Concept Review - 2

Object Detection Systems using RCNNs

Ananya K R

9th January 2021

R-CNNs (Region-Based CNNs)

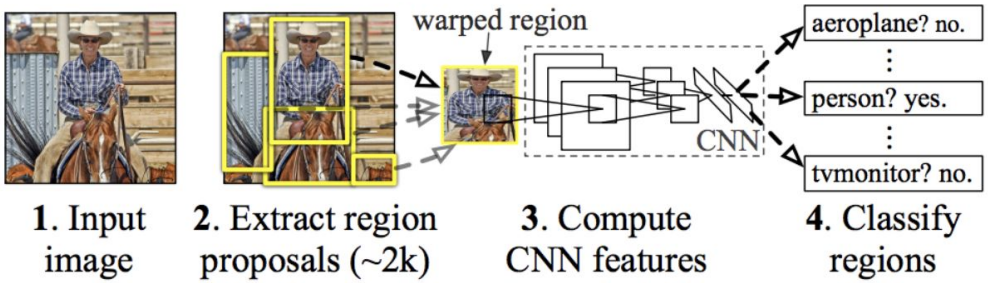
One of the first large and successful application of convolutional neural networks to the problem of object localization, detection, and segmentation. The approach was demonstrated on benchmark datasets, achieving then state-of-the-art results on the **VOC-2012 dataset and the 200-class ILSVRC-2013 object detection dataset**.

Their proposed R-CNN model is comprised of three modules; they are:

Module 1: Region Proposal. Generate and extract category independent region proposals, e.g. candidate bounding boxes.

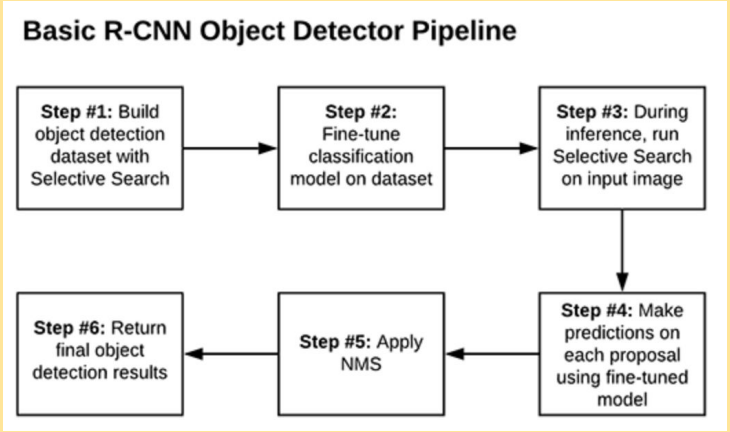
Module 2: Feature Extractor. Extract feature from each candidate region, e.g. using a deep convolutional neural network.

Module 3: Classifier. Classify features as one of the known class, e.g. linear SVM classifier model.



Practical Implementation Overview

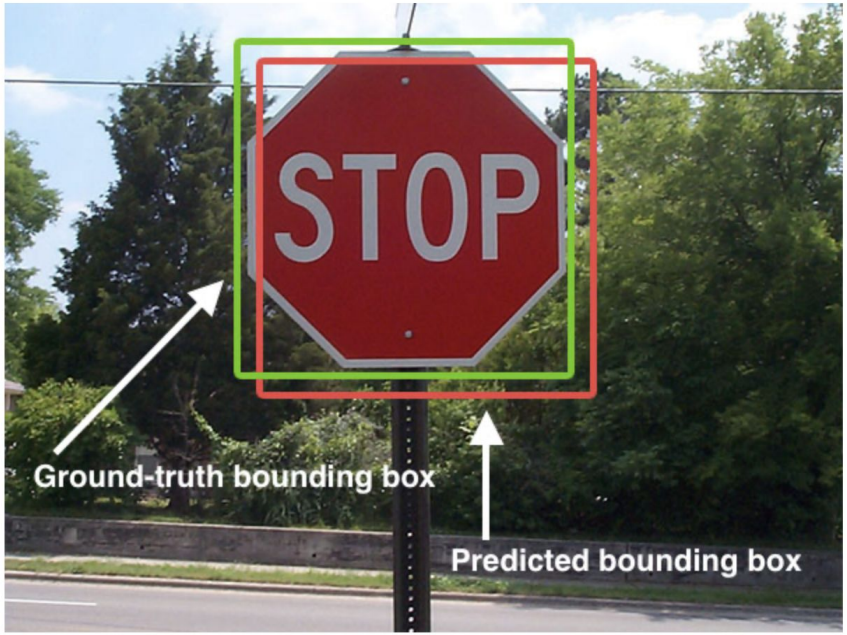
Using the Raccoon object detection dataset curated by **Dat Tran**, **R-CNN object detection with Keras, TensorFlow, and Deep Learning**



