Machine Learning in Applications
AM04 Project Report

Stiven Hidri (s315147), Filippo Greco (s309529), Giovanni Gaddi (s308685), Gabriele Ferro (s308552)
Politecnico di Torino

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Machine Learning in Applications AM04 Project Report

***Abstract*—In recent years additive manufacturing has been getting a lot of recognition: creating objects from the ground up, among all the other benefits, allows to waste less material compared to subtractive manufacturing methods. Although this strategy is highly effective, it is not free of challenges. Different defects can emerge during the printing phase which might lead to unusable manufactured parts. An automatic solution for the defect detection task could be helpful but, as in many other fields, data scarcity is a constraint since few images of the manufacturing process are available. To overcome this problem we managed to isolate the available defects and train an image harmonization model to harmonize newly created defected synthetic images. Such synthetic images consist in imperfections cropped and pasted onto imperfections-free backgrounds. Classification models are then trained and tested on the new artificial dataset. Although promising results are obtained, to further study strengths and weaknesses of the proposed technique an Object Detection Model is trained on the augmented dataset and then further tested on the original provided one. Despite the model did not achieve optimal scores on the original samples, valuable insights about the difficulties of the recognition of different defect's type were gained from the results, laying the base for future works in the topic.**
Source code can be found at https://github.com/figimodi/AM04

I. Introduction

Additive manufacturing is rapidly gaining traction across various industries due to the numerous advantages it offers over traditional subtractive fabrication methods. One of the most significant benefits is the ability to build objects by adding material layer by layer, as opposed to removing excess material from a larger block. This fundamental difference opens the door to the creation of more complex and mechanically efficient objects, considering material amount over structural strength, enabling as well the use of innovative and advanced materials that were previously difficult or impossible to work with. Additionally, additive manufacturing is inherently more efficient, as it significantly reduces material waste, ensuring a more sustainable production process. Furthermore, the capability to produce multiple parts simultaneously on a single machine not only optimizes the manufacturing process, but also drastically cuts down on production time, leading to faster turnaround and greater scalability in production.

The solution is valuable, but it's important to recognize the challenges it entails. During the printing phase, various types of defects can emerge, which may result in the manufactured part being unusable.
Detecting and classifying these defects could help factories trace out the root causes of the problem, develop tailored solutions and halt the additive process to prevent unnecessary expenditure of materials and time.

To accurately classify defects, the classification model must be trained on a large dataset, however, generating the defected samples is both costly and improbable under normal machine operation.

To overcome this challenge, it is essential to develop an effective technique for creating a sufficiently large dataset of defected samples without intentionally causing machine failures or waiting an excessive amount of time for the necessary failures to occur naturally. Creating a new dataset from a small number of defected samples can be effectively achieved using **data augmentation** techniques. One approach involves pasting isolated defects from defected images on non-defected ones (with clean powder bed) in randomly chosen locations.
This method often results in augmented images that appear unrealistic because the defects may not seamlessly blend with the original illumination and textures of the non-defected images. To solve this issue, **harmonization networks** are employed. These networks adjust the defects to match the surrounding background and lighting conditions, making the augmented images appear as realistic as possible.

In this research, methods for augmenting datasets with defected images will be explored. Initially, defects will be isolated from the original images, various color transformations are then applied to these defects, and then they are used to train an harmonization network. Subsequently, we will generate realistic images by harmonizing synthetic ones that contain defect collages onto non-defected backgrounds.
Then, a defect-classification network is trained on the augmented dataset, consisting in images with a single defect type, to accurately discriminate different defects, if present. Given that, in a real case scenario, more than one defect's type could occur at the same time an additional object detection model is trained, that detects multiple defects and their location through bounding boxes. This model is also tested on the real, starting not augmented dataset of images, to further analyse the adoption of object detection on a real case scenario.

