Lecture 1.6
Proposal-based Object Detectors

Dimitrios Papadopoulos
Assistant Professor, DTU Compute

Project 1.2

Detecting waste in the wild

Tasks for simple object detector:
- Extract object proposals
- Finetune a CNN for object detector on object proposals (replace last layer)
- Apply the model onE test images
- Implement NMS
- Evaluate the object detection performance

Save the environment: Detecting waste in the wild

Project 1.2

Deep Learning in Computer Vision

June 2022

Litter has been accumulating around us as most local governments and international organizations fail to tackle this crisis, which is having a catastrophic impact on biodiversity and marine animals. In this project, you are asked to build a deep learning object detection system that can automatically detect trash and litter and in images in the wild. This object detection can then be deployed in robotic machines that can scan areas and collect and clean beaches, forests and roads.

Figure 1: Examples from the TACO dataset.

Project 1.2

Detecting waste in the wild

The task

Your tasks for training and deploying a simple object detector to detect litter and trash are:

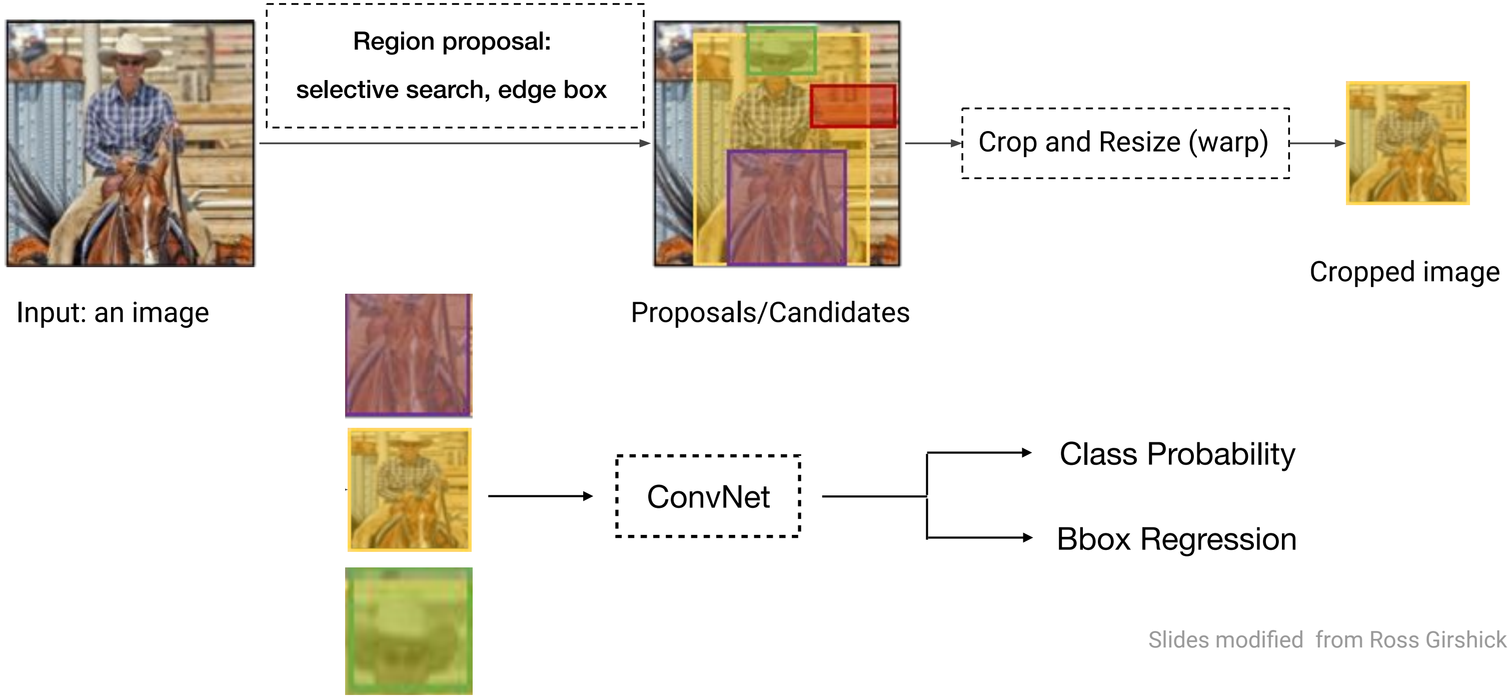
1. Extract object proposals for all the images of the dataset (e.g. Selecting Search, Edge Boxes, etc)
2. Finetune a convolutional neural network to classify object proposals.
3. Apply the model on the test images and implement non-maximum suppresion and Intersection over Union (IoU).
4. Evaluate the object detection performance using standard metrics.

Optional tasks:

1. Improve the simple model above by adding a bounding-box regression output that improves the detection performance.
2. Improve the efficiency of the simple model (i.e., ROI pooling layer inspired by Fast RCNN).
3. Implement a Convolutional Neural Network that is trained to generate generic object proposals to replace the object proposal algorithm (i.e., Region Proposal Network inspired by Faster RCNN).

