

AI Fix Roads SU Data School Hackathon

**Jan Dalhuysen Hugo Bruwer Nathan Sparg Dannike
Zietsman Joseph Carmody**

Introduction

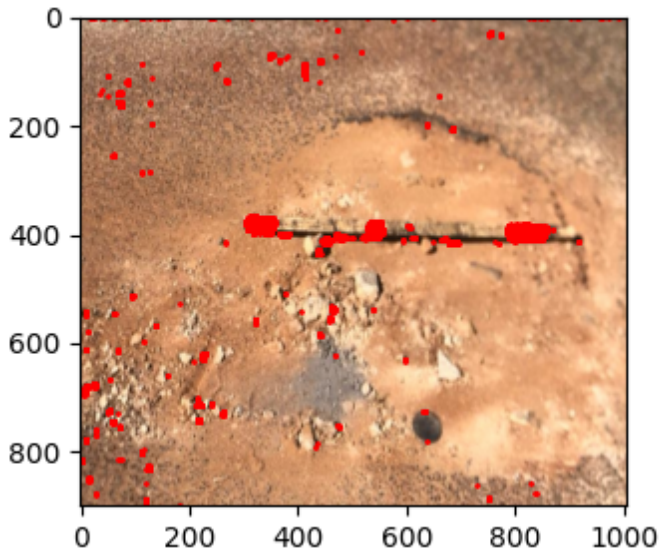
Predict the amount of asphalt required to fill a pothole given just an image of that pothole? How might you commercialise this model? How can your model be used practically in industry?

Detecting Stick length

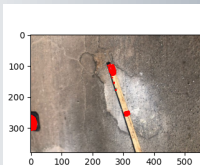
Disjoint-set data structure Invented 1964 by Bernard A. Galler and Michael J. Fischer In computer science, a disjoint-set data structure, also called a union-find data structure or merge-find set, is a data structure that stores a collection of disjoint (non-overlapping) sets. Equivalently, it stores a partition of a set into disjoint subsets. It provides operations for adding new sets, merging sets (replacing them by their union), and finding a representative member of a set. The last operation makes it possible to find out efficiently if any two elements are in the same or different sets.

-Wikipedia For detecting the length of the stick in the image, the disjoint-set algorithm was used. The algorithm was implemented in Python using the **scikit-image** library, that modified the image to only display the red pixels in the image. The DSU then groups red pixels together, while discarding other clutter, like red pixels that don't form part of the red paint on the stick.

Detecting Stick length



Detecting Stick length

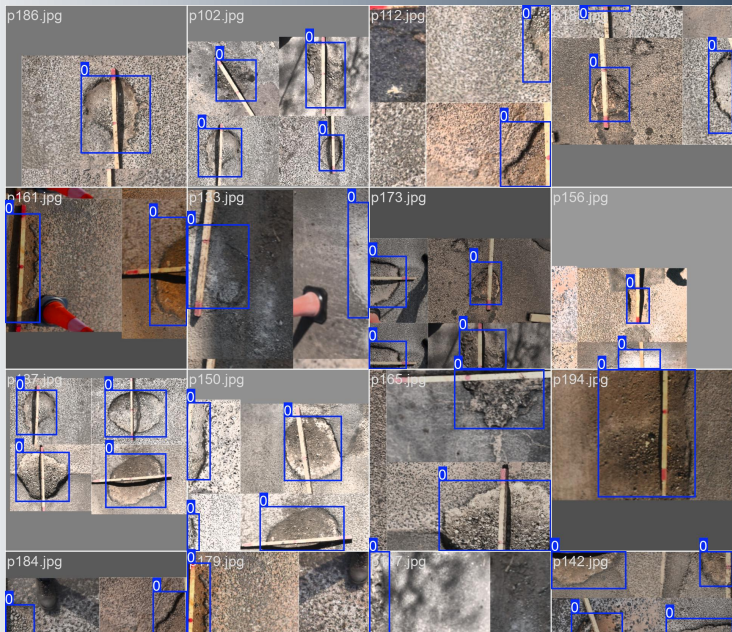


```
There are 3 groups with more than 50 points.
Average x value of group: 265.49655172413793
Average x value of group: 311.7130434782609
Average x value of group: 6.957142857142857
Average y value of group: 112.78275862068965
Average y value of group: 252.30434782608697
Average y value of group: 283.8095238095238
Group 1 - Average (x, y): (265.49655172413793, 112.78275862068965)
Group 2 - Average (x, y): (311.7130434782609, 252.30434782608697)
Group 3 - Average (x, y): (6.957142857142857, 283.8095238095238)
There are no groups that are very close to each other.
Group 1 - Average (x, y): (265.49655172413793, 112.78275862068965)
Group 2 - Average (x, y): (311.7130434782609, 252.30434782608697)
Group 3 - Average (x, y): (6.957142857142857, 283.8095238095238)
The distance between the first and second groups is 146.97699807949053 pixels.
The distance between the second and third groups is 306.3800500638645 pixels.
The distance between the first and third groups is 309.98835518169363 pixels.
The sum of distances between groups 1 and 2, and groups 2 and 3 is not close to
Groups 1 and 2 are the closest to each other.
The distance between the two closest groups is 146.97699807949053 pixels.
```

