Formwork detection in UAV pictures of construction sites

Katrin Jahr, Alexander Braun & André Borrmann
Chair of Computational Modeling and Simulation, Technical University of Munich, Germany

ABSTRACT:

The monitoring of the construction progress is an essential task on construction sites, which nowadays is conducted mostly by hand. Recent image processing techniques provide a promising approach for reducing manual labor on site. While modern machine learning algorithms such as convolutional neural networks have proven to be of sublime value in other application fields, they are widely neglected by the CAE industry so far. In this paper, we propose a strategy to set up a machine learning routine to detect construction elements on UAV photographs of construction sites. In an accompanying case study using 750 photographs containing nearly 10.000 formwork elements, we reached accuracies of 90% when classifying single object images and 40% when locating formwork on multi-object images.

1 INTRODUCTION

The digitization of the construction industry offers various new possibilities for the planning, monitoring, and design process of buildings. In recent years, many research projects are focusing on using methods of computer-aided engineering, such as building information modeling or structural simulations, to facilitate and enhance the planning process. However, as of now, not many of the advantages of using digital support are used after the planning of a construction has been finished. While monitoring of the construction progress by comparing planned conditions to the actual situation is a labor-intensive task, it is still mostly conducted by the workforce on site with little technical support.

During the last decades, image processing techniques have been increasingly adopted by the construction industry, greatly improving and facilitating the process of construction monitoring. These methods gained new potential due to the more affordable and precise acquisition devices like unmanned aerial vehicles (UAVs) or laser scanners. Using the resulting 3D point clouds and information retrieved from the building information model, the possibility to track the progress of construction sites arises (Golparvar-fard, Pena-Mora, and Savarese 2009; Braun et al. 2015).

A detailed geometric as-planned vs. as-built comparison allows to track the current progress of a construction site, assess the quality of the construction work, and to check for construction defects such as cracks.

To generate high-quality point clouds, a significant number of consecutive photographs covering the monitored area is needed, requiring extensive image capturing and processing. However, most monitoring tasks do not entail the need for detailed 3D information. These include the monitoring of the quantity and positions of site equipment, of externally stored construction material, and major construction phases.

The image analysis and object detection on aerial photographs, which can be taken with relatively low effort, offers an alternative to expensively generating 3D point clouds. The scientific field of computer vision provides different solutions to process and, to a certain extent, understand images.

In this contribution, we use two state-of-the-art techniques of image processing to analyze aerial photography of construction sites. On the example of formwork elements, we demonstrate an artificial in-

telligence approach to recognize and locate construction elements on site. In the first part of the paper, we give an overview of the state of the art in image analysis as used on construction sites today, followed by a further description of the used methodology. We conclude the paper with a proof of concept and a summary of our results.

2 STATE OF THE ART

Computer Vision is a heavily researched topic, that got even more attention through recent advances in autonomous driving and machine learning related topics. Image analysis on construction sites, on the other hand, is a rather new topic. Since one of the key aspects of machine learning is the collection of large datasets, current approaches focus on data gathering. In the scope of automated progress monitoring, Han et al. published an approach for Amazon Turk based labeling (Han and Golparvar-Fard 2017). (Kropp, Koch, and König 2018) tried to detect indoor construction elements based on similarities, focusing on radiators.

For effective and efficient image analysis and object recognition, machine learning algorithms have been increasingly used during the last decades. In 2012, the convolutional neural network (CNN) "AlexNet" (Krizhevsky, Sutskever, and Hinton 2017) achieved a top-5 error of 15.3% in the prestigious ImageNet Large Scale Visual Recognition Challenge (Russakovsky et al. 2015). These results were surprisingly accurate at the time, proving the advantages of using CNN. On this account, the software industry shifted towards using CNN for all machine learning based image processing tasks (LeCun, Bengio, and Hinton 2015).

There are different tasks to be solved by image processing algorithms. Well known problems include classification, where single-object images are analyzed, object detection, where several objects in one image may be classified and localized within the image, and image segmentation, where each pixel of an image is classified (Buduma 2017). In this paper, we focus on image classification and object detection.

CNNs are structured in locally interconnected layers with shared weights. Each layer comprises multiple calculation units (called neurons). The neurons of the first layer (input layer) represent the pixels of the analyzed image, the last layer (output layer) comprises the predictable object classes.

