Solid Freeform Fabrication 2021: Proceedings of the 32nd Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference
Reviewed Paper

In-Situ Verification of 3D-Printed Electronics Using Deep Convolutional Neural Networks

Daniel Ahlers
Department of Informatics, Group TAMS, University of Hamburg, Germany
ahlers@informatik.uni-hamburg.de *

Abstract

Printed electronics processes are becoming more stable and evolve into first industrial applications. These industrial applications require proper quality assurance to get a mostly autonomous production process. In this work, we present a new approach to inspect printed electronics and ensure their quality. Our hardware setup extends a fused filament fabrication (FFF) printer with an extruder for direct dispensing of conductive paste, a pick and place unit, and two cameras. The cameras take multiple images during printing. A trained neural network analyzes these pictures to separate the electronic wires from the plastic background. All separated images of a layer are combined to get a full view of the layer. Our algorithms then examine the detected wires to identify printing flaws. The algorithms currently detect connection breaks, shorts, find points that have not been reached, and evaluate the width of the printed wires.

Keywords: printed electronics, visual inspection, neural network, fused filament fabrication

1. Introduction

The global additive manufacturing sector is constantly growing. In 2020, the market revenue was 12.8 billion US-Dollar, and it was growing significantly even during the pandemic [1]. While other processes are already used on an industrial scale, 3D printed electronics are still evolving to be reliable enough for industrial applications. 3D printed electronics is a specialized process to embed electronic circuits and components directly into printed objects. This technique can be used to print out prototypes or novel objects with integrated components like sensors. Although this process will never replace classical printed circuit boards (PCB), there are multiple use cases where 3D printed electronics adds additional freedom to product design. This freedom is also an issue for quality assurance since integrated wires are inaccessible after creating the object. Shorts and connection breaks caused by small imprecisions during printing are not detectable anymore. So, a proper in-situ quality assurance process is needed to use the 3D printed electronics process in industrial applications.

This paper presents a new approach to verify 3D printed electronics during printing. Our FFF electronics printer takes multiple images during the printing process. A deep convolutional neural network then separates the conductive wires from the plastic substrate and the background. The

*This work was supported by the Federal Ministry for Economic Affairs and Energy in the ZIM-KamEl project.

algorithms then take the printed G-code and analyze the wires found by the network. The algorithms currently find connection breaks, shorts, points that have not been reached by the printed wire, and evaluate each wire's width.

The rest of the paper is structured as follows. First, Sec. 2 summarizes the related work. Sec. 3 shortly describes the used hardware. Sec. 4 explains how the conductive wires are separated from the plastic substrate using a deep neural network. Sec. 5 introduces the algorithms used to find the different issues in the printed wires. Sec. 6 shows some results and compares the outcome with previous works. The paper concludes with a summary and an outlook in Sec. 7.

2. Related Work

There are multiple techniques to integrate electronic wires into printed objects. With direct writing, a conductive silver paste is applied directly [2, 3]. Another way is directly embedding copper wires into the surface of an object [4]. Conductive wires can also be applied by aerosol jetting [5] or by ink-jetting [6]. The Stereolithography (SLA) process is also capable of printing 3D electronics [7]. Image processing is already broadly used for PCB inspection [8]. Multiple works show the detection of defect lines both on the surface [9] and on inner lines [10]. Faulty drilled holes are identified [9], and the correct placement of components is inspected [11].

In additive manufacturing, visual inspection is often used to improve the plastic/metal processes. Different approaches aim to quantify the geometric accuracy using different types of machine learning [12, 13, 14], inspect individual layers with a neural network [15], or monitor the printing process itself to detect defects [16, 17, 18]. Oleff et al. [19] presented an overview of the different approaches for process monitoring in additive manufacturing using different sensors like cameras, thermal cameras, accelerometers, 3D cameras, microphones, and other sensors. For conductive wires, Salary et al. [20] showed a method to quantify the attributes in aerosol jetting. This includes the line width, line density, line edge quality/smoothness, overspray, line discontinuity, and internal connectivity. They also showed that the estimated resistance could be calculated using shape-from-shading [21].

This work extends our previous work for optical verification of 3D printed electronics [22]. A support vector machine (SVM) was trained on every layer to classify the pixels into conductive or not conductive pixels by only looking at their color value. This classification is a bit noisy and has problems with dark regions near the conductive material and reflections on top of the wires. A sliding window is shifted along the planned extrusion, and all conductive pixels are summed up to find connection breaks. If the number of pixels falls below a threshold, the section is classified as a connection break, nevertheless if there is a broken connection.

