Detecting the Unexpected via Image Resynthesis
Supplementary Material

Krzysztof Lis Krishna Nakka Pascal Fua Mathieu Salzmann

Computer Vision Laboratory, EPFL

1. Detecting Unexpected Objects

The legend for the semantic class colors used throughout the article is given in Fig. 1. We present additional examples of the anomaly detection task in Fig. 2.

The synthetic training process alters only foreground objects. A potential failure mode could therefore be for the network to detect *all* foreground objects as anomalies, thus finding not only the true obstacles but also everything else. In Fig. 3, we show that this does not happen and that objects correctly labeled in the semantic segmentation are not detected as discrepancies.

In Fig. 4, we illustrate the fact that, sometimes, objects of known classes differ strongly in appearance from the instances of this class present in the training data, resulting in them being marked as unexpected.

We present a failure case of our method in Fig. 5: Anomalies similar to an existing semantic class are sometimes not detected as discrepancies if the semantic segmentation marks them as this similar class. For example, an animal is assigned to the *person* class and missed by the discrepancy network. In that case, however, the system as a whole is still aware of the obstacle because of its presence in the semantic map.

1.1. Discrepancy Network

Our discrepancy network relies on the implementations of *PSP Net* [10] and *SegNet* [1] kindly provided by Zijun Deng. The detailed architecture of the discrepancy network is shown in Fig. 6. We utilize a pre-trained VGG16 [8] to extract features from images and calculate their pointwise correlation, inspired by the co-segmentation network of [6]. The up-convolution part of the network contains SELU activation functions [5]. The discrepancy network was trained for 50 epochs using the Cityscapes [2] training set with synthetically changed labels as described in Section 3.2 of the main paper. We used the Adam [4] optimizer with a learning rate of 0.0001 and the per-pixel cross-entropy loss. We utilized the class weighting scheme introduced in [7] to offset the unbalanced numbers of pixels belonging to each class.

	Full	Labels only	Resynthesis only
Supervised	0.94	0.93	0.96
Unsupervised	0.82	0.79	0.76

Table 1: **Performance of the discrepancy network in a supervised setting.** AUROC scores measured on the *Lost and Found* dataset.

Supervised Discrepancy Network. To get an upper bound on its accuracy, we test the discrepancy network in a supervised setting. To this end, we use the ground-truth anomaly labels of the *Lost and Found* training set, with semantics predicted by PSP Net. The AUROC scores, measured on the test set, are shown in Table 1.

2. Detecting Adversarial Samples

We show additional results on adversarial example detection on the Cityscapes and BDD datasets using the Houdini and DAG attack schemes in Figs. 7 and 8. To obtain these results, we set the maximal number of iterations to 200 in all settings and L_∞ perturbation of 0.05 across each iteration of the attack. We randomly choose 80% of the original validation samples to train the logistic detectors and the rest of the samples are used for evaluation. While evaluating the state-of-the-art Scale Consistency method [9], we found by cross-validation that a patch size of 256×256 resulted in the best performance for an input image of size 1024×512.

3. Image Attribution

We used Wikimedia Commons images kindly provided under the Creative Commons Attribution license by the following authors: Thomas R Machnitzki[1], Megan Beck-

[1] commons.wikimedia.org/wiki/File:Goose_on_the_road_Memphis_TN_2013-03-17_001.jpg

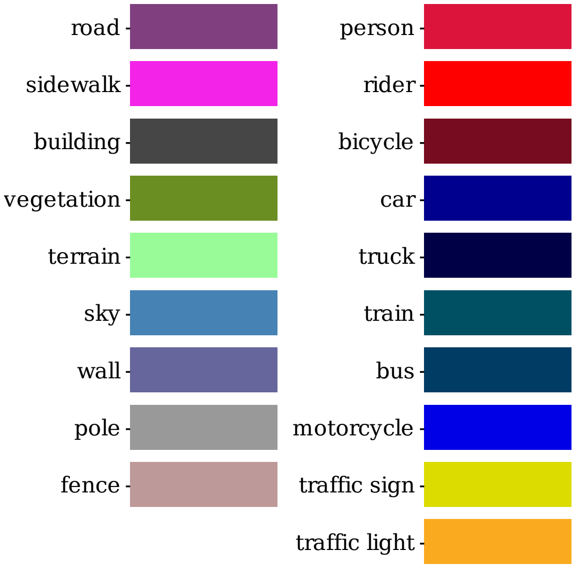


Figure 1: **Semantic map legend.** The colors used in semantic maps throughout this article correspond to the object classes listed above.

ett[2], Infrogmation[3], Kyah[4], PIXNIO[5], Matt Buck[6], Luca Canepa[7], Jonas Buchholz[8] and Kelvin JM[9].

