Hogeschool van Amsterdam

Computer Vision Les 4

Van DL naar Foundation Models

Michiel Bontenbal & Maarten Post

Woensdag 24 oktober
12:00 – 15:20 uur

Creating Tomorrow

Computer Vision les 4

1. Segmentatie en object detectie
   - Notebook : YOLO en/of DETR
   - Notebook : Segment Anything2
2. Nadelen van CNN’s
   - Notebook: adversarial attacks => kat wordt toaster
3. Classification with ResNet and transformer
   - Notebook: ResNet - Olifanten herkennen
   - Notebook: Vision Transformer
4. Foundation Models & Self-Supervised Learning
   - Notebook : Florence 2

Computer Vision sessies

Bootcamp 1: CV highlights
- MNIST handgeschreven cijfers met NN en CNN’s
- Image embeddings & similarity search
- Vision Language Models with ollama

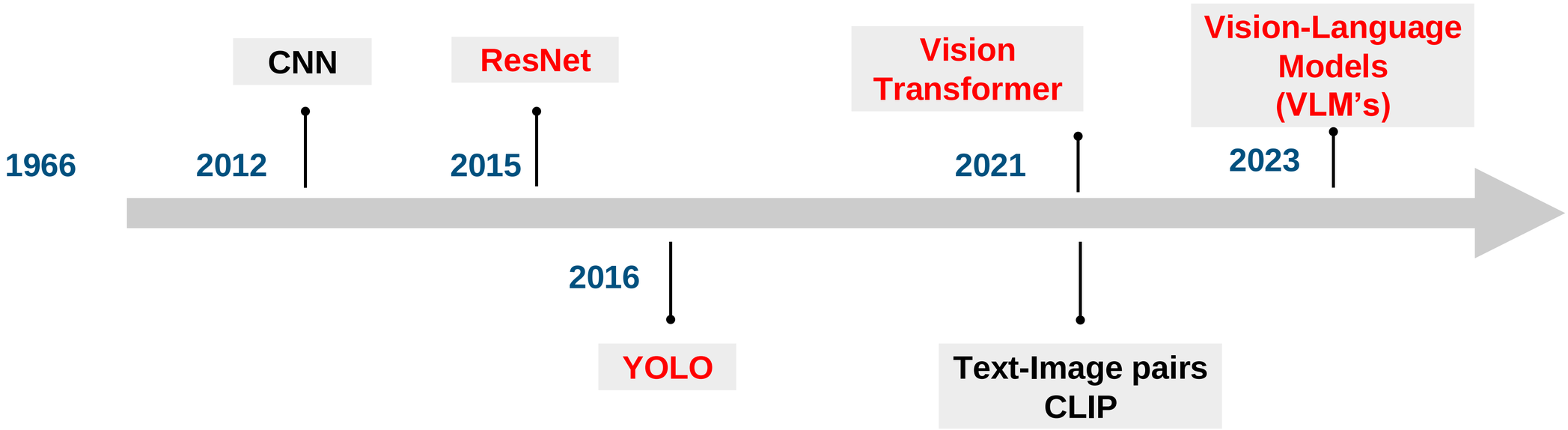
Bootcamp 2: Pre-processing
- OpenCV: Edge Detection en gezichtsherkenning
- Numpy for images

Deep learning - Generative AI for vision
- Auto encoders
- Diffusion models

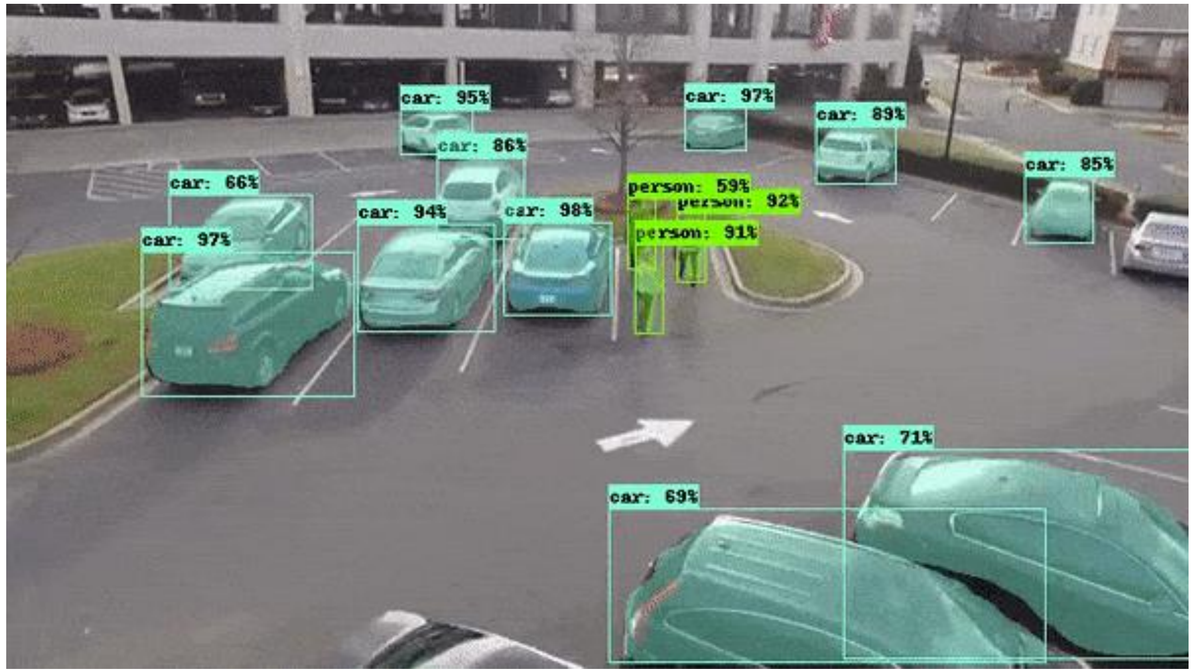
Deep Learning - Computer Vision (vandaag)
- **Segmentatie en object detectie**
- **Nadelen Convolutional Neural Nets + adversarial neural nets**
- **Foundation models & self supervised learning**

Advanced Topics: Data Centric AI (19 november)

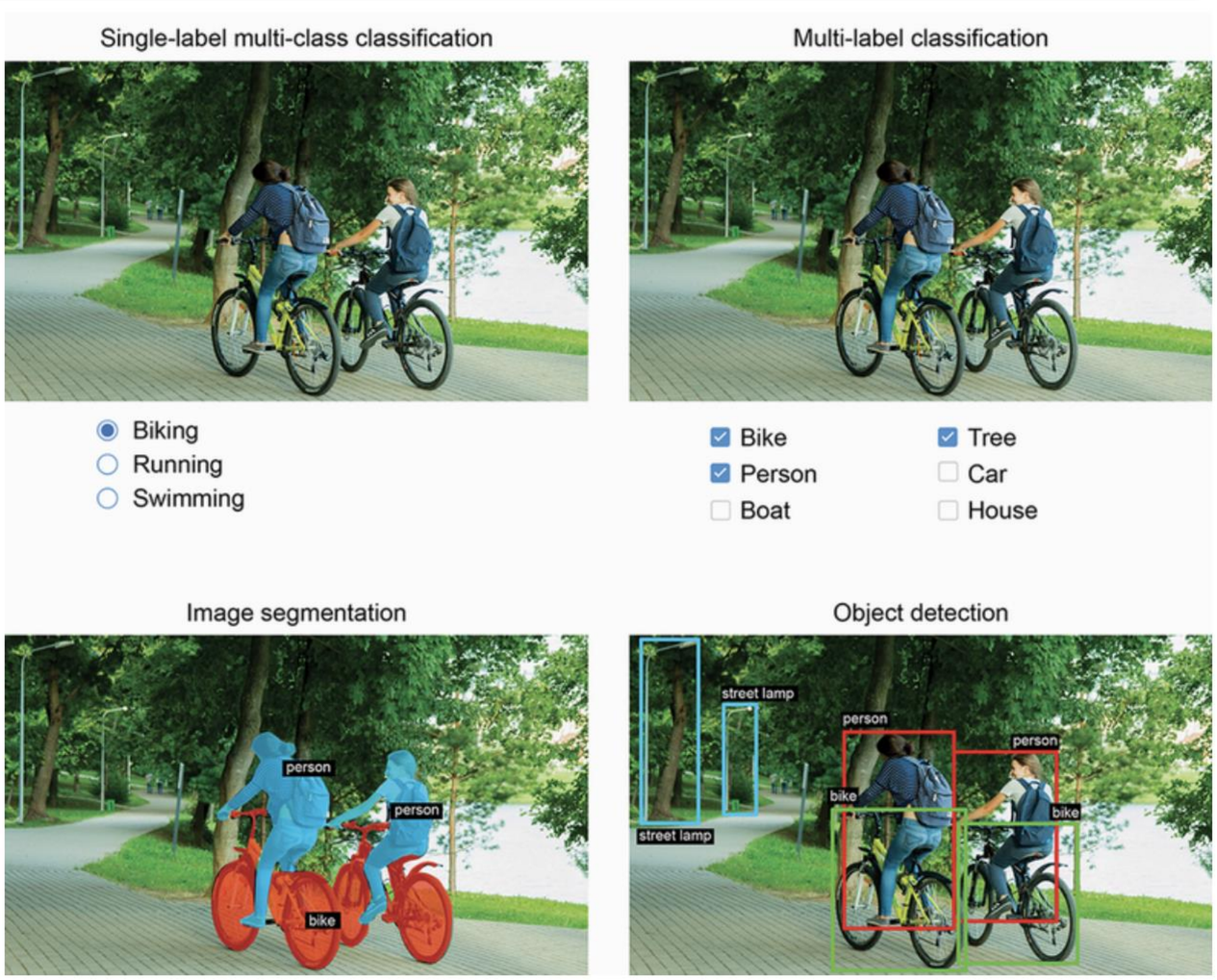
Tijdslijn Computer Vision



Segmentatie en Object Detectie



Classification, segmentation, detection

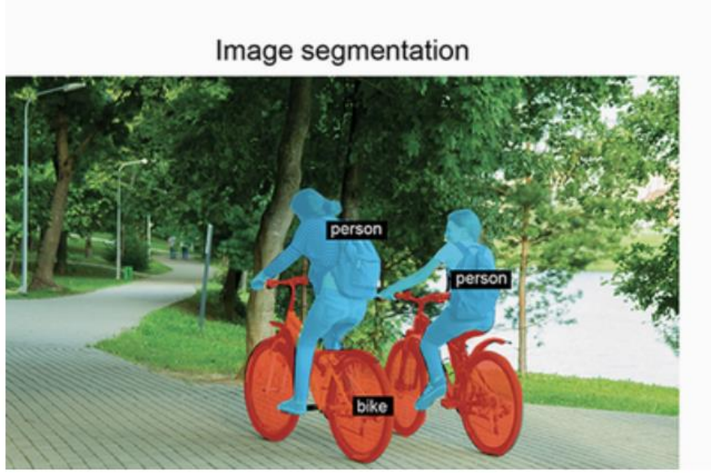


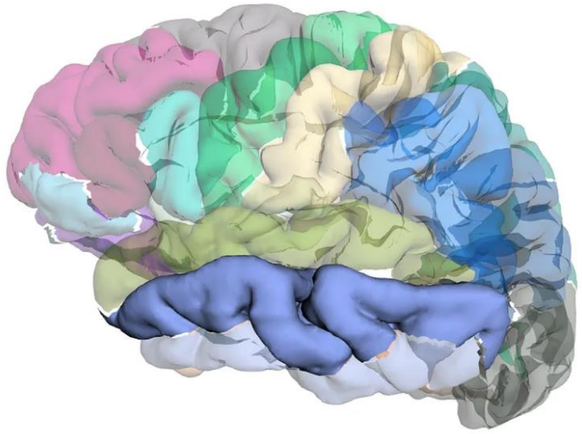
Source: Deep Learning with Python - Francois Chollet

Segmentatie

- Segmentatie is het proces van het verdelen van een digitaal beeld in meerdere **segmenten.**

- Het helpt bij het begrijpen en analyseren van de structuur van afbeeldingen.





Best model = SegmentAnything

Developed in 2023 by Meta, open weights

- Used for image segmentation:
  - Semantic segmentation
  - Instance segmentation

- Maar ook voor:
  - object detectie
  - Depth estimation
  - Cut-outs

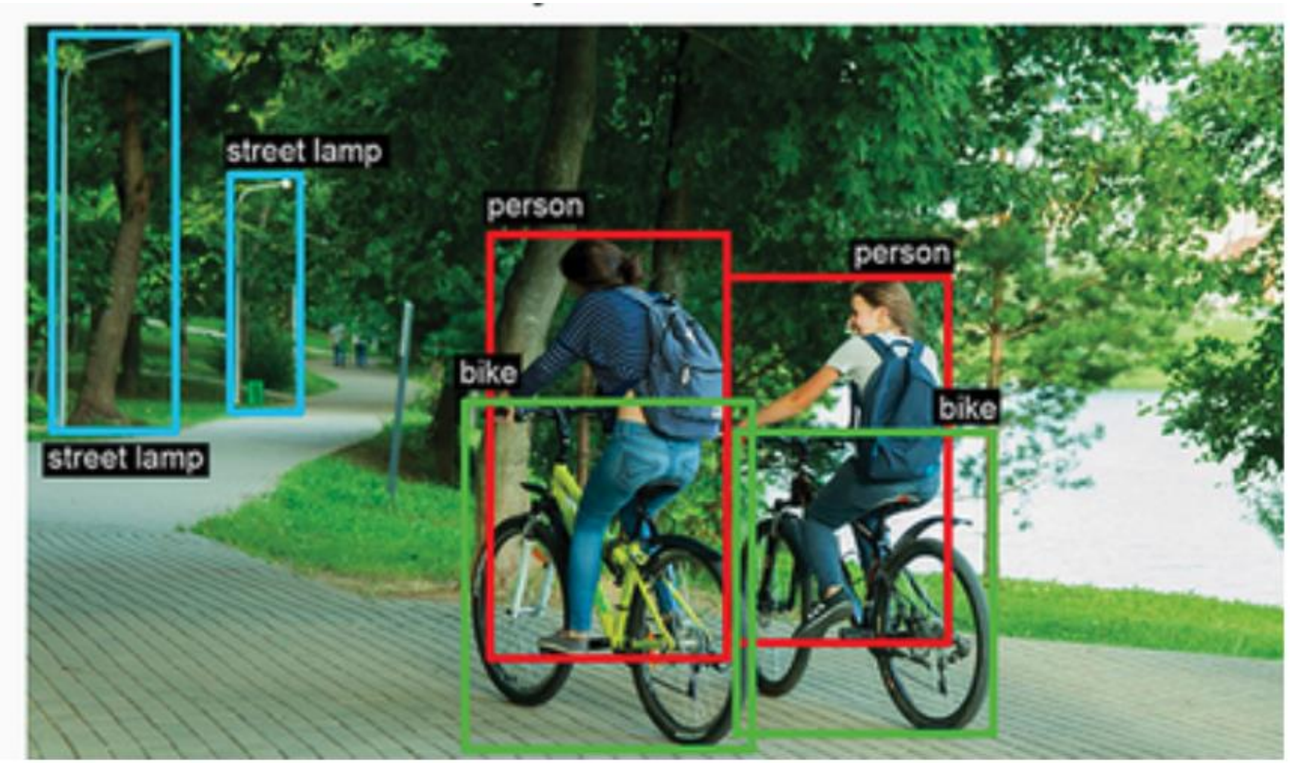
- www.segment-anything.com

Oefening

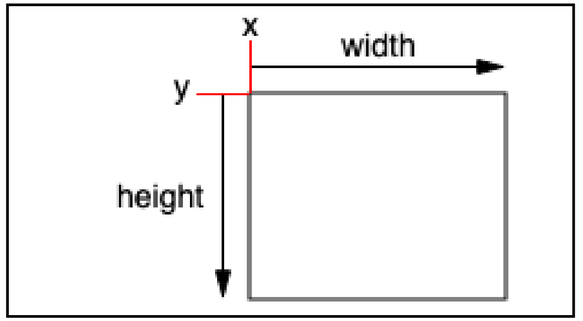
- Speel even met www.segment-anything.com
  - Probeer wat taken uit
  - Wellicht wat images vanuit je project?

