Seminar Computer Vision by Deep Learning
Shadows Detection

Jim Kok
4679970

Dekel Viner
5180929

1 Information about this blog post

Shadow segmentation is the problem in which pixel-wise labeling of shadows/non-shadow in an image needs to be determined. It has use cases in a wide variety of applications (e.g. self-driving) as it can have an impact on perception, and contains powerful environmental cues. Shadow detection can for example aid computer vision systems to better estimate light locations and depth. Estimating depth is for example of great importance for avoiding collisions of self-driving cars. However, shadow detection is found to be difficult in various situations. When looking at pictures that contain shadow it can sometimes be difficult to distinguish shadow from non-shadow regions, even for humans (see for example Figure 1). Currently, there are lots of shadow detector out there, however, we found that this architecture outperformed most of them. The model replicated in this blog post achieves state-of-the-art results, by combining several factors. First of all, it counts the number of shadow regions in the image which can avoid that there are too many noisy dark pixels being classified as shadow. Secondly, it uses the shadow edges to better estimate where the shadow region starts. On top of this, it uses intermediate shadow prediction to better update the network. With this being said, this blog post gives insights into our replication of the Multi-task Mean Teacher model introduced in "A Multi-task Mean Teacher for Semi-supervised Shadow Detection" by Chen et al. [2020]. However, our network turned out to be a bit different from theirs. More details about this will follow in this blog. Finally, This blog post intends to compare our results with the implementation provided by the original authors. Some preliminary results of our work are shown in Figure 2. Our implementation can be found on https://github.com/jimkok9/ShadowPrediction.

In the next sections, we will first provide the necessary information about shadow detection and deep learning models tackling this problem. After this, we discuss the performed experiments, results and a short ablation study that we have conducted. In addition, we give a brief overview of the learning process with the help of learning curves. Finally, we will discuss the visuals and possible future ideas.

TU Delft Seminar Computer Vision by Deep Learning (2020).

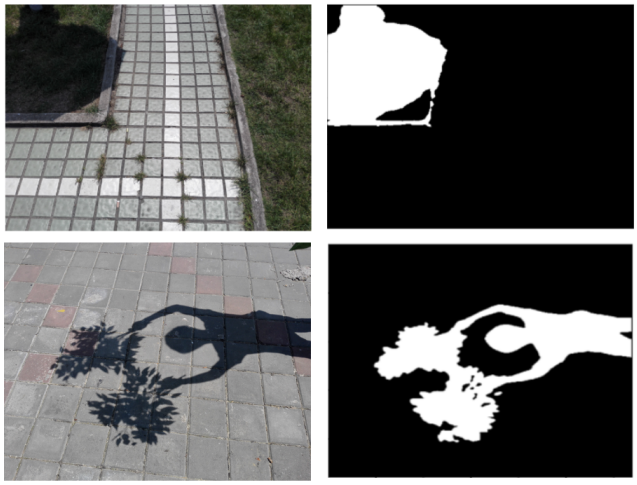

Figure 2: Results ISTD test dataset.

Figure 1: Image in SBU training dataset.

2 Shadow detection using the MTMT-model

2.1 Shadow detection and deep learning

As already mentioned in the previous section shadow detection is about recognizing shadow regions in an image. Shadows can contain lots useful information about the shape of objects, distance cues, light directions, scene geometry etc. One of the reasons that shadow detection is such a difficult problem has to do with the fact that it can be difficult to attain large labelled datasets that would be required to cover the huge variability that may be present in shadowed images. For instance, some images may have dark regions that can easily be mistaken for shadows by a deep net or even a human. Therefore, automatic shadow recognition is very challenging for a computer. Numerous models have been designed over the years, most utilize CNN based architectures and while they have improved upon the classical methods, they still face the problem that they require a huge amount of annotated data and long training times. One such model that tried to combat the data hunger problem has been to use a modified conditional generative adversarial network called scGAN model[Nguyen et al., 2017]. The model uses a U-net for the generator and adversarial training to combat the large data requirement. However, it requires an immense training cost, and furthermore, it requires regulating the number of shadow pixels by a tunable sensitivity parameter. Here, again like the threshold, the problem of some images being relatively darker has an influence on this parameter which needs to be determined beforehand. In the next section, we elaborate on a model that achieves state-of-the-art performance and mitigates the threshold tuning problem.

