

Object detection in automated software testing

Brent Gerets



brighttest
AIM HIGH



Inhoud



- Over Brightest & testing
- Stageopdracht
- Resultaten
- Samenvatting & terugkoppeling
- Bijdrage

Over Brightest &
testing





Over Brightest

Missie

"Aim high by creating innovative services and solutions, with a dedicated team of top-notch consultants who combine creativity and effectivity and enable their clients to realize their quality objectives.

Make high quality software evident and accessible for any organization."

Solutions

- Services
- Advies
- Academy



- Software testing
 - = evalueren en verifiëren dat software werkt zoals het zou moeten
 - Voordelen: voorkomen van fouten, betere performance en security...

- Software testing
 - = evalueren en verifiëren dat software werkt zoals het zou moeten
 - Voordelen: voorkomen van fouten, betere performance en security...
- Test automatisatie
 - = groot deel van testen automatisch uitvoeren d.m.v. code
 - Voordelen: meer frequente tests, minder herhaling voor testers...



Stageopdracht



- Gebruik maken van machine learning modellen & frameworks om patronen te detecteren op websites
- Ontwikkelen van een proof-of-concept om te testen of een bepaald patroon correct wordt weergegeven op een pagina





Probleemstelling

- Probleem: testen van websites door elementen op te vragen via HTML code
 - geen zekerheid dat element correct weergegeven wordt



- Probleem: testen van websites door elementen op te vragen via HTML code
 - geen zekerheid dat element correct weergegeven wordt
- Oplossing: elementen opvragen via screenshot
 - equivalent aan hetgeen de gebruiker ziet



- Probleem: testen van websites door elementen op te vragen via HTML code
 - geen zekerheid dat element correct weergegeven wordt
- Oplossing: elementen opvragen via screenshot
 - equivalent aan hetgeen de gebruiker ziet

Gebruik maken van machine learning?



Hoe kan image processing, en meer specifiek object detection, gebruikt worden in het geautomatiseerd software testing proces?



Resultaten



- 12 object detectie modellen (37 varianten, afbeelding niet volledig)
- Rang op basis van 3 factoren
 - Accuraatheid (op COCO set)
 - Snelheid
 - Beschikbaarheid
- Elk een gewogen score
- Top 3
 - Co-DINO
 - YOLOv9
 - RT-DETR

Model		Accuracy		Speed			Availability	Scoring	
Name	Variant	Score	mAP (%)	Score	FLOPs (B)	FPS	Score	Weighted score	Rank
YOLOv8	n (nano)	34.47	37.30	100.00	8.70		100.00	60.68	25
	s (small)	53.06	44.90	98.88	28.60		100.00	71.50	11
	m (medium)	66.01	50.20	96.06	78.60		100.00	78.43	9
	l (large)	72.62	52.90	91.17	165.20		100.00	80.92	6
	x (extra large)	75.06	53.90	85.95	257.80		100.00	80.82	7
YOLOv9	S (small)	57.70	46.80	98.98	26.70		100.00	74.32	10
	M (medium)	68.95	51.40	96.16	76.80		100.00	80.22	8
	C (compact)	72.86	53.00	94.69	102.80		100.00	82.12	4
	E (extended)	79.22	55.60	89.64	192.50		100.00	84.42	2
RT-DETR	L (large)	72.86	53.00	94.29	110.00		100.00	82.00	5
	X (extra large)	77.26	54.80	87.29	234.00		100.00	82.55	3
Co-DETR	Co-DINO	100.00	64.10	84.76	279.00		0.00	85.43	1
	Co-Deformable-DETR	86.31	58.50	51.99	860.00		0.00	67.38	16
Faster R-CNN	R50-FPN	33.74	37.00	75.28	447.00	10.20	50.00	47.83	30
	R50-FPN	41.56	40.20	13.17		26.32	50.00	33.89	34
	R101-FPN	45.97	42.00	8.85		19.61	50.00	35.23	32
	X101-FPN	48.41	43.00	2.79		10.20	50.00	34.88	33
Weight		0.60		0.30			0.10		



Implementatie modellen

- Ultralytics module
 - YOLOv9 & RT-DETR
 - Training & detectie





- Ultralytics module
 - YOLOv9 & RT-DETR
 - Training & detectie
- MMDetection module
 - Co-DINO
 - Te weinig VRAM → uiteindelijk niet geïmplementeerd







Dataset creator

- Nieuwe dataset of verder gaan
- Kies screenshots
- Maak labels
- Teken bounding boxes

BrightSight

DATASET CREATOR TRAINING TESTING SETTINGS FAQ

✓ Configure Draw bounding boxes Augment data

Upload screenshots and draw bounding boxes around every instance of the elements you wish to detect.

