Kissipo Learning for Deep Learning

Topic 20: Introduction to Object detection (20min)

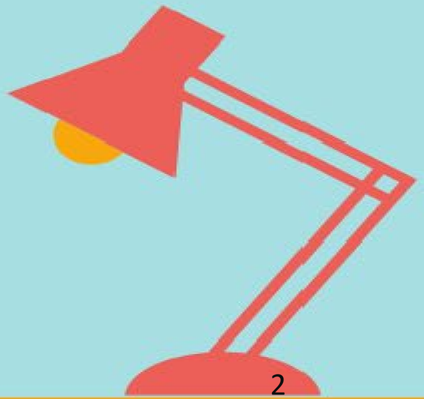
Hsueh-Ting Chu

KLDL-W8-T20

Course Schedule

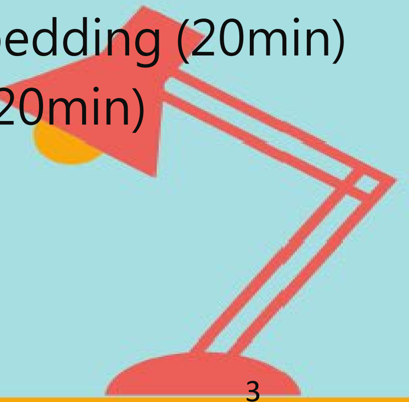
- W1 - Course Introduction
- W2 - DL Programming Basics(1)
- W3 - DL Programming Basics(2)
- W4 - DL with TensorFlow
- W5 - Midterm
- W6 - DL with PyTorch
- W7 - AOI hands-on project
- W8 - RSD hands-on project
- W9 - NLP hands-on project
- W10 - Final exam

DL: Deep Learning
AOI: Automated Optical Inspection
RSD: Road Sign Detection
NLP: Natural Language Processing



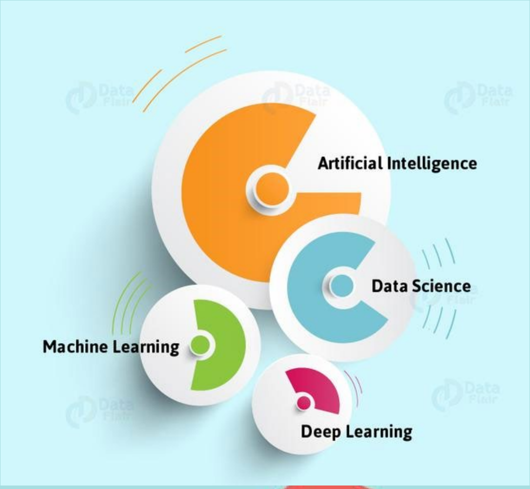
Topics

- Topic 01: Introduction to Deep Learning (20min)
- Topic 02: Kissipo Learning for Deep Learning (20min)
- Topic 03: Python quick tutorial (20min)
- Topic 04: Numpy quick tutorial (15min)
- Topic 05: Pandas quick tutorial (15min)
- Topic 06: Scikit-learn quick tutorial (15min)
- Topic 07: OpenCV quick tutorial (15min)
- Topic 08: Image Processing basics (20min)
- Topic 09: Machine Learning basics (20min)
- Topic 10: Deep Learning basics (20min)
- Topic 11: TensorFlow overview (20min)
- Topic 12: CNN with TensorFlow (20min)
- Topic 13: RNN with TensorFlow (20min)
- Topic 14: PyTorch overview (20min)
- Topic 15: CNN with PyTorch (20min)
- Topic 16: RNN with Pytorch (20min)
- Topic 17: Introduction to AOI (20min)
- Topic 18: AOI simple Pipeline (A) (20min)
- Topic 19: AOI simple Pipeline (B) (20min)
- Topic 20: Introduction to Object detection (20min)
- Topic 21: YoloV5 Quick Tutorial (20min)
- Topic 22: Using YoloV5 for RSD (20min)
- Topic 23: Introduction to NLP (20min)
- Topic 24: Introduction to Word Embedding (20min)
- Topic 25: Name prediction project (20min)