II. Background

Metal Additive Manufacturing. *Sun et al.* [3] explores laser-based powder bed fusion, detailing how the process involves scanning and fusing powder layers under inert gas to prevent oxidation. It provides valuable insights relevant to the domain of our task.
Image Harmonization. *Guo et al.* [4] introduces a new method for image harmonization using a Transformer model

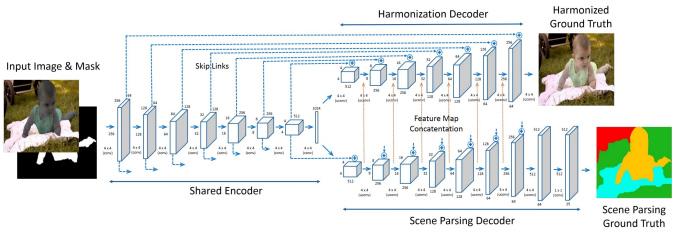


Fig. 1: Representation of the full deep harmonization network from Tsai et al. [1]

that captures global context better, leading to improved results in image harmonization, inpainting, and enhancement.
Hinz et al. [5] explores techniques for training Generative Adversarial Networks (GANs) on a single image to generate realistic variations of that image, that can be interesting in our case study for augmenting the dataset, but, while it does touch on tasks like image harmonization, the main emphasis is on improving the efficiency and quality of image generation from a single sample.

Cong et al. [6] introduces a new dataset called iHarmony4 and a method named DoveNet. DoveNet uses a domain verification discriminator to ensure the foreground and background of an image are in the same domain, enhancing the realism of harmonized images.
However, our chosen harmonization technique relies on the work conducted by *Tsai et al.* [1]. Their network architecture consists of an encoder-decoder framework, designed to harmonize the appearance of image details with their background. The encoder learns to capture the distribution and features of the input images, compressing them into a latent representation. The decoder then decompresses this representation to generate the harmonized image.
To preserve as much detail and texture as possible from the original image, the network includes skip connections that link convolutional layers in the encoder with their corresponding transposed convolutional layers in the decoder. These connections help retain fine-grained information and ensure that the details in the harmonized image closely resemble those in the input.
Fig. 1 illustrates the full architecture of the deep harmonization network. Their implementation also includes an additional decoder designed for semantic segmentation. However, since this component is not pertinent to our study, the focus will be put only on the harmonization aspects omitting the details related to the semantic segmentation decoder.

Image Classification. Object category classification and detection has always been a challenge. Just mentioning the the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [7], that has been run annually from 2010, underscores the significance of these challenges. Already in 1998 *Lecun et al.* [8] proposed LeNet as a CNN to recognize handwritten characters, representing one of the first tries on image classification. With the rapid advancements in neural network architectures, significant effort has been devoted to developing increasingly effective models. *Krizhevsky et al.* [9] utilize AlexNet which was a groundbreaking model in image classification, followed by the work of *Simonyan et al.* [10] on VGG, that introduced a deeper and more uniform architecture. *Szegedy et al.* [11] implemented GoogLeNet and further advanced the field with its innovative inception modules. *He et al.* [12] with ResNet then addressed the challenges of training very deep networks through its residual learning framework. Each of these models has built upon its predecessors, leading to continuous improvements in performance and capabilities.

Object Detection. Object detection has seen significant advancements over the years. The introduction of the Region Proposal Network (RPN) marked a pivotal moment, as it allowed for nearly cost-free region proposals by sharing convolutional features with the detection network. Moreover, the implementation of anchor boxes in RPN, used to handle multiple scales and aspect ratios, further improves the model. In particular, they serve as reference points for detecting objects of various sizes and shapes within an image. This innovation was followed by the unification of RPN and Fast R-CNN into a single network: Faster R-CNN implemented by *Ren et al.* [13]. R-CNN, introduced by *Girshick et al.* [14], was one of the pioneering models that combined region proposals with CNN features for object detection. Fast R-CNN, also developed by *Girshick* [15], improved upon R-CNN by sharing convolutional features across regions of interest, significantly speeding up the process. Faster R-CNN proved to be a cost-efficient solution for practical applications, combining high speed with enhanced accuracy.

