Semantic Scene Composition for Amodal Instance Segmentation

Mihir Parmar, Jie Min, Tejas Srivastava, and Qiao Han

(mihir98, minjie, tjss, hanqiao) @ seas.upenn.edu

University of Pennsylvania, Philadelphia, PA

Abstract

Visual recognition tasks such as object detection, classification and semantic segmentation, are now almost approaching human levels performances due to rapid rate of progress in recent years. However, the task of Amodal instance segmentation, which is to predict the region encompassing both visible and occluded parts of objects in an image, is still far for reaching maturity. Human beings learn and have this innate ability to imagine what the complete objects look like, even if objects might be half occluded. Amodal instance segmentation lets the model act in a similar way, by imagining beyond the visible pixels and providing complex reasoning about full scene structure. But a major roadblock in this task so far has been the lack of a dataset with annotations for amodal instances. In this work, we propose an architecture capable of generating scene composition by placing provided individual objects in the scene, in a context aware manner. We do this by learning a projection matrix based on encoded geometry and semantic information, and use a discriminator loss for training it to produce realistic compositions. Using this pipeline, we would get free amodal mask even in cases where objects are half occluded without human labelling, which can be further used for several other tasks such as scene inpainting of occluded part of the objects or other related segmentation tasks.

1. Introduction

In recent years, visual recognition tasks such as image classification [12, 9], object detection [18, 6], edge detection [2, 4] and semantic segmentation [19, 15] have witnessed a dramatic progress. This has been driven by the availability of large scale image datasets coupled with a renaissance in deep learning techniques with massive model capacity and availability of high computational power at cheaper prices. And with this pace, it can be certainly be said that techniques for many of these tasks is rapidly approaching human levels of performance.

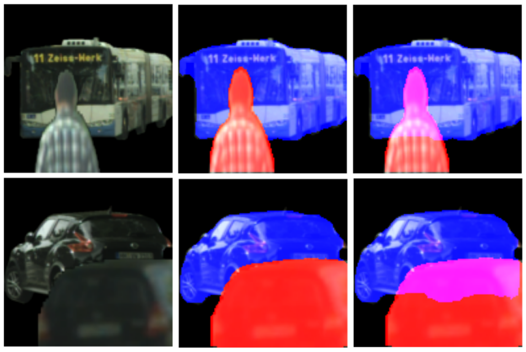


Figure 1: Example of amodal instance segmentation. The left-most column shows the RGB images with occlusion, the middle column shows the modal segmentation performed on the images. The right-most column show the task of amodal segmentation on the images.

Another remarkable property of human perception is the ease with which our visual system interpolates information not directly visible in an image. A particularly prominent example of this, and one on which we focus, is amodal perception: the phenomenon of perceiving the whole of a physical structure when only a portion of it is visible. Humans can readily perceive partially occluded objects and guess their true shape. However, the problem of amodal perception in computer vision is not yet fully explored due to the fact that the data required to train models for these problems is very sparse. Currently, the datasets which do provide amodal annotations in scenes all have been achieved via manual annotations. While manual annotations by humans is not uncommon, specifically for annotati

Thus obtaining masks of occluded parts of objects in an image is also a challenging task and current methods being employed include human intervention to annotate and complete the masks. Also, masks of objects in an image are often required in computer vision tasks such as amodal instance segmentation and scene inpainting, thus obtaining the masks of objects in an image is an important task and cannot be ignored. Hence, in this project we try to automate the process of obtaining the masks of objects in images by composing and generating new scenes from images of individual objects, which are look realistic and semantically correct.

Thus, the main objective of the project is to generate realistic scenes with occlusions in them, from given cropped images from a dataset. Alongwith obtaining the generated compositions with occlusions, as a result we would also obtain free annotation masks for the objects in the scene, even for the occluded portions of the objects. We believe the compositions generated using this method can supervise the mainstream detection frameworks, segmentation frameworks, and edge detection to perform better or help improve any other related tasks since such tasks require understanding object interactions and reasoning about occlusion.

This would be quantified by the performance of the algorithms on our dataset by using a simple metric that focuses on the most salient aspect of our dataset, which is the amodal nature of the segmentation.

2. Related Works

Most large-scale visual recognition datasets facilitate recognizing visible objects in images. ImageNet[3] and Open-Images[11] are used for classification and detection without considering objects precise mask. Meanwhile, segmentation datasets are built to explore the semantic mask of each object in the pixel level. Pascal VOC[5], COCO[14] and ADE20K[20] collect a large number of images in common scenes. KITTI and Cityscapes[16] are created for specific street scenarios. Although widely used in computer vision, these datasets do not contain labeling of invisible and occluded part of objects, thus cannot be used for amodal understanding.

There has been very little work exploring amodal completion due to the lack of publically available amodal segmentation annotations.

Zhu et al. in Semantic Amodal Segmentation [21] proposed an amodal segmentation of where human annotators outline and name all salient regions in the image and specify a partial depth order. The result is a rich scene structure, including visible and occluded portions of each region, figure-ground edge information, semantic labels, and object overlap.