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[3] Clement Creusot and Asim Munawar. Real-Time Small Obstacle Detection on Highways Using Compressive RBM Road Reconstruction. In *Intelligent Vehicles Symposium*, 2015.

[4] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimisation. In *International Conference on Learning Representations*, 2015.

[5] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, 2017.

[6] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep Object Co-Segmentation. In *Asian Conference on Computer Vision*, 2018.

[7] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. *arXiv Preprint*, abs/1606.02147, 2016.

[8] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*, 2015.

[9] Chaowei Xiao, Ruizhi Deng, Bo Li, Fisher Yu, Mingyan Liu, and Dawn Song. Characterizing adversarial examples based on spatial consistency information for semantic segmentation. In *European Conference on Computer Vision*, pages 217–234, 2018.

[10] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[2] commons.wikimedia.org/wiki/File:Rhino_crossing_road.JPG

[3] commons.wikimedia.org/wiki/File:Broadmoor9JanConesSkidloader.jpg

[4] commons.wikimedia.org/wiki/File:Federation_chantier_aout_2006_-_5.JPG

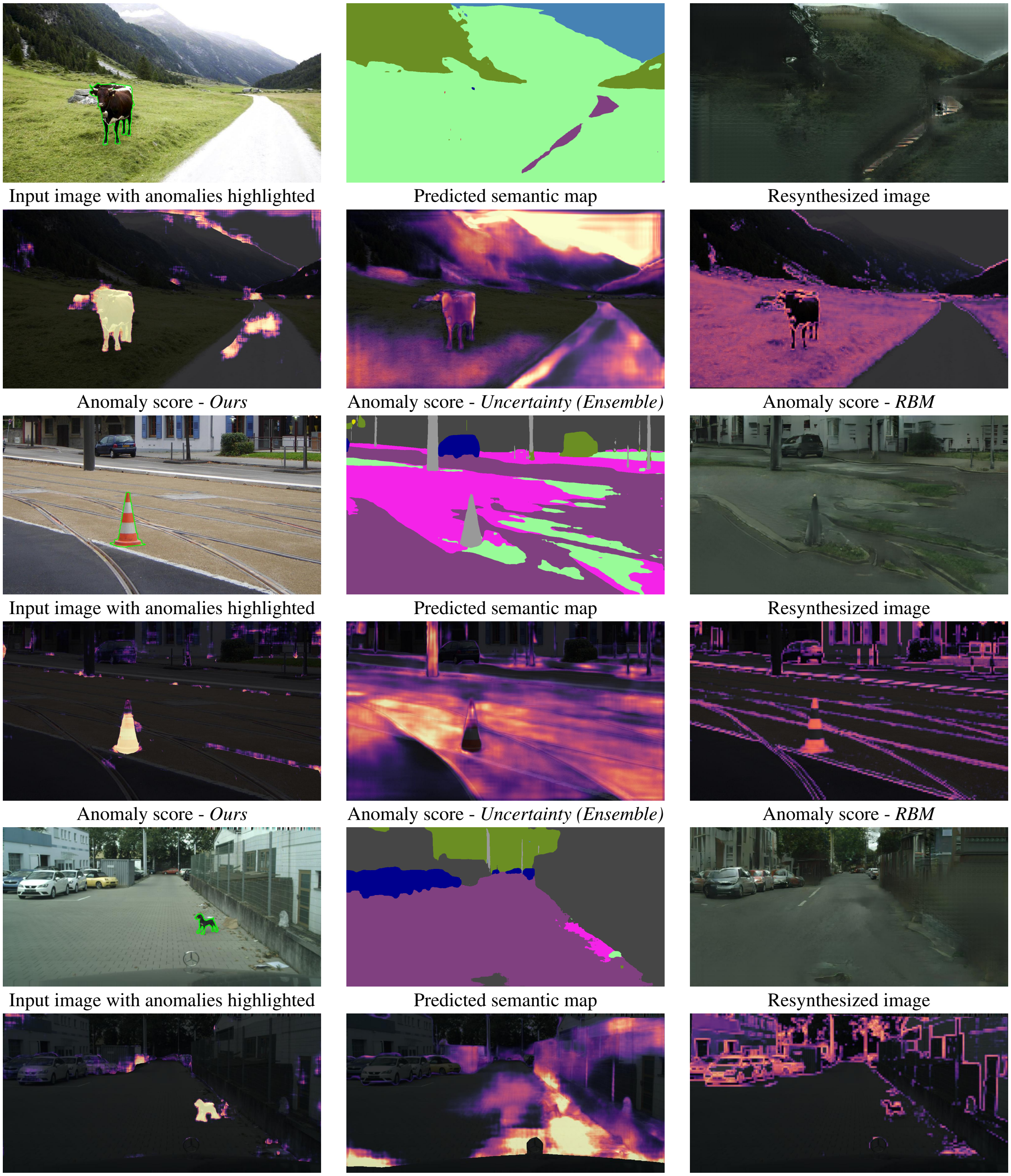
[5] commons.wikimedia.org/wiki/File:Bovine_catle_beside_road.jpg

