Hogeschool van Amsterdam

minor AAI - Bootcamp Computer Vision les 2

Michiel Bontenbal & Maarten Post
Vrijdag 13 september 2024

Tijd: 10:20 tot 12:50

Acknowledgements: Stijn Oomes

Creating Tomorrow

Computer Vision 2

- 1e uur
  - Theorie: Camera
  - Notebook OpenCV
- 2e uur
  - Theorie speciale camera’s & images
  - Notebook NumPy
- 12:00 – 12:30 Renata en Masa over Ethiek
- 13:40 – 17:00 : Presentatie projecten (David / Marcio)

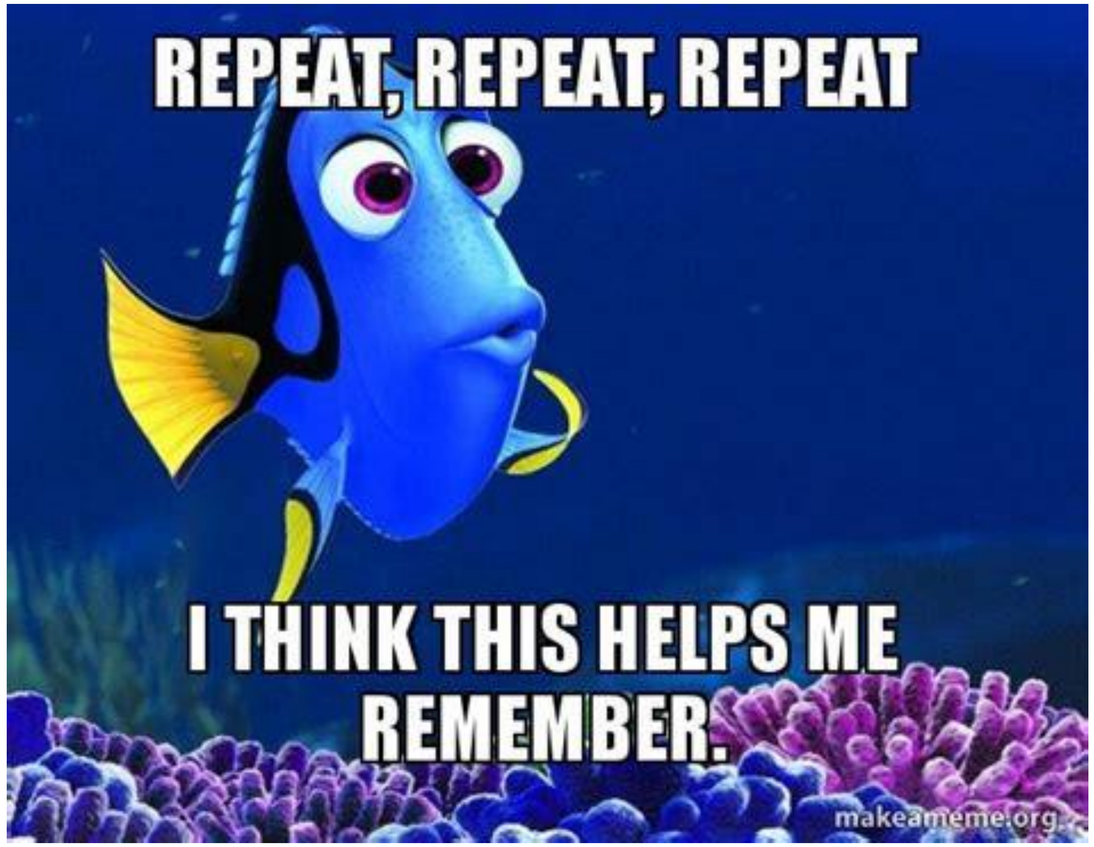
Computer Vision lessen

- **Computer Vision 1: (vorige week)**
  - Convolutional Neural Networks
  - Image embeddings
  - Vision Language Models met ollama

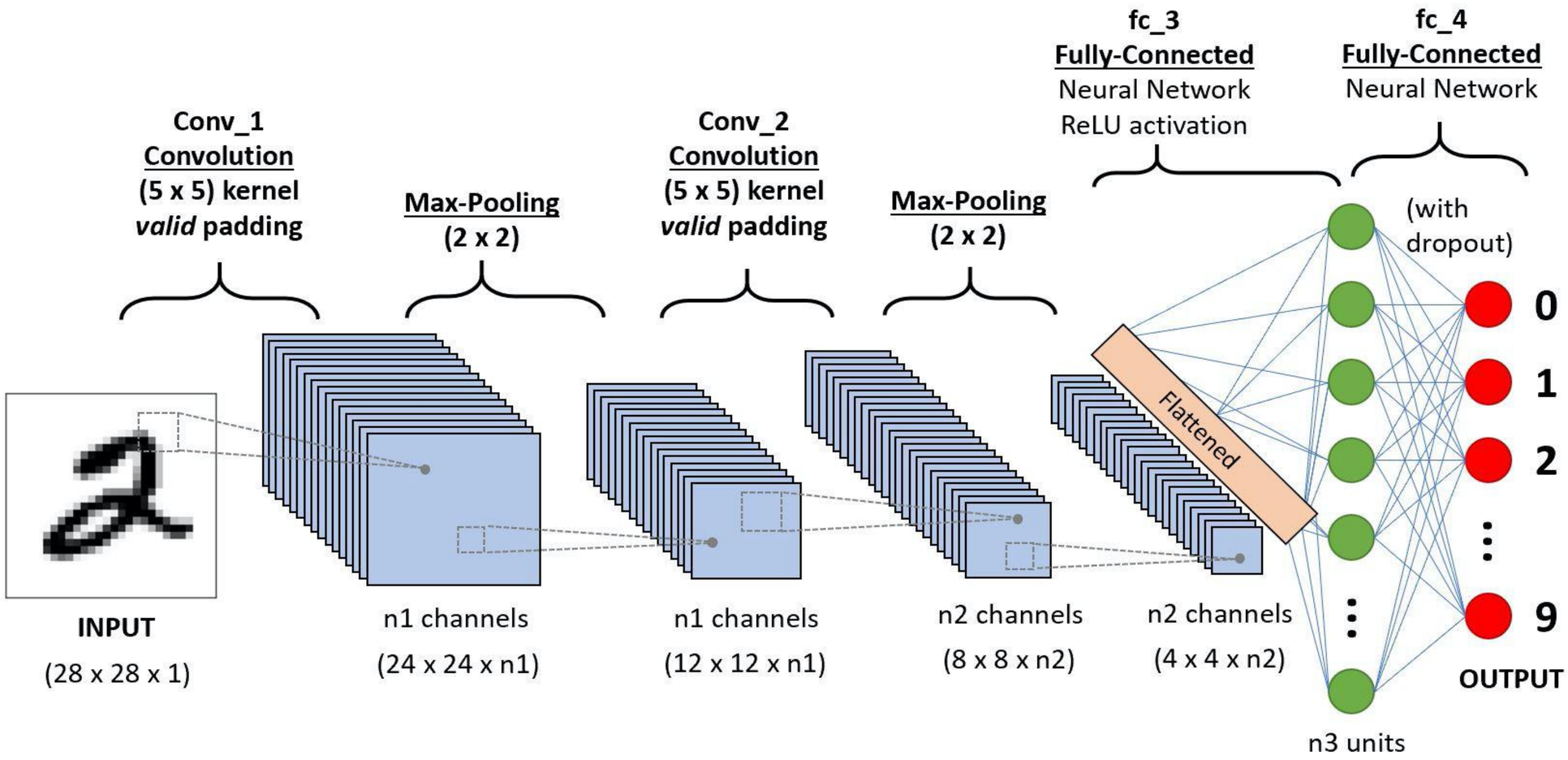
- **Computer Vision 1: (vandaag)**
  - OpenCV -> pre-processing van images & gezichtsdetectie
  - Numpy -> images omzetten naar data

- **Computer Vision 3 + 4: (16 + 23 oktober)**
  - Generative AI voor Images
  - Object detectie, segmentatie
  - Foundation models

Recap les 1


REPEAT, REPEAT, REPEAT
I THINK THIS HELPS ME REMEMBER.
makeameme.org

Convolutional Neural Network



Features herkennen

Neural net voor classificatie

We use an ‘embedding model’ and compare the vectors. Vectors capture the meaning.

‘Paris’

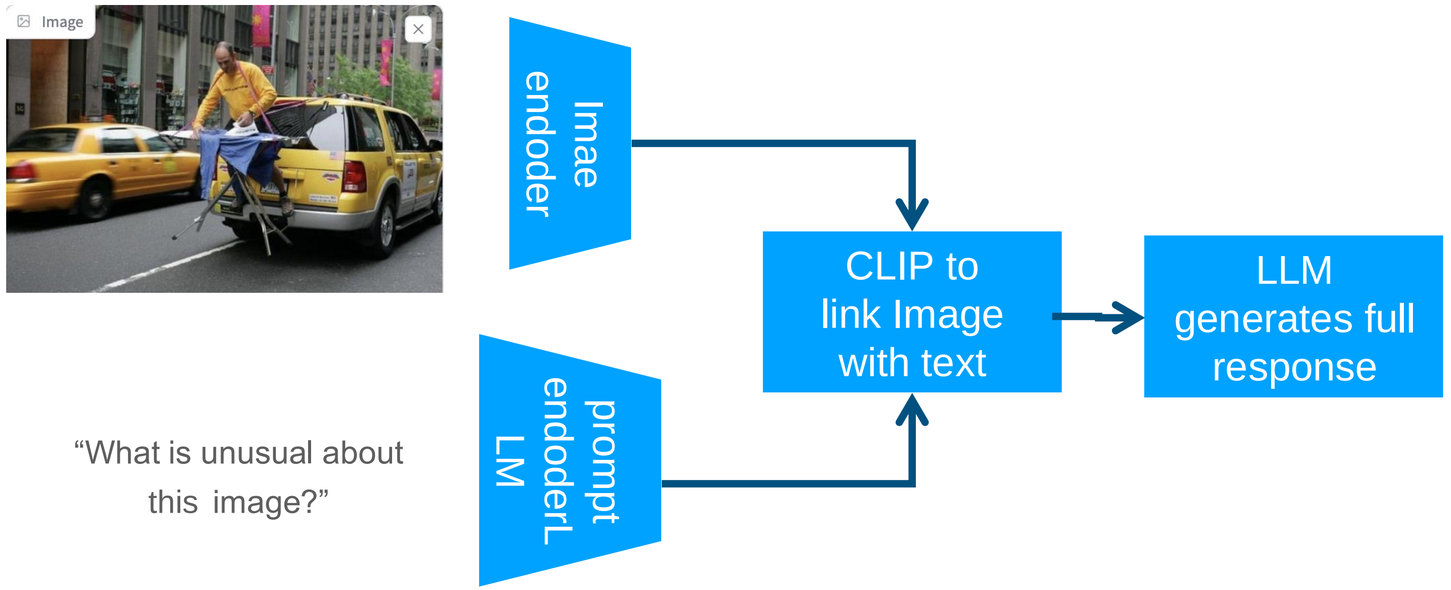
[0.5, 0.3, …, 0.7]

[0.4, 0.3, …, 0.6]

Compare vectors

LLaVA: Vision Language Model

Combines **CLIP with a Large Language Model**



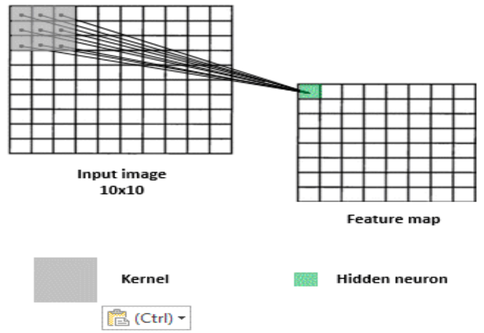
This is also called a ‘neck & head architecture’.

AI is een combinatie van code, wiskunde en concepten

```
cnn.add(Conv2D(filters=32,
               kernel_size=(3, 3),
               activation='relu',
               input_shape=(28,28,1),
               strides=(2, 2)))
```

$$f(x) * g(x)$$

($*$ convolution operator)

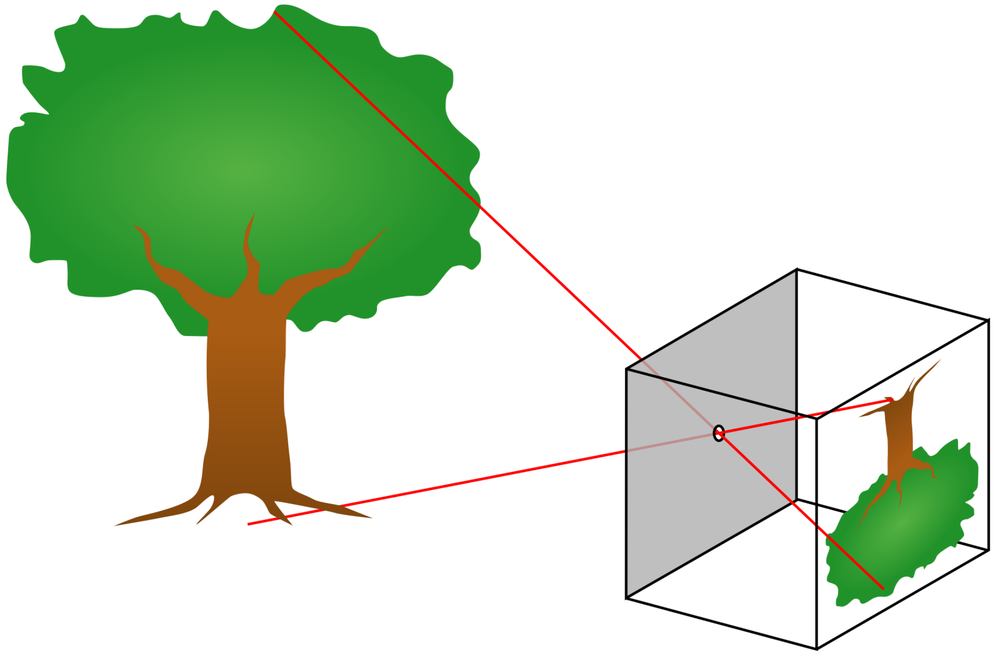


Hier de code, de wiskunde en het concept van een convolutie operatie gebruikt in CNN’s.

