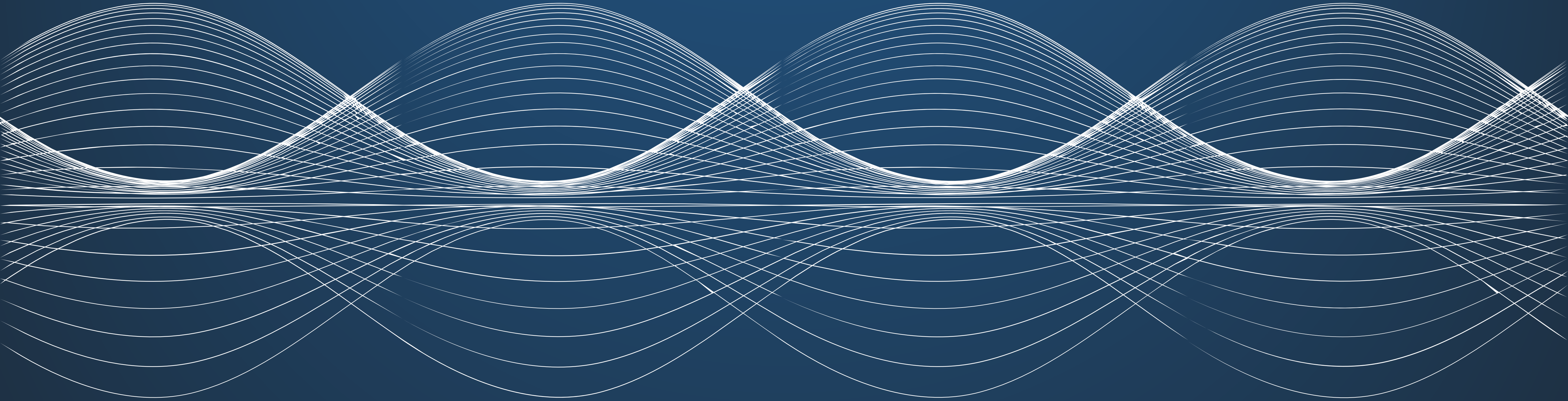


Image Processing I: Object detection and vision transformers

Research Methods in Data Science – 7PAM2015

Dr Vanessa Graber (based on slides by Dr Gülay Gürkan)

Summary Lecture 1

- We discussed the course outline and your lecturer for this module.
- **Image classification** is a fundamental task in image processing, where we assign labels to images. This is generally **simpler than object detection and segmentation**, which we will cover next.
- Large labelled datasets like **ImageNet** were crucial to obtain the superior performance of neural networks in image classification.
- When labels are difficult to obtain or limited data is available, **transfer learning** allows us to use knowledge from previously trained models, because the first layers of a CNN learn generic image features.
- Additionally, **self-organising maps** can be used for unsupervised learning. Inspired by topographic maps, they allow us to reduce the dimensionality of complex data via a competitive learning approach.

Learning outcomes

After this lecture and the tutorial, you will:

- Understand main concepts of object detection & its applications and have seen several methods for this type of image processing.

- Know what attention is, how this concept is implemented in vision transformers (ViTs) and what role ViTs play in image processing.

- Be able to implement object detection in Python using Keras.

- Have set the foundations to understand image segmentation.

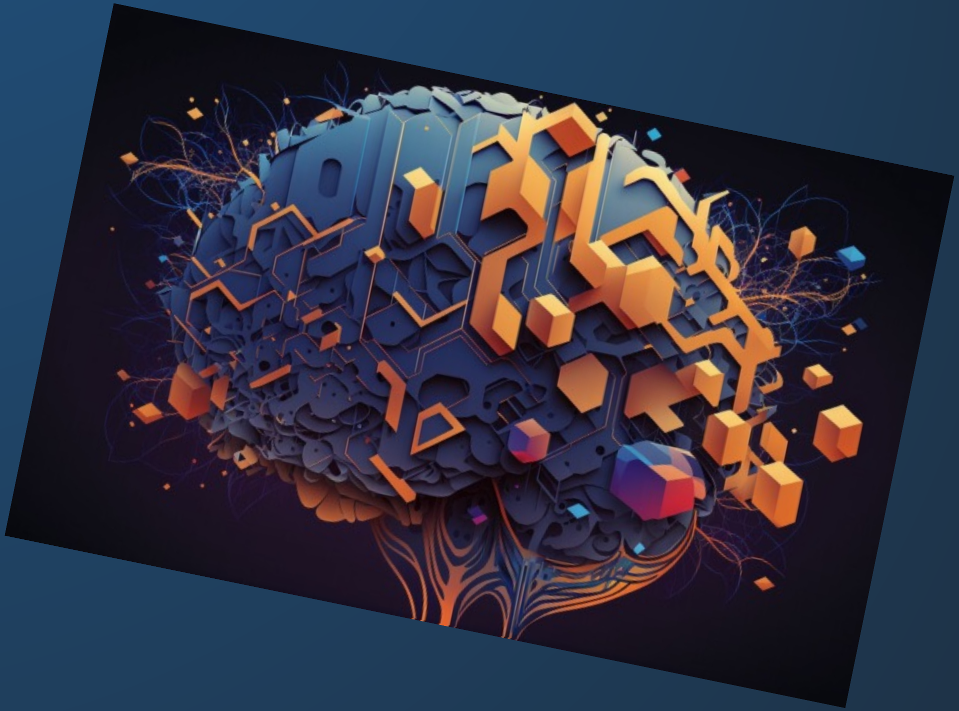
Object detection overview

Object detection with NNs

The concept of attention

Vision transformers

Summary



Object detection overview

Object detection with NNs

The concept of attention

Vision transformers

Summary

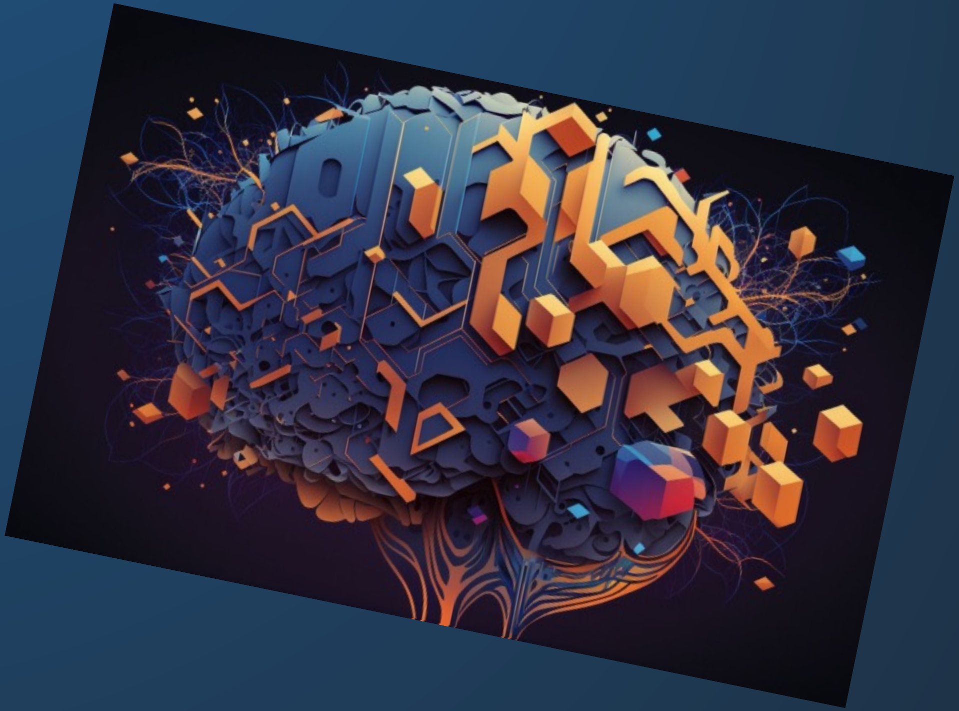


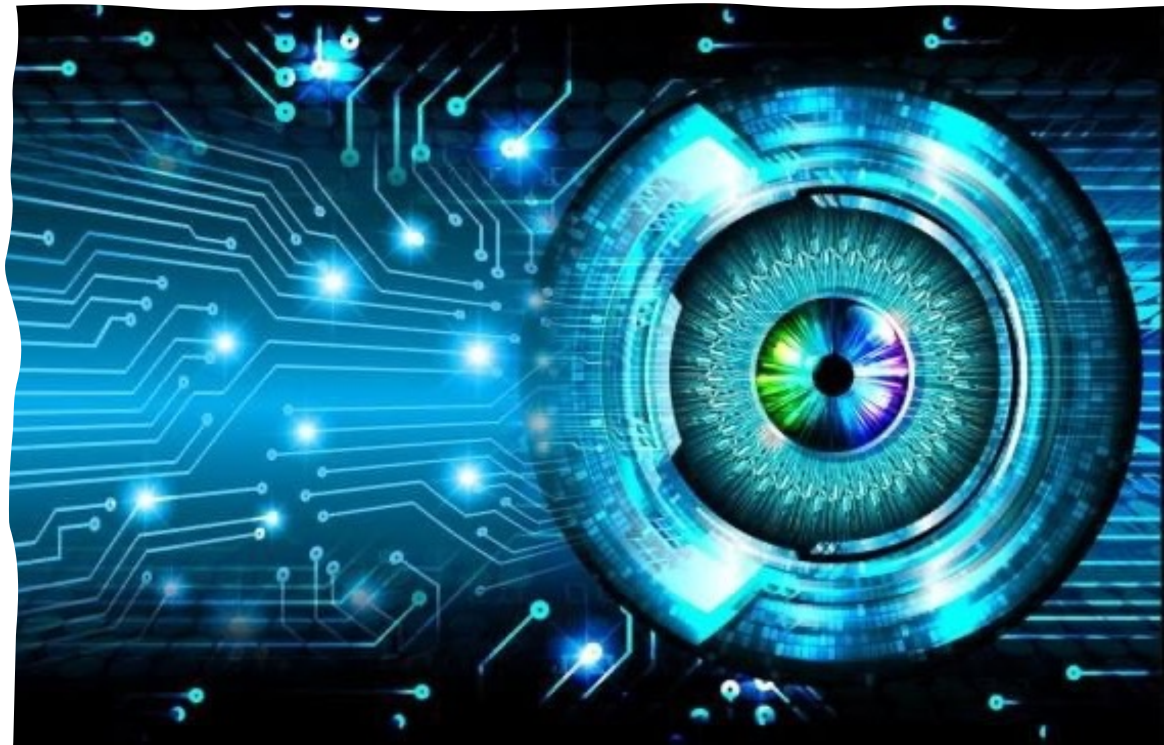
Image recognition

Overview

- Image recognition is a subfield of computer vision that teaches machines how to understand and “see” visual data.

- We typically distinguish the following tasks:
  - Image classification
  - Object detection
  - Image segmentation

In this lecture, we will focus on object detection.



Object detection

The central idea

- Object detection is the field of computer vision that deals with the **localisation and classification** of objects in images and videos.
- As we saw last week, in image classification, we **assign a label** to an entire image. This does not tell us which part of the image led to that specific label.

Credit: Byrdyak

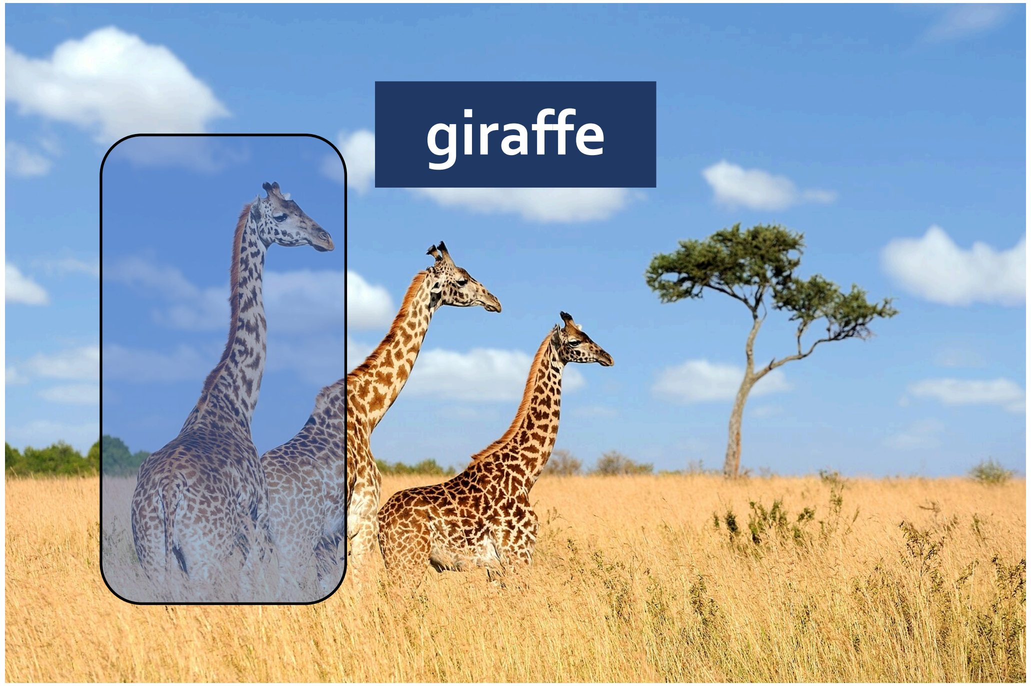
Object detection

The central idea

- Object detection is the field of computer vision that deals with the **localisation and classification** of objects in images and videos.

- As we saw last week, in image classification, we **assign a label** to an entire image. This does not tell us which part of the image led to that specific label.

- In object detection, the goal is to **create a bounding box** around the object and label it.



Credit: Byrdyak

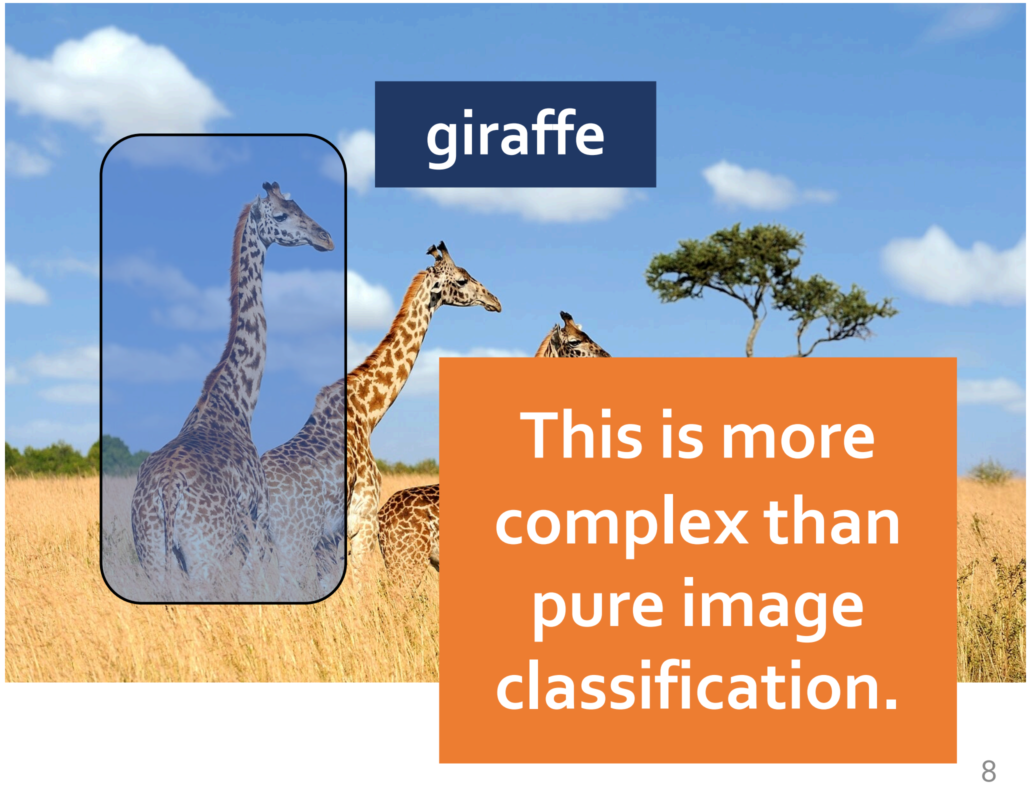
Object detection

The central idea

- Object detection is the field of computer vision that deals with the **localisation and classification** of objects in images and videos.

- As we saw last week, in image classification, we **assign a label** to an entire image. This does not tell us which part of the image led to that specific label.

- In object detection, the goal is to **create a bounding box** around the object and label it.



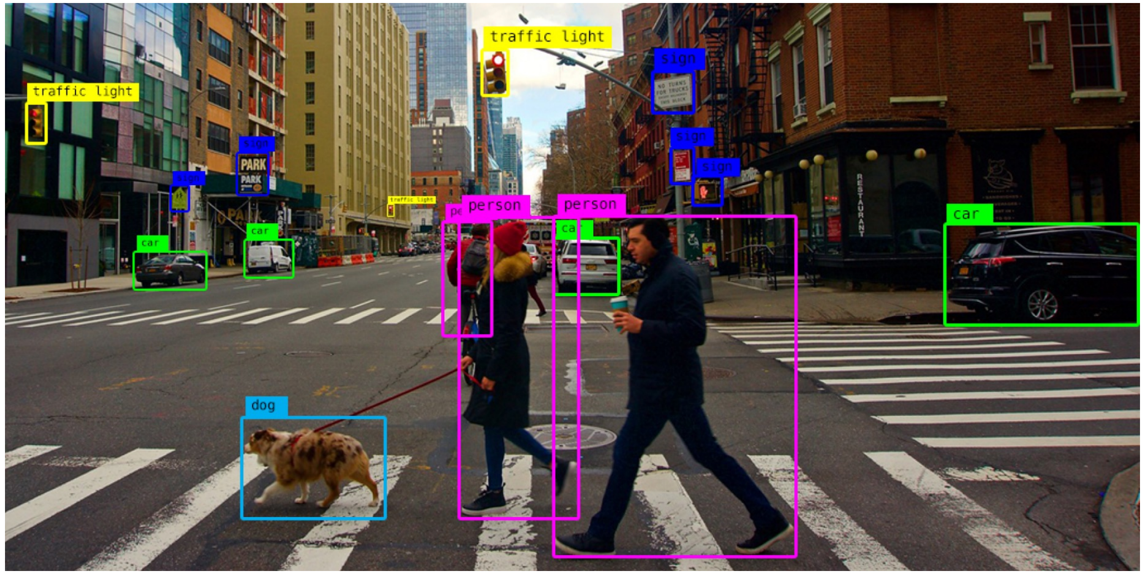
This is more complex than pure image classification.

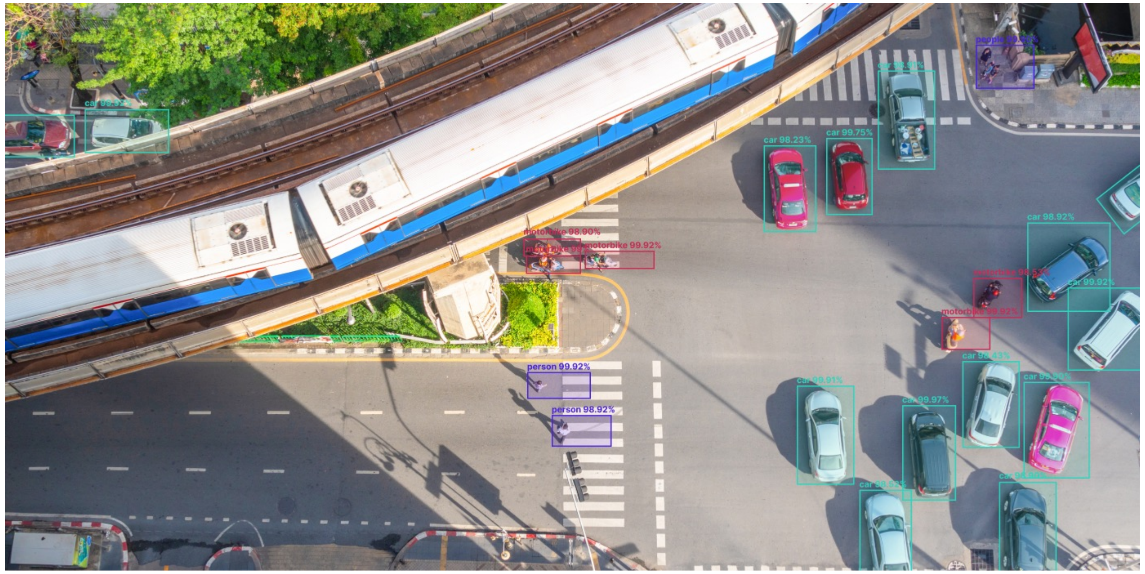
Credit: Byrdyak

Object detection

Comparison to segmentation

- Object detection is focused on **clearly defining the location** of individual objects in an image with boxes. This approach is not pixel-by-pixel based and therefore simpler than segmentation.





