Learning from Images

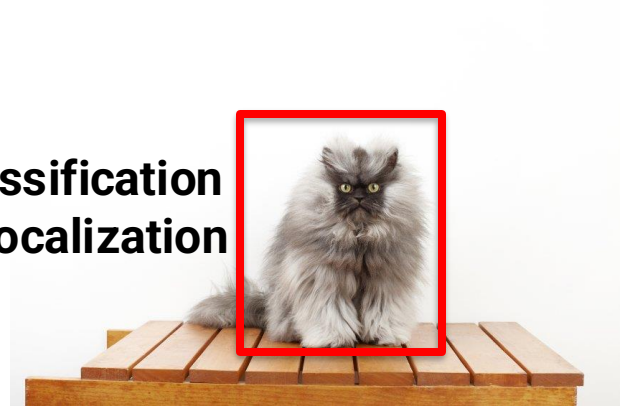
Image Segmentation and Object recognition

Master Data Science

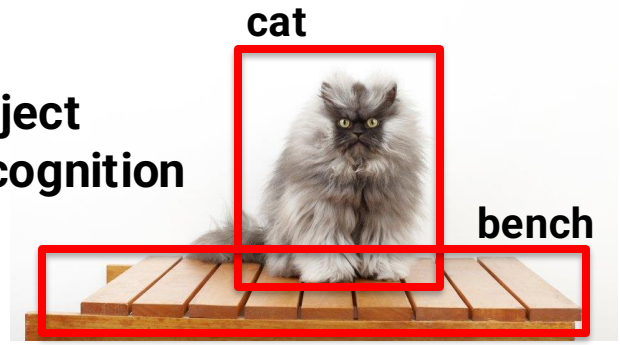
Prof. Dr. Kristian Hildebrand
khildebrand@bht-berlin.de

From classification to segmentation and recognition

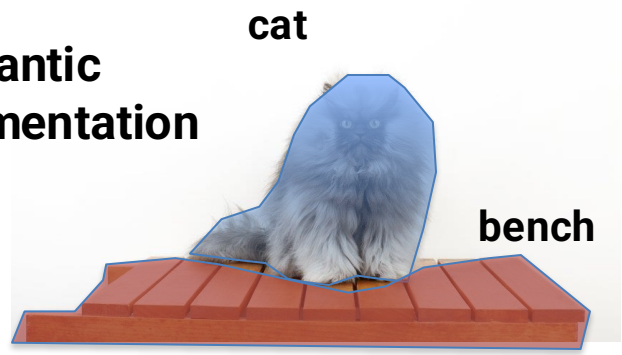
Classification
+ Localization



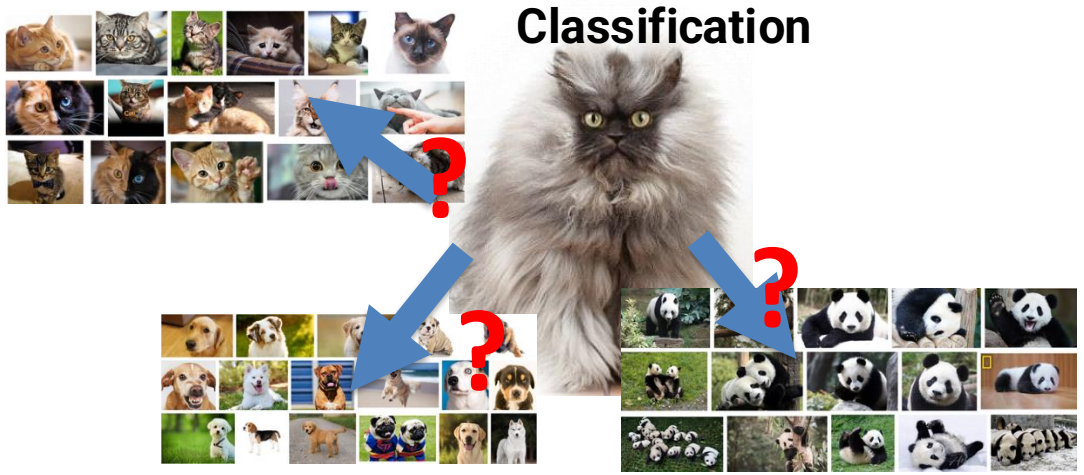
Object
recognition



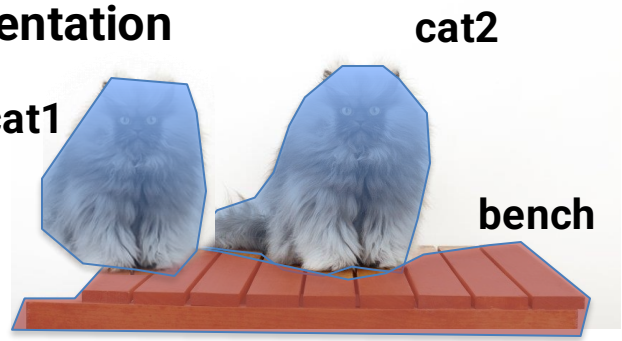
Semantic
Segmentation



Classification



Instance
Segmentation

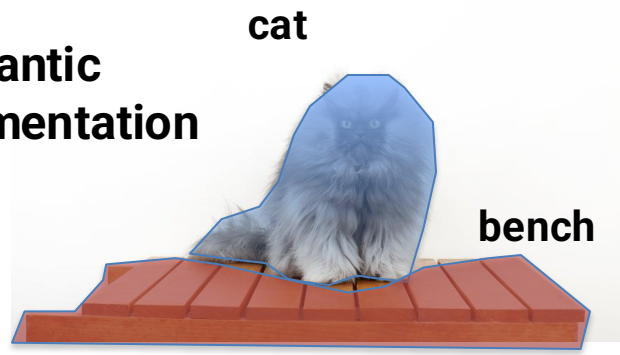


From classification to segmentation and recognition

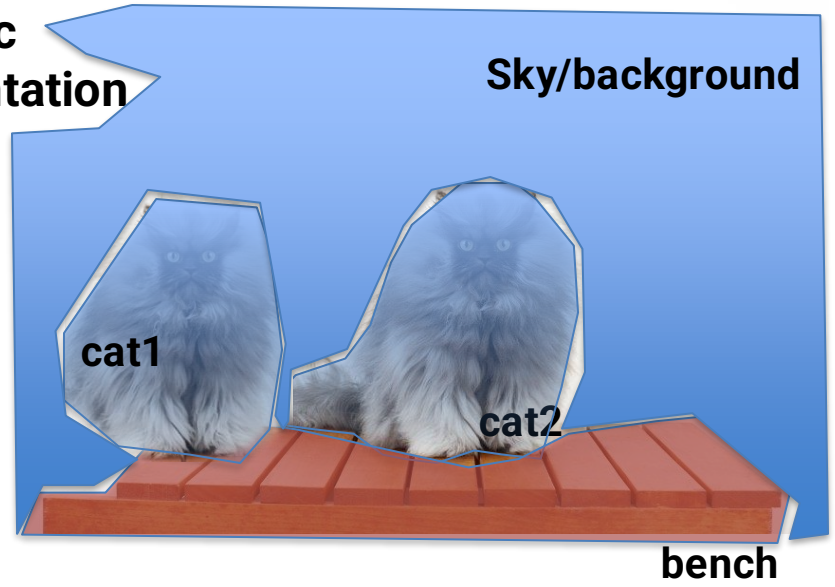
Classification
+ Localization

Object
recognition

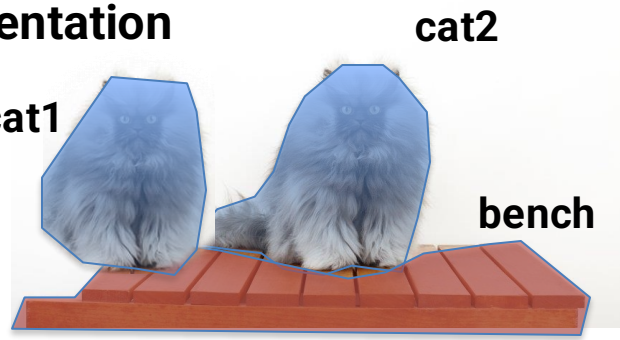
Semantic
Segmentation



Panoptic
Segmentation



Instance
Segmentation



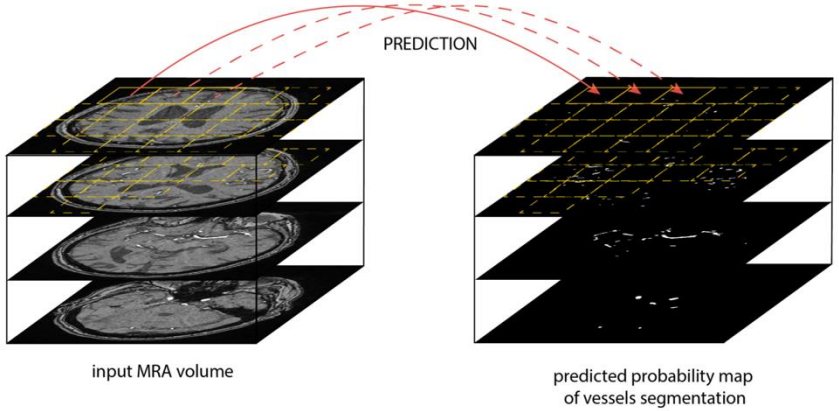
Label each pixel in the scene
Also with unknown or background.
Towards complete scene understanding

Applications

Autonomous Driving

Source: https://www.youtube.com/watch?v=ATlcEDSPWXY

Medical Imaging



Geo Sensing

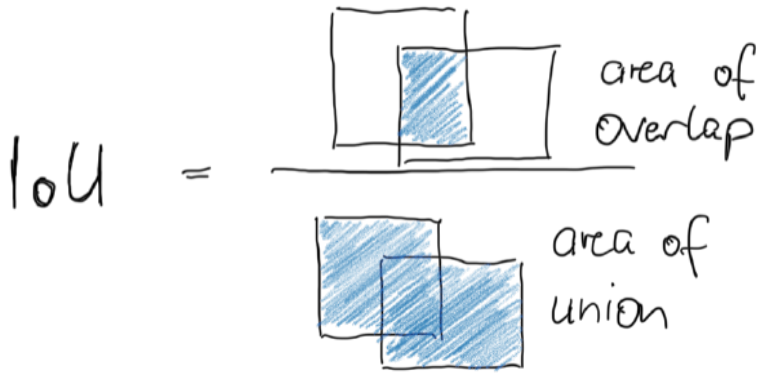
Source: https://blog.playment.io/semantic-segmentation/

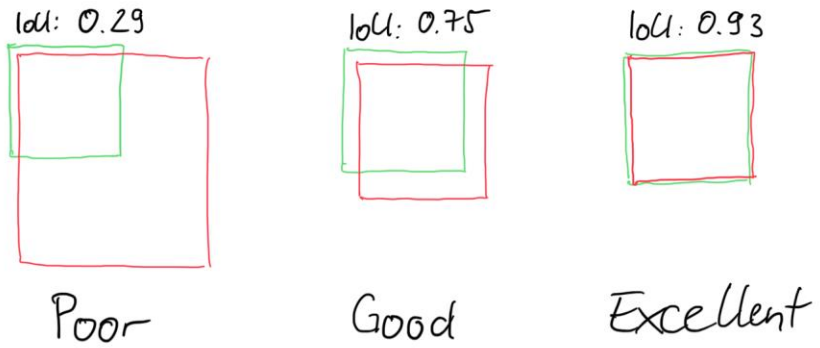
Entertainment / Gaming

http://www.virtualairguitar.com/expertise/

Intersection of Union (IoU)

- Evaluation metric used to measure the accuracy of an object detector
  1. The *ground-truth bounding boxes* (x,y, w,h).
  2. The *predicted bounding boxes* from our model.





Datasets COCO and PASCAL-Context

- 80 object categories
- train/val data has 330k images
- Captions and Segmentations

https://cocodataset.org/#home
https://cs.stanford.edu/~roozbeh/pascal-context

Microsoft COCO: Common Objects in Context

Tsung-Yi Lin Michael Maire Serge Belongie Lubomir Bourdev Ross Girshick
James Hays Pietro Perona Deva Ramanan C. Lawrence Zitnick Piotr Dollár

Abstract—We present a new dataset with the goal of advancing the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. This is achieved by gathering images of complex everyday scenes containing common objects in their natural context. Objects are labeled using per-instance segmentations to aid in precise object localization. Our dataset contains photos of 91 objects types that would be easily recognizable by a 4 year old. With a total of 2.5 million labeled instances in 328k images, the creation of our dataset drew upon extensive crowd worker involvement via novel user interfaces for category detection, instance spotting and instance segmentation. We present a detailed statistical analysis of the dataset in comparison to PASCAL, ImageNet, and SUN. Finally, we provide baseline performance analysis for bounding box and segmentation detection results using a Deformable Parts Model.

1 Introduction

One of the primary goals of computer vision is the understanding of visual scenes. Scene understanding involves numerous tasks including recognizing what objects are present, localizing the objects in 2D and 3D, determining the objects' and scene's attributes, characterizing relationships between objects and providing a semantic description of the scene. The current object classification and detection datasets [1], [2], [3], [4] help us explore the first challenges related to scene understanding. For instance the ImageNet dataset [1], which contains an unprecedented number of images, has recently enabled breakthroughs in both object classification and detection research [5], [6], [7]. The community has also created datasets containing object attributes [8], scene attributes [9], keypoints [10], and 3D scene information [11]. This leads us to the obvious question: what datasets will best continue our advance towards our ultimate goal of scene understanding?

We introduce a new large-scale dataset that addresses three core research problems in scene understanding: detecting non-iconic views (or non-canonical perspectives [12]) of objects, contextual reasoning between objects and the precise 2D localization of objects. For many categories of objects, there exists an iconic view. For example, when performing a web-based image search for the object category "bike," the top-ranked retrieved examples appear in profile, unobstructed near the center of a neatly composed photo. We posit that current recognition systems perform fairly well on iconic views, but struggle to recognize objects otherwise – in the background, partially occluded, amid clutter [13] – reflecting the composition of actual everyday scenes. We verify this experimentally; when evaluated on everyday scenes, models trained on our data perform better than those trained with prior datasets. A challenge is finding natural images that contain multiple objects. The identity of many objects can only be resolved using context, due to small size or ambiguous appearance in the image. To push research in contextual reasoning, images depicting scenes [3] rather than objects in isolation are necessary. Finally, we argue that detailed spatial understanding of object layout will be a core component of scene analysis. An object's spatial location can be defined coarsely using a bounding box [2] or with a precise pixel-level segmentation [14], [15], [16]. As we demonstrate, to measure either kind of localization performance it is essential for the dataset to have every instance of every object

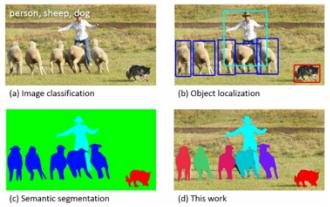
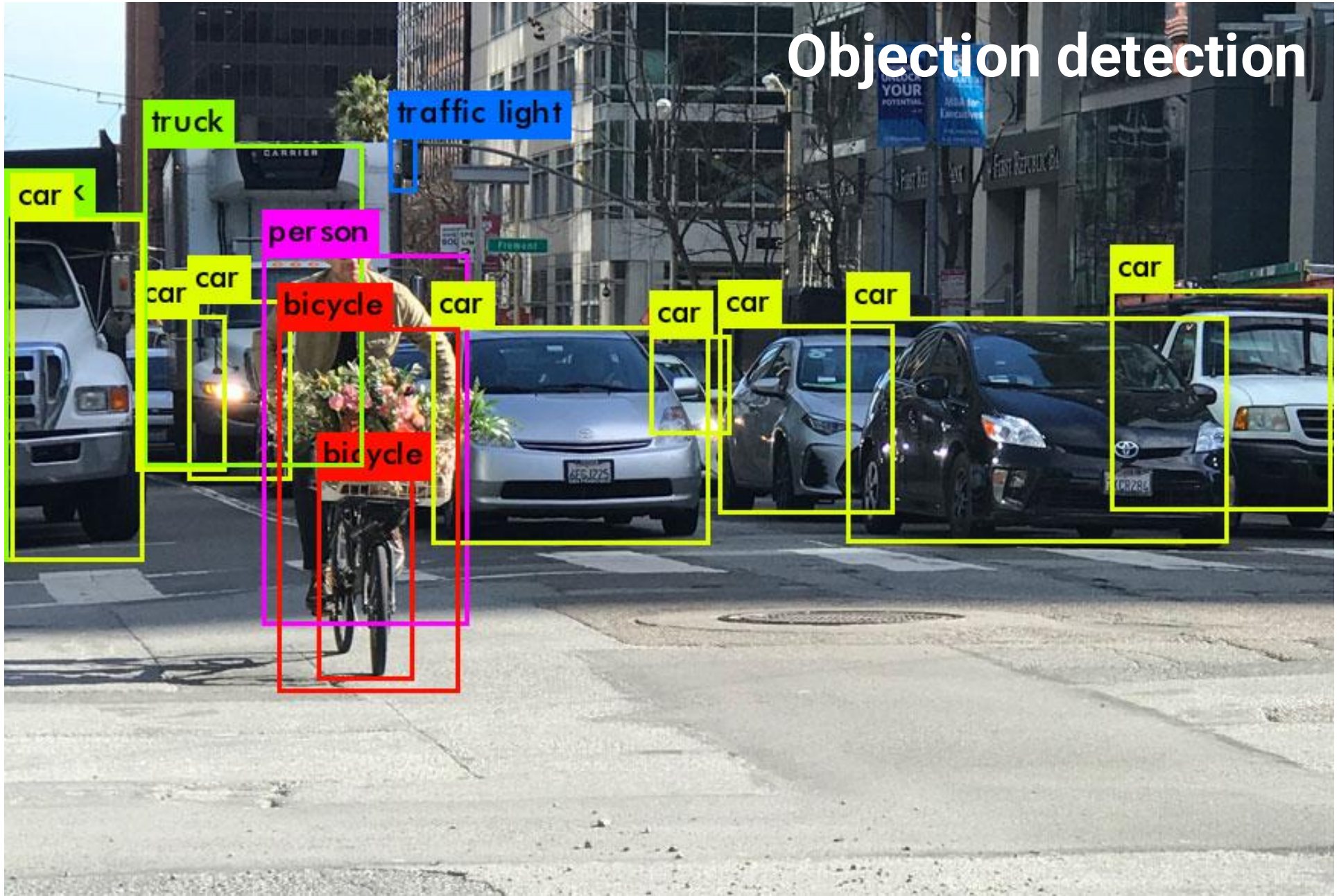


Fig. 1: While previous object recognition datasets have focused on (a) image classification, (b) object bounding box localization or (c) semantic pixel-level segmentation, we focus on (d) segmenting individual object instances. We introduce a large, richly-annotated dataset comprised of images depicting complex everyday scenes of common objects in their natural context.

