AUTOMATED KEY POINT IDENTIFICATION AND DESCRIPTION FOR VISION TRANSFORMERS USING VISION-LANGUAGE MODELS

Siyan Chen, s232894 *
Department of Applied Mathematics and Computer Science
Denmark Technical University
Anker Engelunds Vej 101, 2800 Kongens Lyngby, DK
s232894@dtu.dk
Supervised by Kristoffer Wickstrøm, Lars Kai Hansen

ABSTRACT

Explainable AI (XAI) has emerged as a crucial research field aimed at improving the transparency and reliability of deep learning models. Current approaches, ranging from class activation maps to prototypical models, offer varying levels of interpretability. Yet, these models often fail to provide human-readable explanations. Therefore, in the project, we propose a novel framework called Keypoint Labeling Classifiers that explores the Vision Transformer (ViTs) with vision-language models (VLMs) to further enhance the explainability in fine-grained image classification tasks by enabling automated identification and textual description of matching keypoints between the query and prototypes. Code is available at https://github.com/chensy618/SuperpixelCUB

1 INTRODUCTION

Computer vision is a rapidly evolving field in the context of artificial intelligence, which enables machines to process and interpret complex visual data across various areas such as healthcare (Møller et al., 2025; Østmo et al., 2023) and transportation (Dilek & Dener, 2023). Specifically, image classification, object detection and recognition, and semantic segmentation are the most popular tasks in the current research domain. In 2020, an encoder-only Transformer was adapted for computer vision, yielding the Vision Transformer (ViT), which reached state of the art in image classification, overcoming the previous dominance of Convolution Neural Network (CNN) (Dosovitskiy et al., 2020). Besides, ViT-based foundation models have been shown to perform automatic keypoint matching with high precision (Amir et al., 2021), but a problem is that it remains unclear what exactly is being matched. Furthermore, like many deep learning models, vision transformers are often regarded as black-box models with limited interpretability, making it challenging to understand the decision-making processes. To strengthen the explainability of ViTs, we apply the vision-language models integrated with visual and textual processing, aiming to shed light on why a vision transformer model makes a certain prediction. Therefore, the aim of this project is to investigate how vision-language models can automatically identify what the matches are and generate the semantic textual descriptions that clarify the meaning of matched keypoints.

We began our investigation by integrating our keypoint labeling approach with the Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021) framework, which has been completed as part of the final project for the Deep Learning (02456) (Technical University of Denmark, 2025) course. However, we observed that CLIP struggled with patch-level recognition, which is also supported in the literature (Mukhoti et al., 2023). As a result, to continue our work, we shifted our focus to models specifically designed for part-level recognition, beginning with Patch Aligned Contrastive Learning (PACL) (Mukhoti et al., 2023) and Side Adapter Network (SAN) (Xu et al., 2023), and subsequently exploring Open-Vocabulary Part Segmentation (OV-PARTS) (Wei et al., 2023) and Going Denser with Open-Vocabulary Part Segmentation (VLPart) (Sun et al., 2023).

*https://chensy618.github.io/siyan

2 VISION-LANGUAGE MODEL

Vision-language models originated from the integration of computer vision and natural language processing (NLP), and have developed significantly over the past few decades. Vinyals et al. (2015) proposed the Neural Image Caption (NIC) model, an end-to-end neural network system that can automatically view an image and generate a reasonable description in plain English. This model was the first to unify CNN for image encoding and Long-Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) network for caption generation, establishing a foundation architecture for image-to-text translation tasks. Later, attention mechanisms improved the model performance by introducing visual attention to focus on salient regions (Xu et al., 2015). Subsequently, a major shift occurred with the rise of Transformer architecture in NLP (Vaswani et al., 2017). It replaced the recurrent layers most commonly used in encoder-decoder architecture with multi-headed self-attention. Then, pre-trained language models such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2018) were extended to handle visual inputs, paving the way for Transformer-based multi-modal architectures. These models are typically trained on large-scale paired datasets consisting of images and their corresponding textual descriptions, enabling them to learn joint representations across modalities. More recently, large VLMs became more general-purpose and were treated as foundation models that could be fine-tuned or prompted for a variety of downstream tasks, demonstrating remarkable performance (Radford et al., 2021; Li et al., 2022; Alayrac et al., 2022; Wang et al., 2021) Below, we review some of the most important developments within vision-language models.

2.1 CLIP (CONTRASTIVE LANGUAGE–IMAGE PRE-TRAINING)

CLIP is proposed by Radford et al. (2021) that learns visual concepts from natural language supervision. It is trained on 400 million (image, text) pairs collected from the internet by using a contrastive loss. The architecture of CLIP illustrated in Fig.1 consists of two encoders: an image encoder, typically a CNN or a Vision Transformer (ViT), and a text encoder, usually a Transformer. The model learns to align the embeddings of image-text pairs in a shared space by applying the cosine similarity, which provides scores that quantify how closely each text label corresponds to the given image. For example, given a batch of N (image, text) pairs, CLIP is trained to predict which of the $N \times N$ possible (image, text) pairings across a batch actually occurred. It maximizes the cosine similarity between embeddings of the true image–text pairs in a batch while minimizing the similarity between all incorrect pairings ($N^2 - N$). Crucial to note that a **symmetric** cross-entropy loss is used to ensure the bidirectional alignment (image $\rightarrow$ text and text $\rightarrow$ image). If it is only trained in one direction, it might miss useful gradients from the reverse direction. Moreover, without symmetry, one encoder might dominate the learning process, and the other could collapse. Therefore, symmetric loss keeps both encoders actively learning generalized and semantically meaningful representations mutually, which indirectly forms the foundation of the zero-shot classification ability. During inference, CLIP performs zero-shot classification by comparing the embedding of a query image with the embeddings of a set of textual prompts that describe candidate classes (e.g., "a photo of a cat", "a photo of a dog"). The class whose text embedding has the highest cosine similarity to the image embedding is selected as the prediction.