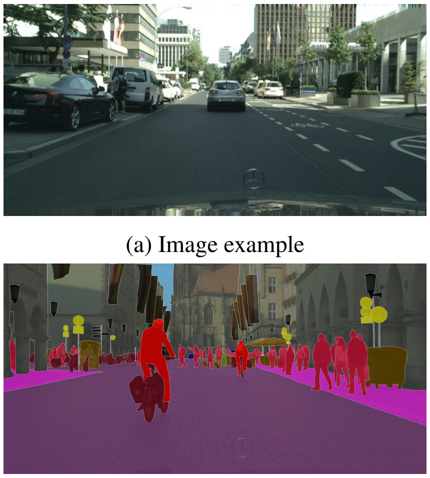
Malik et al. in Amodal Instance Segmentation[13] proposed a novel method for amodal instance segmentation which used a synthetic dataset formed by generating synthetic amodal instance segmentation data from standard modal instance segmentation annotations. They generated composite patches by overlaying randomly cropped object instances on other images on top of the main object. The main objects were generated by cropping image patches that overlap with at least one foreground object instance. For each patch, the pixels belonging to the object are labelled as positive, the pixels belonging to the background are labelled as negative and pixels belonging to the other object as unknown. This process ensures that original modal mask is not affected by the overlaid objects. While in this work, they were able to achieve good results both qualitatively and quantitatively, the method of dataset preparation requires manual annotating of objects which makes it difficult to scale the method for large examples.

Lee et al., in Context-aware synthesis and placement of object instances[1] proposed a novel technique in which they constructed an end-to-end trainable neural network which can coherently place an object in a semantic map. Their method uses two networks consisting of two generative modules where one determines where the inserted object mask should be (i.e., location and scale) and the other determines what the object mask shape (and pose) should look like. However, a major drawback of this technique is that it doesn't demonstrate successful execution of the case when we want the placed object(s) occluded and at the same time want to preserve the entire mask of the occluded object(s).

3. Generating Training Data

3.1. Prepossessing and building data

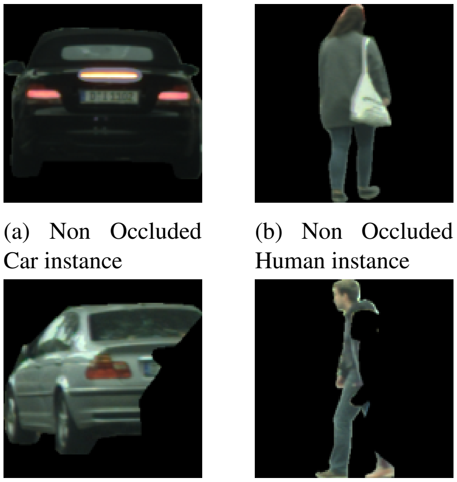
Most recognition datasets such as Cityscapes, COCO [14] already have the ground-truth labels of object classes in the dataset along with the bounding box regions for each of the objects in the images. For this work, we utilize the Cityscape datatset which provides a diverse set of stereo video sequences of street scenes from 50 different cities. The dataset consists of complex scenes with mutiple occlusion situations with different occlusion ratios as well different poses of the object classes. The dataset provides both semantic and instance-level annotation of the scenes. For our use case, we used the instance masks to crop out from Cityscape scenes and created full object instances as well occluded object instances as shown in fig 2 and fig 3 respectively. Even though Cityscape has mask annotations, we used a Mask-RCNN pre-trained for modal segmentation task and obtained segments for the objects. This was done to show that our approach is generalizable to any set of images and need not have any ground truth annotation.



(a) Image example

(b) Annotation example

Figure 2: Image and annotation example from Cityscape dataset



(a) Non Occluded Car instance

(b) Non Occluded Human instance

(c) Occluded Car Instance

(d) Occluded Human instance

Figure 3: Cropped instances of objects

3.2. Labeling Occluded versus Full(Non Occluded) object instances

The obtained segmented images of individual objects had both occluded and non occluded object instances as they were obatined from modal segmentation, and since one of the major applications of our generated images was to improve the amodal tasks, such as amodal segmentation, and appearance recovery, thus we felt it was imperative that the model knows the difference between occluded and non occluded images of individual objects. Previous approaches being employed to solve similar amodal problems do not explicitly use this information of a segmented object being occluded or not, and implicitly learn this information to predict its amodal segmentation. However, in our approach since we tackle this problem by creating situations of occlusions, we needed our model which creates these scenes, to know how a scene with occlusion looks like. Thus, from the segmented images of objects obtained from the dataset in the previous step, we manually labelled them as occluded and non occluded images of objects, and further trained a Resnet-50 classifier, with over 90 percent classification accuracy to confirm that, the labels (occluded or non occluded) of these segmented objects can be trusted upon.

The actual inputs to our model are these segmented non occluded images, which are used to create the compositions, whereas the segmented occluded object images are used to check if the scenes created, have occlusions or not.

