MACHINE LEARNING FOR CFRP QUALITY CONTROL

Sebastian Zambal, Christoph Heindl, Christian Eitzinger
PROFACTOR GmbH
Im Stadtgut A2
4407 Steyr, Austria
E-mail: sebastian.zambal@profactor.at

ABSTRACT

Automation in CFRP production poses multiple challenges. The material at hand is very un-isotropic and deformable, leading to various difficulties in handling. We believe that visual inspection and quality control are key technologies to improve automation in CFRP production. In this paper, we point out possible ways to exploit modern machine learning methods in the context of CFRP quality control. Taking the example of AFP, we show how to transform prior knowledge about the production process into a probabilistic model. By drawing samples from this model, we demonstrate how to infer hidden variables of the process efficiently. We show how to use the methodology to perform inline defect detection and to reconstruct global process parameters. We present results for artificial and selected real AFP monitoring data acquired during inline process monitoring.

1. INTRODUCTION

A lot of effort is put into making CFRP manufacturing more efficient. Nevertheless, the level of automation still is relatively low, compared to other branches of industry. We believe that this relates to the fact that handling of carbon fiber material is quite challenging. Flexibility and high anisotropy of fibrous materials results in the need for sophisticated sensor technology.

Relatively little work has been published related to AFP monitoring systems. Technologies used so far include thermographic imaging (1., 2.) and laser triangulation (3.) Mostly, this work focuses on the hardware components inspection systems. For automated evaluation of sensor data, relatively simple algorithms based on filtering were proposed. In the present work, we propose the use of machine learning to fill this gap. Here, we stick to the use of laser triangulation as sensor technology. However, the data processing approach is extensible to other inspection technologies like thermography.

We argue that it is necessary to introduce powerful data processing methods in order to achieve high defect detection rates for systems that perform automated quality control. The data generated by many sensor systems include uncertainties that are subtle and need careful analysis to perform the correct interpretation. The data processing approach that we propose build upon modern machine learning methods. We deploy a probabilistic model that describes the observed data in combination with a deep neural network to perform inference. Our work is based on recent work (4.) We extend this method by adding the possibility to infer global parameters in addition to performing local surface characterization.

There are currently two important trends in machine learning. The first is deep artificial neural networks. Such networks have shown tremendous performance in many problem domains, which were considered too complex before. The second big trend in machine vision relates to probabilistic models. Probabilistic concepts where not considered fruitful by many researchers in the early days of artificial intelligence. Nowadays, however, probability is widely accepted as the right language to describe many soft problems such as distinguishing between different types of defects showing up in sensor data.

Neural networks are usually very fast when doing inference. Once the network is fully trained (which may take some time), it is in general very fast to apply the network to a new sample. On the other hand, it is very hard to integrate domain expert knowledge into a neural network. The network somehow acts as a black box. Probabilistic (graphical) models on the other hand provide the possibility to explicitly model entities (i.e. random variables) and their relationships (e.g. conditional probabilities in Bayesian Networks) of some system or process. The drawback of such models is that typically inference is computationally very expensive. For non-trivial models no algorithms for fast inference are available.

In this work we combine a Bayesian Network for data modelling with a deep neural network for fast inference. This approach is sometimes referred to as “analysis-by-synthesis” in the literature. It has been used in different fields in the past. To perform face recognition, a system based on renderings of photorealistic images was proposed (5.) Similar ideas were used to develop a system for human hand localization and 3d reconstruction (6.) Another interesting application domain is autonomous driving where generative models were deployed in a similar way (7., 8.)

We extend an existing method (4.) to perform not only segmentation, but to also infer global parameters of the model that generates the data. We propose a neural network architecture that is able to predict tow width at a high accuracy while segmentation performance does not suffer. Essentially, the network that we propose is a multi-task neural network (9.)

1. AFP SENSOR DATA

1.1 Inline Laser Triangulation Sensor

Input to our data processing system comes from a laser triangulation sensor integrated into an AFP lay-up head (Figure 1). The system has a field of view of approximately 120 mm width and 53 mm height (i.e. perpendicular to the tangent plane of the surface at lay-up).

The sensor principle is laser triangulation. A laser line is projected onto the tows right after they are placed by the lay-up head. A camera records images and the laser line is extracted. The result is a height profile of the laser line projected onto the surface for each acquisition. Based on the relative alignment of laser plane and camera, the detected profiles may be converted to a 3d point cloud relative to the sensor. However, we propose not to operate on 3d point clouds to perform data analysis. Instead, we accumulate multiple profiles to create a 2d height map of the surface.

