# Fast R-CNN

# Drawbacks of R-CNN

Generate Region Proposals — **2000 Proposed Regions**

↓

Feature Extraction using CNN — **4096 Features**

↓

Classification and Localization — **SVM and Ridge Regression**

# Fast R-CNN

## Fast R-CNN

Ross Girshick
Microsoft Research
rbg@microsoft.com

### Abstract

*This paper proposes a Fast Region-based Convolutional Network method* (Fast R-CNN) *for object detection. Fast R-CNN builds on previous work to efficiently classify object proposals using deep convolutional networks. Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. Fast R-CNN trains the very deep VGG16 network 9× faster than R-CNN, is 213× faster at test-time, and achieves a higher mAP on PASCAL VOC 2012. Compared to SPPnet, Fast R-CNN trains VGG16 3× faster, tests 10× faster, and is more accurate. Fast R-CNN is implemented in Python and C++ (using Caffe) and is*

while achieving top accuracy on PASCAL VOC 2012 [7] with a mAP of 66% (vs. 62% for R-CNN).[1]

### 1.1. R-CNN and SPPnet

The Region-based Convolutional Network method (R-CNN) [9] achieves excellent object detection accuracy by using a deep ConvNet to classify object proposals. R-CNN, however, has notable drawbacks:

1. **Training is a multi-stage pipeline.** R-CNN first fine-tunes a ConvNet on object proposals using log loss. Then, it fits SVMs to ConvNet features. These SVMs act as object detectors, replacing the softmax classifier learnt by fine-tuning. In the third training stage,

