



Department of Computer Science, Shahid Beheshti University

MSc computer science, Data Mining

Object Detection using R-CNN

Presented by Sara Charmchi

Student ID : 400422066

Advisor : Dr.Katanforoush

Artificial Neural Networks course 1400-2

Girshick, R., Donahue, J., Darrell, T., & Malik, J., Rich feature hierarchies for accurate object detection and semantic segmentation.
In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587). 2014.

TABLE OF CONTENTS

1. **INTRODUCTION** : Defining The Main Problem
2. **METHOD** : Region Proposal CNNs (R-CNN)
3. **DATA** : Ships in Satellite Imagery (San Francisco Bay and San Pedro Bay areas)
 - 4000 80x80 RGB labeled images : ship/no-ship
 - Each image includes : label, scene id, longitude_latitude
4. **METRIC** : Measures and Evaluation Metrics
5. **RESULT** : Preprocess Data, Generate Model, Test Model, Results
6. **REFERENCE**

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles, suggesting a hierarchical or central structure. The lines are thin and gray, connecting the nodes in a non-linear fashion.

1.

Introduction

Defining The Main Problem

What is Object Detection?

- Object Detection is a technology of deep learning, where things, human, building, cars can be detected as object in image and videos.
- **Object detection** is merely to recognize the object with bounding box in the image, where in **image classification**, we can simply categorize(classify) that is an object in the image or not in terms of the likelihood(Probability).

**Classification
+ Localization**



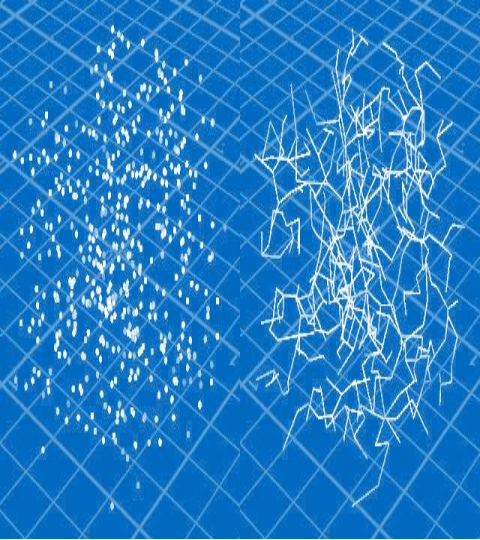
CAT

Single Object

**Object
Detection**



DOG, DOG, CAT



Why is it important?

fundamental problem in computer vision;

Tasks like instance segmentation, image captioning, object tracking, and more.

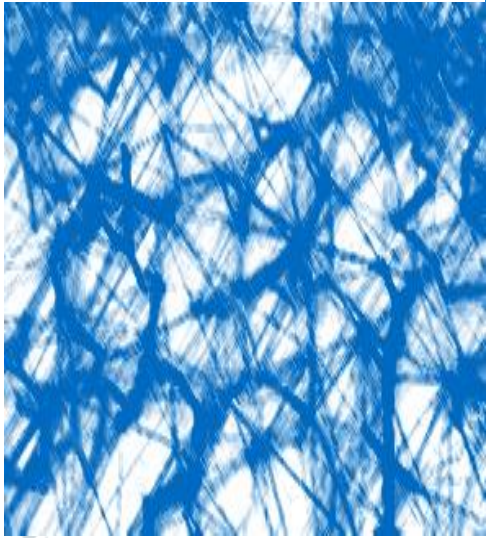


Object Detection Goal

- Object Location
- *Classification*

Real-World Applications

- Object detection in Retail
- Autonomous Driving
- Animal detection in Agriculture
- People detection in Security
- Medical feature detection in Healthcare,...



What are the applications?

- Pedestrian detection
- People counting
- Face detection
- Text detection
- Pose detection
- Number-plate recognition,...



Ship Detection

Ship detection from remote sensing imagery is a crucial application for maritime security like

- Traffic surveillance
- Protection against illegal fisheries
- Oil discharge control and
- Pollution monitoring.

How to detect ships ?
Automated Identification System (AIS) by using VHF radio frequencies

What's the Challenge?
Required connected VHF transponder

How to detect transponder disconnected ships?
Satellite imagery

How to detect by images?
Image processing using deep learning

Ship Detection

What are we looking for?

Bounding box

- Smallest box by small measure that fully contains the object in question
- Typically defined by :
 - top left corner (x,y)
 - wight w
 - height h
 - +classifier confidence



Object Detection as Classification

Suppose that we are given a region of interest, $ROI = (x, y, w, h)$, and asked to decide whether the ROI is an object. We can do this by training a neural network to estimate the classifier output:

$$y_c(ROI) = \begin{cases} 1 & \text{ROI contains an object} \\ 0 & \text{ROI does not contain an object} \end{cases}$$

A neural net trained with MSE or CE will then compute:

$$y_c(ROI) = \Pr(\text{ROI contains an object})$$

Back-prop to the individual pixels can show the degree to which each pixel contributes to the detection probability.

But, Real networks need to deal with situations of partial overlap : IOU

Intersection over union (IOU)

We deal with partial-overlap by putting some sort of threshold on the intersection-over-union measure. Suppose the hypothesis is $(X_{ROI}, Y_{ROI}, W_{ROI}, h_{ROI})$, and the reference is $((X_{REF}, Y_{REF}, W_{REF}, h_{REF}))$, then IOU is:

$$IOU = \frac{I}{U} = \frac{\text{number of pixels in both ROI and REF}}{\text{number of pixels in either ROI or REF}}$$

$$I = (\min(X_{REF} + W_{REF}, X_{ROI} + W_{ROI}) - \max(X_{REF}, X_{ROI})) \times (\min(Y_{REF} + h_{REF}, Y_{ROI} + h_{ROI}) - \max(Y_{REF}, Y_{ROI}))$$

$$U = W_{REF}h_{REF} - W_{ROI}h_{ROI} - I$$

We could use IOU as a soft-measure, or could we put some sort of arbitrary threshold, like:

$$y_c(ROI) = \Pr(IOU > 0.7)$$

SO, Why Object Detection is Hard ? Too Many Rectangles



2.

