Rheinische
Friedrich-Wilhelms-
Universität Bonn

Institute for Computer Science
Department VI
Autonomous Intelligent Systems

Rheinische Friedrich-Wilhelms-Universität Bonn

Master Thesis

Leveraging Feature Correspondences for Domain Adaptation

Author:
Mayara Everlim Bonani

First Examiner:
Prof. Dr. Sven Behnke

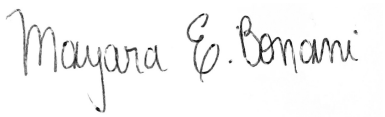
Second Examiner:
Prof. Dr. Florian Bernard

Supervisor:
Max Schwarz

Date: September 6th, 2023

Declaration

I hereby declare that I am the sole author of this thesis and that none other than the specified sources and aids have been used. Passages and figures quoted from other works have been marked with appropriate mention of the source.



Bonn, September 6th 2023

Place, Date

Signature

Abstract

Deep learning methods for computer vision, have allowed the advancement in many areas in robotics; however, such models require large amounts of labeled data. To obtain segmentation ground truth labels for each application is extremely costly. Therefore, the focus of many research areas is to overcome the need of labeled images using different methodologies, such as self-supervised learning methods or using synthetic images. The problem of training a model on synthetic data is the domain gap between synthetic and real data because the differences between their data distributions impair the performance of networks when using only synthetic images.

Our main goal in this thesis is to evaluate different approaches to leverage feature correspondences for Synthetic-to-Real Domain Adaptation, while improving the performance of semantic segmentation models when the training is performed using unlabeled real images and synthetic images. We investigate several approaches to bridge the gap between the features from the synthetic images and the real images, such as adversarial, distance based and variance regularization.

To generate the synthetic images we use the Stillleben framework and we evaluate our models on the YCB-Video dataset. In the first part of this thesis, we consider a frame-level feature adaptation, where we use the distance-based criterion and an adversarial approach. Motivated by exploring local features, in the second part of the thesis, we consider the VICRegL self-supervised approach. Additionally, we use the recently developed Segment Anything Model (SAM) to generate pre-segmentation masks. We use the segments from the SAM masks as a matching criterion to apply the VICReg loss.

In the last part of the thesis, we developed a special block called SAM pooling, where we applied the variance criterion from VICReg to feature regions from different masks. Finally, we also make use of Feature Pyramid Network (FPN) to generate the feature representations, aiming to consider higher resolution feature maps. The combination of the FPN and the SAM Pooling Block leads to a boost in our results, reaching a mean IoU around 85% without the use of ground truth annotations. As a result of our work, we obtained a model that can be trained using unlabeled real images while making the use of the masks from the synthetic image and it is a suitable semantic segmentation model for robotic applications.

Contents

1 Introduction

Deep learning methods for computer vision, such as segmentation, classification, 6D pose estimation, and grasp planning, allowed the advancement in many areas in the robotics and automation field. Although supervised deep learning models have led to great advancements for robotics, they require a large amount of data to be trained that is usually costly to obtain.

The development of autonomous vehicles and mobile robots is intrinsically connected to a good scene understanding of its environments, such as a road or any other path that is surrounding it. In this context, semantic segmentation is an important part in the perception process. The semantic segmentation maps provide for each pixel in an image one class, and therefore, the labels are completely dependent on the set of predetermined classes in the image and in the domain of the autonomous system's specific purpose.

To obtain segmentation ground truth labels for each application is extremely costly. Therefore, the focus of many research areas nowadays is to overcome the need of labeled images using different methodologies, such as self-supervised learning methods (Bardes et al. 2022) or using synthetic images, for which all the labels needed for training are trivially available (Imbusch et al. 2022).

Synthetic data generation is a very promising approach, since precisely-labeled training data can be generated cheaply. In order to obtain synthetic data we use the Stillleben approach (Schwarz and Behnke 2020) that generates training data for perception tasks such as semantic segmentation, object detection, and correspondence or pose estimation. The images provided by Stillleben are obtained through the physics simulation of object meshes and rasterization. It allows the objects to have a randomized appearance and material parameters in addition to noise and transformations that simulate the camera sensors in a scene.

The problem of training a model on synthetic data is the domain gap between synthetic and real data. The differences between their data distributions impair the performance of networks when they are trained using only synthetic images. To overcome this problem, there are some methods which focus on the domain adaptation approach for synthetic images, such as the work by Imbusch et al. (2022). Domain adaptation approaches for synthetic images focus on making the model trained on synthetic data to work well when it is used in another domain, in

this case, when we are testing it for real images. Similarly, in this thesis we focus on using new methods for counterbalancing the need of annotated data through domain adaptation between real and synthetic images.

Our main goal is to use different approaches to leverage feature correspondences for Synthetic-to-Real Domain Adaptation, improving the performance of semantic segmentation models when the training is performed using unlabeled real images and synthetic images. We investigate several alignment approaches, such as adversarial, distance based and variance regularization. All methods are evaluated on the YCB-Video dataset (Xiang et al. 2017).

The first part of our project is called Frame-Level Feature Adaptation. We focus our efforts on using synthetic data in the combination of decreasing the domain gap while improving the semantic segmentation results. We investigate different alignment approaches of the distribution of the feature maps obtained from the RefineNet model (G. Lin et al. 2017) when we use synthetic and real images for the training. We call such approaches frame-level feature adaptation because the alignment between the feature vectors considers the entire region of the image frame, and it does not take into account the scene composition or spatial correspondences. We explore the alignments using two different approaches: l^2 distance-based approach and an adversarial approach.

Motivated its focus on local features, we consider the VICRegL self-supervised approach (Bardes et al. 2022) in the second part of the thesis. First we focus on self-supervised learning with the addition of data augmentation for training models using synthetic data. Specifically, we explore different approaches applying the VICReg variance or using the VICRegL criterion. We evaluate different ways to perform the correspondences between views used for matching features in a local scale, employing real and synthetic images. Then, Segment Anything Model (SAM) (Kirillov et al. 2023) annotations are used as a pre-segmentation step and the features are matched based on the pseudo-labels of SAM masks. The modifications in the model lead us to develop a special block called SAM pooling, where we apply the variance criterion from VICReg to features regions from different masks. Finally, we also make use of Feature Pyramid Network (FPN) to generate the features representations, aiming at improving the quality of the predicted mask. The combination of the FPN and the SAM Pooling Block leads to a boost in our results, matching the performance of a model trained on real annotated data.

This thesis is organized in the following manner. In Chapter 2 we review the related work. In Chapters 3 and 4 we present our results and Chapter 5 is reserved for conclusions, summarizing the results and future work.

2 Related Work

Neural Networks require a large volume of data for training. The nature of the annotation process is costly, time-consuming and can be difficult to obtain a large and diverse dataset. To address the annotation bottleneck problem, techniques have been developed to provide powerful deep feature learning without the requirement of large annotated datasets.

Some approaches make use of synthetic data as an alternative. Unsupervised domain adaptation for synthetic data focuses on improving the synthetic images to get better segmentation results when training only with this type of data.

Another alternative are self-supervised methods that make use of unlabeled data to learn useful representations. The training is performed in a supervised manner using labels generated according to the structure of the data itself, such as different views of the same image. Furthermore, there is also the newest state-of-art image segmentation foundation model, Segment Anything Model (SAM) (Kirillov et al. 2023), which is a promptable segmentation system with zero-shot generalization to unfamiliar objects and images, without the need for additional training.

We focused on these aforementioned methods that avoid the use of labeled data as an inspiration for our work and in this Chapter, we review the most important aspects of each approach.

2.1 Synthetic-to-Real Domain Adaptation using Contrastive Unpaired Translation

Although synthetic data is a viable solution to the annotation bottleneck problem, it suffers from the domain gap between synthetic and real data, i.e. the discrepancy between the statistical properties obtained by the synthetic data distribution and the real data distribution. As a consequence, the model trained using simulated images does not perform as good as the model trained using real-world data.

To overcome the performance degradation when the model is trained using real images, Imbusch et al. (2022) suggested a multi-step learning-based approach to perform synthetic-to-real domain adaptation and consequently minimize the domain gap. It obtains images for the training of a deep neural network for tasks

Figure 2.1: Synthetic (left) and refined (right) image. Source: Imbusch et al. (2022)

like semantic segmentation, focusing on training data for robotic manipulation.

First, synthetic images are generated using the Stillleben library (Schwarz and Behnke 2020). Stillleben operates on arbitrary input meshes and generates random arrangements through the use of a physics engine. Then, the arranged scenes are rendered with a modern physics-based-rendering pipeline, producing realistic images, and a post-processing step adds effects simulating the usage of a real camera.

Secondly, the synthetic images are fed into a domain adaptation network, which outputs images more similar to the real dataset while preserving annotations. The domain adaptation approach in Imbusch et al. (2022) is based on Contrastive Unpaired Translation (CUT) (Park et al. 2020), which is an image-to-image translation technique aimed at preserving the image content while adapting the appearance to the target domain. CUT is a GAN-based approach that uses a contrastive loss on image patches to achieve content preservation by ensuring that a patch of the translated image has more information in common with the same patch in the source image than with other patches from the source image. Instead of applying CUT to images at full resolution, Imbusch et al. (2022) used a patch-based application of the CUT approach and improved the segmentation results obtained from synthetic data for two robotics datasets. An example of a synthetic images and its CUT-refined version is shown in Figure 2.1. Finally, the adapted images are used for training the task network.

In contrast to this method, we do not focus on adapting the images, but we focus on improving the semantic segmentation model performance, leveraging the features for domain adaptation together with the training.

2.2 VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning

A Joint Embedding Architecture, like JEPA proposed by Yann LeCun (LeCun 2022), is a self-supervised learning method in which two networks are trained to produce similar embeddings from different views of the same image. The method also considers the embedding's relation between the images of the batch to avoid the trivial solution in which the network outputs always the same representations for all images. In order to solve this collapse problem, Bardes et al. (2021) proposed a self-supervised method for training joint embedding architectures, called Variance-Invariance-Covariance Regularization (VICReg).

Given an image i sampled from a dataset $\mathcal{D}$, two transformations t and t' (which are random crops of the image followed by color distortions) are sampled from a distribution $\mathcal{T}$ to produce two different views $x = t(i)$ and $x' = t'(i)$ of the image i, as it is shown in Figure 2.2. First the views x and x' are encoded by f_θ (which outputs the representations) into $y = f_\theta(x)$ and $y' = f_\theta(x')$, which are then mapped by the expander h_ϕ (which maps the representations into an embedding space where the loss function will be computed) onto $z = h_\phi(y)$ and $z' = h_\phi(y')$. The loss is then calculated at the embedding level on z and z'.

The method uses a loss function with three terms: variance, invariance and covariance terms. The images are processed in batches of embeddings coming out of the two branches of the siamese architecture, where the two batches $Z = [z_1, z_2, \ldots, z_n]$ and $Z' = [z'_1, z'_2, \ldots, z'_n]$ are composed of n vectors of dimension d each.