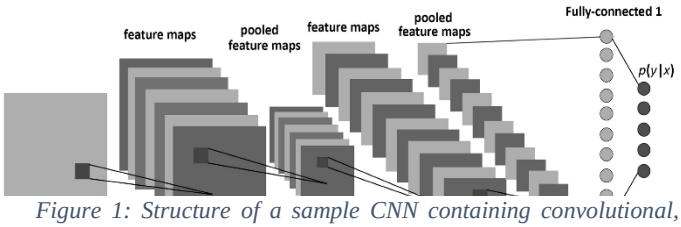


Figure 1: Structure of a sample CNN containing convolutional, pooling and fully connected layers.

In between input and output layer, any number of hidden layers can be arranged. While AlexNet contained 8 hidden layers, GoogLeNet (Szegedy et al. 2015), and Microsoft ResNet (He et al. 2016) use more than 100 hidden layers. The layers are usually convolution layers (sharpening features), pooling layers (discarding unnecessary information), or fully connected layers (enabling classification) (Buduma 2017; Albelwi and Mahmood 2017).

To adapt to different problems, such as recognizing formwork elements on images, CNNs must be trained. During training, the connections between certain neurons are increased, while the connections between other neurons are reduced–the weights connection consecutive layers are weighted. The training is usually carried out using supervised backpropagation, meaning that the network is fed with example input-output pairs (Buduma 2017). The correct solution for each input is called ground truth. To train a CNN towards reliable predictions, a significant amount of training data is required, which has to be prepared in a preprocessing step. ImageNet provides around 1.000 images per class, for example (Russakovsky et al. 2015). To accelerate the training processes, weights of previously trained CNNs can be used. To adapt pretrained CNNs, the fully connected layers are replaced with layers representing the new data and trained with the new data.

3 METHODOLOGY

In the context of the introduced research topics, the paper focusses on the image-based detection of temporary construction elements such as formwork. The detection of recurring, similar objects can be solved by machine-learning approaches. Several tools support the image analysis regarding automated detection of pretrained image sets.

3.1 *Image classification using CNN*

During image classification, which is also known as image recognition, images that contain but exactly one object are classified. Each class, that the CNN can detect, is represented by one output neuron. The activity of the neurons is read as the probability that the image contains an object of the corresponding class. Image classification algorithms will fail on images containing multiple objects. As images of construction sites contain more than one object, image classification algorithms can only be applied after preprocessing of the data. However, they can be very useful to confirm certain questions, e.g. if a wall with a known position is missing, currently shuttered or finished.

3.2 *Object detection using CNN*

The evident solution to analyze multi-object images is using a sliding window on the image and run an image classification on each window, which is computationally very expensive. Different proposals have been made to reduce the computational effort, e.g. region-proposal networks (e.g. R-CNN, (Girshick et al. 2014), (Girshick 2015), (Ren et al. 2017)), which intelligently detect regions of interest within an image and analyze those further, and single shot detectors (e.g. DetectNet (Tao, Barker, and Sarathy 2016) and YOLO (Redmon et al. 2016), (Redmon and Farhadi 2017), (Redmon and Farhadi 2018)), which overlay the image with a grid and analyze each cell.

3.3 *Evaluation of CNN*

To measure the performance of an image classifying CNN, the top-1 error and top-5 error are used. The top-1 error represents the fraction of images, for which the correct class has been predicted with the highest probability. The top-5 error is the fraction of images, for which the correct class is within the 5 classes that have been predicted with the highest probability, accordingly.

To measure the performance of an object detecting CNN, precision *p*, recall *r* and mean average precision *mAP* can be used. They are calculated using the number of true positives *TP*, false positives *FP* and false negatives *FN*:

$$p = \frac{TP}{TP + FP} \qquad r = \frac{TP}{TP + FN}$$

In object detection tasks, a prediction is counted as true positive, if it has an intersection over union *IoU* of a distinct value, usually over 0.5, meaning that more than 50% of the predicted bounding box should overlap the ground truth bounding box (see Figure 2):

