EVML

ADVANCED CNN

JEROEN VEEN

HAN_UNIVERSITY
OF APPLIED SCIENCES

DL REPORT TEMPLATE

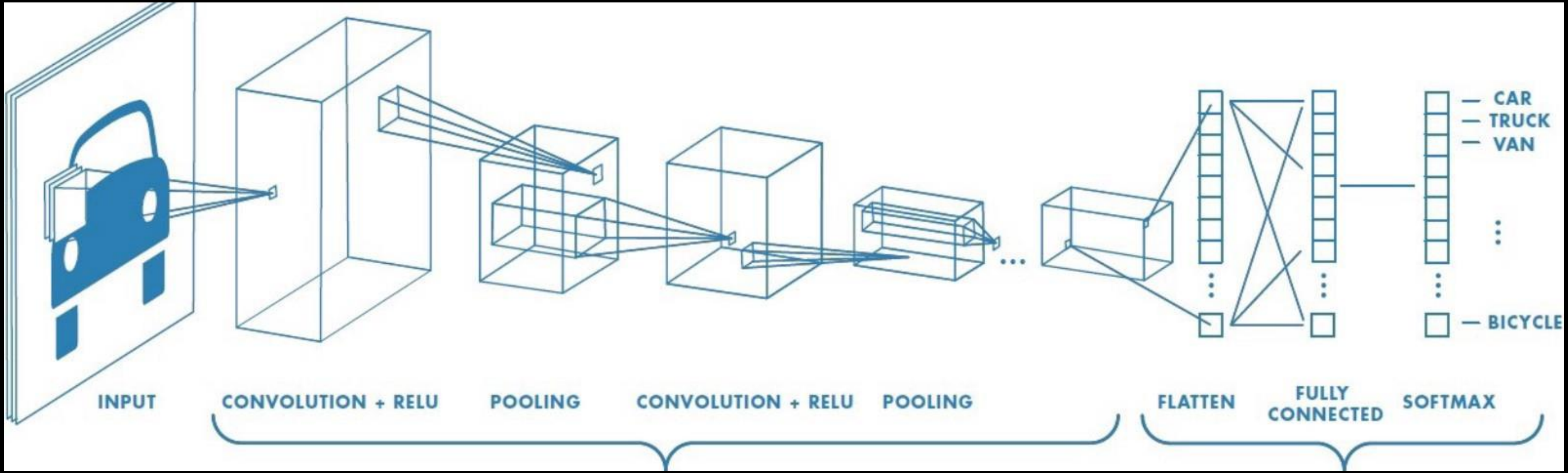
- Questions?

AGENDA

- Object detection
- Object tracking
- Semantic segmentation
- Variational autoencoder

TYPICAL CNN ARCHITECTURE

- Perform classification
- Conv, pool, dense layers



Source: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

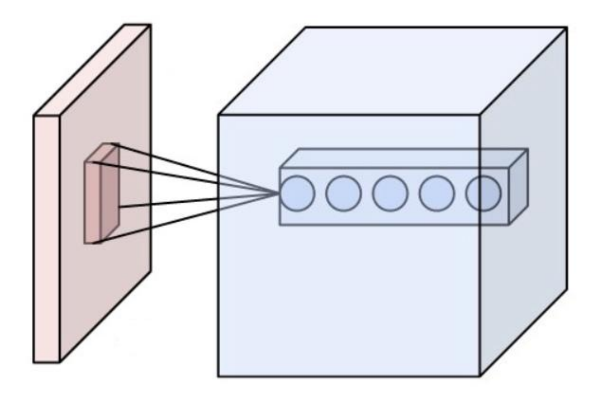
CNN SUMMARY

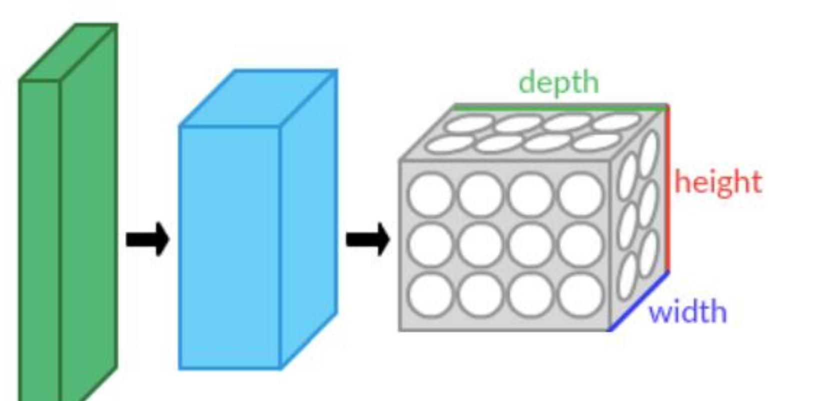
- Emulate the behavior of a visual cortex (e.g. receptive fields)
- Higher-level representations of image content
- No feature definition, but automated extraction

- Biologically inspired perceptrons
- Multilayer perceptrons usually mean fully connected networks, which makes them prone to overfitting
- CNNs can be considered as regularized versions of multilayer perceptrons

CONVOLUTIONAL LAYER SUMMARY

- Local connectivity
- Shared weights
- 3D volumes of neurons
- Output is a stack if feature maps





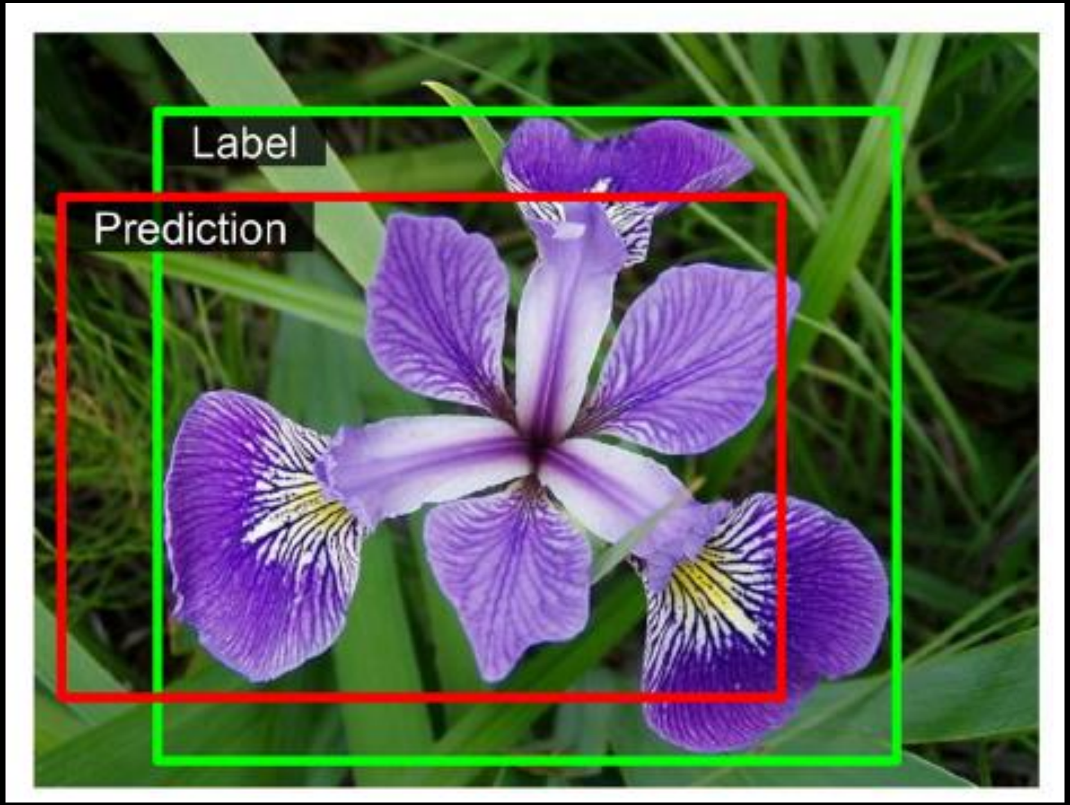
By Aphex34 - Own work, CC BY-SA 4.0,
https://commons.wikimedia.org/w/index.php?curid=45659236

MORE TRANSFER LEARNING HINTS

- https://keras.io/api/applications

CLASSIFICATION AND LOCALIZATION

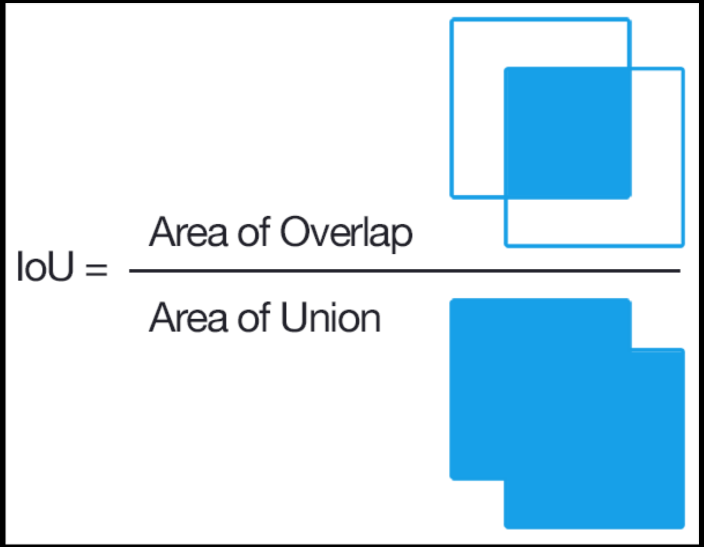
- Add second dense output layer to predict coordinates (regression)

