LIGHT 2-STEP AMODAL SHAPE COMPLETION

Nicolas Pinto
Department of Computer Science
University of Bari
Bari, Italy
n.pinto15@studenti.uniba.it

Emanuele Tanzi
Department of Computer Science
University of Bari
Bari, Italy
e.tanzi2@studenti.uniba.it

July 5, 2024

ABSTRACT

In this study, we address the challenging task of amodal shape completion, a crucial aspect of computer vision involving the prediction of the full shape of partially occluded objects. This problem has significant implications for various applications such as robotics, autonomous driving, augmented reality, and object recognition. Traditional object recognition and segmentation methods often struggle with occluded objects due to their reliance on visible features. Our motivation stems from the need to improve system capabilities in understanding and interacting with environments by accurately predicting hidden parts of objects based on visible portions and contextual information. We propose a two-step approach, "Light 2-Step Amodal Shape Completion" which integrates UNet and CNN architectures to effectively complete occluded object shapes. The first step focuses on predicting the occlusion mask using the visible mask of the object, while the second step generates the amodal mask, representing the complete silhouette of the object, including both visible and occluded parts. Our methodology leverages the COCOA dataset, which provides comprehensive annotations for both modal and amodal masks, enabling robust training and evaluation of our models. Experimental results demonstrate that our two-step approach addresses sufficiently well the complexities of amodal shape completion, achieving notable performance on the COCOA dataset, achieving a modest result, with mIoU = 0.75. Despite the computational limitations, our models exceeded initial expectations, showcasing robustness even under resource constraints. Additionally, the lack of reproducible experiments in existing literature posed challenges, necessitating innovative adaptations from prior studies. The code is available at: https://github.com/npinto97/AmodalShapeCompletion

1 Introduction

Amodal shape completion is a significant challenge in the field of computer vision, where the objective is to predict the full shape of an object that is partially occluded. This task is crucial for various applications, such as robotics, autonomous driving, augmented reality, and object recognition. Accurately identifying and completing the shapes of objects, even when parts of them are not visible, enhances the ability of systems to understand and interact with their environments more effectively.

Traditional methods for object recognition and segmentation often struggle with occluded objects, as they rely heavily on visible features. However, amodal completion requires models to infer and predict the hidden parts of objects based on the visible portions and contextual information from the surrounding scene. This additional layer of complexity necessitates advanced techniques that can generalize well from limited and partially occluded data.

In this project, we propose a novel approach named "Light 2-Step Amodal Shape Completion," which addresses the challenge of amodal shape completion through a two-step process. Our approach leverages both UNet and CNN architectures to efficiently and accurately complete the shapes of occluded objects. The process involves two primary steps:

- Occlusion Mask Prediction: The first phase focuses on predicting the occlusion mask using the visible (modal) mask of the object. The occlusion mask represents the non-visible parts of the object, allowing us to isolate the areas hidden from view.
- Amodal Mask Prediction: In the second phase, we use the occlusion mask obtained in the first step to identify the occlusion boundaries. These boundaries, along with the modal mask, are then used to generate the complete amodal mask, representing the entire silhouette of the object, including both visible and occluded parts.

To validate our approach, we employed the COCOA (Common Object in COntext Amodal) dataset, which provides comprehensive annotations for both modal and amodal masks. This dataset allows for robust training and evaluation of our models. We experimented with two distinct model architectures: UNet, known for its effectiveness in image segmentation tasks, and a simpler CNN model, recognized for its computational efficiency and strong feature extraction capabilities.

Our experimental results demonstrate that the proposed two-step method addresses sufficiently well the challenge of amodal shape completion. UNet provides detailed and accurate segmentations, while the CNN-based approach offers simplicity and efficiency, making it potentially suitable for scenarios with limited computational resources, despite its lower accuracy. Despite the room for improvement, this study could be a good starting point for future experiments because of its reproducibility.

In the following sections, we delve into the related work, discussing previous research in amodal shape completion. We then describe the materials and methods used in our project, detailing the datasets, tools, and techniques employed. The results section presents our findings, describing the metrics used and giving reasons for their selection, then providing an overview of the results obtained. Finally, we conclude by summarizing the strengths and limitations of both proposed architectures and offering insights into potential future directions for amodal shape completion.

2 Related work

The field of amodal completion is relatively new and presents numerous challenges. Unlike standard segmentation tasks, amodal segmentation requires inferring the complete shape of objects, including their occluded parts. The ability to predict amodal masks, which encompass both the visible and hidden portions of an object, is crucial for various downstream tasks such as object detection, smart image editing, 3D reconstruction, and object permanence in video segmentation. This task is inherently complex due to the need to predict the unseen parts of objects, a capability that standard image segmentation models lack [1].

One of the primary difficulties in amodal segmentation is the limited availability of large-scale, high-quality datasets with authentic ground truth amodal masks. Traditional datasets often rely on synthetic images or human annotations, which can be inconsistent and less reliable. The MP3D-Amodal dataset, introduced in [2], addresses this issue by providing real images with authentic amodal segmentation annotations. This dataset is built from MatterPort3D, encompassing a variety of objects in indoor scenes, thus offering a more realistic benchmark for evaluating model performance.

