Shadow Generation for Composite Image in Real-World Scenes Project

Isabella Gomez
University of California, San Diego

1 Project Overview

This project overall aims to generate the shadow of a corresponding object within an image. Given the original image I_g, a composite image I_c, and the composite image foreground object mask M_{fo} generate a prediction of the corresponding foreground object shadow mask and then fill in the area corresponding to the shadow mask. The final output of the process being a new image composed of I_c with a filled shadow area under the foreground object.

2 Method

The methodology of this project is divided into two main parts, Shadow Mask Prediction and Shadow Filling, and an additional data collection section.

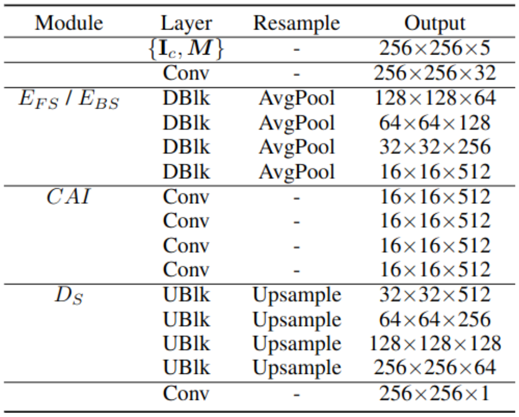
2.1 The DESOBA Dataset

The dataset used for this project is the DESOBA dataset. This dataset belongs to the authors of [1] and is composed of 11,509 training images and 581 testing images. In order to obtain the images, the dataset has to be built from a series of images and binary masks. This can be done using the provided scripts from the authors.

2.2 Shadow Mask Prediction

The Shadow Mask Prediction stage consists of a Shadow Mask Generator G_s. This generator is designed based off of a U-Net architecture. Therefore, it is composed of an encoder E_s, a Cross-Attention Integration Layer (CAI) and a Decoder D_s, where E_s is split into background encoder E_{bs} to capture background details, and foreground encoder E_{fs} to capture the foreground details.

Each encoder is made of 4 Downsampling Blocks followed by an AvgPool layer. Each Block is made up of 1 convolutional layer, followed by Batch Normalization and using the ReLU activation function. On the other hand, D_s is made up of 4 Upsampling blocks. Each of these is composed of a transpose convolutional layer, followed by a convolutional layer with batch normalization and a ReLU activation function. The CAI layer is composed of a total of 4 convolutional layers with spectral normalization. The U-Net model also includes an initial and a final 1x1 convolution in order to put the input and output in the appropriate channel size. The exact architecture can be seen in Figure 1.



Module	Layer	Resample	Output
	$\{\mathbf{I}_c, M\}$	-	256×256×5
	Conv	-	256×256×32
E_{FS} / E_{BS}	DBlk	AvgPool	128×128×64
	DBlk	AvgPool	64×64×128
	DBlk	AvgPool	32×32×256
	DBlk	AvgPool	16×16×512
CAI	Conv	-	16×16×512
	Conv	-	16×16×512
	Conv	-	16×16×512
	Conv	-	16×16×512
D_S	UBlk	Upsample	32×32×512
	UBlk	Upsample	64×64×256
	UBlk	Upsample	128×128×128
	UBlk	Upsample	256×256×64
	Conv	-	256×256×1

Figure 1: U-Net Generator architecture table.

The inputs to the U-Net model are two 256x256x4 tensors which are a concatenation of I_c and M_{fo} which goes into E_{fs} and a concatenation of I_c and M_{bos} for E_{bs}. The final output of the model is $\tilde{M}_{fs}$ which is the foreground object shadow mask, which can now be used in the next stage of the project.

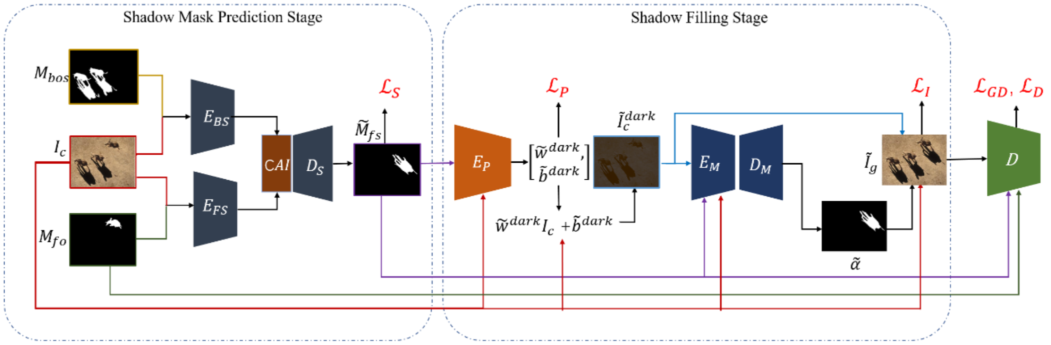


Figure 2: Shadow Generation in the Real-world Network (SGRNet) diagram.