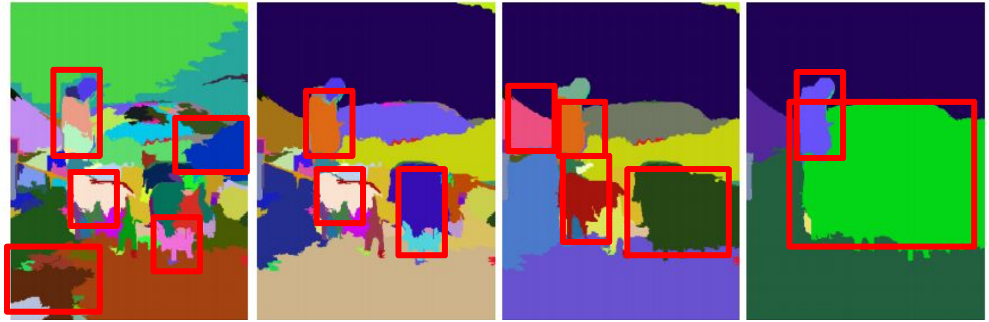
Analytics Vidhya

# Fast R-CNN - Architectural Changes

- Feature Extraction on the complete Image

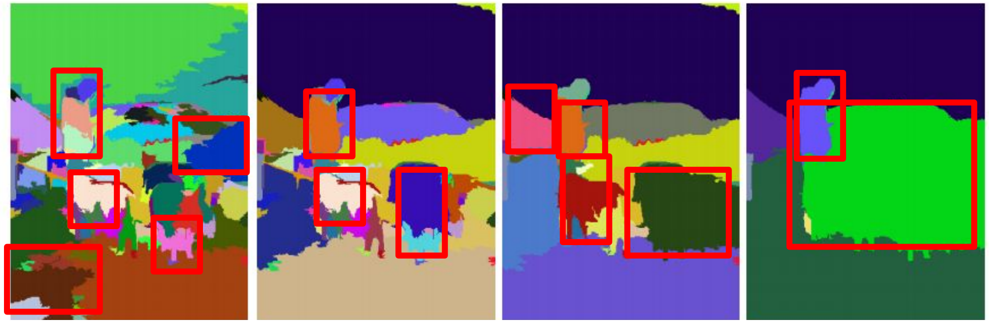
# Fast R-CNN

Generate Region Proposals



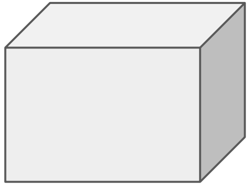
# Fast R-CNN

Generate Region Proposals



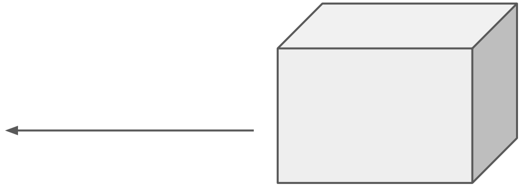
Input Image

Feature Extraction for Image

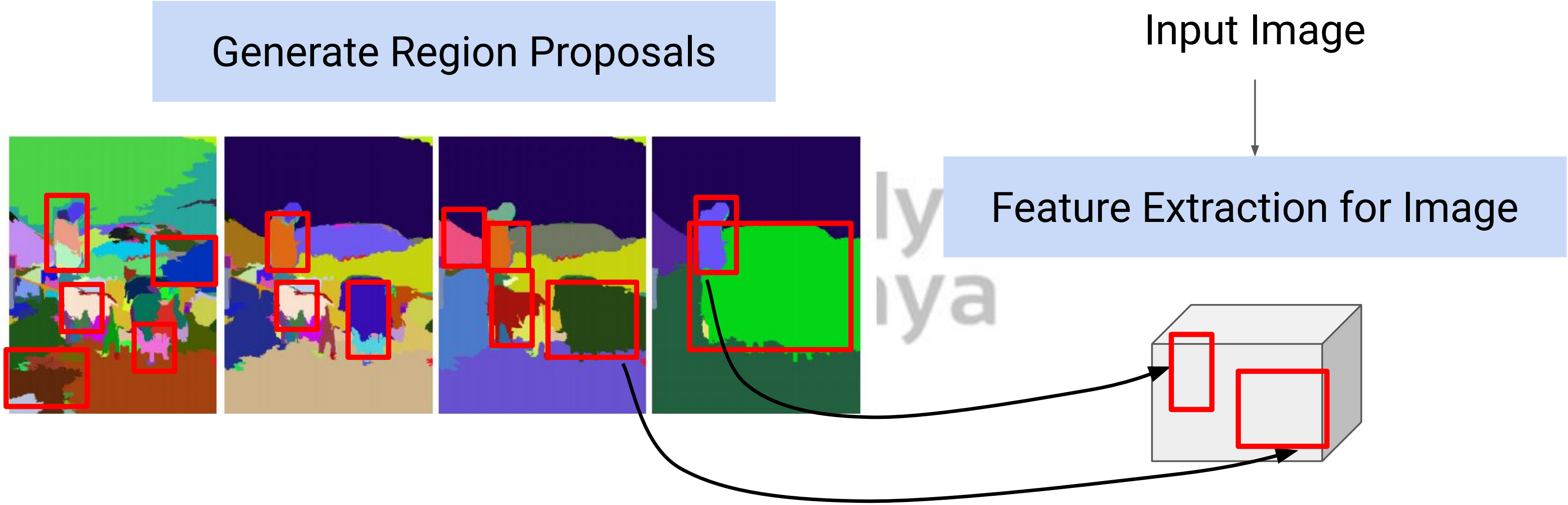


# Feature Extraction in Fast R-CNN

| Layer | # filters | Filter size | Stride | Padding | Size of feature map | Activation function |
|---|---|---|---|---|---|---|
| Input | - | - | | | 224 X 224 X 3 | |
| Conv 1 | 64 | 3X3 | 1 | 1 | 224 X 224 X 64 | ReLU |
| Conv 2 | 64 | 3X3 | 1 | 1 | 224 X 224 X 64 | ReLU |
| Max Pooling 1 | - | 2X2 | 2 | | 112 X 112 X 64 | |
| Conv 3 | 128 | 3X3 | 1 | 1 | 112 X 112 X 128 | ReLU |
| Conv 4 | 128 | 3X3 | 1 | 1 | 112 X 112 X 128 | ReLU |
| Max Pooling 2 | - | 2X2 | 2 | | 56 X 56 X 128 | |
| Conv 5 | 256 | 3X3 | 1 | 1 | 56 X 56 X 256 | ReLU |
| Conv 6 | 256 | 3X3 | 1 | 1 | 56 X 56 X 256 | ReLU |
| Conv 7 | 256 | 3X3 | 1 | 1 | 56 X 56 X 256 | ReLU |
| Max Pooling 3 | - | 2X2 | 2 | | 28 X 28 X 256 | |
| Conv 8 | 512 | 3X3 | 1 | 1 | 28 X 28 X 512 | ReLU |
| Conv 9 | 512 | 3X3 | 1 | 1 | 28 X 28 X 512 | ReLU |
| Conv 10 | 512 | 3X3 | 1 | 1 | 28 X 28 X 512 | ReLU |
| Max Pooling 4 | - | 2X2 | 2 | | 14 X 14 X 512 | |
| Conv 11 | 512 | 3X3 | 1 | 1 | 14 X 14 X 512 | ReLU |
| Conv 12 | 512 | 3X3 | 1 | 1 | 14 X 14 X 512 | ReLU |
| Conv 13 | 512 | 3X3 | 1 | 1 | 14 X 14 X 512 | ReLU |
| Max Pooling 5 | - | 2X2 | 2 | | 7 X 7 X 512 | |
| Fully Connected 1 | | | | | 4096 | ReLU |
| Fully Connected 2 | | | | | 4094 | ReLU |



# Fast R-CNN


Generate Region Proposals
Input Image
Feature Extraction for Image

# Fast R-CNN

| Layer | # filters | Filter size | Stride | Padding | Size of feature map | Activation function |
|---|---|---|---|---|---|---|
| Input | - | - | | | 224 X 224 X 3 | |
| Conv 1 | 64 | 3X3 | 1 | 1 | 224 X 224 X 64 | ReLU |
| Conv 2 | 64 | 3X3 | 1 | 1 | 224 X 224 X 64 | ReLU |
| Max Pooling 1 | - | 2X2 | 2 | | 112 X 112 X 64 | |
| Conv 3 | 128 | 3X3 | 1 | 1 | 112 X 112 X 128 | ReLU |
| Conv 4 | 128 | 3X3 | 1 | 1 | 112 X 112 X 128 | ReLU |
| Max Pooling 2 | - | 2X2 | 2 | | 56 X 56 X 128 | |
| Conv 5 | 256 | 3X3 | 1 | 1 | 56 X 56 X 256 | ReLU |
| Conv 6 | 256 | 3X3 | 1 | 1 | 56 X 56 X 256 | ReLU |
| Conv 7 | 256 | 3X3 | 1 | 1 | 56 X 56 X 256 | ReLU |
| Max Pooling 3 | - | 2X2 | 2 | | 28 X 28 X 256 | |
| Conv 8 | 512 | 3X3 | 1 | 1 | 28 X 28 X 512 | ReLU |
| Conv 9 | 512 | 3X3 | 1 | 1 | 28 X 28 X 512 | ReLU |
| Conv 10 | 512 | 3X3 | 1 | 1 | 28 X 28 X 512 | ReLU |
| Max Pooling 4 | - | 2X2 | 2 | | 14 X 14 X 512 | |
| Conv 11 | 512 | 3X3 | 1 | 1 | 14 X 14 X 512 | ReLU |
| Conv 12 | 512 | 3X3 | 1 | 1 | 14 X 14 X 512 | ReLU |
| Conv 13 | 512 | 3X3 | 1 | 1 | 14 X 14 X 512 | ReLU |
| Max Pooling 5 | - | 2X2 | 2 | | 7 X 7 X 512 | |
| Fully Connected 1 | | | | | 4096 | ReLU |
| Fully Connected 2 | | | | | 4094 | ReLU |