2.2 Multitask architecture

In the paper, the authors introduce a Multi-task Mean Teacher model which is able to outperform all the compared state-of-the-art methods in shadow detection. The Multi-task model is depicted in Figure **??**, here multi-task refers to the fact that a multi-task loss is computed for the network for three different prediction tasks. These are shadow mask, shadow-edge mask, and shadow count.

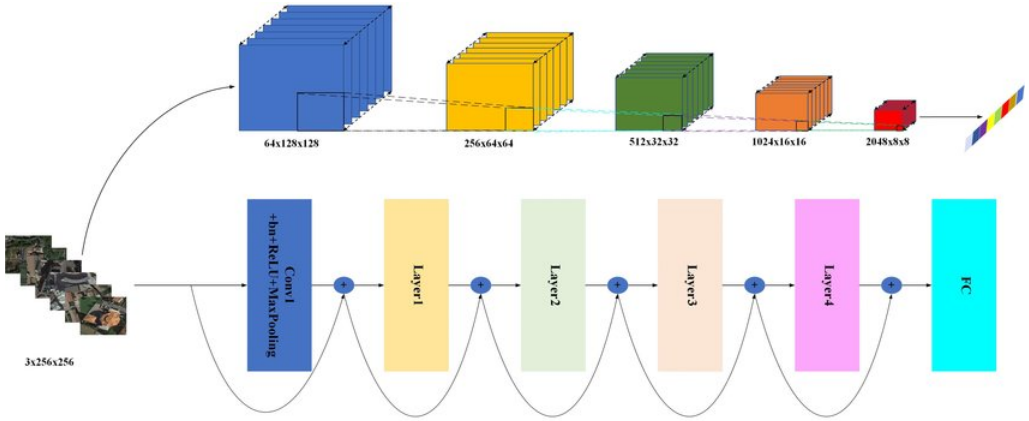

Figure 3: ResNext-101 architecture (Li et al. [2020]).

The first part of this model consists of converting the input image into a set of features at different sizes to account for global and detail information. These are referred to as $EF1$ to $EF5$. They are produced using ResNext architecture which has been pre-trained on ImageNet. This architecture is used for image classification and able to capture the important features of an image. On top of that, training this relatively complex network from scratch requires lots of data and time. After this, the input image is resized to 416 x 416 and then fed into ResNext. ResNext produces features with 64, 256, 512, 1024 and 2048 channels (see Figure 3). A 1 x 1 convolution is applied to each of the intermediate outputs to convert the number of channels to the corresponding EF's number of channels.

2.2.1 Shadow Count (SC) detection

The Shadow Count (SC) is computed by applying average pooling to EF5 on each channel. To make this clear the EF5 dimension of 13 x 13 x 64 is converted to 64. After this, a Fully Connected (fc) layer is applied to get the Shadow Count. According to the paper, by applying shadow count detection, a global constraint is set on the total number of shadow regions.

2.2.2 Shadow region and edge

In the next step the EF layers are combined using short connections to form $DF_{1...5}$. In order to compute DF_k, with $k \geq 2$, all the EF_l layers, with $l \geq k$, are bilinearly up-sampled to the spatial dimension of DF_k, and concatenated. Finally, convolution, Batchnorm and ReLu are applied to the concatenated output to get the desired output DF_k. An example on how to calculate DF_2 is shown in Figure 5. As can be seen from Figure **??**, $DF1$ is calculated differently. A 1 by 1 convolution followed by a ReLu is applied to EF_5 to get the same number of channels as EF_1. Then this result is up-sampled using bilinear interpolation to make element-wise addition possible with EF_1, which results in DF_1. DF_1 and $DF_{2...5}$ are used to predict the shadow edge and shadow region maps respectively with the "Pred" function depicted in Figure **??**. Shadow edge detection enhances the model by setting a detailed-level constraint on the shadow boundaries. Furthermore, Those intermediate levels of shadows are supposed to represent different levels of detail, also global vs more local features. This results in a more appropriate back-propagation for the model's weights.

