An Overview of Object Recognition Tasks

PyML Studio

Vahid Mirjalili
https://vmirly.github.io

Overview of image recognition tasks

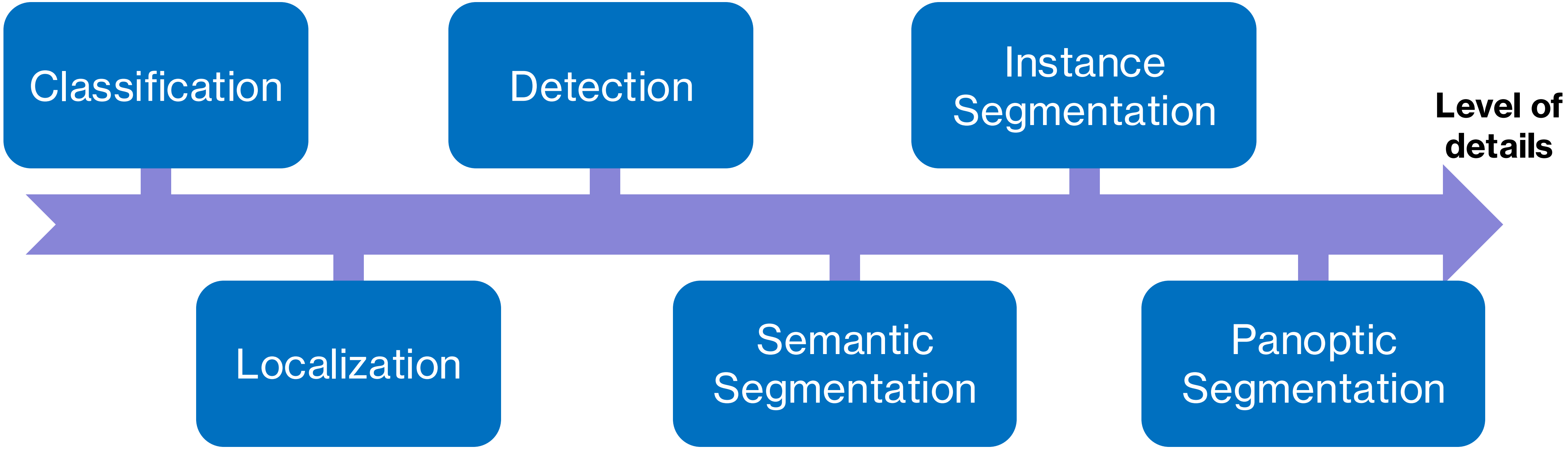
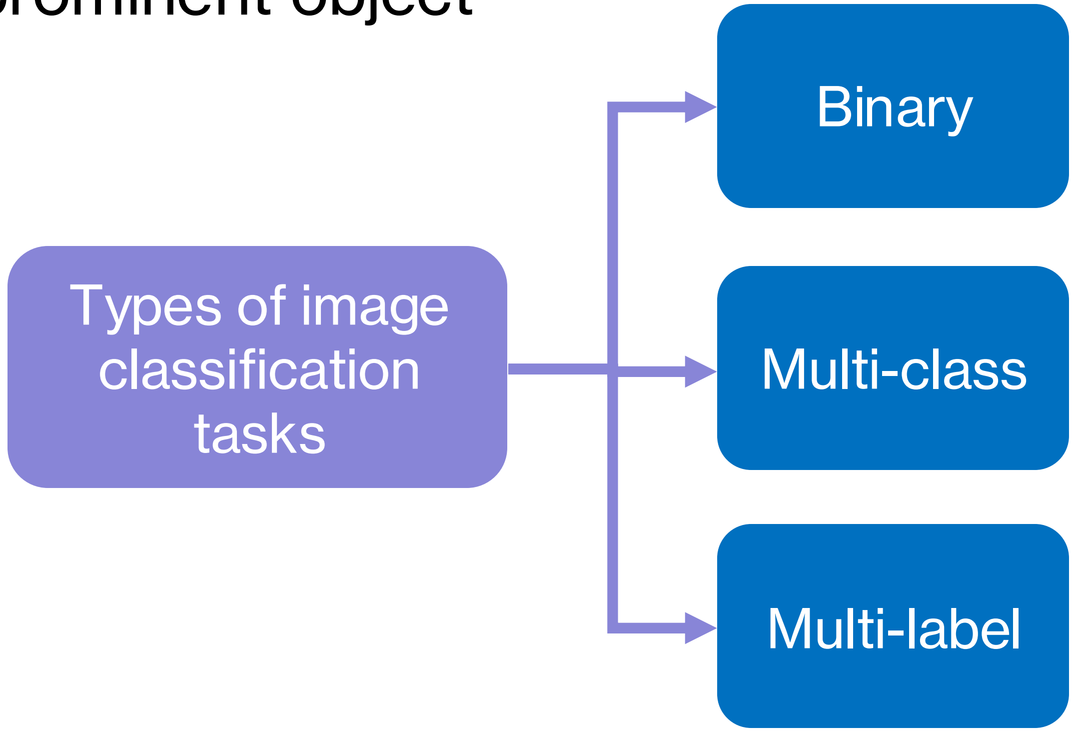


Image Classification

Image classification

- Assigning a label (or multiple labels) to an entire image, identifying the main subject or the most prominent object present in the image



Binary and multi-class classification

Input:

Binary:

"Does this image contain a horse"

Output:

"Yes"

Multi-class:

"Which animal (cat, dog, horse) is present in this image?"

Output:

"Horse"

➔ Assigning a single label

Multi-label classification

Input:

Multi-label:

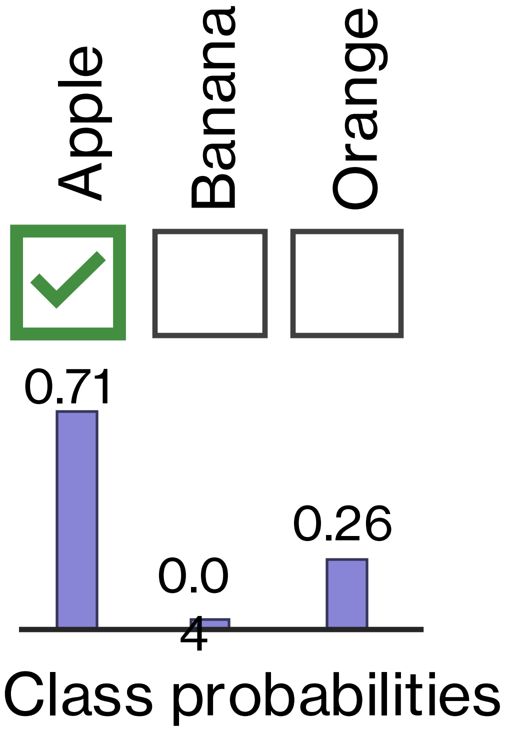
"What animals (cat, dog, horse) are present in this image?"

Output:

"Cat", "Dog"

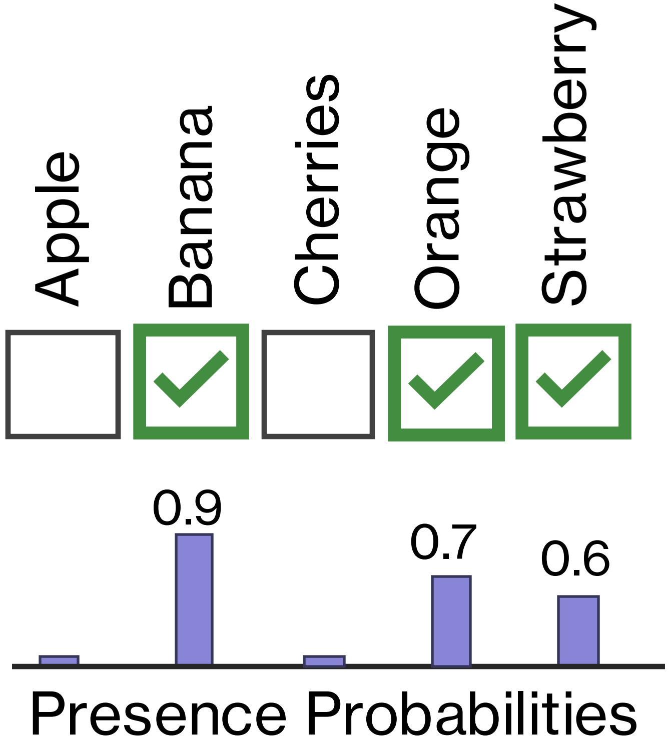
➔ Assign a list of labels to an input image

Multi-class

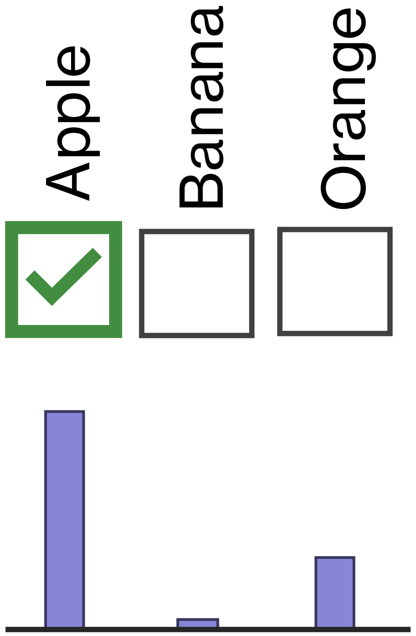


Sum to 1

Multi-label



Do not sum to 1



Class probabilities

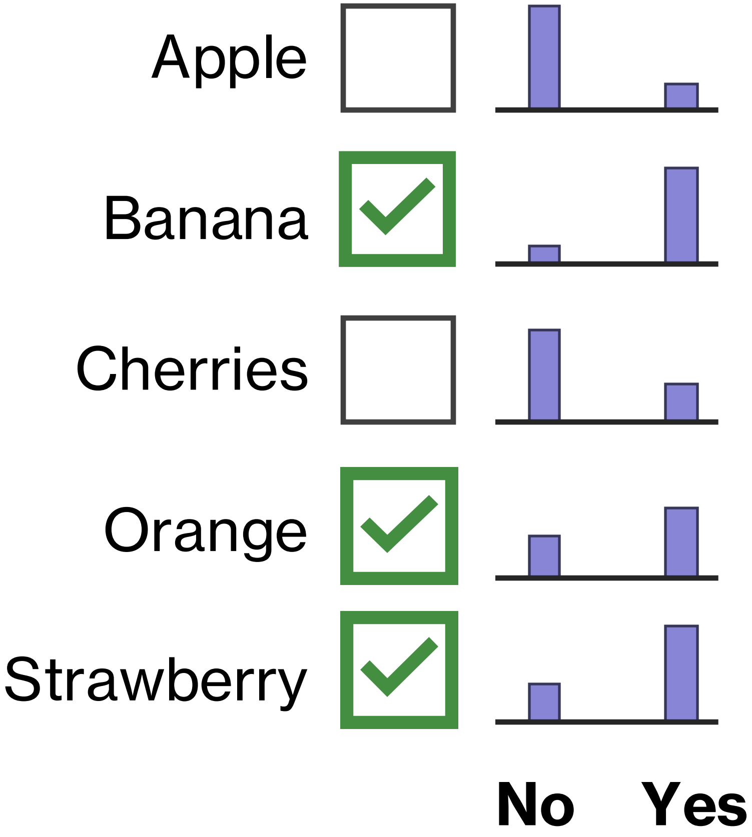
Softmax function

$$P = (\sigma_1, \sigma_2, \ldots, \sigma_c)$$

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}}$$

$$z = (z_1, z_2, \ldots, z_c)$$

Multi-label classification



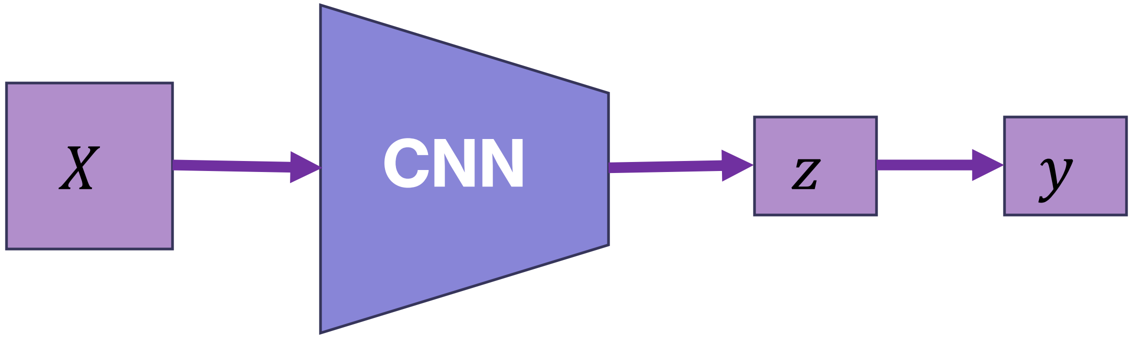
Multiple binary classifiers

Does the image contain this class? $\begin{cases} \text{No} \\ \text{Yes} \end{cases}$