ROI pooling

# Fast R-CNN - Architectural Changes

- Feature Extraction on the complete Image
- Use RoI (Region of Interest) Pooling

# Region of Interest Pooling - Fast R-CNN

Max Pooling Operation

| | | | |
|---|---|---|---|
| 5 | 4 | 4 | 7 |
| 7 | 6 | 6 | 10 |
| 9 | 5 | 6 | 8 |
| 7 | 6 | 1 | 5 |

# Region of Interest Pooling - Fast R-CNN

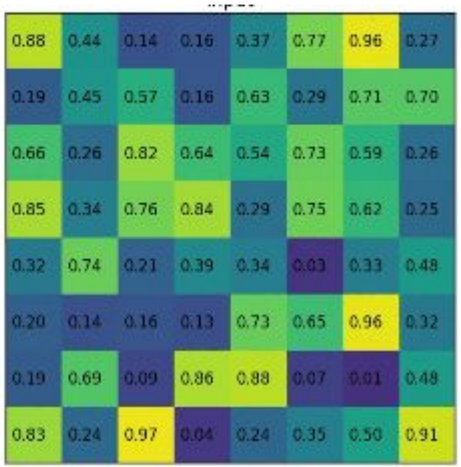
Max Pooling Operation

| | | | |
|---|---|---|---|
| 5 | 4 | 4 | 7 |
| 7 | 6 | 6 | 10 |
| 9 | 5 | 6 | 8 |
| 7 | 6 | 1 | 5 |