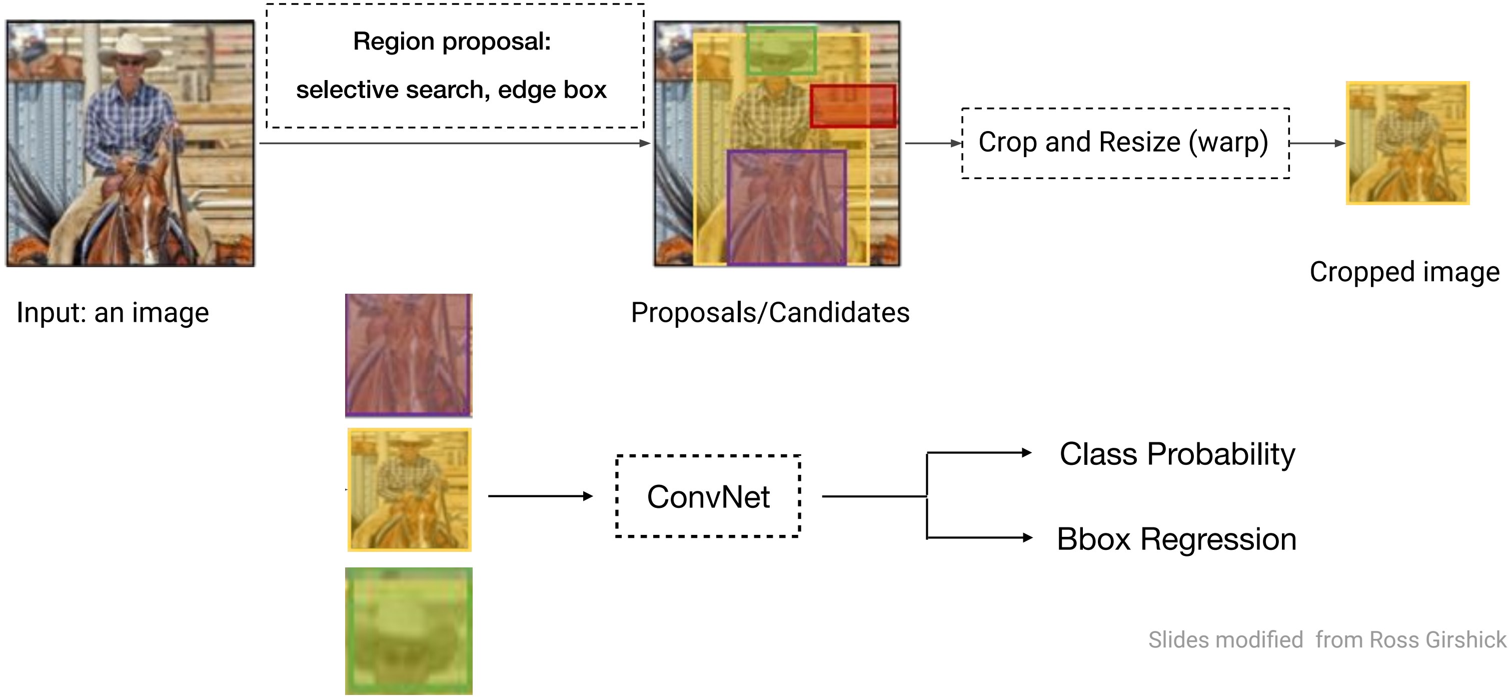
All together:
R-CNN: Region-based CNN

- Propose large number of regions potentially with objects
- Classify each proposed region



R-CNN Training

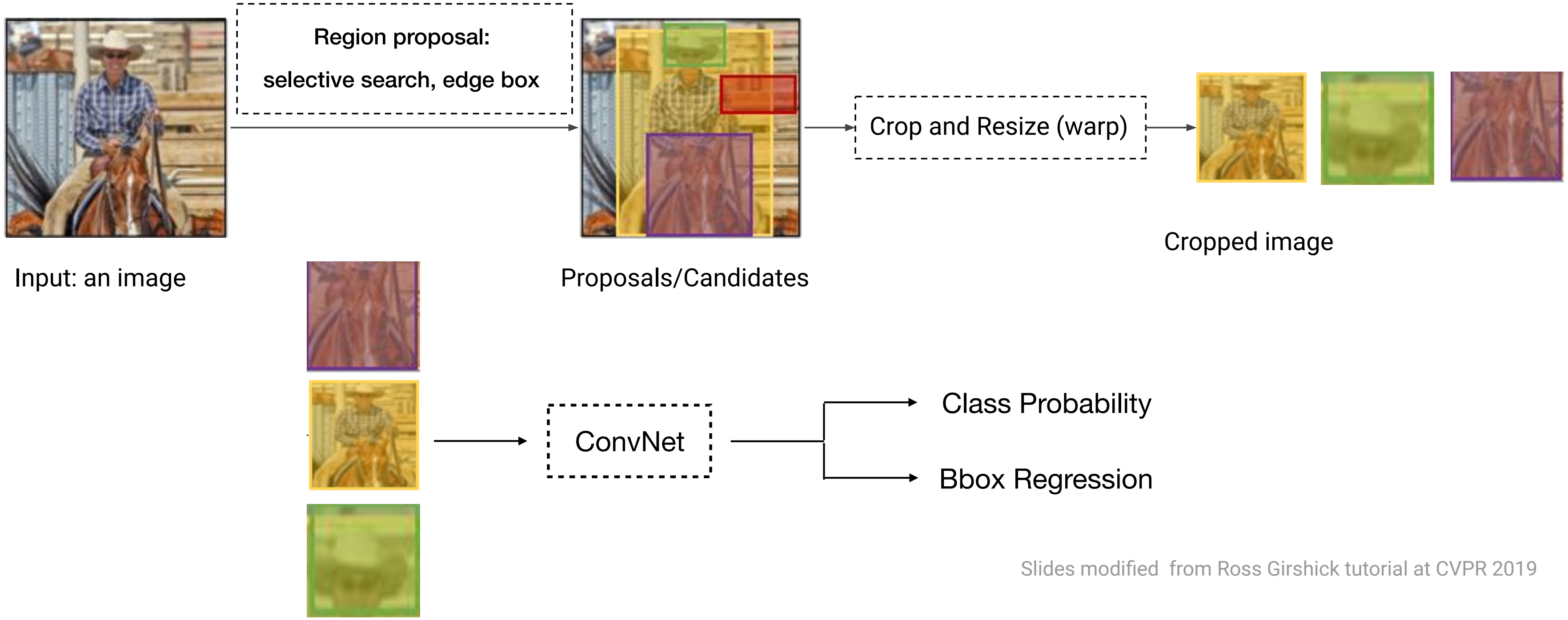
- Step 1: Train (or download) a classification model for ImageNet (AlexNet)