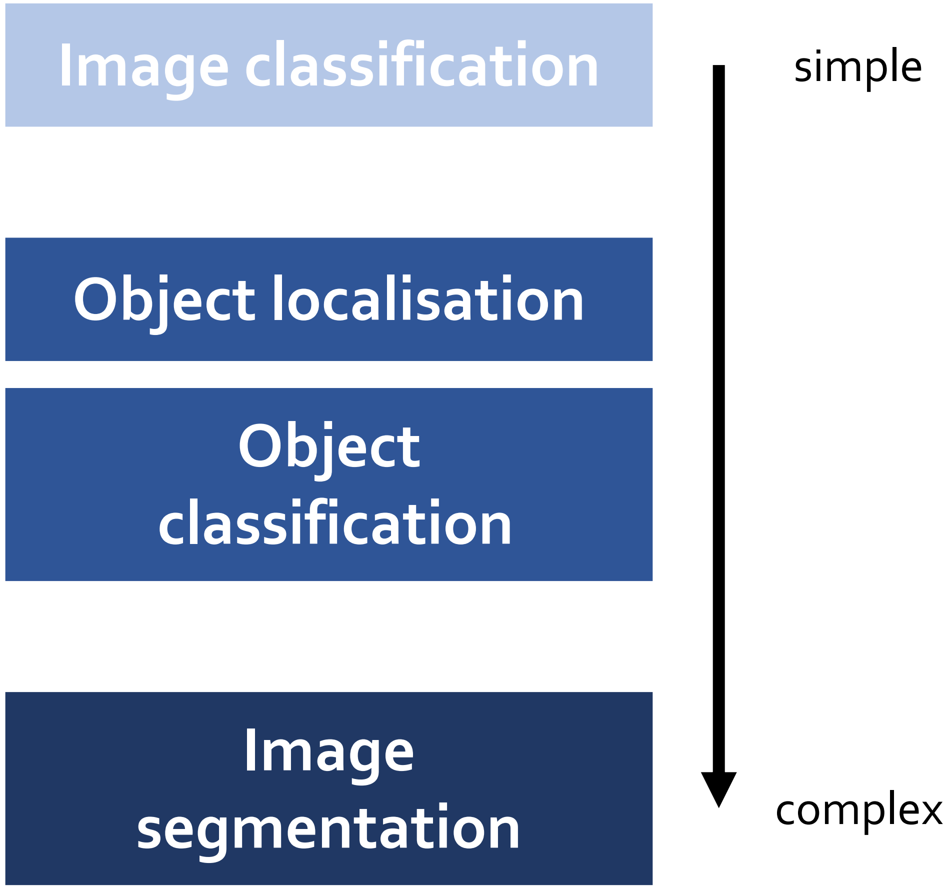
Credit: https://alwaysai.co/blog/object-detection-for-businesses

Object detection

Complexity level

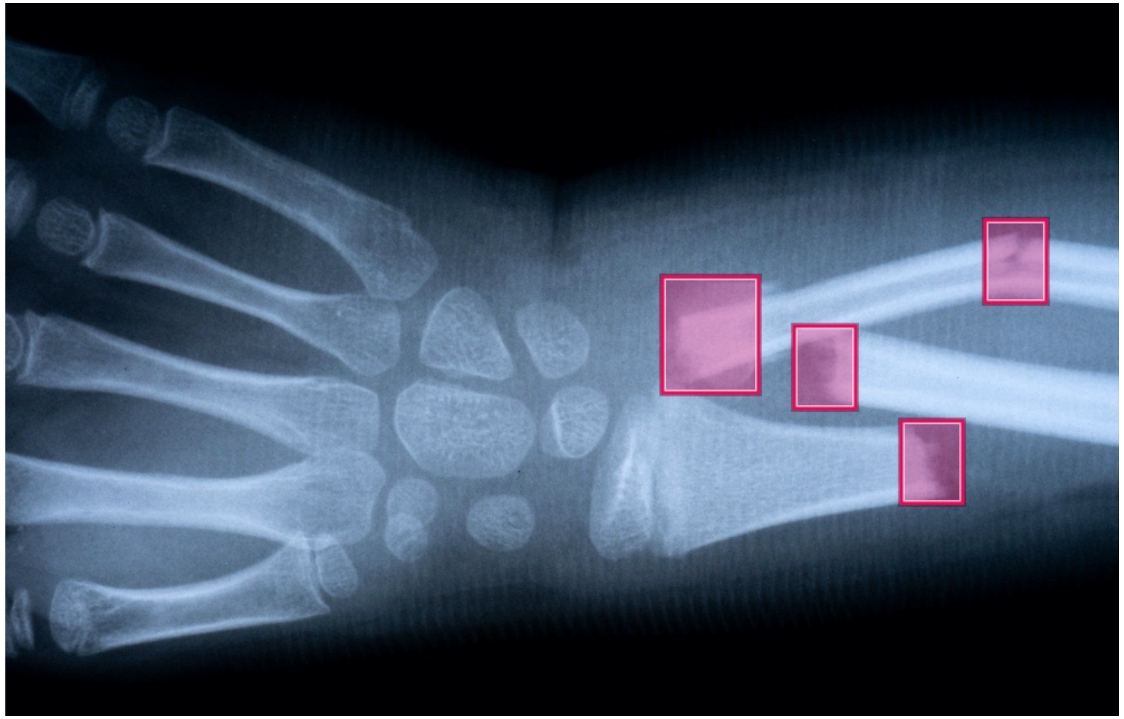
- Object detection is more complex than classification as localising and detecting an object requires the algorithm to learn and process spatial information.

- It is simpler than segmentation as we do not want to delineate the object boundaries, only provide a bounding box and class label.



Object detection
Applications

- Object detection has a **wide range of applications**, including
  - **Retail** (inventory management, customer tracking, fraud detection)
  - **Manufacturing** (quality control, defect detection, equipment maintenance)
  - **Healthcare** (medical image analysis for disease diagnosis, surgical assistance, patient monitoring)
  - **Transportation** (autonomous driving, traffic management, cargo monitoring)
  - **Surveillance** (intruder detection, crowd and animal monitoring, facial recognition, perimeter security)

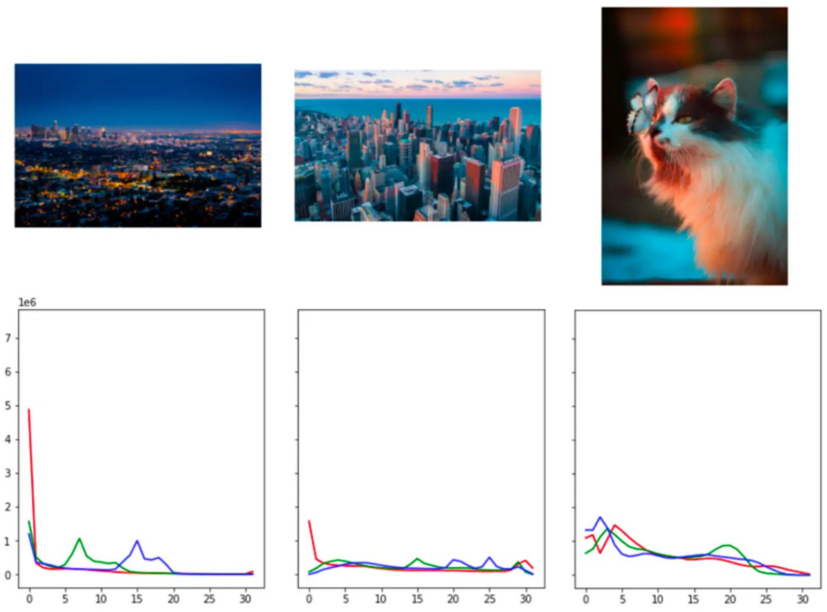


Credit: https://roboflow.com/industries/healthcare-and-medicine/

Object detection techniques

Traditional ML approaches

- Before neural networks were available, classical ML algorithms for object detection focused on extracting information (e.g., common features) using linear regression, random forests or colour histograms:



Credit: https://www.pinecone.io/learn/series/image-search/color-histograms/

- **Scale-invariant feature transforms (SIFTs):** algorithm to identify local features in images regardless of scale or orientation (Lowe 1999)
- **Viola-Jones object detection:** face-detection algorithm that uses a cascading structure of classifiers to identify objects (Viola & Jones 2001)
- **Oriented gradients:** algorithm measuring intensity features (changes in colour/brightness) and their orientations to extract edges and texture patterns (McConnell 1986; Dalal & Triggs 2005)

Object detection techniques

Deep learning

- Deep learning-based approaches using neural networks have been outperforming classical approaches since the early 2010s:
  - **Region-based CNNs:** Based on the seminal paper by Girshick et al. (2014) these CNNs extract rich feature hierarchies for accurate object detection and semantic segmentation (R-CNN, Fast R-CNN, Faster R-CNN, Cascade R-CNN).
  - **YOLO (You Only Look Once) CNNs:** Developed by Redmon et al. (2016) these CNNs enable real-time object detection.
  - **Other alternatives:** Single Shot Multi-detector (SSD; Liu et al. 2016), RetinaNet (Lin et al. 2018), CenterNet (Duan et al. 2019)

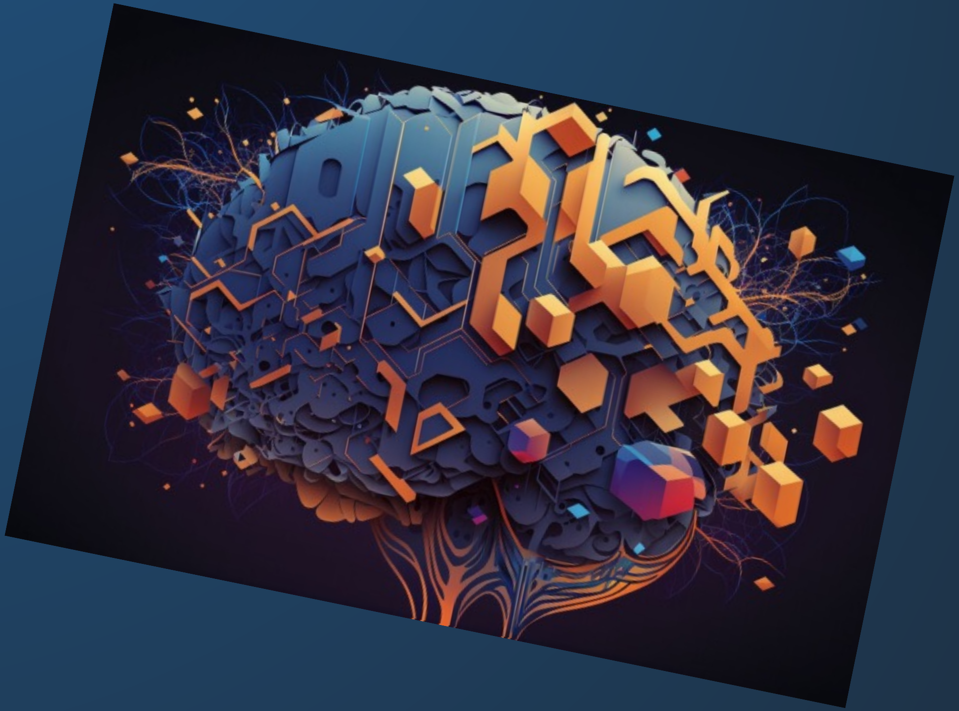
Object detection overview

Object detection with NNs

The concept of attention

Vision transformers

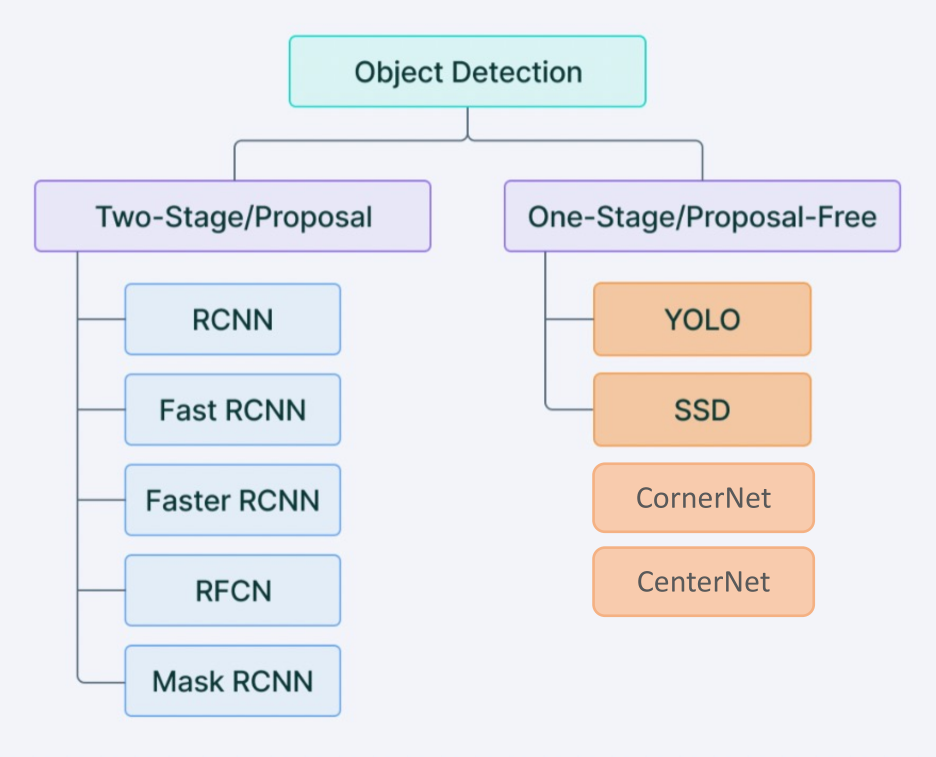
Summary



NN object detection

Two distinct frameworks

- Object detectors with neural networks can be categorised as follows:



Adapted from: https://www.v7labs.com/blog/object-detection-guide

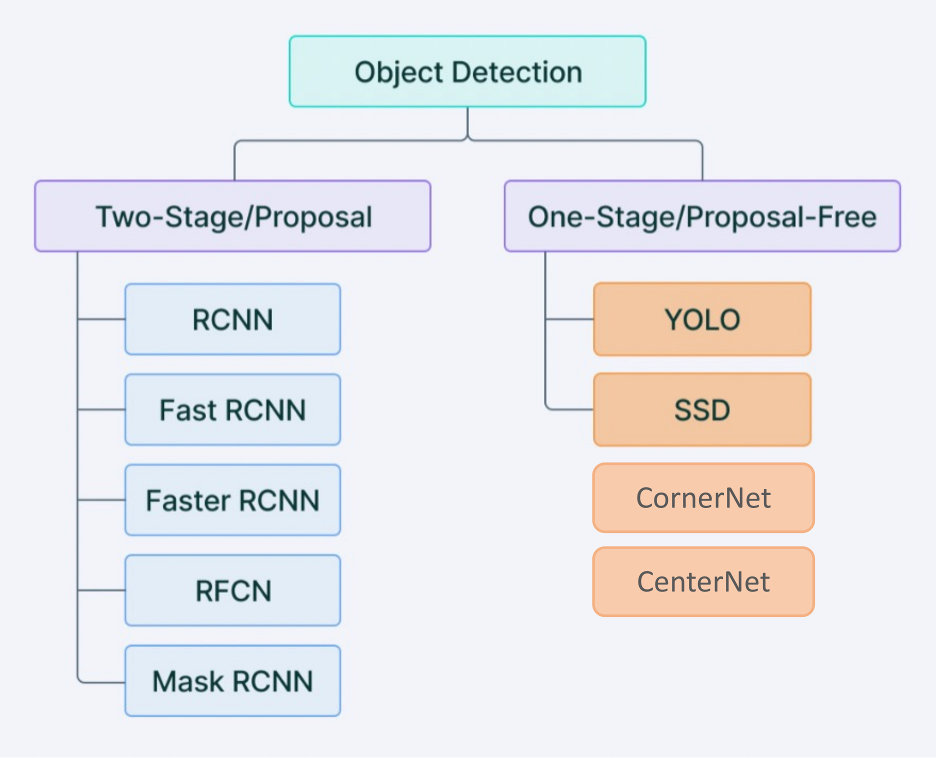
NN object detection

Two distinct frameworks

- Object detectors with neural networks can be categorised as follows:

TWO-STAGE:

- First extracting regions of interest (RoI), then classifying these RoIs.
- Typically, pretrained on a large dataset such as the COCO (Common Objects in Context) or Open Images datasets.



Object Detection

Two-Stage/Proposal

- RCNN
- Fast RCNN
- Faster RCNN
- RFCN
- Mask RCNN

One-Stage/Proposal-Free

- YOLO
- SSD
- CornerNet
- CenterNet

ONE-STAGE:

- Using a single NN trained end-to-end to predict bounding boxes and class label for each.
- No RoI extraction but direct classification of objects enabling real-time object detection.

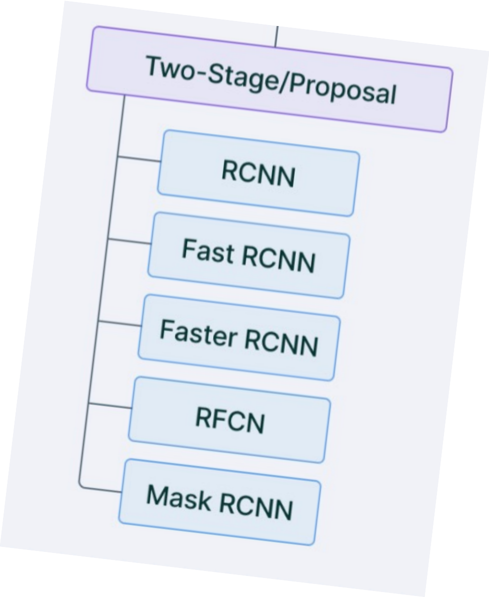
Adapted from: https://www.v7labs.com/blog/object-detection-guide

R-CNNs

Main components

- We first focus on R-CNNs that are based on identifying certain regions of interest (RoIs) in the image and then classifying these.