Method

Region Proposal CNNs (R-CNN)

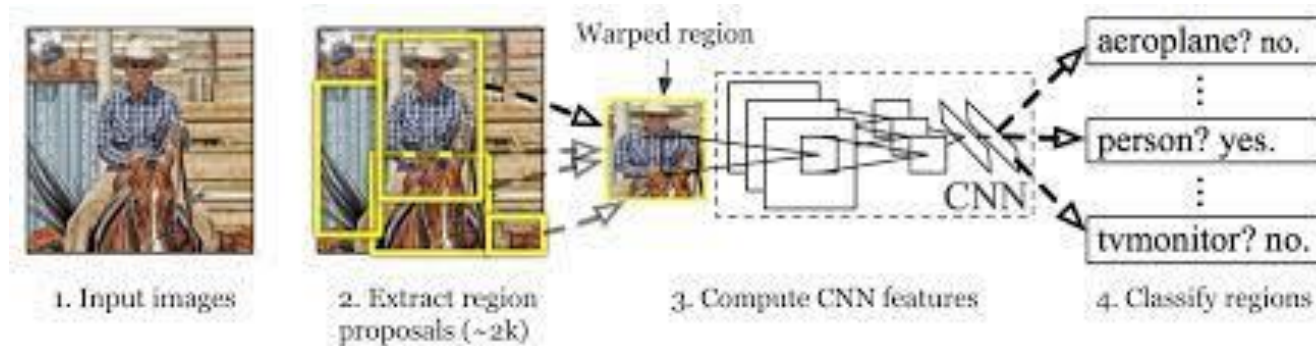
- CNNs are powerful classifiers
- FCN help to improve efficiency (still brute-force)
- Also different Scales and shapes

Can we improve efficiency by only considering interesting regions?

Find interesting regions first, then classify by CNN : Region Proposal CNN



Regional CNN (R-CNN)

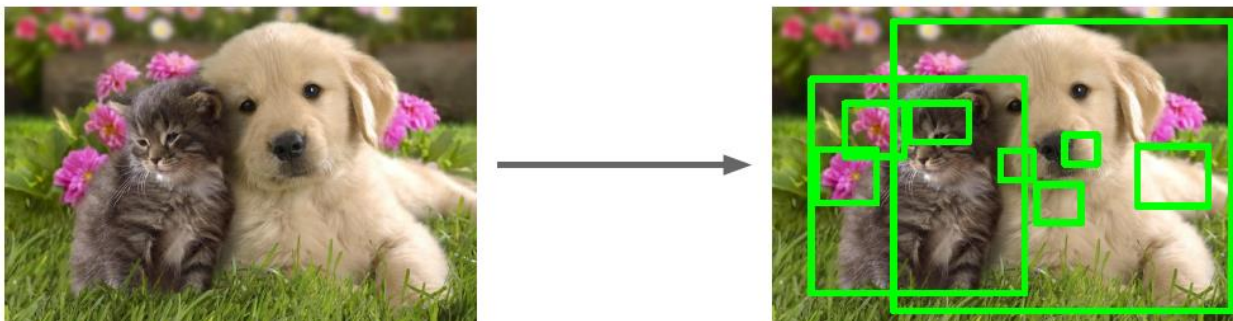


Multi-step approach in R-CNN:

- generate region proposals: selective search
- classify the content of the region proposals within a refined bounding box.

Regional Proposals

- candidate objects by a grouping of pixels with similar texture and color
- Apply for different sized windows



Produce few thousand (~2k) object proposals per image \ll number possible windows

Essentially a form of segmentation

Regional CNN (R-CNN)

Model in detail

Propose category-independent regions of interest by **selective search** (~2k candidates per image).

Those regions may contain target objects and they are of different sizes.



Regions of Interest (RoI) from a proposal method (~2k)

Regional Proposals

Selective search

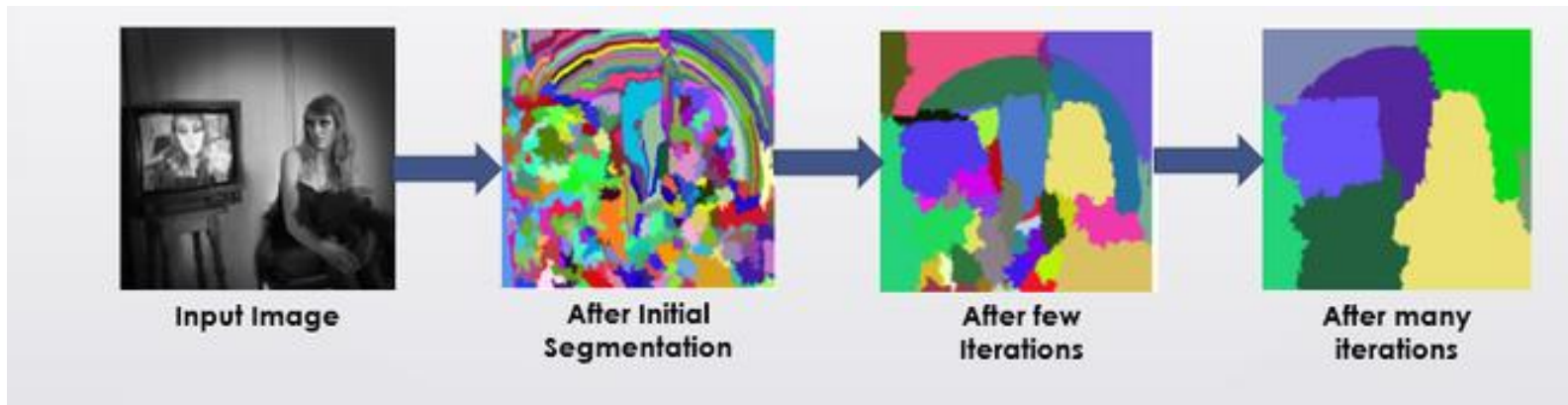
1. Generate initial sub-segmentation of input image using the method describe by *Felzenszwalb et al* in his paper “Efficient Graph-Based Image Segmentation”.



Regional Proposals

Selective search

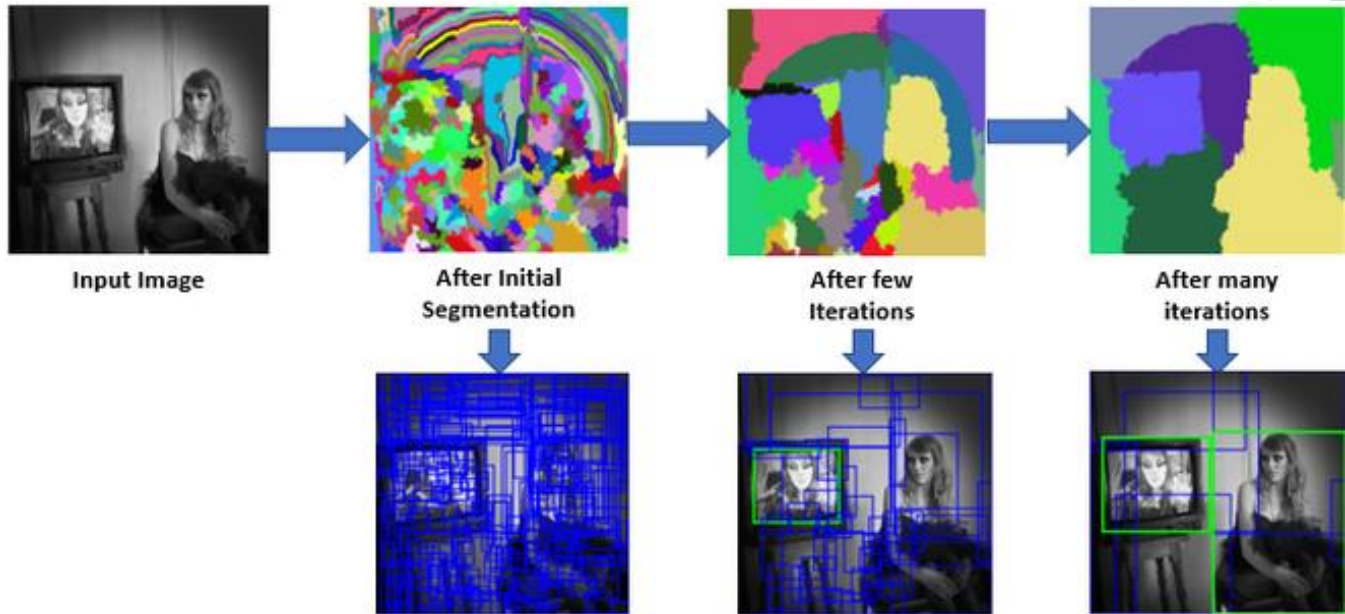
2. Recursively combine the smaller similar regions into larger ones. We use Greedy algorithm to combine similar regions to make larger regions.