2.2 PATCH ALIGNED CONTRASTIVE LEARNING

Patch-Aligned Contrastive Learning (PACL) (Mukhoti et al., 2023) builds upon CLIP by introducing a modified contrastive compatibility function that aligns patch tokens from a ViT-based image encoder with the CLS token from the text encoder. Instead of modifying the core CLIP encoders, PACL integrates a lightweight vision embedder on top of the frozen vision encoder to project patch-level features into a shared multi-modal embedding space. During training, only this embedder is updated. The key idea is to compute a weighted sum of patch embeddings, where the weights are calculated by their similarity to the text embedding. This aggregated representation is then used to compute a revised compatibility score for the contrastive loss, thus enabling fine-grained alignment between visual regions and textual concepts. For semantic segmentation, given an input image x and a set of class names $Y = y_1, ..., y_C$, the model computes the patch-level similarity $s(x, y_C)$ for each class y_C. These similarity matrices act as class-specific segmentation masks, and a softmax is applied across classes at each pixel location to produce the final segmentation map. Therefore, this

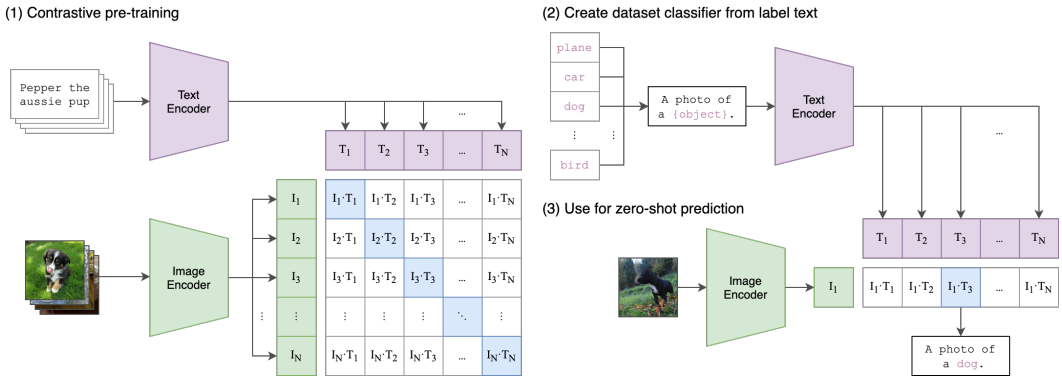

Figure 1: Architecture of the CLIP model (figure from Radford et al. (2021))

approach allows PACL to perform zero-shot segmentation without any segmentation annotations during training.

2.3 Side Adapter Network for Open-Vocabulary Semantic Segmentation

Similar to the PACL, the SAN (Xu et al., 2023) model also leverages a frozen CLIP backbone, but proposes a side adapter network. This network introduces two branches: one for generating object-level mask proposals, and the other for predicting attention bias which is applied in the CLIP model to recognize the class of masks. To enhance mask recognition, SAN creates a set of shadow [CLS] token copies called [SLS] tokens, and each of them corresponds to a predicted mask. These tokens are then updated through attention mechanisms biased toward specific regions, allowing them to capture localized semantic information. The final class predictions are obtained by comparing the [SLS] tokens with CLIP's text embeddings. To generate the segmentation map, the predicted masks $M \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times N}$ are multiplied with the transposed class probability matrix $P \in \mathbb{R}^{C \times N}$, resulting the segmentation map $S \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C}$ over all classes. Through end-to-end training guided by the practice of Cheng et al. (2022), SAN enables the frozen CLIP model to perform open-vocabulary semantic segmentation effectively, while keeping the number of trainable parameters and computational cost low.

2.4 Open-Vocabulary Part Segmentation

The architecture of Open-Vocabulary Part Segmentation(OV-PARTS) (Wei et al., 2023) is built around adapting existing vision-language models, especially CLIP, for part-level segmentation using two primary paradigms. In the two-stage framework, ZSseg (Xu et al., 2022) is selected as the baseline, which includes a proposal generation stage and a zero-shot prediction stage. The proposal generation stage mainly contains a class-agnostic segmentation model like MaskFormer (Cheng et al., 2021) that generates region proposals. These proposals are then classified using CLIP zero-shot capability by computing the similarity between the masked visual features and text embeddings of part labels. To improve classification accuracy, compositional prompt tuning based on CoOP (Zhou et al., 2022) is introduced, where a learnable prompt structure (e.g., "[v_1 OBJECT v_2 PART]") is encoded by the CLIP text encoder to generate part-aware text embeddings. Specifically, mask pooling is adopted to get CLIP visual features for both the object mask and the part mask, which is particularly important in cases where part regions occupy only a small portion of the image or are easily overshadowed by the full object. Empirically, this prompt tuning strategy is effective in enhancing part-level segmentation ability while requiring minimal additional data and training overhead.