• *T.Y. Lin and S. Belongie are with Cornell NYC Tech and the Cornell Computer Science Department.*
• *M. Maire is with the Toyota Technological Institute at Chicago.*
• *L. Bourdev and P. Dollár are with Facebook AI Research. The majority of this work was performed while P. Dollár was with Microsoft Research.*
• *R. Girshick and C. L. Zitnick are with Microsoft Research, Redmond.*
• *J. Hays is with Brown University.*
• *P. Perona is with the California Institute of Technology.*
• *D. Ramanan is with the University of California at Irvine.*

Objection detection



Template Matching

https://docs.opencv.org/master/d4/dc6/tutorial_py_template_matching.html

Template

Image

- Method for searching and finding location of a template image in a larger image
- Slides template image over input image (as in 2D convolution) and compares template and patch of input image under the template image
- Methods return grayscale image, where each pixel denotes how much the neighbourhood of that pixel matches with template
- If input image is of size (W*x*H) and template image is of size (w*x*h), output image will have a size of (W-w+1, H-h+1)
- Min/Max values in that image determine the best match

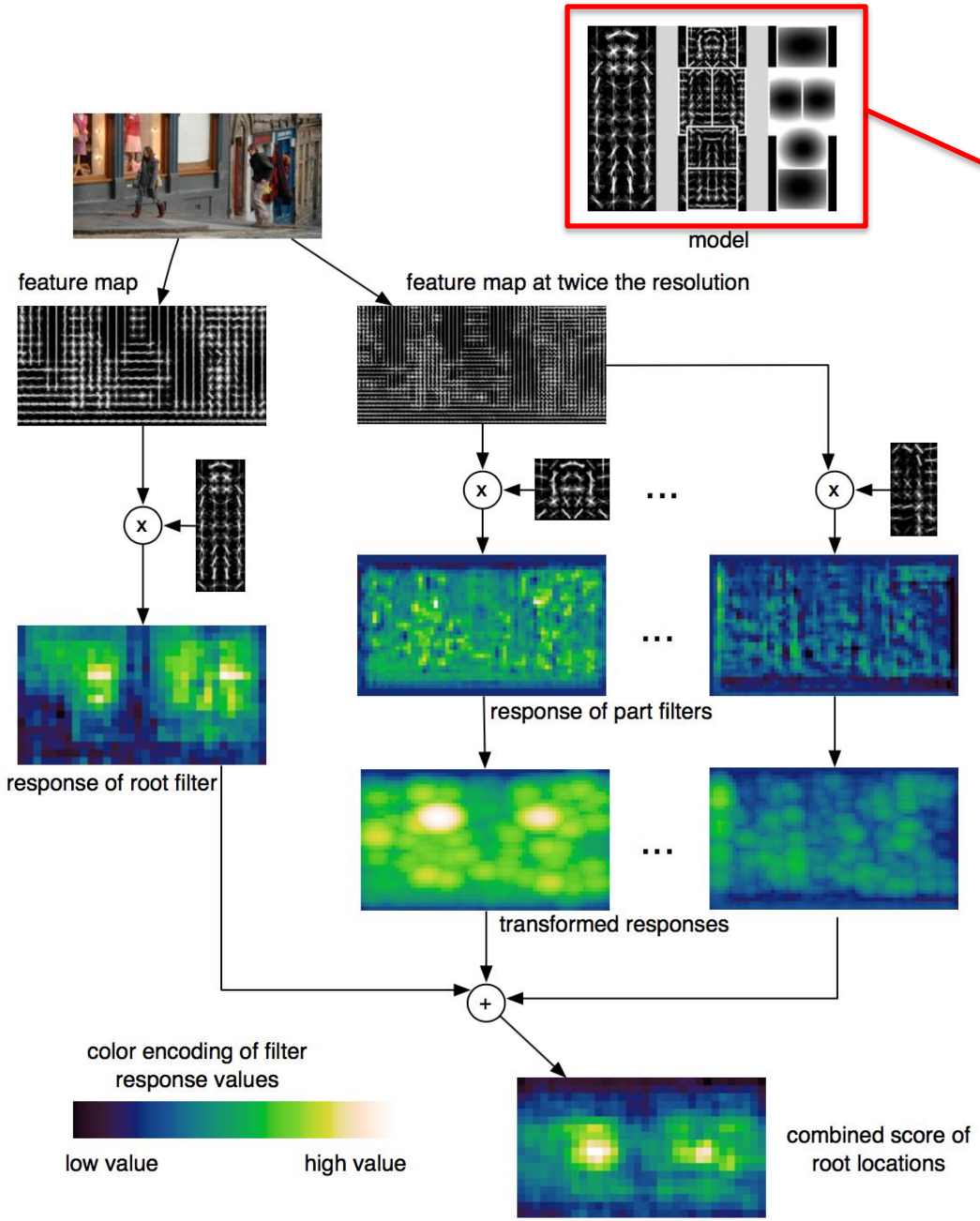
Matching result

Deformable Model Parts (DMP)

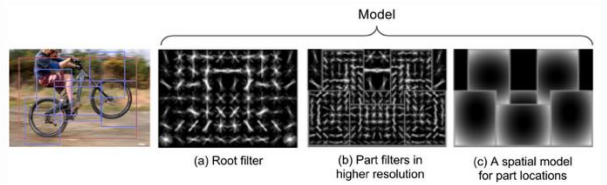
Object Detection with Discriminatively Trained Part Based Models

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester and Deva Ramanan

2010



This is what we learn using **HOG** and a **Support Vector Machine**



Overview – State of the art methods

Model	Goal	Resources	
R-CNN	Object recognition	[paper][code]	https://arxiv.org/abs/1311.2524 https://github.com/rbgirshick/rcnn
Fast R-CNN	Object recognition	[paper][code]	https://arxiv.org/abs/1504.08083 https://github.com/rbgirshick/fast-rcnn
Faster R-CNN	Object recognition	[paper][code]	https://arxiv.org/abs/1506.01497 https://github.com/rbgirshick/py-faster-rcnn
Mask R-CNN	Instance segmentation	[paper][code]	https://arxiv.org/abs/1703.06870 https://github.com/CharlesShang/FastMaskRCNN
YOLO	Fast object recognition	[paper][code]	https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf https://pjreddie.com/darknet/yolo/
SSD	Fast Object recognition.	[paper][code]	https://github.com/weiliu89/caffe/tree/ssd?tab=readme-ov-file

Two-Stage vs. One-Stage Detectors

Feature	Two-Stage Detectors	One-Stage Detectors
Examples	Faster R-CNN, **Mask R-CNN**	**YOLO**, RetinaNet, **MobileNet +** SSD
Pipeline	**Region Proposal → Classification**	**Direct prediction from input image**
Accuracy	**Higher** (especially on small objects)	Slightly **lower**, but improving
Speed	**Slower**	Much faster (**real-time capable**)
Complexity	More complex architecture	Simpler, unified model
Use Case	High-accuracy tasks, offline analysis	Real-time applications, mobile deployment
Training	Harder to train and tune	Easier end-to-end training

Object recognition
YOLO
(You only look once)

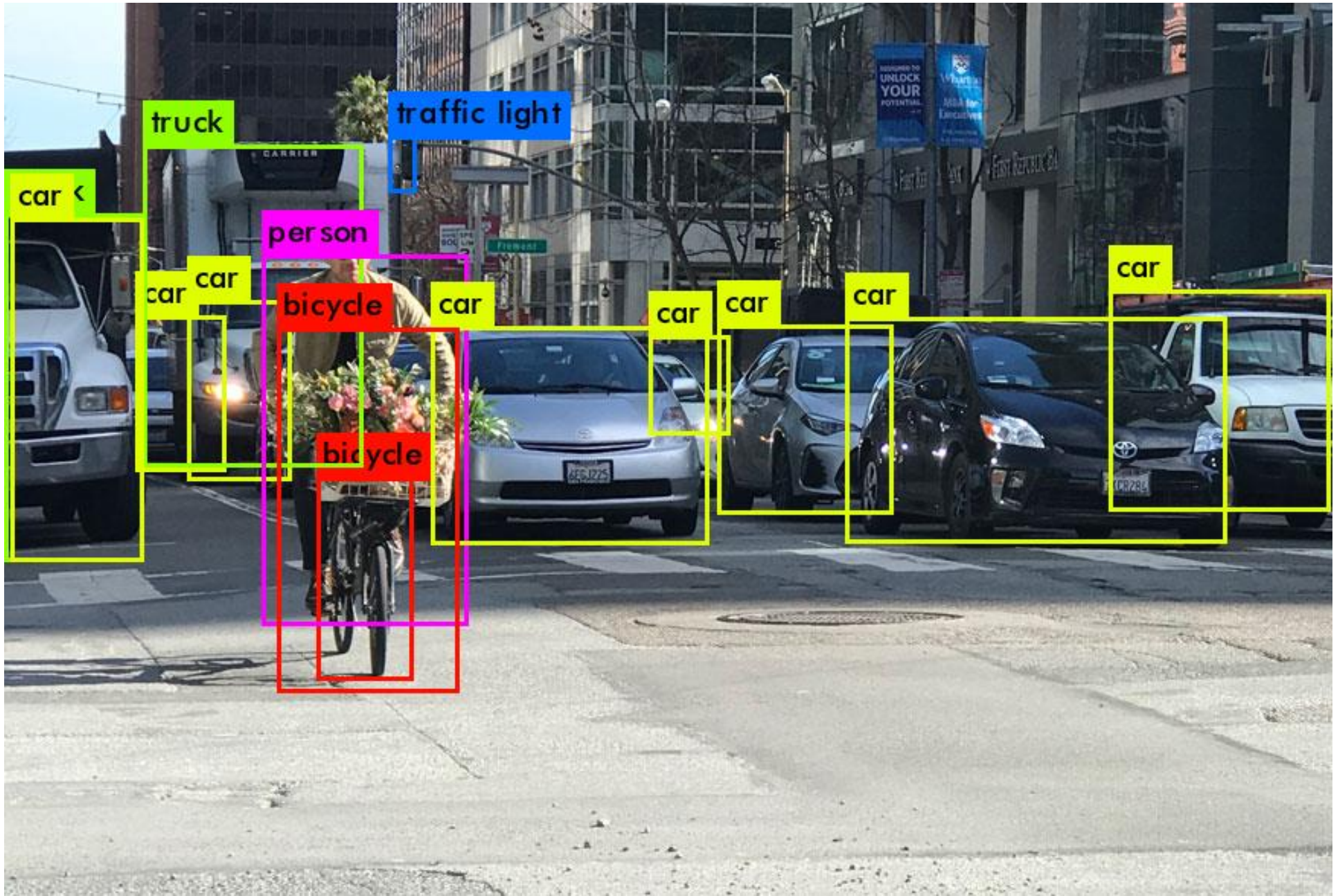
https://www.youtube.com/watch?t=&v=VOC3huqHrss

Pros: Very fast.
Cons: Accuracy tradeoff; not good at recognizing irregularly shaped objects or a group of small objects (i.e. a flock of birds?)

YOLO v1

Images taken from: https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

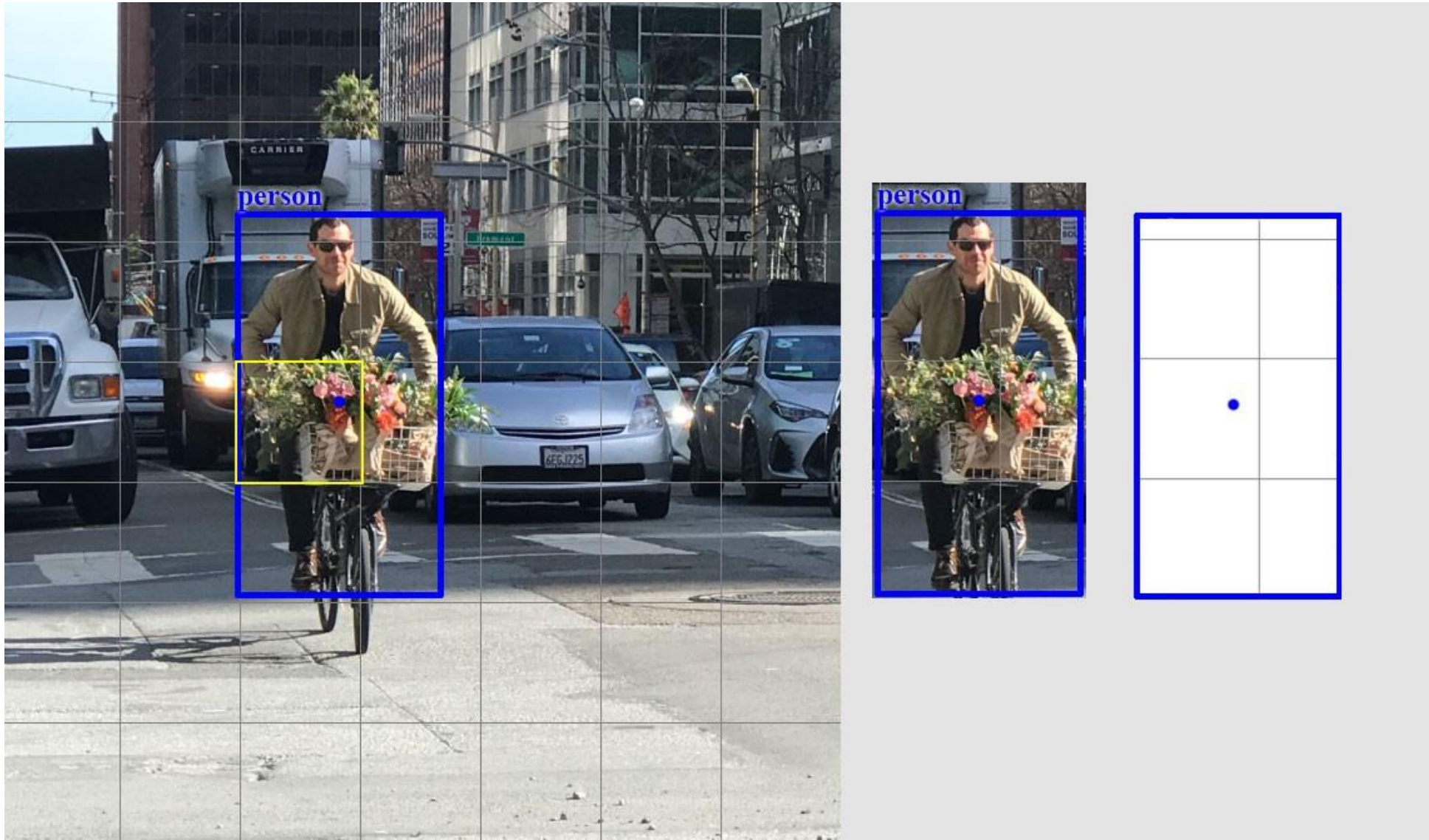
YOLO v1


truck
traffic light
car
person
car
car
bicycle
car
car
car
car
car
bicycle

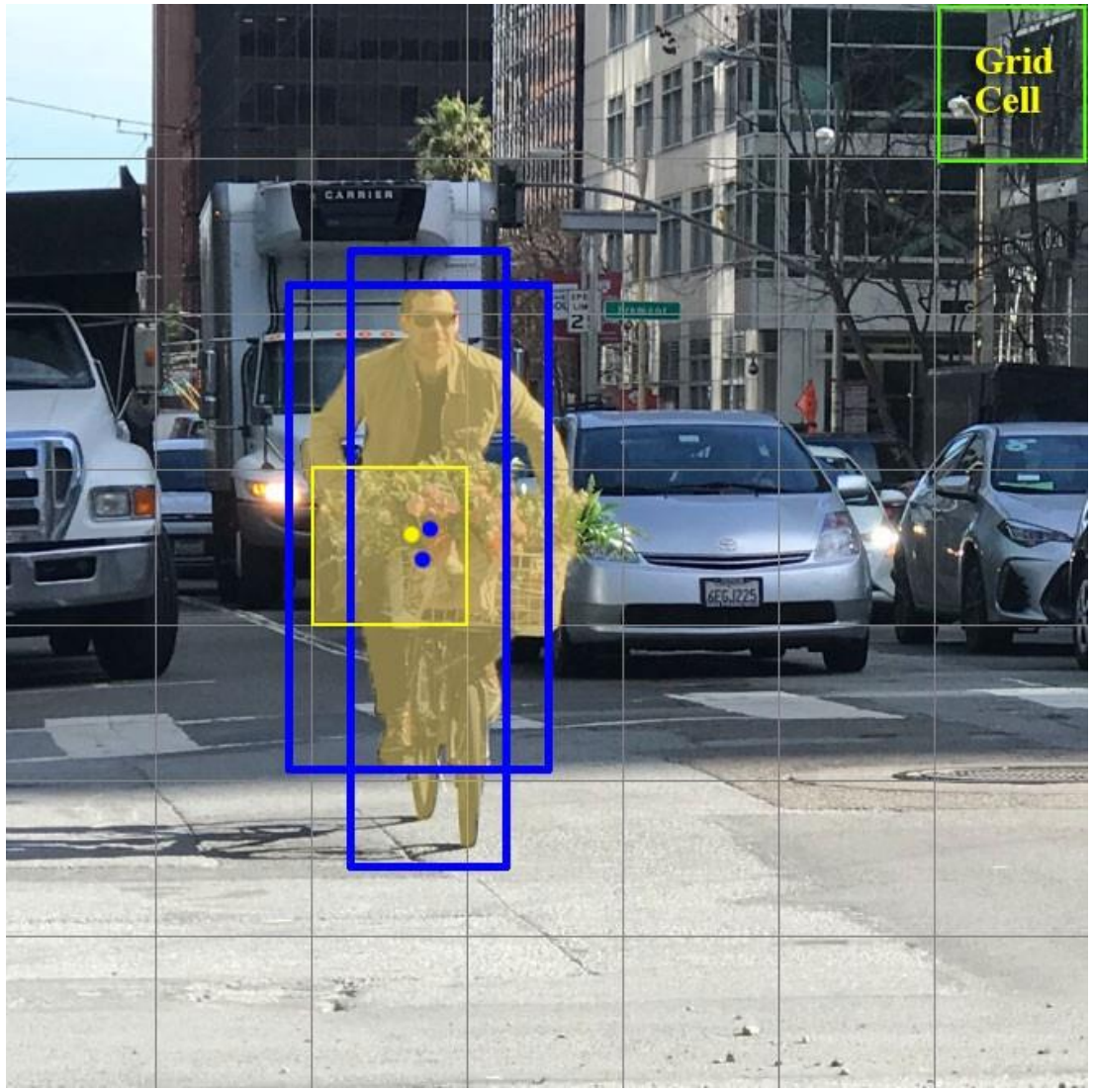
YOLO v1

1. Pre-train a CNN network on image classification tasks.
2. Split an image into S x S cells.
    1. Each cell is responsible for identifying the object (if any) with its center located in this cell.
    2. Each cell predicts
        1. location of B bounding boxes
        2. confidence score
        3. and a probability of object class conditioned on the existence of an object in the bounding box.

