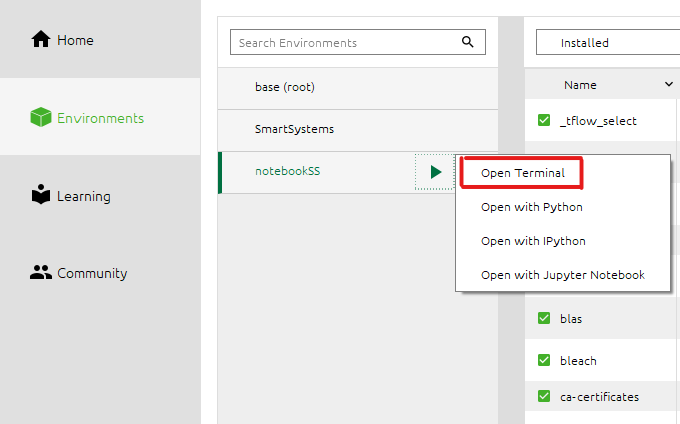
Streamlit

Deze handleiding beschrijft hoe men streamlit applicaties kan openen En hoe je zelf een eigen streamlit applicatie via notebook op heroku kan deployen.

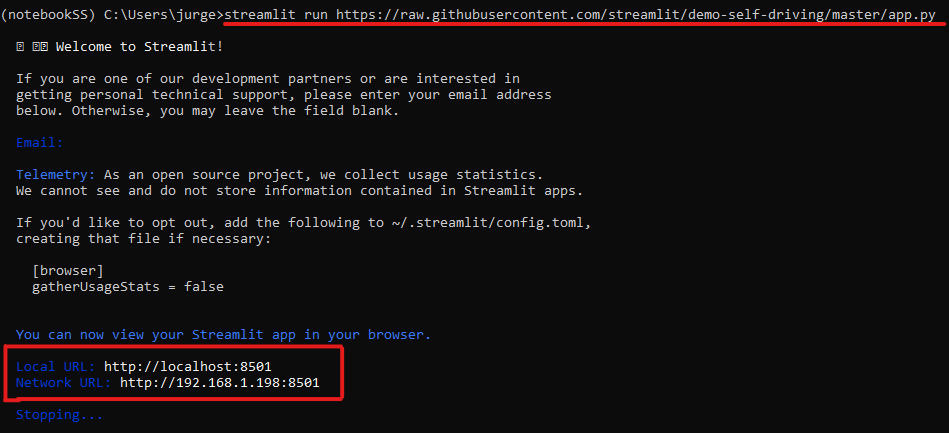
# Demo applicaties:

## Openen van streamlit applicatie demo Self-driving Car

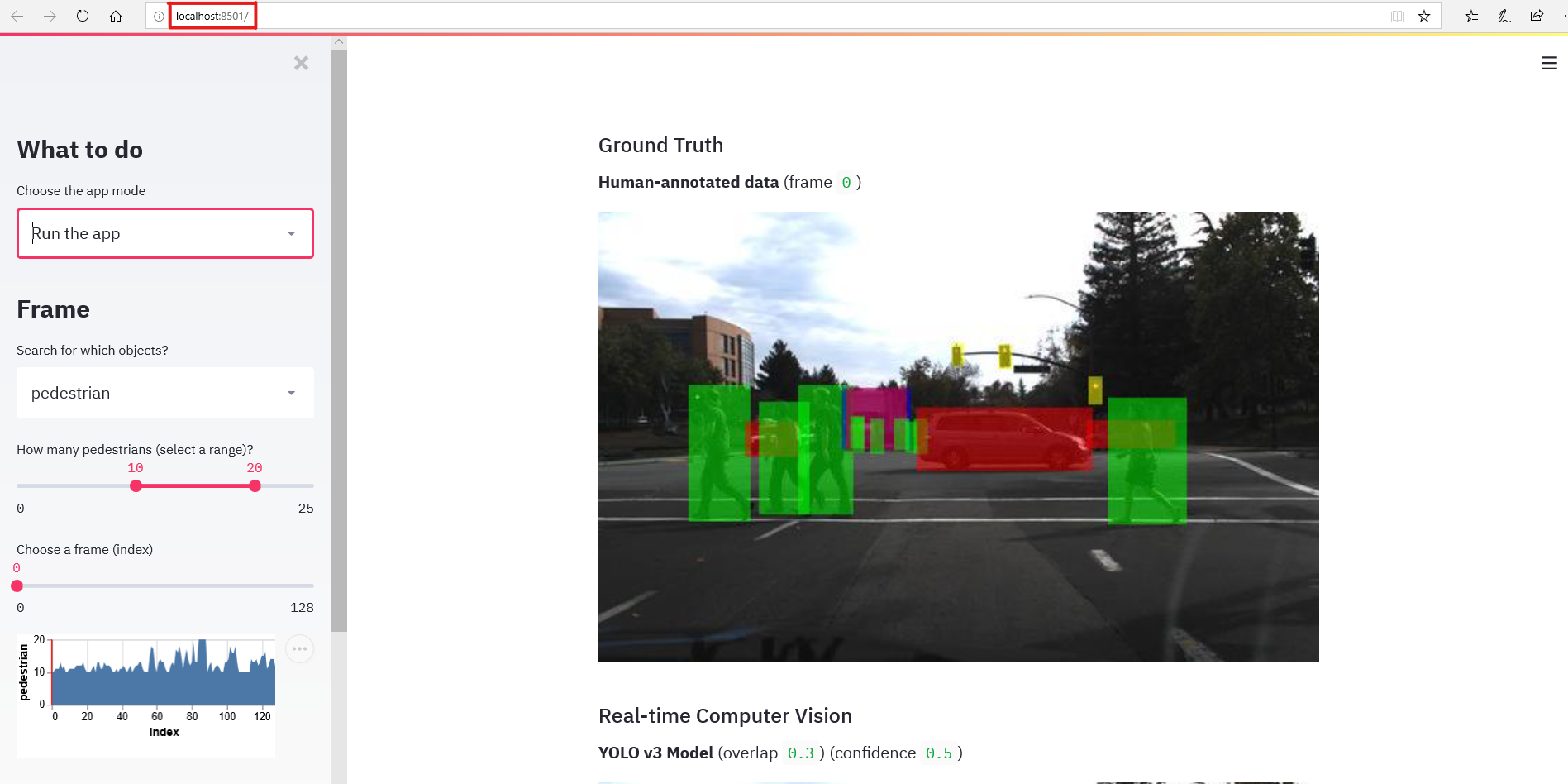
1. Open terminal.



1. Pip install Streamlit.
2. pip install --upgrade streamlit opencv-python.
3. streamlit run <https://raw.githubusercontent.com/streamlit/demo-self-driving/master/app.py>

