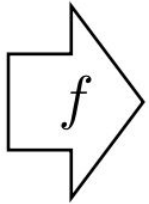
EECS 442 Discussion: PS7

Object Detection

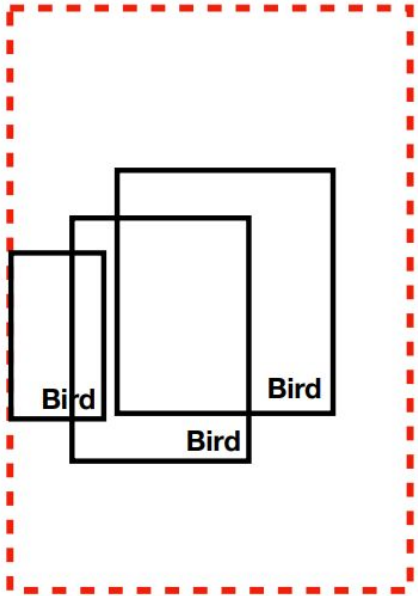
Outline

- Object Detection
- PASCAL VOC 2007 Dataset
- Detector Backbone Network
- Proposal Generation
- Prediction Network
- Intersection Over Union (IOU)
- Non-Maximum Suppression (NMS)

Object detection



Classification and localization



Each bounding box is:
[x,y,w,h]

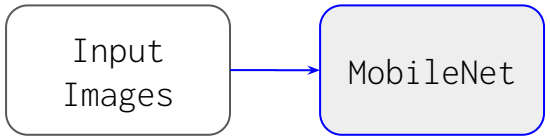
Comprehensive Overview of Single Stage Detector

We'll walk through the detector diagram in these slides.

PASCAL VOC 2007 Dataset

Object detection dataset with ground truth bounding box labels

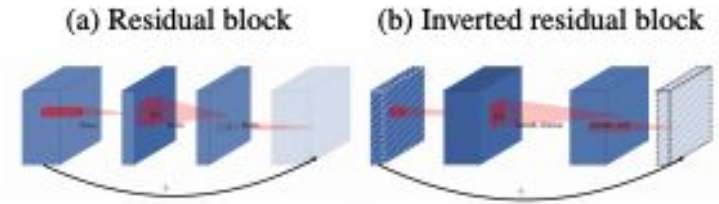
Adding the Feature Extractor



Detector Backbone Network

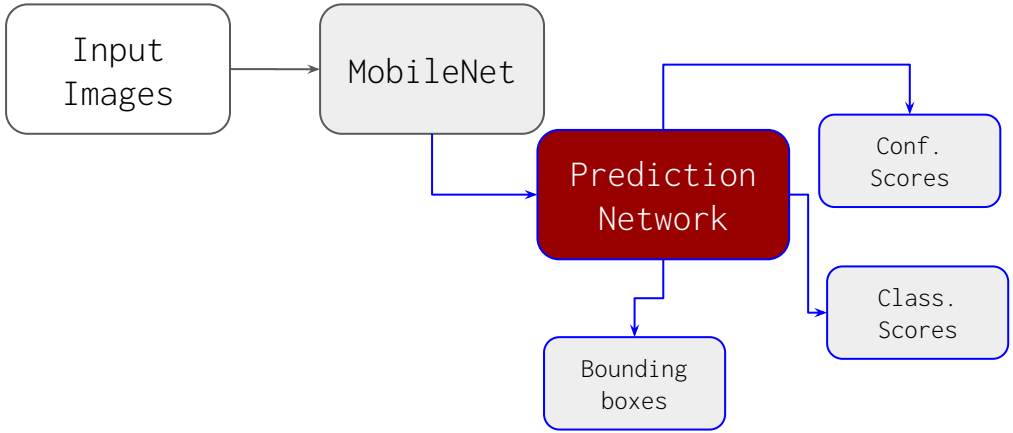
- You can treat this as a black box for your purposes. It simply extracts features from the input images.