YOLO v1



YOLO v1



Setup:

- 7x7 grid cells (SxS)
- boundary boxes (B) with position x, y, width, height and confidence score
- 20 classes (C)

For each grid cell YOLO predicts:

- 2 boundary boxes
- one object for all boundary boxes
- 1 of 20 classes

YOLO v1

YOLO prediction tensors have a shape of
(S, S, B×5 + C) = (7, 7, 2×5 + 20) = (7, 7, 30)

YOLO

(x, y, w, h) + confidence score (objectness+box)

(x, y, w, h) + confidence score (objectness+box)

car: 0.05
person: 0.10
...
truck : 0.02

...

cell(1, 1)

...

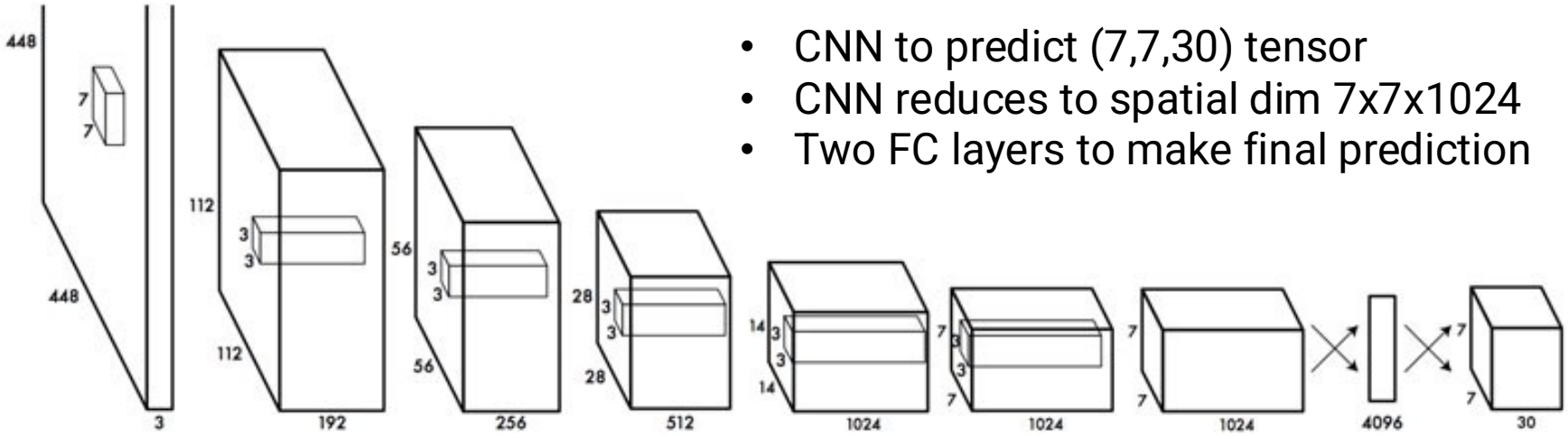
...

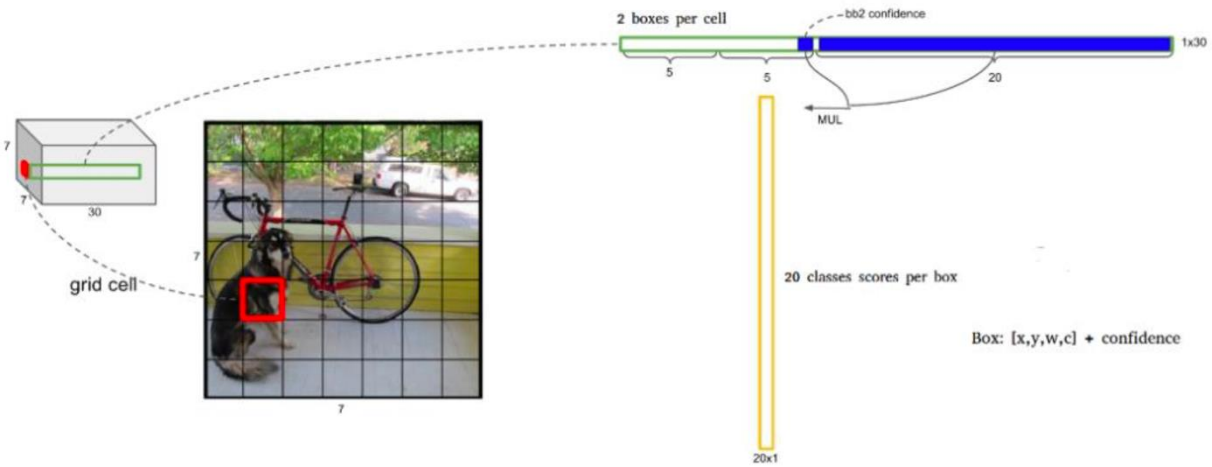
cell(S, S)

YOLO v1

YOLO prediction tensors have a shape of
(S, S, B×5 + C) = (7, 7, 2×5 + 20) = (7, 7, 30)

- CNN to predict (7,7,30) tensor
- CNN reduces to spatial dim 7x7x1024
- Two FC layers to make final prediction





https://www.slideshare.net/TaegyunJeon1/pr12-you-only-look-once-yolo- unified-realtime-object-detection/89

YOLO v1 Loss function

- Loss function consists of three terms:
  - Classification loss

$$\sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2$$

+

  - Localization loss

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2\right]$$

+

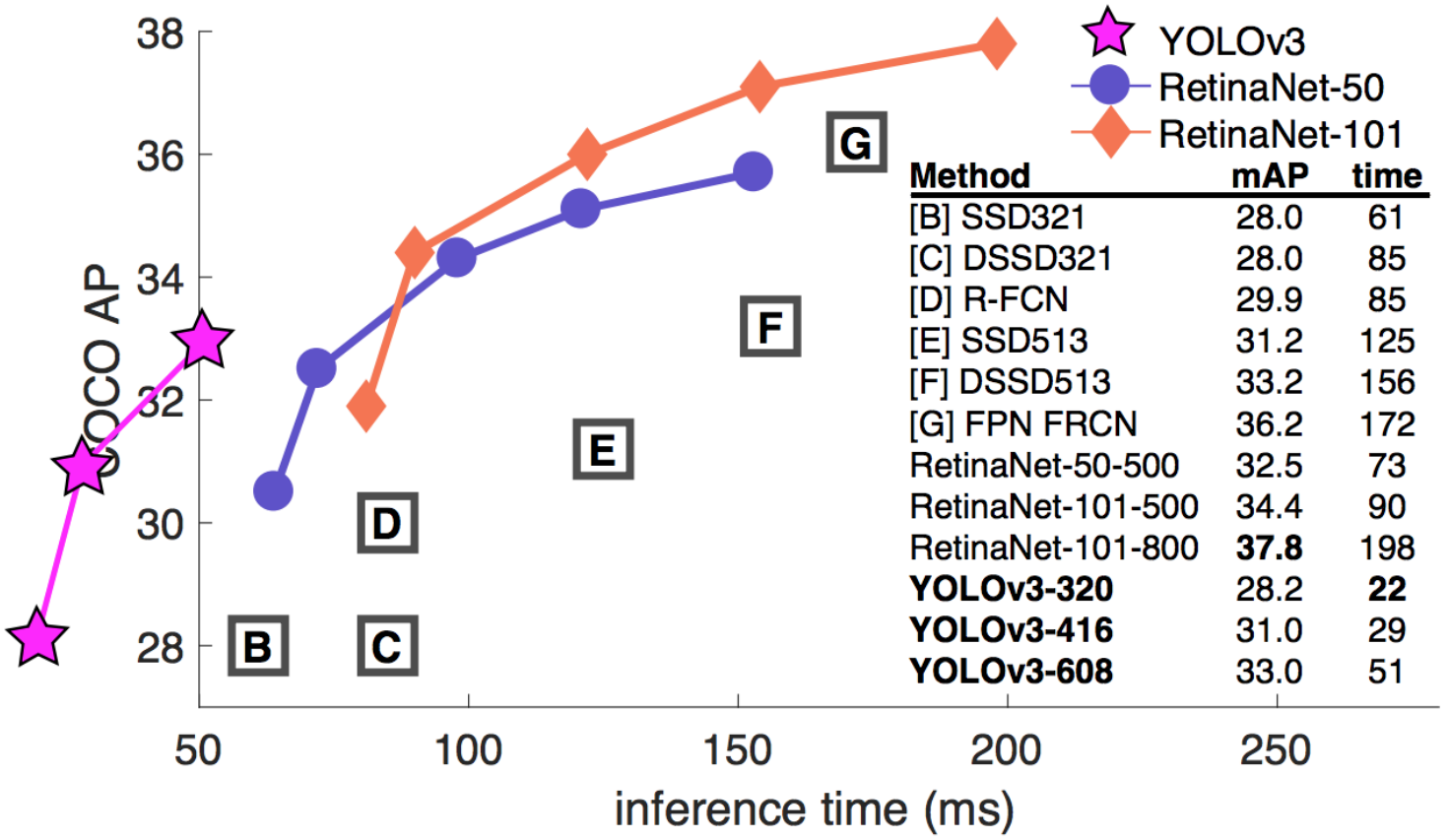
  - Confidence loss

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2$$

$$\lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2$$

YOLO v2 --> YOLO v3

https://www.youtube.com/watch?v=MPU2HistivI



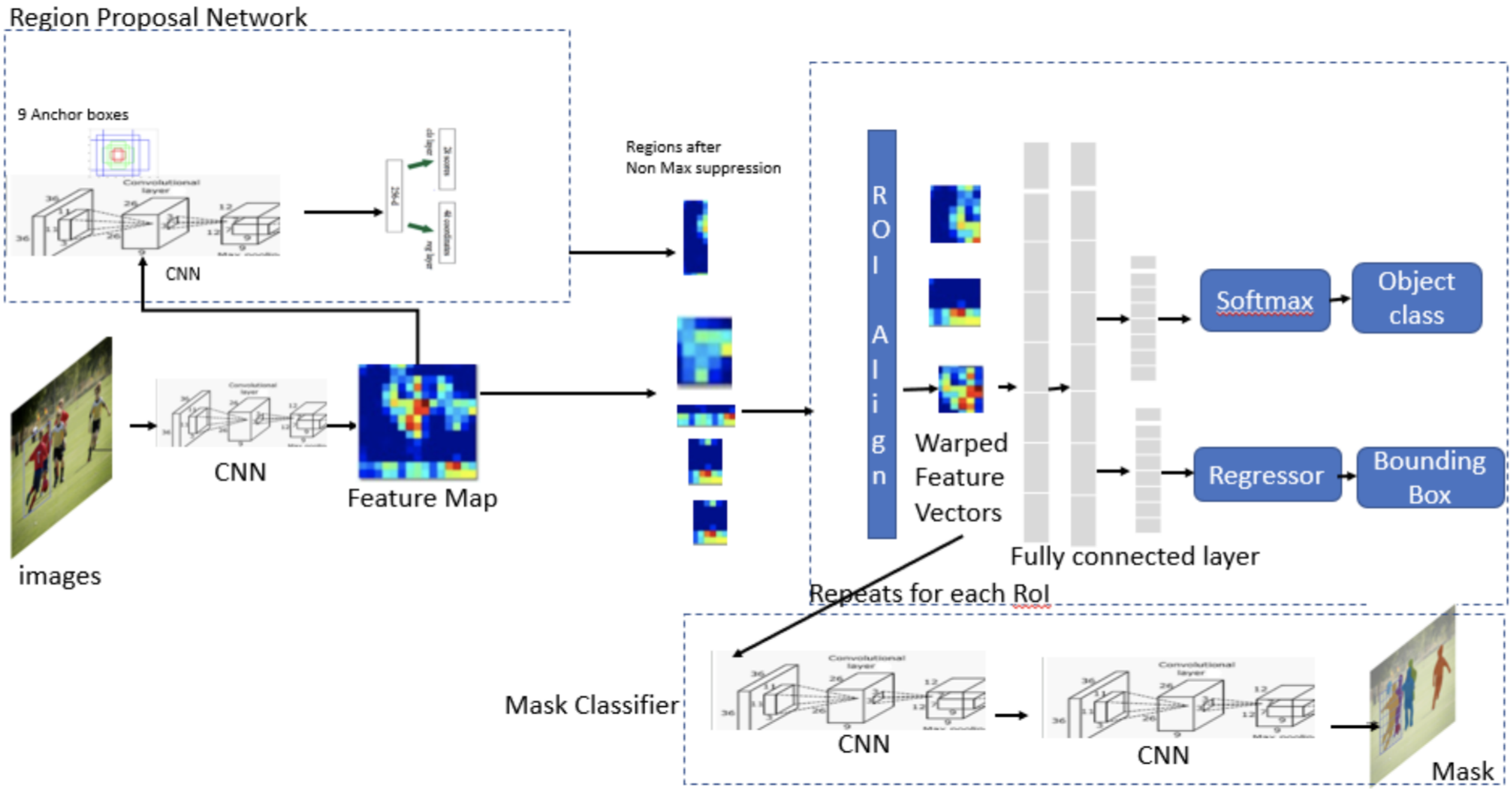
Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

Instance Segmentation

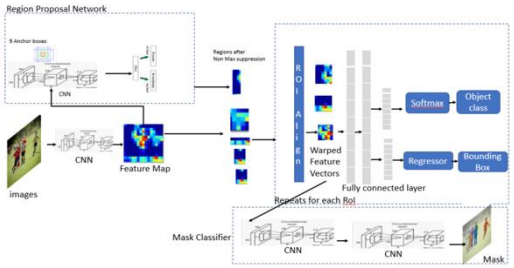
Mask R-CNN


bus.99
person1.00
backpack.99
backpack.93
person.99
sheep.99
sheep1.00
person.99
person1.00
traffic light.96
traffic light.95
traffic light.92
traffic light.84
truck1.00
car1.00
car.99
BUS
chair.99
wine glass1.00
bowl.81
fork.95
knife.83

Mask R-CNN


Region Proposal Network
9 Anchor boxes
CNN
Regions after
Non Max suppression
images
CNN
Feature Map
ROI Align
Warped
Feature
Vectors
Softmax
Object
class
Regressor
Bounding
Box
Fully connected layer
Repeats for each RoI
Mask Classifier
CNN
CNN
Mask