In general, we are interested in detection and localization of different defects. We target at gaps (larger distances between neighboring tows), overlaps (regions where neighboring tows overlap), and fuzz balls (small carbon fibers that accumulate and accidentally fall onto the surface during lay-up). Besides these three types of defects, we define the remaining regions in the recorded height maps as “regular tow”.

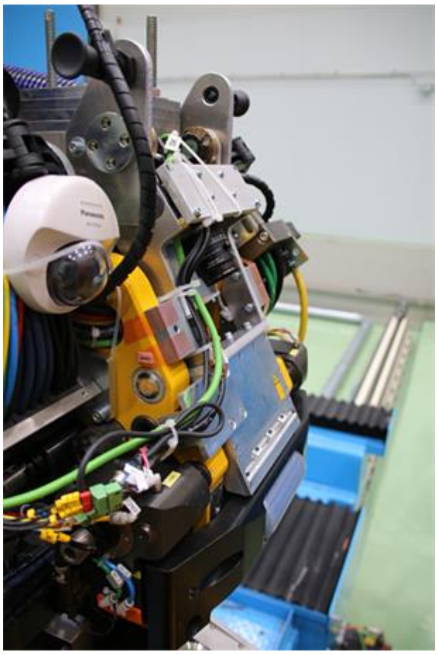

Figure 1: Laser-triangulation sensor on AFP lay-up head.

1.2 Probabilistic Data Model

Our implementation of the data generating model largely follows that of (4.) We model individual entities as probability variables in a probabilistic graphical model. Relationships are modelled as conditional probabilities between these variables. A simplified representation of our model is outlined in Figure 2. We assume that the tow width T, which will be observed by the inspection system, is independent of other random variables in the model. Therefore, T has no incoming arrows. However, T influences other variables Z. While there actually is a large number of hidden variables with different dependencies, we simply represent these by a single node Z. Z can be interpreted as the actual configuration of the inspected carbon fiber surface. Z directly influences the labels Y that are assigned to individual pixels. Z also influences the actual height values X as observed by the laser triangulation system. The gray background of node X indicates that this is an observed variable.

The beauty of a probabilistic graphical model is, that it nicely shows probabilistic relationships in an intuitive way. The directed nature of Bayesian Networks makes it easy to understand the “forward” reasoning in that model. Following the arrows in the model (which represent conditional probability distributions), it is easily possible to sample from that model. However, it is in general computationally very hard to do inference in the backward direction, i.e. determine probabilities for hidden variables given the observed ones. This is especially true when conditional probability distributions are very complex.

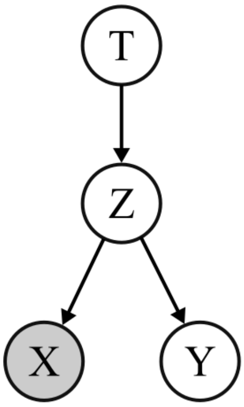

Figure 2: Probabilistic Graphical Model for data generation (simplified). The random variables shown here are a global parameter for tow width T, a hidden state that represents the real configuration of the surface Z, observed X and unobserved Y, which acts as ground-truth information during training.

In this work we consider the problem of finding the most likely configuration of hidden variables given the observed variables. The observations we make are the height maps coming from the laser triangulation system. The hidden variables we are interested in are the hidden tow width T together with the vector of pixel-wise labels Y. Y indicates whether the respective pixels represent regular tow, gap, overlap, or fuzz ball. In the next section we outline the neural network that is used to perform inference of T and Y in combination.

2. INFERENCE WITH DEEP NEURAL NETWORK

As outlined in the previous chapter, we are interested in finding a reasonable configuration of hidden parameters that explains the observed sensor data best. To do this job, we employ a deep neural network.

2.1 Network architecture

The network architecture is based on an encoder-decoder layout. A height map acquired by the sensor system is the input to the neural network. The encoder part of the network consists of multiple encoding layers, each performing a reduction in spatial resolution while increasing the number of features. After the encoding stages, two separate paths are followed. These are outlined in the following.