The variance regularization term v is defined as a hinge function on the standard

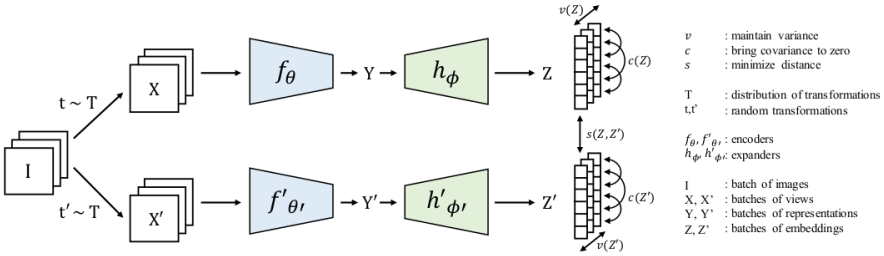


Figure 2.2: VICReg: joint embedding architecture with variance, invariance and covariance regularization. Source: Bardes et al. (2021).

deviation of the embeddings along the batch dimension and above a given threshold

$$v(Z) = \frac{1}{d}\sum_{j=1}^{d} \max(0, \gamma - S(z^j, \epsilon))\,, \tag{2.1}$$

where γ is a constant target value for the standard deviation fixed to 1 in Bardes et al. (2021), ϵ is a small scalar preventing numerical instabilities and S is the regularized standard deviation

$$S(x, \epsilon) = \sqrt{\mathrm{Var}(x) + \epsilon}\,. \tag{2.2}$$

The variance forces the embedding vectors of samples within a batch to be different.

The covariance matrix of Z is defined as

$$C(Z) = \frac{1}{n-1}\sum_{i=1}^{n}(z_i - \bar{z})(z_i - \bar{z})^T\,, \tag{2.3}$$

where $\bar{z} = n^{-1}\sum_{i=1}^{n} z_i$.

It is interesting to define the covariance regularization term c, which encourages the off-diagonal coefficients of $C(Z)$ to be close to zero, decorrelating the different dimensions of the embeddings and preventing an informational collapse in which the variables would vary together or be highly correlated. It is defined as the sum of the squared off-diagonal coefficients of $C(Z)$, with a factor $1/d$ that scales the criterion as a function of the dimension

$$c(Z) = \frac{1}{d}\sum_{i \neq j}[C(Z)]_{i,j}^2\,. \tag{2.4}$$

The invariance criterion s between Z and Z' is defined as the mean-squared Euclidean distance between each pair of embedding vectors, without any normalization

$$s(Z, Z') = \frac{1}{n}\sum_{i} ||z_i - z'_i||_2^2\,. \tag{2.5}$$

The total loss function is then a weighted average of the variance and covariance and invariance terms

$$\ell(Z, Z') = \lambda s(Z, Z') + \mu[v(Z) + v(Z')] + \nu[c(Z) + c(Z')]\,, \tag{2.6}$$

where λ, μ and ν are hyper-parameters controlling the importance of each term in the loss. Bardes et al. (2021) set $\nu = 1$ and performed a grid search on the values

of λ and μ satisfying the condition $\lambda = \mu > 1$.

Finally, the overall objective function taken on all images over an unlabelled dataset $\mathcal{D}$ is given by

$$\mathcal{L} = \sum_{I \in \mathcal{D}} \sum_{t,t' \sim \mathcal{T}} \ell(Z^I, Z'^I)\,, \tag{2.7}$$

where Z^I and Z'^I are the batches of embeddings corresponding to the batch of images I transformed by t and t'. The objective is then minimized for several epochs, over the encoder parameters θ and expander parameters ϕ.

The authors showed that their variance regularization term stabilized the training of other methods and led to performance improvements.

Motivated by the paper's results, we use the same idea to develop a new block, applying the variance to regions in the feature maps that are associated to different masks as it will be shown in the Section 4.7.2.

2.3 VICRegL: Self-Supervised Learning of Local Visual Features

The VICReg approach performs well for classification tasks, however it is not suitable for image segmentation because it removes information related to position and color to satisfy the invariance criterion. To address semantic segmentation tasks, the model needs to focus on learning local information. With this motivation, as a solution to consider global and local features simultaneously, the VICRegL method from Bardes et al. (2022) learns at different scales.

Their paper is one extension of the VICReg mentioned in the previous Section. In the same way, two identical branches of a standard convolutional net architec-

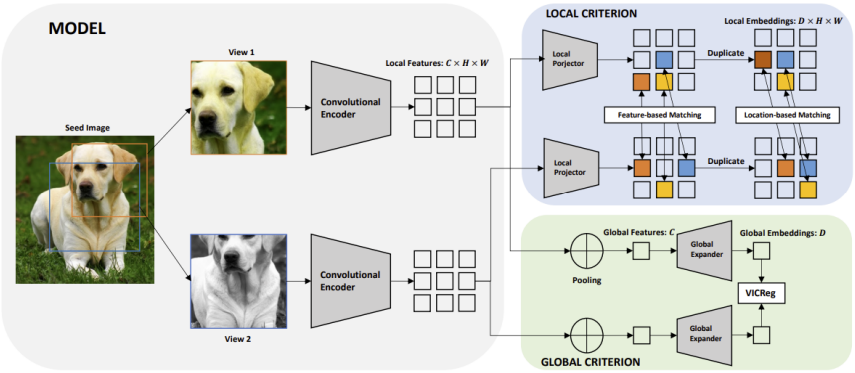


Figure 2.3: Overview of VICRegL: Learning local and global features with VICReg. Source: Bardes et al. (2022)

ture are fed two differently distorted versions of the same image. In Figure 2.3 we have the example of two different views from the same seed image used as an input for the network.

The convolutional neural network is used as an encoder f_θ that outputs local features of size $\mathbb{R}^{C\times H\times W}$, where C is the number of channels and (H, W) is the spatial dimension. The feature maps representations y and $y' \in \mathbb{R}^{C\times H\times W}$ are fed into a local projection network $h^l_\phi : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{D\times H\times W}$, which embed the feature maps y and y' onto feature maps embeddings $z = h^l_\phi(y)$ and $z' = h^l_\phi(y') \in \mathbb{R}^{D\times H\times W}$, preserving its spatial dimension.

The final representations are obtained by performing an average pooling operation $\oplus : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^C$ on the output feature maps. The pooled representations are fed to the expander (same as VICReg) that is here called global expander $h^g_\psi : \mathbb{R}^C \to \mathbb{R}^D$ which maps the pooled representations $y_\oplus$ and $y'_\oplus \in \mathbb{R}^C$ to the embeddings $z_\oplus = h^g_\psi(z_\oplus)$ and $z'_\oplus = h^g_\psi(z'_\oplus) \in \mathbb{R}^D$.

At a global scale, the VICReg criterion is applied in the same way as before. At a local scale, it uses spatial information to match feature vectors that are pooled from close-by regions in the original image (pixels that have a small distance between them), or smallest distance in the embedding space. The VICReg criterion is applied only to the top-γ l^2 nearest-neighbors (NN) feature vectors considering image locations and feature maps. The differential idea is to apply the VICReg criterion between pairs of feature vectors that are associated to a good matching, in the spatial location or in the embedding space. The final goal is to enforce the invariance between them. There are two different types of matching:

- Location-based matching: The idea is to match features that correspond to close positions in the image. The absolute position of the views in the seed image I is known due to the knowledge of each transformation applied to generate it. Using the positions in the seed image, we calculate the l^2 distance between every point of two views, resulting in $H \times W$ pairs of values. From these values, only the top-γ smallest values are used to define the coordinates in the feature maps that are going to be included in the loss, which consists in the VICReg criterion. The loss is defined as

$$\mathcal{L}_s = \sum_{p\in P} \ell(z_p, z'_{\mathrm{NN}(p)}), \tag{2.8}$$

  where the sum is over coordinates p in $P = \{(h, w)|(h, w) \in [1, \ldots, H] \times [1, \ldots, W]\}$.

- Feature-based matching: It consists of matching features that are close in the embedding space with the purpose of capture long-range interactions

between features. Therefore we calculate the l^2 distance for each feature vector z_p at a position p, resulting in the same way, in $H \times W$ pairs. In this case the loss function is defined as

$$\mathcal{L}_d = \sum_{p \in P} \ell(z_p, \mathrm{NN}'(z_p)) \,, \tag{2.9}$$

where the sum is over coordinates p in P and $\mathrm{NN}'(z_p)$ denotes the closest feature vector to z_p in the feature maps z', in terms of the l^2-distance.

The final loss function is a combination of the global loss, which consists in the standard VICReg loss function applied on the pooled representations, and the local loss, which is a combination of the equations (2.8) and (2.9):

$$\mathcal{L}(z, z') = \alpha \ell(z_\oplus, z'_\oplus) + (1 - \alpha)[\mathcal{L}_s(z, z') + \mathcal{L}_s(z', z) + \mathcal{L}_d(z, z') + \mathcal{L}_d(z', z)] \,, \tag{2.10}$$

where α is an hyper-parameter controlling the importance one wants to put on learning global rather than local features.

The quality of the features of the backbone is evaluated by training an additional linear head for semantic segmentation. Bardes et al. (2022) showed that the local criterion significantly improves the performance on segmentation tasks, while preserving the classification accuracy.

We will apply the VICRegL criterion in the same way the paper does. Additionally we will consider different correspondences between the views, using more than one seed image for the training and using synthetic images as a data augmentation, as it is explained in Chapter 4. Furthermore, we include the VICRegL term in our objective when training the model in a supervised manner. Therefore we will consider both: the semantic segmentation loss and the VICRegL loss.

2.4 Segment Anything Model

Foundation models (such as BERT (Devlin et al. 2018), RoBERTa (Y. Liu et al. 2019) and GPT-4 (OpenAI 2023)) enable powerful generalization for tasks and data distributions beyond those seen during training. Kirillov et al. (2023) focused on building a foundation model for image segmentation. The paper contributions are the definition of the promptable segmentation task, the Segment Anything Model (SAM) and the engineered dataset (SA-1B).

The promptable segmentation task's goal is to return a valid segmentation mask given any segmentation prompt, specifying what to segment in an image. A prompt can be any information indicating what to segment in an image, such as a set of

foreground and background points, a rough box or mask or a text information identifying an object.

The SAM can adapt to diverse segmentation tasks via prompt engineering. First, it uses a masked autoencoder pre-trained Vision Transformer (ViT) that is able to process high resolution inputs. There are two types of prompt encoder: one called sparse, to process points, boxes and text information, and other dense, to process masks. The image embeddings and the prompt embeddings are mapped by a mask decoder that predicts multiple valid masks considering the given prompt. Due to this separated structure, the same image embedding can be reused for different prompts. The masks are ranked by an estimated mean IoU. Figure 2.4 shows the explained SAM overview.

As a foundation model it needs strong generalization to new data distributions, and therefore, it is necessary to train SAM on large and more diverse segmentation datasets. Motivated by this, the authors utilized a Segment Anything Data Engine to obtain the Segment Anything Dataset, SA-1B. Therefore, the model was developed with model-in-the-loop dataset annotation. SA-1B consists of 11 million diverse, high-resolution, licensed, and privacy protecting images and 1.1 billion high-quality segmentation masks collected. The automatically generated masks, when compared with professional annotation, are high quality and effective for training models.

For all the evaluations in the paper, SAM uses an MAE pre-trained ViT-H image encoder and SAM was trained only on the automatically generated masks from the dataset SA-1B. The datasets and tasks were not seen during training.

