Reproducibility of "Pixel-wise Anomaly Detection in Complex Driving Scenes " for ML Reproducibility Challenge 2021 (Supplementary Material)

Anonymous Author(s)
Affiliation
Address
email

This material includes a detailed discussion on the model used. It also includes more results that compare the authors' output to ours. We have even shown some results where this model falls short and tried to generalize the reason for these cases.

1 Dissimilarity model

The model mainly has three modules namely segmentation, synthesis, dissimilarity and an ensemble:

Segmentation Module : We employ the pre-trained weights of the model as trained in [6] on Cityscapes dataset. In addition to generating a segmented image, we generate two dispersion maps, softmax entropy H and softmax distance D, which prove beneficial in understanding anomalies within the generated segmentation map(p(c) is the softmax probability for class c). For each pixel x, H and D are calculated as follows:

$$H_x = -\sum_{c \in classes} p(c)\, log_2\, p(c) \tag{1}$$

$$D_x = 1 - \max_{c \in classes} p(c) + \max_{c^1 \in classes \backslash (arg\, max_c\, p(c)} p(c^1) \tag{2}$$

Synthesis Module : To build the realistic image out of the segmentation map, we employ pre-trained weights from the model trained on Cityscapes dataset as a conditional generative adversarial network (c-GAN) [2] [5]. However, because the semantic map lacks information such as color appearance, per-pixel value comparison between the original input and the synthesized image is not possible. As a result, we use perceptual difference, which employs a pre-trained VGG16 model as a feature extractor to compare overall spatial structure rather than features such as color and texture, allowing us to better classify anomalies. For every pixel x of the input image and corresponding pixel r from the synthesized image V is defined as follows :

$$V(x, r) = \sum_{1}^{N} \frac{1}{M_i} ||F^i(x) - F^i(r)||_1 \tag{3}$$

Spatial-Aware Dissimilarity Module : We adopt the authors' method of representation. *ck – sn* denotes a 3x3 Convolution-RELU layer with k filters and stride n. *dk* denotes a 7x7 Convolution-RELU layer with k filters and stride 1. *m2* denotes a 2x2 max pooling layer. *sp – 19* denotes a SPADE normalization-SELU layer [1][3], which uses 19 channels from the predicted semantic map as one of its inputs. *tk* denotes a 2x2 transposed convolution with k filters. *r2* denotes a 1x1 Convolution layer with two filters. This module takes as input the original image, generated image, semantic map, and uncertainty maps (softmax entropy, softmax distance, perceptual difference) calculated in the previous steps to predict the anomaly segmentation map. It is mainly divided into three modules namely encoder, fusion and decoder:

1. **Encoder :** We used pre-trained VGG16[4] as an encoder to extract features of resynthesis image and input image. A CNN d32, c64 – s2, c128 – s2, c256 – s2 to extract features from all uncertainty maps concatenated and semantic map.

2. **Fusion Module :** Concatenates and passes features extracted from resynthesis, input, segmented maps through a 1×1 convolution which is then passed into correlation block along with encoded uncertainty map where pointwise correlation is performed outputting four feature map resolutions corresponding to each of the four layers of the decoder.

3. **Decoder :** There are four decoder blocks used in the dissimilarity network. The first and second blocks follow the structure: c256 s1, sp 19, c256 s1, sp 19, t256. The third follows: c384s1, sp19, c128s1, sp 19, t1258, while the last one follows: c192 s1,sp19, c64s1, sp19, r2. The first decoder block takes the lowest resolution feature map. The concatenation of the feature map from the fusion module and the output of the preceding decoder block is used as the input for all subsequent decoder blocks.