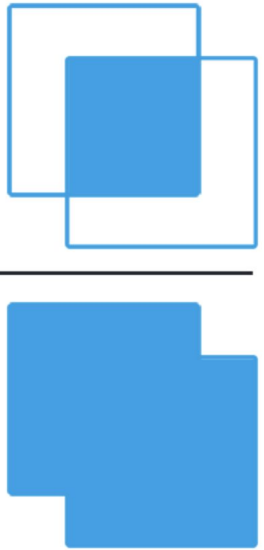
DanTran dataset vs My very own dataset

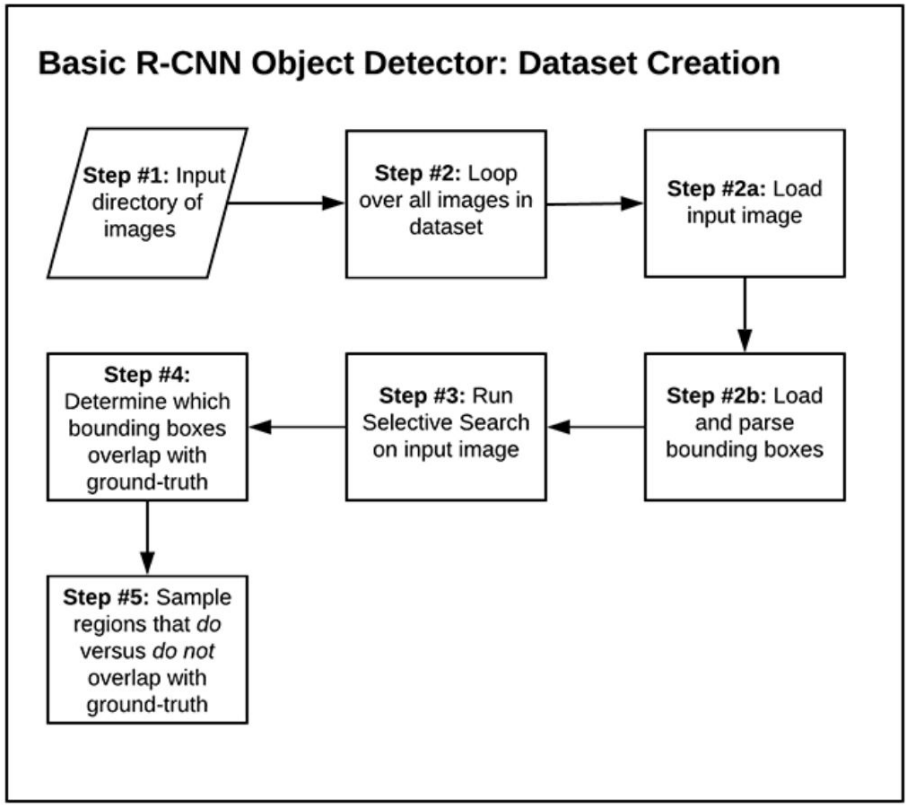
Selective Search to obtain region proposals

Metric used to find accuracy and create dataset - **IOU (Intersection over Union)**