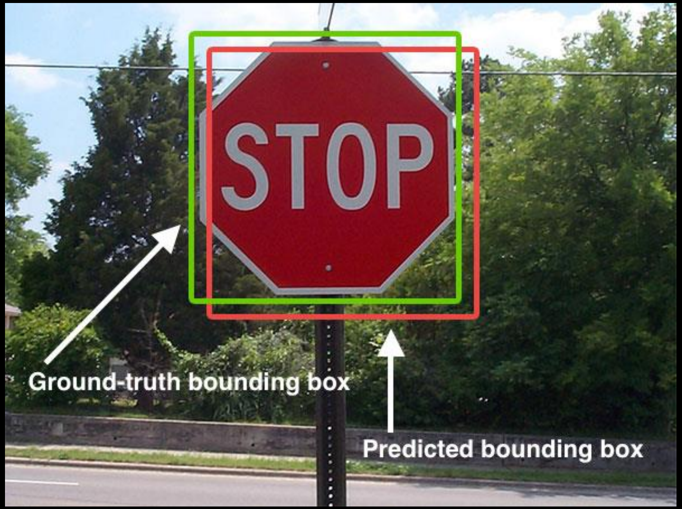


Source: Géron, ISBN: 9781492032632

JACCARD INDEX

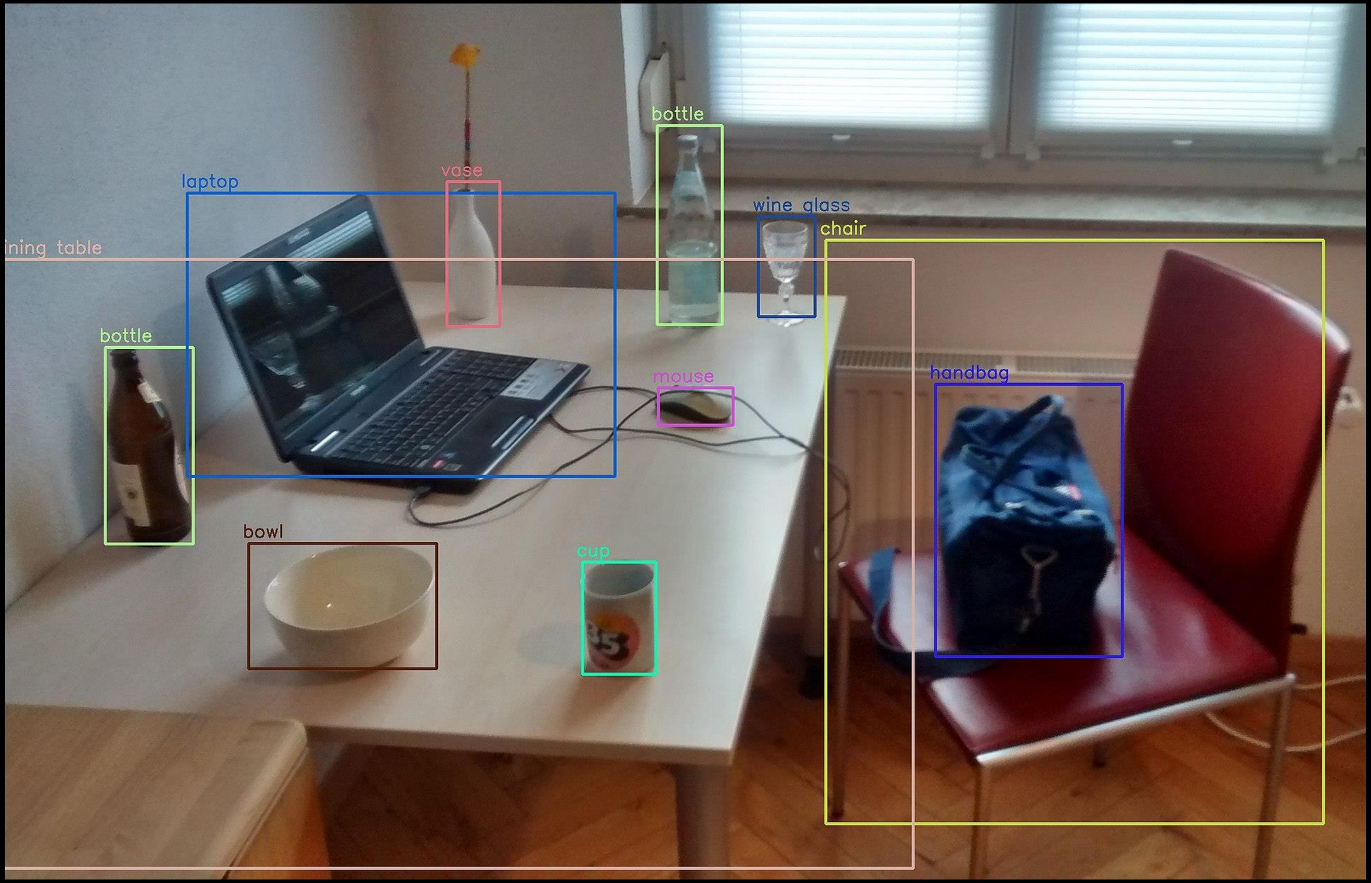
- Performance metric: intersection over Union (IoU)
- measures similarity between finite sample sets





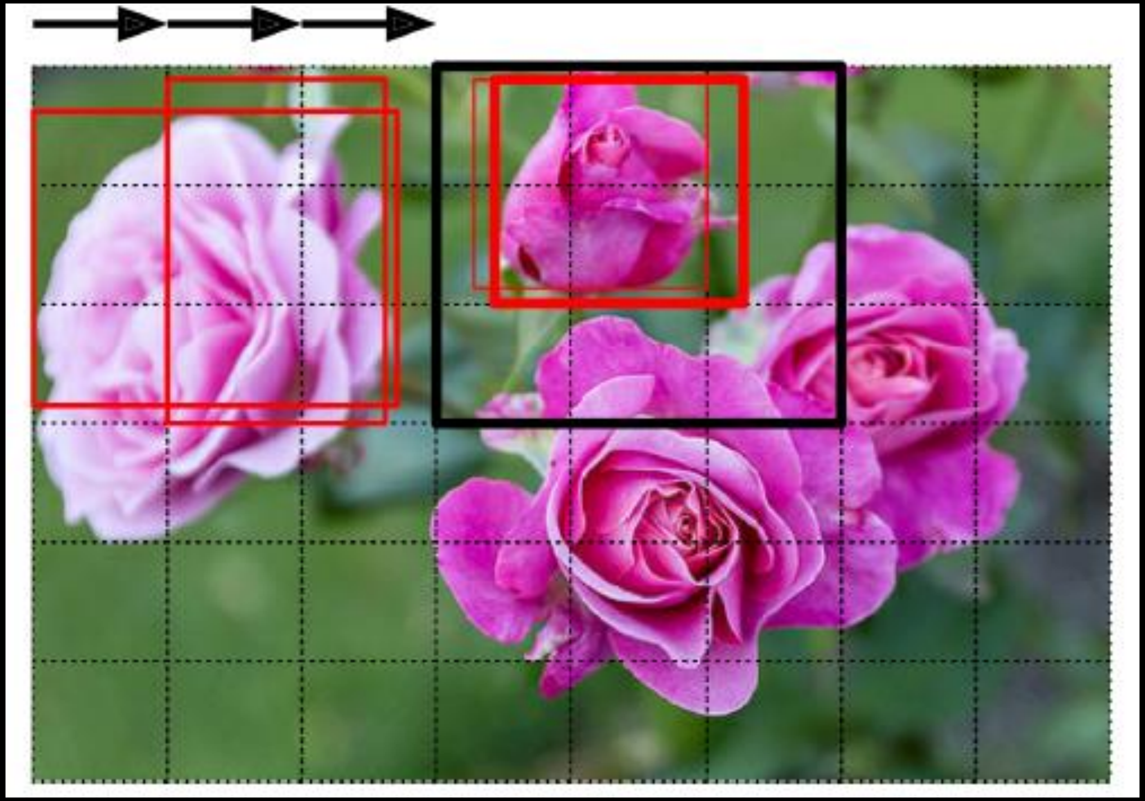
Source: By Adrian Rosebrock - http://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=57718560

OBJECT DETECTION



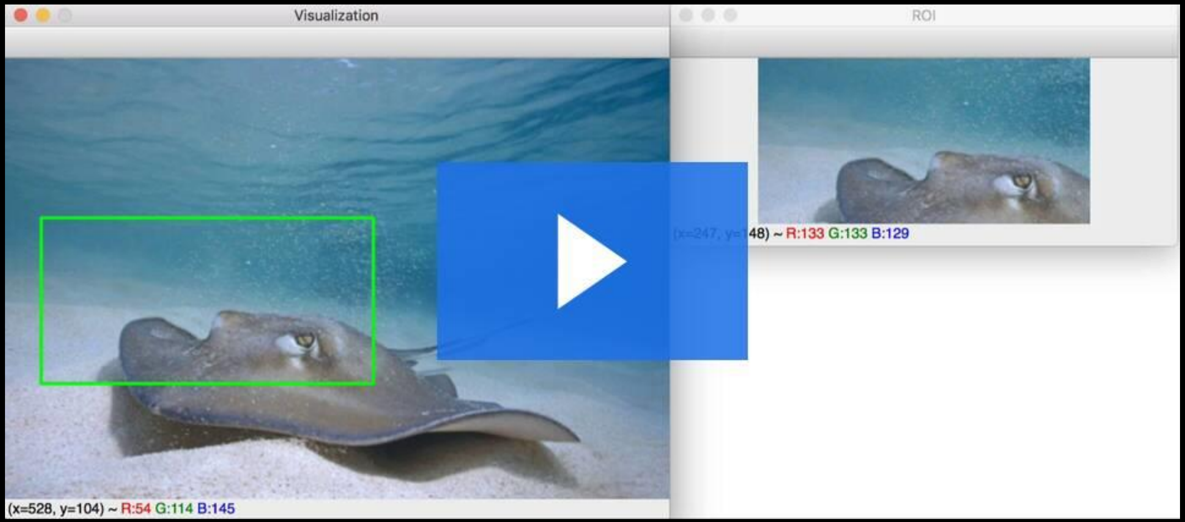
By (MTheiler) - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=75843378

