AI Tech Challenge - Child Seat Location

AI Solutions

Author: Dr. Ilan Sousa Figueirêdo

AI Tech Challenge

Title: Child Seat Localization

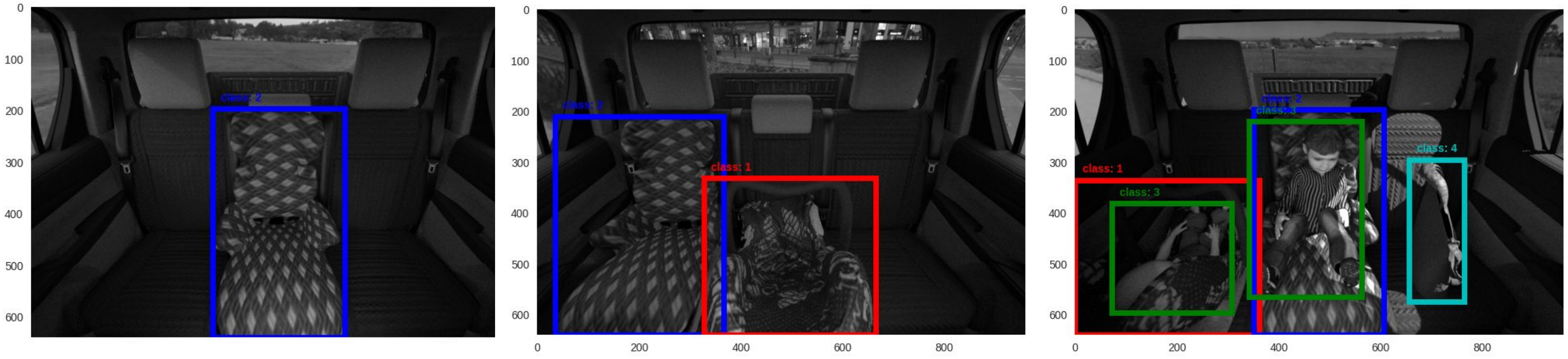
Goal

Build AI solution for child seat localization in the passenger vehicle.

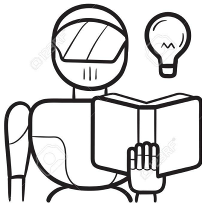
Problem statement

Localization of the child seat in the passenger vehicle enable the business to high-level services and use cases.