Early methods in amodal segmentation often required explicit occluder masks to infer the occluded parts of objects [3, 4]. These methods were limited by their dependency on occluder masks, making them less practical for real-world applications. However, recent approaches like OccAmodal and SDAmodal [5] have made significant strides in overcoming this limitation. The OccAmodal model uses a two-stage process that first infers the occluder before completing the amodal mask, whereas the SDAmodal model utilizes the outpainting capabilities of a pre-trained Stable Diffusion network in a single-stage process [6]. The pipeline can be extended to other datasets like ScanNet [7], which combine real images with the 3D structure of objects. The authors developed two state-of-the-art methods for amodal completion capable of handling situations with undefined occluders and a wide variety of object classes. By incorporating a lightweight occluder predictor and Stable Diffusion representations, these models achieve superior performance across different domains and object categories [5].

Despite these advancements, the task remains challenging due to the complexities involved in predicting the occluded regions accurately. [8] propose a method that combines visible region segmentation with category-specific shape priors to enhance amodal segmentation accuracy. Their approach involves a coarse mask segmentation module, a visible mask segmentation module, and an amodal mask segmentation module, which together refine the amodal mask by leveraging shape priors encoded through an auto-encoder.

[9] introduce a zero-shot amodal segmentation approach using denoising diffusion models trained on synthetic datasets. Their method synthesizes whole objects from occluded images, achieving state-of-the-art performance in a zero-shot

setting on multiple benchmarks. This demonstrates that diffusion models, when trained on large-scale data, can implicitly learn amodal representations that generalize well across various domains and object categories.

Moreover, [10] present an end-to-end pipeline for amodal scene analysis that infers occlusion relationships and completes occluded regions. Their dual-branch structure, which includes an occlusion relationship inference branch and an amodal mask completion branch, leverages a generative model trained on a large-scale dataset and a graph neural network for modeling occlusion relationships. This approach effectively handles complex scenes with multiple occluded objects, achieving state-of-the-art performance on benchmarks such as COCOA [11] and KINS [12].

AISFormer [13] represents a significant advancement in amodal instance segmentation by leveraging Transformer-based architectures. The framework consists of four primary modules: feature encoding, mask transformer decoding, invisible mask embedding, and mask predicting. By treating the complex relationships between occluder, visible, amodal, and invisible mask instances as learnable queries, AISFormer can model these interactions explicitly within the region of interest (ROI). Experimental results and ablation studies on benchmarks such as KINS, D2SA, and COCOA-cls have demonstrated the effectiveness of these learnable queries, with AISFormer achieving state-of-the-art (SOTA) performance in AIS tasks. Future research will explore integrating shape priors into AISFormer and applying it to other modalities like time-lapse and videos [13].

Pix2gestalt [14] presents a novel zero-shot amodal segmentation approach that leverages synthesis. This model utilizes whole object priors learned by large-scale diffusion models, fine-tuned on synthetically generated datasets of realistic occlusions. By synthesizing entire objects, the model equips various computer vision methods with enhanced capabilities to handle occlusions. Pix2gestalt demonstrates SOTA results on several benchmarks for amodal segmentation, occluded object recognition, and 3D reconstruction.

[15]'s approach mimics human amodal perception by using shape priors to infer invisible regions based primarily on visible features. Unlike other methods that rely on the appearance of the entire region-of-interest (ROI), this method focuses on visible regions using attention mechanisms and a codebook to store amodal shape prior embeddings. This strategy addresses the ambiguity present in conventional methods where similar appearances may require different predictions. Experimental results show that this method outperforms existing state-of-the-art methods by effectively simulating human-like imagination from visible regions and shape priors.

The framework [16] introduces two key modules: the Holistic Occlusion Relation Inference (HORI) module and the Generative Mask Completion (GMC) module. The HORI module predicts an occlusion relationship matrix in a single pass, improving inference efficiency and enabling reasoning about mutual occlusions. The GMC module frames amodal mask completion as a generative process, offering multiple high-quality plausible solutions. Experimental results on COCOA, KINS, and the ASW benchmark show that this framework achieves SOTA performance and robustness across various occlusion scenarios, making it a vital baseline for future research in amodal scene analysis.

The C2F-Seg framework [17] employs transformers to learn shape priors in the latent space, starting with the generation of a coarse mask. It then utilizes a dual-branch refinement module to produce an attention mask, which is combined with the coarse mask and ResNet-50 features to predict both visible and amodal masks. For video datasets, the transformer block is adapted into a spatial-temporal version to capture spatio-temporal features effectively. This step-by-step refinement process results in precise amodal masks, achieving new SOTA performance on both image and video amodal segmentation tasks.

Comparative Discussion Across these various approaches, a few common themes and limitations emerge. Transformer-based architectures, as seen in AISFormer [13] and C2F-Seg [17], have shown great promise in modeling complex relationships and refining segmentation masks progressively. However, these models often require substantial computational resources and extensive training data to achieve optimal performance.

The integration of shape priors, as demonstrated by Xiao et al. [15] and C2F-Seg [17], enhances the ability to predict occluded regions accurately by leveraging learned embeddings or latent representations. Yet, these methods might struggle with objects that significantly deviate from learned shapes or with highly irregular occlusions.

Models like those developed in [5] and [14] emphasize the importance of using real 3D data and synthetic datasets to improve generalization across various domains. These approaches highlight the potential for zero-shot learning and robust performance in diverse scenarios. Nevertheless, the dependency on high-quality 3D data or large-scale diffusion models can be a limiting factor for broader applicability.