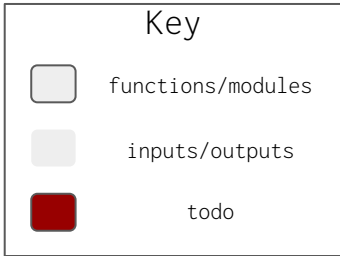
[MobileNet v2]



Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Adding the Prediction Network



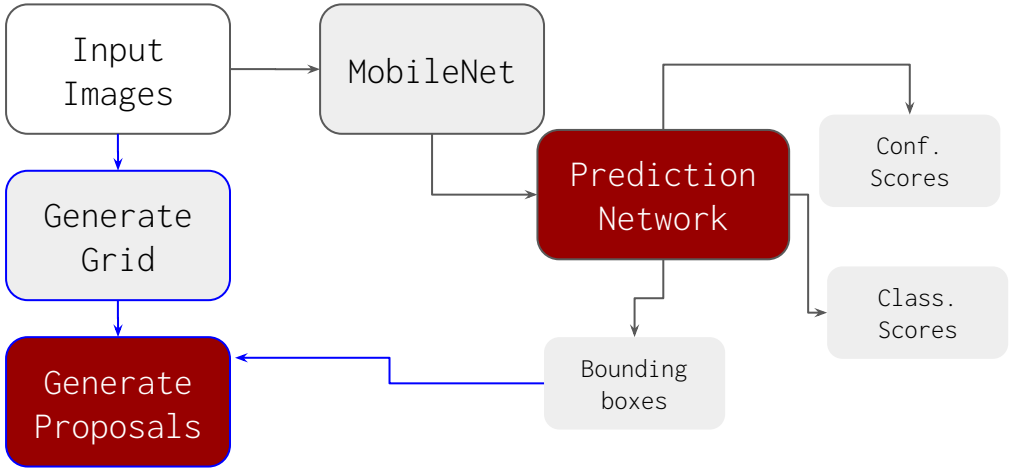


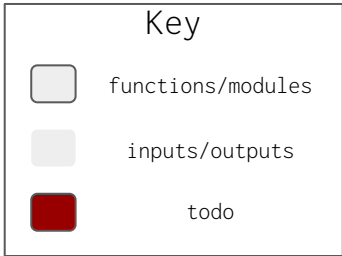
Prediction Network

- Given features from the backbone network, it outputs the classification scores and offsets for each bounding box.

- We have implemented the init function for you in PS7, use it to implement the forward pass

