Material Classification in Construction Sites

Moritz Ritzl
Karlsruhe Institute of Technology
unkxk@student.kit.edu

Laurenz Thiel
Karlsruhe Institute of Technology
ueecz@student.kit.edu

Abstract

Computer vision is being used in more and more areas of our life. However, it is still underrepresented in construction site contexts. But there are many possible applications in this area, for example, to automatically monitor the construction process and, thus, potentially improve efficiency. One possibility is to estimate the construction status based on the materials visible in a certain phase of construction. For example, the floor is initially made of concrete and will then be covered with wood in a later step. To monitor this progress, it is sufficient to take photographs of the construction site. These can then be analyzed in more detail. For this purpose, a pipeline was developed and implemented within the scope of this work. It uses a photograph of the construction site as input. The image is then segmented with respect to the spatial components of the room, which makes it possible to subsequently classify the material on the ground and, thus, obtain an approximation of the distribution of the materials used. The segmentation and classification are done with different CNNs. These were trained using different public datasets (ADE20k, OpenSurfaces, and MINC-2500). For testing, an own dataset, which consists of construction site images, was annotated. The results show that the segmentation works very well, but that there is still room for improvement in the classification.

1. Introduction

Building and monitoring the progress of construction sites has historically been a labor- and time-intensive task. Automating both processes to improve efficiency can be done by leveraging technological advancement. The construction process can be automated by using 3D-printing technology [4] to print parts of a building. The monitoring part can be automated by using computer vision. This is possible due to the technological advancement in cheap and good image capturing devices like smartphones, outdoor cameras etc. in the last decade. The first research in the area of computer vision on construction sites analyzed the color profile of the images to retrieve information about the materials [3]. But with the recent advancement in Convolutional Neural Networks (CNNs) as state-of-the-art image classifiers [15, 11, 18, 30, 25] this process can be improved. These CNNs can be trained on material databases like Open Surfaces [2], ImageNet [18] or OpenImages [13] and then be used to classify the different materials in the images of construction sites more accurately. With the classified materials, a logic can be composed which allows us to identify the progress on the construction side.

Figure 1: The two photographs show a construction site at an early and late stage in the construction progress.

We are interested in capturing the difference between multiple stages in the construction progress by identifying the materials in the different images. Figure 1 shows two photographs of different construction phases of a room. Our aim is to identify the materials in both images using a CNN classifier trained on a material database and, therefore, allow assessing how far the construction process is completed.

Our paper is structured as follows. Section 2 gives a background on prior work. In Section 3 we describe our methodology. Section 4 presents our results and Section 5 summarizes our work and gives an outlook into the future.

2. Related Work

Computer Vision on construction sites. Different forms of computer vision have been used to tackle the problem of identifying materials on images from construction sites. Brilakis *et al.* [3] proposed in 2005 the idea of analyzing the color profile of an image to recognize materials in the image. Zhu *et al.* [32] presented the idea of using edge detection and the Hough transform to identify

concrete columns based on their structure. Son *et al.* [23] benchmarked different machine learning algorithms that leverage color information for concrete detection. Dimitrov and Golaparvar-Fard [9] introduce a Construction Material Library with 20 different material categories and a method for discriminative classification of construction site materials. Han *et al.* [10] present an idea that uses a 4D Building Information Modeling (BIM) and 3D point clouds generated from site images to monitor the progress of the construction process. Deng *et al.* [8] use a support vector machine and combine it with a BIM to track the progress of tiling in a building.

Material Databases. A lot of early work on material recognition focused on classifying instances of textures or material samples. The CUReT [7] database contains 61 materials, each of them captured under 205 different viewing and lighting conditions. This lead to research focused on instance-level texture or material classification [26]. In the area of categorical material databases, Sharan *et al.* [21] released the Flickr Material Database (FMD). FMD contains ten different material categories, each with 100 samples drawn from Flickr images. The different images in the database illustrate a wide range of appearances for these categories. Subsequently, Bell *et al.* released OpenSurfaces [1] which contains over 20,000 scenes from the real world. The scenes are labeled with both materials and objects, using a multi-stage crowdsourcing pipeline. Bell *et al.* use OpenSurfaces as a foundation for their Materials in Context Database (MINC) [2]. MINC contains 23 different materials. It provides patches from the different segments which can be used to train image classifiers. Our work is based on OpenSurfaces and MINC and which are described in detail in Section 4.1.