The model is evaluated on the following tasks:

- Segmenting an object from a single foreground point using 23 datasets with diverse image distributions. Compared to the strongest single point segmenter Reviving Iterative Training with Mask Guidance for Interactive Segmentation RIT , SAM yields higher mIoU on 16 of the 23 datasets.

- Classic low-level task of edge detection using BSDS500 dataset. Even though

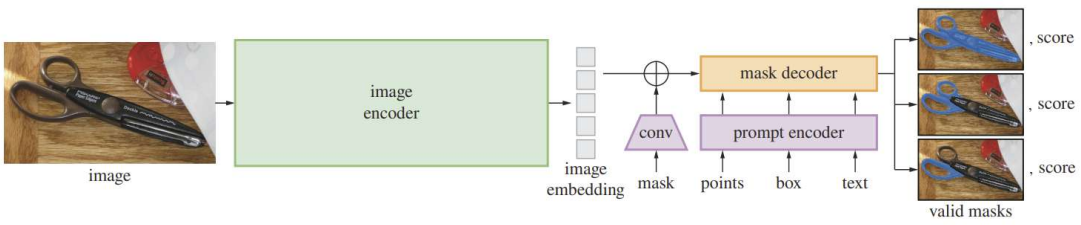


Figure 2.4: SAM overview. Source: Kirillov et al. (2023).

SAM was not trained for edge detection, it produces reasonable edge maps. Compared to the ground truth, SAM predicts more edges, including sensible ones that are not annotated.

- Mid-level task of object proposal generation outputting the masks as proposals. Using the standard average recall (AR) metric on LVIS the ViTDet-H detector outperforms SAM, but the gap shrinks on the higher-quality LVIS masks. Notably, it outperforms ViTDet-H on medium and large objects, as well as rare and common objects.

- Segmentation module of an instance segmenter. It runs an object detector (the ViTDet used before) and prompts SAM with its output boxes. By visualizing outputs, we observed that SAM masks are often qualitatively better than those of ViTDet, with crisper boundaries.

- Segmenting objects from free-form text. SAM can segment objects based on simple text prompts.

As a conclusion, SAM enables zero-shot generalization for new datasets and tasks often by using a prompting technique. Although the model has some limitations, such as to miss fine structures, to segment small disconnected components, and so on, the trained SAM model can segment unfamiliar objects and images without requiring any training or annotations and can be used for different tasks.

SAM cannot be applied directly to YCB-Video because it has no knowledge of the target classes. As it is explained, the model requires good prompting or produces all possible segmentations of the image. The produced segments can intersect each other, being necessary additional processing to obtain only one label per pixel. However, we can use it as a pre segmentation stage. If we use the SAM model to obtain the masks at every training, the performance would be slower. Therefore, we produce the SAM masks for the YCB-Video dataset and store it on disk. Details will be given in Section 4.5.

3 Frame-Level Feature Adaptation

As a first attempt to solve the annotation bottleneck problem, we use synthetic images to avoid the real label annotations. Although synthetic data generation is a very promising approach, since precisely-labeled training data can be generated cheaply, training on synthetic data suffers from the domain gap between real and synthetic data. The models' performance drops significantly compared to training on real data. Two effects come into play here:

1. The distribution of the synthetic dataset may not encompass the distribution of real data (and thus the trained model does not recognize real frames during inference) or

2. The distribution of the synthetic dataset is too wide-spread, making learning impossible.

As a result, we should align the two distributions more closely. One solution is to generate and refine images, in the same way Imbusch et al. (2022) did. However, we focus on our domain adaptation problem to improve the learning process directly. In our case we aim to improve the semantic segmentation model performance when using synthetic images, as it is shown in Figure 3.1. Our goal is to have the features of both domains, $z(X_{\text{real}})$ and $z(X_{\text{syn}})$, to be as indistinguishable as possible keeping the $\mathcal{L}_{\text{Segmentation}}$ as an objective.

To achieve this, the training process will be modified: on synthetic images, where ground truth is available, the target network will be trained as usual in a supervised manner. On the unlabeled real images, we will analyze the distribution of feature activations in the network and try to align this distribution more closely to the distribution on synthetic images.

We focus on the semantic segmentation task, using the RefineNet model (G. Lin et al. 2017), an established network architecture for semantic segmentation. Figure 3.2 shows the structure of the RefineNet network and the feature maps in different scales.

We will investigate several alignment approaches. We call this Chapter Frame-Level Feature Adaptation because we compare the feature maps entirely, differently from the next Chapter, which takes into account the local regions in the feature maps.

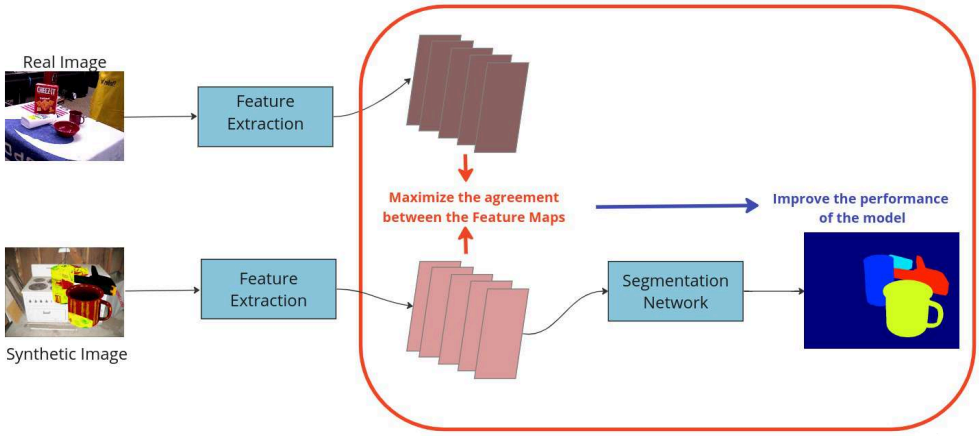


Figure 3.1: Our goal is to have the features of both domains to be as indistinguishable as possible keeping the segmentation loss as an objective.

In order to align the features from both domains, as it is shown in Figure 3.1, the Convolutional Neural Network (CNN) will be trained with two images: one real and one synthetic. It is possible to have two images with the same pose, which we call the paired case, and pairs in which the objects are not in the same position in the image. One limitation of the paired approach is the need for annotation from the object 6D poses to generate the synthetic data. For the unpaired approach it is not necessary to have any annotation from the real image, since the object is rendered in an arbitrary pose in the synthetic image.

We generate synthetic training images using the Stillleben framework (Schwarz and Behnke 2020). Stillleben operates on arbitrary input meshes and generates random arrangements through the use of a physics engine. Figure 3.3 shows the two pairs, paired and unpaired, with synthetic images generated by the Stillleben

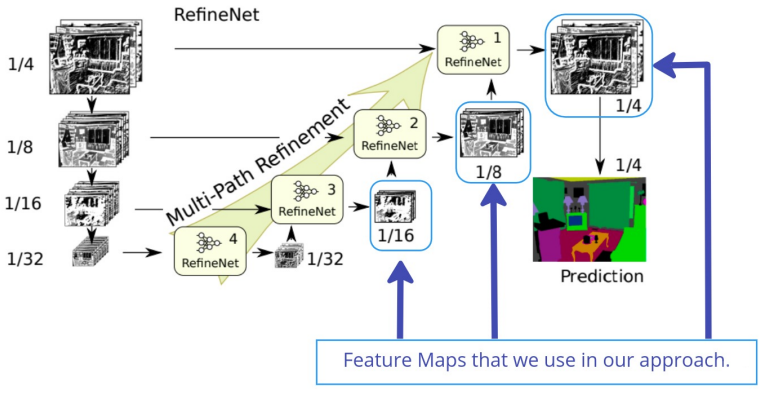


Figure 3.2: RefineNet architecture. Source: G. Lin et al. (2017)

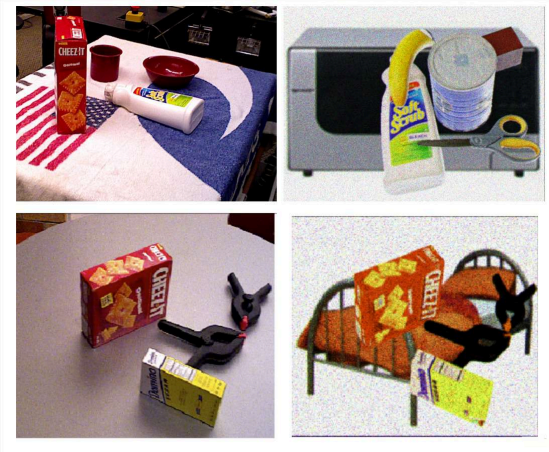


Figure 3.3: Real (left) and synthetic (left) images, where the top ones are unpaired and the bottom ones are paired.

framework.

In our experiments, we show results using a distance based approach on paired images and an adversarial approach on paired and unpaired images.

We choose the YCB-Video dataset (Xiang et al. 2017) as a dataset and we train the RefineNet network from scratch on 450k images, subdivided into 300 epochs of 1500 images. We evaluate both approaches considering the mean Intersection over Union (IoU) over the segmentation obtained from each model in the YCB-Video test dataset. The results shown in this Chapter are obtained by the average of the mean IoU over the last 20 epochs of the training. For all experiments we use the Adam optimizer, learning rate equal to 10^{-5}, and a batch size of 1. For didactic reasons we will present the results along with the methods in this Chapter.

Although we obtained a small improvement when using the paired approach, the conclusions from this Chapter led us to search for an approach that considers local relations between the representations.

3.1 Distance-Based approach

For the distance based approach we use the l^2 norm between the feature maps at the tree last levels of the RefineNet model (G. Lin et al. 2017), as it is shown in Figure 3.2. The total loss is given by

$$\mathcal{L} = \mathcal{L}_{\text{Seg}} + ||z(X_{\text{real}}) - z(X_{\text{syn}})||_2^2 \,. \tag{3.1}$$

Note that we mask the feature activations in the background (labels = 0) so that deviations there are not penalized. Therefore, we only consider regions in the feature maps that are associated to objects of the dataset and we do not consider the background. We aim to decrease the difference between the features generated by the network in points that are associated to the same object, although we consider all the objects at the same time in the loss. Therefore, the backgrounds are not taken into account in this calculation.

3.1.1 Experimental results

We compare the perfomance of our model using the values for the mean IoU when we train RefineNet only considering the training for real and synthetic images.

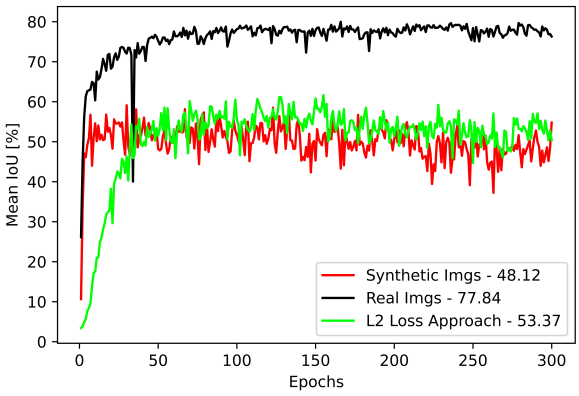


Figure 3.4: Mean IoU for the Distance-Based approach. The numbers represent the average IoU over the 20 last epochs.