Slides modified from Ross Girshick tutorial at CVPR 2019

R-CNN Training

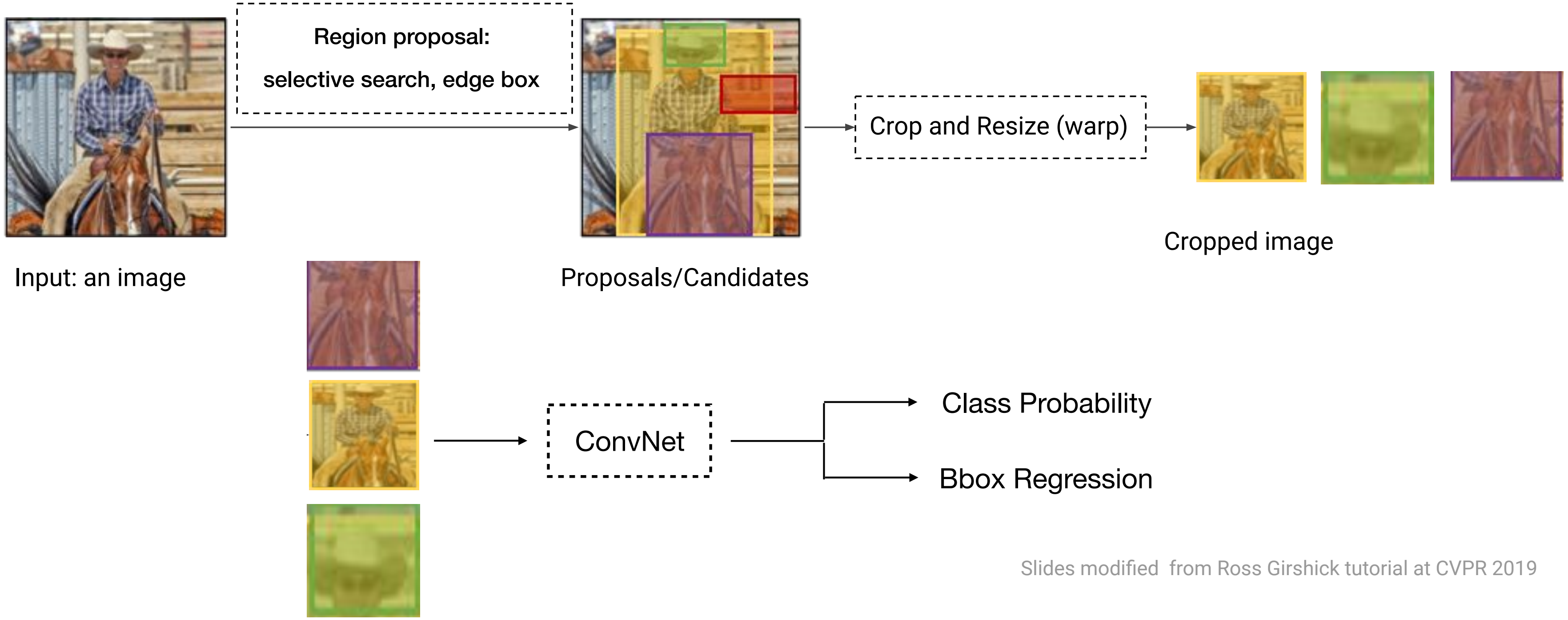
- Step 2: Fine-tune model for detection:
  - Instead of 1000 ImageNet classes → 20 object classes + 1 background
  - Throw away fc layer, re-initialize it
  - Input: Instead of images → Region Proposals (cropped and resized)