In the one-stage setup, the CLIPSeg (Lüddecke & Ecker, 2022) is employed as the baseline model. It leverages frozen CLIP encoders to generate the visual and textual embeddings, and these frozen features are then passed through the extra light-weight CLIP-Adapter modules (Gao et al., 2024) to transform them into task-specific representations that are suitable for pixel-level part segmentation. Subsequently, these representations are fused by the FiLM (Feature-wise Linear Modulation) (Perez

et al., 2018) module to form the cross-modal embeddings, which will be fed into a multi-class pixel decoder that produces the final segmentation masks.

Overall, based on performance comparisons on the dataset of Pascal-Part-116 (Chen et al., 2014) and ADE20K-Part-234 (Zhou et al., 2017; 2019), the two-stage setup offers superior segmentation accuracy with object-aware part reasoning, while the one-stage setup is more efficient and demonstrates better generalization to unseen parts.

2.5 Going Denser with Open-Vocabulary Part Segmentation

VLPart (Sun et al., 2023) introduces a detector with the joint ability of open-vocabulary object detection and part segmentation by extending the standard Mask R-CNN (He et al., 2017) framework. Notably, the important components of VLPart's capability are two powerful foundation models: CLIP (Radford et al., 2021) and DINO (Caron et al., 2021). The architecture consists of an image encoder, a detection decoder, a CLIP-based classification head, and a class-agnostic mask decoder. The image encoder can be a CNN (e.g., ResNet (He et al., 2016)) or Transformer-based model (e.g., Swin (Liu et al., 2021)) with a Feature Pyramid Network (Lin et al., 2017) to produce multi-scale feature maps. The detection decoder includes a Region Proposal Network (RPN) (Ren et al., 2015) that generates region proposals for both objects and parts, followed by an R-CNN recognition head that predicts bounding boxes and classification scores. Unlike standard detectors, VLPart replaces the classifier weights with text embeddings generated by the CLIP text encoder, which allows the model to classify regions by computing the dot product between region features and text embeddings of object-part names. The mask decoder is adapted from Mask R-CNN but modified to be class-agnostic, enabling segmentation of novel object parts without requiring class-specific mask heads. To handle unseen object categories, VLPart integrates DINO as a visual semantic matcher to find a base object similar to each novel object and uses dense feature correspondence to transfer part annotations across categories.

Furthermore, this detector is trained on the joint of part-level, object-level, and image-level data for the establishment of multi-granularity alignment. In particular, the part segmentation data sourced from Chen et al. (2014); He et al. (2022); Ramanathan et al. (2023) includes part mask segmentation and its category, serving as essential supervision for learning finer-grained part understanding. It is worth noting that each part is defined in the context of its associated object, as the same semantic part (e.g., "tail") can appear visually different depending on the object it belongs to (e.g., a cat versus a dog). To reflect this, they defined part categories as object-part pairs, where each label explicitly encodes both the object and its corresponding part. The category names are formalized as:

$$C_{\text{part}} = [\text{“dog: head”}, \text{“dog: nose”}, \ldots, \text{“cat: tail”}]$$

This formulation stimulates the model to better distinguish parts with shared names but different appearances across object categories. To further improve the part recognition ability, the novel object is parsed into parts based on its nearest base object identified through feature similarity. Overall, extensive experiments showed that VLPart can significantly improve the open-vocabulary part segmentation performance and achieve favorable performance on a wide range of datasets such as Pascal Part (Chen et al., 2014), PartImageNet (He et al., 2022) and PACO(Ramanathan et al., 2023).

3 Adapting Vision-Language Models for Keypoint Descriptions

As a starting point, we evaluated the PACL model, as it presents a promising potential for enhancing the explainability of our proposed methodology. However, the source code was not publicly available, which limited our ability to reproduce their results. Therefore, our initial goal was to build it from scratch, but it proved to be time-consuming and involved a heavy workload due to the lack of insufficient implementation details. Then, we explored alternative approaches that are more accessible and better supported by publicly available implementations. Among them, we first identified the SAN model as its segmentation results on ImageNet (Deng et al., 2009) demonstrated strong object segmentation capabilities, even on complex and cluttered images. We later extended our evaluation to more advanced models, including the OV-PARTS and the VLPart, both of which offer structured part-level supervision and richer semantic correspondence.

In order to identify the most suitable model for our task, we conducted a series of qualitative experiments, as shown in Appendix A, where the VLPart demonstrated the best overall performance. Thus, VLPart was chosen as the foundation for our method.

To enable semantic keypoint labeling, we integrated the VLPart into our framework called Keypoint Labeling Classifiers (KLCs), which can automatically generate interpretable keypoints that align part-level visual features with textual descriptions. As aforementioned, the VLPart is also one of the ViT-based models and is enhanced with semantic correspondence mechanisms. Therefore, KLCs extract semantic segments with interpretable labels for both queries and prototypes by leveraging the VLPart model. Then, KLCs compare *parts* of the query image with *parts* from the prototypes, which is achieved by applying the part-correspondence in the tokens of ViTs (Amir et al., 2021). Following this, a set of matching regions containing keypoints between the query and the prototypes is determined using the best-buddies similarity (Oron et al., 2017). In particular, KLCs discard keypoint pairs that are visually aligned but semantically inconsistent. Finally, these matched keypoints are visualized with human-readable labels to provide qualitative insight into the alignment between query and prototype representations. As a result, the ViT-based model's classification process can be explained through consistent part-level matches and made transparent by meaningful semantic labels. Below, we provide a formal description of the KLCs methodology.

