SPRING

Deliverable D2.7: Learning scene representations in relevant environments

Due Date: 31/07/2023

Main Author: CVUT

Contributors: UNITN

Dissemination: Public Deliverable

This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

DOCUMENT FACTSHEET

Deliverable	D2.7: Learning scene representations in relevant environments
Responsible Partner	CVUT
Work Package	WP2: Environment Mapping, Self-localisation and Simulation
Task	T2.4: Scene Representations for Localization and Mapping
Version & Date	31/07/2023
Dissemination	Public Deliverable

CONTRIBUTORS AND HISTORY

Version	**Editor**	**Date**	**Change Log**
1	CVUT	24/07/2023	First Draft
2	CVUT	31/07/2023	Final version

APPROVALS

Authors/editors	Martin Zderadicka, Nikita Sokovnin, Rakshith Madhavan, Michal Polic, Tomas Pajdla
Task Leader	CVUT
WP Leader	CVUT

Contents

Executive Summary

This deliverable aims to develop models, representations, and learning algorithms for creating and updating a map of the environment. This directly serves StO-1 and SpO-1.3. The outcome comprises the Visual Robot Localization module, enabling robot self-localization through vision T2.1; the Visual Semantics module, facilitating language-driven robot self-localization T2.3; and the Online Map Update module, tasked with updating the current environment map with semantic information T2.4. In terms of Key Performance Indicators, the deliverable addresses and improves the results reached concerning the KPI-StO-1.2 (Multiple object tracking accuracy) and KPI-StO-1.7 (Object recognition mean average precision) by Openscene3D segmentation of the environment. Further, the deliverable addresses the updates reached in the scope of the KPI-StO-1.8 (Image retrieval precision and recall) by employing the semantic information of the environment, KPI-StO-1.9 (Localization accuracy), and KPI-StO-1.10 (3D structure coverage by the Visual/3D map) by using simultaneously forward and backward facing camera for the localization. The current milestone reached aligns with MS5: Validation of the intermediate software architecture, completed in controlled environments (laboratory) and initiated in relevant domains (hospital), as expected by the grant agreement. We will continue finetuning and evaluating the frameworks to achieve MS6 and MS7, i.e., by systematically validating the overall intermediate software architecture in relevant environments and quantifying usability and acceptance using standard protocols.

1 Contributions

1.1 Introduction

For assistive robots, it's crucial that they are capable of understanding and discussing their environment. This is pivotal for task completion and facilitates appropriate responses and conversations that are contextually accurate and consider both the ongoing dialogues and their location. The term 'Visual language grounding' denotes the association between linguistic expressions and their visual counterparts, such as objects, their characteristics, and their interrelationships.

Creating a visually grounded dialogue for artificial systems presents several hurdles: 1) The task of correlating language with visual perceptions is complex. Language is incredibly diverse, and the same expressions can hold different meanings based on context or individual interpretation. 2) Language is fluid, with word meanings shifting over time and new expressions being added. Agents working on a task can also dynamically form and negotiate meanings. Given this, it's impractical to encode all this knowledge in advance. 3) Once an artificial agent like a robot can comprehend language and relate it to the visual world, another challenge arises. How does it generate responses? How does it appropriately formulate pertinent and coherent answers given a particular scene, prior dialogue, and question?

The answer partially lies in the robot's ability to dynamically learn about new unknown objects, their properties, affordances, and relationships. As the robot navigates through its environment, it continually encounters new objects that are domain-specific and can be employed in the localization process. These continuously learned representations of the environment and its specific properties can then be utilized as priors for finding similar images for localization. This dynamic learning approach enables the robot to create a more enriched and context-specific understanding of its environment and world meaning.

This document provides an overview of software modules developed for visually-based semantic scene comprehension, with evaluations performed on the Broca dataset. The Broca dataset comprises five recordings from the ARI robot, collected in April 2022, which include images from the front and rear fisheye cameras. The dataset contains 2500 images (250 per recording, two cameras, and five recordings). It also includes two scans captured by the Matterport scanner. Section 1.2 delves into multi-view semantic localization. It explains how the front and rear-facing fish-eye cameras are used to improve localization and mapping. Section 1.3 focuses on how Large Language Models (LLMs) can extract object relationships. This part highlights the results of the recent MiniGPT4 and LLaVA models on input images from Broca. We discuss the strengths, weaknesses, and limitations of these techniques. Recognizing unknown objects gets special attention, with more extensive details appearing in Section 1.4. This section discusses multi-view open-set object classification and focuses on the evaluation of various anomaly scores. These scores help us detect new environment-specific objects, which can be useful for pre-filtering map images during localization. Section 1.5 shifts the focus to Open-Set Object Classification in 3D. It presents the latest results in scene modeling, where individual scene entities come equipped with queryable attributes, affordances, and classes using natural language. Lastly, Section 1.6 discusses Localization Using Open-Set Object Classification. Here, we show how the outputs of Section 1.5 come into play in localization. Specifically, how to use them for finding similar places in the map before running localization based on matching local feature points.

The software will be made available through the code repositories. In accordance with European Commission guidelines, the repository will be publicly accessible for a minimum of four years following the conclusion of the SPRING project. Those interested can request software access by contacting the project coordinator at spring-coord@inria.fr.

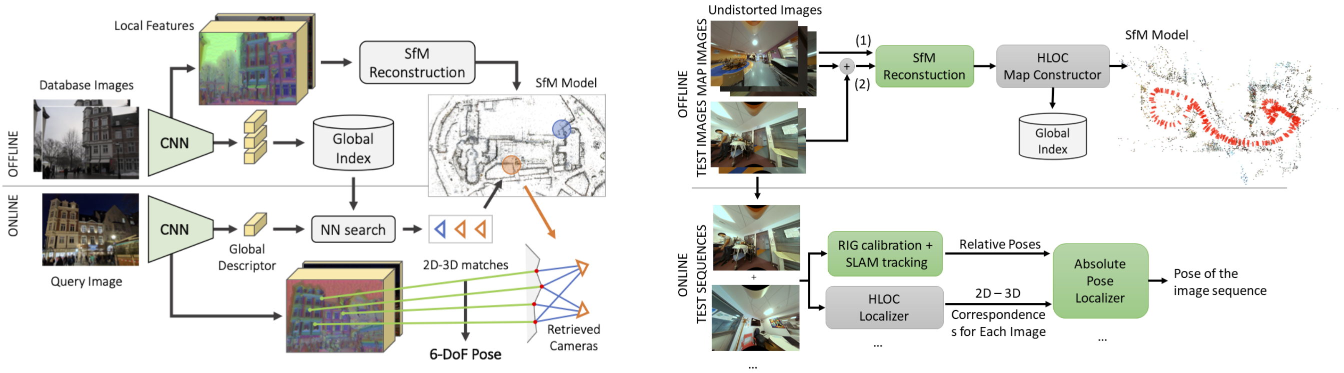


Figure 1.1: The left part of the image visualizes the Hloc architecture as presented in [14]. The offline step of the algorithm takes camera poses and images as input. The Hloc map creator detects local feature points using a selected detector, triangulates them, and calculates a global descriptor for each image. At the time of localization, the query image is processed by the same feature point detector and image descriptor. Based on the image descriptor, neighboring images from the map are retrieved. Next, 2D-3D matches between the query image and selected images from the map are utilized to run the absolute pose solver and localize the camera of the query image. On the right part, the top scheme shows the map creation process. We assume two sets of images, i.e., the map and the test images, that are used to create two reconstructions using Structure from Motion (SfM) [16]. One reconstruction consists of only the map images, while the second includes all the images. The test images are further utilized to select sequences from the recording. Each sequence uses the extrinsic camera calibration and Simultaneous Localization and Mapping (SLAM) [11] or SfM reconstructed relative poses plus the 2D-3D correspondences from the original Hloc inside the generalized absolute pose solver. The generalized absolute pose solver finds the pose of the image sequence in the map reconstruction (map coordinate system). Then, we compare the obtained poses with the poses from the reconstruction where all the images are used.

1.2 Multiview semantic localization

The ARI robots used in SPRING are equipped with two 180 degrees field of view fish-eye cameras, providing a suitable foundation for constructing global visual maps using global structure from motion pipelines such as COLMAP [16]. This led to the exploration of Hloc, an image-based localization pipeline, for global ARI localization in the relevant Broca hospital environment [14, 15]. This section discusses shortly an extension by multiview localization. The relevant codes for this section are in GitLab[1]. By using multiple images taken simultaneously or sequentially by the ARI robot, particularly the front and rear-facing fish eye cameras, we gain a broader field of view of the surroundings, aiding precise estimation of the camera pose. This approach bolsters accuracy and leverages a semantic understanding of the environment but necessitates the generalization of the absolute pose estimation algorithm. The prefiltering of candidate images in the map is discussed in sections 1.5, 1.6.