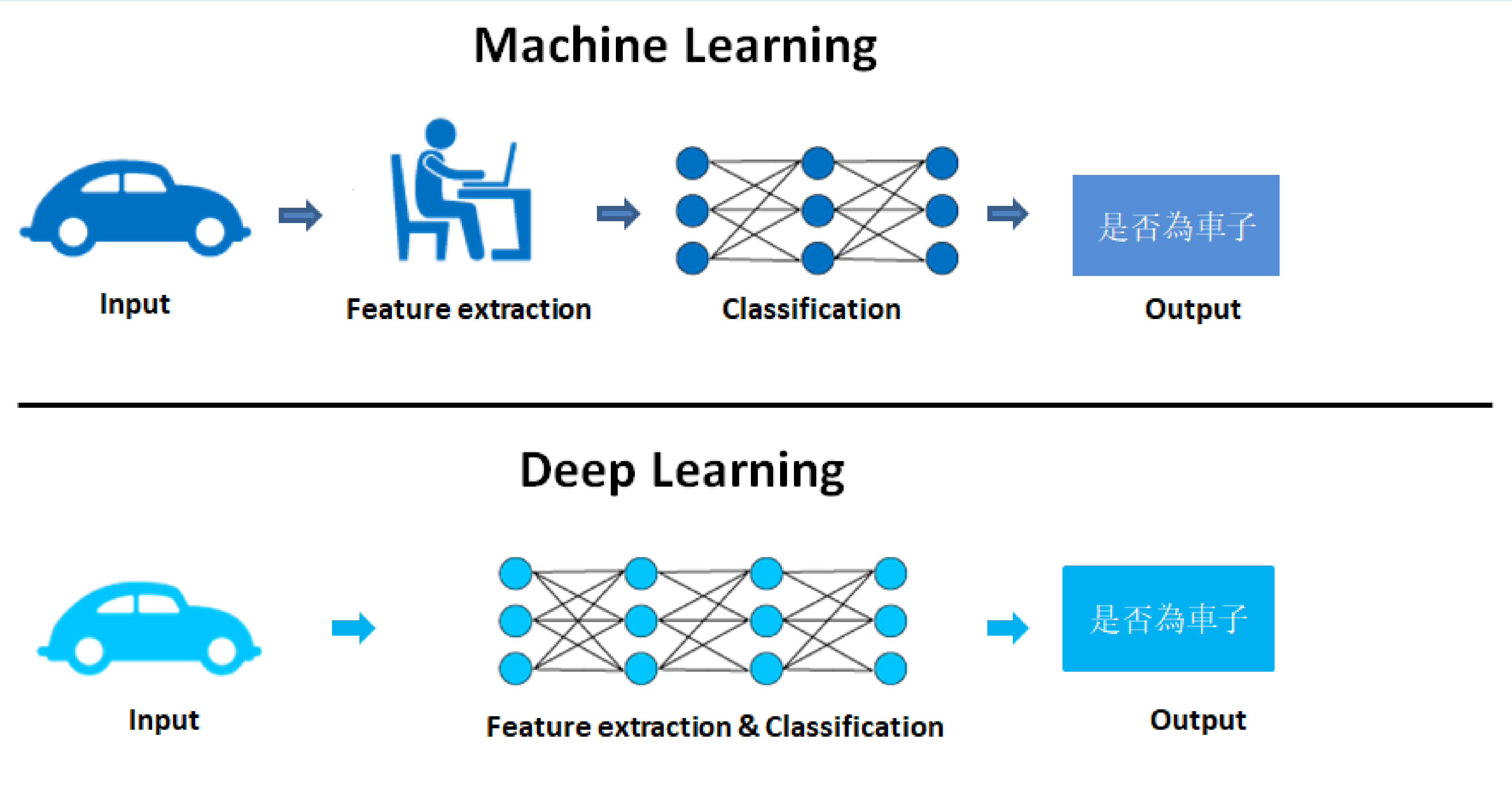


Week 8 Topics

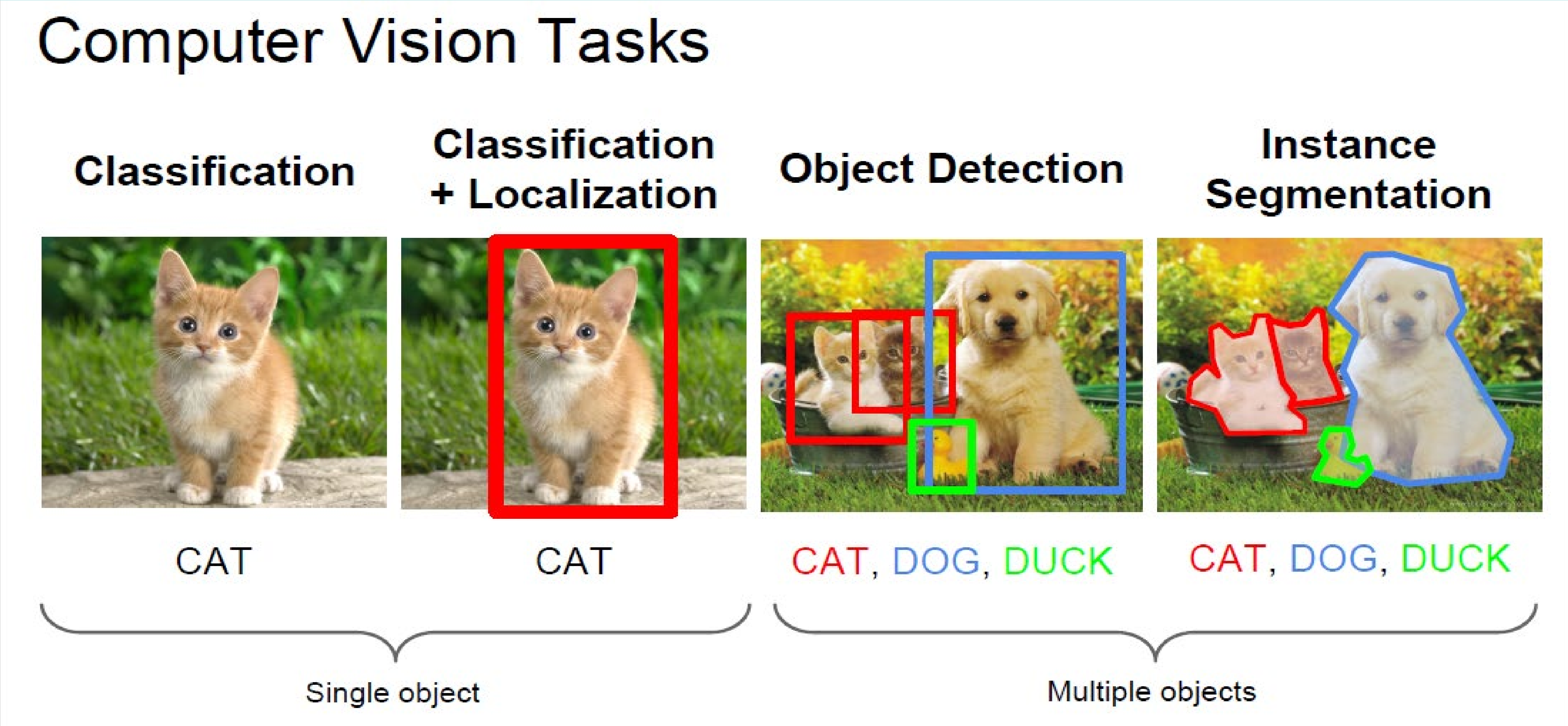
- Topic 20: Introduction to Object detection (20min)
- Topic 21: YoloV5 Quick Tutorial (20min)
- Topic 22: Using YoloV5 for RSD (20min)



Machine learning and Deep learning



CV jobs



Computer Vision Tasks

Classification

CAT

Classification + Localization

CAT

Object Detection

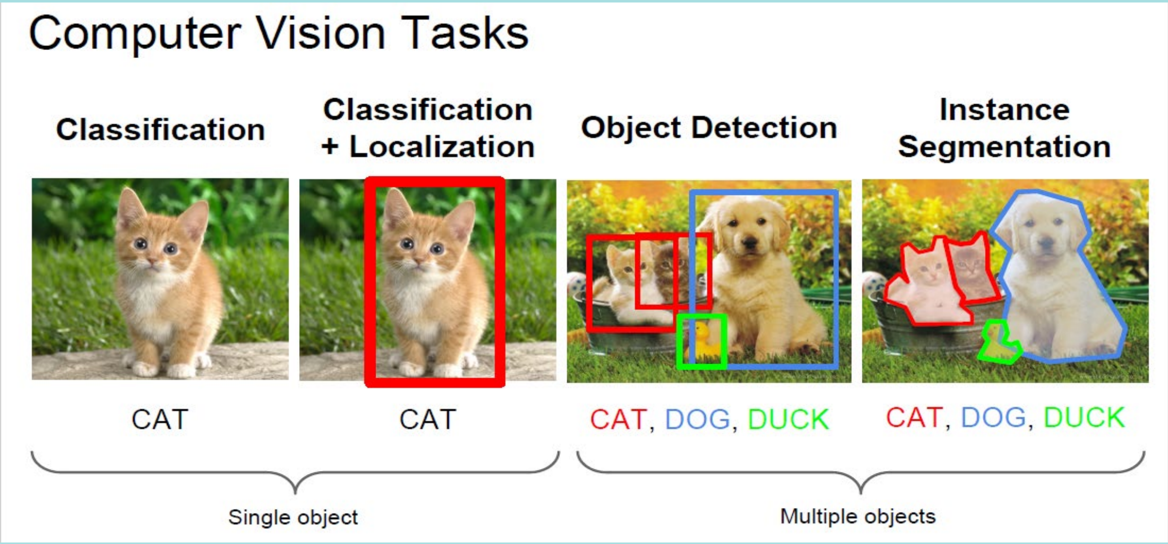
CAT, DOG, DUCK

Instance Segmentation

CAT, DOG, DUCK

Single object

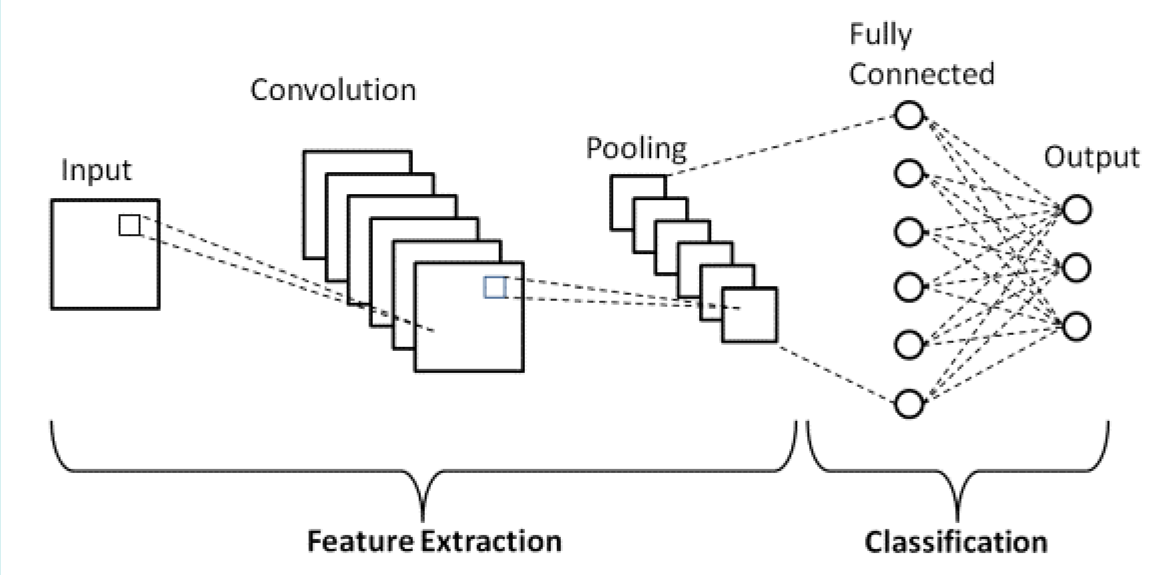
Multiple objects

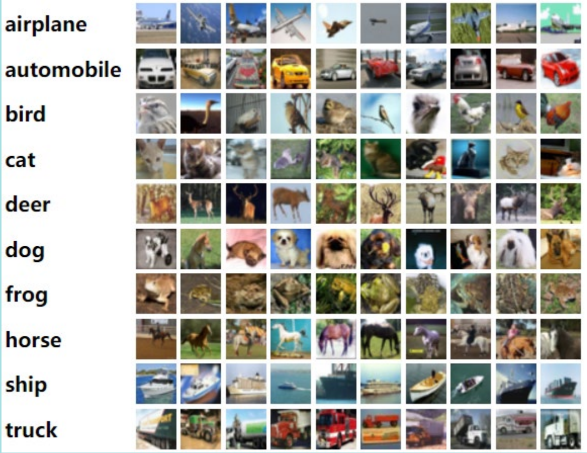


- Classification
- Positioning (Classification + Localization): mark the position and size of an object.
- Object Detection: Mark the location and size of multiple objects.
- Semantic Segmentation: Do not distinguish instances.
- Instance Segmentation: Mark instances.

