AFRICAN UNIVERSITY OF SCIENCE & TECHNOLOGY
COMPUTER SCIENCE DEPARTMENT

**Masters Thesis**

# Fast and Accurate Feature-based Region Identification

**Abstract**

There have been several improvement in object detection and semantic segmentation results in recent years. Baseline systems that drives these advances are Fast/Faster R-CNN, Fully Convolutional Network and recently **Mask R-CNN** and its variant that has a weight transfer function. Mask R-CNN is the state-of-art. This research extends the application of the state-of-art in object detection and semantic segmentation in drone based datasets.

Existing drone datasets was used to learn semantic segmentation on drone images using **Mask R-CNN**. And a new drone dataset will be collected, labelled, annotated with a bounding box object detection.

Drone based images will be collected with the drone developed by the Robotic team at African University of Science and Technology, Abuja, which will be made public for academic researches.

This work is the result of my own activity. I have neither given nor received unauthorized assistance on this work.

JUNE 2019

MADUAKOR FRANCIS

THESIS SUPERVISOR:
PROF. DR. LEHEL CSATÓ

FACULTY OF MATHEMATICS AND INFORMATICS,
BABEŞ BOLYAI UNIVERSITY OF CLUJ-NAPOCA,
ROMANIA

African University of Science & Technology
Computer Science Department

Masters Thesis

# Fast and Accurate Feature-based Region Identification

Thesis supervisor:

Prof. dr. Lehel Csató

Faculty of Mathematics and Informatics,
Babeş Bolyai University of Cluj-Napoca,
Romania

Student:

Maduakor Francis

June 2019

# Contents

## 1. Chapter

# Introduction

## 1.1 Introduction

Images and videos are collected everyday by different sources. Recognizing objects, segmenting localizing and classifying them has been a major area of interest in computer vision. Significant progress has been made commencing from use of low-level image features, such as **scale invariant feature transform** SIFT [A] and **histogram of oriented gradients HOG** [B] , in sophisticated machine learning frameworks to the use of multi-layer convolutional networks to compute highly discriminative, and invariant features [C]. SIFT and HOG are feature descriptor and semi-local orientation histograms that counts occurrences of gradient orientation in localized portions of an image. Just as Convolutional Neural Network (CNN) is traced to the Fukushima's **"neocognitron"** [D], a hierarchical and shift-invariant model for pattern recognition, the use of CNN for region-based identification (R-CNN)[C] can also be traced back to the same. After CNN was considered inelegant in the 1990s due to the rise of support vector machine (SVM), in 2012 it was revitalize by Krizhevsky et al. [D] by demonstrating a valuable improvement in image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [E] and included new mechanisms to CNN like rectified linear unit (ReLU) and, dropout regularization. To perform object detection with CNN and in attempt to bridge the gap between image segmentation and object detection two issues were fixed by R.Girshick et al [C]. First was the localization of objects with a Deep Network and training a high-capacity model with only a small quantity of annotated detection data. Use of a sliding-window detector was proposed for the localization of object but was not preferred because it can only work for one object detection and all object in an image has to have a common aspect ratio for its use in multiple object detection. Instead the localization problem was solved by operating within the "recognition using regions" paradigm.