$$IoU = \frac{area\ of\ overlap}{area\ of\ union}$$

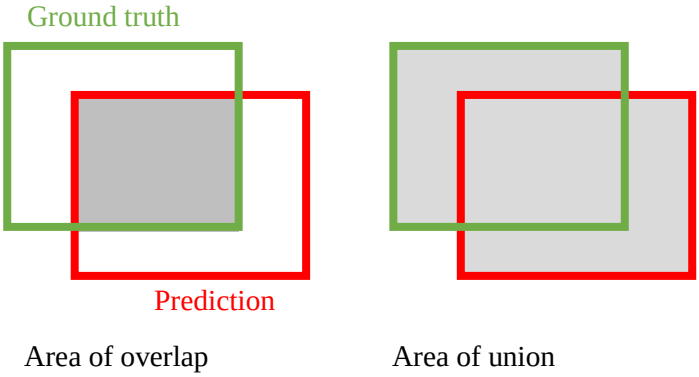


Figure 2: Area of overlap and area of union for predicted and labeled bounding boxes

For object detection, the mAP is the average of the possible precision at different recall values across all classes. To calculate the AP for each class, (Russakovsky et al. 2015) propose to consider 11 recall values according to the proposal by ImageNet:

$$AP = \frac{1}{11} \sum_{r \in \{0.0,\ldots,1.0\}} p_i\,(r)$$

With p_i = maximum precision for any recall value exceeding *r*.

3.4 *Labeling*

Labeling defines the approach of marking all regions of interest in a set of pictures and defining the type of the marked region. A subset of labeled pictures is depicted in Figure 5 a). The labels are marked with green bounding boxes.

As the labeling work takes a lot of time, a novel approach for automated labeling has been introduced by (Braun et al., 2018). In the frame of the research project ProgressTrack focusing on automated progress monitoring with photogrammetric point clouds, an algorithm has been developed to validate detection results of the as-built vs. as-planned comparison. As depicted in Figure 3, the projected 2D geometry of construction elements can be transformed from the building information model's coordinate system into the 2D coordinate system of each picture, the element is included in. This is possible, as the pictures were aligned and oriented during the photogrammetric process and thus making it possible to know the exact position in relation to the Building Information Model.

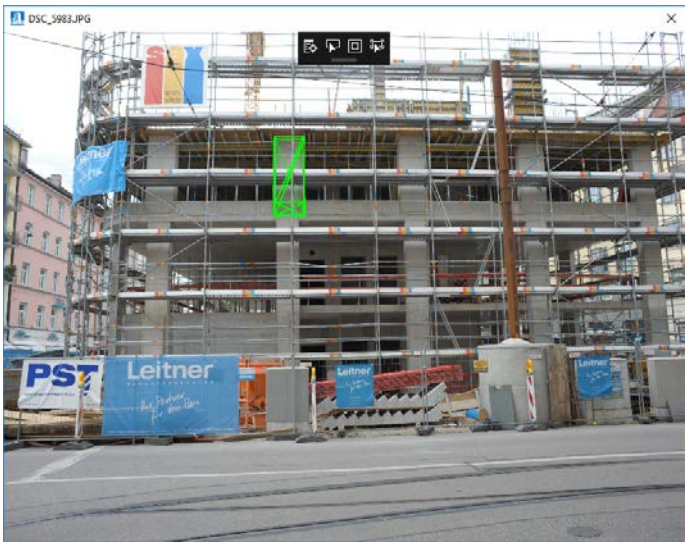
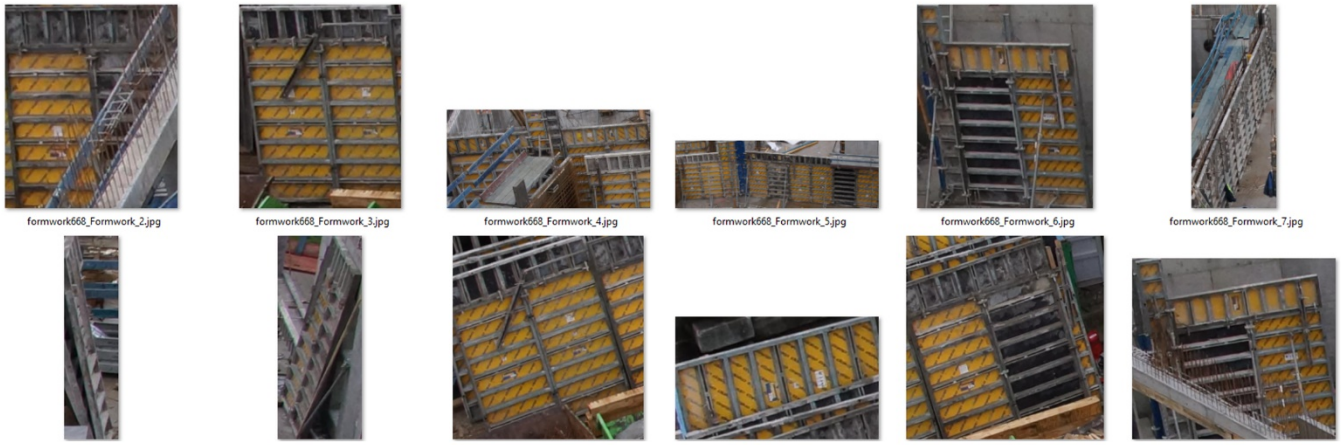