Slides modified from Ross Girshick tutorial at CVPR 2019

R-CNN Training

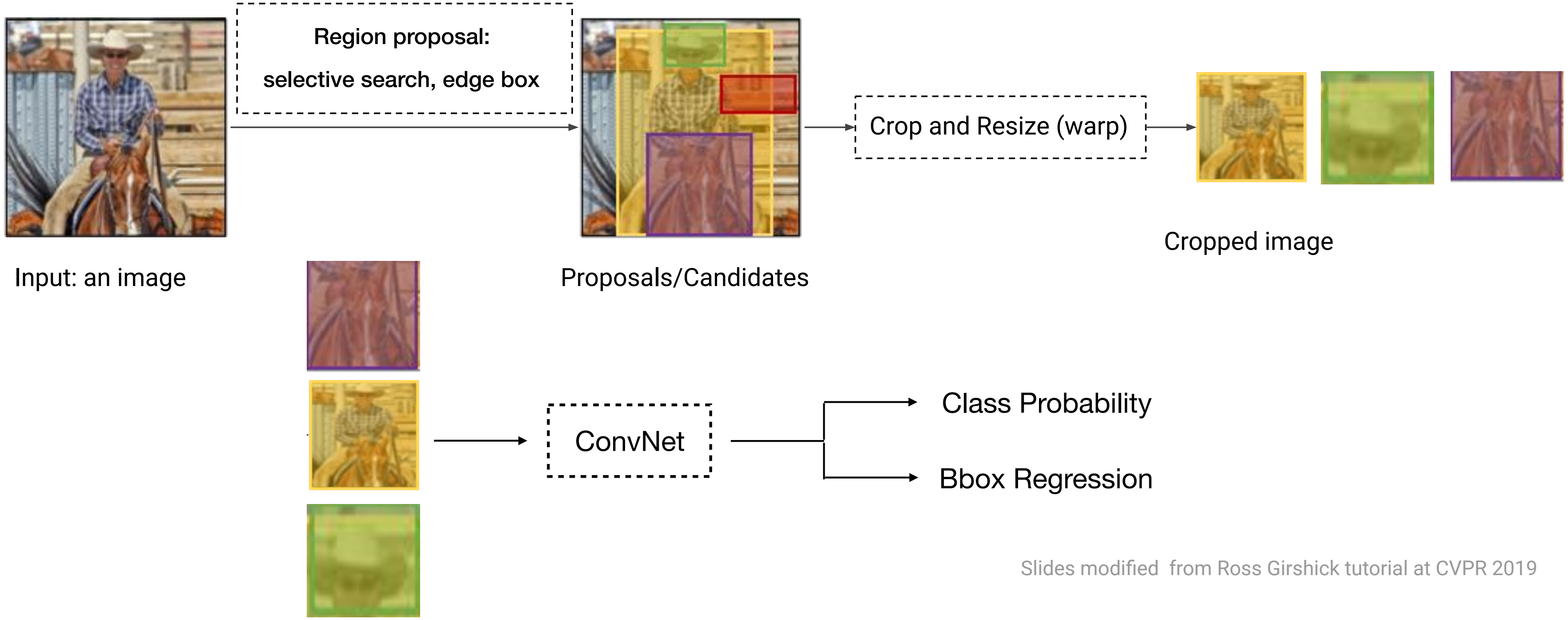
- Step 3: Extract features:
  - Input: Instead of images → Region Proposals (cropped and resized)
  - Save pool5 features to disk → ~100GB for a dataset of 10k images with 20 object classes (PASCAL VOC 2007)



Slides modified from Ross Girshick tutorial at CVPR 2019

R-CNN Training

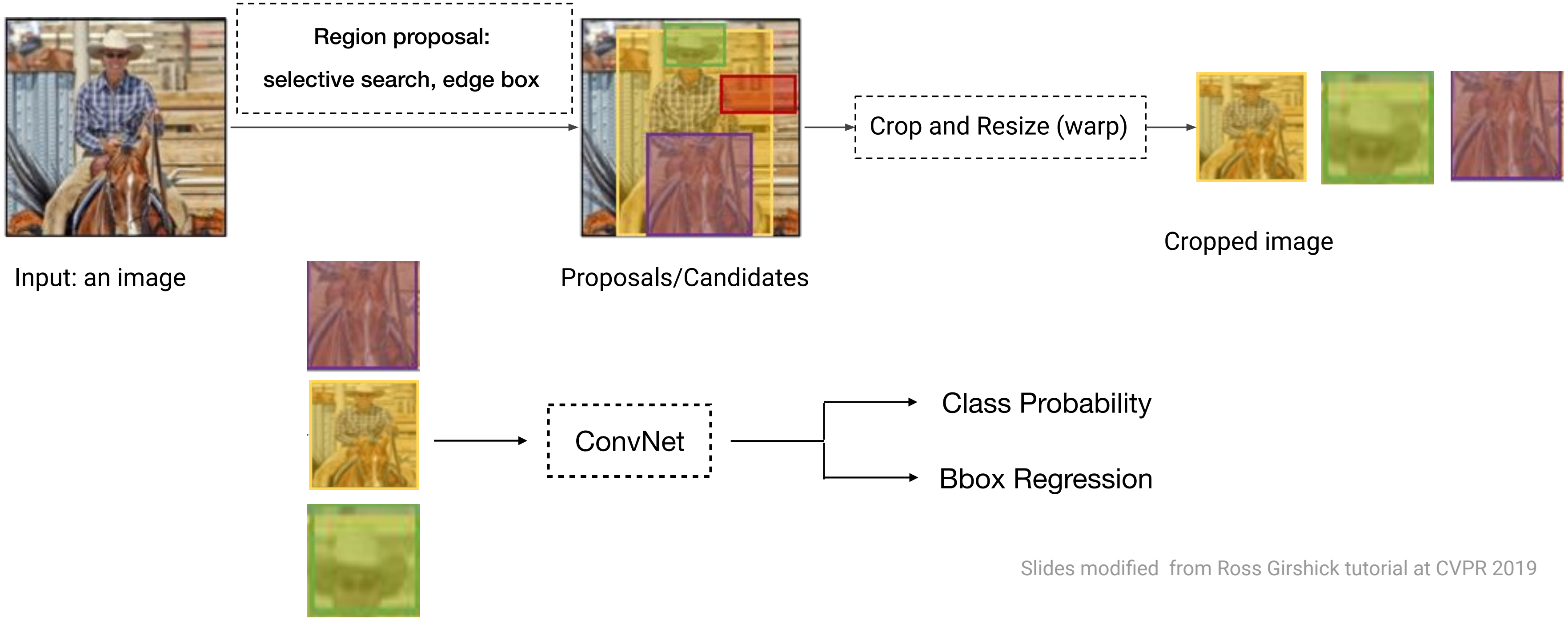
- Step 4: Train a binary SVM per class to classify region features



