Summarization of AI project 3D printing defect

Michal Raczkowski

26-01-2024

Contents

1 Objectives

- **Objective**: Identify ”spaghetti” defects in 3D printed models.
- **Target Variable**: ”Defect Status” (0 for absent, 1 for present).

2 Goal

- **Goal**: Detect ”spaghetti” defects at an early stage of 3D printing to halt the process, thereby conserving materials and electricity.

3 Data Requirements

- **Type**: Images
- **Color**: Black and White (Grayscale)
- **Format**: .jpeg, .png
- **Resolution**: min. 640x480 px
- **Content**: Displaying ”spaghetti” defect with nozzle above and print on the 3d printer bed
- **Example**:

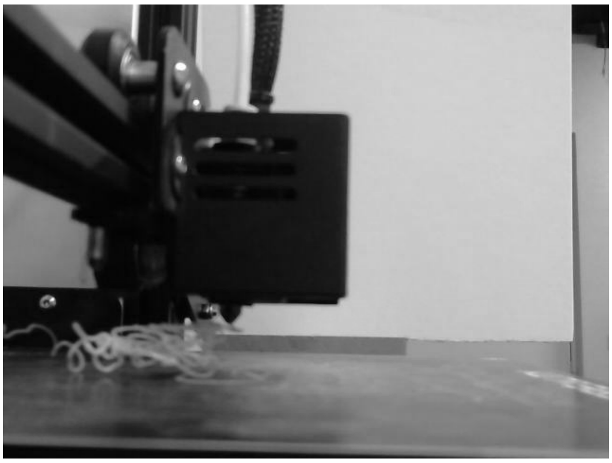

Figure 1: Example picture [1]

4 Data Sources

- Homemade pictures and open-source online repositories. [1] [2]

5 Data Preparation

- Choose relevant (with ”spaghetti” defect) images from datasets
- Change color to black and white (grayscale)
- Rotate some of them

6 Data Legality and Ethics

- Data legally obtained with rights of open source

7 Data Diversity

- Marge images from online sources and homemade images, some images are rotated to increase amount of data and increase diversity

8 Version Control

- GIT

9 Used tools

- Computer Vision Annotation Tool: CVAT [3]
- Object Tagging Tool: VoTT [4]
- Labeling tool: MakeSense [5]
- Image processing tool: Resizepixel [6]

10 Modeling

- **Approach**: Use YOLOv8 (CNN)
- **Metrics** IoU, AP, mAP, Precision, Recall, F1 Score.

11 Code

The absence of a notebook is not due to a lack of extensive code for training the model; rather, it’s attributed to the nature of YOLOv8. Additionally, data manipulation was carried out using external tools, as this approach was more efficient.

Here is code:

```
from ultralytics import YOLO

model = YOLO("yolov8n.yaml")

results = model.train(data="config.yaml", epochs=20)
```

Here is configuration which is crucial for YOLOv8:

```
ath: /home/michal/Repos/OpenWeekProject/data # dataset root dir
train: images/train # train images (relative to ’path’)
val: images/val # val images (relative to ’path’)

# Classes
names:
  0: spaghetti
```