Finally, the holistic approaches combining occlusion reasoning with generative processes, as seen in [16], offer a comprehensive solution to amodal segmentation by addressing mutual occlusions and generating multiple plausible completions. These methods, while highly effective, may introduce complexity in model design and inference processes, necessitating further optimization for real-time applications.

Our work, Light 2-Step Amodal Shape Completion, takes inspiration from the OccAmodal approach proposed in [5]. Building upon the two-stage process of inferring occluders before completing the amodal mask, we introduce a lightweight methodology designed to enhance the applicability of amodal segmentation in real-world scenarios. A significant advantage of our approach is its ease of replication, contrasting with the complexity and difficulty in testing and reproducing results seen in other state-of-the-art systems. Our approach involves a streamlined two-step pipeline that first predicts the occluder mask and then completes the amodal shape, leveraging both traditional segmentation techniques and advanced shape priors. In the following sections, we detail the architecture and workflow of our model, making it accessible for a wider range of applications and researchers.

3 Materials

For our project, we employed the COCOA (Common Objects in COntext Amodal) dataset, an extension of the well-known COCO (Common Objects in Context) dataset specifically designed for amodal segmentation tasks. In particular, we used the 2014 version, the most widely used. The COCOA dataset provides additional annotations that include both modal (visible) and amodal (occluded) masks for various objects in diverse scenes. This dataset is crucial for training and evaluating models designed for amodal shape completion, as it offers a comprehensive view of objects, including both their visible and hidden parts.

The dataset contains two image folders, one employed for training while the other for testing. Associated with the two folders are two JSON files containing annotations. The annotations provide detailed information about the visible and occluded parts of objects in the images, making it ideal for developing and assessing models for amodal segmentation.

To process the dataset and perform the necessary preprocessing steps, we utilized several tools and libraries. The Pycocotools library was fundamental in this process, as it is designed to work with COCO-style annotations, offering essential functions for manipulating and handling annotation data. Additionally, we employed Torch (PyTorch), a widely-used machine learning library that facilitates the implementation and training of deep learning models. For image processing and visualization, we relied on OpenCV and PIL, which were useful for preprocessing images and displaying results.

Data preprocessing was a critical step in our project. Using the Pycocotools library, we loaded and processed the COCOA annotations. Once the annotations were loaded, we extracted the visible (modal) masks and the occlusion masks from the annotations. The visible masks represent the parts of the objects that are visible in the images, while the occlusion masks represent the occluded parts of the objects, inferred from the surrounding context. These masks are essential for generating the amodal masks, which represent the entire silhouette of the objects, including both visible and occluded parts. Not all annotations possess amodal segmentation information, so as a first pre-processing step, it was necessary to store pairs (consisting of image id and annotation, the main information to identify an instance) having the necessary information.

To ensure the quality and relevance of our dataset, we applied several preprocessing steps. First, we filtered the images based on the occlusion area, removing those where the occluded objects had a total surface area below a certain threshold (1% of the entire image). This step ensured that only significant occlusions were considered, improving the training process by focusing on meaningful data. This pre-processing step resulted in a reduction of about 61.5% in the training set and about 62.0% in the test set (Figure 1).

Next, we applied data augmentation techniques to enrich the dataset and enhance the robustness of our models. The data augmentation techniques included random horizontal flipping of images and random adjustments to brightness and contrast. These techniques increased the variability of the training data, improving the models' ability to generalize to unseen data.

To efficiently manage and process the dataset, we implemented a custom dataset class. This class encapsulated all relevant information for each image, including the image ID, annotation index, modal mask, occlusion mask, occlusion boundary, and amodal mask. Additionally, the class applied the filtering and augmentation steps during data loading, ensuring that our models were trained on high-quality and varied data. At the end of this phase, we obtained a dataset consisting of 3801 instances for training and 2151 instances for testing (the dataset dimensions for each step are reported in Table 1).

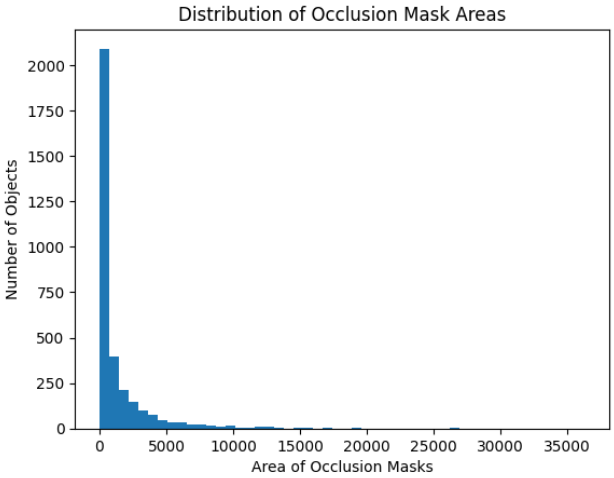

Figure 1: Distribution of Occlusion Mask Areas

Pre-processing step	Train Set Dimension	Test Set Dimension
Post filtering	3287	1884
Occlusion mask based reduction	1267	717
Post data augmentation	3801	2151

Table 1: Dataset dimensions for each step

4 Methods

4.1 Proposed Method

Our strategy for amodal shape completion involves a two-step approach, designed to incrementally build the understanding necessary for predicting the complete shape of occluded objects. For this reason, we trained two different model to face two aspects of the problem:

The first one, called "Amodal Occlusion", focuses on predicting the occlusion mask. The model takes as input the image, resized to 256x256 pixels for ease of processing, and the modal mask, which represents the visible part of the object. The task here is for the model to learn to predict the occlusion mask, which indicates the parts of the object that are hidden from view. This task is complex because it requires the model to infer the non-visible parts of an object based on the visible parts and the context provided by the image. However, the precision of the occlusion mask is not critically important; what matters is that the model learns the general areas where the visible mask should be extended to form the complete, amodal mask. This prediction provides a rough estimation of the occluded areas, which is sufficient for our needs.