Adding Proposal Generation

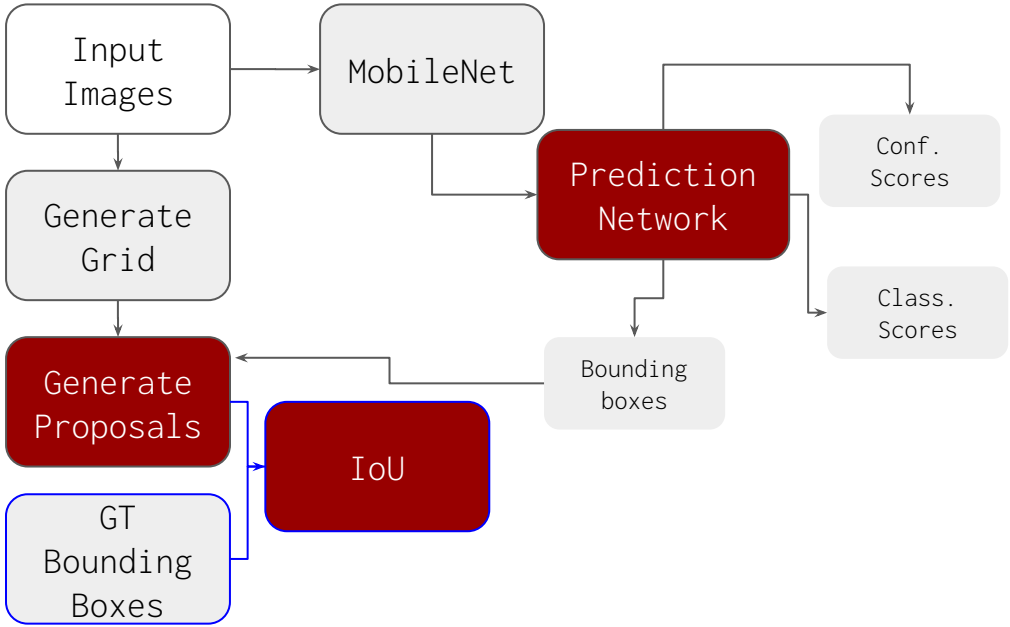


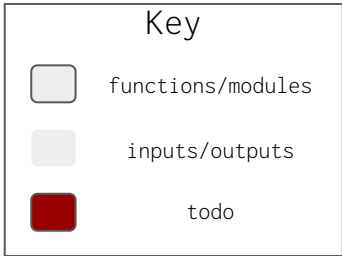


Proposal Generation

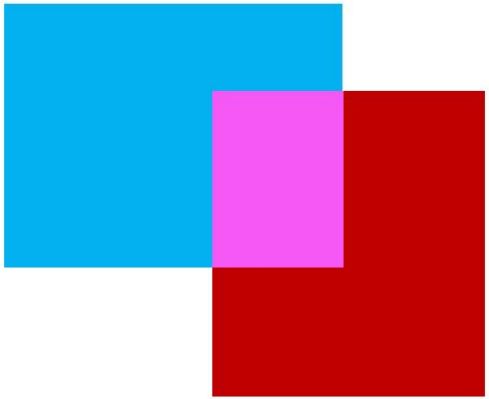
- The prediction network outputs offsets, which are bounding box coordinates with respect to grid centers
- During proposal generation, we transform these offsets into bounding box coordinates (top left x, top left y, bottom right x, bottom right y)

Adding IoU


Input Images
MobileNet
Generate Grid
Prediction Network
Conf. Scores
Class. Scores
Bounding boxes
Generate Proposals
IoU
GT Bounding Boxes


Key
functions/modules
inputs/outputs
todo

Intersection Over Union (IOU)

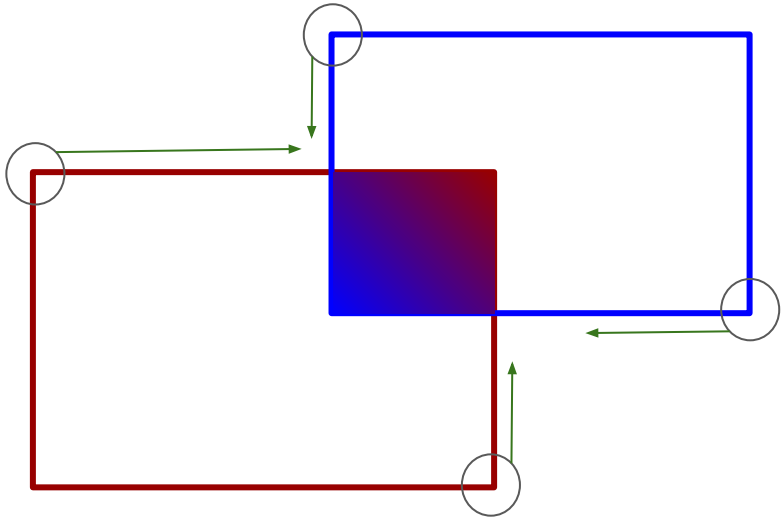


$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Intersection over union (also known as Jaccard similarity)

Source: B. Hariharan

Intersection Over Union (IOU)

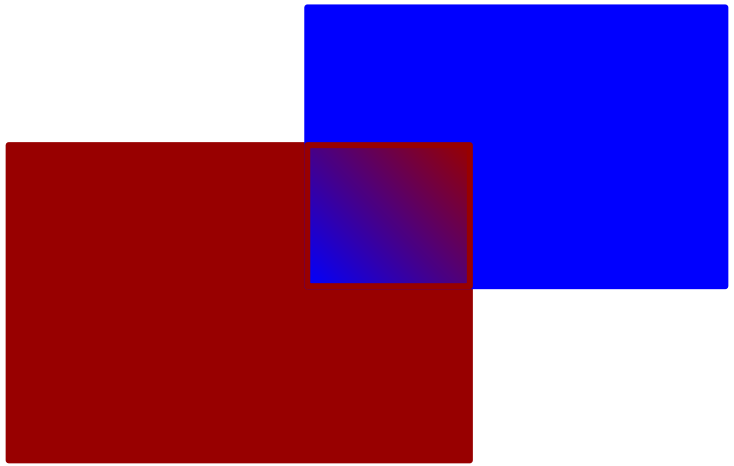


Consider how you can determine the intersection of two boxes using their top-left and bottom-right endpoints!