Figure 3.4 shows the mean IoU results for when we apply the l^2 criterion to each stage of the feature maps in the RefineNet network. Comparing when we use only synthetic images there is an improvement of 11%, although training on real labels yields even higher performance.

We observed quantitatively and qualitatively that the distance between the feature maps goes to zero, as it is shown in Figure 3.5 and in the images obtained for the feature maps in Figure 3.6. We see that l^2 loss goes to zero, and the difference considering the objects between the feature maps is also small, although the difference between the background is still relevant. Since the performance does not reach real supervision, we conclude that l^2 apparently is not well suited for encouraging feature robustness. Additionally, this approach requires 6D pose annotations to be able to generate paired images.

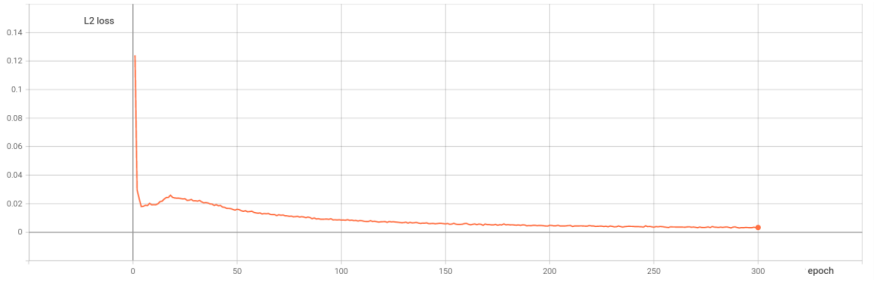


Figure 3.5: l^2 loss.

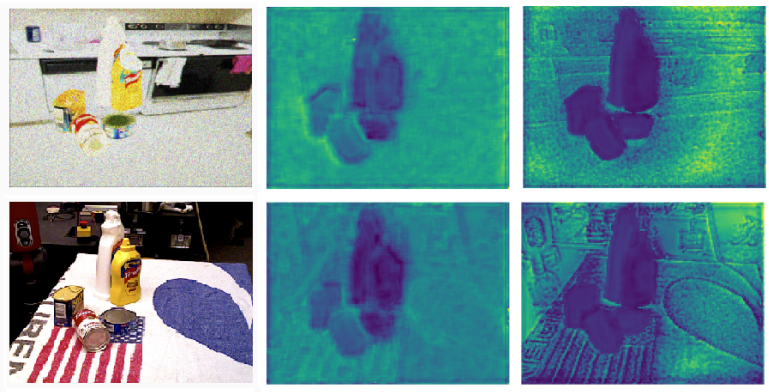


Figure 3.6: Example of features from synthetic image (top row) and real image (bottom row) for 1/4 image size.

3.2 Adversarial approach

Inspired by Generative Adversarial Networks (GANs) proposed by Goodfellow et al. (2014), we compare the features generated by the RefineNet Model in a discriminative manner.

The goal is to train the generator, i.e. the RefineNet model, to produce features from the real and the synthetic image that have a close distribution. The discriminator is trained to distinct such features. We adopt the Least Squares Generative Adversarial Network, LSGAN, or Least Squares GAN (X. Mao et al. 2017), which is a type of generative adversarial network that adopts the least squares loss function for the discriminator.

The minimax objective for GANs can be formulated as

$$\min_G \max_D V_{\text{GAN}}(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))], \quad (3.2)$$

where D is the discriminator, G is the generator, $p_z(z)$ is the distribution of the input variables z and $p_{\text{data}}(x)$ is the distribution of data x.

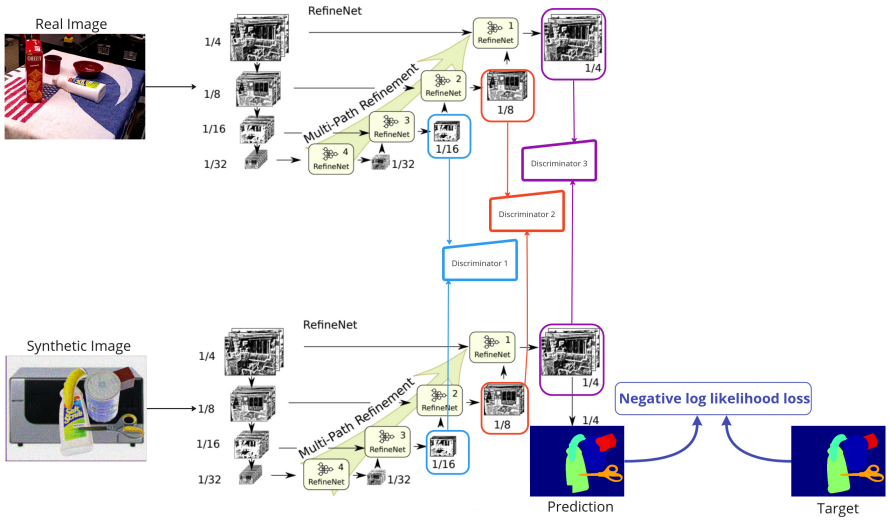


Figure 3.7: Adversarial RefineNet Architecture. We have one discriminator for the last tree stages of the RefineNet model. We also consider in the objective the semantic segmentation loss.

Table 3.1: Mean IoU results for adversarial approach – paired images.

Training data source	Mean IoU (%)
Real images	**77.84**
Synthetic images	48.12
Adversarial approach – 1/16 image size	56.66
Adversarial approach – 1/8 image size	53.85
Adversarial approach – 1/4 image size	51.53
Adversarial approach – 3 feature maps	**57.11**

We show in Figure 3.7 the architecture we used. We applied the loss considering the RefineNet feature maps as the generator and use one discriminator for each one of the last tree stages of the RefineNet model. Considering our goal is to improve the semantic segmentation results, we add the negative log likelihood loss.

3.2.1 Experimental results

In the experiments we perform the training considering the paired and unpaired cases. We show the results when the images are paired.

We apply the discriminator individually to each level of the RefineNet features. The levels have the dimensions 1/16, 1/8 and 1/4 the size of the image. We also compare the results when we apply the discriminator to the 3 levels together. Table 3.1 shows the results and in Figure 3.8 we show the result for the best case in the adversarial approach (3 feature maps). The best improvement was for the

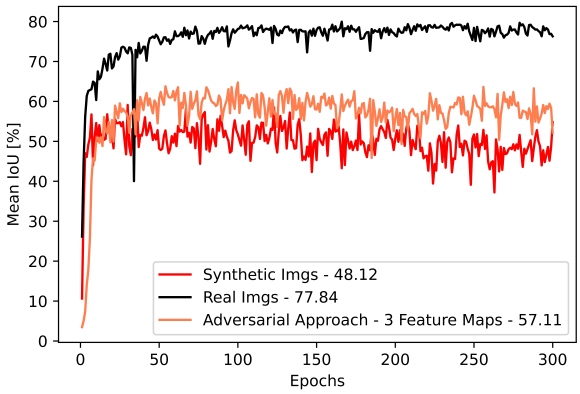


Figure 3.8: Mean IoU for the Adversarial approach.

3 feature maps, with an improvement of 19%.

For the unpaired images, we obtained a mean IoU less than 30%. Due to the competitive nature of the two networks of the GANs' architecture, its training is difficult to converge and unstable. From this approach we conclude that we should take into account correspondence information of some kind.

4 VICRegL Approach

In this Chapter we explain the methods in which we explore the different approaches using the VICRegL criterion or applying the VICReg variance main idea.

First we evaluate different ways to perform the correspondences between views used for matching features in a local scale, employing real and synthetic images. Secondly, we use SAM annotations as a pre-segmentation step and we match the features based on the pseudo-labels of SAM masks.

All the modifications of the model led us to create a special block called SAM pooling, in which we apply the variance idea from VICReg to features regions from different masks. Focusing on improving the quality of the predicted mask, we also make use of Feature Pyramid Network (FPN) to generate the features representations.

We provide details of the implementation of each one of the methods and evaluate their performance. Initially we verify the quality of each way of the local-based matching in the image. Then we focus on evaluating the mean IoU obtained using the methods we developed.

4.1 ConvNeXt Backbone

Considering the excellent performance in VICRegL (Bardes et al. 2022), we use the ConvNeXt architecture (Zhuang Liu et al. 2022) as a backbone. This recent model is similar to to the well-known ResNet architecture (He et al. 2016), however, its modifications allow it to have a performance for computer vision benchmarks similar or better to state-of-the-art hierarchical vision transformers, while keeping the structure of a pure ConvNet.

The architecture ConvNeXt comes from the original ResNet network, being built entirely from convolutional networks. The modifications to the model are inspired by the Swin Transformer design (Ze Liu et al. 2021). For instance, it uses inverted bottlenecks. Additionally, as an activation function, instead of using a Rectified Linear Unit (ReLU), it replaces it by the Gaussian Error Linear Unit, or GELU (Hendrycks and Gimpel 2016), which can be thought of as a smoother variant of ReLU and it is utilized in the most advanced Transformers, such as Google's

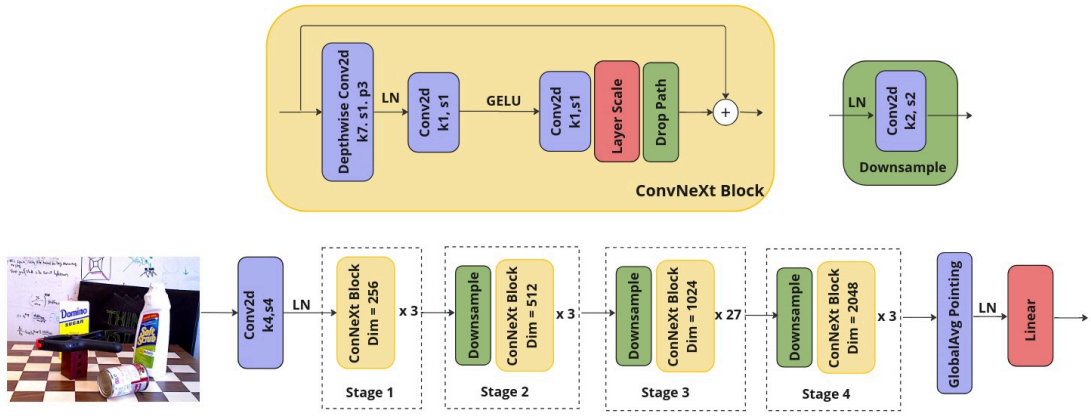


Figure 4.1: Four stages of the ConvNeXt network. The letter **k** followed by a number indicates the kernel size and **s** the stride size.

BERT (Devlin et al. 2018).

Figure 4.1 shows the four stages of the ConvNeXt network that we use as a backbone to extract the features. Similar to Swin Transformers, ConvNeXt uses a downsampling layer between stages. The first downsampling layer uses a kernel size of 4×4 with stride 4 to form the Patchify layer (larger kernel size and non-overlapping convolution). The remaining spatial downsampling blocks use a 2×2 kernel with stride 2. Additionally, the network uses Layer Normalization (LN) instead of Batch normalization in each one of the residual blocks. Each four stages of the ConvNeXt-based network (shown in Figure 4.1) provides output features at different scales.