- As a result, R-CNNs are composed of the following components



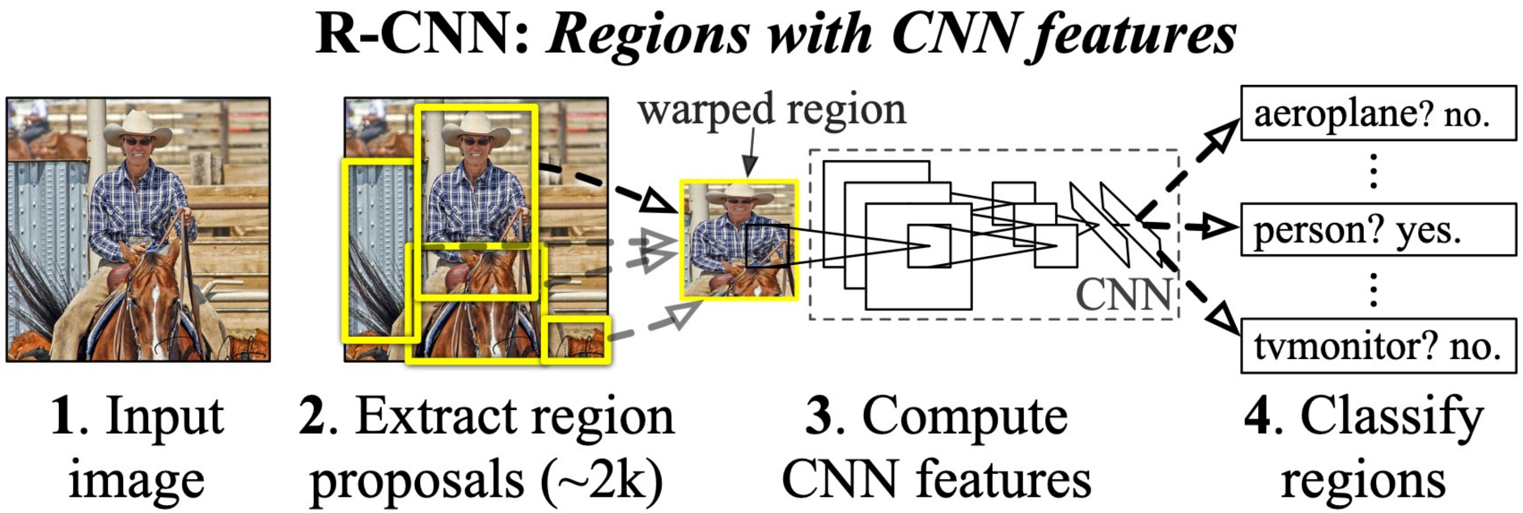
Credit: https://www.v7labs.com/blog/object-detection-guide

- **Region proposer:** construct and extract category independent region proposals, i.e., candidate bounding boxes / regions
- **Feature extractor:** extract features from each candidate box using a CNN
- **Classifier:** classify the features extracted from the candidate region

R-CNNs

Original algorithm

- In their seminal 2014 paper, Girshick et al. used the following setup:

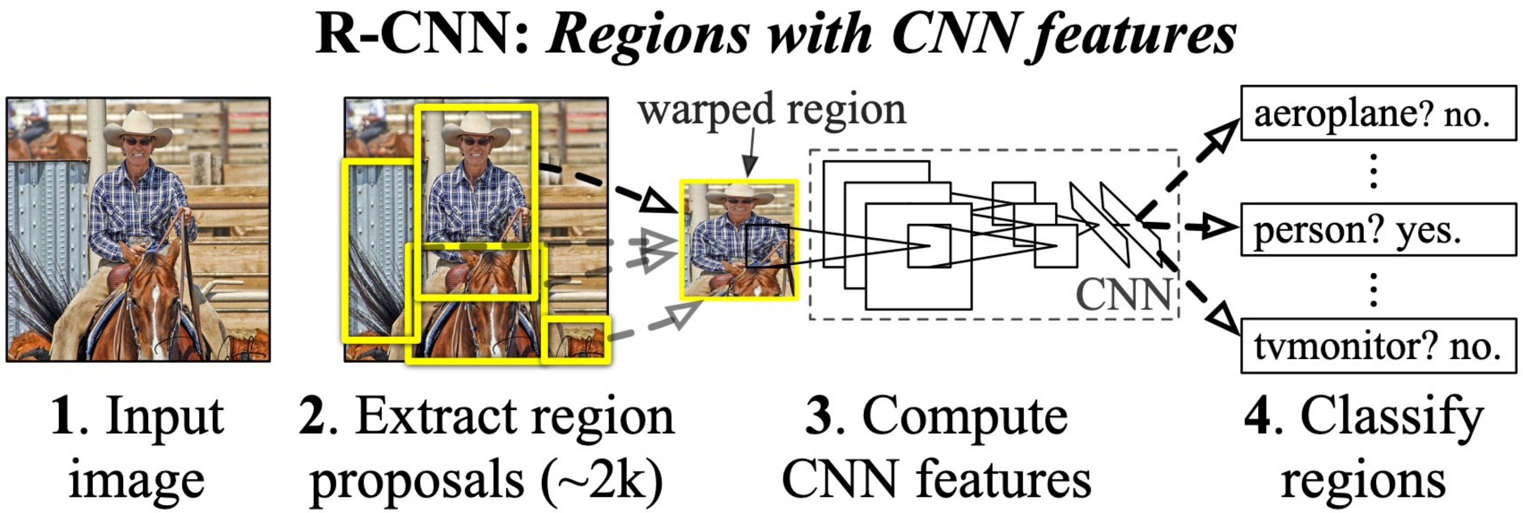


Credit: Girshick et al. (2014)

R-CNNs

Original algorithm

- In their seminal 2014 paper, Girshick et al. used the following setup:

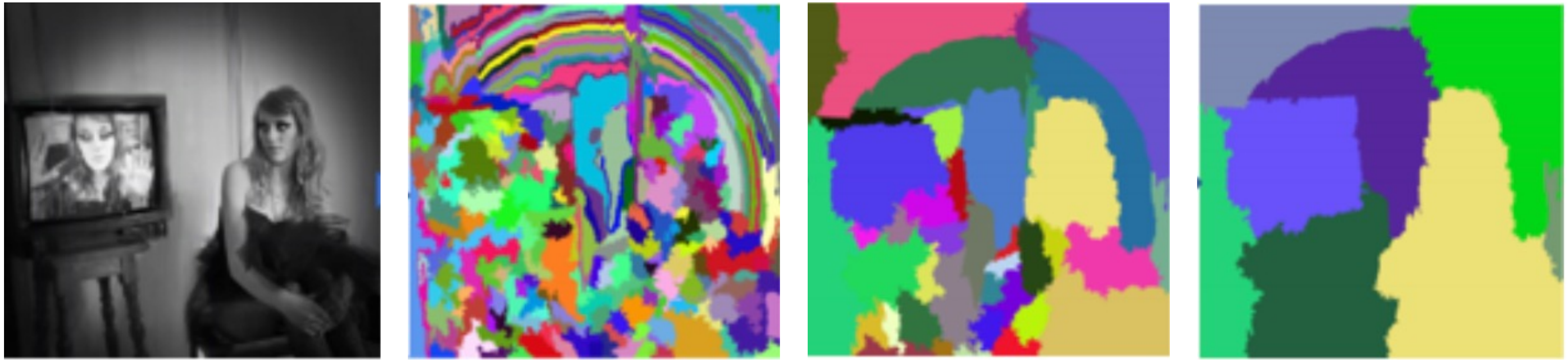


Credit: Girshick et al. (2014)

R-CNNs

Region proposal

- By 2014, several classical ML methods had been developed to generate **category-independent region proposals**. Girshick et al. focused on **Selective Search** (Uijlings et al. 2012). This fast method starts with image sub-segments based on similarities such as colour, texture, etc., which are then iteratively combined to larger regions.



Input image

First iteration

Several iterations

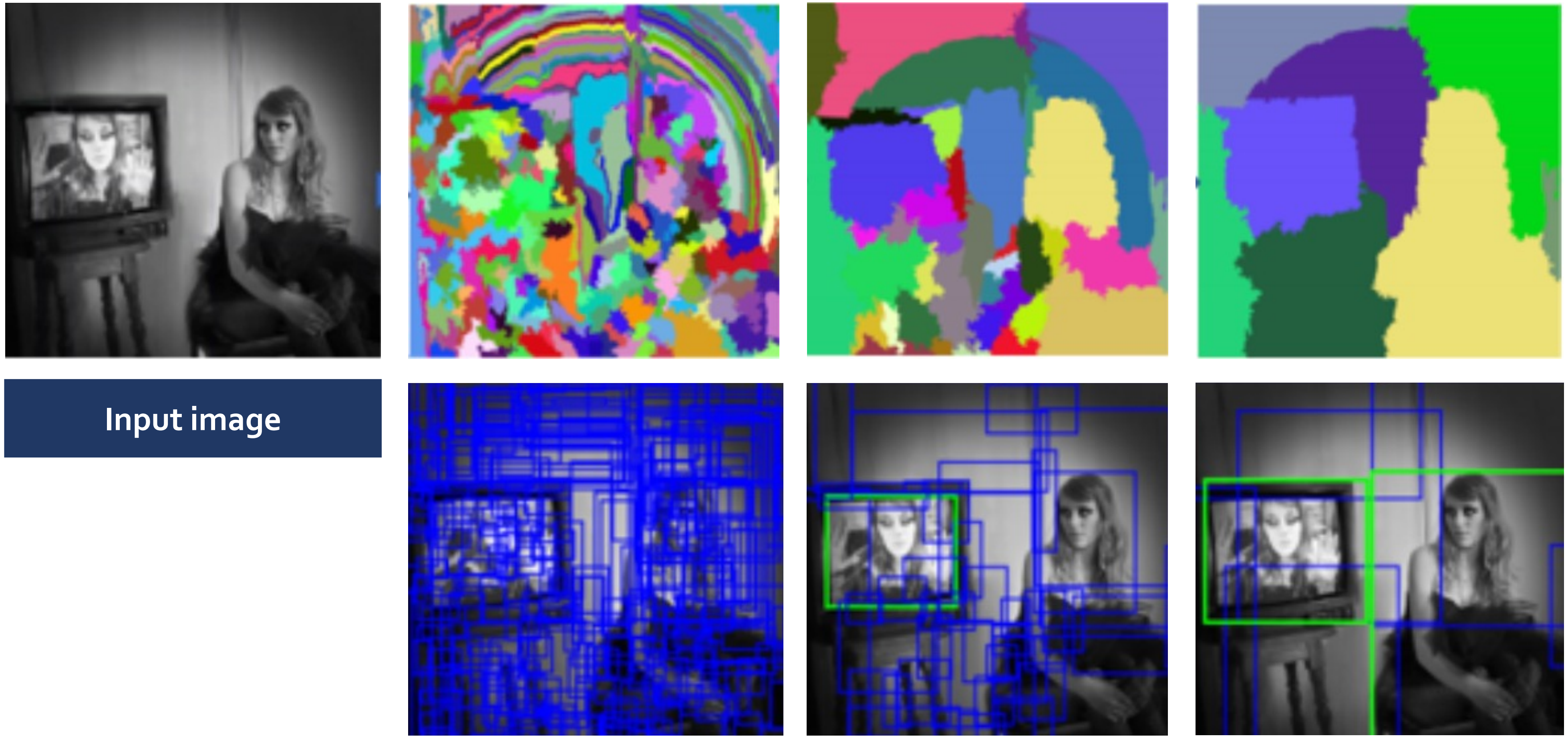
Many iterations

Credit: Uijlings et al. (2012)

R-CNNs

Region proposal

- Individual regions can then be associated with bounding boxes.

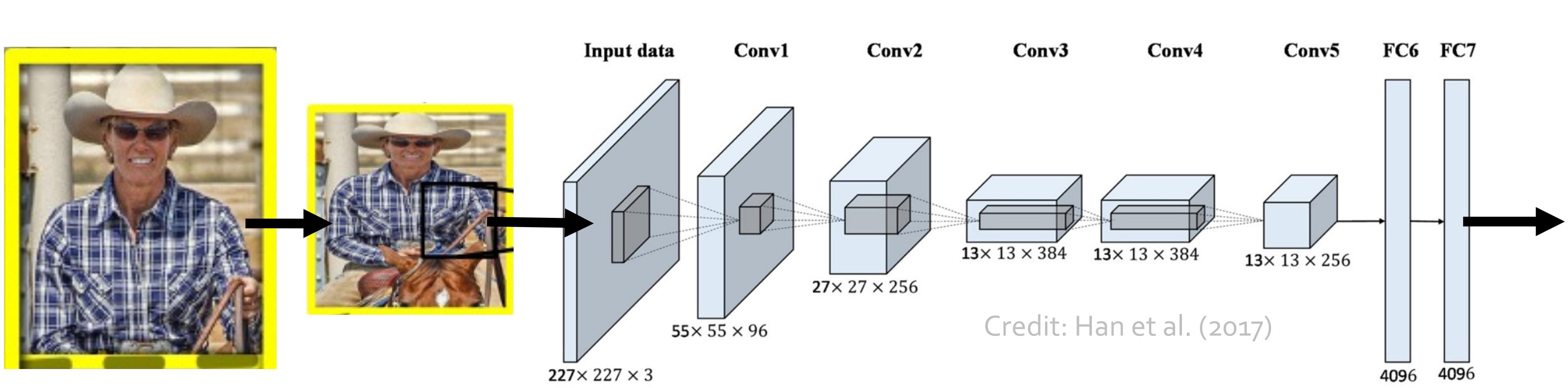


Credit: Uijlings et al. (2012)

R-CNNs

Feature extraction

- Based on the final RoIs, Girshick et al. (2014) then resized the images to a 227 x 227 pixel RGB image and used AlexNet (Krizhevsky et al. 2012) to extract a single feature vector for each proposal region.



Credit: Han et al. (2017)

(N+1)-dimensional feature vector (N = no. of classes plus background) for each given RoI

The CNN enables the extraction of RoI features in form of a compressed (lower-dimensional) vector!

R-CNNs

Classifier

- The feature vectors are then fed into a separate ML classifier for each object class. R-CNNs typically use Support Vector Machines (SVMs). For each class, a unique binary SVM is trained to determine whether the proposal region contains an instance of that class. Positive (negative) samples are those where objects are present (absent).

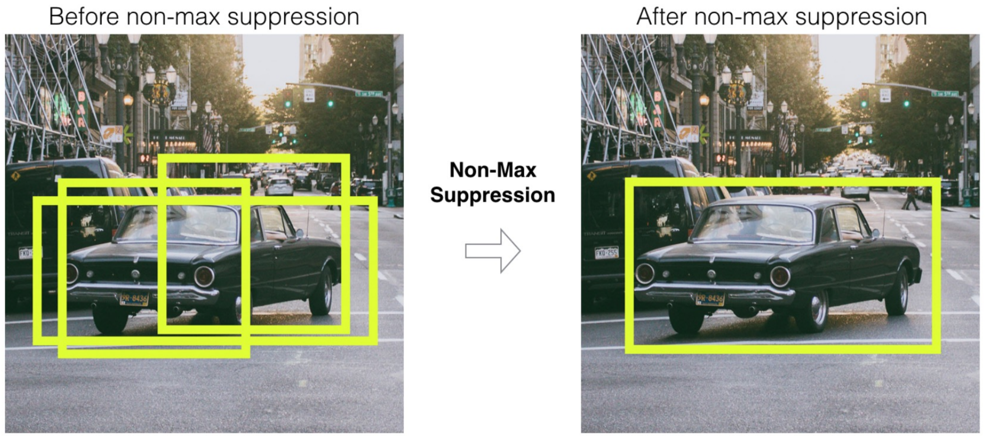
R-CNNs

Classifier

- The feature vectors are then fed into a separate ML classifier for each object class. R-CNNs typically use Support Vector Machines (SVMs). For each class, a unique **binary SVM is trained** to determine whether the proposal region contains an instance of that class. Positive (negative) samples are those where objects are present (absent).

- R-CNNs also perform **bounding box regression** to refine the location and size of bounding boxes around detected objects. This **refinement** leads to better accuracy in object localisation.

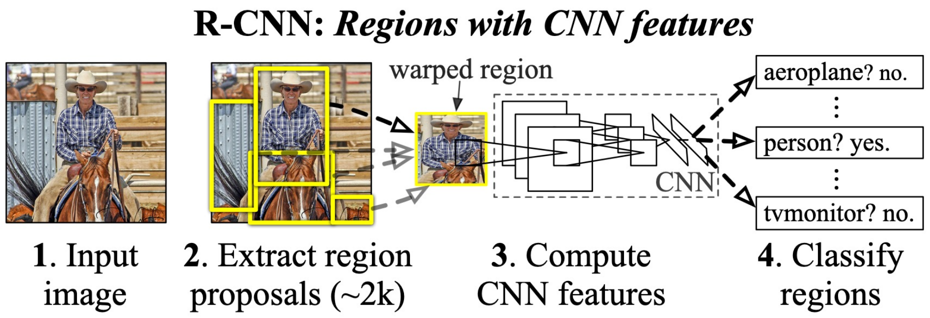
- The final step is **non-maximum suppression,** which is used to eliminate duplicate or overlapping bounding boxes.



Credit: Jain & Nandy (2019)

R-CNNs

Advantages and disadvantages



Credit: Girshick et al. (2014)

ADVANTAGES

- Accurate object detection due to region-based CNN features.
- Can handle objects of different sizes, orientations, and scales, making them suitable for real-world scenarios and complex backgrounds.
- Flexible and can be adapted to various object detection tasks by modifying final layers of the network.

DISADVANTAGES

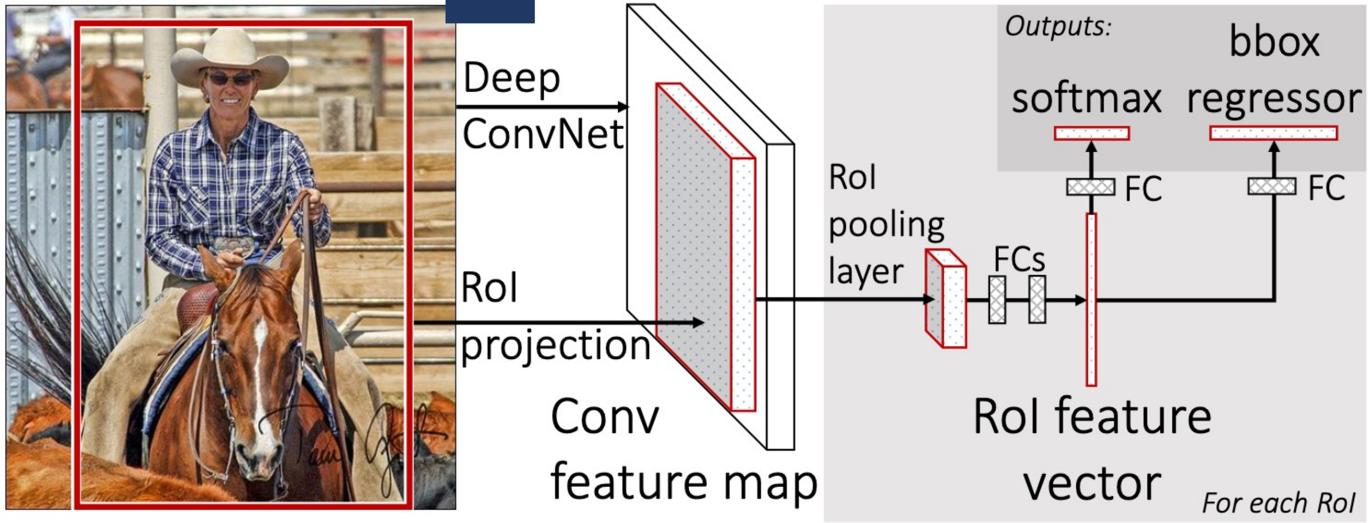
- Require separate components such as selective search and AlexNet.
- Computationally expensive due to multi-stage process.
- Slow inferences make them unsuitable for real-time application.
- Overlapping regions generate redundant computations.

Fast R-CNNs

Improvements

- To overcome the issues of the original R-CNN implementation Girshick (2015) developed Fast R-CNN, where the CNN, SVM and bounding box regressor are combined into one architecture:

1



Credit: Girshick (2015)

Step 1: Fast R-CNN processes the entire image (not multiple RoIs) with several object proposals through Deep ConvNet, a trained model for feature extraction. Sharing the CNN layers makes the process much faster.