3. Hardware Setup

Our new hardware setup to fully automatically print 3D electronics is a modified E3D ToolChanger [23]. The open-source ToolChanger is a platform that can attach four different tools to a motion system. These tools can be parked and picked up again. Fig. 1 shows our printer with the different tools used for conductive printing. All tools only move along the X-, Y-plane while the printbed moves along the Z-axis. A camera is attached to the toolhead facing downwards to move around with the attached tools permanently. Currently, two FFF extruder printheads are attached to the printer. One is directly driven, and the other is a bowden driven extruder. A pneumatic conduc-

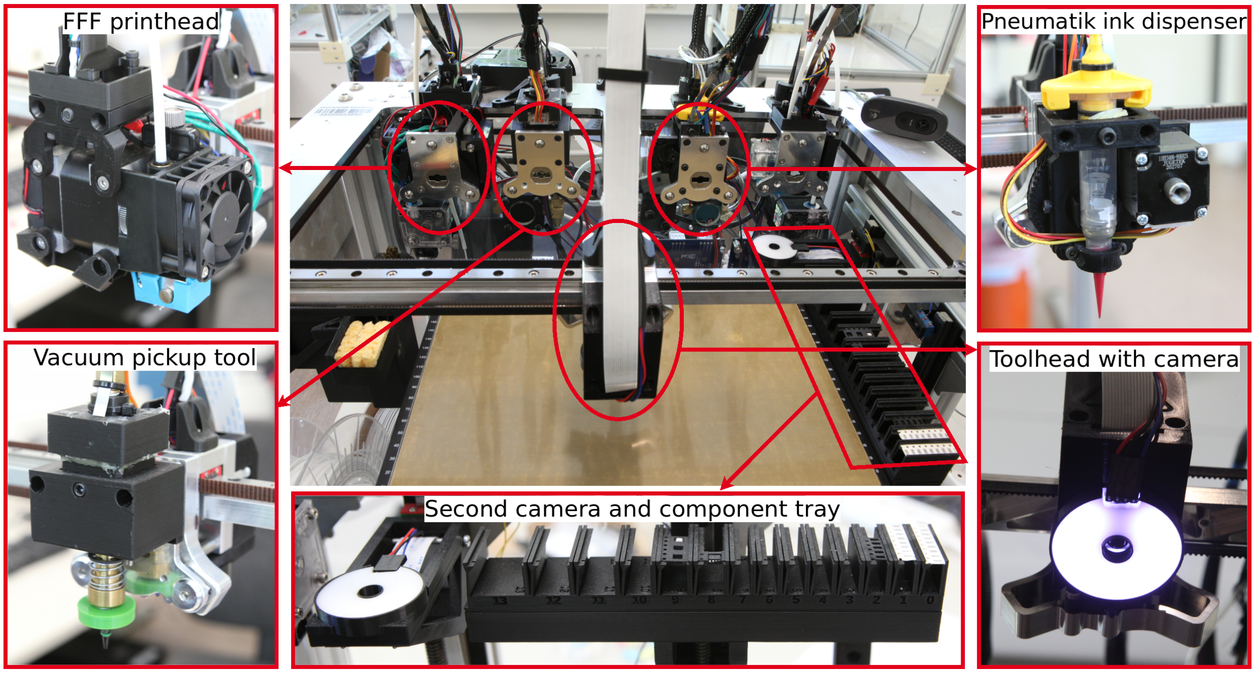

Figure 1 – The modified E3D ToolChanger with the FFF printhead, the vacuum pickup tool, the rotating conductive extruder, the camera attached to the toolhead, and the component tray with the second camera next to it.

tive ink dispenser can extrude conductive silver paste to create electronic wires when pressure is applied. A vacuum pick and place tool is used to grab components from the component tray and place them into the objects. The component tray can be moved up and down to pick up the components. The second camera is used to align components and calibrate the printer's different tools, similar to our previous work [24]. Since all tools have a different length, the camera can be moved up and down to shift the focus to the tool's tip. Both cameras are a Raspberry Pi Camera Module V2 with a resolution of 3280×2464 pixels, a short focus distance, and an attached led ring to have reproducible light conditions. Since these cheap cameras tend to produce distorted images, we calibrated the intrinsic parameters of the cameras and corrected every image before further processing.

4. Wire Segmentation

The images for the verification of the 3D printed electronics are taken during printing each time a layer is finished. The camera in the print head takes multiple images per layer to cover the whole printed object. The pictures are aligned and recorded automatically by our software [22]. For better matching, the tiles are taken with an overlap to align them automatically. This is necessary because of positioning inaccuracies from the belt-driven gantry system. The tiles are combined into a large image where each pixel covers $20\,\mu m \times 20\,\mu m$.