In the second step, the goal is to generate the amodal mask, which represents the complete shape of the object, including both visible and occluded parts. Here, the model uses the occlusion boundaries derived from the predicted occlusion mask in the first step. These boundaries highlight the transitions between visible and occluded regions. Toghether with the occlusion boundaries, the original images and the modal masks are used by the model to predict the amodal masks.

4.2 Architectures

4.2.1 UNet-based Architecture

The UNet model is named for its U-shaped architecture, which consists of a contracting path (encoder) and an expansive path (decoder). This design allows the network to effectively learn hierarchical features and spatial relationships within the input images.

We had to manipulate the UNet architecture for the two steps, since they had to manage different kinds of inputs. However, we managed to keep the architecture coherent in both definitions.

The contracting path is responsible for capturing the context of the input image. It progressively reduces the spatial dimensions while increasing the depth, thus enabling the network to learn more abstract features. Our UNet implementation includes the following layers in the contracting path:

- Initial Convolution (inconv): This layer processes the input image and modal mask, applying a double convolution operation that includes ReLU activations and batch normalization.
- Downsampling Layers (down): These layers consist of a max-pooling operation followed by a double convolution. Each downsampling step reduces the spatial dimensions by half and increases the number of feature channels, allowing the network to capture increasingly complex features.

The expansive path reconstructs the spatial dimensions of the image while combining the learned features from the contracting path. This path helps to accurately localize the features within the image. Our UNet's expansive path includes:

- Upsampling Layers (up): Each upsampling layer uses either bilinear interpolation or a transposed convolution to increase the spatial dimensions. The upsampled feature maps are concatenated with the corresponding feature maps from the contracting path, providing the network with high-resolution features from earlier layers.
- Final Convolution (outconv): This layer generates the final output, producing the desired number of segmentation classes (masks).

To enhance the feature extraction capability, we integrated a pre-trained ResNet-18 model into the UNet architecture. This ResNet-18 model serves as a powerful image encoder, providing rich feature representations that are further processed by the UNet. The integration is achieved by:

- Image Encoder: The ResNet-18 model is truncated to remove its final classification layers, retaining only the feature extraction layers.
- Dimension Reduction (reduce_dim): A convolutional layer is used to match the dimensions of the feature maps from the ResNet-18 encoder with those of the UNet's expansive path.

The UNet architecture combines the feature maps from the ResNet-18 encoder and the UNet decoder through concatenation. This combination leverages both the detailed features learned by the ResNet-18 and the spatial information captured by the UNet's contracting path. This synergy allows the model to produce accurate and detailed segmentation masks.

In the forward pass, the input image and modal mask are concatenated and passed through the contracting path. The feature maps from the image encoder (ResNet-18) are integrated at the bottleneck, and the expansive path then reconstructs the high-resolution feature maps to produce the final segmentation output.

The version specialized in predicting amodal segmentation differs only in input, adding another channel for managing the occlusion boundary masks, so adding another mask to concatenation. Moreover, it separates the processing of the image through the ResNet-18 encoder, reduces its dimensionality, and then combines it with the features extracted from the concatenated image, modal mask, and occlusion boundary. This allows the network to effectively utilize both the high-level features from ResNet-18 and the detailed boundary information from the occlusion boundary. More qualitative results are in the appendix.

4.2.2 CNN-based Architecture

For the second experiment, we employed a CNN-based architecture. Using CNNs is the first, typical approach to solve this task. This architecture is relatively straightforward compared to more complex models like UNet, yet it effectively captures spatial features and reconstructs them to produce the desired output. Here's an in-depth look at the components and workings of this model. There are multiple reasons for which the CNN approach is still valid:

- Simplicity and Efficiency: The SimpleCNN architecture is straightforward and less computationally intensive compared to more complex models, making it faster to train and suitable for scenarios with limited computational resources.
- Strong Feature Extraction: Despite its simplicity, the CNN architecture is capable of extracting powerful features from the input images, thanks to multiple convolutional and pooling layers.
- Effective for Image Processing: CNNs have a proven track record in image processing tasks, including segmentation, due to their ability to capture spatial hierarchies in images.

- Flexibility: The SimpleCNN approach is highly flexible and can be easily adapted to different input modalities and segmentation tasks, making it a versatile choice for our project.

The encoder comprises a series of convolutional layers designed to extract spatial features from the input images. These layers are followed by the ReLU activation function with `inplace=True`, which introduces non-linearity into the model while optimizing memory usage. To further improve the model's efficiency, MaxPool2d layers are incorporated, reducing the size of the feature maps. This reduction not only increases computational efficiency but also mitigates the risk of overfitting by downsampling the input. The encoder structure can be summarized as follows:

1. First Layer: Convolutional layer with 64 filters, ReLU activation, and max-pooling.
2. Second Layer: Convolutional layer with 128 filters, ReLU activation, and max-pooling.
3. Third Layer: Convolutional layer with 256 filters, ReLU activation, and max-pooling.
4. Fourth Layer: Convolutional layer with 512 filters, ReLU activation, and max-pooling.