The experiments in Bardes et al. (2022) show that the largest version of the backbone (ConvNext-XL) achieves higher mean IoU scores. For this reason we use the ConvNext-XL, which has the number of the blocks at each stage equal to [3, 3, 27, 3], and the feature dimension (dim) of each stage equal to [256, 512, 1024, 2048]. We also use the same networks for the additional VICRegL Blocks. The local projector of the VICRegL architecture has dimensions [2048, 512, 512, 512] and the global expander is a 3-layers fully-connected network with dimensions [2048, 8192, 8192, 8192].

The VICRegL criterion uses only the last feature map values to calculate its local and global criterion.

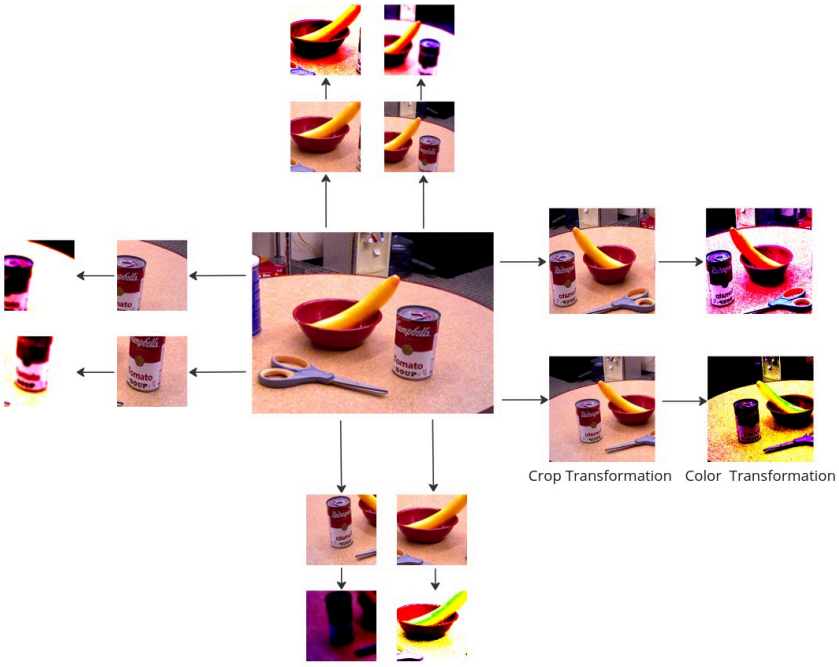


Figure 4.2: From one frame of the YCB Video dataset we obtain 8 views of the same image with different crops and augmentations applied to each one of them. The original image frame is at the center and it has the size of 480, 640 while the views have the size from 224×224 (views from the right) and 96×96 (left, up and bottom positions).

4.2 Single-Frame Training

The first approach considers the case in which all correspondences between views are from one single-frame from the YCB-Video dataset (see Section 2.3 for the review on VICRegL). Figure 4.2 shows the 8 different views from the same seed frame. For this case, the correspondences are between crops from the same image. This is the simplest version and it is also the one used in the VICRegL original paper (Bardes et al. 2022).

The crop transformation returns a crop of random size (values from 0.08 to 1.0) of the original size and a random aspect ratio (from 3/4 to 4/3) of the original aspect ratio. Each crop is finally resized to a given size: 224×224 or 96×96. There are 8 crops in total: 2 larger ones and 6 crops with the small size, and the cropping transformation is shown in Figure 4.2. The horizontal flipping of the crop is randomly selected with a probability of 50%.

In addition to the cropping and random flipping, there is also a random color distortion applied to each, such as color jittering, Gaussian blurring, solarization

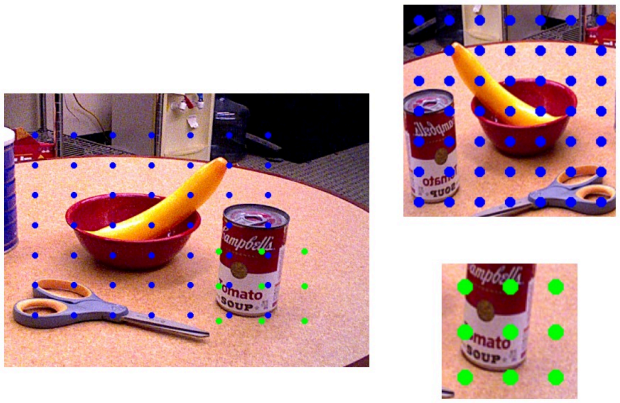


Figure 4.3: Each one of the views is associated with a grid of size 7×7 (in blue) and 3×3 (in green) for the large and small views, respectively. Each point of this grid is used to associate square areas in the feature maps that have a smaller distance in the seed image.

and grayscale transformation. All the images are normalized using the mean and standard deviation values from ImageNet.

The backbone network generates a 7×7 grid of features for the large views and 3×3 for the small views. The location of each of these points in the grid is associated with its corresponding location in the seed image, as it is shown in Figure 4.3. These grid points are used for the distance-based local loss: the K nearest points in the seed image from different views select the areas in the feature maps that are used in the calculation of the VICReg criterion.

4.3 Paired Synthetic and Real Images

Synthetic data has been used as a source of data augmentation to avoid overfitting of a particular dataset distribution by images that provide to the network different lighting changes, camera variations, and backgrounds. Tremblay et al. (2018) showed, for instance, that the model DOPE for 6D pose estimation when trained only with synthetic data can achieve state of-the-art performance compared with a network trained on real data. Therefore, motivated by this, we use synthetic data as a data augmentation for the training of the backbone.

To generate different views we use two images as seeds. Such data augmentation approach is called paired because we use one real image and one synthetic image that share the same objects with identical poses in the image.

One of the images corresponds to an image frame from the YCB-Video dataset.

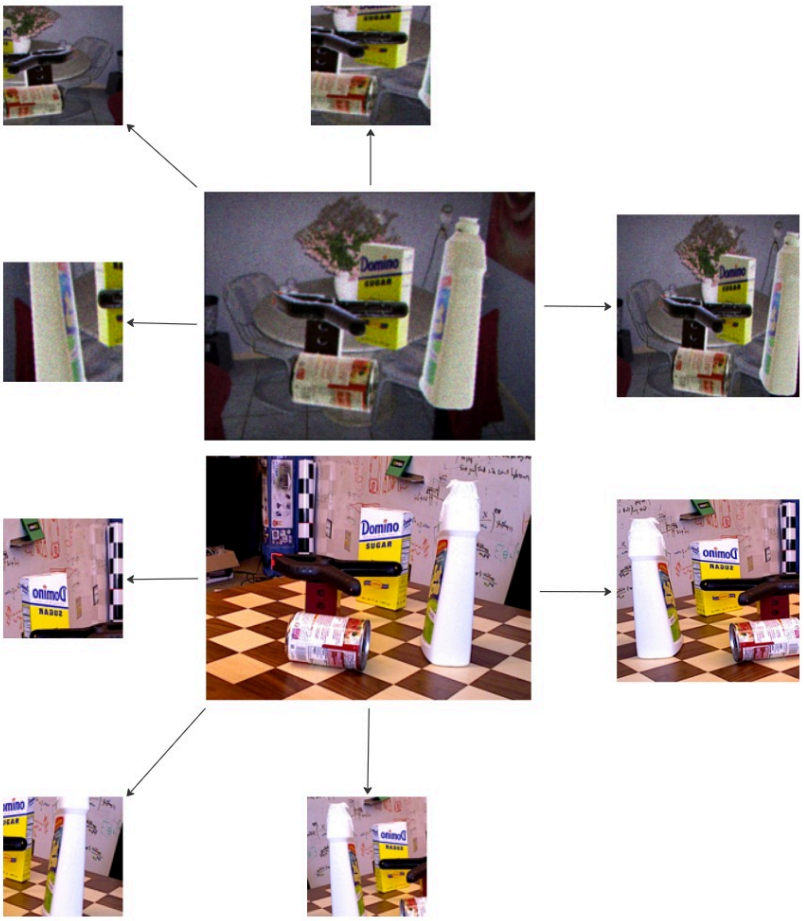


Figure 4.4: Paired approach. The two images at the center are seeds in which each view is generated from. The image at the top is the synthetic image rendered by the Stillleben framework and the one below is the real image from the YCB-Video dataset. The arrows show the random crops obtained from each one of the paired images. At the right are the crops with size 224 and at the other directions, the ones with size 96.

In order to obtain the paired image, this approach requires pose annotation. Each object in the scene and its corresponding 6D poses are provided to the Stillleben framework (Schwarz and Behnke 2020), which renders each object in an Image-based lighting (IBL) environment that provides diverse lighting and background options. Figure 4.4 shows an example of two paired images. The total number of views stays the same with the difference that half of the views are from the real image and the other half are from the synthetic image.

Although the objects are in the same pose in the image, they have a different appearance and background. Our goal is to provide to the network an effective way for learning to localize an object without relying on spurious correlations with the background. Furthermore, matching feature vectors with respect to their distance in the pixel space will create a comparison between features that describe the objects in a more generalized manner.

4.4 Multiple Arrangements

The method VICRegL explores the similarity between the features that correspond to pixels that are close to each other in the image (adjacent positions in the image). In the previous Section we investigated the correspondences between views generated from scenes in which the objects are in the same configuration. As an extension, we explore the correspondences between objects that are in different poses. In order to obtain objects in different arrangements we use synthetic images generated by the Stillleben framework (Schwarz and Behnke 2020).

For a frame of the YCB-Video dataset, as a way to ensure that each object in the scene is visible, we generate synthetic images rendering only one single object per image in a different pose. Therefore, we guarantee the objects are not occluded. In our case, we use the 6D pose annotation to calculate the correspondences between every view and the seed image, as it is shown in Figure 4.5. As a result, the model will bring closer features related to the same position of each object, however, from different arrangements in the image, in addition to different backgrounds and appearance.

Figure 4.5 shows the synthetic images generated by Stillleben and the correspondences between each view of these images with the seed image. As it is shown, since the 6D poses are arbitrary, it happens that some synthetic images do not have any correspondences in the original image and therefore, are not used as a view for the model. For the same reason, not all the points in the grid of each view have correspondences in the image. Therefore, only the positions in the synthetic images which have correspondence in the seed image are taken into account.

