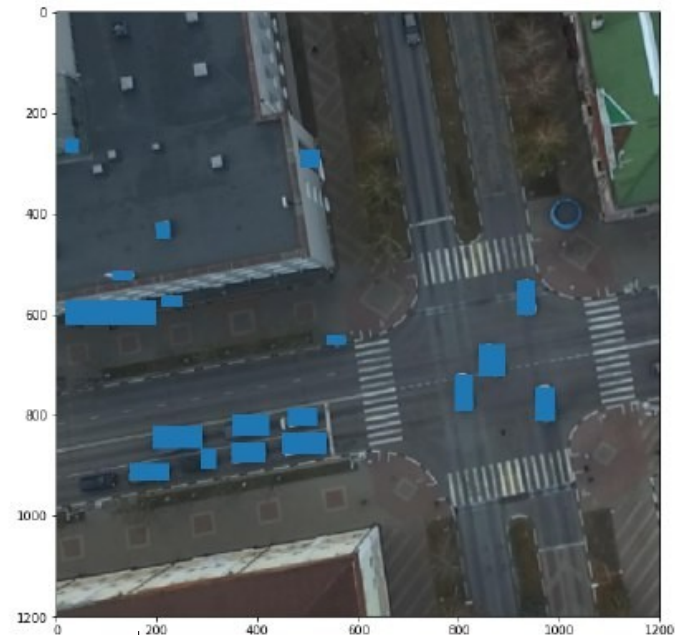
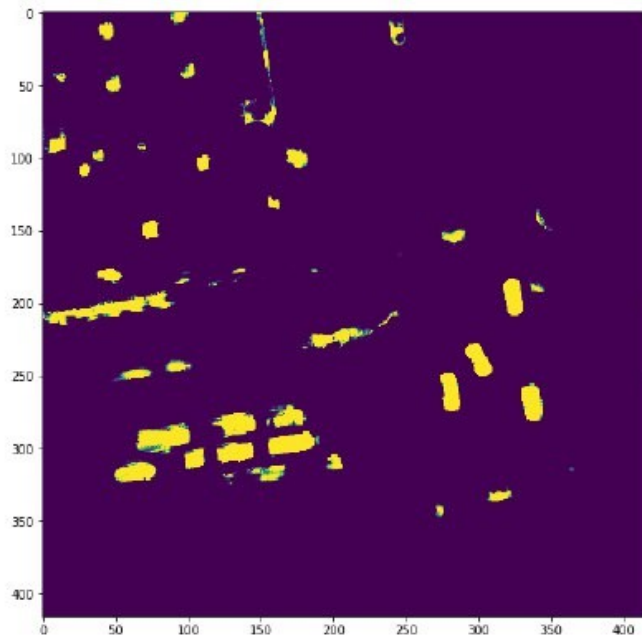


Results on the task:

Computer vision and reinforcement learning

Amin Kashi

Edward Paul



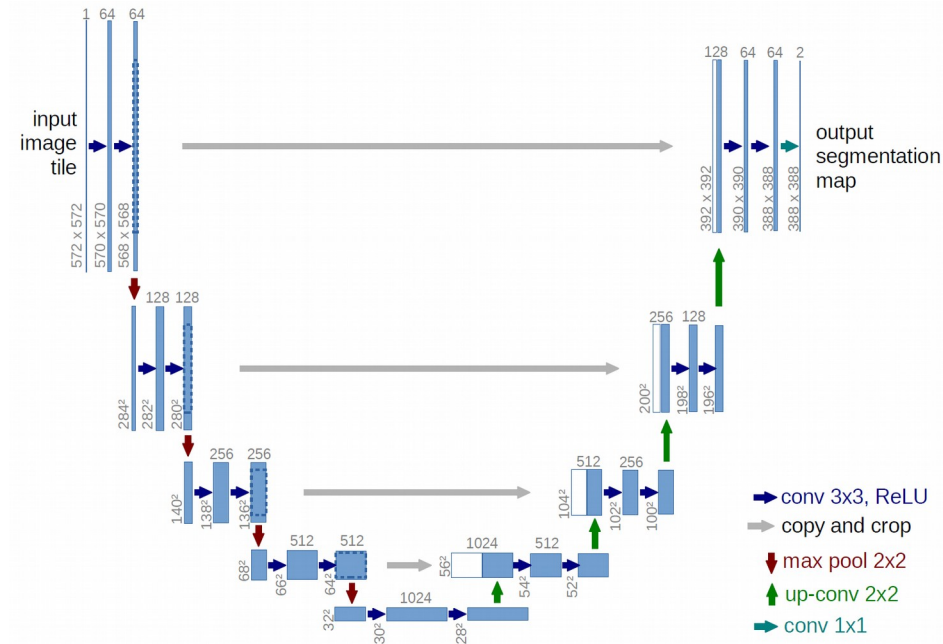
Action

Agent

States

Location of cars and obstacles

Car recognition



+



scikit-image
image processing in python

U-net DNN

Reinforcement learning

Dense : 100 neurons, ReLU



```
graph TD; A[Dense : 100 neurons, ReLU] --> B[Dense : 100 neurons, ReLU]; B --> C[Dense : 4 neurons, sigmoid];
```

The diagram illustrates a neural network architecture for reinforcement learning. It consists of three layers, each represented by a blue rectangular box. The top layer is labeled 'Dense : 100 neurons, ReLU'. A vertical arrow points down from this layer to the middle layer, which is also labeled 'Dense : 100 neurons, ReLU'. Another vertical arrow points down from the middle layer to the bottom layer, which is labeled 'Dense : 4 neurons, sigmoid'.

Dense : 100 neurons, ReLU

Dense : 4 neurons, sigmoid

Future

- More real examples instead of synthesized
- A faster and lighter network for car detection



U-Net + Atrous convolution
(or ASPP)

- Using the segmentation map as the state of the environment