Material recognition. Prior Work on material recognition has focused on the classification problem (identifying the material on an image patch). Liu *et al.* [16] used FMD and introduced a reflectance-based edge features together with general image features. Hu *et al.* [12] extracted features based on variances of oriented gradients. Qi *et al.* [17] proposed a pairwise local binary pattern (LBP) feature. Nishino and Schwartz [19] presented the idea of material traits that incorporate learned convolutional auto-encoder features. Xue *et al.* [29] introduced a new approach for material recognition called texture-encoded angular network (TEAN). TEAN is based on the Ground Terrain in Outdoor Scenes (GTOS) [28] database that combines deep encoding pooling of RGB information and differential angular images angular-gradient features to utilize the GTOS dataset.

Convolutional neural networks. CNNs have been around for some decades already. Early networks like the LeNet [14] already were successful, but recently the advancements in the research community led to state-of-the-art results in object classification and detection. Driven by the ILSVRC challenge [18], with more recent architectures, including GoogLeNet [24], we have seen several successful CNN architectures [22, 20], led by Russakovsky *et al.* [18] work on their SuperVision (a.k.a. AlexNet) network. He *et al.* [11] proposed a residual network architecture (ResNet), which allows networks to be substantially deeper than before. Zagoruyko and Komodakis [30] refined the ResNet architecture by widening the network, which leads to improved performance and accuracy. Tan and Le [25] introduced in their EfficientNet architecture a technique for compound model scaling, leading to smaller but more efficient networks with state-of-the-art performance. We adapted the wide ResNet and EfficientNet to solve our problem of material classification.

3. Method

We present our model to segment and classify our images from the construction site. Our goal is to segment different parts of the image using a semantic segmentation model. With the output of the semantic segmentation model, we apply a patch classifier to retrieve the different materials displayed in the construction site image.

3.1. Semantic Segmentation

Semantic segmentation is the task of clustering parts of an image together that belong to the same object class. It is a form of pixel-level prediction because each pixel in an image is classified according to a category. We compare the performance of two different state-of-the-art semantic segmentation models (HRNetV2 [31] and Deeplab V3+ [5]) on our dataset. Both were trained on the ADE20K [27] dataset. The ADE20K dataset consists of images with indoor and outdoor scenes with 150 object classes. It also includes the classes ceiling, floor ,and wall which are relevant for our work. We benchmarked the two models on our dataset and their performance is shown in Section 4.3.

3.2. Patch Classifier

For training CNNs and other types of classifiers, it is helpful to have data in the form of fixed-sized patches. We adapted the idea of a patch classification from Ciresan *et al.* [6]. Our classification pipeline is visualized in Figure 2. It starts with the semantic segmentation, after the input image is segmented we cut patches out of each segment and then feed that into our CNN classifier. After each patch is classified we compare the result of the CNN against the ground truth. For calculating the accuracy we used two different metrics:

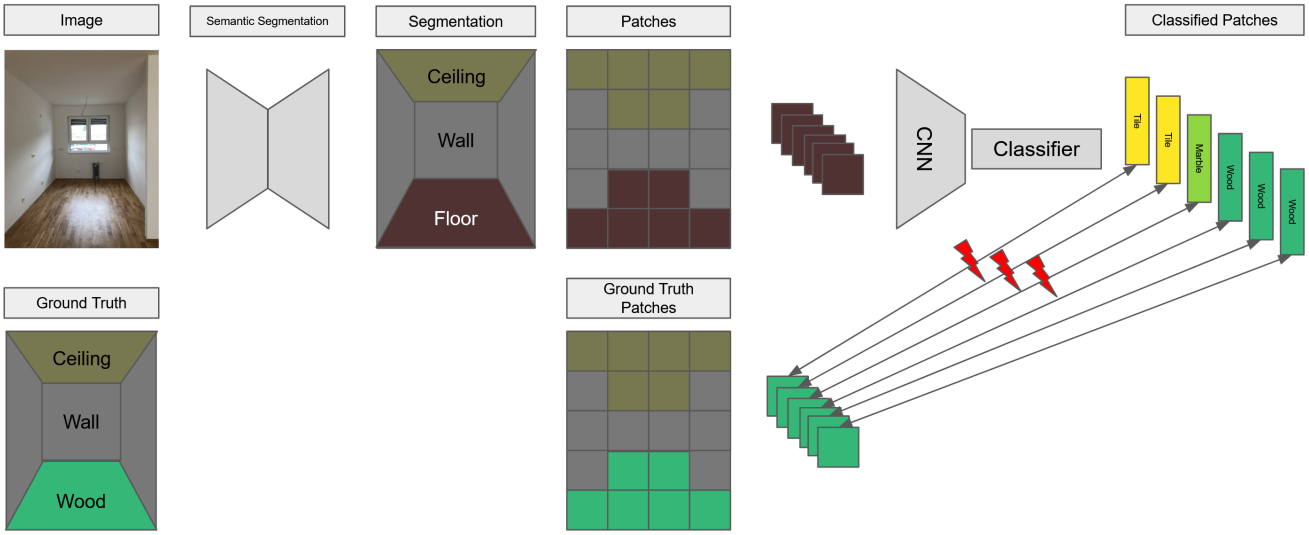

Figure 2: This figure shows the pipeline of our approach. As input we use a construction site image and as output we receive, both, a semantic segmentation of the parts of the room and an approximation of the material distribution of the floor. First, a semantic segmentation is performed. The floor is then divided into patches. At the same position of the floor patches we lookup the real material in the ground truth. After classification, we calculate the accuracy using the classified patches and the ground truth patches.

$$Accuracy = \frac{\sum c_i}{\sum a_i}$$

where c_i is the sum of correct patches per segment and a_i is the sum of all patches in the segment. If the correct material is classified over 50% in the image then the segment is classified correctly for the metric we call segment accuracy. The second accuracy of every patch is compared against the ground truth. This gives us a relative accuracy of all the patches in the segment. We call that the patch accuracy. Our patch classifier performance is shown in Section 4.4.

4. Evaluation

In this chapter, the datasets used are first presented, before the training is described in the following section. Finally, the experiments used to examine the performance of the individual sections of the pipeline are presented.

4.1. Datasets

Two datasets were used for the training and evaluation of the classifiers. These are the MINC-2500 [2] and the Opensurfaces [1] dataset.

Opensurfaces consists of over 105,000 segments. Each segment was cut out of a photograph and shows a specific material. In total, 36 different materials are distinguished. In this work, however, only a fraction of these materials are of interest (*concrete*, *granite/marble*, *tile* and *wood*). Therefore, for this work, only a subset with a size of 12 GB of the dataset was downloaded and processed. Since this data is used for training and evaluation of the patch classifier, the segments must first be cut into square patches. For this purpose, an equidistant grid is placed over the segment and all patches that are completely filled with texture are cut out. Due to the shape of the segments and the selected patch size, it is possible that no patches can be found for some segments. The dataset is not balanced with regard to the number of samples per material. This means that after all segments have been divided into patches, the number is not balanced here either. Therefore, the material with the lowest number of patches serves as the upper limit. If more patches exist for another material, the excess patches are discarded. Two different patch sizes were sampled, resulting in 22,120 patches for 128*128px and 4,740 patches for 224*224px.

MINC Dataset. In addition, the MINC-2500 dataset was also used for training and evaluation of the classifier. This combines the photographs from Opensurfaces with additional photographs from Flickr and Houzz. The additional photographs were annotated using crowdsourcing. In total, the dataset consists of 23 different materials. For each material there are 2,500 batches with a resolution of 362*362px. However, of the available materials, only *polished stone*, *tile* and *wood* are used. The material *concrete* is not available in the dataset.

Figure 3: The figures shows three example images from our own dataset with the respective semantic segmentation (ground truth). The colors correspond to the class or the material of the segment.

Own Dataset. To test the method implemented in this work, a separate dataset was created. It consists of 61 indoor photographs of a renovated apartment. Most of the images were taken in the room with the optical axis of the camera parallel to the ground and the main room axes aligned with the image axes. For each image, a ground truth was created manually using a tool[1]. In total, the segments were assigned to seven different categories (*ceiling*, *wall*, *tile*, *granite/marble*, *wood*, *other*), where *tile*, *granite/marble* and *wood* are interpreted as *floor*. Three example images of the annotation can be found in Figure 3

4.2. Training

The process consists of a pipeline in which the image is first segmented with the aid of a neural network. The areas identified as floor are then cut up and the resulting patches are assigned to a material with the aid of the classifier.

For the semantic segmentation no own training is performed. Instead, a network that has already been pre-trained on ADE20k [27] dataset is used.