12 Results

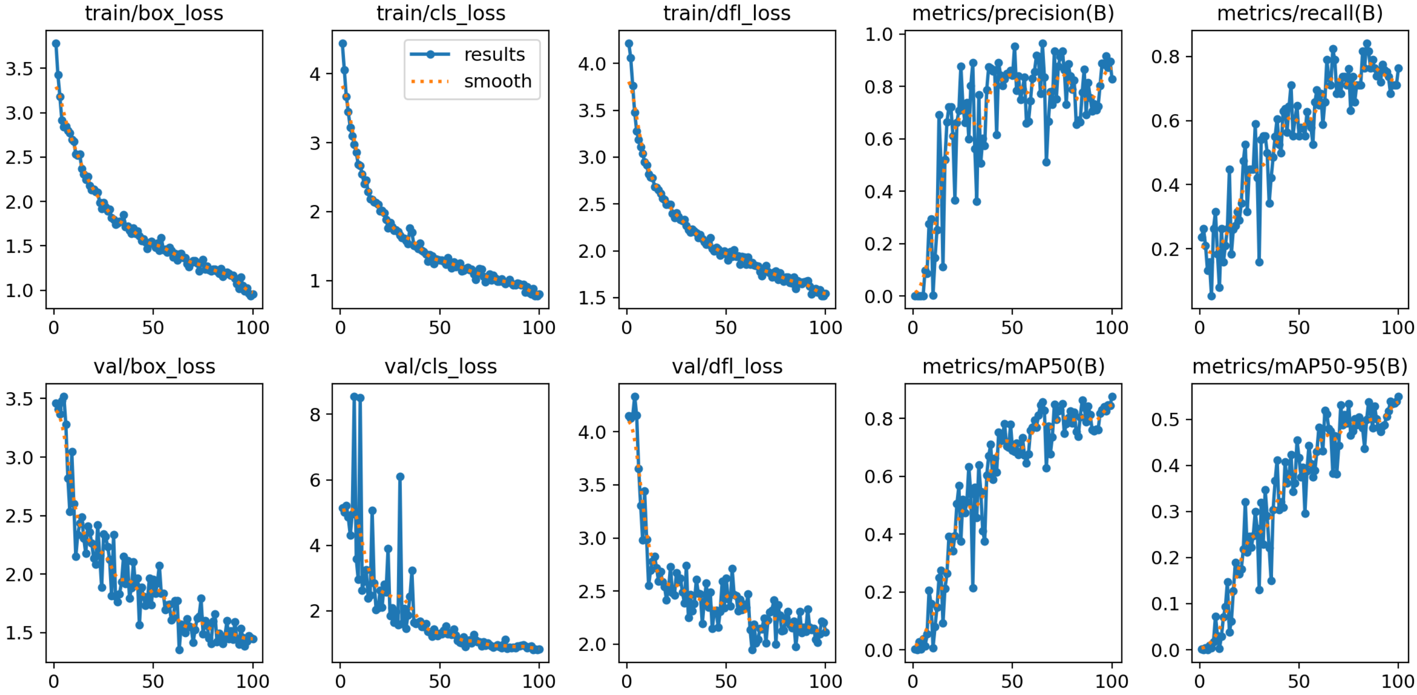

Figure 2: Graph Result

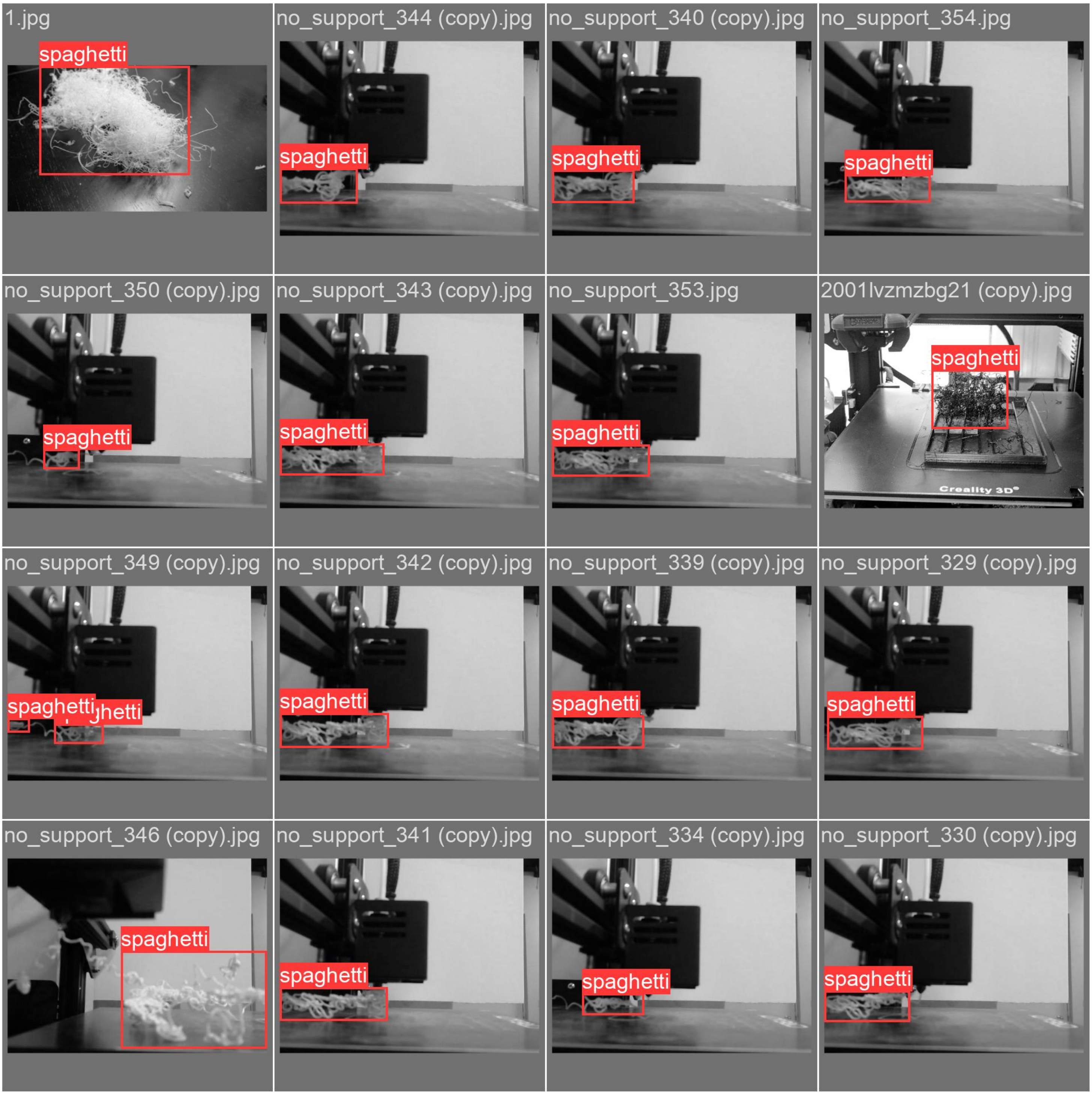
1.jpg
spaghetti
no_support_344 (copy).jpg
spaghetti
no_support_340 (copy).jpg
spaghetti
no_support_354.jpg
spaghetti
no_support_350 (copy).jpg
spaghetti
no_support_343 (copy).jpg
spaghetti
no_support_353.jpg
spaghetti
2001lvzmzbg21 (copy).jpg
spaghetti
Creality 3D
no_support_349 (copy).jpg
spaghetti
ghetti
no_support_342 (copy).jpg
spaghetti
no_support_339 (copy).jpg
spaghetti
no_support_329 (copy).jpg
spaghetti
no_support_346 (copy).jpg
spaghetti
no_support_341 (copy).jpg
spaghetti
no_support_334 (copy).jpg
spaghetti
no_support_330 (copy).jpg
spaghetti