| | |
|---|---|
| 7 | 10 |
| 9 | 8 |

# Region of Interest Pooling - Fast R-CNN

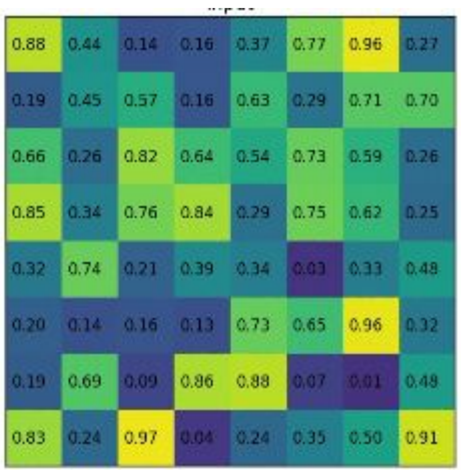
Region of Interest (RoI) Pooling



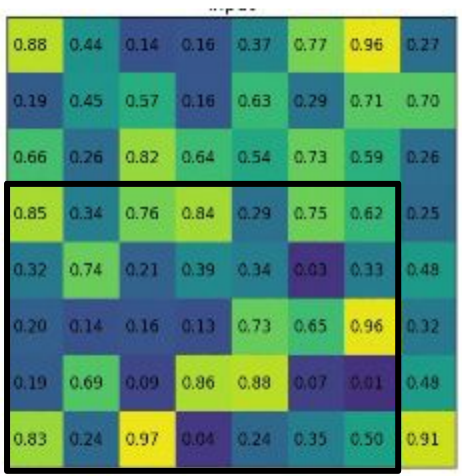

Feature Map

# Region of Interest Pooling - Fast R-CNN

## Region of Interest (RoI) Pooling



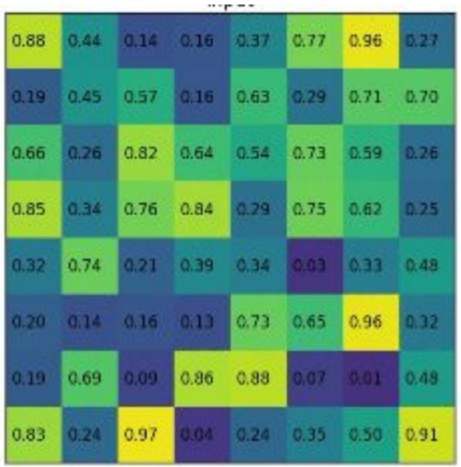

Feature Map



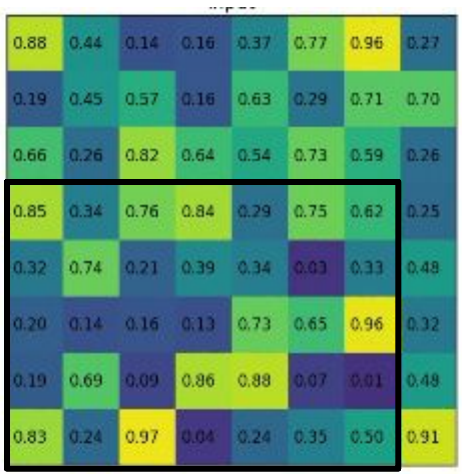

Region of Interest

# Region of Interest Pooling - Fast R-CNN

## Region of Interest (RoI) Pooling




Feature Map



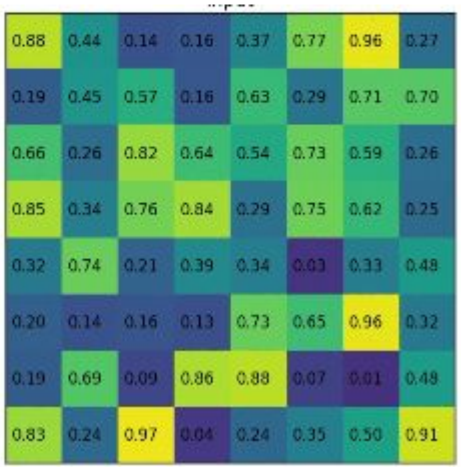

Region of Interest

Output shape 2x2

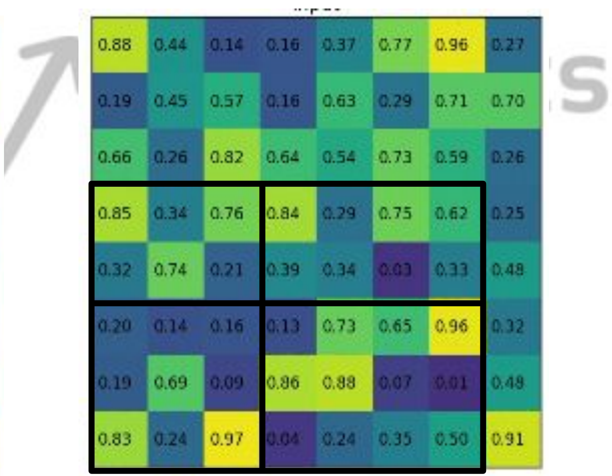