For the classification of the patches, different CNN's, with different configurations were trained on the Opensurfaces and the MINC-2500 dataset. One CNN used was the Wide RestNet (50_2) [30] and the EfficientNet (b1) [25]. Both architectures are widely used and have been successfully applied as feature extractors for classification in many different application areas. For both architectures, models pre-trained on ImageNet [18] were used and fine-tuned on the subset of the Opensurfaces dataset and the MINC-2500 dataset. This already results in four different variants. In addition, we tested the effect of two different optimizers. For this the Adam optimizer (lr=0,0001) and the SGD optimizer with a learning rate scheduler (ReduceLROnPlateau [2]) were used.

In total, eight different configurations were used for training. Other parameters which were chosen to be the same for all configurations are as follows: For training and evaluation the datasets were split up. 80% of the data was used for training and 20% of the data for evaluation. A total of 50 epochs, with a batch size of 32, was performed. The metric used was accuracy. This describes the proportion of correctly estimated patches to the total number of patches in the evaluation subset. Results of the training evaluation are reported in Table 2.

4.3. Semantic Segmentation Performance

In the following, the quantitative and qualitative results of the evaluation for the first processing step are presented. The first processing step is the semantic segmentation. In the input image, ceiling, wall ,and floor can be found. Two different CNN architectures were used for this purpose. These are Deeplab V3+ [5] and HRNetV2 [31]. For both architectures, a model already pre-trained on the ADE20k dataset was used and tested on the self-created dataset. For this step, the differently annotated materials are all combined and interpreted as floor. The metric used is the well-known *Intersection over Union*, first averaged per category over all of the 61 input images and finally the average over all of the categories was calculated. The quantitative results are shown in Table 1. It can be seen that Deeplab V3+ performs better than HRNetV2 on average, especially in the important floor category. Qualitative results can be seen in the form of positive and negative examples in Figure 5.

[1] https://segments.ai/

[2] https://pytorch.org/docs/stable/optim.html#torch.optim.lr_scheduler.ReduceLROnPlateau

Model	Ceiling mIoU	Wall mIoU	Floor mIoU	Avg. mIoU
D	62,65 %	**84,16** %	**90,86** %	**79,21** %
H	**63,82 %**	81,11 %	89,83 %	78,25 %

Table 1: This table shows the results of semantic segmentation tests on the own dataset. The accurateness of the determined segments for *ceiling*, *wall* and *floor* was examined. Compared Deeplab V3+ (*D*) and HRNetV2 (*H*) using *mean Intersection over Union (mIoU)*.

4.4. Patch Classifier Performance

In this section, the results of the patch classifier are presented. The patch classifier tries to determine the material in the segment that the semantic segmentation has identified as the floor. For this purpose, this segment is divided into patches of equal size, which then serve as input for the classifier. The classifier then determines the material of each patch. In this way, a distribution of the patches can be determined over all of the patches and all of the materials of the floor by the means of majority voting. The ground truth is used to determine whether the individual patches and the entire segment have been correctly determined. For each patch that was cut out of the segmented image, the ground truth is additionally checked at the same position to determine the material at this position. For this purpose, the central pixel of the patch is evaluated.

The tests were performed with the self-annotated dataset. The different configurations which are listed in Table 2 were investigated. On the one hand, it was tested if a whole segment could be detected by a majority vote (Segment Accuracy) and, on the other hand, how many patches in total (Patch Accuracy) were detected correctly. The results are listed in Table 2. It can be seen that *No. 1* achieves the highest accuracy for the Segment Accuracy. For the Patch Accuracy, *No. 7* achieves the highest accuracy. Both, *No. 1* and *7* use the Wide RestNet (50_2) as model and Adam as optimizer. Additionally, the results of the EfficientNet (b1) with Adam optimizer and *OS (224) + flipped* should be considered (*No. 15*). Although no best values were achieved with this configuration, the results of Segment and Patch Accuracy are on par at a high level.

5. Conclusion

Monitoring construction sites is a long-standing problem, but with our approach to automate this process using computer vision we show that future research can be built on our work. Our segmentation and classification pipeline has shown that our idea of using patch-based image classification works for the given problem. Some lessons learned are:

- The semantic segmentation performs well on most images of our dataset. If the semantic segmentation fails then it is due to the perspective of the image. Figure 5c and Figure 5d show examples when the segmentation fails completely. Our explanation is that these images miss spatial information and the segmentation model needs this information to perform well as shown in Figure 5a and Figure 5b. Therefore, we suggest taking pictures from construction sites, where at least the floor and wall are visible, ideally also the ceiling.