The images are then fed into the neural network to separate the conductive wires from the plastic substrate and the background. Such a large image can not be feed directly into the network, and scaling it down to a usable size would lead to a loss of detail. So the image of the whole layer is

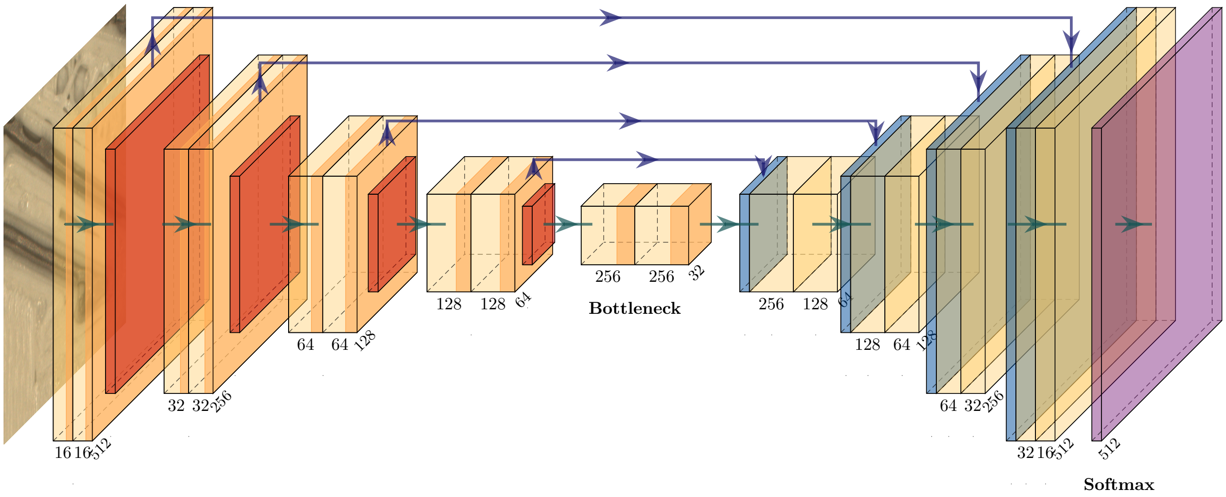

Figure 2 – The structure of the U-net with a 512 × 512 pixel input and output. All layers use ReLu as an activation function except the output layer, which uses Sigmoid.

again split up into multiple tiles where each tile is 512 × 512 pixels. Because convolution layers with active padding add zeros on the edges to keep the input and output size the same, the detection from the network is wrong along the edges of the tiles. This is compensated by adding an overlap of 32 pixels to the tiles. The overlap is removed when rearranging the output tiles back into one image after all tiles are processed.

Different network architectures were implemented and tested with the same dataset to find the best working network architecture. The tree architectures are U-net [25], DenseNet56 [26], and DeepLabv3+ [27], all implemented with TensorFlow. While all architectures performed similarly in detection rates, the U-net architecture was chosen because it performed best on small detail regions like plastic strings and small gaps in the material. The U-net takes an image with the tree color channels as an input. The contraction part contains four blocks, each with two convolution layers and a max-pooling layer. The bottleneck only contains two convolution layers. The layers are then expanded with four blocks to get back to the input resolution. Each block starts with a transposed convolution, followed by the concatenation with the bypass and two convolution layers. All convolution layers use Relu as an activation function. Lastly, a convolution layer outputs three probabilities for each pixel using a Sigmoid activation function. The three classes are conductive wire, plastic substrate, or background. Fig. 2 illustrates the layout of the network as described previously.

The training data for the network is generated from images recorded for documentation purposes in previous prints. The images are extracted from six different prints. They are printed in white and black plastic. The image tagging of the electronic wires, the plastic, and the background was done by hand. To get the images in the same size as the network, they are split up into 512 × 512 pixel sub-images with an overlap of 256 pixels. Since the dataset has only two colors, a copy of the images is added where the color of the plastic was changed with GIMP. Printing more colors with the printer instead of artificially creating images was planned but could not be realized due to a closed lab during the pandemic. More data augmentation is added by rotating a small proportion