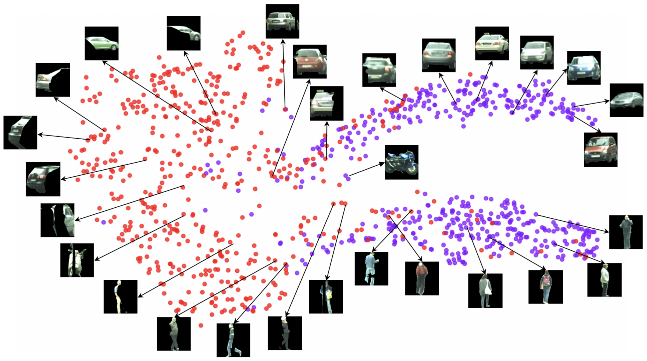


Figure 4: Occluded object versus full object Decision boundary

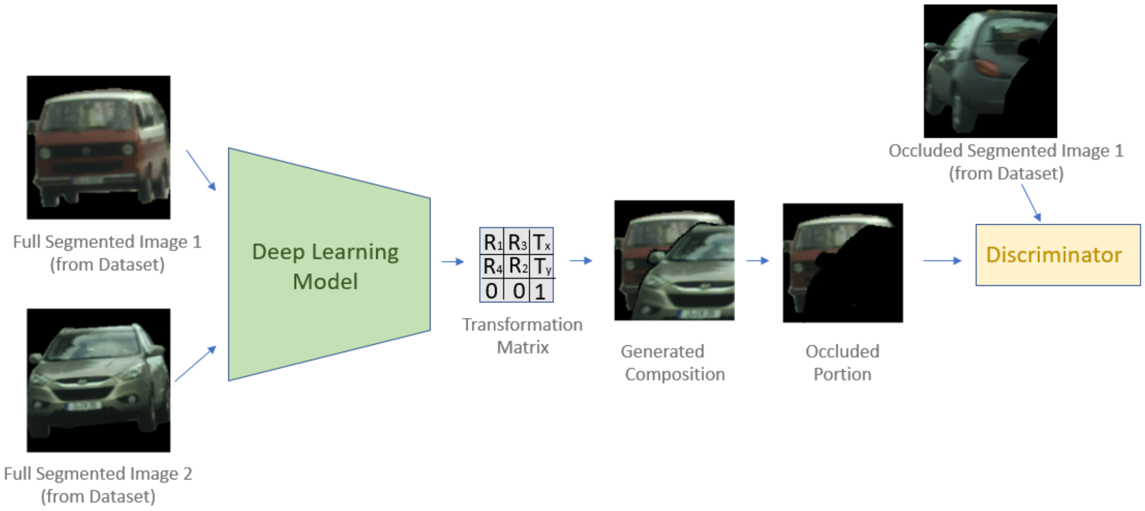
4. Methods

The main objective of our frameworks is to use complete (non occluded) single object instances to compose realistic scenes with occlusion, which also give us their complete ground truth masks, which could further be used to improve and assist amodal tasks such as completing amodal masks, and portion of occluded object images.

Since ours is a novel approach, we had to come up with our own simple baseline methods, to compare the results of our main approach.

4.1. A Non - Deep Learning Baseline

The objective was to come up with a simple way to compose our individual object images in a manner that the compositions looked realistic. While it should be simple, it should still be able to leverage semantic information at some level when creating these compositions. One obvious method for this approach would be to regress the affine matrix values for transformation. But a major roadblock in learning these parameters properly was the lack of a ground truth.



Occluded Segmented Image 1 (from Dataset)
Full Segmented Image 1 (from Dataset)
Deep Learning Model
R₁ R₃ Tₓ
R₄ R₂ Tᵧ
0 0 1
Transformation Matrix
Generated Composition
Occluded Portion
Discriminator
Full Segmented Image 2 (from Dataset)

Figure 5: Basic Pipeline of our Deep Learning Methods

To counter this, we used a small subset of scenes from the Cityscape dataset to find the most common values of rotation, translation along x and y, scaling factors along the same two axes for the objects from our considered classes. After collecting about seven values for each of these parameters, given a pair of object instances, we randomly applied these transformations to one of the objects, and then simply pasted the resultant transformed object on the other (background) object to generate a composite scene. As discussed in the results section, while this method did produce images with both objects present in the scene, a majority of scenes were semantically meaningless. Although very few results did looked realistic doing some justice to this approach of learning an affine transformation to create scene compositions. This motivated further research to come up with better ways of learning these affine transformations.

Affine Transformation: Affine transformations scale, rotate, translate, mirror, and shear images, controlled by six parameters in the affine transformation matrix. They transformations are analogous to printing on a sheet of rubber and stretching the sheet's edges parallel to the plane. This transform relocates pixels requiring intensity interpolation to approximate the value of moving pixels, bicubic interpolation is the standard for image transformations in image processing applications.

4.2. Towards Deep Learning based architectures

The architectural pipeline, as shown in fig 5 for both our deep learning baseline and our novel approach are similar, but vary in terms of the architectural components used for the tasks within the pipeline.

The basic pipeline takes in two complete (non occluded) segmented images, and uses deep learning based approaches in order to predict a transformation matrix for the given images. This transformation is applied on one of the object images (considered foreground object) and is combined with the other object image (background object) to create a scene with the background being partially occluded. In order to learn both, the appropriate values of the transformation matrix (more details later) for a pair of objects, and to check whether the scene created has some occlusion or not, we take the cropped out visible part of the background from the composed scene and pass it to a discriminator. The discriminator essentially classifies if the scene has occlusion or not. It takes two inputs, the first one being the occluded portion of the background from our composed scene and a second input which comes from actual occluded images of objects from our segmented dataset; thus trying to learn if the input from our composition matches the actual occluded input. The objective of the first half of the network is to come up with realistic occlusion situations to fool the discriminator into classifying them as real. To achieve this, the loss from discriminator is backpropagated to better estimate the affine matrix values. The overall architecture is inspired from the Generative Adverserial Network [7] architecture.