- Our used dataset is quite small with only 61 images from one renovated flat. The pictures were also not taken in different stages of the construction phase. For more valid results of our approach, a larger and more diverse dataset would be required. Also because of the late construction phase of our flat, we could not validate our performance for the concrete class, because there was no more concrete visible in the images.

- The final performance of classifying the patches on our dataset also did not achieve the accuracy we expected as shown in Table 2.

- Different materials such as concrete or tile have inherently little surface structure and thus provide a homogeneous texture which makes classification for CNN more difficult.

- In Figure 4b and Figure 4c two different materials can be seen, which look similar when comparing the patches. Granite/marble and tile are two classes that tend to overlap making a distinct classification difficult. For instance, in Figure 4a you can see a floor made of marble tiles. To tackle the problem of similar-looking textures we also benchmarked different patch sizes (see Table 2). This improved our performance sometimes but also reduced the number of pictures we could train on (details can be found in Section 4.1). We implemented a horizontal flipping of the patch sizes for 224*224px sized patches to increase the training set and received better results. But a larger training set from the beginning would be beneficial.

- The different perspectives and lighting conditions in the image are important factors for the classifier to perform well. Reflections from the ground (see Figure 4c) can confuse our CNN classifier because both of our training sets did not account for light reflections in the images. A possible solution might be the announced database Ground Terrain in Indoor Scenes[3] (GTIS). Like the GTOS database [28] it will account for multiple perspectives and lighting conditions. A

[3] https://www.ece.rutgers.edu/~kdana/gts/gtis.html

(a) (b) (c)

Figure 4: Figure (a) shows an example image from our own dataset. Figure (b) and (c) show crops with the size of 128*128px from this image, whose locations are marked as red squares in Figure (a). Our classifier had difficulties classifying these patches correctly.

CNN trained on GTIS will then most likely perform better on our dataset.

Many future avenues of work remain. Expanding the test dataset to contain more images of different construction sites and different stages of construction will be necessary. Also utilizing a material database that accounts for different lighting conditions and perspectives will be beneficial. Our initial idea of the pipeline can be adapted easily to a new database and test datasets. Our test dataset, codebase ,and trained models are available online [4].

References

[1] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Opensurfaces: A richly annotated catalog of surface appearance. *ACM Transactions on Graphics*, 32(4):1–17, 2013.

[2] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[3] Ioannis Brilakis, Lucio Soibelman, and Yoshihisa Shinagawa. Material-based construction site image retrieval. *Journal of Computing in Civil Engineering*, 19(4):341–355, 2005.

[4] Craig Buchanan and Leroy Gardner. Metal 3d printing in construction: A review of methods, research, applications, opportunities and challenges. *Engineering Structures*, 180:332–348, 2019.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, 2018.

[6] Dan Ciresan, Alessandro Giusti, Luca Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Proceedings of the Advances in Neural Information Processing Systems*, 2012.

[7] Kristin J Dana, Bram Van Ginneken, Shree K Nayar, and Jan J Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, 1999.

[8] Hui Deng, Hao Hong, Dehuan Luo, Yichuan Deng, and Cheng Su. Automatic indoor construction process monitoring for tiles based on bim and computer vision. *Journal of Construction Engineering and Management*, 146(1), 2020.

[9] Andrey Dimitrov and Mani Golparvar-Fard. Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections. *Advanced Engineering Informatics*, 28(1):37–49, 2014.

[10] Kevin K Han and Mani Golparvar-Fard. Appearance-based material classification for monitoring of operation-level construction progress using 4d bim and site photologs. *Automation in Construction*, 53:44–57, 2015.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[12] Diane Hu, Liefeng Bo, and Xiaofeng Ren. Toward robust material recognition for everyday objects. In *Proceedings of the British Machine Vision Conference*, 2011.

[13] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship

[4] https://github.com/m0ritzl/material-classification-in-construction-sites

detection at scale. *International Journal of Computer Vision*, pages 1–26, 2020.

[14] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998.

[16] Ce Liu, Lavanya Sharan, Edward H Adelson, and Ruth Rosenholtz. Exploring features in a bayesian framework for material recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[17] Xianbiao Qi, Rong Xiao, Chun-Guang Li, Yu Qiao, Jun Guo, and Xiaoou Tang. Pairwise rotation invariant co-occurrence local binary pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2199–2213, 2014.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[19] Gabriel Schwartz and Ko Nishino. Visual material traits: Recognizing per-pixel material context. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013.