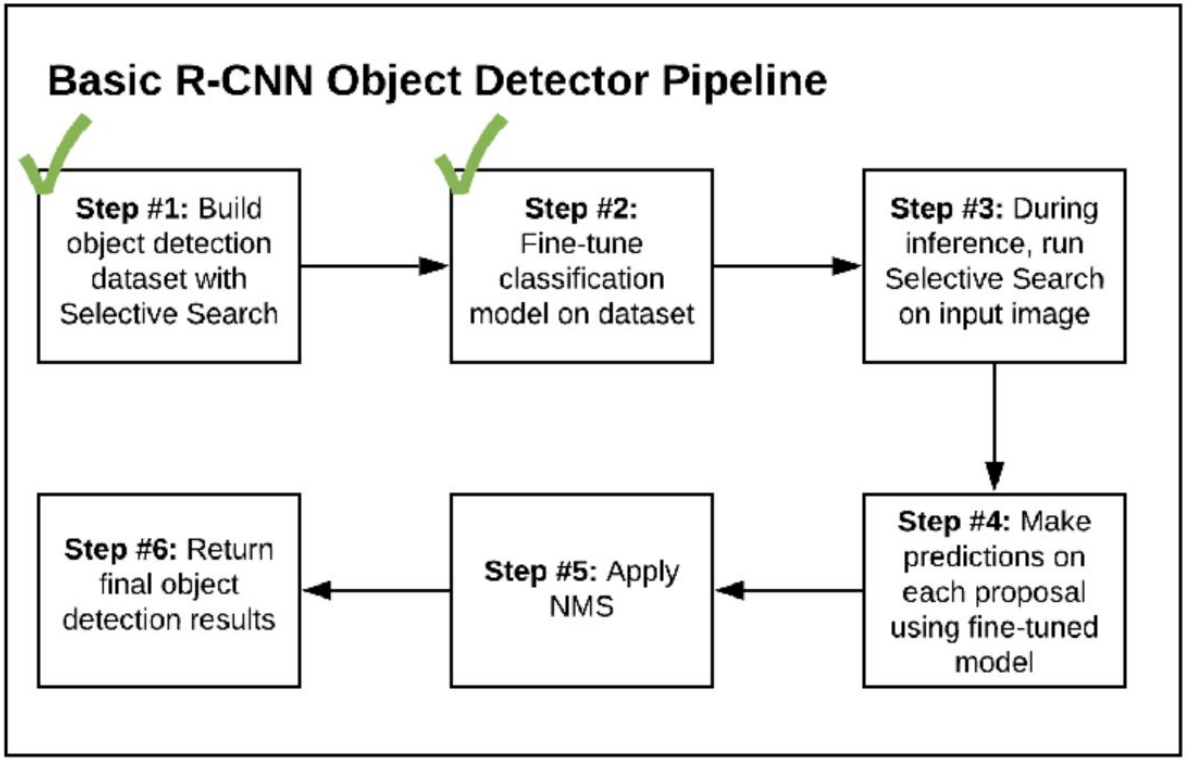
$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$




Basic R-CNN Object Detector: Dataset Creation
Step #1: Input directory of images
Step #2: Loop over all images in dataset
Step #2a: Load input image
Step #2b: Load and parse bounding boxes
Step #3: Run Selective Search on input image
Step #4: Determine which bounding boxes overlap with ground-truth
Step #5: Sample regions that do versus do not overlap with ground-truth

Preparing my image dataset for object detection using pre-trained MobileNet V2 CNN on ImageNet

Training the R-CNN object detection network with Keras and TensorFlow


Basic R-CNN Object Detector Pipeline
Step #1: Build object detection dataset with Selective Search
Step #2: Fine-tune classification model on dataset
Step #3: During inference, run Selective Search on input image
Step #4: Make predictions on each proposal using fine-tuned model
Step #5: Apply NMS
Step #6: Return final object detection results

- **Step #3:** Create an object detection inference script that utilized Selective Search to propose regions that *could* contain an object that we would like to detect
- **Step #4:** Use our fine-tuned network to classify each region proposed via Selective Search
- **Step #5:** Apply non-maxima suppression to suppress weak, overlapping bounding boxes
- **Step #6:** Return the final object detection results

Thank you