- N.B. We gaan aan de slag met Segmentation Notebook.

Object detection



Bounding Box



Use cases:

- **Meerdere types objecten**
- **Lokaliseren van objecten**
- **Objecten tellen**

Confidence wordt berekend.
Gebruik treshold.
Eén bounding box per object obv algoritme (non-max suppresion).

Hoe vind je een fiets?

Hoe vind je een fiets? Exhaustive…

Three types of Object Detection algo’s

‘Region’ or ‘two stage‘ approach

- R-CNN
- Fast-RCNN
- Faster-RCNN
- DETECTRON (2)

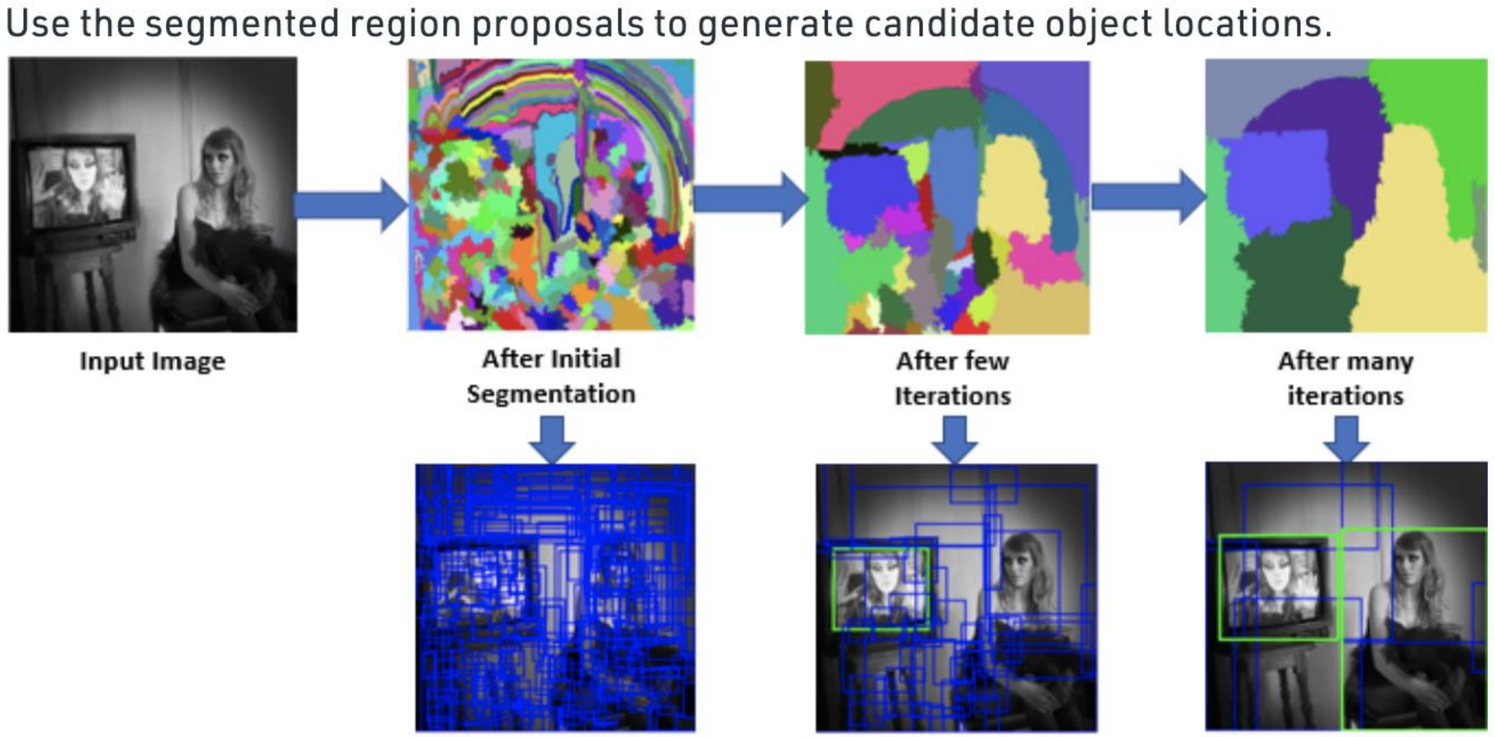
‘Grid’ or one stage approach

- YOLO
- Single Shot Detector
- RetinaNet

Transformer based

- DETR
- RT-DETR

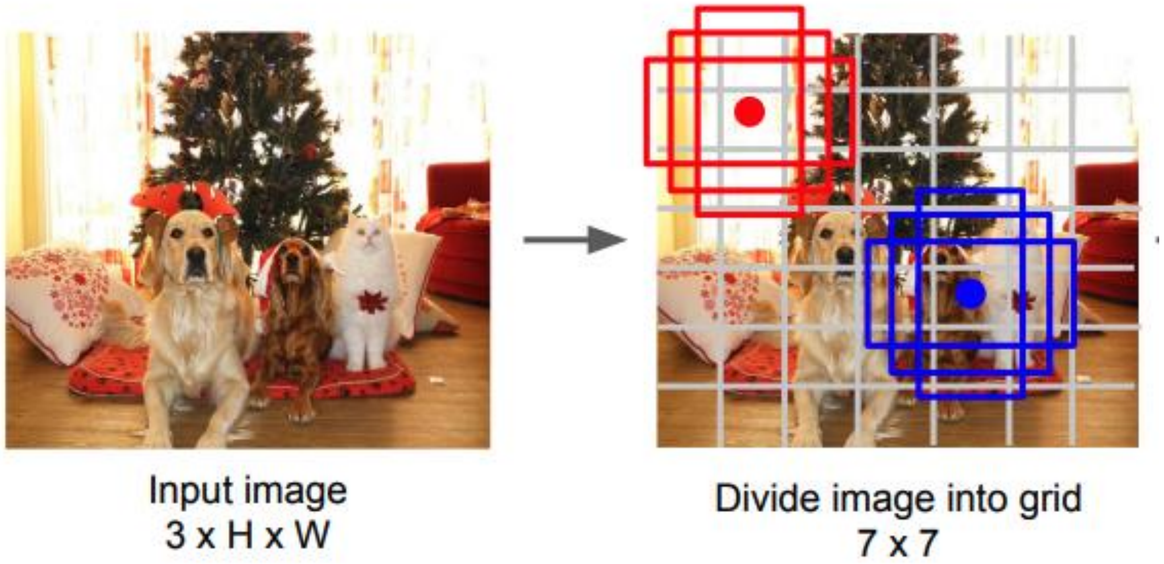
Region based approach OD met R-CNN



Aanpak: Gebruik *selective search* om regions te vinden, combineer deze en bepaal mogelijke object locaties

Sources: https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013/UijlingsIJCV2013.pdf
https://www.geeksforgeeks.org/selective-search-for-object-detection-r-cnn/

Grid based approach (YOLO)

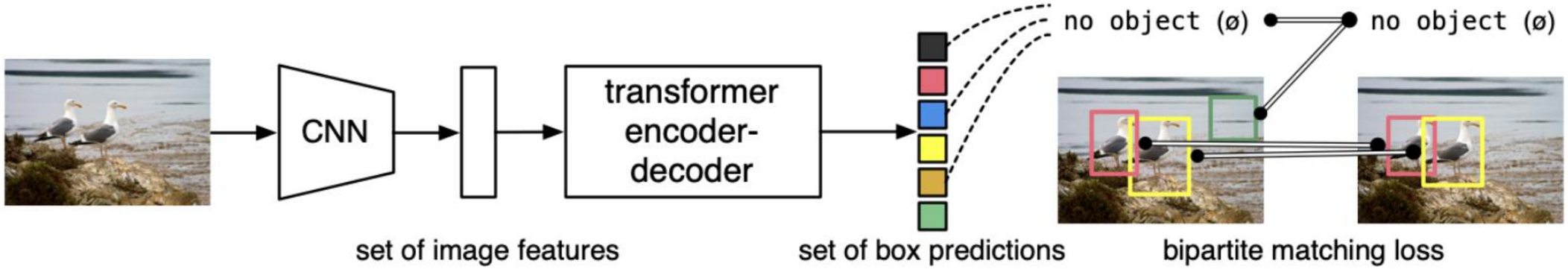


Maak een grid aan.
Voor elke grindcel:
- **Maak meerdere bounding boxes**
- **Bepaal probability voor elke klasse**

Toon de hoogste probability met bijbehorende bounding box.

- YOLO = You Only Look Once
- Geintroduceerd in 2016 door Joseph Redmon
- Wordt nog steeds doorontwikkeld => inmiddels YOLOv11

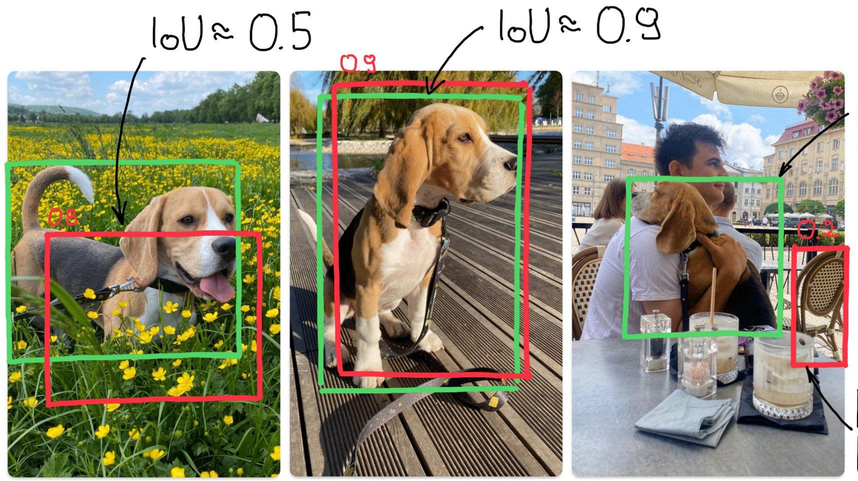
Transformer Based

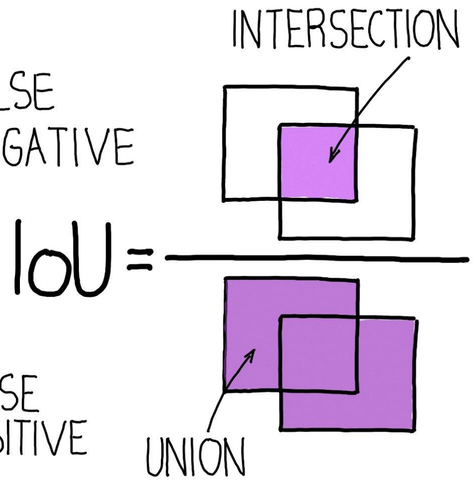


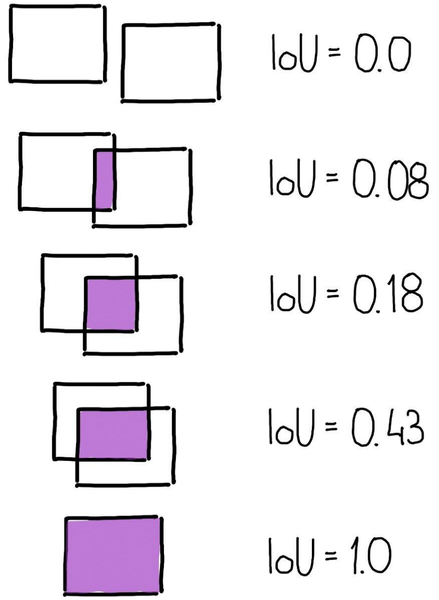
- Met CNN/ResNet worden de features onderscheiden.
- Vervolgens worden ook de posities bepaald (‘positional encoding’)
- Deze worden aan een encoder – transfomer gevoerd. (die is getraind op images niet op text !)
- Dit leidt tot predictions – waarbij de belangrijkste worden getoond.