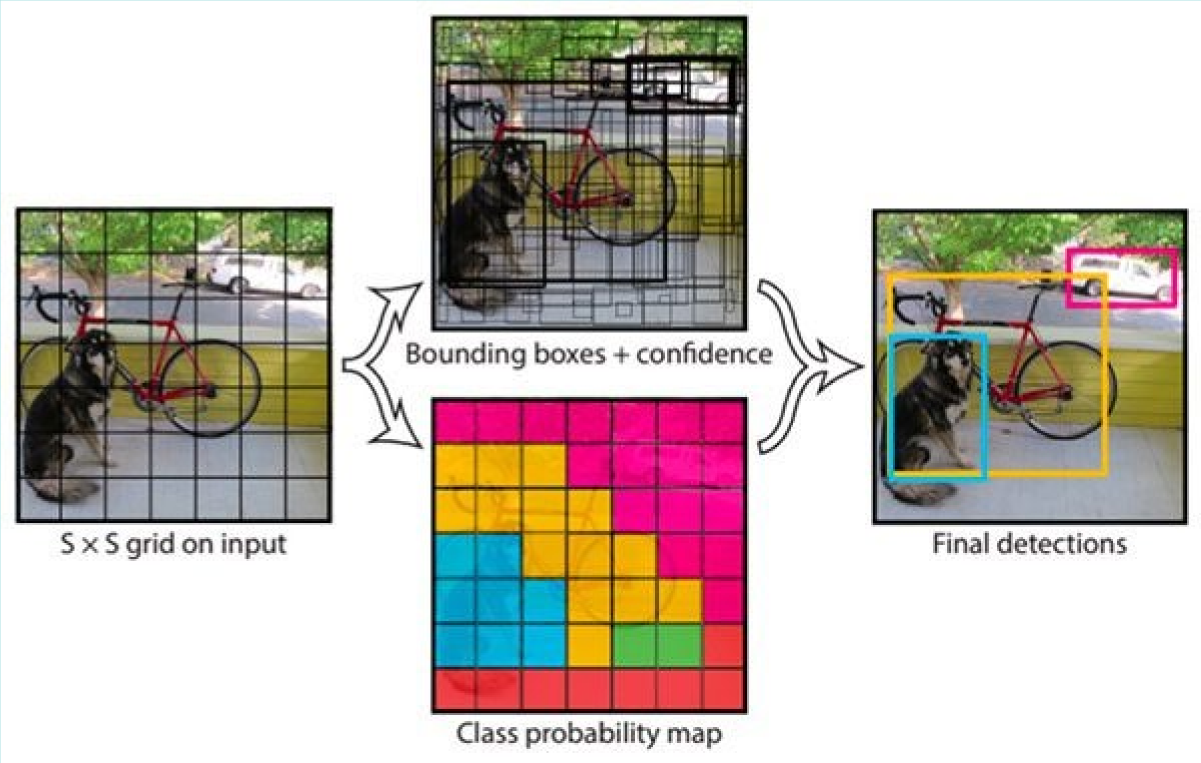
影像分類

Class 7 (horse)
0
5
10
15
20
25
30
0
5
10
15
20
25
30


Input
Convolution
Pooling
Fully Connected
Output
Feature Extraction
Classification

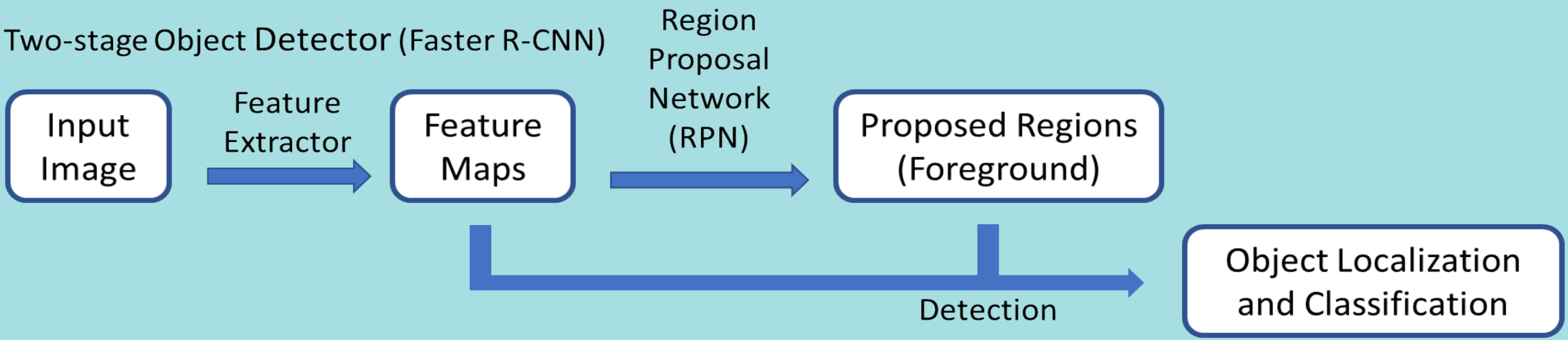

airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck

物件偵測


S × S grid on input
Bounding boxes + confidence
Class probability map
Final detections

One-stage object detection algorithm

Two-stage Object Detector (Faster R-CNN)



Input Image

Feature Extractor

Feature Maps

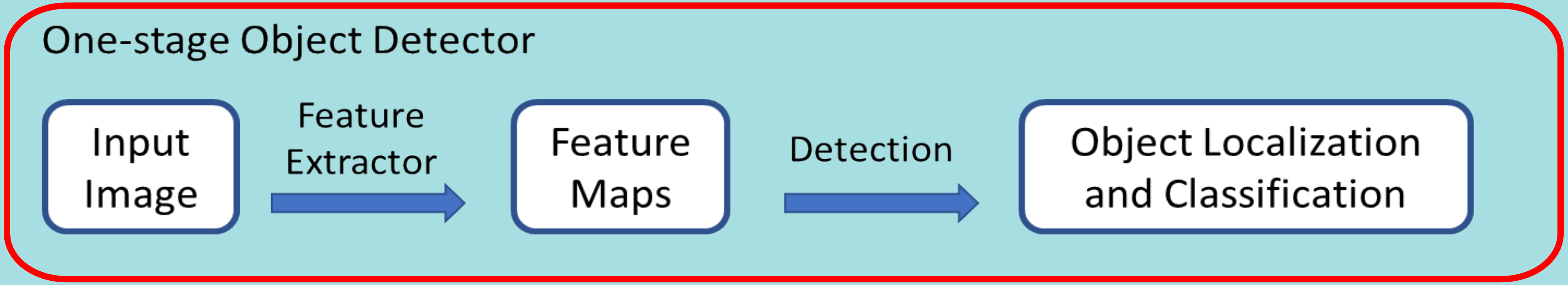
Region Proposal Network (RPN)

Proposed Regions (Foreground)

Detection

Object Localization and Classification

One-stage Object Detector



Input Image

Feature Extractor

Feature Maps

Detection

Object Localization and Classification

You Only Look Once (YOLO)

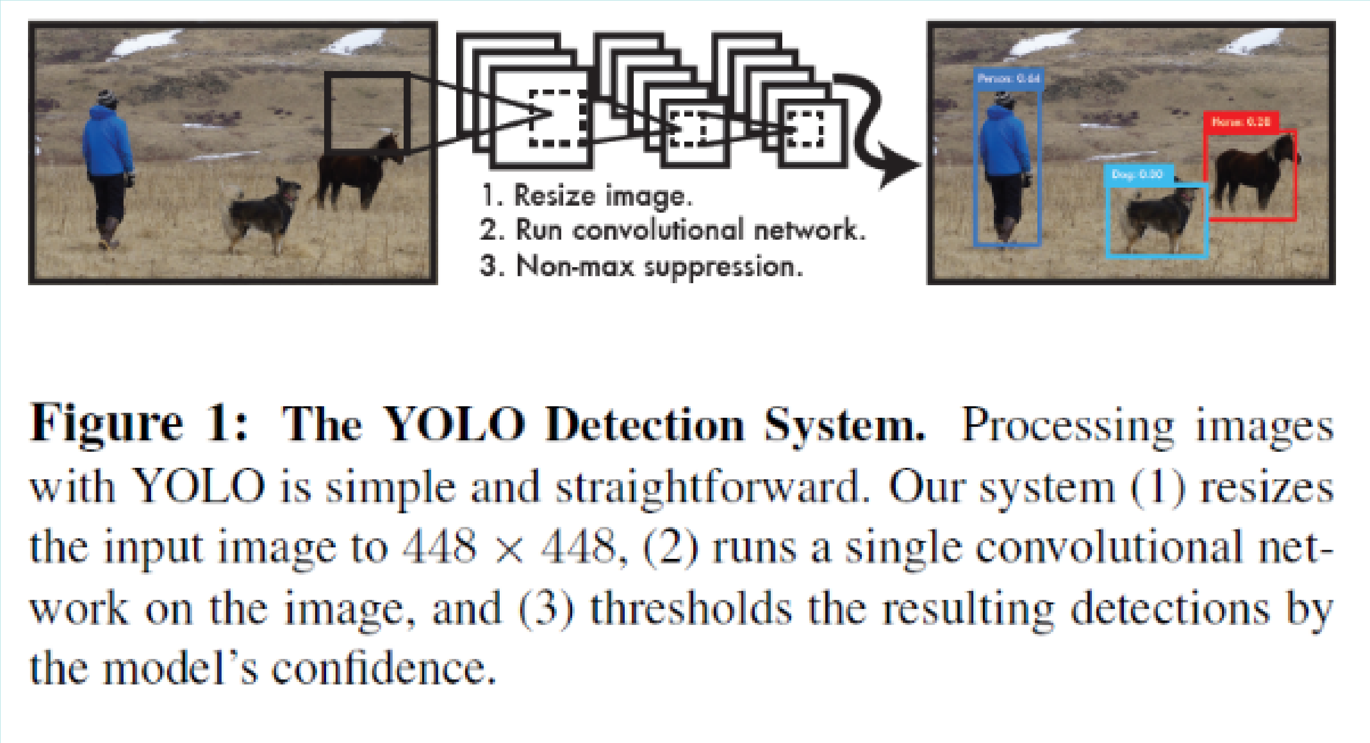
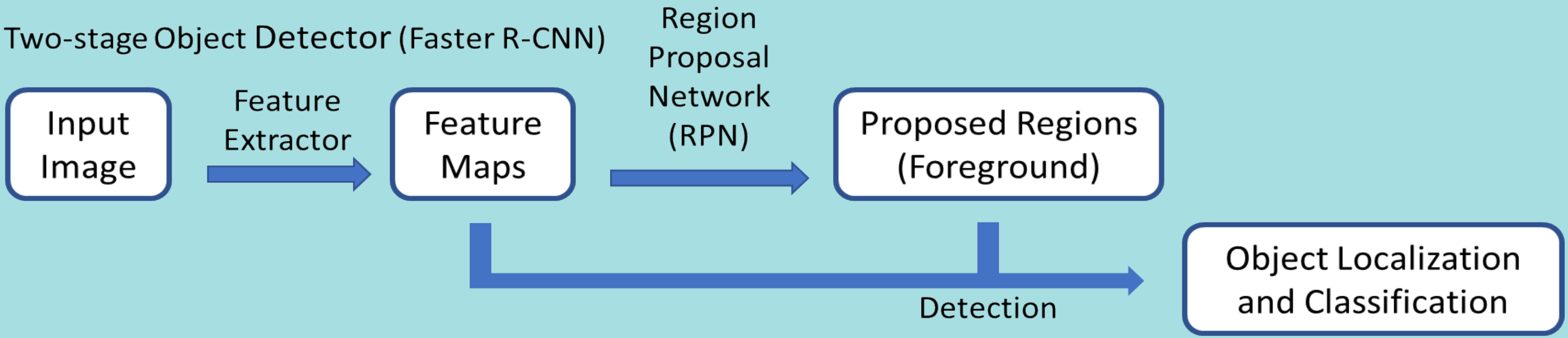


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

In the part of object detection, YOLO basically divides the graph into many grid cells, and then predicts the probability of 2 bounding boxes and which category it belongs to in each grid cell, and finally uses the threshold and NMS (Non-Maximum Suppression) way to get the result.