The individual architectural details of the Deep Learning Models used for feature extraction from images and for generating values of transformation matrix are different in both deep learning baseline and our novel approach.

4.3. Basic Deep Learning Approach - Deep Learning baseline

Our Deep learning baseline takes segmented objects as inputs, flattens the images to obtain feature vectors, and

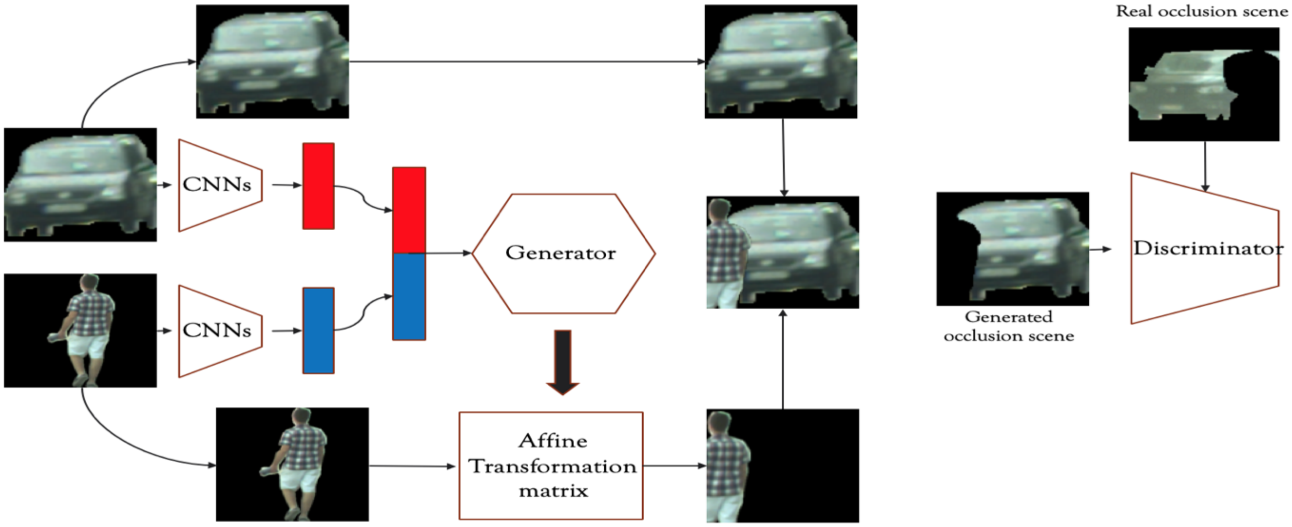


Figure 6: Semantic reasoning composition GAN pipeline: Our Method

adds the vectors together. Next, a multilayer perceptron is used to learn six parameters of the affine matrix, which is then applied to the foreground object and a scene is composed. The rest of the architecture (the discriminator) remains the same as discussed in the basic pipeline.

The compositions obtained from this approach were better than those obtained from the non Deep Learning Baseline approach, since it does take into account the classes which are being given as input to the network to calculate the affine matrix, whereas the simple non Deep Learning baseline, does not take that into consideration and randomly selects the parameters for tr out of the few best ones. However, we realized that we could further improvise this approach by trying to take into factor the spatially shared features and semantic information from the images, and the relation between a pair of foreground and background object images passed as input. We addressed these factors in proposed our advanced deep learning based approach, which is also our main novel approach.

4.4. Improvised Deep Learning Approach: Our Method

Our novel method is still based on the basic deep learning pipeline discussed in the previous sections, but leverages a comparatively advanced deep learning model for feature extraction and affine matrix estimation. The overall architecture is shown in fig 6.

4.4.1 Feature Extractor

The first part of our method is similar to a composition GAN architecture. We randomly sample two complete (non occluded) images I_x and I_y from the segmented dataset S which are given as inputs to our network. Consider I_x to be the foreground object, for example, the image of the man in Figure 6, and I_y as the background object, some portion of which will be occluded, the car in our diagram. Next in our network are two convolutional encoders, which are used to extract meaningful features and information from each of the input images I_x and I_y but separately. We decided to keep two separate encoders for the foreground and background object, since the semantic information which we require from the foreground object is different from the information we need from the background object. Thus, we obtain two feature vectors,

$$Z_x = E_x(I_x)$$

$$Z_y = E_y(I_y)$$

as the output of these encoder architectures. These feature vectors Z_x and Z_y are now concatenated to combine the information extracted separately from the foreground and the background object, thus combining them into a single embedding, Z, which is the output of our feature extractor, and is then passed on to the next part of our model, the STN based GAN.