Slides modified from Ross Girshick tutorial at CVPR 2019

R-CNN Training

- Step 5: bounding-box regression:
  - For each class, train a linear regression model to map from features to offsets to ground-truth bounding boxes → makes up for ”slightly wrong” proposals



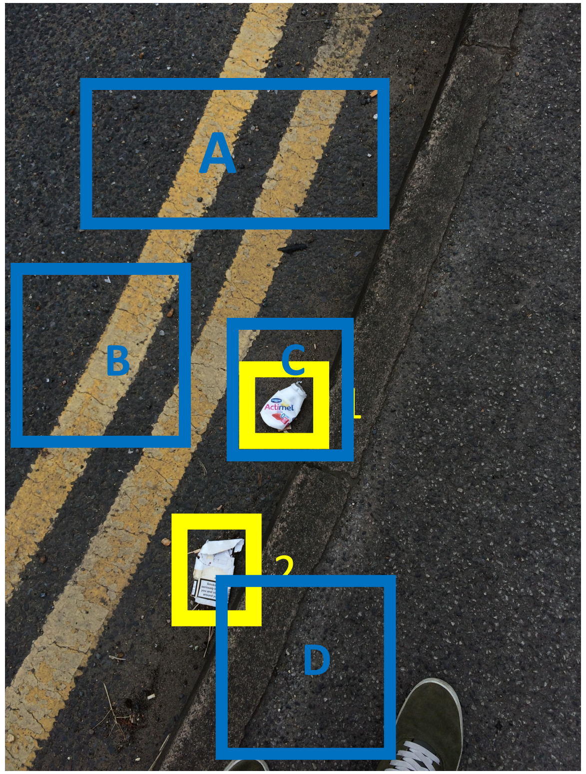
Slides modified from Ross Girshick tutorial at CVPR 2019

Basic Object Detector - Training

Basic Object Detector - Training

annotations.json

Basic Object Detector - Training

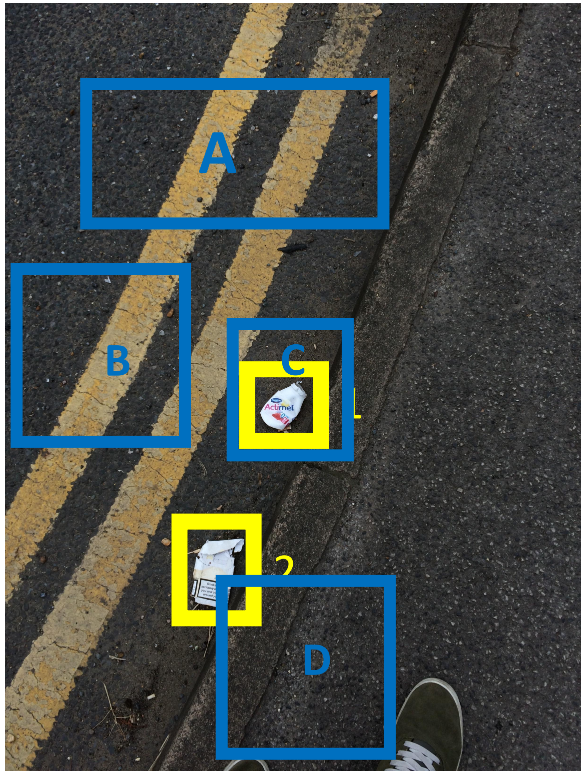


Run Selective Search, Edge Boxes, etc..

Tips:

- If slow, resize very large images beforehand (e.g. largest dimension 500)
- In case of SS, be sure that you use the ‘fast’ mode
- EdgeBoxes method is much faster

Basic Object Detector - Training



crop and
resize (wrap)