Analytics Vidhya

# Region of Interest Pooling - Fast R-CNN

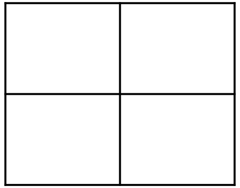
## Region of Interest (RoI) Pooling




Feature Map

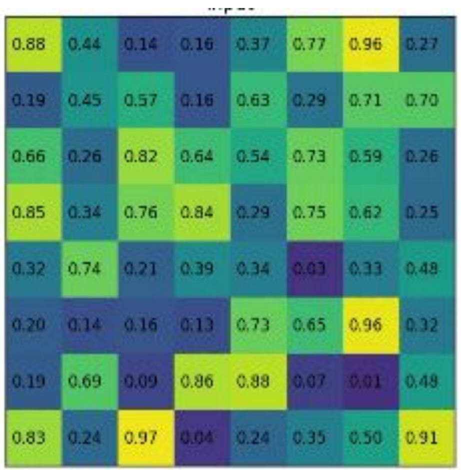



Region of Interest

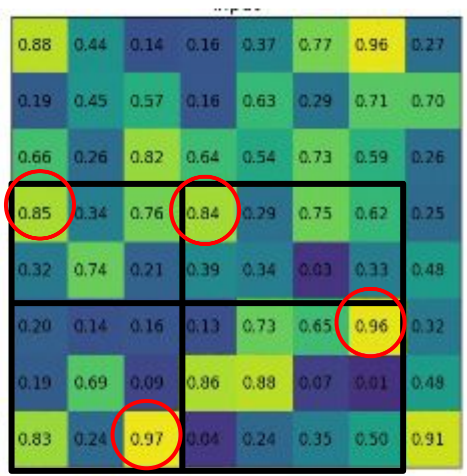


Output shape 2x2

# Region of Interest Pooling - Fast R-CNN




Feature Map

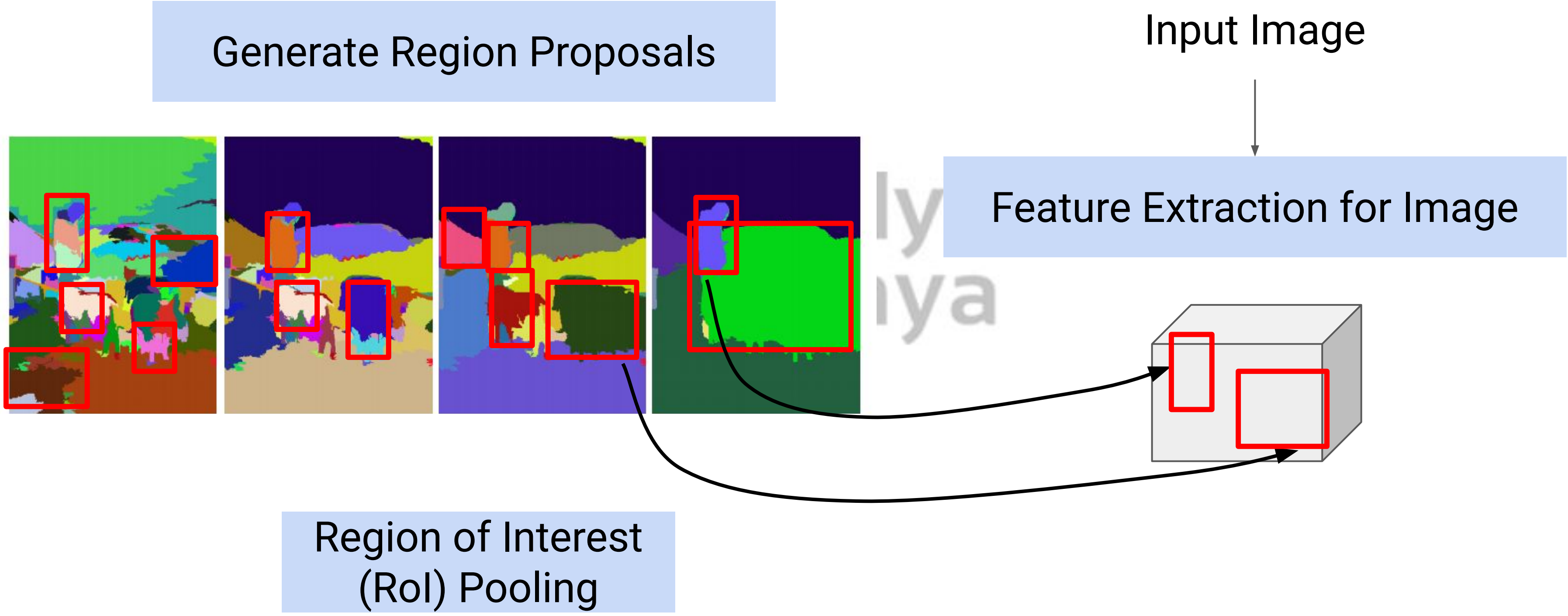



Region of Interest

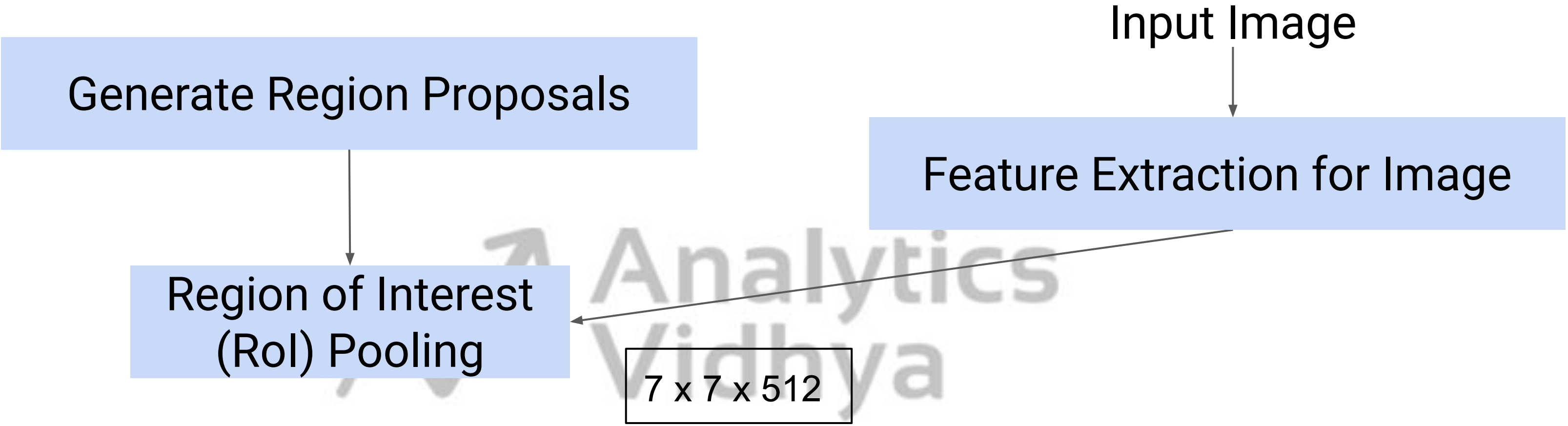
| | |
|---|---|
| 0.85 | 0.84 |
| 0.97 | 0.96 |

Output shape 2x2

# Fast R-CNN


Generate Region Proposals
Input Image
Feature Extraction for Image
Region of Interest
(RoI) Pooling