The process of measuring the accuracy of the localization consists of several steps. Initially, two sets of images - the map and the test images - are utilized to create two distinct reconstructions using SfM [16]. The first reconstruction is solely based on the map images, whereas the second includes all images. Then, test images are employed to select randomly short image sequences from the recording. Every sequence incorporates the extrinsic camera calibration, relative poses reconstructed via SLAM or SfM, and the 2D-3D correspondences from the original Hloc. This information is sent to the generalized absolute pose solver responsible for determining the pose of the image sequence within the map coordinate system. Lastly, the obtained poses are compared with the poses from the ground truth reconstruction, which includes all images. This comparison is the best measure we can get without having precise ground truth from external tracking system in Broca. It's important to note that the timing of image capture doesn't significantly impact the process. Whether we use images taken at a single point in time or a sequence of images captured over the last few seconds of recording (even when the robot is in motion) makes no considerable difference to the outcome. The reason lies in the fact that our primary requirement is the relative poses and the 2D-3D correspondences provided by SLAM and Hloc algorithms.

Drawing from statistical theory, the standard error - for instance, the error of the camera pose or the pose of the image sequence - diminishes inversely with the square root of the number of observations. This implies that using four images could double the accuracy of the results. Furthermore, it's recognized that omnidirectional cameras tend to have significantly smaller drift compared to their perspective counterparts. This is due to their ability to constrain the camera pose from all directions, as opposed to only the forward direction visible in perspective images. However,

[1] https://gitlab.inria.fr/spring/wp2_mapping_localization/hloc-mapping-localization

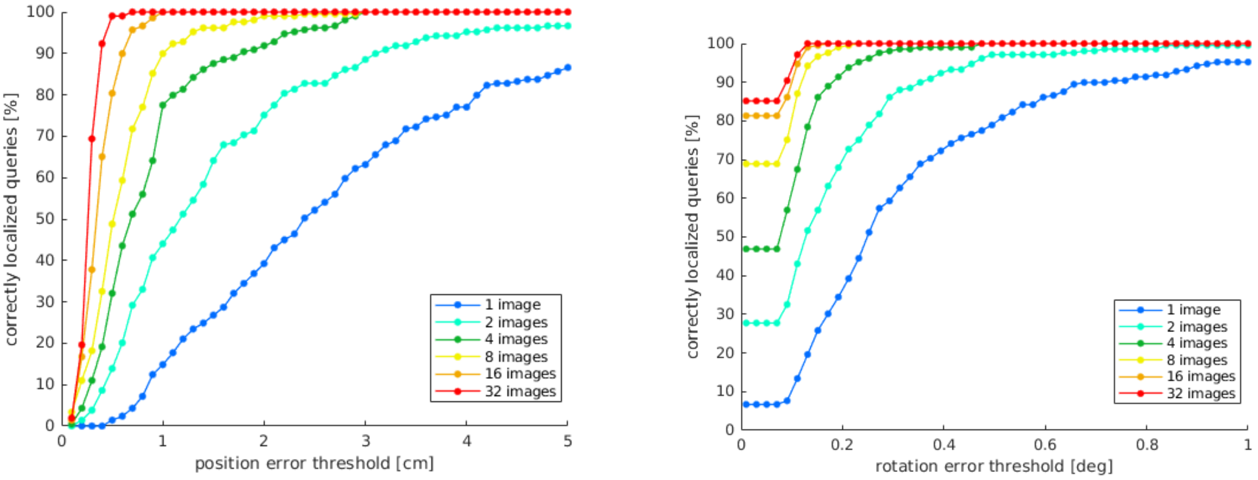


Figure 1.2: Visualization of the percentage of 209 localized queries within a given positional and rotational threshold. The image sequences are from two different recordings of the ARI robot in Broca Hospital. The first recording forms a map, and the second is used to compose the query sequences. Note that the front and rear-facing cameras were not synchronized well, and therefore, the sequences were collected based on the spatial distance between cameras. As a result, some camera poses may be less accurate because both cameras look in the same direction. Despite this, the results confirm the theoretical assumption that the accuracy increases with the addition of more observations.

these benefits are subject to certain limitations. Regarding our evaluation, the precision of the SfM reconstruction conducted with all the images truly influences the localization accuracy. Next, the accuracy of the relative poses obtained from SLAM (which does use only the front camera now) over long sequences may exhibit considerable drift that affects the camera rig's pose.

1.2.1 Experimental evaluation

The experimental setup and evaluation procedure are described in Figure 1.1. We calculate the localization map, excluding the test images, and create a reference reconstruction using all the photos. Both are then aligned using the Similarity transformation, i.e., the reference camera centers are aligned to the camera centers of map images. Subsequently, the test images are composed into sequences of different lengths [2, 4, 8, 16, 32], amounting to approximately 1k sequences in total, and localized with respect to the map. The residuals concerning the reference reconstruction are visualized as a recall curve, as shown in Figure 1.2.

1.2.2 Conclusion

This section's results address SpO-1.1, aiming to perform self-localization and tracking in cluttered and populated spaces. The related KPI-StO-1.9, requiring a localization accuracy of less than 0.5 meters, is shown to be achieved. Moreover, the increased accuracy of cameras within the environment further contributes to a more detailed 3D structure coverage. Consequently, these advancements directly contribute to reaching our target for KPI-StO-1.10, which mandates a 3D structure coverage by the Visual/3D map of over 80%.

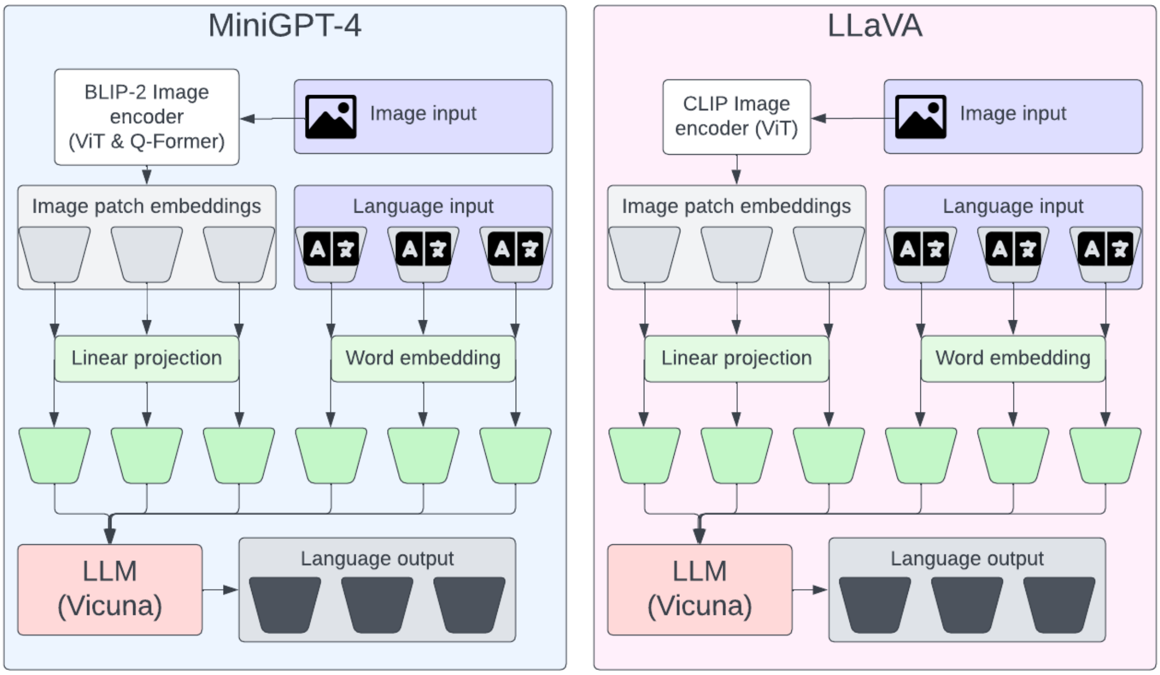


Figure 1.3: Both LLaVA and MiniGPT-4 utilize the same visual instruction tuning process. Notable differences are LLaVA's use of the Contrastive Language-Image Pre-Training (CLIP) image encoder and MiniGPT-4's use of BLiP-2, along with differing tuning datasets.