- Model van Facebook 2020

Metrics







DETECTIONS \ ANNOTATIONS	POSITIVE	NEGATIVE
POSITIVE	TRUE POSITIVE [TP]	FALSE POSITIVE [FP]
NEGATIVE	FALSE NEGATIVE [FN]	TRUE NEGATIVE [TN]

THE SAME DETECTION CAN BE TRUE POSITIVE OR FALSE NEGATIVE DEPENDING ON **IoU THRESHOLD**

IF IoU VALUE GREATER THAN THRESHOLD IT IS **TP**. IF IT IS SMALLER DETECTION IS **FN**.

$$\text{PRECISION} = \frac{TP}{TP + FP} \qquad \text{RECALL} = \frac{TP}{TP + FN}$$

$$\text{F1 SCORE} = 2 \times \frac{(\text{PRECISION} \times \text{RECALL})}{(\text{PRECISION} + \text{RECALL})}$$

; Tomorrow

Object detectie + classificatie

Stap 1: Object detectie ‘auto’

Stap 2: Classificeer type auto

Trade offs bij selecteren model

- Snelheid versus nauwkeurigheid
- Lokaal OD vraagt om stevige laptop (cpu vs gpu)
- Aantal klassen (bijv. auto’s eerst OD dan classificatie)

Ook YOLO maakt fouten…

rtlnieuws 10 ° 0 km 2 OV

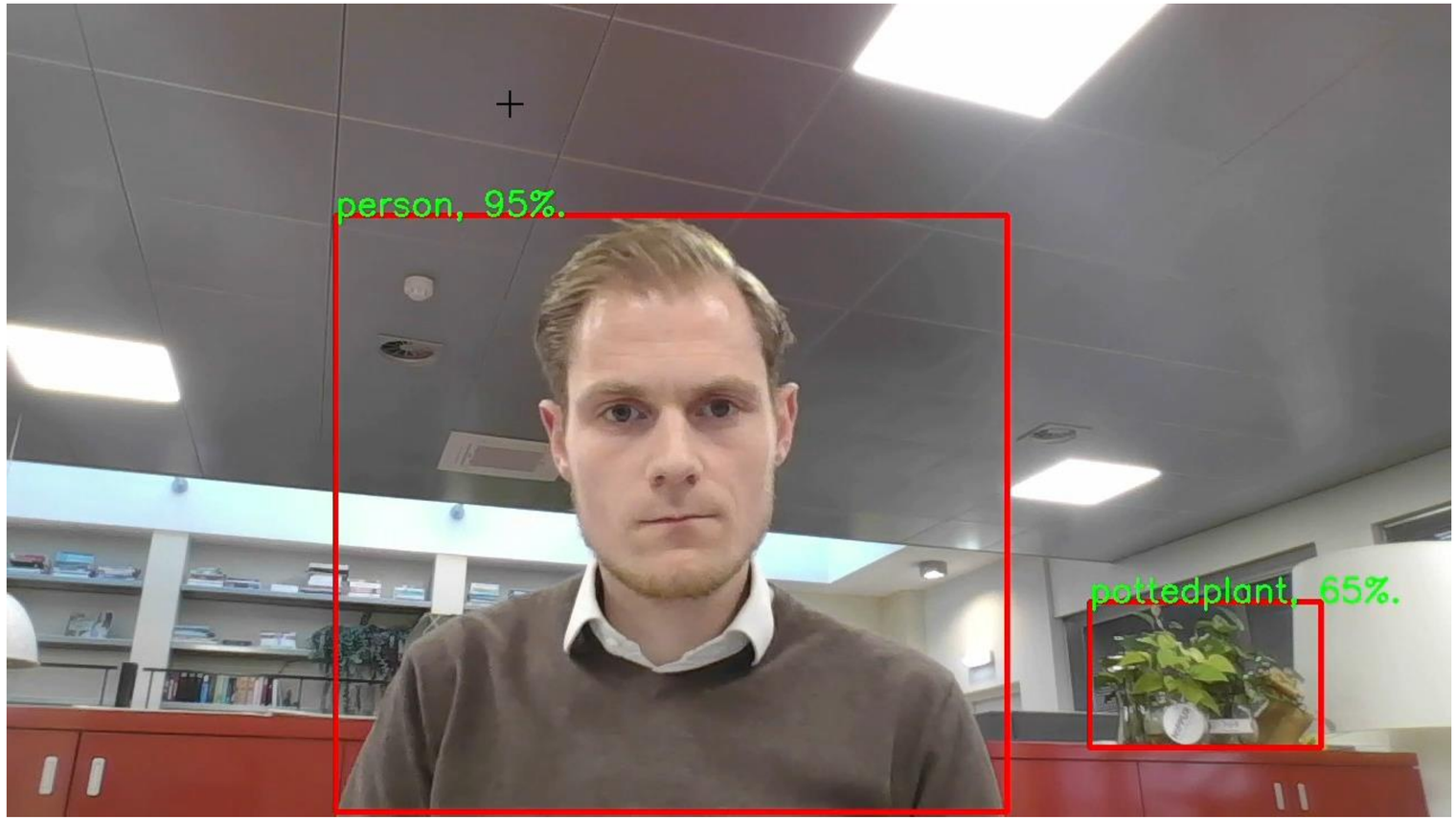
Nieuws Economie Sport Entertainment Tech Lifestyle Editie NL Uitzendingen

Nederland

Tim krabde aan hoofd, maar kreeg boete van 380 euro voor bellen achter het stuur

21 februari 2024 11:48 • Aangepast 21 februari 2024 15:00

Tim versus het algoritme



- Source: https://nippur.nl/tim-versus-politie-algoritme/

Huggingface Transformers

- Huggingface heeft een python package gemaakt waar heel veel modellen in staan.
  - Vooral Transformer architecturen maar ook niet-transformers zoals ResNet.
- Open Source dus alle modellen vrij te gebruiken.
- Top 10 GitHub repo’s.
- Ook in Javascript: Transformers.js
- https://huggingface.co/docs/transformers/index

Notebooks:
Object Detectie en Segmentatie

`24_10_24_YOLO.ipynb`

`24_10_24_segment_anything_transformers.ipynb`

Object Detectie - meer lezen

Arxiv papers:
R-CNN paper: https://arxiv.org/pdf/1311.2524.pdf
YOLO paper: https://arxiv.org/abs/1506.02640

Stanford Course Computer Vision - Object Detection les:
http://cs231n.stanford.edu/slides/2022/lecture_9_jiajun.pdf

Metrics
https://blog.roboflow.com/mean-average-precision/#what-is-object-detection

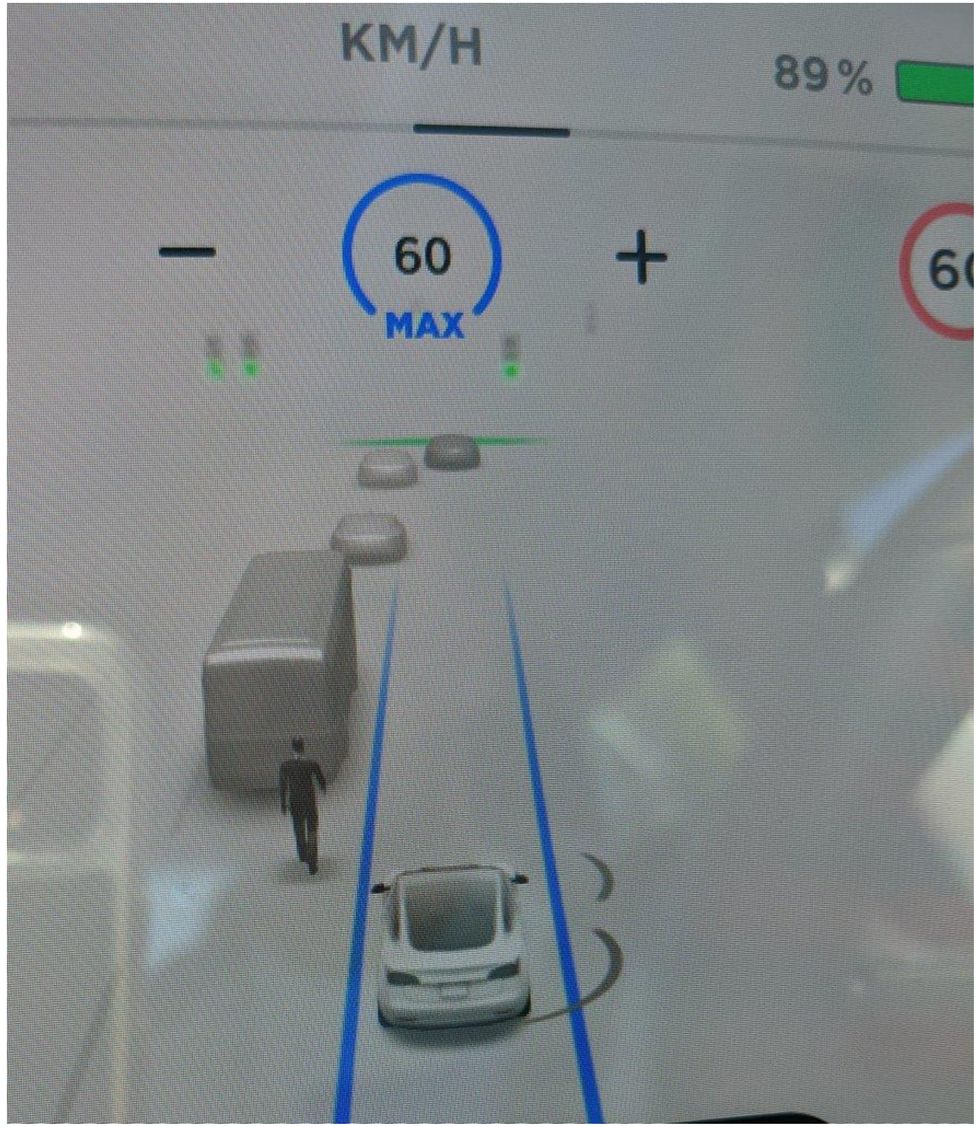
Overzichtsartikelen
https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

https://analyticsindiamag.com/r-cnn-vs-fast-r-cnn-vs-faster-r-cnn-a-comparative-guide/#:~:text=This%20is%20the%20basic%20difference,call%20the%20regional%20proposal%20network