➔ Sigmoid function

$$\sigma(z_i) = \frac{1}{1 + e^{-z_i}}$$

Building an image classifier



Approach 1:

Train from scratch

- CNN parameters are initialized randomly

Approach 2:

<u>Transfer Learning</u>

- Using pre-trained models
- Fine-tune the last layer

Performance Metrics

Accuracy

$$\text{Accuracy} = \frac{\#\ \text{correctly classified}}{\#\ \text{total}}$$

- Not suitable for imbalanced classes
- No information on individual class performance

Confusion Matrix

Ground Truth

	Apple	Banana	Orange
Apple	0.65	0.11	0.24
Banana	0.07	0.9	0.03
Orange	0.16	0.1	0.74

Performance Metrics

Precision & Recall

TP_i	FP_i
FN_i	TN_i

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i}$$

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i}$$

Macro-averaging:
treating classes equally

Micro-averaging:
treat each sample equally

F1 score

Harmonic mean of precision and recall

$$\text{F1 Score} = 2 \times \frac{\text{Precicion} \times \text{Recall}}{\text{Precicion} + \text{Recall}}$$

- Balancing precision and recall

Multi-label Performance Metrics

Zero-One Accuracy

(aka subset accuracy)

Ground Truth	0	1	0	1	1
Predicted	0	1	0	1	0

➔ Incorrect prediction

A sample is counted as correct if the two vectors match entirely.

Multi-label Performance Metrics

Hamming Accuracy

Ground Truth	0	1	0	1	1
Predicted	0	1	0	1	0
	✓	✓	✓	✓	✗

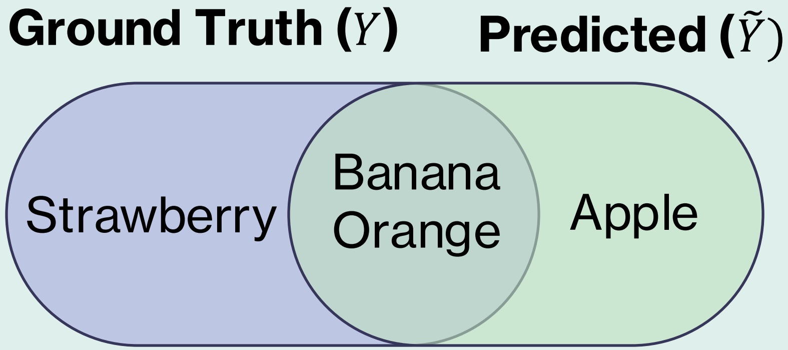
Count the number of matches

➔ 4 out of 5 are predicted correctly

Multi-label Performance Metrics

Jaccard Similarity

(Intersection over Union – IoU)



$$IoU = \frac{|Y \cap \tilde{Y}|}{|Y \cup \tilde{Y}|}$$

- Size of intersection = 2
- Size of union = 4

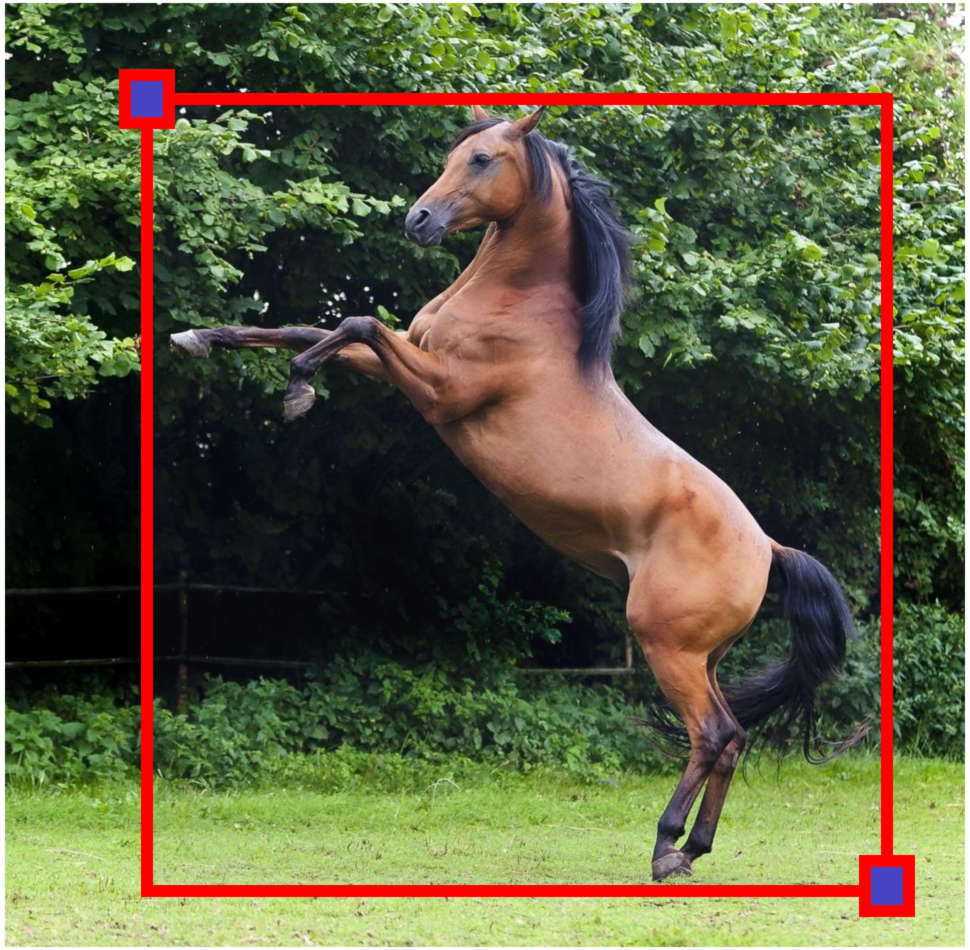
➔ $IoU = \frac{2}{4}$

Object Localization

Object localization

- What is the object in the image?
- Where is it?

Object localization involves not only identifying what the primary object in an image is (classification) but also determining its specific location with a bounding box.

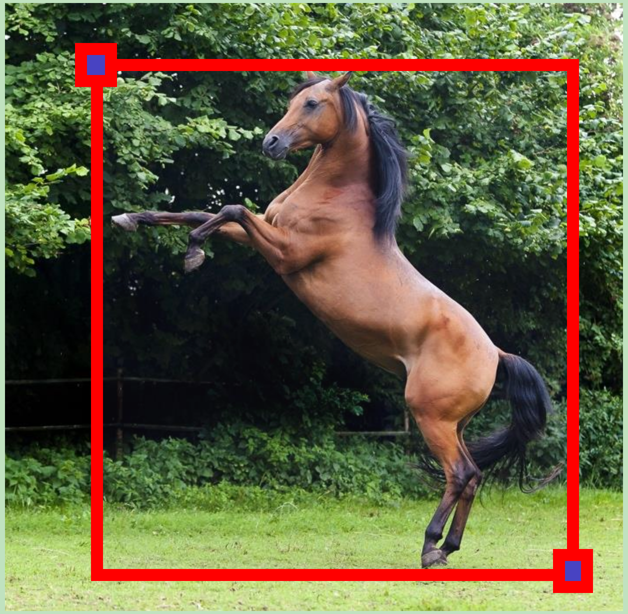


Object localization

Input:

Output:

Bbox Coordinates: (x_1, y_1, x_2, y_2)



Limitations of object localization

Single Object Focus:

Traditional object localization techniques may struggle when multiple instances of the same object are present, often creating a bounding box that encompasses all instances as a single object.

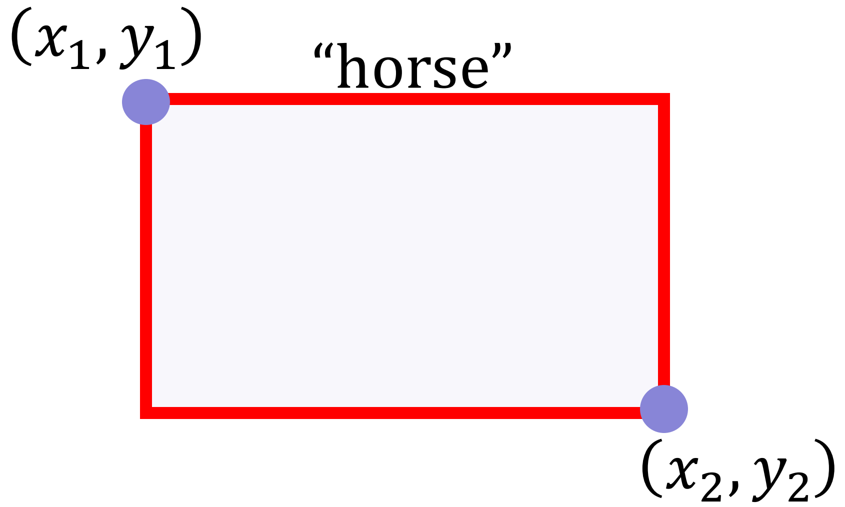
Object Detection

Object detection

- Detecting multiple instances of semantic objects (pedestrians, cars, ...)

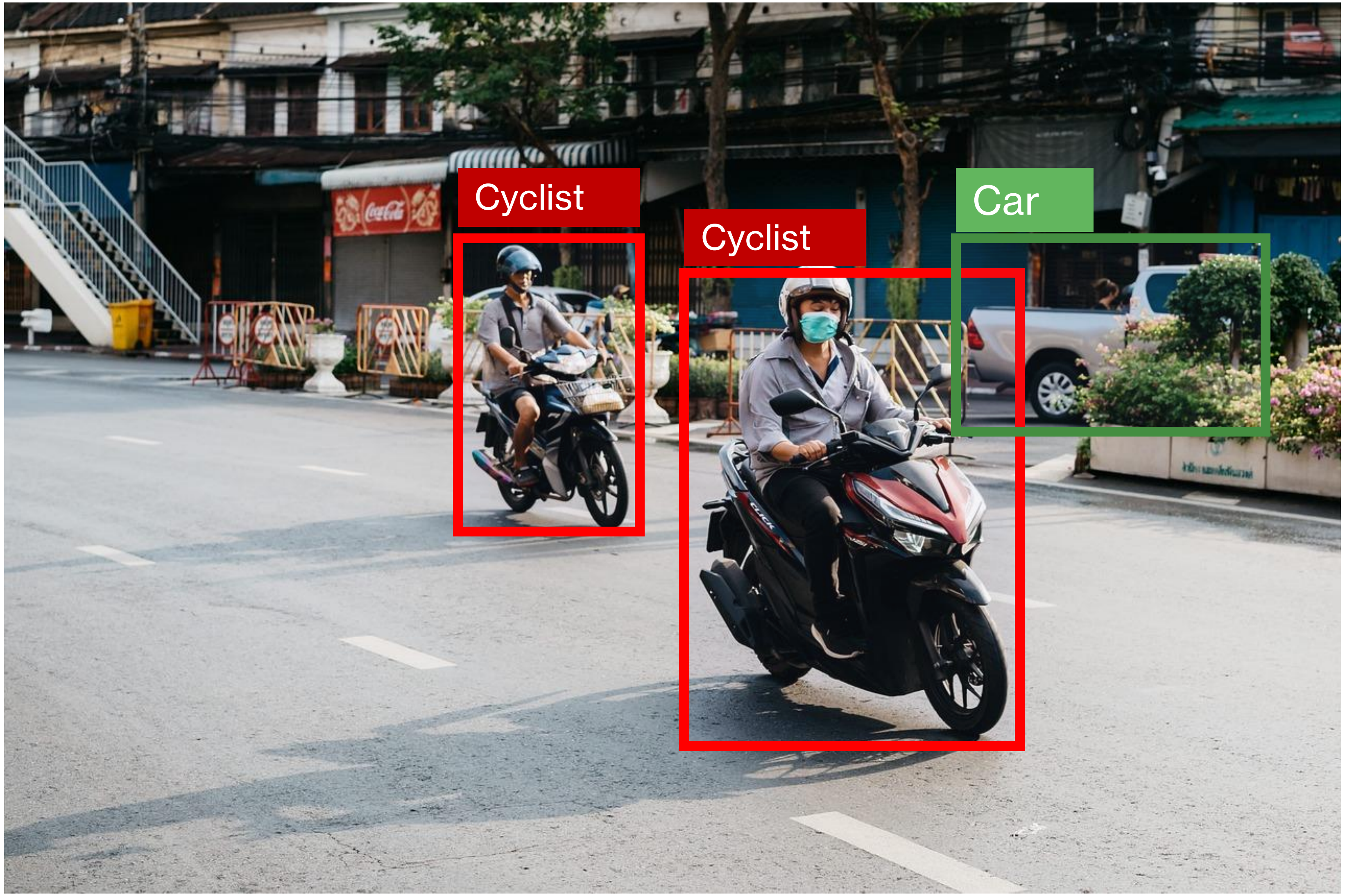
- Finding bounding-boxes around each instance