1.3 Object relationships extraction by LLM

1.3.1 The Visual instruction tuning approach

We have conducted preliminary experimentation with visually grounded large language models, namely MiniGPT-4 [19] and LLaVA [7], in tasks related to visually grounded reasoning and dialogue, aiming to enhance human-robot interaction within hospital locations. Effectively, this helps the robot understand a visual scene and facilitates meaningful, contextual dialogue based on that understanding. Such capability is essential in healthcare environments, where precise comprehension of the scene and smooth communication can be crucial in assisting patients and medical staff. The relevant codes are in GitLab repositories[2,3].

The cornerstone of the approach utilized in the two vision-language models [19], [7] lies in their innovative technique for integrating the outputs of a Vision Transformer into the inputs of a Large Language Model (LLM). The Vision Transformer serves as a pipe for transforming raw visual data into a structured, semantically rich format that the LLM can comprehend and interpret. The visualization of these model architectures is in Figure 1.3. This integration process, often called visual instruction tuning, bridges visual and language domains. Once the Vision Transformer processes the visual data, the resulting output, a sequence of image patch embeddings, is projected directly into the input space of the LLM. This effectively enables the LLM to perceive the visual information as if it is text, thereby using its semantic and syntactic understanding capabilities to interpret the visual input.

An essential part of this approach is the fine-tuning process. By adjusting the LLM's parameters based on visual instructions that correlate image properties with descriptions, the model learns better to understand the visual data in the context of language. This synergy between the Vision Transformer and LLM facilitates learning complex visual-language relationships and significantly enhances the model's ability to provide accurate responses and make insightful predictions. For our future research, we aim to explore this methodology's potential in projecting 3D scene models into the inputs of an LLM. The challenge lies in ensuring the transformed input adequately captures the spatial and relational characteristics of the original 3D data. The visual instruction tuning approach presents a compelling alternative to a scene graph-based technique. The advantages of this technique are multifold. Firstly, it can simultaneously process large amounts of visual and textual data. This allows the models to learn complex representations across visual and language domains, often leading to a richer understanding of the scene and the ability to gener-

[2] https://gitlab.inria.fr/spring/wp2_mapping_localization/MiniGPT-4
[3] https://gitlab.inria.fr/spring/wp2_mapping_localization/LLaVA

Evaluation results			
Method	Description	Relation	Usage
MiniGPT-4	0.58	0.1	0.2
MiniGPT-4 (cropped)	-	4.5	**0.7**
LLaVA	**0.64**	0.35	0.55
LLaVA (cropped)	-	**0.7**	**0.7**

Table 1.1: The table visualizes the mean score obtained by evaluating ten queries for each topic and each language model. Note that the purpose of this visualization is to show the difference between the cropped images, which directly focus on the object of interest, and the zoomed-out images, highlighting the limitations of current visual information extraction from the LLM.

ate more accurate dialogue responses. Secondly, visual instruction tuning models inherently encode spatial relations, eliminating the need for explicit graph-based representations, which, as seen in previous work, often suffer from noise, misclassifications, and a tendency to highlight obvious or irrelevant relations. The limitation of such an approach is the requirement of the finetuning dataset and hallucinations in out-of-the-domain environments.

1.3.2 Experimental evaluation

Evaluating this approach quantitatively is a non-trivial task. Current benchmarks focus predominantly on either object detection or specific visual tasks and hence, need to be revised to evaluate the complex visual relationships these models can capture. Therefore, we provide only a qualitative overview of this model's ability to interpret and respond to visual scenes.

We employed a three-pronged approach for qualitative evaluation. This methodology emphasizes questions from three topics: object description, relationships between objects, and usage of the things. The results are shown in Table 1.1. For the *description category*, the model is tasked with articulating an exhaustive and precise image description. Scoring for this task is based on a scale from 0 to 1. The answer with hallucinations (information not present in the image) and overlooked or mistaken details are marked by zero. On the opposite side, accurate and precise descriptions are evaluated as one. The *relation category* requires the model to discern and interpret relationships between the image's components. Questions in this category can include comparative reasoning between objects or referencing an object via its association with others. Similarly, the *usage category* questions the model's understanding of objects' functions and use cases within the image context. In the last two categories, scores range from 1 for a correct answer, 0.5 for an incomplete or inaccurate response, and 0 for a wrong answer. The evaluation is done by humans, which see the question, image, and model answer.

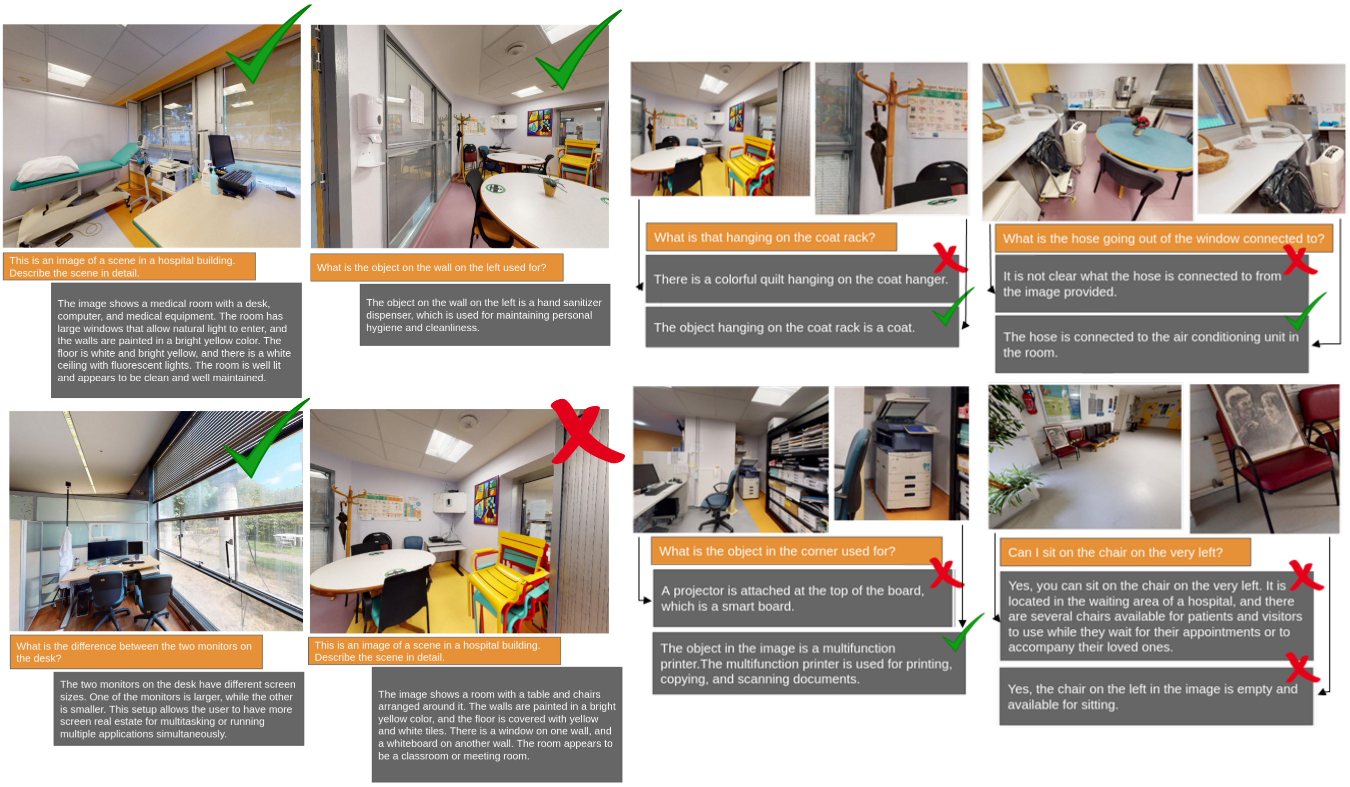
Taking into account the known limitations of the vision transformer architecture when dealing with objects that occupy a small portion of the image, we also conducted a supplementary evaluation. In these cases, the same questions for relation and usage tasks were posed using images cropped to focus solely on the relevant part of the image. Ten images were evaluated for each category to ensure a varied and balanced representation of different types of visual data. It's worth noting that while this evaluation provides valuable insights, it remains a manual and thus a somewhat subjective measure of the model's capabilities.