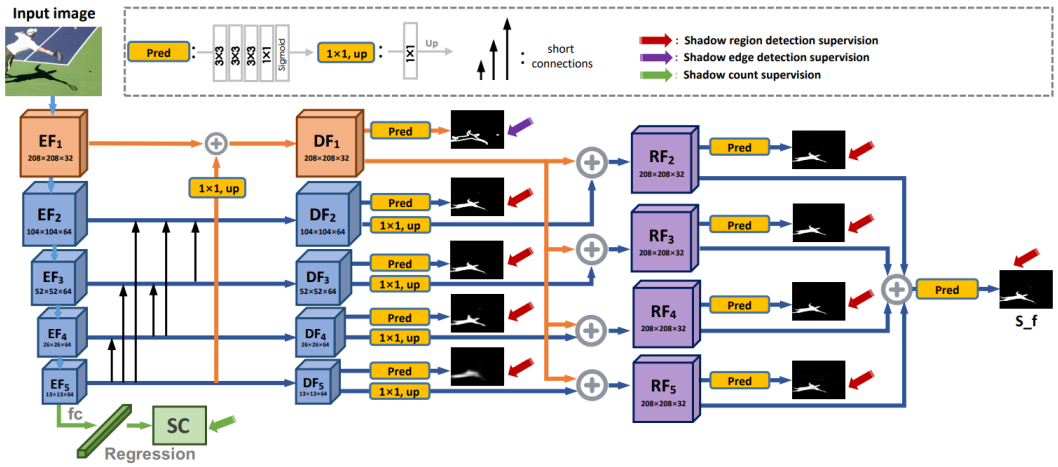

Figure 4: MTMT model

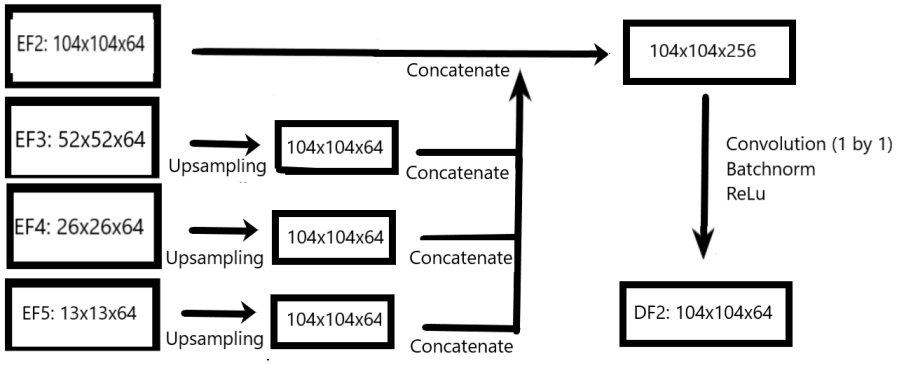

Figure 5: Calculation of DF_2. Here the dimensions are HxWxC, where C stand for the number of channels.

2.2.3 Shadow region detection part 2

$RF_{2..5}$ are computed by applying sequentially a 1 times 1 convolution and bilinear upsampling to DF_k (to get the appropriate spatial and channel dimension). Secondly, DF_k is element-wise added to DF_1 to obtain RF_k. Then, the "Pred" function is applied to get the corresponding shadow maps. Again, those intermediate shadow predictions allow to update the loss function more appropriately. Finally, the resulting output of the network is calculated by adding all the RF layers and applying the "Pred" function.

2.2.4 Mean teach concept

The model consists of a Student and Teacher Network, as depicted in Figure 6. Both the Student and Teacher model follow the same multi-task architecture. The Student Network trains on labeled images, for which all the predictions are compared against the ground truth. The predictions consist of the 9 shadow masks, the edge mask and the Shadow count. We generate the shadow edge masks using the shadow ground truths, and a canny edge detector [Bradski, 2000]. The same applies to the shadow count which is computed by first removing small shadow objects from the ground truth and then calculating the number of connected areas. Finally, the loss is calculated for all the shadow maps, edge masks and shadow count, which is called the supervised loss.

The second part of the MTMT model is the Teacher Network, on which only unlabeled data is fed as input. To be more concise, the unlabeled data is passed through the Student Network without noise,