2 Results that matched

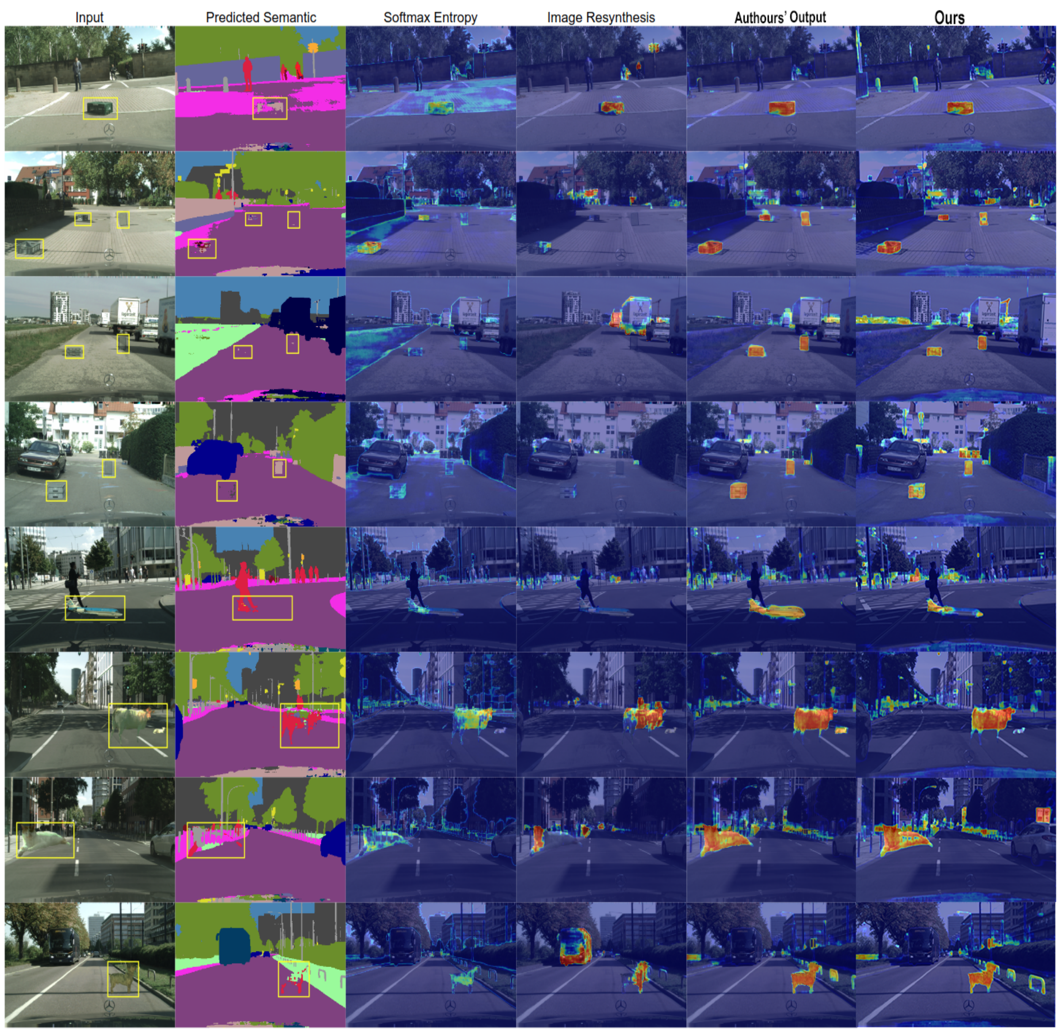

Figure 1: Our predictions are compared to the authors' in detecting the main anomaly. Image Resynthesis [24] and Softmax Entropy [14] are also mentioned.