[6] commons.wikimedia.org/wiki/File:Beeston_MMB_A6_Middle_Street.jpg

[7] commons.wikimedia.org/wiki/File:Zebra_Crossing_Abbey_Road_Style_(63894353).jpeg

[8] commons.wikimedia.org/wiki/File:Aihole-Pattadakal_road.JPG

[9] commons.wikimedia.org/wiki/File:A_man_carrying_dry_grass_on_bicycle_for_domestic_animal_like_cows.jpg



Input image with anomalies highlighted

Predicted semantic map

Resynthesized image

Anomaly score - *Ours*

Anomaly score - *Uncertainty (Ensemble)*

Anomaly score - *RBM*

Input image with anomalies highlighted

Predicted semantic map

Resynthesized image

Anomaly score - *Ours*

Anomaly score - *Uncertainty (Ensemble)*

Anomaly score - *RBM*

Input image with anomalies highlighted

Predicted semantic map

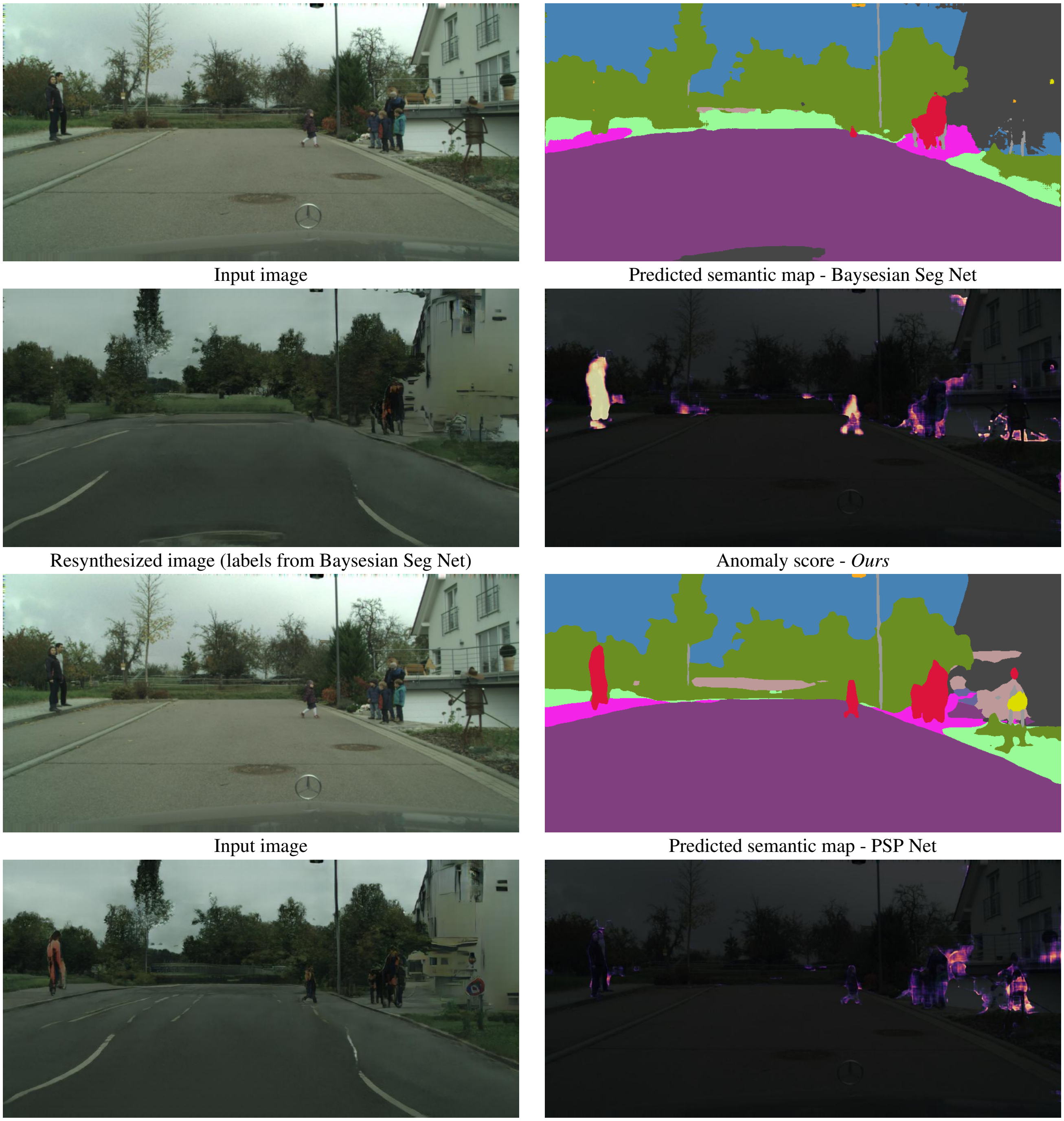
Resynthesized image

Anomaly score - *Ours*