Fast R-CNN was introduced in 2015 by Girshick [F]. A single-stage training algorithm that jointly learns to classify object proposals and refine their spatial locations was demonstrated. This tackled the problem of complexity that arises in other deep ConvNets [D,G, H], caused by the multi-stage pipelines that are slow. The slow nature is due to the fact that detection requires accurate localization of objects that creates the challenge of that many proposals (candidate object locations) must be processed and these proposals provides only rough localization that must be refined to achieve precise localization. Fast R-CNN is 9 X faster than R-CNN [C] and 3 X faster than SPPnet [I]. R-CNN was speed up by **Spatial pyramid pooling networks (SPPnets)**[I] by sharing computation. A convolutional feature map for the entire input image was computed by SPPnet method. After which it then classifies each object proposal using a feature vector extracted from the shared feature map. SPPnet also has obvious pitfalls. It is a multi-stage pipeline similarly to R-CNN that involves extracting features, refining a network with log loss, training SVMs, and lastly fitting bounding-box regressors. Features are also written to disk. But unlike R-CNN, the refining algorithm demonstrated in SPPnet cannot update the convolutional layers that precede the spatial pyramid pooling. This constraint limits the accuracy of very deep networks. Additional efforts were made to reduce the running time of deep ConvNets for object detection and segmentation. Regional proposal computation is the root of this expensive running time in detection networks. A fully convolutional network that simultaneously predicts object bounds and objectness scores at each position called **Region Proposal Network (RPN)** was developed by Ren et al [J]. RPN shares full-image convolutional features with the detection network, thus permitting virtually cost-free region proposals and it is trained end-to-end to generate high-quality region proposals. Integrating RPN and Fast R-CNN into a unit network by sharing their convolutional features results to Faster R-CNN. Anchor boxes that acts as reference at multiple scales and aspect ratios were introduced in Faster R-CNN instead of the pyramids of filters used in earlier methods. RPNs are developed to coherently speculate region proposals with an extensive range of scales and aspect ratios. Changing the architecture of the pyramids of filter to a top-down architecture with lateral connections improved the efficiency of this pyramids [K]. This is applied in building high-level semantic feature maps at all scales. This new architecture is called **Feature Pyramid Network (FPN)** [K]. In various applications and uses it displayed a notable improvement as a generic feature extractor. When used in a Faster R-CNN it achieved results that supersedes that of Faster R-CNN alone. In order to generate a high-quality segmentation mask for object instances in an image, Mask R-CNN was developed [L]. Mask R-CNN add another branch to the Faster R-CNN. In addition to the bounding box recognition system a branch for predicting an object mask in parallel was added. It affixes only a bijou overhead to Faster R-CNN, running at **5 fps**.

The accessibility and use of drone technology is at the increase currently. It is tackling challenges in various spheres and areas like defence, shipping of consumer goods, disease controls, events coverage and so on. One of the most important application of drone is for collection of images and videos. These data collected can be used for different purposes. This work will extend the state-of-the-art Mask R-CNN for segmentation of objects in image instances collected by a drone. It detects about 22 classes including tree, grass, other vegetation, dirt, grave, rocks, water, paved area, pool, person, dog, car, bicycle, roof, wall, fence, fence-pole, window, door, and obstacle. For the training of the model high resolution images at 1Hz with pixel-accurate annotation was used.

In Chapter 2 of this work will discuss theory of CNN and Mask RCNN deeply. The first part will discuss the backbone of Mask RCNN, followed by Regional Proposal Network, ROI Classifier and Bounding Box Regressor and lastly Segmentation Mask. Chapter 3 will discuss fully segmentation on drone dataset. Chapter 4 will explore the methodology and implementation of the work.

## 2. Chapter

# Chapter about theory

## 2.1 Computer Vision Tasks

With the advance of computer vision, task in computer vision has moved from simple tasks of image classifaication to complex task like semantic and instance segmentation. Deep learning has made this possible, especially using the Vo

### 2.1.1 Image Classification

Image classification is the process of assigning land cover classes to pixels. Image classification refers to the task of extracting information classes from a multiband raster image. The resulting raster from image classification can be used to create thematic maps. Depending on the interaction between the analyst and the computer during classification, there are two types of classification: supervised and unsupervised. The image classification plays an important role in environmental and socioeconomic applications. In order to improve the classification accuracy, scientists have laid path in developing the advanced classification techniques. Image classification analyzes the numerical properties of various image features and organizes data into categories. Classification algorithms typically employ two phases of processing: training and testing. In the initial training phase, characteristic properties of typical image features are isolated and, based on these, a unique description of each classification category, i.e. training class, is created. In the subsequent testing phase, these feature-space partitions are used to classify image features. The description of training classes is an extremely important component of the classification process. In supervised classification, statistical processes (i.e. based on an a priori knowledge of probability distribution functions) or distribution-free processes can be used to extract class descriptors. Unsupervised classification relies on clustering algorithms to automatically segment the training data into prototype classes. In either case, the motivating criteria for constructing training classes are that they are:

1. Independent, e.a change in the description of one training class should not change the value of another,

2. Discriminatory, e.different image features should have significantly different descriptions, and

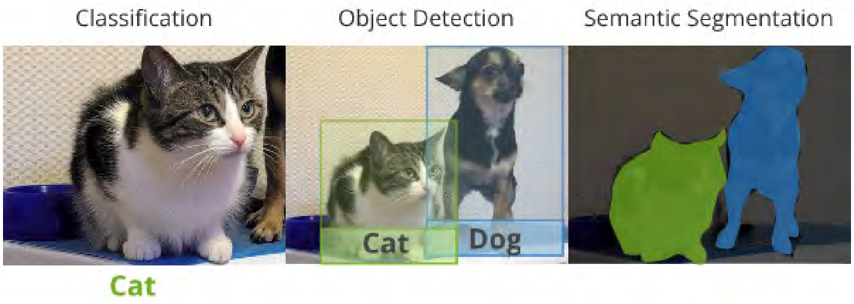

Figure 2.1: Image Classification,Object detection Semantic Segmentation.

3. Reliable, all image features within a training group should share the common definitive descriptions of that group.

This representation allows us to consider each image feature as occupying a point, and each training class as occupying a sub-space (i.e. a representative point surrounded by some spread, or deviation), within the n-dimensional classification space. Viewed as such, the classification problem is that of determining to which sub-space class each feature vector belongs.

### 2.1.2 Object Detection

The goal of object detection is to detect all instances of objects from a known class, such as people, cars or faces in an image. Typically only a small number of instances of the object are present in the image, but there is a very large number of possible locations and scales at which they can occur and that need to somehow be explored. Each detection is reported with some form of pose information. This could be as simple as the location of the object, a location and scale, or the extent of the object defined in terms of a bounding box. In other situations the pose information is more detailed and contains the parameters of a linear or non-linear transformation. For example a face detector may compute the locations of the eyes, nose and mouth, in addition to the bounding box of the face. An example of a vehicle and person detection that specifies the locations of certain parts is shown in Figure 1. The pose could also be defined by a three-dimensional transformation specifying the location of the object relative to the camera. Object detection systems construct a model for an object class from a set of training examples. In the case of a fixed rigid object only one example may be needed, but more generally multiple training examples are necessary to capture certain aspects of class variability.

Object detection methods fall into two major categories, generative and discriminative. The first consists of a probability model for the pose variability of the objects together with an appearance model: a probability model for the image appearance conditional on a given pose, together with a model for background, i.e. non-object images. The model parameters can be estimated from training data and the decisions are based on ratios of posterior probabilities. The second typically builds a classifier that can discriminate between images (or sub-images) containing the object and those not containing the object. The parameters of the classifier are selected to minimize mistakes on the training data, often with a

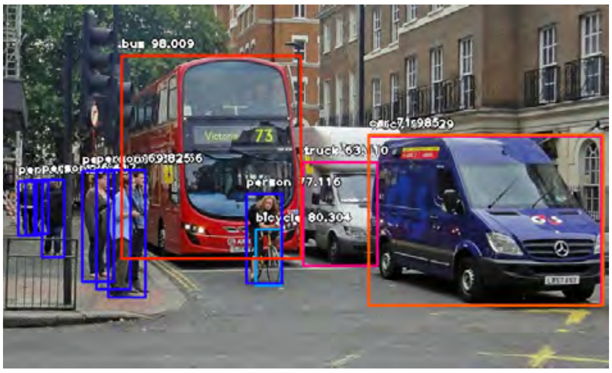

Figure 2.2: Object detection with bounding boxes.

regularization bias to avoid overfitting. Other distinctions among detection algorithms have to do with the computational tools used to scan the entire image or search over possible poses, the type of image representation with which the models are constructed, and what type and how much training data is required to build a model.

### 2.1.3 Semantic Segmentation

Segmentation is essential for image analysis tasks. Semantic segmentation describes the process of associating each pixel of an image with a class label, (such as flower, person, road, sky, ocean, or car). Semantic image segmentation can be applied effectively to any task that involves the segmentation of visual information. Examples include road segmentation for autonomous vehicles, medical image segmentation, scene segmentation for robot perception, and in image editing tools. Whilst currently available systems provide accurate object recognition, they are unable to delineate the boundaries between objects with the same accuracy.

Oxford researchers have developed a novel neural network component for semantic segmentation that enhances the ability to recognise and delineate objects. This invention can be applied to improve any situation requiring the segmentation of visual information.