Camera

- Vervorming door camera

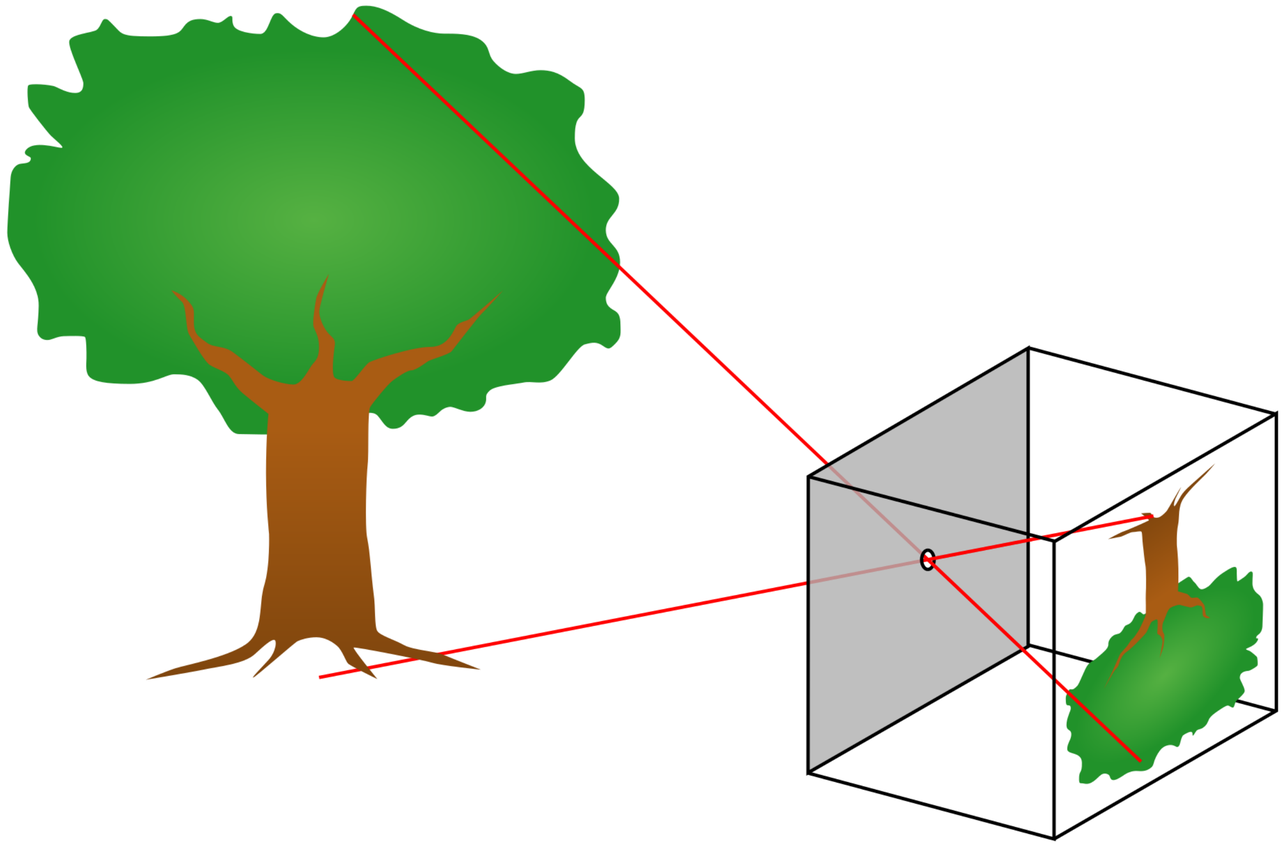
- Pinhole camera
Afstanden meten



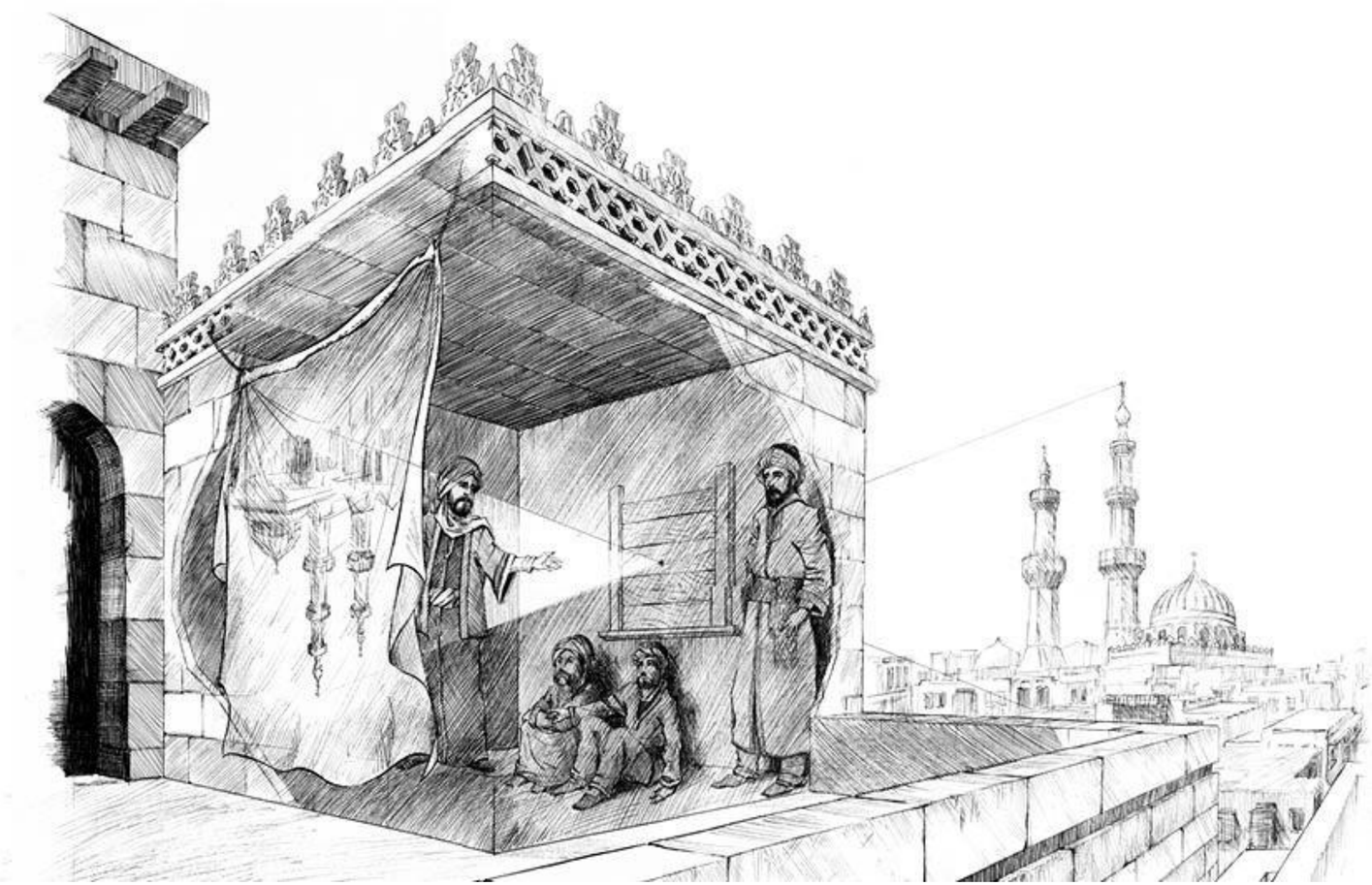
Wat doet een camera?

- camera detecteert licht
- zet het om in een “image” (afbeelding, foto)

Pinhole camera. Lichtstralen gaan in rechte lijn door gaatje.



“Camera obscura"

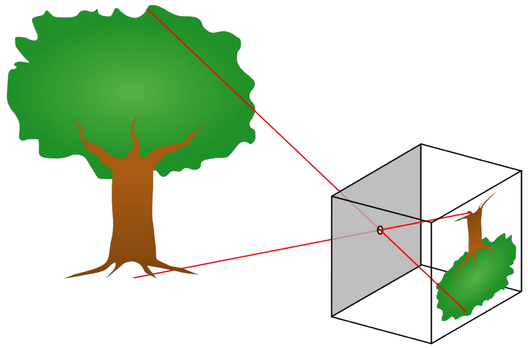


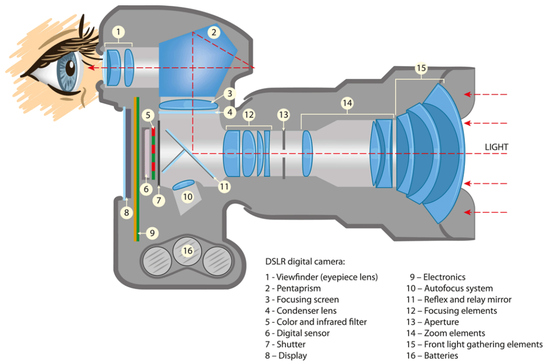
Plaatje ter iluustratie

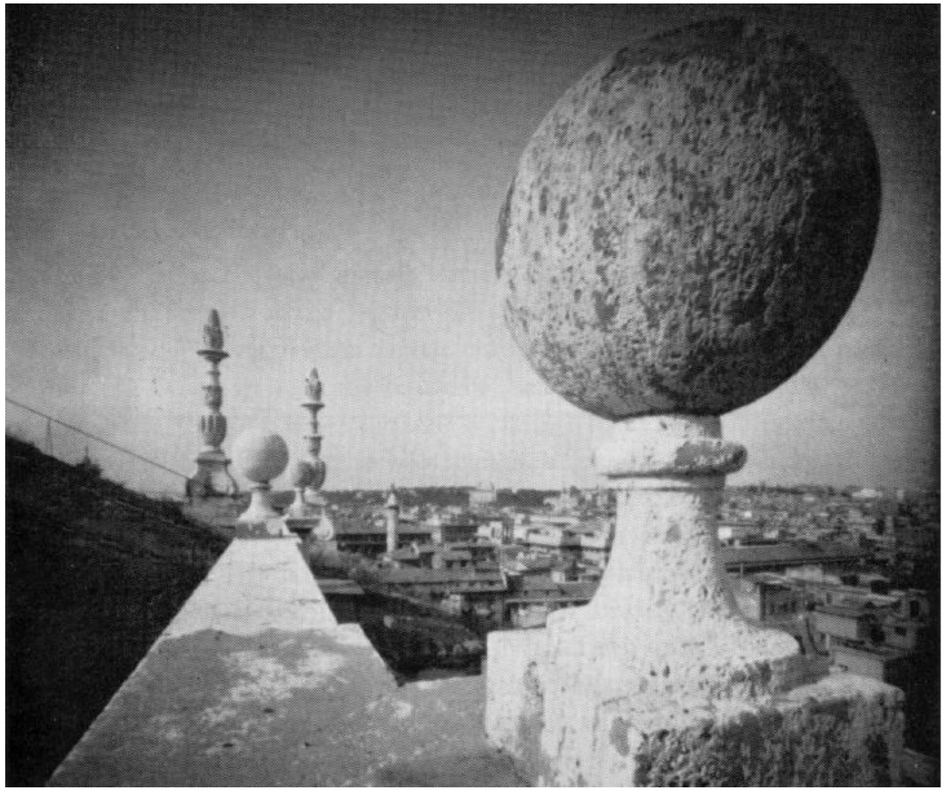
Camera obscura voor het eerst beschreven door

Abu Ali al-Hasan ibn al-Haytham in 1021.

Camera vervormt het beeld in hoek

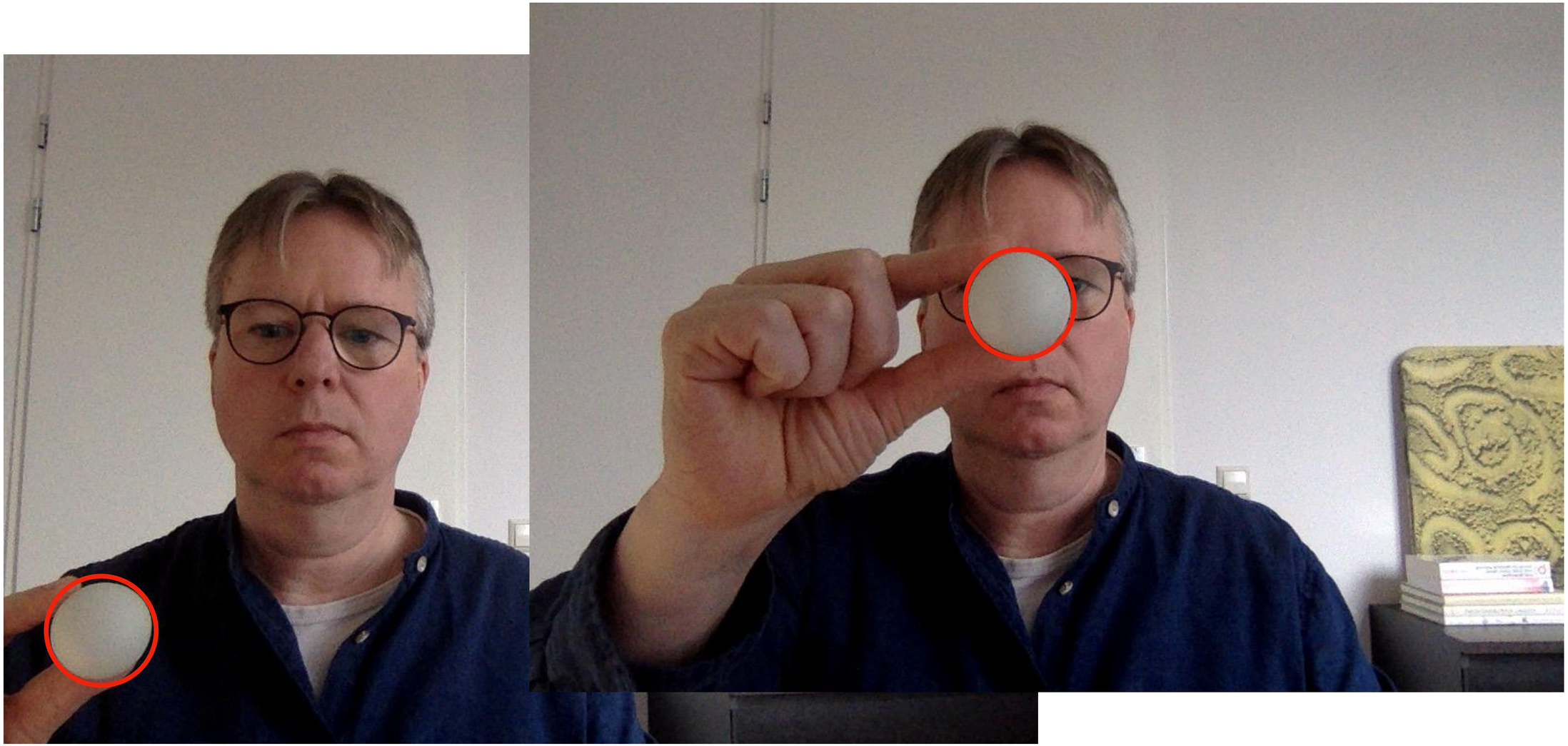




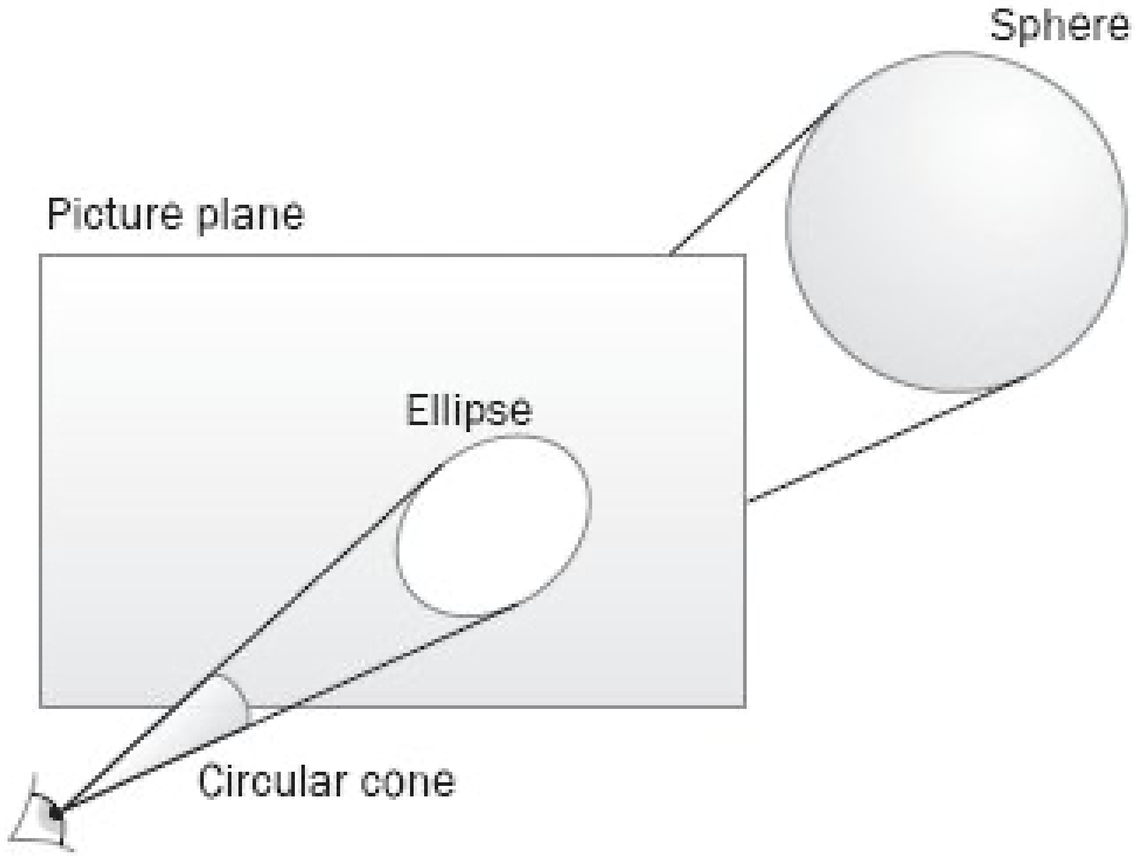


Bolle lens!