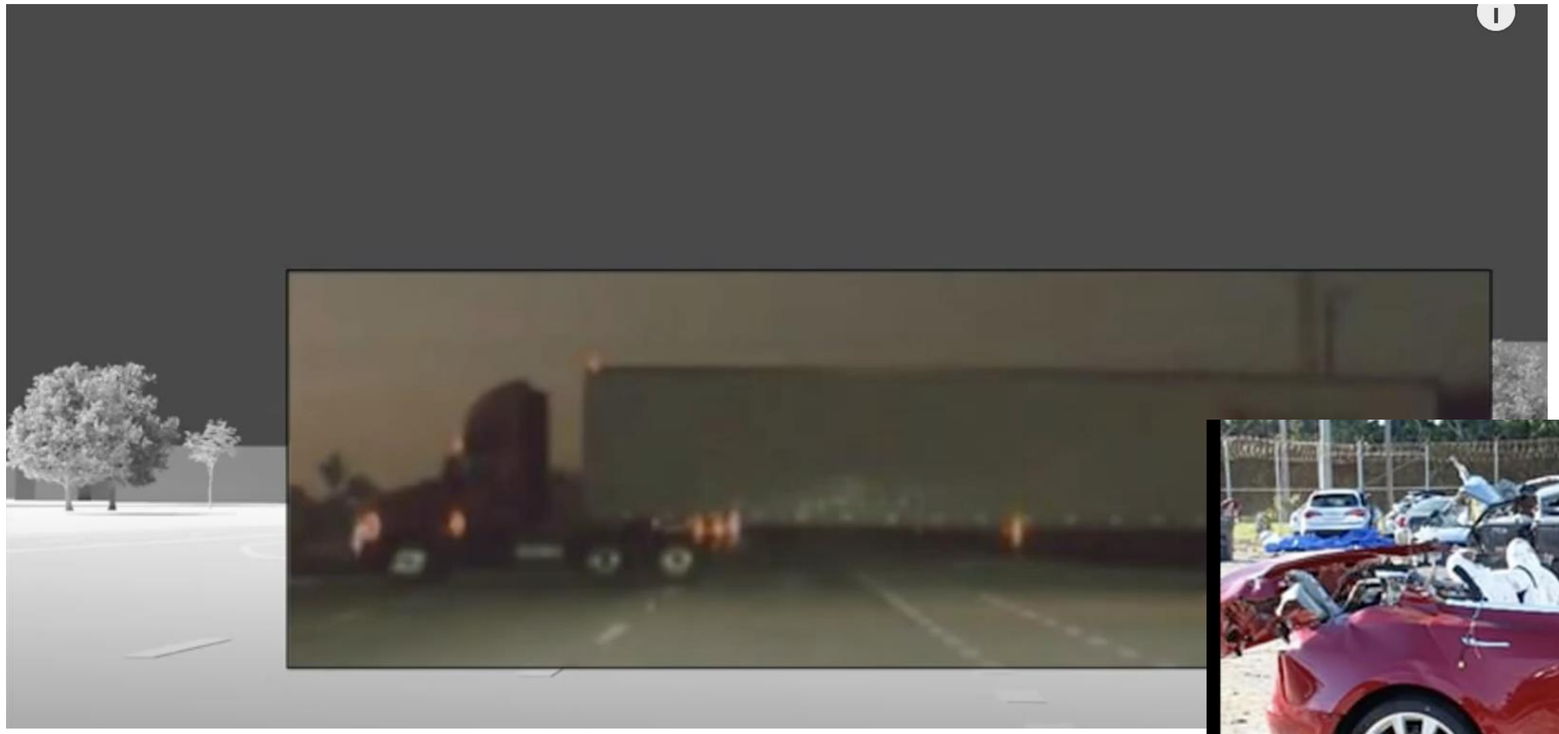
Nadelen van Neural Nets

Neural nets maken fouten

Neural Nets maken fouten



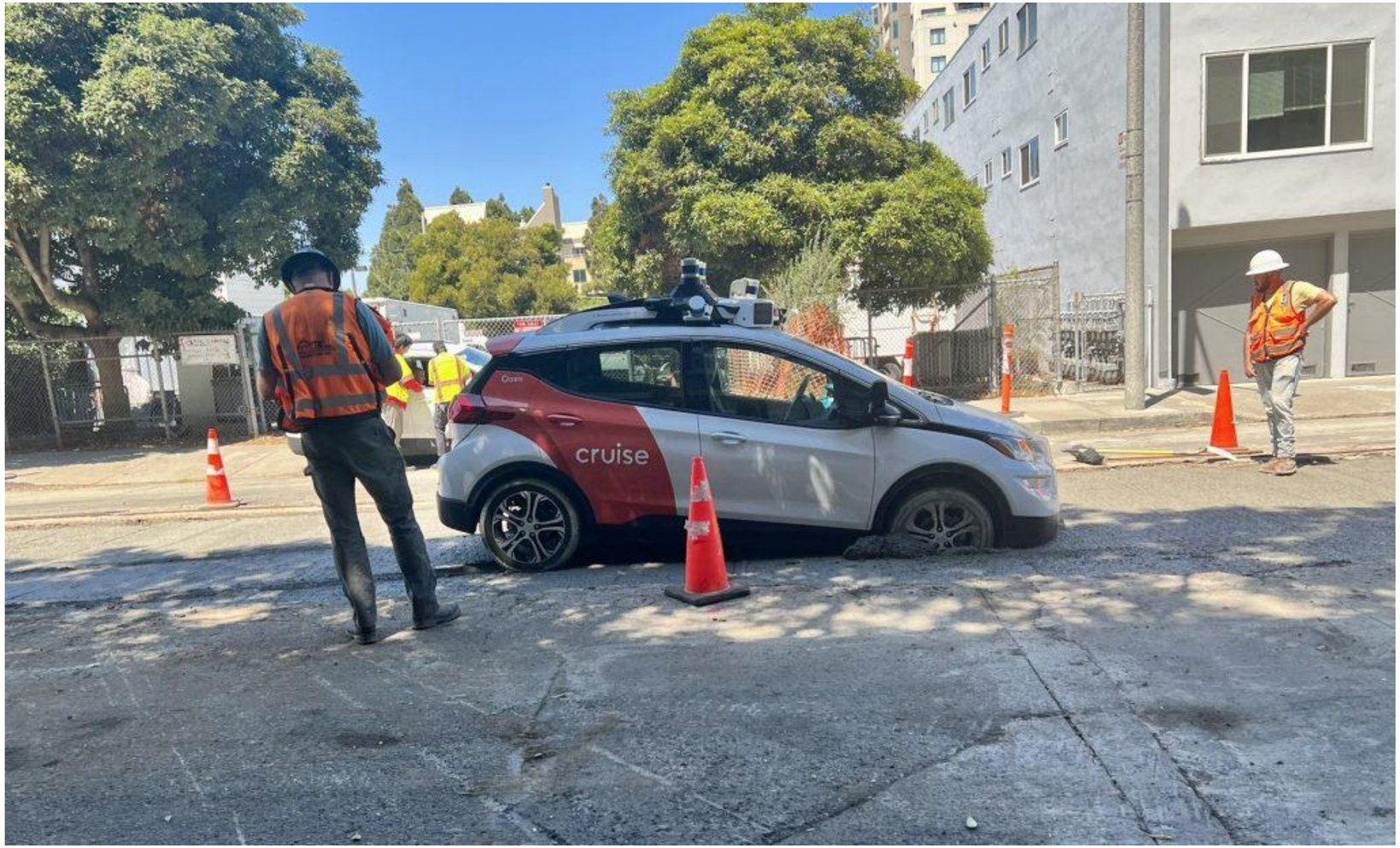
Neural nets maken fouten



https://www.youtube.com/watch?v=7oprWTnnBqM

Prank

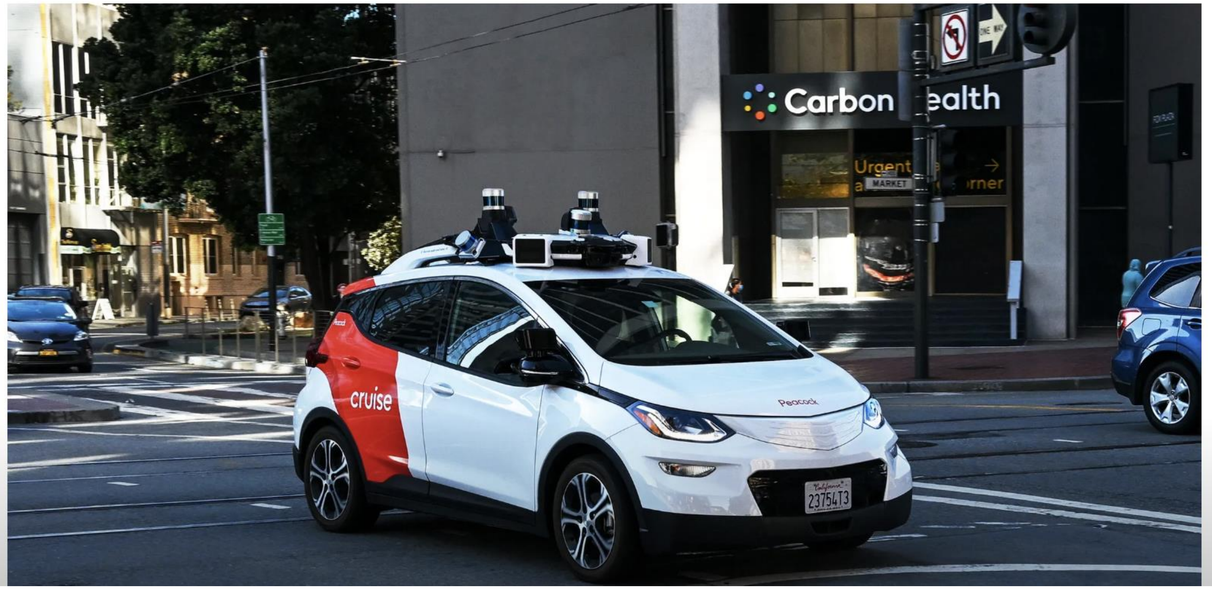
Neural nets maken fouten



En dat heeft gevolgen…

GM's Cruise Loses Its Self-Driving License in San Francisco After a Robotaxi Dragged a Person

The California DMV says the company's autonomous taxis are "not safe" and that Cruise "misrepresented" safety information about its self-driving vehicle technology.

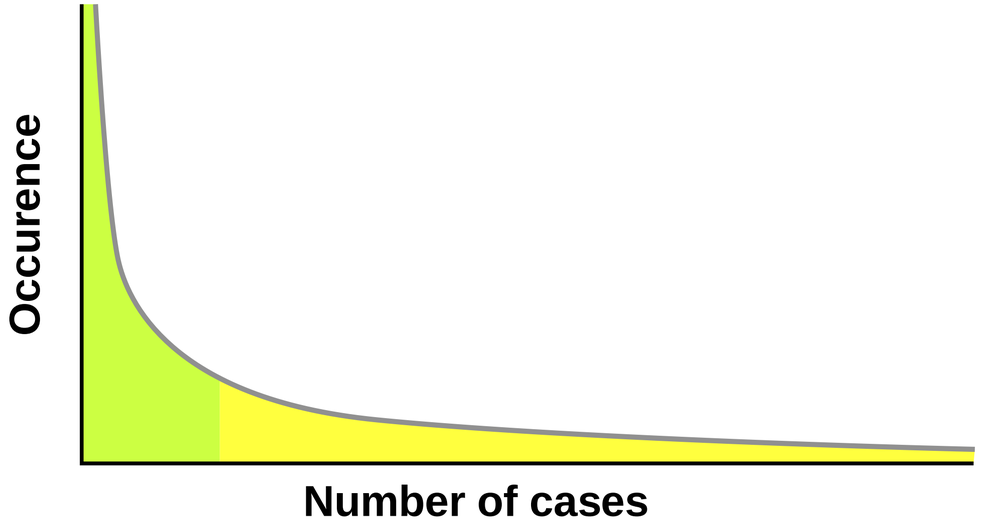


https://www.wired.com/story/cruise-robotaxi-self-driving-permit-revoked-california/

Geloofde Elon het eigenlijk zelf wel…?

https://youtu.be/zhr6fHmCJ6k?si=56NcdCWyScPJGGSO&t=9

Waarom is de zelfrijdende auto nog steeds een belofte?



Heel veel ‘edge cases’ zijn niet te automatiseren…

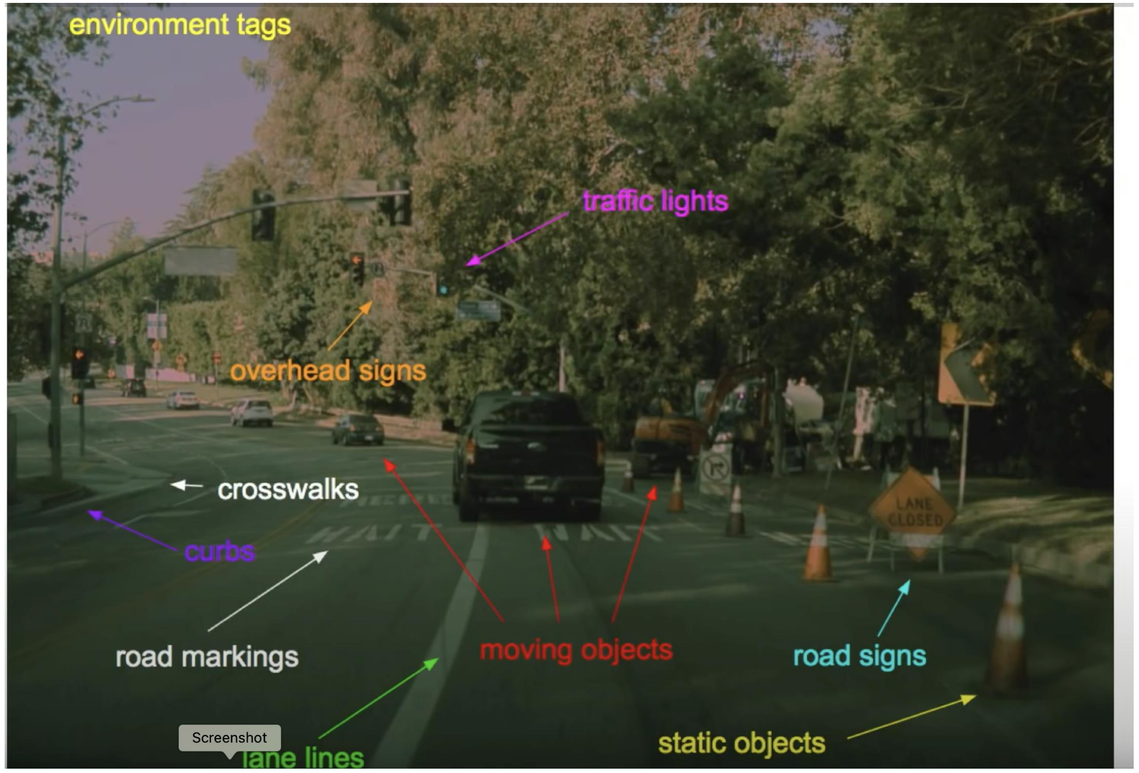
Businessweek | The Big Take

Even After $100 Billion, Self-Driving Cars Are Going Nowhere

They were supposed to be the future. But prominent detractors—including Anthony Levandowski, who pioneered the industry—are getting louder as the losses get bigger.

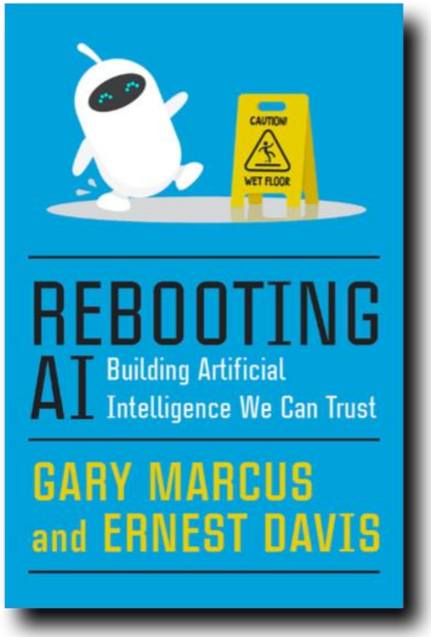
https://archive.ph/sMhUt

Tegelijkertijd is het razend knap wat een auto al kan.



Nadelen Convolutional Neural Nets

- **“Greedy”** = gulzig
  - vraagt heel veel data en heel veel berekeningen
- **“Opaque”** = ondoorzichtig
  - het is niet precies duidelijk hoe ze werken
- **“Brittle”** = breekbaar
  - kleine veranderingen kunnen grote gevolgen hebben



http://rebooting.ai

Gary Marcus, Ernest Davis

https://twitter.com/GaryMarcus

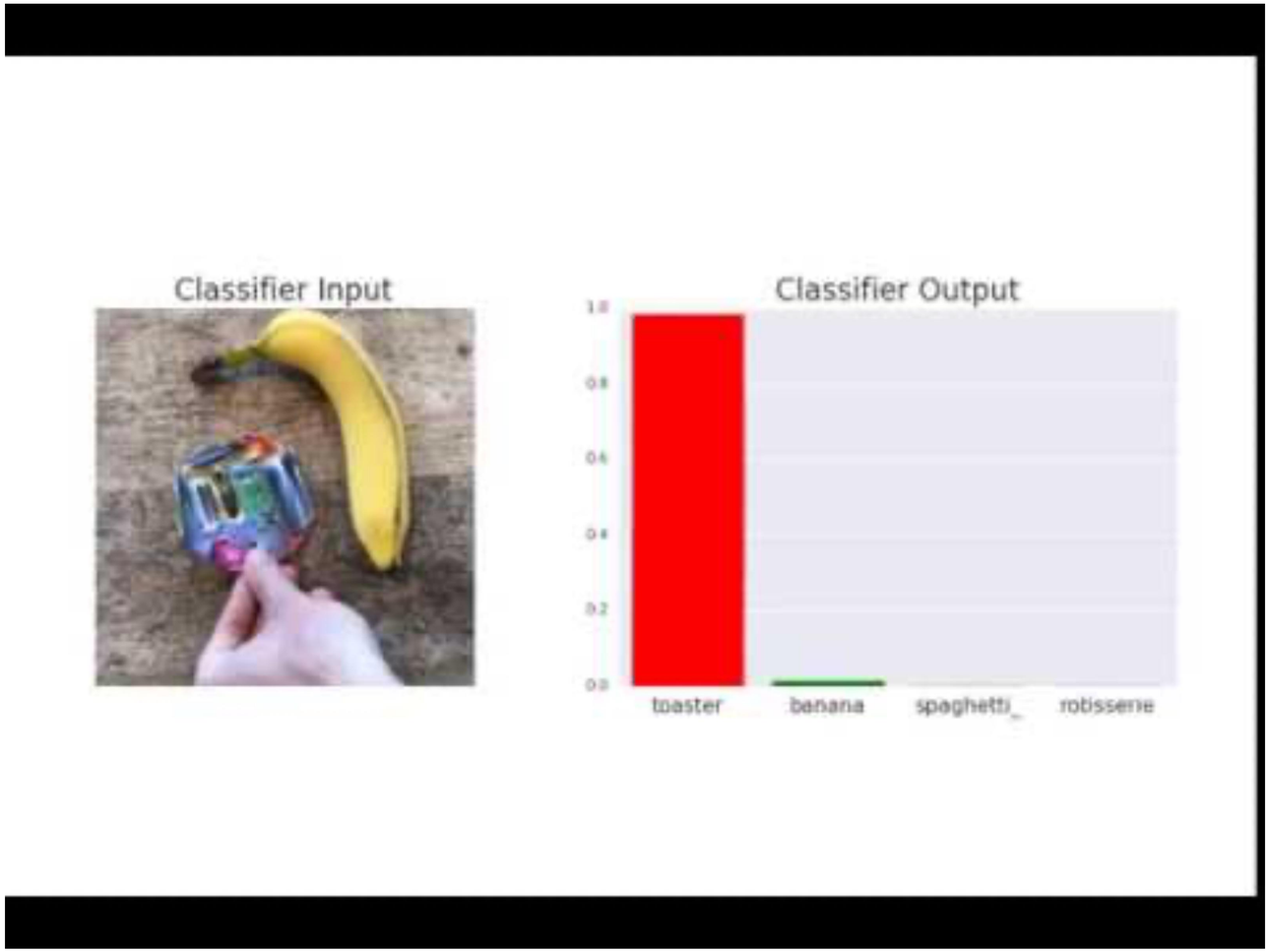
Conclusie: Hoe goed zijn CNNs?

