

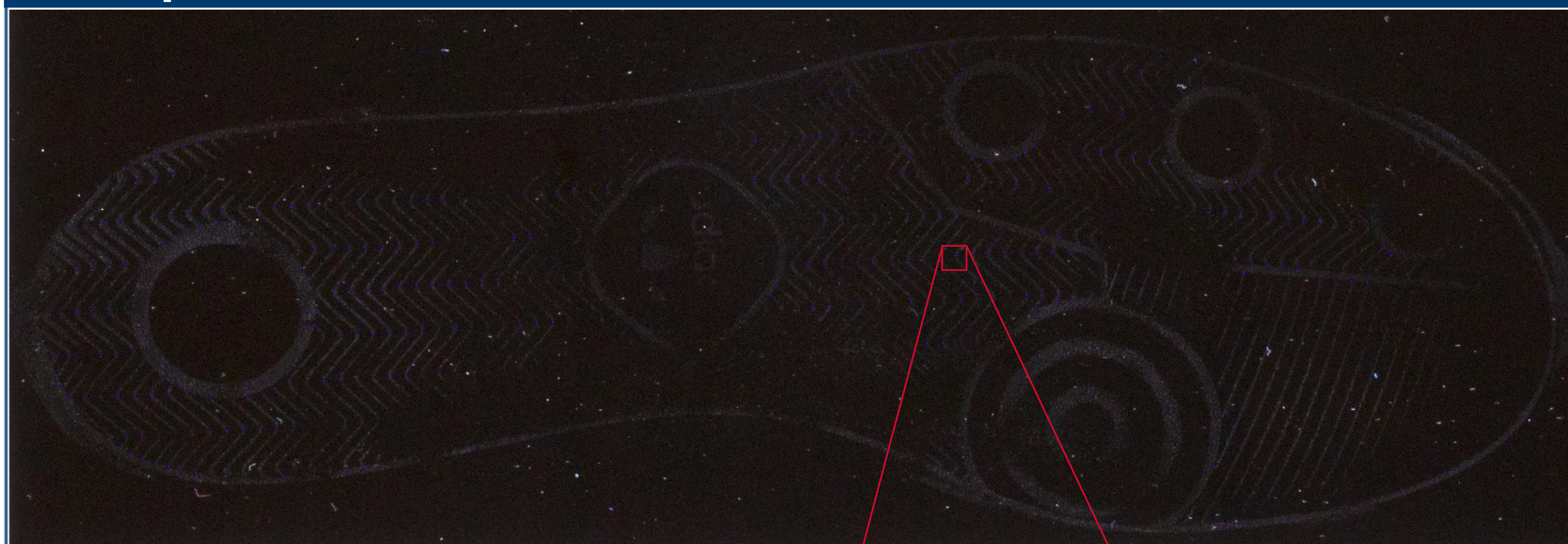
Introduction

- The goal was to create a machine learning algorithm to detect interest points on pictures of shoe treads, starting with corners
- Current corner detection methods are only good at picking up on sharp, well defined corners which shoeprints rarely have
- In the future, this type of technique could be used to aid further analysis by mapping the corner points onto shoe treads before more advanced techniques could work

Methods

- We first manually marked shoeprint photos with dots on the corners, and sampled from these images to create our dataset of corner pixels and non-corner pixels
- Sampling involved taking pixels labeled corners, and the surrounding areas, and non corner pixels with surrounding area
- We then experimented with different neural network structures to optimize accuracy on testing data that the network hasn't seen
- Examples of hyperparameters we checked are learning rates, number of training cycles, image size, and layer numbers
- The network was constructed with Python code, and the PyTorch package