SLIDING WINDOW



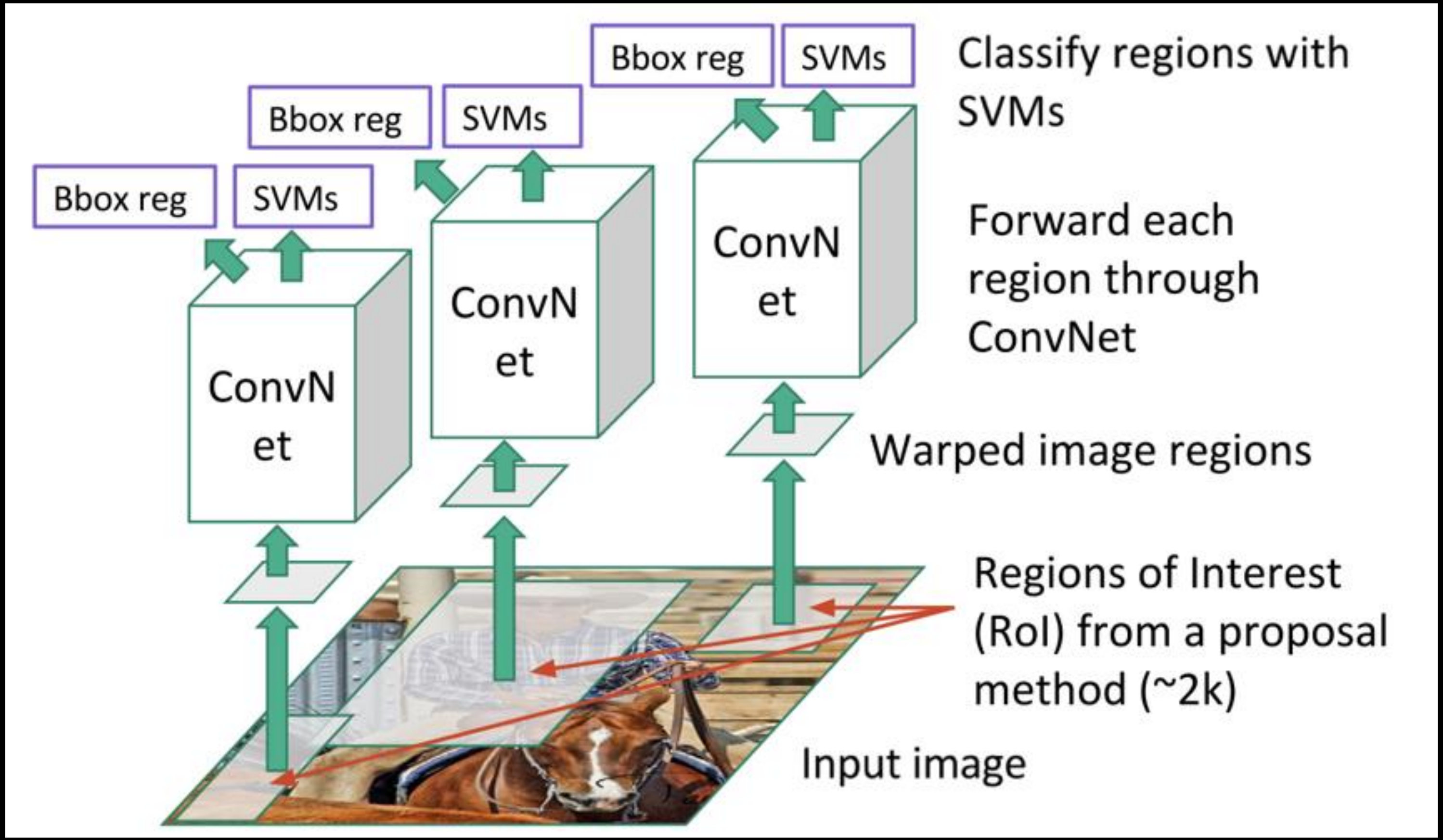
Source: Géron, ISBN: 9781492032632

SLIDING WINDOWS


Visualization
ROI
(x=247, y=148) ~ R:133 G:133 B:129
(x=528, y=104) ~ R:54 G:114 B:145

Source: https://www.pyimagesearch.com/2020/06/22/turning-any-cnn-image-classifier-into-an-object-detector-with-keras-tensorflow-and-opencv/

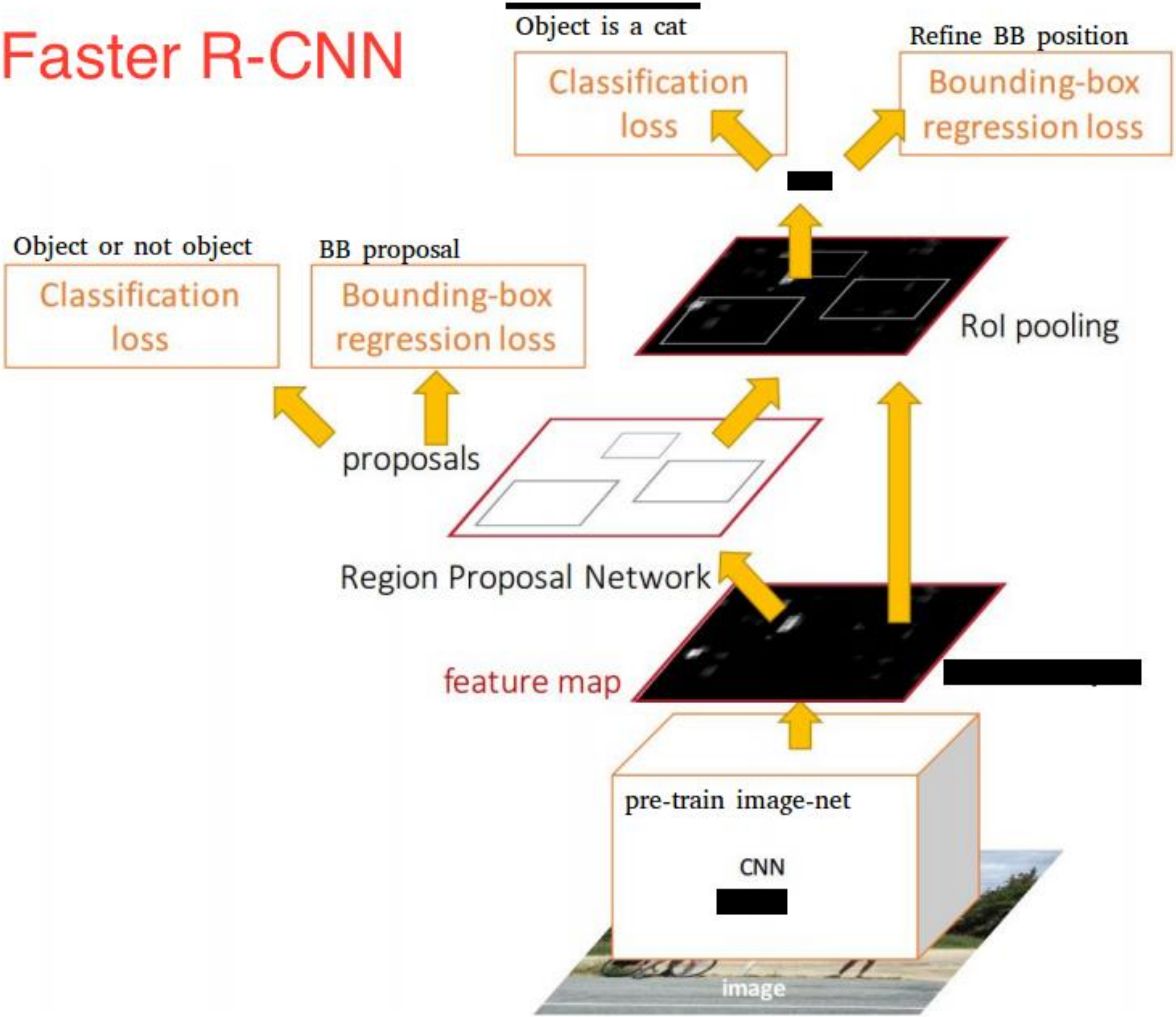
REGION-BASED CNN


Bbox reg
SVMs
Bbox reg
SVMs
Bbox reg
SVMs
Classify regions with SVMs
ConvNet
ConvNet
ConvNet
Forward each region through ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image