- Output:
  - Object class
  - Bounding boxes can be represented with a vector of 4 values: (x_1, y_1, x_2, y_2)



Object detection

Example output ➔

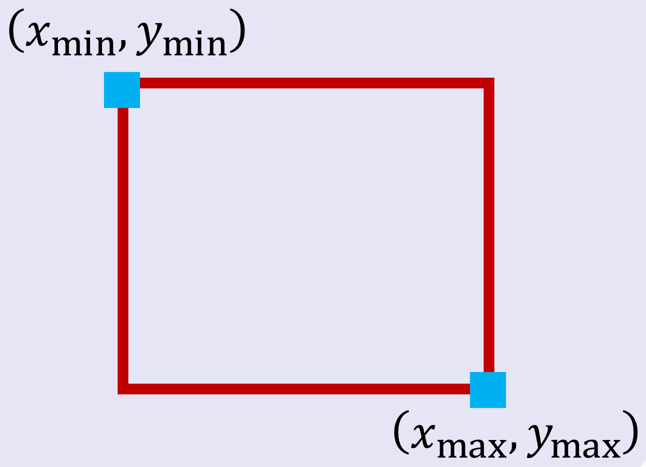


Bounding box formats

Pascal VOC

“xyxy”

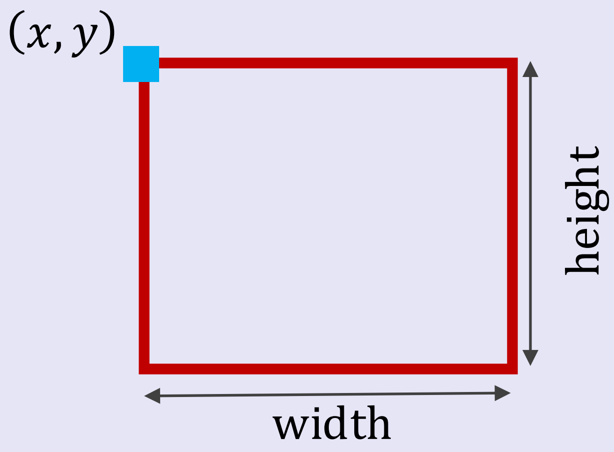
$[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$



COCO

“xywh”

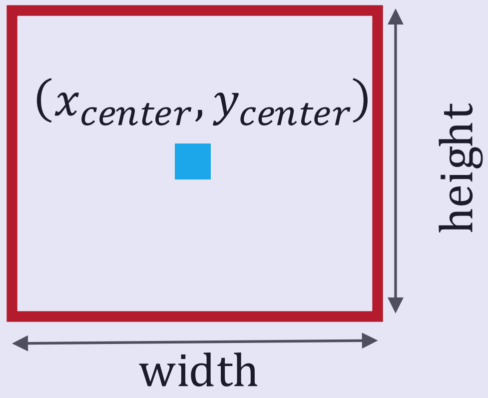
$[x_{\min}, y_{\min}, \text{width}, \text{height}]$



YOLO

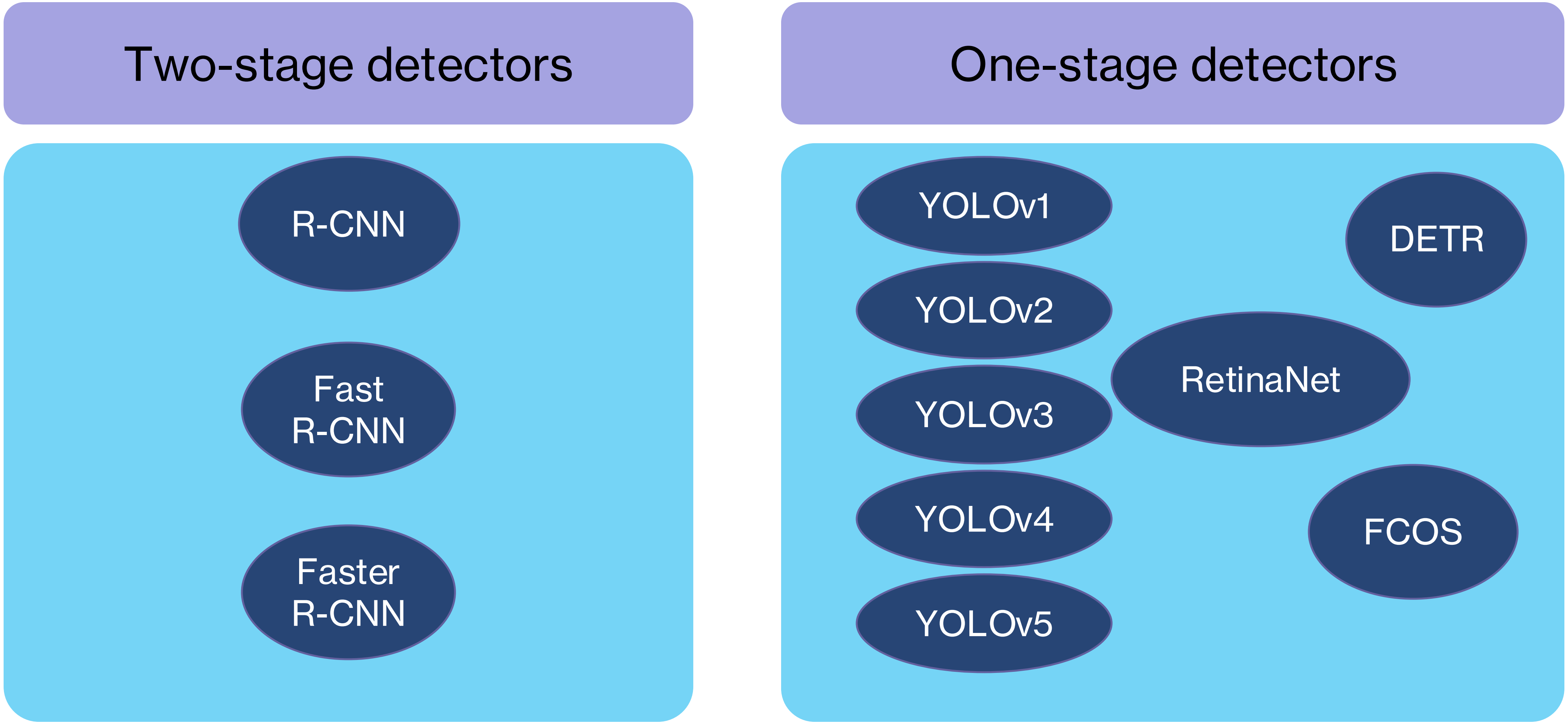
“cxcywh”

$[x_{\text{center}}, y_{\text{center}}, \text{width}, \text{height}]$

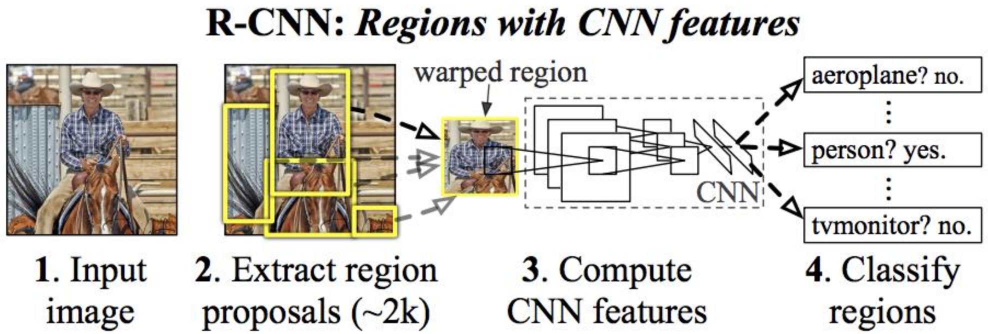


Read more: https://albumentations.ai/docs/getting_started/bounding_boxes_augmentation/