- CNNs werken voor sommige domeinen heel goed
- Toch maken CNNs vreemde en ongewenste fouten
- Zijn voor de gek te houden ‘adversarial attack’.

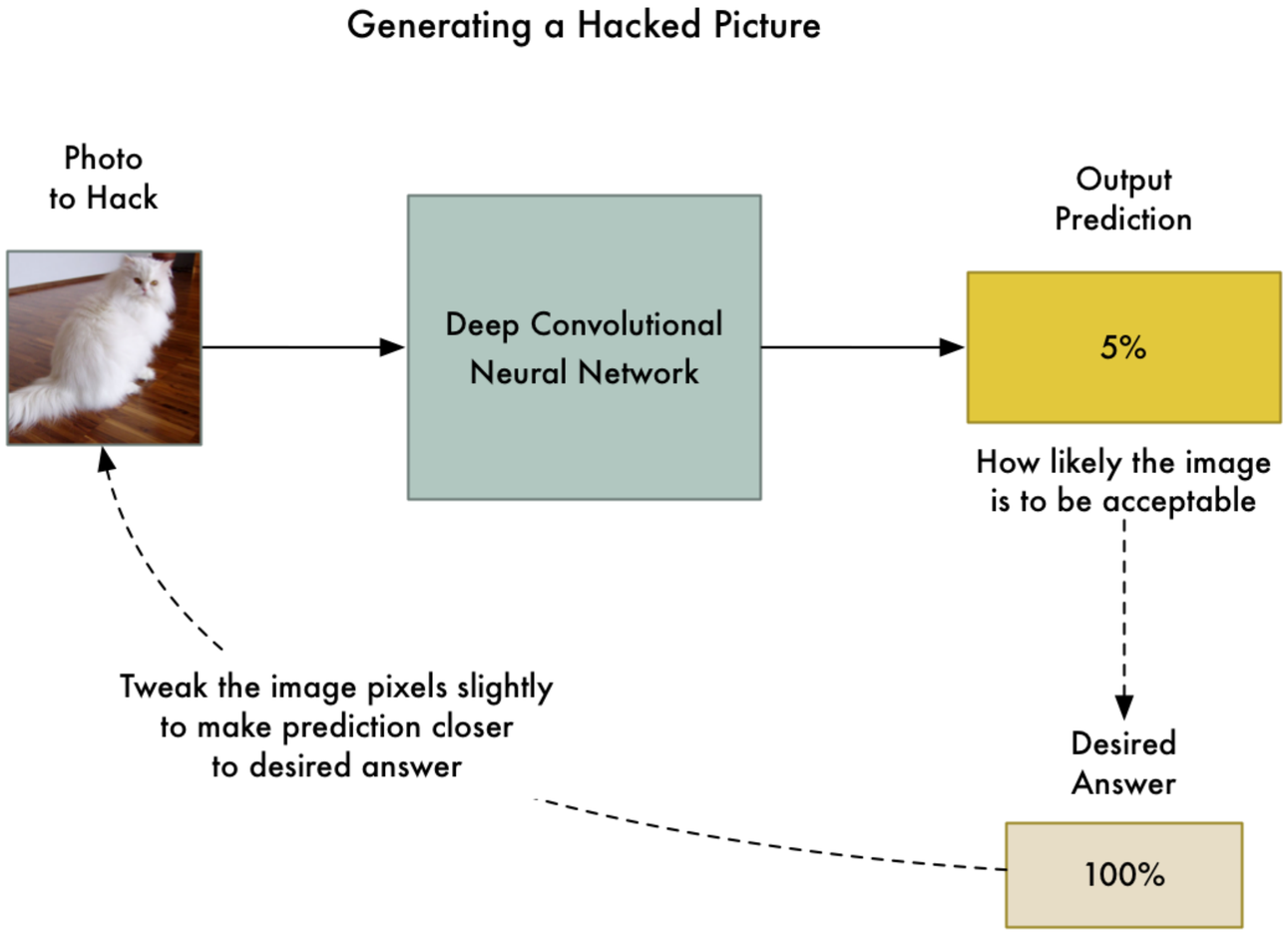
Niet toepassen in kritieke situaties:

- Verkeer, longarts, etc
- Zorg dat een mens de controle houdt
- ‘Human-in-the-loop’

Adversarial attacks



Cat wordt toaster



source: https://medium.com/@ageitgey/machine-learning-is-fun-part-8-how-to-intentionally-trick-neural-networks-b55da32b7196

Oefening fake a toaster

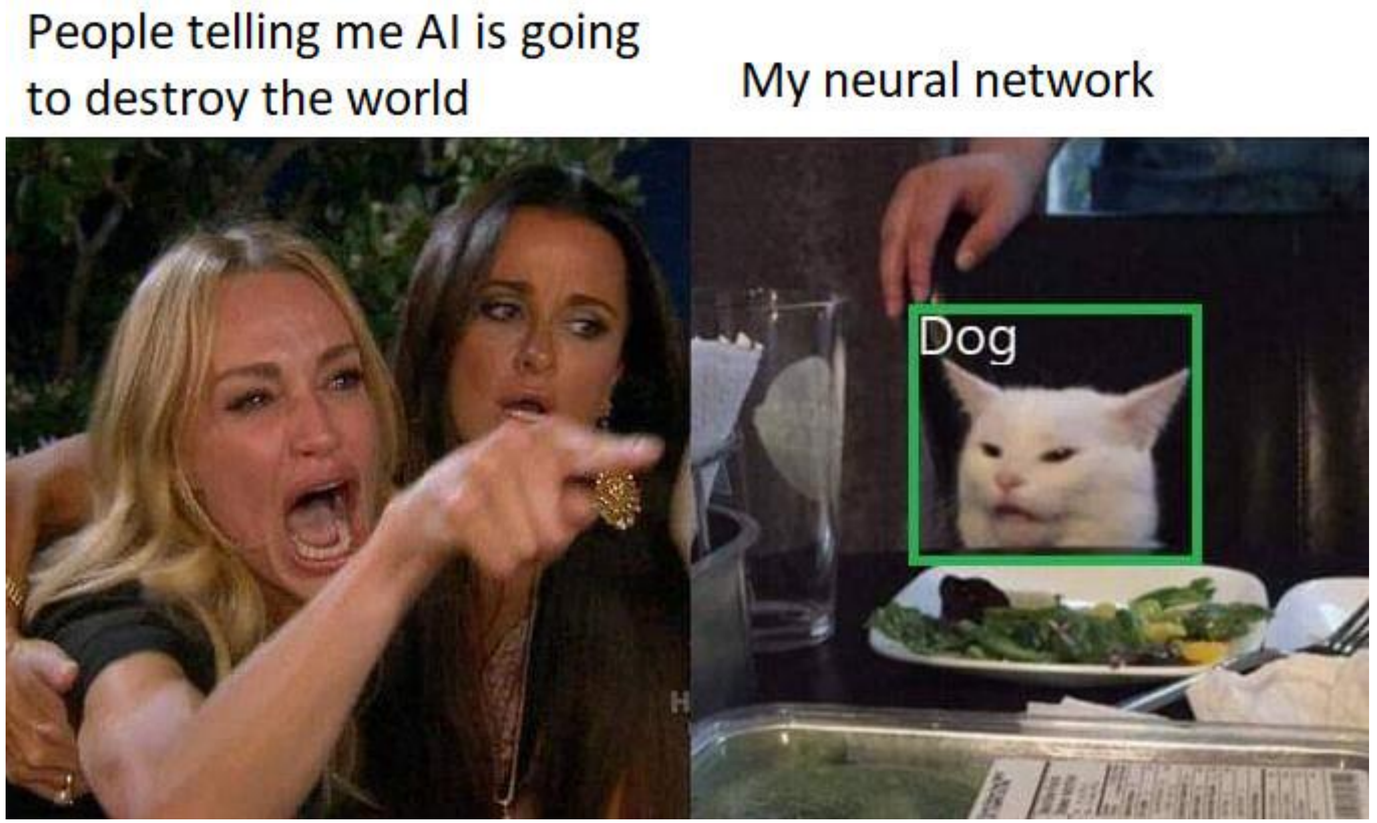
Run het volgende notebook en bekijk of jouw cat een toaster wordt.

Zie het artikel: **https://medium.com/@ageitgey/machine-learning-is-fun-part-8-how-to-intentionally-trick-neural-networks-b55da32b7196**
Bonus: het wetenschappelijke paper: **https://archive.ph/nBKAK**

Gebruik de image cat.png van DLO.

`2024_10_24_Fake_a_toaster.ipynb`

Bespreking

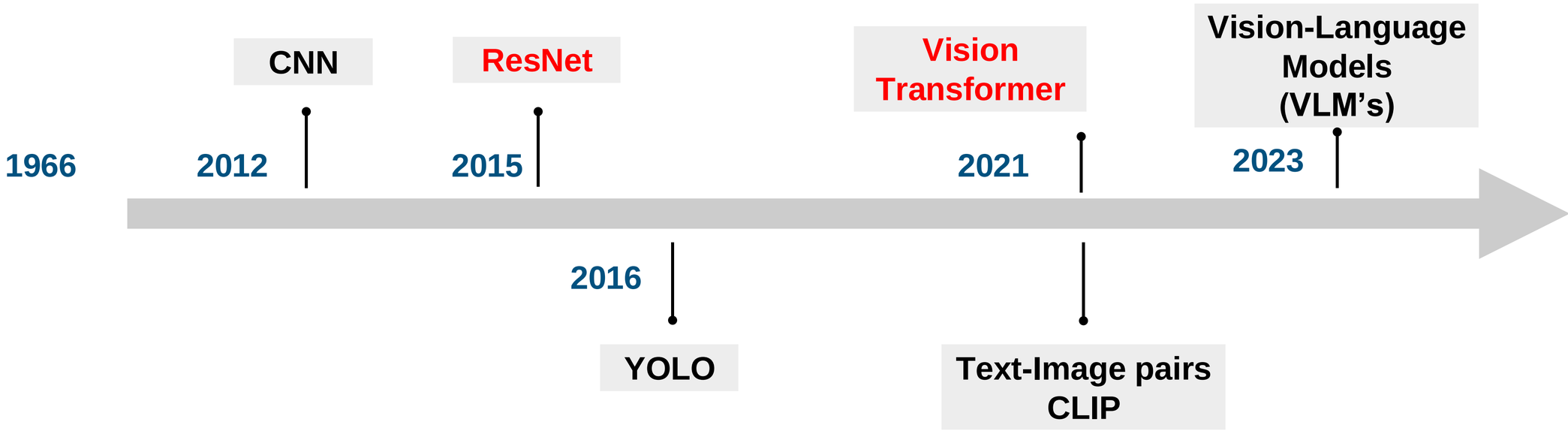

People telling me AI is going to destroy the world
My neural network
Dog

Hogeschool van Amsterdam

Classification with ResNet & Vision Transformer

Creating Tomorrow

Tijdslijn Computer Vision

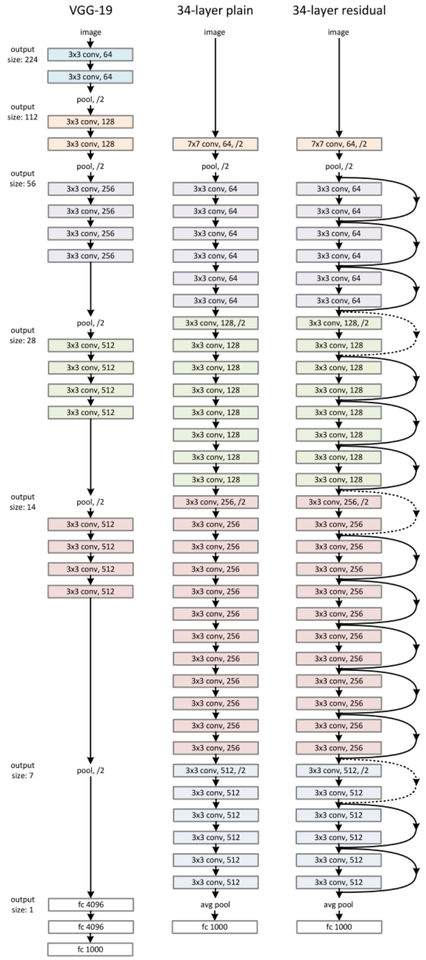


ResNet

ResNet

- Getraind op dataset ImageNet met nauwkeurigheid 0.964
- Geplubliceerd 2015 door Microsoft
- Wordt nog steeds gebruikt – vooral ResNet50 en ResNet18

- Opdracht: bestudeerd en tel de lagen in het model

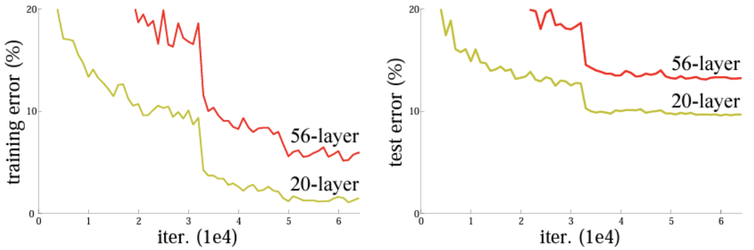


- https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/
- https://arxiv.org/abs/1512.03385 - **ResNet paper**

De residual layer is de kern van ResNets

- Tijdens onderzoeken naar Neural Nets kwam men achter het
- Volgende:
  - Meer lagen in een NN kunnen tot slechte resultaten leiden
  - => *Vanishing Gradient Problem*

  - Oplossing: gebruik een ***residual layer***.
  - Werking in pseudo-code:
    - Als resultaat != goed
      - Skip die laag