In the decoder, transposed convolutional layers (ConvTranspose2d) are employed to incrementally restore the spatial dimensions of the feature maps to their original size. These layers are complemented by standard convolutional layers, which refine the feature maps at each stage of the decoder. The ReLU activation function is again utilized, adding non-linearity to each layer. This non-linearity is crucial as it enables the network to learn and represent complex patterns within the data. The decoder structure is:

1. First Layer: Transposed convolutional layer to upsample, followed by a convolutional layer with 256 filters, and ReLU activation.
2. Second Layer: Transposed convolutional layer to upsample, followed by a convolutional layer with 128 filters, and ReLU activation.
3. Third Layer: Transposed convolutional layer to upsample, followed by a convolutional layer with 64 filters, and ReLU activation.
4. Fourth Layer: Transposed convolutional layer to upsample, followed by a final convolutional layer to produce the output mask.

The forward method begins by concatenating the input image and the modal mask along the color channel using torch.cat. This concatenation allows the network to simultaneously process both the visible parts of the image and the modal mask. The combined data then flows through the encoder, where essential features are extracted. Subsequently, it passes through the decoder, which reconstructs the spatial dimensions to produce the final occlusion mask. This method ensures that the network can effectively predict the occluded regions in the input image.

For the segmentation step, the architecture presents some difference with the one used for occlusion step:

- Number of channel used: unlike the amodal occlusion network, which uses a single mask, this network requires two masks for training. Consequently, the input channel count is adjusted accordingly.
- Forward method: in this network, the modal mask and the occlusion boundary are concatenated along the channel dimension. This expands on the forward method of the amodal occlusion network, where only the image and the modal mask were concatenated.

More qualitative results are in the appendix.

4.3 Amodal Occlusion Computation

Firstly, we utilize the trained occlusion prediction model to obtain the occlusion masks for the test set. These occlusion masks represent the hidden parts of the objects that are not visible due to occlusion. Alongside the modal masks (which represent the visible parts of the objects), these occlusion masks are employed to compute the occlusion boundaries. These boundaries are critical for the next steps, as they provide precise delineation of the occluded regions.

Once we have the occlusion boundaries, we modify the original test set to incorporate these newly computed boundaries. Specifically, the original occlusion boundaries in the test set are replaced with the new boundaries obtained from the occlusion prediction model. This results in a new test dataset that accurately reflects the occlusion scenarios as predicted by the model. This step ensures that the test set is up-to-date with the latest occlusion boundary predictions, which is essential for accurate evaluation.

Finally, we use this updated test set to evaluate the performance of the amodal segmentation model. The model, trained to predict the entire silhouette of occluded objects, utilizes the new test set with the updated occlusion boundaries to make its predictions.

4.4 Experiments Setup

To preserve consistency between experiments, we decided to adopt the same setup for both experiments, which is as follows[1]:

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = Model(n_channels=x, n_classes=1).to(device)
criterion = BCEDiceLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)
```

In our model training, we employed a combined loss function that integrates both Dice Loss and Binary Cross-Entropy (BCE) Loss. This combination was chosen to harness the complementary strengths of both loss functions, leading to improved performance and convergence during training.

BCE Loss is a widely used loss function for binary classification tasks, and it is particularly effective for pixel-wise classification in segmentation tasks. BCE Loss measures the discrepancy between the predicted probability and the true label for each pixel. It is defined as:

$$\text{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

Where N is the number of pixels, y_i

Dice Loss is designed to measure the overlap between the predicted segmentation mask and the ground truth mask. It directly optimizes the segmentation performance by focusing on the regions of interest. Dice Loss is defined as:

$$\text{Dice} = 1 - \frac{2\sum_{i=1}^{N} p_i y_i + \epsilon}{\sum_{i=1}^{N} p_i + \sum_{i=1}^{N} y_i + \epsilon}$$

where ϵ is a smoothing term to prevent division by zero, N is the number of pixels, p_i is the predicted probability, and y_i is the ground truth label for pixel i

By combining BCE Loss and Dice Loss, we leverage the strengths of both methods, taking the accurate pixel-wise classification with smooth gradients for stable training from BCE Loss and the ability to enhance the overall overlap between predicted and true masks, focusing on critical regions and handling class imbalance, from Dice Loss.

Since we used the resources provided by Colab to conduct our experiments, we couldn't implement a particularly complex parameter search strategy, such as grid search or more advanced techniques, to optimize the learning rate.

Nonetheless, we performed several separate experiments with different learning rate values. We also applied a weight decay strategy, which did not lead to significant improvements.

5 Results

For the evaluation, in both amodal occlusion and amodal segmentation, we employed three key metrics: mean Intersection over Union (mIoU), Dice Coefficient, and Inverse Mean IoU.

Mean Intersection over Union (mIoU) is a standard metric used in segmentation tasks to evaluate how well the predicted segmentation matches the actual object. It calculates the ratio of the intersection of the predicted and ground truth masks to their union, providing a comprehensive measure of accuracy.

$$\text{IoU} = \frac{|\text{Pred} \cap \text{Target}|}{|\text{Pred} \cup \text{Target}|}$$

[1]The number of channels is computed as the numbers of channels in RGB (3) plus the number of concatenated masks: one in case of amodal occlusion, two in case of amodal segmentation.

Where $Pred$ is the predicted mask and $Target$ is the ground truth mask. IoU is a widely accepted metric in image segmentation because it balances both false positives and false negatives, offering a robust view of the model's performance across different objects and scenes. It helps in understanding how well the predicted regions align with the actual regions.

The Dice Coefficient is another measure of overlap, similar to IoU, but it is particularly sensitive to the presence of small objects and regions within the segmentation. It calculates the overlap between the predicted and ground truth masks, considering the size of the overlap relative to the total number of pixels in both masks.

$$\text{Dice Coefficient} = \frac{2 \times |\text{Pred} \cap \text{Target}|}{|\text{Pred}| + |\text{Target}|}$$

The Dice Coefficient is particularly useful for ensuring that even finer details are accurately captured, making it a valuable metric for applications where precise boundary delineation is crucial. It is sensitive to small objects, which helps in maintaining the integrity of segmentation in complex scenes.

Inverse Mean IoU provides a different perspective by focusing on the errors made by the model. It is calculated as one minus the mean IoU, highlighting the proportion of the segmentation that is not accurate.

$$\text{Inverse IoU} = 1 - \text{IoU}$$

Inverse Mean IoU helps in understanding the areas where the model's predictions are lacking, offering insights into the errors and guiding further improvements. By focusing on what the model gets wrong, it can help in diagnosing and addressing specific weaknesses in the segmentation process.