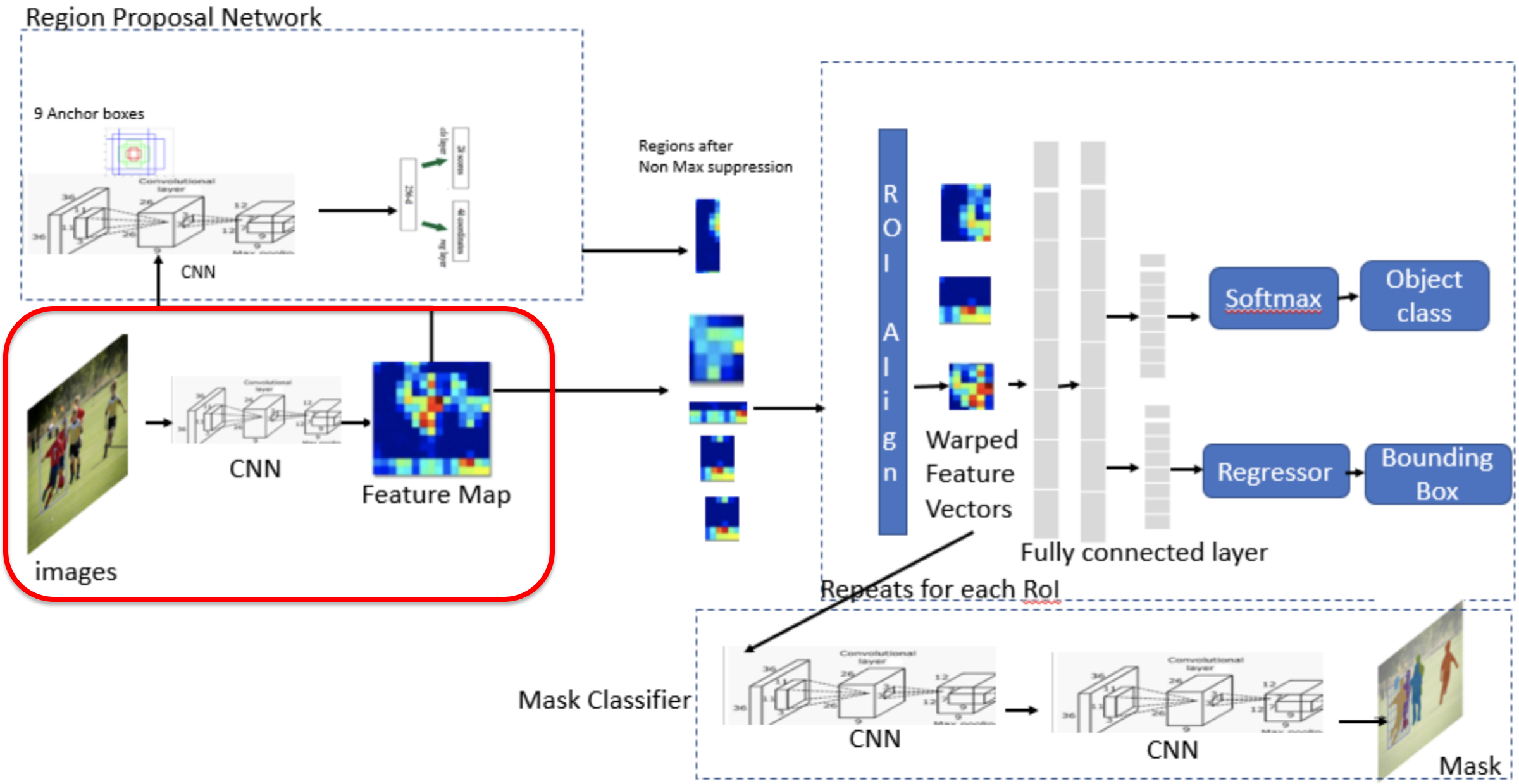
Mask R-CNN



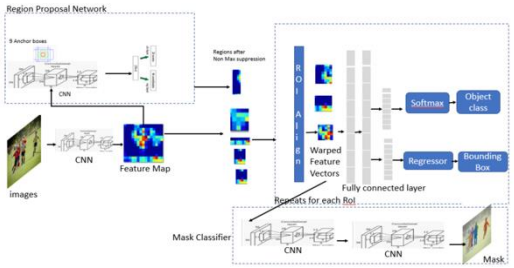
- Model is divided into two parts:
  - **Region proposal network (RPN) to proposes candidate object bounding boxes aka Object detection** (bounding box + class)
  - **Binary mask classifier to generate mask for every class aka Instance segmentation** (mask per object)

1. Image is run through the CNN to generate the feature maps.

Mask R-CNN


Region Proposal Network
9 Anchor boxes
CNN
Regions after
Non Max suppression
images
CNN
Feature Map
ROI Align
Warped
Feature
Vectors
Softmax
Object
class
Regressor
Bounding
Box
Fully connected layer
Repeats for each RoI
Mask Classifier
CNN
CNN
Mask

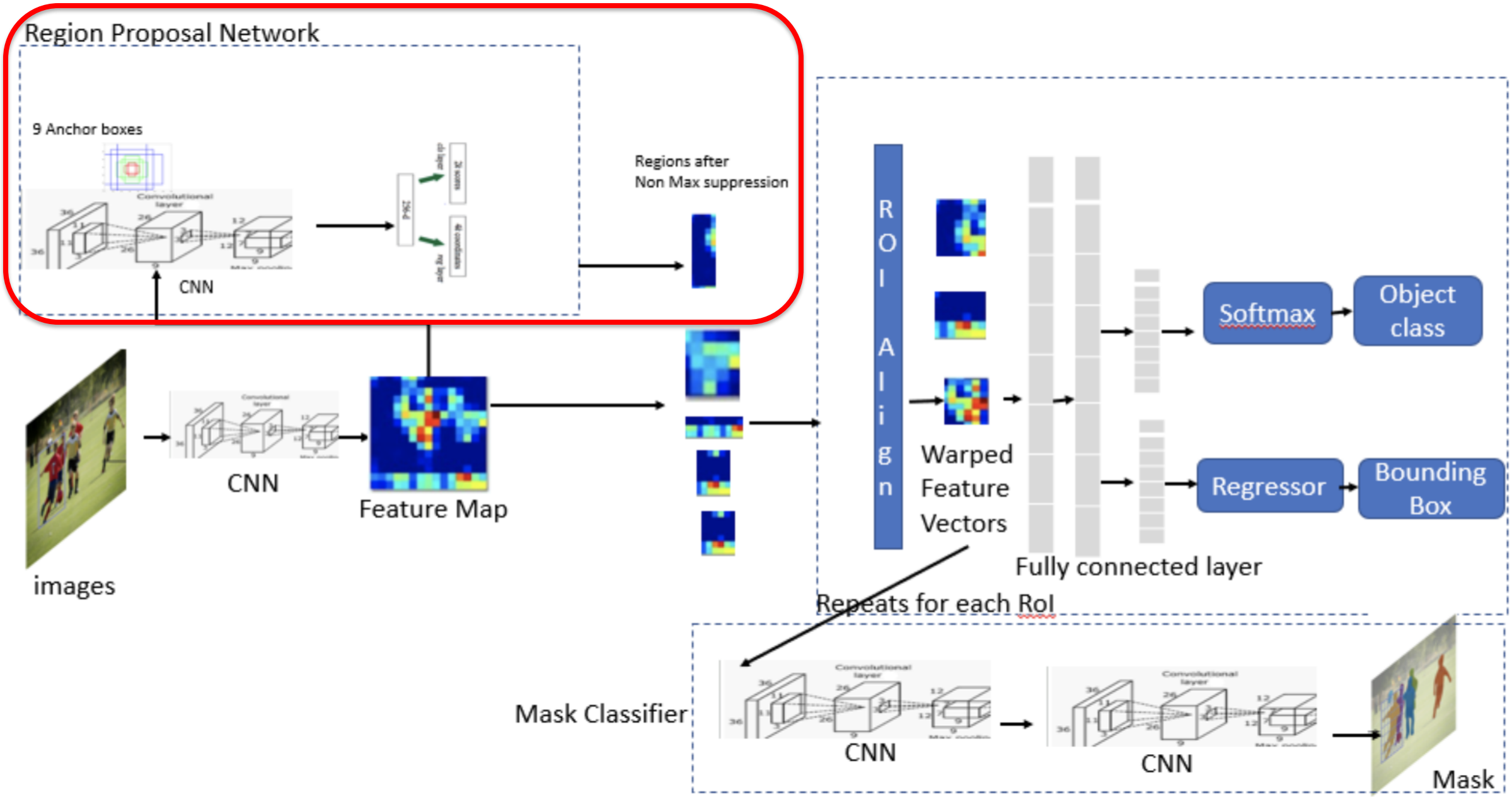
Mask R-CNN



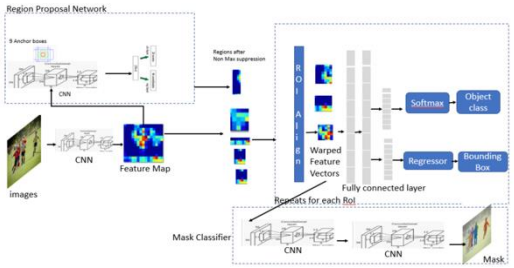
- Model is divided into two parts:
  - **Region proposal network (RPN) to proposes candidate object bounding boxes aka Object detection** (bounding box + class)
  - **Binary mask classifier to generate mask for every class aka Instance segmentation** (mask per object)

1. Image is run through the CNN to generate the feature maps.
2. Region Proposal Network(RPN) uses a CNN to generate multiple Region of Interest anchors boxes over the image, e.g., 2000 regions. The classifier returns object/no-object scores. Non-Max suppression is applied to Anchors with high object score

Mask R-CNN


Region Proposal Network
9 Anchor boxes
Convolutional layer
CNN
Regions after
Non Max suppression
images
CNN
Feature Map
ROI Align
Warped
Feature
Vectors
Softmax
Object
class
Regressor
Bounding
Box
Fully connected layer
Repeats for each RoI
Mask Classifier
CNN
CNN
Mask

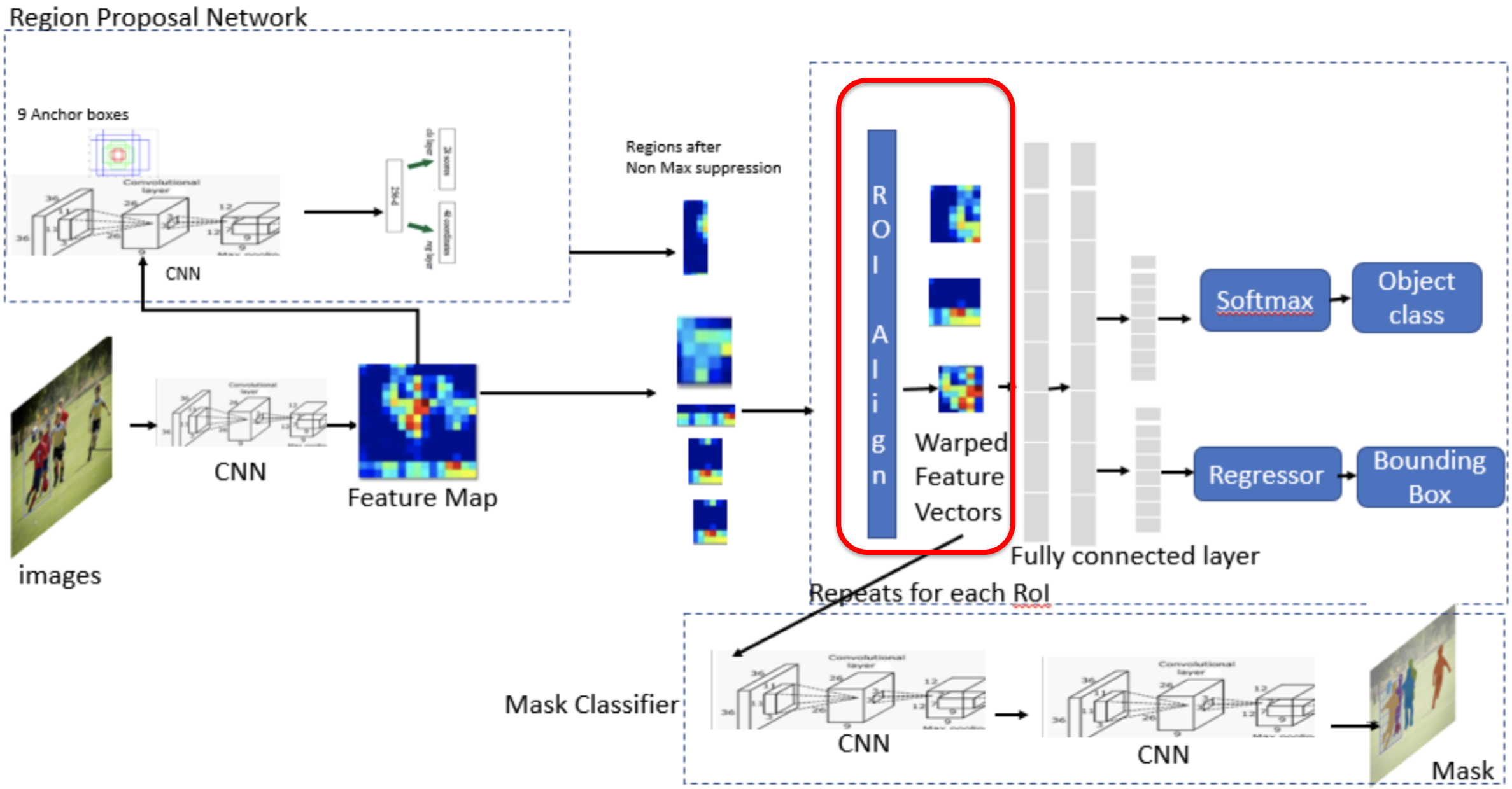
Mask R-CNN



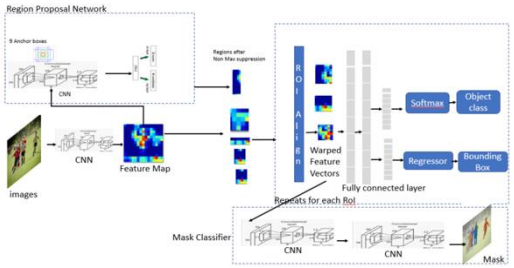
- Model is divided into two parts:
  - **Region proposal network (RPN) to proposes candidate object bounding boxes aka Object detection** (bounding box + class)
  - **Binary mask classifier to generate mask for every class aka Instance segmentation** (mask per object)

1. Image is run through the CNN to generate the feature maps.
2. Region Proposal Network(RPN) uses a CNN to generate multiple Region of Interest anchors boxes over the image. The classifier returns object/no-object scores. Non-Max suppression is applied to Anchors with high object score
3. The RoI network outputs multiple bounding boxes rather than a single definite one and warp them into a fixed dimension

Mask R-CNN


Region Proposal Network
9 Anchor boxes
CNN
Regions after
Non Max suppression
images
CNN
Feature Map
ROI Align
Warped
Feature
Vectors
Softmax
Object
class
Regressor
Bounding
Box
Fully connected layer
Repeats for each RoI
Mask Classifier
CNN
CNN
Mask

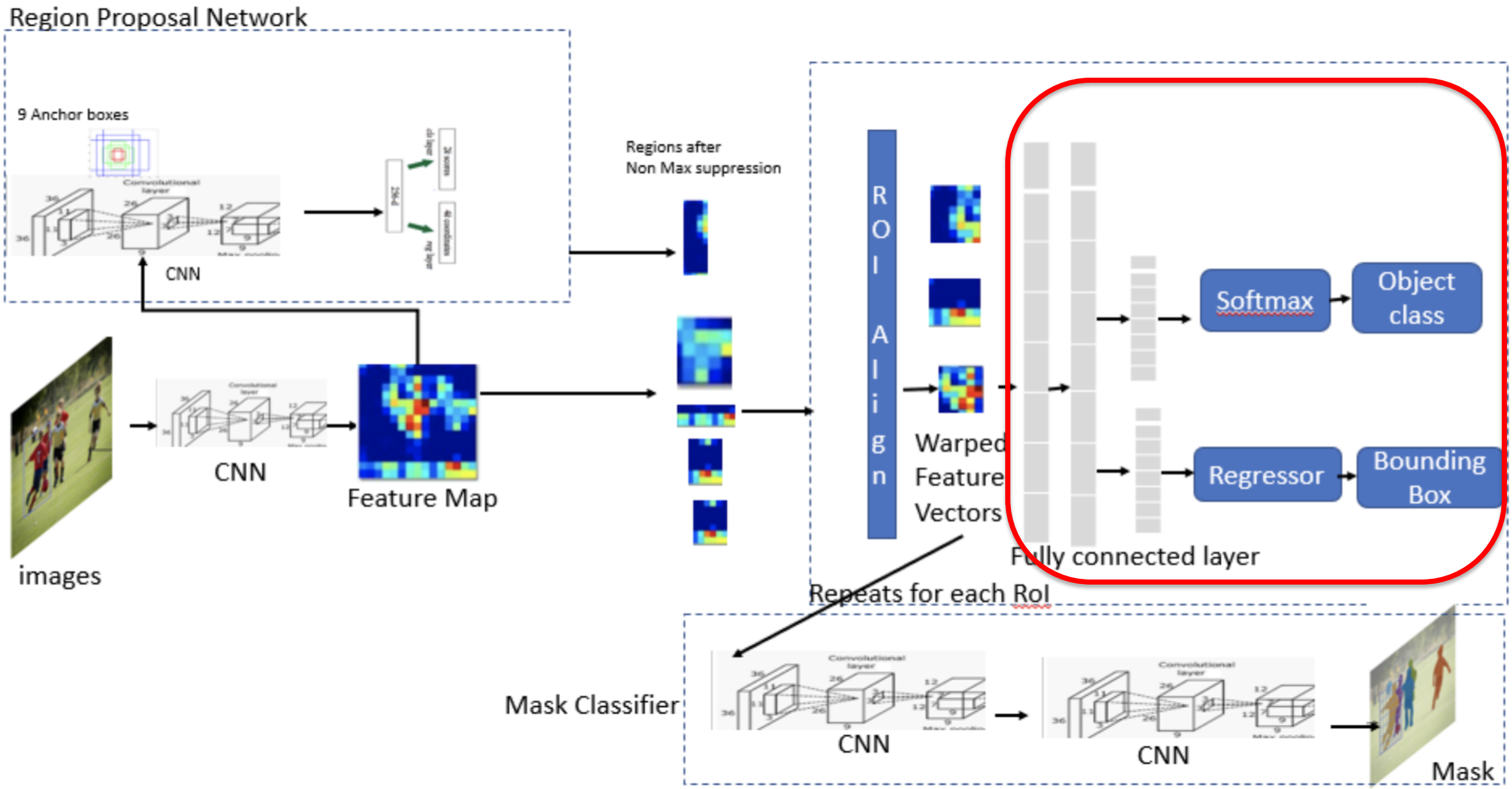
Mask R-CNN



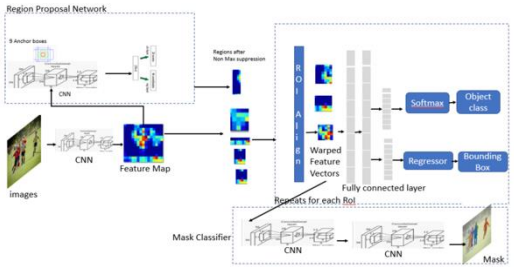
- Model is divided into two parts:
  - **Region proposal network (RPN) to proposes candidate object bounding boxes aka Object detection** (bounding box + class)
  - **Binary mask classifier to generate mask for every class aka Instance segmentation** (mask per object)