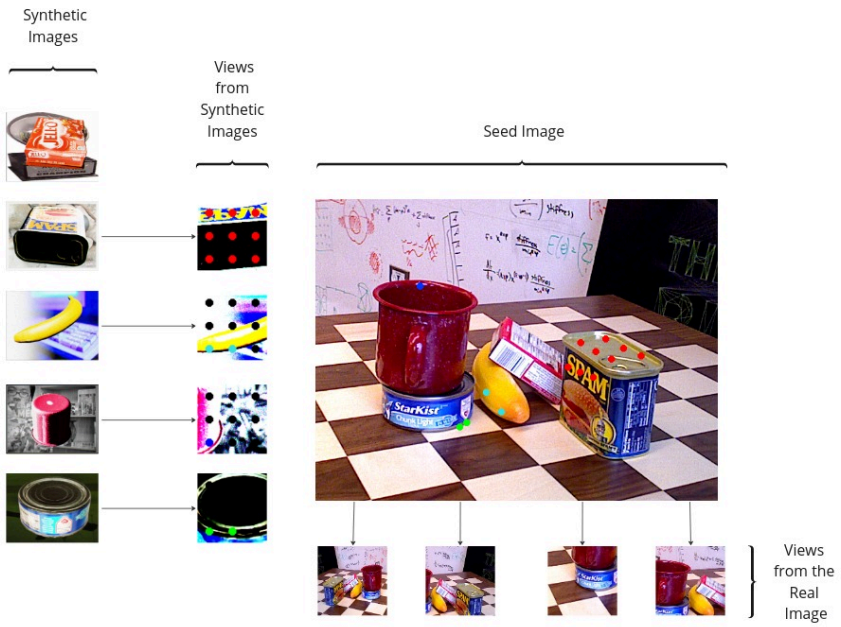


Figure 4.5: The seed image is a frame of the YCB-Video dataset. For each one of the objects, Stillleben generates one synthetic image in a different arrangement, as it is shown in the left column of the plot. The arrows indicate the image source of every view. To illustrate the correspondences between the synthetic images and the seed, we use colored points. Points of the same color indicate correspondences. Black points represent points that do not have a correspondence in the seed image.

As a choice of implementation, half of the smaller views are generated from the synthetic images, while the remaining ones are transformations of the YCB-Video frame used as a seed.

To further explore the configuration of multiple arrangements, we consider the multiple arrangements between different frames of the YCB-Video dataset.

4.5 Leveraging Pre-Segmentation for Correspondences

For the following methods we make use of the SAM annotations to apply it in two different cases:

- As a base for matching vectors in which we apply the VICRegL criterion,
- To calculate the mean and the variance for each one of the segments inside an image. Here we apply one idea for the variance similar to VICReg, with

the difference that we compare the variance between region on the feature maps that belong to different SAM segments. This allows us to apply ideas from VICReg on a segment-level rather than pixel-level.

4.5.1 SAM Annotations as Feature Matching Criterion

In Section 2.4 it was shown that the SAM model allows zero-shot generalization for new datasets and tasks given a specific prompt. Our goal is to use the SAM model to produce masks for the YCB-Video dataset. The SAM model generates a grid of point prompts over the image, then filters low quality and duplicate masks. The default settings are chosen for SAM with a ViT-H backbone, as it is specified in Kirillov (2023a).

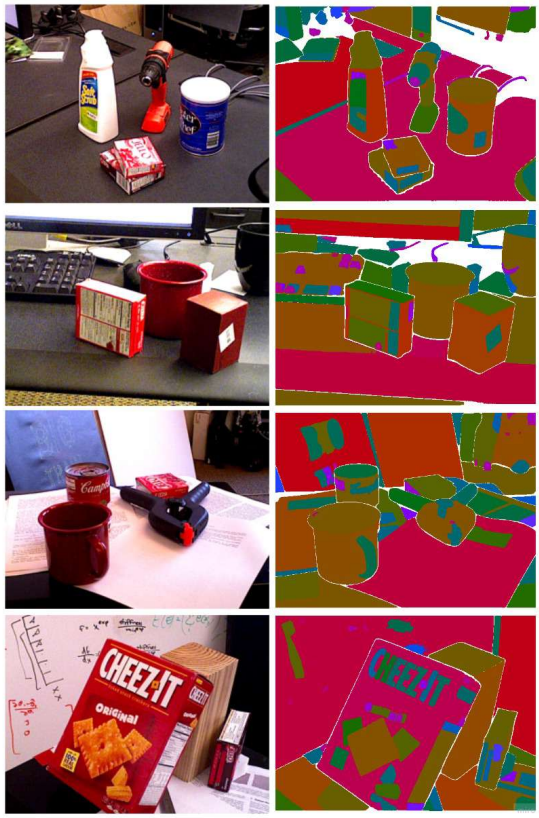


Figure 4.6: SAM masks obtained from the YCB-Video dataset.

Considering occlusions may occur, we select a minimum area of the mask equal to 1000 pixels. Post-processing is applied to remove disconnected regions and holes in masks with an area smaller than this value. The number of points to be sampled along each side of the image is 24. Figure 4.6 shows the resulting segmentation map obtained from the SAM model considering these mask generation options.

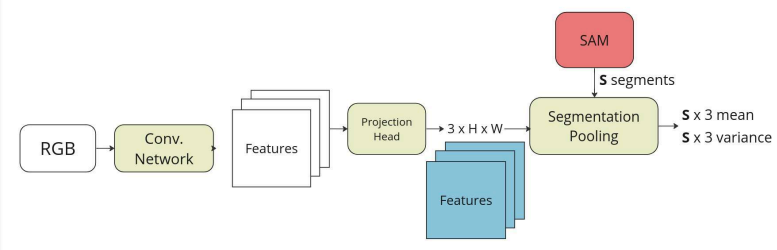


Figure 4.7: Structure of the SAM Segmentation Pooling Block.

As it is mentioned in the original paper, there are multiple masks for one single prompt. The masks are sorted by size (descending) and then drawn on top of each other. That means the smaller masks overwrite larger masks. In this way, we obtain only one pseudo label per pixel in the SAM's segmentation maps. We called it a pseudo label in the view of the fact that the model predicts the object masks, but it does not associate it to a label related to the domain of the application. As can be seen in Figure 4.6, by human rating the mask quality is high because it can segment even small parts, such as the written letters and drawings inside some objects, with well defined boundaries. We use such masks as a matching basis to apply the VICReg criterion. In the same way we applied VICReg to pairs that are a good match based on their distance in the seed image or in the feature maps, we apply it to pairs that have the same SAM label in different views. The SAM-based matching loss function is given by:

$$\mathcal{L}_{\text{SAM}}(z, z') = \sum_{p \in P} \ell(z_p, z'_p)\,, \tag{4.1}$$

if and only if $p^{\text{SAM}}_{\text{label}} = p'^{\text{SAM}}_{\text{label}}$, i.e. the points share the same pseudo SAM label, where the sum is over coordinates p in $P = \{(h, w) | (h, w) \in [1, \ldots, H] \times [1, \ldots, W]\}$. Similarly, we select only a γ amount of pairs, with the difference that the points are equally considered and randomly selected.

According to Kirillov (2023b), the image encoder takes ∼0.15 seconds on an NVIDIA A100 GPU and the prompt encoder and mask decoder take ∼50 ms on CPU in the browser using multithreaded SIMD execution. Considering the inference time, in order to have faster training, we obtained SAM Masks for the whole YCB-Video and saved its results on disk. It would be impractical to use the SAM for real-time applications, such as processing a real-time video stream.

4.5.2 Pooling Features using SAM Mask

One different way to use the SAM annotations is to pool together features that belong to the same region in the mask. In Hénaff et al. (2021), for instance, the feature vectors coming from the same region in the mask are jointly pooled and the resulting vectors are contrasted between each other with a contrastive loss function. This allows spatial vectors far away in the original image to be pooled together if they belong to the same object. Following the same idea, we use SAM masks obtained to pool together regions in the feature maps. The general idea is presented on the diagram shown in Figure 4.7.

We use the same nomenclature of Section 2.3, where the projection head is $h_\phi^l : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{D\times H\times W}$. In the same way the local projector from VICRegL (Bardes et al. 2022), the spatial information is preserved. We choose the dimension $D = 3$ to enable the visualization of the feature maps as a colored image. Using the S masks provided by the SAM annotations for the input image, we compute the mean and the squared deviation of the features for each segment.

For each one of 3 channels, the mean is calculated for the points that are from the same mask. Therefore, for each one of the S segments, we will have one mean vector

$$\mu_{S_i} = \frac{\sum_{p\in S_i} z_p}{n_{S_i}}\,, \tag{4.2}$$

where n_{S_i} is the number of points in S_i. The squared deviation is

$$v_{S_i} = \sum_{p\in S_i} (z_p - \mu_{S_i})^2\,. \tag{4.3}$$

We want to keep the squared deviation of each segment low. Therefore, we have the invariance loss in order to enforce for each segment S_i that its features z_p, where $p \in S_i$, have the same value. The invariance loss is calculated as

$$\mathcal{L}_{\text{Inv}} = \frac{\sum_{S_i\in S} v_{S_i}}{\sum_{S_i\in S} n_{S_i}}\,. \tag{4.4}$$

Between segments we want to enforce the feature distances to have a minimum value, similarly to what is done in VICReg. Instead of applying the VICReg between the different images in the batch, we apply a similar loss between different SAM segments. Accordingly, we define the squared deviation loss as

$$\mathcal{L}_V = \frac{\sum_{i\neq j} \max(0, \beta - |\mu_{S_i} - \mu_{S_j}|)}{S(S-1)/2}\,, \tag{4.5}$$

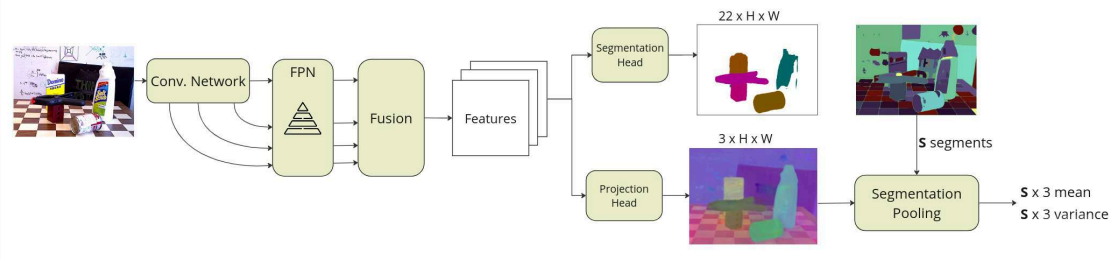


Figure 4.8: FPN and SAM pooling blocks applied together.

where we choose $\beta = 0.5$ and the sum is over all $S(S-1)/2$ different elements coming from the subtraction $|\mu_{S_i} - \mu_{S_j}|$ for S_i, $S_j \in S$ and $i \neq j$.

For each channel, the SAM Regularization loss is calculated as

$$\mathcal{L}_{\text{SAM Reg}} = \mathcal{L}_{\text{Inv}} + \mathcal{L}_V \,, \tag{4.6}$$

where $\mathcal{L}_{\text{Inv}}$ and $\mathcal{L}_V$ are given by Equations (4.4) and (4.5), respectively. The total loss is the mean of $\mathcal{L}_{\text{SAM Reg}}$ over the D channels

$$\bar{\mathcal{L}}_{\text{SAM Reg}} = \sum_{D} \mathcal{L}_{\text{SAM Reg}} \,. \tag{4.7}$$

4.6 Feature Pyramid Network

Unrelated to our correspondence efforts, whe noticed that ConvNeXt suffers from low resolution at the last stage, which is detrimental to segmentation performance. FPN (T.-Y. Lin et al. 2017) is a method for creating high-resolution feature maps and involves creating a pyramid of feature maps at different spatial resolutions, each capturing information at a specific scale. The architecture takes advantage of a pyramidal shape of a CNN feature hierarchy while creating a feature pyramid that combines low-resolution, semantically strong features with high-resolution, semantically weak features via a top-down pathway and lateral connections.