NEXT BACK

SAVE

Screenshots

- test.png
- test2.png
- test3.png

Labels

- x2 Brightest logo
- x4 Performance logo
- x2 Learn more
- x1 Arrows
- x4 Logo

Bounding boxes

- #1 Brightest logo
- #2 Performance logo
- #3 Performance logo
- #4 Learn more
- #5 Learn more

brightest

Solutions Cases About Insights Careers Contact Academy

Software Quality AI (SQAI)

SQAI is a conversational AI software testing toolkit for enterprises to build RAG-enabled testing assistants for their development teams and key users.

Learn more

Coaching

We guide your organization or team in the right direction, bring new insights and quickly add expertise.

Learn more

Performance testing

Performance testing is the practice of determining how applications,

Security Testing



Dataset creator

- Nieuwe dataset of verder gaan
- Kies screenshots
- Maak labels
- Teken bounding boxes
- Data augmentatie
 - Schaal wijzigen
 - Bijsnijden
 - Verplaatsen
 - Noise

BrightSight

DATASET CREATOR

✓ Configure

✓ Draw bounding boxes

Use data augmentation to expand the dataset.

DONE

BACK

Configure

Augmentation progress

Batch size
5

Resize amount
5

Resize range

0.4

0.9

Crop amount
0

Crop range

0.6

0.8

Translate amount
5

Translate padding
100

Noise probability
0.01

Screenshot multiplier

1

0%

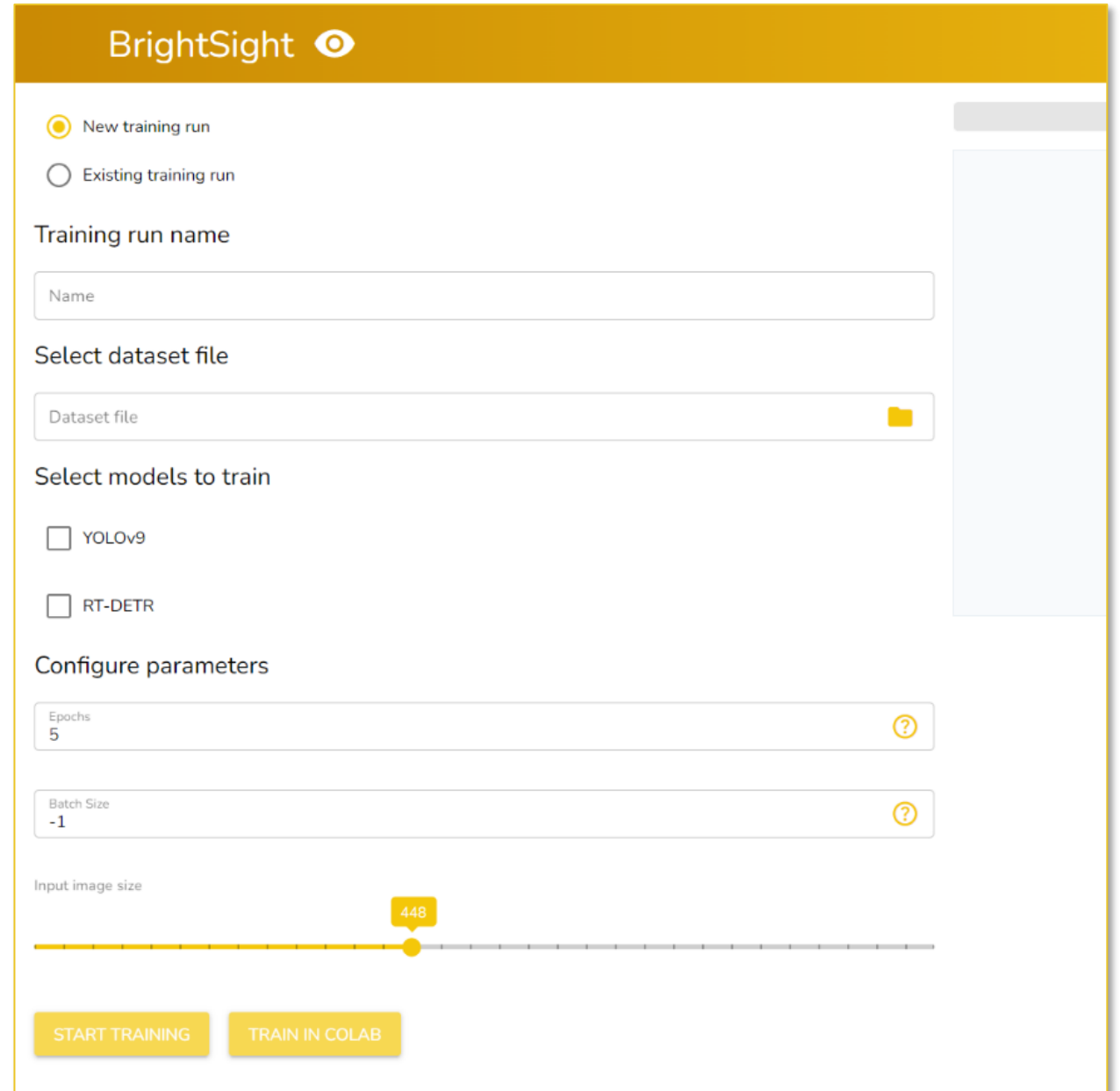
Current process


0%

	screenshots	bounding boxes	total
input	3	16	19
output	108	576	684

START DATA AUGMENTATION


- Nieuwe run of verder gaan
- Selecteer modellen om te trainen
- Selecteer dataset
- Configureer parameters
 - Epochs
 - Batch size
 - Afbeeldingsgrootte
- Weergave training vooruitgang