Object detection models



R-CNN
Regions with Convolutional Neural Networks

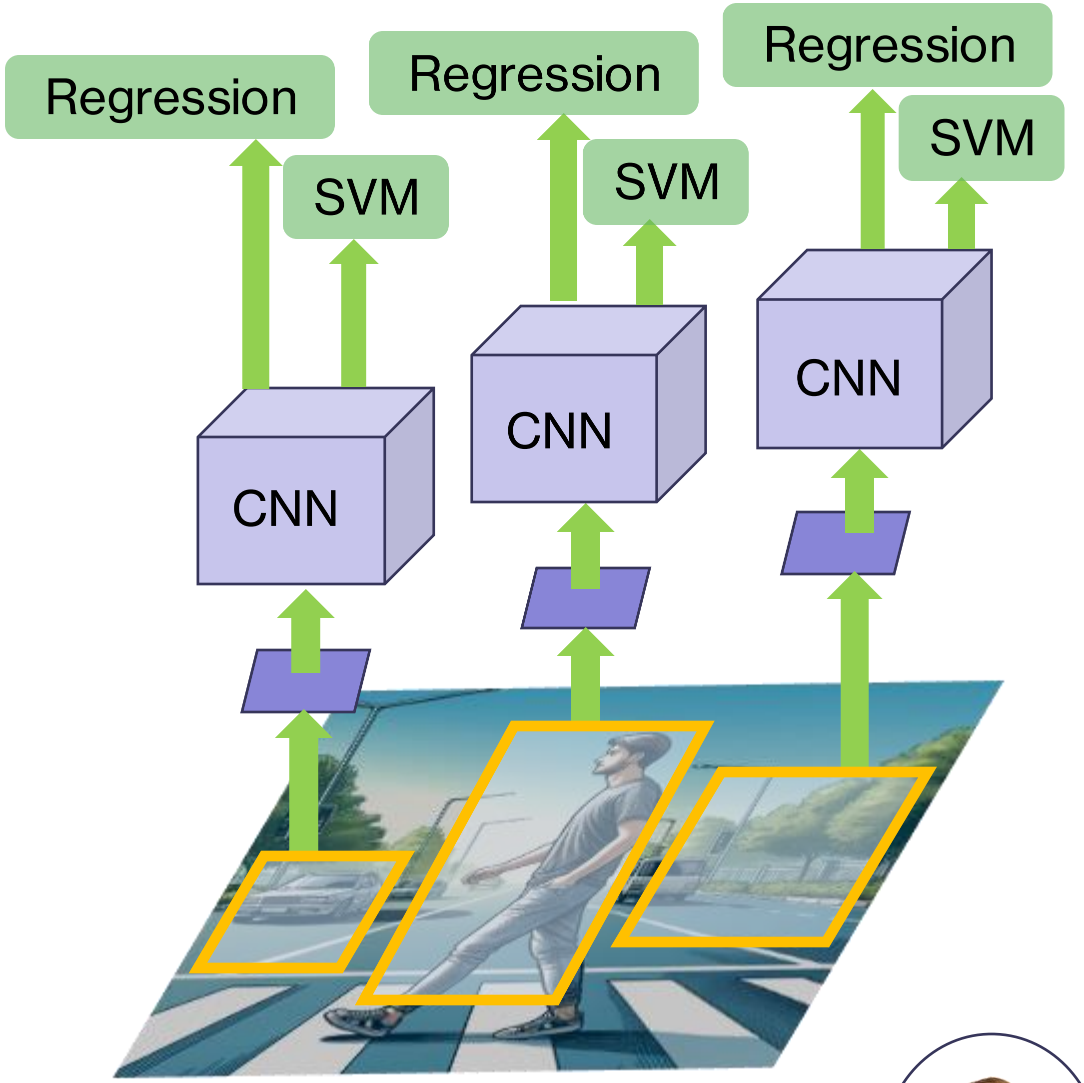


Step 1: Extract ~2000 region proposals using **selective search algorithm**

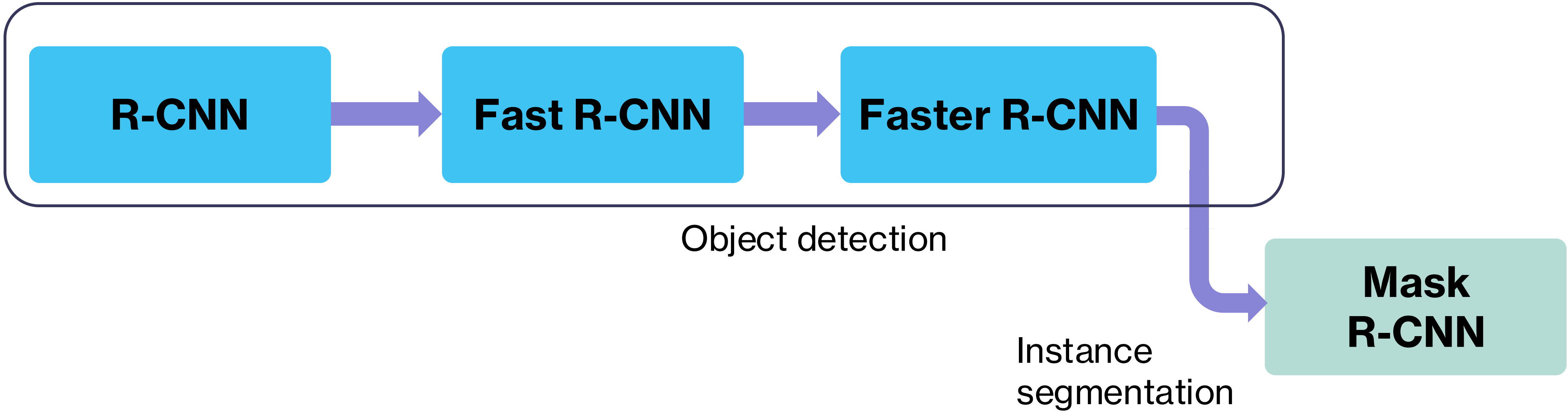
Step 2: Compute features for each region

Step 3: Classify each region using linear-SVM

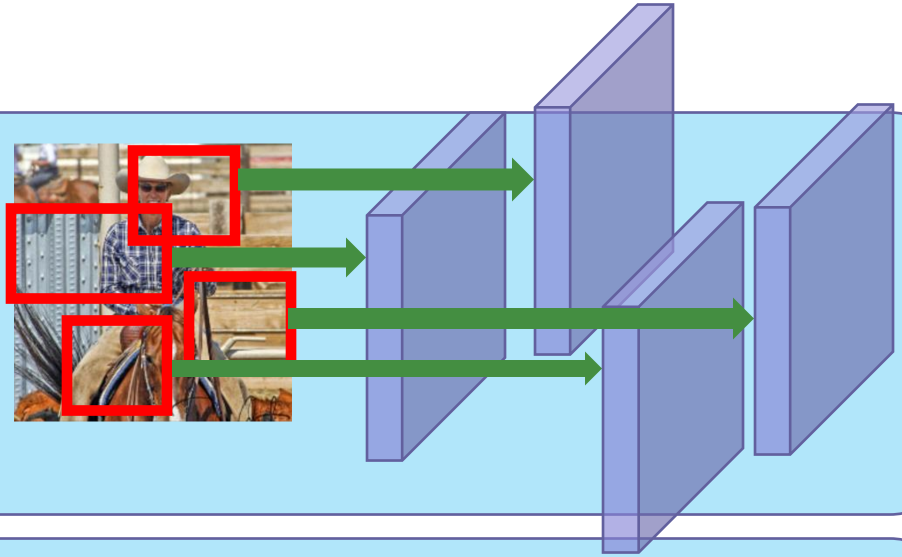
Drawback: redundant computations



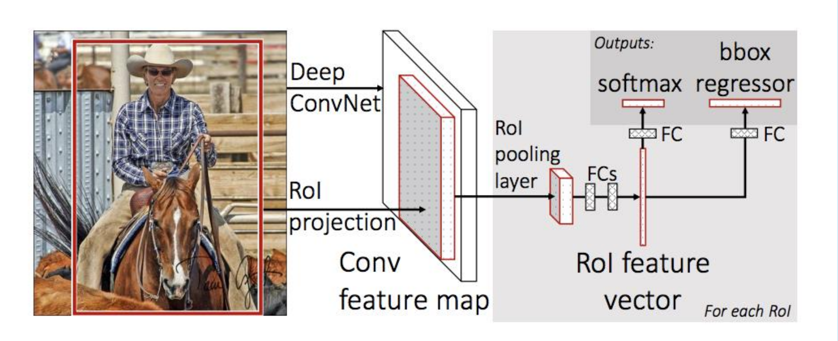
Evolution of R-CNN Models



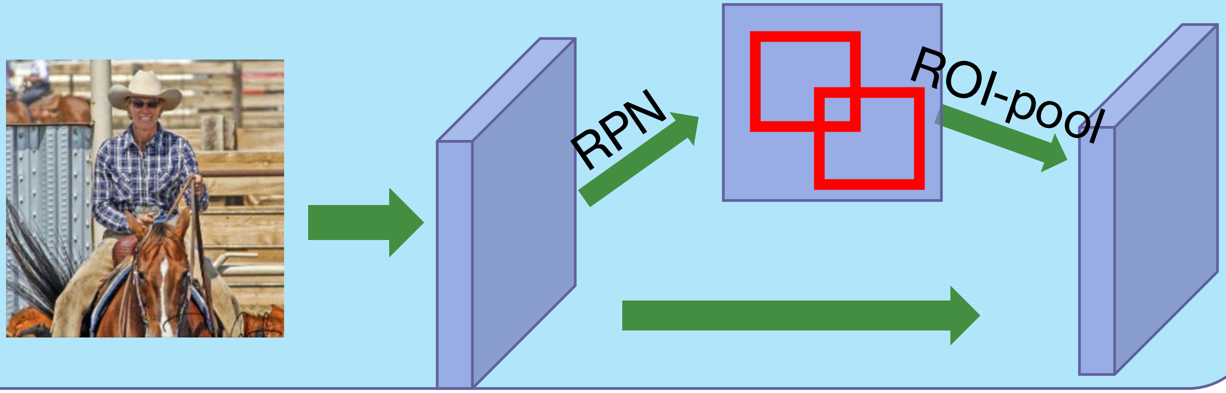
- R-CNN
  - Selective search on the original image
  - Computes the features for each region proposal



- Fast R-CNN
  - Sharing the computations for different region proposals
    ➔ shared feature maps
  - Extract feature vectors from the shared feature maps