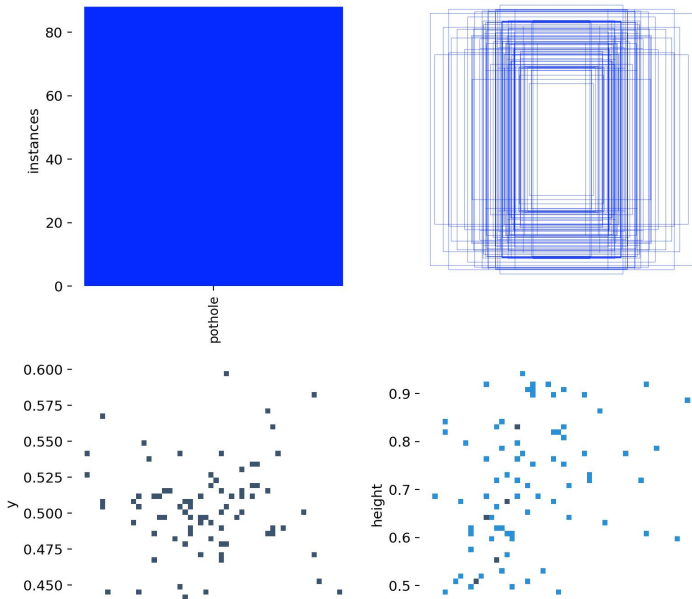
Detecting Pothole dimensions

The pothole yolov8 nano model by Hugging Face user sovitrath was initially used to detect the pothole in the image, however this model was not good enough to detect the dimensions of the pothole reliably. The next step was then to use the base YoloV8 model from the ultralytics library and then train it on the pothole images and labels provided. The training photos with their labels showing the bounding boxes of the potholes and the models improvement over time are clearly shown below.

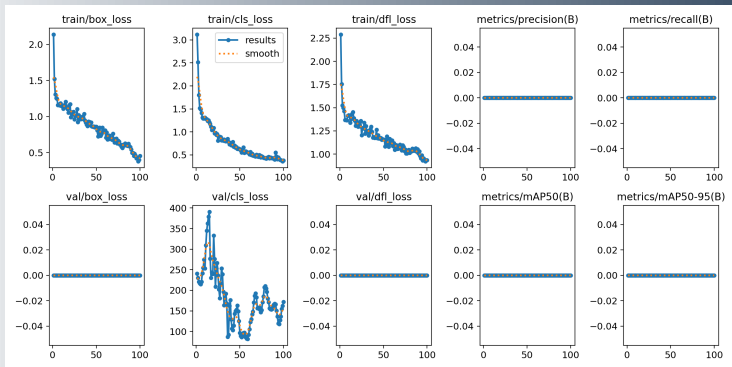
Detecting Pothole dimensions



Detecting Pothole dimensions



Detecting Pothole dimensions



Predicting

To find the correlation between the area of the pothole and the number of bags of asphalt used that is needed to fill them the **scikit-learn LinearRegression** model was used. The model was trained on the data set that was provided and the plot below clearly shows the correlation between the two variables. As the area of the pothole increases so does the number of bags of asphalt needed to fill it. An example of the csv file used to train the model is also included below.

Detecting Pothole dimensions

```
import pandas as pd

from sklearn.linear_model import LinearRegression

data = pd.read_csv("pothole_area_vs_bags_used.csv")

X = data.drop(columns=['output'])
y = data["output"]

model = LinearRegression()
model.fit(X, y)
```

Example of csv file:

```
input,output
0.01807917701,0.25
0.0247474859,0.25
0.03104582122,0.25
...
0.3071829833,0.5
0.3151921838,0.5
0.3200815111,0.5
...
0.4896235766,1
0.4922149956,1
0.4966567092,1
...
0.7860071056,1.5
...
0.7998276765,2
0.8079945704,2
0.8811756529,2
0.9715512358,2
1.118149991,2
1.323262264,2
1.3691919,2
1.446414526,2
...
```

Detecting Pothole dimensions