1. Image is run through the CNN to generate the feature maps.
2. Region Proposal Network(RPN) uses a CNN to generate multiple Region of Interest anchors boxes over the image. The classifier returns object/no-object scores. Non-Max suppression is applied to Anchors with high object score
3. The RoI network outputs multiple bounding boxes rather than a single definite one and warp them into a fixed dimension
4. Warped features are then fed into fully connected layers to make classification using softmax and boundary box prediction

Mask R-CNN


Region Proposal Network
9 Anchor boxes
CNN
Regions after
Non Max suppression
images
CNN
Feature Map
ROI Align
Warped
Feature
Vectors
Softmax
Object
class
Regressor
Bounding
Box
Fully connected layer
Repeats for each RoI
Mask Classifier
CNN
CNN
Mask

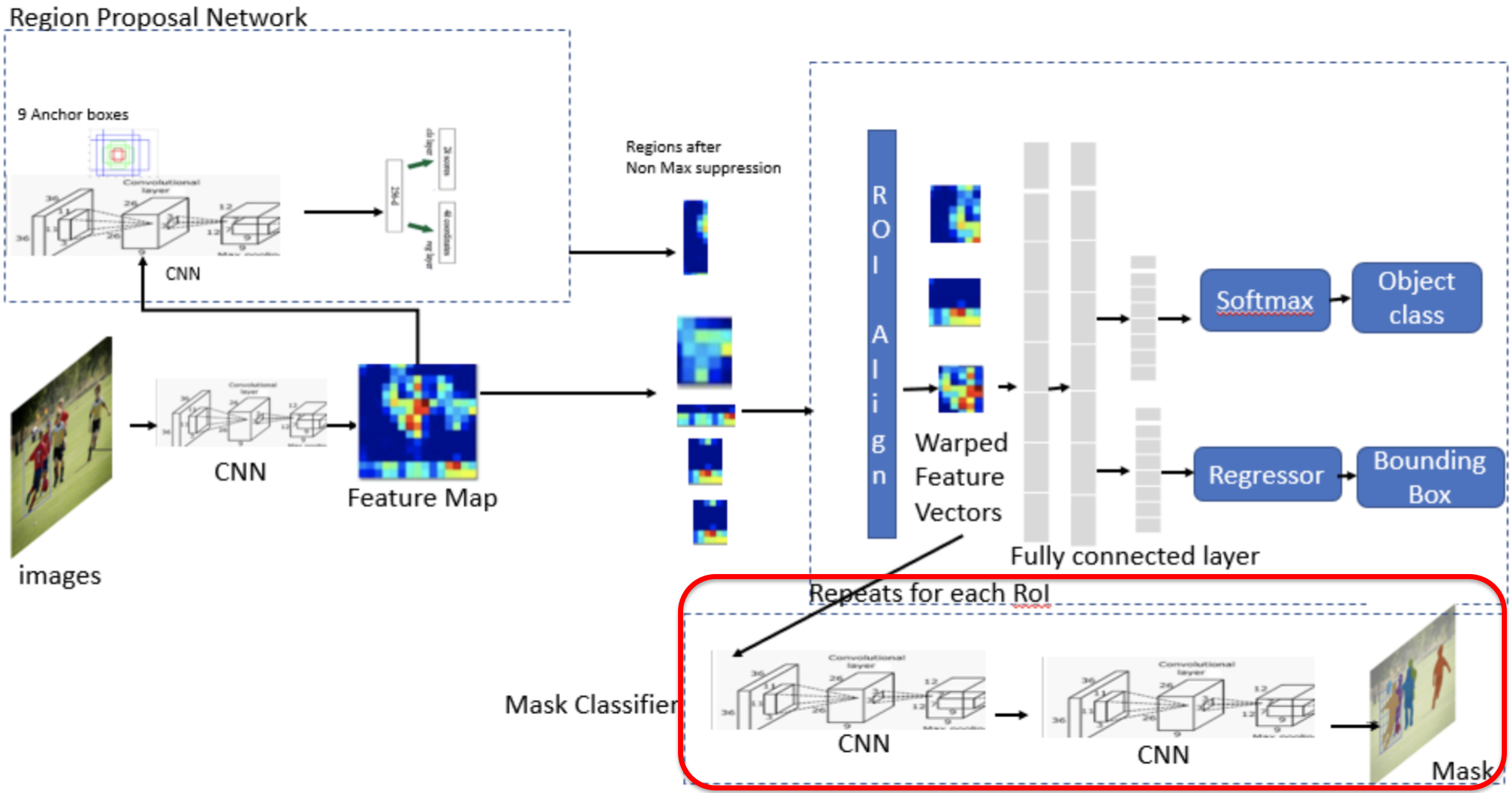
Mask R-CNN



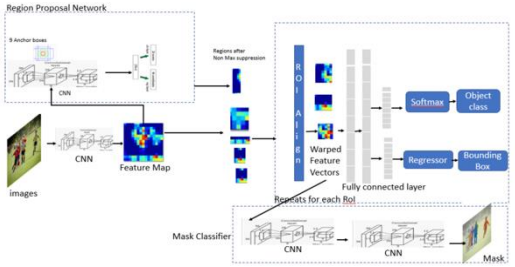
- Model is divided into two parts:
  - **Region proposal network (RPN) to proposes candidate object bounding boxes aka Object detection** (bounding box + class)
  - **Binary mask classifier to generate mask for every class aka Instance segmentation** (mask per object)

1. Image is run through the CNN to generate the feature maps.
2. Region Proposal Network(RPN) uses a CNN to generate multiple Region of Interest anchors boxes over the image. The classifier returns object/no-object scores. Non-Max suppression is applied to Anchors with high object score
3. The RoI network outputs multiple bounding boxes rather than a single definite one and warp them into a fixed dimension
4. Warped features are then fed into fully connected layers to make classification using softmax and boundary box prediction
5. Warped features are also fed into mask classifier, which consists of two CNN's to output a binary mask for each RoI

Mask R-CNN


Region Proposal Network
9 Anchor boxes
CNN
Regions after
Non Max suppression
images
CNN
Feature Map
ROI Align
Warped
Feature
Vectors
Softmax
Object
class
Regressor
Bounding
Box
Fully connected layer
Repeats for each RoI
Mask Classifier
CNN
CNN
Mask

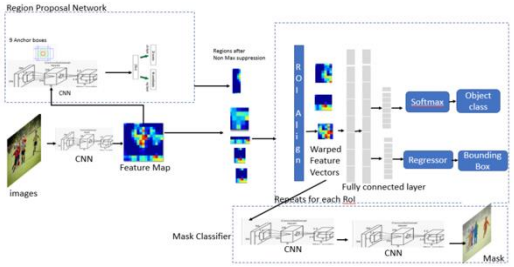
Mask R-CNN



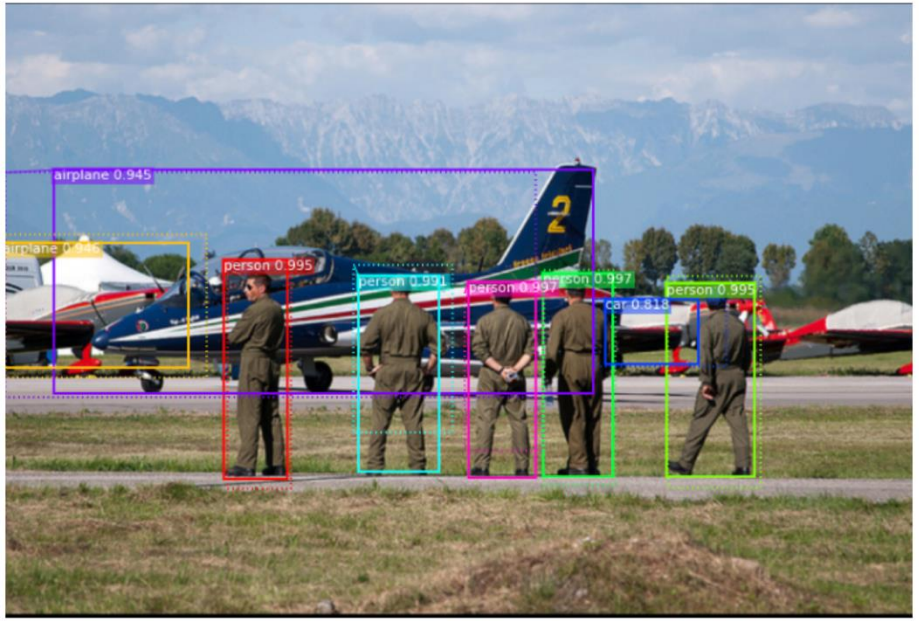
- **Anchor *boxes*** are a set of predefined bounding boxes of a certain height and width
- are defined to capture the scale and aspect ratio of specific object classes you want to detect
- To predict multiple objects or multiple instances of objects in an image, Mask R-CNN makes thousands of predictions

- **Final object detection** is done by removing anchor boxes that belong to the background class and the remaining ones are filtered by their confidence score (IoU > 0.5, non-max suppression)

Non-maximum suppression



- Non-Max Suppression will remove all bounding boxes where IoU is less than or equal to 0.7
- Pick the bounding box with the highest value for IoU and suppress the other bounding boxes for identifying the same object



https://github.com/matterport/Mask_RCNN

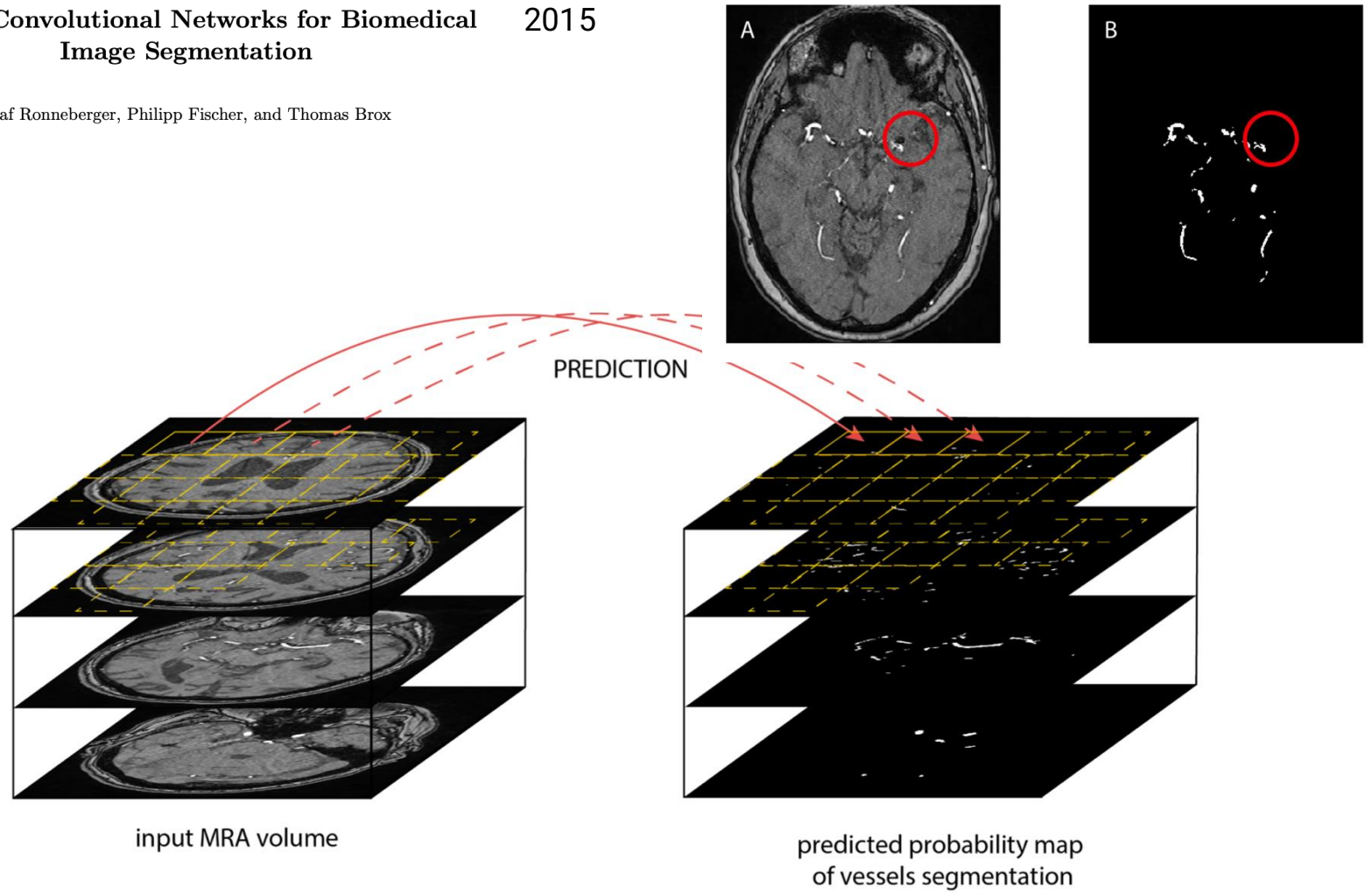
Mask R-CNN

Semantic Segmentation

U-Net for image segmentation

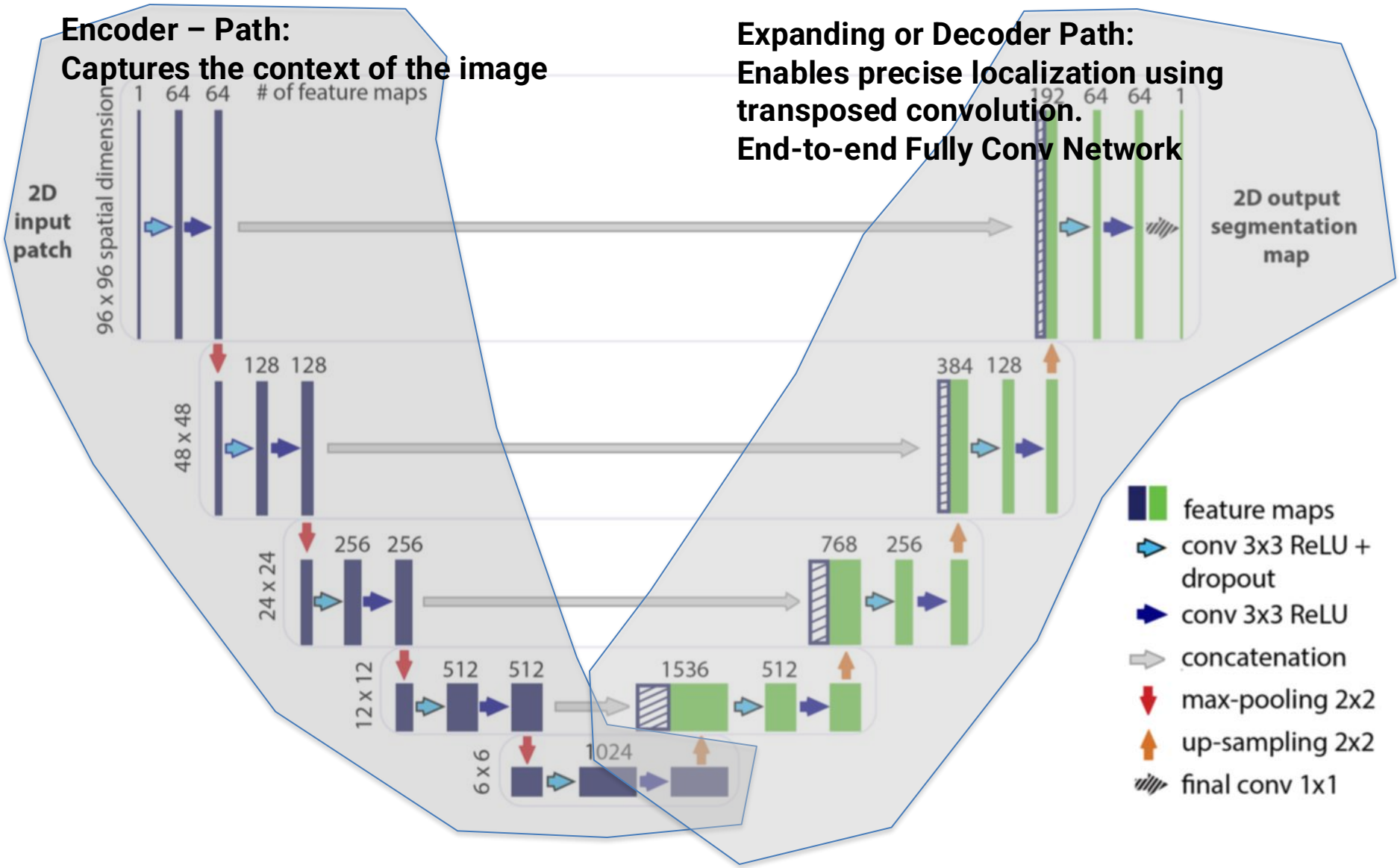
U-Net: Convolutional Networks for Biomedical Image Segmentation 2015

Olaf Ronneberger, Philipp Fischer, and Thomas Brox



U-net Deep Learning Framework for High Performance Vessel Segmentation in Patients with Cerebrovascular Disease Frontiers in Neuroscience. 2019. Livne, Michelle and Rieger, Jana and Aydin, Orhun and Taha, Abdel and Akay, Ela and Kossen, Tabea and Sobesky, Jan and Kelleher, John and Hildebrand, Kristian and Frey, Dietmar and Madai, Vince