III. Materials and Methods

A. Defect Isolation (Defect Masks Creation)

The approach used to augment the dataset involves creating new synthetic images by incorporating defects extrapolated from existing defected images and embedding them onto non-defected ones. To facilitate this process, individual defects are firstly segmented producing the relative masks, which can be used to extract and paste the defects onto the non-defected images as described above.
5 types of different defects have been recognized [2]:

- *Holes*: These are small dark spots in the powder bed caused by missing metallic powder. They usually result from improper regulation of the powder dosing mechanism, leading to localized powder deficiencies.
- *Spattering*: This refers to the droplets of molten metal that are ejected from the melt pool and spread around it.
- *Incandescence*: This describes high-intensity bright areas in the powder bed layer, typically caused by excessive laser energy that prevents the metal from cooling properly.
- *Horizontal Defects*: These are dark horizontal streaks in the layer image, resulting from geometric imperfections

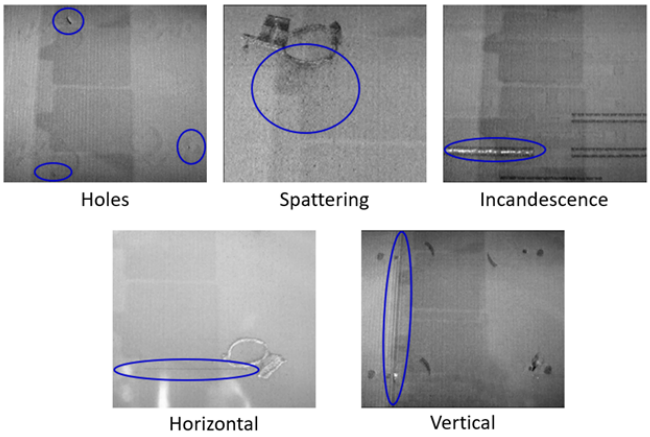


Fig. 2: Examples of each kind of defect in the dataset [2].

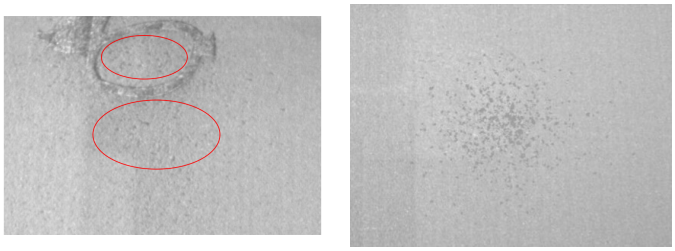
in the piece that disrupt the even spreading of the metallic powder.

- *Vertical Defects*: These consist of alternating dark and light vertical lines in the powder bed along the recoater's path. They are usually caused by either defects in the recoater's surface or mechanical interference between the object and the recoater.

An example of each one of the defects is shown in Figure 2. The starting dataset consists of 47 defected images. Due to the limited amount of data, we have manually segmented and classified the defects one by one using graphic tools. In particular, the area of the defect was separated, and then, based on the gray-value in relation to the background, the defect was further isolated. The resulting area forms the mask relative to the defect.

Spattering has revealed to be tricky to isolate, given the multitude of different little spots of similar colour to the background. The first thought was of selecting the spattering covered region all together, hoping that the harmonization model would be able to successfully harmonize it with the background, but the result showed a very evident halo. To address this problem, a plausible spattering effect was created. Firstly, the correct range of colours of the spattering spots is identified, to recreate trusty points. Next, to emulate the shape and distances of the original defect, the responsible algorithm chooses a center on a free-defect image, generates some random points following a Gaussian distribution (to emulate the denser center and sparser edges) and then slightly proliferates the points in order to make them similar to the original case. In this way we are able to build a mask relative to the little spattering parts, something that was hard to achieve by using only graphical tools given the countless little shapes to be segmented. It must be said that this technique may introduce some bias since it is not guaranteed that the artificial spattering perfectly follows the distribution and characteristics of the original one. Figure 3 shows both an original and a synthetic spattering defect.

Table I shows the number of different isolated defects:



(a) Real spattering. (b) Artificial spattering.

Fig. 3: Comparison between an image containing a real spattering and one generated artificially by the algorithm.

Defect Type	Number of isolated defects*
Horizontal	15
Incandescence	88
Spattering	10
Vertical	2
Hole	65
Total Defects	179

TABLE I: Number of different isolated defects.
* The numbers don't refer to the number of defected images, but rather to the number of individual isolated defects.

B. Color Manipulation

The harmonization model accepts as input an image with the relative foreground mask highlighting the defect. Different versions of the same image must be fed into the network for a better training phase. The different versions consist in the same image where the appearance of the target, in this case the defect, is manipulated with respect to the background. Many colour manipulation techniques are available, but initially only brightness and contrast were tweaked. Since they provided similar results, given that the images are gray scaled, the brightness manipulation was discarded to reduce the number of duplicated samples. Specifically for each image 8 different versions are produced, consisting in the slight alteration of the contrast as showed in Fig. 4. The following scaling factors are applied: 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2.

