Semantic Correspondence with Visual Foundation Models

TA: claudia.cuttano@polito.it

Introduction

Semantic correspondence is the task of finding pixel-level matches between semantically similar parts of objects across different images. For example, given a point on the left eye of a dog in one image, the goal is to find the location of the corresponding left eye in another image of a dog, or even a semantically related object such as a wolf or a cartoon version of the dog.

This problem is particularly challenging because:

- Objects may appear in different viewpoints, scales, or contexts.
- Images may come from different domains (*e.g.*, photos vs paintings).
- Models must distinguish semantically similar but geometrically different parts (*e.g.,* left vs right paw).

Recent research has shown that large Vision Foundation Models contain rich internal representations that can be used for semantic correspondence. These representations emerge without explicit supervision, offering a powerful starting point for dense matching tasks.

Literature Review

Several works have laid the foundation for modern semantic correspondence:

- In [1, 2, 3], you will find how **correspondence can emerge naturally** from large pretrained models.
- [4] introduces **SPair-71k**, a benchmark dataset that provides annotated keypoints for evaluating correspondence methods.
- [5–8] present the **evolution of visual foundation models**, including DINO and SAM. DINO [5–7] shows how self-supervised ViTs capture semantic structure, while SAM [8] offers strong segmentation capabilities.

High-level goal

The goal of this project is to **establish pixel-accurate semantic correspondences** between two images using features extracted from **pretrained visual encoders**.

Given i) a **source image** with annotated keypoints, and ii) a **target image**: the task is to predict, for each source keypoint, the **corresponding location in the target image**.

The project proceeds in **four main stages**:

1. **Training-free baseline:** use frozen features to perform semantic correspondence.
2. **Light fine-tuning:** adapt the last layers of the backbone to improve performance.
3. **Better prediction rule:** replace argmax with window soft-argmax.
4. **Extension**

1) Training-free baseline

To evaluate how well different models encode correspondence, we use **SPair-71k**, a standard benchmark for semantic correspondence. Each image pair in this dataset comes with annotated keypoints that represent the same semantic part (e.g., the tip of a dog's ear, the wheel of a car) across different object instances or viewpoints.

For each pair of source and target images:

1. We extract **dense features** from a pretrained backbone.
2. For every source keypoint, we compute its **cosine similarity** with all patch features in the target image, producing a similarity map.
3. The location with the highest similarity is selected as the predicted match.

Backbone Candidates

We will compare different pretrained **encoders** to understand how well they support correspondence:

- DINOv2
- DINOv3
- Segment Anything (SAM)

Evaluation Protocol

We follow the standard protocol from DIFT [1], using **PCK@T** (Percentage of Correct Keypoints) as the main metric. PCK measures the percentage of keypoints predicted within a certain normalized distance from the ground truth. We use multiple thresholds (e.g., 0.05, 0.1, 0.2) to analyze performance at different precision levels.

Results will be reported:

- **Per keypoint**

- **Per image**

This analysis will show how each backbone behaves across categories and difficulty levels.

2) Light Finetuning of the Last Layers

In the second stage, we keep the same pipeline but **unfreeze the last layers** of the backbone and fine-tune them using keypoint supervision from SPair-71k.

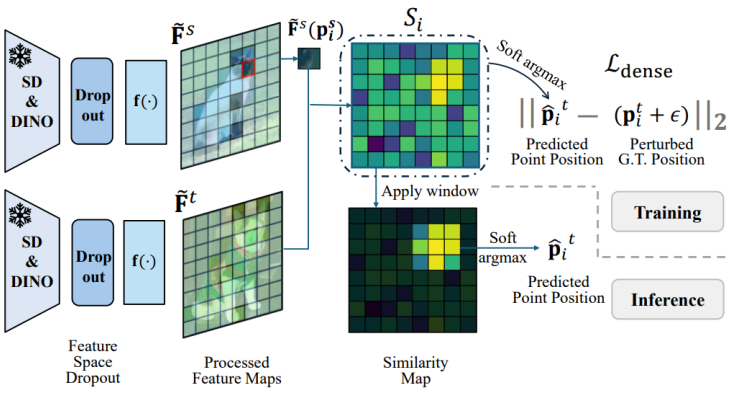
By testing different numbers of finetuned layers, we can observe how performance evolves as the model is given more flexibility to adapt to the task. This highlights how a small amount of fine-tuning can significantly boost correspondence quality.

3) Prediction

In the baselines above, the final correspondence is obtained using argmax on the similarity map. However, this has clear limitations: *i)* it only predicts discrete pixel locations and *ii)* it is sensitive to local noise and can miss subtle details.
As proposed by Zhang et al. [3], we replace this with window soft-argmax:

1. Find the peak location with argmax.
2. Apply soft-argmax only within a small fixed window around the peak.

This allows sub-pixel refinement and makes the prediction more robust to noisy similarity maps. In this step, you will evaluate how this change affects PCK across different thresholds.



4) Mandatory Extension

After completing the main steps of the project, students are encouraged to **explore and experiment**. Below are some **example directions** you can take, but you are free to choose others:

- **Try different backbones** beyond DINOv2 and SAM, such as Stable Diffusion features [1, 2].
- **Test on new datasets**, e.g., PF-Pascal, PF-Willow, or AP-10K, to see how well your method generalizes across domains.

- **Add Adapter** [9]**, Adapformer** [10] **or LoRA** [11] to explore efficient fine-tuning strategies.
- **Test on Geometric tasks**, *e.g.* point tracking (DAVIS)

---

[1] Tang et al., NeurIPS 2023 — *Emergent Correspondence from Image Diffusion*

[2] *Zhang et al., NeurIPS 2023* — *A Tale of Two Features: Stable Diffusion Complements DINO for Zero-Shot Semantic Correspondence*

[3] *Zhang et al., CVPR 2024* — *Telling Left from Right – Identifying Geometry-Aware Semantic Correspondence*

[4] Min et al., ICCV 2019 — *SPair-71k: A Large-scale Benchmark for Semantic Correspondence*

[5] Caron et al., ICCV 2021 — *Emerging Properties in Self-Supervised Vision Transformers*

[6] Oquab et al., CVPR 2023 — *DINOv2: Learning Robust Visual Features without Supervision*

[7] Simeoni et al., 2025 — *DINOv3*

[8] Kirillov et al., ICCV 2023 — *Segment Anything (SAM)*

---

[9] Houlsby et al., ICML 2019 — *Parameter-Efficient Transfer Learning for NLP*

[10] Chen et al., CVPR 2022 — *AdaptFormer: Adapting Vision Transformers for Scalable Visual Recognition*

[11] Hu et al., ICLR 2022 — *LoRA: Low-Rank Adaptation of Large Language Models*