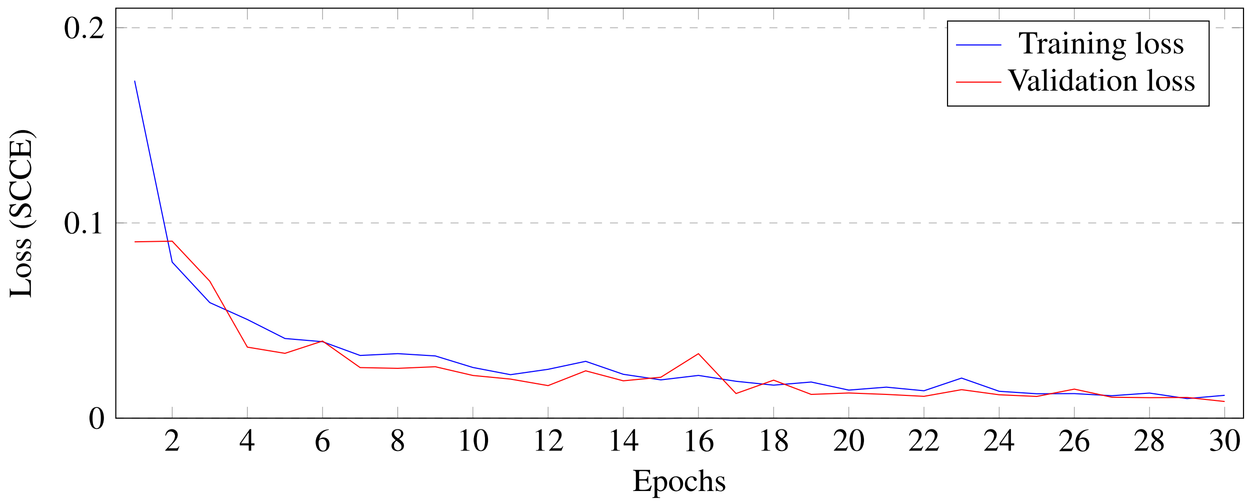

Figure 3 – The plotted loss of the training and validation sets during the epochs.

of the images, changing their brightness and contrast. The resulting dataset has roughly 20,000 images split into 80% training set and 20% validation set. The network was trained for 30 epochs and converged pretty fast. The resulting training and validation loss from the Sparse Categorical Cross Entropy (SCCE) function is plotted in Fig 3.

5. Wire Verification

With the separated image from the network, the verification algorithms can be applied. First, the printed wires are extracted from the G-code file. Then they are converted into the image coordinate system. This is possible because the positions where the corrected pictures are taken and the pixels per mm are known. Fig. 4 shows the output from the neural network overlayed with the G-code instructions. Since the wires in the G-code have no information about the relationship of the different lines, they are grouped before starting the verification. The grouping algorithm tests every line if one endpoint is near any other endpoint of another line. If two points are closer together than a threshold (1/2 wire width), they belong to the same wire and are grouped. If a line has points close to two groups, they are merged because they belong to the same wire. When all lines are grouped, the printed wires can be evaluated. The following paragraphs describe how the different appearing problems can be identified. All tests are repeated for all grouped wires that were extracted from the G-code.

Shorts are unwanted connections between two or more wires often caused by too much conductive material, imprecise movement, or a leaking conductive extruder. A short can be found by checking if any other wire can be reached from the current wire. To test this, the segmented image from the network is taken and the starting point of the wire is flood-filled. Flood-fill recursively marks every neighbor pixel that is classified the same as the starting pixel. Then, every point of all other wires is checked if they got filled. If they got filled, these two wires are connected and have a short.