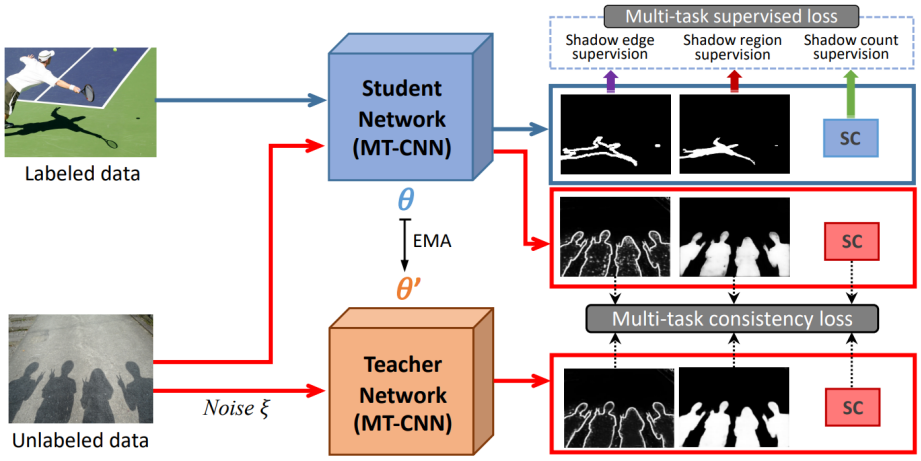

Figure 6: Student and Teacher network.

and through the Teacher Network with noise included. By comparing outputs of both networks the consistency loss is computed as:

$$\mathscr{L}^c(y) = \mathscr{L}^c_r + \mathscr{L}^c_e + \mathscr{L}^c_c \tag{1}$$

where

$$\mathscr{L}^c_r = \sum_{j=1}^{9} MSE(S_{rj}, T_{rj})$$
$$\mathscr{L}^c_e = MSE(S_e, T_e)$$
$$\mathscr{L}^c_c = MSE(S_c, T_c)$$

and the total loss is computed as:

$$\mathscr{L}_{total} = \sum_{i=1}^{N} \mathscr{L}^s(x_i) + \lambda \sum_{j=1}^{M} \mathscr{L}^c(y_j) \tag{2}$$

where N and M are the number of labelled and unlabelled images in a batch. λ as in the original paper is set as $\lambda = \lambda_{max} e^{(5(1t/t_{max})2)}$, where $\lambda_{max} = 10$, t is the iteration number and t_{max} is the total number of iterations.

The idea behind this concept is that the network should be robust against noise. To be more specific, to keep the consistency loss as low as possible the outputs of the Student (which takes as input the image without noise) and Teacher (which takes as input the noisy image) should be as close as possible. This is achieved when adding noise to the image does not or little affect the shadow predictions. As a final remark the parameters of the Teacher network are updated by the exponential moving average of the Student parameter; $\theta_{new} = \eta\, \theta_{old} + (1 - \eta)\theta_{Student}$.

3 Our work

3.1 Changes

As compared to the original paper which used a batch size of 6 consisting of 4 labelled images and 2 unlabelled images, we lowered the batch size to 3 using 2 labelled and 1 unlabelled image. This was a consequence of the huge number of parameters needed to fit the network in the GPU. We are running the training loop on a Geforce GTX 1060 and the largest possible batch size was 2 for labeled and 1 for unlabeled.

3.2 Challenges

In the process of replicating the paper, we have also examined the code that the original authors have provided for their model. This was necessary as large parts of the implementation details were vague in the paper. For instance, in the paper, they mention that they calculate the DF layers using a convolutional layer, but do not mention what kind of convolution (e.g. the filter size and stride).

At other times the implementation of their code was noticeably different from what was reported in the paper. This was most noticeable at the predict function where in the paper they report 3 3x3 convolution layers followed by a 1x1 convolution and finally a sigmoid. In the code their implementation consisted of a 3x3 convolutional layer, a batch normalization, followed by a Relu function, followed by a 0.1 dropout, and finally a 1x1 convolution.

We wanted to determine the influence of the different prediction methods so we devised two models, one model was based on the paper which we used throughout all our experiments. Another was based on the code they have provided, which we compared to method 1 but only running them both on the base model(student model). We call the model based on the paper "our method 1". And the model based on their code "our method 2". This comparison is found the ablation study section.

3.3 Experiments

We tested the models on two datasets SBU [Vicente et al., 2016] and ISTD [Wang et al., 2018] trained it for 10000 iterations like in the original paper. We furthermore conducted ablation studies where we only trained the base(Network). Our Method 1 was used in experiments on the full model. We used a batch size of 3, 2 labelled images and 1 unlabelled. All experiment run in the ablation studies section use a batch size of 2 labelled images.

As a consequence of not having enough processing power, we had to lower the Batch Size during training. The paper uses a Batch Size of 4 and 2 for unlabeled and labeled data, respectively. Whereas, we use a Batch Size of 2 for labeled and 1 for unlabeled data (see Table 3.4).

