Sprint 01: Induction to Data Science

*“Detecting Ships in Satellite Imagery”*

|  |  |
| --- | --- |
| **Stakeholder** |  |
| **Business contact** |  |
| **Duration** | 5-10 days |

# Overview:

## Key findings:

1. Using modules such as matlabplot and keras are used in image detection for AI
2. How to train your program, through feeding it data to learn and test on; given that the longer the program can train, usually the more correct it becomes at detecting/predicting the next occurrence in data
3. How to create Python environments
4. How to download packages using conda and pip
5. How the program must be provided correct samples to train on to be able to work on new data
6. The differences between CNN and Mask R\_CNN

## Sprint Aim:

To evaluate and better understand image classification techniques.

## Sprint Objectives:

1. To develop knowledge of Python environments and create an environment in conda.
2. To run through two tutorials on image classification: one simple, one complex.
3. To review the differences, strengths and weaknesses of each image classification approach.
4. To document the skills you have learnt.

# Results

Working on two image classification approaches allow

Contents

[Overview: 1](#_Toc20210185)

[Key findings: 1](#_Toc20210186)

[Sprint Aim: 1](#_Toc20210187)

[Sprint Objectives: 1](#_Toc20210188)

[Results 1](#_Toc20210189)

[Objective 1: To develop knowledge of Python environments and create an environment in conda. 2](#_Toc20210190)

[Objective 2: To run through two tutorials on image classification: one simple, one complex. 3](#_Toc20210191)

[Objective 3: To review the differences, strengths and weaknesses of each image classification approach. 4](#_Toc20210192)

[Objective 4: To document the skills you have learnt. 5](#_Toc20210193)

## Objective 1: To develop knowledge of Python environments and create an environment in conda.

**Pip:** I became more comfortable using pip to install packages and modules into my environment, this is what was used mainly in setting up modules in python i.e. Pandas, Numpy, OS etc.

**Conda:** I used conda to not only install modules but to also set up environments – for example an environment I created called Pi, used to hold Pandas, Numpy, OS, Tensorflow modules. These environments used helped me in VS Code to run my code.

**Git Terminal:** Using the Git Bash was more of a new concept to me. I had used Github before and their prepacked software to Push and Pull projects to and from my repositories. However I learned how to use Git to manually created, push and pull repositories and edit documents.

**Anaconda Terminal:** Just like Git or CMD Anaconda is another terminal that can be used. I used this terminal to install modules, create my environments and check all packages.

Troubleshooting: There were multiple issues just setting up the project, ranging from setting up my modules and packages to checking directories. Using Github and StackOverflow I was able to pinpoint the issues and find a solution.

Containers – I haven’t used containers yet during this project, however I have researched and learnt the use for containers using docker. Containers allow one to package applications in containers, allowing them to be portable and run on various OS.

**Note of useful commands:**

**Pip install <package> --cert=P:\data\_Science\cacert.pem** - this command uses pip to install the wanted package and applies a certificate to bypass security protocols which normally prevent anyone from downloading packages.

**Conda activate <environment>** - this command uses conda to activate an environment, allowing me to use the set packages installed uniquely to that environment.

**Conda deactivate** – this command uses conda to deactivate an environment, allowing me to change environment once finished.

**Conda create –n <envName> python=x.x anaconda** – this command uses conda to create your environment and assign it a name and the Python model that will be used.

## Objective 2: To run through two tutorials on image classification: one simple, one complex.

**Simpler project**

1. Studies the dataset and saves it as an 8 -bit image
2. Inputs the data and reshapes the image and shows it through 3 channels (RGB)
3. Outputs the data and declares the set of images that are ships and not ships
4. Prepares the data by shuffling indexes, transposing the image matrix and then normalizes the data
5. Trains the data using the set of training images
6. Downloads image and plots the 8-bit image
7. Searches the image x,y times (checking each combination) and checks to see if there is a ship within each pixel and which one is near the ship. Splitting them into a list of (near and not near) to cut the image into the selected area
8. Images are then shown by plotting a box over the original image

**Complex Project**

1. Download data and save it into set paths
2. Takes RLE Mask into image mask to be used for training in the R\_CNN library
3. Package and set up the docker container to run the large dataset (can be done without docker. To get more accurate results, using AWS allows the program to train with a larger dataset).
4. Train the program
5. Predict ship segments by loading in dataset and RLE pixels; then run the predicted pixels against mask regions
6. If matched select the mask region and continue to do so until all ships are found/selected

## Objective 3: To review the differences, strengths and weaknesses of each image classification approach.

**Image classification**

**Pro**

* Better suited to a single subject I the image
* Suited for still images
* Simpler to implement on basic models

**Cons**

* Becomes more of a challenge to identify multiple objects due to interference

**Object Recognition**

**Pro**

* Can detect multiple subjects within an image
* Better suited for moving objects
* Works on multiple models

**Cons**

* Takes longer to produce results as it must focus on a single region at a time to minimise interference before CNN can classify

**Complex project VS Simpler project**

**Simple**

**Pro**

* Works well with smaller datasets

**Con**

* Not good for large datasets due to inefficient search methods

**Complex**

**Pro**

* Works well with large datasets and can be used for a wide range of cases
* Simpler search method since it converts

**Cons**

* To produce better results, the program has to be left to run for a long amount of time with the large dataset

## Objective 4: To document the skills you have learnt.

I’ve learnt;

1. How to set up and load environments using conda
2. How to download packages using conda and pip
3. How to use modules such as matlabplot and keras
4. How to train my program- through feeding it data to learn and test on; given that the longer the program can train, usually the more correct it becomes at detecting/predicting the next occurrence in data
5. How the program must be provided correct samples to train on to be able to work on new data
6. The differences between CNN and R\_CNN
7. The differences between image classification and object recognition