BrightSight 


☒ New training run
☐ Existing training run

Training run name

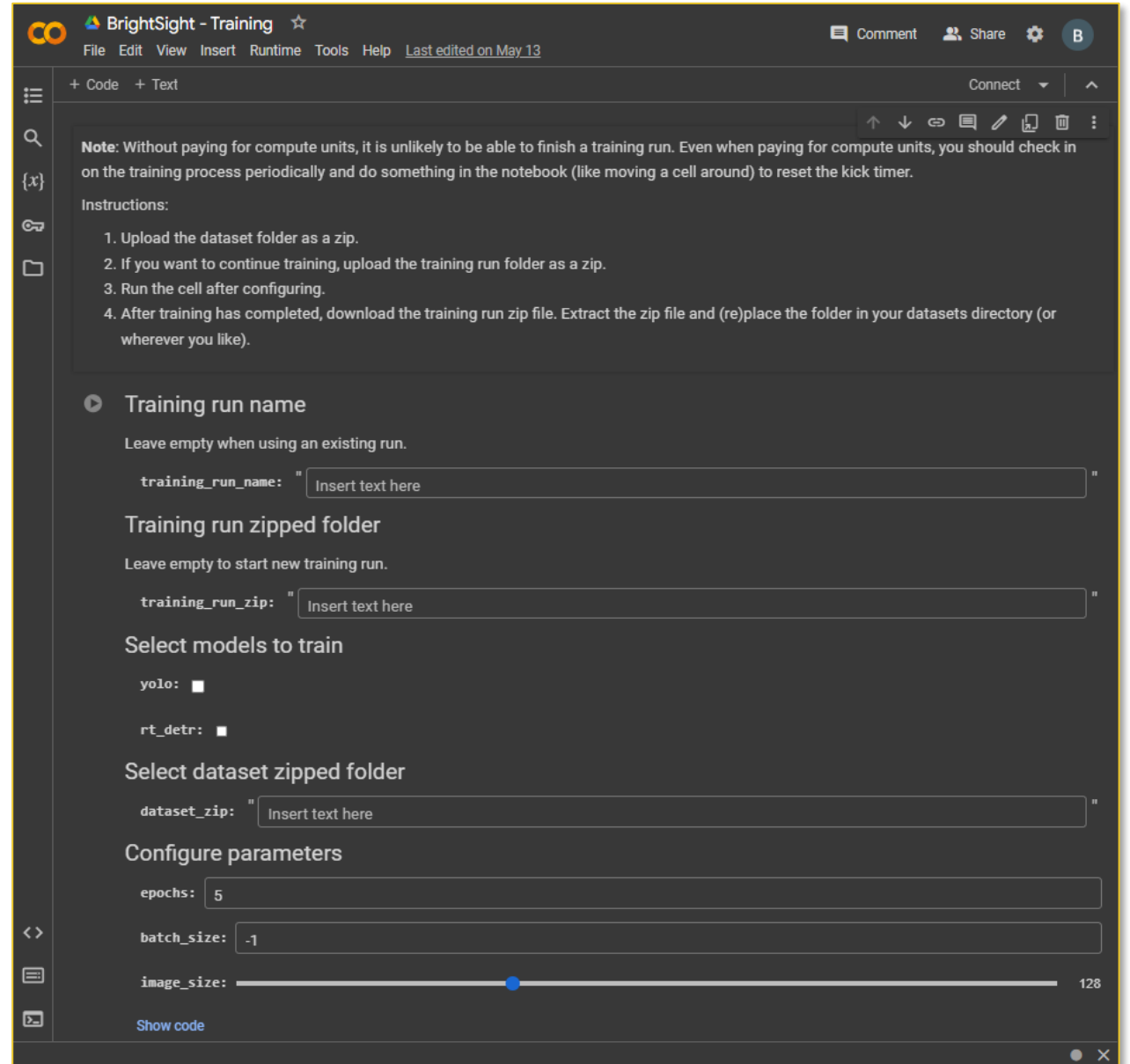
Select dataset file
 

Select models to train
☐ YOLOv9
☐ RT-DETR

Configure parameters
Epochs 
Batch Size 

Input image size


- Nieuwe run of verder gaan
- Selecteer modellen om te trainen
- Selecteer dataset
- Configureer parameters
 - Epochs
 - Batch size
 - Afbeeldingsgrootte
- Weergave training vooruitgang
- Training in Colab optie



The screenshot shows the BrightSight - Training web interface. At the top, there's a header with the logo, title, and navigation links. Below the header, a sidebar on the left contains icons for file management and execution. The main content area is divided into sections: a 'Note' about compute units, 'Instructions' for training, and a 'Training run name' section with input fields for 'training_run_name' and 'training_run_zip'. Below these are sections for 'Select models to train' (with checkboxes for 'yolo' and 'rt_detr'), 'Select dataset zipped folder' (with a 'dataset_zip' input field), and 'Configure parameters' (with input fields for 'epochs' set to 5, 'batch_size' set to -1, and a slider for 'image_size' set to 128). A 'Show code' link is at the bottom.

BrightSight - Training ☆
File Edit View Insert Runtime Tools Help [Last edited on May 13](#)

+ Code + Text

Note: Without paying for compute units, it is unlikely to be able to finish a training run. Even when paying for compute units, you should check in on the training process periodically and do something in the notebook (like moving a cell around) to reset the kick timer.

Instructions:

1. Upload the dataset folder as a zip.
2. If you want to continue training, upload the training run folder as a zip.
3. Run the cell after configuring.
4. After training has completed, download the training run zip file. Extract the zip file and (re)place the folder in your datasets directory (or wherever you like).

Training run name
Leave empty when using an existing run.

training_run_name: "Insert text here"

Training run zipped folder
Leave empty to start new training run.

training_run_zip: "Insert text here"

Select models to train

yolo: ☐

rt_detr: ☐

Select dataset zipped folder

dataset_zip: "Insert text here"