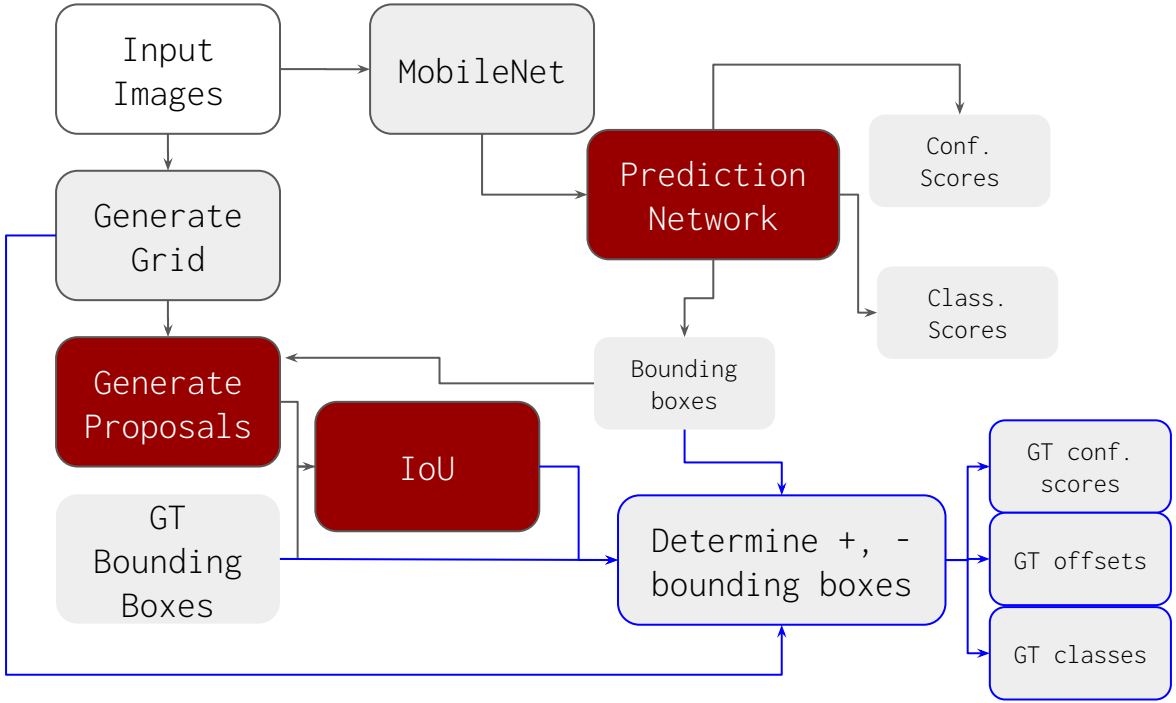
If you want to think about it in coordinates, (0,0) is located at the upper left corner of the "grid".

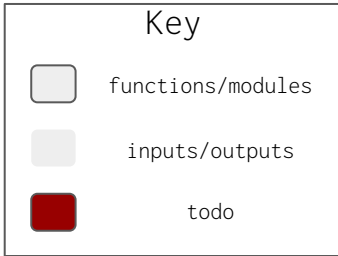
Intersection Over Union (IOU)