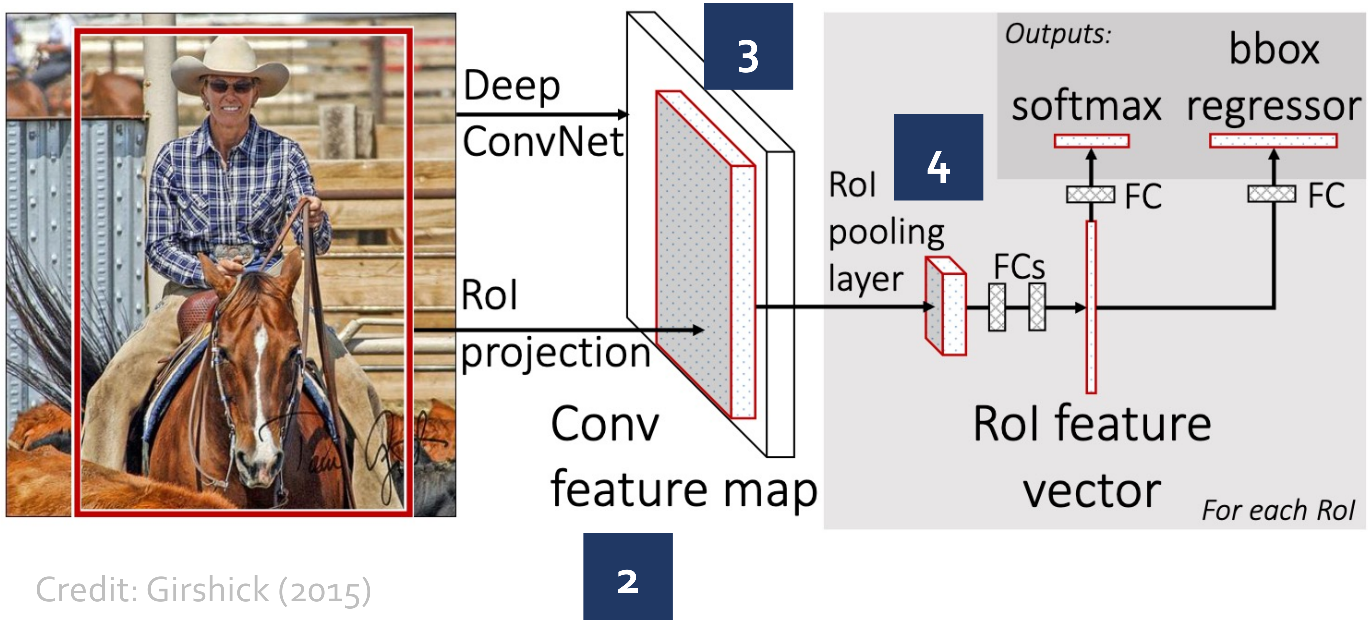
Fast R-CNNs II

Improvements

Step 2: The output is a feature map of the entire image (features are stored relative to where they are detected in the image).

Step 3: We then extract regions of interest (RoIs) from this feature map based on the object proposals.

Step 4: RoIs for all object proposals in the image are downsampled to equal-size features in a RoI pooling layer.



Credit: Girshick (2015)

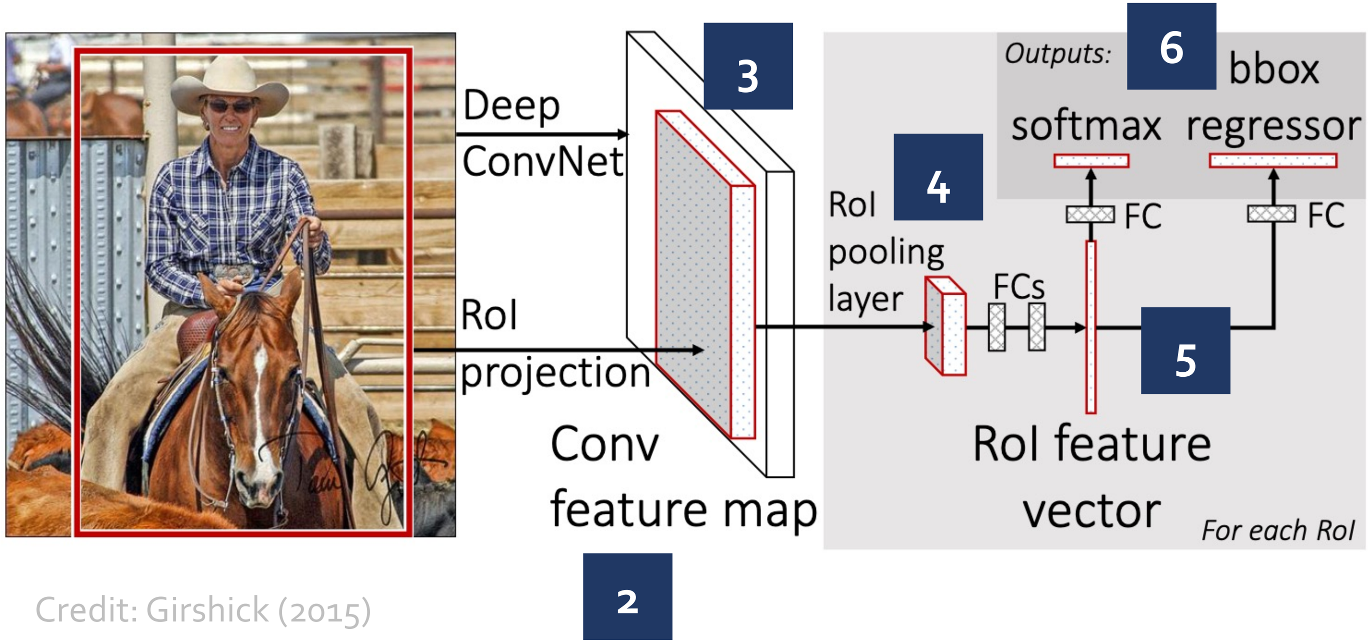
Fast R-CNNs II

Improvements

Step 2: The output is a feature map of the entire image (features are stored relative to where they are detected in the image).

Step 3: We then extract regions of interest (RoIs) from this feature map based on the object proposals.

Step 4: RoIs for all object proposals in the image are downsampled to equal-size features in a RoI pooling layer.



Credit: Girshick (2015)

Step 5: Feature maps are flattened to a 1D RoI feature vector.

Step 6: The feature vector branches into two output layers. One produces a softmax probability estimate over our object classes. The regressor outputs 4 values that define the refined bounding box edges.

Fast R-CNNs II

Improvements

Step 2: The output is a
map of the entire imag
are stored relative to
are detected in th

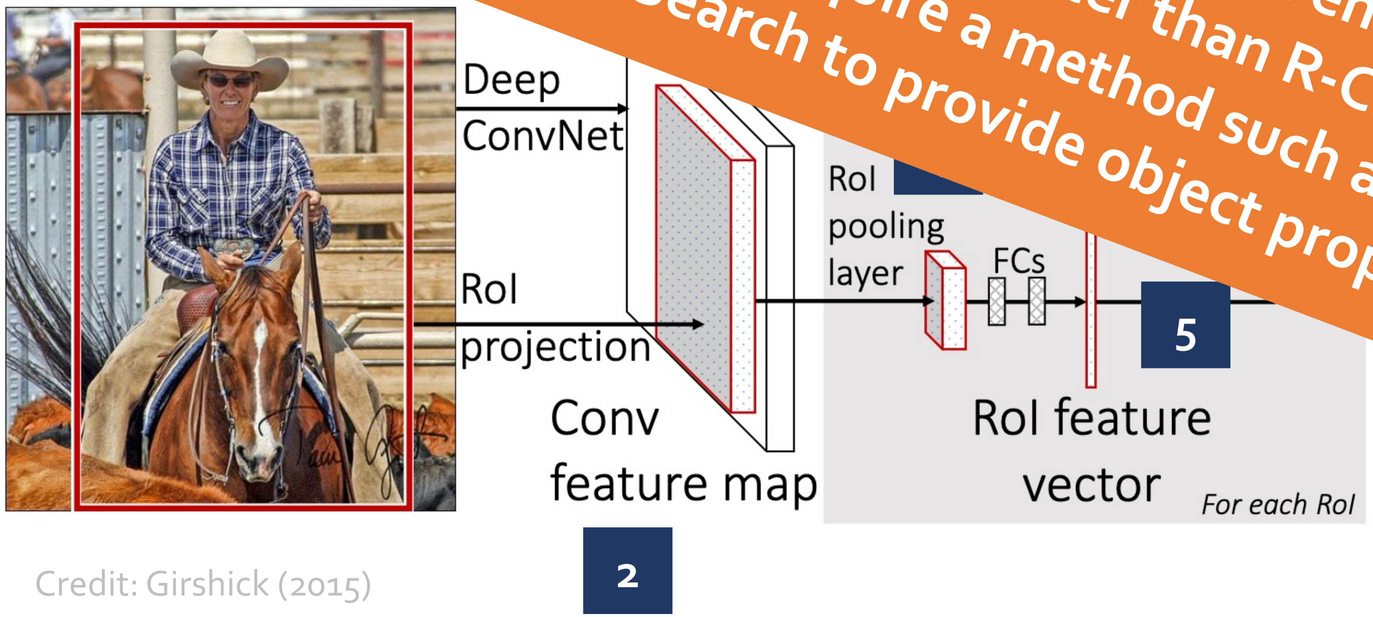
Step 3: We then extract regions of interest (RoIs)
this feature map based
ct proposals.

Step 4: RoIs for all object proposals in the image are downsampled to equal-size features in a RoI pooling layer.

The entire pipeline is trained end-to-end which makes this much faster than R-CNN. However, we still require a method such as Selective Search to provide object proposals.

: Feature maps are flattened
RoI feature vector.

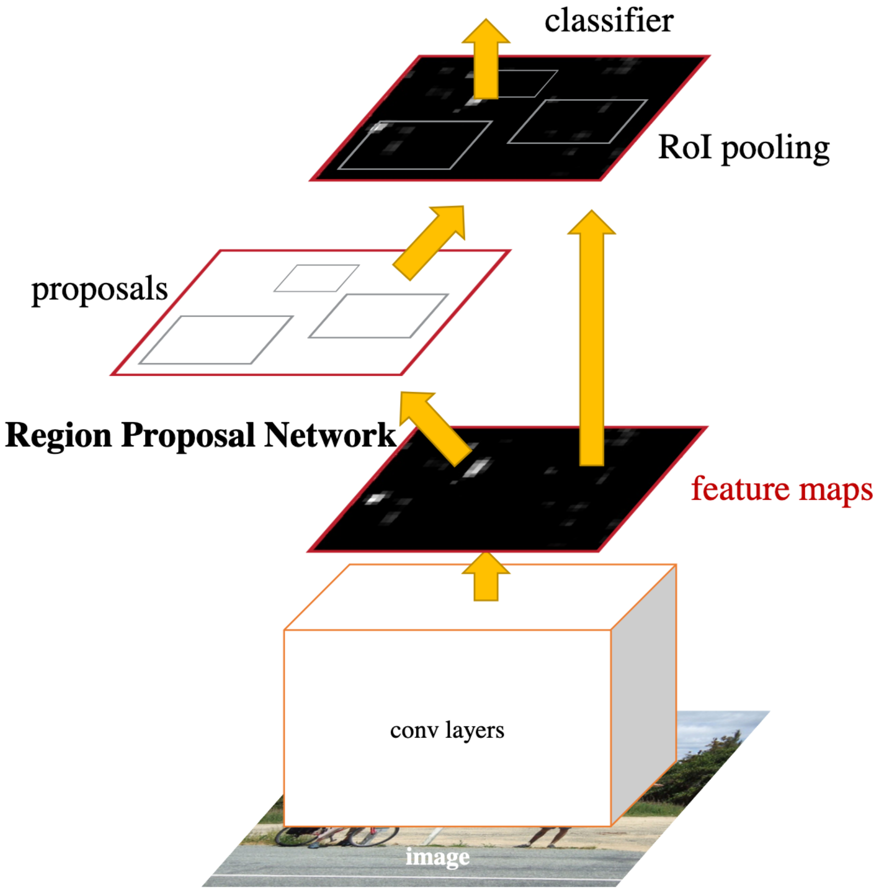
ature vector branches
tput layers. One produ-
ax probability estimate
over
bject classes. The regressor outputs 4 values that define the refined bounding box edges.



Credit: Girshick (2015)

Faster R-CNNs

Further improvements

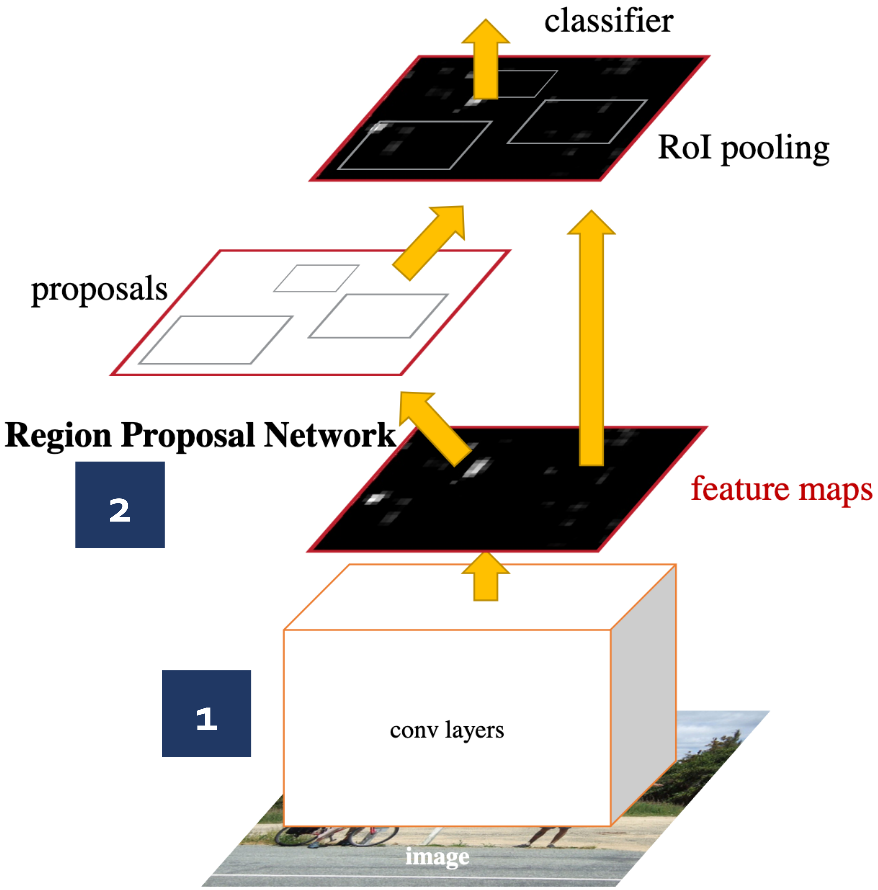


Credit: Ren et al. (2016)

- To overcome the issue of determining object proposals separately, Ren et al. (2016) add a **region proposal network** (RPN) directly into the R-CNN architecture. This CNN generates proposals of various scales and aspect ratios.

Faster R-CNNs

Further improvements



Credit: Ren et al. (2016)

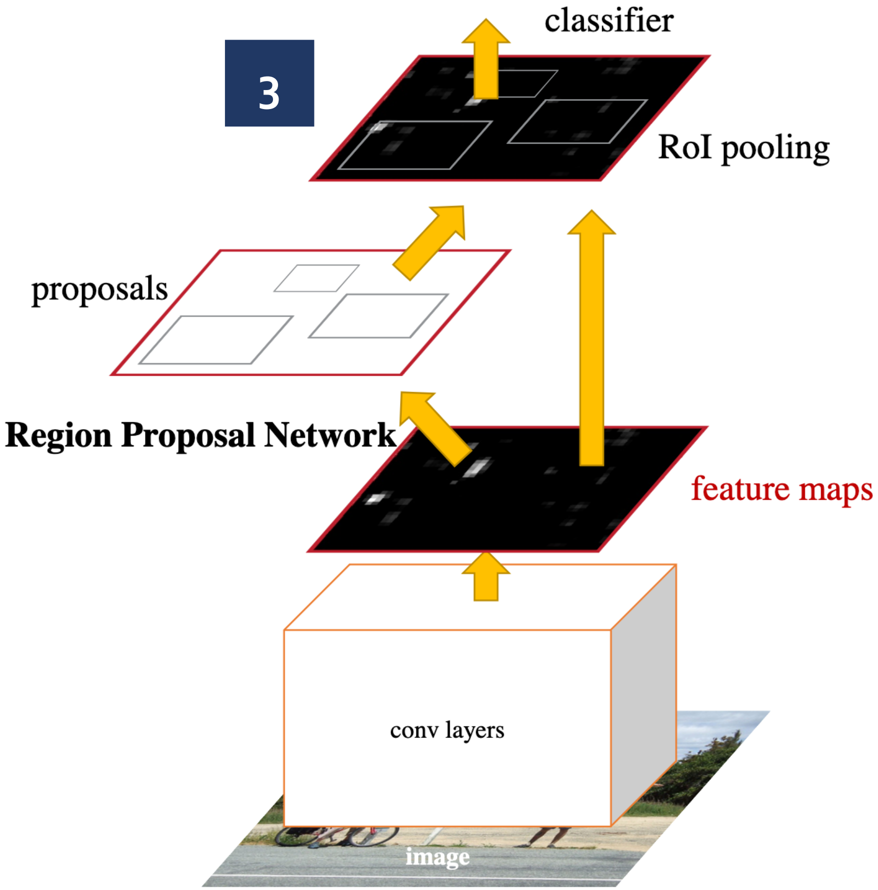
- To overcome the issue of determining object proposals separately, Ren et al. (2016) add a **region proposal network** (RPN) directly into the R-CNN architecture. This CNN generates proposals of various scales and aspect ratios.

Step 1: The input image is fed into a pre-trained CNN that extracts the features of the image and generates feature maps as before.

Step 2: The RPN generates object bounding box proposals by sliding small convolutional layers over the feature maps to produce proposals relative to certain (random) anchor positions.

Faster R-CNNs

Further improvements



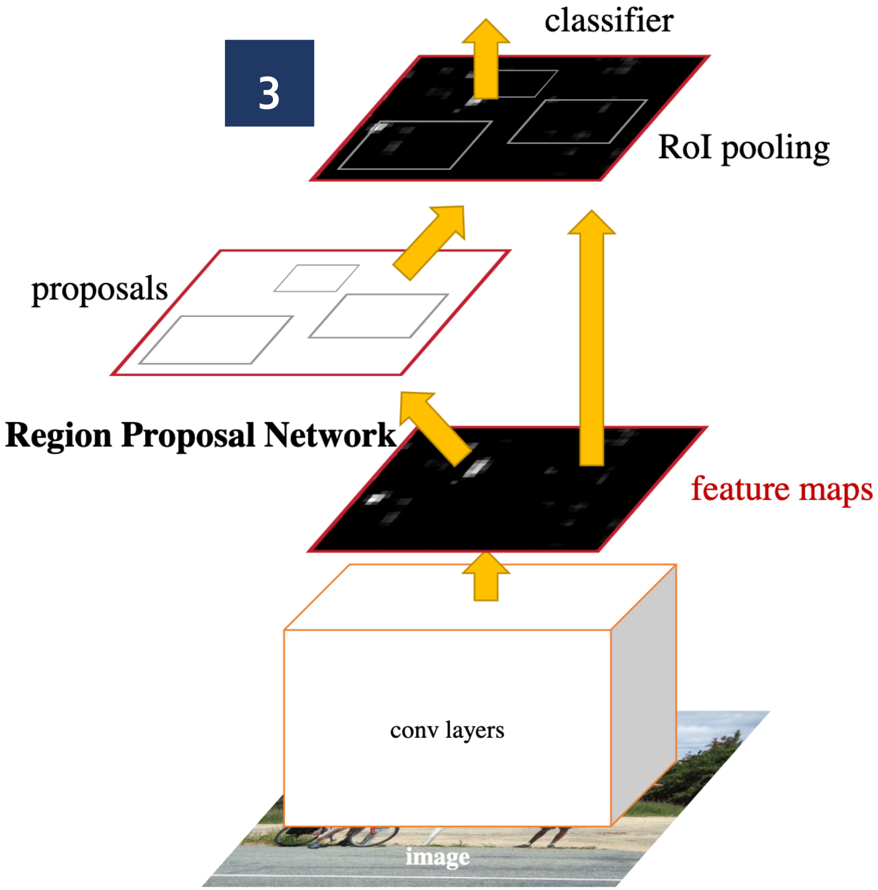
Step 3: Once the proposals are generated, they are fed into the RoI pooling layer (see Fast R-CNN) which is used to transform the RoIs of variable size into a fixed size.