C. Train Deep Harmonization Network

The training approach described by *Tsai et al.* [1] will be followed. The input to the network comprises two

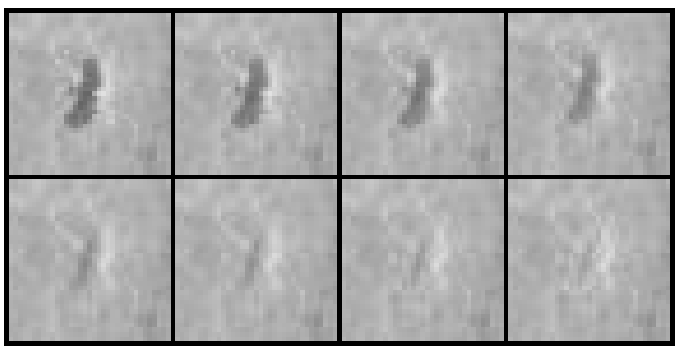


Fig. 4: 8 different scaling factors applied to the contrast for colour manipulation on hole defect.

components: the gray scale image and a binary mask. Both inputs are resized to a resolution of 512x512 pixels and consist of a single channel—grayscale. The mask is designed to highlight the foreground areas that require harmonization to blend seamlessly with the background.

The network architecture begins with 8 sequential convolutional layers, each using a 4x4 kernel. This encoder stage generates a latent space representation of size 1024. Following the encoder, the decoder consists of an equivalent number of transposed convolutional layers, implemented as transposed convolution, again using a 4x4 kernel. To ensure that the final output retains fine textures and details, skip connections are employed between each convolutional layer and its corresponding transposed convolutional layer. The skip connections help preserve critical information, enhancing the overall quality and realism of the generated images.

During training, the network is provided with real images that have undergone color transformations, accompanied by their respective binary masks that isolate the transformed areas. The original, untransformed images serve as the ground truth, allowing the network to learn how to adjust the foreground details to achieve a seamless blend with the background. This training process helps the network understand and replicate the nuances required for realistic image harmonization.

At inference, the process is simplified: the network receives the input image with the relative mask and generates an harmonized version where the foreground details are effectively integrated with the background. This results in augmented images that appear natural and visually coherent.

D. Synthetic Images

To augment the dataset the previously explained defect isolation technique is used: thanks to the obtained masks we are able to crop the defects and randomly paste them onto a defect-free background. While pasting the defects eventual overlapping cases are avoided and the boundaries of the powder bed are respected. In this work each new generated image contains only one type of defect, from 1 to 6 times. The background image is randomly chosen among the available ones and the defects too. The augmented dataset is balanced, i.e. 200 synthetic images for each defect type are generated, for a total of 1000, and 200 without defects, as described in Table II.

Defect Type	Number of Synthetic Images
No Defect	200
Horizontal	200
Incandescence	200
Spattering	200
Vertical	200
Hole	200
Total Images	1200

TABLE II: Number of generated synthetic images.

E. Defect Classification

For the image classification task, all the considered classifiers were modified to accept as input a single channel image, instead of a 3-channels RGB one, given that the provided dataset is composed of gray scale images. For the training phase, the augmented dataset is employed, considering only the synthetic images with at most one defect type each. The original images where not considered given that some of them contains more than one type of defect at the time and the classification wouldn't be possible. As classifiers ResNet-18, ResNet-34 and GoogLeNet are being considered. The results for those classifiers are shown in Section IV.

F. Object Detection on Defects

From the necessity to detect multiple kind of defects on the same image, a further defect detection model has been trained, based on the Faster R-CNN model [13], a network able to classify and locate objects by highlighting their presence through the employment of a bounding box.

Object detection through bounding boxes requires annotations regarding the area boundaries of each object, along with the corresponding labels. To generate these annotations we masks are exploited, manually created for the defect classification task, to calculate the top-left and bottom-right coordinates of each defect. To address this lack of information, during the generation of the synthetic images a pickle file is also created containing relative annotations about bounding boxes coordinates and labels.