2.3 Shadow Filling

The Shadow Area Filling stage consists of a few main parts, creating $I_c dark$, generating α and producing the output $\tilde{I}_g$. To do this, the first step is to generate the ground truth shadow parameters $\{w_{dark}, b_{dark}\}$ using the ground truth shadow masks and Linear Regression. With those ground truth values, we can use the E_p encoder to estimate the values of the parameters for our generated shadow masks. Then we use the equation $\tilde{w}_{dark} * I_c + \tilde{b}_{dark}$ in order to create $I_c dark$.

$I_c dark$ is then used, alongside I_c and the predicted shadow masks $\tilde{M}_{fs} dark$, as input to the generator G_M, which has the same U-Net architecture as used with G_S. The output to this generator is the shadow matte parameter $\tilde{\alpha}$, which can be used alongside $I_c dark$ in order to create the final image $\tilde{I}_g$. $\tilde{I}_g$ is the image with the generated shadow from our network.

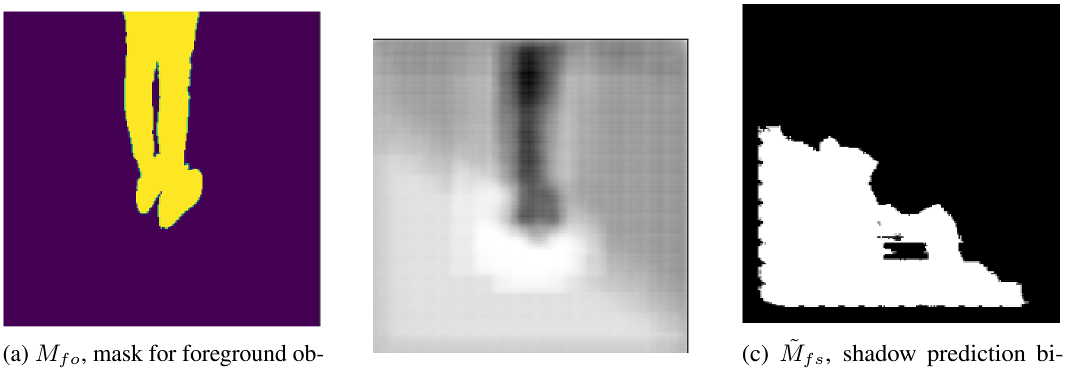
2.4 Results

Examples of the produced images can be seen in the following pages. These figures show the images captured through the process. The most significant of which represent the predicted shadow masks $\tilde{M}_{fs}$, $I_c dark$, and the final output $\tilde{I}_g$.

One issue that arose is that, as a raw output, α **??** had a large border around them. This made α unusable for the computation of $\tilde{I}_g$. Instead, the predicted shadow mask $\tilde{M}_{fs}$ was used, therefore, the color (darkness, lightness) of the shadows will become affected.

References

[1] Yan Hong, Li Niu1, and Jianfu Zhang. Shadow generation for composite image in real-world scenes. `https://arxiv.org/pdf/2104.10338.pdf`, 2022.

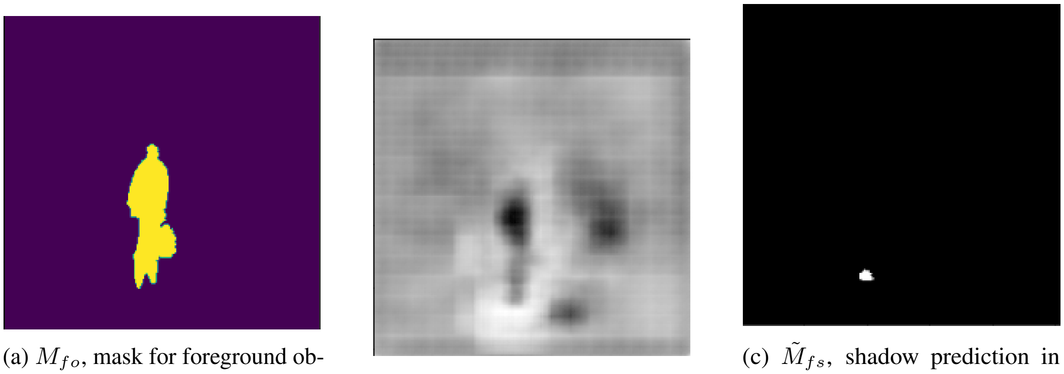


(a) M_{fo}, mask for foreground object

(b) Raw output from U-Net.