Semantic image segmentation plays a crucial role in image understanding, allowing a computer to recognise objects in images. Recognition and delineation of objects is achieved through classification of each pixel in an image. Such processes have a wide range of applications in computer vision, in diverse and growing fields such as vehicle autonomy and medical imaging.

The previous state-of-the-art image segmentation systems used Fully Convolutional Neural Network (FCNN) components, which offer excellent accuracy in recognising objects. Whilst this development represented a significant improvement in semantic segmentation, these networks do not perform well in delineating object boundaries. Conditional Random Fields (CRFs) can be employed in a post-processing step to improve object boundary delineation, however, this is not an optimum solution owing to a lack of integration with the deep network. Oxford researchers have developed a neural network component for semantic segmentation that harnesses the exceptional object recognition of FCNNs and the powerful boundary delineation of CRFs. CRFs are fully integrated as recurrent neural networks, resulting in a

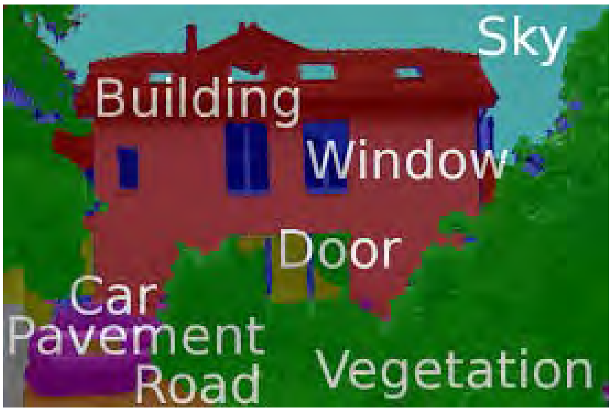

Figure 2.3: Image with semantic segmentation.

system that offers enhanced performance compared to the previous state-of-the-art. The novel system can be applied to any task that involves the segmentation of visual information. Examples include road segmentation for autonomous vehicles, medical image segmentation, scene segmentation for robot perception, and in image editing tools. Oxford University Innovation is seeking industrial partners that wish to explore the use of this system for commercial applications.

### 2.1.4 Instance Segmentation

Instance segmentation is one step ahead of semantic segmentation wherein along with pixel level classification, we expect the computer to classify each instance of a class separately. For example in the image above there are 3 people, technically 3 instances of the class "Person". All the 3 are classified separately (in a different color). But semantic segmentation does not differentiate between the instances of a particular class.

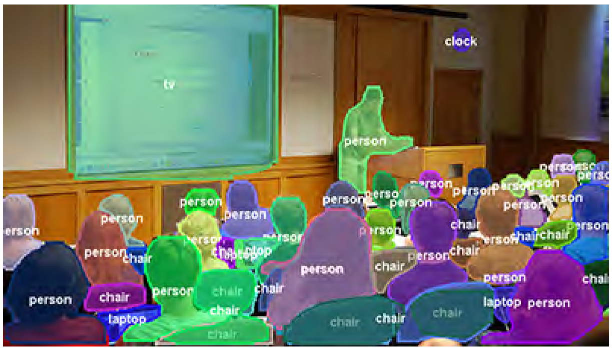

Figure 2.4: Image with instance segmentation.

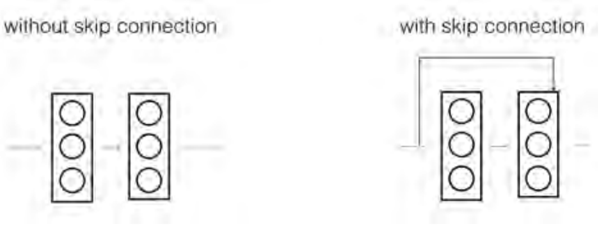

Figure 2.5: Skip connection of ResNet.