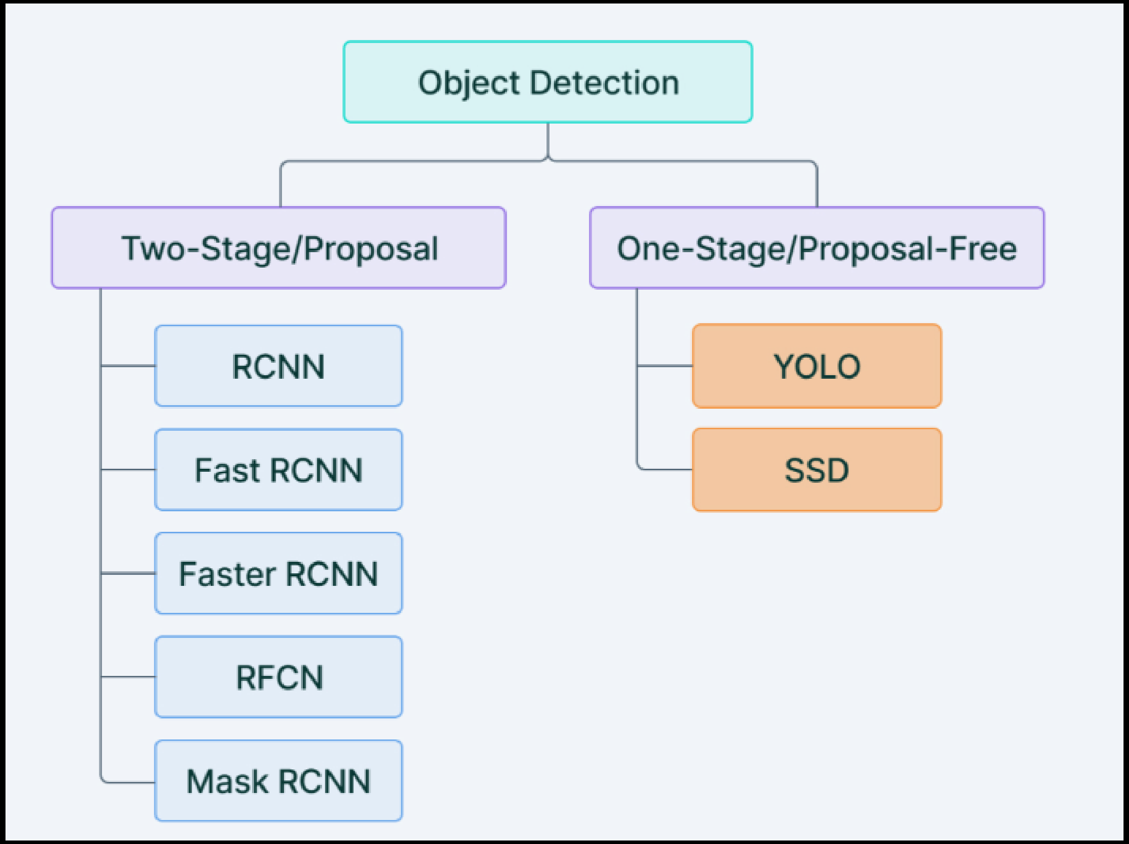
FPN: FEATURE PYRAMID NETWORK

- FPN: Feature Pyramid Network (2016) - KiKaBeN

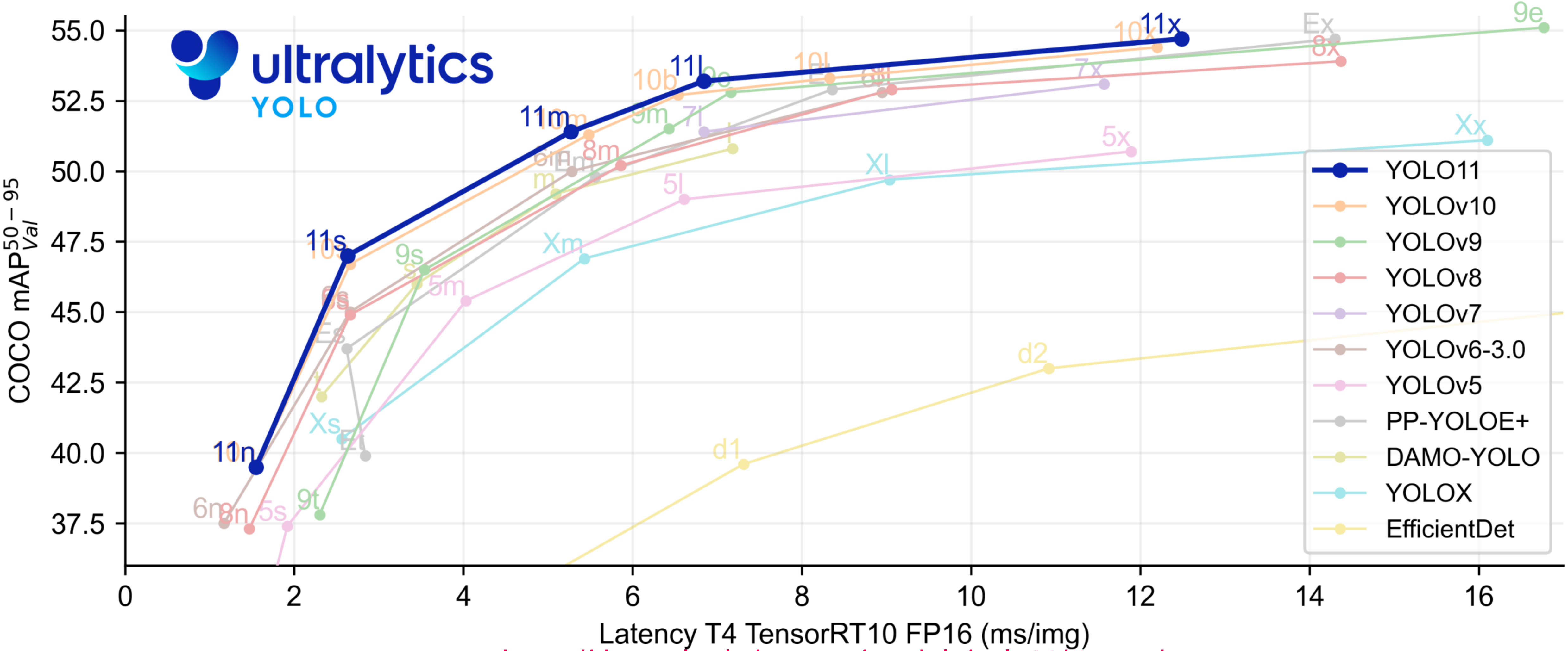
Faster R-CNN


[redacted]
Object is a cat
Classification loss
Refine BB position
Bounding-box regression loss
[redacted]
Object or not object
Classification loss
BB proposal
Bounding-box regression loss
RoI pooling
proposals
Region Proposal Network
feature map
[redacted]
pre-train image-net
CNN
[redacted]
image

OBJECT DETECTION METHODS



YOLO V9



https://docs.ultralytics.com/models/yolo11/#overview

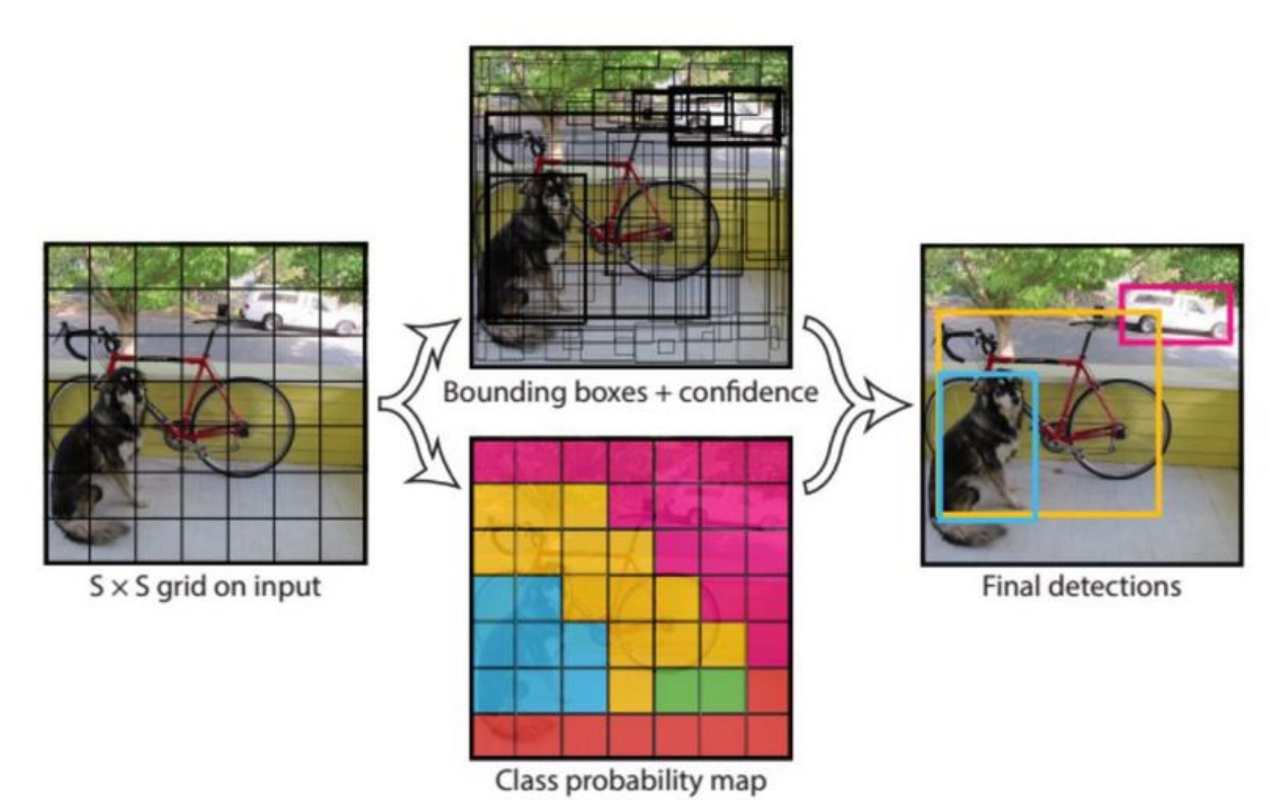
YOLO

- Intro: https://youtu.be/LvArU0AH8s8?t=35s

- Explanation: https://www.youtube.com/watch?v=svn9-xV7wjk&t=170s

  I’ll give a short version ☺

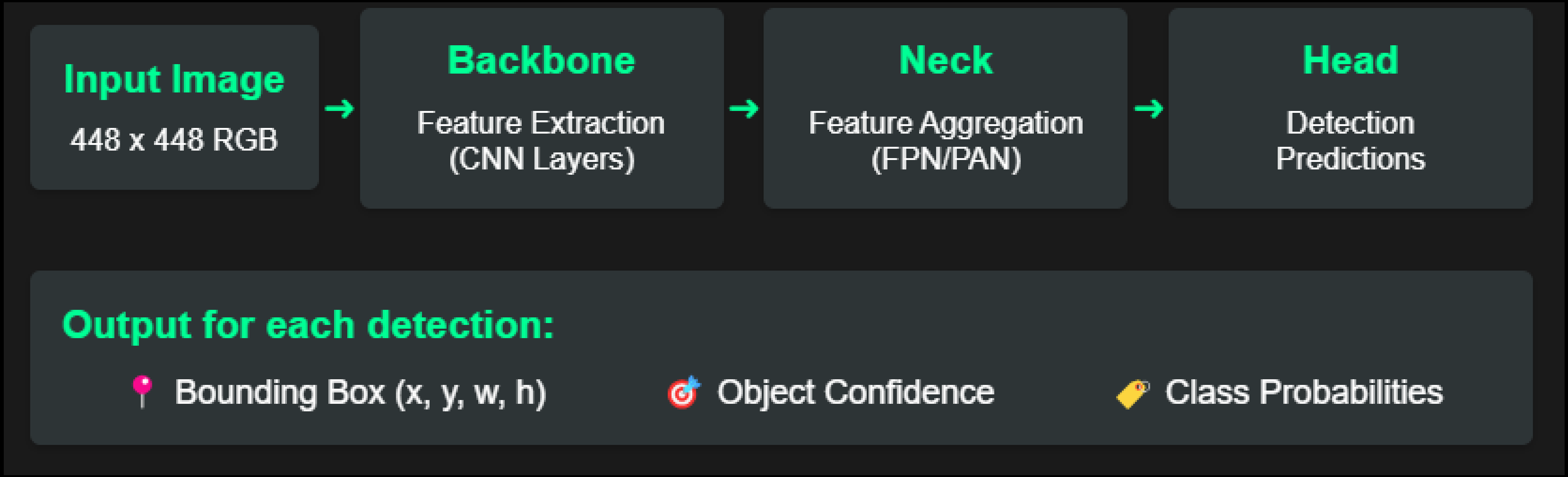
YOU ONLY LOOK ONCE (YOLO)



Source: Redmon et al., You Only Look Once:Unified, Real-Time Object Detection, 2016.