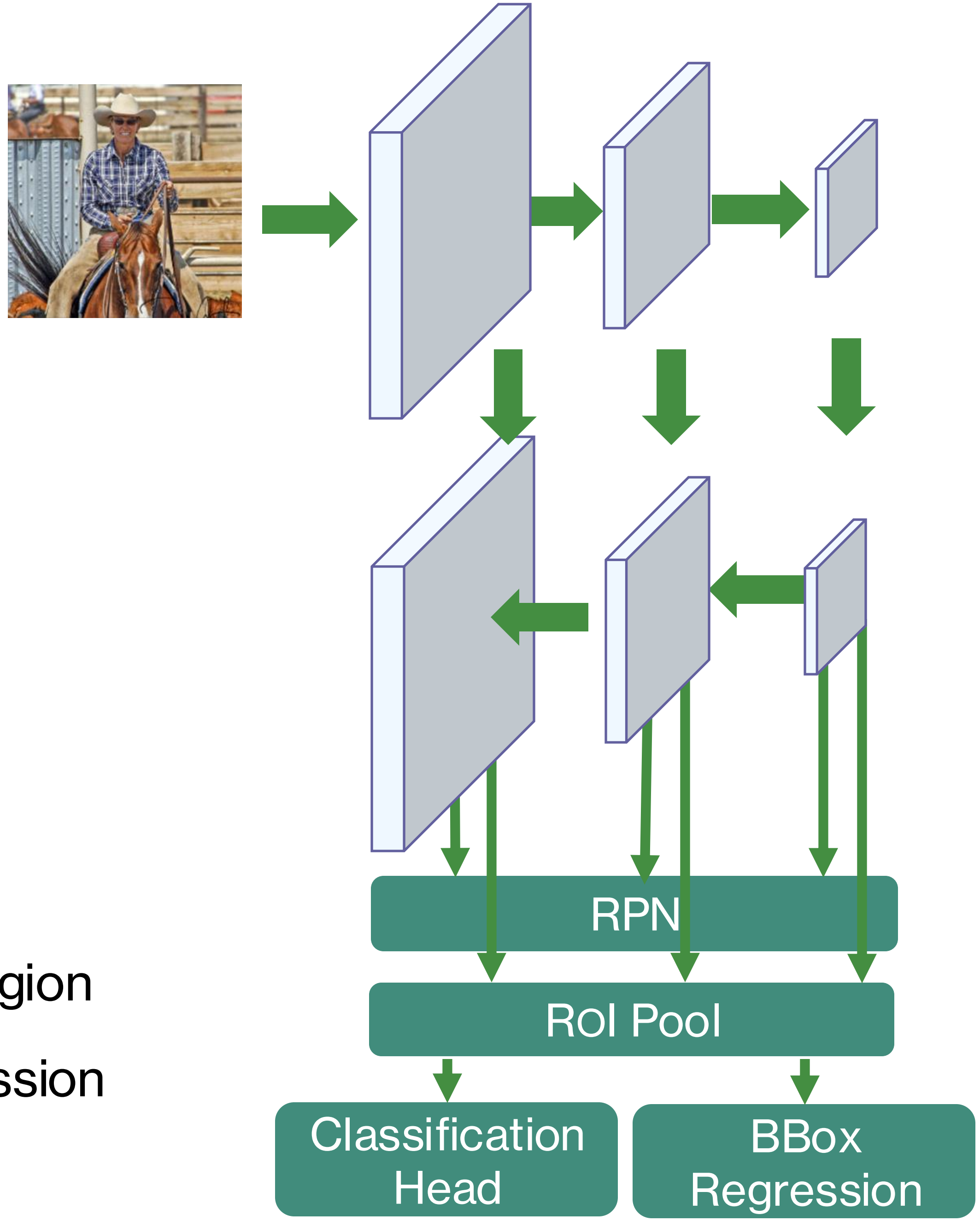


- Faster R-CNN
  - Region proposal network (RPN)
  - ROI-Pool



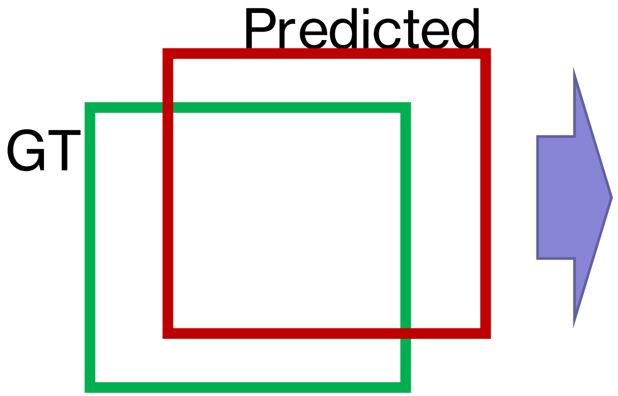
Architecture of Faster R-CNN

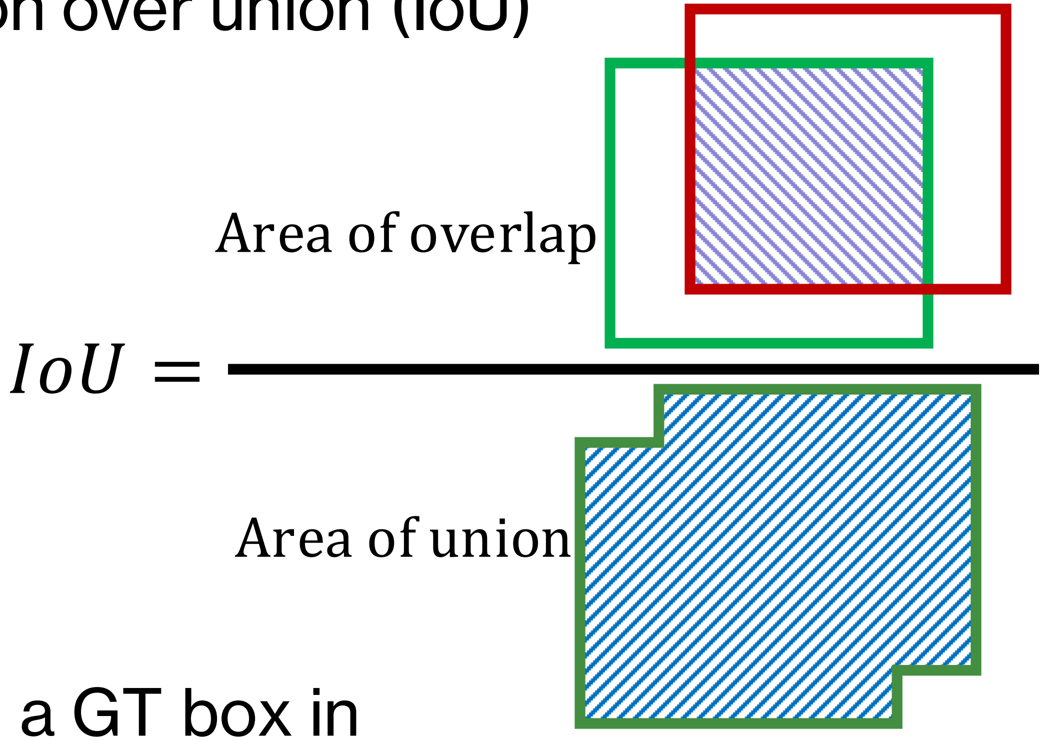
- Backbone network
  - ResNet-50
  - ResNet-101
- Feature Pyramid Network (FPN)
  - Using lateral connections at different feature scales
  - Extracts different feature hierarchies
  - Multiscale object recognition
- RPN and ROI-pool: Extracting features for each region
- Two heads: classification and bounding-box regression



Object Detection Performance Metrics

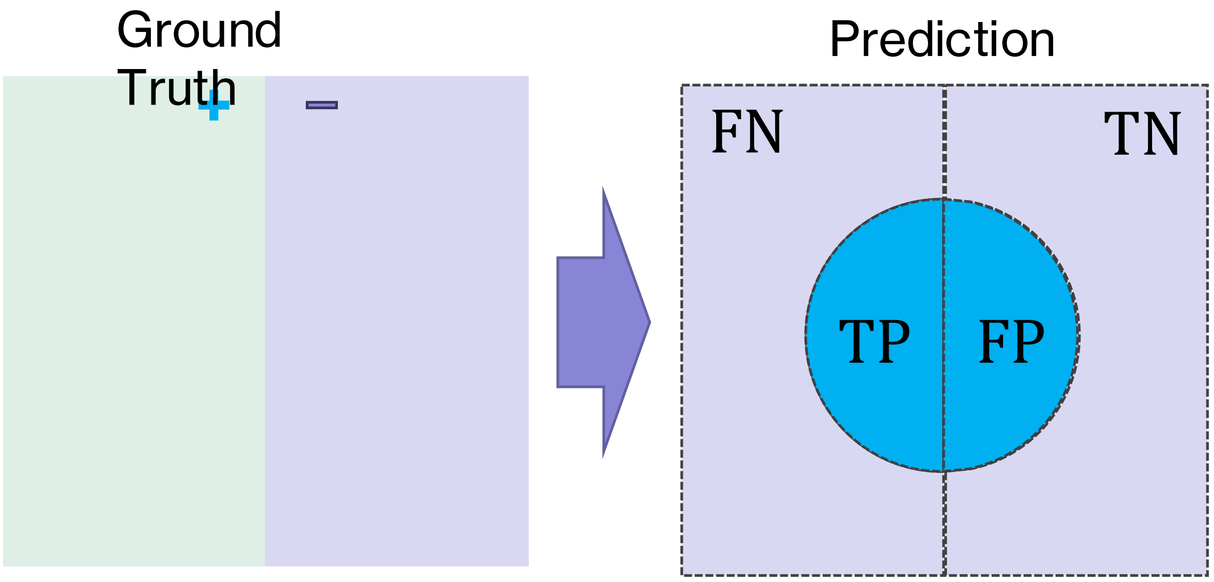
- Mapping predictions and ground truth boxes into TP, FP, or FN based on intersection over union (IoU)



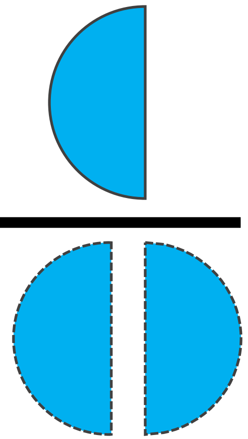


- TP: a predicted box matched with a GT box in the same class with IoU > threshold

Precision and Recall

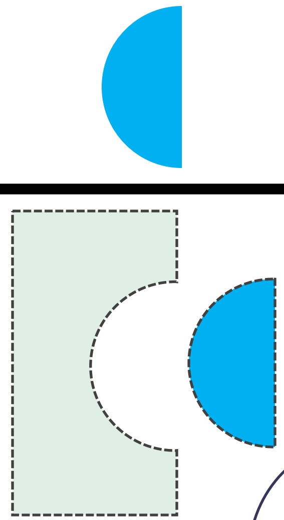


$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$



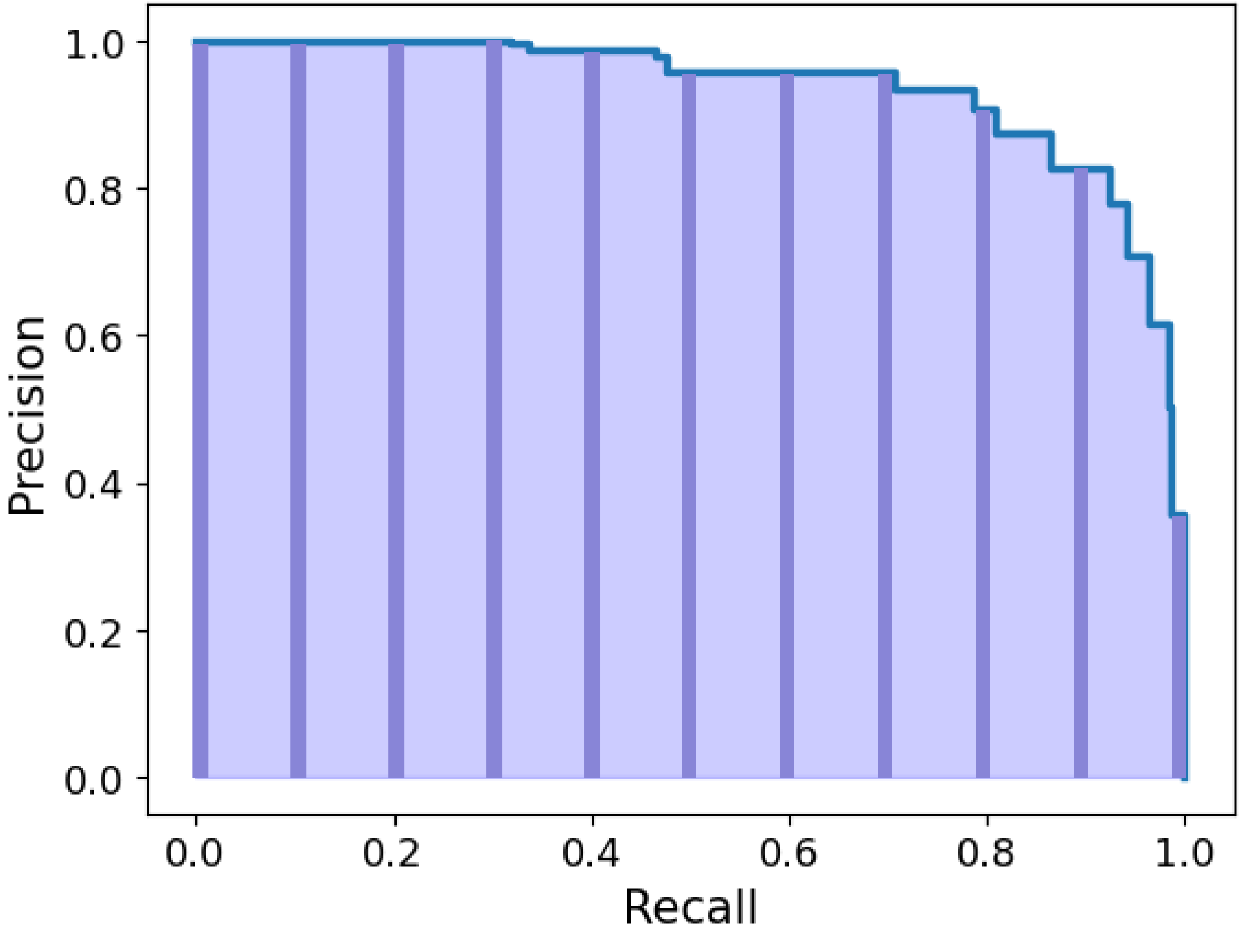
Total number of predicted boxes

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$



Total number of GT boxes

Average Precision – AP



- **Calculated for each class individually**
- Precision-recall curve based on confidence values
- Area under the curve of the precision-recall curve
  - For class *k:*

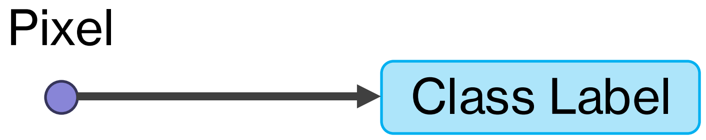
$$AP_k = \int_0^1 p(r)\, dr$$

$$AP_k \approx \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1.0\}} p\,(r)$$

Semantic Segmentation

Semantic segmentation

- **Pixel-wise classification:** Assigns a class label to each individual pixel in an image, effectively partitioning it into regions.