Step 4: Finally, classification and bounding box regression are applied to determine if an object is present within the anchor box and adjust box position & size, respectively.

Credit: Ren et al. (2016)

Faster R-CNNs

Further improvements



Credit: Ren et al. (2016)

Step 3: Once the proposals are generated, they are fed into the RoI pooling layer (see Fast R-CNN) which is used to transform the RoIs of variable size into a fixed size.

Step 4: Finally, classification and bounding box regression are applied to determine if an object is present within the anchor box and adjust box position & size, respectively.

- The CNN RPN (not a fully connected NN) acts as an attention mechanism to generate high quality proposals of interesting feature areas. As this is not based in the input image plane as during Selective Search but feature maps, this approach is much faster.

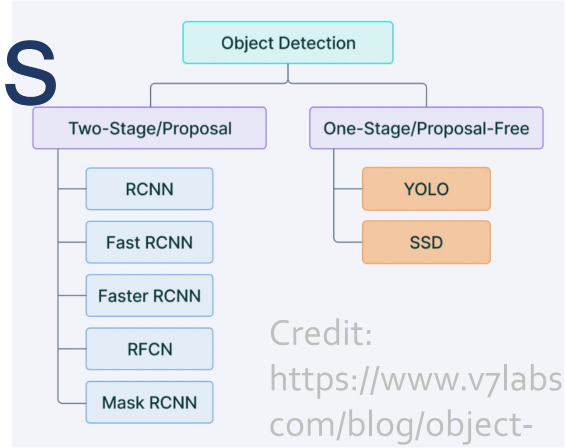
R-CNNs

Other approaches in the family

- Several additional improvements on the Faster R-CNN architecture have been developed in recent years:
  - R-FCN (Dai et al. 2016): Improve object detection by replacing fully connected layers in Faster R-CNN with position sensitive score maps.
  - Mask R-CNN (He et al. 2017): Add a mask prediction for subsequent segmentation onto the Faster R-CNN architecture. This allows object detection and mask prediction at the same time (multi-task learning; see next lecture for details on masks and image segmentation).
  - Cascade R-CNN (Cai & Vasconcelos 2018): To overcome issues with overfitting at training and quality mismatch during the inference, this approach trains a sequence of image detectors for different binary classification criteria between object detected vs absent.

You Only Look Once (YOLO)-CNNs

Main idea from Redmon et al. (2015) I



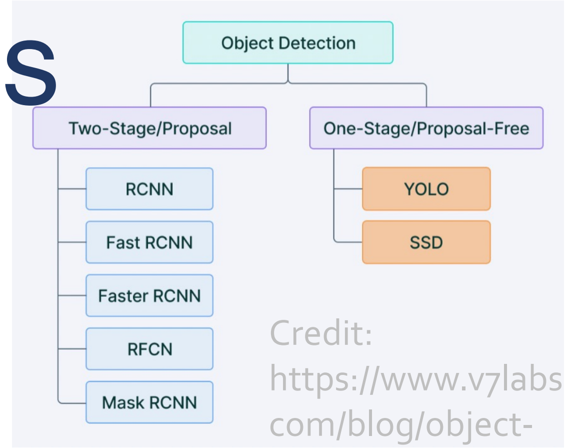
Credit: https://www.v7labs.com/blog/object-detection-guide

- The previous architectures were based on a two-stage process, where we determine proposal regions that are subsequently classified.

- Proposal-free methods do not define RoIs but focus on entire images to determine bounding boxes and class probabilities, which makes them faster than two-stage methods.

You Only Look Once (YOLO)-CNNs

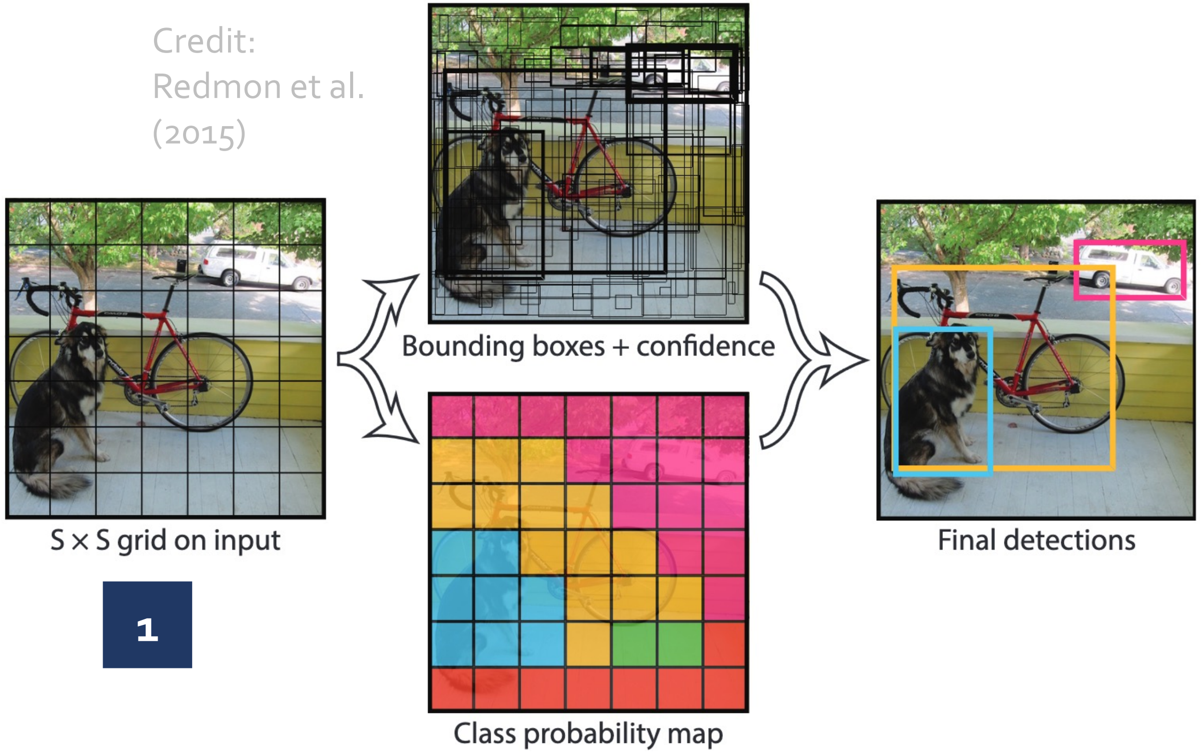
Main idea from Redmon et al. (2015) I



Credit: https://www.v7labs.com/blog/object-detection-guide

- The previous architectures were based on a two-stage process, where we determine proposal regions that are subsequently classified.

Credit: Redmon et al. (2015)



1

- Proposal-free methods do not define RoIs but focus on entire images to determine bounding boxes and class probabilities, which makes them faster than two-stage methods.

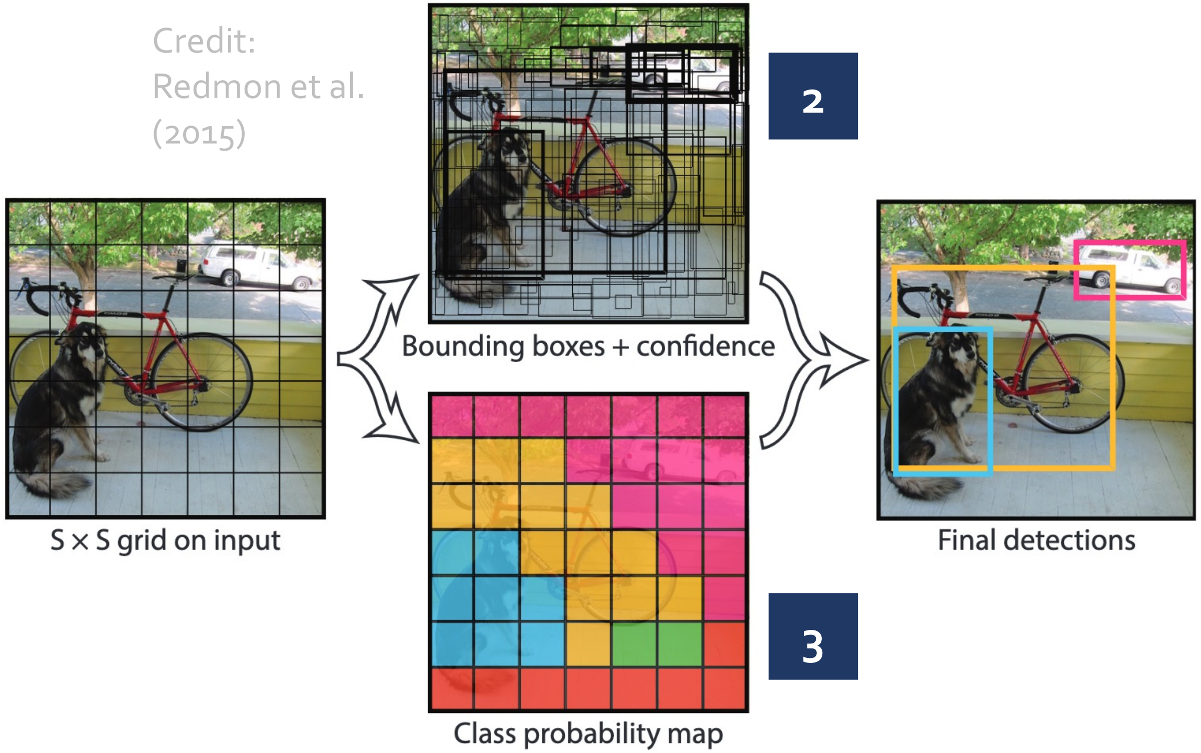
Step 1: Images are resized to 448 x 448 pixel. The model then splits the input into S x S grid cells.

You Only Look Once (YOLO)-CNNs

Main idea from Redmon et al. (2015) II

Step 2: Using a regression approach, each cell is responsible for predicting bounding boxes (centre coordinates plus height and width) and a confidence score for the object being present.

Step 3: Each cell also predicts a predefined number of N conditional class probabilities. Note there is a single class probability even if the cell contains multiple bounding boxes.



Credit: Redmon et al. (2015)

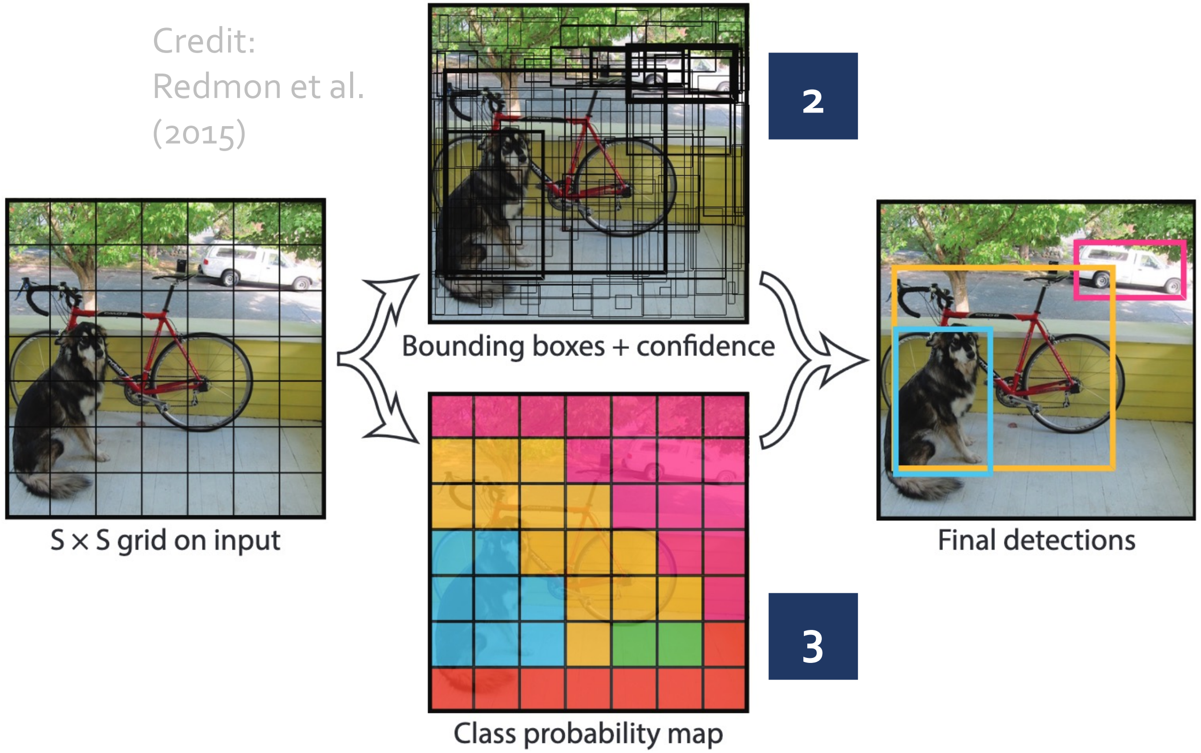
You Only Look Once (YOLO)-CNNs

Main idea from Redmon et al. (2015) II

Step 2: Using a regression approach, each cell is responsible for predicting bounding boxes (centre coordinates plus height and width) and a confidence score for the object being present.

Step 3: Each cell also predicts a predefined number of N conditional class probabilities. Note there is a single class probability even if the cell contains multiple bounding boxes.

Credit: Redmon et al. (2015)



2

3

4

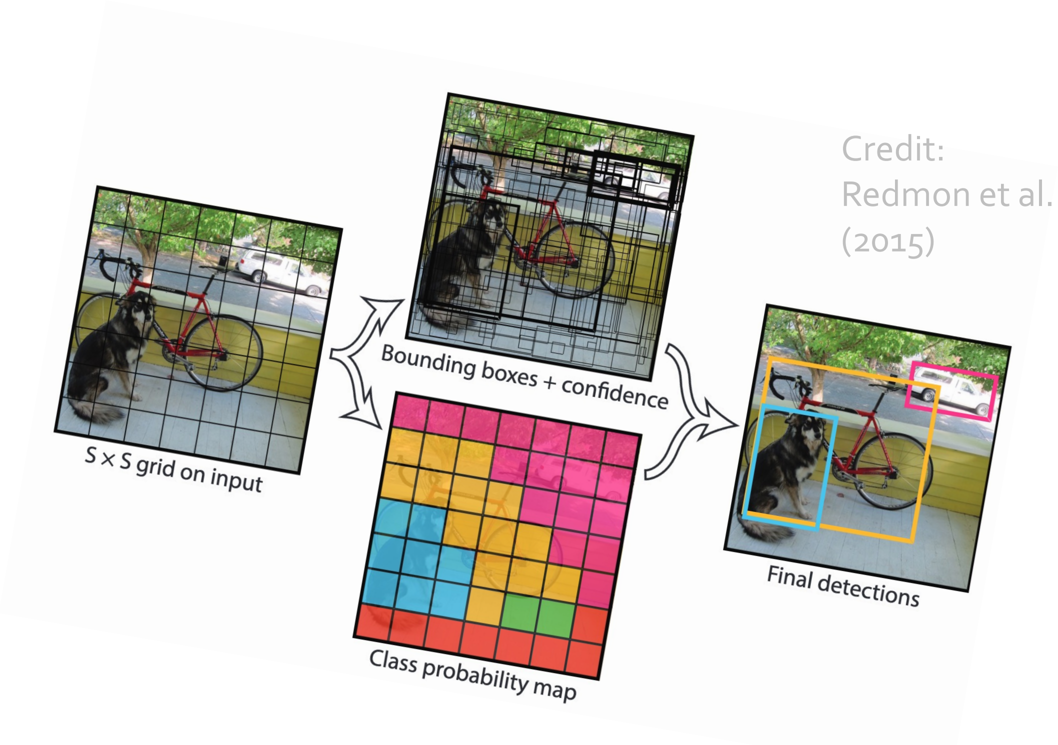
Step 4: Select only the most relevant boxes and then pick the one with the highest class probability to determine the final detections.

YOLO-CNNs

Advantages and disadvantages

ADVANTAGES

- End-to-end pipeline makes it easy to implement and train.
- Fast due to use of single CNN for regression (only a single forward pass to detect objects needed) enables real-time use.
- Fewer issues with misclassifying background patches as objects.
- Good generalisation capability across different domains making them robust.



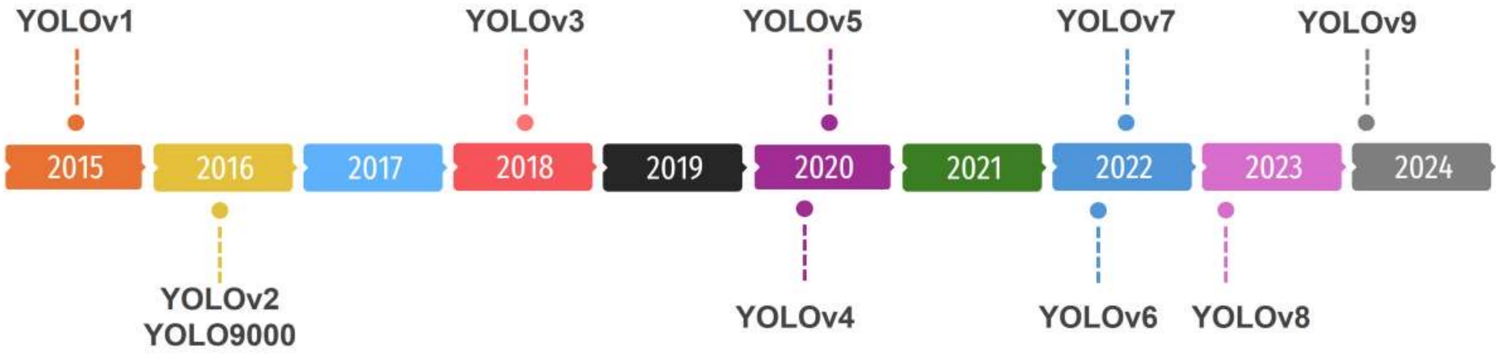
Credit: Redmon et al. (2015)

DISADVANTAGES

- Less accurate than two-stage methods due to single staging.
- Grid architecture makes it possible for the detector to miss objects smaller than the grid spacing.

YOLO-CNNs

Different versions



Credit: https://www.ikomia.ai/blog/what-is-yolo-introduction-object-detection-computer-vision

- The initial YOLO architecture has been updated repeatedly since 2015 with numerous advances. These have included:
  - v2 (YOLO9000): faster and more accurate than v1 able to detect a wider range of classes; uses a different CNN backend (Darknet) and different loss function.
  - v4 (Bochkovskiy et al. 2020): further improvements on speed and accuracy and accessible to a wider range of devices and different platforms
  - v5 (not published as a paper but released by Ultralytics): different architecture based on PyTorch not Darknet with better scalability and performance
  - v9 (Wang et al. 2024): current state-of-the-art in real-time object detection with several new features that increased performance even further

Question time

Mentimeter quiz

Let’s take a few minutes to recap what we have discussed so far:

Go to menti.com and enter the code 8217 746.