See: https://youtu.be/NM6lrxy0bxs?list=PLrrmP4uhN47Y-hWs7DVfCmLwUACRigYyT
https://pjreddie.com/darknet/yolo/

YOLO BASIC ARCHITECTURE



Input Image
448 x 448 RGB

→

Backbone
Feature Extraction
(CNN Layers)

→

Neck
Feature Aggregation
(FPN/PAN)

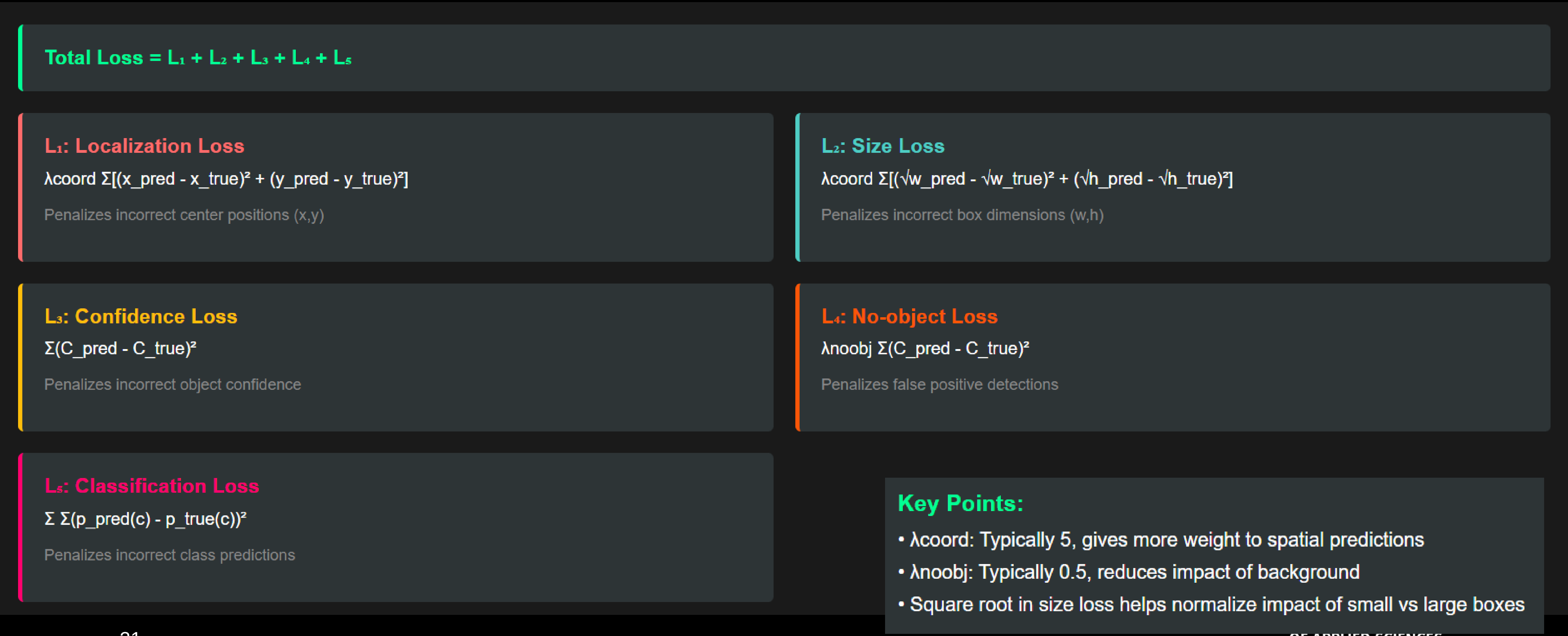
→

Head
Detection
Predictions

Output for each detection:

📍 Bounding Box (x, y, w, h) 🎯 Object Confidence 🏷️ Class Probabilities

LOSS FUNCTION



Total Loss = $L_1 + L_2 + L_3 + L_4 + L_5$

L_1: Localization Loss

λcoord $\Sigma[(x\_pred - x\_true)^2 + (y\_pred - y\_true)^2]$

Penalizes incorrect center positions (x,y)

L_2: Size Loss

λcoord $\Sigma[(\sqrt{}w\_pred - \sqrt{}w\_true)^2 + (\sqrt{}h\_pred - \sqrt{}h\_true)^2]$

Penalizes incorrect box dimensions (w,h)

L_3: Confidence Loss

$\Sigma(C\_pred - C\_true)^2$

Penalizes incorrect object confidence

L_4: No-object Loss

λnoobj $\Sigma(C\_pred - C\_true)^2$

Penalizes false positive detections

L_5: Classification Loss

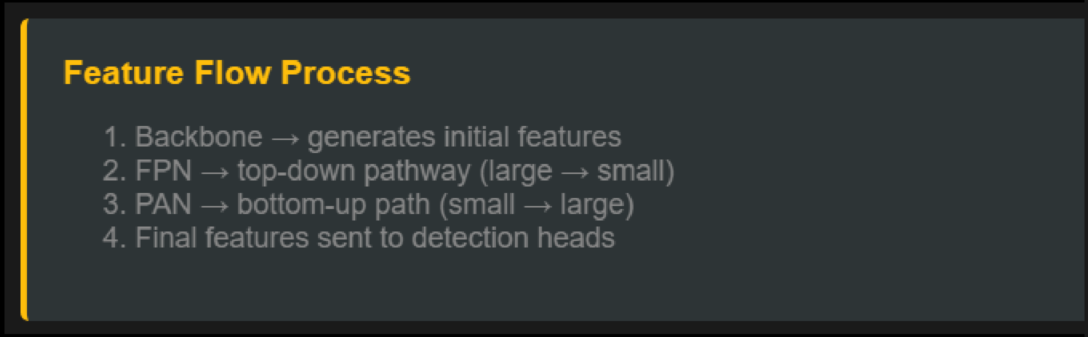
$\Sigma\ \Sigma(p\_pred(c) - p\_true(c))^2$

Penalizes incorrect class predictions

Key Points:

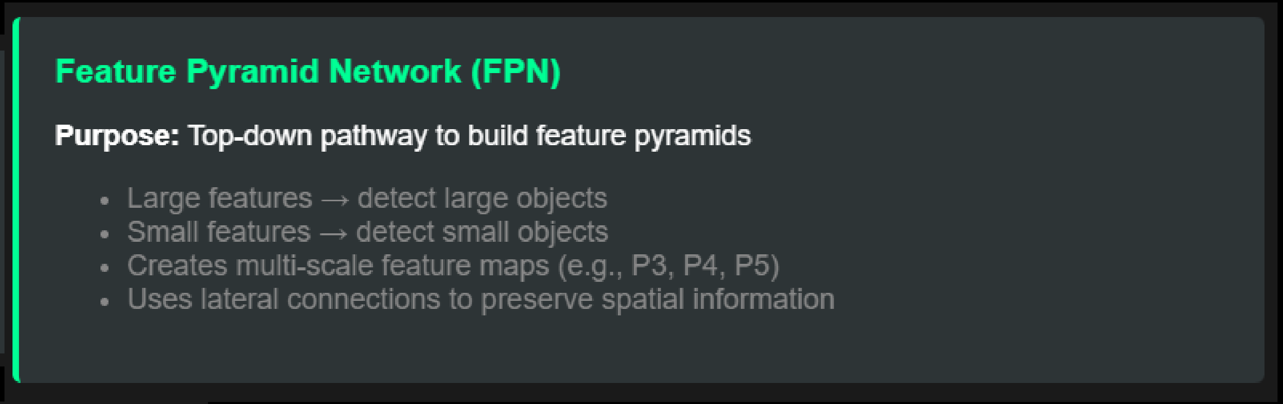
- λcoord: Typically 5, gives more weight to spatial predictions
- λnoobj: Typically 0.5, reduces impact of background
- Square root in size loss helps normalize impact of small vs large boxes

NECK



Feature Flow Process

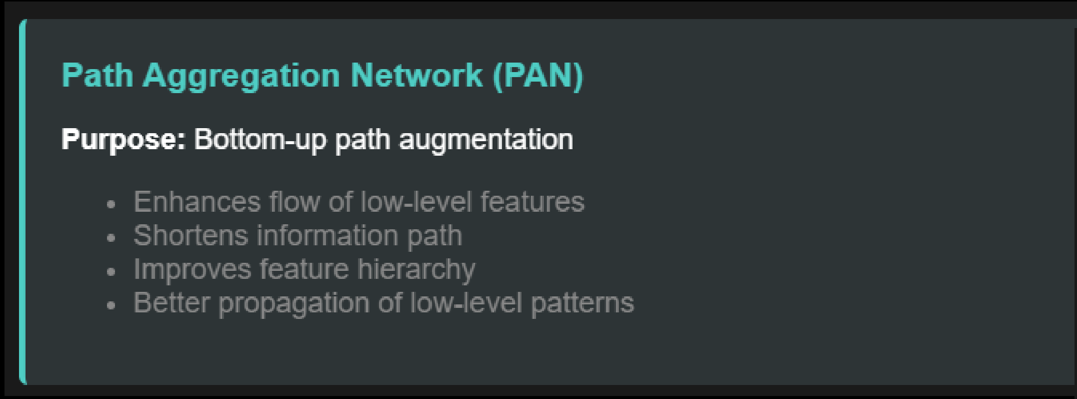
1. Backbone → generates initial features
2. FPN → top-down pathway (large → small)
3. PAN → bottom-up path (small → large)
4. Final features sent to detection heads



Feature Pyramid Network (FPN)

Purpose: Top-down pathway to build feature pyramids

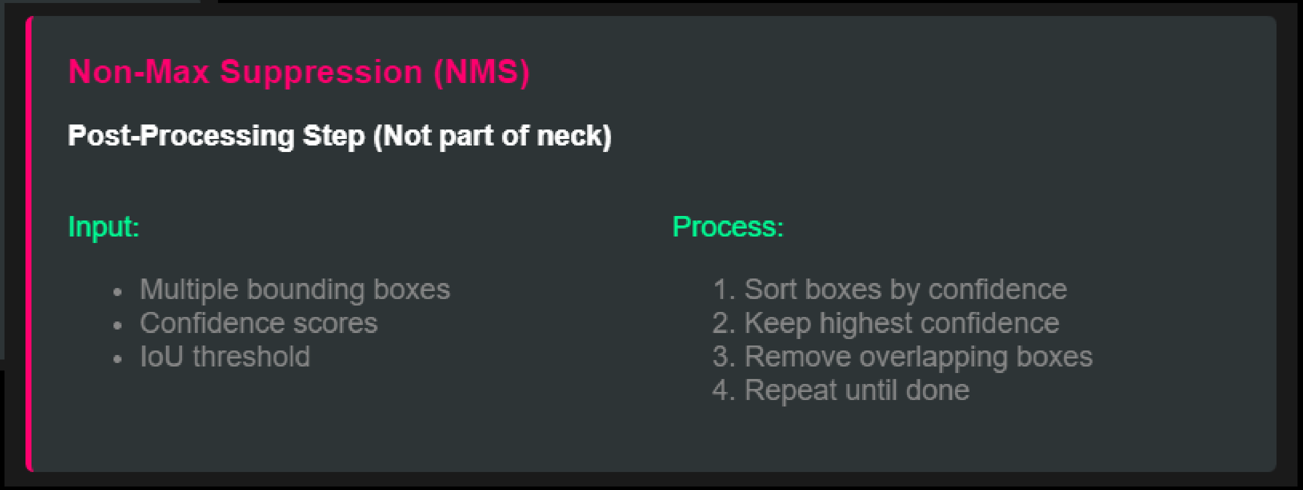
- Large features → detect large objects
- Small features → detect small objects
- Creates multi-scale feature maps (e.g., P3, P4, P5)
- Uses lateral connections to preserve spatial information