3.1 PART 1 - SEMANTIC KEYPOINT IDENTIFICATION OF QUERY-PROTOTYPE

The fundamental idea of this project is to automatically identify keypoints between a query and a set of prototypes. As previously noted by Amir et al. (2021), the tokens of Vision Transformers (ViTs) encode spatial information about semantic parts. We build upon this finding to establish meaningful correspondences between the query and prototype representations.

We begin by identifying semantically coherent segments in the query image and across the set of prototype images. Let the input consist of a query image I_q and a set of prototype images $\{I_p^i\}_{i=1}^N$, where N is the number of prototypes.

$$I_q \in \mathbb{R}^{H\times W\times C}, \tag{1}$$

$$I_p^i \in \mathbb{R}^{H\times W\times C}, \quad \text{for } i = 1,\dots,N \tag{2}$$

Each image is a standard RGB image of resolution $H \times W$.

To obtain part-level semantic information, we apply a part segmentation model $\mathcal{S}(\cdot)$ (e.g., VLPart) to both the query and prototype images:

$$S_q = \mathcal{S}(I_q), \tag{3}$$

$$S_p^i = \mathcal{S}(I_p^i), \quad \text{for } i = 1,\dots,N \tag{4}$$

where $S_q, S_p^i \in \mathbb{N}^{H\times W}$ are the segmentation maps assigning a semantic part label $l \in \mathcal{L}$ (e.g., wing, beak, leg) to each pixel. These maps serve as the foundation for subsequent keypoint identification and semantic correspondence analysis.

For each semantic part label $l \in \mathcal{L}$, we define the set of pixels corresponding to that label as:

$$R_q^l = \{(x,y) \mid S_q(x,y) = l\}, \quad R_p^{i,l} = \{(x,y) \mid S_p^i(x,y) = l\} \tag{5}$$

We then compute the centroid of each segmented region and define it as the representative keypoint location:

$$k_q^l = \frac{1}{|R_q^l|}\sum_{(x,y)\in R_q^l}(x,y), \quad k_p^{i,l} = \frac{1}{|R_p^{i,l}|}\sum_{(x,y)\in R_p^{i,l}}(x,y) \tag{6}$$

Here, k_q^l and $k_p^{i,l}$ represent the 2D coordinates of the keypoint for part l in the query and the i-th prototype image, respectively.

3.2 PART 2 - ALIGNMENT OF MATCHED KEYPOINTS VIA MUTUAL NEAREST NEIGHBORS

With keypoints located, the next phase of KLCs is to find visually aligned matching keypoints. To perform the matching, we are inspired by the template matching using mutual nearest neighbors (Oron et al., 2017), which had been demonstrated to effectively locate matching keypoints between two images (Amir et al., 2021).

Let $\mathcal{K}_q = \{k_q^l\}_{l \in \mathcal{L}_q}$ be the set of keypoints from the query image and $\mathcal{K}_p^i = \{k_p^{i,l'}\}_{l' \in \mathcal{L}_p^i}$ be the keypoints from prototype image i. For each keypoint $k_q^l \in \mathcal{K}_q$, we define its nearest neighbor in $\mathcal{K}_p^i$ by:

$$\text{NN}_{p \leftarrow q}(k_q^l) = \arg \min_{k_p^{i,l'} \in \mathcal{K}_p^i} \|k_q^l - k_p^{i,l'}\|_2 \tag{7}$$

Similarly, we compute the nearest neighbor of each prototype keypoint in the query set:

$$\text{NN}_{q \leftarrow p}(k_p^{i,l'}) = \arg \min_{k_q^l \in \mathcal{K}_q} \|k_p^{i,l'} - k_q^l\|_2 \tag{8}$$

A best-buddies pair is established if the keypoints are mutual nearest neighbors:

$$\text{MatchSet}(k_q^l, k_p^{i,l'}) \iff \text{NN}_{p \leftarrow q}(k_q^l) = k_p^{i,l'} \ \wedge \ \text{NN}_{q \leftarrow p}(k_p^{i,l'}) = k_q^l \tag{9}$$

All such matched pairs are then used to form the set of aligned keypoints between the query and prototype image i.

Additionally, we construct a set $\mathcal{Y}_M$ that contains the semantic label pairs corresponding to each matched keypoint from the query and prototype images.

$$Let \quad \mathcal{Y}_q = \{y_q^l\}_{l \in \mathcal{L}_q}, \quad \mathcal{Y}_p^i = \{y_p^{i,l'}\}_{l' \in \mathcal{L}_p^i} \tag{10}$$

$$\mathcal{Y}_M = \left\{ (y_q^l, y_p^{i,l'}) \;\middle|\; (k_q^l, k_p^{i,l'}) \in \text{MatchSet} \right\} \tag{11}$$

This set $\mathcal{Y}_M$ captures which semantic parts (e.g., wing, tail) were matched between the query and prototype images.

3.3 PART 3 - VISUALIZATION OF MATCHED SEMANTIC LABELS

To ensure interpretability, we visualize only the matched keypoints whose semantic labels are consistent across the query and prototype images.