Regional Proposals

Selective search

3. Use the segmented region proposals to generate candidate object locations.



Regional Proposals

Selective search

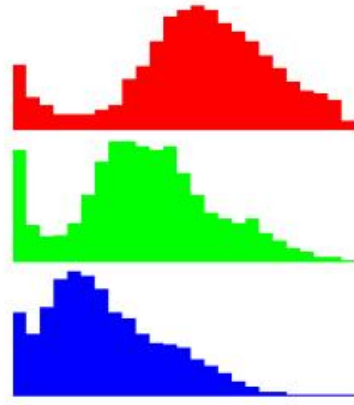
Similarity Metrics:

1. Color Similarity

Create a color histogram C for each channel in region r .

In the paper, 25 bins were used, for 75 total dimensions. We can measure similarity with histogram intersection:

$$s_{colour}(r_i, r_j) = \sum_{k=1}^n \min(c_i^k, c_j^k)$$



Regional Proposals

Selective search

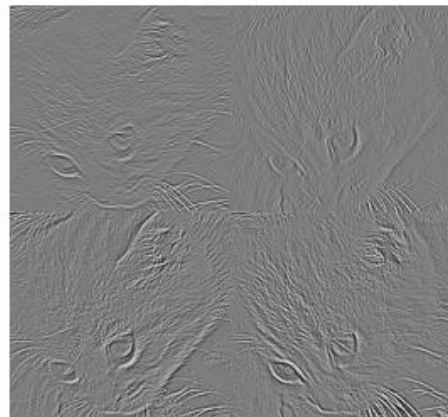
Similarity Metrics:

2.Texture Similarity :

Can measure textures with a HOG-like feature:

1. Extract gaussian derivatives of the image in 8 directions and for each channel.
2. Construct a 10-bin histogram for each, resulting in a 240-dimensional descriptor.

$$S_{\text{texture}}(\mathbf{r}_i, \mathbf{r}_j) = \sum_{k=1}^n \min(t_i^k, t_j^k)$$



Regional Proposals

Selective search

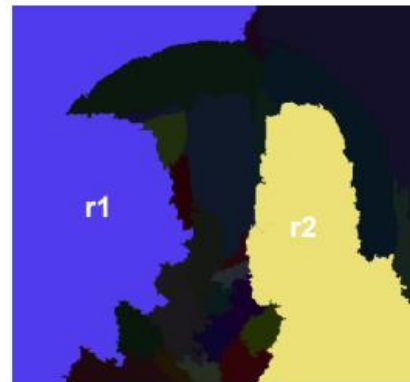
Similarity Metrics:

3. Size Similarity :

We want small regions to merge into larger ones, to create a balanced hierarchy.

Solution: Add a size component to our similarity metric, that ensures small regions are more similar to each other.

$$s_{size}(r_i, r_j) = 1 - \frac{\text{size}(r_i) + \text{size}(r_j)}{\text{size}(im)}$$



Regional Proposals

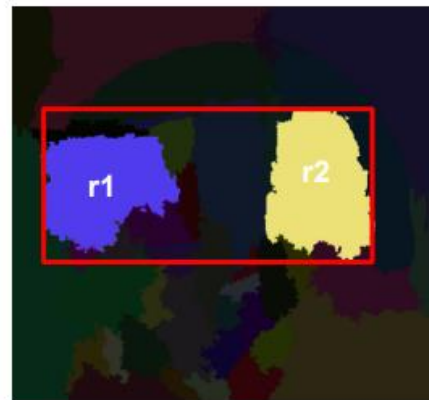
Selective search

Similarity Metrics:

4.Fill Similarity : Shape Compatibility

We also want our merged regions to be cohesive, so we can add a measure of how well two regions “fit together”.

$$fill(r_i, r_j) = 1 - \frac{size(BB_{ij}) - size(r_i) - size(r_j)}{size(im)}$$



Regional Proposals

Selective search

Final similarity metric

We measure the similarity between two patches as a linear combination of the four given metrics:

$$s(r_i, r_j) = a_1 s_{colour}(r_i, r_j) + a_2 s_{texture}(r_i, r_j) + a_3 s_{size}(r_i, r_j) + a_4 s_{fill}(r_i, r_j),$$

Then, we can create a diverse collection of region-merging strategies by considering different weighted combinations in different color spaces.

Regional Proposals

Selective search

Evaluation

To measure the performance of this method. The paper describes an evaluation parameter known as MABO (Mean Average Best Overlap).

called Average Best Overlap:

$$ABO = \frac{1}{|G^c|} \sum_{g_i^c \in G^c} \max_{l_j \in L} \text{Overlap}(g_i^c, l_j)$$

Overlap between ground truth and best selected box.

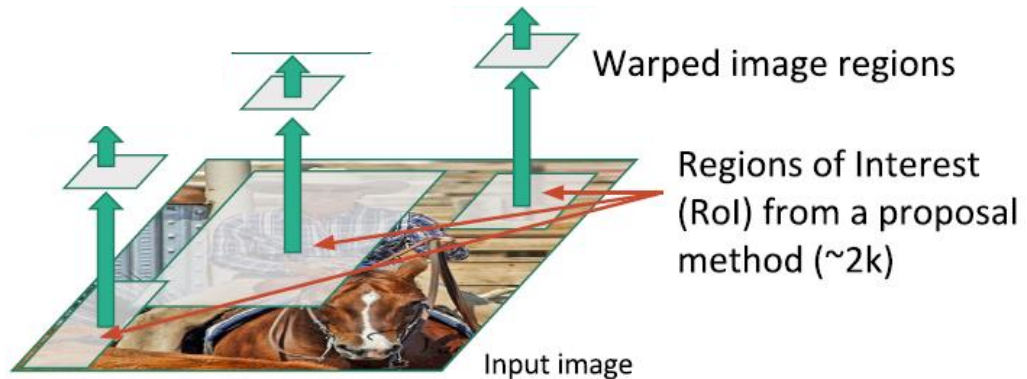
Average of "best overlaps" across all images.