Path Aggregation Network (PAN)

Purpose: Bottom-up path augmentation

- Enhances flow of low-level features
- Shortens information path
- Improves feature hierarchy
- Better propagation of low-level patterns



Non-Max Suppression (NMS)

Post-Processing Step (Not part of neck)

Input:

- Multiple bounding boxes
- Confidence scores
- IoU threshold

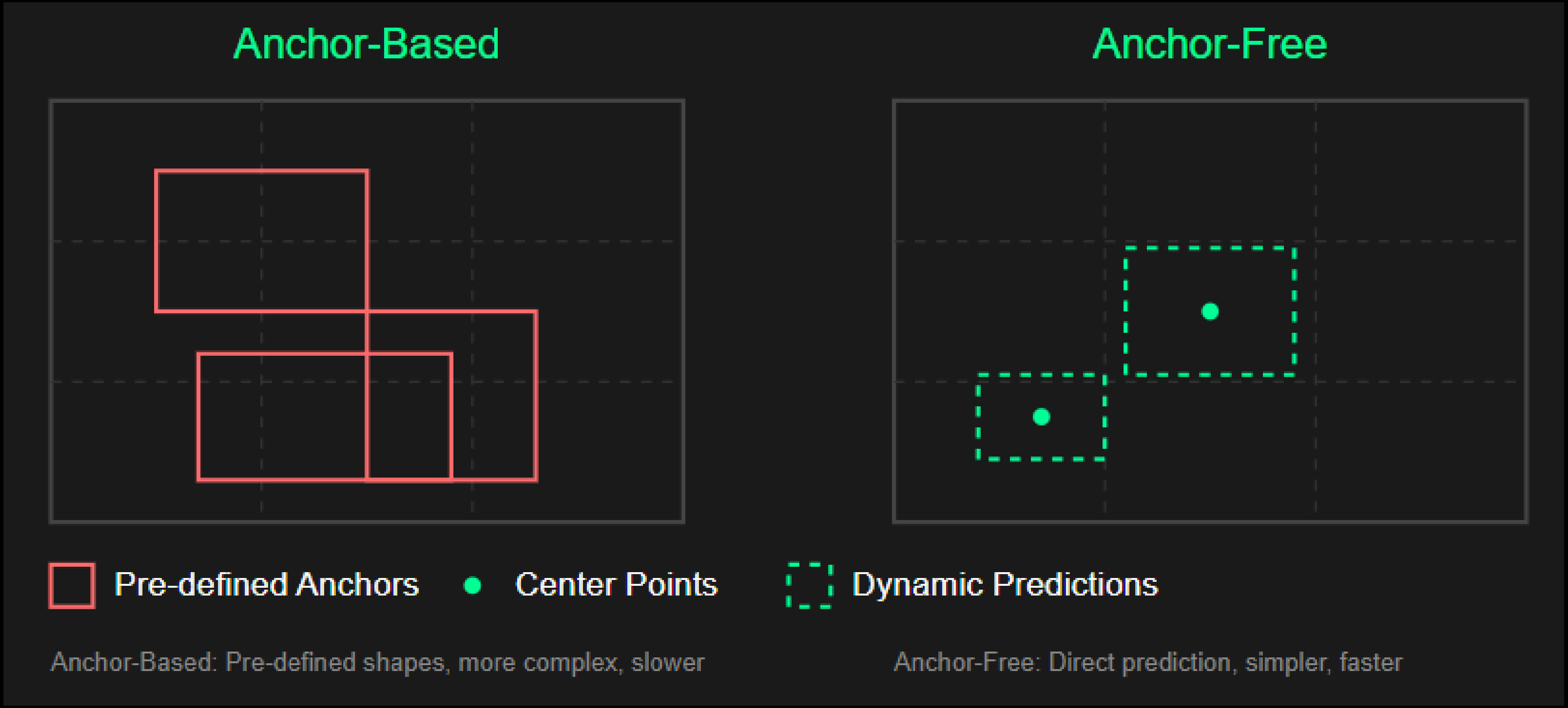
Process:

1. Sort boxes by confidence
2. Keep highest confidence
3. Remove overlapping boxes
4. Repeat until done

KEY TECHNICAL IMPROVEMENTS ACROSS VERSIONS:

- Early Versions (v1-v3)
  - v1: Single-stage detection
  - v2: Batch norm & anchor boxes
  - v3: Multi-scale predictions
- Middle Era (v4-v7)
  - v4: CSP & Mosaic augmentation
  - v5: Adaptive anchors
  - v7: Dynamic head & label assignment
- Modern Era (v8-v11)
  - V8: Anchor-free detection
  - v9: Programmable gradients
  - v11: Transformer integration

ANCHOR-BASED VS ANCHOR-FREE APPROACHES

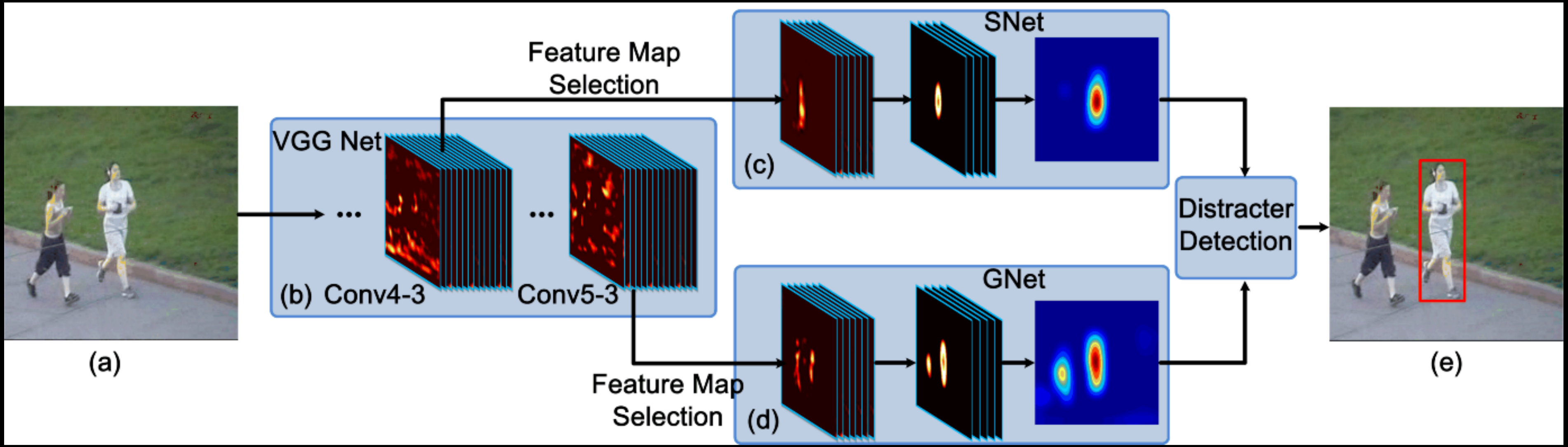


Anchor-Based: Pre-defined shapes, more complex, slower

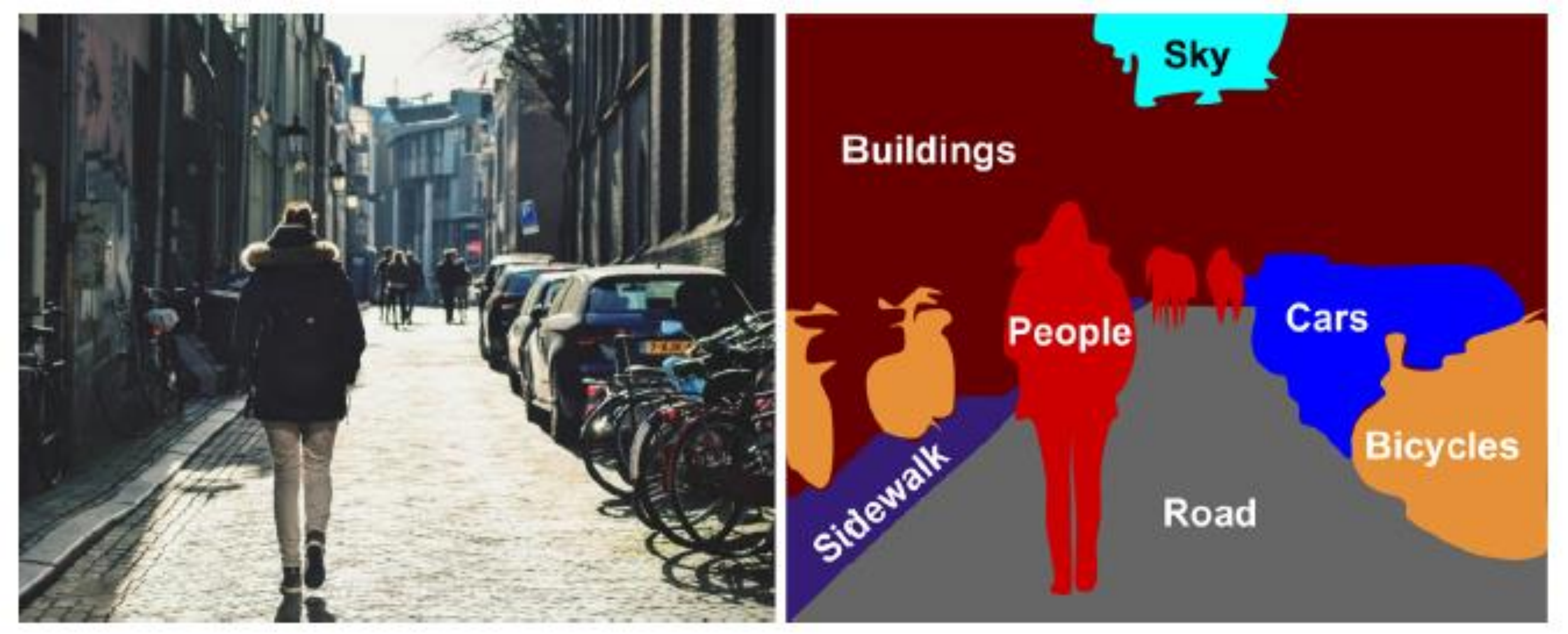
Anchor-Free: Direct prediction, simpler, faster