3.4 Results

Our Full model (Student and Teacher) achieves a significantly worse Balanced Error Rate (BER) for both Shadow and Non-Shadow pixel classification. As part of the ablation study, we have trained the exact same model, however with the Teacher part disabled(only student 1). This allows us to investigate the significance of the mean teacher concept. We can see that for the SBU data set this has produced significantly worse results but this could also be linked to a final prediction threshold of 90/255 intensity score that was advised by the main authors. However still with the same setting, the full model produced much better results. Surprisingly however, in the experiments run on the ISTD model we achieved better results for the only student model for method 1. For our method 2, we can see that on the SBU dataset we have achieved far better results than the one obtained for method 1. We have not run the experiments on the ISTD dataset but in order to obtain an idea of how it would have behaved, we have provided learning curves for only the first 200 iterations on fig. 8. From this figure, we can observe that indeed the two methods behave differently and likely the ISTD results would have produced similarly good results.

	Batch Size	SBU	ISTD
Network	Labeled/Unlabeled	BER/Shadow/Non Shadow	BER/Shadow/Non Shadow
Full model	2/1	$10.77/17.38/4.16$	$8.1/13.55/2.83$
Only student 1	2/-	$26.73/53.09/\mathbf{0.36}$	$7.13/12.66/\mathbf{1.61}$
Only student 2	2/-	$4.87/7.30/2.44$	$-$
Their full model	4/2	$\mathbf{3.15}/\mathbf{3.73}/2.57$	$\mathbf{1.72}/\mathbf{1.36}/2.08$
Their student only	4/2	3.61/-/-	2.03/-/-