4.4.2 STN based Generative Adversarial Network

Spatial Transformer Networks: Spatial Transformer Networks(STN)[10] have been widely used in many previous works for the spatial manipulation of data within the network. They can be inserted into existing convolutional architectures, giving neural networks the ability to actively spatially transform feature maps, conditional on the feature map itself, without any extra training supervision or modification to the optimisation process. They are used to esti-

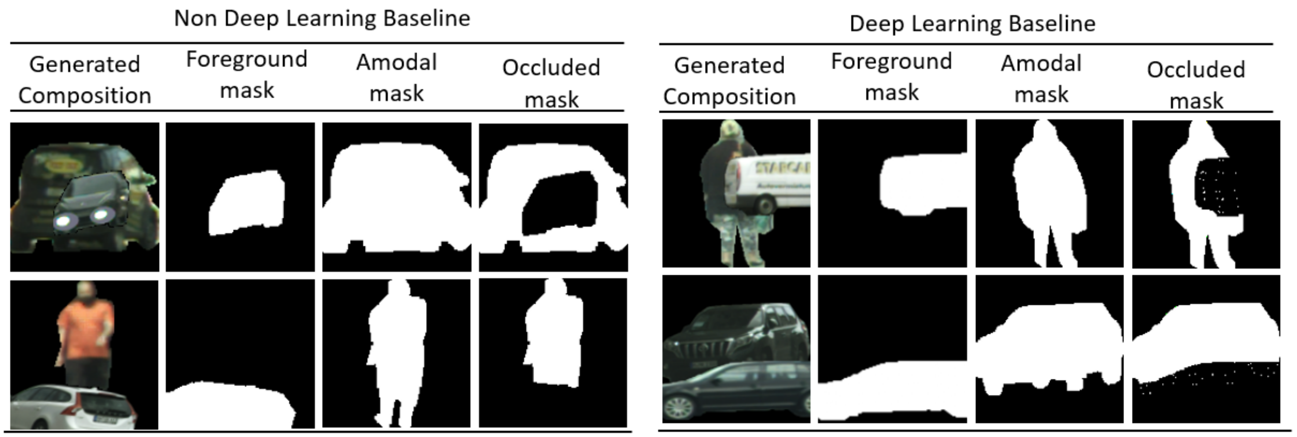


Table 1: Few examples from generated images with the corresponding masks

mate the affine transformation matrix of input feature map and maintain the shift-invariant property of deep networks.

The output of our feature extractor Z, which is a single embedding vector for the pair of input object images is then passed to the Spatial Transformer Network, which consists of some 1- d convolutional layers, followed by some fully connected layers to extract the features and relation between the embeddings of the foreground and the background object to obtain $\hat{Z}$.

$$\hat{Z} = \sigma(...\sigma(W_1^T * \sigma(W_0^T * Z + b_0) + b1) + ...)$$

The values in $\hat{Z}$, which is a six dimensional vector are considered as the rotation and translation parameters of the affine matrix for the current composition. Further, Parametrised Sampling Grid is used to find the corresponding pixel locations after applying this transformation, given the original images and estimated matrix.

Thus, our approach uses Spatial Transfer Network as a generator (Φ), to estimate the affine matrix. This affine matrix is a 3x3 matrix,

$$M_a = [R|t] = \Phi(FCs(Z))$$

Next, this estimated affine matrix M_a is applied to the foreground image I_x, and outputs a composition based on both the objects I_x and I_y, giving

$$\hat{I} = I_y + M_a * I_x$$

The generator thus outputs a generated occlusion scene with knowledge of the ground truth masks of all objects in the scene. The visible cropped portion of the background object in the generated scene is then given as input to the discriminator along with an image of real occluded object from our input dataset and tells if both of these are similar or not (being able to distinguish between generated occluded scenes) and thus forcing the generator to produce a more realistic occlusion scene and a more meaningful composition of objects. The loss function for this setup hence turns into a similar one used in [7]. The discriminator seeks to maximize the average of the log probability for real occluded images and the log of the inverted probabilities of non occluded images.

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right] \tag{1}$$

here, $\boldsymbol{x}^{(i)}$ denotes a real-occluded image and $\boldsymbol{z}^{(i)}$ denotes the generated occlusion scene.

The generator seeks to minimize the log of the inverse probability predicted by the discriminator for occluded images. This has the effect of encouraging the generator to generate samples that have a low probability of being detected as not occluded.

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \tag{2}$$

The entire model architecture including the layers involved in each of the three modules along with the hyperparameters is given in Table 2.

5. Experiments and Analysis

Since there is no existing work based on scene composition which preserves the original masks of object instances even in case of occlusions, a major challenge in this work

Type	Feature Extractor	Discriminator	Generator
Learning Rate	2e-4	2e-4	2e-4
Input Size	32(Batch_Size) * 3 * 128 * 128	32(Batch_Size) * 3 * 128 * 128	32(Batch_Size) * 512 * 8 * 8
Layers	Conv(16,32)->Conv(32,64)->Conv(64,128)->Conv(128,256) ->Conv(256,512) ->Conv(512,1024)->Conv(1024,1024)	Conv(3,32,4,2,1)->Conv(32,64,4,2,1)->Conv(64,128,4,2,1)->Conv(128,256,4,2,1) ->Conv(256,512,4,2,1) ->Conv(512,1024,4,2,1)->Conv(1024,1,1,1)	Conv(512,256,3)->Conv(256,128,3)->Conv(128,64,3) ->Conv(64,32,1) ->Conv(32,16,1) ->FC(16,6)
Output Size	32(Batch_Size) * 256 * 8 * 8	32(Batch_Size) * 1	32(Batch_Size) * 9 (Affine Matrix)
Activation function	ReLU()	LeakyReLU(0.2)	LeakyReLU(0.2)

Table 2: Few examples from generated images using Novel Method with the corresponding masks

was to do a quantitative evaluation of how good the results are. We first provide qualitative results of the images generated by our model.