Figure 3: Reprojected bounding box of a column on a picture gathered during acquisition

a) Labeled images

b) Image patches for classification



c) Prepared snippets for DetectNet

Figure 2: Sample data from a) labeling, b) image snippets for classification, as well as c) snippets for DetectNet

The process of labeling can benefit from this work because by this method, labels for all building element can be marked in all pictures, that were taken and aligned accordingly. Future research will focus on this method to extract labels for all construction elements and train a CNN accordingly without the time consuming, manual labeling work to be done.

4 CASE STUDY

In the following sections, we present an image analysis routine including data preparation as well as the training of convolutional neural networks to be able to recognize formwork elements. We focus on two different image analysis tasks: image classification and object detection.

4.1 *Data preparation*

As an initial dataset, 9.956 formwork elements were labeled manually on pictures of three construction sites that were collected during different case studies in the recent years. The images contain formwork elements from two different, German manufacturers and vary in size (30cm up to 2,70m length) as well as color (red, yellow, black, grey). They were taken at varying weather conditions on partly cloudy, as well as sunny days. The image acquisition was achieved with aerial photography by different UAVs, but also from the ground with regular digital cameras, resulting in image sizes from 4000 x 3000 px up to 6000 x 4000 px. The manual labeling process for this data set took around 130h to complete.

The gathered data is processed as plain text files for each picture and processed for the various neural networks according to their respective requirements.

4.2 *Image analysis*

For image analysis, we used the Nvidia Deep Learning GPU Training System DIGITS (Yeager 2015), which provides a graphical web interface to the widespread machine learning frameworks TensorFlow, Caffe, and Torch (NVIDIA 2018). It enables data-management, network design and visualization of the training process.

4.2.1 *Image classification*

We used a standard GoogLeNet CNN implemented in Caffe for the image classification task. The training is performed using the Adam Solver (Kingma and Ba 2014). We retrieved a classification dataset of formwork elements from the labeled data (Section 4.1) by automatically trimming the images around the bounding boxes of the labeled formwork elements (see a subset in Figure 4 b)). The automation was achieved by a self-written tool that takes all labeled data and images as input and crops them automatically. The tool is made available on GitHub as an OpenSource solution[1]. To assure relatively even image sizes with sufficient detailing, we removed all images with resulting dimensions under 200 x 200 pixels.

To train the algorithm not only on formwork elements but on several classes, we added seven classes (see Table 1) that are related to construction sites from the Caltech 256 dataset (Griffin, Holub, and Perona 2007). The Caltech 256 provides single object images of 256 classes that need no further preprocessing for image classification.

Table 1: Classes and number of images per class used for training of an image classification CNN

Class	Origin	Number of images
Barrel	Caltech 256	47
Bulldozer	Caltech 256	110
Car	Caltech 256	123
Chair	Caltech 256	62
Formwork	Own dataset	1410
Screwdriver	Caltech 256	102
Wheelbarrow	Caltech 256	91
Wrench	Caltech 256	39

As GoogLeNet requests input images of 256 x 256 pixels, all images are resized to that dimensions by DIGITS. For image classification, DIGITS automatically splits the data into training and validation data.