Moreover, to be able to use the original dataset as the test set, a script performs the labeling operation for the original samples. The information about the bounding boxes and the label are saved in a pickle file, used for testing the performance of the Faster R-CNN model. The training dataset is composed of synthetic images with multiple defect types, in order to emulate the provided original dataset. Furthermore some particular cases are generated to mimic the original patterns: vertical defects are usually present near the left and right edges of the powder bed and incandescences, some times, are very densely present (up to 20 occurrences in a single image). As usual, harmonization is applied afterwards. Faster R-CNN consists of three main components: a CNN backbone network, that extracts feature maps from the input image, integrated with a Feature Pyramid Network (FPN), that enables the backbone to produce a multi-scale feature pyramid to enhance detection of objects at different sizes; a Region Proposal Network (RPN), that uses these feature maps to generate region proposals where objects are likely to be found; and a detection head, that performs object classification and bounding box regression on these proposals.

The RPN slides over the feature maps, generating anchor boxes of various sizes and aspect ratios. These anchor boxes, defined by all the combinations of the parameters in Table III, are placed at each spatial location on the feature maps, to accommodate objects of different scales and shapes. The

RPN uses these predefined anchors to predict objectness scores and refine bounding box coordinates, adjusting the anchors' positions and sizes to better fit the detected objects. The Region of Interest (RoI) Align step then extracts a fixed-size feature map for each proposed region, which is fed into the detection head. The detection head outputs the final class labels and refined bounding boxes.
A Non-Maximum Suppression (NMS) step is applied to remove redundant detections and produce the final set of bounding boxes and class predictions.

The previously trained ResNet-18 network is employed as the backbone for the Faster R-CNN, deprived of the last fully connected layers. The pre-trained classifier of the augmented dataset has learned discriminative features of the different defects, which may help the object detection task.

Parameter Name	Values
Feature Maps Size	8, 16, 32, 64, 128
Aspect Ratio	0.03, 1.0, 5, 10

TABLE III: RPN parameters.

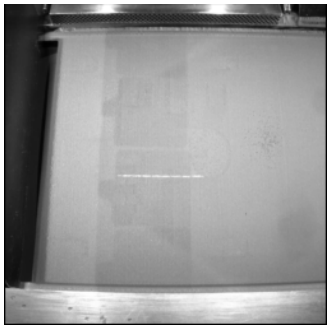
IV. Results and discussion

A. Harmonization model

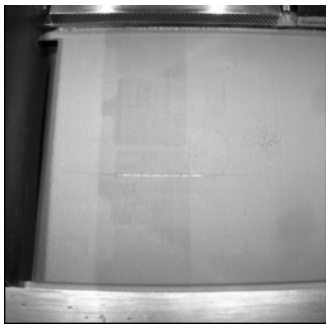
In order to train the harmonization model, the dataset of harmonized synthetic images is split in 50% of samples for the training phase, 25% for the validation set and the remaining 25% as test set. Input images are shrunk to 512x512 pixels to reduce the overall computational demand. The training of the deep harmonization network consists in 200 epochs, a learning rate of 1e-4 and the Adam optimizer with weight decay and plateau scheduler, which reduces the loss by a factor of 1/10 each time it doesn't improve after 5 epochs (patience). The loss employed is the MSE loss between the whole original image and the whole harmonized image. The model's weight are initialized with Xavier uniform initialization. Figure 5 shows the effectiveness of the process, from the generation of the synthetic images to their harmonization through the deep harmonization model. It can be seen, in Figure 6, that the artificial image looks realistic and it can so be used as a sample for the classification task, resolving the problem of data scarcity. Now that as many images as needed can be generated, they can be used as samples for the defect classification task.

B. Classifier

For the classification task, three models are tested: ResNet-18, ResNet-34 and GoogLeNet. For all the models the augmented dataset, obtained through the previous mentioned steps, is split in 50% of the samples for the training phase, 25% for the validation and 25% for the final test part. All three models are trained for 200 epochs employing a batch size of 16. The learning rate starts at $1e^{-5}$ and uses the Adam optimizer with plateau scheduler, with 5 epoch patience and 1/10 as updating factor. Cross entropy is employed as the loss. All the models

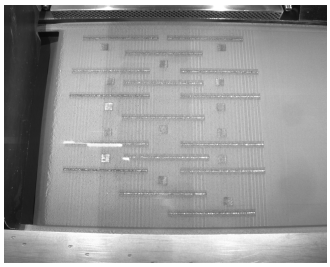


(a) Composed image, without color correction nor harmonization

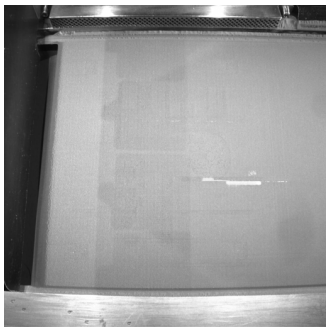


(b) Harmonized composed image

Fig. 5: Comparison between the image as it is composed and its harmonized version. They both show incandescence as defect.