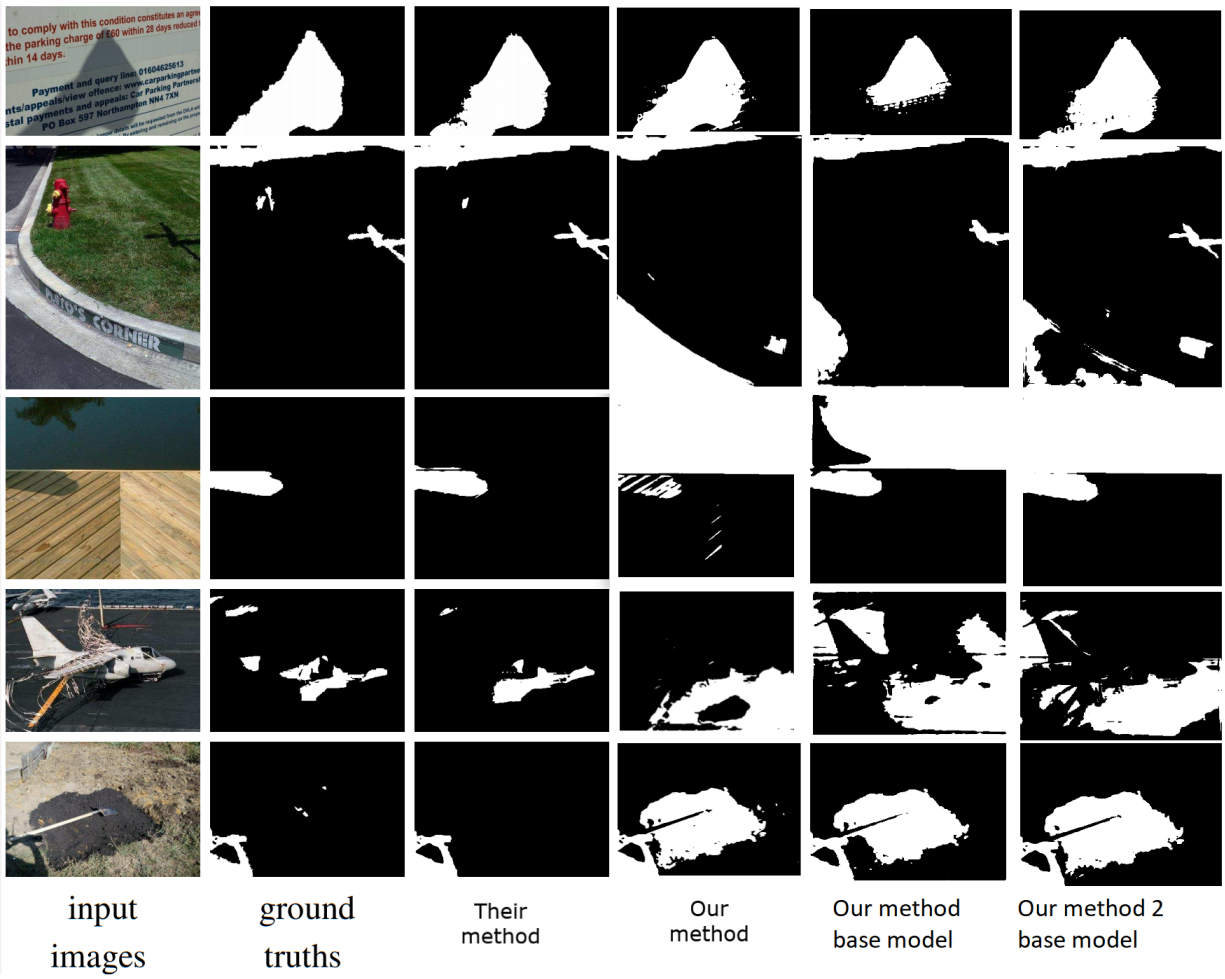

Figure 7: Visual comparison of the models

4 Discussion

4.1 Performance

Interestingly, the adjustment of the network significantly improved the results. The second student model even outperformed the full model. This suggests that the model adjustment that we made has more influence on the performance than the lowered batch size. In Figure 7, the visual comparison between the models is shown. It is also worth noticing that the results for the ISTD dataset are much better than for the SBU dataset. This is simply because the ISTD dataset is relatively easier. Most images in SBU contain dark regions that are hard to distinguish from real shadows. Examples of the SBU and ISTD dataset are shown in Figure 9 and 10, respectively.

4.2 Ablation studies

As can be seen from the table above the network for which the Teacher is disabled achieves a score that is much lower than the full model. In the full model, the BER score is 10.77 whereas, the BER score for the only student model is 26.73. Interestingly the BER score for non-shadow is much better

