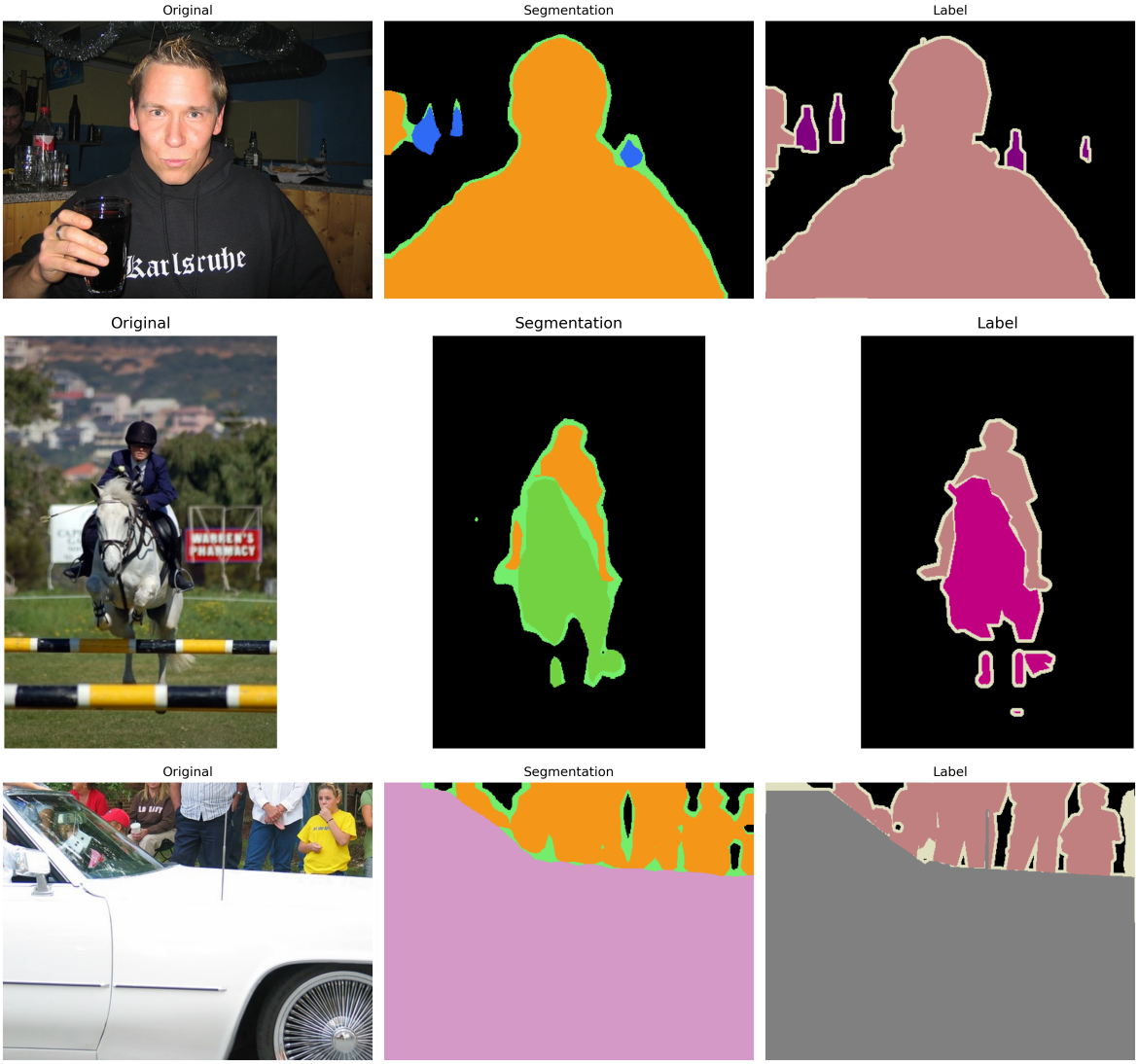

Figure 15: Sematic segmentation result

Experimental Protocol:

Table 9: Hyperparameter settings for the semantic segmentation experiment

Hyperparameter	**Value**
Backbone model	DINOv3 (pre-trained, frozen)
Backbone variants	ViT-B/16
Segmentation head	Lightweight convolutional decoder
Optimizer	AdamW
Learning rate	1×10^{-4}
Batch size	2, 4, 8
Number of training epochs	20
Input resolution	448×448
Patch size	16
Loss function	Cross-entropy loss
Upsampling method	Bilinear interpolation
Evaluation metric	Mean Intersection over Union (mIoU)

Semantic Segmentation Evaluation Metrics

Total Loss Total loss represents the overall training objective of the semantic segmentation model. It is computed using a pixel-wise cross-entropy loss between the predicted segmentation maps and the ground-truth semantic labels, and reflects how well the model learns dense, per-pixel classification during training.