Bal wordt ellipsvormig in de hoek

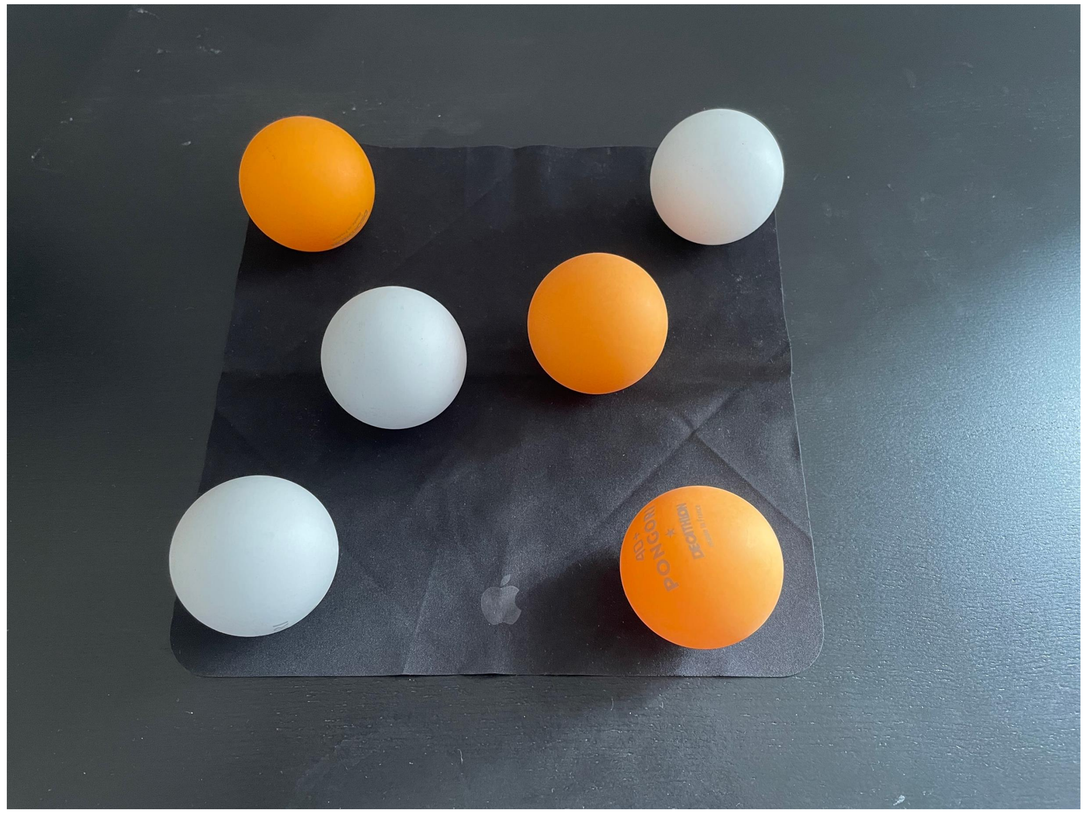


Pinhole camera image

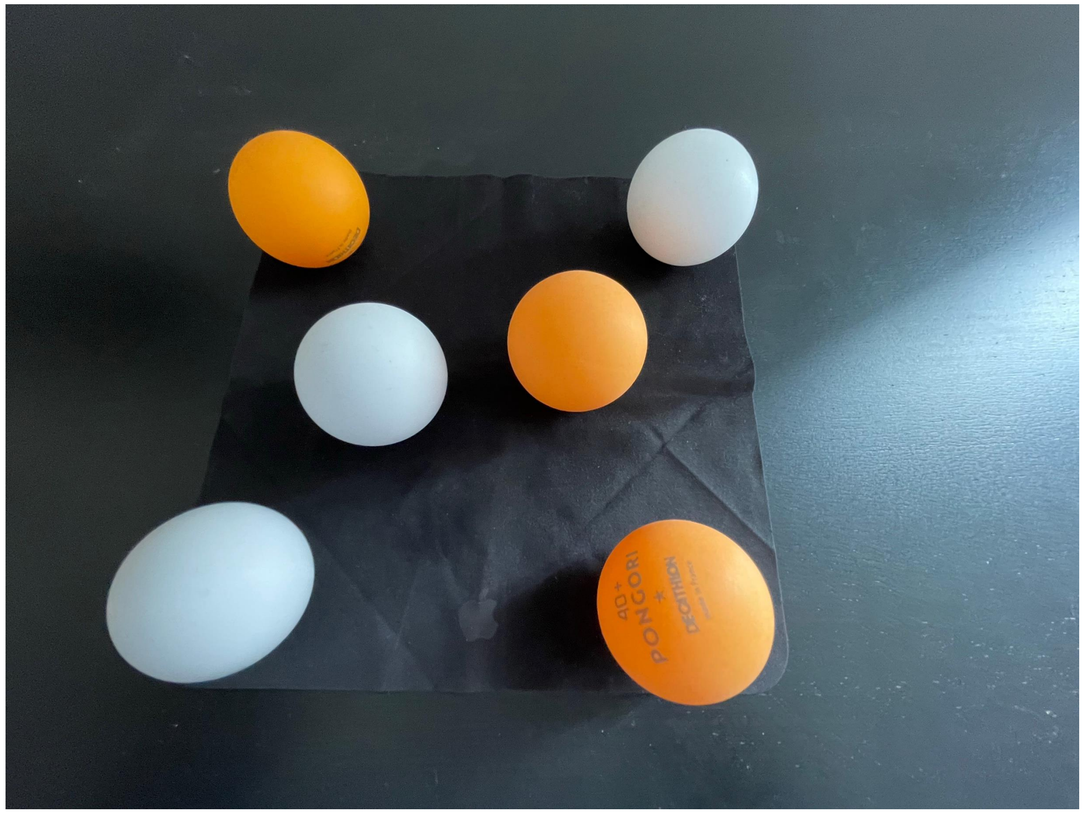


Bol wordt een ellips in de hoek van de afbeelding

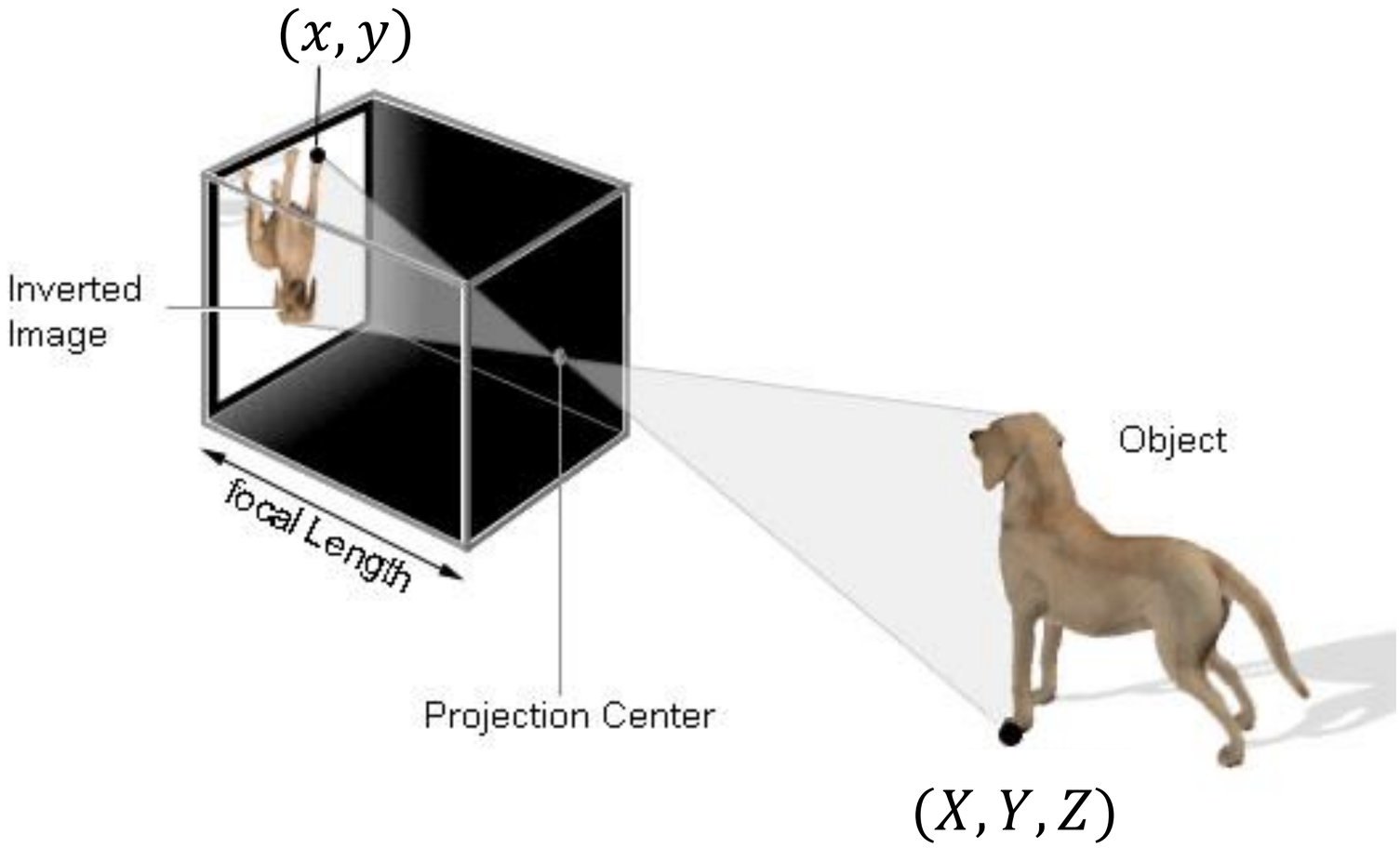
Normale lens (67° x 53°)



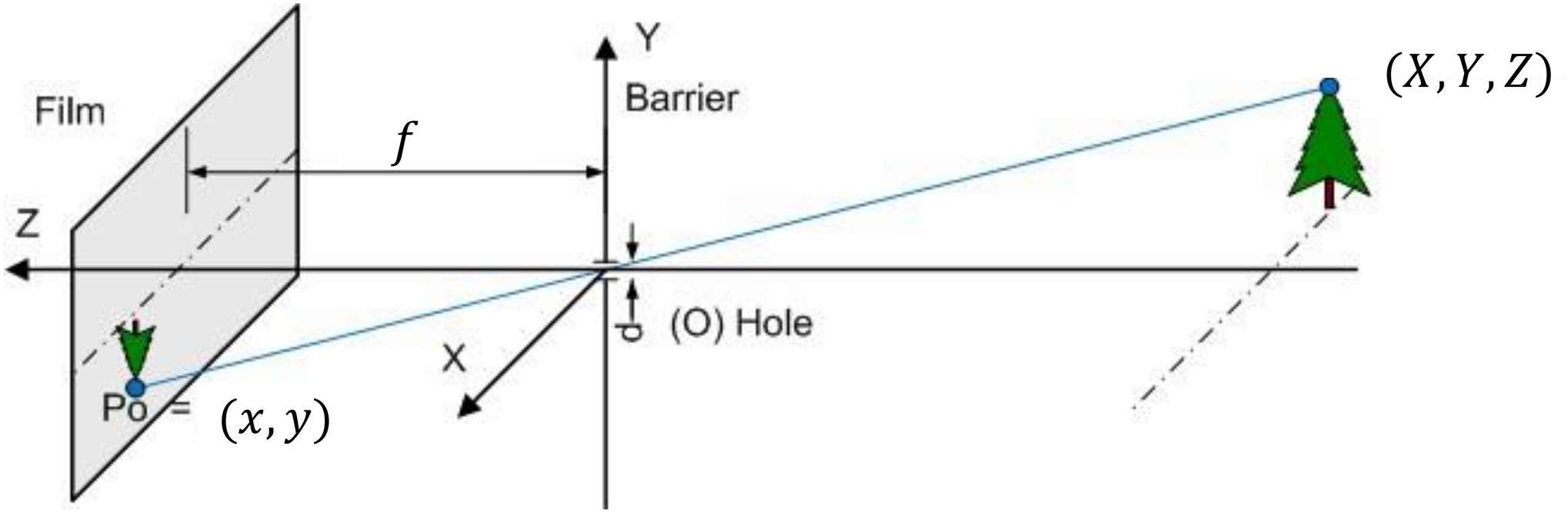
Groothoek lens (101° x 85°)



Camera

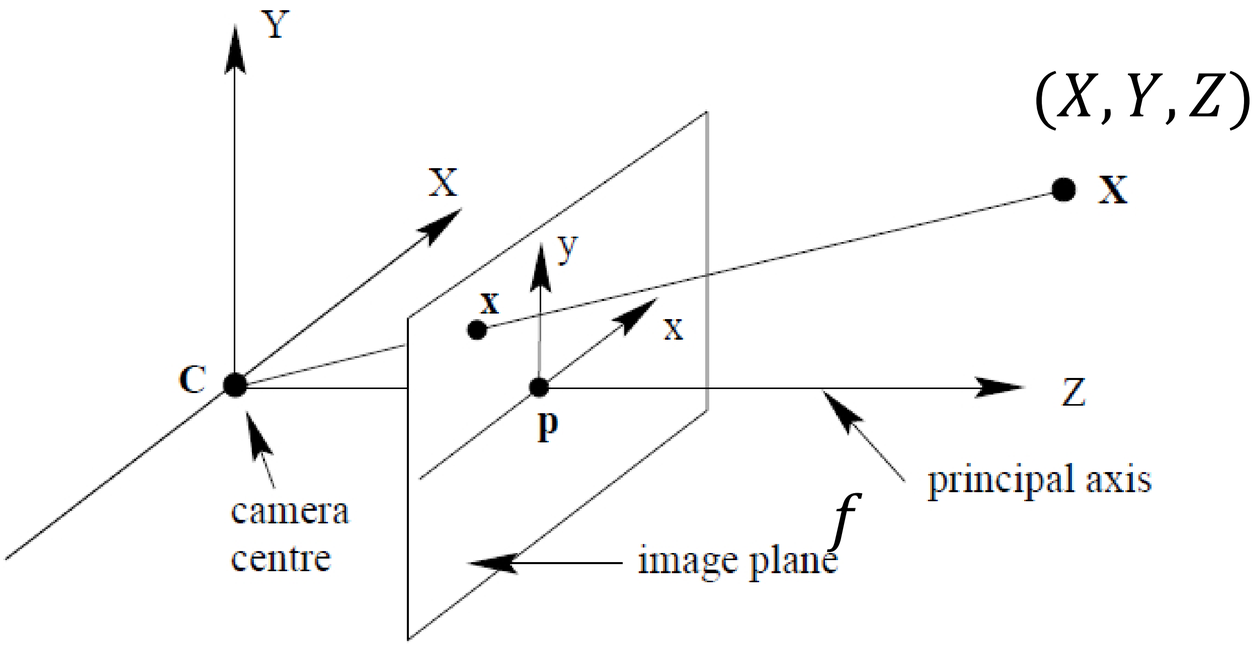

(x, y)
Inverted
Image
focal Length
Projection Center
Object
(X, Y, Z)