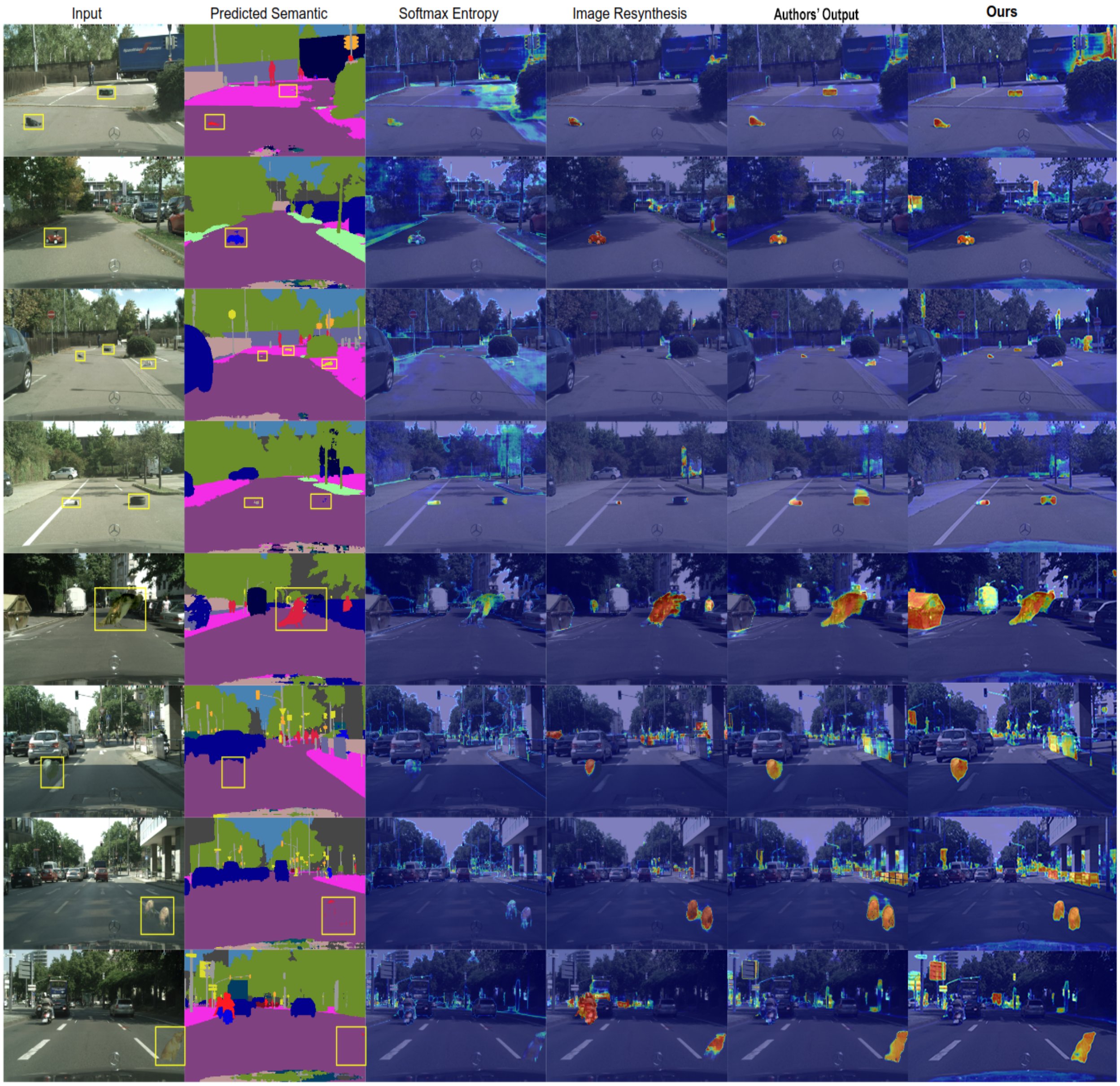

Figure 2: The picture above compares the authors' output to ours (last two columns, respectively). The first four images are from FS Lost and Found, followed by four from FS Static. We can see that the results are rather comparable.