Union = sum of their areas - area of their intersection



Adding positive/negative bounding box determination



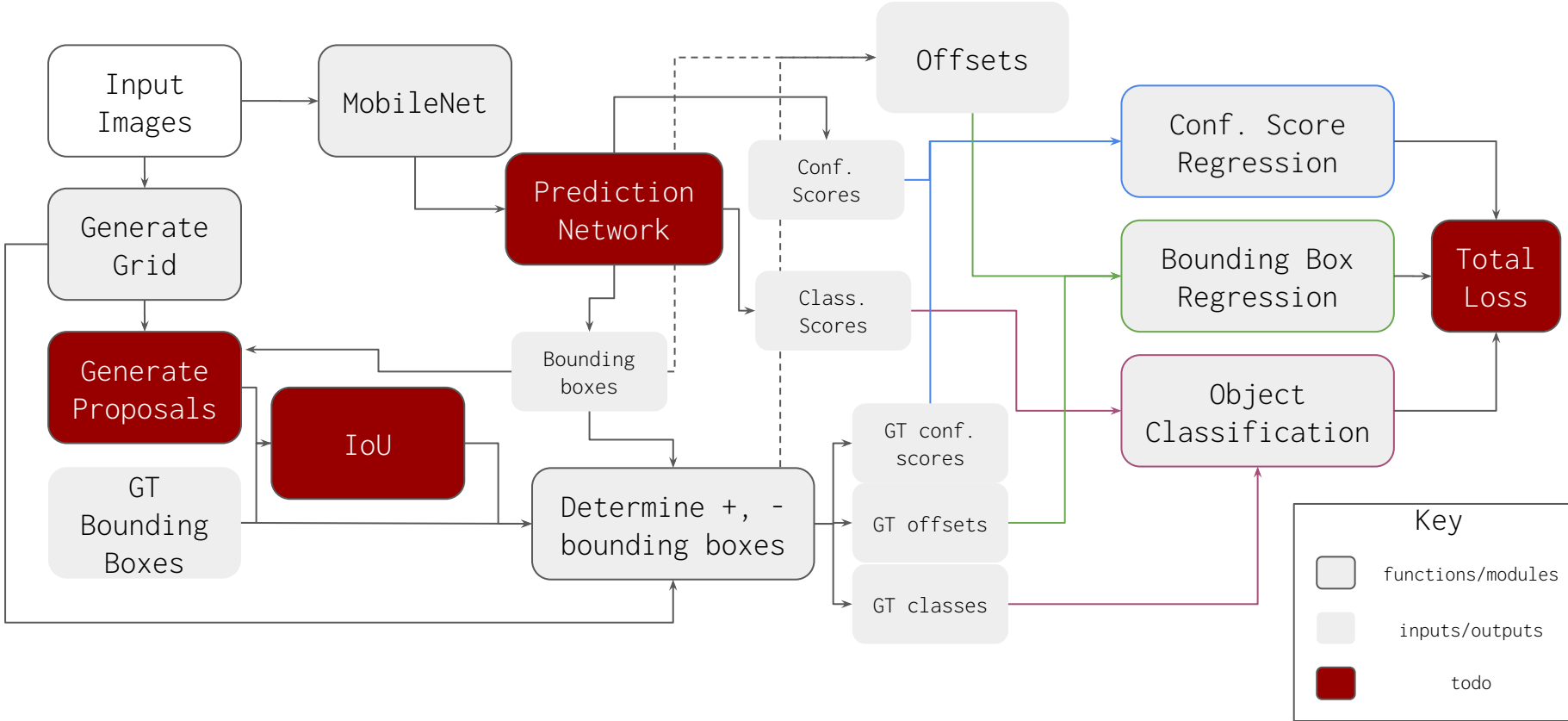


Determining positive and negative bounding boxes

During training, after we calculate the offset values for all bounding boxes, we need to match the ground-truth boxes against the predicted bounding boxes to determine the classification labels for the bounding boxes -- which boxes should be classified as containing an object and which should be classified as background?

We wrote this for you, but digest this carefully.