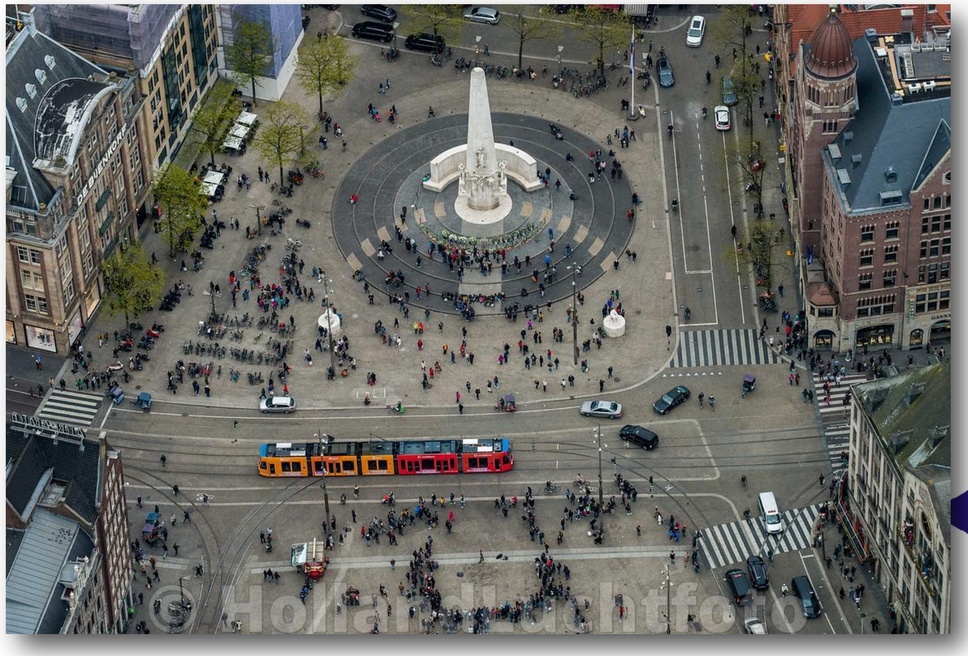
Camera model



$$x = -\frac{fX}{Z}, \quad y = -\frac{fY}{Z}$$

Bereken hoogte van objecten in foto's

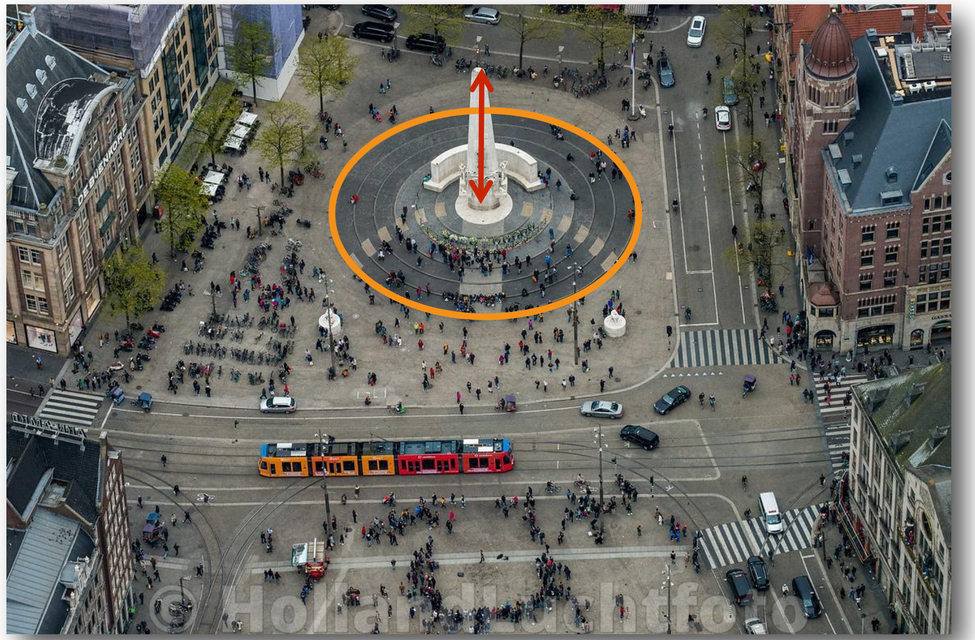




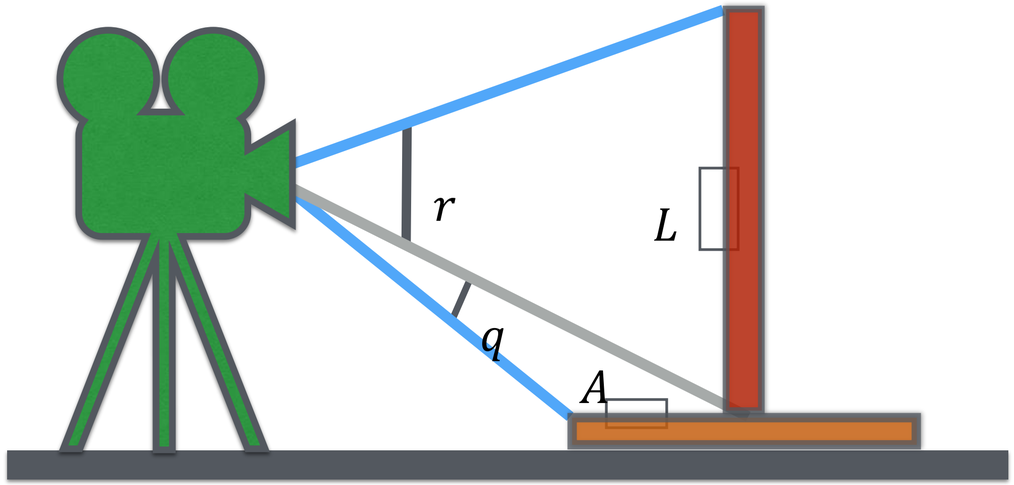
Tutorial van onze collega Stijn Oomes:
https://www.youtube.com/watch?v=2l__8MQJ4AA

Hoogte van monument

$$L = \frac{(1-p)^2 qr}{[1-(1-p)q][pr + (1-p)q(1-2r) - 2\sqrt{(1-p)q(p-(1-p)q)r(1-r)}]} A$$



directe meting van hoogte
22.0 m



visuele voorspelling van hoogte
22.7 m

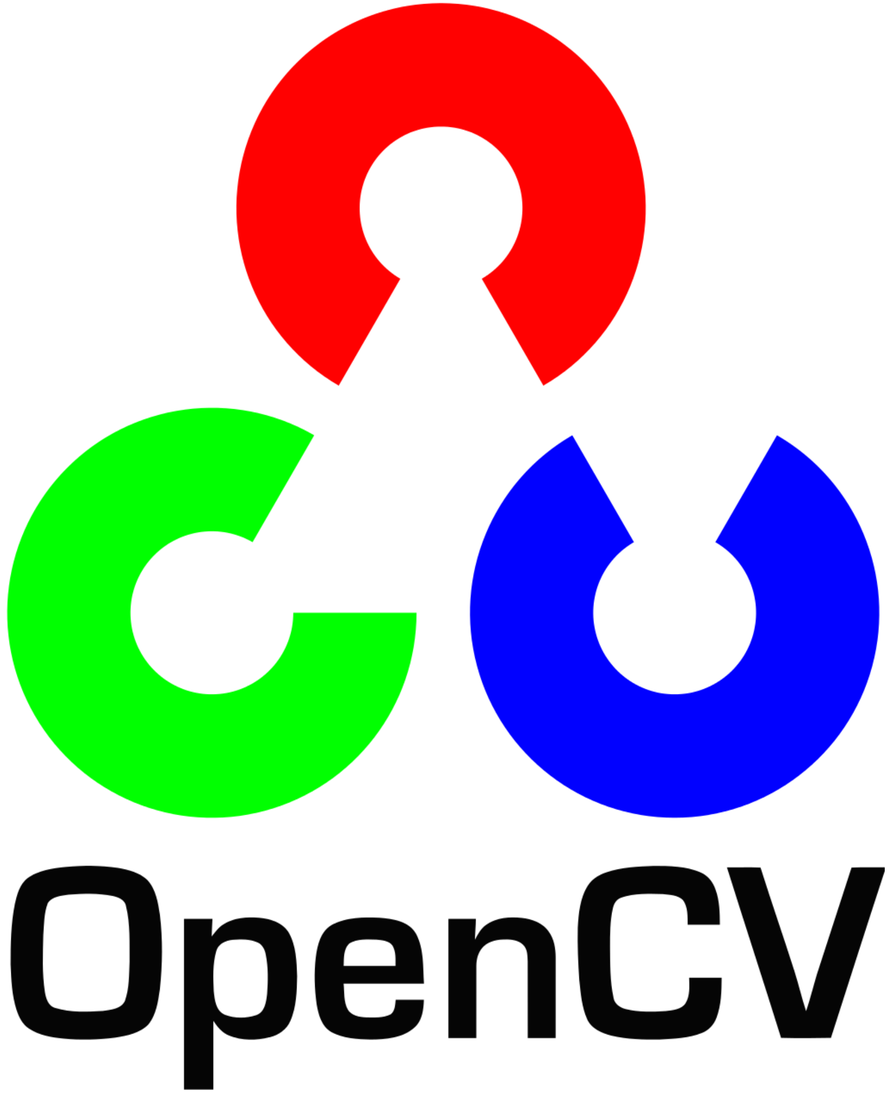
www.stijnoomes.com

OpenCV

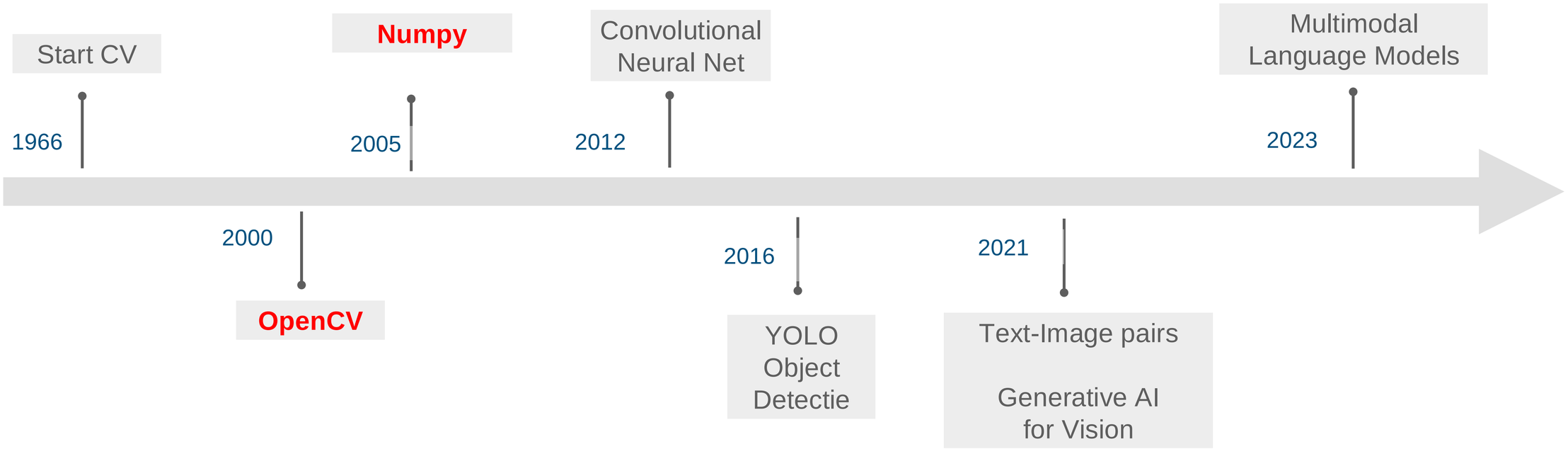
- Intro

- Kernel operaties

- Gezichtsdetectie



Tijdslijn Computer Vision (CV)



OpenCV

- Een verzameling van tools en algoritmes voor (basic) Computer Vision.
- Gestart in 2000 in C++
- Wij gebruiken de opencv-python library.
- Verouderd maar nog steeds gebruikt: inmiddels versie 4.10 (june 2024)

- Moderne computer vision voor classificatie of object detectie doen we Niet met OpenCV
  - We gebruiken een neural net en libraries als Keras of Pytorch.

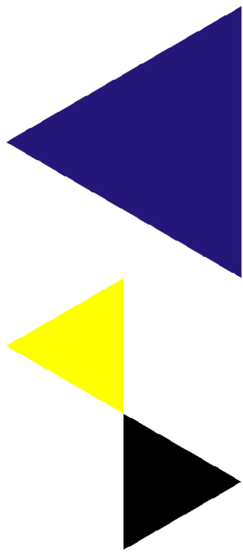
Pre-processing

Preprocssing gebruik je om images geschikt te maken voor een CV taak.

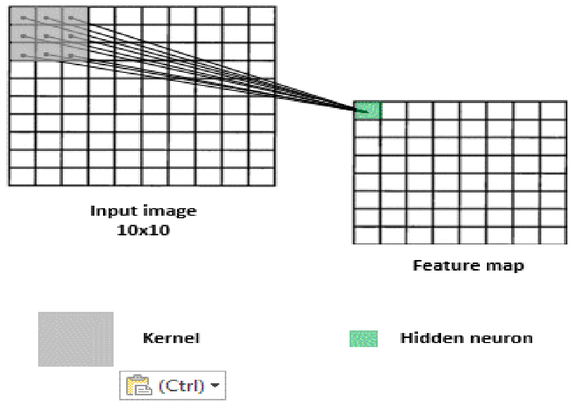
Pre-processsing:

- Kleurwaarden (RGB) omzetten naar Grijswaarden
- Kleurcorrectie en versterken van kleuren, false colours
- Resizing => alle images zelfde formaat
- Image Augmentation => flippen of spiegelen
- Noise reduction => bijv smoothing of filtering

- Gaan we nu doen met OpenCV en daarna met Numpy



Kernel operation



With kernel you can:

- Blurring: Smooths out noise and details in an image
- Sharpening: Enhances edges and fine details
- Edge detection: Identifies boundaries of objects in an image
- Embossing: Creates a 3D-like effect

Gebruik van een kernel

Gaussian blur

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Edge detection

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

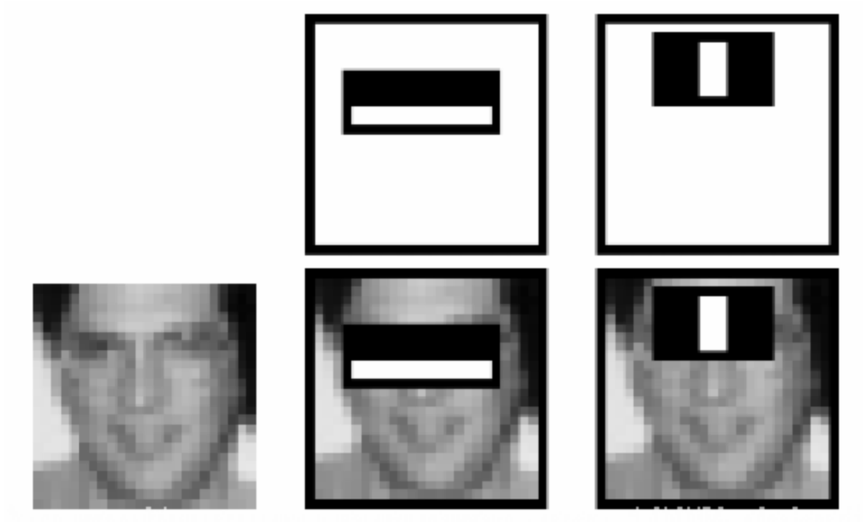
Source: https://en.wikipedia.org/wiki/Kernel_(image_processing)

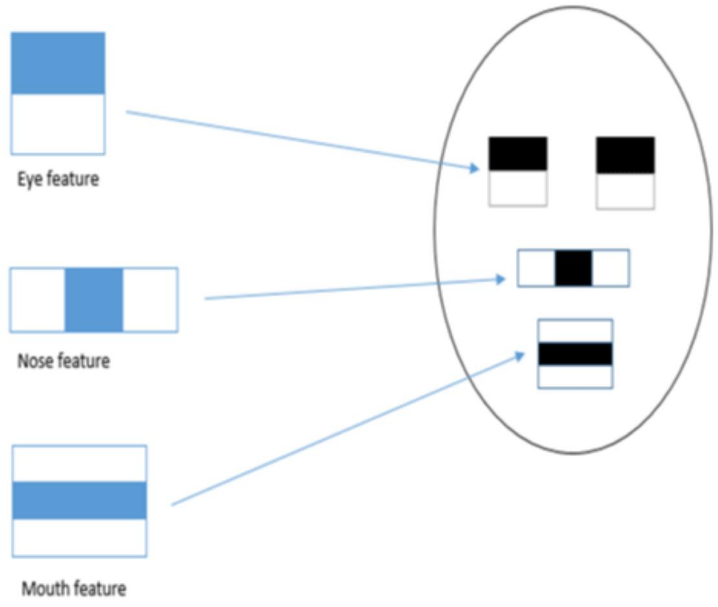
Gezichtsdetectie: twee algoritmes

- **Haar detector (2001)**
  - Weinig rekenkracht nodig, maar wel verouderd.
  - OpenCV
  - Vernoemd naar meneer Haar (wiskundige)