The results are reported in the evaluation tables:

- Amodal Occlusion Evaluation (Table 2): focuses on evaluating the amodal occlusion model.
- Amodal Segmentation Evaluation (Table 3): evaluates the amodal segmentation model using the original test set.
- Overall Pipeline Evaluation (Table 4): the main table for out comparison, assesses the entire pipeline, evaluating the amodal segmentation model using the test set modified with the predicted occlusion boundaries from the amodal occlusion model.

To thoroughly evaluate the effectiveness of our proposed pipeline, we have included three distinct tables, each corresponding to different components and stages of the amodal shape completion process. These tables are designed to separately assess the performance of the individual components as well as the overall pipeline.

Including these three tables is vital for a few reasons. Firstly, evaluating each component separately helps identify specific areas that may require improvement. For instance, if the occlusion mask prediction shows lower performance, efforts can be focused on enhancing this model without altering the entire pipeline. Similarly, analyzing the amodal segmentation on the original test set allows for targeted refinements in that area.

Evaluating the components separately and together serves a dual purpose. Firstly, it allows us to identify the strengths and weaknesses of each model individually. If one component significantly underperforms, targeted improvements can be made without altering the entire pipeline. Secondly, it provides a clear picture of how each step contributes to the final outcome, ensuring that any enhancements are data-driven and effectively address the specific needs of each component.

By presenting these evaluations, we ensure that any potential improvements can be systematically approached, either by refining the occlusion prediction model, enhancing the segmentation model, or optimizing the interaction between the two within the pipeline.

Model	Mean IoU	Mean Dice Coefficient	Inverse Mean IoU
UNet	0.44	0.56	0.56
CNN	0.27	0.39	0.73

Table 2: Performance metrics for amodal occlusion.

For the amodal occlusion task, although it was not necessary to achieve outstanding results in terms of Mean IoU and Inverse Mean IoU, it is important to verify that the UNet-based model achieved better results compared to the

CNN-based model. The UNet model also outperformed in terms of the Mean Dice Coefficient, which was included to capture the finer details of the contours, an important aspect in this step.

Model	Mean IoU	Mean Dice Coefficient	Inverse Mean IoU
UNet	0.84	0.90	0.16
CNN	0.80	0.88	0.20

Table 3: Performance metrics for amodal segmentation.

Regarding the amodal segmentation task, the results are slightly better for the UNet model, indicating a higher potential for solving the problem. Nevertheless, the CNN model also shows satisfactory results. The UNet model's slight edge in performance metrics suggests it may be promising in handling the nuances of amodal segmentation.

Model	Mean IoU	Mean Dice Coefficient	Inverse Mean IoU
UNet	0.75	0.84	0.25
CNN	0.68	0.79	0.32

Table 4: Overall performance metrics.

The overall evaluation of the entire pipeline demonstrates that the UNet-based model performs better than the CNN-based model in both the overlap of the masks and in recognizing the details of the contours. However, it is worth noting that even the UNet model is limited and does not perfectly solve the problem of amodal segmentation. Despite this, the superior performance of the UNet model suggests it is a more robust choice for this application. More qualitative results are in the appendix.

6 Conclusion

6.1 Conclusions and Limitations

In this study, we investigated the challenging task of amodal shape completion using a novel two-step approach that integrates UNet and CNN architectures. Our main findings indicate that the proposed methodology effectively addresses the complexities of predicting the occluded parts of objects, demonstrating notable performance on the COCOA dataset.

The two-step approach proved convincing, especially the UNet-based approach, allowing a fair reconstruction of occluded objects. Moreover, despite the constraints of using limited resources on Google Colab, the models achieved results that exceeded initial expectations, showcasing the robustness of our approach even under resource limitations.