1.3.3 Limitations

While the approach exhibits outstanding potential, the vision-language models, as noted in the referenced papers [19], [7], have shown a propensity to hallucinate significant visual details. This refers to their tendency to generate output based on assumptions or learned patterns that do not strictly adhere to the actual content of the visual input, which could notably impair the models' usability in real-world applications. This issue is particularly problematic in scenarios where precision and accurate representation of visual details are paramount. Another hurdle relates to the substantial computational cost associated with LLM inference. The complexity of large language models demands considerable computational resources and power, which could present a significant constraint, especially in applications requiring real-time or near-real-time responses.

Navigation by LLMs

In an effort to probe the utility of these vision-language models in a more diverse set of applications, we have conducted experiments to test their proficiency in navigation tasks based on top-down views of an environment. Unfortunately, the performance in this particular use-case was largely limited. The models showed difficulties in correctly interpreting spatial relationships and movement directives from this perspective, making their outputs inadequate



This is an image of a scene in a hospital building. Describe the scene in detail.

The image shows a medical room with a desk, computer, and medical equipment. The room has large windows that allow natural light to enter, and the walls are painted in a bright yellow color. The floor is white and bright yellow, and there is a white ceiling with fluorescent lights. The room is well lit and appears to be clean and well maintained. ✓

What is the object on the wall on the left used for?

The object on the wall on the left is a hand sanitizer dispenser, which is used for maintaining personal hygiene and cleanliness. ✓

What is that hanging on the coat rack?

There is a colorful quilt hanging on the coat hanger. ✗

The object hanging on the coat rack is a coat. ✓

What is the hose going out of the window connected to?

It is not clear what the hose is connected to from the image provided. ✗

The hose is connected to the air conditioning unit in the room. ✓

What is the difference between the two monitors on the desk?

The two monitors on the desk have different screen sizes. One of the monitors is larger, while the other is smaller. This setup allows the user to have more screen real estate for multitasking or running multiple applications simultaneously. ✓

This is an image of a scene in a hospital building. Describe the scene in detail.

The image shows a room with a table and chairs arranged around it. The walls are painted in a bright yellow color, and the floor is covered with yellow and white tiles. There is a window on one wall, and a whiteboard on another wall. The room appears to be a classroom or meeting room. ✗

What is the object in the corner used for?

A projector is attached at the top of the board, which is a smart board. ✗

The object in the image is a multifunction printer.The multifunction printer is used for printing, copying, and scanning documents. ✓

Can I sit on the chair on the very left?

Yes, you can sit on the chair on the very left. It is located in the waiting area of a hospital, and there are several chairs available for patients and visitors to use while they wait for their appointments or to accompany their loved ones. ✗

Yes, the chair on the left in the image is empty and available for sitting. ✗

Figure 1.4: Examples of description, relation, and usage task outputs with varying scores. The figures show examples of good and bad scores, the effect of cropping images, and the impact of the input image dimensions.

for reliable, real-world application in navigation tasks. Consequently, while the models demonstrate strong potential in certain areas of scene understanding and visual dialogue, their use in navigation-based tasks, at least under the present state, is not sufficiently robust to be viable.

Open-set classifier by LLMs

We have also explored the possibility of leveraging these vision-language models as open set classifiers, through an approach commonly referred to as prompt engineering. This methodology involves carefully crafting textual inputs, or prompts, in order to guide the model towards generating specific, class-related outputs. Despite our diligent efforts in creating these prompts, the results yielded were varied and less than ideal. A prominent issue encountered was the tendency of the models to "hallucinate," or infer elements and details not present in the input data. This could result in unpredictable and often inconsistent textual output. This effect was particularly noticeable when adjusting the temperature parameter of the models. Typically, a higher temperature encourages more diverse output, potentially increasing the model's ability to cover a wide range of classes. However, in this instance, increasing the temperature resulted in a significant degradation of the output's consistency and reliability. The issue of hallucination and erratic output became substantially more pronounced, further hindering the utility of these models in an open set classification context.

1.3.4 Conclusion

The extraction of object relationships by LLM focuses on enhancing 3D geometric maps with semantic information. This information is learned from associations between images (color and depth) and natural language queries, addressing SpO-1.3. The related KPI-StO-1.7, concerning object recognition mean average precision, is achieved through the use of LLM, trained on a vast amount of internet data. Despite the challenging nature of evaluation, due to the multitude of ways the same intention can be expressed, a qualitative study demonstrates a high level of scene relationship understanding as so as the limits of the usage for open-set classifier and the robot or patient navigation.

1.4 Multiview open-set object classification

This section briefly updates object detection and classification from multiple views. This object detection is utilized to distinguish between known and unknown objects in the scene to guide the localization and human interaction to enrich the knowledge about the environment. The most specific objects and landmarks are the most valuable priors for semantic localization. Suppose an example of localization in a room with a magnetic resonance machine within the map of a hospital. By learning the unknow magnetic resonance, that was not included in standard datasets, we can localize only from a few images that contain it, i.e., prune the map before the localization by Hloc.

In Computer Vision, there are many extensive studies on both closed-set problems and open-set problems (OSR). Usually, they utilize a single view of the object to make a prediction. Most large modern datasets such as ImageNet [13] are single-view because it is much harder to construct a dataset with high variability of objects, where each object is captured from multiple views. Under the "view," we understand the image of the object captured from the specific position of the camera. While making a decision where only one view is available is a crucial problem for many applications, multiview settings might be beneficial for many problems. There are works such as [17] showing that the use of multiple views of the same object leads to a better closed-set performance of the classifier, especially for problems with large inter-class visual similarity and low intra-class similarity. Our work shows that utilizing multiple views is also beneficial for Open Set Recognition. Information from different views helps to better understand whether the object is from the novel unseen class. Previous work showed that the Vision Transformer [18][4] outperforms other architectures in OSR tasks. We use ViT to extract features from images and use them to compute anomaly score, which is used for known/unknown classification. The relevant codes for this section are in GitLab repository [4].

1.4.1 Anomaly score

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

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

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

Maximum Logit Score (MLS). The authors of [18] propose to use the MLS instead of the softmax probabilities as an open-set indicator. Logits are outputs of the last layer in the network before softmax is applied, which normalizes them. In [2], it was noticed that the logits of unknown instances have a lower magnitude.

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

Average cluster center distance (CCD). As pointed out in [9], unknown objects tend to be farther away in the feature space from the center of clusters consisting of known object's features. Here, we compute an average vector produced by the penultimate layer of the trained network on the training data and call it c_i for the i^{th} class. We compute an average cluster center distance (CCD) to measure novelty.

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

Entropy. Another way to reject unknown objects is to measure the epistemic uncertainty of the detector. Instead of using the maximal probability, we can use the full softmax vector with all class probabilities. In [8] $\overline{s}$ is treated as an average vector of class probabilities over a set of score vectors produced by multiple forward passes of the same data with enabled dropout. In other words, it is the average result of the model ensemble. In our case, $\overline{s}$ represents the mean score vector on multiple views of the same 3D object. Therefore, uncertainty may arise when the model is not able to classify the same object from different perspectives identically. To measure uncertainty, we can use the entropy of the probability vector:

[4] https://gitlab.inria.fr/spring/wp2_mapping_localization/multiview-osr

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

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

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

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

where M is the number of singular values of the embedded matrix Y and σ_1,σ_2,...,σ_M are the normalized singular values of Y. Y is the matrix that contains feature vectors obtained from different views.

GMM score Instead of using single cluster center to represent the class we can use Gaussian Mixture Models (GMM) [10] We obtain a measure of epistemic uncertainty for each known class by computing the log-likelihood of the data l for every known class model G_i

$$a_{GMM} = P = (log(p(l;G_1)), ..., log(p(l;_N)))$$

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

1.4.2 Experiments

In this section, we describe the conducted experiments and discuss the obtained results. The authors of [18] showed that there is a roughly linear relationship between the top-1 accuracy used for the closed-set performance evaluation and the Area Under the Receiver Operating Characteristic (AUROC) used for the open-set performance evaluation. This is why we use the most prominent architecture for image classification, which is Vision Transformer (ViT) [3], respectively, its ViT-B/16 variant. We take the pre-trained model on ImageNet and train it on selected classes from the TinyImagenet for ten epochs, with input size 128x128. TinyImageNet is a subset of ImageNet [13], which has 200 classes and 500 images per class. Usually, it is considered the most challenging case for open-set recognition due to the large number of classes and their complexity.