Adding Loss Calculations



Calculating the Object Detector's Loss

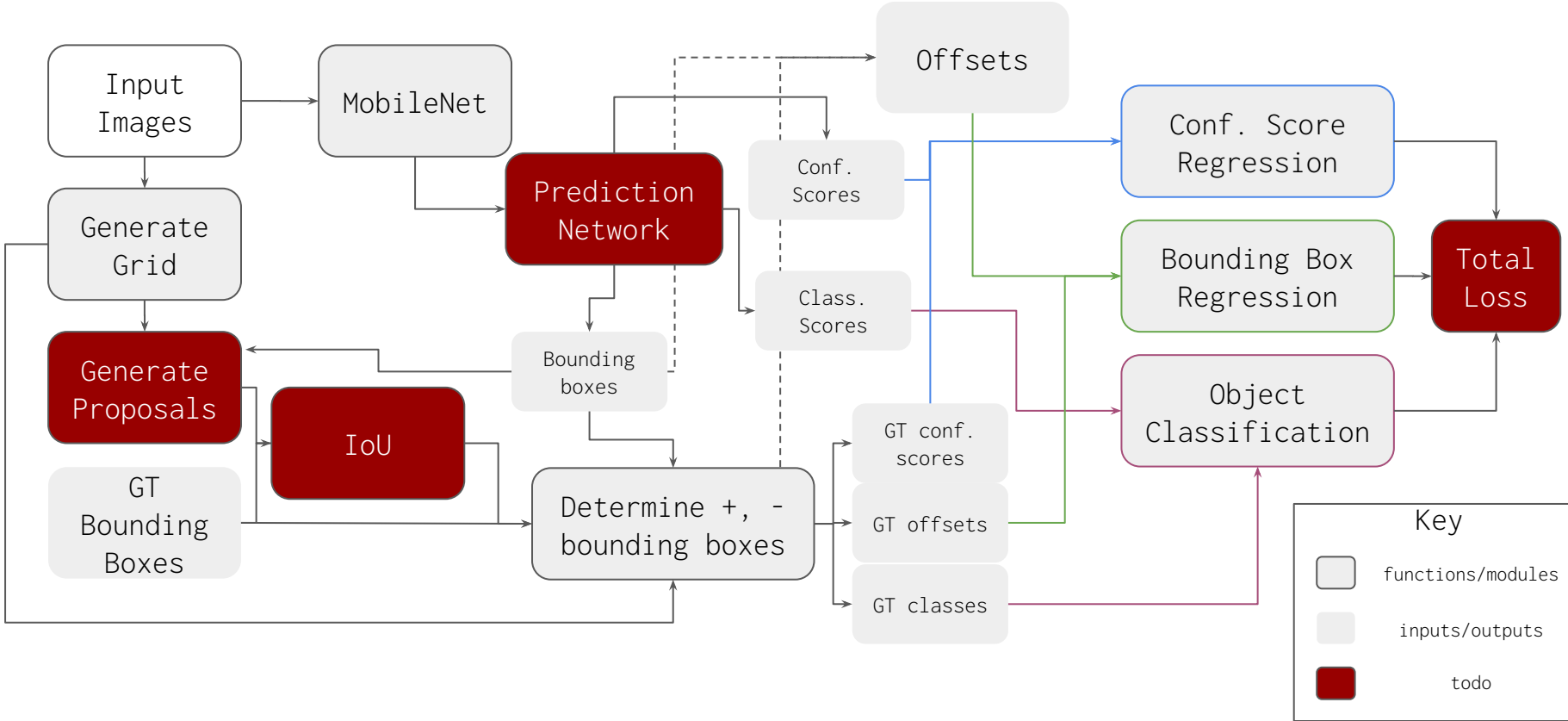
- The confidence score regression loss captures how confident the model is compared to how confident it should be
  - for both activated/negative bounding boxes / proposals

- The bounding box regression loss compares the offsets for activated bounding boxes and their ground truth offsets