Main types of object detection algorithm

Two-stage Object Detector (Faster R-CNN)



Input Image

Feature Extractor

Feature Maps

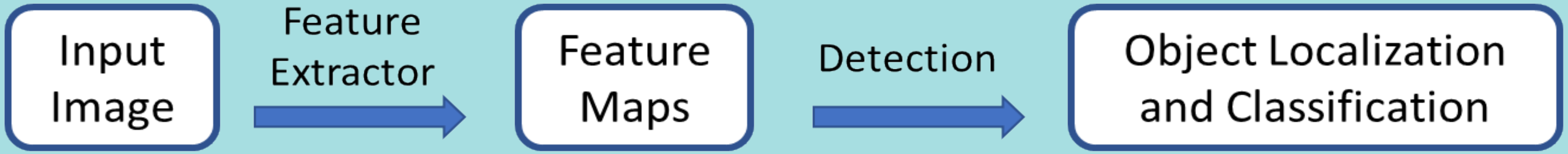
Region Proposal Network (RPN)

Proposed Regions (Foreground)

Detection

Object Localization and Classification

One-stage Object Detector



Input Image

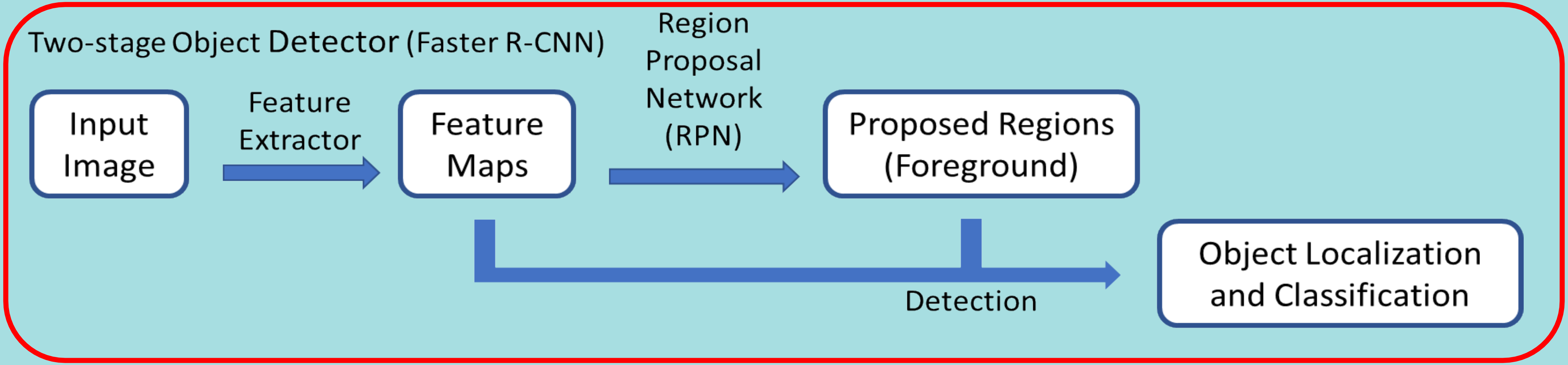
Feature Extractor

Feature Maps

Detection

Object Localization and Classification

Two-stage object detection algorithm



Two-stage Object Detector (Faster R-CNN)

Input Image

Feature Extractor

Feature Maps

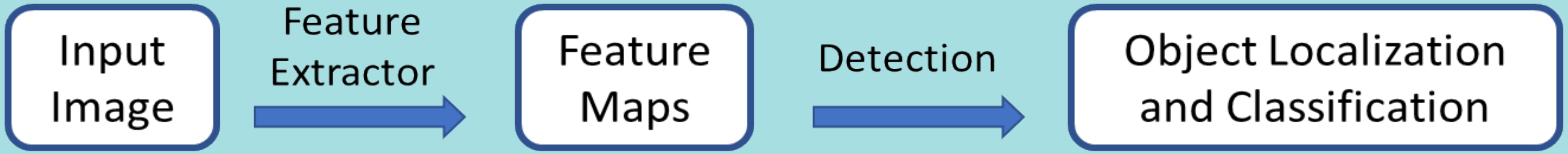
Region Proposal Network (RPN)

Proposed Regions (Foreground)