To evaluate the multiview performance of the proposed model, we use A Large-Scale Hierarchical Multi-View RGB-D Object Dataset [6]. It contains 300 objects common in office and home environments organized into 51 categories. There are 3-10 distinct object instances within each category. Each object is spinning on a turntable at constant speed. Three cameras mounted at three different angles relative to the turntable (approximately 30°, 45° and 60°) record a video sequence at 20 Hz, which results in around 250 RGB-D images per camera and around 250,000 RGB-D images in total. In our work, we use only RGB data, and depth images are ignored.

The number of common classes between the two above-mentioned datasets is very low. We choose ten classes (keyboard, water bottle, flashlight, pitcher, plate, bell pepper, orange, lemon, banana, coffee mug) that have one-to-one correspondence and consider them as known. We train ViT only on TinyImageNet images from 10 selected classes, the other 190 classes are not used. Objects from 41 classes in the RGB-D object data set are labeled unknown.

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

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

Feature embeddings visualisation

To make sure that our approach of measuring the distance to the center of the cluster in the feature space (CCD) is valid, we visualise features of known and unknown objects in 2D (Figure 1.6). We plot 2D T-SNE latent representations

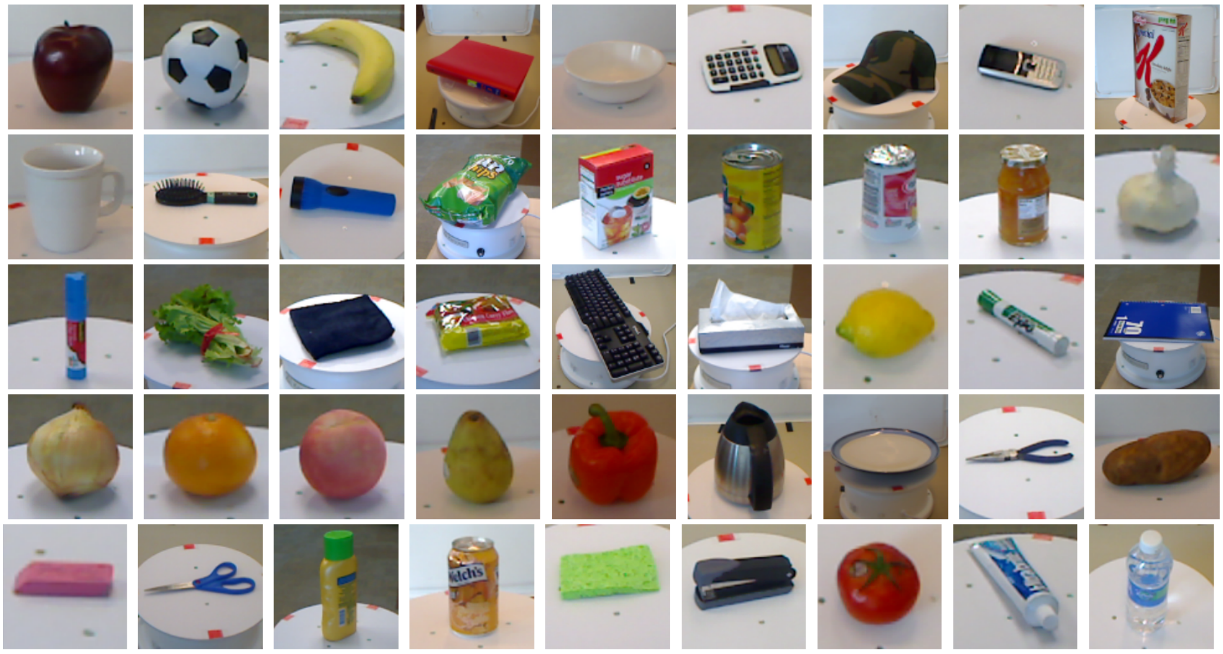


Figure 1.5: Objects from the RGB-D Object Dataset [6]. Each object shown here belongs to a different category. Objects are placed on a turntable which allows to capture them from different sides.

of 300 objects from the multiview dataset. As an input for 2D T-SNE, we provide outputs of the penultimate layer (with dimension 768) averaged over five different views. We see that known objects form tight clusters and are close to the cluster centers computed from the single-view training dataset. Interestingly, semantically similar objects like "orange" and "lemon" or "pitcher" and "water bottle" are placed close to each other. Figure 1.6 demonstrates that unknown objects are usually farther away from the cluster centers than known ones.

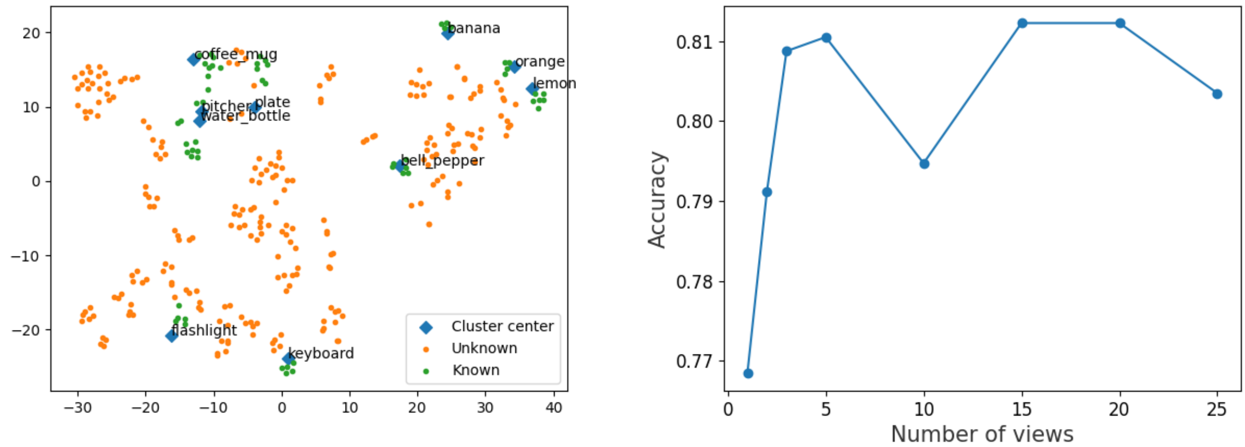


Figure 1.6: Left plot shows 2D T-SNE latent representations of cluster centers, known and unknown objects. The right figure visualize classification accuracy for a different number of available object views. Each result is averaged over ten runs where random views are selected.

Closed set performance

After training the classifier on ten classes from TinyImageNet with a validation accuracy of 95.7% we test it on the multiview dataset. Figure 1.6 shows that the accuracy increases when multiple views are used for classification. Just two additional views result in more than a 3% increase. However, more additional views do not affect performance much. The maximum classification accuracy achieved on the test multiview dataset is 81.2% (15 and 20 views).

Open Set performance

In a similar setting, we evaluate the open-set classification performance. Figure 1.7 shows that all methods benefit from using multiple views. The greatest boost in performance is observed when only one view is added. Also, we can observe that using more than 20 views is not very beneficial. We can see that using raw output (MLS) instead of softmax probabilities (MSP) results in a large increase in AUROC. CCD, MLS, and entropy methods perform very similarly. However, using the SVD entropy of probability scores as an uncertainty measure shows the best performance. The maximum AUROC achieved is 0.919 (50 views). GMM score shows the best performance on a low number of views (1-3).

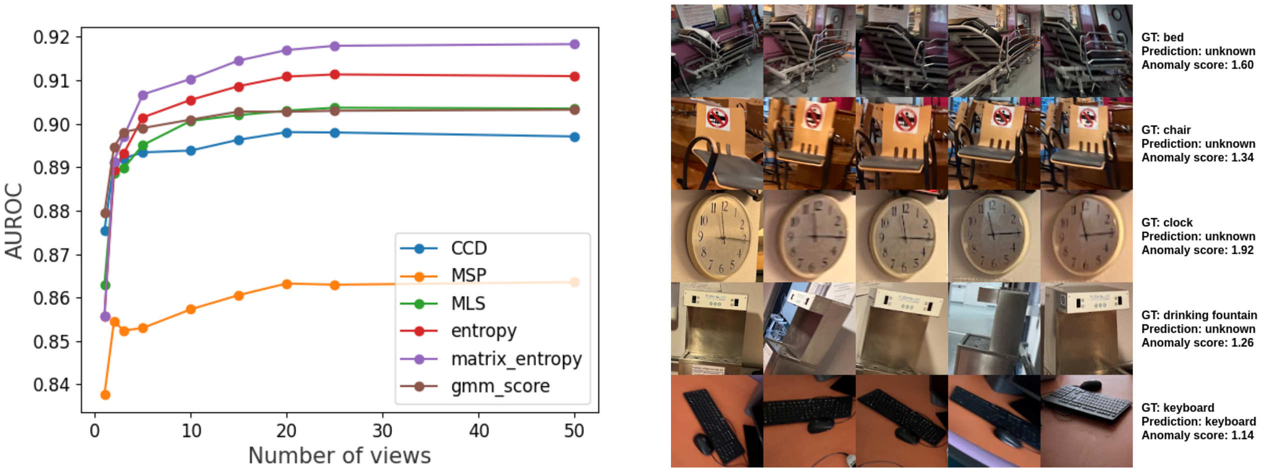


Figure 1.7: The left plot shows the AUROC for different numbers of views. CCD stands for Cluster Center Distance, MSP for Maximum Softmax Probability, and MLS for Maximum Logit Score. These are evaluated on the dataset visualized in Figure 1.5. The right plot shows the evaluation of the best performing anomaly score, i.e., SVD entropy (ViT-B trained on 10 classes), on the Broca dataset. Bed, chair, clock and drinking fountain classes were not used during training.