# Fast R-CNN

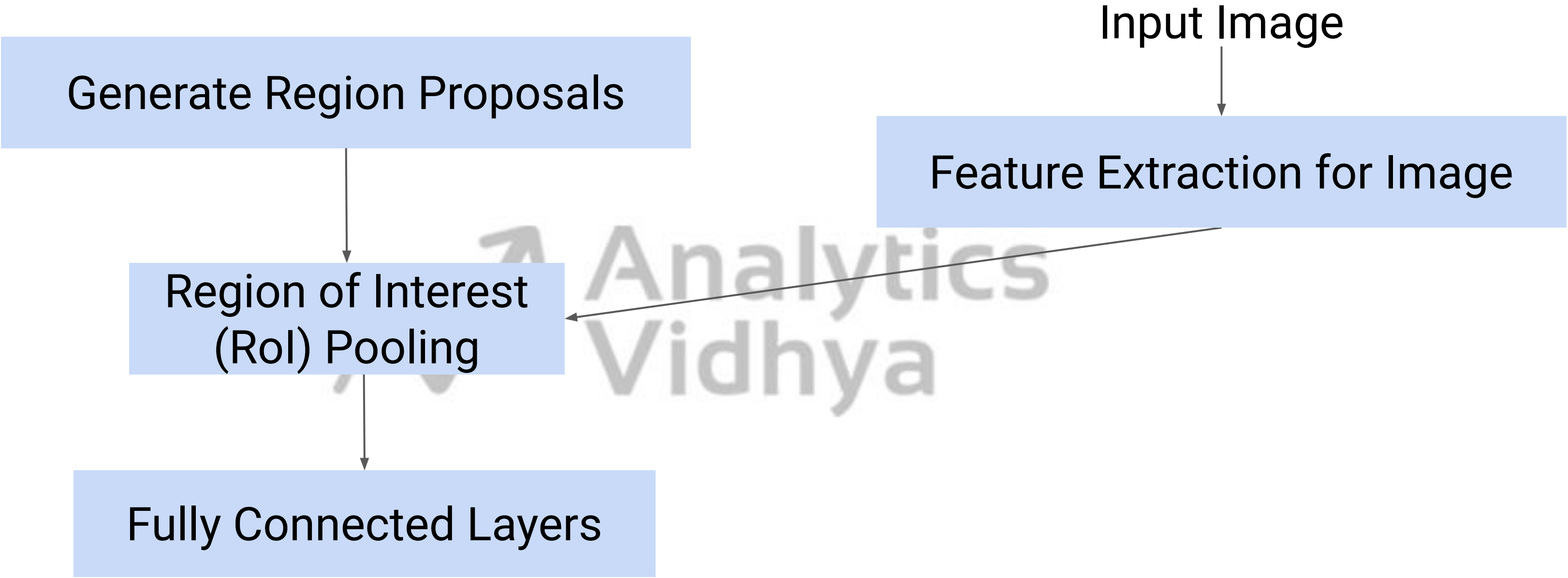


Input Image
Generate Region Proposals
Feature Extraction for Image
Region of Interest
(RoI) Pooling
7 x 7 x 512

# Fast R-CNN - Architecture

| Layer | # filters | Filter size | Stride | Padding | Size of feature map | Activation function |
|---|---|---|---|---|---|---|
| Input | - | - | | | 224 X 224 X 3 | |
| Conv 1 | 64 | 3X3 | 1 | 1 | 224 X 224 X 64 | ReLU |
| Conv 2 | 64 | 3X3 | 1 | 1 | 224 X 224 X 64 | ReLU |
| Max Pooling 1 | - | 2X2 | 2 | | 112 X 112 X 64 | |
| Conv 3 | 128 | 3X3 | 1 | 1 | 112 X 112 X 128 | ReLU |
| Conv 4 | 128 | 3X3 | 1 | 1 | 112 X 112 X 128 | ReLU |
| Max Pooling 2 | - | 2X2 | 2 | | 56 X 56 X 128 | |
| Conv 5 | 256 | 3X3 | 1 | 1 | 56 X 56 X 256 | ReLU |
| Conv 6 | 256 | 3X3 | 1 | 1 | 56 X 56 X 256 | ReLU |
| Conv 7 | 256 | 3X3 | 1 | 1 | 56 X 56 X 256 | ReLU |
| Max Pooling 3 | - | 2X2 | 2 | | 28 X 28 X 256 | |
| Conv 8 | 512 | 3X3 | 1 | 1 | 28 X 28 X 512 | ReLU |
| Conv 9 | 512 | 3X3 | 1 | 1 | 28 X 28 X 512 | ReLU |
| Conv 10 | 512 | 3X3 | 1 | 1 | 28 X 28 X 512 | ReLU |
| Max Pooling 4 | - | 2X2 | 2 | | 14 X 14 X 512 | |
| Conv 11 | 512 | 3X3 | 1 | 1 | 14 X 14 X 512 | ReLU |
| Conv 12 | 512 | 3X3 | 1 | 1 | 14 X 14 X 512 | ReLU |
| Conv 13 | 512 | 3X3 | 1 | 1 | 14 X 14 X 512 | ReLU |
| Max Pooling 5 | - | 2X2 | 2 | | 7 X 7 X 512 | |
| Fully Connected 1 | | | | | 4096 | ReLU |
| Fully Connected 2 | | | | | 4094 | ReLU |