Figure 3: Labels on Images

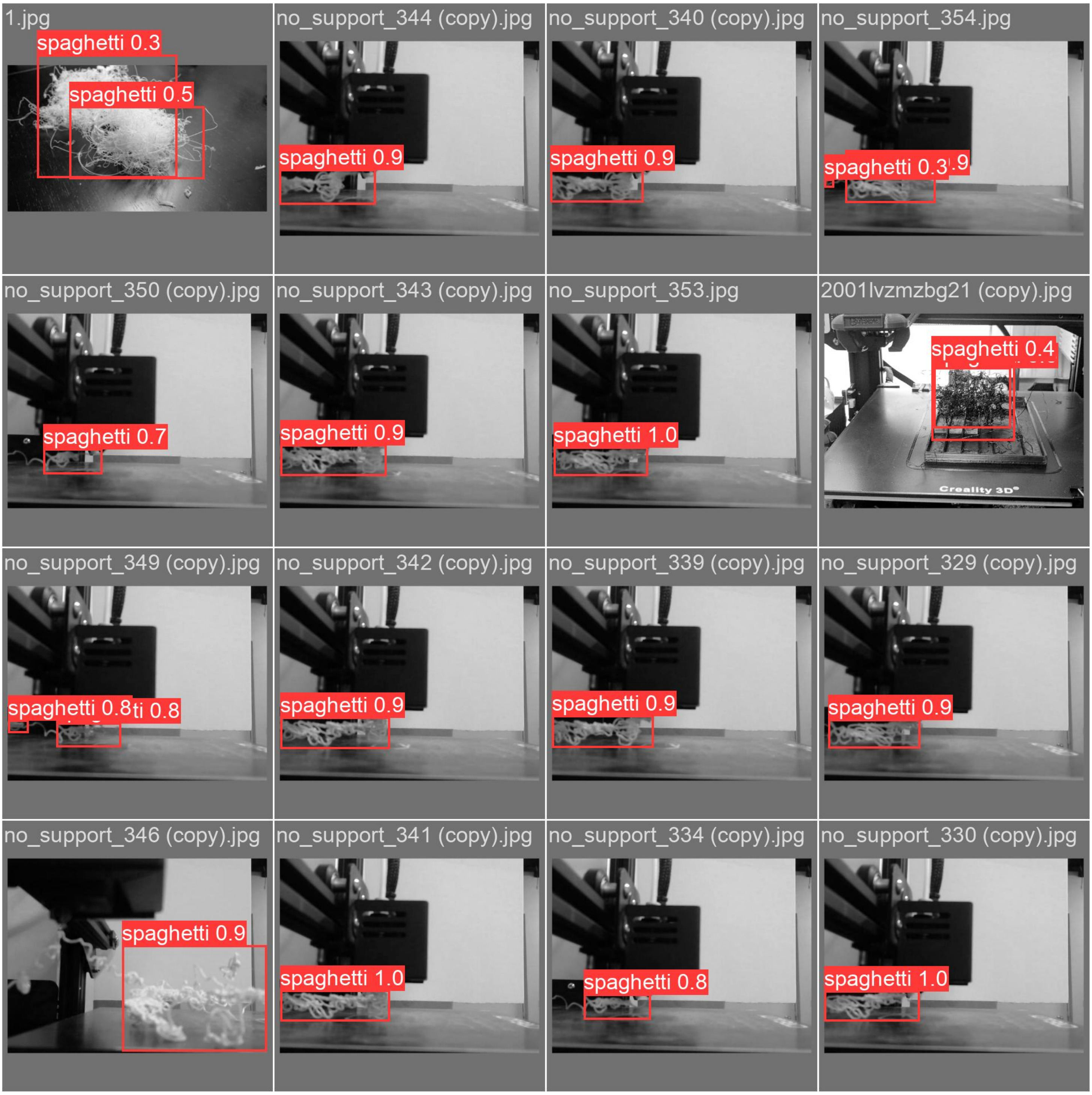

Figure 4: Prediction of model on Images

13 Conclusion

The model's accuracy is currently satisfactory, but there is potential for improvement by incorporating additional data that includes real-life images.

References

[1] Dataset: 3D-Printer Defected Dataset:
https://www.kaggle.com/datasets/justin900429/3d-printer-defected-dataset

[2] Dataset: 3D printing errors:
https://www.kaggle.com/datasets/mikulhe/3d-printing-errors

[3] Computer Vision Annotation Tool: CVAT:
https://www.cvat.ai/

[4] Visual Object Tagging Tool: VoTT:
https://github.com/microsoft/VoTT

[5] MakeSense: MakeSense:
https://www.makesense.ai/

[6] Resizepixel: Resizepixel:
https://www.resizepixel.com