Configure parameters

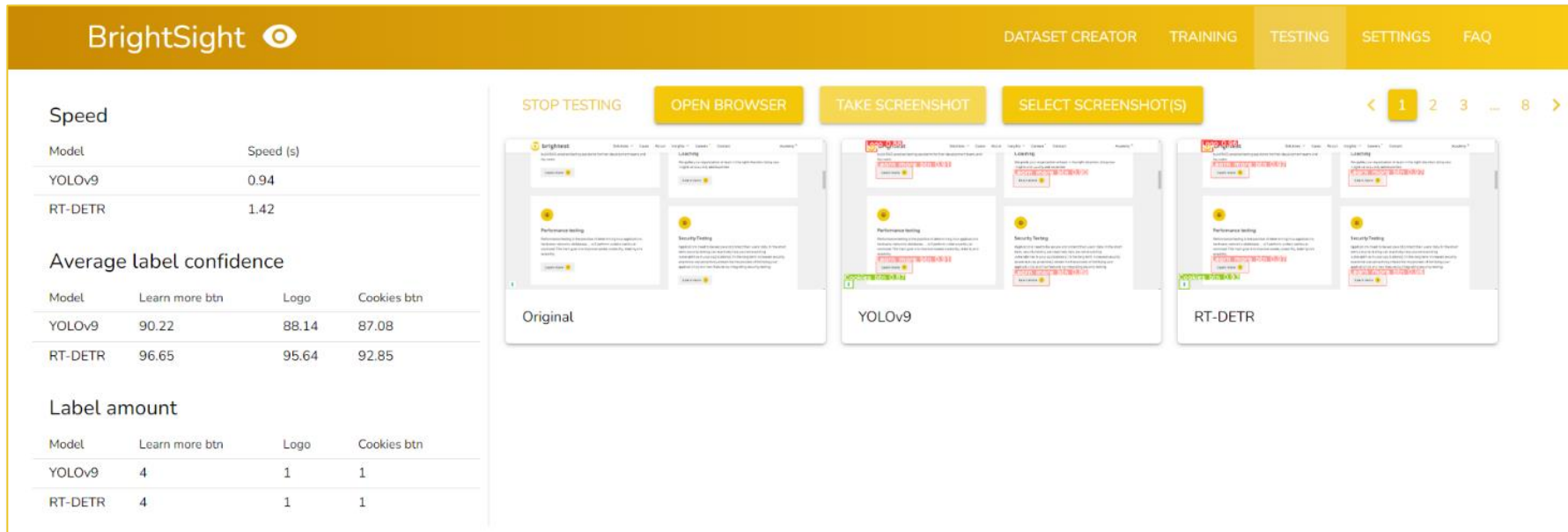
epochs: 5

batch_size: -1

image_size:

[Show code](#)

- Nieuwe run of verder gaan
- Selecteer training run
- Selecteer screenshots of open website a.d.h.v. Selenium en neem screenshots
- Bekijk de detectie output



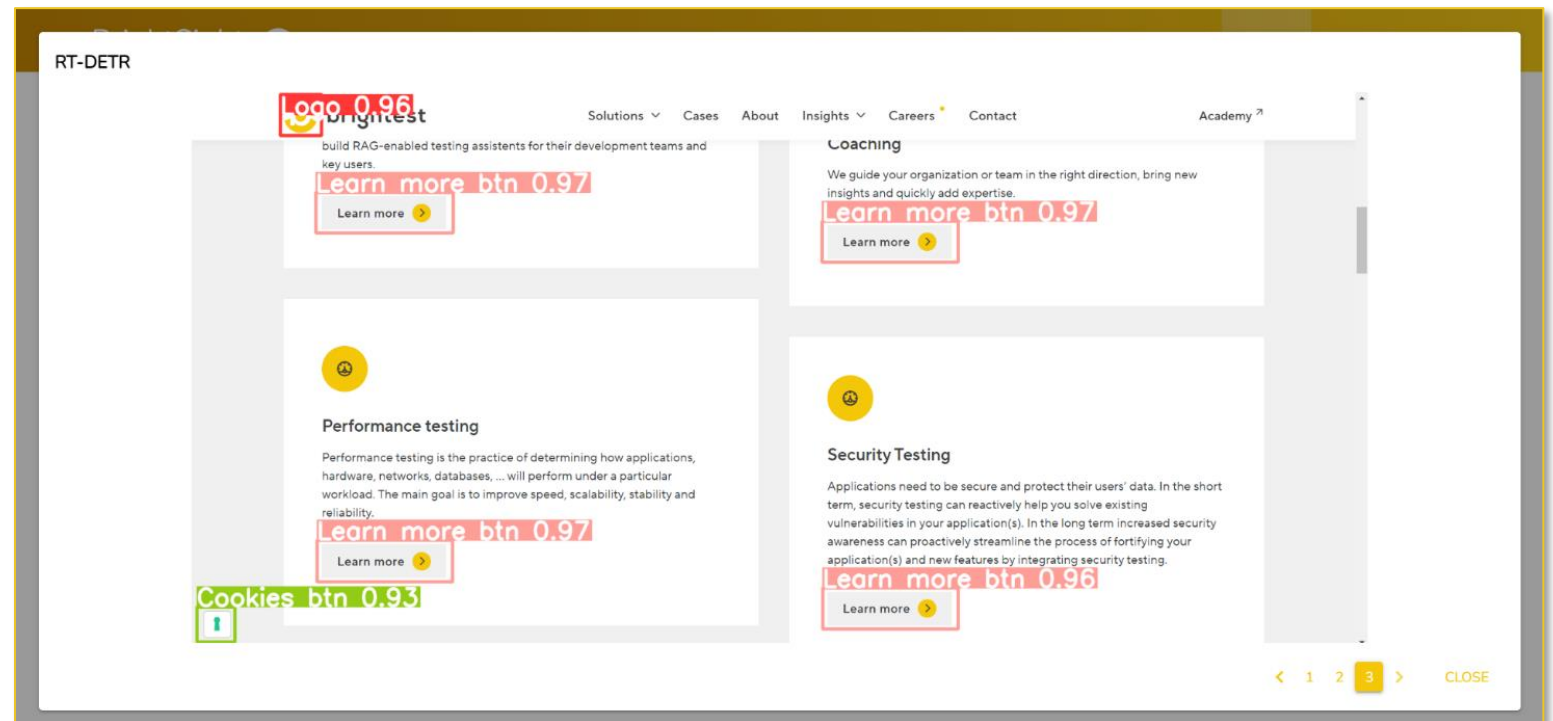
The screenshot displays the BrightSight Testing interface. On the left, there are three tables: 'Speed', 'Average label confidence', and 'Label amount'. The 'Speed' table compares YOLOv9 (0.94s) and RT-DETR (1.42s). The 'Average label confidence' table shows YOLOv9 with 90.22% confidence and RT-DETR with 96.65%. The 'Label amount' table shows both models detecting 4 objects. On the right, a 'STOP TESTING' button is followed by 'OPEN BROWSER', 'TAKE SCREENSHOT', and 'SELECT SCREENSHOT(S)' buttons. Below these are three panels: 'Original', 'YOLOv9', and 'RT-DETR', each showing a screenshot of a website with detection boxes. The 'Original' panel shows the raw screenshot, while the 'YOLOv9' and 'RT-DETR' panels show the same screenshot with red bounding boxes around detected objects.