Detection

Object Localization and Classification

One-stage Object Detector



Input Image

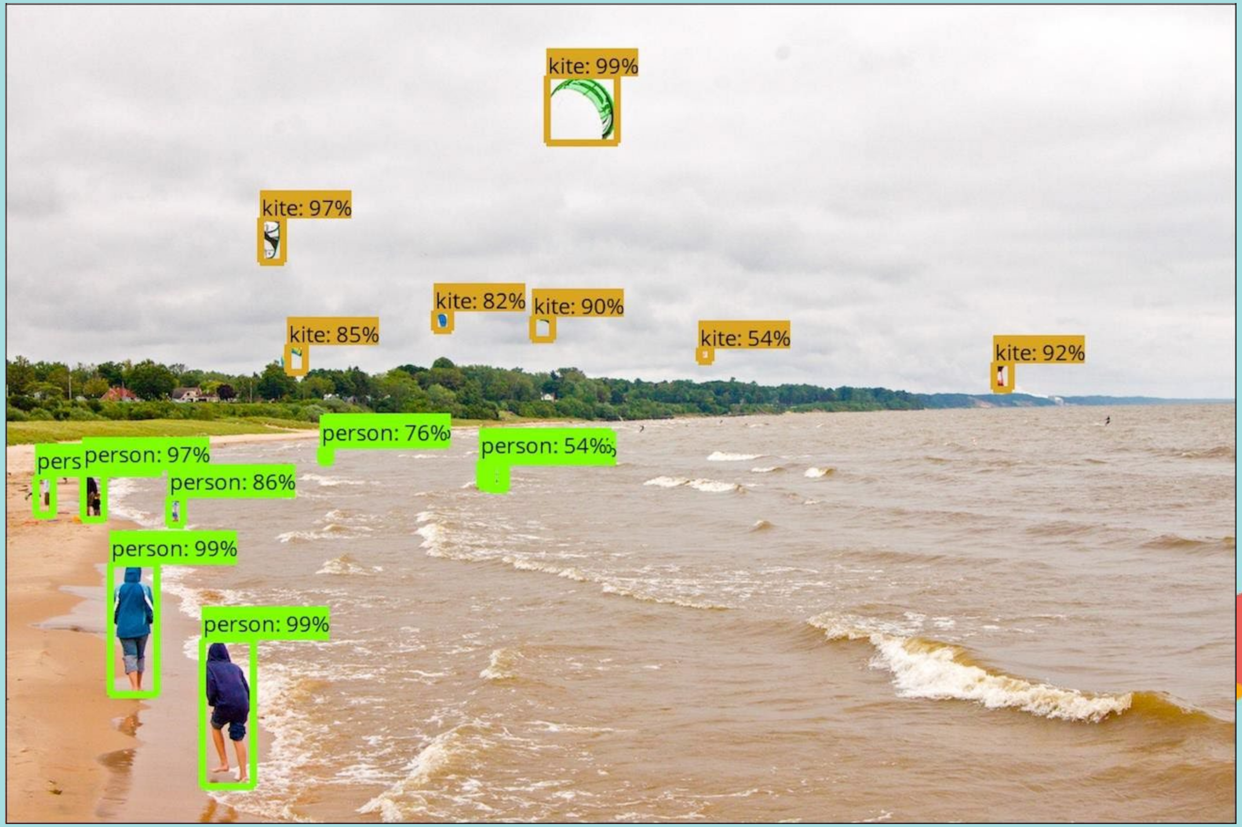
Feature Extractor

Feature Maps

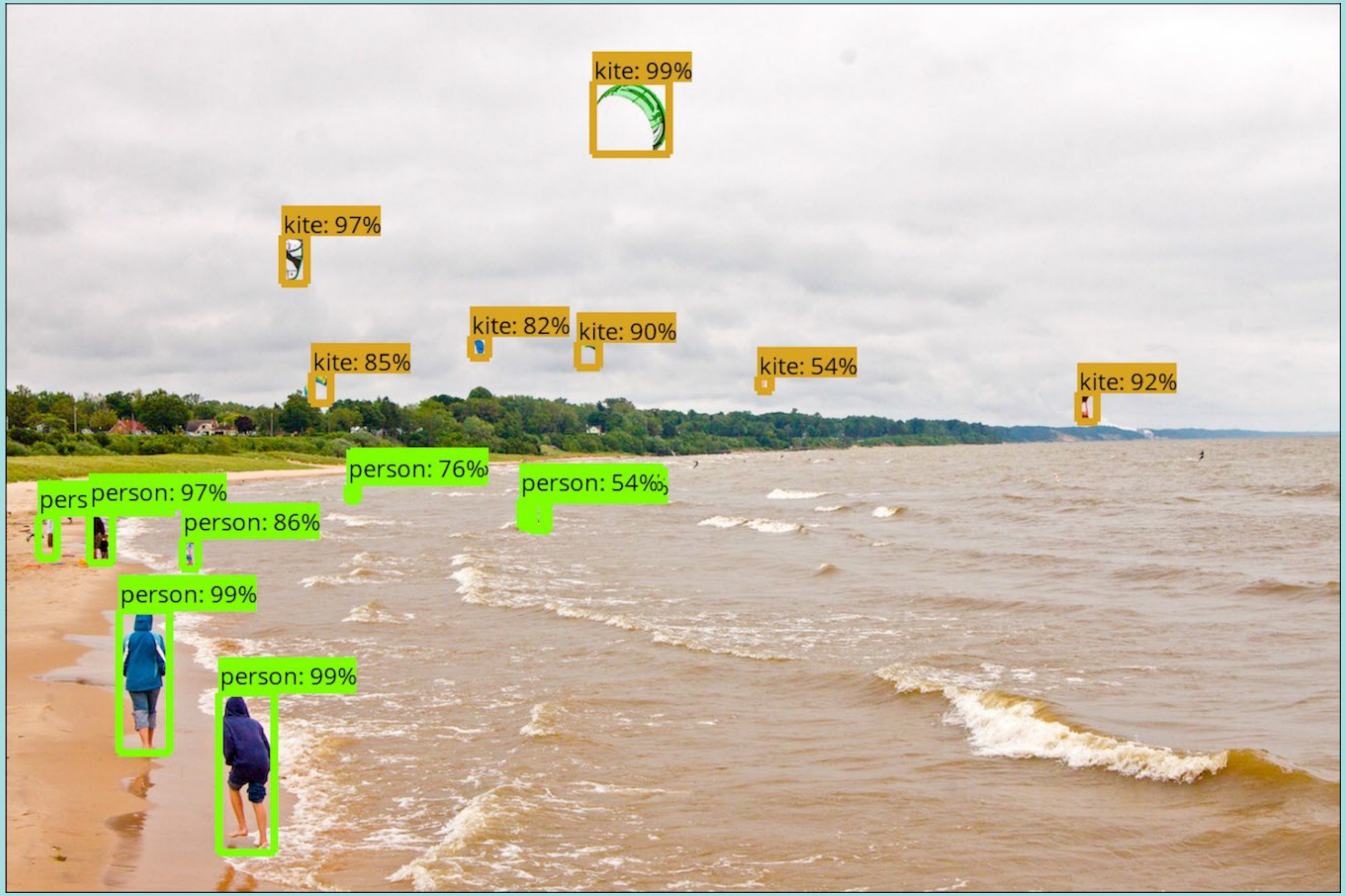
Detection

Object Localization and Classification

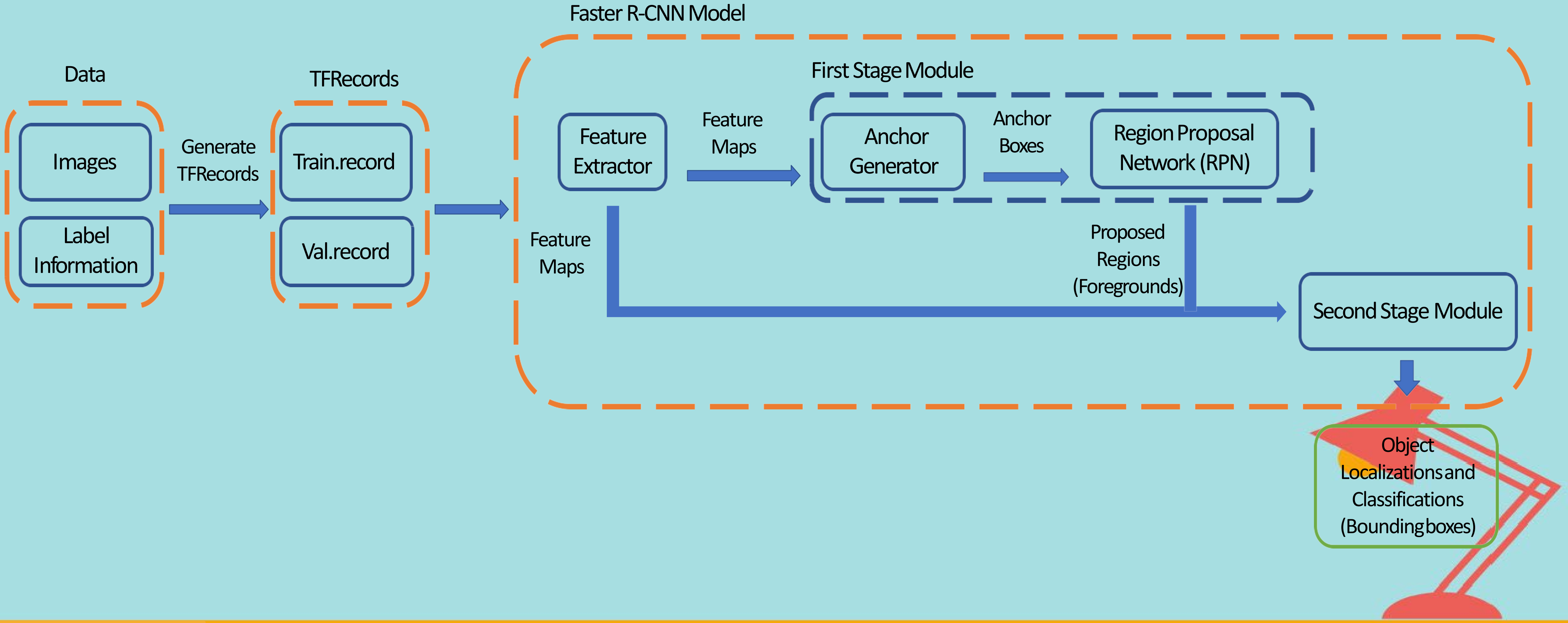
TensorFlow Object Detection API
- Faster R-CNN



TensorFlow Object Detection API



Faster R-CNN Model



Faster R-CNN Model

Data

Images

Label Information

Generate TFRecords

TFRecords

Train.record

Val.record

Feature Extractor

Feature Maps

First Stage Module

Anchor Generator

Anchor Boxes

Region Proposal Network (RPN)

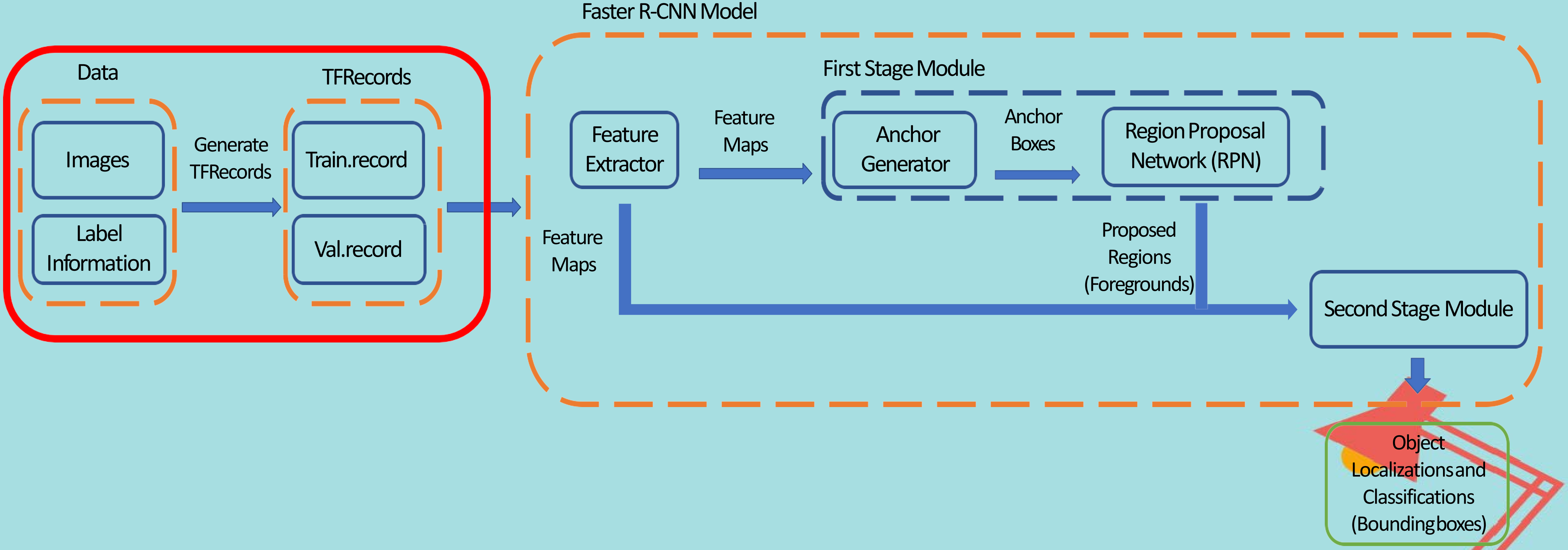
Feature Maps

Proposed Regions (Foregrounds)