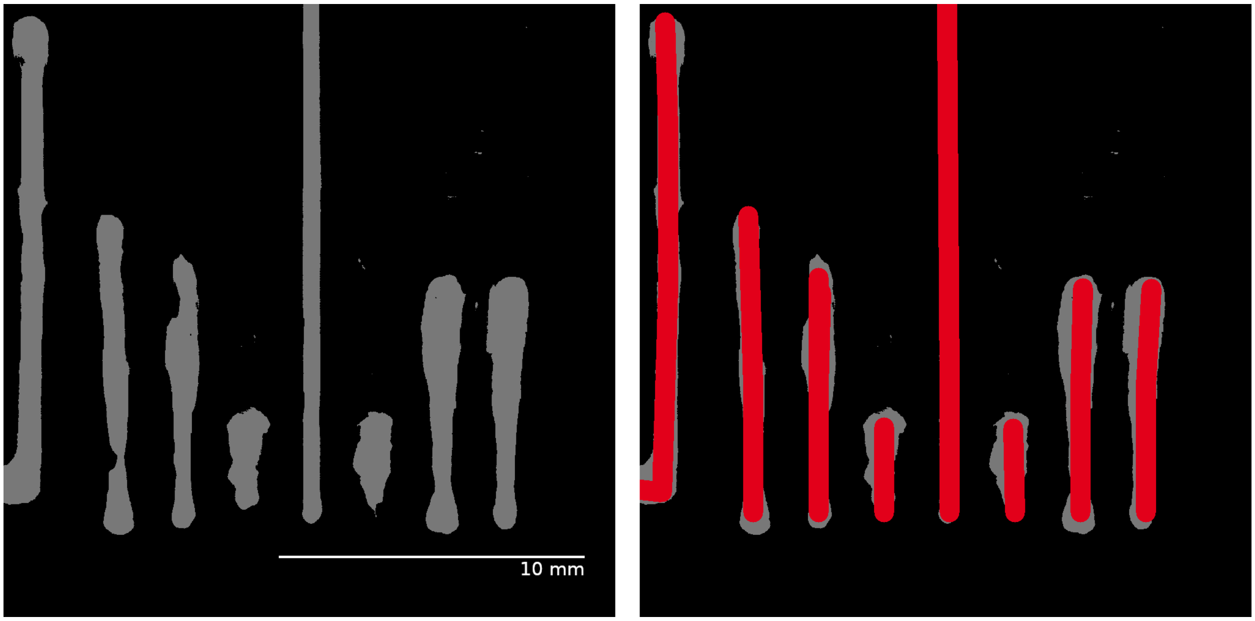

Figure 4 – Left: The output of the wire segmentation where gray areas are identified as wires and black areas are plastic substrate. Right: The overlayed wires extracted from the G-code in red.

Connection breaks are unwanted interruptions in the conductive material. Unconstant extrusions or small plastic strings often cause these issues. Connection breaks are also checked with the flood-fill technique. The starting point of the wire is flood-filled and then every point of this wire that got extracted from the G-code is checked if they got filled correctly. If a point got not filled, there is no connection between the starting point and this point, so there must be a connection break somewhere in between. The part of the wire that is broken can be found by looking for the line extracted from the G-code where one end of this line is filled and the other is not.

Points that are not reached often appear on sharp corners when the material is slightly off due to an imprecise motion system or on the ends of a wire due to inconsistent material flow. They can be found by checking if all points from the G-code of a wire are classified as conductive material. If not, this point is not reached during printing. While a connection break or a short is a clear fault, a point that was not reached can still be tolerable since there is a general connection in the wire, and reaching this exact point might not be important.

Wire width is often crucial since the width of the wire influences its resistance. A wire can be connected but have a thin section along its path. To measure a single wire and not accidentally include another one nearby, all other wires are removed from a copy of the segmented image. Then each wire segment from the G-code is extracted into a single image and rotated vertically. So the image only contains a segment from one corner to another. The width is limited to four times the configured extrusion width of the conductive wire to reduce the impact of the corners. Then all pixels are then summed up in every pixel row to get the width of the wire in this pixel row. The pixel values are then converted into mm to get real-world values. Fig. 5 illustrates the evaluation of one wire segment.