The decoding path brings the spatial resolution back to the original input resolution. In order not to lose spatial accuracy, activations are forwarded from the encoding layers on the same level. Each encoding layer consists of the following operations: max pooling, convolution, padding, relu, convolution, padding and relu. Each decoding layer consists of: upscaling, padding, concatenation, convolution, padding, relu, convolution, padding, and relu. For more details we refer to (4.) The output of the encoding-decoding path in the network delivers a pixel-wise prediction of labels.

Three additional pairs of convolutions followed by relu realize the second path that follow the encoding stages. The final convolution reduces the number of features to two. The interpretation of these two low-resolution feature maps is such that one of them represents the prediction of tow width. The second one represents the confidentiality of the correctness of the corresponding tow width prediction.

All parameters of the network are contained in convolutions. No fully connected layers are used. This provides the advantage that the neural network adapts to different input sizes.

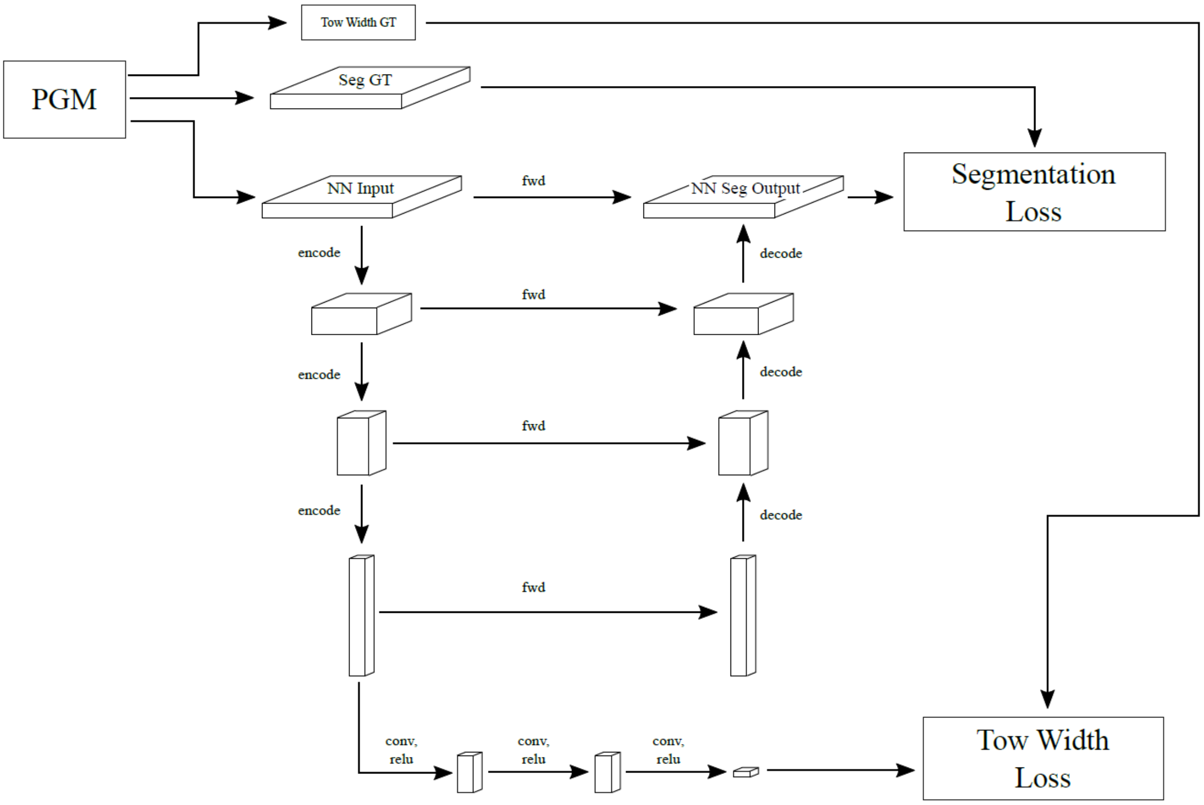

Figure 3: Multi-task neural network for inference. A probabilistic graphical model (PGM) provides network input and ground truth. The data takes two paths through the network: for segmentation and for prediction of tow width.

2.2 Loss function

Each of the two data processing paths through the neural network terminate in a separate loss function: Segmentation loss and tow width loss. The overall loss used for neural network training is the averaged sum of both losses. In the following, we outline segmentation and tow width loss in detail.