Second Stage Module

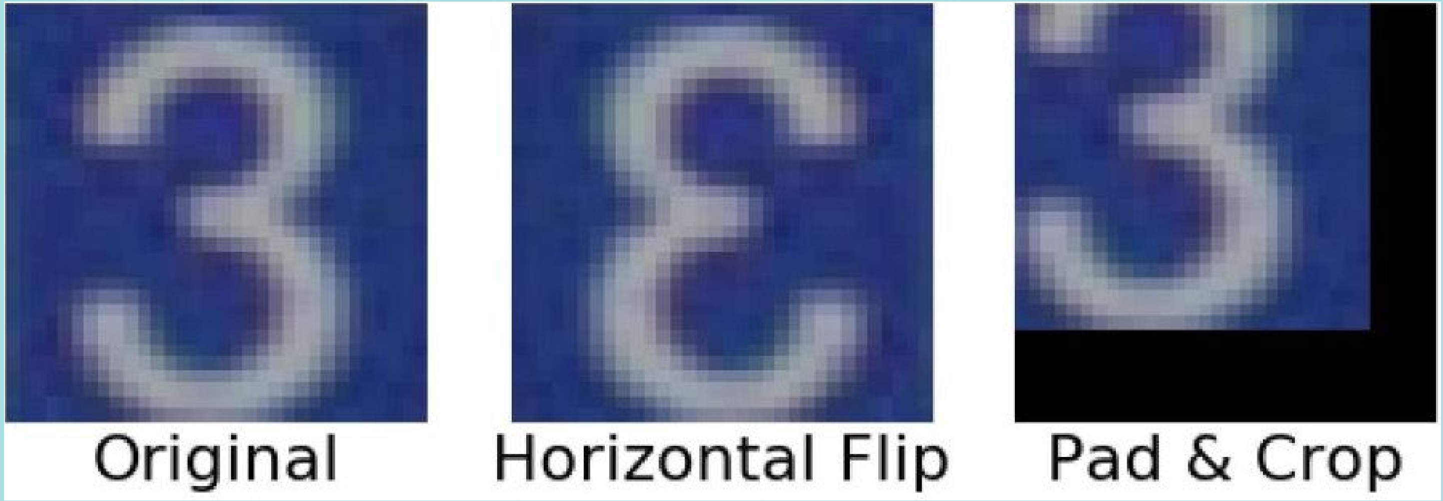
Object Localizations and Classifications (Bounding boxes)

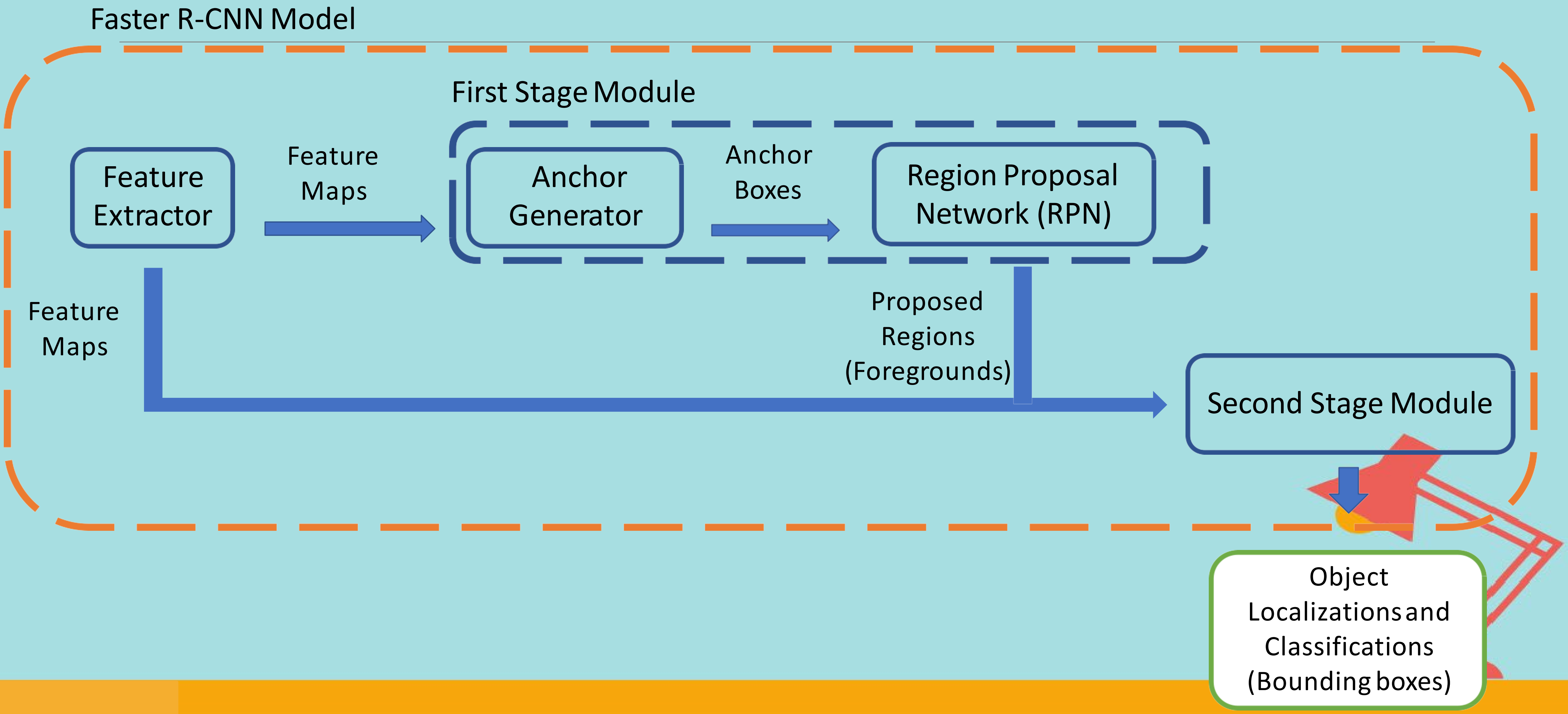
Faster R-CNN Model



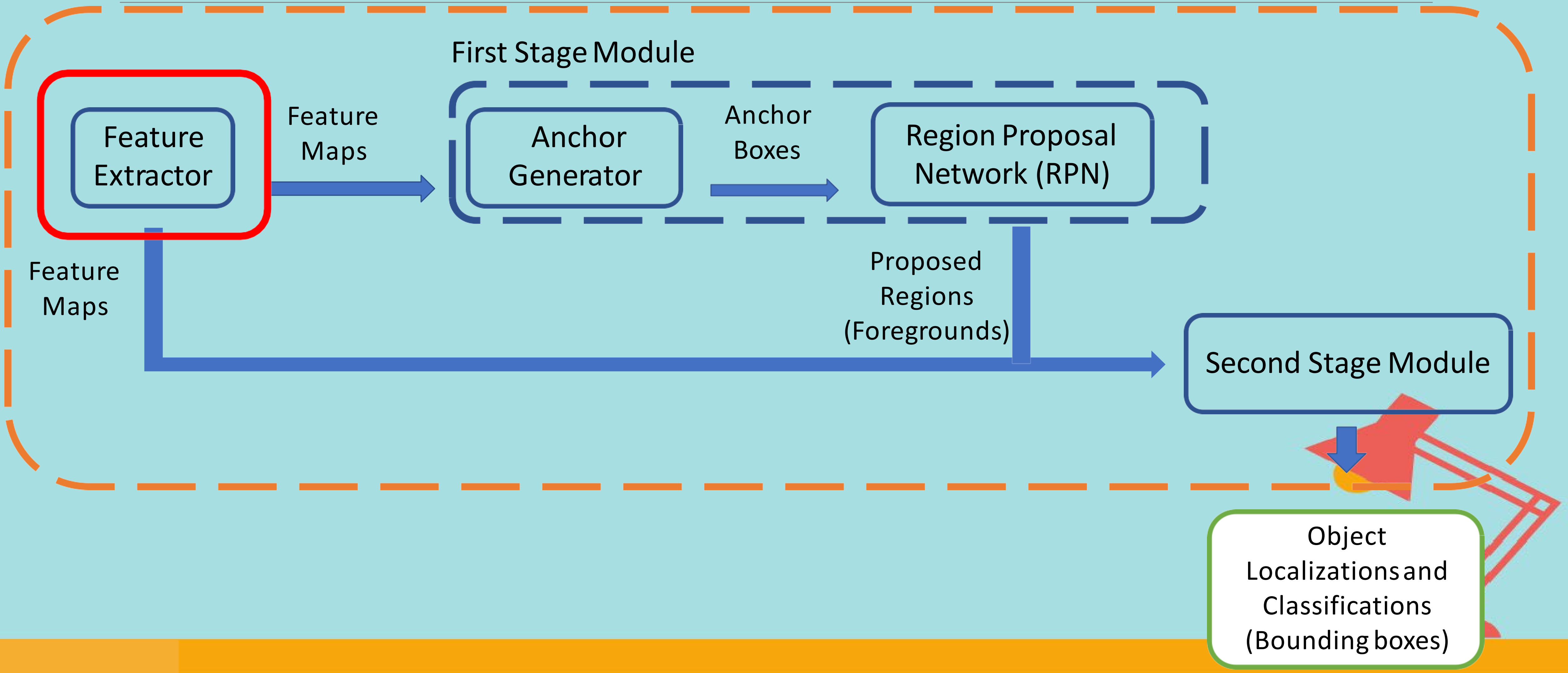
Data Augmentation

- Horizontal flip
- Image scale
- Crop image
- Pad image




Faster R-CNN Model
First Stage Module
Feature Extractor
Feature Maps
Anchor Generator
Anchor Boxes
Region Proposal Network (RPN)
Feature Maps
Proposed Regions (Foregrounds)
Second Stage Module
Object Localizations and Classifications (Bounding boxes)