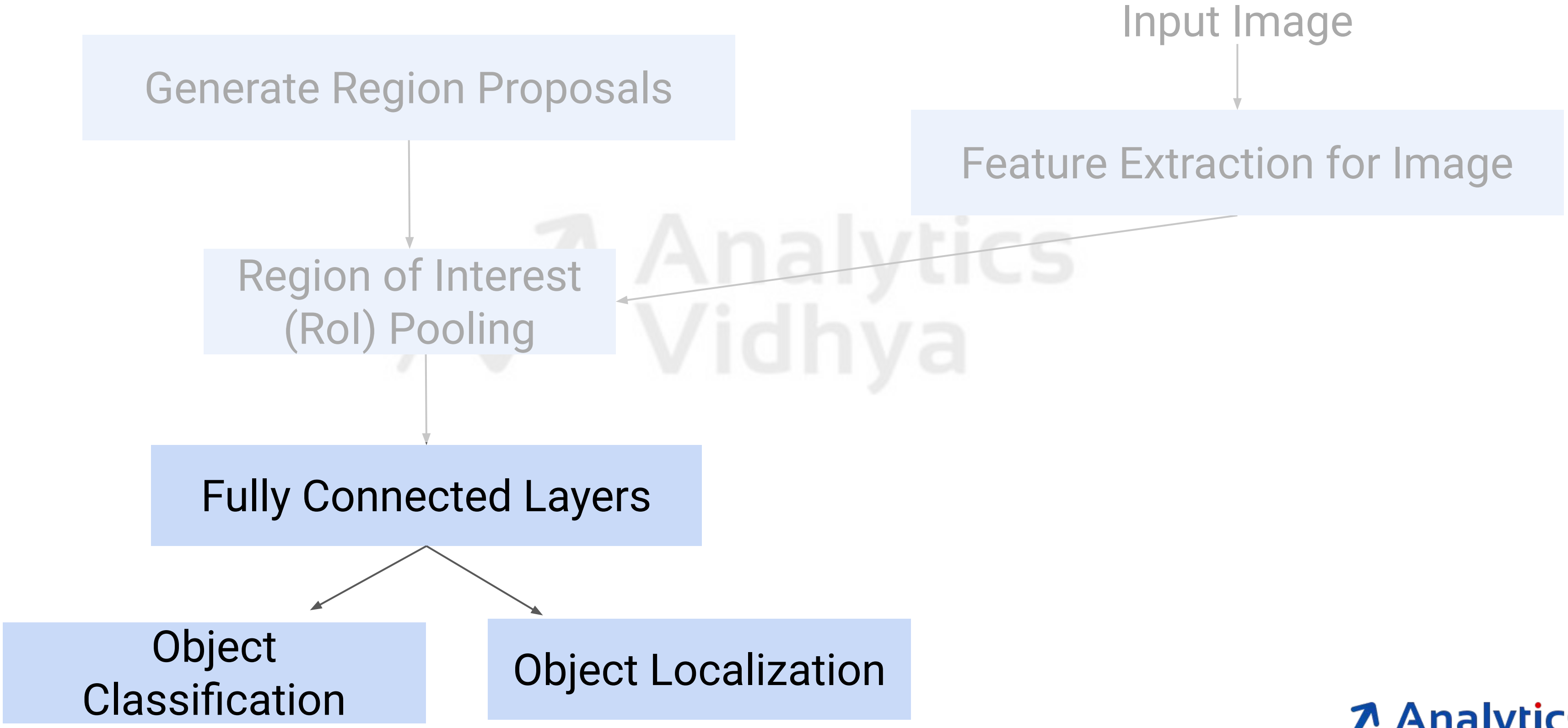
# Fast R-CNN


Input Image
Generate Region Proposals
Feature Extraction for Image
Region of Interest
(RoI) Pooling
Fully Connected Layers

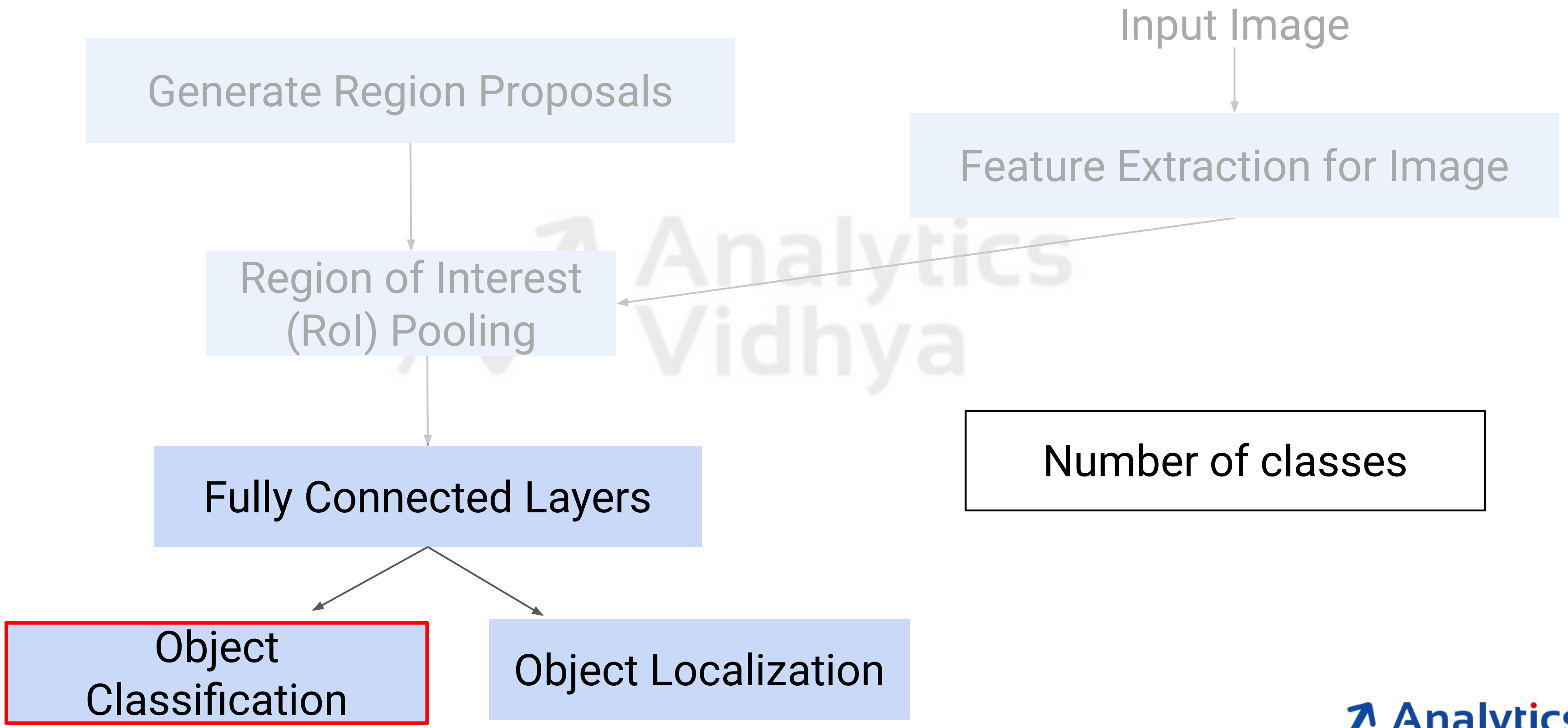
# Fast R-CNN - Architectural Changes

- Feature Extraction on the complete Image
- Use RoI (Region of Interest) Pooling
- Use Dense Layers instead of SVM and Ridge Regression