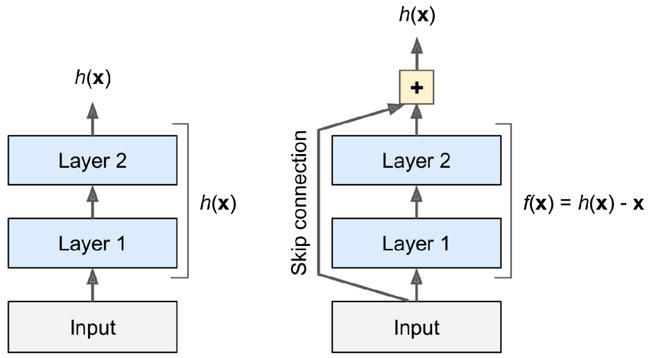
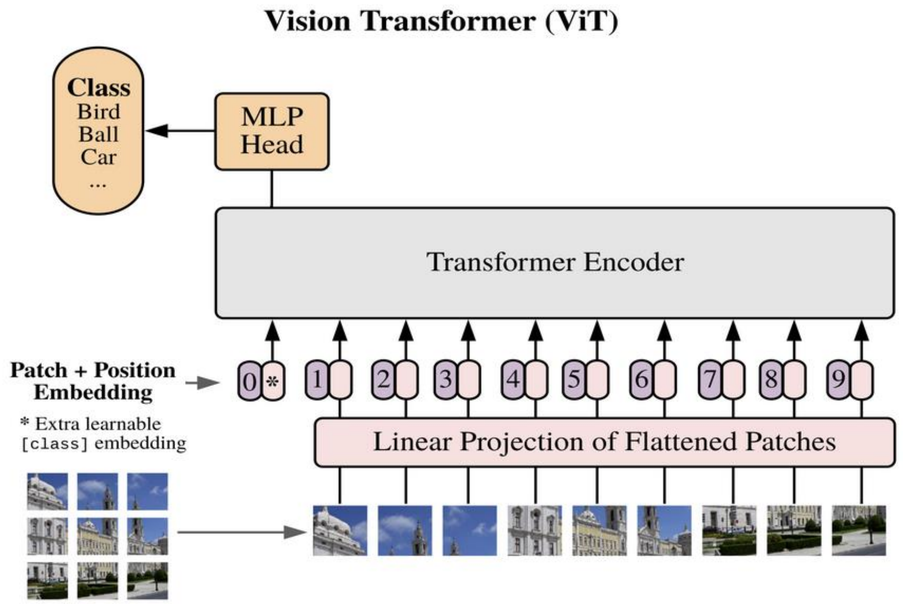


Figure 14-15. Residual learning

Vision Transformers (ViT)

- Volgend op het succes van Transformers in NLP, werd dit in 2021 toegepast in Computer Vision.
- Het model maakt gebruik van ‘embeddings’ en gebruikt informatie over de positie
- De transformer is een encoder-transformer.
- De resultaten waren beter dan ResNet, maar het bleef een beperkte verbetering. Recent artikel concludeert: CNN’s zijn net zo goed als ViT



Vision Transformer (ViT)

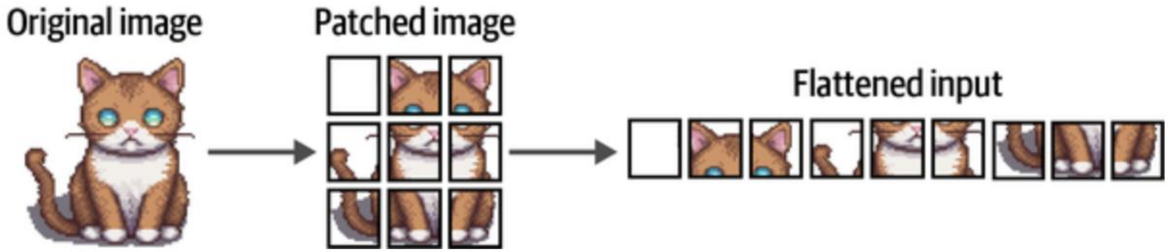


Figure 9-4. The “tokenization” process for image input. It converts an image into patches of subimages.

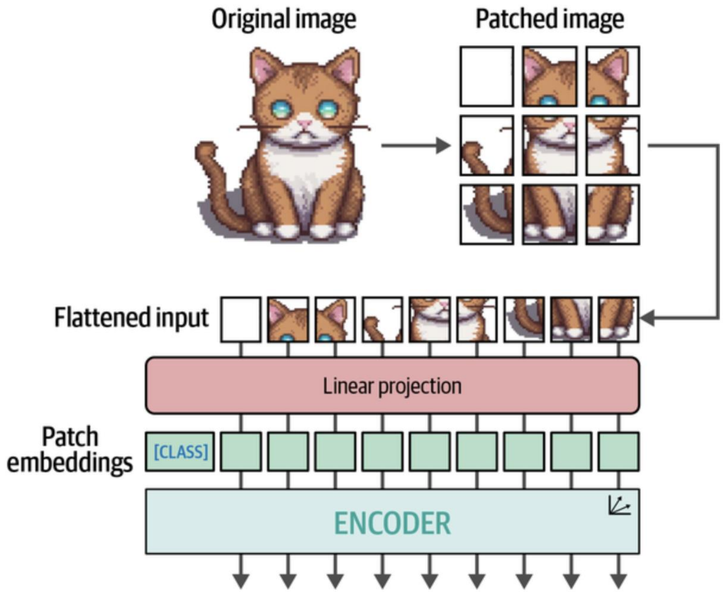


Figure 9-5. The main algorithm behind ViT. After patching the images and linearly projecting them, the patch embeddings are passed to the encoder and treated as if they were textual tokens.

- Source: Hands on Large Language Models, Alammar/Grootendorst, 2024, o’Reilly

Opdracht: Olifantenherkenning

https://medium.com/@robertgeirhos/why-deep-learning-works-differently-than-we-thought-ec28823bdbc

We gaan een pretrained ResNet gebruiken getraind op ImageNet

- **AI = model + data**

- Jullie gaan gebruik maken van de ImageNet dataset.

- Gebruikt voor Imagenet Competitie
- Imagenet = 14 miljoen images, 20.00 classes
- 2009 opgezet door Fei-Fei Li van Stanford
- => Imagenet 1k : 1000 classes, 1.3 miljoen images

- https://www.image-net.org/

Opdrachten

- Notebooks op DLO

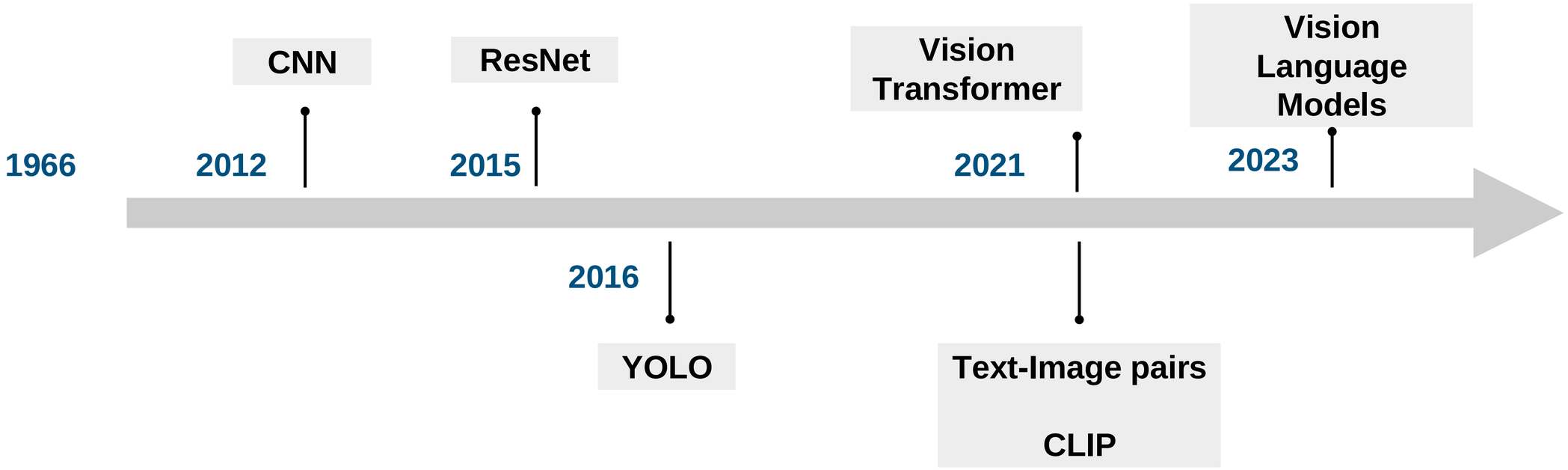
```
24_10_24_ResNet.ipynb

24_10_24_Vision_transformer
```

Foundation Models Self-Supervised Learning in Computer Vision

- Foundation Models
- Self-Supervised Learning

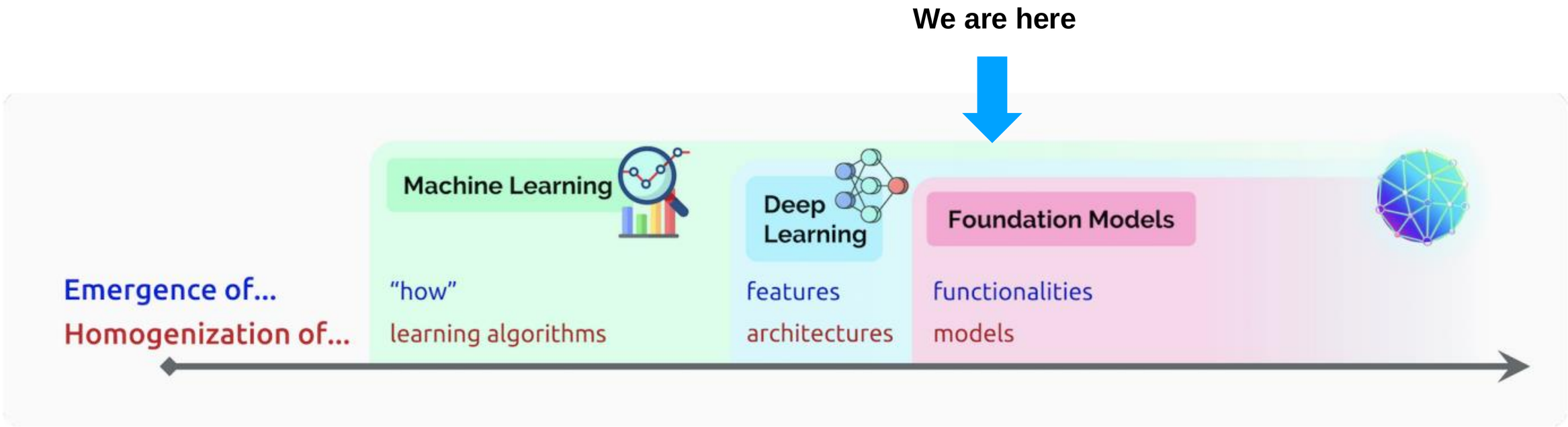
Tijdslijn Computer Vision



Deep Learning / Supervised Learning

Foundation Models

Van DL naar Foundation models



Source: On the Opportunities and Risks of Foundation Models, Stanford, 2021

Foundation models

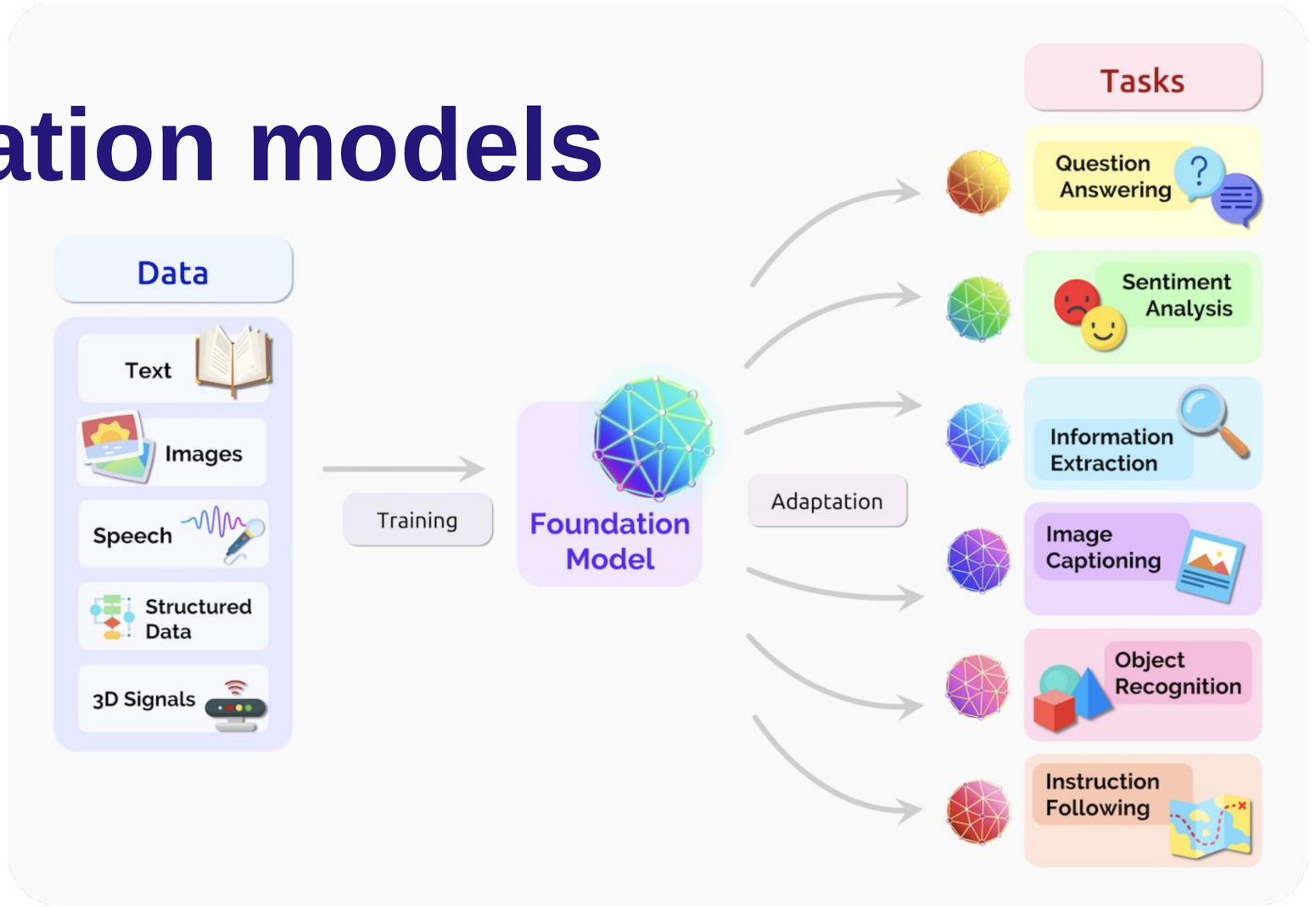


Fig. 2. A foundation model can centralize the information from all the data from various modalities. This one [illegible] be adapted to a wide range of downstream tasks.