1.4.3 Results

Table 1.2 shows that using multiple views is beneficial for both closed-set and open-set classification. Observing an object from different angles provides reacher information about it. If the classifier response from different points of view is similar, then the object can be considered as known. On the other hand, if the object looks different from different sides and the model outputs different predictions, then this object can be classified as unknown. From Table 1.2, we can draw an empirical conclusion that 20 views is the best choice in terms of accuracy and AUROC, but 5 views seem to be optimal since the performance drop is not significant, but it is easier/faster to obtain them.

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

Table 1.2: Classification results on RGB-D Object Dataset (10 classes are known, 41 are unknown). For AUROC results, the best open-set method (entropy) is used.

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

Based on our findings, we further plan to fine-tune a model, which has been initially trained on a standard image classification dataset, using a multiview environment-specific dataset. The data collection is under process, and the results of the retrained classifier will be reported in the final technical report. This process will encourage the model to produce consistent outputs for the same objects captured from different viewpoints. We expect this will improve the model's output consistency for known objects, while unknown objects will continue to have varying predicted labels when captured from different perspectives.

1.4.4 Conclusion

This section concentrates on SpO-1.1: performing self-localization and tracking in cluttered and populated spaces, as well as SpO-1.3: augmenting 3D geometric maps with semantic information learned from associations between images and natural language queries. The KPI-StO-1.7, concerning object recognition mean average precision, and KPI-StO-1.8, regarding image retrieval precision and recall, are improved by distinguishing between known and unknown objects. When an object is deemed unknown, we refrain from guessing its identity and instead generate a system message to ask the user for clarification. Furthermore, new specific things serve as optimal guides for pre-filtering images in the map used for localization, thereby addressing KPI-StO-1.9, which pertains to localization accuracy.

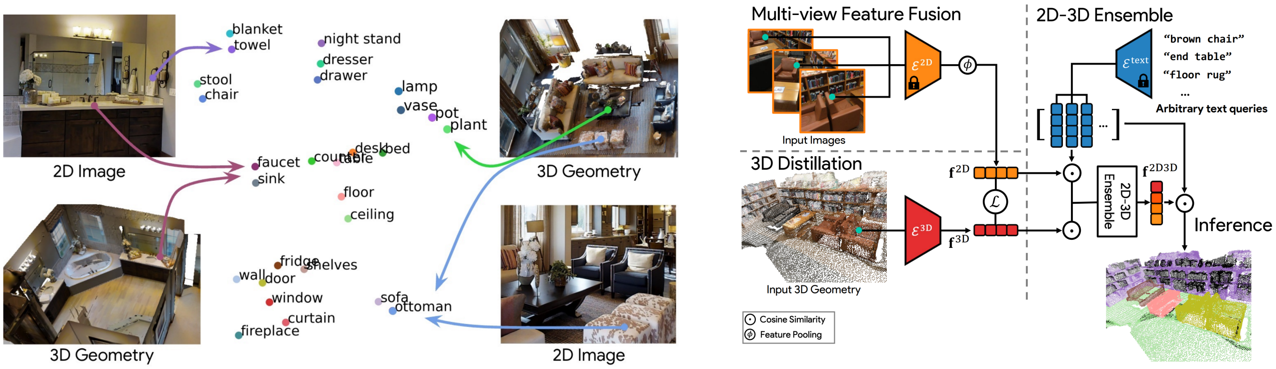


Figure 1.8: The left part of the figure visualizes the main idea of the OpenScene3D paper, i.e., learning feature vectors of 2D pixels and 3D points to achieve the same values by minimizing their cosine similarity loss. The right part of the figure presents an overview of the algorithm. Feature vectors for each image pixel are extracted by pre-trained segmentation models based on CLIP features [5]. Each 3D point has multiple projections into the images, and its feature vector is constructed using an average pooling operator on top of these multi-view features. Another sparse convolutional neural network is trained to produce feature vectors for each 3D point to be the same as those obtained from fused pixel feature vectors. In this way, we achieve the same embedding for text queries, pixels in images, and 3D points. The similarity is further measured as the maximum from the cosine similarity between query-pixels and query-points feature vectors. The visualizations are from OpenScene3D [12].

1.5 Open-set object classification in 3D

This section presents the application of OpenScene3D [12], a recent novel method for 3D scene understanding, with the Broca hospital environment. OpenScene is a groundbreaking approach that leverages text-image embedding models, like CLIP, to offer open-vocabulary 3D scene comprehension. Unlike traditional models that rely on labeled 3D datasets for task-specific supervision, OpenScene uses a zero-shot approach to predict dense features for 3D scene points, enabling task-agnostic training and open-ended queries. The unique aspect of OpenScene lies in its ability to compute dense features for 3D points that are co-embedded with text strings and image pixels within the CLIP feature space. By combining features extracted from both 2D images and 3D geometry, OpenScene is able to identify and understand various aspects of 3D scenes, including objects, materials, affordances, activities, and room types, thus enabling a wider range of 3D scene understanding queries.

Our contributions can be found in the GitLab repository[5]. These consist of methods for running this algorithm in an environment that is actively being mapped by the ARI robot. The results of the experimental evaluation are qualitatively visualized in 1.9, 1.11.

1.5.1 Experiments

We conduct qualitative testing of the Openscene3D scene representation on environment-specific reconstructions. The testing methodologies we employ are as follows:

1. The scene is obtained by Structure from Motion utilizing images from the ARI dataset, and

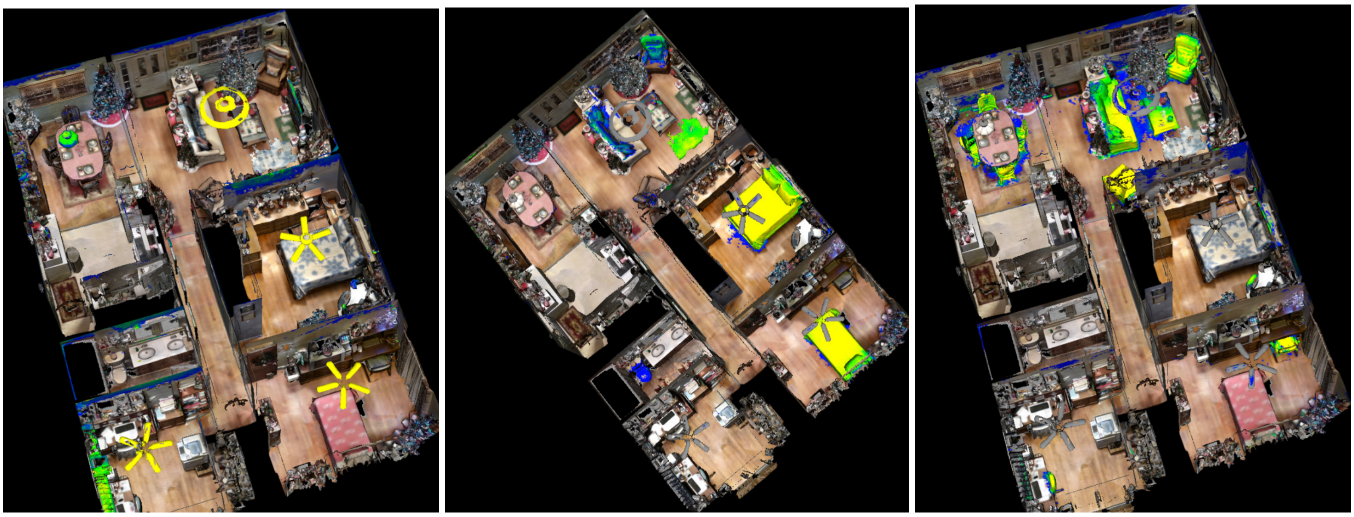
2. A model obtained via a scanning process using a Matterport 3D camera.