There are two version of selective search came **Fast** and **Quality**.

Regional CNN (R-CNN)

For **each** region proposal window :

- Wrap to standard window size



Regional CNN (R-CNN)

For **each** region proposal window :

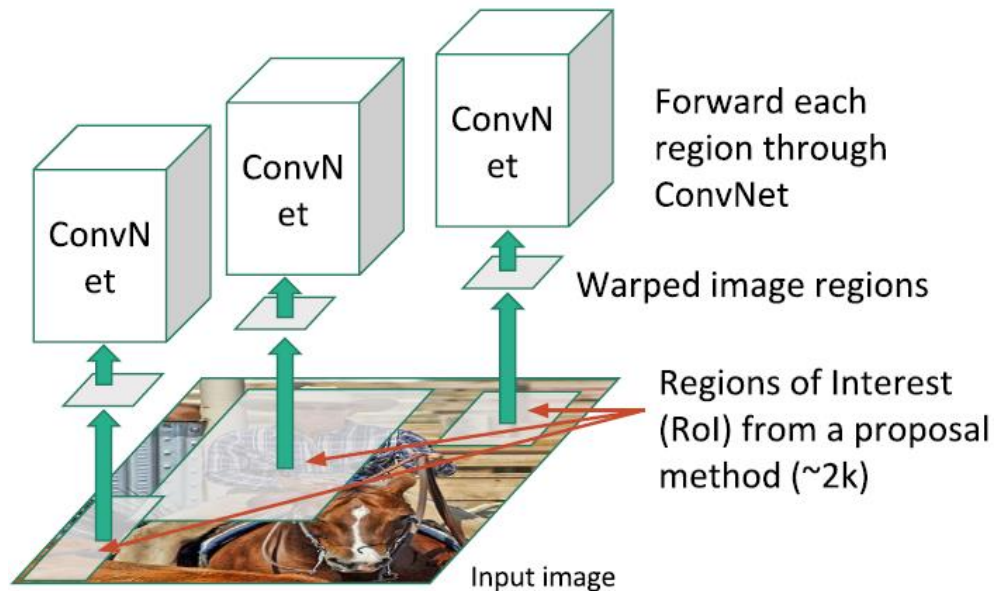
- Pre-trained CNN for feature extraction :

$K + 1$ classes

one class refers to the background (no object of interest).

In the fine-tuning stage:

use a much smaller learning rate and the mini-batch oversamples the positive cases because most proposed regions are just background.



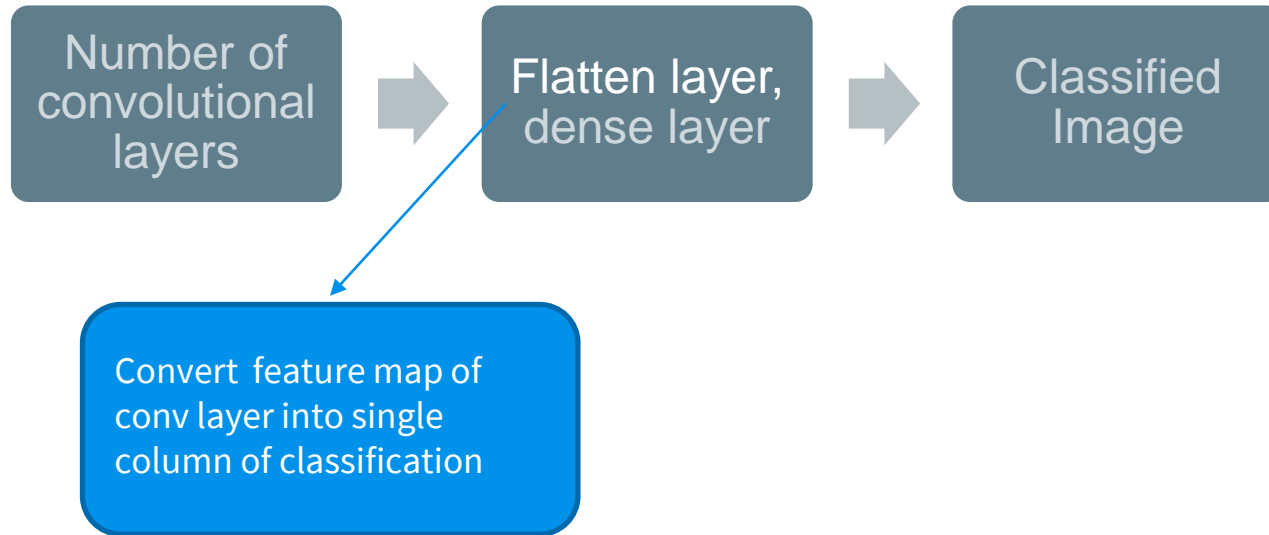
Model Creation

Pre-trained CNN for feature extraction



Model Creation

Pre-trained CNN for feature extraction



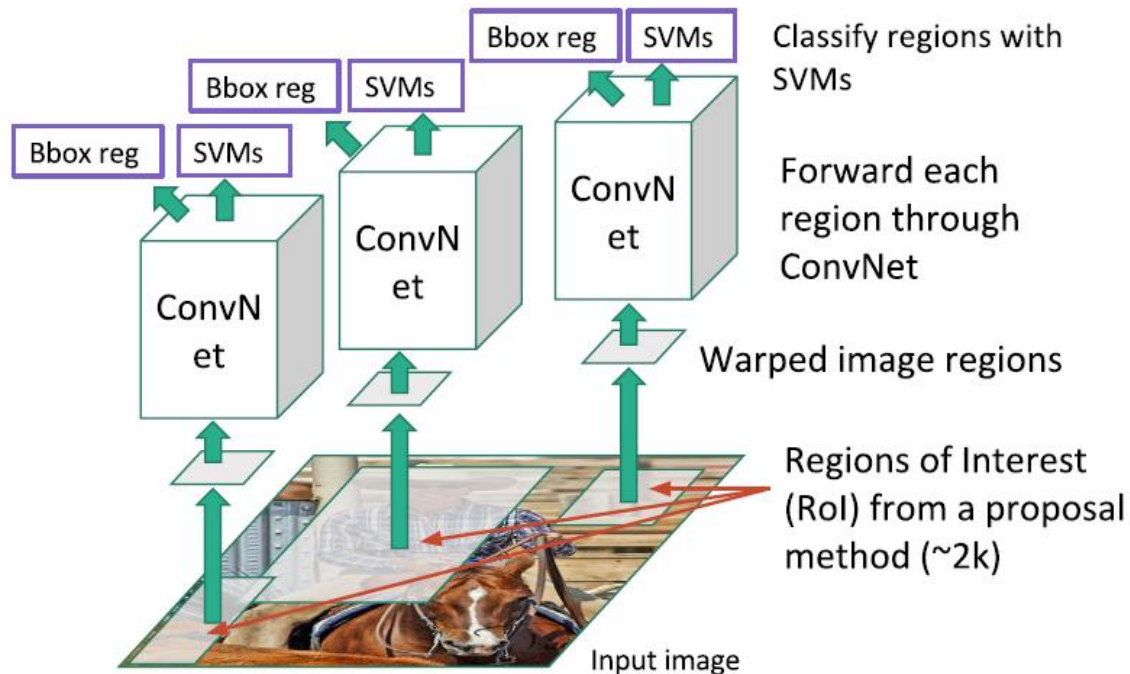
Regional CNN (R-CNN)

For **each** region proposal window :

- Linear SVM for object classification

Given every image region, one forward propagation through the CNN generates a feature vector. This feature vector is then consumed by a binary SVM trained for each class independently.