- **Convolutioneel Neuraal Network (2012)**
  - Veel rekenkracht nodig + deep learning
  - Heel goed & wordt nog steeds gebruikt
  - Keras en PyTorch

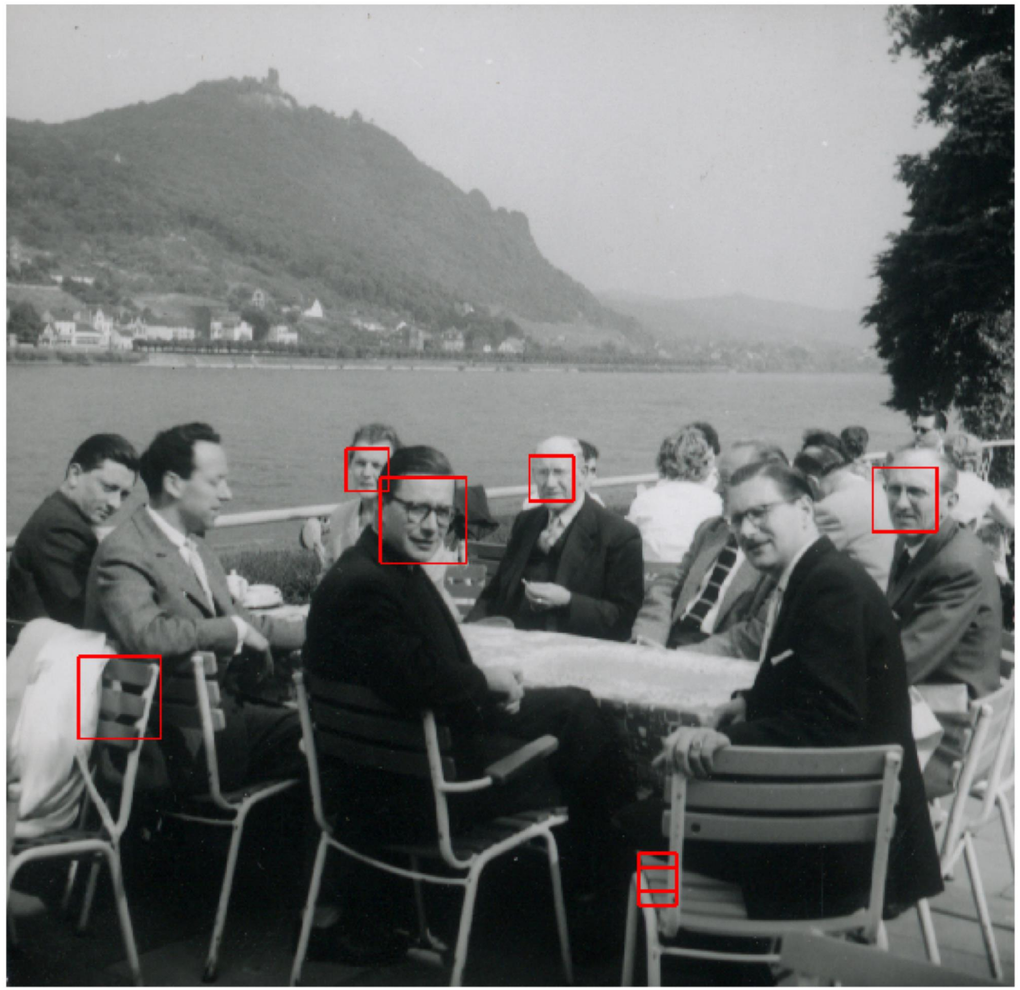
Pattern matching met Haar detector





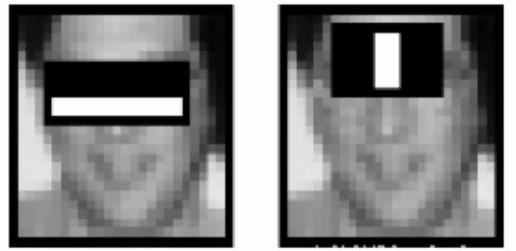
Creating Tomorrow

Resultaten zijn matig



Haar detector samenvatting

- Pattern matching => komt een bepaald patroon voor?
  - 6000 detectors zeggen samen ‘ja’
- Cascade
  - Probeer de belangrijkste detectors eerst
- Gebruikt een voorgetraind model in .xml formaat



Opdracht

- Notebook op DLO:
  - `OpenCV opdracht.ipynb`

- Gebruik de Geeks for Geeks tutorial (link in notebook)
- Maak de oefeningen

- Circa 25 minuten **tot 11:20**

Vraag: wat is verschil tussen ‘pattern matching’ en CNN’s?

- **Pattern matching**: je zoekt met een vooraf gegeven patroon over de afbeelding heen om een object te vinden.

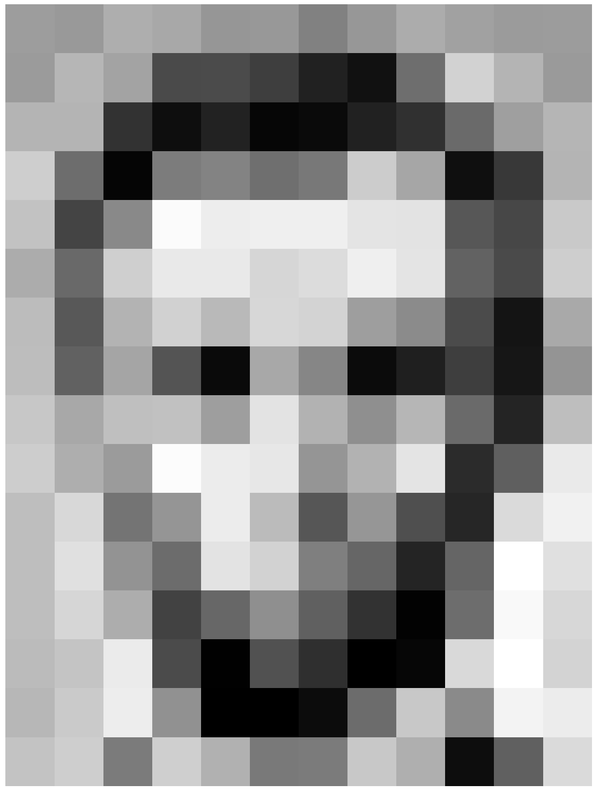
- **CNN**: Dit neuraal netwerk is in staat zelf patronen en objecten te herkennen

Ditigale foto’s

Hoe ‘werkt’ een digitale foto?

- Wat zien mensen, wat ‘zien’ computers?
- Wat is het ‘elektromagnetische spectrum’?
- Waarom zijn digitale foto’s eigenlijk ‘R,G,B’?

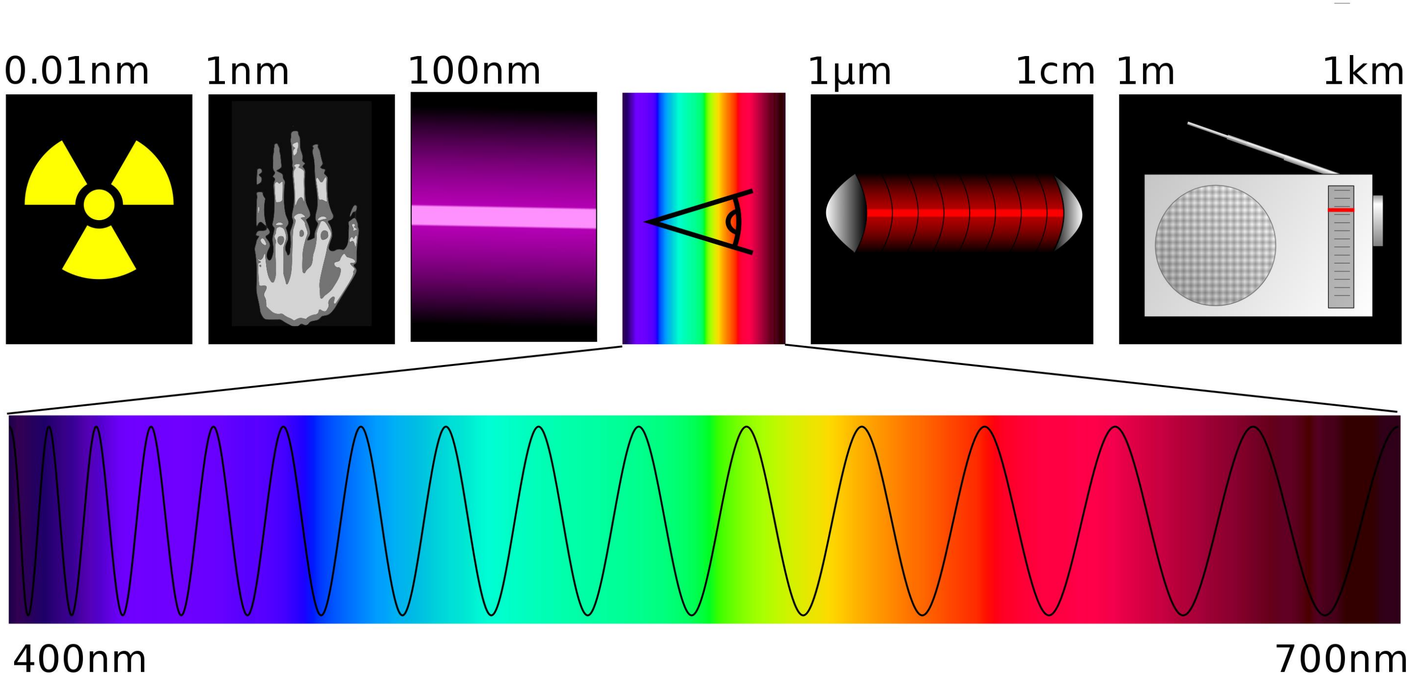
Wij zien plaatjes, computers ‘zien’ cijfers



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

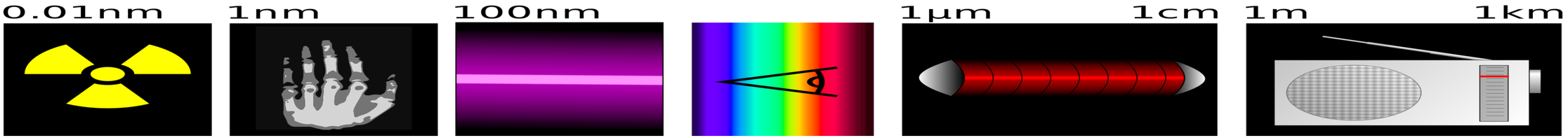
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

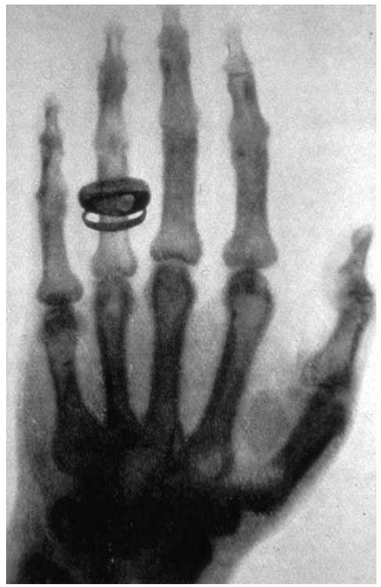
Elektromagnetische straling: onze ogen zien maar een klein deel van spectrum.