Data set: SVIRO



Methodology



Exploratory Data Analysis

Raw data

Understanding the SVIRO dataset

Browse other datasets

Downloaded and extracted data

Model select

Pretrained model

Deep Learning for object detection

Level of network architecture

Stable and efficient model

Light model

Preprocessing

Data for training

Color mode

Filter on images and labels

Label standardization

Organize Directories

Data set build

Model Training

Model trained

Train data set

Model parameterization

Train evaluation

Model Evaluation

Model Prediction

Test data set

Inference in test data set

Test evaluation

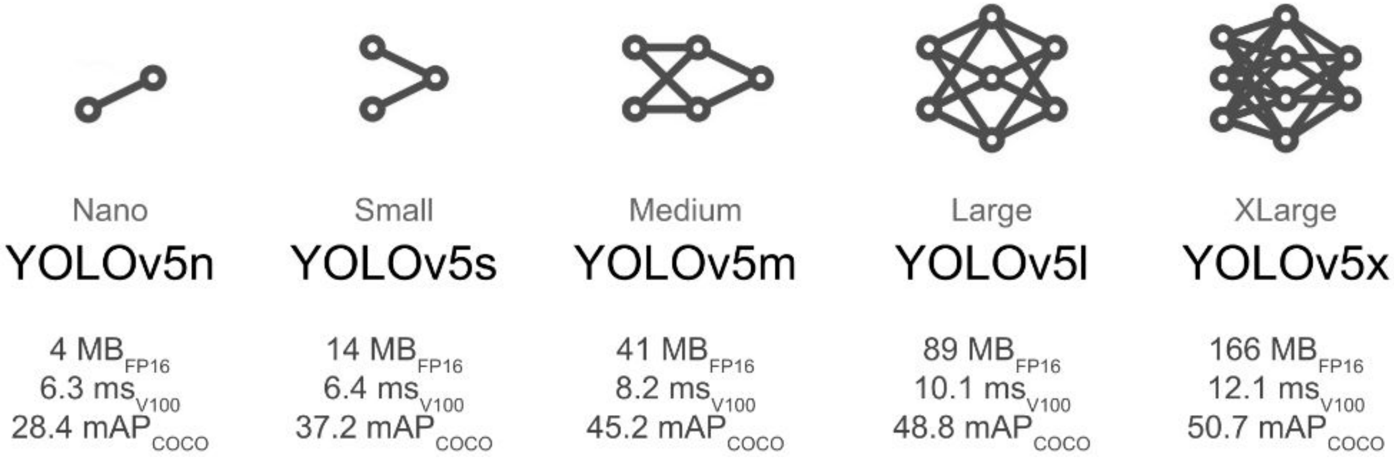
Exploratory Data Analysis

- **SVIRO and TICaM data sets**

Data set	**Train**	**Test**	**Labels**	**Classes**
SVIRO grayscale / RGB / depth images	22,000	5,000	Bounding boxes, Masks and Key points	Infants seat, child seat, person and everyday object
TICaM grayscale synthatic images	3,306	N/A	Bounding boxes, Masks and Key points	Infants seat, child seat, person and everyday object
Total	25,306	5,000		

Model Selection

The one-stage detectors, such as **YOLO, are significantly more time-efficient and have greater applicability to real-time object detection** (XIAO, et al; WU, et al, 2020).



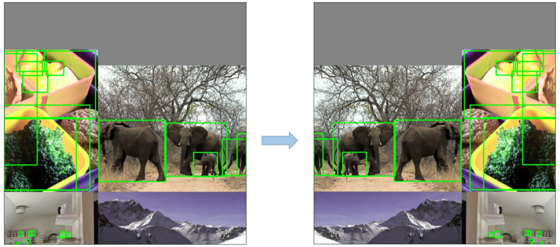
The architectures of YOLOv5s is based on Deep Convolutional Neural Network (DCNN).

Preprocessing

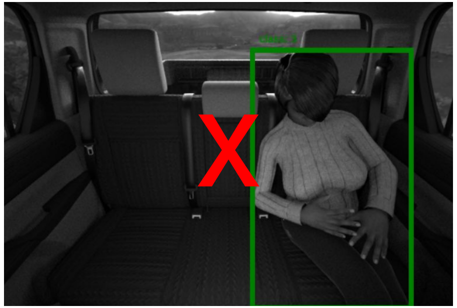
1. Labels with one row per object
2. Removing the coordinates of objects other than the child's seat
3. Converting labels to the following format: *class x_center y_center width height*
4. Normalized of bounding Box coordinates to xywh format (from 0 - 1)
5. Removal of images that doesn't have a child seat (only for train data)
6. Grayscale images for the training dataset
7. Data augmentation in the trainloader*
8. Normalize the images (0 - 1)*
9. Resize the images to 640x640*

** Inside Yolov5s model*

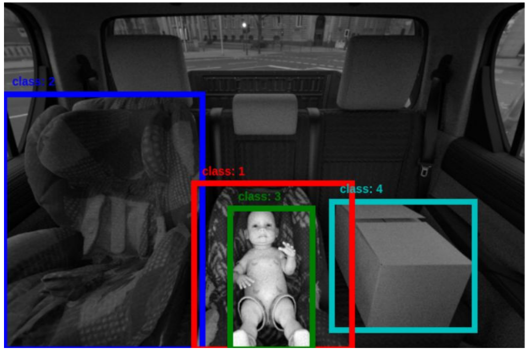
(7)



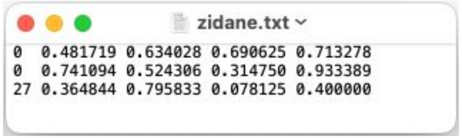
(5)



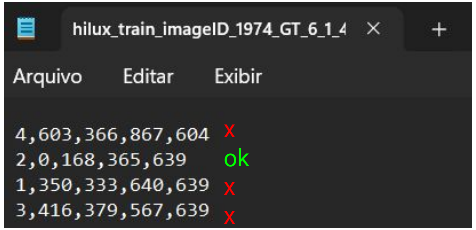
(1,2)