Our idea is to use of the feature maps from each stage of the ConvNeXt backbone as the input for a FPN. Figure 4.8 shows the diagram of the combined approaches: FPN and SAM pooling. The feature maps obtained from the FPN are interpolated to have the same size of the image and concatenated in the Fusion Block.

In this case, the total loss will be a sum over the semantic segmentation loss and $\mathcal{L}_{\text{SAM Reg}}$, defined in equation (4.6)

$$\mathcal{L}_{\text{FPN, SAM}} = \mathcal{L}_{\text{Seg}} + 0.05\bar{\mathcal{L}}_{\text{SAM Reg}} \,, \tag{4.8}$$

where $\bar{\mathcal{L}}_{\text{SAM Reg}}$ is the mean of $\mathcal{L}_{\text{SAM Reg}}$.

4.7 Experimental Results

In this Section we will present the results of each method developed in this thesis. For the first two cases below we use the ConvNeXt-XL backbone pre-trained on the 1000-class unlabelled ImageNet dataset and train it on 450k images, subdivided into 100 epochs of 1500 images. The segmentation performance is evaluated on a test set of annotated real images by calculating the mean Intersection over Union (IoU) over all classes present in the dataset.

When we use the architecture provided by the VICRegL paper (Bardes et al. 2022), we consider the contributions of each term of the VICReg loss are set to 25, 25, and 1 for the variance, invariance and covariance, respectively. For all simulations, we use the AdamW optimizer (Loshchilov and Hutter 2017), a weight decay of 10^{-6}, a batch size of 3 and a learning rate of 0.001. The learning rate follows a cosine decay schedule, starting from 0 with 10 warm-up epochs and with a final value of 0.00001. The number of selected best matches are set to 20 for feature maps from large views and 4 for feature maps from small views. We also experiment the training using the values 40, and 8, respectively.

First we present some qualitative results related to the matches of the VICRegL approach. Then, we present mean IoU for each one of the methods above, comparing the performance between them. The presented values for the mean IoU are the average between the last 20 epochs.

4.7.1 Visualization of the Best Matches by VICRegL

We provide in Figure 4.9 a visualization of the pairs of matched feature vectors based on the distance in the seed image (distance-based), on the top, or in the distance in the feature maps (feature-based), on the bottom.

As we described in Section 4.1, the backbone network generates a square grid that divides the features in N spaces. Considering that the feature map representations have the same dimension as the image, we show in the seed image the best matching locations in which we apply the VICRegL criterion. The circular points represent only the center of each region of the feature maps that are used for the local loss.

As we may see in the feature-based matching, the semantic content at a position in the image is coherent for some cases, showing that matching pools feature vectors have similar backgrounds and texture. For instance, there are points that are associated to same color in the image, like for instance, the white table or the

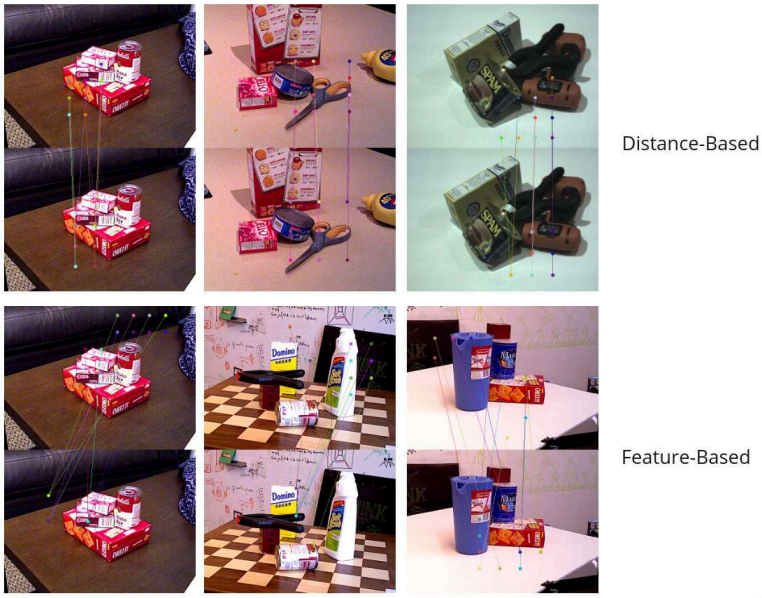


Figure 4.9: Visualization of 10 pairs of distance-based matching points (top images) and feature-based matching points (bottom images). Same colors for the lines represent the matchings.

black sofa. Therefore, the feature-based matchings associate points that are far from each other represent related concepts. However, some matching seems not to be relevant or to make a logical sense. A reasonable explanation for these cases is that the receptive field of these feature vectors is larger than just the square grid area represented in the figures. Convolution networks cover the entire image and not only the matched areas.

The distance-base matchings are intuitive, as we may notice closest points in the image are grouping in pairs.

4.7.2 Performance for the Segmentation Task

Using all the described methods we can divide the approaches into 3 main divisions according to the model and the way the total loss is calculated:

Backbone and Linear Head trained separately: In this case the loss is given by:

$$\mathcal{L}_{\text{backbone}} = \rho_{\text{VICRegL}} \mathcal{L}_{\text{VICRegL}} \,, \tag{4.9}$$

where $\mathcal{L}_{\text{VICRegL}}$ is given by equation (2.10). The coefficient ρ_{VICRegL} is the contribution of the loss.

When training the linear head, we froze the backbone weights and consider only

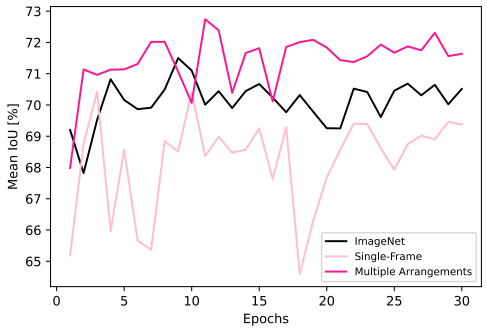


Figure 4.10: Comparison between the three approaches when we train the linear head using real images. The curves represent the approaches used to train the backbone. For the black curve we use the backbone pretained on ImageNet. For the light pink and dark pink curves, we use the Single-Frame and Multiple Arrangements approaches, respectively.

the semantic segmentation loss $\mathcal{L}_{\text{Seg}}$. Table 4.1 shows the results when the network was trained in this manner.

Between all the methods, for the real images, the Multiple Arrangements approach performs better. We may see in Figure 4.10 the comparison between the segmentation maps obtained by the features representations from 3 different training methods. The black curve represents the case when we use only the pre-trained backbone on ImageNet. The two pink curves represent the linear head using features obtained by the Single-Frame Approach and the Multiple Arrangements approach. However, when we increase the number of views that are generated from synthetic images, keeping the remaining views in the same amount (2 large and 4 small views from one real-frame image), there is no improvement in the learned features for semantic segmentation.

We may notice that using the YCB-Video real images to train the backbone worsen the performance of the segmentation task. When we use the Multiple Arrangements from more than one real-frame as a seed, the value is even inferior when we use only one image as a seed.

We associate the improvement to the data augmentation provided by the synthetic images when using it together with the real ones, as we may notice that the Paired Synthetic and Real Images Approach also gave superior results. However, when we use only synthetic images to perform the training, the backbone features modifications with training do not improve the results for the semantic segmentation task.

Backbone and Linear Head trained together: When we trained both back-

Table 4.1: Backbone Training Method – Mean IoU (%)

Backbone Training Approach	Linear Head Trained on Real Images	Linear Head Trained on Synthetic Images
Pre-trained on ImageNet[1]	70.24	**62.33**
Single-Frame[2]	68.94	42.40
Paired Synthetic and Real Images	70.45	62.18
Multiple Arrangements from syn and real images	**71.71**	61.64
Multiple Arrangements from syn and real images[3]	71.64	-
Multiple arrangements from real images	68.07	-

[1] We use only the pre-trained weights.
[2] In this case, the backbone is also trained using real images (left column) or synthetic images (right column).
[3] We use more views from synthetic seed images. In this case, we use 21 small views from the synthetic images.

bone and linear head together, we consider the total loss to be

$$\mathcal{L}_{\text{backbone}} = \rho_{\text{VICRegL}}\mathcal{L}_{\text{VICRegL}} + \rho_{\text{SAM}}\mathcal{L}_{\text{SAM}} + \rho_{\text{Seg}}\mathcal{L}_{\text{Seg}}\,, \tag{4.10}$$

where $\mathcal{L}_{\text{VICRegL}}$ is given by the equation (2.10), $\mathcal{L}_{\text{SAM}}$ is given by the equation (4.1) and we fixed $\rho_{\text{Seg}} = 1$.

We consider the supervised methods when we make use of the real ground truth labels for the semantic segmentation masks, and when we do not use any annotation from real data. In Table 4.2 we show the results when we use real images and its masks annotations for the $\mathcal{L}_{\text{Seg}}$ calculation. In this case, the same images that generate the views are also used in the semantic segmentation loss. We select specific values for the coefficients based on different trials we performed, i.e. these values return the highest scores for the mean IoU for the 20 trials we performed. The first line shows the result when we consider only the semantic segmentation loss for the training. Compared with the result when we use only the segmentation loss, the worst result improved the mean IoU by 0.8%, whereas the best one improved by 2.67%.

The training performance is sensitive to the hyperparameters ρ_{VICRegL}, ρ_{SAM} and the number of matches. Indeed, there are other hyperparameters related to the VICRegL loss and to the training, (such as α, weight decay, and learning rate) that may alter the results.

Motivated by the problem of the annotation bottleneck, we perform the training

Table 4.2: Training performed using and real images for the case of Single-Frame approachs and for all methods, the ground truth labels from the real images.

Backbone Training Approach	ρ_{VICRegL}	ρ_{SAM}	number of matches	Mean IoU (%)
Single-Frame	0.000	0.000	[20,4]	74.45
Single-Frame	0.014	0.000	[20,4]	75.08
Multiple Arrangements	0.014	0.014	[20,4]	75.43
Single-Frame	0.014	0.014	[20,4]	75.40
Single-Frame + SAM labels	0.000	0.140	[20,4]	**76.44**
Single-Frame + SAM labels	0.000	0.140	[40,8]	75.10

Table 4.3: No ground truth labels available.

Backbone Training Approach	ρ_{VICRegL}	ρ_{SAM}	number of matches	Mean IoU (%)
Single-Frame	0.000	0.000	0	57.72
Multiple Arrangements	0.014	0.000	[20,4]	44.08
Single-Frame synthetic image	0.000	0.014	[20,4]	57.92
Single-Frame Unpaired Images $\alpha = 0$	0.014	0.014	[40,8]	62.49
Single-Frame Unpaired Images	0.000	0.014	[40,8]	**68.44**
Single-Frame Unpaired Images	0.000	0.140	[40,8]	63.21
Single-Frame Unpaired Images	0.000	1.400	[20,4]	51.25

using only the labels from the synthetic images. The results when we do not make use of real annotations are in Table 4.3. We call Single-Frame Unpaired Images when we use the real images to perform the training of the backbone in a self-supervised manner, but we use a different synthetic image for the $\mathcal{L}_{\text{Seg}}$ calculation. When we do not specify that the approach is unpaired, it means that we use synthetic image to generate the views and the semantic segmentation loss. The first line shows the result when we use only $\mathcal{L}_{\text{Seg}}$ loss. Not all the approaches improve the segmentation loss. The multiple arrangements loss and the Single-Frame with $\rho_{\text{SAM}} = 1.4$ worsen the segmentation results. However, when we consider the unpaired approach we obtain a result that is 18.57% better than when we do not use the loss $\mathcal{L}_{\text{Seg}}$.