A

B

C

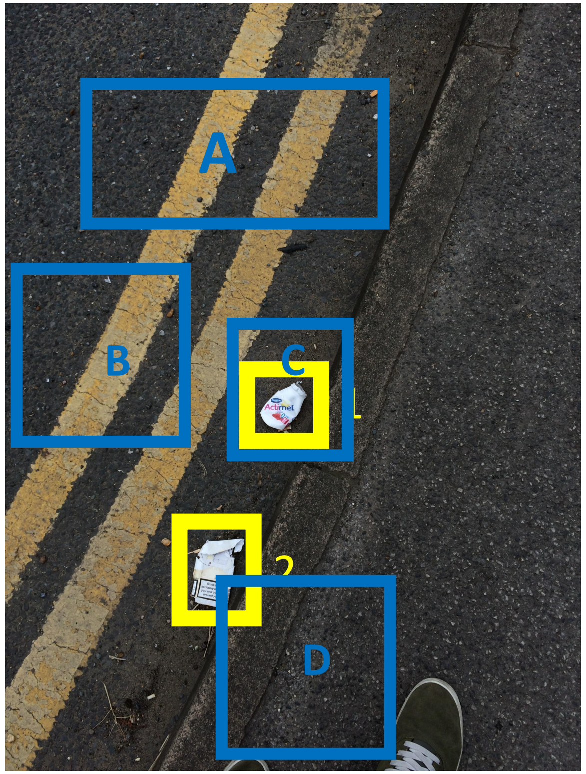
D

1

2

Basic Object Detector - Training

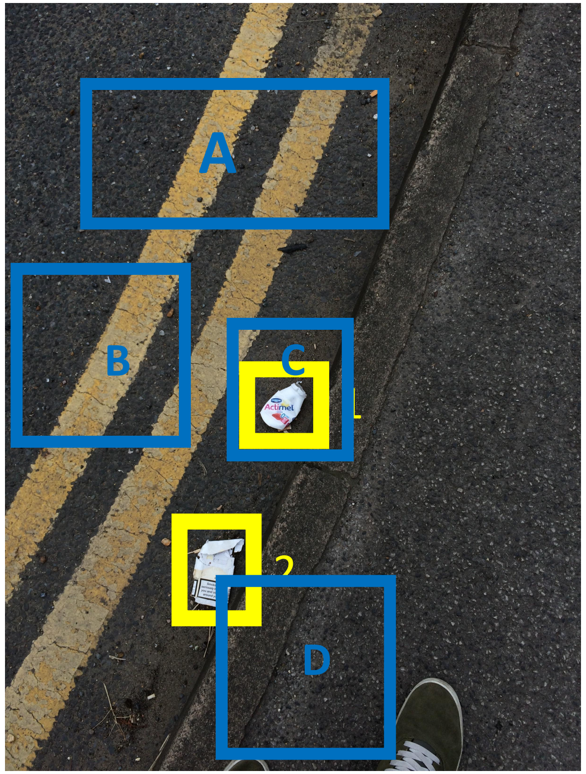
x y



crop and resize (wrap)