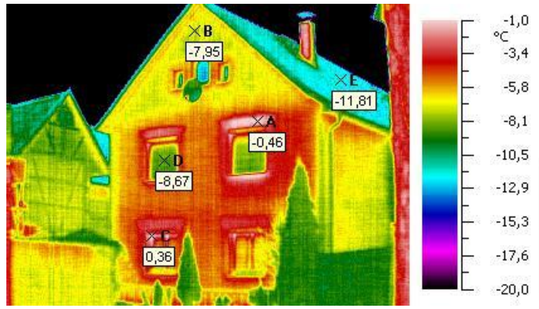


- https://nl.wikipedia.org/wiki/Elektromagnetische_straling

Camera’s buiten het zichtbare licht

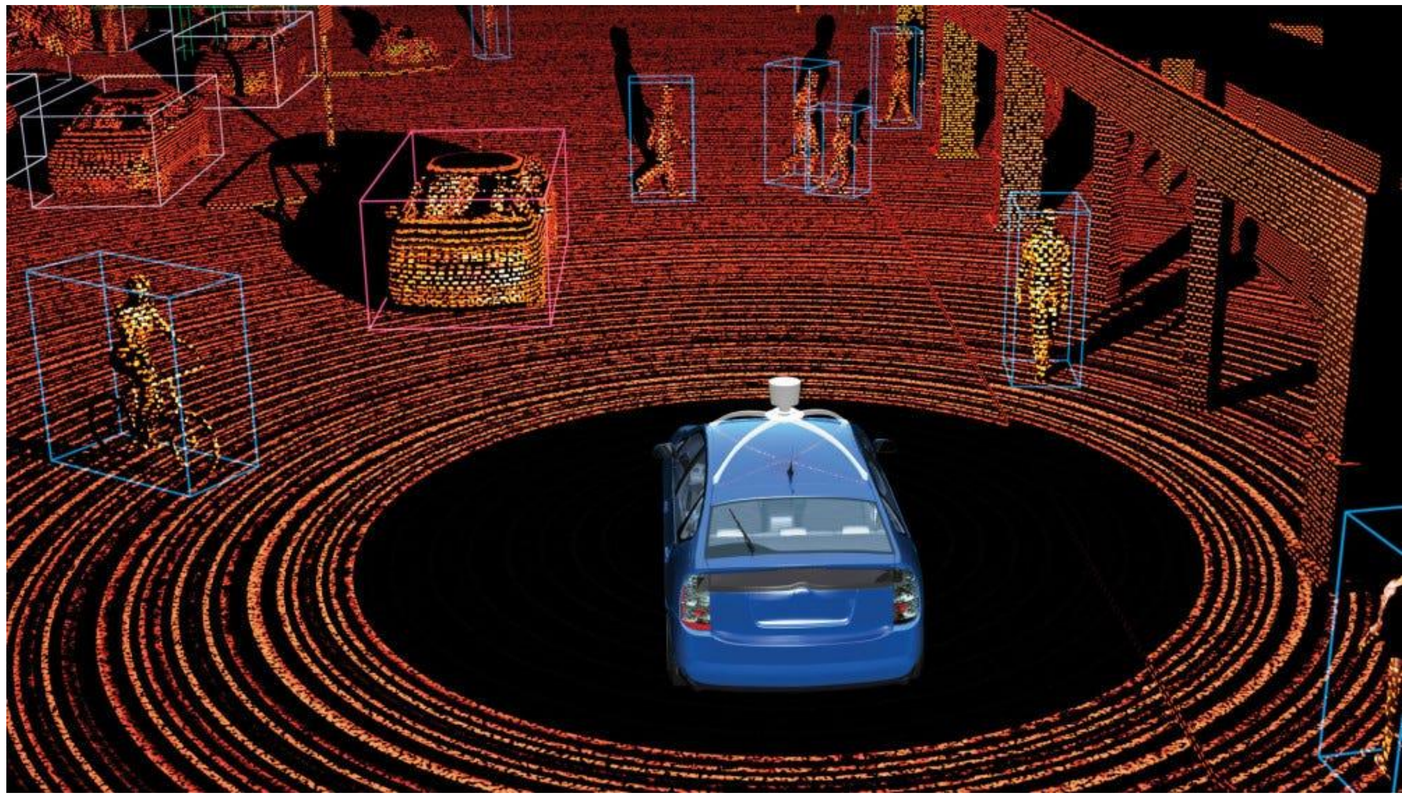




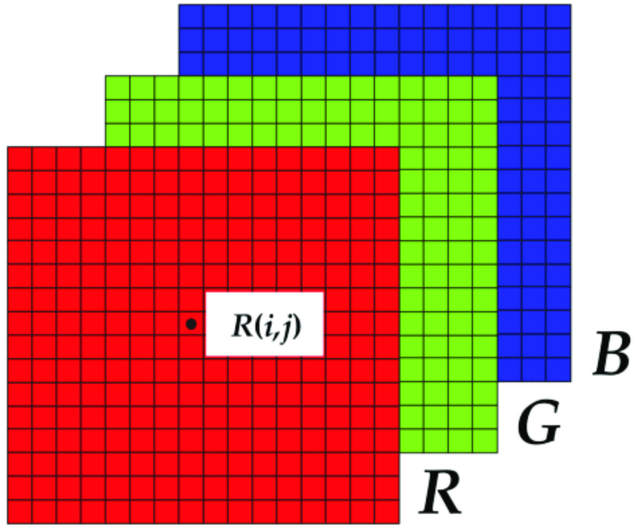


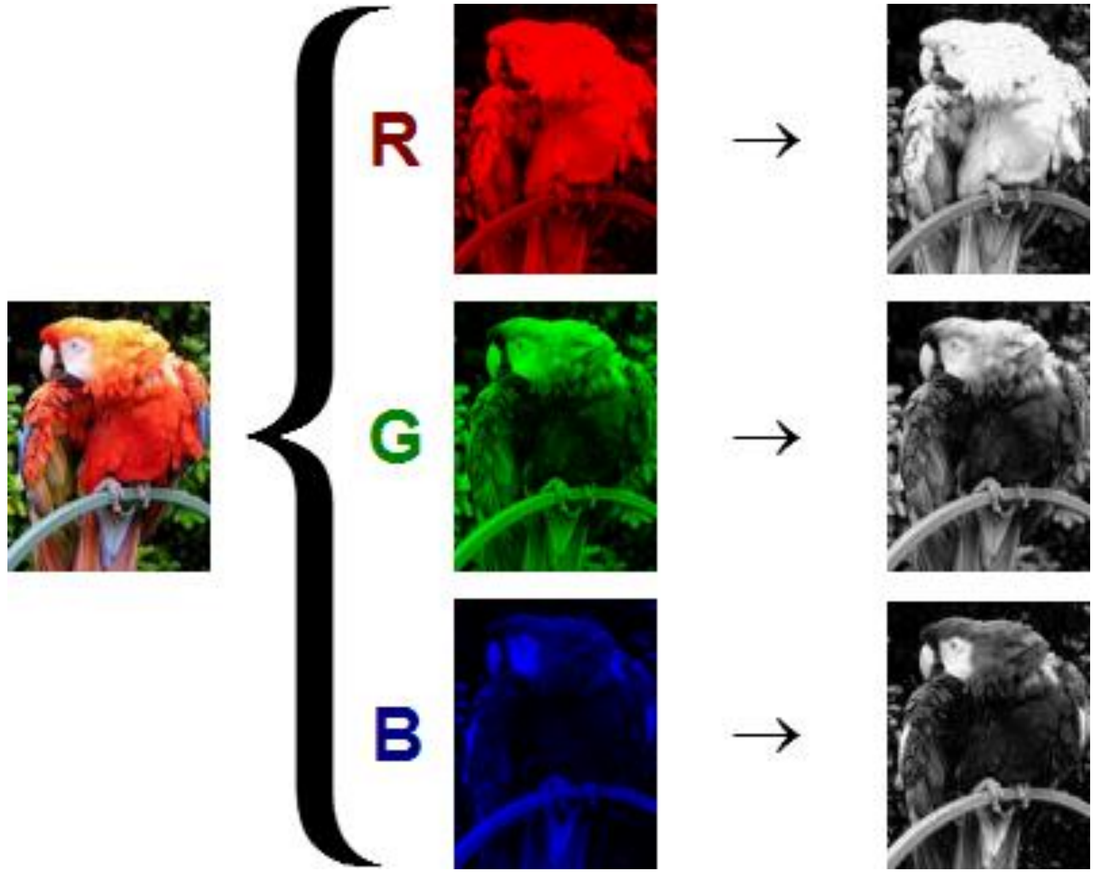
Lidar en point clouds

- Lidar = LIght Detection and Ranging.
- Lidar registreert reflectie
- Elk punt (X, Y, Z) coordinaten.
- Je kan een AI-model trainen om objecten te herkennen



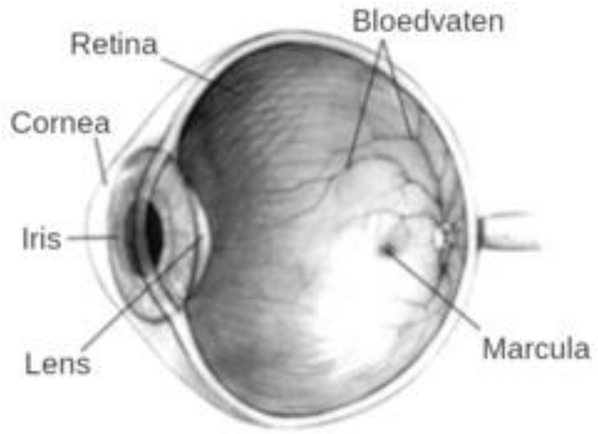
Digitale foto R,G,B

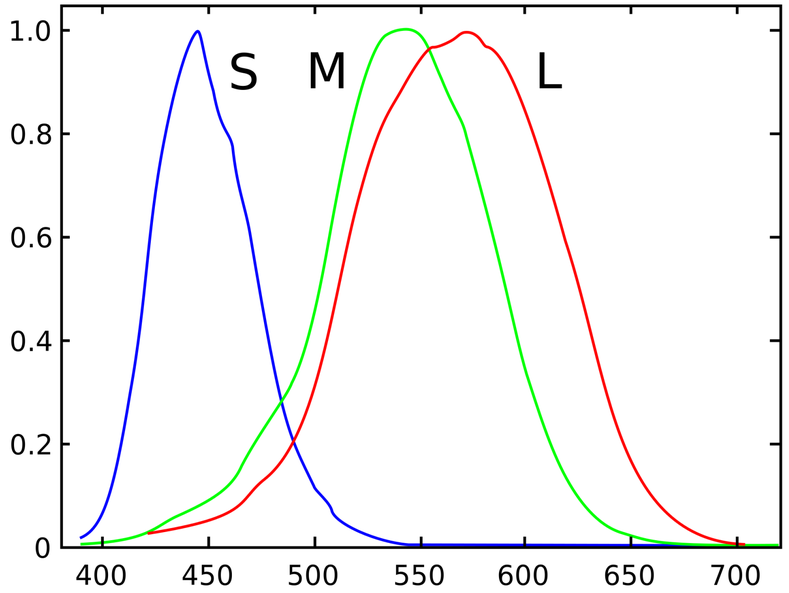




Mensen nemen vooral R,G,B waar. Andere kleuren minder.

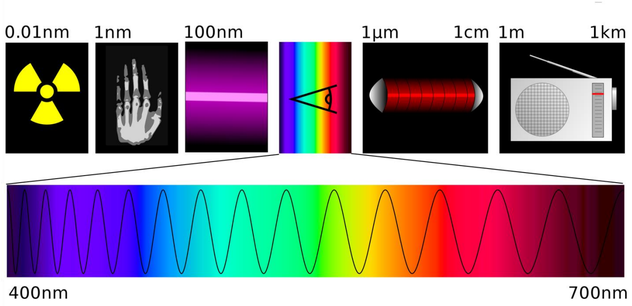
- Mensen zijn ‘tri-chromatisch’. Tri = drie en chroma = kleur
- Netvlies (of retina) vangt vooral drie kleuren op: RGB

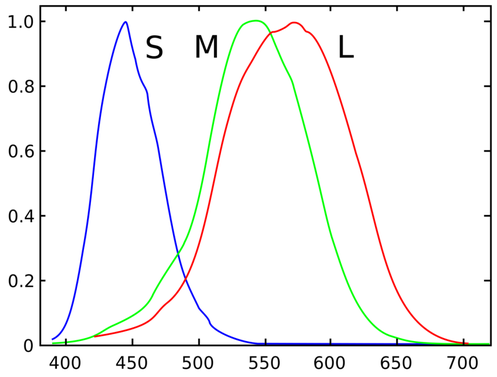


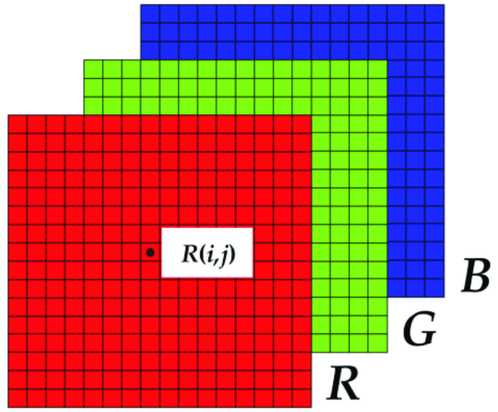


https://en.wikipedia.org/wiki/Trichromacy

Samenvattend: van het zichtbare licht vangen onze ogen vooral R,G,B op.

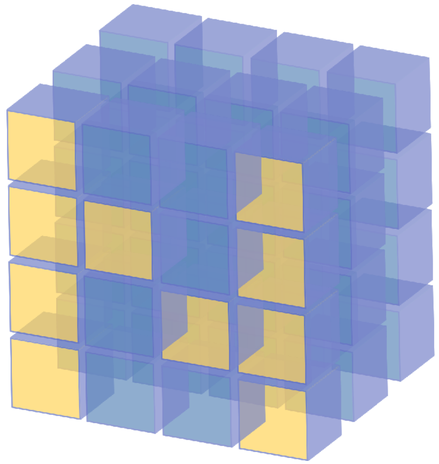






Let op: circa 3% van de mensen is kleurenblind en ziet niet alle kleuren!

NumPy



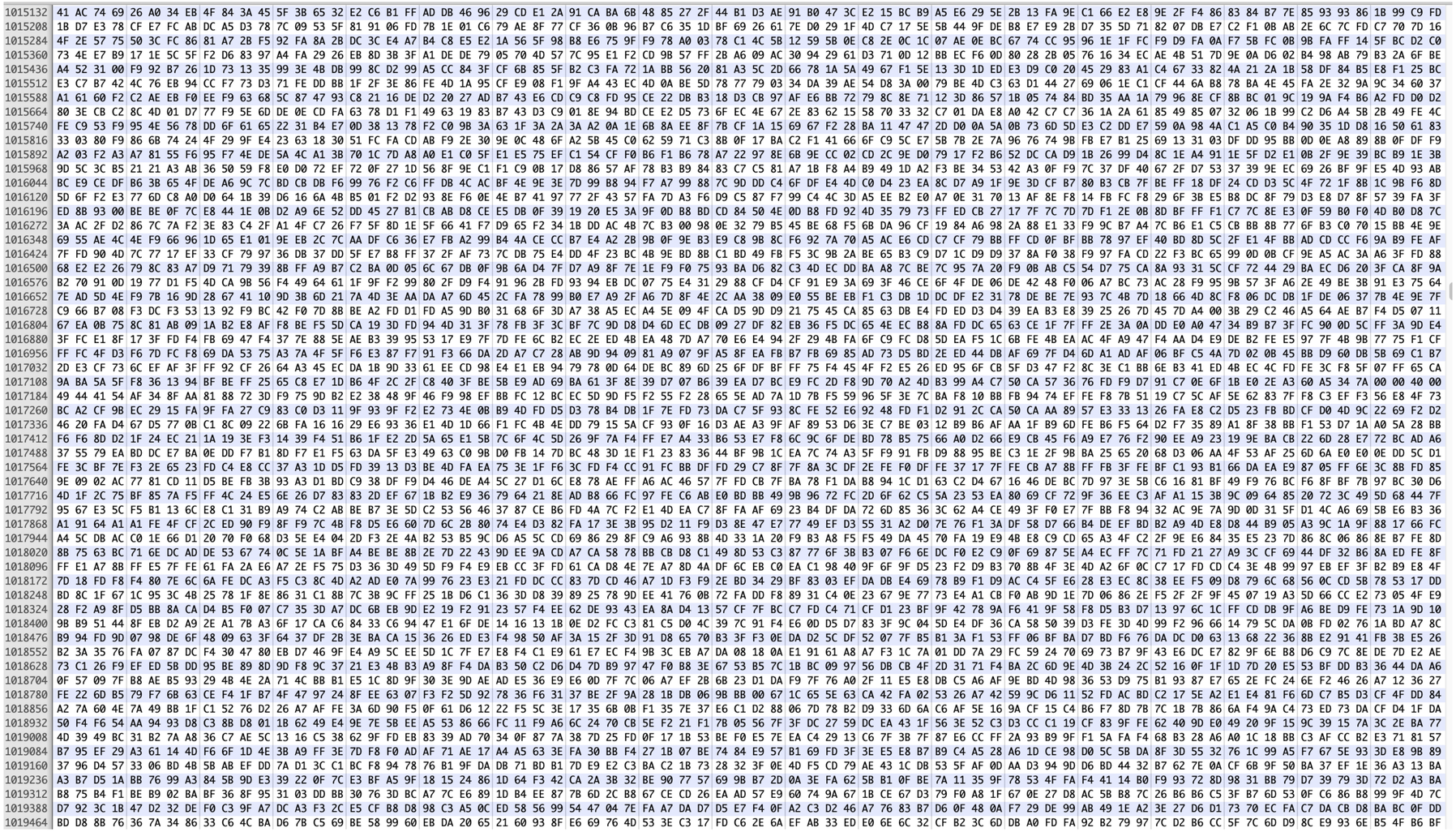
NumPy

NumPy is een Python library

- **Num**erical **Py**thon
- Wiskundige library vooral voor arrays en matrices
- documentatie : https://numpy.org

```
import numpy as np
```