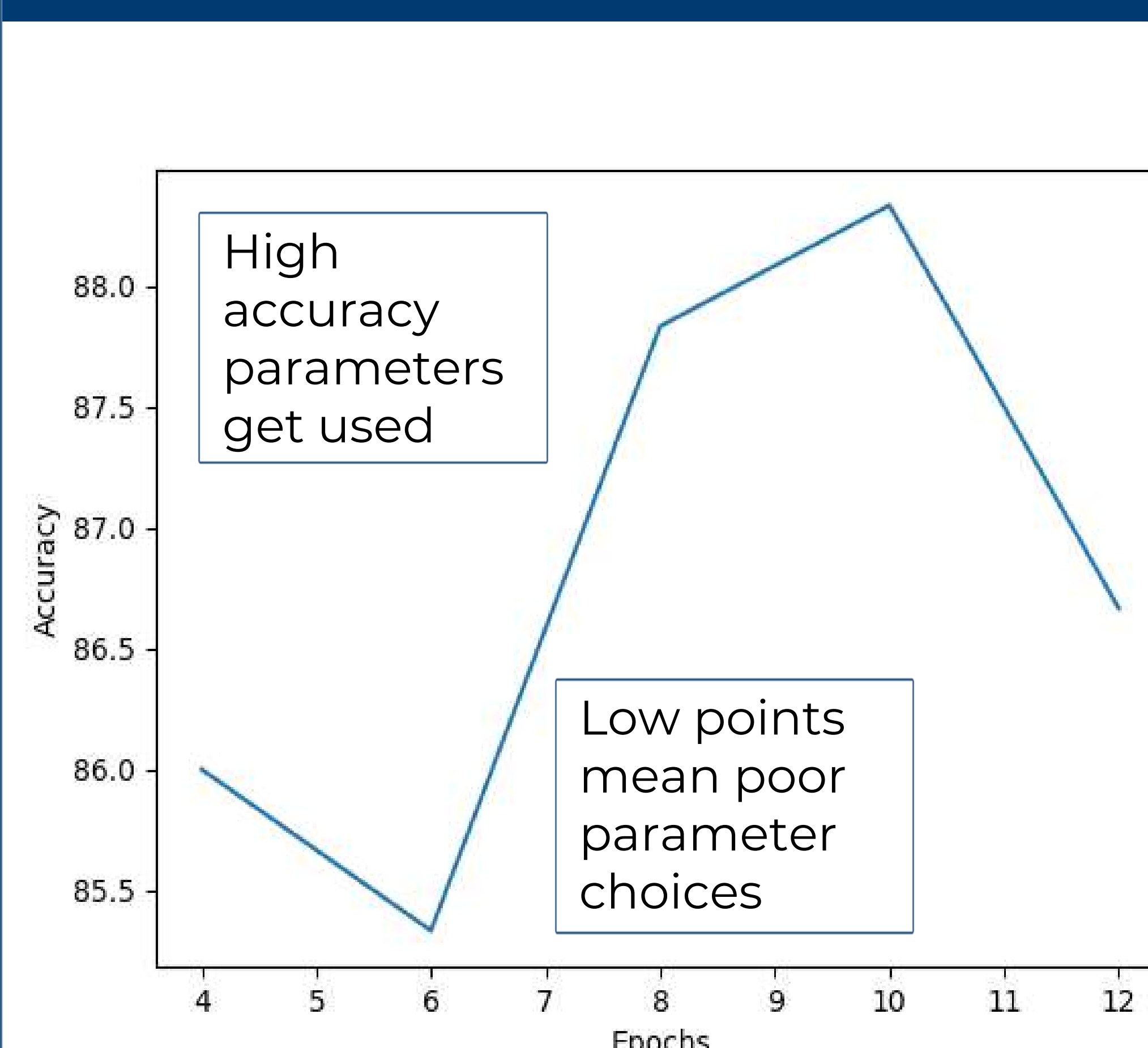
Example of a Marked Photo



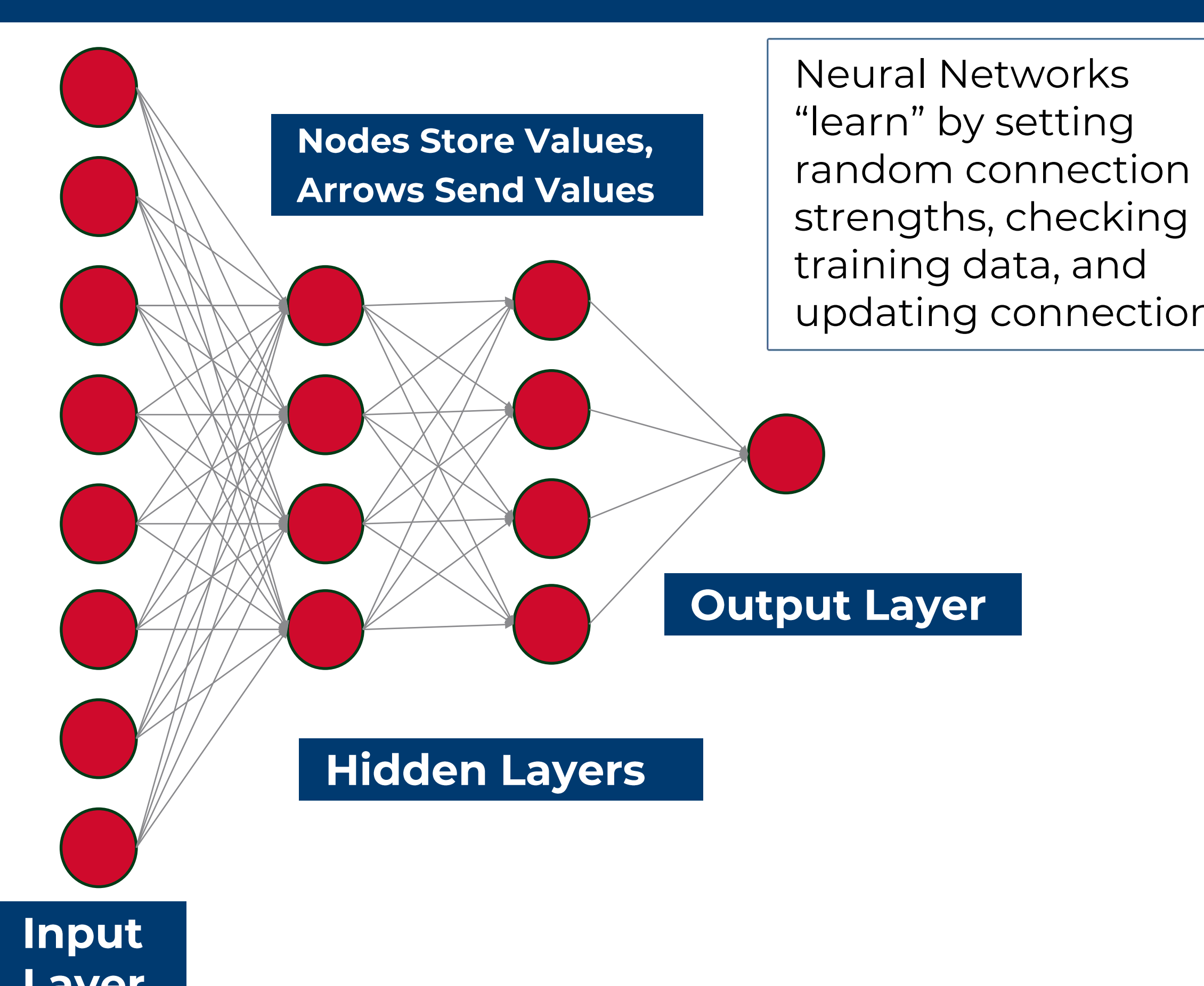
This square is a piece of data given to the network. The network must detect the fuzzy white lines that create a corner.



Parameter Data



What is a Neural Network?



Example Code

```
class Net(nn.Module):
    def __init__(self, in_shape):
        super().__init__()

        self.features = nn.Sequential(
            nn.Conv2d(in_channels=1, out_channels=3, kernel_size=5, padding=2),
            nn.BatchNorm2d(3),
            nn.MaxPool2d(2, 2),
            nn.ReLU(inplace=True)
        )

        self.classification = nn.Sequential(
            nn.Linear(108, 66),
            nn.ReLU(inplace=True),
            nn.Linear(66, 66),
            nn.ReLU(inplace=True),
            nn.Linear(66, 1),
            nn.Sigmoid()
        )

    def forward(self, x):
        x = self.features(x)
        x = x.reshape(x.shape[0], -1)
        x = self.classification(x)
        return x
```

Results

- Largely successful, creating a model with 95% accuracy on data it had never seen before
- Simple models worked best overall, allowing for strong outcomes while preventing overfitting and speeding up computation times
- A single convolutional layer followed by 2 fully connected linear layers worked best

Conclusion

This model will need to be continuously improved, because shoeprint pictures contain thousands of pixels and marking even just 5% of them incorrectly means hundreds of wrong marks. What we have now will work as a powerful starting point for this future research.

After interest points can be quickly marked by a network, more advanced statistical comparisons can be made.