Next, to get a direct quantitative evaluation, we use the generated scene compositions consisting of amodal masks, and use them as ground truth to train a Mask-RCNN [8] to make it capable of predicting segment for amodal instances. Since predicting amodal segments requires understanding object interaction and reasoning about occlusion, by obtaining high scores on this task and comparing with scores of other existing works on the same task, we demonstrate that our model is able to learn these inductive biases. We report the amodal segmentation performance by reporting the COCO-type mean-Average Precision (mAP) proposed and used in the COCO challenges [14].

5.1. Qualitative Results

In this section of analysis, we report the generated scene compositions using a pair of individual instances of different combinations from the 10 classes previously mentioned in the dataset section. Table 1 shows some examples of the generated images by using our Non Deep Learning baseline and Deep learning baseline approach, along with the corresponding binary mask for foreground object, amodal mask and the occluded partial mask for background object. Table 3 shows the same set of images, but created by our novel deep learning approach.

5.2. Quantitative Results

In this section, we begin by reiterating that the rich quality of images and corresponding ground truth masks generated using our model subsume many mainstream vision tasks such as edge detection, object detection and classification, and instance segmentation. To verify this for the instance segmentation task, we used around 1500 generated images along with the ground truth masks to fine-tune a Mask-RCNN pre-trained on the COCO dataset. We split into 1200/100/200 images for train/val/test set and evaluate on test set.

5.2.1 Evaluation Metric

To evaluate the segment quality, we use the popular metric for object proposal, mean-Average Precision (mAP) used in the COCO challenge [14]. For both the average recall and the average precision, the respective values are computed for multiple IoU thresholds and then averaged. For our analysis, during evaluation, we compute the IoU against:

1. Modal + Amodal masks
2. Modal masks

Additionally, to demonstrate the importance of having amodal masks for training to improve the performance of segmentation network on occluded objects, we train two different models:

(A) Mask-RCNN fine tuned with the ground truth mask for background (occluded) object being the full amodal mask.

(B) Mask-RCNN fine-tuned with the ground truth mask for background (occluded) object being the partial occluded mask.

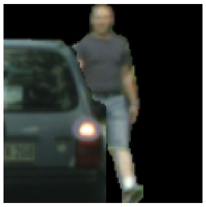
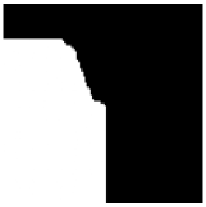
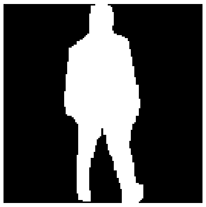
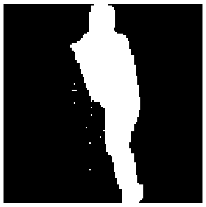
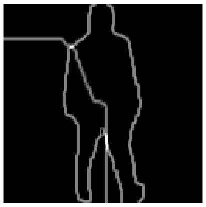
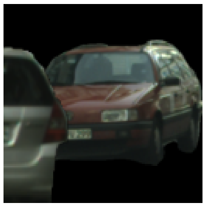
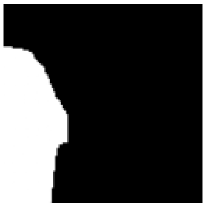
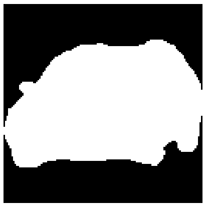
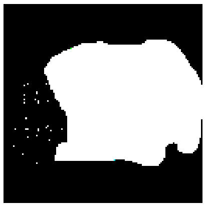
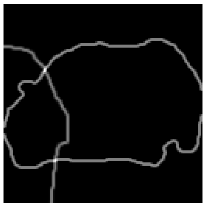
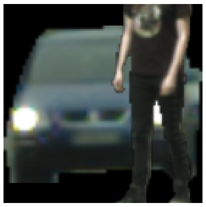
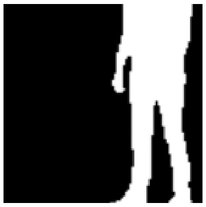
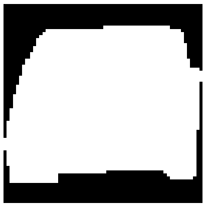
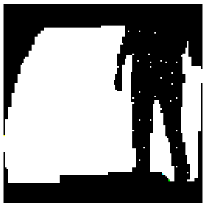
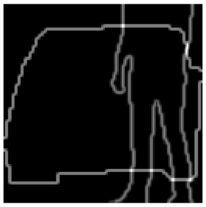
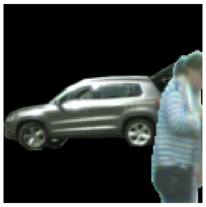
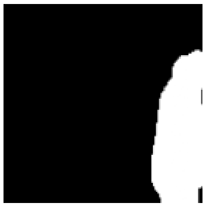
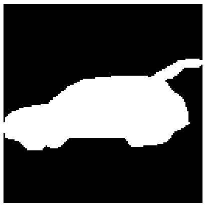
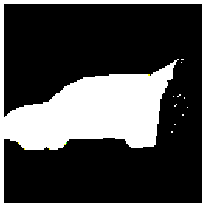
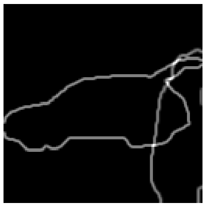
Nr.	Generated Image	Foreground Mask	Amodal Mask	Occluded Mask	Object Edges
1					
2					
3					
4					

Table 3: Few examples from generated images using Novel Method with the corresponding masks

Henceforth, to reduce description, we refer these models as model A and model B respectively.