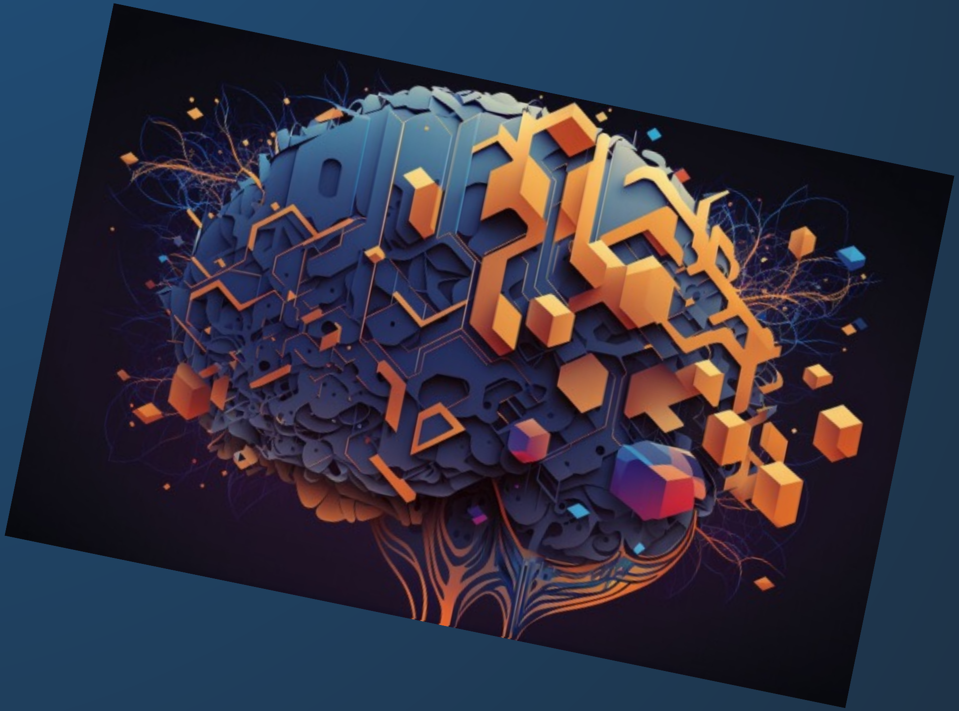
Object detection overview

Object detection with NNs

The concept of attention

Vision transformers

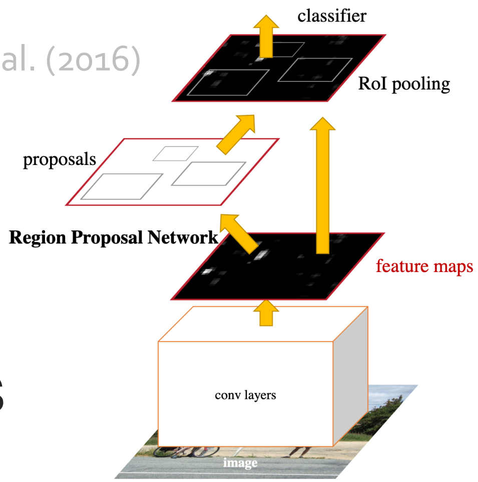
Summary



Attention

Human brains I

Credit: Ren et al. (2016)



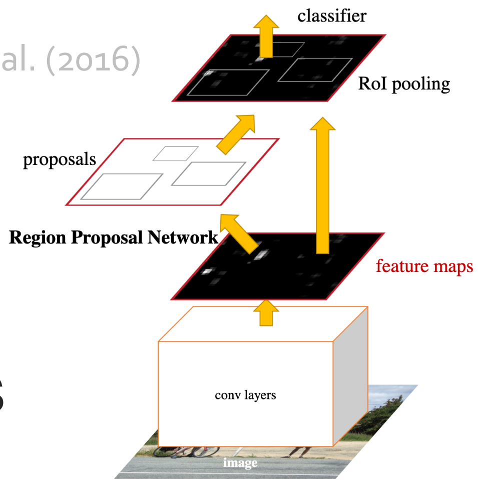
- When introducing Faster R-CNN, we discussed that the Region Proposal Network gives **attention** to relevant areas in the feature maps for subsequent classification.

- Let's talk about what attention means in the context of **human brains**.

Attention

Human brains I

Credit: Ren et al. (2016)



- When introducing Faster R-CNN, we discussed that the Region Proposal Network gives **attention** to relevant areas in the feature maps for subsequent classification.

- Let’s talk about what attention means in the context of **human brains**.

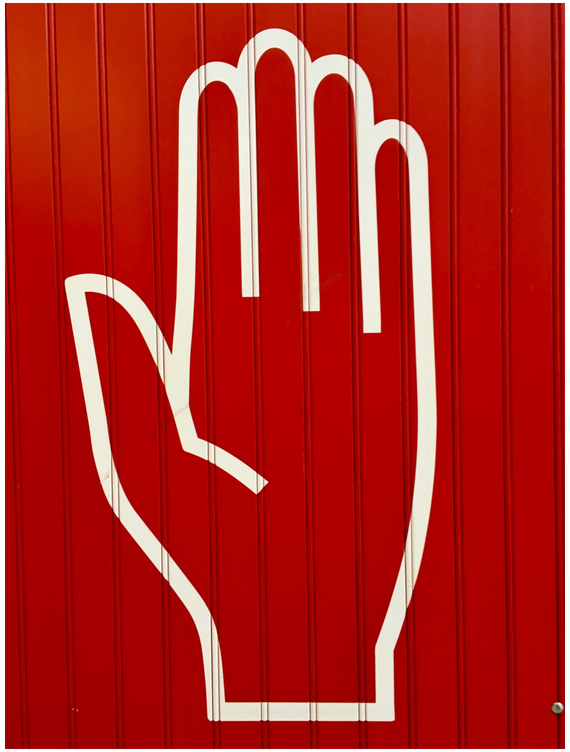
- Our brains have **limited memory capacity**, so our brains have to decide which information to store. This is where attention comes in.

When we look at images, the movements of our eyes can reveal the most relevant image parts that our attention is attracted to.

Attention

Human brains I

- While deeper neuronal layers extract specialised high-level information, the first neurons in the “processing pipeline” react to **simple visual attributes** like intensity, contrast, colour, orientation, direction.



- These features automatically attract attention, and we refer to them as *visually salient*. Directing attention to other image attributes requires our brains “effort”, but both types of processing typically operate in parallel.

While salient features can differ from person to person, human brains are generally attracted to similar salient visual structures.

Attention

Applications in ML

- In ML, we use attention in a similar way to the human brain. That is, we want to **dynamically highlight and use salient parts of our input**.
- Attention-based systems contain **three components**:
  - **Reading process:** Read in raw data (e.g., words in a sentence) and convert them to a distributed representation (e.g., words become feature vectors).
  - **Storage:** These feature vectors can be seen as “memories”, containing a sequence of information. Individual information pieces can be retrieved later, but not necessarily in the stored order.
  - **Application:** Take advantage of the memories to sequentially perform certain tasks with the ability to pay attention to single (or a few) memory elements.

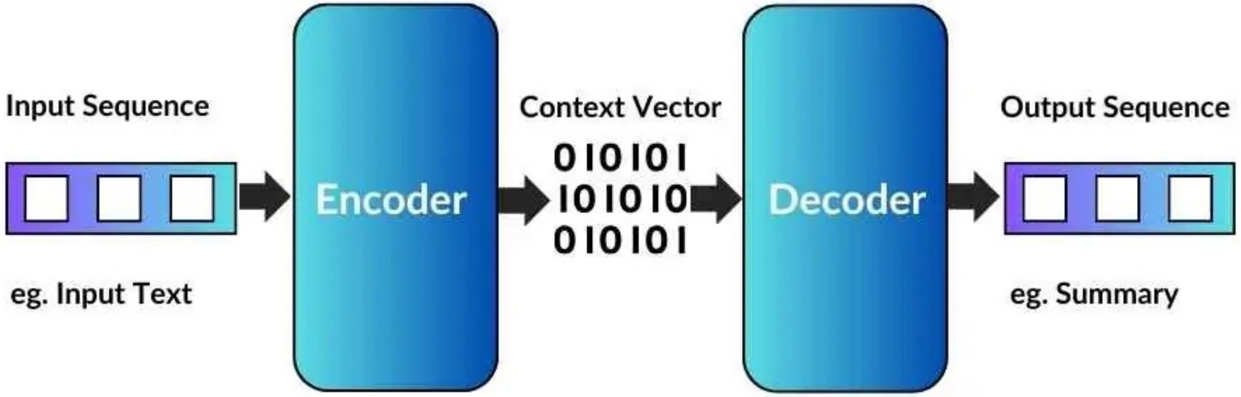
Attention

Role in Natural Language Processing

- To understand why this is important, let's look into **Natural Language Processing** (NLP), where we want to **translate a sentence**.

- To understand this input sequence of sentence words and summarise its meaning into a compact representation, an **encoder** reads in each word, analyses its relationship to other words and captures that the resulting information into a fixed-length **summary or context vector**.

- The **decoder** then takes the summary vector to generate a translation by considering the vector itself plus its own "vocabulary".



Credit: https://spotintelligence.com/2023/09/28/sequence-to-sequence/

Attention

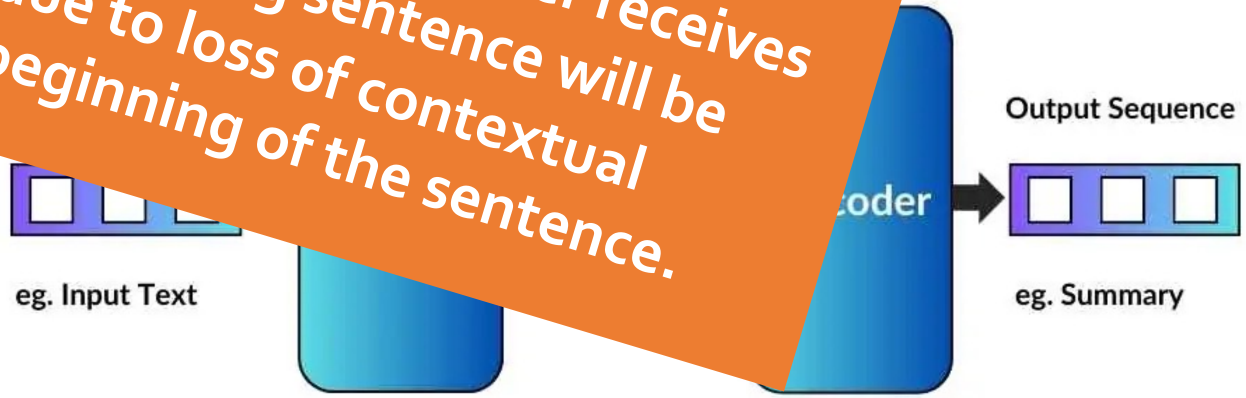
Role in Natural Language Processing

- To understa is important, let's look into **Natural Language Processin** ant to **translate a sentence**.

- To unders words and summarise its meani reads in each word, ar at the resultin **vector**.

- The **decoder** then tak summary vector to generate a translation by considering the vector itself plus its own "vocabulary".

Since the encoder generates fixed-length vectors irrespective of input length or complexity (no highlighting of salient features!) the decoder receives limited information. E.g., a long sentence will be harder to translate due to loss of contextual information from the beginning of the sentence.



Credit: https://spotintelligence.com/2023/09/28/sequence-to-sequence/

Self-attention

Transformers

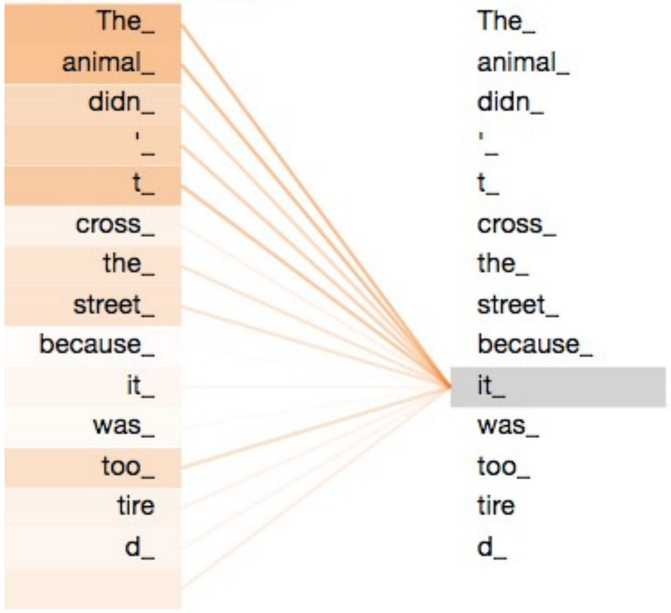
- *Attention* addresses this issue in the sequence-to-sequence picture by adding an additional step that determines the significance of sequence elements (Bahdanau et al. 2014). Attention scores encode how much attention each element deserves based on its relevance for decoding.

Self-attention

Transformers

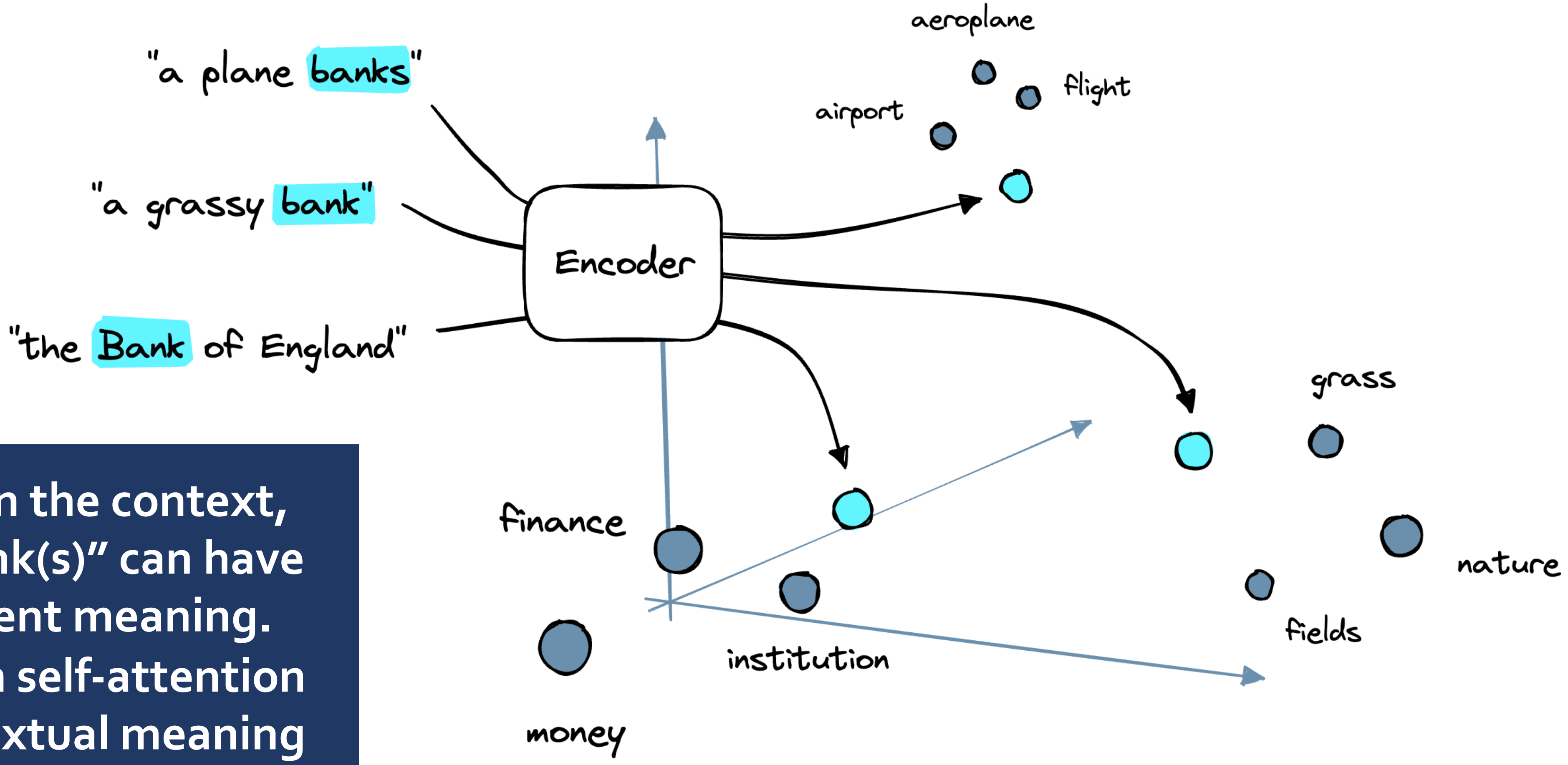
- *Attention* addresses this issue in the sequence-to-sequence picture by adding an additional step that determines the significance of sequence elements (Bahdanau et al. 2014). Attention scores encode how much attention each element deserves based on its relevance for decoding.
- Vaswani et al. (2017) developed this concept further and introduced **self-attention** in their seminal paper "Attention is all you need" which introduces so-called **transformers**.

In self-attention each word in a sentence is compared to all other words including itself. Each word embedding is then reweighted to include this contextual relevance, so that the output has context.



Self-attention

Another example



Depending on the context, the word "bank(s)" can have a very different meaning. Encoders with self-attention can add contextual meaning to embeddings.

Credit: https://www.pinecone.io/learn/series/image-search/vision-transformers/

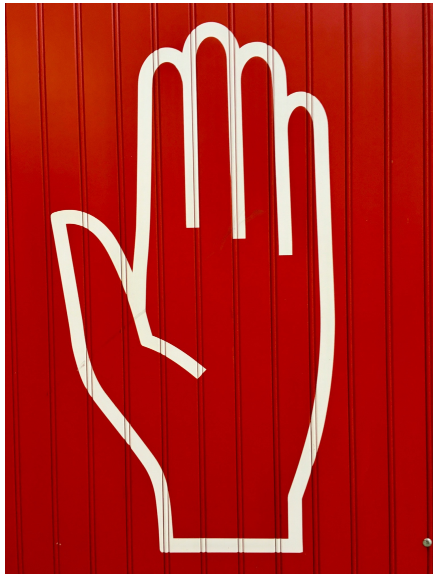
Attention

Computer vision

The attention mechanism was originally introduced to analyse word sequences and improve machine translation. While developed for sequential/temporal data, attention can also help in processing static information such as image data.

- In computer vision, the attention mechanism is used to **dynamically adjust weights** of our neural network using a function g(x) for an input feature map x that is superimposed between the convolutional layers.

- Aim is to provide information about which features should be paid attention to to the next NN layers.



Attention

Computer vision

Th ttention mechanism was originally introduced to analyse
nces and improve machine translation. While
d l/temporal data, attention can also help
tion such as image data.

As transformers replaced (recurrent) neural networks, we should ask if we can do something similar in computer vision and replace CNNs with something else?

- In computer visio
to **dynamically adjust weig**
using a function g(x) for an input fe
is superimposed between the convolutiona
- Aim is to provide information about which features should be paid attention to to the next NN layers.