We define a filtered subset $\mathcal{M}_{\text{valid}}$ that includes only those keypoint pairs with matching labels:

$$\mathcal{M}_{\text{valid}} = \left\{ (k_q^l, k_p^{i,l'}) \;\middle|\; (k_q^l, k_p^{i,l'}) \in \text{MatchSet}, \ y_q^l = y_p^{i,l'} \right\} \tag{12}$$

Only the keypoint pairs in $\mathcal{M}_{\text{valid}}$ are visualized, each annotated with their shared semantic label $y^l = y_q^l = y_p^{i,l'}$.

4 EXPERIMENTAL SETUP

While our primary objective is to automatically generate semantic descriptions for part-level keypoints to enhance the explainability of the ViT-based models, we additionally conducted a classification experiment to show the effectiveness of our keypoint matching framework.

We used the CUB-200-2011 (Wah et al., 2011) dataset as the standard benchmark for evaluating the class-level classification performance based on the part-level segments. CUB-200-2011 contains

location annotations of object parts for each image, including 15 categories of object parts (back, breast, eye, leg, ...) that cover the bird's whole body. For matching visual features across instances, we rely on the ViTs as dense feature descriptors, inspired by prior work (Amir et al., 2021; 2022) and implemented through dense semantic correspondence. As the visual backbone, we use the *dinov2-base* variant of the DINOv2 vision transformer, which provides the precomputed labels and part-level representations of the prototypes. These serve as the reference for comparison against the segment maps generated by VLPart for the final classification decision. The prediction is made based on the number of matched segments across all prototypes. Specifically, for each prototype, we count how many of its segments are matched to the query. Here, the segments are all generated by the VLPart model. These match counts are aggregated by class label. The final predicted label is the one with the highest normalized count (i.e., the label whose prototypes contribute the most matched segments, relative to all matches). If the predicted label matches the ground truth (i.e., a hit), we count it as a correct prediction and record the result.

The setup is outlined below:

- Image Resolution : All images resized to 224×224 pixels.
- Segmentation : part-level segmentation generated by VLPart.
- Number of Prototypes : 3 prototypes per evaluation.
- Matching Method : Cosine similarity on CLS tokens to select top-1 prototypes (k=3), followed by mutual nearest neighbor (MNN) matching of segments.
- Prediction Strategy : Count the number of matched segments per class and apply **soft voting** (Salih, 2021)

$$\text{Score}(c) = \frac{\text{matched segments from class } c}{\text{total matched segments}} \tag{13}$$

- Evaluation Size : 1000 randomly selected test images.
- Metric : Top-1 classification accuracy (on Computer Science, 2022).

5 RESULTS

The entire visual pipeline of KLCs is illustrated in Fig. 2, where we input a query image and three prototype birds. After loading and segmenting the images with VLPart, segments with label IDs (i.e. a numerical identifier representing part categories such as beak, head, or torso) and part labels are decoded (e.g., beak, head, foot, torso). Keypoints are then identified and matched across images. Finally, unmatched parts are masked out to focus on consistent semantic correspondences. By providing semantically human-readable explanations between the query and matched prototypes, the classification decision of the ViT-based model becomes more interpretable and transparent, thus turning ViT-based foundation models into self-explainable models (Chen et al., 2019; Gautam et al., 2022; Nauta et al., 2023) without retraining.

Table 1: Comparison of KLCs with SOTA models on CUB-200-2011 dataset

				CUB200	
method	ViT?	no retraining of encoder	no training of clf head	Top-1 Accuracy	Prototypes Count
ProtoPNet	✗	✗	✗	79.2	2000
ProtoTree	✗	✗	✗	82.2	202
ProtoPool	✗	✗	✗	85.5	202
PIP-Net	✗	✗	✗	84.3	4
ProtoS-ViT	✓	✓	✗	85.2	6
ST-ProtoPNet	✓	✓	✗	82.7	40
KLCs (ours)	✓	✓	✓	82.7	3