As per our hypothesis, for the task of predicting segment of occluded objects, the performance of the first model which leverages the full mask of occluded object given by our model should lead to superior performance when compared to the second model which uses only the partial occluded masks.

5.2.2 Segmentation Network - Mask-RCNN

We use Mask-RCNN, current state-of-the-art method for modal class-agnostic object segmentation, pre-trained on the COCO dataset from PyTorch's torchvision framework [17] and perform fine-tuning to fit our dataset. The Mask-RCNN model uses a ResNet-50-FPN backbone with the pre-trained box-predictor head replaced with a new Faster-RCNN box predictor module and the pre-trained mask predictor head replaced with a new MaskRCNN mask predictor module. The number of input-features for the box predictor head is 1024, and the number of input-features for the mask classifier is 256. For this work, since the object instances in the generated compositions are unlabelled, the class labels for all objects were considered to be same. This method of implementation will be incorrect if the end goal was to have a high accuracy on object classification, however, since for this work we are concerned only with the accuracy of mask prediction of the object instances, giving incorrect class labels should have no effect on the metrics reported. Hence, there were two classes one corresponding to background and one for the objects.

5.2.3 Results and Analysis

To demonstrate the superiority of our STN-GAN implementation over the non-DL and DL baselines quantitatively, we first train a Mask-RCNN for amodal instance segmentation task using the dataset generated by the three respective implementation methods.Table 5 compares the mAP and mAR values at different epochs during training and the mAP values on the test set.

The better values obtained by the STN-GAN implemen-

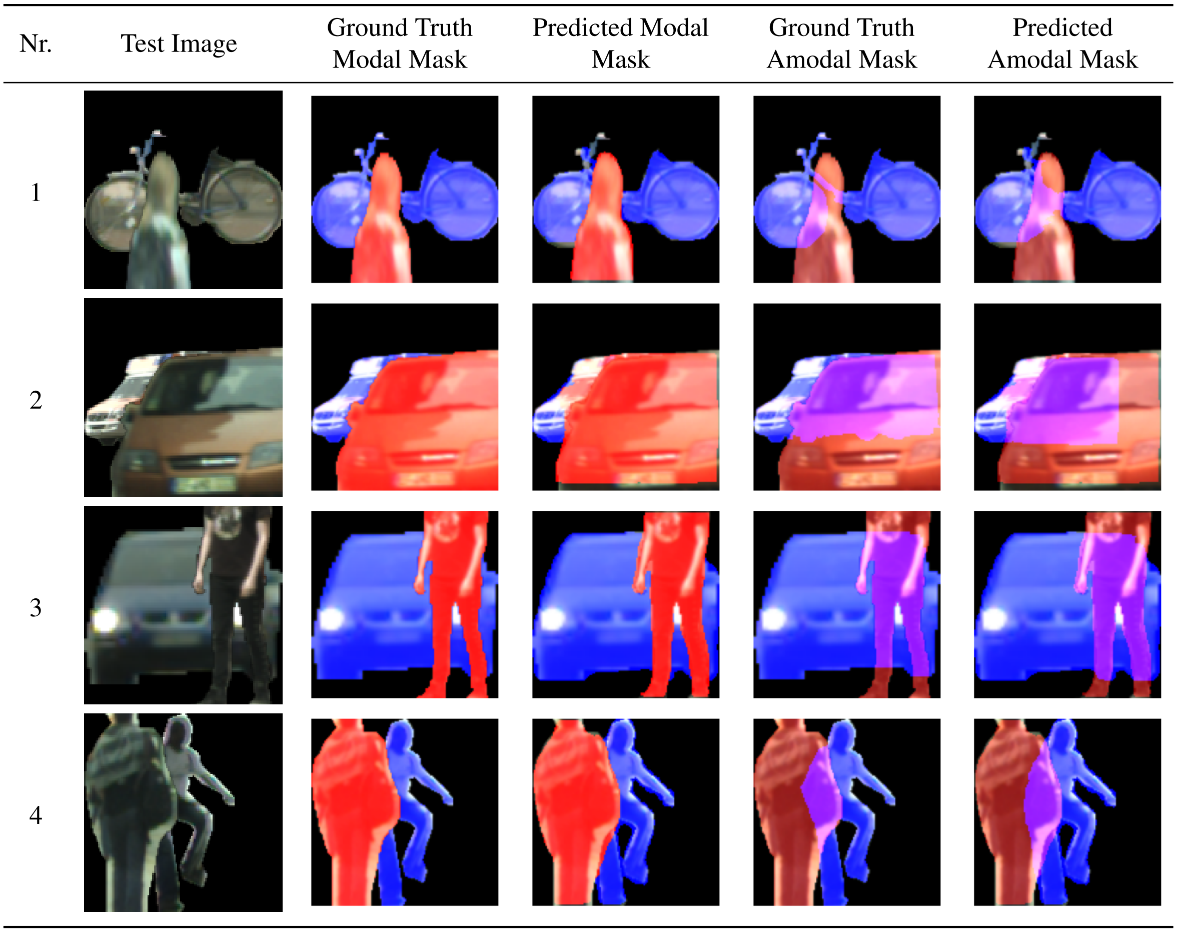


Table 4: Few examples from generated images using Novel Method with the corresponding masks

