nodeflux

Extending Vision Beyond Imagination

Introduction to Object Detection Algorithm from Image

Introduction

Common Visual Recognition Tasks

Classification

What objects are contained in the image?

= Cat

= Dog

= ? % Dog
? % Person

Common Visual Recognition Tasks

Classification

Applications

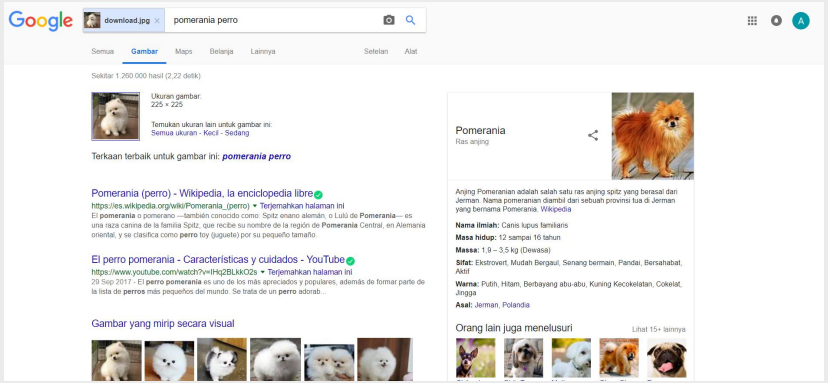
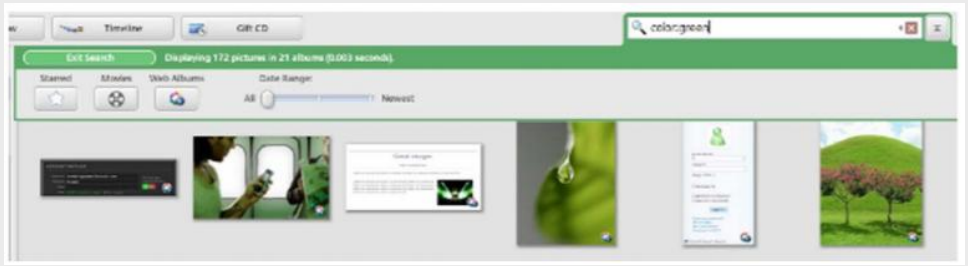


Image Search



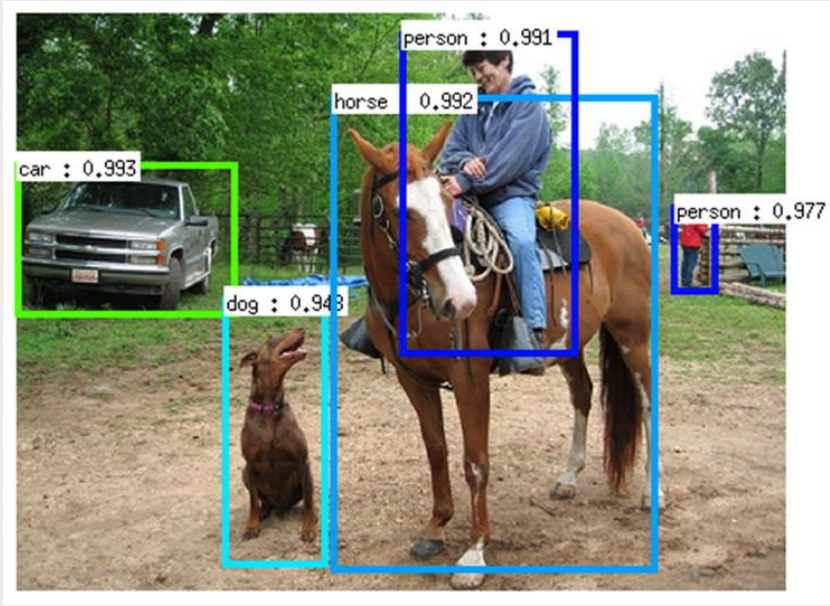
Organizing Photo Collections

Common Visual Recognition Tasks

Detection

What objects are contained in the image? +

Where



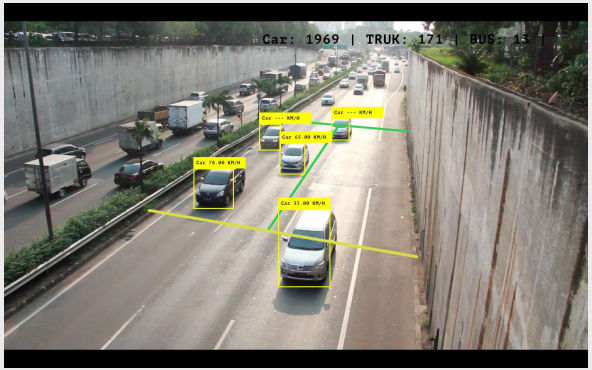
Object Detection (bounding box)