(3,4)


zidane.txt

```
0  0.481719 0.634028 0.690625 0.713278
0  0.741094 0.524306 0.314750 0.933389
27 0.364844 0.795833 0.078125 0.400000
```


hilux_train_imageID_1974_GT_6_1_4

Arquivo Editar Exibir

```
4,603,366,867,604 X
2,0,168,365,639   ok
1,350,333,640,639 X
3,416,379,567,639 X
```

Model Training

Overview of main parameters:

- Epochs: 400
- Batch-size: 128
- Optimizer: SGD
- Patience: 200
- Class: 0 (child seat)
- Initial learning rate: (1, 1e-5, 1e-1)
- Final One Cycle learning rate: (1, 0.01, 1.0)
- Weight decay: (1, 0.0, 0.001)
- Feed: 0

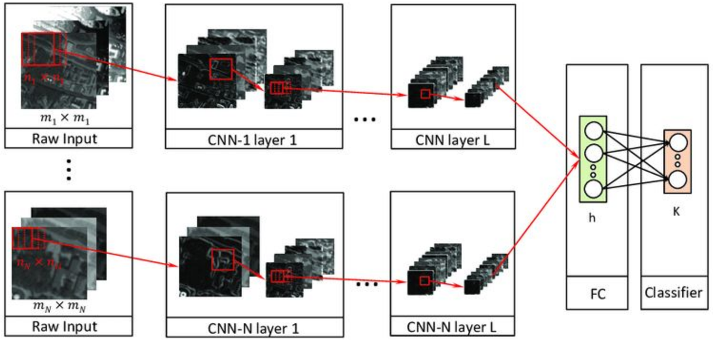
Fitness:

- mAP@0.5 contributes 10% of the weight
- mAP@0.5:0.95 contributes the remaining 90%

Main training techniques:

- Warmup and Cosine LR Scheduler
- Multiscale Training

CNN architecture used for multiscale feature learning.



(Längkvist, et al; 2020).

Model Evaluation

Model summary:

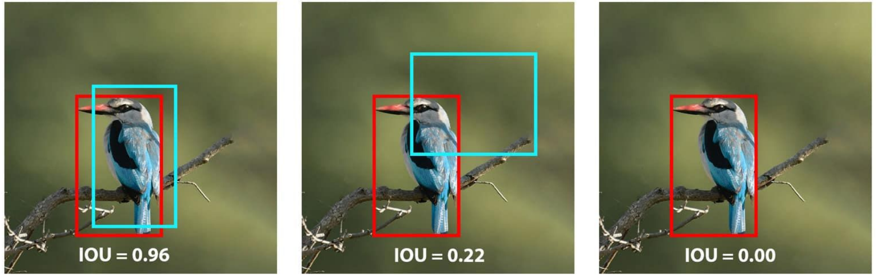
- 157 layers
- 7,012,822 parameters
- 0 gradients
- 15.8 GFLOPs

Main inference settings:

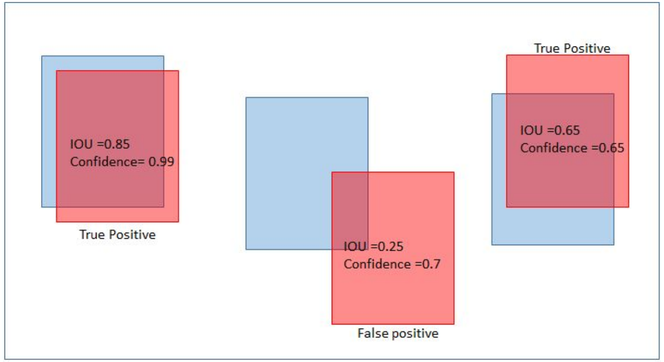
- IoU threshold: 0.45
- Confidence threshold: 0.25

Compute Losses:

- Objectness Loss (Obj Loss) - the confidence of object presence is the objectness loss
- Location Loss (Box Loss) - bounding box regression loss (Mean Squared Error)