Mask-RCNN fine-tuned for Amodal Segmentation			
Method	epoch = 3	epoch = 6	epoch= 10
Non-DL	0.662	0.750	0.763
DL	0.701	0.835	0.858
STN-GAN	0.787	0.797	0.803

Table 5: Comparison of mAP scores at different stages of training for Non-DL, DL and STN-GAN method

Mask-RCNN fine-tuned for Amodal Segmentation	
Method	Test dataset score
Non-DL	0.494
DL	0.651
STN-GAN	0.721

Table 6: Comparison of mAP scores on test set for Non-DL, DL and STN-GAN method

tation on the test set as shown in table 6 could be explained by the ability of this method to learn to generate compositions with a better semantic meaning due to the presence of convolutional layers which help extract better spatial features. Since the aforementioned evaluations demonstrate that the STN-GAN method leads to superior result, hereafter in further evaluations reported, we used the data generated by the STN-GAN method as the training data to finetune Mask-RCNN.

While intuitively it is clear and there are already works [21, 13] which have demonstrated that amodal masks are necessary to solve the task of Amodal Instance Segmentation, to further demonstrate it we trained two different Mask-RCNN models. One was trained with amodal masks being included, while other was trained with partial masks of occluded objects. The mAP values obtained during training are reported in table 7

Using STN-GAN generated data			
Method	epoch = 3	epoch = 6	epoch= 10
without Amodal masks	0.689	0.707	0.721
with Amodal masks	0.787	0.797	0.803

Table 7: Comparison of mAP scores at different stages of training of model A and model B

The performance of both models on the test set is as shown in table 8. Mask-RCNN trained on amodal masks outperforms model B with a significant increase in the mAP score. The qualitative results are shown in tabel 4

Using STN-GAN generated data	
Method	Test dataset score
without Amodal masks	0.607
with Amodal masks	0.721

Table 8: Comparison of mAP scores on test set for model A and model B

Next, we try to answer if whether training a Mask-RCNN with amodal masks provided. makes the performance of the model improve on the task of modal segmentation. Intuitively, this would seem justifiable since by providing amodal masks we are providing more semantically meaningful training data to the Mask-RCNN. To verify if this is true, we evaluated the mAP scores obtained by model A and model B with IoU computed only against the modal masks. The results are reported in table **??**

IoU computed against the modal masks	
Method	Test dataset score
without Amodal masks	0.682
with Amodal masks	0.811

Table 9: Comparison of mAP scores on test set for model A and model B with IoU computed only against modal masks

As the results demonstrate, the network does end up improving it's performance on the modal segmentation task.

6. Discussion

In this work, we presented a novel deep learning architecture capable of creating semantically meaningful scene compositions given two object instances. While not restricted to this task, we demonstrated that the resulting image data generated by our network could be used to significantly improve performance of segmentation models like Mask-RCNN on amodal instance segmentation task. Our proposed architecture could have a great potential in many currently growing fields such as autonomous driving wherein different situations of driving scenes could be generated and used to train perception tasks. The motivation of our work is to encourage and improve reasoning about object interactions and scene structure, and amodal perception.

Future work will involve generating scenes with high information in terms of number of objects instances involved as well as incorporating non-rigid objects, geometric features and depth based compositions in world coordinate frame.

7. Acknowledgements

We would like to thank Prof. Konrad Kording and all the other instructors of this course for providing us the platform to work on this project idea. We are especially grateful of Sadat Shaik for providing valuable feedback and suggestion at different stages of the project timeline which helped us refine our methods.

References

[1] Context-aware synthesis and placement of object instances. *Advances in Neural Information Processing Systems*, 2018-December:10393–10403, 1 2018.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[4] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *CoRR*, abs/1406.5549, 2014.

[5] M. Everingham, M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2).

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. pages 770–778, 06 2016.

[10] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[11] I. Krasin. *Openimages: A public dataset for large-scale multi-label and multi-class image classification*.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[13] K. Li and J. Malik. Amodal instance segmentation. In *ECCV*, 2016.

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[15] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[16] S. R. T. R. M. E. R. B. U. F. S. R. B. S. Marius Cordts, Mohamed Omran. The cityscapes dataset for semantic urban scene understanding. *2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] PyTorch. *PyTorch: Torchvision*.

[18] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. 2nd international conference on learning representations, iclr 2014. Jan. 2014.

[19] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision – ECCV 2006*, pages 1–15, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[20] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[21] Y. Zhu, Y. Tian, D. Metaxas, and P. Dollár. Semantic amodal segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1464–1472, 2017.