SfM using ARI images

We reconstruct the 1st floor in the Broca hospital, with data captured with ARI's front and rear fisheye cameras, using a COLMAP [16] dense reconstruction pipeline.

The scene comprises a dining/waiting room, a hallway or corridor, and the main hall. We first achieve a sparse SfM reconstruction, which estimates a sparse point cloud, along with the camera's intrinsic and extrinsic parameters (up to a similarity transformation). The estimated intrinsic parameters are then utilized to correct the distortion in the fisheye images. These undistorted images are subsequently used for the dense reconstruction employing Multi-View Stereo.

[5] https://gitlab.inria.fr/spring/wp2_mapping_localization/semantic-scene-represenation



(a) Where are the fans?

(b) I'm tired. Where can I lie down?

(c) Where can I sit?

Figure 1.9: The Figure depicts a dense point cloud of a scene, accompanied by a related query and a highlighted area. This area of colored points achieves significant cosine similarity of the feature vectors of the 3D points, pixels in images, and extracted query feature vector. We calculated all the feature vectors by the OpenScene3D approach.

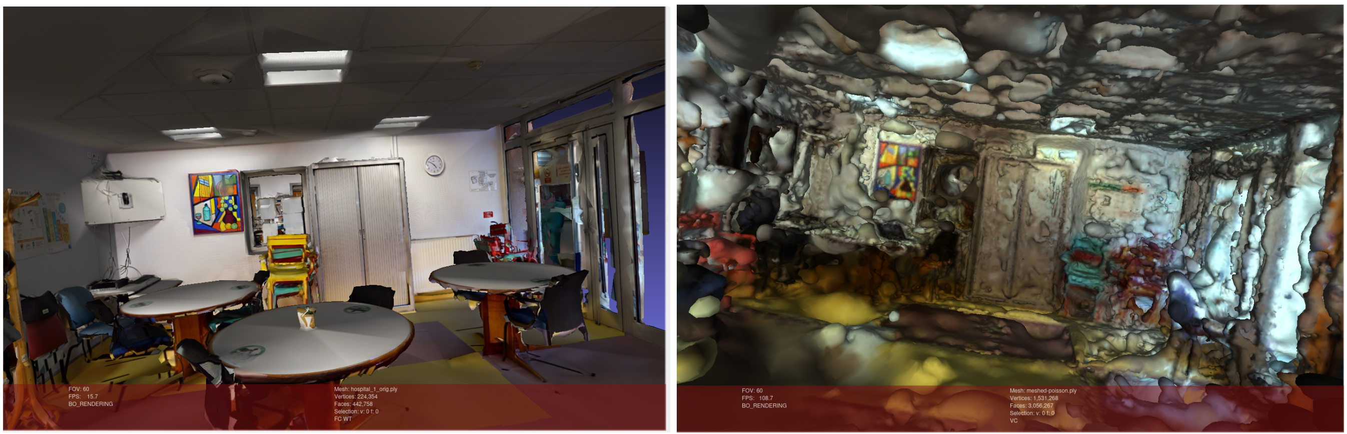


Figure 1.10: Dense reconstruction of the first floor of the Broca hospital. Snapshots of the same region in the Broca hospital are presented, with the left image showing the results of Matterport 3D scanning and the right image displaying the SfM reconstruction from ARI images. The dense model derived from Matterport is complete and less noisy than the reconstruction obtained from ARI images.

Matterport model

We also test the data acquired through manual scanning of the hospital using a Matterport 3D camera. This method delivers higher quality 3D models compared to the aforementioned approach of using SfM with ARI images. Examples of these models, in comparison with the SfM derived from ARI, are illustrated in Figure 1.10.

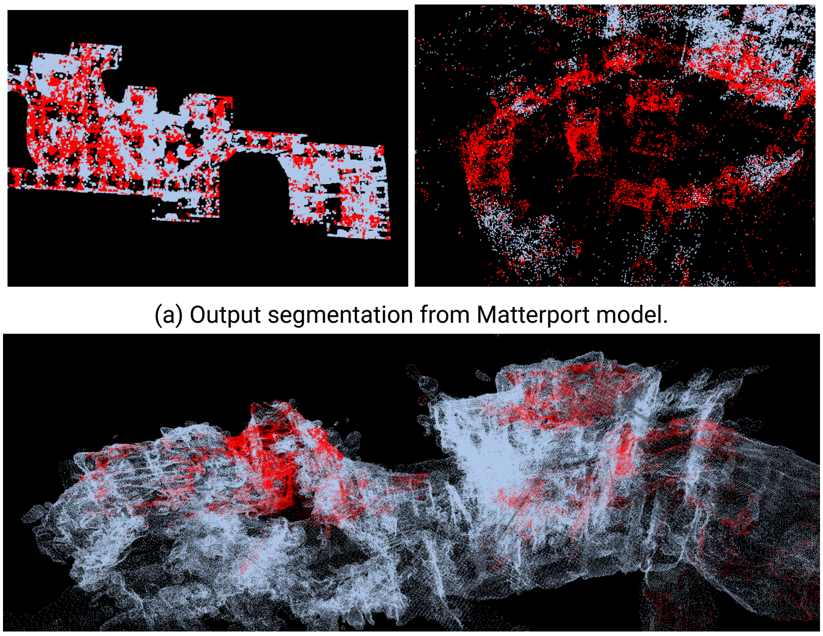
With the dense 3D representation of the scene, we can generate per-point CLIP-aligned features, achieving a semantic representation of the scene.

Extracting OpenScene Feature Vectors and Class Predictions

We extract the semantic feature vectors of each point within the scene. These vectors are utilized to identify regions in the scene that correspond to semantic queries. An illustration of this is provided in Figure 1.9

The process of extracting the semantic feature vectors for each 3D point in the Broca scene unfolds as follows:

- Data is pre-processed to include the images, corresponding depth maps, and the dense scene model in the *.ply* format.



(a) Output segmentation from Matterport model.

(b) Output segmentation from the COLMAP model

Figure 1.11: Segmentation output for the query: "Where can I sit?" The points in red are those with a value higher than a threshold for the query. For the Matterport model, we can see that the chairs are detected, along with some parts of the floor. However, there are false positives on the ceiling and some false negatives, in terms of missed chairs. For the COLMAP model, the output segmentation is significantly poorer. Almost all the chairs are missed, and most of the segmented points are on the ceiling or the walls.

- Using the dense model of Broca from the previous step, we extract the multi-view fused image feature for each 3D point with the OpenSeg [5] model.
- Inference is performed on the collected data using input prompts.
- For each 3D point, we obtain:
  1. A 768-float semantic feature vector,
  2. Scores for each input query prompt,
  3. The class with the highest prediction score in instances of multiple prompts.

Semantic Queries: We test the following queries:

1. Where can I sit?
2. Where is the door?

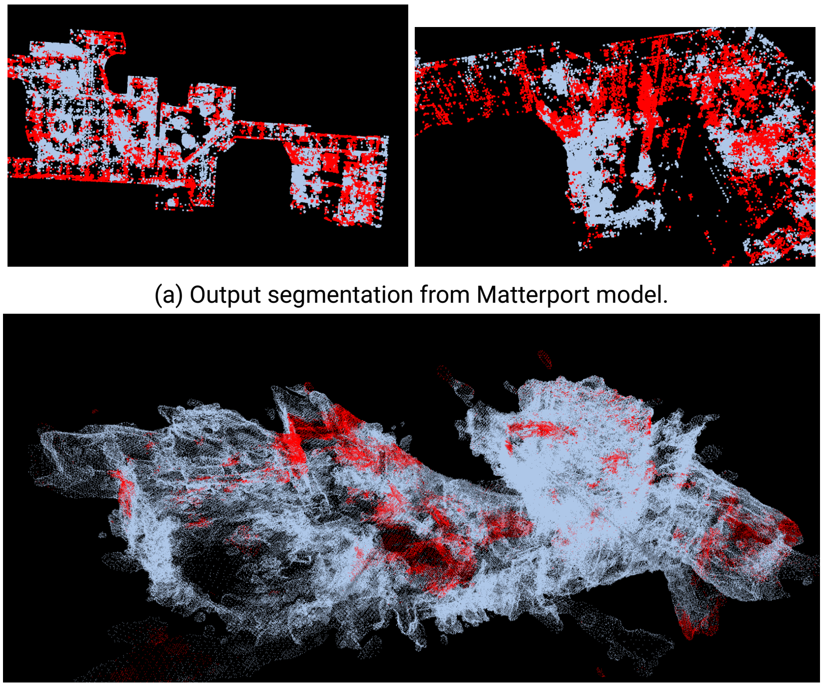
The qualitative results are detailed in figures 1.11 and 1.12

1.5.2 Conclusion

The method proposed in [12] provides a robust semantic representation of the scene, integrating spatial understanding with semantics, thus enabling natural language queries for objects within the map. This capability could prove highly advantageous in hospital scenarios, such as when a patient needs to locate the nearest rest area or toilet. For instance, if a patient prefers to wait in an area with a painting, as opposed to a television, or vice versa. The efficacy of this approach is demonstrated with the demo data in Figure 1.9.