The CNN converged quickly towards high accuracies (top-1-error) around 85% (Figure 4) and stagnated at 90% after 100 epochs, which is a satisfying result. To achieve even higher accuracies throughout all classes, the number of images per class could be evened out by adding additional images to the underrepresented classes of the training data in future work.

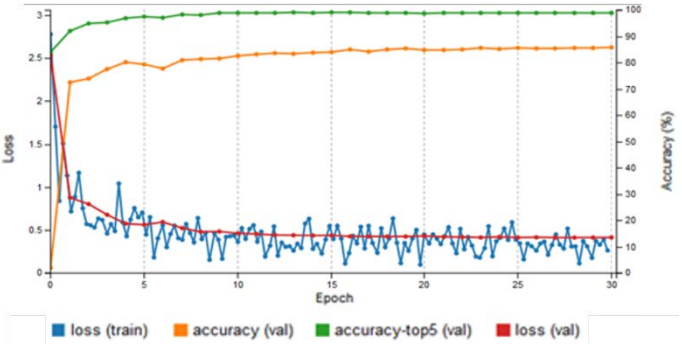


Figure 3: Loss and accuracy of the GoogLeNet after 30 epochs of training for classifying images of formwork elements and objects typically found on construction sites.

[1] https://github.com/tumcms/Labelbox2DetectNet

4.3 *Object detection*

As next step, an object detection algorithm is introduced, to exactly detect certain elements in images and also precisely find the position of these elements. For this purpose, the dataset depicted in Figure 4 c) is used. To detect several formworks within an image of a construction site, we used a CNN with DetectNet architecture, implemented in Caffe. To reduce training time, we used the weights of the "BVLC GoogleNet model"[2], which has been pretrained on ImageNet data. The training again is performed using the Adam Solver.

We split the labeled images into 85% of training data and 15% of validation data. The images were recorded at a high resolution between 4000 x 3000 and 6000 x 4000 pixels. To minimize the necessary computational effort, we split the images into smaller patches with a size of 1248 x 384 pixels.

We trained the CNN twice with 300 epochs each. Both precision and recall reached values around 63%, the mAP stagnated around 44% (Figure 5). The network manages to detect most formwork elements correctly with low rates of false detections. In Figure 6, the resulting bounding box for one example image is depicted. For this image, a very good result was retrieved.

Further steps to improve the object detection algorithm entail more extensive preprocessing of the data, longer training periods and adjustments of both the network architecture and the solving algorithms.

Table 2: Number of images and number of formwork elements contained in that images for training and validation of the object detection

Purpose	Nr. of images	Nr. of formwork elements
Training	646	8429
Validation	99	1487

5 SUMMARY

The presented research focusses on image analysis of construction site images. To make automated assumptions on the construction elements depicted on an image, machine learning tools need to be trained. First, the current state of the art for machine learning approaches is introduced and examined for their suitability of application in the domain of construction.

Then, these approaches are tested on construction site elements. For the training, 750 images of construction sites were labeled, resulting in nearly 10.000 labeled formwork elements. The images were used as input to various classification and detection algorithms, resulting in very high success rates for the classification of single object images and mediocre success rates for object detection on multi-object images. However, as object detection is a highly demanding task concerning a large community of researchers, the results give a promising starting point for future improvements.

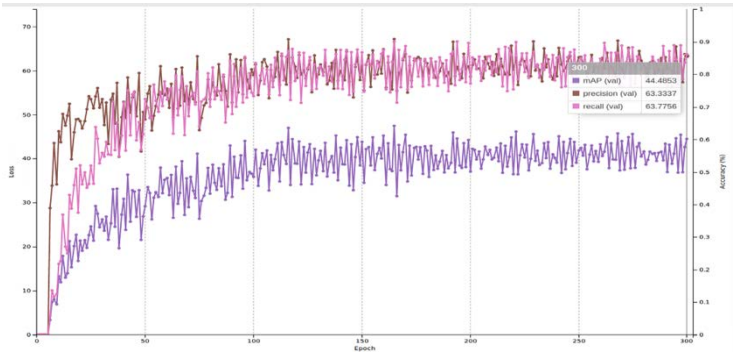


Figure 4: Precision, recall and mAP of the DetectNet after one round of 300 epochs of training for detecting formwork on images of construction sites.

Figure 5: Detected bounding box for formwork elements on a photography of a construction site.