Screenshot

Foundation Model

Foundation model definitie:

> "***any model***[1] *that is trained on broad data,*
> *generally using* ***self-supervision***[2] *at scale,*
> *that can be* ***adapted***[3] *to a wide range of downstream tasks.*"

1. Modellen voor zowel tekst, vision, audio en multimodal
2. Self-Supervision learning: model leert zelf op basis van grote hoeveelheden data
3. Adapted – model aanpassen voor taak zoals Classificatie, Segmentatie of Object Detectie

Source: https://en.wikipedia.org/wiki/Foundation_model

Deep Learning vs Foundation Models

Deep learning

Keras of Pytorch

Zelf een model maken
- CNN, autoencoder

Groot ± miljoenen parameters

1 model voor 1 taak

Foundation models

o.a. Transformers library

Gebruiken (finetunen) van een model
- CLIP, VLM, Diffiusion

Zeer groot ± miljarden parameters

1 model voor meer taken

Problemen met supervised learning

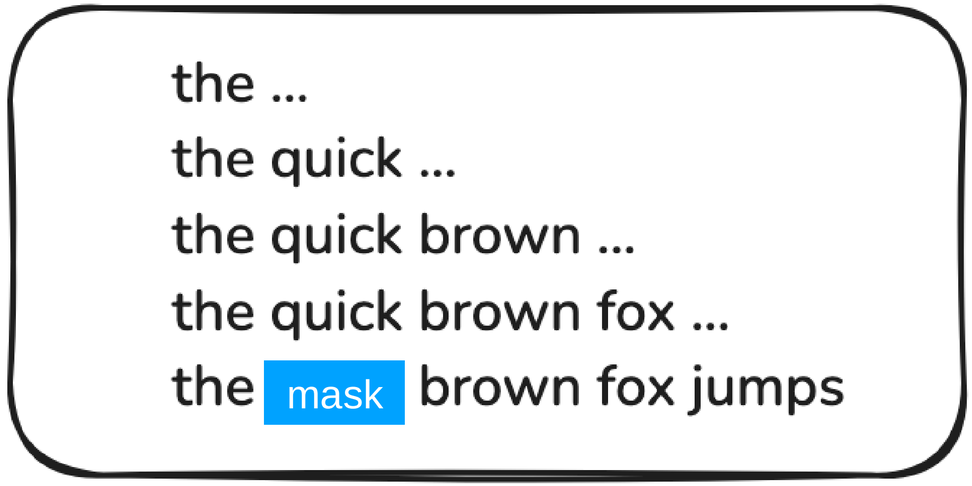
Computer Vision vooral supervised learning: CNN, ResNet, ViT.

1. Veel gelabelde data nodig en labelen is arbeidsintensief / duur
   - denk bijv. aan medische data waar een radioloog moet labelen.

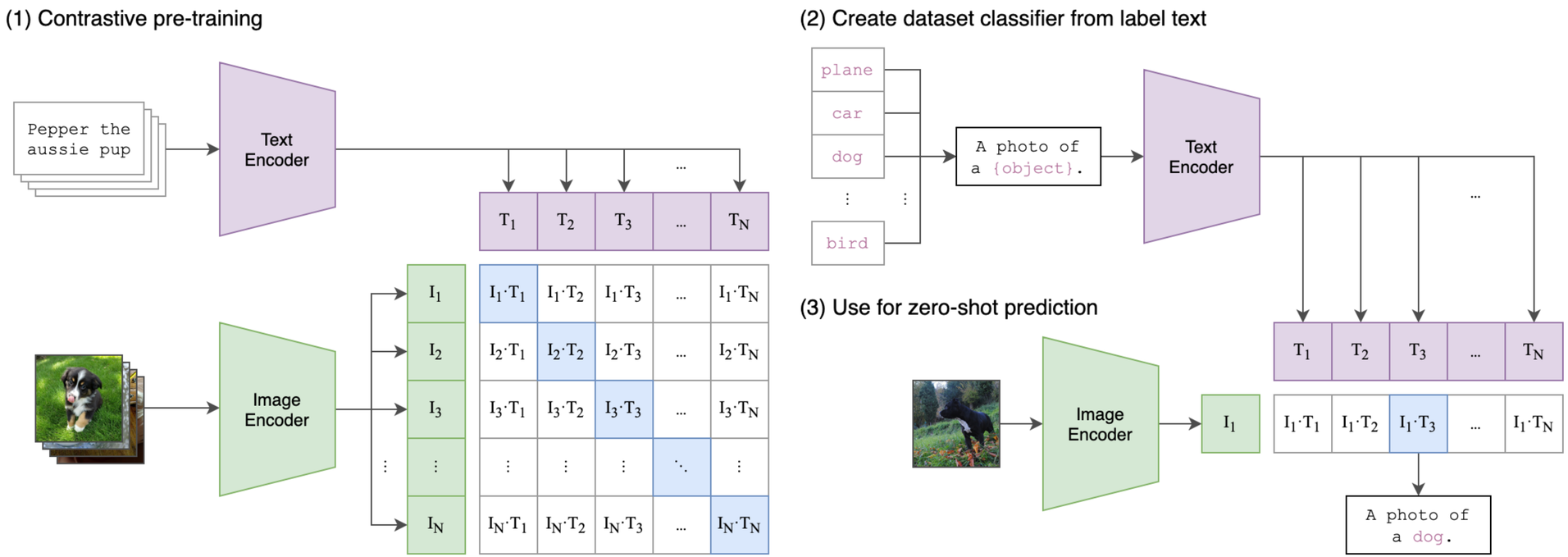
2. CNN kijkt niet alleen naar vorm maar ook naar textuur en/of achtergrond
   - denk aan de zwemmende olifant uit ResNet.

Self-supervised learning ken je uit NLP

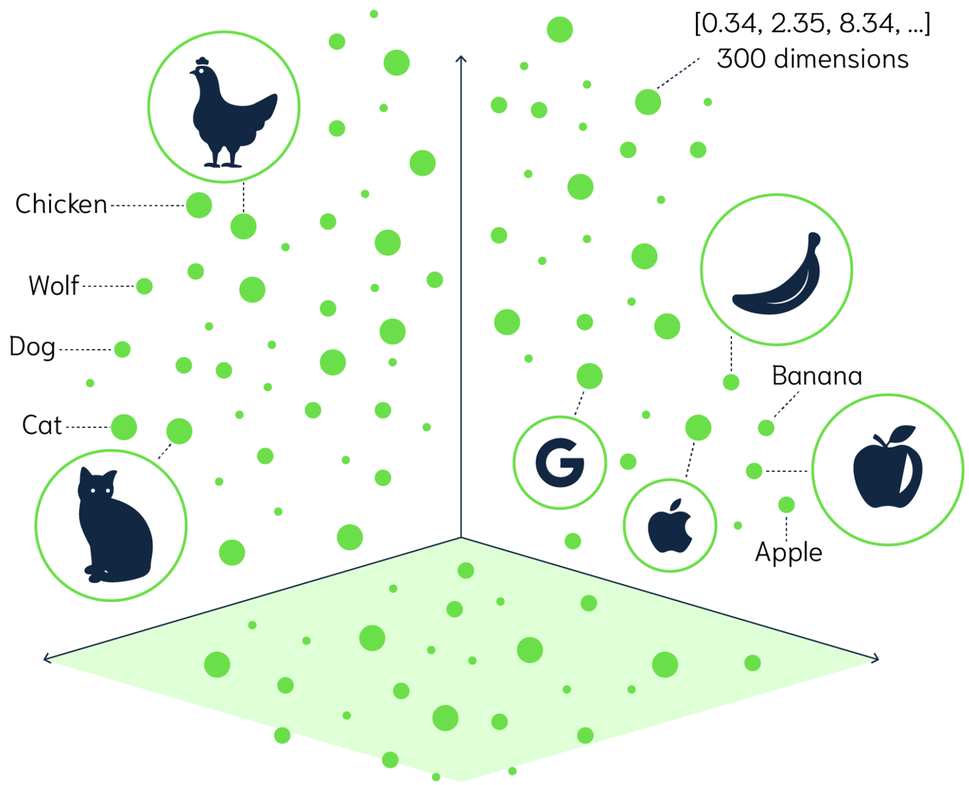
- Voor het eesrt gebruikt in Word2Vec
- Dit is een vorm van self-supervised learning die we ‘autoregression’ noemen
- “You can know a word by the company it keeps”
- Kan met woroden, zinnen, masks etc etc.



Self-supervised learning in Vision with OpenAI's CLIP.



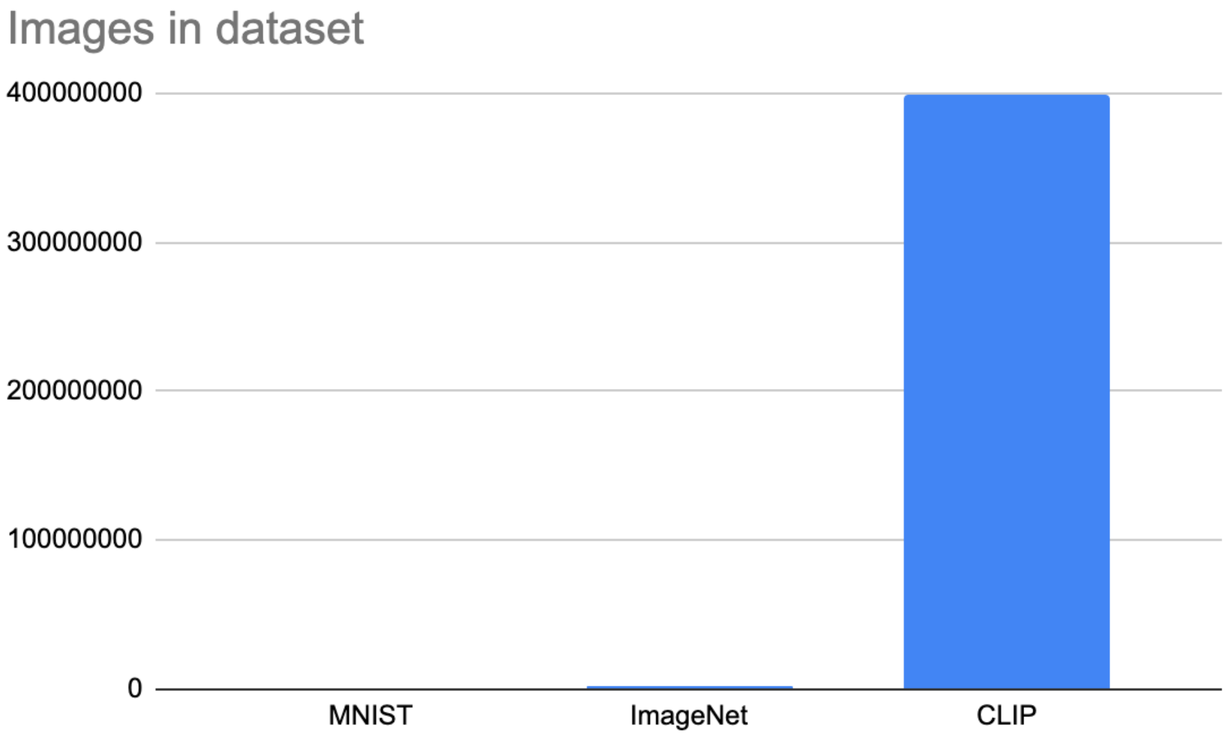
Text en image in dezelfde ’latent space’



Woorden en afbeeldingen met dezelfde betekenis staan in de ‘latent space’ dicht bij elkaar.

Self-Supervised Learning vraagt grote hoeveelheden data

MNIST	70.000 images
ImageNet	1.400.000 images
CLIP	400.000.000 images



morrow

Summary: 3 types of learning in CV

Unsupervised

Patterns and structures

K-means
(Iris dataset)

Supervised

Features

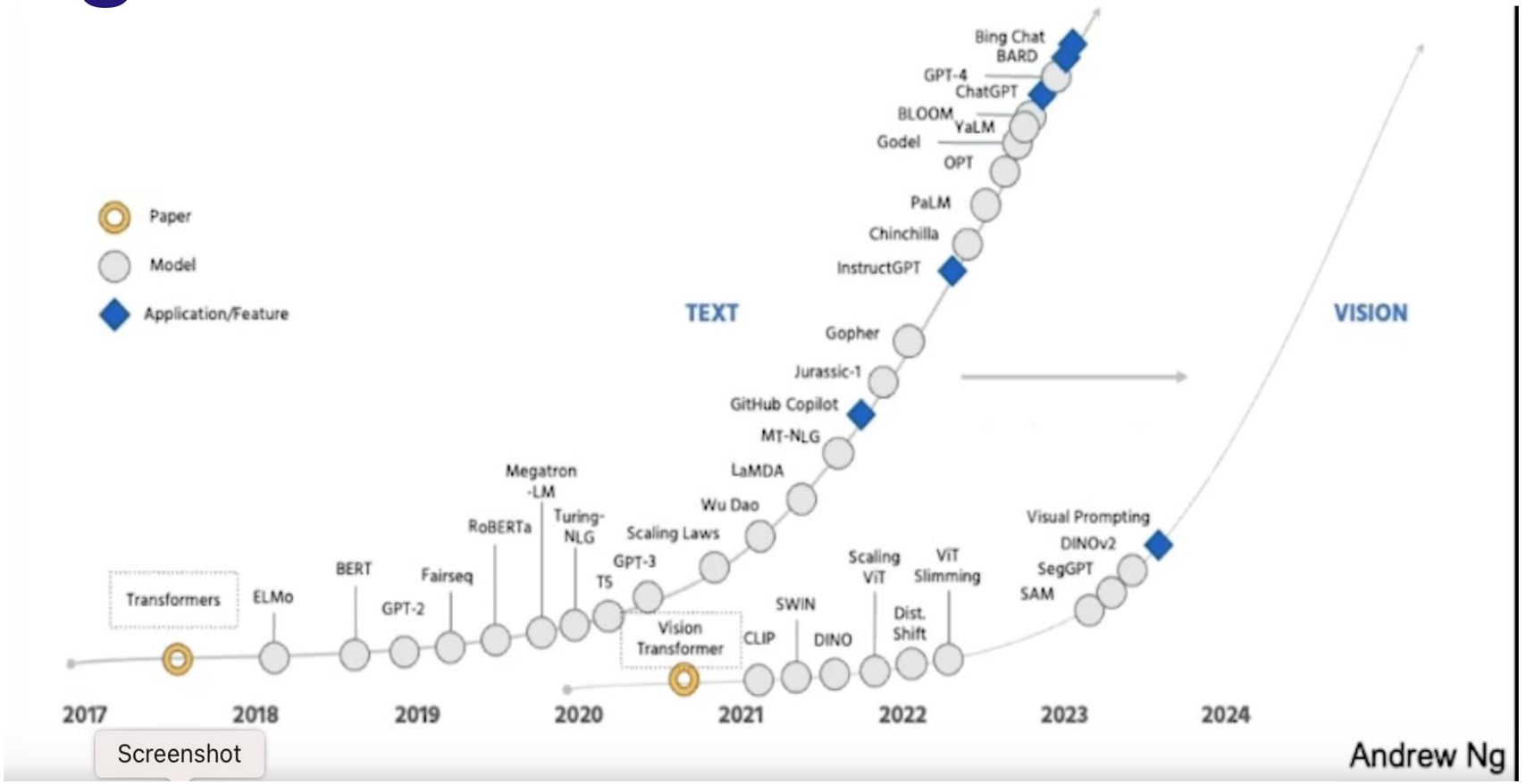
CNN
ResNet
YOLO
Autoencoder

Self Supervised

Tasks

CLIP
VLM's
Diffusion

The text (ChatGPT) revolution is coming to vision!