U-Net for image segmentation

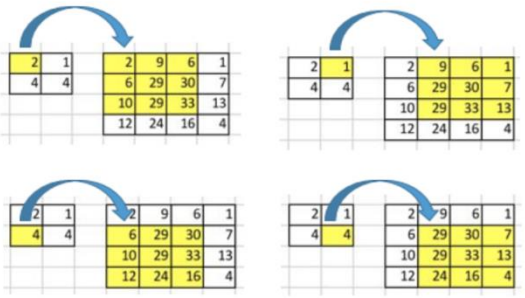


Up Sampling

Upscaling

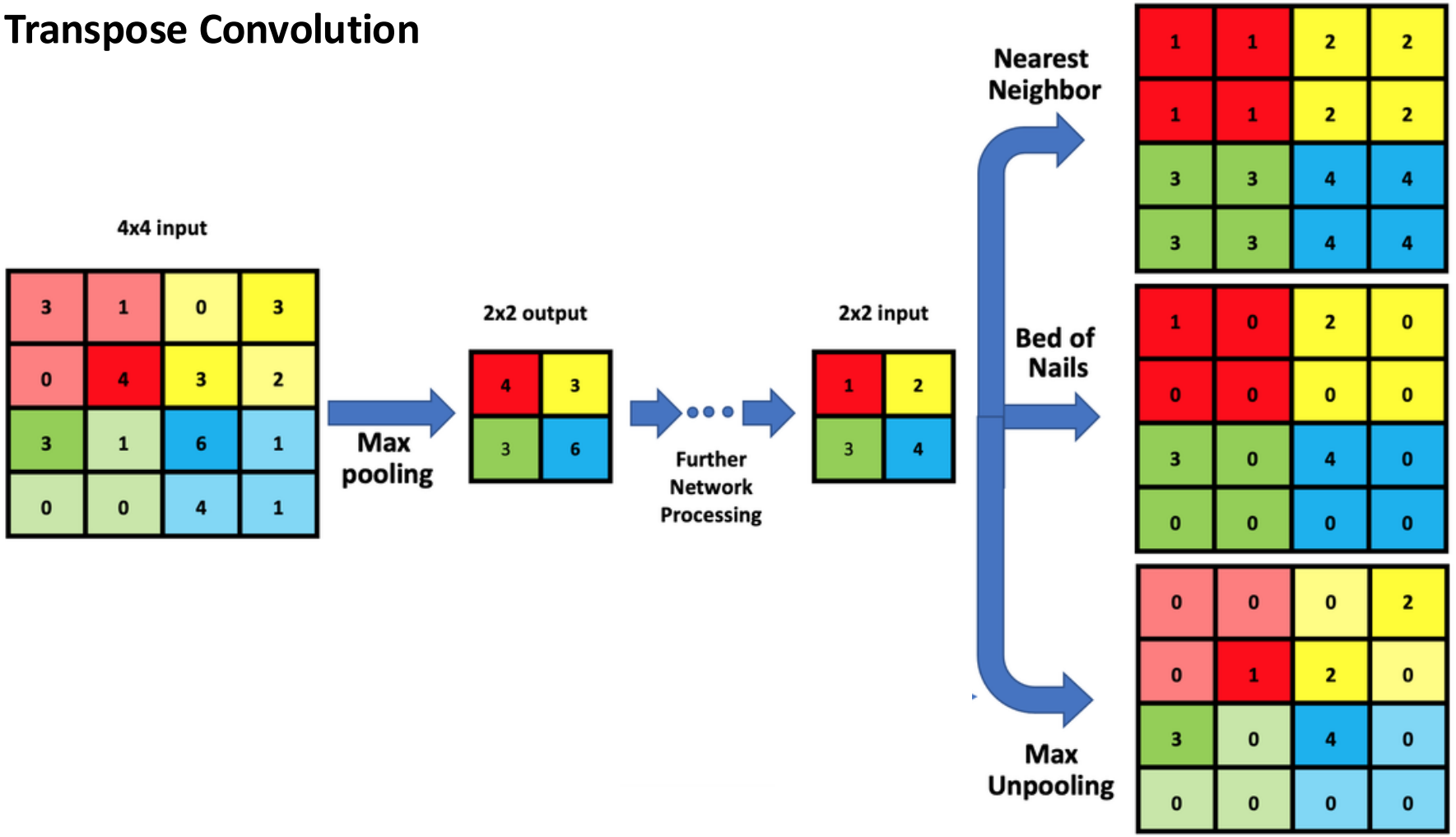
Without global interpolation method
e.g. bi-linear or cubic interpolatin

Transposed Convolution



Upscaling

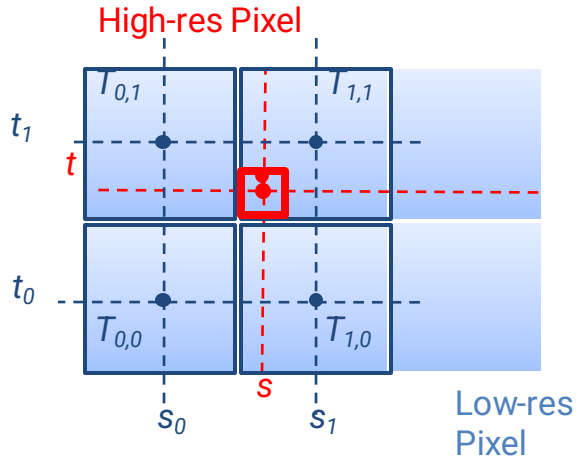
- Interpolation
- Transpose Convolution



Upsampling: Bilineare Interpolation

Nearest Neighbour

Bilinear



- Bilinear Interpolation uses a weighted mean from values $T_{0,0}$, $T_{0,1}$, $T_{1,0}$, $T_{1,1}$.

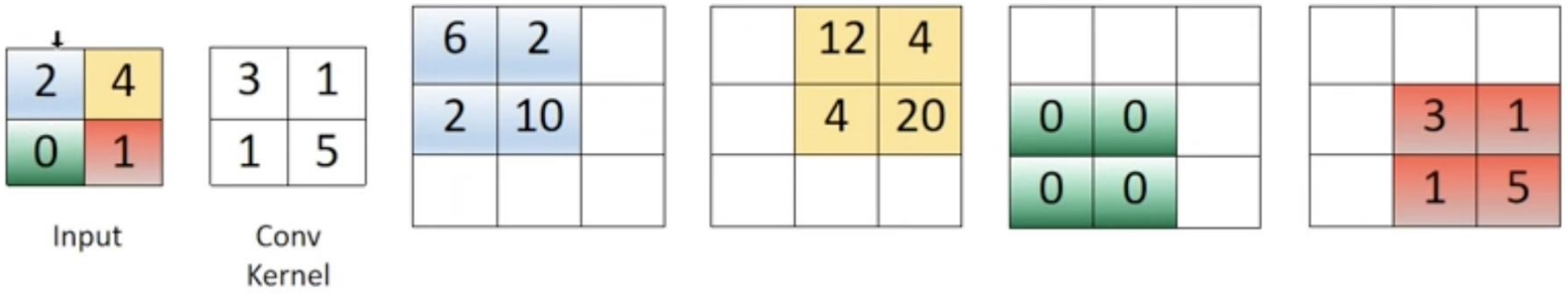
$$T(s,t) = (1-s')(1-t')T_{0,0} + (s')(1-t')T_{1,0} + (s')(t')T_{1,1} + (1-s')(t')T_{0,1}$$

$$\text{mit} \quad s' = \frac{s-s_0}{s_1-s_0} \quad \text{und} \quad t' = \frac{t-t_0}{t_1-t_0}$$

Akenine-Möller et al., *Real-Time Rendering (3rd Edition)*, A K Peters

Transposed Convolution

Transpose convolution 2x2 with stride of 1 and padding of 0



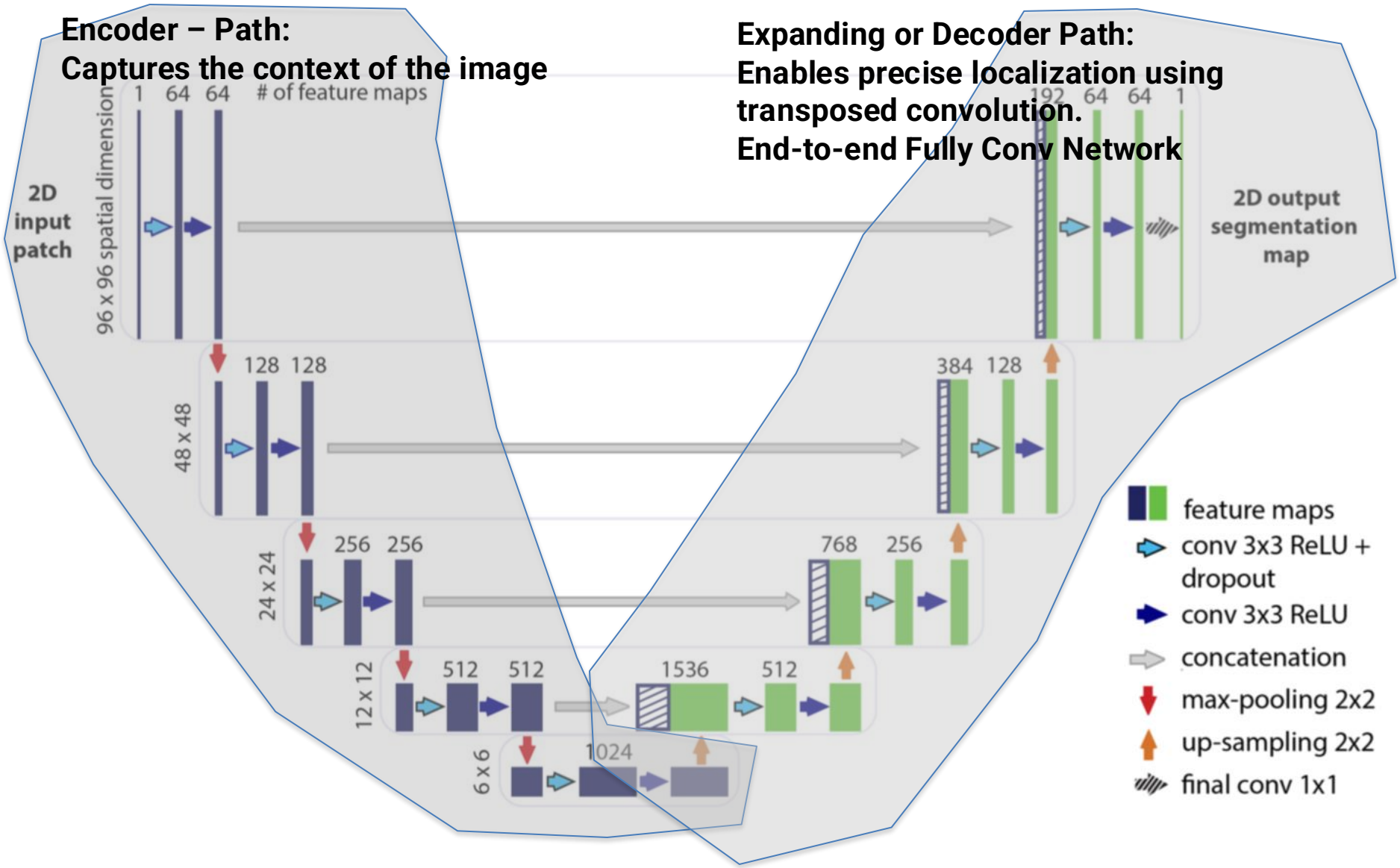
6	14	4
2	17	21
0	1	5

Output

What is an Autoencoder?

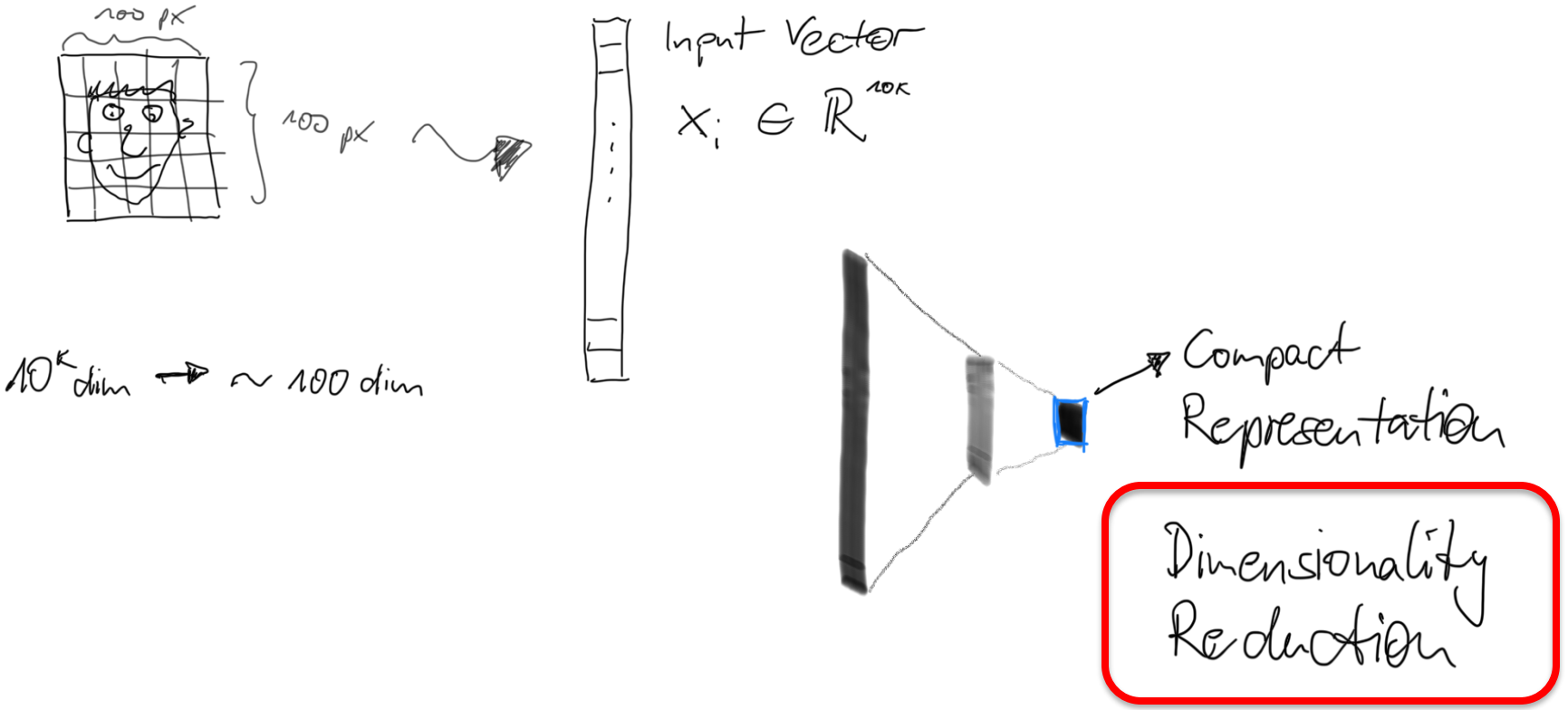
http://localhost:8888/notebooks/Beuth/Teaching/Courses/lfi/Notebooks/IRIS-AE_vs_PCA.ipynb

U-Net for image segmentation

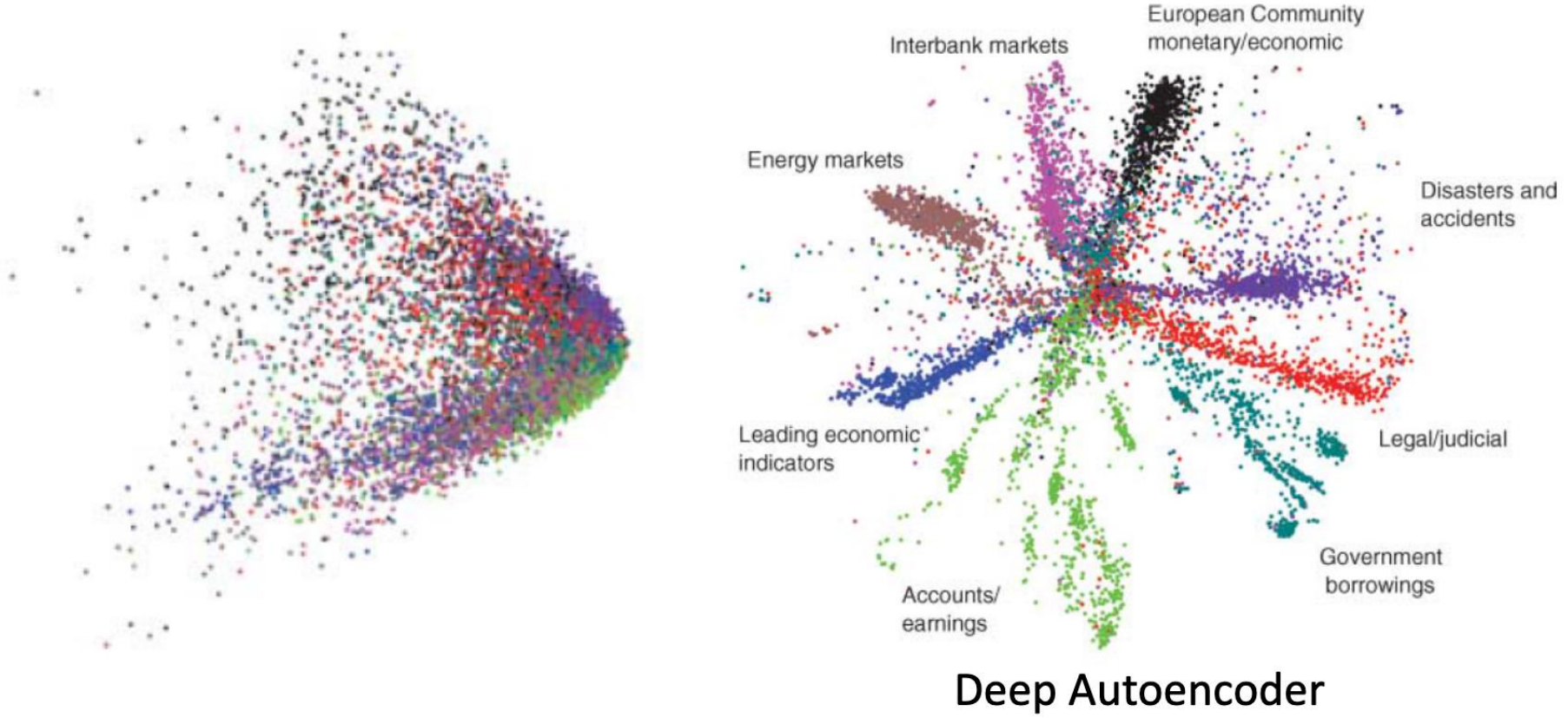


Autoencoder

- Neural network to learn an efficient and compact encoding of the underlying training data