data

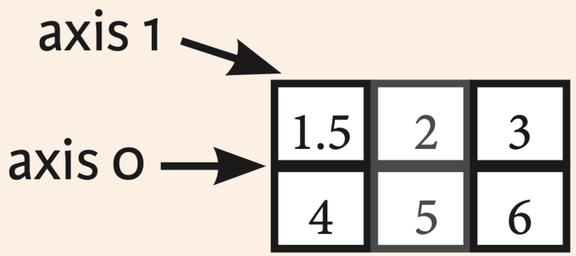

1015132 41 AC 74 69 26 A0 34 EB 4F 84 3A 45 5F 3B 65 32 E2 C6 B1 FF AD DB 46 96 29 CD E1 2A 91 CA BA 6B 48 85 27 2F 44 B1 D3 AE 91 B0 47 3C E2 15 BC B9 A5 E6 29 5E 2B 13 FA 9E C1 66 E2 E8 9E 2F F4 86 83 84 B7 7E 85 93 93 86 1B 99 C9 FD
1015208 1B D7 E3 78 CF E7 FC AB DC A5 D3 78 7C 09 53 5F 81 91 06 FD 7B 1E 01 C6 79 AE 8F 77 CF 36 0B 96 B7 C6 35 1D BF 69 26 61 7E D0 29 1F 4D C7 17 5E 5B 44 9F DE B8 E7 E9 2B D7 35 5D 71 82 07 DB E7 C2 F1 0B AB 2E 6C 7C FD C7 70 7D 16
1015284 4F 2E 57 75 50 3C FC 86 81 A7 2B F5 92 FA 8A 2B DC 3C E4 A7 B4 C8 E5 E2 1A 56 5F 98 B8 E6 75 9F F9 78 A0 03 78 C1 4C 5B 12 59 5B 0E C8 2E 0C 1C 07 AE 0E BC 67 74 CC 95 96 1E 1F FC F9 D9 FA 0A F7 5B FC 0B 9B FA FF 14 5F BC D2 C0
1015360 73 4E E7 B9 17 1E 5C 5F F2 D6 83 97 A4 FA 29 26 EB 8D 3B 3F A1 DE DE 79 05 70 4D 57 7C 95 E1 F2 CD 9B 57 FF 2B A6 09 AC 30 94 29 61 D3 71 0D 12 BB EC F6 0D 80 28 2B 05 76 16 34 EC AE 4B 51 7D 9E 0A D6 02 B4 98 AB 79 B3 2A 6F BE
1015436 A4 52 31 00 F9 92 B7 26 1D 73 13 35 99 3E 4B DB 99 8C D2 99 A5 CC 84 3F CF 6B 85 5F B2 C3 FA 72 1A BB 56 20 81 A3 5C 2D 66 78 1A 5A 49 67 F1 5E 13 3D 1D ED E3 D9 C0 20 45 29 83 A1 C4 67 33 82 4A 21 2A 1B 58 DF 84 B5 E8 F1 25 BC
1015512 E3 C7 B7 42 4C 76 EB 94 CC F7 73 D3 71 FE DD BB 1F 2F 3E 86 FE 4D 1A 95 CF E9 08 F1 9F A4 43 EC 4D 0A BE 5D 78 77 79 03 34 DA 39 AE 54 D8 3A 00 79 BE 4D C3 63 D1 44 27 69 06 1E C1 CF 44 6A B8 78 BA 4E 45 FA 2E 32 9A 9C 34 60 37
1015588 A1 61 60 F2 C2 AE EB F0 EE F9 63 68 5C 87 47 93 C8 21 16 DE D2 20 27 AD B7 43 E6 CD C9 C8 FD 95 CE 22 DB B3 18 D3 CB 97 AF E6 BB 72 79 8C 8E 71 12 3D 86 57 1B 05 74 84 BD 35 AA 1A 79 96 8E CF 8B BC 01 9C 19 9A F4 B6 A2 FD D0 D2
1015664 80 3E CB C2 8C 4D 01 D7 77 F9 5E 6D DE 0E CD FA 63 78 D1 F1 49 63 19 83 B7 43 D3 C9 01 8E 94 BD CE E2 D5 73 6F EC 4E 67 2E 83 62 15 58 70 33 32 C7 01 DA E8 A0 42 C7 C7 36 1A 2A 61 85 49 85 07 32 06 1B 99 C2 D6 A4 5B 2B 49 FE 4C
1015740 FE C9 53 F9 95 4E 56 78 DD 6F 61 65 22 31 B4 E7 0D 38 13 78 F2 C0 9B 3A 63 1F 3A 2A 3A A2 0A 1E 6B 8A EE 8F 7B CF 1A 15 69 67 F2 28 BA 11 47 47 2D D0 0A 5A 0B 73 6D 5D E3 C2 DD E7 59 0A 98 4A C1 A5 C0 B4 90 35 1D D8 16 50 61 83
1015816 33 03 80 F9 86 6B 74 24 4F 29 9F E4 23 63 18 30 51 FC FA CD AB F9 2E 30 9E 0C 48 6F A2 5B 45 C0 62 59 71 C3 8B 0F 17 BA C2 F1 41 66 6F C9 5C E7 5B 7B 2E 7A 96 76 74 9B FB E7 B1 25 69 13 31 03 DF DD 95 BB 0D 0E A8 89 8B 0F DF F9
1015892 A2 03 F2 A3 A7 81 55 F6 95 F7 4E DE 5A 4C A1 3B 70 1C 7D A8 A0 E1 C0 5F E1 E5 75 EF C1 54 CF F0 B6 F1 B6 78 A7 22 97 8E 6B 9E CC 02 CD 2C 9E D0 79 17 F2 B6 52 DC CA D9 1B 26 99 D4 8C 1E A4 91 1E 5F D2 E1 0B 2F 9E 39 BC B9 1E 3B
1015968 9D 5C 3C B5 21 21 A3 AB 36 50 59 F8 E0 D0 72 EF 72 0F 27 1D 56 8F 9E C1 F1 C9 0B 17 D8 86 57 AF 78 B3 B9 84 83 C7 C5 81 A7 1B F8 A4 B9 49 1D A2 F3 BE 34 53 42 A3 0F F9 7C 37 DF 40 67 2F D7 53 37 39 9E EC 69 26 BF 9F E5 4D 93 AB
1016044 BC E9 CE DF B6 3B 65 4F DE A6 9C 7C BD CB DB F6 99 76 F2 C6 FF DB 4C AC BF 4E 9E 3E 7D 99 B8 94 F7 A7 99 88 7C 9D DD C4 6F DF E4 4D C0 D4 23 EA 8C D7 A9 1F 9E 3D CF B7 80 B3 CB 7F BE FF 18 DF 24 CD D3 5C 4F 72 1F 8B 1C 9B F6 8D
1016120 5D 6F F2 E3 77 6D C8 A0 D0 64 1B 39 D6 16 6A 4B B5 01 F2 D2 93 8E F6 0E 4E B7 41 97 77 2F 43 57 FA 7D A3 F6 D9 C5 87 F7 99 C4 4C 3D A5 EE B2 E0 A7 0E 31 70 13 AF 8E F8 14 FB FC F8 29 6F 3B E5 B8 DC 8F 79 D3 E8 D7 8F 57 39 FA 3F
1016196 ED 8B 93 00 BE BE 0F 7C E8 44 1E 0B D2 A9 6E 52 DD 45 27 B1 CB AB D8 CE E5 DB 0F 39 19 20 E5 3A 9F 0D B8 BD CD 84 50 4E 0D B8 FD 92 4D 35 79 73 FF ED CB 27 17 7F 7C 7D 7D F1 2E 0B 8D BF FF F1 C7 7C 8E E3 0F 59 B0 F0 4D B0 D8 7C
1016272 3A AC 2F D2 86 7C 7A F2 3E 83 C4 2F A1 4F C7 26 F7 5F 8D 1E 5F 66 41 F7 D9 65 F2 34 1B DD AC 4B 7C B3 00 98 0E 32 79 B5 45 BE 68 F5 6B DA 96 CF 19 84 A6 98 2A 88 E1 33 F9 9C B7 A4 7C B6 E1 C5 CB BB 8B 77 6F B3 C0 70 15 BB 4E 9E
1016348 69 55 AE 4C 4E F9 66 96 1D 65 E1 01 9E EB 2C 7C AA DF C6 36 E7 FB A2 99 B4 4A CE CC B7 E4 A2 2B 9B 0F 9E B3 E9 C8 9B 8C F6 92 7A 70 A5 AC E6 CD C7 CF 79 BB FF CD 0F BF BB 78 97 EF 40 BD 8D 5C 2F E1 4F BB AD CD CC F6 9A B9 FE AF
1016424 7F FD 90 4D 7C 77 17 EF 33 CF 79 97 36 DB 37 DD 5F E7 B8 FF 37 2F AF 73 7C DB 75 E4 DD 4F 23 BC 4B 9E BD 8B C1 BD 49 FB F5 3C 9B 2A BE 65 B3 C9 D7 1C D9 D9 37 8A F0 38 F9 97 FA CD 22 F3 BC 65 99 0D 0B CF 9E A5 AC 3A A6 3F FD 88
1016500 68 E2 E2 26 79 8C 83 A7 D9 71 79 39 8B FF A9 B7 C2 BA 0D 05 6C 67 DB 0F 9B 6A D4 7F D7 A9 8F 7E 1E F9 F0 75 93 BA D6 82 C3 4D EC DD BA A8 7C BE 7C 95 7A 20 F9 0B AB C5 54 D7 75 CA 8A 93 31 5C CF 72 44 29 BA EC D6 20 3F CA 8F 9A
1016576 B2 70 91 0D 19 77 D1 F5 4D CA 9B 56 F4 49 64 61 1F 9F F2 99 80 2F D9 F4 91 96 2B FD 93 94 EB DC 07 75 E4 31 29 88 CF D4 CF 91 E9 3A 69 3F 46 CE 6F 4F DE 06 DE 42 48 F0 06 A7 BC 73 AC 28 F9 95 9B 57 3F A6 2E 49 BE 3B 91 E3 75 64
1016652 7E AD 5D 4E F9 7B 16 9D 28 67 41 10 9D 3B 6D 21 7A 4D 3E AA DA A7 6D 45 2C FA 78 99 B0 E7 A9 2F A6 7D 8F 4E 2C AA 38 09 E0 55 BE EB F1 C3 DB 1D DC DF E2 31 78 DE BE 7E 93 7C 4B 7D 18 66 4D 8C F8 06 DC DB 1F DE 06 37 7B 4E 9E 7F
1016728 C9 66 B7 08 F3 DC F3 53 13 92 F9 BC 42 F0 7D 8B BE A2 FD D1 FD A5 9D B0 31 68 6F 3D A7 38 A5 EC A4 5E 09 4F CA D5 9D D9 21 75 45 CA 85 63 DB E4 FD ED D3 D4 39 EA B3 E8 39 25 26 7D 45 7D A4 00 3B 29 C2 46 A5 64 AE B7 F4 D5 07 11
1016804 67 EA 0B 75 8C 81 AB 09 1A B2 E8 AF F8 BE F5 5D CA 19 3D FD 94 4D 31 3F 78 FB 3F 3C BF 7C 9D D8 D4 6D EC DB 09 27 DF 82 EB 36 F5 DC 65 4E EC B8 8A FD DC 65 63 CE 1F 7F FF 2E 3A 0A DD E0 A0 47 34 B9 B7 3F FC 90 0D 5C FF 3A 9D E4
1016880 3F FC E1 8F 17 3F FD F4 FB 69 47 F4 37 7E 88 5E AE B3 39 95 53 17 E9 7F 7D FE 6C B2 EC 2E ED 4B EA 48 7D A7 70 E6 E4 94 2F 29 4B FA 6F C9 FC D8 5D EA F5 1C 6B FE 4B EA AC 4F A9 47 F4 AA D4 E9 DE B2 FE E5 97 7F 4B 9B 77 75 F1 CF
1016956 FF FC 4F D3 F6 7D FC F8 69 DA 53 75 A3 7A 4F 5F F6 E3 87 F7 91 F3 66 DA 2D A7 C7 28 AB 9D 94 09 81 A9 07 9F A5 8F EA FB B7 FB 69 85 AD 73 D5 BD 2E ED 44 DB AF 69 7F D4 6D A1 AD AF 06 BF C5 4A 7D 02 0B 45 BB D9 60 DB 5B 69 C1 B7
1017032 2D E3 CF 73 6C EF AF 3F FF 92 CF 26 64 A3 45 EC DA 1B 9D 33 61 EE CD 98 E4 E1 EB 94 79 78 0D 64 DE BC 89 6D 25 6F DF BF FF 75 F4 45 4F F2 E5 26 ED 95 6F CB 5F D3 47 F2 8C 3E C1 BB 6E B3 41 ED 4B EC 4C FD FE 3C F8 5F 07 FF 65 CA
1017108 9A BA 5A 5F F8 36 13 94 BF BE FF 25 65 C8 E7 1D B6 4F 2C 2F C8 40 3F BE 5B E9 AD 69 BA 61 3F 8E 39 D7 07 B6 39 EA D7 BC E9 FC 2D F8 9D 70 A2 4D B3 99 A4 C7 50 CA 57 36 76 FD F9 D7 91 C7 0E 6F 1B E0 2E A3 60 A5 34 7A 00 00 40 00
1017184 49 44 41 54 AF 34 8F AA 81 88 72 3D F9 75 9D B2 E2 38 48 9F 46 F9 98 EF BB FC 12 BC EC 5D 9D F5 F2 55 F2 28 65 5E AD 7A 1D 7B F5 59 96 5F 3E 7C BA F8 10 BB FB 94 74 EF FE F8 7B 51 19 C7 5C AF 5E 62 83 7F F8 C3 EF F3 56 E8 4F 73
1017260 BC A2 CF 9B EC 29 15 FA 9F FA 27 C9 83 C0 D3 11 9F 93 9F F2 E2 73 4E 0B B9 4D FD D5 D3 78 B4 DB 1F 7E FD 73 DA C7 5F 93 8C FE 52 E6 92 48 FD F1 D2 91 2C CA 50 CA AA 89 57 E3 33 13 26 FA E8 C2 D5 23 FB BD CF D0 4D 9C 22 69 F2 D2
1017336 46 20 FA D4 67 D5 77 0B C1 8C 09 22 6B FA 16 16 29 E6 93 36 E1 4D 1D 66 F1 FC 4B 4E DD 79 15 5A CF 93 0F 16 D3 AE A3 9F AF 89 53 D6 3E C7 BE 03 12 B9 B6 AF AA 1F B9 6D FE B6 F5 64 D2 F7 35 89 A1 8F 38 BB F1 53 D1 1A A0 5A 28 8B
1017412 F6 F6 8D D2 1F 24 EC 21 1A 19 3E F3 14 39 F4 51 B6 1F E2 2D 5A 65 E1 5B 7C 6F 4C 5D 26 9F 7A F4 FF E7 A4 33 B6 53 E7 F8 6C 9C 6F DE BD 78 B5 75 66 A0 D2 66 E9 CB 45 F6 A9 E7 76 F2 90 EE A9 23 19 9E BA CB 22 6D 28 E7 72 BC AD A6
1017488 37 55 79 EA BD DC E7 BA 0E DD F7 B1 8D F7 E1 F5 63 DA 5F E3 49 63 C0 9B D0 FB 14 7D BC 48 3D 1E F1 23 83 36 44 BF 9B 1C EA 7C 74 A3 5F F9 91 FB D9 88 95 BE C3 1E 2F 9B BA 25 65 20 68 D3 06 AA 4F 53 AF 25 6D 6A E0 E0 0E DD 5C D1
1017564 FE 3C BF 7E F3 2E 65 23 FD C4 E8 CC 37 A3 1D D5 FD 39 13 D3 BE 4D FA EA 75 3E 1F F6 3C FD F4 CC 91 FC BB DF FD 29 C7 8F 7F 8A 3C DF 2E FE F0 DF FE 37 17 7F FE CB A7 8B FF FB 3F FE BF C1 93 B1 66 DA EA E9 87 05 FF 6E 3C 8B FD 85
1017640 9E 09 02 AC 77 81 CD 11 D5 BE FB 3B 93 A3 D1 BD C9 38 DF F9 D4 46 DE A4 5C 27 D1 6C E8 78 AE FF A6 AC 46 57 7F FD CB 7F BA 78 F1 DA B8 94 1C D1 63 C2 D4 67 16 46 DE BC 7D 97 3E 5B C6 16 81 BF 49 F9 76 BC F6 8F BF 7B 97 BC 30 D6
1017716 4D 1F 2C 75 B7 85 7A F5 FF 4C 24 E5 6E 26 D7 83 83 2D EF 67 1B B2 E9 36 79 64 21 8E AD B8 66 FC 97 FE C6 AB E0 BD BB 49 9B 96 72 FC 2D 6F 62 C5 5A 23 53 EA 80 69 CF 72 9F 36 EE C3 AF A1 15 3B 9C 09 64 85 20 72 3C 49 5D 68 44 72
1017792 95 67 E3 5C F5 B1 13 6C E8 C1 31 B9 A9 74 C2 AB BE B7 3E 5D C2 53 56 46 37 87 CE B6 FD 4A 7C F2 E1 4D EA C7 8F FA AF 69 23 B4 DF DA 72 6D 85 36 3C 62 A4 CE 49 3F F0 E7 7F BB F8 94 32 AC 9E 7A 9D 0D 31 5F D1 4C A6 69 5B E6 B3 36
1017868 A1 91 64 A1 A1 FE 4F CF 2C ED 90 F9 8F F9 7C 4B F8 D5 E6 60 7D 6C 2B 80 74 E4 D3 82 FA 17 3E 3B 95 D2 11 F9 D3 8E 47 E7 77 49 EF D3 55 31 A2 D0 7E 76 F1 3A DF 58 D7 66 B4 DE EF BD B2 A9 4D E8 D8 44 B9 05 A3 9C 1A 9F 88 17 66 FC
1017944 A4 5C DB AC C0 1E 66 D1 20 70 F0 68 D3 5E E4 04 2D F3 2E 4A B2 53 B5 9C D6 A5 5C CD 69 86 29 8F C9 A6 93 8B 4D 33 1A 20 F9 B3 A8 F5 F5 49 DA 45 70 FA 19 E9 4B E8 C9 CD 65 A3 4F C2 2F 9E E6 84 35 E5 23 7D 86 8C 06 86 8E B7 FE 8D
1018020 8B 75 63 BC 71 6E DC AD DE 53 67 74 0C 5E 1A BF A4 BE BE 8B 2E 7D 22 43 9D EE 9A CD A7 CA 58 78 BB CB D8 C1 49 8D 53 C3 87 77 6F 3B B3 07 F6 6E DC F0 E2 C9 0F 69 87 5E A4 EC FF 7C 71 FD 21 27 A9 3C CF 69 44 DF 32 B6 8A ED FE 8F
1018096 FF E1 A7 8B FF E5 7F FE 61 FA 2A E6 A7 2E F5 75 D3 36 3D 49 5D F9 F4 E9 EB CC 3F FD 61 CA D8 4E 7E A7 8D 4A DF 6C EB C0 EA C1 98 40 9F 6F 9F D5 23 F2 D9 B3 70 8B 4F 3E 4D A2 6F 0C C7 17 FD CD C4 3E 4B 99 97 EB EF 3F B2 B9 E8 4F
1018172 7D 18 FD F8 F4 80 7E 6C 6A FE DC A3 F5 C3 8C 4D A2 AD E0 7A 99 76 23 E3 21 FD DC CC 83 7D CD 46 A7 1D F3 F9 2E BD 34 29 BF 83 03 EF DA DB E4 69 78 B9 F1 D9 AC C4 5F E6 28 E3 EC 8C 38 EE F5 09 D8 79 6C 68 56 0C CD 5B 78 53 17 DD
1018248 BD 8C 1F 67 1C 95 3C 4B 25 78 1F 8E 86 31 C1 8B 7C 3B 9C FF 25 1B D6 C1 36 3D D8 39 89 25 78 9D EE 41 76 0B 72 FA DD F8 89 31 C4 0E 23 67 9E 77 73 E4 A1 CB F0 AB 9D 1E 7D 06 86 2E F5 2F 2F 9F 45 07 19 A3 5D 66 CC E2 73 05 4F E9
1018324 28 F2 A9 8F D5 BB 8A CB D4 B5 F0 07 C7 35 3D A7 DC 6B EB 9D E2 19 F2 91 23 57 F4 EE 62 DE 93 43 EA 8A D4 13 57 CF 7F BC C7 FD CA 71 CF D1 23 BF 9F 42 78 9A F6 41 9F 58 F8 D5 B3 C7 13 97 6C 1C FF CD DB 9F A6 BE D0 FE 73 1A 9D 10
1018400 9B B9 51 44 8F EB D2 A9 2E A1 7B A3 6F 17 CA C6 84 33 C6 94 47 E1 6F DE 14 16 13 1B 0E D2 FC C3 81 C5 D0 4C 39 7C 91 F4 E6 0D D5 D7 83 3F 9C 04 5D E4 DF 36 CA 58 50 39 D3 FE 3D 4D 99 F2 96 66 14 79 5C DA 0B FD 02 76 1A BD A7 8C
1018476 B9 94 FD 9D 07 98 DE 6F 48 09 63 3F 64 37 DF 2B 3E BA CA 15 36 26 ED E3 F4 98 50 AF 3A 15 2F 3D 91 D8 65 70 B3 3F F3 0E DA D2 5C DF 52 07 7F B5 B1 3A F1 53 FF 06 BF BA D7 BD F6 76 DA DC D0 63 13 68 22 36 8B E2 91 41 FB 3B E5 26
1018552 B2 3A 35 76 FA 07 87 DC F4 30 47 80 EB D7 46 9F E4 A9 5C EE 5D 1C 7F E7 E8 F4 C1 E9 61 E7 EC F4 9B 3C EB A7 DA 08 18 0A E1 91 61 A8 A7 F3 1C 7A 01 DD 7A 29 FC 59 24 70 69 73 B7 9F 43 E6 DC E7 82 9F 6E B8 D6 C9 7C 8E DE 7D E2 AE
1018628 73 C1 26 F9 EF ED 5B DD 95 BE 89 8D 9D F8 9C 37 21 E3 4B B3 A9 8F F4 DA B3 50 C2 D6 D4 7D B9 97 47 F0 B8 3E 67 53 B5 7C 1B BC 09 97 56 DB CB 4F 2D 31 71 F4 BA 2C 6D 9E 4D 3B 24 2C 52 16 0F 1F 1D 7D 20 E5 53 BF DD B3 36 44 DA A6
1018704 0F 57 09 7F B8 AE B5 93 29 4B 4E 2A 71 4C BB B1 E5 1C 8D 9F 30 3E 9D AE AD E5 36 E9 E6 0D 7F 7C 06 A7 EF 2B 6B 23 D1 DA F9 7F 76 A0 2F 11 E5 E8 DB C5 A6 AF 9E BD 4D 98 36 53 D9 75 B1 93 87 E7 65 2E FC 24 6E F2 46 26 A7 12 36 27
1018780 FE 22 6D B5 79 F7 6B 63 CE F4 1F B7 4F 47 97 24 8F EE 63 07 F3 F2 5D 92 78 36 F6 31 37 BE 2F 9A 28 1B DB 06 9B BB 00 67 1C 65 5E 63 CA 42 FA 02 53 26 A7 42 59 9C D6 11 52 FD AC BD C2 17 5E A2 E1 E4 81 F6 6D C7 B5 D3 CF 4F DD 84
1018856 A2 7A 60 4E 7A 49 BB 1F C1 52 76 D2 26 A7 AF FE 3A 6D 90 F5 0F 61 D6 12 22 F5 5C 3E 17 35 6B 0B F1 35 7E 37 E6 C1 D2 88 06 7D 78 B2 D9 33 6D 6A C6 AF 5E 16 9A CF 15 C4 B6 F7 8D 7B 7C 1B 7B 86 6A F4 9A C4 73 ED 73 DA CF D4 1F DA
1018932 50 F4 F6 54 AA 94 93 D8 C3 8B D8 01 1B 62 49 E4 9E 7E 5B EE A5 53 86 66 FC 11 F9 A6 6C 24 70 CB 5E F2 21 F1 7B 05 56 7F 3F DC 27 59 DC 64 A3 1F 56 3E 52 C3 D3 C1 19 CF 83 9F FE 62 40 9D E0 49 20 9F 15 9C 39 15 7A 3C 2E BA 77
1019008 4D 39 49 BC 31 B2 7A A8 36 C7 AE 5C 13 16 C5 38 62 9F FD EB 83 39 AD 70 34 0F 87 7A 38 7D 25 FD 0F 17 1B 53 BE F0 E5 7E EA C4 29 13 C6 7F 3B 7F 87 E6 CC FF 2A 93 B9 9F F1 5A FA F4 68 B3 28 A6 A0 1C 18 BB C3 AF CC B2 E3 71 81 57
1019084 B7 95 EF 29 A3 61 14 4D F6 6F 1D 4E 3B A9 FF 3E 7D F8 F0 AD AF 71 AE 17 A4 A5 63 3E FA 30 BB F4 27 1B 07 BE 74 84 E9 57 B1 69 FD 3F 3E E5 E8 B7 B9 C4 A5 28 A6 1D CE 98 D0 5C 5B DA 8F 3D 55 32 76 1C 99 A5 F7 67 5E 93 3D E8 9B 89
1019160 37 96 D4 57 33 06 BD 4B 5B AD EF DD 7A D1 3C C1 BC F8 94 78 76 B1 9F DA DB 71 BD B1 7D E9 E2 C3 BA C2 1B 73 28 32 3F 0E 4D F5 CD 79 AE 43 1C DB 53 5F AF 0D 94 9D D6 BD 44 32 B7 62 7E 0A CF 6B 9F 50 BA 37 EF 1E 36 A3 13 BA
1019236 A3 B7 D5 1A BB 76 99 A3 84 5B 9D E3 39 22 0F 7C E3 BF A5 9F 18 15 24 86 1D 64 F3 42 CA 2A 3B 32 BE 90 77 57 69 9B B7 2D 0A 3E FA 62 5B B1 0F BE 7A 11 35 9F 78 53 4F FA F4 41 14 B0 F9 93 72 8D 98 31 BB 79 D7 39 79 3D 72 D2 A3 BA
1019312 B8 75 B4 F1 BE B9 02 BA BF 36 8F 95 31 03 DD BB 30 76 3D BC A7 7C E6 89 1D B4 EE 87 7B 6D 2C B8 67 CE CD 26 EA AD 57 E9 60 74 9A 67 1B CE 67 D3 79 F0 A8 1F 67 0E 27 D8 AC 5B B8 7C 26 B6 B6 C5 3F B7 6D 53 0F C6 86 B8 99 9F 4D 7C
1019388 D7 92 3C 1B 47 D2 32 DE F0 C3 9F A7 DC A3 F3 2C E5 CF B8 D8 98 C3 A5 0C ED 58 56 99 54 47 04 7E FA A7 DA D7 D5 E7 F4 0F A2 C3 D2 46 A7 76 83 B7 D6 0F 48 0A F7 29 EC E4 79 DE 99 AB 49 1E A2 3E 27 D6 D1 73 F7 20 CA CB D8 BA EC D6 20
1019464 BD D8 8B 76 36 7A 34 86 33 C6 4C BA D6 7B C5 69 BE 58 99 60 EB DA 20 65 21 60 93 8F E6 69 76 4D 53 3E C3 17 FD C6 2E 6A EF AB 33 ED 0E 6E 6C 32 CF B2 3C 6D DB A0 FD FA 92 B2 79 97 7C D2 B6 CC 5F 7C 6D D9 8C E9 93 6E B5 4F B6 BF