(c) $\tilde{M}_{fs}$, shadow prediction binary mask.

Figure 3: Matching composite image mask and shadow prediction.

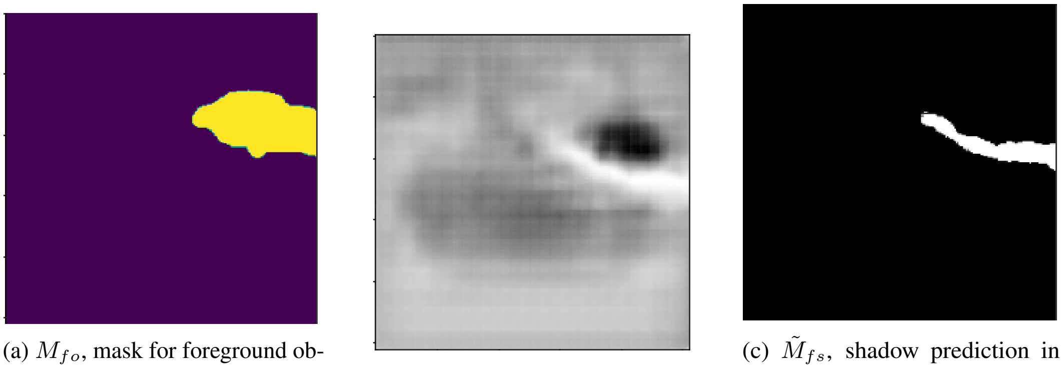


(a) M_{fo}, mask for foreground object

(b) Raw output from U-Net.

(c) $\tilde{M}_{fs}$, shadow prediction in white.

Figure 4: Matching composite image mask and shadow prediction.

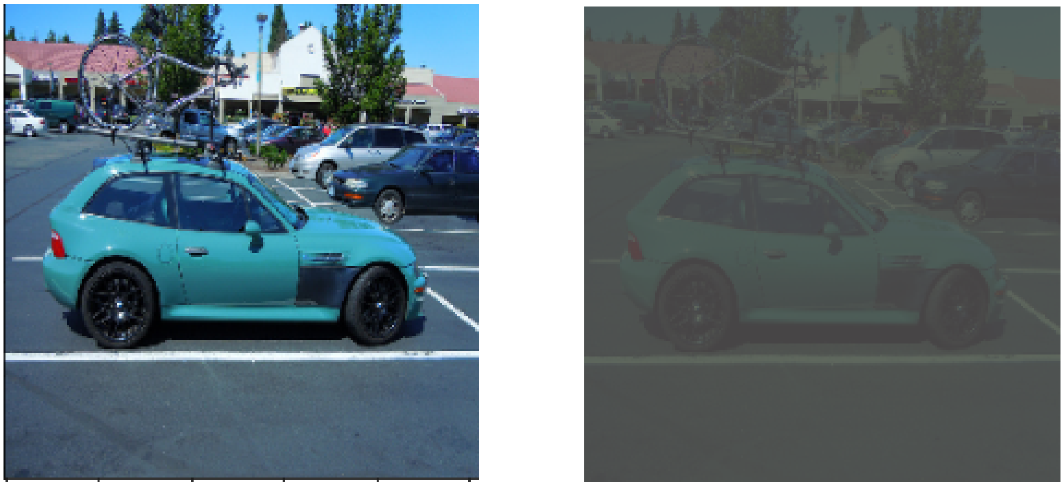


(a) M_{fo}, mask for foreground object

(b) Raw output from U-Net.

(c) $\tilde{M}_{fs}$, shadow prediction in white.