OBJECT TRACKING WITH CNN

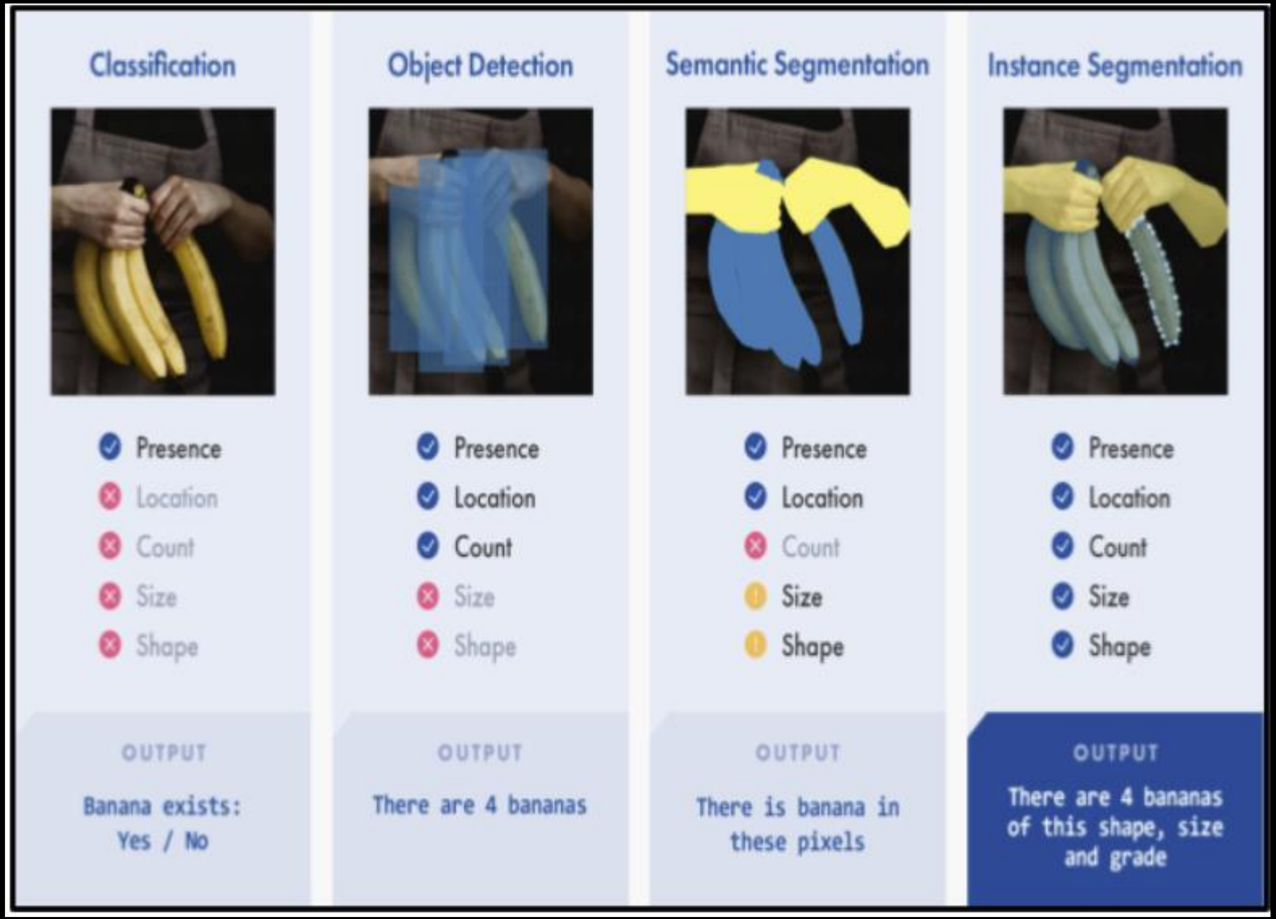
- Fully-convolutional network tracker (FCNT)



SEMANTIC SEGMENTATION

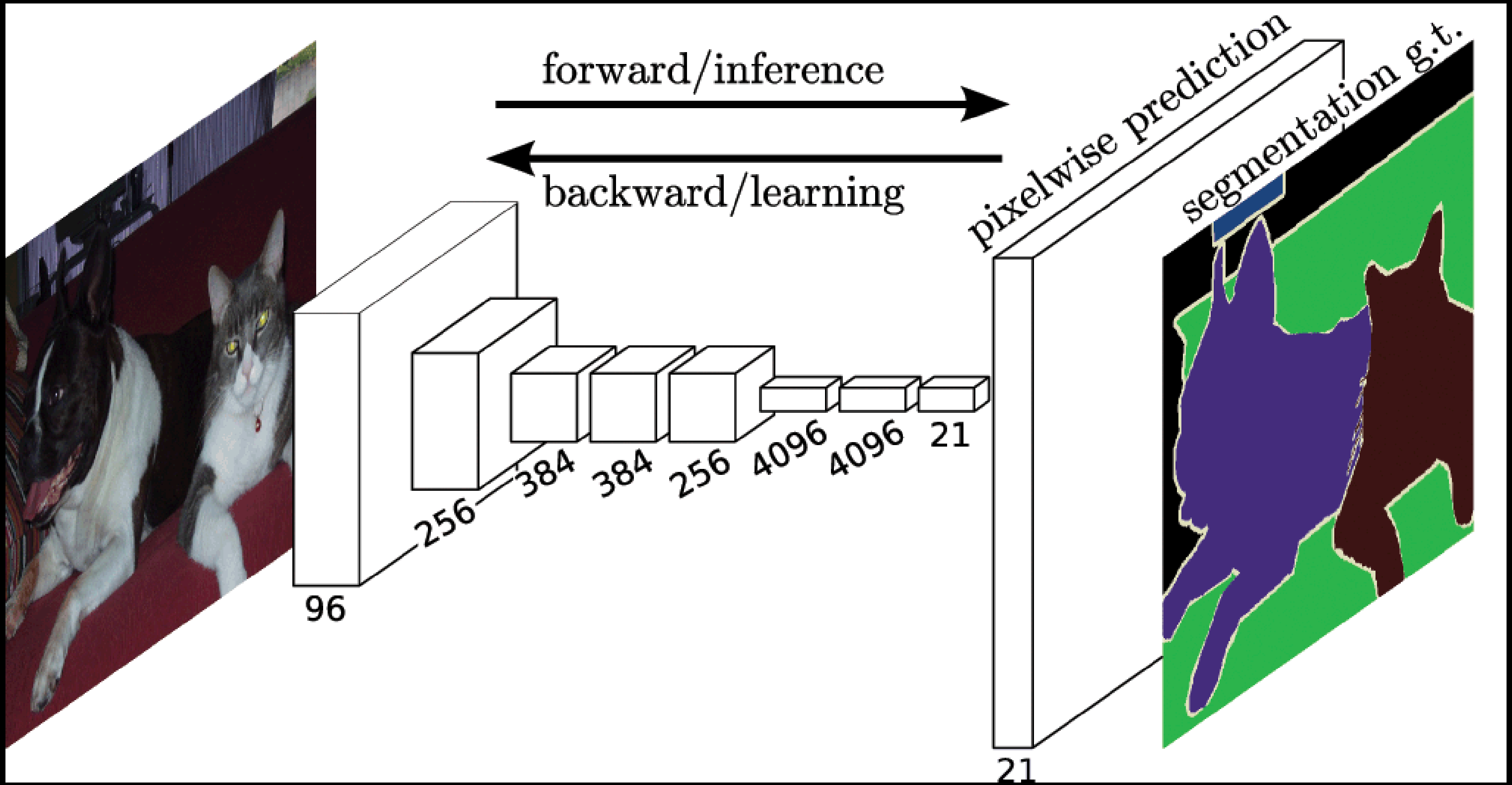


CLASSIFICATION VS. DETECTION VS. SEMANTIC SEGMENTATION VS INSTANCE SEGMENTATION



Classification	Object Detection	Semantic Segmentation	Instance Segmentation
✓ Presence	✓ Presence	✓ Presence	✓ Presence
✗ Location	✓ Location	✓ Location	✓ Location
✗ Count	✓ Count	✗ Count	✓ Count
✗ Size	✗ Size	! Size	✓ Size
✗ Shape	✗ Shape	! Shape	✓ Shape
OUTPUT	OUTPUT	OUTPUT	OUTPUT
Banana exists: Yes / No	There are 4 bananas	There is banana in these pixels	There are 4 bananas of this shape, size and grade