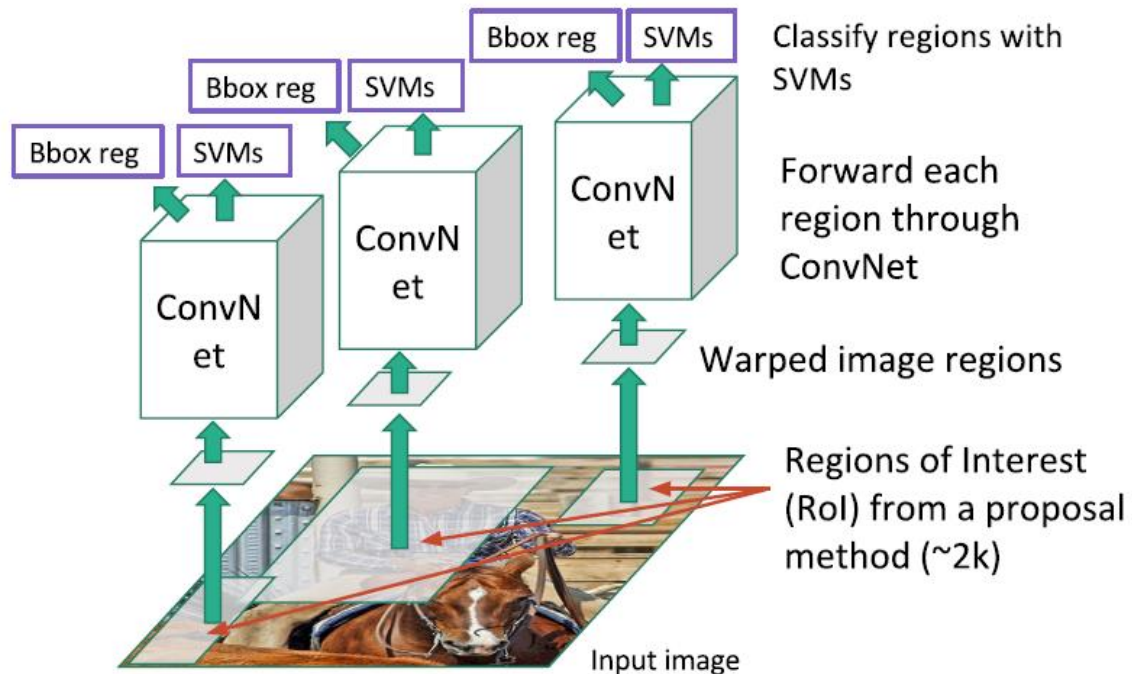
- positive samples are proposed regions with IoU
- negative samples are irrelevant others.



Regional CNN (R-CNN)

For **each** region proposal window :

- To reduce the localization errors, a regression model is trained to correct the predicted detection window on bounding box correction offset using CNN features.



Regional CNN (R-CNN)

Bounding Box Regression

predicted bounding box coordinate (center coordinate, width, height):

$$\mathbf{p} = (p_x, p_y, p_w, p_h)$$

its corresponding ground truth box coordinates

$$\mathbf{g} = (g_x, g_y, g_w, g_h)$$

the regressor is configured to learn scale-invariant transformation between two centers and log-scale transformation between widths and heights.

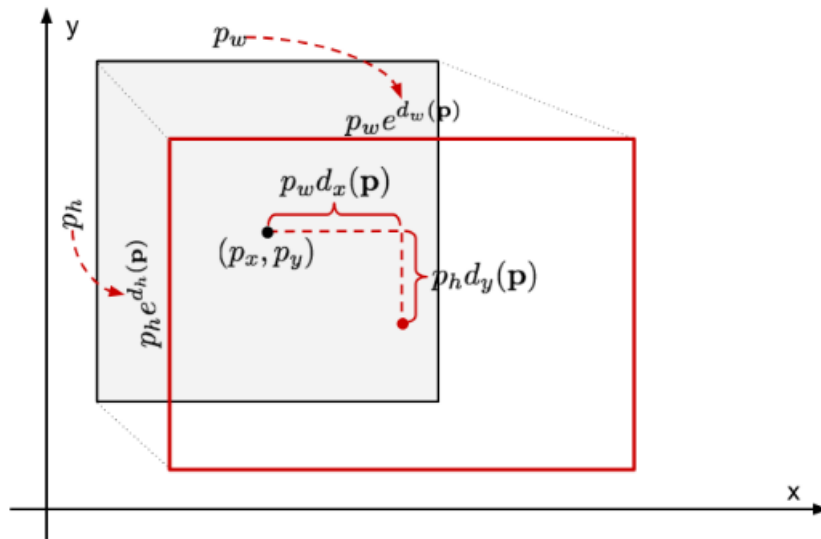
All the transformation functions take \mathbf{P} as input.

$$\hat{g}_x = p_w d_x(\mathbf{p}) + p_x$$

$$\hat{g}_y = p_h d_y(\mathbf{p}) + p_y$$

$$\hat{g}_w = p_w \exp(d_w(\mathbf{p}))$$

$$\hat{g}_h = p_h \exp(d_h(\mathbf{p}))$$



Regional CNN (R-CNN)