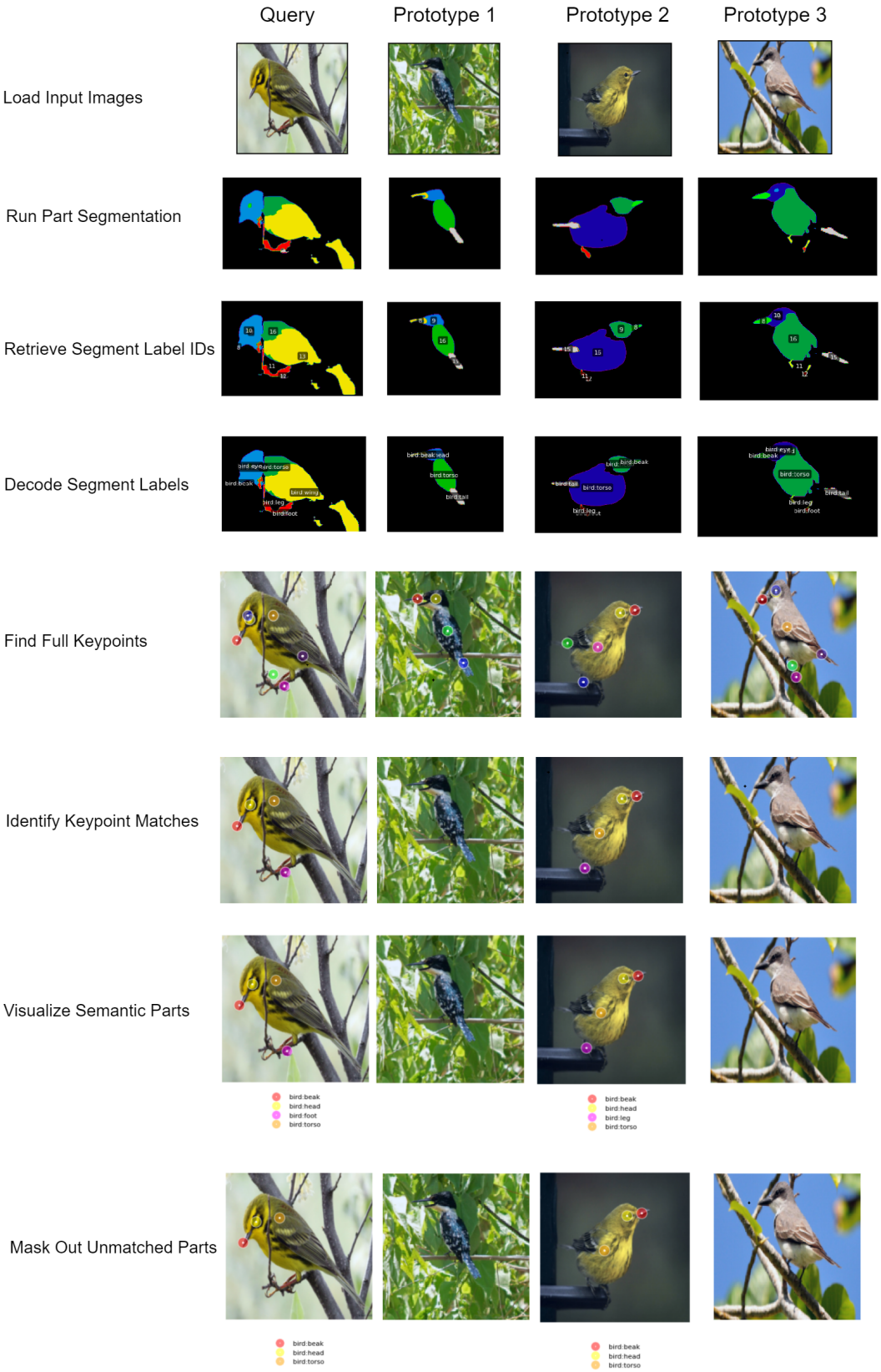

Figure 2: Automated key point identification and description for ViTs using vision-language model

In the experimental evaluation, we showed the comparison results with various state-of-the-art (SOTA) methods for both the CNN and ViT regimes, such as ProtoPNet (Chen et al., 2019), ProtoTree (Nauta et al., 2021), PIP-Net (Nauta et al., 2023), and ProtoS-ViT (Turbé et al., 2024). As presented in Table 1, our approach reaches a classification accuracy of 82.7% using only 3 prototypes, demonstrating a strong capability to capture and align semantic correspondence. Notably, our method is the only one that is not trained on the CUB-200-2011 dataset among the compared approaches, yet it achieves comparable and even superior performance as many recent methods.

6 Conclusion

In conclusion, we developed an automated keypoint labeling framework that leverages the part-level vision-language models to enable interpretable and consistent identification and explanation of critical anatomical features for classification tasks in ViT-based models. Our pipeline supports accurate region segmentation, label assignment, and visual and semantic correspondence matching, which enhances the explainability and reasoning behind the model's decision process. We conducted both qualitative and quantitative experiments, and the VLPart was chosen to serve our framework. Experimental results further prove the effectiveness of our method in supporting the fine-grained classification task while maintaining strong interpretability, showing superior performance compared to existing approaches. We believe that this framework marks a significant step toward improving the reliability and transparency of ViT-based foundation models.

Acknowledgments

By finishing this project, I would like to take this opportunity to firstly express my sincere gratitude to my supervisor, Kristoffer Wickstrøm, for his continuous support, insightful guidance, and invaluable feedback throughout the course of this project. His expertise and encouragement have been instrumental in shaping the direction and quality of my work. Secondly, I also want to show my appreciation to my co-supervisor, Lars Kai Hansen, who took charge of the administrative procedures, for his support and help during this project.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021.

Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. On the effectiveness of vit features as local semantic descriptors. In *European Conference on Computer Vision*, pp. 39–55. Springer, 2022.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1971–1978, 2014.

Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in neural information processing systems*, 34:17864–17875, 2021.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1290–1299, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: a large-scale hierarchical image database. *2009 Ieee Conference on Computer Vision and Pattern Recognition (cvpr)*, pp. 248–255, 2009. doi: 10.1109/CVPRW.2009.5206848.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Esma Dilek and Murat Dener. Computer vision applications in intelligent transportation systems: a survey. *Sensors*, 23(6):2938, 2023.

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.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2):581–595, 2024.

Srishti Gautam, Ahcene Boubekki, Stine Hansen, Suaiba Salahuddin, Robert Jenssen, Marina Höhne, and Michael Kampffmeyer. Protovae: A trustworthy self-explainable prototypical variational model. *Advances in Neural Information Processing Systems*, 35:17940–17952, 2022.

Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision*, pp. 128–145. Springer, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7086–7096, 2022.