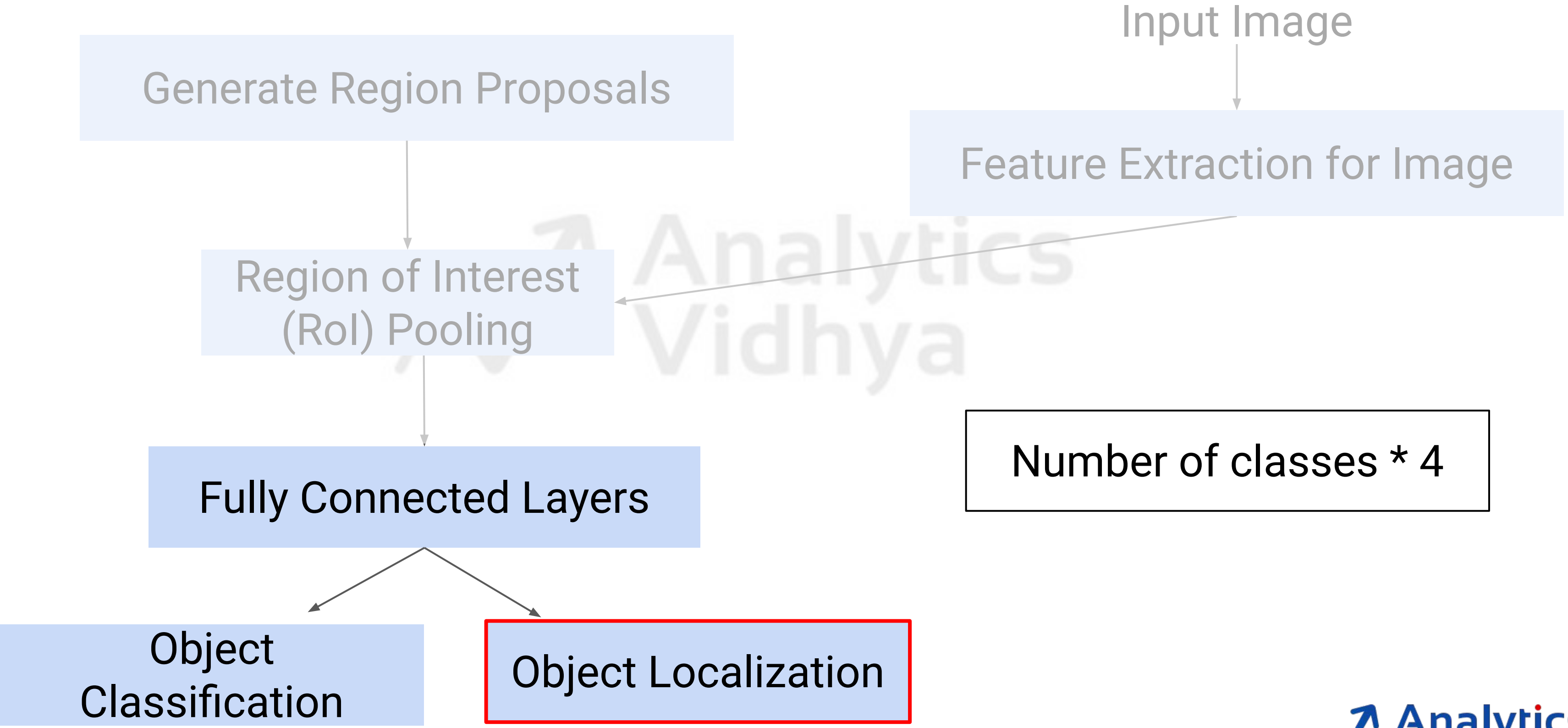
# Fast R-CNN


Input Image
Generate Region Proposals
Feature Extraction for Image
Region of Interest (RoI) Pooling
Fully Connected Layers
Object Classification
Object Localization

# Fast R-CNN


Input Image
Generate Region Proposals
Feature Extraction for Image
Region of Interest (RoI) Pooling
Fully Connected Layers
Number of classes
Object Classification
Object Localization

# Fast R-CNN


Input Image
Generate Region Proposals
Feature Extraction for Image
Region of Interest (RoI) Pooling
Fully Connected Layers
Number of classes * 4
Object Classification
Object Localization

# Advantages of Fast R-CNN (over R-CNN)

- Significantly faster than RCNN

Inference time RCNN
**~47 seconds**

Inference time Fast RCNN
**~ 2.3 seconds**

# Advantages of Fast R-CNN (over R-CNN)

- Significantly faster than RCNN

| Inference time RCNN<br>**~47 seconds** | Inference time Fast RCNN<br>**~ 2.3 seconds** |
| --- | --- |

- Feature Extraction process for complete image

# Can we Make fast R-CNN faster?

- Extracting 2000 Regions per image is time consuming
- Alternative to selective search?

Thank You