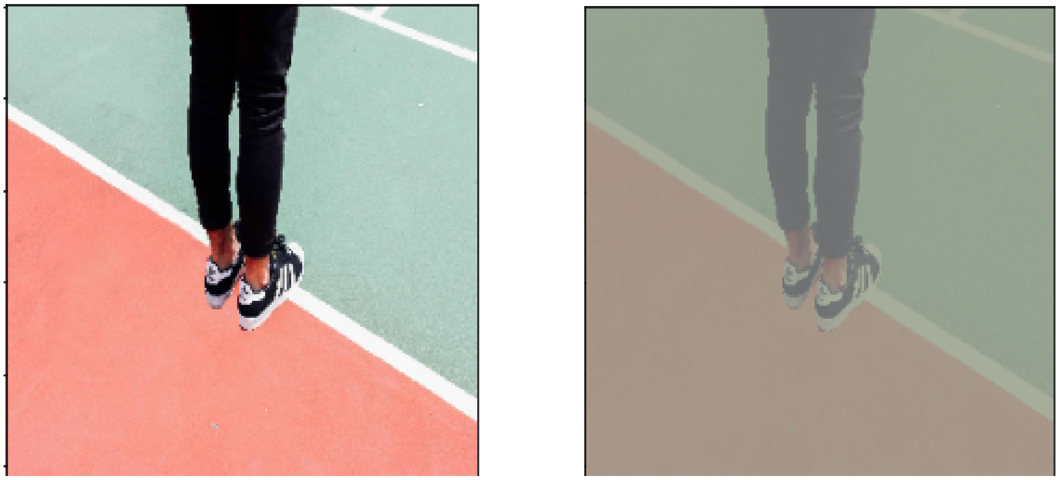
Figure 5: Matching composite image mask and shadow prediction.



(a) I_c, composite image.

(b) $I_c dark$

Figure 6: Composite image with darkened composite image.



(a) I_c, composite image.

(b) $I_c dark$

Figure 7: Composite image with darkened composite image.