Bounding Box Regression

An obvious benefit of applying such transformation is that all the bounding box correction functions, $d_i(p)$ where $i \in \{x, y, w, h\}$, can take any value between $[-\infty, +\infty]$. The targets for them to learn are:

$$t_x = (g_x - p_x)/p_w$$

$$t_y = (g_y - p_y)/p_h$$

$$t_w = \log(g_w/p_w)$$

$$t_h = \log(g_h/p_h)$$

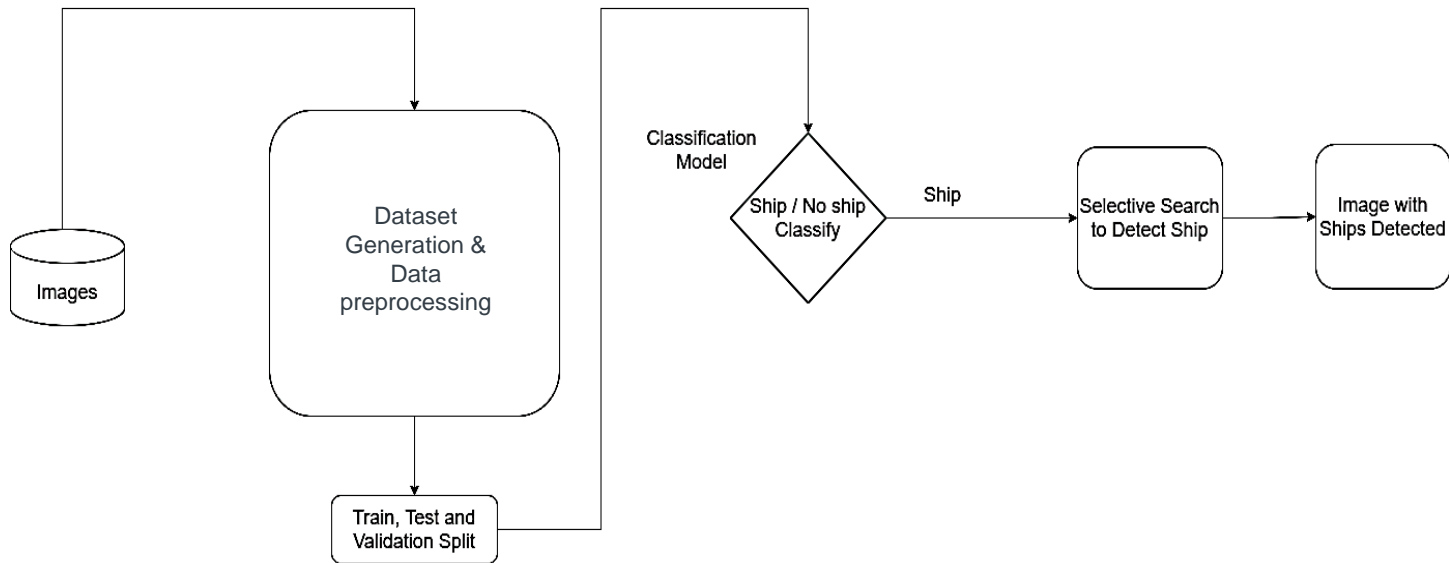
A standard regression model can solve the problem by minimizing the SSE loss with regularization:

$$\mathcal{L}_{\text{reg}} = \sum_{i \in \{x, y, w, h\}} (t_i - d_i(\mathbf{p}))^2 + \lambda \|\mathbf{w}\|^2$$

The regularization term is critical here and RCNN paper picked the best λ by cross validation.

Implementation Overview

Basic RCNN Model for Ship Detection



Basic R-CNN object detectors, rely on the concept of region proposal generators. In Classical implementation the location from the regional proposal was treated as the bounding box, while the SVM produced the class label for the bounding box region.

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles, suggesting a hierarchical or central structure. The lines are thin and gray, connecting the nodes in a non-linear fashion.

3.

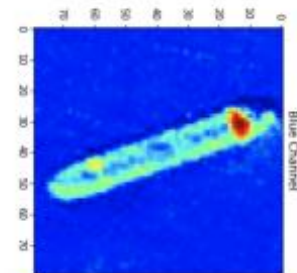
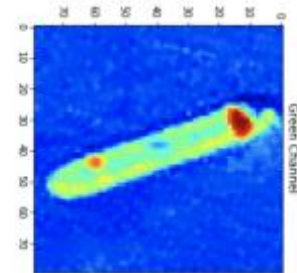
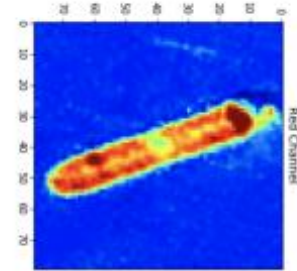
Data

Ships in Satellite Imagery

Datasets

Ships in Satellite Imagery

- **Source :** Satellite imagery ; PlanetScope
- **Location:** California ;San Francisco Bay /San Pedro Bay areas
- **Number:** 4000
- **Label:** ship/no-ship
- **Size:** 80x80 RGB
- **Format:** {label} __ {scene id} __ {longitude} _ {latitude}.png
- **List of 19200 integers**
 - First : 6400 ; red channel values
 - Second: 6400 ; green channel values
 - Third: 6400 ; blue channel values

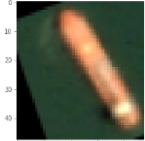
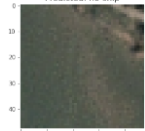


Datasets

Ships in Satellite Imagery

Dataset includes:

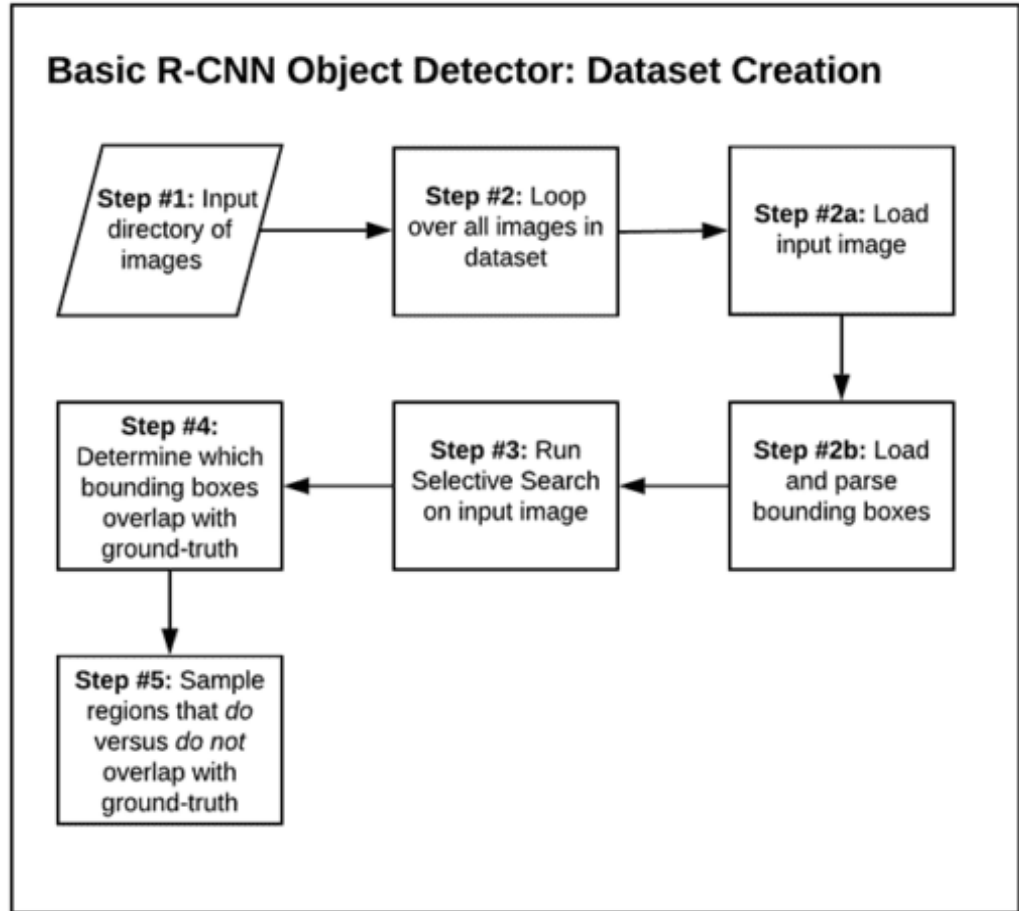
- **shipsnet.json** : JSON formatted file containing data, labels, scene id's, and location metadata
- **shipsnet** : 4000 images, entire dataset as .png image chips
- **Scenes** : satellite images including 8 .png images