Anomaly score - *Uncertainty (Dropout)*

Anomaly score - *RBM*

Figure 2: Additional examples of the anomaly detection task



Input image

Predicted semantic map - Baysesian Seg Net

Resynthesized image (labels from Baysesian Seg Net)

Anomaly score - *Ours*

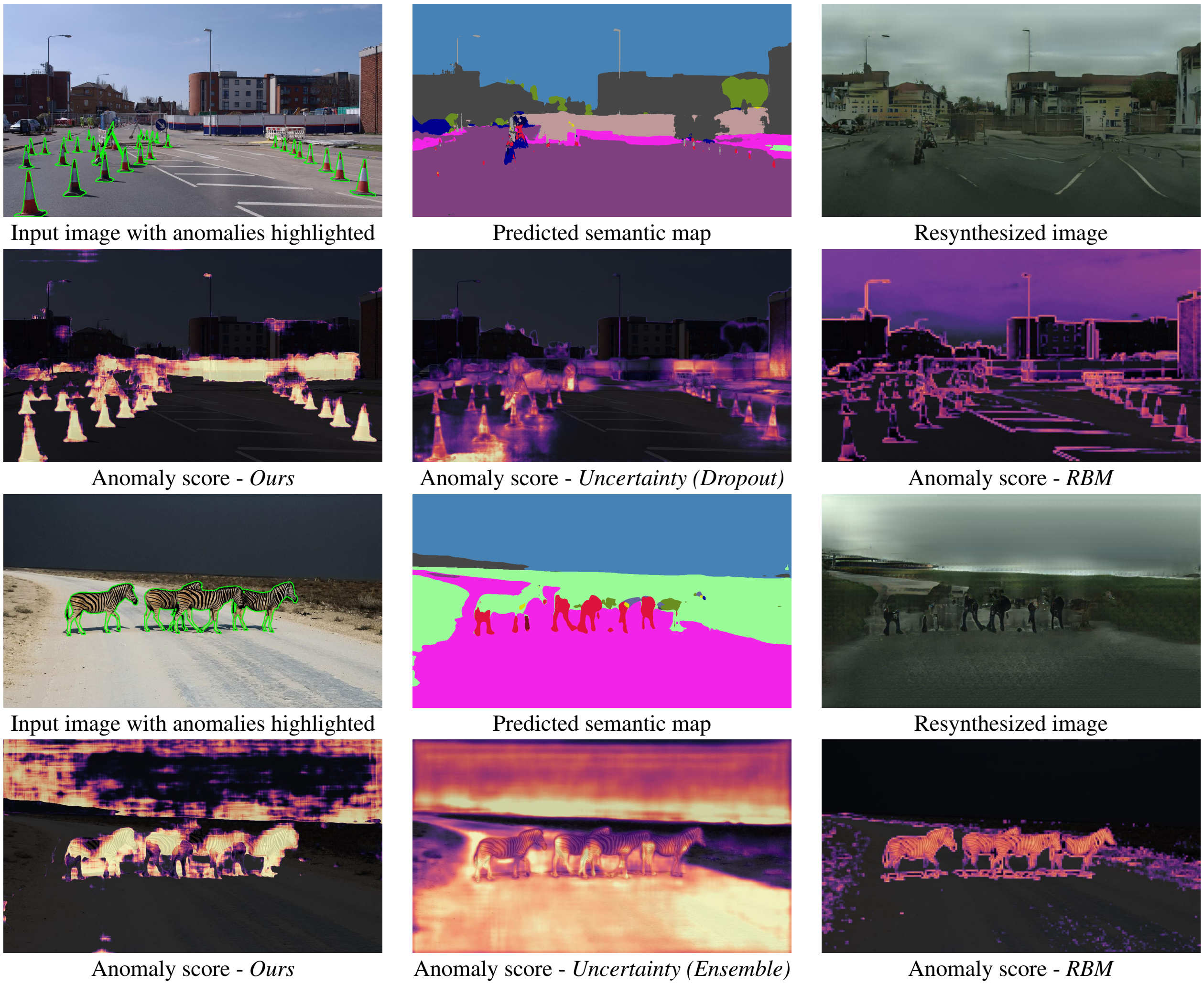
Input image

Predicted semantic map - PSP Net

Resynthesized image (labels from PSP Net)

Anomaly score - *Ours*

Figure 3: The synthetic training process alters only foreground objects, but that does not mean our discrepancy network learns to blindly mark all such objects. In the top row, we show an example where the *Bayesian SegNet* failed to correctly label some of the people present, and this discrepancy is detected by our network. However, our detector reports no discrepancy when the *PSP Net* correctly labels the people in the image (third row).