Bjørn Leth Møller, Sepideh Amiri, Christian Igel, Kristoffer Knutsen Wickstrøm, Robert Jenssen, Matthias Keicher, Mohammad Farid Azampour, Nassir Navab, and Bulat Ibragimov. Nemt: Fast targeted explanations for medical image models via neural explanation masks. In *Northern Lights Deep Learning Conference 2025*, 2025.

Jishnu Mukhoti, Tsung-Yu Lin, Omid Poursaeed, Rui Wang, Ashish Shah, Philip HS Torr, and Ser-Nam Lim. Open vocabulary semantic segmentation with patch aligned contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19413–19423, 2023.

Meike Nauta, Ron Van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14933–14943, 2021.

Meike Nauta, Jörg Schlötterer, Maurice Van Keulen, and Christin Seifert. Pip-net: Patch-based intuitive prototypes for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2744–2753, 2023.

Baeldung on Computer Science. Top-n accuracy metrics. `https://www.baeldung.com/cs/top-n-accuracy-metrics`, 2022.

Shaul Oron, Tali Dekel, Tianfan Xue, William T Freeman, and Shai Avidan. Best-buddies similarity—robust template matching using mutual nearest neighbors. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1799–1813, 2017.

Eirik A Østmo, Kristoffer K Wickstrøm, Keyur Radiya, Michael C Kampffmeyer, and Robert Jenssen. View it like a radiologist: Shifted windows for deep learning augmentation of ct images. In *2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7141–7151, 2023.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Ilyas Bin Salih. What is hard and soft voting in machine learning? `https://ilyasbinsalih.medium.com/what-is-hard-and-soft-voting-in-machine-learning-2652676b6a32`, 2021.

Peize Sun, Shoufa Chen, Chenchen Zhu, Fanyi Xiao, Ping Luo, Saining Xie, and Zhicheng Yan. Going denser with open-vocabulary part segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15453–15465, 2023.

Technical University of Denmark. 02456 Deep Learning. `https://kurser.dtu.dk/course/02456`, 2025. Accessed May 2025.

Hugues Turbé, Mina Bjelogrlic, Gianmarco Mengaldo, and Christian Lovis. Protos-vit: Visual foundation models for sparse self-explainable classifications. *arXiv preprint arXiv:2406.10025*, 2024.

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.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

Meng Wei, Xiaoyu Yue, Wenwei Zhang, Shu Kong, Xihui Liu, and Jiangmiao Pang. Ov-parts: Towards open-vocabulary part segmentation. *Advances in Neural Information Processing Systems*, 36:70094–70114, 2023.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pp. 2048–2057. PMLR, 2015.

Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. 2022.

Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2945–2954, 2023.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

A Qualitative Experiments

A.1 SAN

Initially, we started the evaluation by extracting patch-aligned visual embeddings to compare them with the given textual embeddings using the pre-trained weight of *san_vit_b_16.pth*. We observed that the model's performance in terms of part-level alignment and semantic segmentation was comparable to that of CLIP-based approaches. Therefore, we decided to adopt the segmentation mask outputs produced by SAN and apply our inference method. Specifically, we used the predicted class logits and spatial masks to compute a semantic segmentation map via a weighted sum over queries. Our method uses softmax-normalized class probabilities and sigmoid-activated spatial masks to generate per-class heatmaps, following the equation:

$$\text{semseg}_{c,h,w} = \sum_{q} \text{softmax(mask\_cls)}_{q,c} \cdot \text{sigmoid(mask\_pred)}_{q,h,w}$$

This means that for each class c, we take the spatial prediction from each query q, weighted by the probability that query q belongs to class c. Based on this approach, we conducted a controlled qualitative experiment, where the original category list used by SAN is modified

to investigate how custom vocabulary affects the SAN's ability to verify bird part representations in the input images. We commented out all default COCO categories related to "bird" in `register_coco_stuff_164k.py` and designed several custom vocabulary configurations to evaluate the model's segmentation behavior. We tested five different vocabulary settings:

1. **Part-only vocabulary (without "bird" but COCO merge)**: We used a vocabulary containing only bird parts such as `"beak"`, `"tail"`, and `"wing"` without including the category `"bird"` itself.
2. **Part vocabulary + "bird"**: We extended the above vocabulary by appending the general category `"bird"`.
3. **Possessive part vocabulary**: We used a detailed list such as `"bird's neck"`, `"bird's back"`, etc., to evaluate whether SAN could localize individual parts.
4. **Animal-level vocabulary (COCO-like)**: We included general animal classes including `"bird"`, `"dog"`, `"cat"`, etc.
5. **Strictly limited parts (no COCO merge)**: We evaluated the part-only vocabulary in isolation, without merging it with any COCO categories.

Results. We observed that the presence of the term `"bird"` in the vocabulary played a critical role in enabling SAN to detect the entire shape of the bird accurately, as shown in Fig.5. Besides, when the vocabulary was restricted to only part names, the model frequently over-predicted the dominant part label for the entire object. For example, in many test cases, the whole bird was labeled as `"beak"` despite the presence of other parts in the vocabulary, as illustrated in Fig.4. Moreover, in Fig.7, using possessive-form part labels (e.g., `"bird's back"`) failed to capture the correct semantic correspondence on birds.