Mean Intersection over Union (mIoU) Mean Intersection over Union (mIoU) is the primary evaluation metric for semantic segmentation. It is computed by averaging the Intersection over Union across all semantic classes and quantifies the overlap between predicted segmentation masks and ground-truth labels. This metric provides a robust measure of segmentation quality by jointly considering false positives and false negatives at the pixel level.

Hyperparameter Benchmark: Batch Size

We benchmark the effect of batch size on semantic segmentation performance to analyze its impact on optimization stability and feature learning quality. Batch size controls the variance of gradient updates during training: smaller batches introduce noisier gradients, while larger batches provide more stable optimization, which can be beneficial when training lightweight decoders on top of frozen pretrained backbones. All experiments are conducted on the Pascal VOC 2012 segmentation dataset using a frozen DINOv3 ViT-B/16 backbone and a simple convolutional decoder. Models are

trained for a fixed number of epochs with the same learning rate (1×10^{-4}) and identical data augmentation settings. Performance is evaluated using mean Intersection-over-Union (mIoU) on the validation set.

Table 10: Semantic segmentation performance under different batch sizes

Batch Size	**Validation mIoU**
2	0.6141
4	0.6459
8	**0.6593**

The results show a consistent improvement in segmentation performance as the batch size increases. A small batch size of 2 yields the lowest mIoU, likely due to high gradient variance and less stable optimization. Increasing the batch size to 4 substantially improves performance, indicating more reliable gradient estimates during training.
The best performance is achieved with batch size 8, reaching an mIoU of 0.6593. This suggests that, in the frozen-backbone setting, larger batches help the decoder better leverage the rich spatial features provided by DINOv3, leading to more accurate pixel-level predictions.

Evaluation:

For the semantic segmentation task, we evaluate DINOv3 on the Pascal VOC 2012 dataset using mean Intersection-over-Union (mIoU) as the evaluation metric. Despite employing a frozen DINOv3 backbone with a lightweight decoder, the model achieves competitive segmentation performance, demonstrating that DINOv3 patch-level representations retain strong spatial and semantic cues that are well suited for dense prediction.
Interestingly, the achieved mIoU is comparable to results reported in prior DINOv3-based segmentation studies, even though those works rely on substantially larger backbone models and are trained or evaluated on larger-scale datasets. This suggests that, for VOC-scale segmentation, the representational quality of DINOv3 features already saturates performance to a large extent, and increasing model capacity alone does not necessarily yield proportional gains.
These results indicate that effective adaptation of pretrained representations, combined with careful optimization, can partially compensate for reduced model and data scale. At the same time, larger models and datasets remain advantageous for improving robustness and generalization beyond the VOC benchmark, particularly for more complex scenes and fine-grained categories.

4.4 Unsupervised Object Discovery

Task Description:

Unsupervised object discovery aims to localize the dominant foreground object in an image without using any class labels or bounding-box annotations. Given a single image, the goal is to separate object regions from background by exploiting internal image cues and learned visual representations. This task is commonly evaluated using the CorLoc metric, which measures whether the predicted object localization overlaps sufficiently with ground-truth bounding boxes.

Implementation:

We implement unsupervised object discovery by combining dense patch features extracted from a self-supervised Vision Transformer (DINOv3) with the TokenCut algorithm. First, the input image is resized to a resolution compatible with the ViT patch size, and patch-level features are extracted from a selected transformer layer while discarding the CLS and register tokens. Pairwise cosine similarities between patch features are used to construct an affinity matrix. TokenCut then applies spectral clustering on the thresholded affinity graph and uses the second eigenvector of the normalized Laplacian to partition patches into foreground and background regions. The resulting binary mask is reshaped to the patch grid, optionally refined using adaptive thresholding and morphological post-processing, and finally upsampled to the original image resolution for evaluation and visualization.

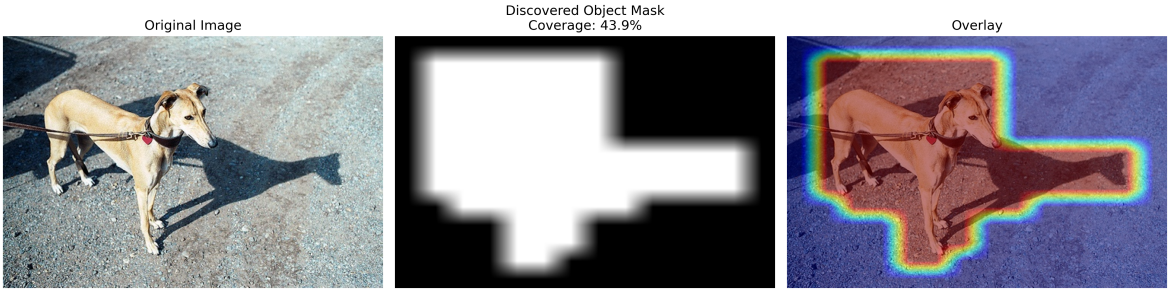

Figure 16: Unsupervised object discovery result

Experimental Protocol:

Table 11: Experimental settings for the task

Setting	**Value**
Backbone model	DINOv3 ViT-S/16 (pre-trained, frozen)
Pre-training method	Self-supervised learning
Input image size	Variable (shorter side resized to 518)
Patch size	16×16
Feature representation	Patch token embeddings
Feature dimension	384
Affinity metric	Cosine similarity between patch features
Graph construction	Thresholded affinity graph
TokenCut threshold (τ)	$\{0.1, 0.2, \ldots, 0.9\}$
Spectral clustering	Normalized cuts (second eigenvector)
Foreground selection	Adaptive median thresholding
Post-processing	Morphological opening, closing, hole filling
Training paradigm	Fully unsupervised (no fine-tuning)
Evaluation dataset	Pascal VOC 2007
Evaluation split	trainval
Evaluation metric	CorLoc (IoU ≥ 0.5)
Additional analysis	Mean IoU, qualitative visualizations

Unsupervised Object Discovery – Key Hyperparameters
TokenCut Threshold (τ): Controls the sparsity of the affinity graph by removing weak patch-to-patch connections. Lower values preserve more global context, while higher values emphasize strong local similarities and can lead to smaller, more compact object regions.

Hyperparameter Benchmark: TokenCut Threshold (τ)

We benchmark the effect of the TokenCut affinity threshold τ on unsupervised object discovery performance. The threshold τ controls the sparsity of the patch affinity graph by retaining only patch-to-patch similarities above a given value. Lower thresholds preserve more global connections, while higher thresholds emphasize strong local affinities and tend to produce more compact object regions.

All experiments are conducted on the Pascal VOC 2007 trainval split using a frozen DINOv3 ViT-S/16 backbone. Performance is evaluated using the CorLoc metric with an IoU threshold of 0.5, along with the corresponding mean IoU across samples.

Table 12: Object discovery performance under different TokenCut thresholds

Threshold (τ)	**CorLoc (%)**	**Mean IoU**
0.10	29.40	0.3362
0.20	28.66	0.3292
0.30	18.66	0.2247
0.40	19.58	0.2387
0.50	27.40	0.3121
0.60	28.66	0.3232
0.70	27.14	0.3187
0.80	27.04	0.3187
0.90	**29.93**	**0.3380**

The results show that the TokenCut threshold has a non-monotonic impact on object localization performance. Very low thresholds ($\tau \leq 0.2$) retain excessive background connections, while intermediate values ($\tau \in [0.3, 0.4]$) significantly degrade performance due to over-fragmentation of the affinity graph. Higher thresholds recover performance by focusing on strongly correlated patch regions, leading to more coherent object masks.
Overall, the best performance is achieved at $\tau = 0.9$, yielding a CorLoc of 29.93% and a mean IoU of 0.3380. Based on these results, $\tau = 0.9$ is selected as the default threshold for subsequent experiments unless stated otherwise.

Evaluation:

For the unsupervised object discovery task, DINOv3 combined with TokenCut is evaluated on the Pascal VOC 2007 dataset using the CorLoc metric. The method is able to consistently localize the dominant foreground object in a subset of images, demonstrating that patch-level features extracted by DINOv3 capture meaningful object-centric structure even without supervision.
However, the achieved performance does not match the results reported in the original TokenCut and DINOv3 studies, where CorLoc scores of approximately 66% are obtained. This performance gap is primarily attributed to the significant difference in model scale. In our experiments, we employ a DINOv3 ViT-S/16 backbone with roughly 21 million parameters, whereas the reported state-of-the-art results rely on substantially larger models with billions of parameters (up to 7B). Larger models pro-

vide richer and more globally consistent representations, which are particularly important for dense prediction tasks such as object discovery.

5 Conclusion

Overall, the findings indicate that DINOv3 is a powerful and flexible vision foundation model that can be effectively applied to a wide range of computer vision tasks, including classification, detection, and dense prediction. This project demonstrates the potential of self-supervised learning approaches in reducing dependency on labeled data while maintaining strong performance. Future work may focus on extended training, task-specific fine-tuning, and applying DINOv3 to additional domains such as video understanding and geospatial analysis.

6 References

1. How AI Taught Itself to See [DINOv3] , Youtube, url, accessed: 25/12/2025
2. DINOv3 Explained: Technical Deep Dive, LIGHTLY, url, accessed: 25/12/2025
3. DINOv3, arxiv, url, accessed: 25/12/2025