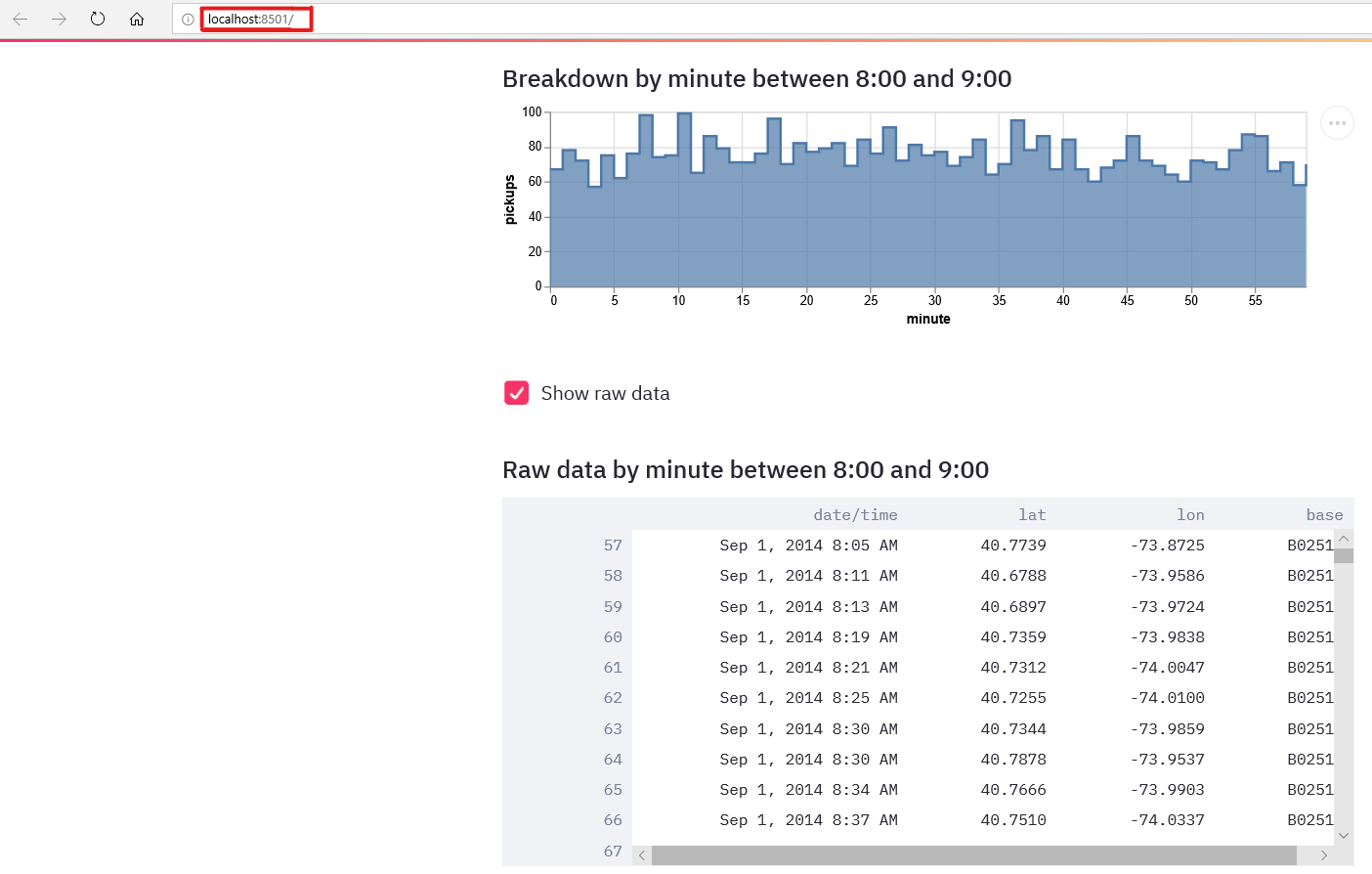


1. Normaal start de applicatie vanzelf anders vind je deze in de opgegeven ip-adres.
2. De applicatie gaat open in een internet tablad.
3. Links zie je de data inputs, dit zijn verschillende selectie-/ inputboxen en sliders die je kan veranderen om op de foto rechts andere dingen weer te geven. Dit zorgt voor een directe aanpassing die zichtbaar is op je app.



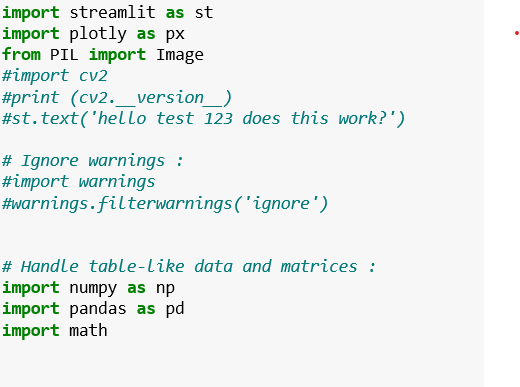
## Openen van streamlit applicatie demo uber drives

1. Open terminal (zie vorige).
2. pip install --upgrade streamlit (indien er een nieuwe versie is, anders ga je naar stap 3).
3. streamlit run <https://raw.githubusercontent.com/streamlit/demo-uber-nyc-pickups/master/app.py>

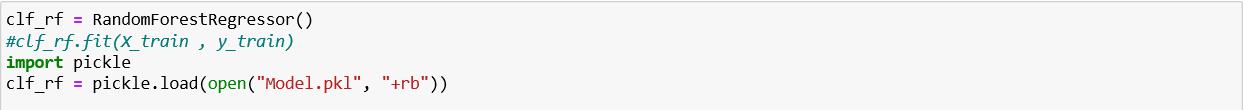


# Deploy een streamlit applicatie op Heroku

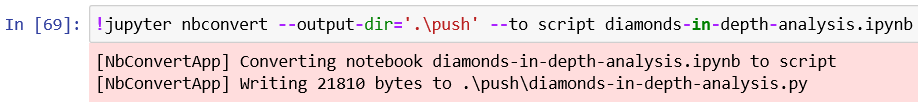
1. Open een jupyter notebook.
2. Importeren van de juiste libraries.

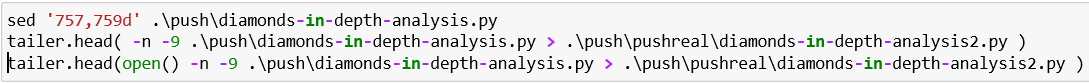


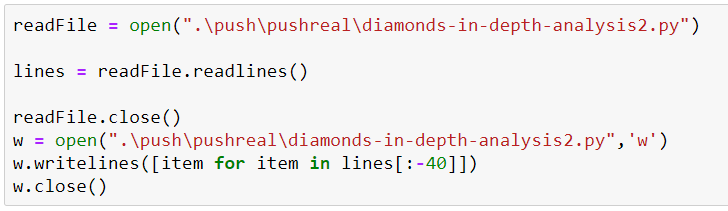
1. Inlezen van een pickle in notebook.