Faster R-CNN Model



Feature Extractor

Feature Maps

First Stage Module

Anchor Generator

Anchor Boxes

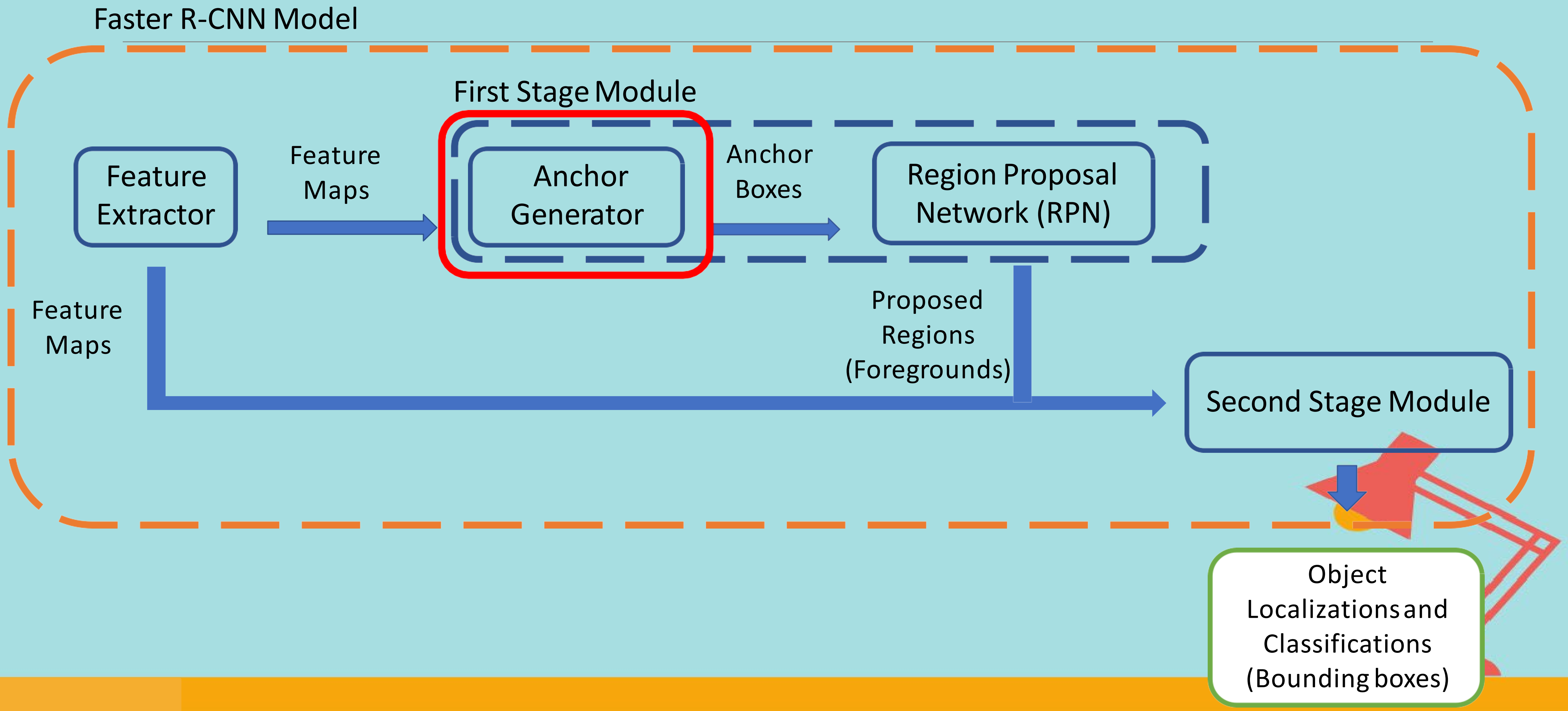
Region Proposal Network (RPN)

Feature Maps

Proposed Regions (Foregrounds)

Second Stage Module

Object Localizations and Classifications (Bounding boxes)



Faster R-CNN Model

First Stage Module

Feature Extractor

Feature Maps

Anchor Generator

Anchor Boxes

Region Proposal Network (RPN)

Feature Maps

Proposed Regions (Foregrounds)

Second Stage Module

Object Localizations and Classifications (Bounding boxes)

First Stage Module - AnchorGenerator

- Hyperparameter: **scale** and **aspect ratio**

Image size = 800x600

height = base anchor size* scale / √(aspect ratio)

weight = base anchor size*scale*√(aspect ratio)