Object Detection (pixel-based segmentation)

Common Visual Recognition Tasks

Detection

Applications



Defense and Security

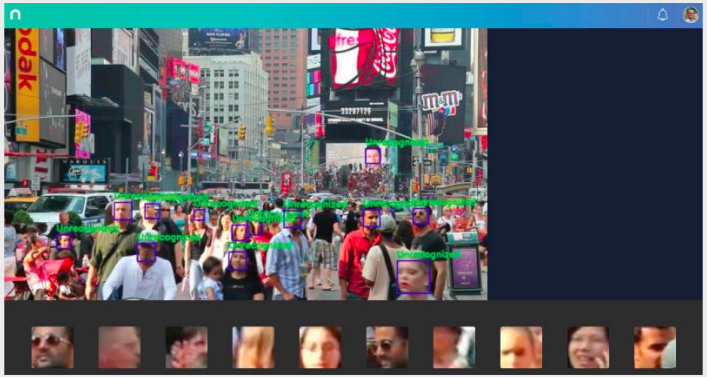
- Face Recognition
- 2D to 3D Face Reconstruction
- Time Compression Analysis
- Pixel Enhancement
- License Plate Recognition
- Crowd Behaviour Analysis

Smart City

- Traffic Monitoring
- Road and River Monitoring
- Flood Monitoring
- Vehicle Detection
- Illegal Parking Detection
- Dynamic Traffic Lights

Store Analytics

- Visitor Counting
- Visitor Trajectory Flow
- Visitor Heat Map
- Product View Rank
- Queue Analysis



Algorithm

Object Detection Key Components

- An algorithm to inspect parts of images, e.g. : sliding window, region proposal
- Obtain extracted features (image patterns) from the inspected parts , e.g. : using CNN
- Classify them whether they’re an object or not using machine learning : using SVM, Fully Connected

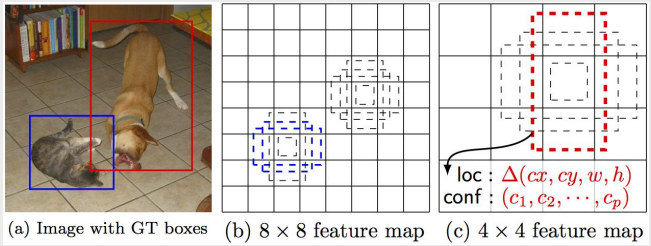
First Step of Object Detection

Where should we tell the computer to inspect whether there are objects in it?

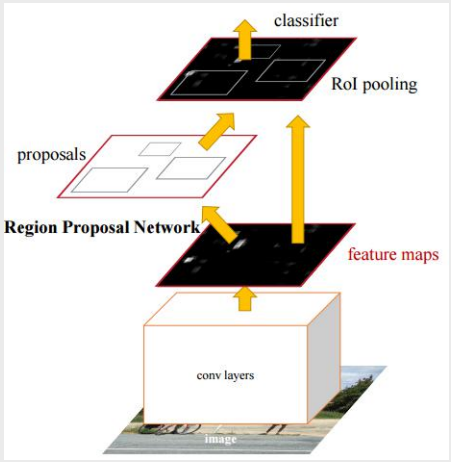
Traditional (Old Style)

Sliding Window + Image Pyramid

State-of-the-Art Style



Feature Maps
Anchor Boxes

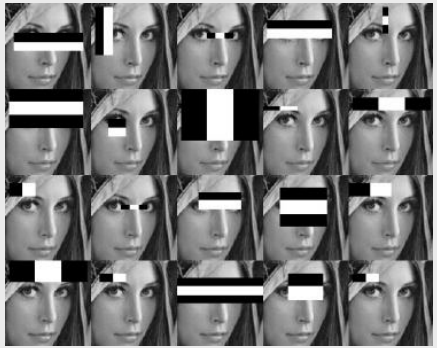


Region Proposal Network

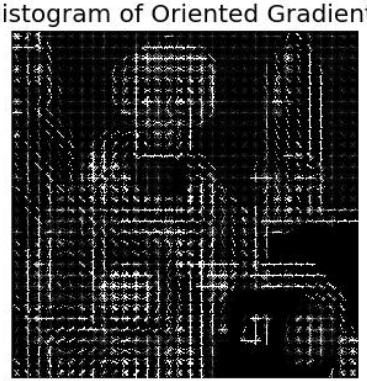
Second Step of Object Detection

What is the object's pattern?