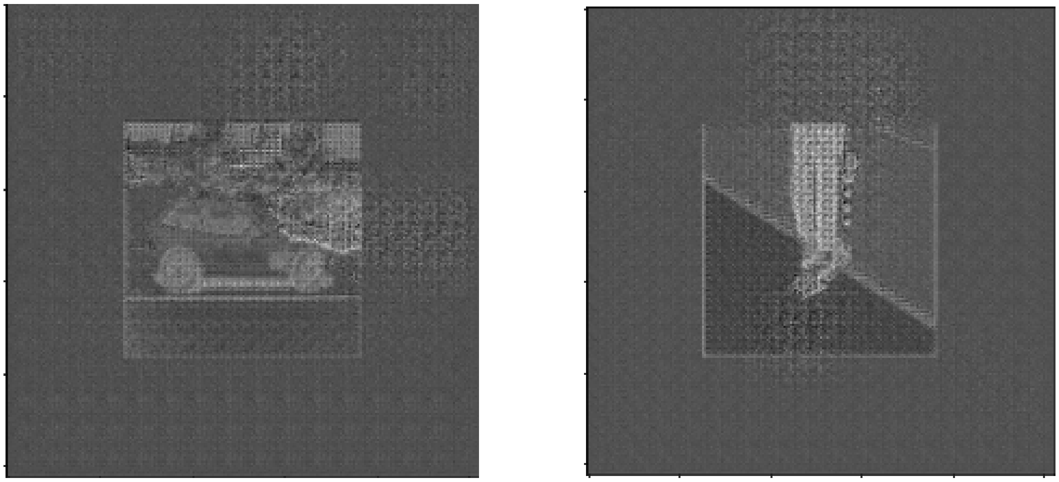


Figure 8: Examples of α

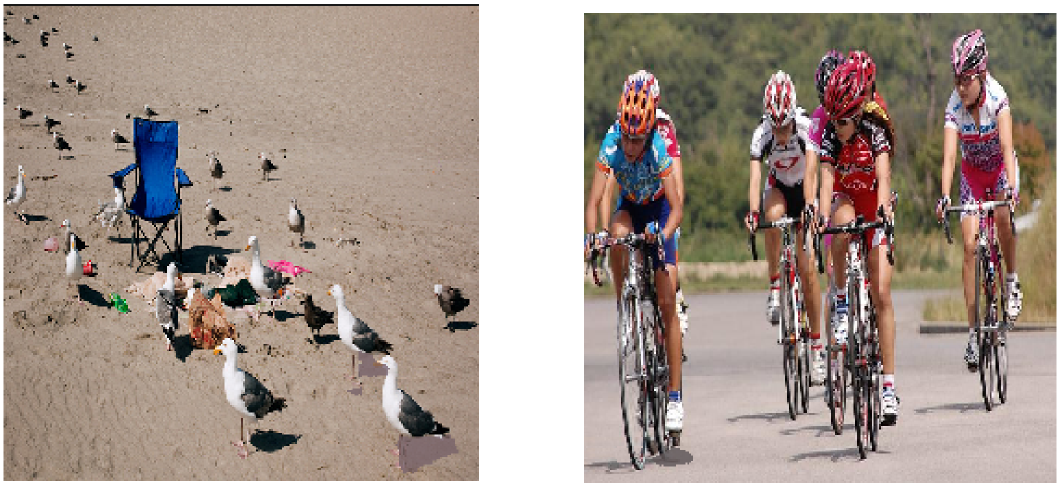


Figure 9: $\tilde{I}_g$ of birds and bicycle scenes.

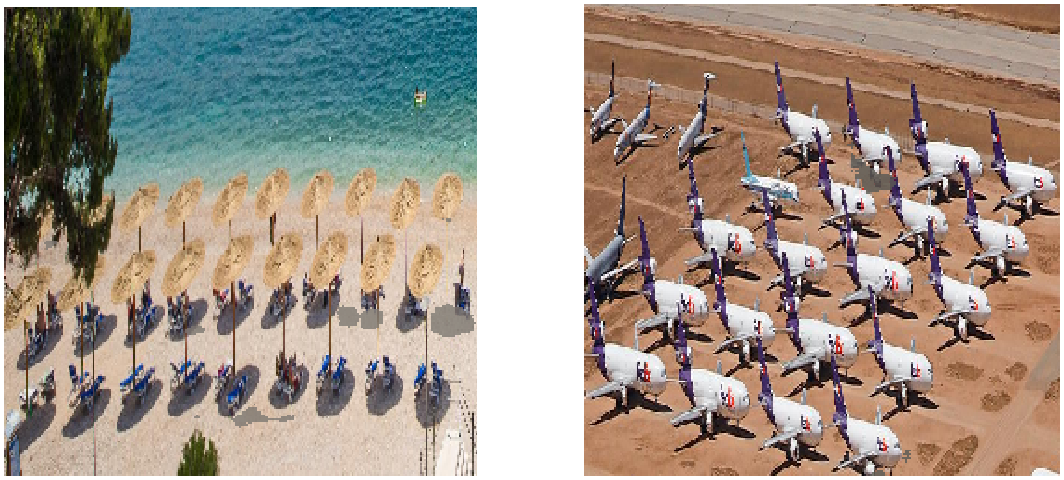


Figure 10: $\tilde{I}_g$ of beach and plane scenes.

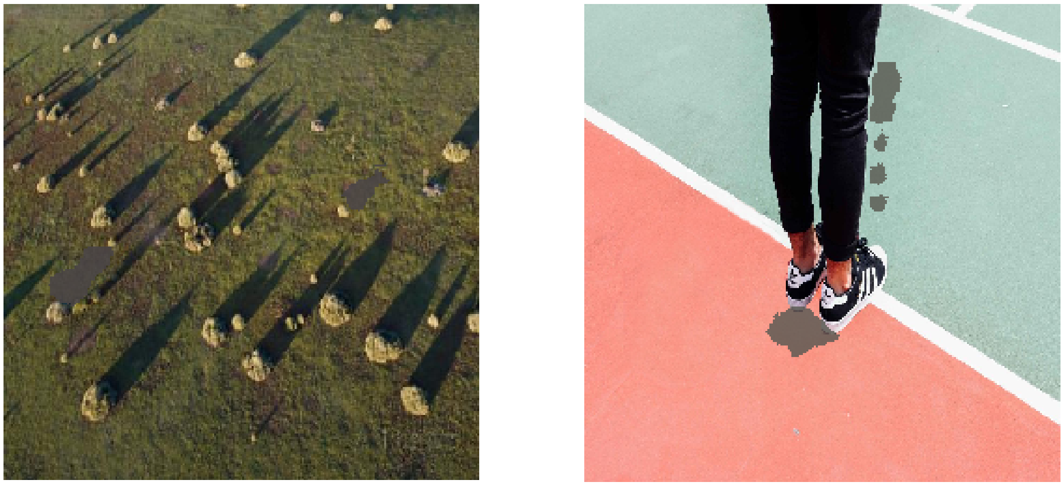


Figure 11: $\tilde{I}_g$ of court and field scenes.