(a) Real image



(b) Synthetic harmonized image

Fig. 6: Comparison between a real image and a synthetic one generated artificially and harmonized with deep harmonization network. They both show incandescence as defect.

are trained to recognize the defect's type if one is present in the image, otherwise it will be labeled as non-defected (6 total labels).

As is it shown in Table IV the best performing model obtained is GoogLeNet, evaluated considering the top-1 accuracy metric. ResNet models, although are known to be better most of the cases for classification tasks could be potentially too big for our dataset, which seems to prefer a simpler model such as GoogLeNet.

Model	Top-1 Accuracy
ResNet-18	88.33
ResNet-34	87.33
GoogLeNet	92.33

TABLE IV: Top-1 accuracies on defect classification for different classification models.

C. Defect detection

The synthetic harmonized images, with multiple defect types, are split in the usual 50-25-25 ratios for training, validation and testing. The Faster R-CNN model is trained with 20 epoch, batch size of 4, learning rate 1e-5 with Adam optimizer and plateau scheduler. 4 losses are jointly considered: RPN

IoU Threshold	AP				
	Hole	Horizontal	Incand.	Spattering	Vertical
0.5	0.00	49.40	64.69	96.49	98.11
0.6	0.00	37.35	55.15	96.49	98.11
0.7	0.00	26.51	41.18	94.74	98.11
0.8	0.00	7.23	26.47	52.63	94.34
0.9	0.00	0.00	2.94	7.02	73.58

TABLE V: Average Precision for different IoU thresholds and different classes.

IoU Threshold	mAP
0.5	61.74
0.6	57.42
0.7	52.11
0.8	36.13
0.9	16.71

TABLE VI: mAP at different thresholds.

classification, RPN regression, Object classification and Object regression.

The small amount of epochs was necessary to keep the computation feasible for the hardware we were provided with, since object detection tasks are very expensive in terms of resources. Also batch size of 4 is kept to reduce memory usage during training.

1) Synthetic Images: The first step was training and testing the object detection model's performances on the augmented dataset. The final results are reported in Table V and Table VI, indicating Average Precision (AP) and Mean Average Precision (mAP) calculated with the area under the Precision-Recall curve, where the correctness of the prediction is decided upon the Intersection Over Union (IoU), at different thresholds, of the predicted and the target bounding box.

From these results it can be seen that the model is confident to predict correctly vertical and spattering defects. Their areas, being in average the widest, contain much more information with respect to other smaller defects, such as incandescences and horizontal lines, whose area is on average smaller, thus containing less information. Holes are not detected at all. Two are the main causes: the very small area defining them and the fact that the image is shrunk to 512x512 from 1024x1280 before being fed to the model, losing more of the already scarce information. Image resizing is also necessary to reduce computational cost when training the model. Examples of object detection on synthetic images are shown in Figure 8.

2) Real Images: For a deeper analysis, the original samples are then used as test for the object detection model. This is done since the proposed augmentation technique is expected to allow to train models on few samples and obtain good results on real case scenarios. Although the defects present in both original and augmented samples are the same, the obtained results are insufficient (Table VII and Table VIII).

A major problem could be the influence of shapes generated during production (Figure 7), that can cause confusion to the Faster R-CNN model. In Figure 9b (Appendix) it is possible to see an example of mis-detection of spattering, when the shapes aforementioned are overlapping with the defect. In the

IoU Threshold	AP				
	Hole	Horizontal	Incand.	Spattering	Vertical
0.5	0.00	0.00	32.18	32.35	0.00
0.6	0.00	0.00	25.29	20.59	0.00

TABLE VII: AP for different IoU thresholds and different classes. Metrics calculated on the starting (not augmented) dataset.