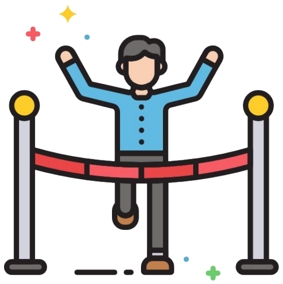
True Positive | False Positive | False Negative



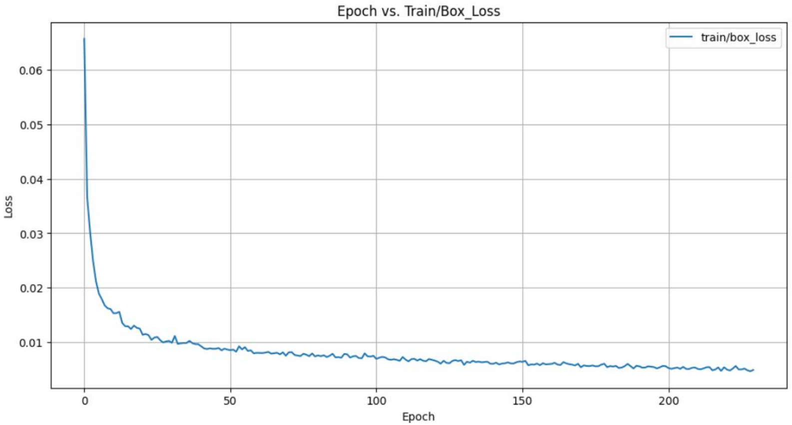
$$Precision = \frac{True\ Postive}{True\ Positives + False\ Positives} = \frac{count(True\ Positives)}{count(all\ red\ boxes)} = \frac{2}{3}$$

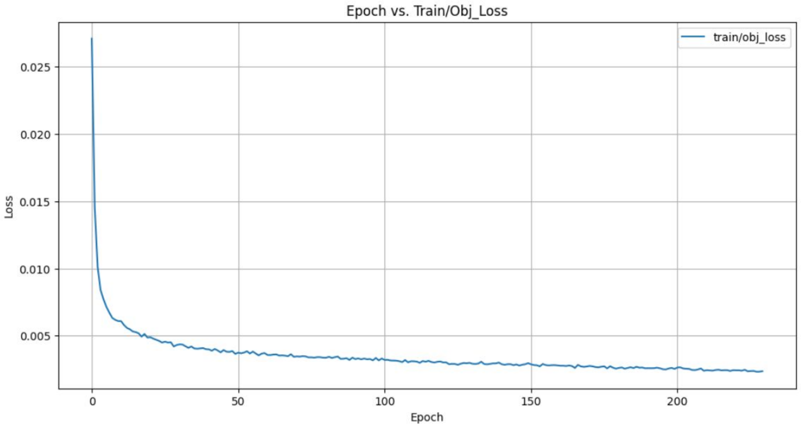
$$Recall = \frac{True\ Postive}{True\ Positives + False\ negatives} = \frac{count(True\ Positives)}{count(all\ blue\ boxes)} = \frac{2}{3}$$