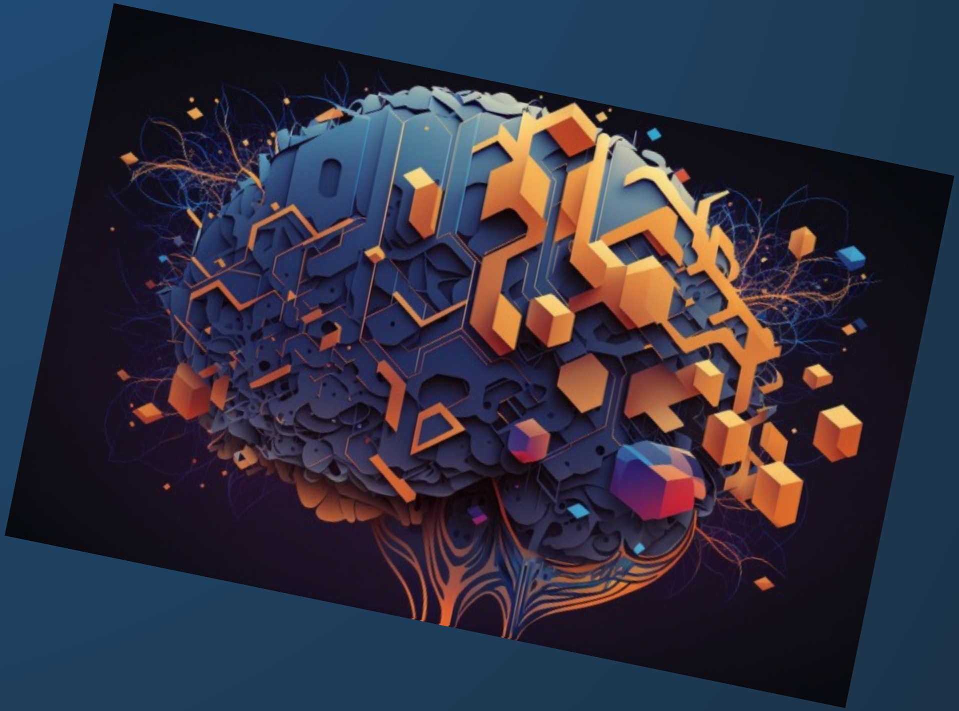
Object detection overview

Object detection with NNs

The concept of attention

Vision transformers

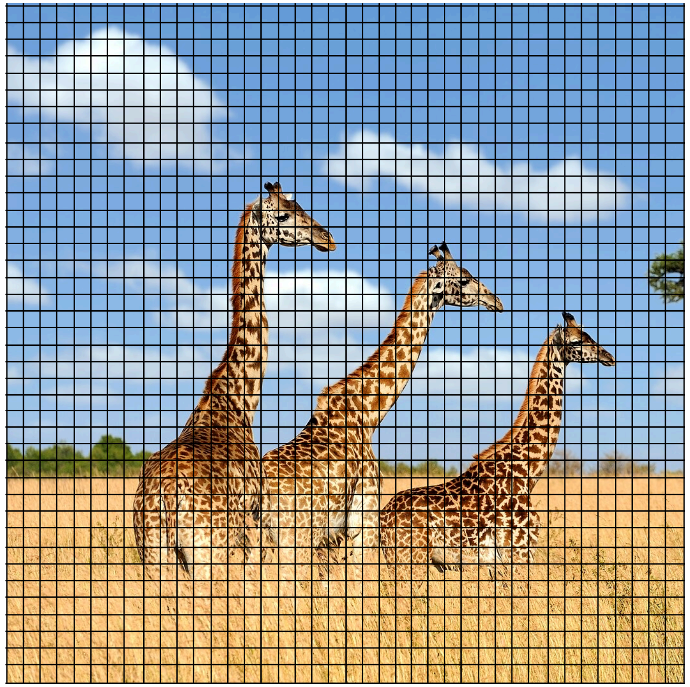
Summary



Vision transformers

Main challenge

- The idea of **vision transformers** (ViT) was introduced in 2020 by Dosovitskiy et al. in a paper titled “An image is worth 16x16 words: Transformers for image recognition at scale”.

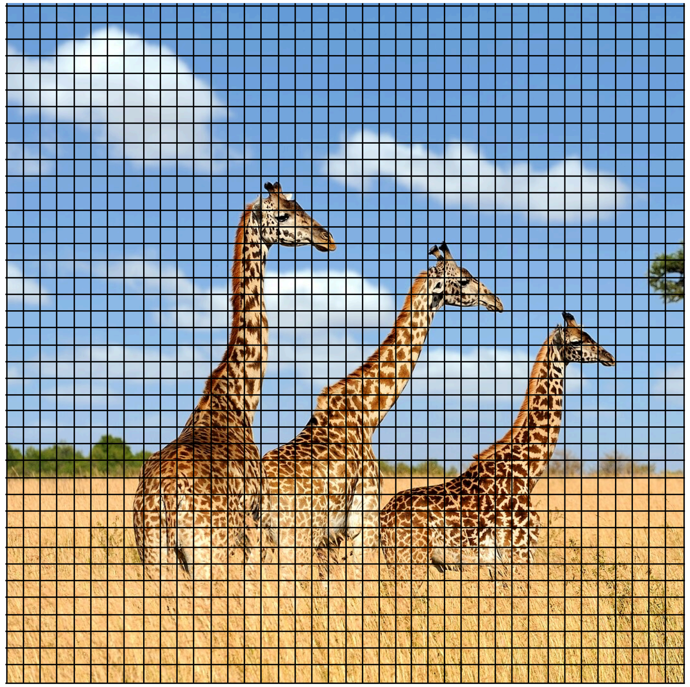


- Thinking about the self-attention mechanism again, we need to ask “Can each pixel pay attention to all the other pixel in an image?”

Vision transformers

Main challenge

- The idea of **vision transformers** (ViT) was introduced in 2020 by Dosovitskiy et al. in a paper titled “An image is worth 16x16 words: Transformers for image recognition at scale”.

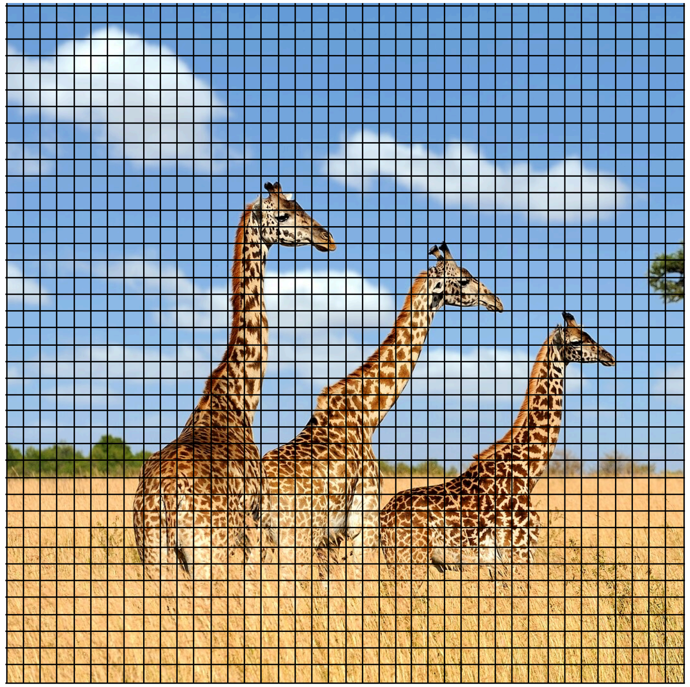


- Thinking about the self-attention mechanism again, we need to ask “Can each pixel pay attention to all the other pixel in an image?”

For an image with 100 x 100 resolution, this would require $(100 \times 100)^2 = 10^8$ interactions, which is way too large to handle computationally. In NLP, sequences typically have a length of a few hundred.

Vision transformers

Solution(s)

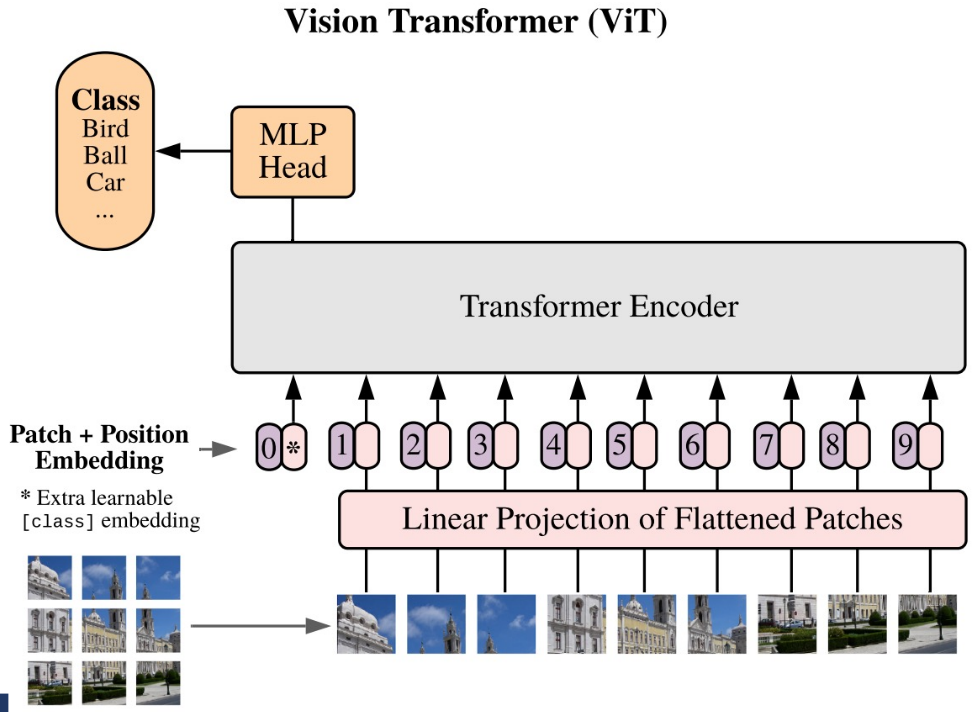


- There are **two avenues** to overcome this issue of computational tractability:
  - We could use convolutional layers to produce a smaller feature map and then apply the self-attention mechanism. However, that requires a significant reduction in dimensionality (and potential loss of information) and still relies on the use of CNNs.
  - Alternatively, we could try and find an approach to using the self-attention mechanisms all the way. The (big!) question that remains is “How could this work?”.

Vision transformers

Algorithm by Dosovitskiy et al. (2020)

- To take advantage of existing transformer architectures, they followed the approach by Vaswani et al. as closely as possible.


Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Step 1: To deal with an image, they divide the input into patches, where each patch has a resolution of P x P (P=16) pixel. Each patch can be considered similar to a word (also referred to as a token) in NLP.

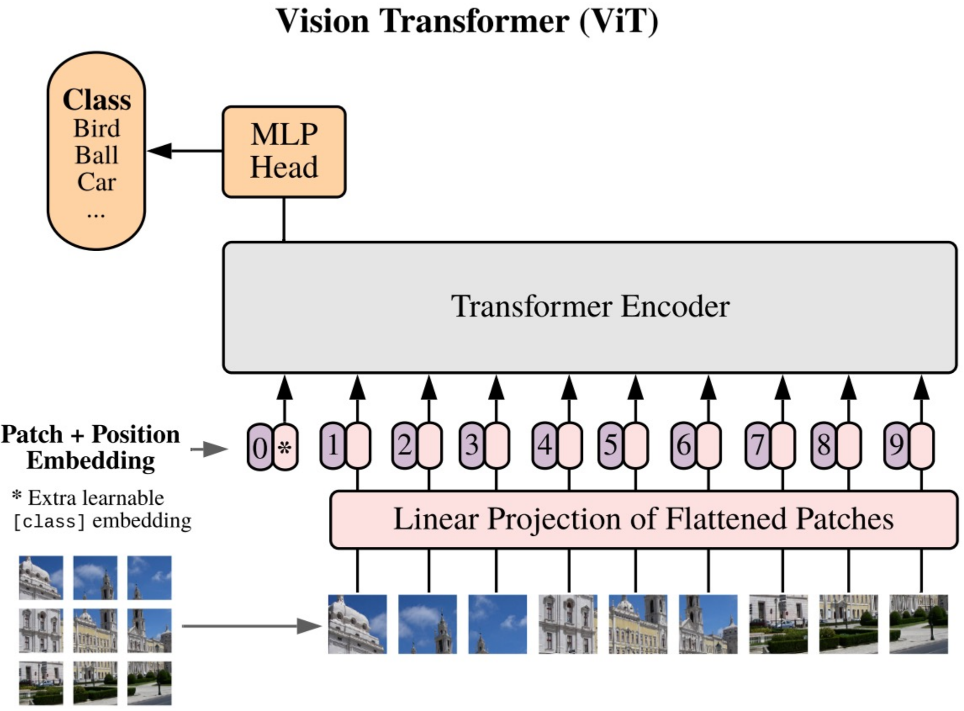
1

Credit: Dosovitskiy et al. (2020)

Vision transformers

Algorithm by Dosovitskiy et al. (2020)

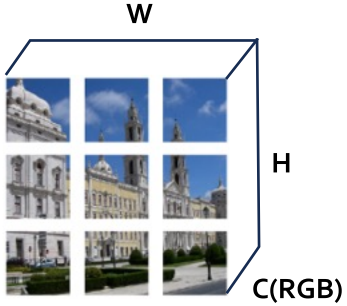
- To take advantage of existing transformer architectures, they followed the approach by Vaswani et al. as closely as possible.


Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 *
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

1

Credit: Dosovitskiy et al. (2020)

Step 1: To deal with an image, they divide the input into patches, where each patch has a resolution of P x P (P=16) pixel. Each patch can be considered similar to a word (also referred to as a token) in NLP.



N=9 number of patches

Image input dimensions: $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$

Dimension of 2D flattened patches: $\mathbf{x}_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$

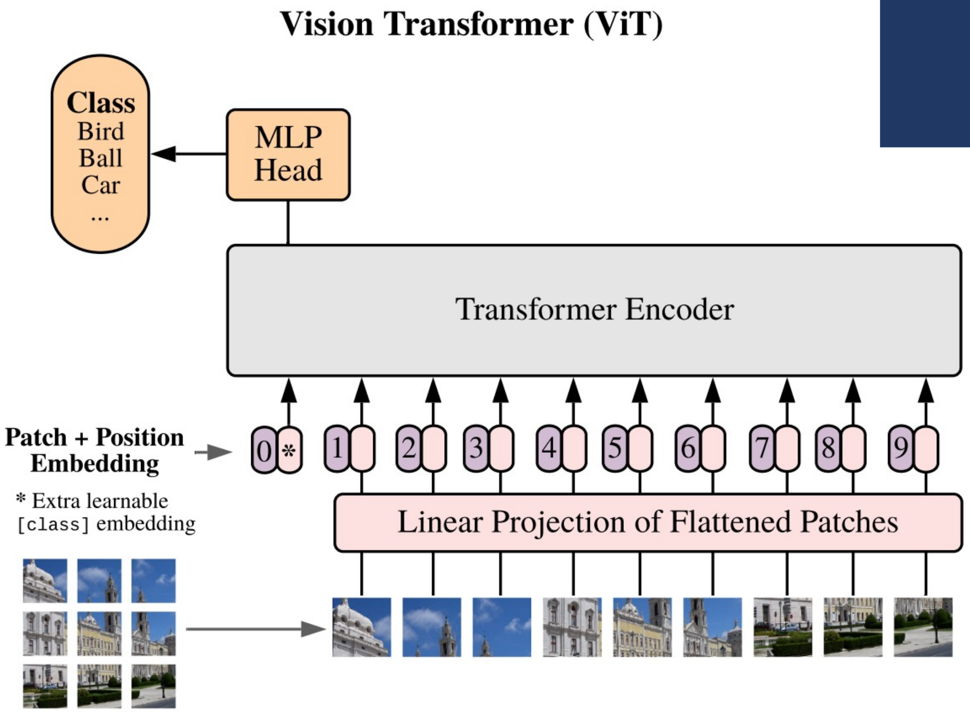
Vision transformers

Algorithm by Dosovitskiy et al. (2020)

Dimension of 2D flattened patches:

$$\mathbf{x}_p \in \mathbb{R}^{N\times(P^2\cdot C)}$$

Step 2: The resulting sequence of image patches has a dimension of N*(P²*C)=9*(16*16*9)=9*768. This is still too large for the transformer encoder. We can compress each patch to a chosen dimension D by multiplying it with an embedding matrix E (a trainable quantity).



Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

2

Credit: Dosovitskiy et al. (2020)

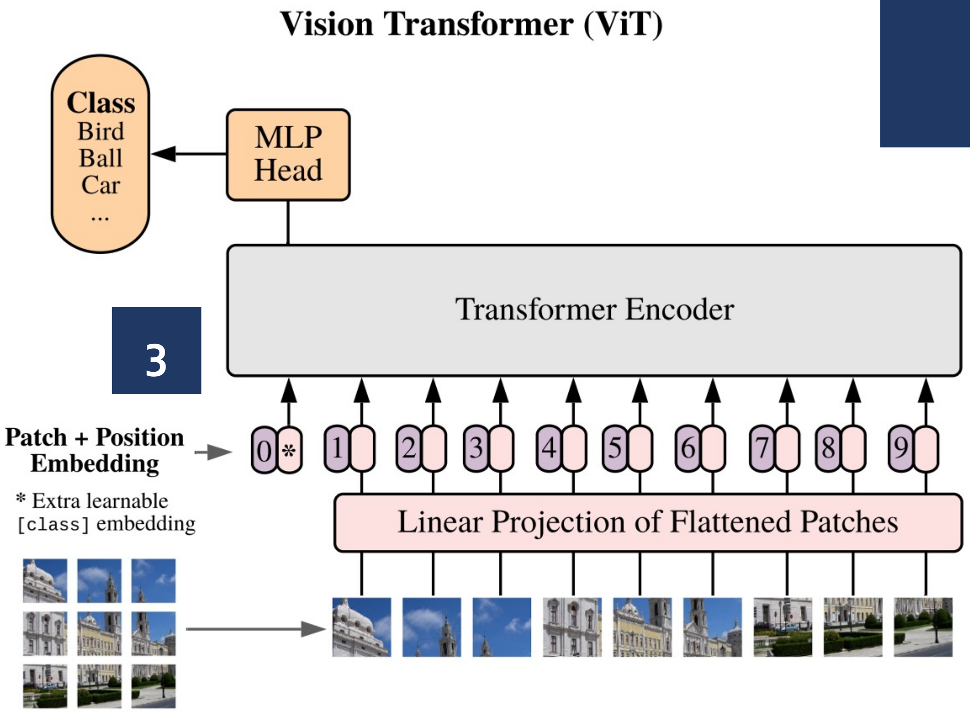
Vision transformers

Algorithm by Dosovitskiy et al. (2020)

Dimension of 2D flattened patches:

$$\mathbf{x}_p \in \mathbb{R}^{N\times(P^2\cdot C)}$$

Step 2: The resulting sequence of image patches has a dimension of $N*(P^2*C)=9*(16*16*9)=9*768$. This is still too large for the transformer encoder. We can compress each patch to a chosen dimension D by multiplying it with an embedding matrix E (a trainable quantity).


Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
3
Patch + Position Embedding
* Extra learnable [class] embedding
0 *
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

2

Step 3: The attention operation does not care about the location of a certain patch in the sequence. However, this matters for image processing. So, we need to add information on the patch position via an in additional "label".

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}$$

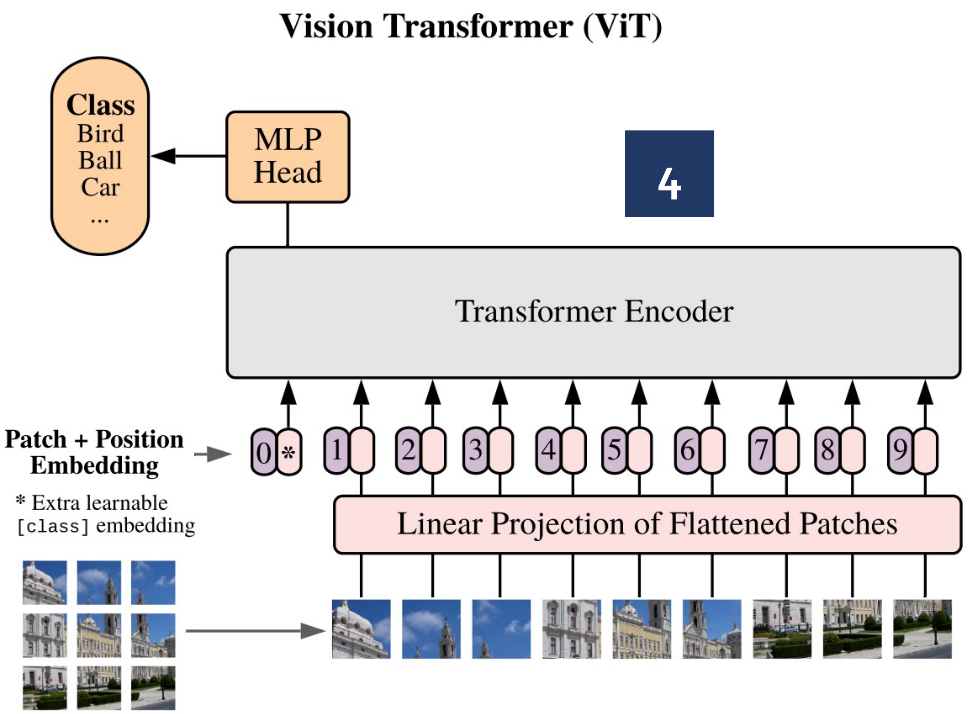
Extra learnable (blank) class embedding

Credit: Dosovitskiy et al. (2020)

Vision transformers

Algorithm by Dosovitskiy et al. (2020)

Step 4: At this stage, the resulting sequence of vectors is fed into the standard Transformer Encoder architecture.