IoU Threshold	mAP
0.5	12.90
0.6	9.18

TABLE VIII: mAP at different thresholds. Metrics calculated on the starting (not augmented) dataset.

same figure are also shown other examples of object detection on real images, one of them regarding incandescences (Figure 9c, Appendix), that are only detected when the defect is large enough. On the other hand if the defect is too small, such as holes (Figure 9a, Appendix), the model is not able to detect them at all, a condition present also on the augmented dataset. The model fails to predict Horizontal and vertical lines too, which might indicate that the synthetic images containing these defects do not reflect their true characteristics. It must be highlighted that the approach of training the classifier using synthetic spatterings may potentially work: the AP scores is similar to the one of the incandesceses even though original spattering defects were not used in the training phase. Ad hoc techniques must be adopted to improve the small object detection, and a deeper study must be pursued to understand how to better process horizontal and vertical defects.

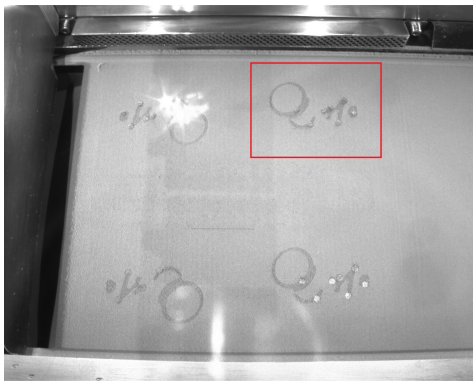


Fig. 7: Shapes printed during production.

V. Conclusions and Future Works

This work addresses the challenge of defect classification and detection in additive manufacturing within a scarce data context, proposing an augmentation pipeline that allows to create a larger datasets starting from a limited one. New synthetic samples are created, by copy-pasting isolated defects onto defect-free powder bed images. To make the synthetic images look more realistic and in tone with the new background, a deep harmonization model is adopted; this is trained using colour manipulated versions of the original samples. Few CNNs are then trained using the newly generated samples to

learn discriminative features among all defect types, reaching really high performances. To further study the benefits and downsides of such proposed method an object detection algorithm is trained to check the classifier's ability to discriminate different defect. Although good precision is reached using the augmented dataset, insufficient performance is obtained while testing the model on the original provided samples, showing room for improvement, specifically regarding small defects detection (holes) and better representation of synthetic horizontal and vertical defects. It is shown that representing the spattering through artificially created spots that mimic the original sample can be a viable solution, which may potentially lead future researches into the creation of artificially created defects without the need of original samples. Moreover, many other techniques may help to speed up and improve the current proposed pipeline: automatic segmentation will replace the manual segmentation task and improve the object detection task by indicating not only the position of the defect, but also indicating its exact extension.

May this work be a solid start for future readers, that can extend the work done by us and help improve quality control in additive manufacturing processes.

REFERENCES

[1] Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, and M.-H. Yang, "Deep image harmonization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] D. Cannizzaro, A. G. Varrella, S. Paradiso, R. Sampieri, Y. Chen, A. Macii, E. Patti, and S. Di Cataldo, "In-situ defect detection of metal additive manufacturing: an integrated framework," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 1, pp. 74–86, 2021.

[3] S. Sun, M. Brandt, and M. Easton, "Powder bed fusion processes: An overview," *Laser additive manufacturing*, pp. 55–77, 2017.

[4] Z. Guo, D. Guo, H. Zheng, Z. Gu, B. Zheng, and J. Dong, "Image harmonization with transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 14 870–14 879.

[5] T. Hinz, M. Fisher, O. Wang, and S. Wermter, "Improved techniques for training single-image gans," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1300–1309.

[6] W. Cong, J. Zhang, L. Niu, L. Liu, Z. Ling, W. Li, and L. Zhang, "Dovenet: Deep image harmonization via domain verification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8394–8403.