Traditional (Old Style)



HAAR Features

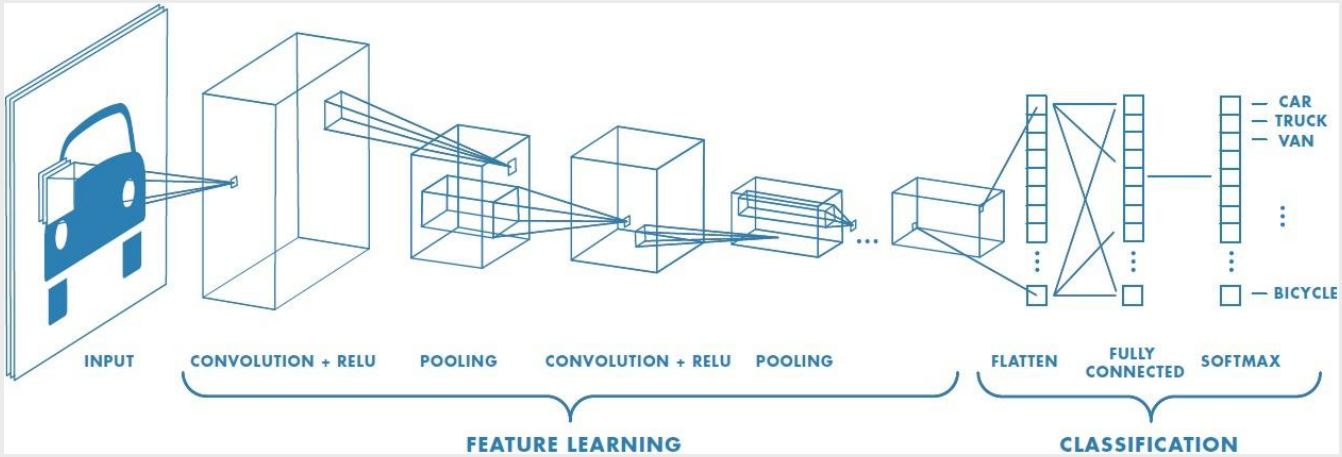


Histogram of Oriented Gradients

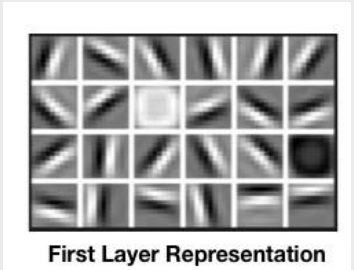
Second Step of Object Detection

What is the object's pattern?

State-of-the-Art Style



Convolutional Neural Network

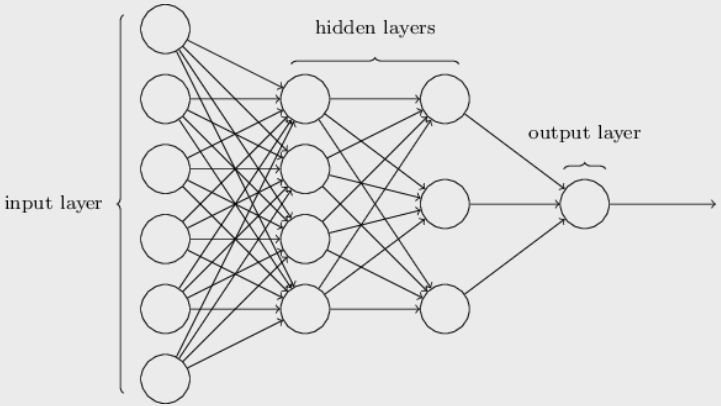

First Layer Representation


Second Layer Representation

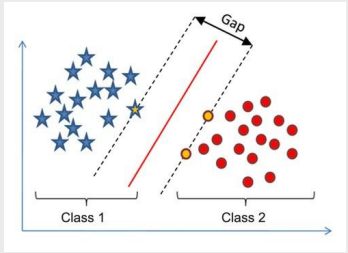

Third Layer Representation

Final Step of Object Detection

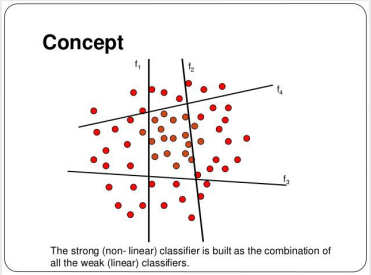
How do we classify the object?