A
B
C
D
1
2

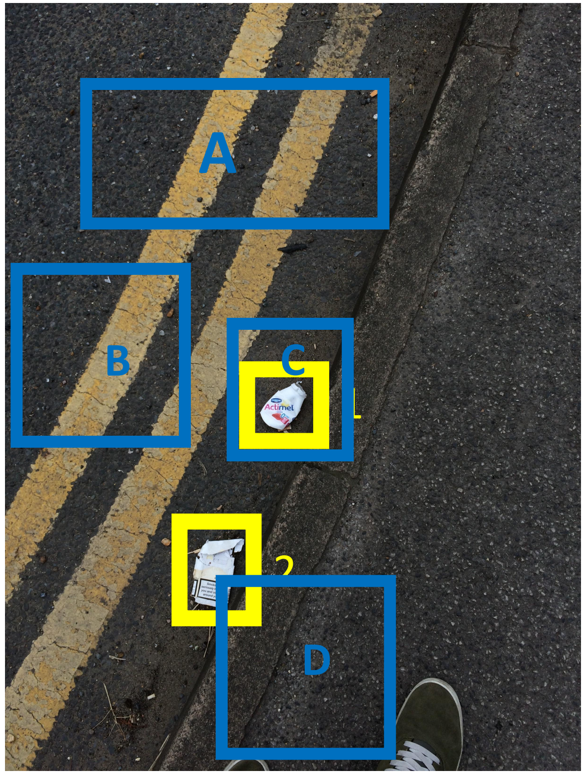
Basic Object Detector - Training



crop and resize (wrap) →

	x	y
A		?
B		?
C		?
D		?
1		class1
2		class2

Basic Object Detector - Training



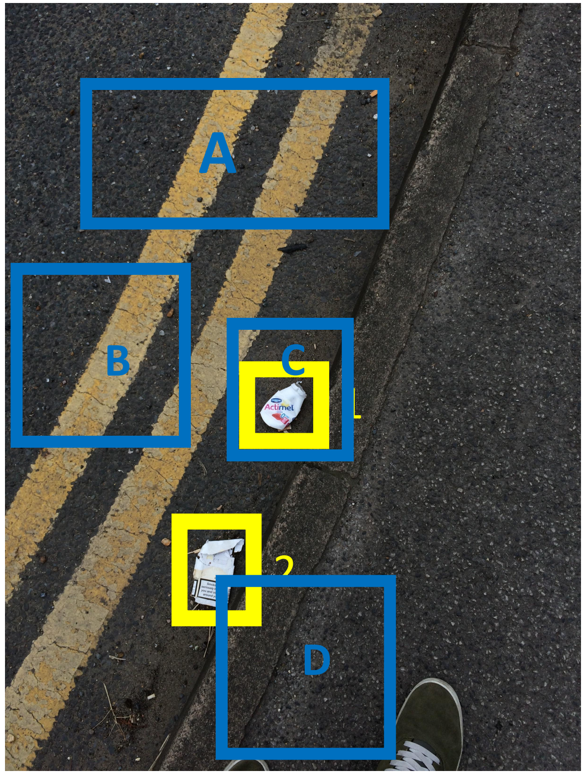
crop and resize (wrap) →

	x	y
A		?
B		?
C		?
D		?
1		class1
2		class2

If $\max_i(\mathrm{IoU}(A, GT_i)) < k1$, then A is background

If $\max_i(\mathrm{IoU}(A, GT_i)) >= k2$, then A is class of GT_i

Basic Object Detector - Training



crop and resize (wrap) →

x y

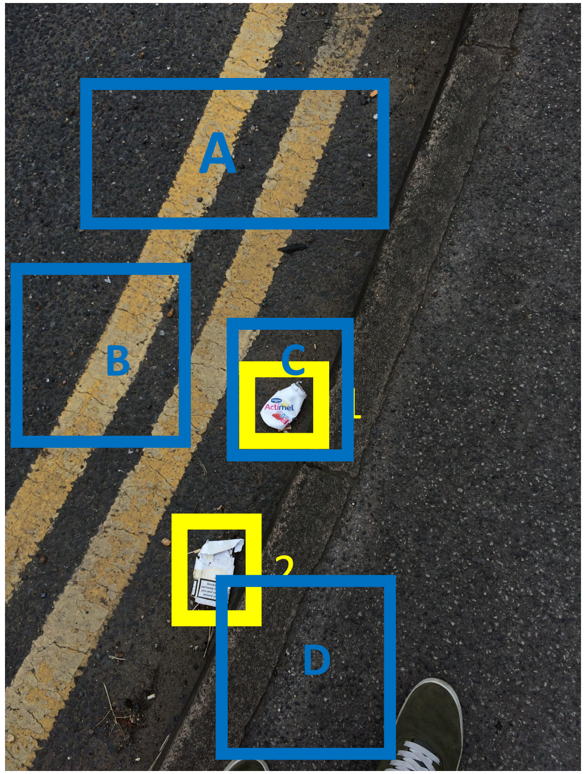
	y
A	background
B	background
C	class1
D	class2
1	class1
2	class2

If $\max_i(\mathrm{IoU}(A, GT_i)) < k1$, then A is background

If $\max_i(\mathrm{IoU}(A, GT_i)) >= k2$, then A is class of GT_i

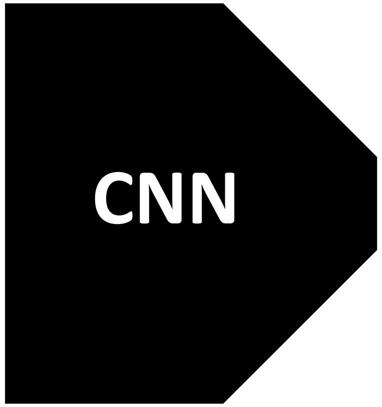
- k1=k2=0.5
- k1=0.5, k2=0.3
- k1=0.7 k2=0.3

Basic Object Detector - Training



crop and resize (wrap)

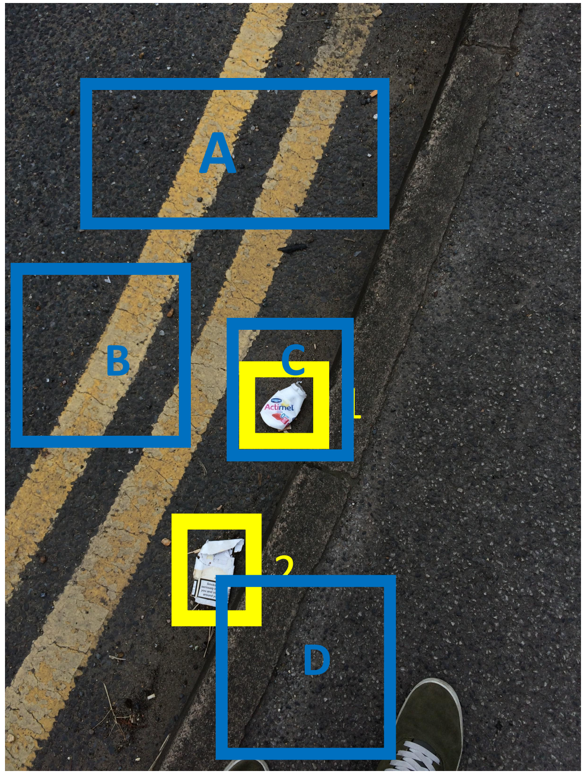
	x	y
A		background
B		background
C		class1
D		class2
1		class1
2		class2



Finetune a CNN pretrained on ImageNet to classify proposal windows

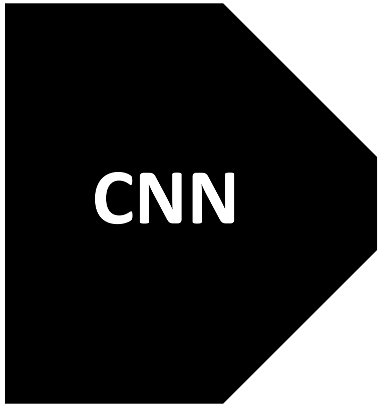
N+1 output classes

Basic Object Detector - Training



crop and resize (wrap)

	x	y
A		background
B		background
C		class1
D		class2
1		class1
2		class2



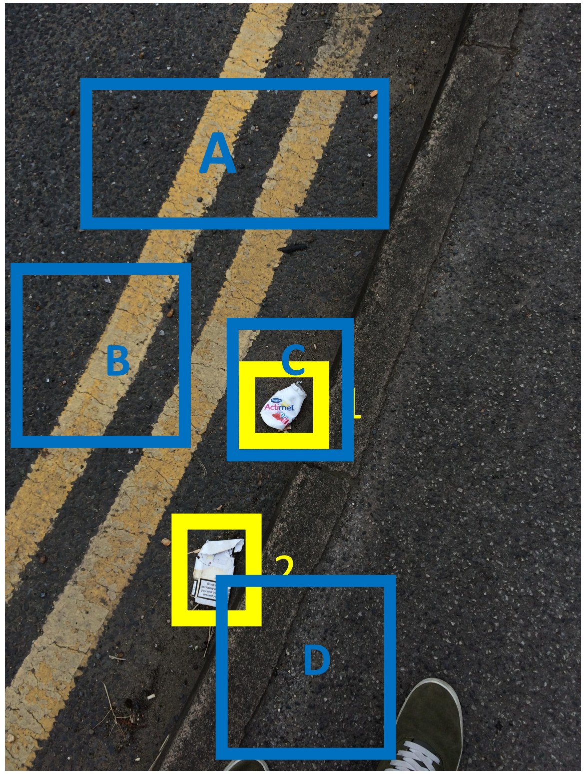
Finetune a CNN pretrained on ImageNet to classify proposal windows