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[10] K. Simonyan, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

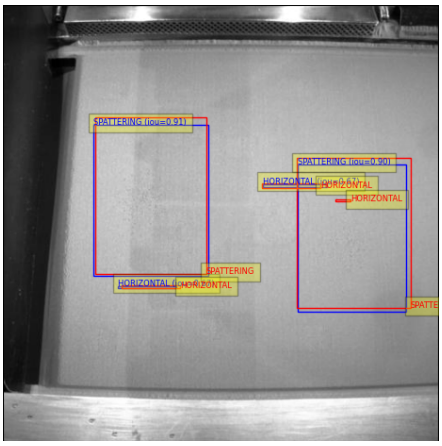
[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

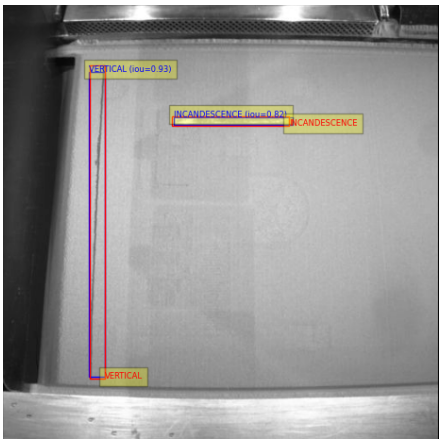
[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[15] R. Girshick, "Fast r-cnn," *IEEE International Conference on Computer-Vision (ICCV)*, 2015.

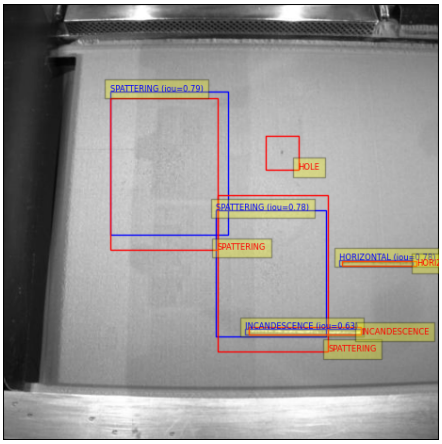
APPENDIX



(a) Spattering perfectly detected, missed one horizontal detection.

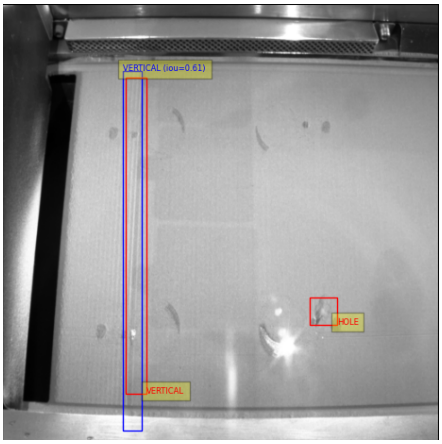


(b) Incandescence and vertical defects detected.

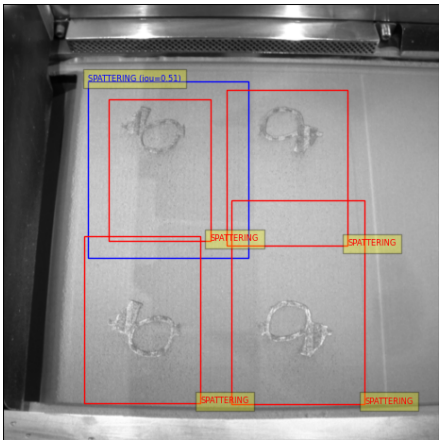


(c) Holes are not detected

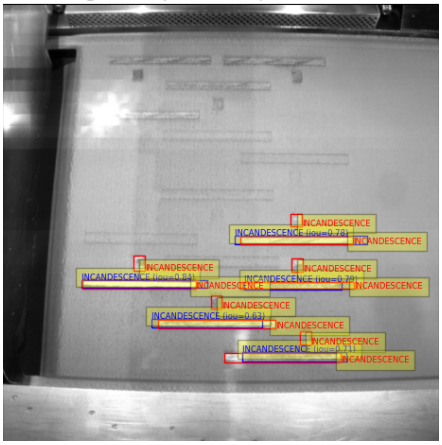
Fig. 8: Object Detection on three synthetic images. Faster R-CNN is not able to detect holes. On the other hand it detects spattering, horizontal and incandescence without any problem, with spattering being the most precise detection.



(a) Vertical lines are detected, but holes are not.



(b) Spattering is mostly not detected.



(c) Incandescence is detected only when the defect is large enough.

Fig. 9: Object Detection on three real images. Spattering is detected with low confidence, or not detected at all, probably due to the shapes that overlap on it. Incandescence is not detected when it is too small, like holes.