Fully Connected Neural Network



SVM



Adaboost

Machine Learning

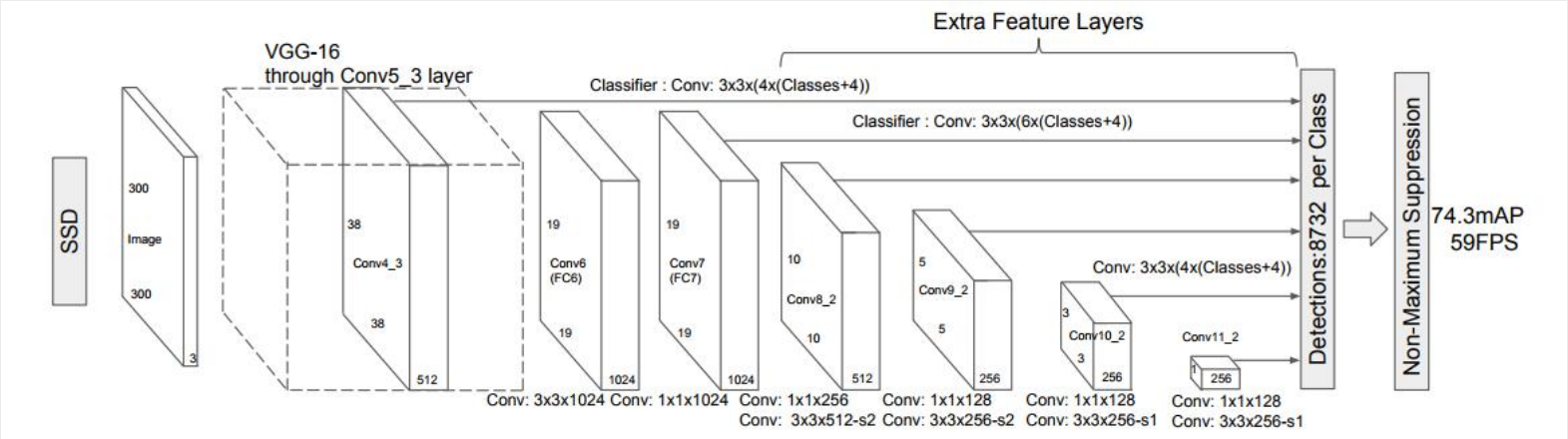
Introduction to Singleshot MultiBox Detector

nodeflux
Extending Vision Beyond Imagination

Singleshot Multibox Detector

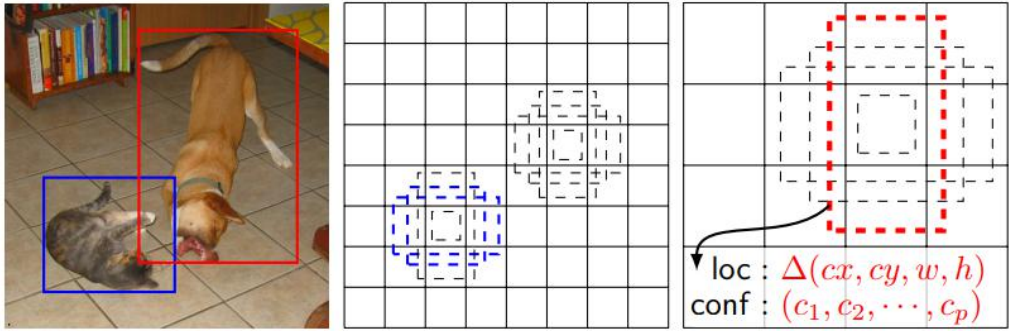
- Proposed by Wei Liu et al. in their paper ‘SSD: Single Shot MultiBox Detector’, presented at ECCV 2016
- The term came from these reasons:
  - Single Shot : The localization and classification tasks will be completed only with single forward pass of the network
  - Multibox : a bounding box regression technique which can adapt to multi-scale object
  - Detector : this framework will detect and classify object presented in an image

Architecture



- Consist of *base architecture* like VGG16, Mobilenet, ResNet,..
- Substituting the last fully connected layer with several *auxiliary* convolutional layers to extract feature maps at multiple scale (inspired by Multibox work)

Anchor Boxes



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

- SSD predict the presence of objects using feature maps extracted from *base architecture*
- Each feature maps is split into multiple fixed size *cells*
- In each *cells*, SSD use a pre-computed fixed size default boxes that matched closely with the distribution of the ground-truth boxes from training data which are called *priors* or *anchors*
- These *anchors* will be regressed to match the ground truth bounding box to perform detection

Losses

- *Confidence Loss* : this loss is used to calculate how confidence is the network to present that an area is containing any object in it. This loss is calculated using categorical cross-entropy
- *Location Loss* : this loss is a calculated smooth L1 loss to present how far the predicted bounding box coordinates from the ground truth
- *Combined Loss* : the overall combined loss is a weighted sum over the confidence and location loss. The alpha is a hyper-parameter which measure how much the contribution of the location loss

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

alvinprayuda@nodeflux.io