Figure 3: SAN input image (a)

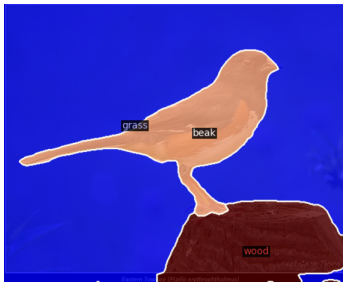

Figure 4: Vocabulary excludes `"bird"`

Figure 5: Vocabulary includes `"bird"`

Figure 6: SAN input image (b)

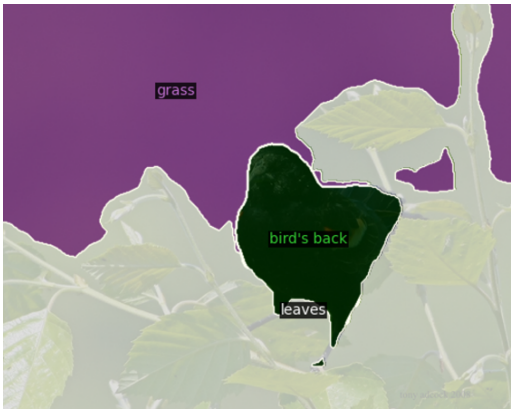

Figure 7: Possessive-form part vocabulary

Conclusion. To sum it up, these results suggest that the SAN model relies on global context provided by higher-level object categories to generate accurate and coherent object-level segmentations, but fails in the part-level recognition even if we provided the part-level vocabulary list.

A.2 OV-PARTS

We tested OV-PARTS in a zero-shot setting using CLIPSeg and CATSeg backbones, with different configuration files and vocabulary prompts.

1. CLIPSeg test setup
   - Mode: Zero-Shot
   - Config: clipseg_voc.yaml
   - Inputs: Fig.8
   - Text prompts: [“bird”, “dog”, “cat”, “branch”, “building”]
2. CATSeg test setup
   - Mode: Zero-Shot
   - Config: clipseg_ade.yaml
   - Inputs: Fig.8
   - Text prompts: [“bird”, “dog”, “cat”, “branch”, “building”]

Observation. The CLIPSeg could identify multiple bird parts such as the head, torso, leg, foot, and tail based on text prompts, despite minor misclassifications where the bird’s foot is labeled as “dog’s paw.” The CATSeg failed to recognize the bird in the image. This is likely because its training dataset did not include bird-related vocabulary. However, after manually adding ”bird” to OBJ_CLASS_NAMES, and part-level labels such as ”bird’s wing”, ”bird’s tail”, ”bird’s head”, ”bird’s eye”, ”bird’s beak”, ”bird’s torso”, ”bird’s neck”, ”bird’s leg”, and ”bird’s foot” to CLASS_NAMES in the configuration file, the model became capable of segmenting the bird and its corresponding parts. Nonetheless, some segments remain noisy or inaccurate for some of the parts. Overall, while OV-PARTS is capable of part-level recognition, its outputs are still relatively coarse and could benefit from further refinement.

A.3 VLPART

To qualitatively evaluate the performance of the VLPart, we conducted inference using the provided *demo.py* script on two sample images from the CUB-200-2011 dataset.
Test setup:

- Mode: Zero-Shot
- Config: r50_pascalpart.yaml
- Pre-trained weight: r50_pascalpart.pth
- Inputs: Fig.12, Fig.14
- Text prompts: [“bird back”, “bird head”, “bird foot”, “bird neck”, “bird beak”, “bird belly”, “bird breast”, “bird crown”, “bird eye”, “bird wing”, “bird nape”, “bird leg”, “bird torso”, “bird tail”]

Summary. The VLPart shows strong performance in zero-shot part segmentation and accurately identifies parts like the beak, eye, torso, wings, and legs. The segmentation masks are highly well-aligned, which clearly illustrate the shape and structure of the bird, highlighting the model’s great spatial understanding and generalization ability.

Figure 8: OV-PARTS input image (c)

Figure 9: CLIPSeg output

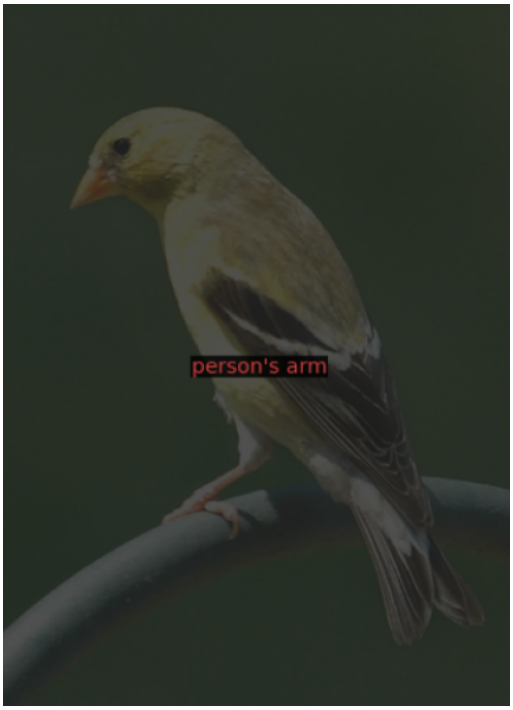

Figure 10: CATSeg output

Figure 11: Adapted CATSeg output

Figure 12: VLPart input (a)

Figure 13: VLPart output (a)

Figure 14: VLPart output (b)

Figure 15: VLPart output (b)