N+1 output classes

- Be careful with the class imbalance:
>>99% of proposals will be "background"

- Sample proposals so that you train with 75% background and 25% of any class

Basic Object Detector - Training



crop and resize (wrap)

x y

A background

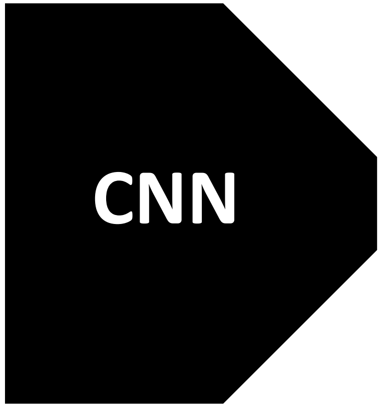
B background

C class1

D class2

1 class1

2 class2



Finetune a CNN pretrained on ImageNet to classify proposal windows

N+1 output classes

- Be careful with the class imbalance:
>>99% of proposals will be "background"

- Sample proposals so that you train with 75% background and 25% of any class
(e.g. batch:64, 38 background windows, 16 "positive"

Basic Object Detector - Test

Run Selective Search, Edge Boxes, etc..

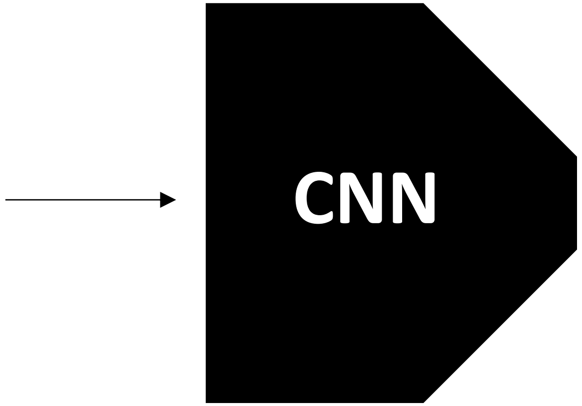
Tips:

- If slow, resize very large images beforehand (e.g. largest dimension 500)
- In case of SS, be sure that you use the ‘fast’ mode
- EdgeBoxes method is much faster

Basic Object Detector - Test

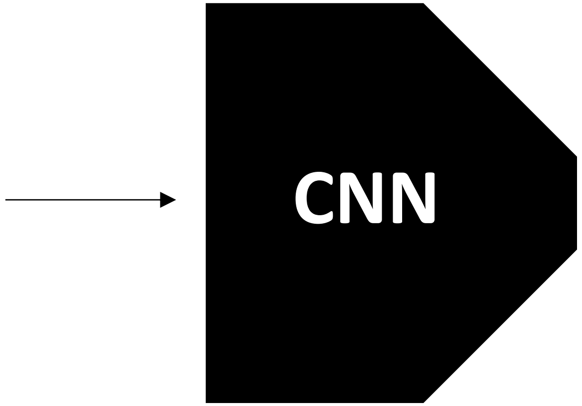
Basic Object Detector - Test

Apply the model to predict the classes of the proposals



Basic Object Detector - Test

Apply the model to predict the classes of the proposals



background (score 0.7)

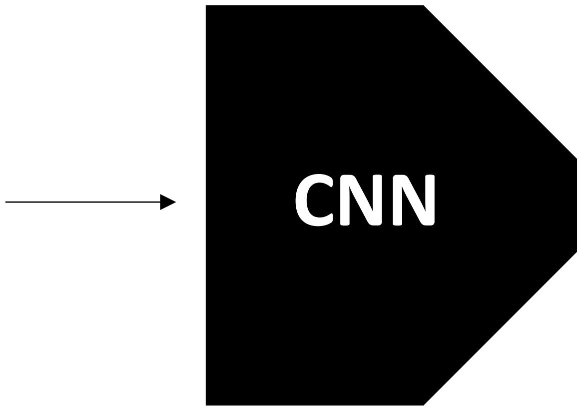
Background (score 0.5)

Background (score 0.8)

Class1 (score 0.9)

Basic Object Detector - Test

Apply the model to predict the classes of the proposals



Basic Object Detector - Test

Post-processing: NMS

Basic Object Detector - Test

Evaluate the output of the object detector: mAP

Prediction

Ground-truth

Project 1.2

Detecting waste in the wild

The task

Your tasks for training and deploying a simple object detector to detect litter and trash are:

1. Extract object proposals for all the images of the dataset (e.g. Selecting Search, Edge Boxes, etc)
2. Finetune a convolutional neural network to classify object proposals.
3. Apply the model on the test images and implement non-maximum suppresion and Intersection over Union (IoU).
4. Evaluate the object detection performance using standard metrics.

Optional tasks:

1. Improve the simple model above by adding a bounding-box regression output that improves the detection performance.
2. Improve the efficiency of the simple model (i.e., ROI pooling layer inspired by Fast RCNN).
3. Implement a Convolutional Neural Network that is trained to generate generic object proposals to replace the object proposal algorithm (i.e., Region Proposal Network inspired by Faster RCNN).