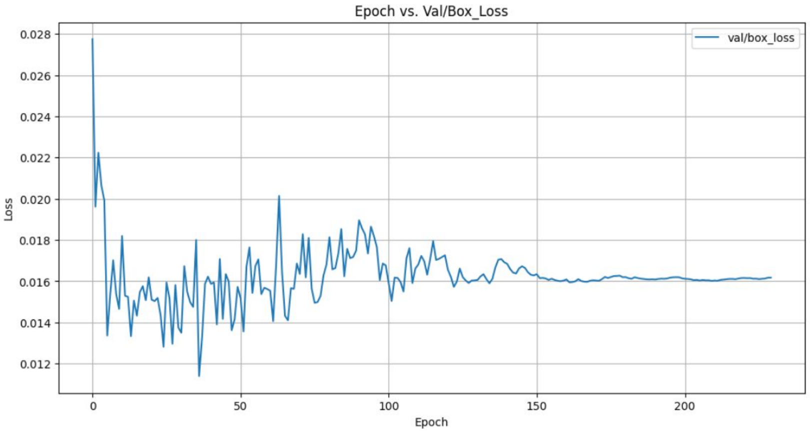
Results

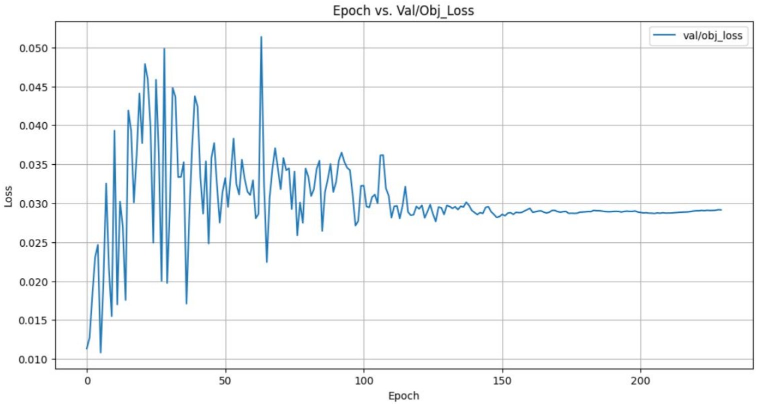


Train Results

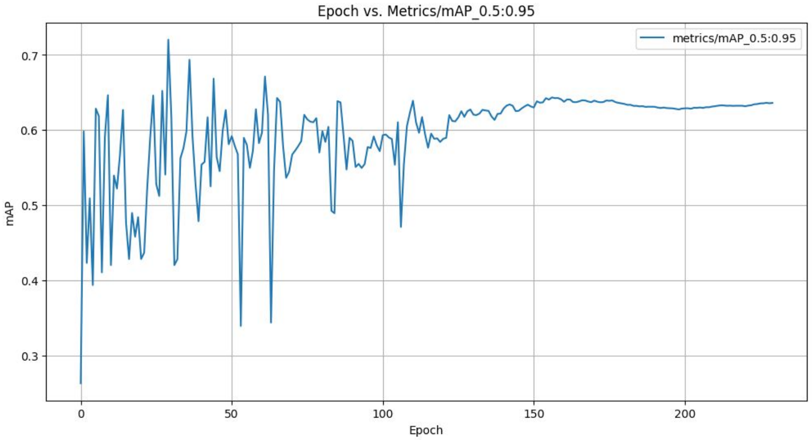


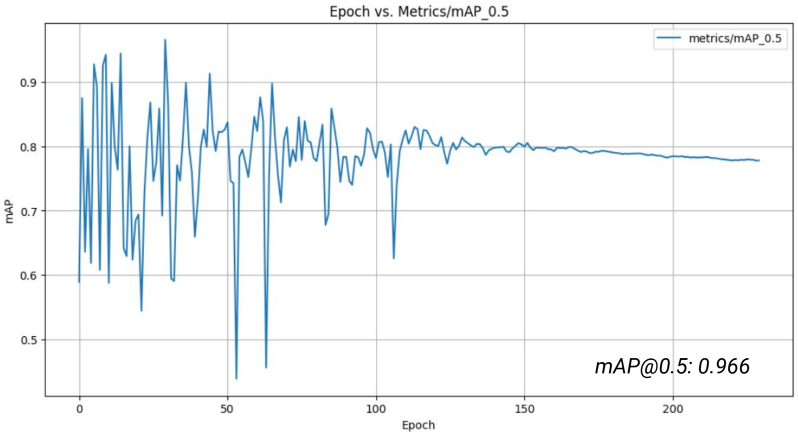


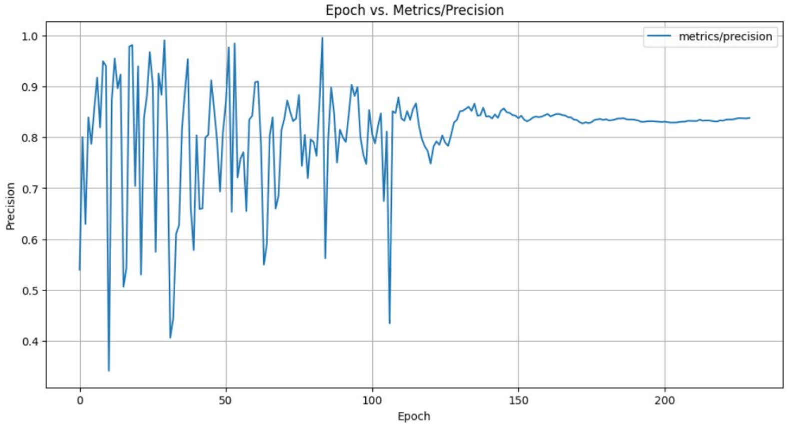


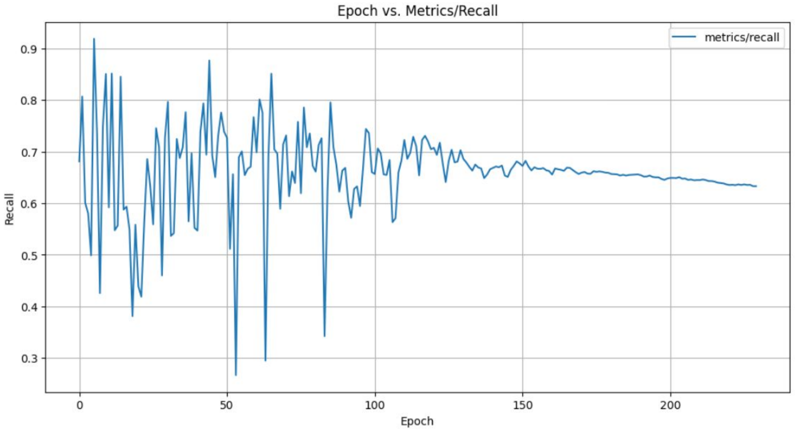


Train Results







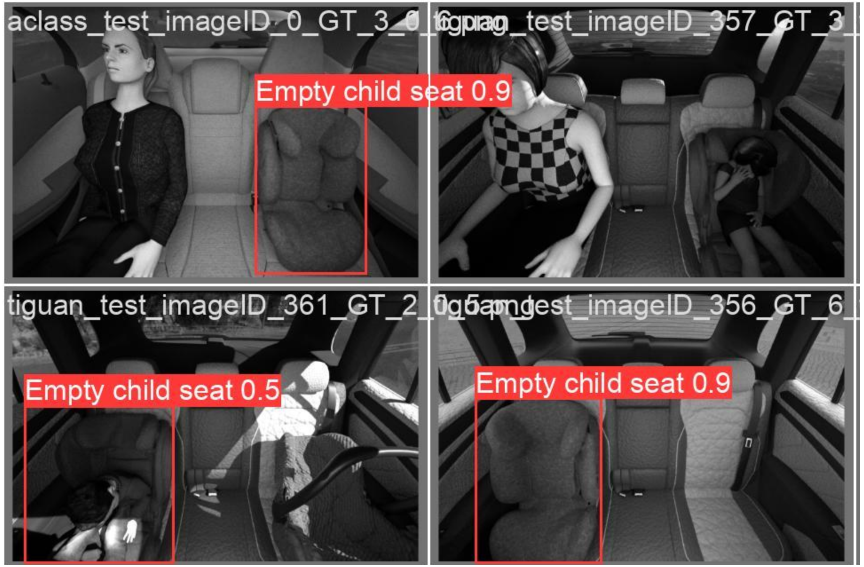


Test Results

- Test data: 5,000 images of SVIRO data set
- Speed: 24.8ms pre-process, 14.3ms inference, 1.8ms NMS per image at shape (1, 3, 448, 640)
- mAP@0.5: 0.874

Labels

Predicted



Technologies

python

YOLOv5

NumPy

OpenCV

matplotlib

PyTorch

pandas

References

XIAO, Youzi et al. A review of object detection based on deep learning. Multimedia Tools and Applications, v. 79, p. 23729-23791, 2020.

WU, Xiongwei; SAHOO, Doyen; HOI, Steven CH. Recent advances in deep learning for object detection. Neurocomputing, v. 396, p. 39-64, 2020.

LÄNGKVIST, Martin et al. Classification and segmentation of satellite orthoimagery using convolutional neural networks. Remote Sensing, v. 8, n. 4, p. 329, 2016.

Thank you!

Ilan Figueirêdo
PhD in Computational Modelling
ilan.figueiredo@gmail.com