3 Results that did not match

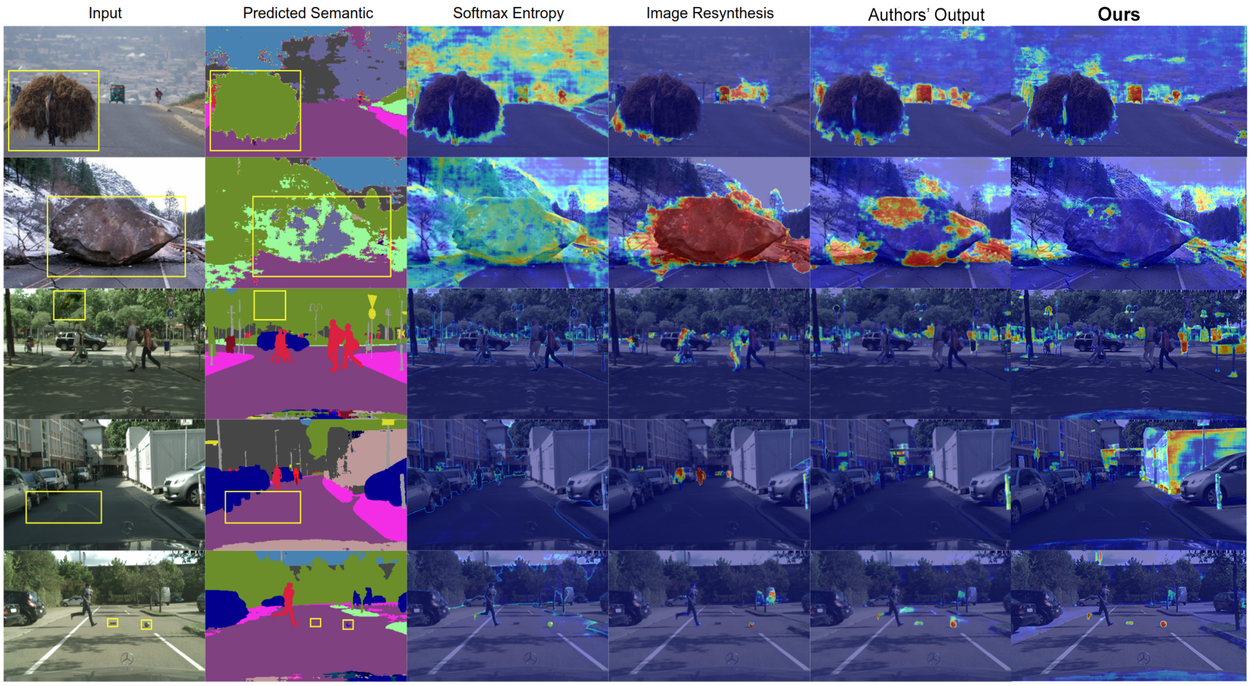

Figure 3: Failure cases. This framework is still unable to detect some challenging anomaly cases. The top two images show derived scenes with different metropolitan landscapes, the middle two images show anomaly cases that blend in nicely with the backdrop, and the bottom two images show small anomalous items that only cover a few pixels in the image.

4 Discussion

We can see from the above results that our model provides results comparable to the authors. We believe minor differences are related to this model's sensitivity to ensemble weights, which is explained in detail in the reproducibility report. We can see that there is a lot of room for improvement in the outcomes that do not match with the ground truth.

References

[1] Günter Klambauer et al. "Self-Normalizing Neural Networks". In: *CoRR* abs/1706.02515 (2017). arXiv: 1706.02515. URL: http://arxiv.org/abs/1706.02515.

[2] Xihui Liu et al. "Learning to Predict Layout-to-image Conditional Convolutions for Semantic Image Synthesis". In: *CoRR* abs/1910.06809 (2019). arXiv: 1910.06809. URL: http://arxiv.org/abs/1910.06809.

[3] Taesung Park et al. "Semantic Image Synthesis With Spatially-Adaptive Normalization". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2332–2341. DOI: 10.1109/CVPR.2019.00244.

[4] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *arXiv 1409.1556* (Sept. 2014).

[5] Ting-Chun Wang et al. "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs". In: *CoRR* abs/1711.11585 (2017). arXiv: 1711.11585. URL: http://arxiv.org/abs/1711.11585.

[6] Yi Zhu et al. "Improving Semantic Segmentation via Video Propagation and Label Relaxation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.