The segmentation loss describes how well the network predicts the correct label (“gap”, “overlap”, “regular tow”, “fuzz ball”) for all pixels. The network outputs the predicted labels in four channels, one for each possible label. The output of a single pixel is converted to a probability distribution via the softmax function first. Then, the cross entropy with the ground truth is calculated for each pixel. The average cross entropy over all pixels represents the final segmentation loss.

The loss for prediction of tow width is used to assess how well the network predicts the average tow width that shows up in the data. Because the layout of the network was chosen to be fully convolutional (i.e. no fully connected layers), the spatial size of the map that represent tow width depends on the spatial size of the input to the network. Therefore, we need to convert the spatially distributed tow width predictions to a single value. We define the first output feature map **w** as the actual tow width prediction. The second feature map **c** is defined as the expected error for the corresponding entry in **w**, i.e. $\mathbf{c}_{i,j}$ describes the predicted absolute error of the estimation **w**$_{i,\,j}$ of tow width. i and j denote the spatial position of the prediction within the feature maps. The tow width loss is defined as:

$$L_{tw} = \frac{1}{N}\sum_{i,j}(\epsilon_{i,j} * \lambda + \eta_{i,j})$$

with

$$\epsilon_{i,j} = |w_{i,j} - w_{gt}|$$

and

$$\eta_{i,j} = |c_{i,j} - \epsilon_{i,j}|$$

where N is the total number of pixels. The parameter λ describes the relative weight of predicted tow width and its error. We use λ = 3, i.e. the predicted tow width is three times more important than the corresponding predicted absolute error.

3. RESULTS

We present results on purely artificial data and selected data from a real sensor system. Both types of evaluation are based on the same neural network trained exclusively on a set of 5000 artificial data samples. Two examples of these with different tow width (44.2 and 112.4 pixels) are shown in Figure 4.

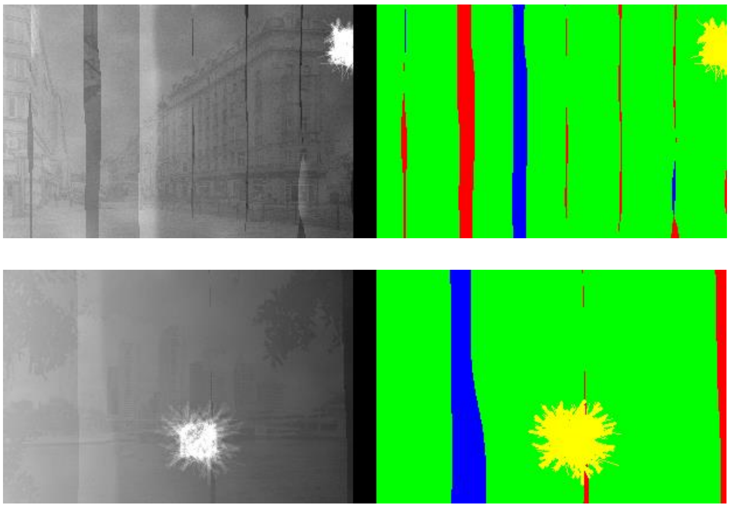

Figure 4: Two examples of artificially generated data sets: normalized height map (left) and different labels for image regions (right). Colors represent regular tow (green), gap (red), overlap(blue), and fuzzball (yellow). The example on top is based on tow width of 44.2 pixels. The bottom example is based on tow width of 112.4 pixels.

3.1 Evaluation for artificial test data

We generate 100 separate test samples for evaluation. The generative model is the same as that to generate the 5000 samples for neural network training. However, a different random seed is chosen for the test samples. Therefore, the test samples are different although drawn from the same distribution.

We feed the test samples to the neural network and record the predicted tow width values. We also record the ground truth of the tow width as used in the generative model. Figure 5 shows the ground truth values on the horizontal axis. The vertical axis shows the output of the neural network. There is a good correlation between the values. The mean absolute error of the predictions for all 100 test samples is 0.77 pixel. The maximum absolute error is 7.02 pixel.