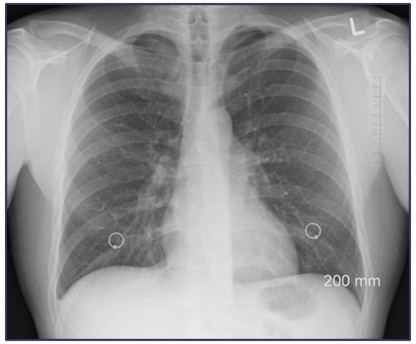


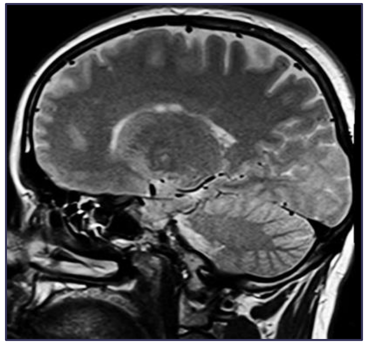
- Provides detailed comprehension of the scene, distinguishing between different objects and background.

Applications:

- Medical imaging
- Autonomous vehicles
- Scene understanding tasks.

Applications



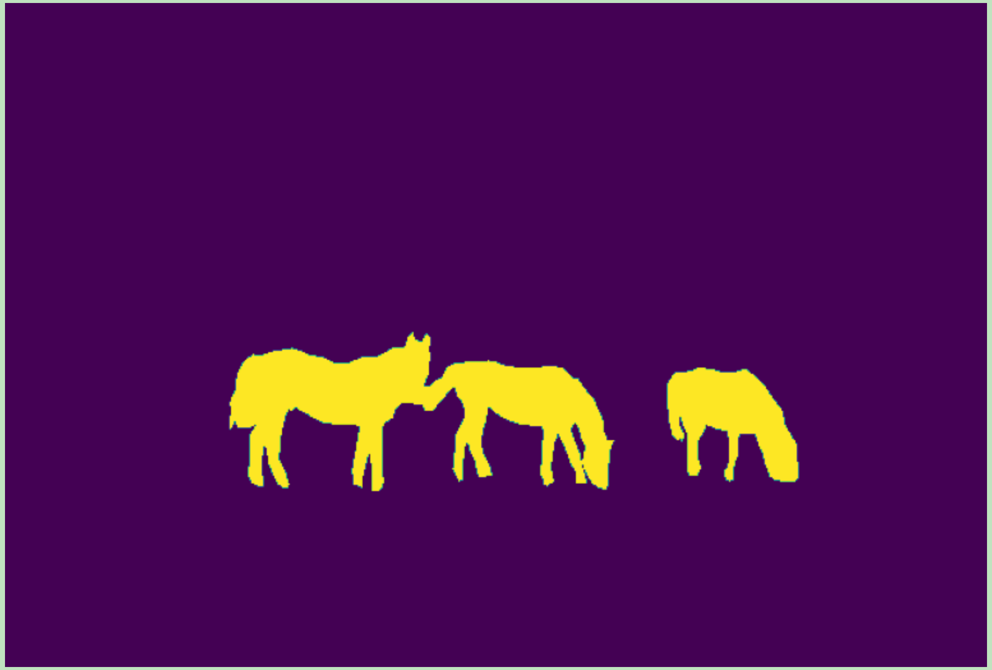


- Medical imaging
- Self-driving cars
- Robotics
- Image editing
- Agriculture automation
- …

Semantic segmentation

Input:

Output:

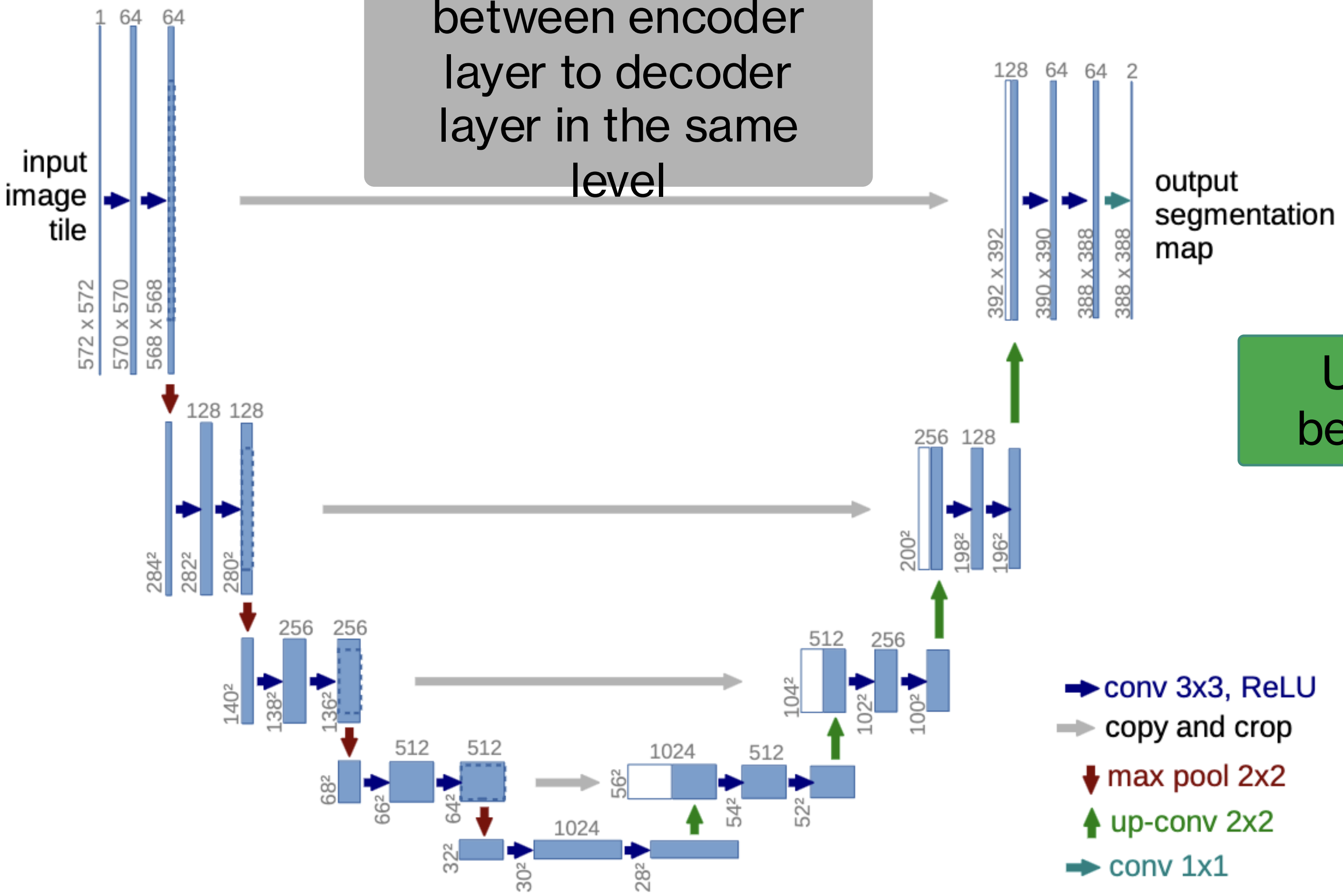


Semantic segmentation with UNet

Skip-connections between encoder layer to decoder layer in the same level

Down-sampling between levels

Up-sampling between levels



“U-Net: Convolutional Networks for Biomedical Image Segmentation, O Ronneberger et al.”, 2015, https://arxiv.org/pdf/1505.04597.pdf

Performance metrics for semantic segmentation

Pixel Accuracy

$$\frac{\text{\# of correctly classified pixels}}{\text{Total \# of pixels}}$$

➔ Not symmetric

➔ Biased towards predictions larger than GT

➔ Suffers from imbalanced classes

Per-class Precision & Recall

For each class, map each pixel into TP, FP, TN, FN categories

$$\mathrm{Precision_k} = \frac{TP_k}{TP_k + FP_k}$$

$$\mathrm{Recall_k} = \frac{TP_k}{TP_k + FN_k}$$

➔ Provides detailed class-level performance measure

➔ Not symmetric

IoU

$$IoU_k = \frac{|g_k \cap p_k|}{|g_k \cup p_k|}$$

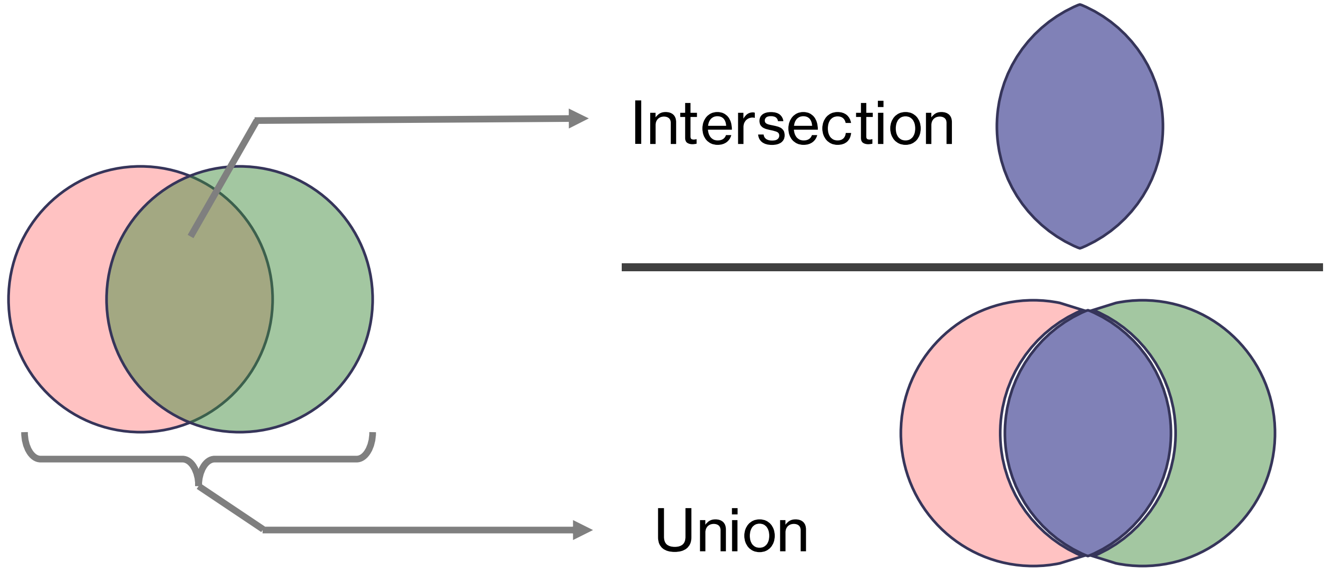
Dice Coefficient (F1 score)