| | Number | Type | Sample Image |
|---------|--------|---|---|
| SHIP | 1000 | Single Ship |  |
| NO-SHIP | 3000 | Random sampling/Partial ships/Mislabeled by ML models |  |

Datasets

Dataset Generation

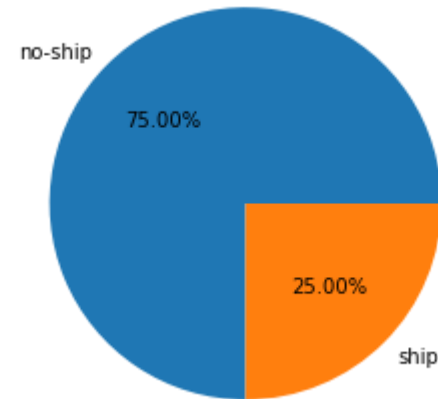
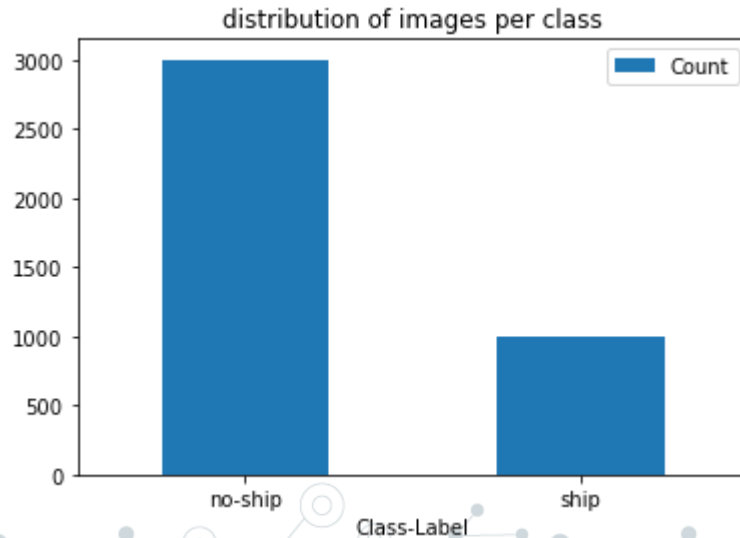
- Build object detection dataset from original dataset by applying Selective Search.
- Selective Search + post-processing logic
- identify regions of an input image that *do* and *do not* contain a potential object of interest.
- take these regions and use them as training data



Dataset

Data Visualization

- Generated Dataset used as training data consists of 2 directories, no-ship and ship, with each containing images as specified in the original dataset.



A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles, suggesting different levels of connectivity or importance. The lines are thin and gray, creating a mesh-like structure.

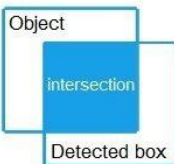
4. Metric

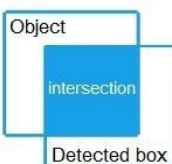
Measures and Evaluation metrics

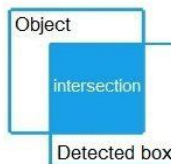
RCNN Evaluation Metrics

Goal: minimize or maximize through the modeling process

- **IOU** : intersection over union; IoU metric determines how many objects were detected correctly and how many false positives were generated
- **Precision** : measures how accurate your predictions are. i.e. the percentage of your predictions that are correct.
- **Recall** : measures how good you find all the positives.
- **F1 score** is HM (Harmonic Mean) of precision and recall.


$$\text{Precision} = \frac{\text{Intersection}}{\text{Detected box}}$$


$$\text{Recall} = \frac{\text{Intersection}}{\text{Object}}$$


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

Loss Function

Goal : the quantity which the model will minimize over the training

Classification Loss Function : In most cases CNNs use a **cross-entropy loss** on the one-hot encoded output. For a single image the cross entropy loss :

$$-\sum_{c=1}^M (y_c \cdot \log \hat{y}_c)$$

- where M is the number of classes and \hat{y}_c is the model's prediction for that class
- CNNs can also use several other kinds of loss functions

Bounding Box Regression Loss function: Bounding-box regression loss is best measured by the two loss functions : **IOU** and **MSE**

- MSE loss is the mean over seen data of the squared differences between true and predicted values, or writing it as a formula.

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^N (y - \hat{y}_i)^2$$

Non Maximum Suppression (NMS)

Goal : Improve object detection model

After classified region proposals ,

How to deal with extra bounding box generated by model ? Non- maximum suppression.

It works in 3 steps:

- Discard those objects where the confidence score is less than a certain threshold value(say 0.5).
- Select the region which has the highest probability among candidates regions for object as predicted region.
- In the final step we discard those regions which has **IoU** (intersection Over Union) with predicted region over 0.5.

Before non-max suppression



**NON-MAX
SUPPRESSION**



After non-max suppression



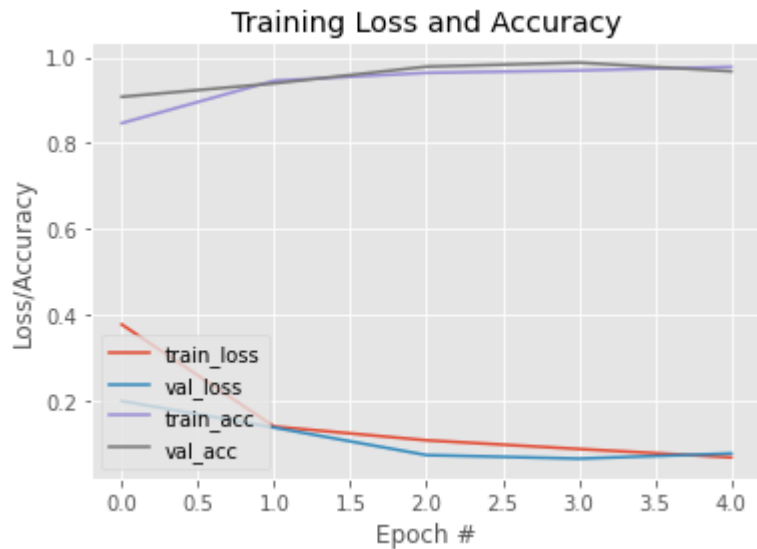
A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by circles of varying sizes, some with solid centers and others with dashed outlines. The lines are thin and gray, creating a mesh-like structure.