[20] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *Proceedings of the International Conference on Learning Representations*, 2014.

[21] Lavanya Sharan, Ruth Rosenholtz, and Edward Adelson. Material perception: What can you see in a brief glance? *Journal of Vision*, 9(8):784–784, 2009.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2015.

[23] Hyojoo Son, Changmin Kim, and Changwan Kim. Automated color model–based concrete detection in construction-site images by using machine learning algorithms. *Journal of Computing in Civil Engineering*, 26(3):421–433, 2012.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[25] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*, 2019.

[26] Manik Varma and Andrew Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1-2):61–81, 2005.

[27] Weihao Xia, Zhanglin Cheng, Yujiu Yang, and Jing-Hao Xue. Cooperative semantic segmentation and image restoration in adverse environmental conditions. *Computing Research Repository*, 2019.

[28] Jia Xue, Hang Zhang, Kristin Dana, and Ko Nishino. Differential angular imaging for material recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[29] J. Xue, H. Zhang, K. Nishino, and K. Dana. Differential viewpoints for ground terrain material recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[30] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, 2016.

[31] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.

[32] Zhenhua Zhu and Ioannis Brilakis. Automated detection of concrete columns from visual data. In *Proceedings of the ASCE International Workshop on Computing in Civil Engineering*, 2009.

Appendix

No.	Model	Dataset	Optim.	Concrete	Granite/ Marble	Tile	Wood	Avg.	Segment Accuracy	Patch Accuracy
1	*R*	*MINC*	*A*	-	56%	58,4%	47,2%	66,37%	**58%**	31,53%
2	*R*	*MINC*	*S*	-	70,4%	73,6%	55,2%	75,23%	52%	29,06%
3	*R*	*OS* (128)	*A*	45,12%	56,42%	54,21%	73,06	57,2%	42%	37,15%
4	*R*	*OS* (128)	*S*	54,88%	50,72%	60,09%	74,59%	60,07%	43,5%	41,88%
5	*R*	*OS* (224)	*A*	39,24%	54%	71,61%	82,28%	61,78%	47,7%	52,25%
6	*R*	*OS* (224)	*S*	46,84%	51,06%	73,73%	**89,87%**	65,364%	50,7%	50,61%
7	*R*	*OS* (224) flipped	*A*	35,02%	48,95%	63,56%	79,33%	56,71%	49,23%	**55,01%**
8	*R*	*OS* (224) flipped	*S*	**62,45%**	49,37%	74,58%	89,45%	68,96%	47,69%	50,72%
9	*E*	*MINC*	*A*	-	78,4%	**76,8%**	71,2%	78,3%	39,1%	-
10	*E*	*MINC*	*S*	-	**80,8%**	72%	68,8%	**79,8%**	32%	-
11	*E*	*OS* (128)	*A*	52,4%	51,3%	50,8%	60,6%	53,7%	52,17%	49,76%
12	*E*	*OS* (128)	*S*	50,9%	51,8%	57,4%	74%	58,5%	50,72%	44,87%
13	*E*	*OS* (224)	*A*	52,74%	52,74%	83,48	85,23%	68,53%	47,69%	47,65%
14	*E*	*OS* (224)	*S*	57,38%	49,79%	72,03%	83,12%	57,38%	47,7%	31,94%
15	*E*	*OS* (224) flipped	*A*	50,63%	46,84%	69,49%	88,61%	63,89%	52,31%	52,83%
16	*E*	*OS* (224) flipped	*S*	37,13%	45,99%	68,22%	80,59%	57,97%	44,61%	49,58%

Table 2: The models used are *R* for Wide RestNet (50_2) and *E* for EfficientNet (b1). The abbreviation *OS* stands for the Opensurfaces subset and *MINC* for MINC-2500. The optimizers Adam and SGD+Scheduler have been abbreviated as *A* and *S*, respectively. The columns of the table are divided into three parts. The left part shows the configuration used while the middle part shows the results on the evaluation subset of the training dataset. The right part shows the results of the configurations on the self-created test dataset.

(a) (b)

(c) (d)

Figure 5: Example images of a construction site on the left hand side. On the right hand a semantic segmentation is shown which was generated at the first step of our pipeline. Figure (a) and (b) show examples where segmentation worked very well. The Figure (c) and (d) show negative examples. The photographs show a floor, but the semantic segmentation has recognized it as a wall. For an interpretation of the color code take a look at Figure 3.