1. Converteren de notebook file naar een python file en sla het op een gekende plaats op.



1. Snij de laatste commando’s weg uit de file.
   1. Linux commando’s: Deze werken dus niet standaard op windows. Vervolgens maken we een copy van deze file en slaan hem op in een andere file die dan naar Heroku wordt gepushd.
   2. Python commando’s: Deze werken zowel in windows als in linux.

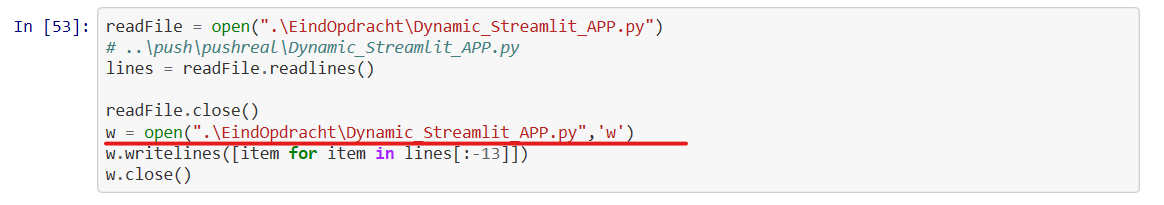


1. Push nu de hele map naar heroku via git commit en git push (je kunt dit best twee maal achter elkaar doen want soms vindt hij bij de eerste push geen veranderingen).



# Maken van een streamlit applicatie

1. /\*Open en maak een python file in notebook
2. Download de file als een ‘.py’ file



1. Open in uw environment de terminal (zie eerste)
2. Streamlit run “path locatie van de ‘.py’ file die je hebt gedownload”
3. Site gaat automatisch open
4. Categorieën bekijken
5. inputs koppelen in python file
6. Code voor het voorspellen schrijven naar python file
7. In notebook via !git commiten naar heroku\*/

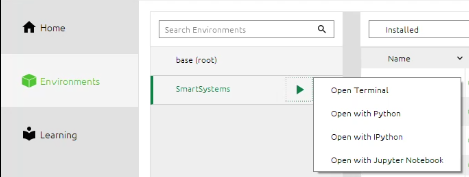
Numbers als input kunnen voor errors zorgen.

# Uitleg eigen applicaties:

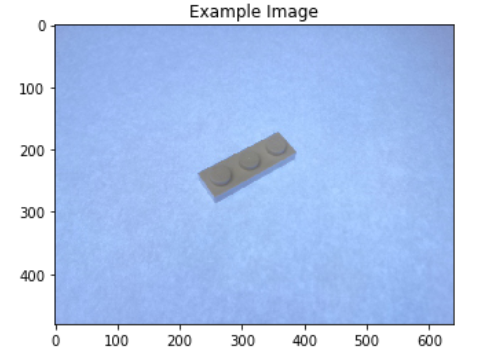
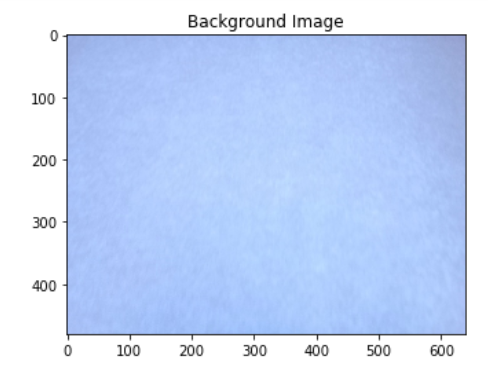
## Lego Dataset

Het doel van deze notebook is om verschillende soorten lego blokken te herkennen aan de hand van verschillende foto’s. Dit gaan we doen door middel van OpenCV.

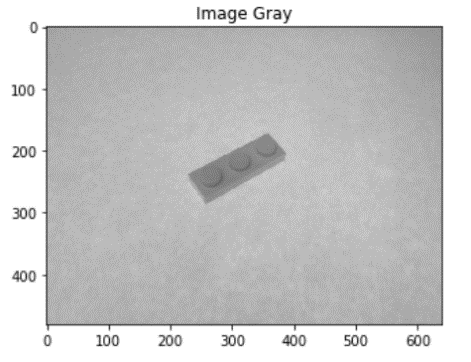
1. Open anaconda
2. Open uw environment
3. Open met jupyter notebook



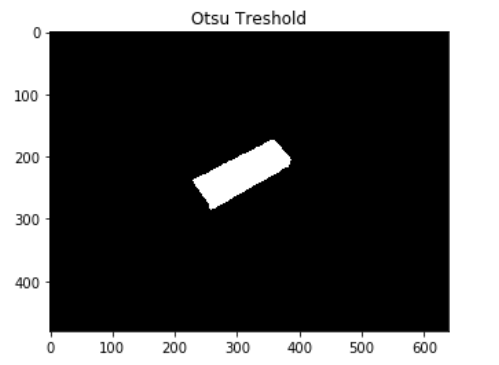
1. De notebook gaat open in een internet tablad.
2. Selecteer de map en open de python file.
3. Run de code met .
4. Als de applicatie klaar is.
   1. Zie je afbeeldingen van bepaalde lego blokken en van de achtergrond.