Considering the limitations, the reliance on Google Colab limited the extent of our experiments and exploration. With more powerful computational resources, further improvements in model performance could be anticipated. Moreover, the lack of reproducible experiments in the existing literature posed a significant challenge, forcing us to build upon ideas from previous studies rather than following established benchmarks.

6.2 Future Work

These experiments can serve as a starting point for further interesting experiments. Underlying these could be the use of a more powerful architecture to allow longer trainings with fewer constraints. In addition, despite the hard work in making the dataset consistent and usable for our task, it might be worth exploring other datasets, larger and better annotated than the COCOA dataset (still chosen for its simplicity and relative small size).

Considering our approach, it might ideally be possible to train a model (based on UNet, CNN or other) that instead of predicting the occlusion mask, directly predicts the occlusion boundary. Also, another possible approach might involve training amodal segmentation directly on the predicted occlusion boundary, rather than using those in the dataset and then using the model for prediction.

Finally, one might consider using other advanced techniques for two-step training, such as Generative Adversarial Networks, or transformer-based models, if not merging, using these models, the two training steps into one.

References

[1] Guanqi Zhan, Weidi Xie, and Andrew Zisserman. A tri-layer plugin to improve occluded detection. *arXiv preprint arXiv:2210.10046*, 2022.

[2] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[3] Khoi Nguyen and Sinisa Todorovic. A weakly supervised amodal segmenter with boundary uncertainty estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7396–7405, 2021.

[4] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3784–3792, 2020.

[5] Guanqi Zhan, Chuanxia Zheng, Weidi Xie, and Andrew Zisserman. Amodal ground truth and completion in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28003–28013, 2024.

[6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[8] Ziheng Zhang, Anpei Chen, Ling Xie, Jingyi Yu, and Shenghua Gao. Learning semantics-aware distance map with semantics layering network for amodal instance segmentation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2124–2132, 2019.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[10] Ke Li and Jitendra Malik. Amodal instance segmentation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 677–693. Springer, 2016.

[11] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1464–1472, 2017.

[12] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2019.

[13] Minh Tran, Khoa Vo, Kashu Yamazaki, Arthur Fernandes, Michael Kidd, and Ngan Le. Aisformer: Amodal instance segmentation with transformer. *arXiv preprint arXiv:2210.06323*, 2022.

[14] Ege Ozguroglu, Ruoshi Liu, Dídac Surís, Dian Chen, Achal Dave, Pavel Tokmakov, and Carl Vondrick. pix2gestalt: Amodal segmentation by synthesizing wholes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3931–3940, 2024.

[15] Yuting Xiao, Yanyu Xu, Ziming Zhong, Weixin Luo, Jiawei Li, and Shenghua Gao. Amodal segmentation based on visible region segmentation and shape prior. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2995–3003, 2021.

[16] Bowen Zhang, Qing Liu, Jianming Zhang, Yilin Wang, Liyang Liu, Zhe Lin, and Yifan Liu. Amodal scene analysis via holistic occlusion relation inference and generative mask completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6997–7005, 2024.

[17] Jianxiong Gao, Xuelin Qian, Yikai Wang, Tianjun Xiao, Tong He, Zheng Zhang, and Yanwei Fu. Coarse-to-fine amodal segmentation with shape prior. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1262–1271, 2023.

Appendix

A More Qualitative Amodal Prediction Examples

Figure 2 displays amodal masks predicted by UNet architecture. Figure 3 displays amodal masks predicted by CNN architecture. Figure 4 displays a comparison of the masks produced by the two proposed models.

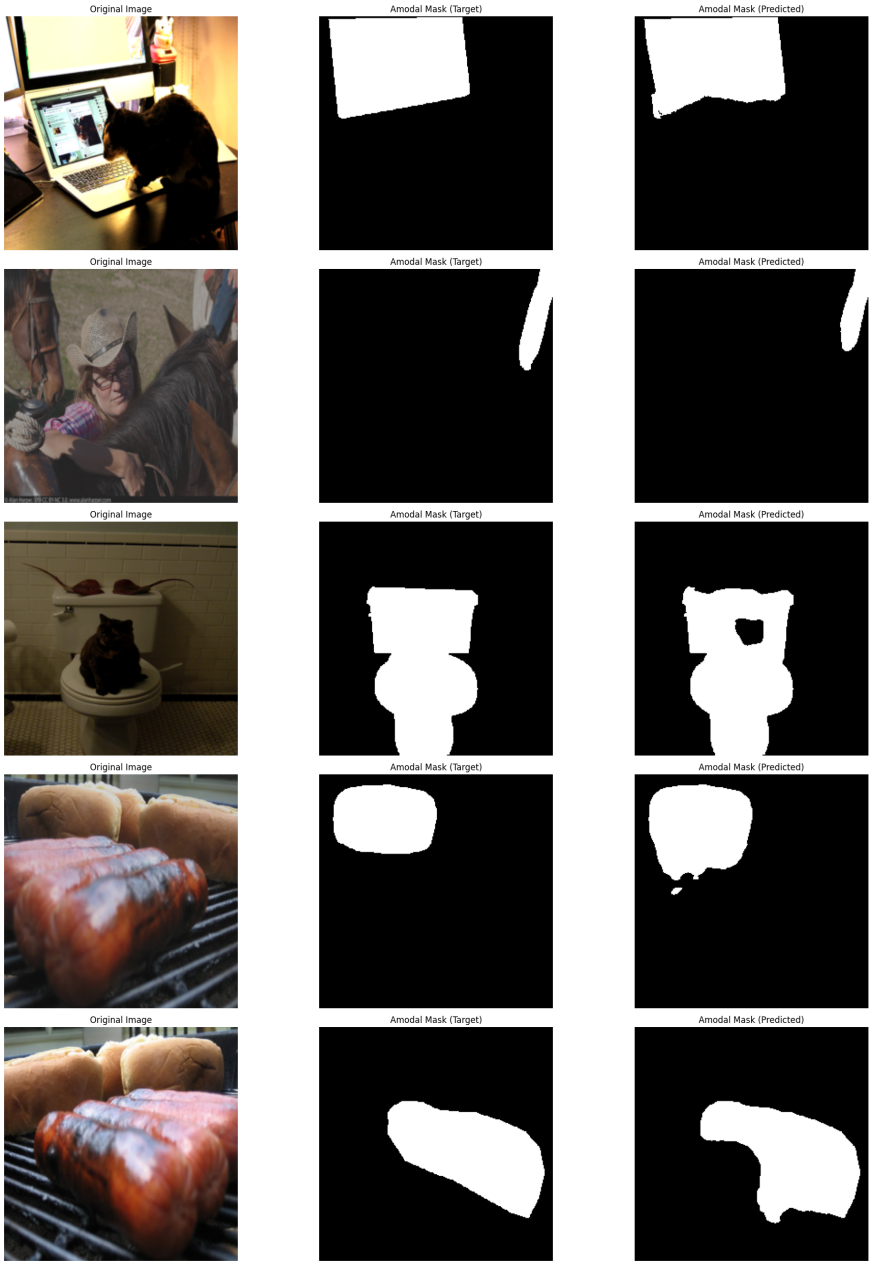

Figure 2: UNet Qualitative Examples

Figure 3: UNet Qualitative Examples

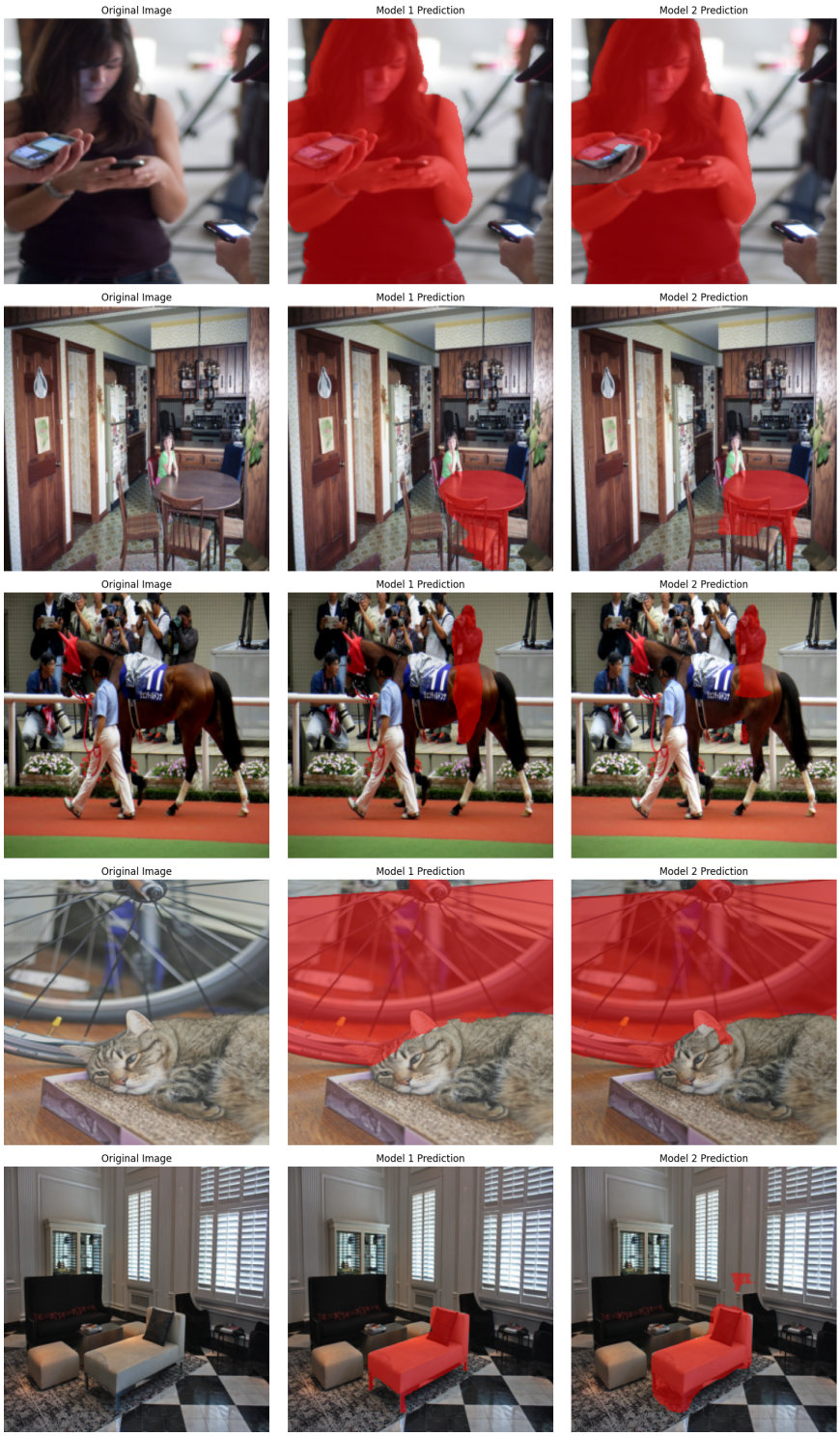

Figure 4: A qualitative comparison