Autoencoder

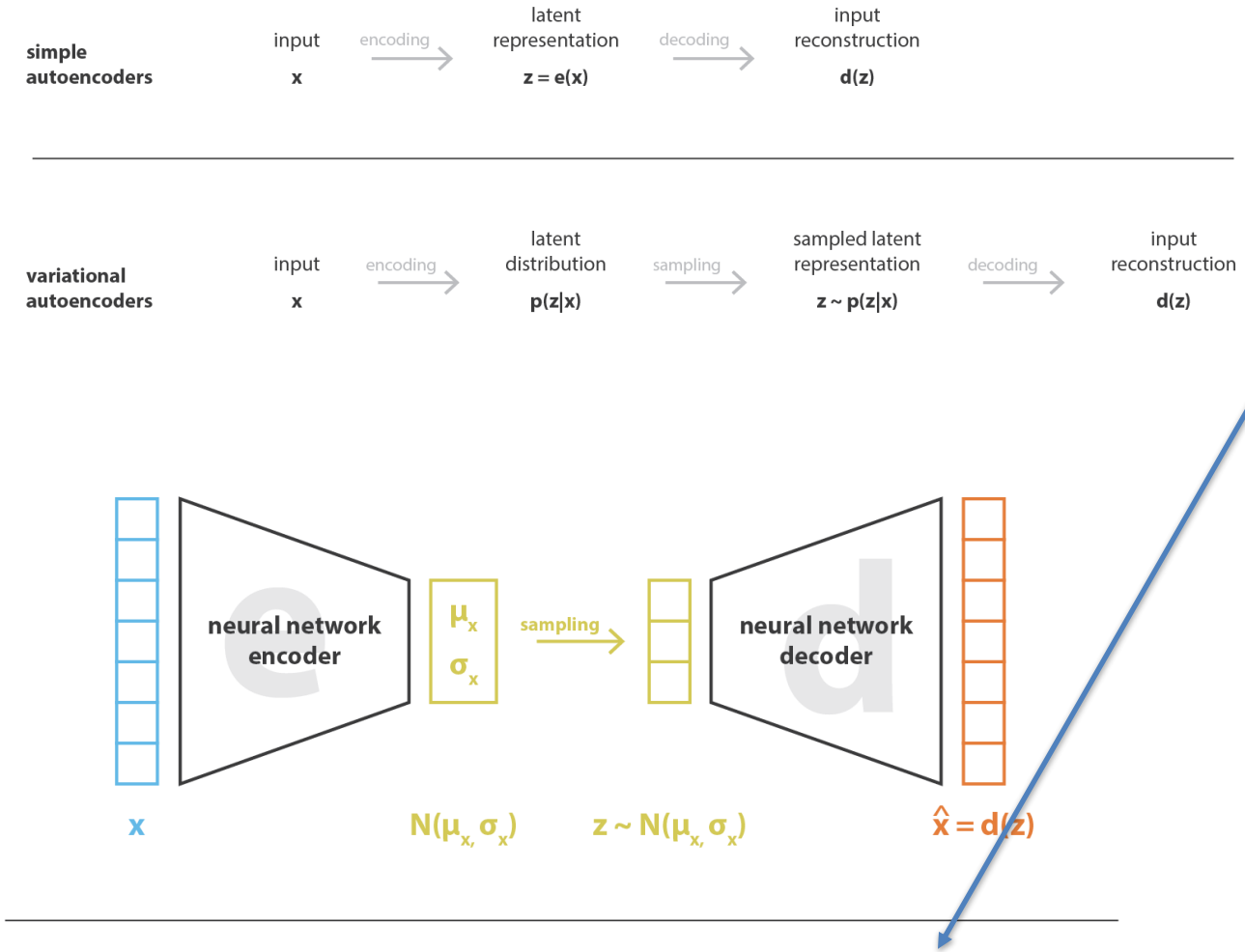


Source: Hinton et al., Reducing the Dimensionality of Data with Neural Networks

Lets talk about PCA and Autoencoder

Variational Autoencoder - Idea

https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73



Variational Autoencoders (VAEs) ~ Autoencoders

- tackle the problem of the latent space irregularity by making encoder return distribution over latent space instead of single point
- adding in the loss function a regularisation term over returned distribution in order to ensure better organisation of latent space

Going towards Generating content.

Training steps:

1. Input is encoded as distribution over the latent space
2. Point from the latent space is sampled from distribution
3. Sampled point is decoded and reconstruction error can be computed
4. Reconstruction error is backpropagated through network

Variational Autoencoder - Regularization

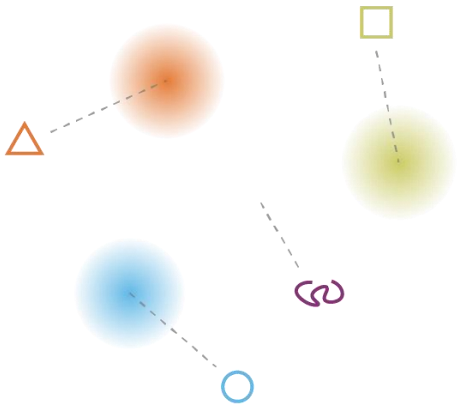
https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

$$KL[\, N(\mu_x, \sigma_x), N(0, I) \,]$$

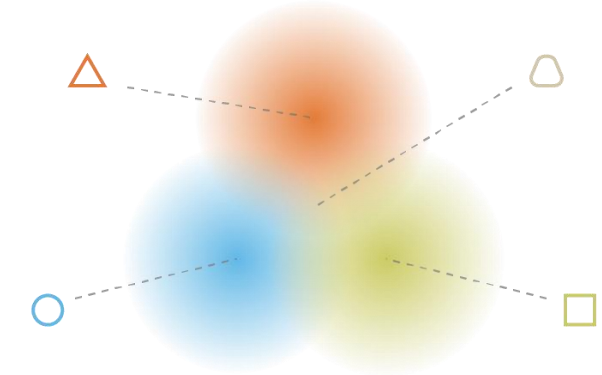
- Regularisation term is expressed as the **Kullback-Leibler divergence** between the returned distribution and a standard Gaussian (is expressed in terms of means and covariance matrices of the two distributions)
- Kullback-Leibler divergence measures the distance between probability distributions
- Relates to **Optimal Transport Problem**: Optimal transport is originally a classical problem, which, starting from a given initial distribution and a desired final distribution, searches for the most favorable transport in which the initial distribution is transformed into the final distribution

Variational Autoencoder – Variational Inference

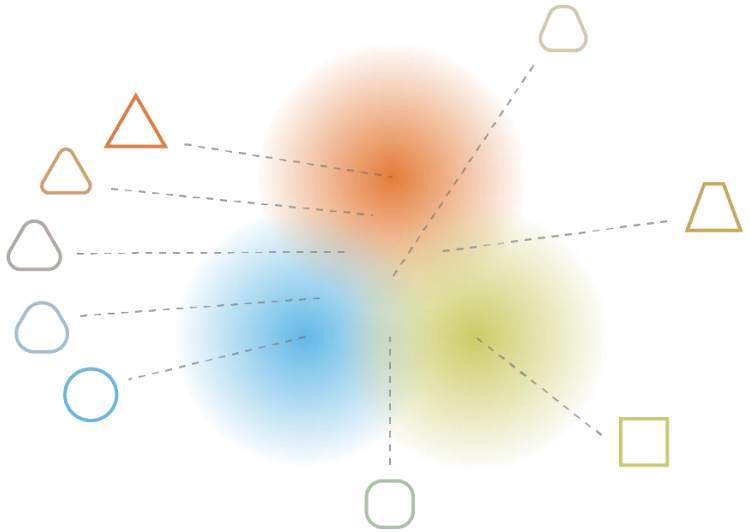
https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73



what can happen without regularisation

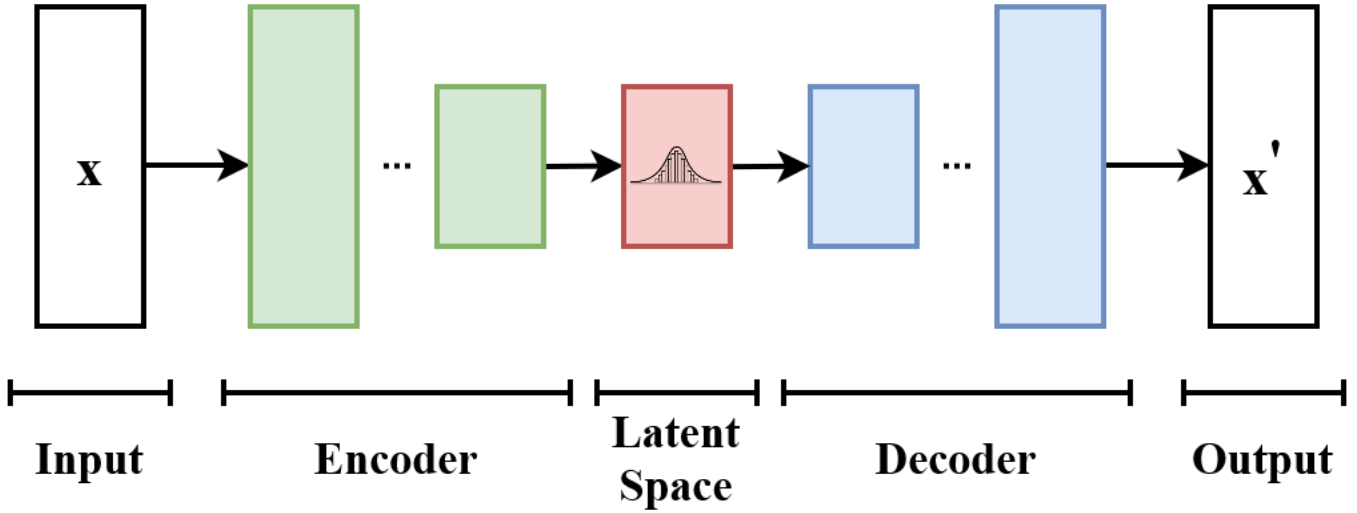


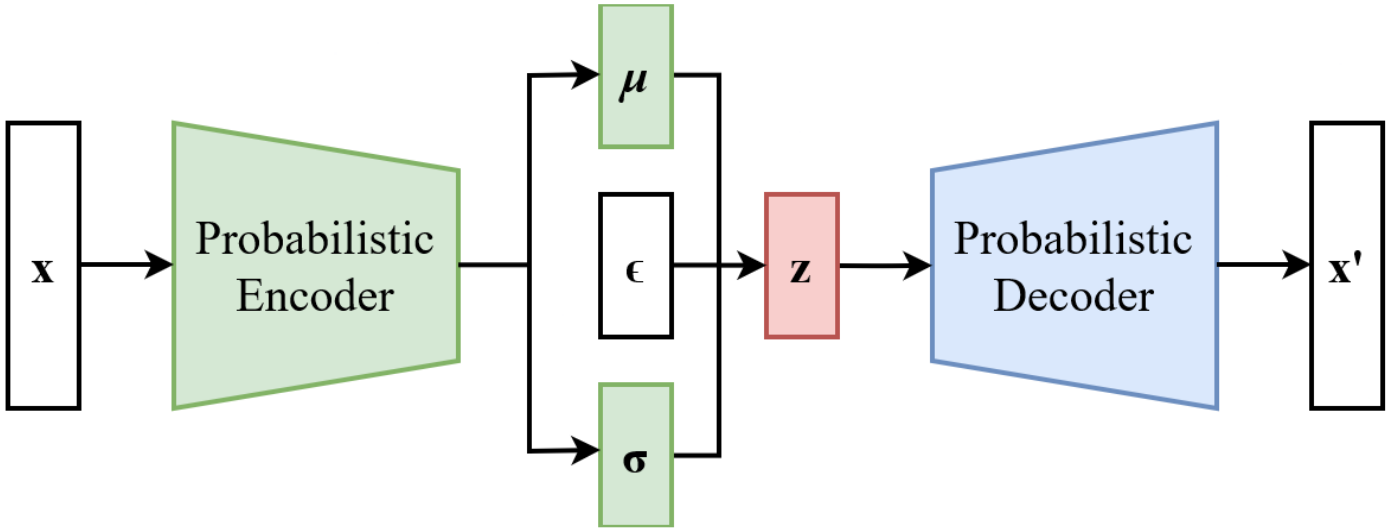
what we want to obtain with regularisation



Autoencoder vs. Variational Autoencoder

https://medium.com/@amir_shakiba/variational-autoencoders-d7928774c9f1

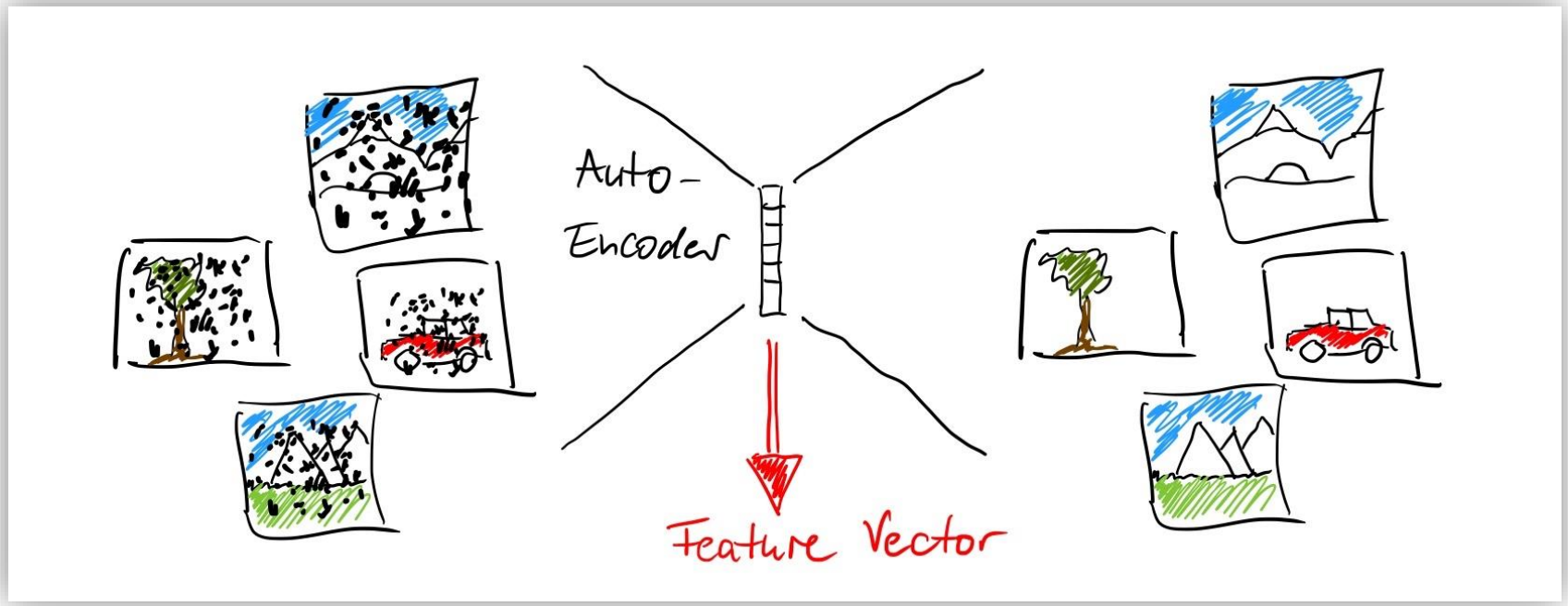




Let's look at some code

Autoencoder – Architectures and Applications

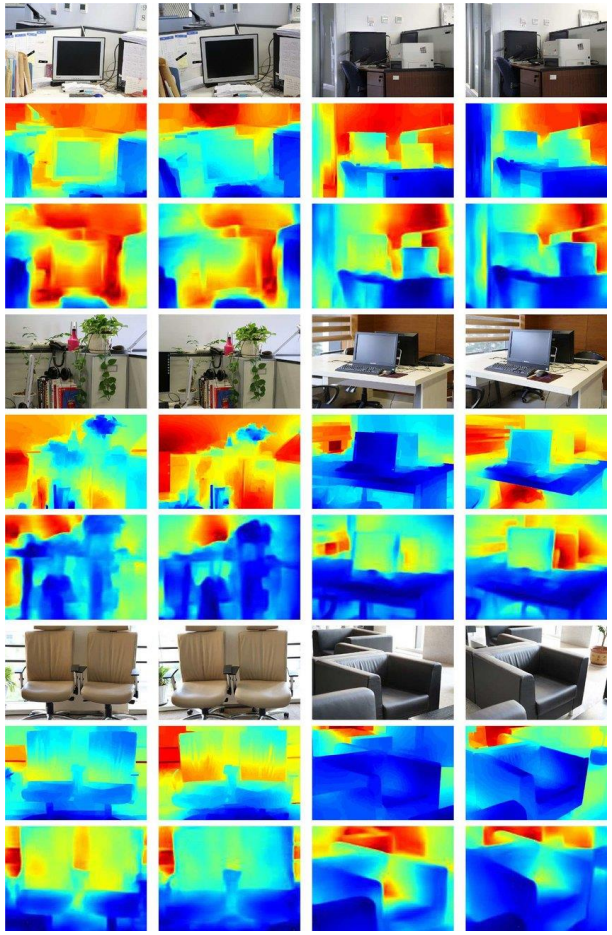
Feature Extraction / Image Retrieval etc.