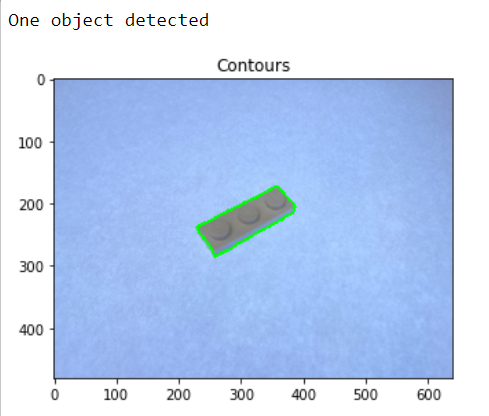
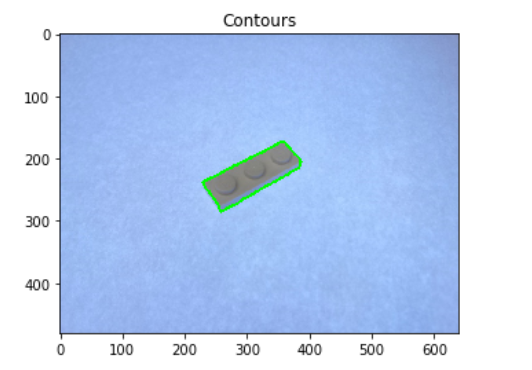
* 1. Vervolgens worden de afbeeldingen in Grayscale gezet.



* 1. De achtergrond word zwart en het object zelf word wit gemaakt zodat het onderscheid tussen object en achtergrond groot is.



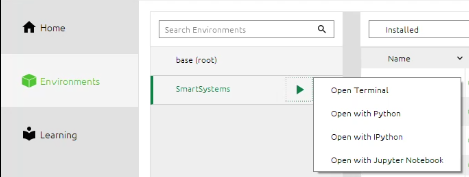
* 1. Er word een contour gemaakt rond het object dit is handig als er meerdere objecten zijn om ze zo gemakkelijk uit elkaar te halen en om te weten hoeveel objecten er zijn.



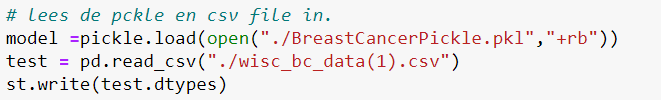
## Automatische Pickle file

In deze applicatie gaan we verschillende csv en pickle files online zetten op heroku, zodat je deze kan gaan gebruiken als streamlit applicatie.

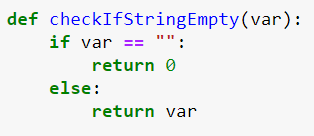
1. Open anaconda
2. Open uw environment
3. Open met jupyter notebook



1. De notebook gaat open in een internet tablad.
2. Selecteer de map en open de python file.
3. Run de code met .
   1. Er worden files ingelezen (csv en pickle)



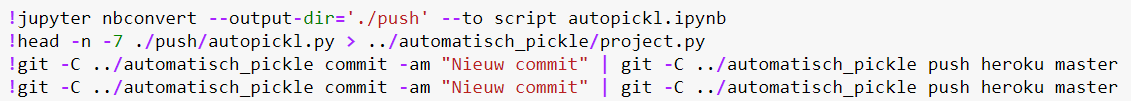
* 1. Een functie controleert of er geen lege string gebruikt word.



* 1. We doorlopen alle kolommen die er zijn in de csv file.



* 1. We kennen aan elke kolom toe of het een ingang of een uitgang is.
  2. Daarna converteren we de notebook file naar een python file.
     1. We verwijderen de laatste regels omdat deze na het runnen niet meer uitgevoerd mogen worden.
  3. We pushen de python file daarna naar heroku, waardoor er een streamlit applicatie aangemaakt word die je kan raadplegen.



1. Als de applicatie klaar is kan je applicatie openen.
   1. Je kan de kolommen zien en welke type elke kolom is.

## Diamond