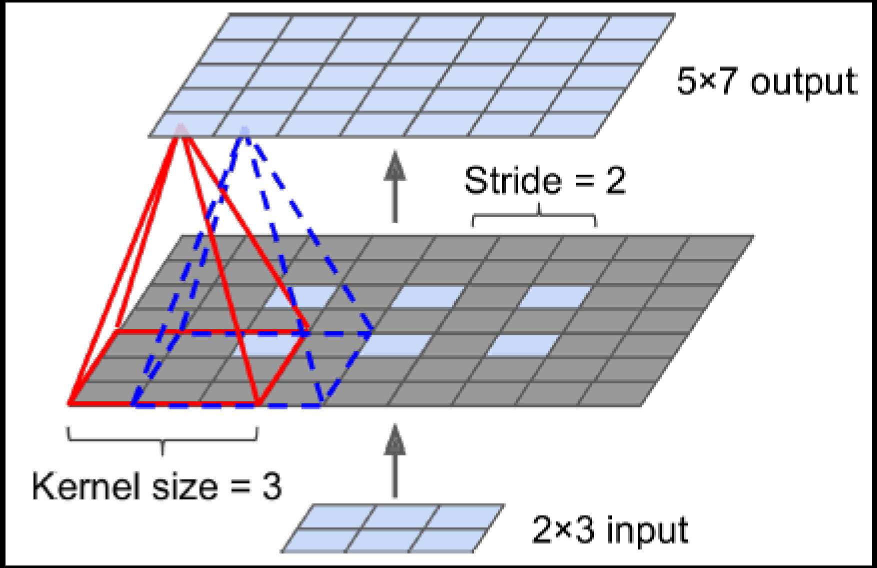
FULLY CONVOLUTIONAL NETWORKS



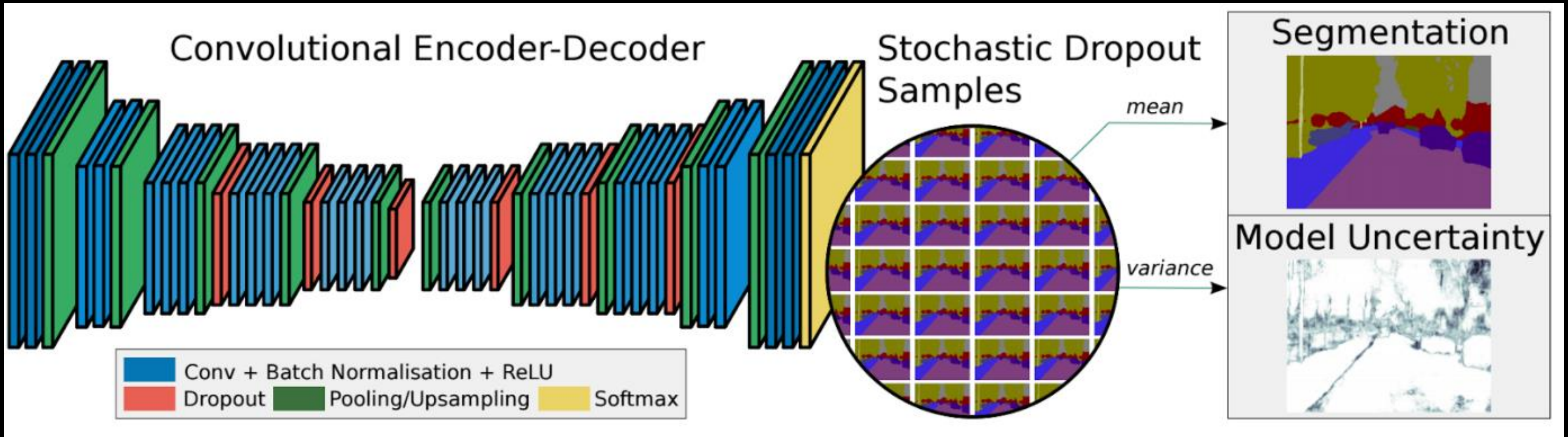
TRANSPOSED CONVOLUTION

- Upsampling layer to recover spatial information

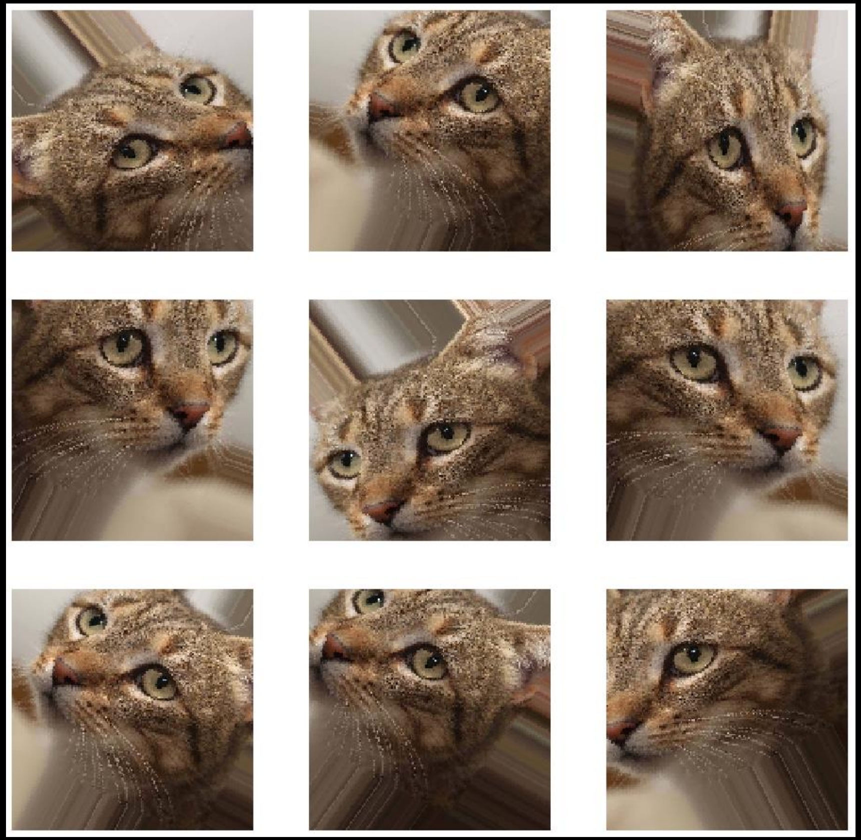
  1. stretching
  2. filtering



SEGNET

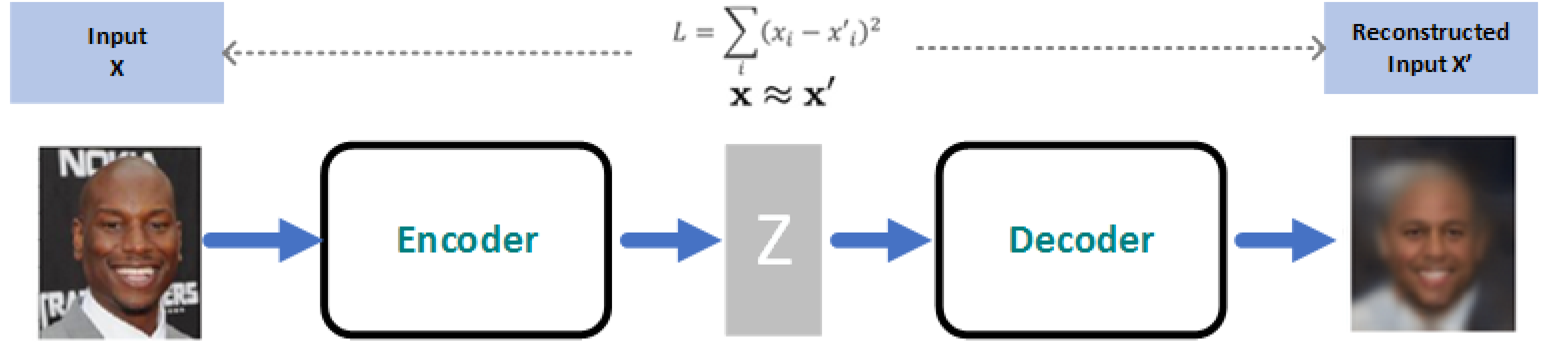

Convolutional Encoder-Decoder
Stochastic Dropout Samples
mean
variance
Segmentation
Model Uncertainty
Conv + Batch Normalisation + ReLU
Dropout
Pooling/Upsampling
Softmax

RECALL DATA AUGMENTATION



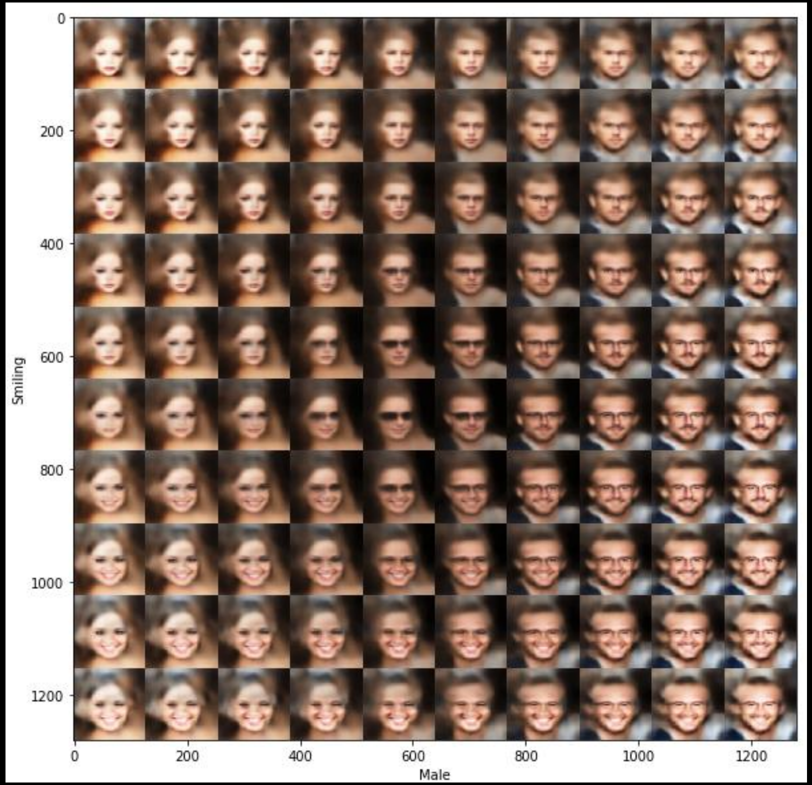
AUTOENCODERS

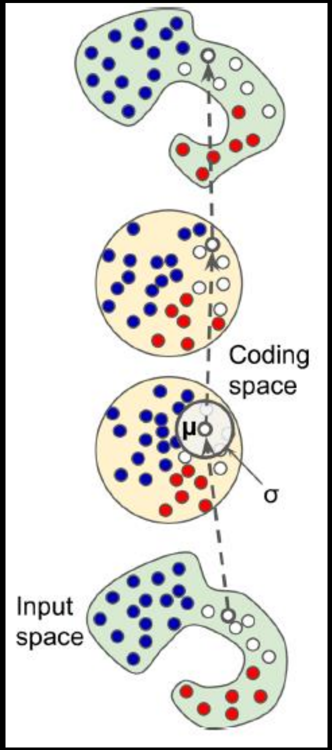
- Generating augmented data



CONDITIONAL VARIATIONAL AUTOENCODER

- Output determined by latent variables, chance and metadata.





HAN_UNIVERSITY OF APPLIED SCIENCES