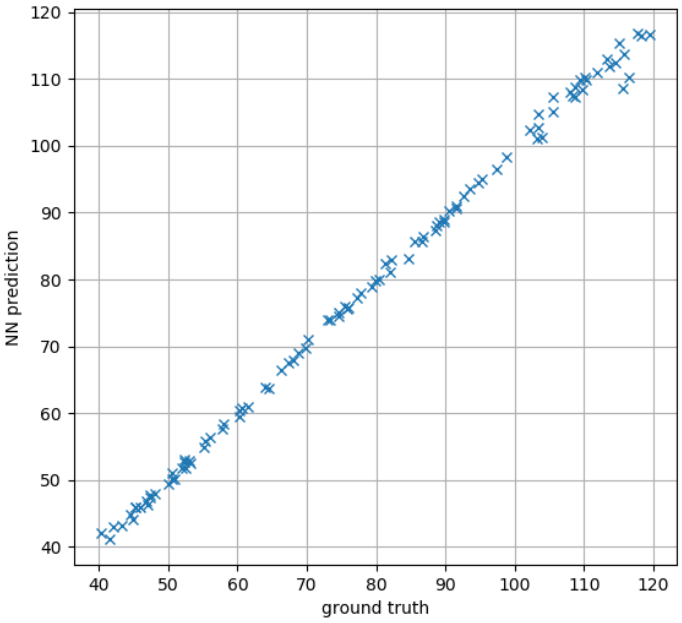

Figure 5: Inference of tow width in pixels. The horizontal axis shows values of the true parameter T. The vertical axis represents predictions as output of the neural network.

In parallel to the accuracy for predictions of the global tow width, we evaluate segmentation performance. Table 1 shows this result as a confusion matrix. Across all 100 test samples, 99.62% of pixels were correctly classified (diagonal elements of the confusion matrix).

		Ground truth				
		Gap	Tow	Overlap	Fuzzball	Σ
NN prediction	Gap	8.08	0.04	0.000000	0.00	8.12
	Tow	0.06	80.80	0.10	0.02	80.97
	Overlap	0.00	0.12	5.909983	0.00	6.03
	Fuzzball	0.00	0.04	0.004167	4.84	4.88
	Σ	8.14	90.98	6.01	4.86	100.00

Table 1: Confusion matrix for over all 100 artificial test samples. The numbers represent the percentage of pixels. Correctly classified pixels show up in the diagonal of the confusion matrix.

Figure 6 shows an example for the segmentation output calculated by the neural network. Only very few pixels are given wrong labels.

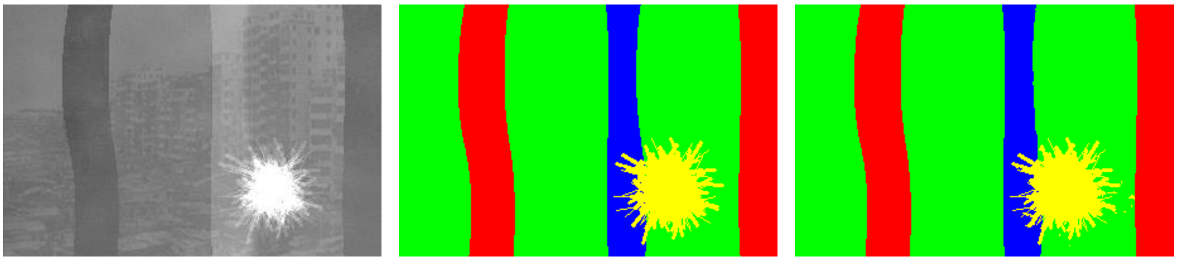

Figure 6: Example for segmentation calculated by the neural network on unseen artificial data. Normalized input range image (left), ground truth segmentation (center), and neural network output (right). Color coding is the same as in Figure 4.

3.2 Evaluation on real data

We evaluate our method also on a single sample of real sensor data. Other than for artificial data, ground truth cannot be derived from a generative model. Instead, the ground truth for segmentation comes from manual labelling. Figure 7 shows the segmentation result. Compared to the manual labelling, the neural network predicts labels for 95.4% of pixels correctly.

The ground truth for tow width is manually defined as 109 pixels. Three horizontal lines with this length are drawn at different locations over the top image in Figure 7. The prediction of the neural network is 112.7 pixel for this input data. This corresponds to an error of 3.7 pixel, i.e. 3.3% of the ground truth value.