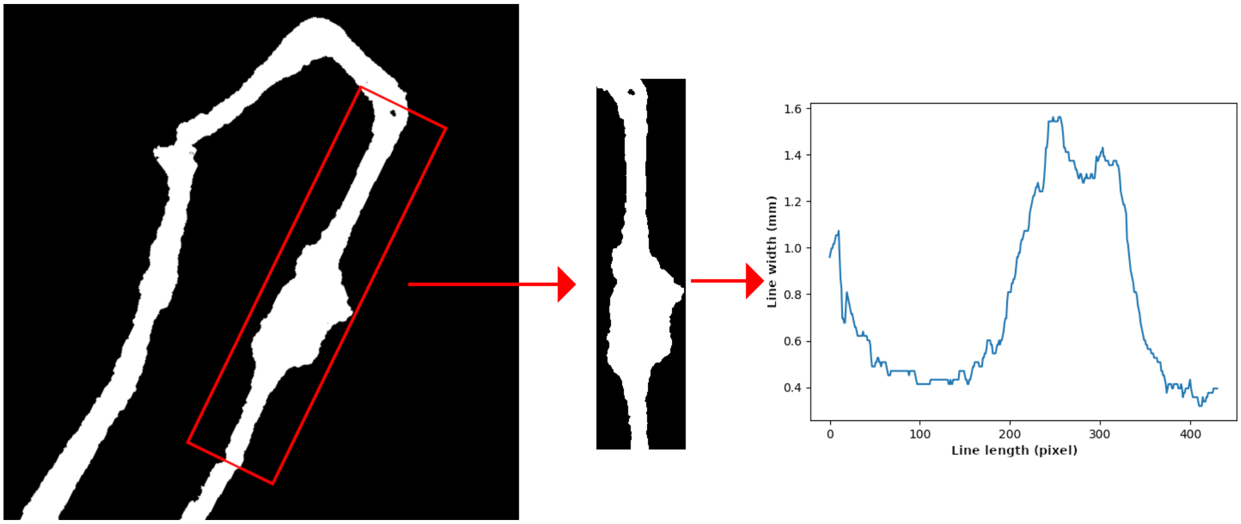

Figure 5 – Left: The isolated wire group. Middle: The cropped and rotated wire segment. Right: A plot of the summed-up pixel rows representing the wire segments width (mm) in printing direction.

6. Evaluation

The wire segmentation is evaluated by comparing the detection performance of the neural network with our previous work [22]. Multiple images are processed by the neural network and the SVM and then compared with a manually labeled version of the images. Table 1 lists the correct classified pixels, the false-negative pixels, and the false-positive pixels for both approaches. Note that the summed-up pixels are only classified as conductive wire by the different approaches and the percentage rate is compared to the actual number of existing conductive pixels in the manually labeled images. False-positive means that pixels are marked as wires but are no wires. False-negative is the other way round where wire pixels are marked as plastic or background. Both cases are a problem since false positive pixels tend to indicate a short when there are none and can lead to missed connection breaks, while false-negative pixels might lead to missed shorts and detect connection breaks that are nonexistent. The results show that the rate of correctly classified and false-negative pixels stays similar to the SVM approach while the false-positive rate drops significantly. When inspecting which regions are wrongly classified, it shows that dark regions are generally less often wrongly classified. However, when the dark region lies between two conductive wires, the classification sometimes fails and marks the region between these two wires also as wire. Reflections are less problematic, but very bright ones are still sometimes wrongly classified.

Table 1 – Pixel classification of conductive wires in absolute pixels and the detection rate.

	correct	false-positive	false-negative
U-net	916304 (96.6%)	60008 (6.3%)	32407 (3.4%)
SVM	904626 (95.4%)	279645 (29.5%)	44085 (4.6%)

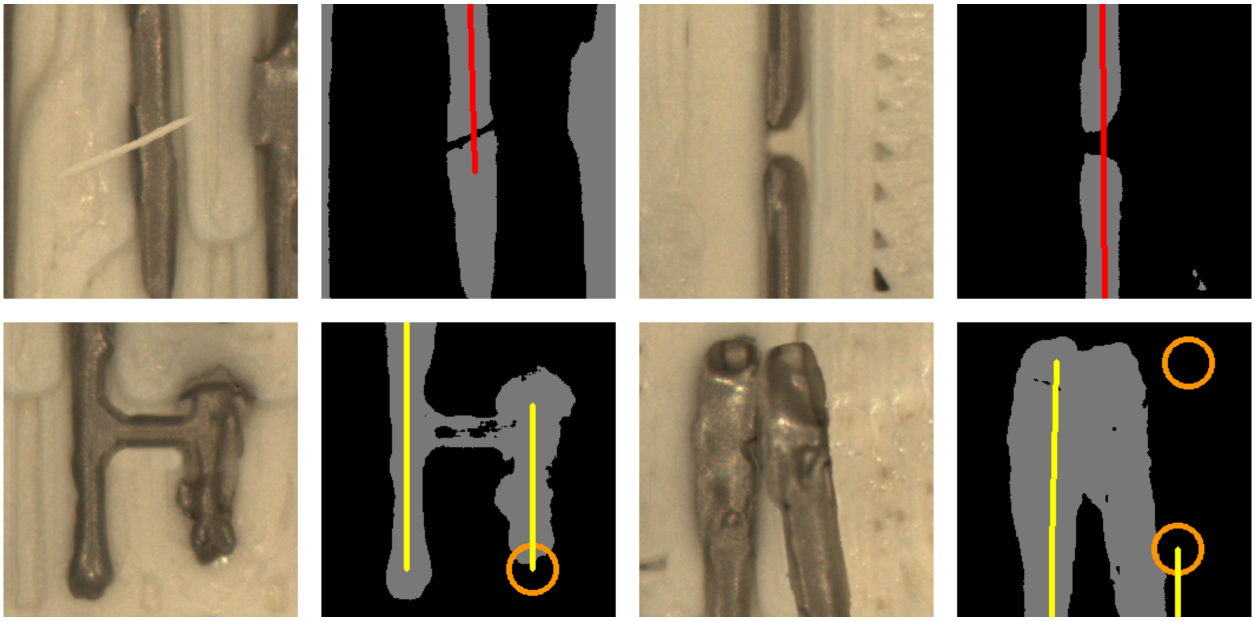

Figure 6 – Four different defects and the output from the verification algorithms. From top left to bottom right: A connection break created by a plastic string over the conductive material, a connection break from missing material, a short from an accidentally printed line, and a short from a misplaced conductive line. In the results, connection breaks are marked red, shorted wires are drawn yellow, and points that are not reached by the conductive material are marked orange.