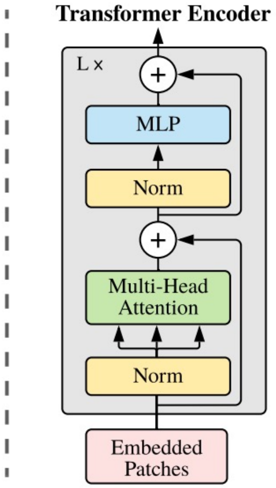
Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
4
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Credit: Dosovitskiy et al. (2020)

Vision transformers

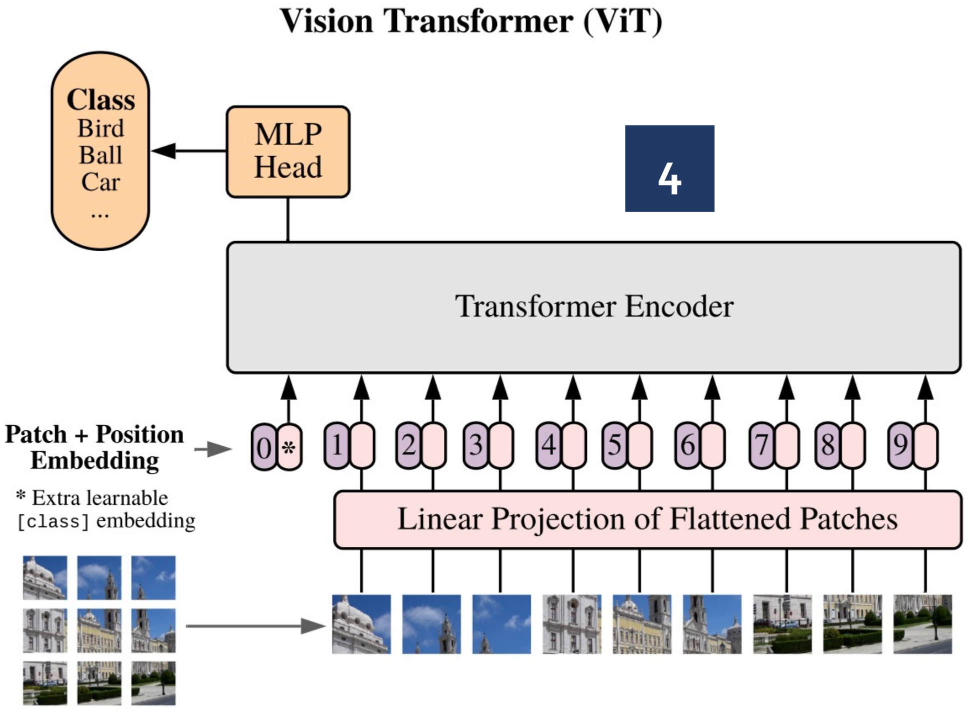
Algorithm by Dosovitskiy et al. (2020)

Step 4: At this stage, the resulting sequence of vectors is fed into the standard Transformer Encoder architecture.



5

Credit: Dosovitskiy et al. (2020)



Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
0
*
1
2
3
4
5
6
7
8
9
* Extra learnable [class] embedding
Linear Projection of Flattened Patches

4

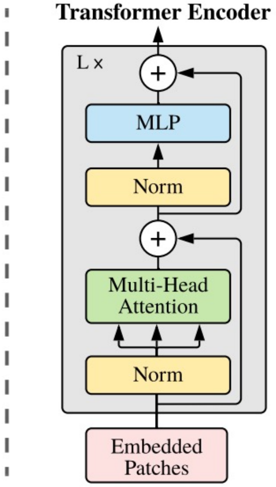
Credit: Dosovitskiy et al. (2020)

Step 5: The transformer encoder involves repeated iterations of normalisation, Multi-Head Attention (weighted sum of sequence elements with attention weights) and multi-layer perceptrons (fully connected neural network with a non-linear activation function).

Vision transformers

Algorithm by Dosovitskiy et al. (2020)

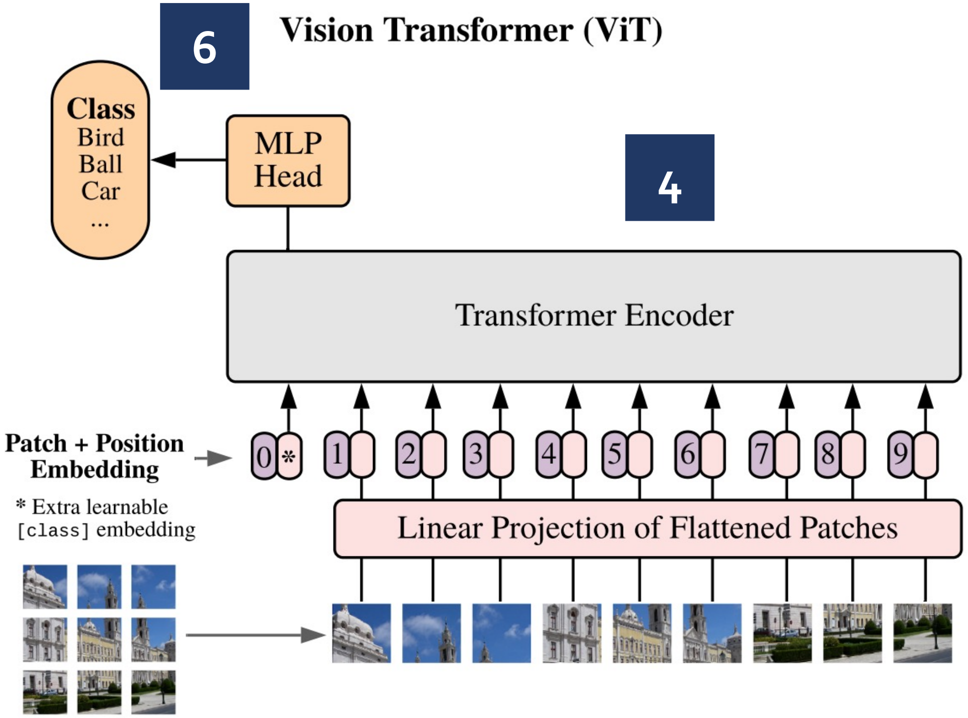
Step 4: At this stage, the resulting sequence of vectors is fed into the standard Transformer Encoder architecture.



5

Credit: Dosovitskiy et al. (2020)

6



Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

4

Credit: Dosovitskiy et al. (2020)

Step 5: The transformer encoder involves repeated iterations of normalisation, Multi-Head Attention (weighted sum of sequence elements with attention weights) and multi-layer perceptrons (fully connected neural network with a non-linear activation function).

Step 6: The encoder outputs sequence of vector representations. To use these further, e.g., for classification, a head (shallow MLP) is added to output a probability distribution over classes.

Vision transformers

Attention maps: an example

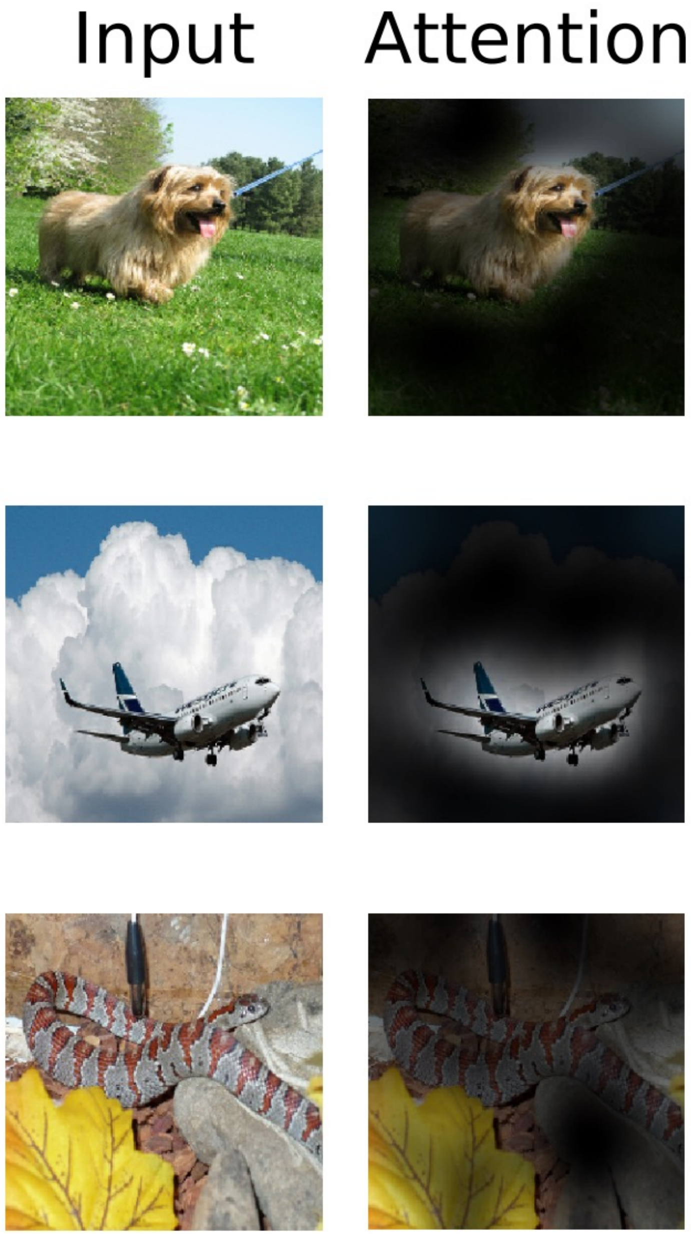
Similar to NLP where representation vectors flow through a transformer and learn more and more semantic relationships between words, the attention mechanism in a ViT repeatedly transforms representation vectors of image patches, incorporating more and more semantic relations between image patches.

Vision transformers

Attention maps: an example

Similar to NLP where representation vectors flow through a transformer and learn more and more semantic relationships between words, the attention mechanism in a ViT repeatedly transforms representation vectors of image patches, incorporating more and more semantic relations between image patches.

- To understand how ViTs integrate information across the image, we can compute **attention maps** by multiplying the weight matrices of all layers.



Credit: Dosovitskiy et al. (2020)

Vision transformers

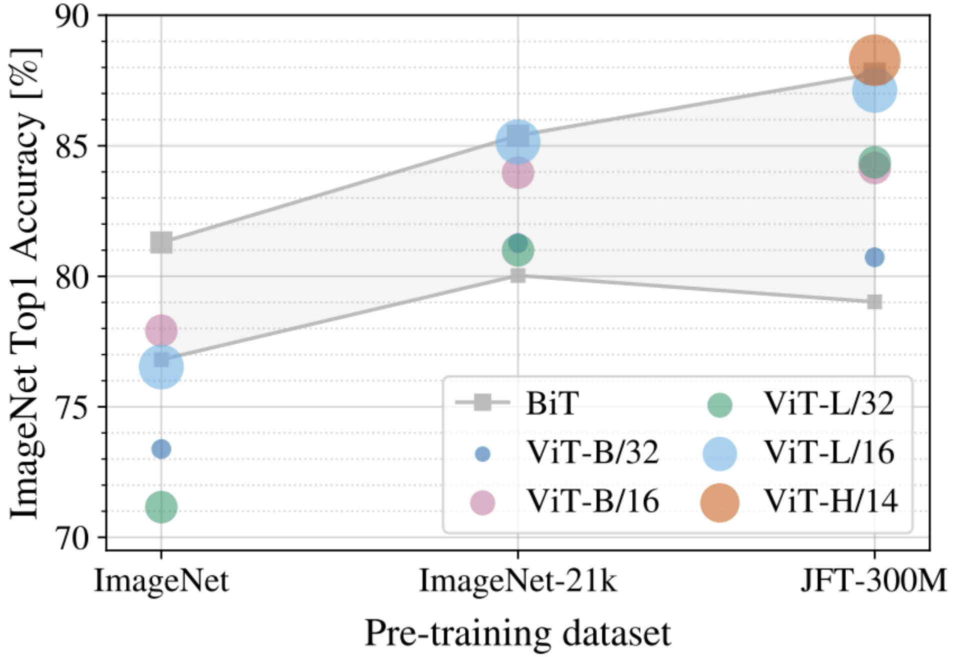
A success story?

- Vision transformers marks first time that a NN algorithm has been successfully applied to both computer vision and NLP.

By implementing different ViT training experiments, Dosovitskiy et al. showed that a CNN approach to image classification is generally superior to ViT when trained on a small dataset. However, as the dataset size increases, ViT begin to outperform CNNs.

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.



BiT (Big Transfer): supervised transfer learning model with CNN-based ResNet

Credit: Dosovitskiy et al. (2020)

Vision transformers

A success story?

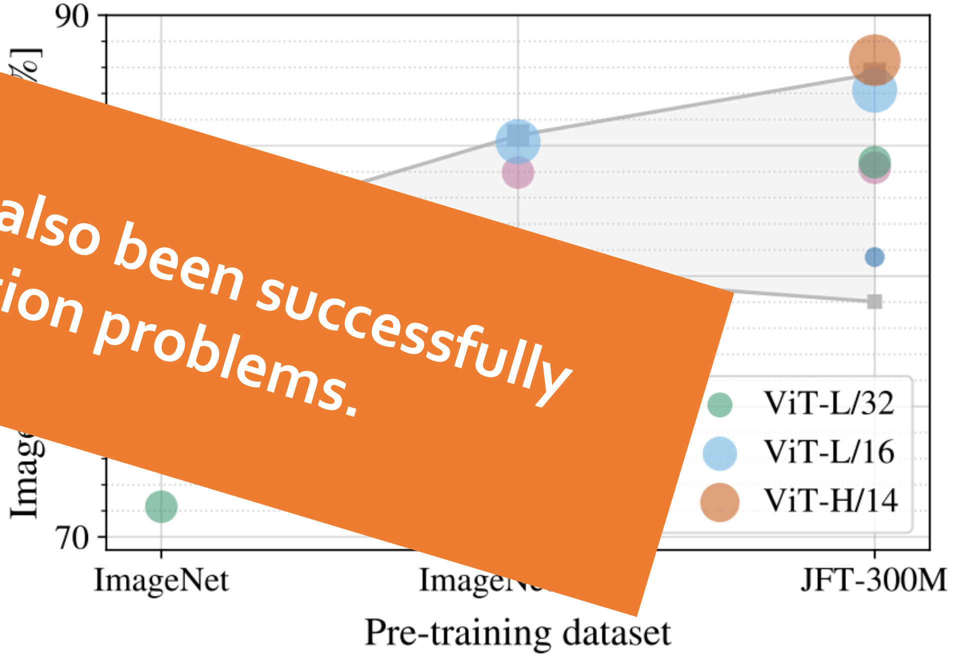
- Vision tra sformers marks first time that a NN algorithm has been success lied to both computer vision and NLP.

Vision transformers have now also been successfully applied to Object Detection problems.

By impl experime that a CNN app classification is generally su ViT when trained on a small dataset. However, as the dataset size increases, ViT begin to outperform CNNs.

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.



BiT (Big Transfer): supervised transfer learning model with CNN-based ResNet

Credit: Dosovitskiy et al. (2020)

Question time

Mentimeter quiz

Let’s take a few minutes to recap what we have discussed so far:

Go to menti.com and enter the code 5601 5416.

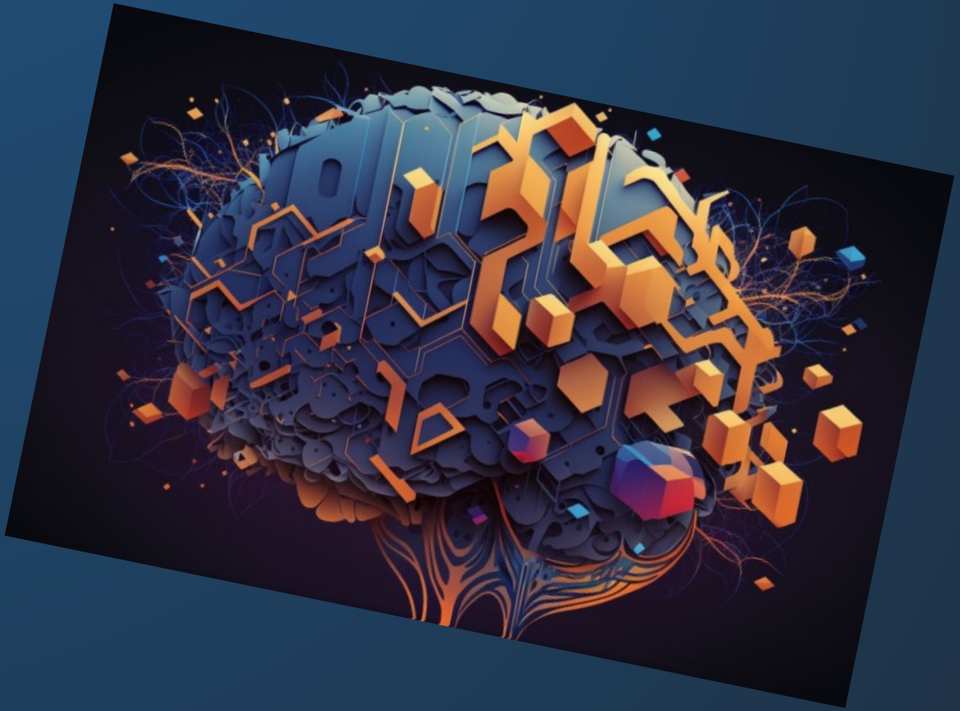
Object detection overview

Object detection with NNs

The concept of attention

Vision transformers

Summary



Further reading I

Summary of several references discussed today

- Viola and Jones, *Rapid object detection using a boosted cascade of simple features,* Proceedings IEEE Computer Society Conference on Computer Vision, I-I (2001)
- Uijlings et al., *Selective Search for Object Recognition*, International Journal of Computer Vision, 104, 154 (2013)
- Girshick et al., *Rich feature hierarchies for accurate object detection and semantic segmentation,* Proceedings IEEE Computer Society Conference on CV, 580 (2014)
- Girshick, *Fast R-CNN*, IEEE International Conference on Computer Vision, 1440 (2015)
- Redmon et al., *You Only Look Once: Unified, Real-Time Object Detection,* Proceedings IEEE Computer Society Conference on Computer Vision, 779 (2016)

Further reading II

Summary of several references discussed today

- Lin et al., *Focal Loss for Dense Object Detection,* IEEE Transactions on Pattern Analysis and Machine Intelligence, 42, 318 (2018) (RetinaNet)
- Duan et al., *CenterNet: Keypoint Triplets for Object Detection,* IEEE/CVF International Conference on Computer Vision, 6568 (2019)
- Bahdanau et al., *Rapid object detection using a boosted cascade of simple features,* arXiv e-print, arXiv:1409.0473 (2014)
- Vaswani et al., *Attention is all you need,* arXiv e-print, arXiv:1706.03762 (2017)
- Dosovitskiy et al.*, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,* arXiv e-print, arXiv:2010.11929 (2020)

Summary

- Object detection is the field of image processing that is concerned with creating bounding boxes around objects and labelling them.
- It is a key research area in computer vision that has numerous applications across many sectors. This is a key skill to have!
- Classical ML is outperformed by DL methods, which can be separated into two-stage (R-CNN family) and single-stage (YOLO family) frameworks. Both have advantages depending on the application.
- Inspired by our brains, self-attention (a key concept in transformers) captures contextual relationships between elements in a sequence.
- Vision transformers apply self-attention in the context of computer vision by learning semantic relationships between different images. They are particularly powerful when pretrained on large datasets.