5. Result

Preprocess Data, Generate Model, Test Model, Benchmarking Results

RCNN : region proposal

Loss Function & Accuracy



On Generated Dataset

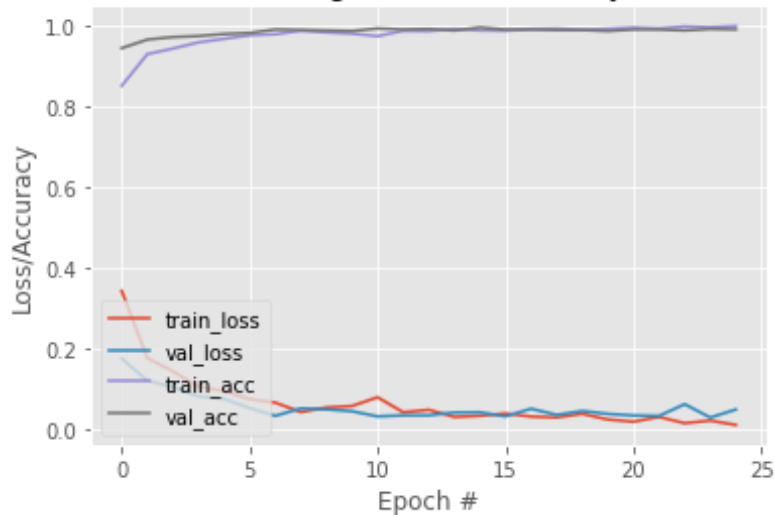
[INFO] evaluating network...

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| no-ship | 0.99 | 0.95 | 0.97 | 600 |
| ship | 0.88 | 0.98 | 0.93 | 200 |
| accuracy | | | 0.96 | 800 |
| macro avg | 0.94 | 0.97 | 0.95 | 800 |
| weighted avg | 0.96 | 0.96 | 0.96 | 800 |

RCNN : bounding box

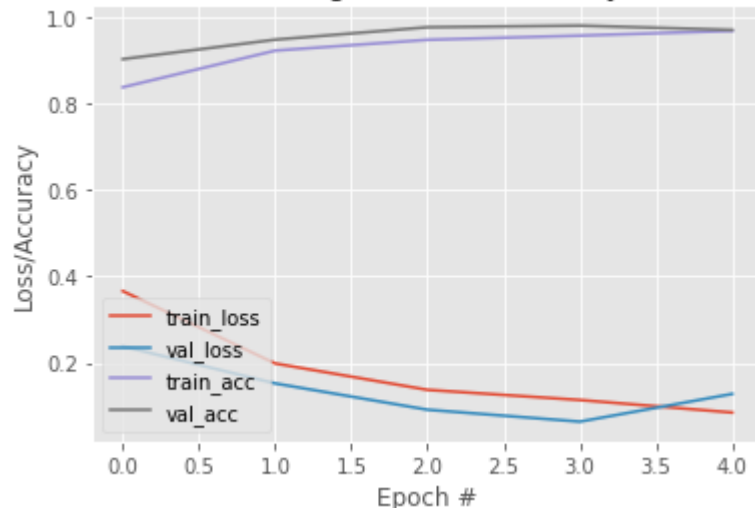
Loss Function & Accuracy

CNN Blocks & DENSE & Flatten Layers
Training Loss and Accuracy



On coordinated

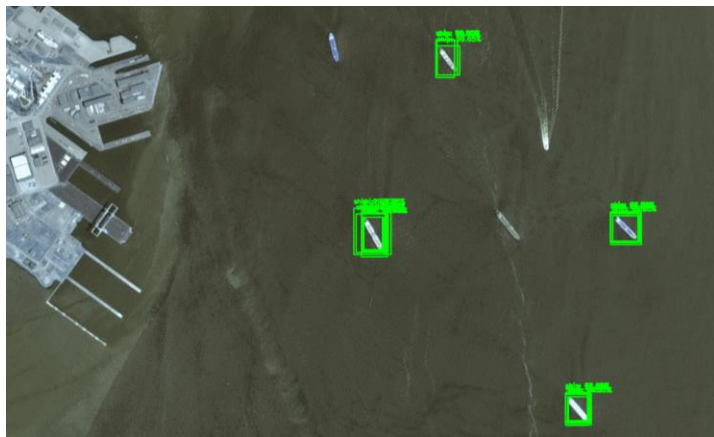
Pre-trained VGG16 & DENSE & Flatten Layers
Training Loss and Accuracy



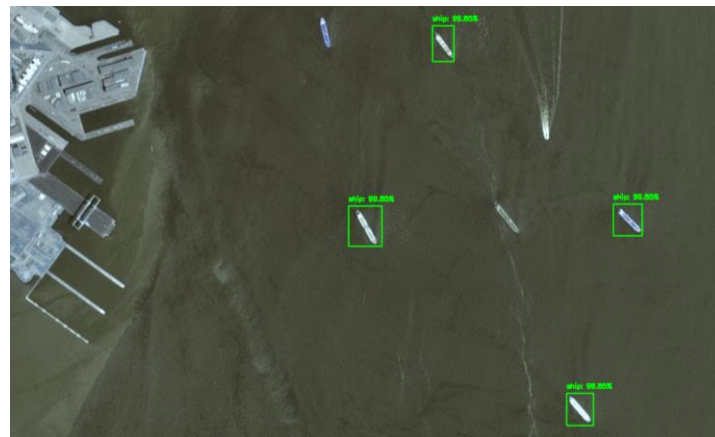
processing each RoI for every image by a pretrained model such as the VGG16 will takes alot of time even when by using the GPU.

RCNN : Object Detection

Before NMS



After NMS



Let's review some concepts



Since there are approximately *2000* candidate proposals. It takes **a lot of time to train** the network. Also we need to train multiple steps separately (CNN architecture, SVM model, bounding box regressor). So, This makes it very **slow** to implement.



Selective Search algorithm is very rigid and there is no learning happens in that. This sometimes leads to **bad region proposals generation** for object detection.

References

- [1] Girshick, R., Donahue, J., Darrell, T., & Malik, J., Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587), 2014.
- [2] Fei-Fei Li & Justin Johnson & Serena Yeung,, *Detection and Segmentation*, Lecture 11, Deep Learning Course, Stanford University School of Engineering, 1 May 2017
- [3] Hannah Peterson, “A Beginner’s Guide to Segmentation in Satellite Images”, Medium, 2020.
- [4] Adrian Rosebrock, Object detection: Bounding box regression with Keras, TensorFlow, and Deep Learning, Pyimagesearch University, 2020.



Thanks!

Any questions?

Contact me :

charmchisara@yahoo.com