6 ACKNOWLEDGMENTS

This work is supported by the Bavarian Research Foundation under grant 1156-15.

We thank the Leibniz Supercomputing Centre (LRZ) of the Bavarian Academy of Sciences and Humanities (BAdW) for the support and provisioning of high-performance computing infrastructure essential to this publication.

7 REFERENCES

Albelwi, Saleh, and Ausif Mahmood. 2017. "A Framework for Designing the Architectures of Deep Convolutional Neural Networks." *Entropy* 19 (6): 242. doi:10.3390/e19060242.

Braun, Alexander, Sebastian Tuttas, André Borrmann, and Uwe

[2] Released for unrestricted use at https://github.com/NVIDIA/DIGITS/tree/master/examples/object-detection

Stilla. 2015. “A Concept for Automated Construction Progress Monitoring Using BIM-Based Geometric Constraints and Photogrammetric Point Clouds.” *ITcon* 20: 68–79.

Buduma, Nikhil. 2017. *Fundamentals of Deep Learning : Designing Next-Generation Machine Intelligence Algorithms*. Vol. 44. doi:10.1007/s13218-012-0198-z.

Girshick, Ross. 2015. “Fast R-CNN.” In *2015 IEEE International Conference on Computer Vision (ICCV)*, 1440–48. IEEE. doi:10.1109/ICCV.2015.169.

Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation.” In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 580–87. IEEE. doi:10.1109/CVPR.2014.81.

Golparvar-fard, Mani, F Pena-Mora, and S Savarese. 2009. “D4AR - a 4 Dimensional Augmented Reality Model for Automation Construction Progress Monitoring Data Collection, Processing and Communication.” *Journal of Information Technology in Construction* 14 (June): 129–53.

Griffin, G., A. Holub, and P. Perona. 2007. “Caltech-256 Object Category Dataset.” http://www.vision.caltech.edu/Image_Datasets/Caltech256/.

Han, Kevin K., and Mani Golparvar-Fard. 2017. “Potential of Big Visual Data and Building Information Modeling for Construction Performance Analytics: An Exploratory Study.” *Automation in Construction* 73 (January): 184–98. doi:10.1016/j.autcon.2016.11.004.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. “Deep Residual Learning for Image Recognition.” In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–78. IEEE. doi:10.1109/CVPR.2016.90.

Kingma, Diederik P., and Jimmy Ba. 2014. “Adam: A Method for Stochastic Optimization,” December.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2017. “ImageNet Classification with Deep Convolutional Neural Networks.” *Communications of the ACM* 60 (6): 84–90. doi:10.1145/3065386.

Kropp, Christopher, Christian Koch, and Markus König. 2018. “Interior Construction State Recognition with 4D BIM Registered Image Sequences.” *Automation in Construction* 86 (February): 11–32. doi:10.1016/j.autcon.2017.10.027.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. “Deep Learning.” *Nature* 521 (7553): 436–44. doi:10.1038/nature14539.

NVIDIA. 2018. “Nvidia Digits - Deep Learning Digits Documentation,” no. May.

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. “You Only Look Once: Unified, Real-Time Object Detection.” In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–88. IEEE. doi:10.1109/CVPR.2016.91.

Redmon, Joseph, and Ali Farhadi. 2017. “YOLO9000: Better, Faster, Stronger.” In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517–25. IEEE. doi:10.1109/CVPR.2017.690.

Redmon, Joseph, and Ali Farhadi. 2018. “YOLOv3: An Incremental Improvement,” April.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. 2017. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (6): 1137–49. doi:10.1109/TPAMI.2016.2577031.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, et al. 2015. “ImageNet Large Scale Visual Recognition Challenge.” *International Journal of Computer Vision* 115 (3): 211–52. doi:10.1007/s11263-015-0816-y.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. “Going Deeper with Convolutions.” In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9. IEEE. doi:10.1109/CVPR.2015.7298594.

Tao, Andrew, Jon Barker, and Sriya Sarathy. 2016. “DetectNet: Deep Neural Network for Object Detection in DIGITS.” https://devblogs.nvidia.com/detectnet-deep-neural-network-object-detection-digits/.

Yeager, Luke. 2015. “DIGITS : The Deep Learning GPU Training System.” *ICML AutoML Workshop*.