## 2.2 CNN for Object Detection and Segmentation

### 2.2.1 Backbone

**Residual Networks (RESNET)**

ResNet is an essential neural network that serves as a backbone to Mask R-CNN and numerous computer vision tasks. It makes the training of extremely deep neural networks possible which was very difficult before then due to the challenge of vanishing gradients, that hampers convergence in the network. According to [M] before RESNET, the problem of *vanishing gradient* has been mainly addressed by normalized initialization [N, O, P, Q] and intermediate normalization layers [R], which enable networks with tens of layers to start converging for stochastic gradient descent (SGD) with backpropagation [S]. Looking at the sample scenario of the vanishing gradients or degradation. A worst-case scenario of vanishing gradient is the case was the early layers of a deeper model can be replaced with a shallow network and the other layers can act as an identity function. The shallow network and its deeper counterpart give the same accuracy. So deeper models do not perform well due to degradation. When a deeper network is used it approximates the mapping than its shallower variant and decreases the error by a notable margin. Also, the deeper network had issues of degradation. To solve this problem ResNet introduced the concept of skip connection.

Without a skip connection, deep convolution networks are stacked together one after the other. With a skip connection deep convolution networks are tacked together but this time the original input is added to the output of the convolution block. Mathematically representing this, we can consider a mapping or space G(x) to be fitted by some stacked layers of an entire network. X denotes the inputs in the first layer of the net. This layer will approximate a residual function Z(x) = G(x)-x by hypothesizing. Therefore the original function or mapping G(x) becomes Z(x) + x. The input and output dimensions are expected to be of the same dimension for this work properly. It is worth noting that ResNet, contained 152 layers, won ILSVRC 2015 with an error rate of 3.6 percent beating even humans with their error rate of circa 5 – 10 percent, and replacing VGG-16 layers in Faster R-CNN with ResNet-101 produced relative improvements of 28 percent. It also trained networks with 100 layers and 1000 layers.

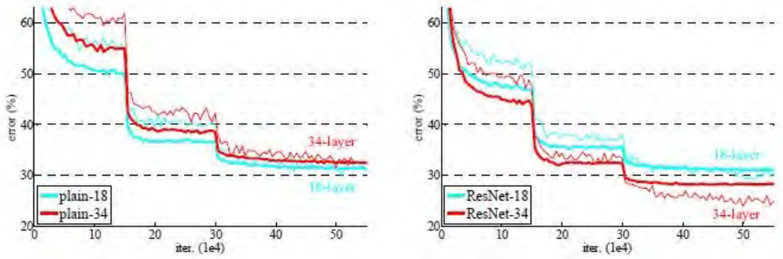

Figure 2.6: Training on ImageNet. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts [M].

### Feature Pyramid Network

Feature Pyramid Network (FPN) is a generic feature extractor used in various application for recognizing objects at different scales, developed by Lin et al [K]. For the recognition system for detecting objects at various scales, feature pyramid is a primary constituent of such a system. Pyramid representation has the problem of computing and memory intensiveness and has been avoided in the deep learning object detectors. Feature Pyramid Network (FPN) solves this problem by restructuring the architecture of the pyramid to a top-down architecture with lateral connections. The multi-scale, pyramidal hierarchy of deep ConvNet was leveraged to develop Feature Pyramid Network (FPN). Pyramids of FPN are scale-invariant. This means that when an object scale changes it is offset by shifting its level in the pyramid. Before the introduction of FPN, some ways were used in the extraction of features from images. Initially, hand-engineered features [U] were used and it makes use of featurized image pyramids. ConvNet is more robust to variance in scale, capable of representing higher-level semantics, and so features from it have quickly replaced engineered features. According to [L] this ConvNets gives multi-scale feature representation in which all levels are semantically strong, including the high-resolution levels. Featuring each level of an image pyramid comes with the profound limitation of increase in inference time which makes it impractical for real applications. FPN explored the pyramidal shape of a ConvNet's feature hierarchy to build a feature pyramid that has strong features with high-resolution at all scales. This gives a feature pyramid that has profound semantics at all phases and is constructed quickly from a unit input image scale

FPN was applied in Regional Proposal Network (RPN) and Fast R-CNN. With the new adaptations, RPN could be naturally trained and tested with FPN, Using FPN in a basic Faster R-CNN system, the result surpasses all existing single-model entries including those from the COCO 2016 challenge winners.