FPN model in conjunction with the SAM Pooling Block: For this approach we use the loss provided by the equation (4.8). For all simulations, we use the

Table 4.4: FPN + SAM pooling. Ground Truth Real Labels or Synthetic Labels are used in the images for training the corresponding network if the number in the corresponding column is 1. The network is trained with images without one of the labels if the number in the column is 0.

	Ground Truth Real Labels	Synthetic Labels	Mean IoU (%) (between 80-100 ep)
SAM Pooling Block + FPN	0	1	**84.84**
FPN	0	1	80.29
FPN	1	0	83.52
SAM Pooling Block	0	1	66.81

Table 4.5: Comparison between the methods that are trained without real labeled data.

Method	Mean IoU (%)
ConvNeX Backbone + Linear Head	62.33
Single-Frame Umpaired Images	68.44
SAM Pooling Block + FPN	**84.84**

Adam optimizer, no weight decay, a batch size of 1 and a learning rate of 0.00001. The results are shown in Table 4.4. We evaluate the model with and without the FPN block for synthetic images.

Using only the SAM Pooling Block we obtained an performance for the synthetic images inferior to the one obtained in the previous Section. When we use only the FPN, we obtain best results for the mean IoU, with the difference of only 4%, in the performance of the model when we use real images and synthetic images, showing itself as a potential solution for the annotation bottleneck problem.

Finally, when we apply the SAM Pooling Block and the FPN and trained the model on real data without labels and synthetic data with labels, we obtain the best performance over all methods: 84.84%. Among all the approaches we investigated, this one was the most promising, providing the highest score without relying on annotation data.

Comparison between the methods that are trained without real labeled data

We summarize the methods that are trained without labeled real data in Table 4.5. The method using the SAM Pooling Block + FPN also performs better than the other methods that make use of real images. It even outperforms the best method trained using real labels, which was also based on the FPN architecture. While it is clear that the SAM Pooling Block improves results when used in combination with FPN, further research on its application is indicated.

5 Conclusions and Future Work

In this thesis we explored different approaches to improve the semantic segmentation results when considering unlabeled data.

In the first part of the thesis described in Chapter 3, we used synthetic images focusing on decreasing the domain gap during training while keeping the semantic segmentation loss as part of the objective. We tried to align the distribution of the feature maps obtained from the RefineNet model when we used synthetic and real images for the training. We evaluated 2 different methods: distance-based approach and adversarial approach.

When using paired images, for the l^2 distance-based approach we had in the best case an improvement of 11% for the mean IoU over the test set compared when we trained the network only with synthetic images. The disadvantage of this approach is the need of annotation to render the objects with identical 6D poses in the image. For the paired case, when we use the adversarial approach, we obtained an improvement of 19%.

For the unpaired case, we do not need any annotations. However, contrary to our expectations, when we applied the adversarial approach to unpaired configuration the network performs worse than when we use only synthetic images. We also face some difficulties to train our "RefineNet-GAN'' model related to its convergence and instability.

We conclude that for the Frame-Level Feature adaptation, when we compare the feature maps using the distance approach or the adversarial approach, we miss important local information, considering that the comparison is between the entire region of the feature maps. The comparison at a local scale is important to improve the semantic segmentation results. Even more critically, the comparison between *corresponding* regions, i.e. regions belonging to the same object, need to be identified and made use of.

Motivated by the need to use local features in our approach and by the need to use unlabeled data, we used the VICRegL approach in Chapter 4 of the thesis. Differently from Chapter 3, we used a modern network ConVNeXt-XL as a backbone for our model. We made use of synthetic images and different configurations for the correspondences between views that are used for the local VICRegL loss: we used synthetic images as data augmentation and we explored the correspondences

between images in which the objects are in different poses (Multiple arrangements approach).

For the training of the backbone and the linear head separately, when we used ground truth labels, most of the approaches improved the results in comparison when we used only the pretrained backbone or when we used only one image as seed for the different views. However, there were no improvements when we trained the linear head using synthetic images.

At the same time we were working on this project, the SAM model appeared as a solution to generate pseudo-masks to any dataset. Although when we used SAM masks there were no real labels associated with the domain of our application, we used the location of the masks to match features that belong to the same segment, applying the VICRegL criterion to these matching regions in the feature maps. When we considered the SAM masks as a matching criterion to apply the VICReg loss, the performance of the model improved but only for certain values for the contribution of each one the loss.

Training the backbone and the linear head together, we had to tune the contribution of each one of the losses, which affected the performance of the model. The best result when we did not use any ground truth label was obtained when we used only the VICRegL criterion applied to regions that are part of the same segment in the SAM masks. Further investigations need to be performed to understand the specific aspect that each one of the losses have in the network.

Using the modifications we performed to the loss of the VICRegL model, we obtained better results than when we use the original objective function. However, for the real images, the mean IoU of 76.44% was similar to the RefineNet model obtained using only the semantic segmentation loss, 77.84%. For the synthetic images, we obtained using the Single-Frame unpaired approach the best result over all the approaches from the previous Sections: a mean IoU of 68.44%. Nevertheless, there is still a gap of 10.5% when we train the model on unlabeled data. Consequently, we may state that we decreased the domain gap when we used synthetic and real images for training, but there is still space for improvement.

Focusing on obtaining a better model performance, we developed the SAM Pooling Block, in which we considered the variance between the segments and not between different images in the batch. The main idea of this block is to keep the difference between the features inside each segment as low as possible, and between different segments above a minimum value, following the same principle as VICReg variance. When we used this block during the training of the backbone we did not obtain a considerable effect. We associated that one of the reasons the model does not perform better is the low-resolution of feature maps obtained from the backbone. Motivated by this we used a FPN to have the contribution of the

features in different scales.

The FPN applied together with the SAM Pooling Block approach, provided a considerable boost in our results: we obtained a mean IoU around 85% when we trained the model on real data without labels and synthetic data with labels. Compared to the methods developed in this thesis, this last approach generated the highest scores for real and synthetic images. Additionally, the IoU of 84% is suitable for robotic applications, making this model even more promising for different applications.

As a future work, there is still a lot to explore related to the contribution of each one of the losses for Section 4 in conjunction with the ConvNeXt-XL-FPN model Additionally, we want to evaluate this model in different datasets, considering that in this project we only used the YCB-Video.

During the months we worked on these ideas we proposed different ideas. As in many research areas, the judgment whether one method is good or not comes from the results of the experiments. Consequently, each approach took an amount of effort and time to be implemented before being assessed as a good option. In our case, it was an extensive research work until we obtained the final method that could succeed as an outstanding result. Additionally, we were using state-of-art and recent projects to guide the directions in which we thought it was worth investing time and effort on. SAM, for instance, was available only in April, 2023.

Our main motivation was due to the fact that visual domain adaptation is an interesting and promising area, considering the need to solve the annotation bottleneck problem for different applications. Although the method we created produced better results than previous works in the same area, such as the CUT-refining approach for domain adaptation (Imbusch et al. 2022), which obtained 76.3% mean IoU on YCB-Video, we are not sure which block is responsible for the improvements: if it is the ConvNeXt backbone, the FPN (which can also be applied to CUT as well), the SAM Pooling or VICRegL approach. This is a motivation for further investigation of this method.

We conclude that the thesis was a successful research experience in the field of computer vision, where the scientific method was applied and various methodologies were investigated and tested.

List of Figures

List of Tables

Bibliography

Bardes, Adrien, Jean Ponce, and Yann LeCun (2021). "Vicreg: variance-invariance-covariance regularization for self-supervised learning." In: *Arxiv preprint arxiv:2105.04906*.

– (2022). "Vicregl: self-supervised learning of local visual features." In: *Advances in neural information processing systems* 35, pp. 8799–8810.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: pre-training of deep bidirectional transformers for language understanding." In: *Arxiv preprint arxiv:1810.04805*.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative adversarial nets." In: *Advances in neural information processing systems* 27.

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

Hénaff, Olivier J, Skanda Koppula, Jean-Baptiste Alayrac, Aaron Van den Oord, Oriol Vinyals, and Joao Carreira (2021). "Efficient visual pretraining with contrastive detection." In: *Proceedings of the ieee/cvf international conference on computer vision*, pp. 10086–10096.

Hendrycks, Dan and Kevin Gimpel (2016). "Gaussian error linear units (gelus)." In: *Arxiv preprint arxiv:1606.08415*.

Imbusch, Benedikt T, Max Schwarz, and Sven Behnke (2022). "Synthetic-to-real domain adaptation using contrastive unpaired translation." In: *2022 ieee 18th international conference on automation science and engineering (case)*. IEEE, pp. 595–602.

Kirillov, Alexander (2023a). *Github segment anything*. URL: `https://github.com/facebookresearch/segment-anything/blob/main/segment_anything/automatic_mask_generator.py`.

– (2023b). *Segment anything research*. URL: `https://segment-anything.com/`.

Kirillov, Alexander, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. (2023). "Segment anything." In: *Arxiv preprint arxiv:2304.02643*.

LeCun, Yann (2022). "A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27." In: *Open review* 62.

Lin, Guosheng, Anton Milan, Chunhua Shen, and Ian Reid (2017). "Refinenet: multi-path refinement networks for high-resolution semantic segmentation." In: *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 1925–1934.

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

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). "Roberta: a robustly optimized bert pretraining approach." In: *Arxiv preprint arxiv:1907.11692*.

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

Liu, Zhuang, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie (2022). "A convnet for the 2020s." In: *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pp. 11976–11986.

Loshchilov, Ilya and Frank Hutter (Nov. 2017). "Decoupled Weight Decay Regularization." In: *Arxiv e-prints*, arXiv:1711.05101, arXiv:1711.05101. arXiv: `1711.05101 [cs.LG]`.

Mao, Xudong, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley (2017). "Least squares generative adversarial networks." In: *Proceedings of the ieee international conference on computer vision*, pp. 2794–2802.

OpenAI (Mar. 2023). "GPT-4 Technical Report." In: *Arxiv e-prints*, arXiv:2303.08774, arXiv:2303.08774. arXiv: `2303.08774 [cs.CL]`.

Park, Taesung, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu (2020). "Contrastive learning for unpaired image-to-image translation." In: *European conference on computer vision*. Springer, pp. 319–345.

Schwarz, Max and Sven Behnke (2020). "Stillleben: realistic scene synthesis for deep learning in robotics." In: *2020 ieee international conference on robotics and automation (icra)*. IEEE, pp. 10502–10508.

Tremblay, Jonathan, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield (2018). "Deep object pose estimation for semantic robotic grasping of household objects." In: *Arxiv preprint arxiv:1809.10790*.

Xiang, Yu, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox (Nov. 2017). "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes." In: *Arxiv e-prints*, arXiv:1711.00199, arXiv:1711.00199. arXiv: `1711.00199 [cs.CV]`.