NumPy array

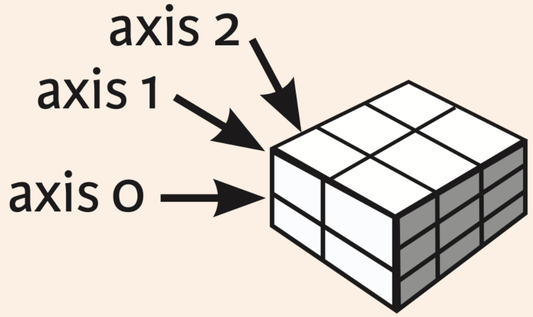
NumPy Arrays

1D array

2D array

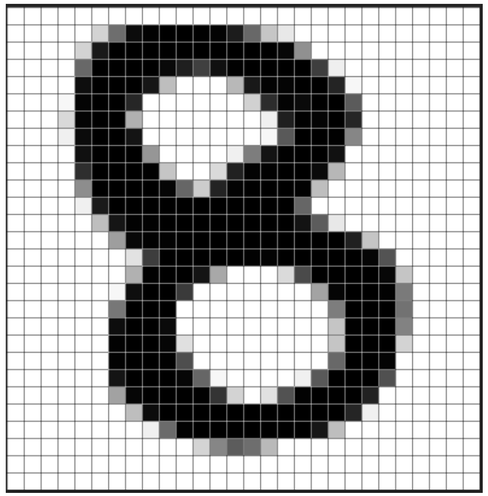


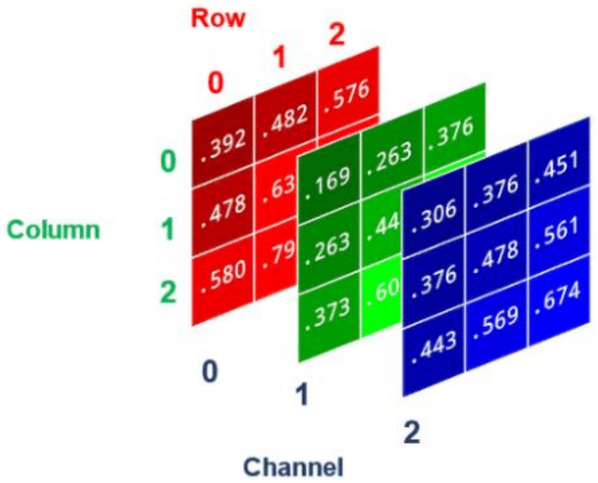
3D array



Numpy: image => array

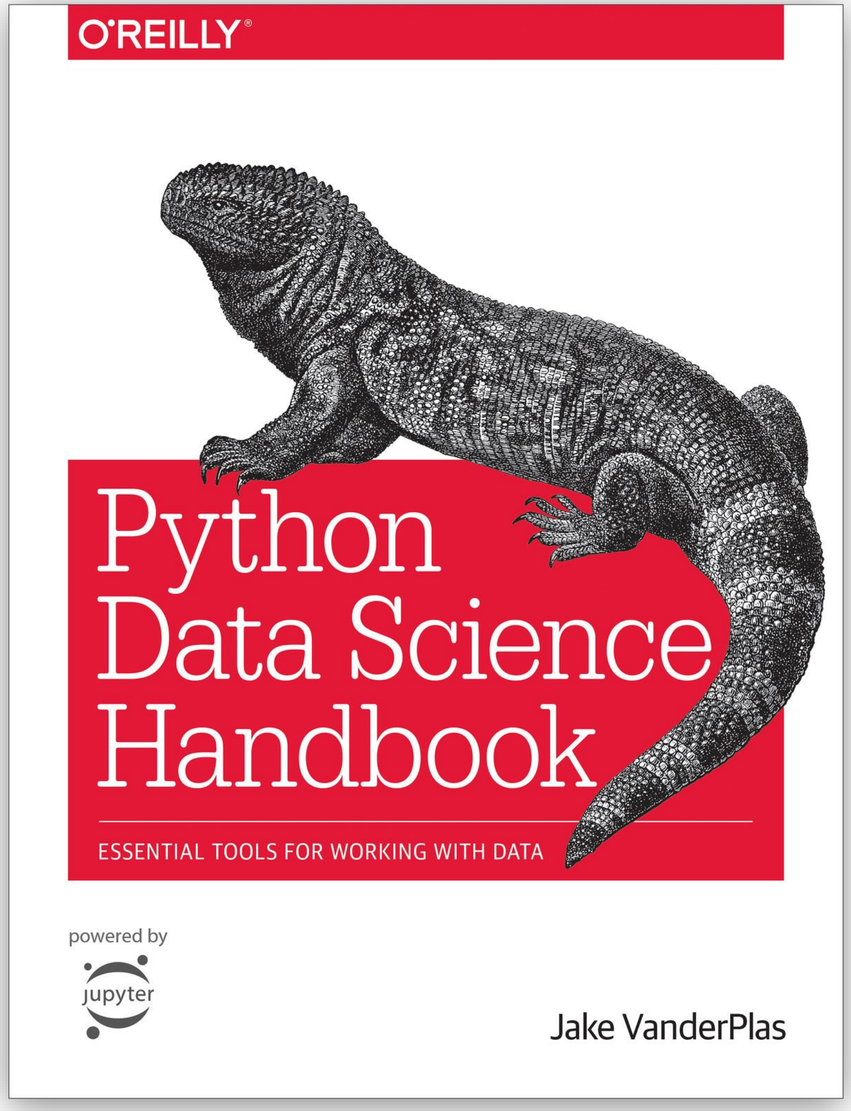
Hoeveel dimensies heeft het linker plaatje?
En het rechter?





Numpy wiskunde

Operator	Equivalent ufunc	Description
`+`	`np.add`	Addition (e.g., `1 + 1 = 2`)
`-`	`np.subtract`	Subtraction (e.g., `3 - 2 = 1`)
`-`	`np.negative`	Unary negation (e.g., `-2`)
`*`	`np.multiply`	Multiplication (e.g., `2 * 3 = 6`)
`/`	`np.divide`	Division (e.g., `3 / 2 = 1.5`)
`//`	`np.floor_divide`	Floor division (e.g., `3 // 2 = 1`)
`**`	`np.power`	Exponentiation (e.g., `2 ** 3 = 8`)
`%`	`np.mod`	Modulus/remainder (e.g., `9 % 4 = 1`)



Handbook
Ch. 2 Intro to NumPy

https://learning.oreilly.com/library/view/python-data-science/9781491912126/

https://lib.hva.nl/discovery/dbfulldisplay?context=L&vid=31UKB_UAM2_INST:HVA&docid=alma9939149327405132

Opdracht

- Notebook op DLO:

- `Numpy for images.ipynb`