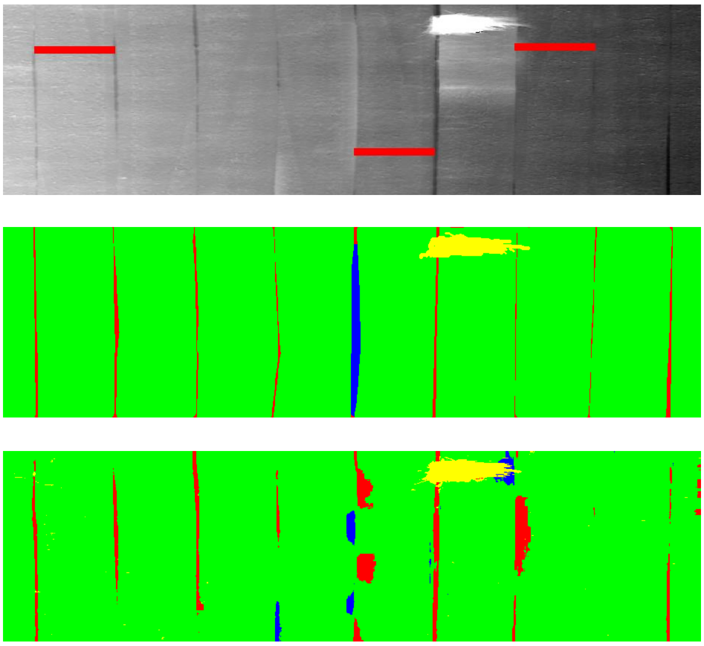

Figure 7: Example for segmentation calculated by the neural network on unseen artificial data. Normalized input range image (left), ground truth segmentation (center), and neural network output (right). Color coding is the same as in Figure 4.

4. CONCLUSIONS

In this work, we extend the concept of analysis-by-synthesis to data processing for inline monitoring of AFP processes. We model data that is acquired by the monitoring system in an explicit and probabilistic way. A neural network infers hidden variables from observed values. Hidden variables can be of different type: local (i.e. per-pixel) in case of segmentation or global parameters (e.g. tow width).

The fact that a probabilistic model generates complete data samples makes tedious manual labelling of training data obsolete. Changes in the data (e.g. due to modifications of the sensor hardware) would usually require tedious acquisition and labelling of new data. With the proposed method it is only necessary to adapt the generative model in order to adapt the system.

Although the method is promising and has several advantages, there is one critical aspect. In general, there is some discrepancy between the real process and the probabilistic model. Interesting future research questions relate to how this discrepancy can be understood, quantified, and minimized.

5. ACKNOWLEDGMENTS

Work presented in this paper has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 721362 (project "ZAero") and by the European Union in cooperation with the State of Upper Austria within the project "Investition in Wachstum und Beschäftigung" (IWB).

6. REFERENCES

1. P. D. Juarez, K. E. Cramer, and J. P. Seebo, "Advances in in situ inspection of automated fiber placement systems", in Thermosense: Thermal Infrared Applications XXXVIII, 9861, p. 986109, International Society for Optics and Photonics, 2016.
2. B. Denkena, C. Schmidt, K. Völtzer, and T. Hocke, "Thermographic online monitoring system for automated fiber placement processes", Composites Part B: Engineering 97, pp. 239-243, 2016.
3. J. Cemenska, T. Rudberg, and M. Henscheid, "Automated in-process inspection system for AFP machines", SAE International Journal of Aerospace 8(2015-01-2608), pp. 303-309, 2015.
4. S. Zambal, C. Heindl, C. Eitzinger, and Josef Scharinger, "End-to-End Defect Detection in Automated Fiber Placement Based on Artificially Generated Data", International Conference on Quality Control by Artificial Vision (QCAV), 2019.
5. I. Yildirim, T. D. Kulkarni and W. Freiwald and J. B. Tenenbaum, "Efficient analysis-by-synthesis in vision: A computational framework", Annual Meeting of the Cognitive Science Society (CogSci), 2015.
6. F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "GANerated hands for real-time 3d hand tracking from monocular RGB", in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
7. A. Prakash, S. Boochoon, D. Acuna, E. Cameracci, G. State, O. Shapira, S. Birchfeld, "Structured Domain Randomization: Bridging the Reality Gap by Context-Aware Synthetic Data", International Conference on Robotics and Automation (ICRA), 2019.

8. M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?", in IEEE International Conference on Robotics and Automation (ICRA), pp. 746-753, 2017.

9. X. Li, L. Zhao, L. Wei, M. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang, "DeepSaliency: Multi-Task Deep Neural Network Model for Salient Object Detection", IEEE Transactions on Image Processing 25(8), pp. 3919-3930, 2016.