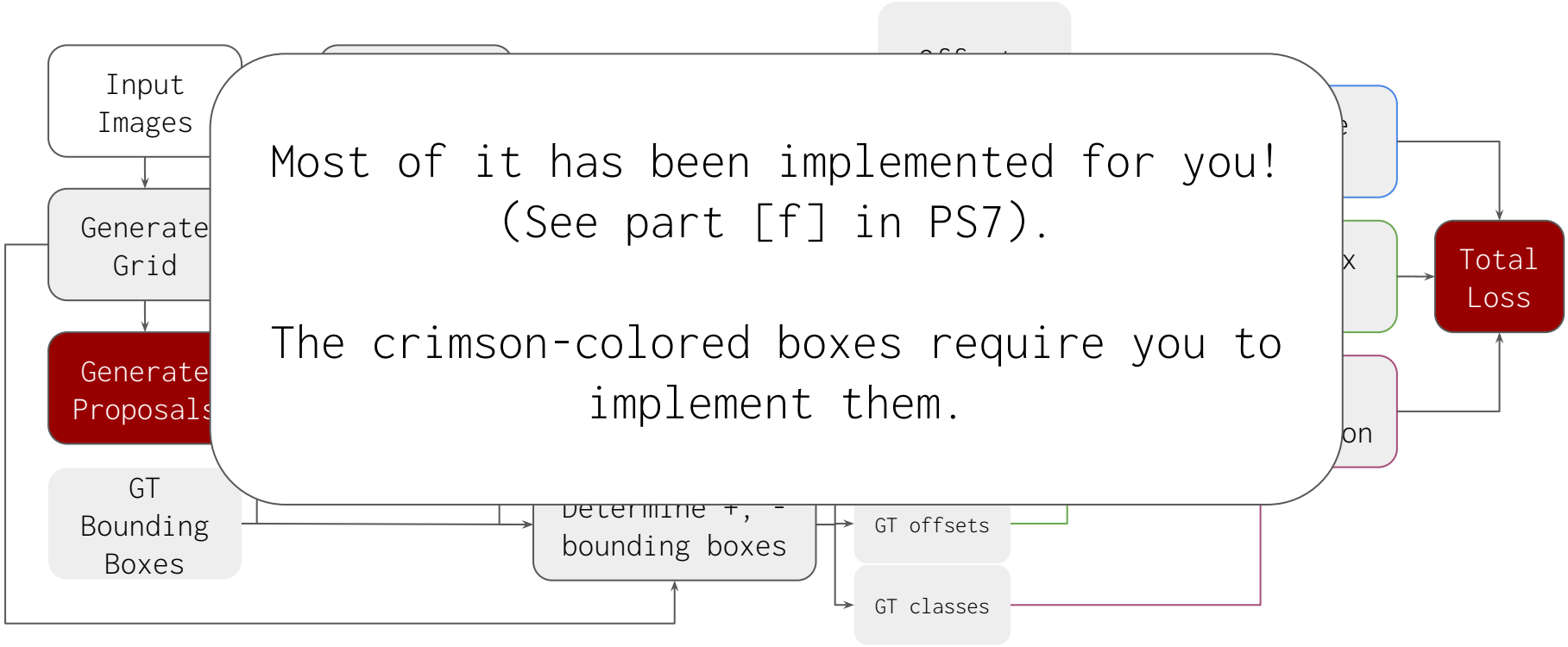
- The object classification loss compares classification output to ground truth
  - Both bbox regression loss and object cls loss are for activated bounding boxes only.

These have all been implemented for you! You just need to use these functions to calculate the total loss.

Comprehensive Overview of Single Stage Detector

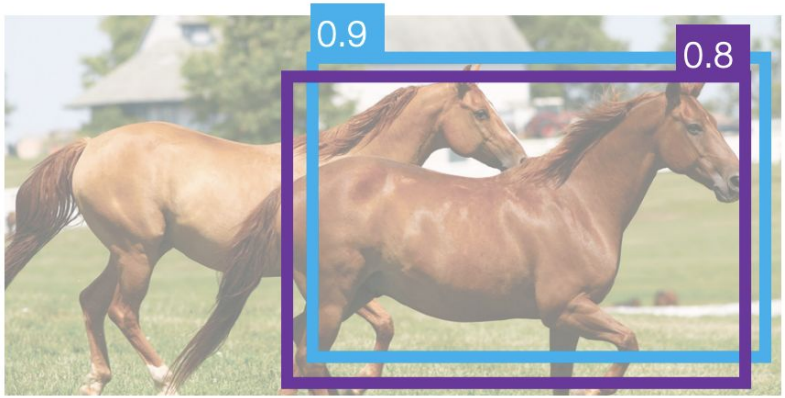


Comprehensive Overview of Single Stage Detector



Non-Maximum Suppression (NMS)

For inference, you will have to implement NMS to get rid of redundant bounding boxes.



- Subtlety: we predict a bounding box for every sliding window. Which ones should we keep?
- Keep only “peaks” in detector response.
- Discard low-prob boxes near high-prob ones
- Often use a simple greedy algorithm

Non-Maximum Suppression (NMS)

Greedy algorithm, run on each class independently

let A be the set of all bounding boxes
let D be the set of detections we'll keep, and set $D = \emptyset$
    while $A \neq \emptyset$:
        remove x, the box with highest probability from A
        if x doesn't significantly overlap with an existing box in D (IoU > 0.5):
            $D = D \cup \{x\}$
return D

Inference + Evaluation

We have implemented inference based on unseen input images for you. The inference step is dependent on NMS.

You will also evaluate your model via mean average precision (mAP). You should see 12%+ performance.