Input image with anomalies highlighted

Predicted semantic map

Resynthesized image

Anomaly score - *Ours*

Anomaly score - *Uncertainty (Dropout)*

Anomaly score - *RBM*

Input image with anomalies highlighted

Predicted semantic map

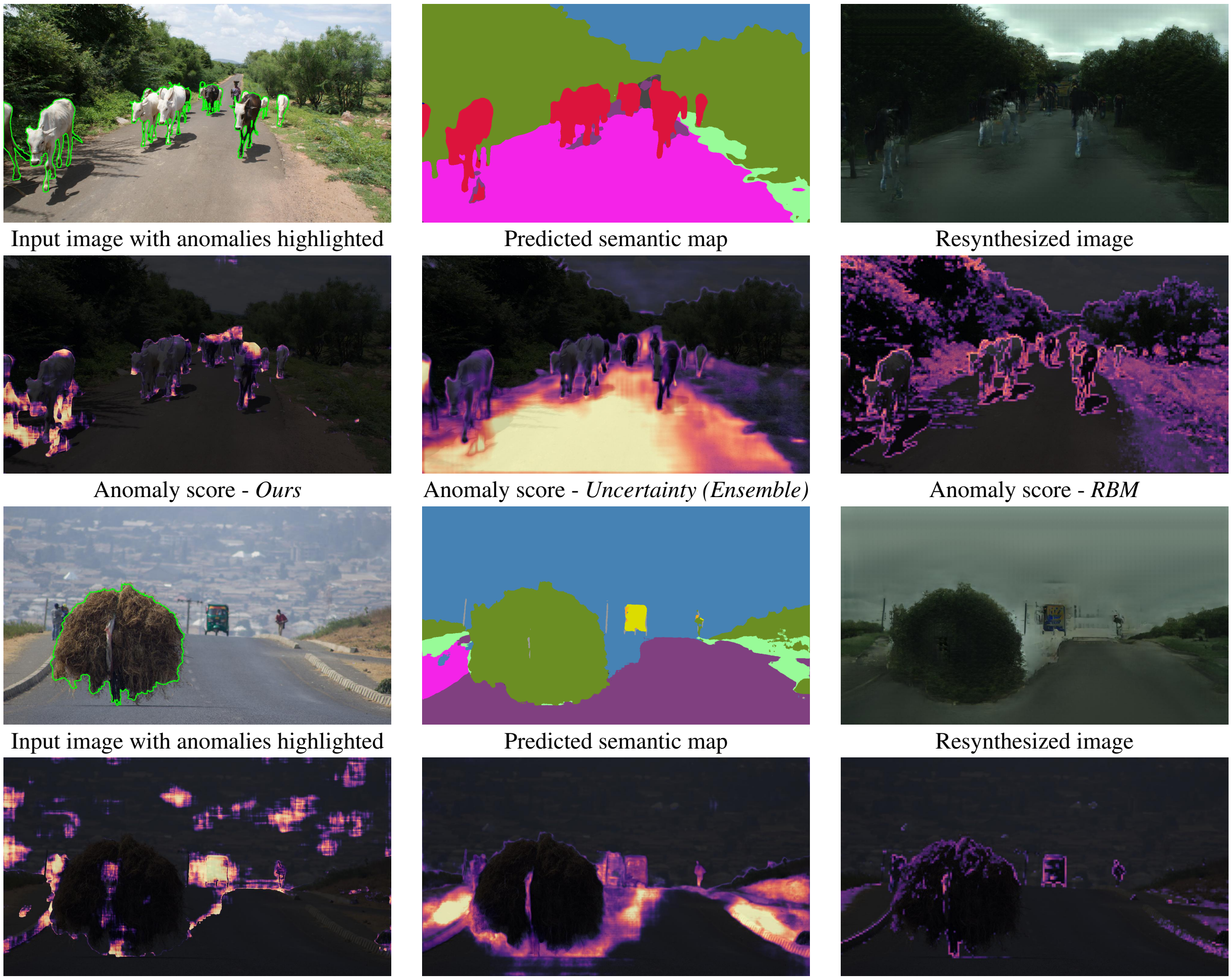
Resynthesized image

Anomaly score - *Ours*

Anomaly score - *Uncertainty (Ensemble)*

Anomaly score - *RBM*

Figure 4: **Unusual versions of known objects**. Objects of known classes are marked as anomalies because their appearance differs from the examples of this class present in the training data, for example the fence in the first row (*fence* class) and the dark sky in the third row. Note that the *RBM* patch-based method [3] is especially sensitive to edges and so it detects the zebras very well.