Limitations and Proposed Solutions This method does not perform as well with our data, whether the scene is reconstructed from ARI images or a Matterport scan, even when evaluated qualitatively. Possible reasons include:

1. Density of 3D maps; the networks were trained on data considerably denser than either of the Broca models we tested.



(a) Output segmentation from Matterport model.

(b) Output segmentation from the COLMAP model

Figure 1.12: Segmentation output for the query: "Where is the door?" The points in red are those with a value higher than the threshold for the query. For the Matterport model, we can see that the doors are detected. However, there are false positives where it identifies windows as doors, along with a few points on the ceiling and other miscellaneous points. The count of false negatives does not appear to be high. For the COLMAP model, the output segmentation is significantly poorer. Almost all the doors are missing, and most segmented points are on the ceiling or the walls.

2. Noisy reconstructions, especially in the case of the SfM reconstructions from ARI.

A first step towards enhancing performance would involve distilling and training the network on our data, as opposed to using pre-trained model weights. This should significantly improve the query segmentation. Given the impracticality of manually scanning every scene with a 3D scanner, it would be best to use the 3D reconstructions derived from ARI images of the scene. One potential approach in this direction could involve leveraging different reconstruction pipelines such as RealityCapture or Meshroom, which might yield better results.

This section focuses on SpO-1.3: the augmentation of 3D geometric maps with semantic information acquired through associations between images and natural language queries. Given the broad spectrum of environment-specific classes we anticipate in a hospital setting, the KPI-StO-1.7 - (object recognition mean average precision) is enhanced by employing CLIP features trained on a large dataset of internet images and their respective captions. As a result, very few object classes remain unknown, enabling the robot to handle most discussions about them effectively. The relationship with KPI-StO-1.8 (image retrieval precision and recall) and KPI-StO-1.9 (localization accuracy) is discussed in the subsequent section.

1.6 Localization using open-set object classification

Localization typically involves finding correspondences between feature points in map images and query images. However, maps can contain tens of thousands to millions of images, making such searches time-consuming, potentially taking tens of minutes. Thus, a common solution is to prefilter the images using a simpler and less accurate approach, such as image retrieval. One standard method is training the weights of the descriptors of the points of interest in the images, for instance, by utilizing NetVLAD [1]. This approach extracts a single feature vector for each image, reducing the number of comparisons required to identify pictures with similar content.

This deliverable focus on scene representation for semantic localization, i.e., let us briefly discuss the proposed localization on top of the OpenScene segmentation. We suggest utilizing the semantic classes, i.e., their names, attributes, and affordances, that offer a more comprehensive understanding of the current environment than individual feature points. Therefore, similar images can be prefilter by training the aggregation neural network weights for the feature vectors extracted using the OpenScene3D approach. Moreover, the semantic classes of objects can also be utilized to prefilter correspondences based on class membership. For example, we do not need to match the feature points on a window with the feature points of a chair. We are going to evaluate this approach on standard and hospital datasets.

2 Conclusions

This deliverable, focused on enhancing 3D scene representation, represents a significant step toward achieving SpO-1.1 objectives: performing self-localization and tracking in cluttered and populated spaces. Particularly, the results presented are based on the application of these developments on the Broca dataset. The ability to accurately localize is crucial for the effective functioning of assistive robots. We have demonstrated a multi-view localization approach that can reach an accuracy of below 3cm using the last four images captured from a 2-second recording at Broca Hospital at a frame rate of 2 fps. This accuracy achieves about an order better result than required by KPI-StO-1.9. Our efforts also include integrating these localization codes into a Docker container, making the software ready to use on the ARI robot.

Applying Large Language Models (LLMs) for language and image inputs is highly popular because it can extract relevant information about the scene object properties. Recent models such as LLaVA and MiniGPT-4 have shown promising qualitative results, while quantitatively, the outputs are appropriate in approximately 60-70% of examples. These models provide a method for extracting object descriptions, relationships, and usage, thus enriching the robot's environmental knowledge and enhancing human-robot conversation. While these models present several benefits, they also have limitations, which are discussed in detail. The main one is the consistency of the answers across multiple similar queries and hallucinations when the objects are far from the camera or out of the training dataset. The presented experimental evaluation of LLMs addresses SpO-1.3 and the related KPI-StO-1.7.

To enhance localization speed, we investigated using semantic information to pre-filter candidate images in the image retrieval task, i.e., the KPI-StO-1.8 (image retrieval precision and recall). Moreover, the anomaly cores are employed to identify rare but significant semantic classes not typically present in segmentation datasets. Such objects can be used to filter out irrelevant map images during the online localization process. The best-performing anomaly score was tested on a small Broca dataset composed of pictures of objects recorded by the ARI robot.

The OpenScene segmentation approach is evaluated in both standard and hospital environments. Properly trained networks could select objects based on user questions such as "Where can I sit?" and "I'm tired. Where can I lie down?". This directly serves StO-1 and SpO-1.3 and addresses KPI-StO-1.2 (Multiple object tracking accuracy) and KPI-StO-1.7 (Object recognition mean average precision) as so as KPI-StO-1.8 (Image retrieval precision and recall) and KPI-StO-1.9 (Localization accuracy). This capability significantly enhances the natural discussions between the robot and human users and can improve the speed of localization and image retrieval, as proposed in the last section. However, a fundamental limitation is that the model requires retraining when moving from high-quality 3D environmental scans to sparse and quasi-dense 3D scenes. This limitation emphasizes the challenge of generalizing this approach to out-of-domain data sources.

As we move forward, we will continue to refine our models and techniques to overcome current limitations and enhance the capabilities of assistive robots. In keeping with European Commission guidelines, the software developed in the course of this project will be made available via the GitLab code repositories and will remain publicly accessible for a minimum of four years following the conclusion of the SPRING project. This ensures that our contributions can be leveraged by the broader research and development community.

Bibliography

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016. 20

[2] Akshay Raj Dhamija, Manuel Günther, and Terrance Boult. Reducing network agnostophobia. *Advances in Neural Information Processing Systems*, 31, 2018. 11

[3] 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. In *ICLR*, 2021. 12

[4] Silvio Galesso, Max Argus, and Thomas Brox. Far away in the deep space: Nearest-neighbor-based dense out-of-distribution detection. *arXiv preprint arXiv:2211.06660*, 2022. 11

[5] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European Conference on Computer Vision*, pages 540–557. Springer, 2022. 16, 18

[6] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *2011 IEEE international conference on robotics and automation*, pages 1817–1824. IEEE, 2011. 12, 13

[7] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 8, 9

[8] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3243–3249, 2018. 11

[9] Dimity Miller, Niko Sunderhauf, Michael Milford, and Feras Dayoub. Class anchor clustering: A loss for distance-based open set recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3570–3578, 2021. 11

[10] Dimity Miller, Niko Sünderhauf, Michael Milford, and Feras Dayoub. Uncertainty for identifying open-set errors in visual object detection. *IEEE Robotics and Automation Letters*, 7(1):215–222, 2021. 12

[11] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 6

[12] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–824, June 2023. 16, 18

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 11, 12

[14] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019. 6

[15] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 6

[16] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 6, 16

[17] Marco Seeland and Patrick Mäder. Multi-view classification with convolutional neural networks. *Plos one*, 16(1):e0245230, 2021. 11

[18] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. *arXiv preprint arXiv:2110.06207*, 2021. 11, 12

[19] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 8, 9