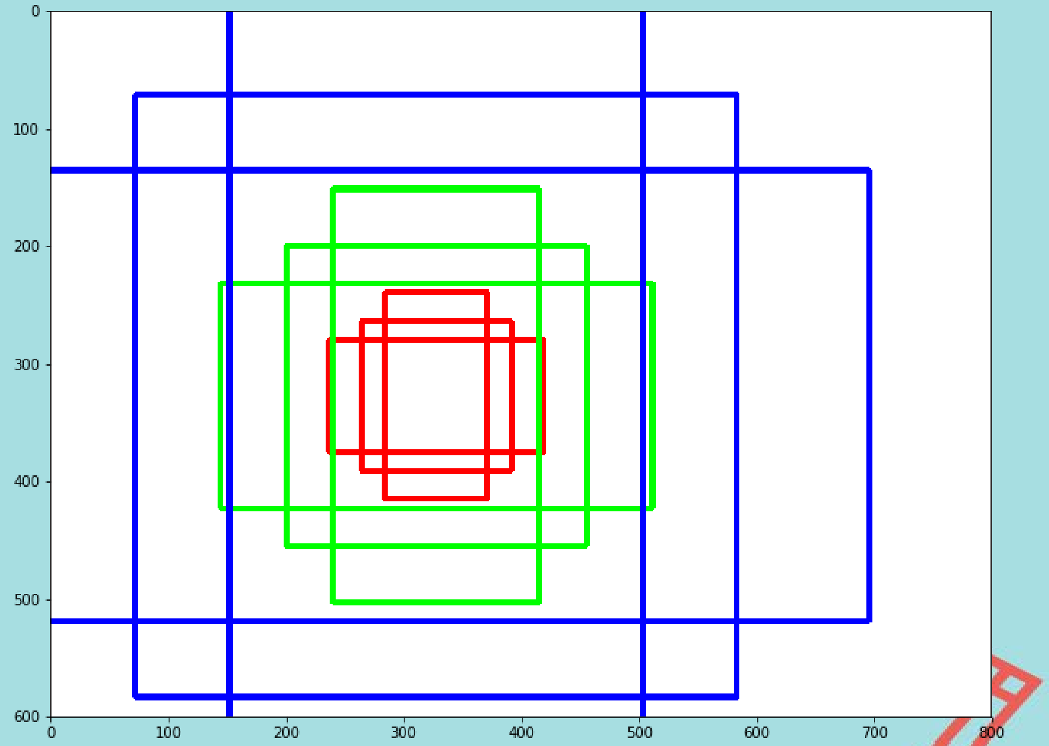
base anchor size = 256

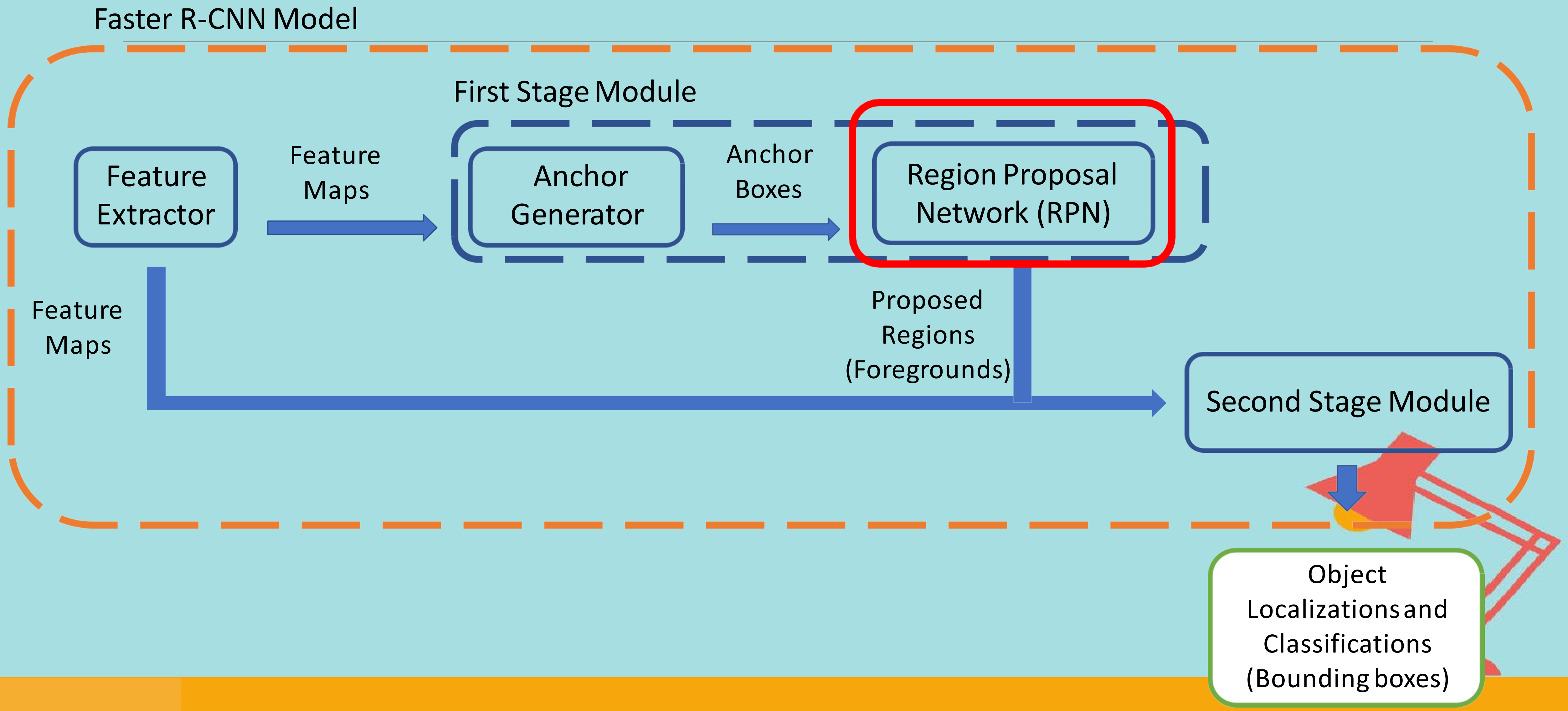
The maximal square anchor box is

256(base anchor size)*2(scale)*1(aspect ratio)=512

The minimal square anchor box is

256(base anchor size)*0.125(scale)*1(aspect ratio)=32.





Faster R-CNN Model

First Stage Module

Feature Extractor

Feature Maps

Anchor Generator

Anchor Boxes

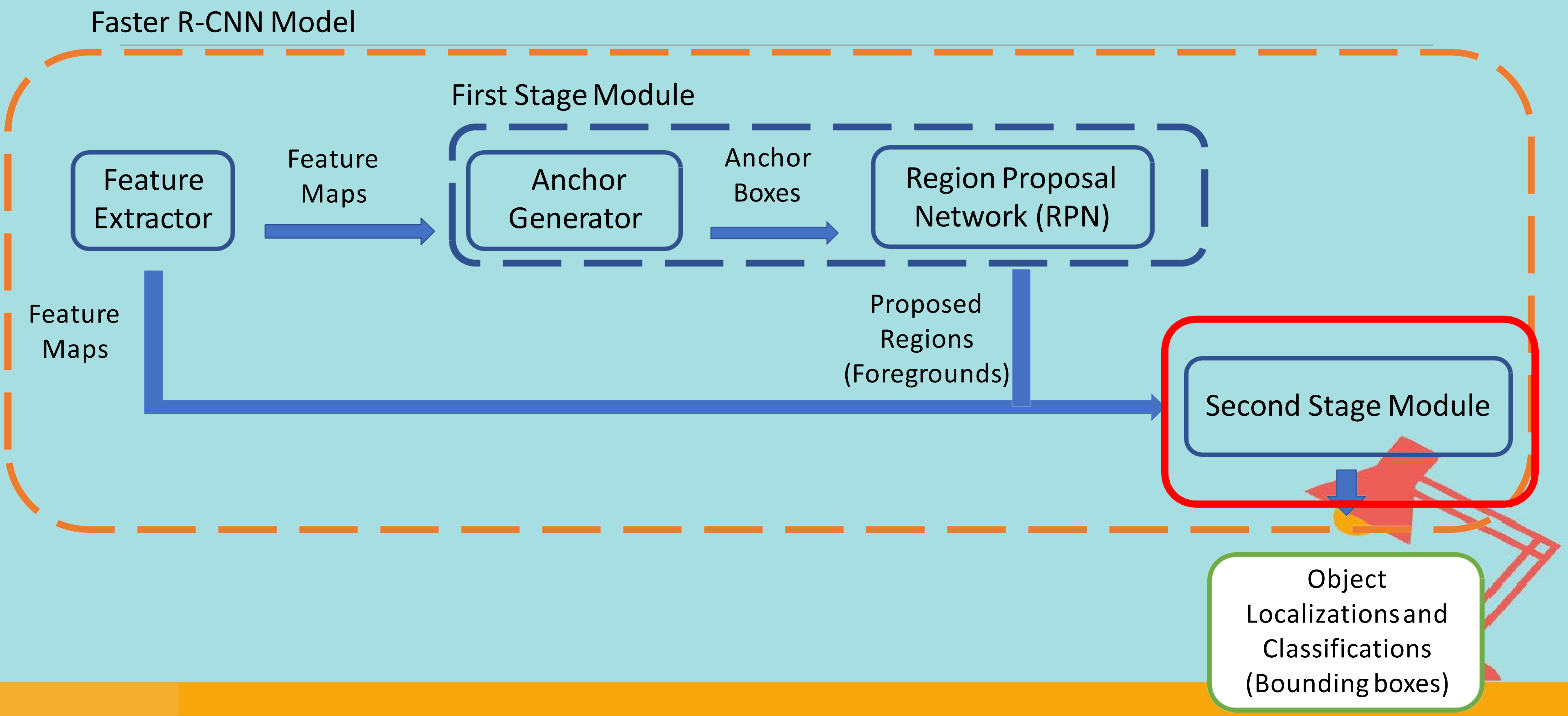
Region Proposal Network (RPN)

Feature Maps

Proposed Regions (Foregrounds)

Second Stage Module

Object Localizations and Classifications (Bounding boxes)



Faster R-CNN Model

First Stage Module

Feature Extractor

Feature Maps

Anchor Generator

Anchor Boxes

Region Proposal Network (RPN)

Feature Maps

Proposed Regions (Foregrounds)

Second Stage Module

Object Localizations and Classifications (Bounding boxes)

EfficientDet物件偵測模型

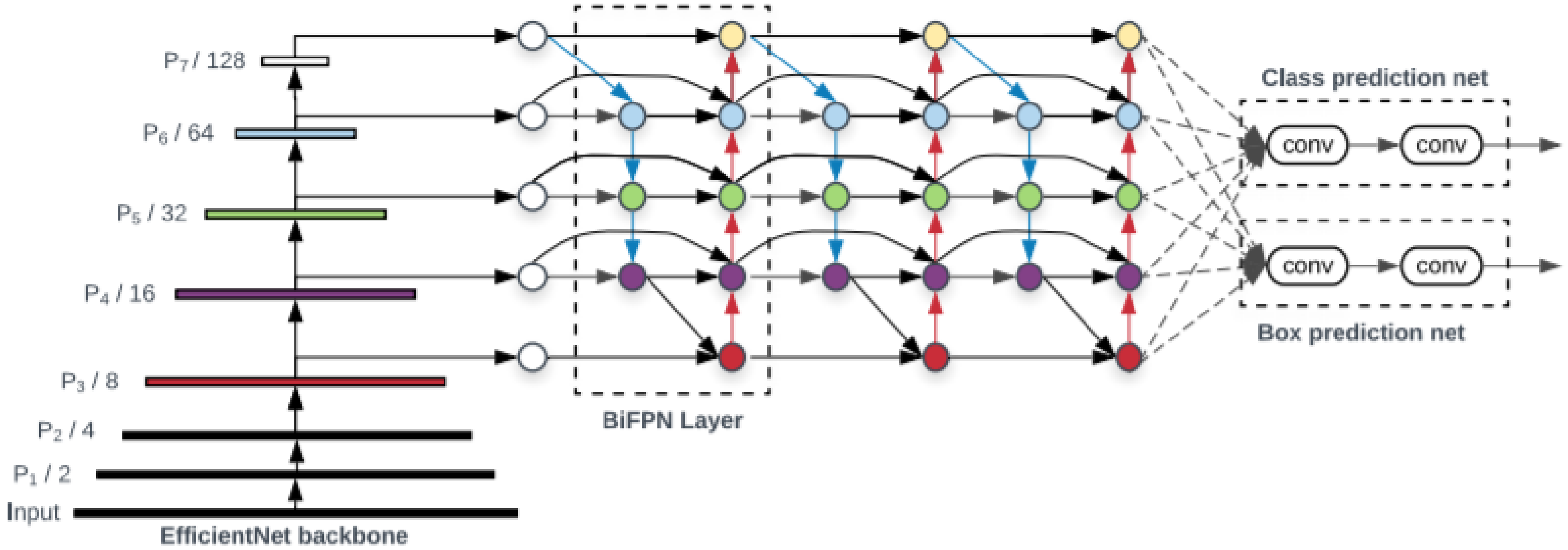


Figure 3: **EfficientDet architecture** – It employs EfficientNet [39] as the backbone network, BiFPN as the feature network, and shared class/box prediction network. Both BiFPN layers and class/box net layers are repeated multiple times based on different resource constraints as shown in Table 1.

Thanks!

Q&A