## 2.2.2 Regional Proposal Network

Regional Proposal Network (RPN) is a useful network effectively used in R-CNN that scans the image in a sliding window pattern over the anchors. It proposes multiple objects that are recognizable in a

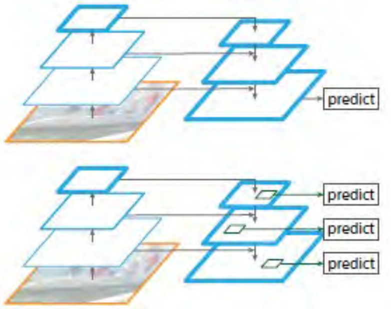

Figure 2.7: Top: a top-down architecture with skip connections, where predictions are made on the finest level (e.g., [T]). Bottom: FPN model that has a similar structure but leverages it as a feature pyramid, with predictions made independently at all levels [K].

particular image. The last convolutional layer that is produced by the Faster R-CNN is called the feature map. A proposal is generated for the region where the object lies by sliding over a feature map a network, which is the RPN. RPN suggest where an object lies in an image.

Analyzing the architecture of RPN, the intermediate layer divides into a classifier and regressor layers, and the concept of the anchor was introduced. Anchor are boxes of different sizes and aspect ratio that are generated over an image that determines the ideal location, shape, and size of objects in the image. They overlap to fill up as much of the image as possible. Thousands of anchor boxes are generated for this. For each anchor box, the object's bounding box that has the highest overlap is divided by non-overlap. This is termed Intersection Over Union (IOU). If the highest IOU is greater than 50 percent, the anchor box determines the object that gave the highest IOU. But if it is greater than 40 percent the true detection is ambiguous, and if it is less than 40 percent it predicts no object. Classifier gives the probability of a proposal containing the target object. Regression regresses the coordinates of

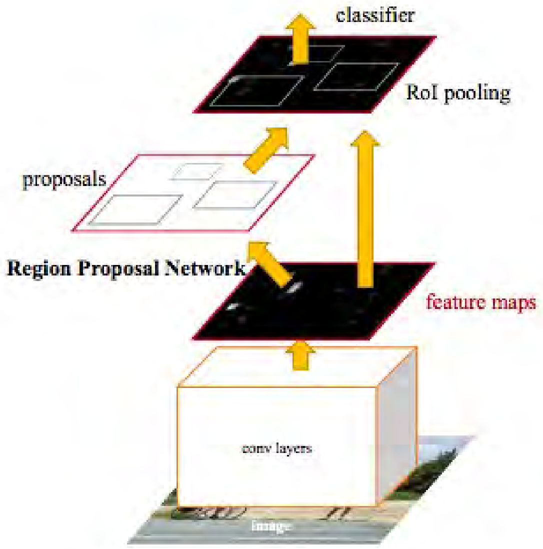

Figure 2.8: The architecture of Faster R-CNN. RPN generate the proposal for the objects. [J].

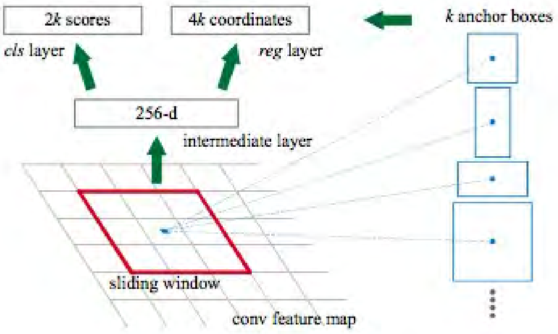

Figure 2.9: RPN Architecture [J].

the proposals. RPN is also used in Mask RCNN.

The RPN produces two results for each anchor; Anchor Class i.e. either the foreground or the background, and Bounding Box Refinement, an estimation to rectify the anchor box that fits the object well. This is a change in x,y, width, height.

### 2.2.3 ROI Classifier and Bounding Box Regressor

Region of Interest (ROI) classifier is proposed by the Region Proposal Network (RPN) and similarly, like the RPN, it produces two results for each ROI. First, the *class*, it produces the classes of the object in the ROI, but it is deeper and can classify regions to specific classes (car, person, nucleus, etc). And secondly the *Bounding Box Refinement*, which further refines the location and size of the bounding box to envelope the object.