Input image with anomalies highlighted

Predicted semantic map

Resynthesized image

Anomaly score - *Ours*

Anomaly score - *Uncertainty (Ensemble)*

Anomaly score - *RBM*

Input image with anomalies highlighted

Predicted semantic map

Resynthesized image

Anomaly score - *Ours*

Anomaly score - *Uncertainty (Dropout)*

Anomaly score - *RBM*

Figure 5: **Failure cases.** Our approach sometimes fails when the anomaly bears resemblance to an existing class: For example, animals classified as people in the first row or transported hay classified as vegetation in the third row. The system as a whole is nonetheless still aware of the obstacle because of its presence in the semantic map.

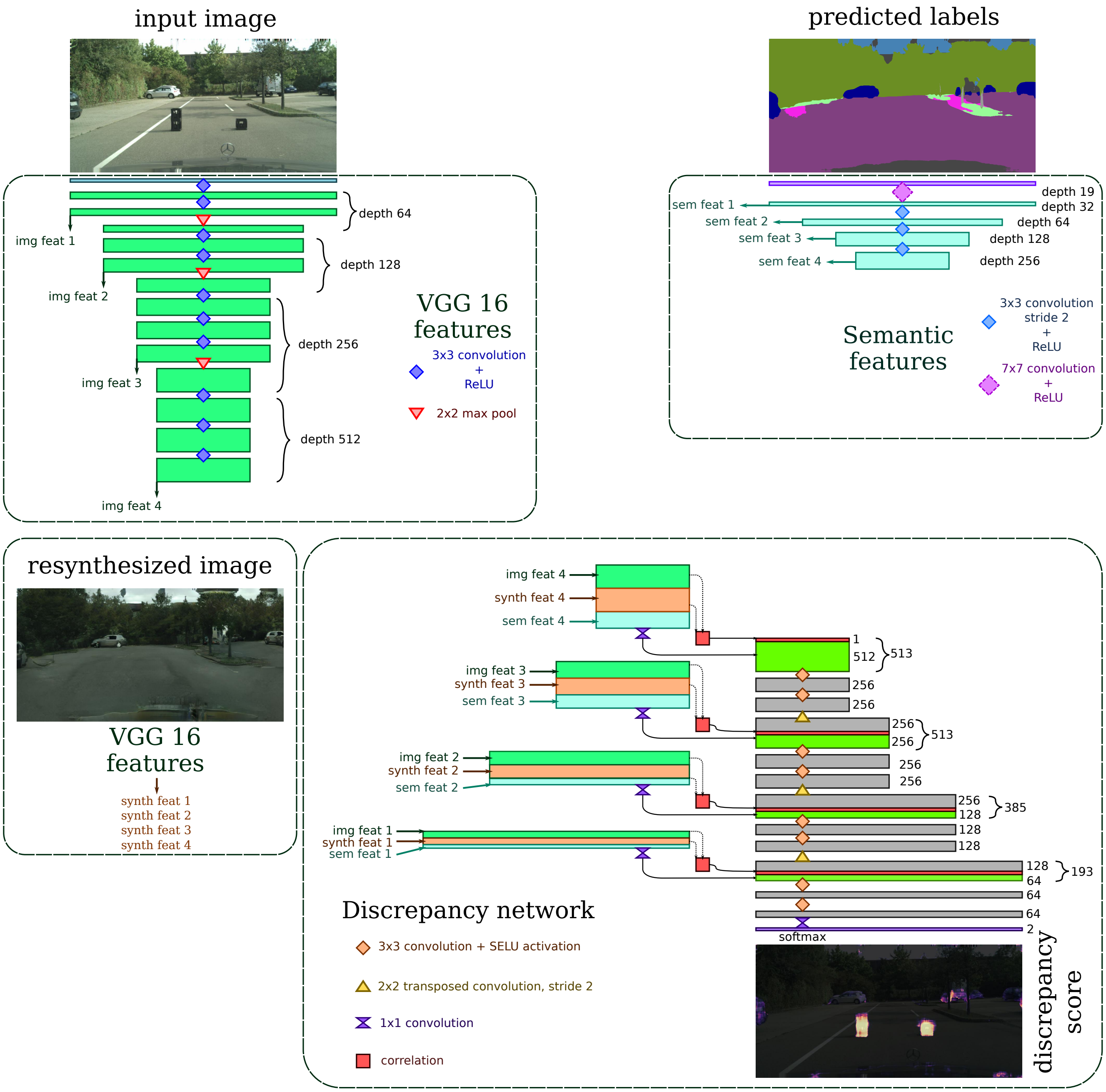
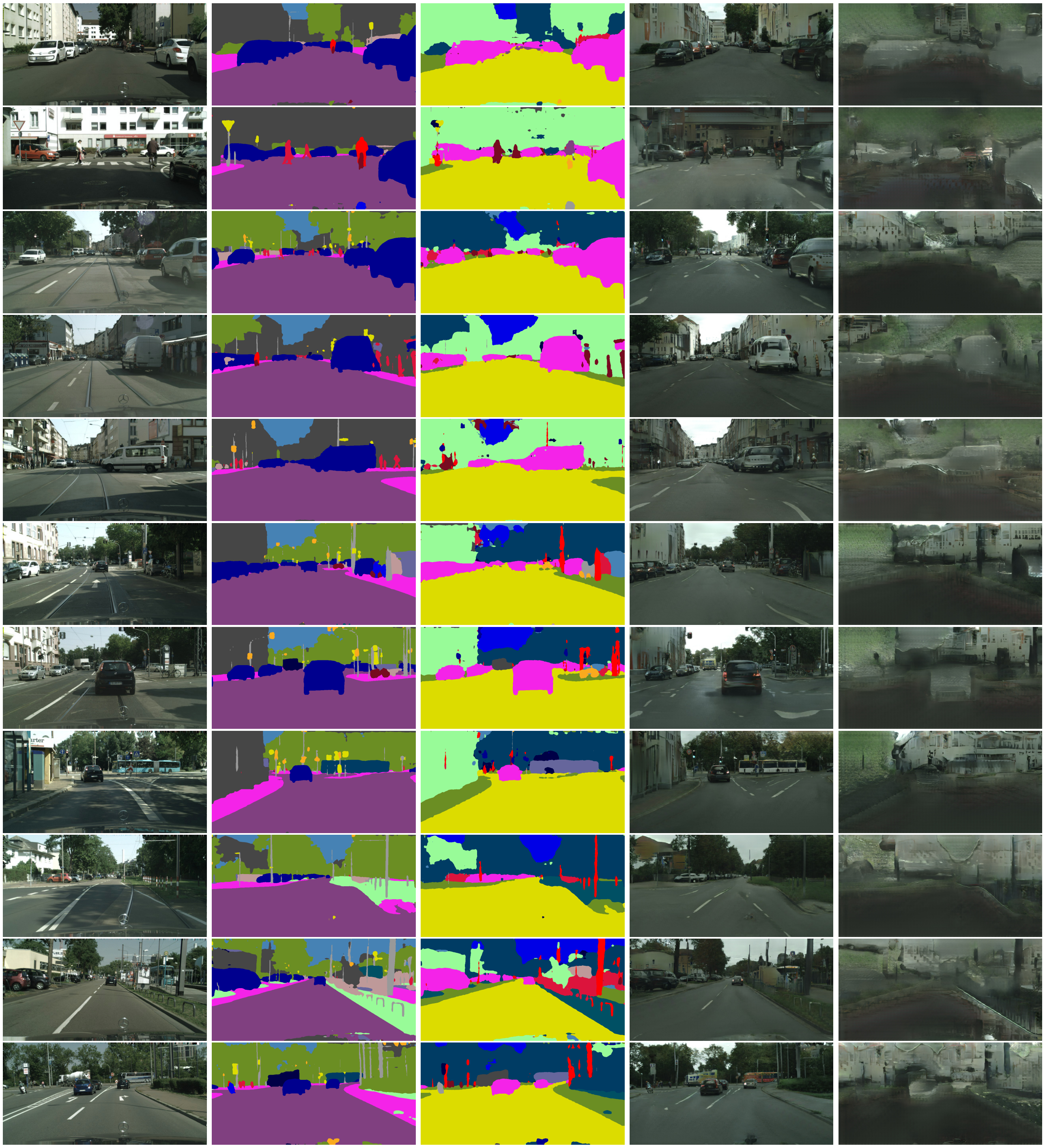
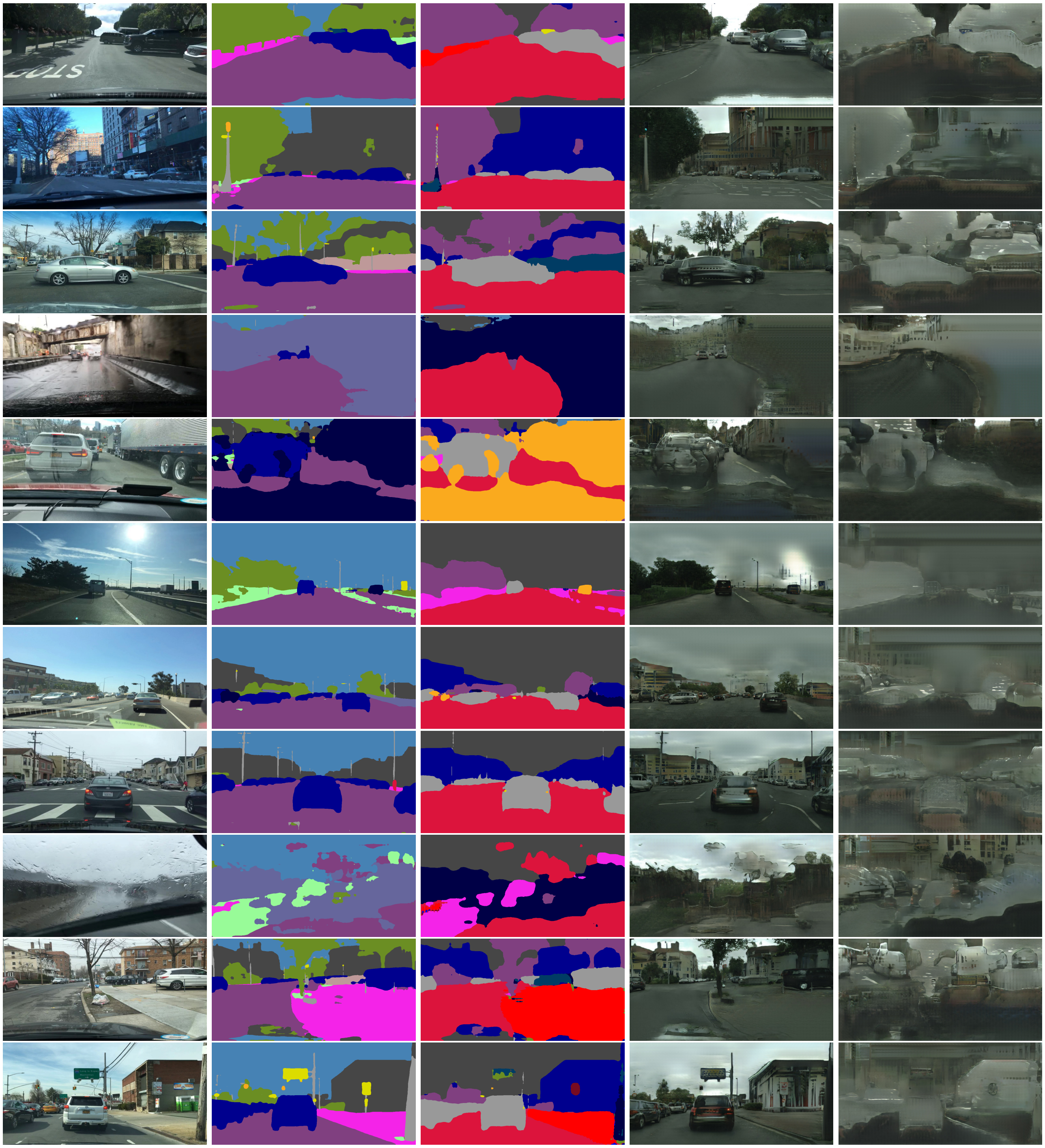


Figure 6: **Architecture of our discrepancy network.**



(a) Input image (normal) (b) Predicted map (normal) (c) Predicted map (Shift) (d) Resynthesized image (normal) (e) Resynthesized image (Shift)

Figure 7: **Detecting Houdini adversarial attacks on Cityscapes.** Without attack, the re-synthesized image **(d)** obtained from **(b)** looks similar to it. By contrast, the resynthesized image **(e)** obtained from the semantic maps **(c)** computed from a Houdini-compromised input differs massively from the original one.



(a) Input image (normal) (b) Predicted map (normal) (c) Predicted map (Shift) (d) Resynthesized image (normal) (e) Resynthesized image (Shift)

Figure 8: **Detecting DAG adversarial attacks on the BDD dataset.** Without attack, the re-synthesized image **(d)** obtained from **(b)** looks similar to it. By contrast, the resynthesized image **(e)** obtained from the semantic maps **(c)** computed from a DAG-compromised input differs massively from the original one.