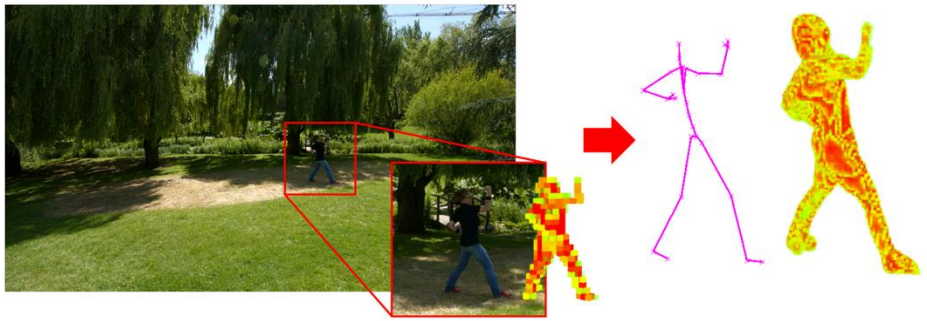
Use Noise to learn stable representations

Autoencoder – Architectures and Applications

Depth Estimation



Human Pose Estimation



Content Generation

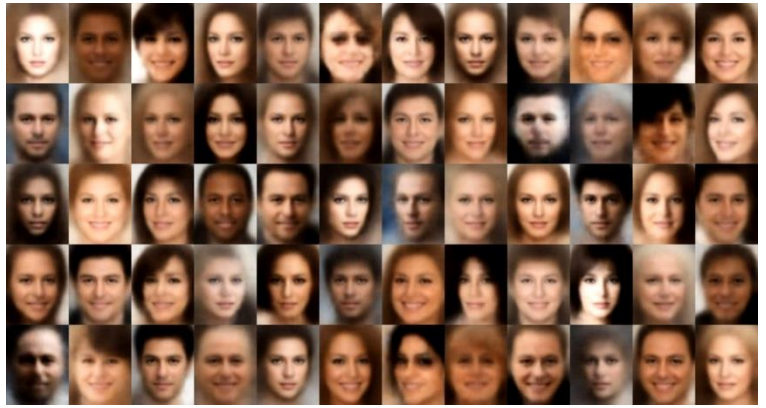
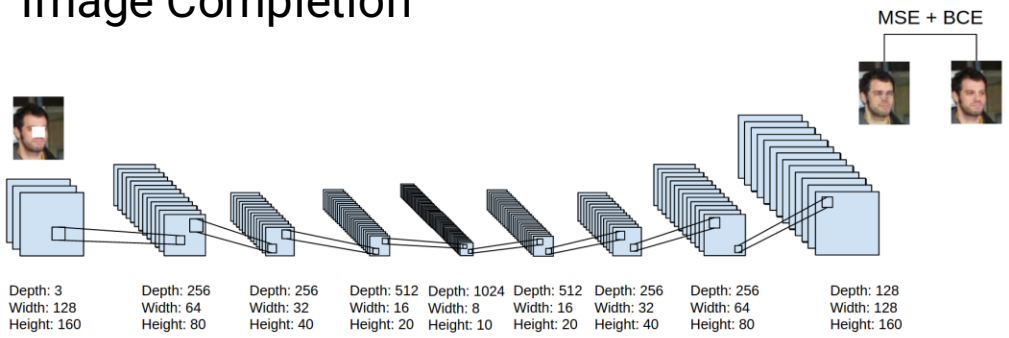


Image Completion

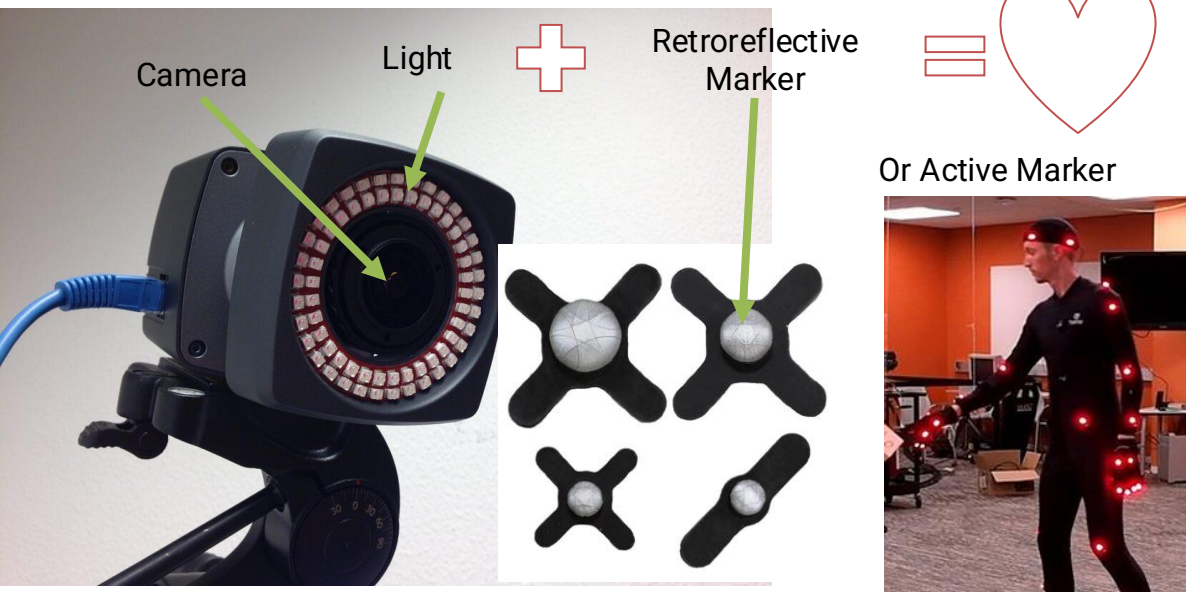


Human Pose Estimation

Optical MoCap Pipeline – Keypoint Detection

More on this when we talk about 3D reconstruction

- Marker-Based approach relatively simple simpel



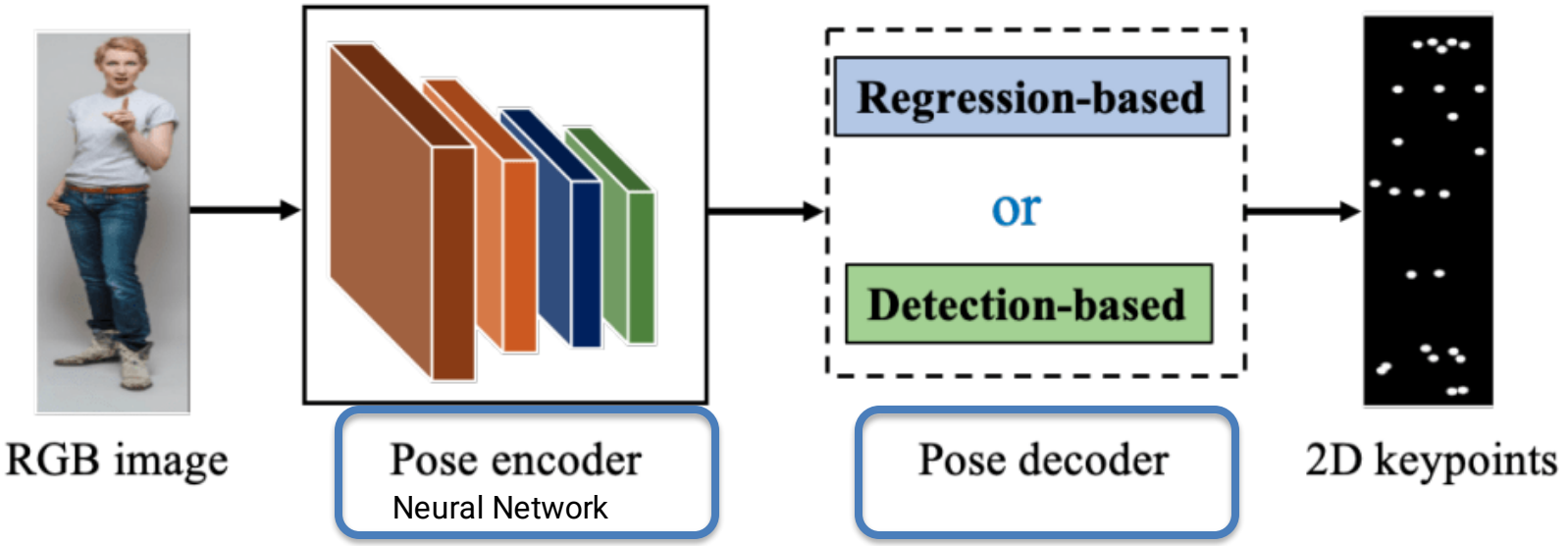
Or Active Marker

- Markers can be extracted relatively easily from the image using a thresholding procedure

- https://www.iaw.rwth-aachen.de/cms/iaw/forschung/methoden/~mqwpg/bewegungstracking/
- https://www.qualisys.com/accessories/markers/
  https://www.cs.utexas.edu/~dana/vrlab/equipment.html
  https://motion-database.humanoids.kit.edu/marker_set/

Markerless Human Pose Estimation

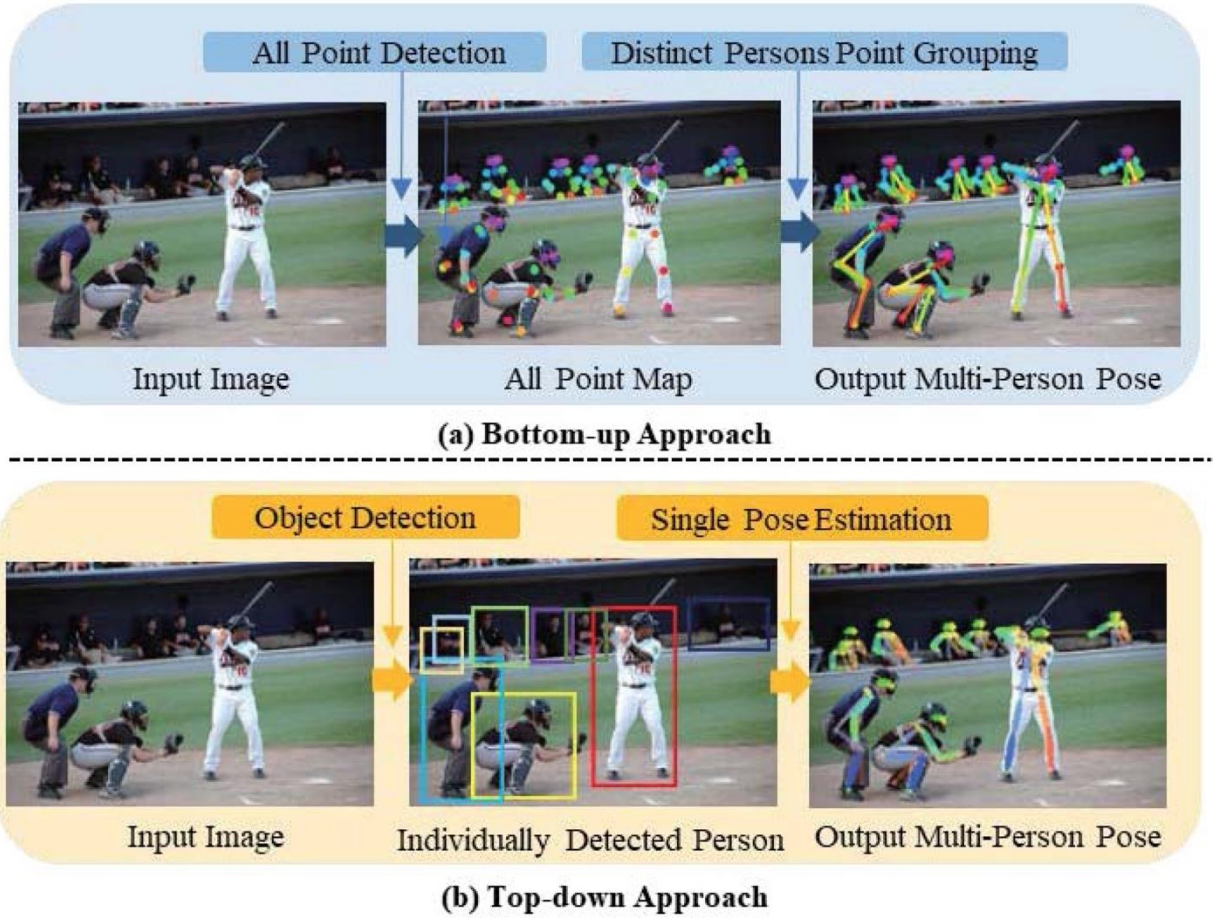
- Markerless (ML-approach) is more difficult
- **Human Pose Estimation**



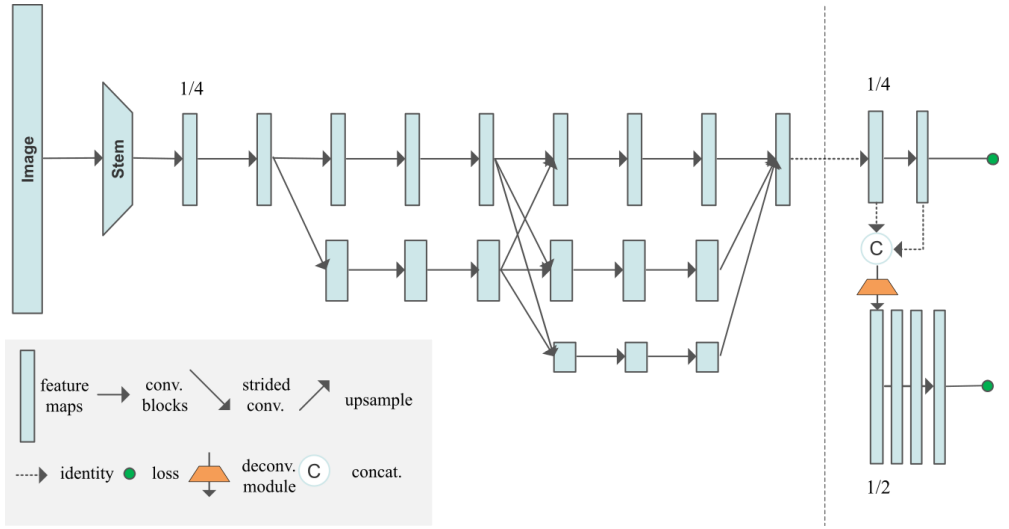
- https://ilyestalbi75.medium.com/introduction-to-human-pose-estimation-abb4e628703

Markerless Human Pose Estimation

- What about several persons?



HRNet, MMPose, OpenPose are possible Standard models

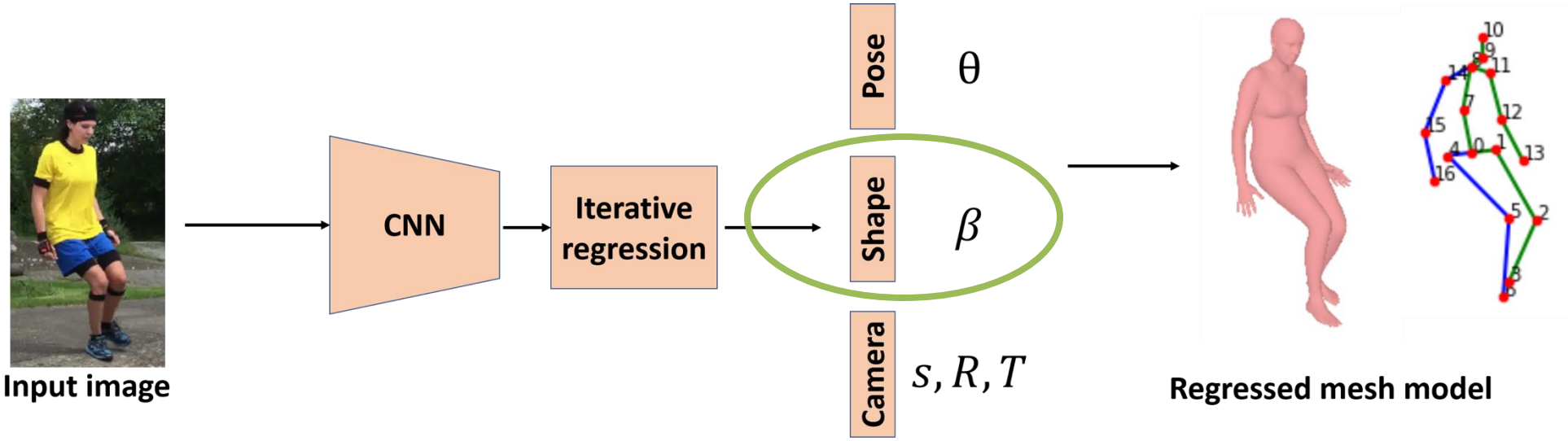


https://github.com/HRNet/HigherHRNet-Human-Pose-Estimation?tab=readme-ov-file

- https://doi.org/10.3390/s21227640

Markerless Human Pose Estimation

- SMPL -> Skinned Multi-Person Linear Model



$$L = L_{smpl} + L_{reproj} + L_{generic}$$

- https://sites.ecse.rpi.edu/~cvrl/HumanActionActivity/3Dpose_Yufei/3D_latest.html

Figure 4: Overview of the proposed model architecture