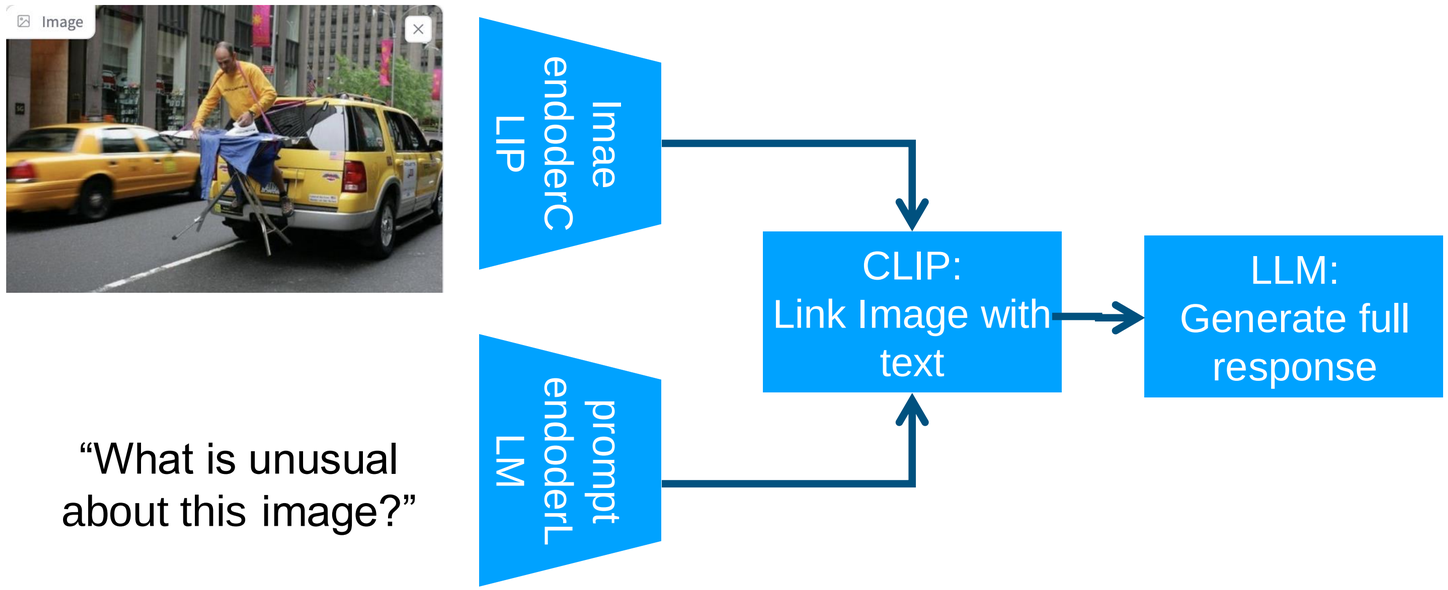


Screenshot

Large Vision Models a.k.a. Large Multimodal Models

Vision Language Model

Combineer CLIP met een LLM



Florence 2

Florence-2 is 'vision foundation model’

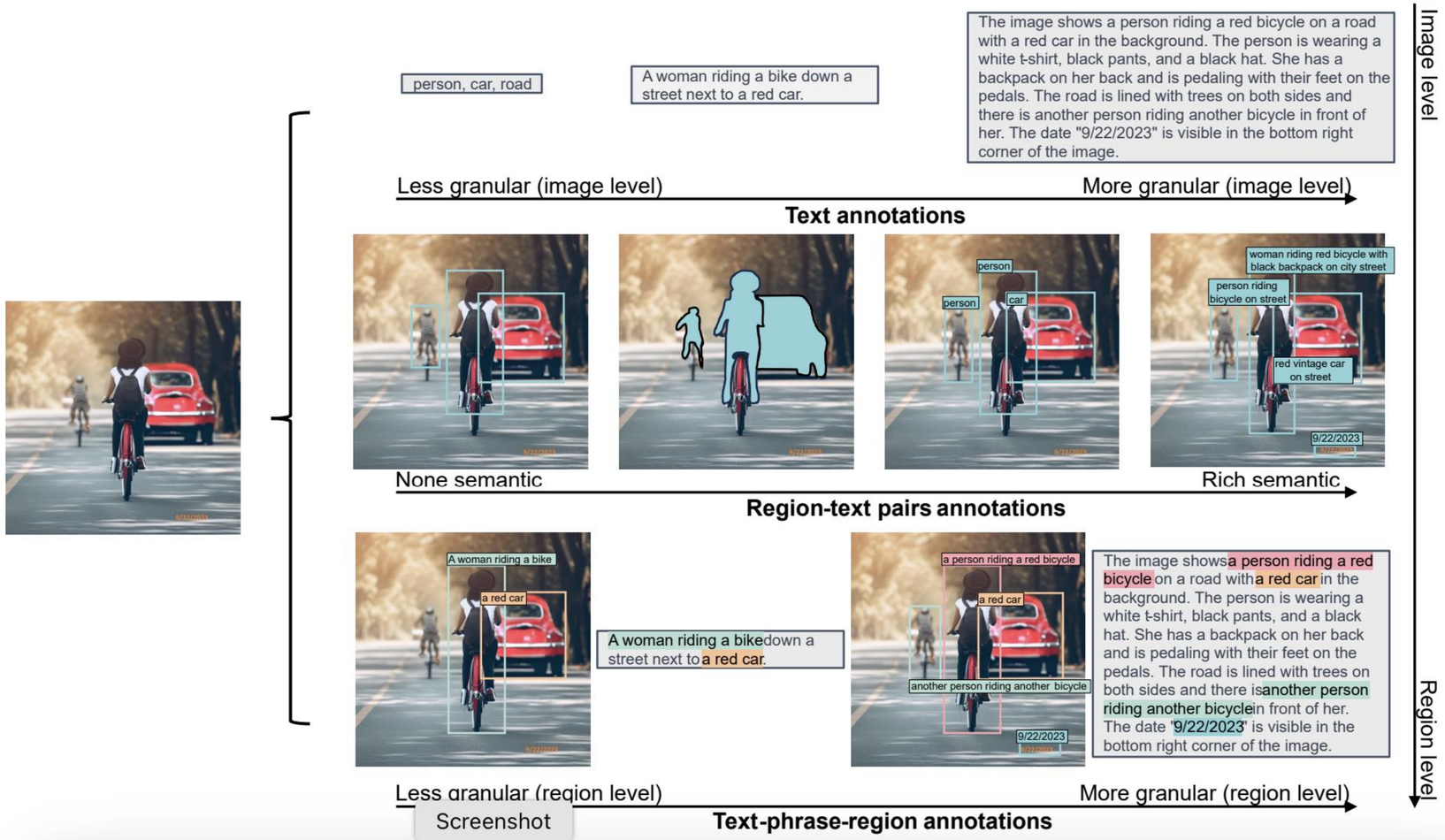
Eén model voor meerdere taken.

Een klein model in twee maten:
- Base 0.2B params
- Large = 0.7B params

Microsoft juni 2024

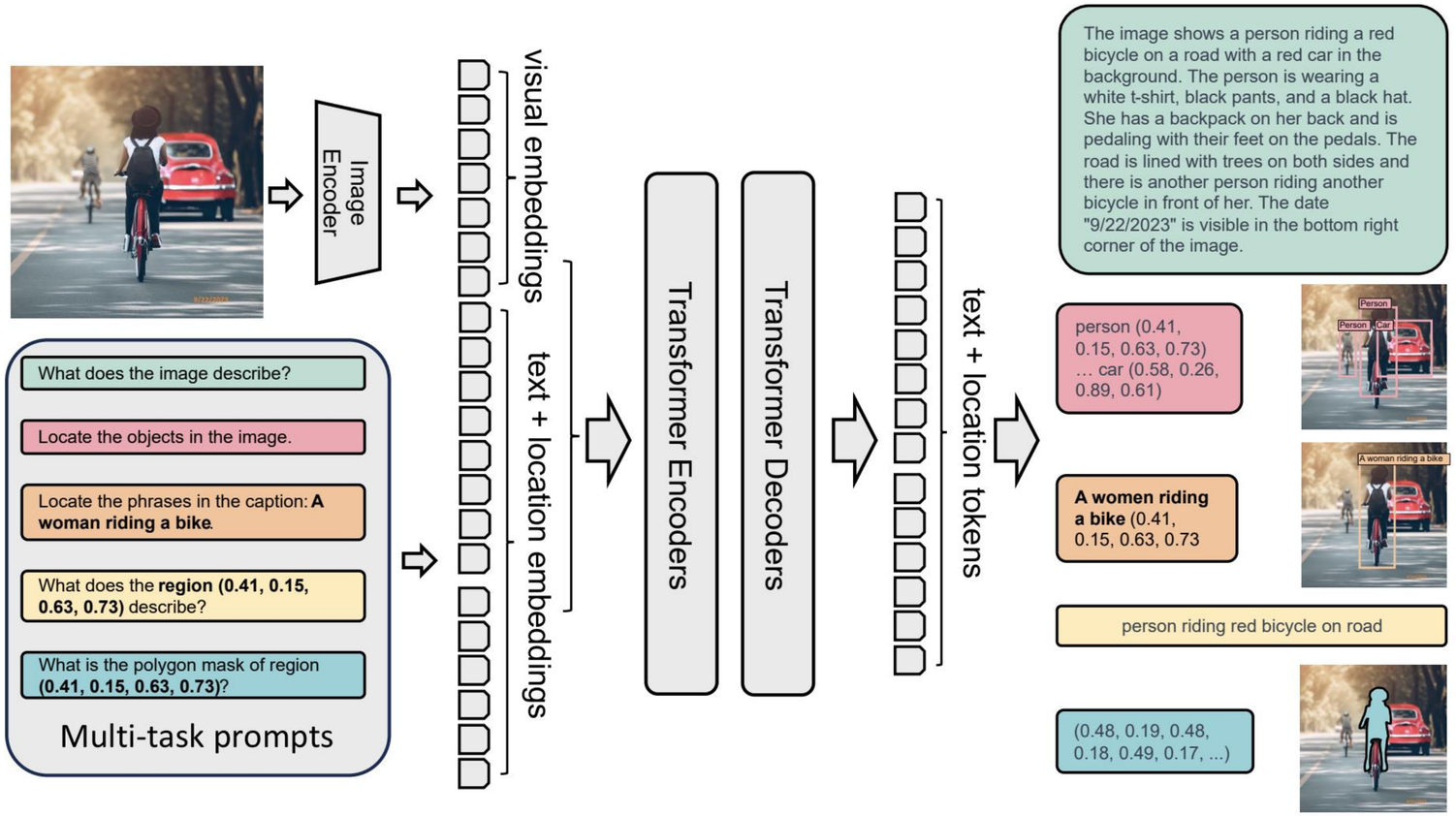
- Source: https://huggingface.co/microsoft/Florence-2-base
- Paper: https://arxiv.org/pdf/2311.06242

1 model voor meerdere taken



Screenshot

Florence 2 architectuur



Notebook Florence 2

Gebruik het notebook:

`24_10_24_Florence2.ipynb`

Bijlage: Welke VLM gebruiken?

🤗 Leaderboard :

- https://huggingface.co/spaces/opencompass/open_vlm_leaderboard

Meer lezen:

- https://github.com/gokayfem/Awesome-VLM-Architectures

BONUS

Running quantised models with ollama

Quantised models

- Say, you want to run LlaVa on your machine
  - This model is 15Gb large!
  - You need a very good laptop, even might need a GPU and it is still slow

  - Solution: use a smaller version of this model
    - We call this a ‘quantised model’ => it’s been made smaller by:
      - Quantisation => changing datatype of embeddings from float32 to float16 or Int8 or Int4
      - Pruning => removing unused nodes
      - Other tricks

ollama

- Ollama is een tool waarmee je lokaal LLM’s kan runnen
- Ook LLaVA kan je hier mee runnen.

- Download en installeer via www.ollama.com (Mac / Windows / Linux)

- Start het vanaf de CLI:

```
ollama run tinyllama
```

- Stel je vraag

- Quit met CTRL+C of /bye

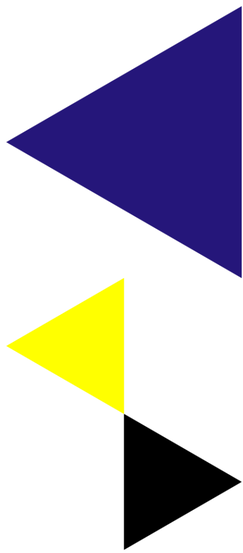
Ollama python package

```
pip install ollama
```

Gebruik het bijgeleverde notebook om tegen je lokale ollama aan te praten.

`ollama.ipynb`

N.B. Ook als Node.js package: `npm install ollama`



Meer lezen over ollama

Het originele project om llama om te zetten naar C++ code:

- https://github.com/ggerganov/llama.cpp

- Quantization of you own models
- https://www.youtube.com/watch?v=mNE_d-C82lI