$$\mathrm{Dice}_c = \frac{2 \times |g_k \cap p_k|}{|g_k| + |p_k|}$$

IoU

$$IoU_k = \frac{|g \cap p|}{|g \cup p|}$$

$$= \frac{TP_k}{TP_k + FP_k + FN_k}$$

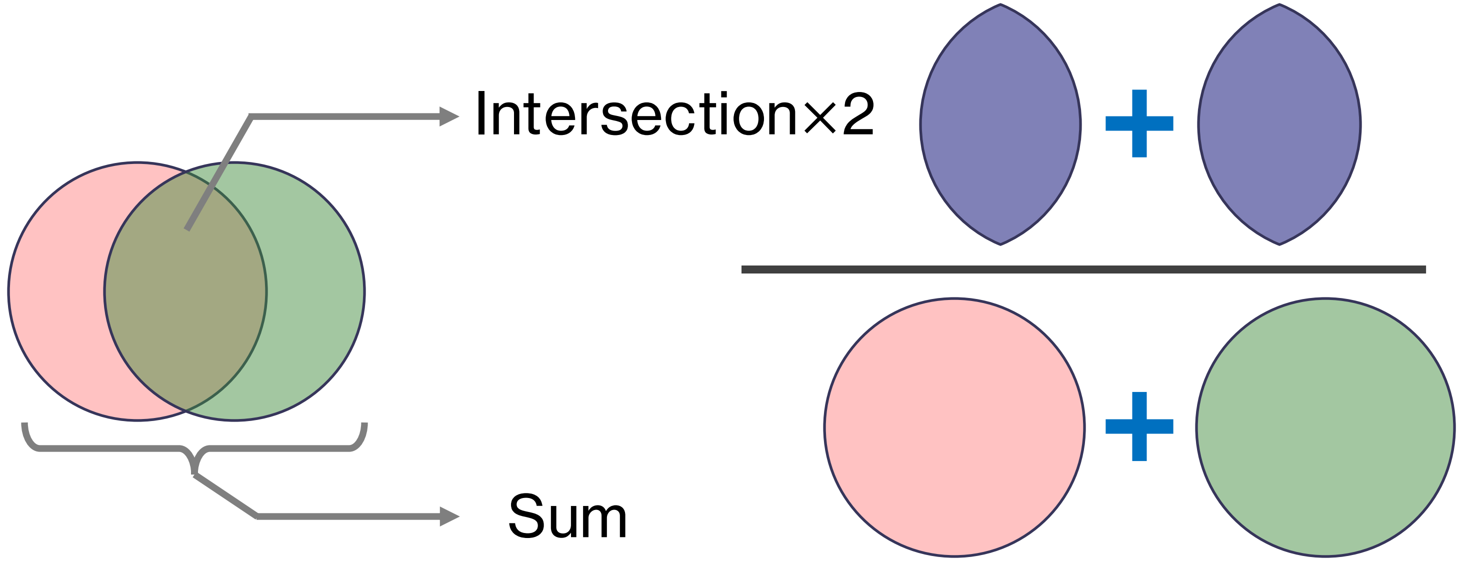


Dice Coefficient

$$\mathrm{Dice}_k = \frac{2 \times |g \cap p|}{|g| + |p|}$$

$$= \frac{2TP_k}{2TP_k + FP_k + FN_k}$$

More sensitive for small objects



Boundary IoU

(gt) p1 p2 p3

	p1	p2	p3
Mask IoU	89%	92%	97%
Boundary IoU	69%	78%	91%

Mask IoU: $\Delta = 8\%$

Boundary IoU: $\Delta = 22\%$

Source: https://arxiv.org/pdf/2103.16562

Instance Segmentation

Instance segmentation

Individual Object Identification:

- Identifying instances of objects in an image
- Differentiating between multiple instances of the same class
- Combining detection and segmentation
- Applicable to countable things (not stuff)

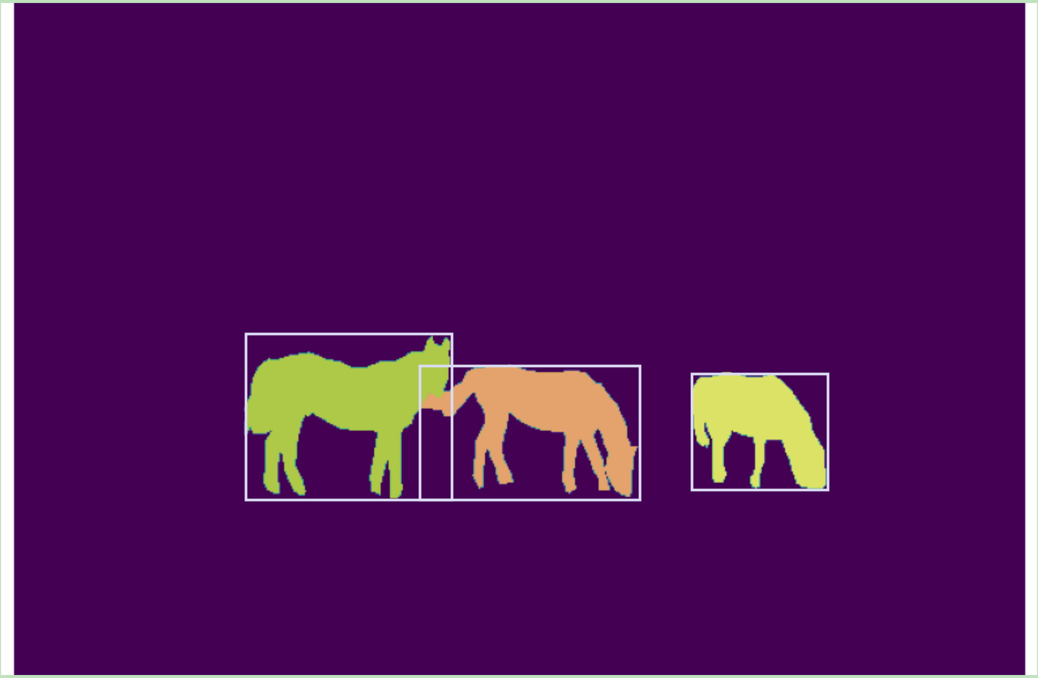
Objects (things) vs. stuff

- Objects: discrete and countable things
  - E.g., people, animals, cars, …
- Stuff: continuous areas (not countable)
  - E.g., sky, grass, road, …

Instance segmentation

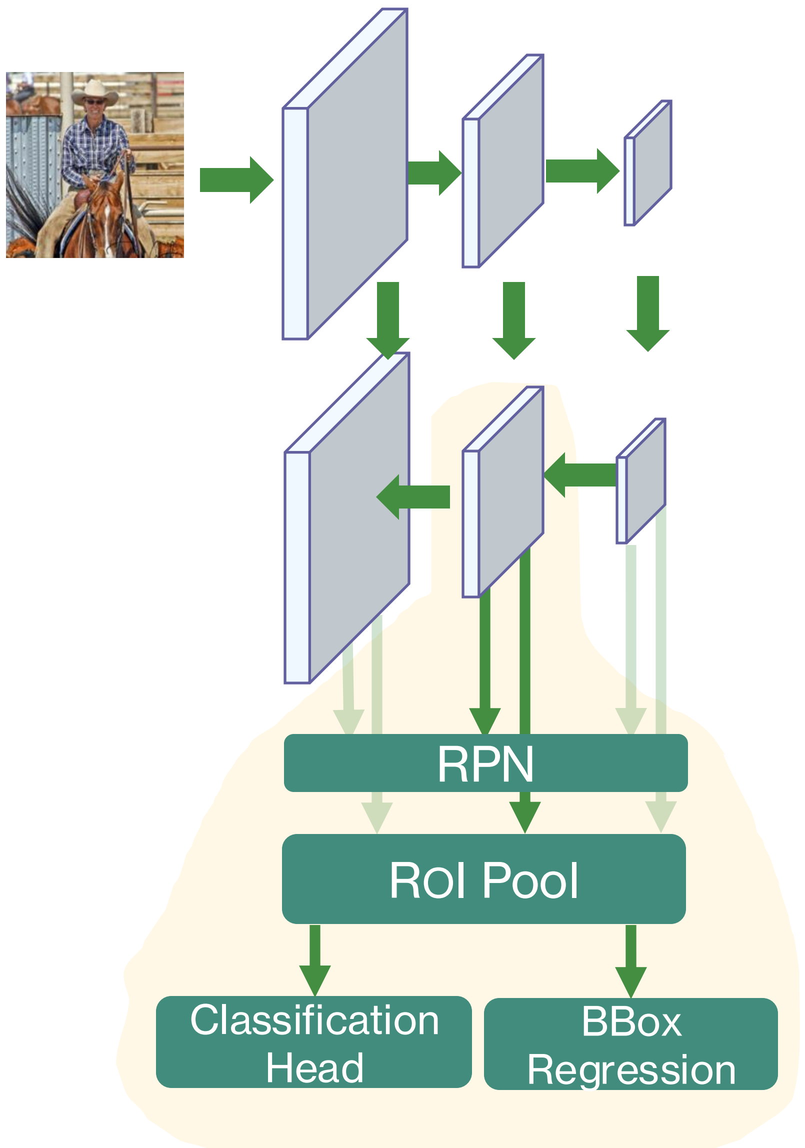
Input:

Output:



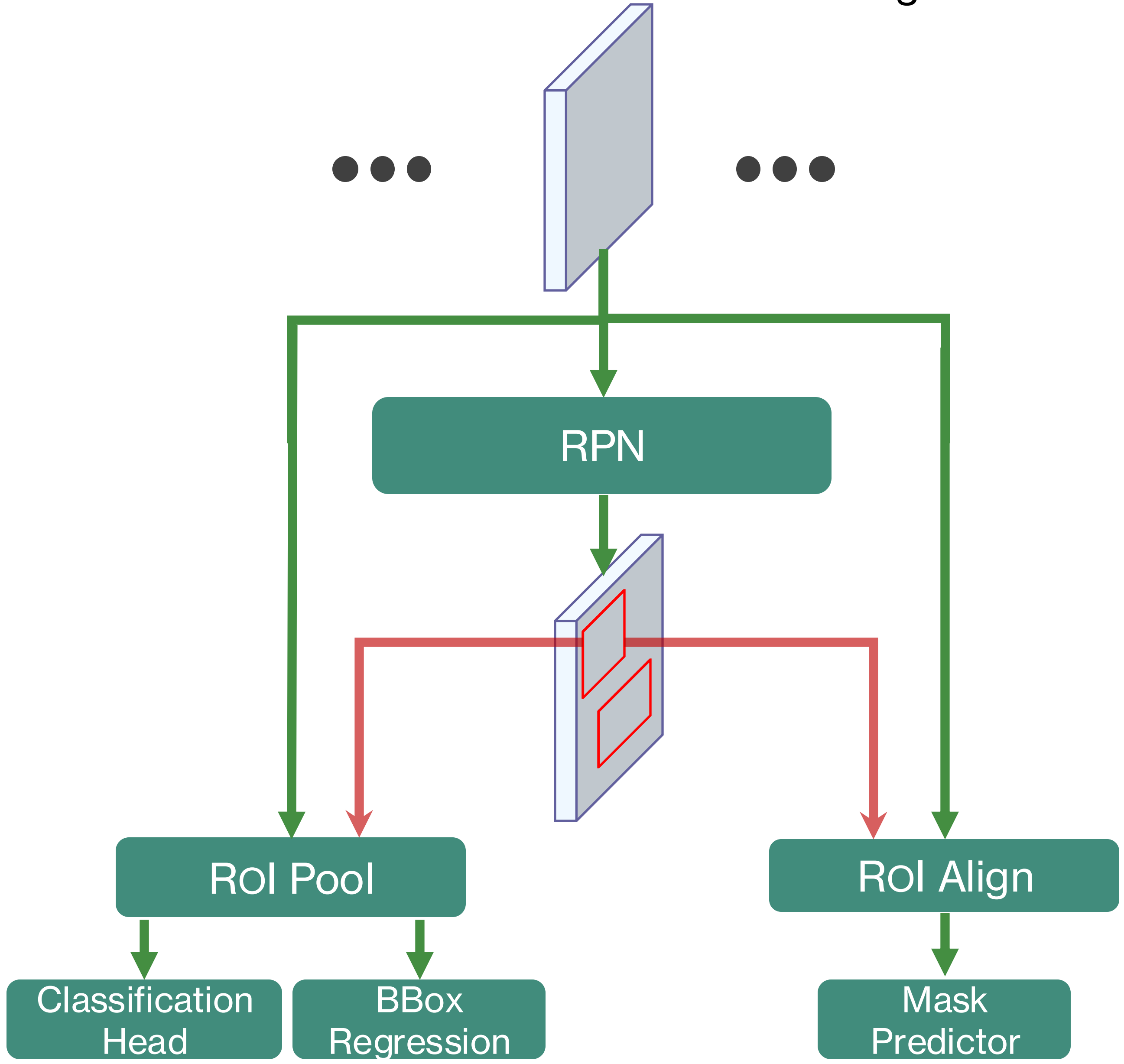
Faster R-CNN

Only detection

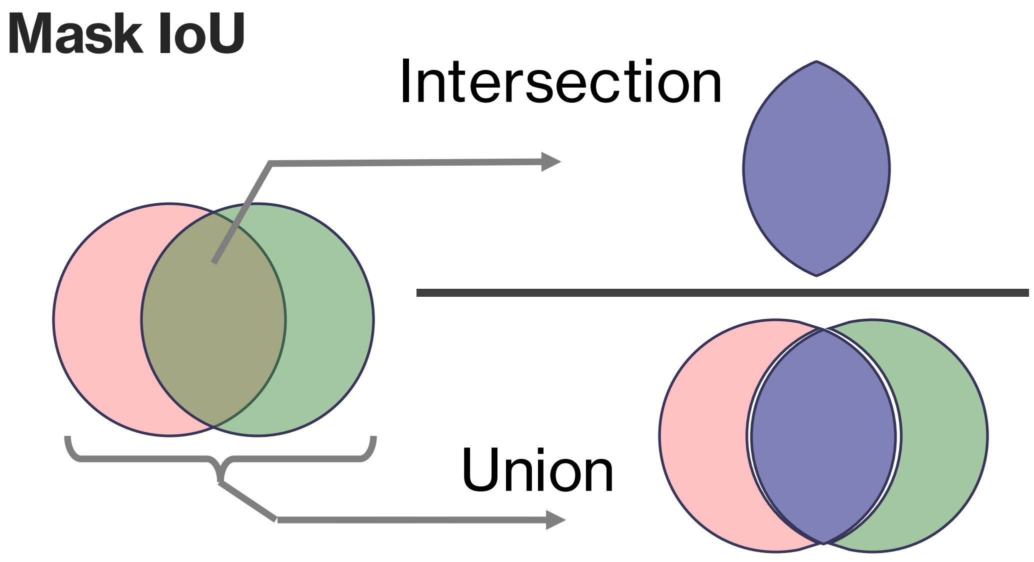


Mask R-CNN

Detection & segmentation



Performance metrics for instance segmentation



(Mask) Average Precision: $AP^{k}_{IoU=0.5}$

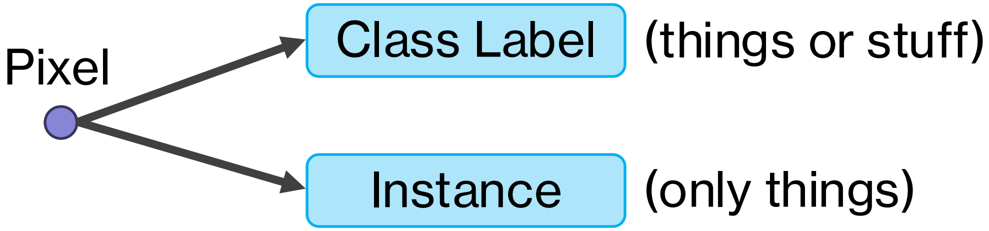
- For each class k, assign predicted masks to GT masks based on their Mask-IoU ➔ determine TP, FP, FN
- Sort predictions based on their confidences
- Construct precision-recall curve and calculate area under the curve

Mean Average Precision (mAP)

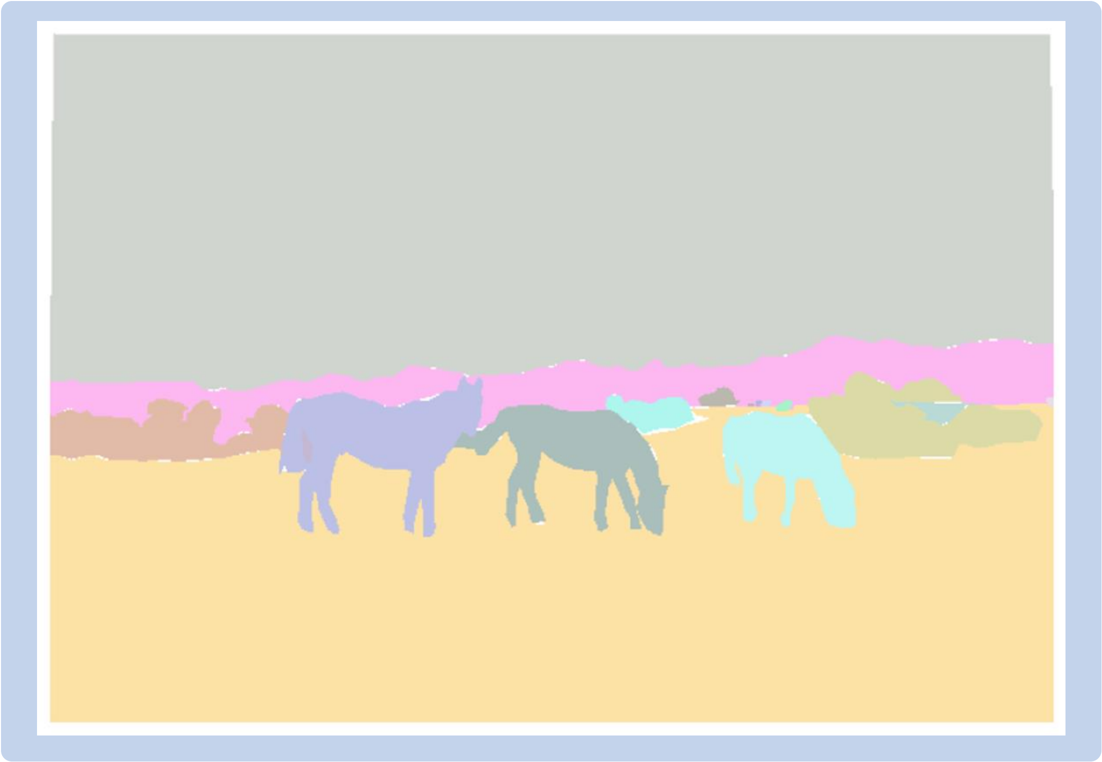
- Mean of average precisions for all classes

Panoptic Segmentation

Panoptic segmentation

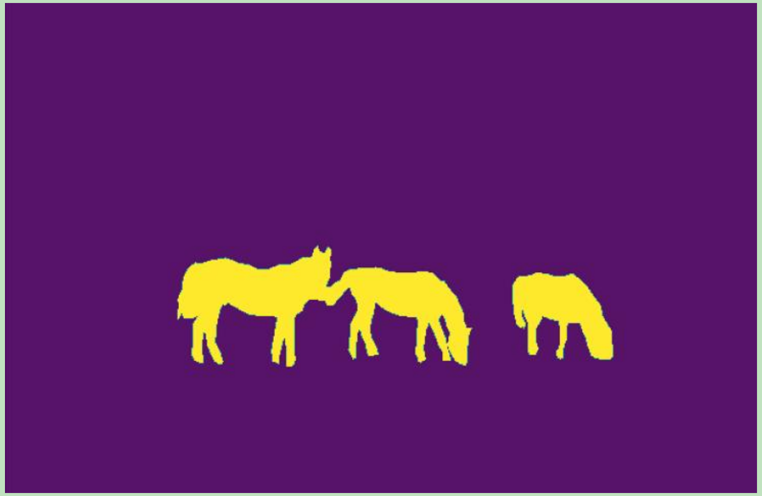


- Integrates **semantic** and **instance** segmentation
- Applicable to both things and stuff



Segmentation Tasks

Semantic

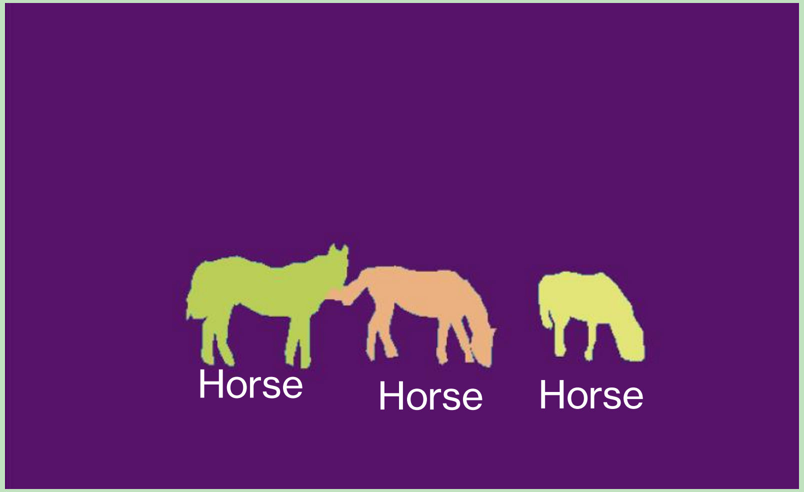


Classes color legend

Horse

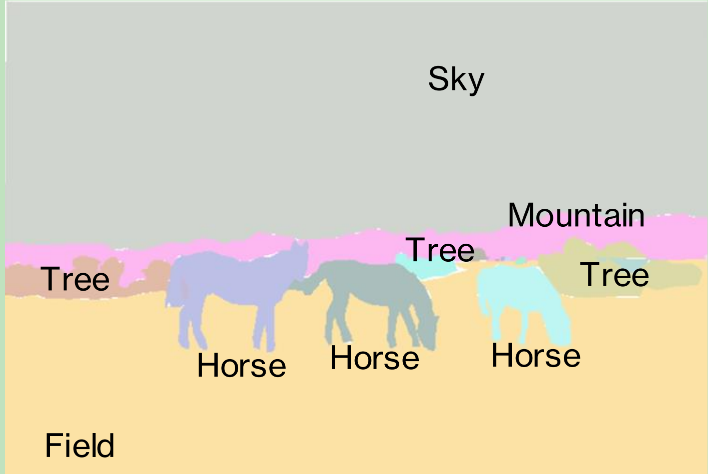
Background

Instance



Instances of countable objects (things)

Panoptic



Things and stuff

Panoptic segmentation metrics

Panoptic Quality (PQ)

Matching predicted and ground truth segments based on $IoU > 0.5$ → TP_k, FP_k, FN_k

$$PQ_k = \frac{\sum_{p,g \in TP_k} IoU(p,g)}{|TP_k| + \frac{1}{2}|FP_k| + \frac{1}{2}|FN_k|}$$

Panoptic segmentation metrics

Segmentation Quality (SQ)

Recognition Quality (RQ)

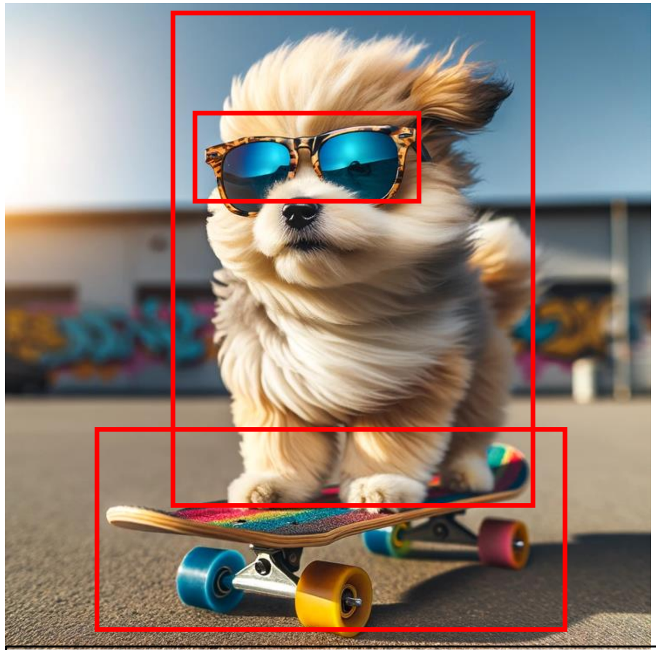
$$PQ_k = \frac{\sum_{p,g \in TP_k} IoU(p,g)}{|TP_k|} \times \frac{|TP_k|}{|TP_k| + \frac{1}{2}|FP_k| + \frac{1}{2}|FN_k|}$$

Decomposition provides further insights for error analysis

Visual Grounding

Visual Grounding

- Links language descriptions to specific objects or regions in visual data, enhancing interpretability and interaction.
- Identifies specific image regions corresponding to textual descriptions.



“a cool dog skateboarding”

Visual Grounding

- Understanding the **context**
- Identifying the **objects**
- Identifying the **relationship** between objects
- Provide a **description** of the image in natural language

Summary: Overview of object recognition tasks

- **Classification:** Assigns a label to the entire image (or multiple labels in multi-label).
- **Object Localization:** Identifies a **single** object's locations with a bounding box.
- **Object Detection:** Combines classification and localization for **multiple** objects.
- **Semantic Segmentation:** **Pixel-wise** classification.
- **Instance Segmentation:** Combining detection and segmentation (only for things).
- **Panoptic Segmentation:** Segmenting stuff and things.