To evaluate the verification algorithms, we created images with intentional defects that these algorithms should find. The defects are connection breaks due to inconsistent extrusions, connection breaks produced by thin plastic strings from leaking plastic extruders, and different shorts. Since we had no pictures of shorted connections, we created shorts using GIMP to test our algorithms. Fig. 6 shows different defects with the output of the wire verification algorithms. It can be seen that our algorithms detect connection breaks correctly and marks the broken wire segments in red. The shorts from the accidentally printed line and the misplaced conductive line are also identified and marked in yellow to indicate that these two wires are shorted. The algorithms also found three points that are not reached and marked them with orange circles. While the point on the first short is not reached because the material is not printed until the end, the other two are not reached because the conductive material is not in its intended position.
The extra time that it takes to run the whole verification process is crucial. The wire segmentation with the neural network takes ∼6 seconds, while the wire verification takes ∼1 second for one layer. Both measured on a notebook with a 4 core 3.6 GHz CPU. So the influence on the print time per layer is less than 1% which is similar to our previous work and does not increase the printing time significantly. However, capturing the images during the printing still takes about 10% extra time on every layer, increasing the printing time significantly.

7. Conclusion and Outlook

In this work, we showed our new approach to verify 3D printed electronic wires while printing it. The approach focuses on FFF printed plastic objects with pressure extruded conductive silver ink but could also be applied to different printing techniques, substrates, and wire embedding techniques. It compares the G-code instructions with the actual printed object without any needed supervision and stores all data for a high-resolution documentation. A professional can look at the found errors and decide if the device can be used or not. Knowing the wire width of each wire can be used to improve the printing process.

Our future work will focus on improving the wire verification to identify the exact position of a short or connection break. A found connection break could be printed again to get a stable connection during printing and close the control loop. Since shorts and connection breaks can also appear between layers, we want to connect these individual layers to find defects that appear between two adjacent layers. The correct placement of SMD components also needs further inspection. Here, the printed footprints and the placement of the components need to be verified. The wire thickness measurement can also be fed back into the printer to adjust the material flow during printing.

References

[1] Terry Wohlers, Robert Ian Campbell, Olaf Diegel, Ray Huff, and Joseph Kowen. *Wohlers Report 2021*. Wohlers Associates, Fort Collins, Colorado, 2021.

[2] Cassie Gutierrez, Rudy Salas, Gustavo Hernandez, Dan Muse, Richard Olivas, Eric MacDonald, Michale D Irwin, Ryan Wicker, K Churck, M Newton, and Brian Zufelt. CubeSat Fabrication through Additive Manufacturing and Micro-Dispensing. *Proceedings from the IMAPS Symposium*, 2011.

[3] Florens Wasserfall, Daniel Ahlers, Norman Hendrich, and Jianwei Zhang. 3D-Printable Electronics - Integration of SMD Placement and Wiring into the Slicing Process for FDM Fabrication. In *Proceedings of the 27th International Solid Freeform Fabrication Symposium*, pages 1826–1837, Austin, 2016.

[4] C.Y Kim, a. Cuaron, M.a. Perez, D. Espalin, E. MacDonald, and R.B. Wicker. Cooperative Fabrication Methodology for Embedding Wireon Curved Surfaces. *Proceedings of the International Solid Freeform Fabrication Symposium*, pages 185–196, 2014.

[5] Martin Hedges and Aaron Borras Marin. 3D Aerosol Jet Printing - Adding Electronics Functionality to RP/RM. In *DDMC 2012 conference*, pages 1–5, 2012.

[6] J. Ledesma-Fernandez, C Tuck, and R Hague. High Viscosity Jetting of Conductive and Dielectric Pastes for Printed Electronics. *Proceedings of the International Solid Freeform Fabrication Symposium*, pages 40–55, 2015.

[7] Amit Joe Lopes, Eric MacDonald, and Ryan B. Wicker. Integrating stereolithography and direct print technologies for 3D structural electronics fabrication. *Rapid Prototyping Journal*, 18:129–143, 2012. ISSN 1355-2546. doi: 10.1108/13552541211212113.