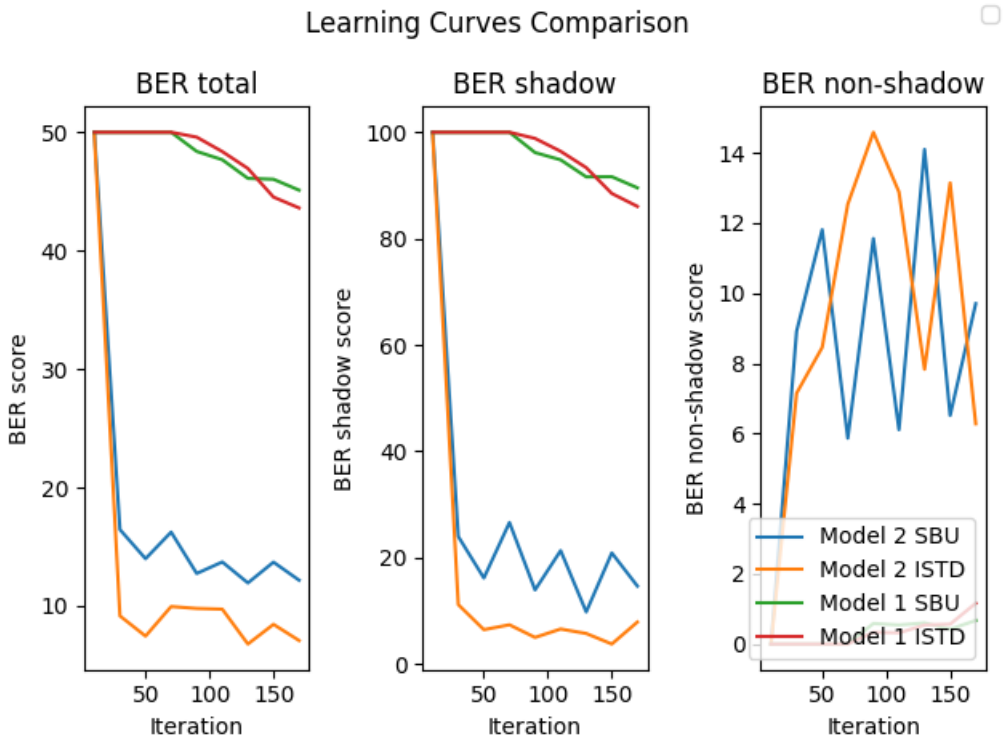

Figure 8: Learning curves comparison of the models

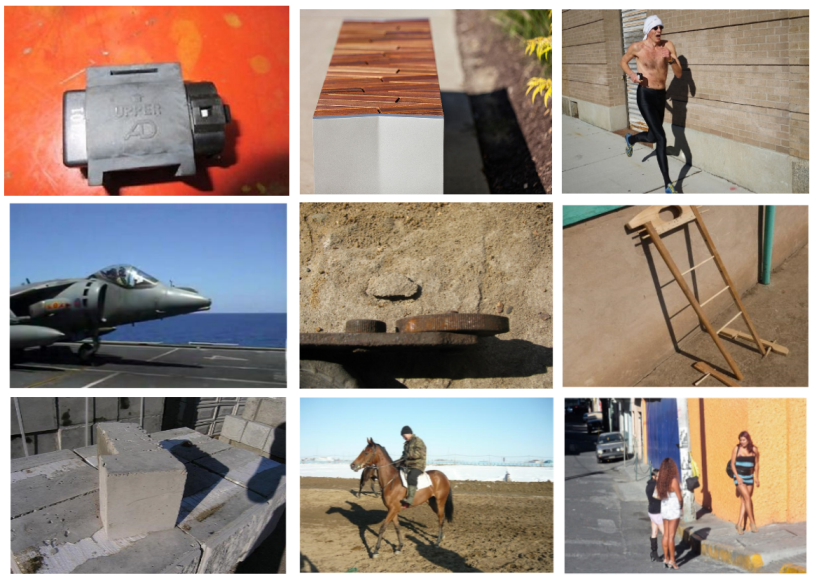

Figure 9: Samples of the SBU test set.

for the only-student model. However, this simply means that the model classifies a pixel with a higher probability as non-shadow and thus automatically achieves a higher BER for shadow.

Figure 10: Samples of the ISTD test set. As can be seen, these images are relatively easy in comparison to the SBU dataset.

4.3 Visual comparison

From the visual comparison, we can see that our models are able to capture some of the shadow regions quite well. However, in general, it fails in a similar fashion as many previously produced methods have. That is the colour contrasts and variations in images lead to dark regions to be falsely predicted as shadows. The results that have been presented by Chen et al. [2020] show a significantly better and more nuanced performance. We believe there may be two potential reasons as to why this may be the case. When considering our only student model 2, the main difference between our implementation and the implementation that could be found in their code is the use of a Relu function in the prediction step compared to a sigmoid in ours and a different batch size. While a larger batch size could increase the stability of the network and therefore the result. The lack of nuance in the produced shadow masks may hint at a vanishing gradient problem where highly specific scenarios are not well represented. That being said more experimentation would be needed to confirm this.

4.4 Future ideas

Currently, the system works on 2d images. One possible way of extending this is by adjusting the model to allow for the processing of sequences of images. Video processing of shadows can be used in all sorts of applications think of for example self-driving cars. Self-driving cars possibly are better able to estimate depth with the help of this shadow video detection.

Another potential use case would be to use a similar Teacher Student model in image re-lighting. Training such a model using sub prediction steps such as shadow count, shadow masks, and edge masks may help impose similar such constraints on the final prediction that will help ground the predictions more. The architecture of the network could even be modified to work with a GAN framework, where the adversary network would attempt to differentiate between a shaded image and the generated one. Likewise this could be applied to shadow detection task. Using a cGan As in Nguyen et al. [2017] the adversary network could be trained to differentiate between the true and generated shadow mask conditioned on the original image.

References

G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

Z. Chen, L. Zhu, L. Wan, S. Wang, W. Feng, and P.-A. Heng. A multi-task mean teacher for semi-supervised shadow detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5611–5620, 2020.

F. Li, R. Feng, W. Han, and L. Wang. An augmentation attention mechanism for high-spatial-resolution remote sensing image scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:3862–3878, 2020.

V. Nguyen, T. F. Yago Vicente, M. Zhao, M. Hoai, and D. Samaras. Shadow detection with conditional generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4510–4518, 2017.

T. F. Y. Vicente, L. Hou, C.-P. Yu, M. Hoai, and D. Samaras. Large-scale training of shadow detectors with noisily-annotated shadow examples. In *European Conference on Computer Vision*, pages 816–832. Springer, 2016.

J. Wang, X. Li, and J. Yang. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1788–1797, 2018.