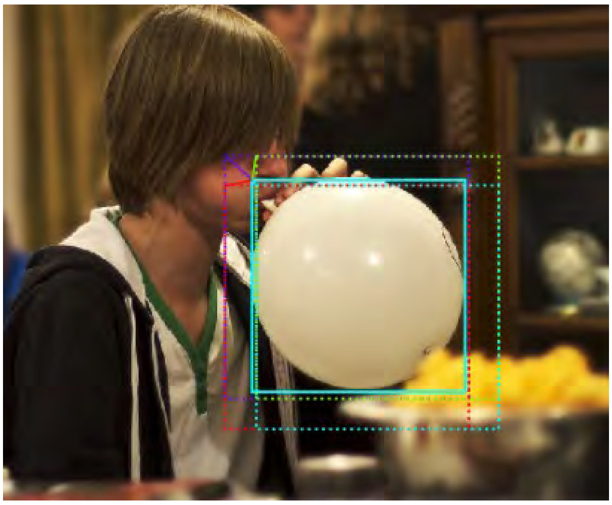

Figure 2.10: Anchor boxes (dotted) and the shift/scale applied to them to fit the object precisely (solid). Several anchors can map to the same object.

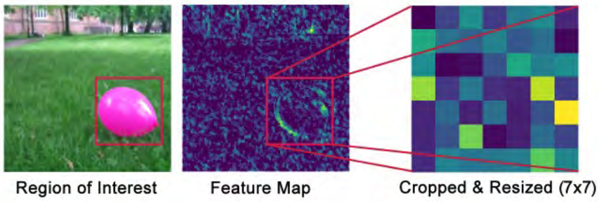

Figure 2.11: IROI Pooling.

### ROI Pooling

Input sizes of a classifier vary. Classifiers require fixed, stable input size and can't manage varying input sizes. Bounding box refinement in RPN produces ROI boxes of various sizes. ROI Pooling tackles this challenge. Cropping a part of a feature map and resizing it to a fixed size is termed ROI pooling. It is very much alike to the concept of resizing a cropped image.

## 3. Chapter

# Results and analysis

## 3.1 The travelling salesman problem

bla

bla

### 3.1.1 Heuristics

etc, the rest is in Hungarian.

## 4. Chapter

# Matlab and Netlab tutorial

***Summary:*** *In the summary again we are explaining what is going to happen: this chapter originally had a short description of Matlab and its accompanying NETLAB package.*

*Once more, the rest of the chapter is in Hungarian; important is the logics of the SOURCE file...*

## 4.1 Programming summary

# A. Appendix

# Principal program codes

We use the package **listings** , see [Heinz and Moses, 2007], that is well written and documented.

If space is important, one could use:

```
\lstinputlisting[multicols=2]{progfiles/ex_prg.py}
```

```python
# coding: utf-8
import numpy as np
import scipy.io as spio
import matplotlib.pyplot as plt


# reading in the data using PANDAS
# import pandas as pd
# data = pd.read_csv("weightHeight.dat",header=None,sep=’\s+’)
# PANDAS is an EXCEL-like processing... the "sep=..." specifies
# a regular expression that accepts multiple spaces

# this is a second alternative using numpy
data = np.loadtxt("weightHeight.dat")

# getting the INPUTS - x0 -- the heights, and the outputs y0 the WEIGHTS
x0    = data[:,1].reshape( (-1,1) )
y0    = data[:,0].reshape( (-1,1) )
# visualising the data
plt.scatter(x0, y0)

# to enrich the regression function, we define
#
# a FEATURE GENERATOR -- example of a function
def PHI(X):
    nData = X.shape[0]
    return np.concatenate(
         ( np.ones( (nData,1) ), X, X ** 2, X ** 3, X ** 4 ),
           axis = -1
         )

### PERFORMING REGRESSION
X  = PHI(x0)
Y = y0
# solving the linear system
theta = np.linalg.lstsq(X , Y, rcond=None)[0]


xMin, xMax = min(x0), max(x0)
allX = np.linspace(xMin-20,xMax+20,50).reshape( (-1,1) )
allY = PHI(allX) @ theta

# second visuaisation
plt.plot( allX, allY,’r--’)
```

# Bibliography

C. Heinz and B. Moses. The listings package. Technical report, CTAN TEX Archive, 2007. URL http://www.ctan.org/tex-archive/macros/latex/contrib/listings.