[8] M. Moganti, Ercal F., Dagli C.H., and Tsunekawa S. Automatic PCB inspection algorithms: a survey. *Computer Vision and Image Understanding*, 63(2):287–313, 1996.

[9] C. K. Huang, C. W. Liao, A. P. Huang, and Y. S. Tarng. An automatic optical inspection of drill point defects for micro-drilling. *The International Journal of Advanced Manufacturing Technology*, 37(11):1133–1145, Jul 2008.

[10] Jie-xian Huang, Dong-tao Yang, and Cang-lai Gong. Inspection of PCB line defects based on directionality measurements. *Circuit World*, 38(3):130–141, 2012.

[11] Pedro MA Vitoriano and Tito G Amaral. Improved Pattern Matching Applied to Surface Mounting Devices Components Localization on Automated Optical Inspection. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 11(4):429–433, 2017.

[12] L. Lu, J. Zheng, and S. Mishra. A Layer-To-Layer Model and Feedback Control of Ink-Jet 3-D Printing. *IEEE/ASME Trans. Mechatronics*, 20:1056–1068, 2015.

[13] M. Khanzadeh, P. Rao, R. Jafari-Marandi, B.K. Smith, M.A. Tschopp, and L. Bian. Quantifying Geometric Accuracy With Unsupervised Machine Learning: Using Self-Organizing Map on Fused Filament Fabrication Additive Manufacturing Parts. *J. Manuf. Sci. Eng.*, 140: 031011, 2017.

[14] Samie Tootooni, Dsouza A. M., R. Donovan, P.K. Rao, Z. Kong, and P. Borgesen. Classifying the Dimensional Variation in Additive Manufactured Parts From Laser Scanned Three-Dimensional Point Cloud Data Using Machine Learning Approaches. *J. Manuf. Sci. Eng.*, 139:091005, 2017.

[15] Zeqing Jin, Zhizhou Zhang, and Grace X Gu. Autonomous in-situ correction of fused deposition modeling printers using computer vision and deep learning. *Manufacturing Letters*, 22:11–15, 2019.

[16] Jeremy Straub. Initial Work on the Characterization of Additive Manufacturing (3D Printing) Using Software Image Analysis. *Machines*, 3(2):55–71, 2015.

[17] U. Delli and S. Chang. Automated Process Monitoring in 3D Printing Using Supervised Machine Learning. *Procedia Manufacturing*, 26:865–870, 2018.

[18] Yuanbin Wang, Jiakang Huang, Yuan Wang, Sihang Feng, Tao Peng, Huayong Yang, and Jun Zou. A CNN-Based Adaptive Surface Monitoring System for Fused Deposition Modeling. *IEEE/ASME Transactions on Mechatronics*, 25(5):2287–2296, 2020.

[19] Alexander Oleff, Benjamin Küster, Malte Stonis, and Ludger Overmeyer. Process monitoring for material extrusion additive manufacturing: a state-of-the-art review. *Progress in Additive Manufacturing*, pages 1–26, 2021.

[20] Roozbeh Ross Salary, Jack P Lombardi, M Samie Tootooni, Ryan Donovan, Prahalad K Rao, Peter Borgesen, and Mark D Poliks. Computational Fluid Dynamics Modeling and Online Monitoring of Aerosol Jet Printing Process. *Journal of Manufacturing Science and Engineering*, 139(2):021015, 2017.

[21] Roozbeh Ross Salary, Jack P Lombardi, Prahalad K Rao, and Mark D Poliks. Online monitoring of functional electrical properties in aerosol jet printing additive manufacturing process using shape-from-shading image analysis. *Journal of Manufacturing Science and Engineering*, 139(10):101010, 2017.

[22] Florens Wasserfall, Daniel Ahlers, and Norman Hendrich. Optical In-Situ Verification of 3D-Printed Electronic Circuits. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 1302–1307. IEEE, 2019.

[23] E3D ToolChanger. URL `https://www.e3d-online.com/pages/toolchanger`. [Accessed 31.05.2021].

[24] Daniel Ahlers, Florens Wasserfall, Norman Hendrich, Arne Bungener, Jan-Tarek Butt, and Jianwei Zhang. Automated In-Situ Placing of Metal Components Into 3D Printed FFF Objects. *IEEE/ASME Transactions on Mechatronics*, 2021.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[26] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The One Hundred Layers Tiramisu: Fully Convolutional Densenets for Semantic Segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 11–19, 2017.

[27] 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 (ECCV)*, pages 801–818, 2018.