Speed	
Model	Speed (s)
YOLOv9	0.94
RT-DETR	1.42

Average label confidence			
Model	Learn more btn	Logo	Cookies btn
YOLOv9	90.22	88.14	87.08
RT-DETR	96.65	95.64	92.85

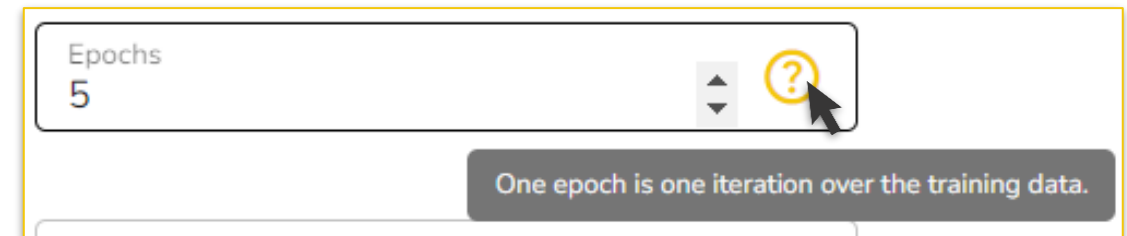
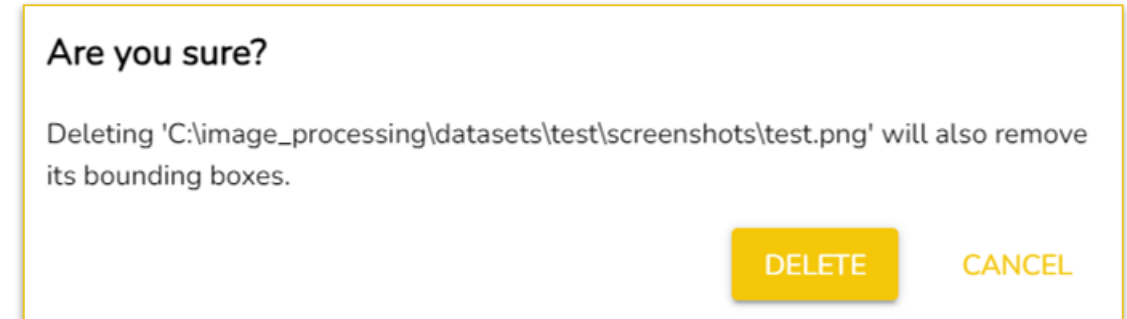
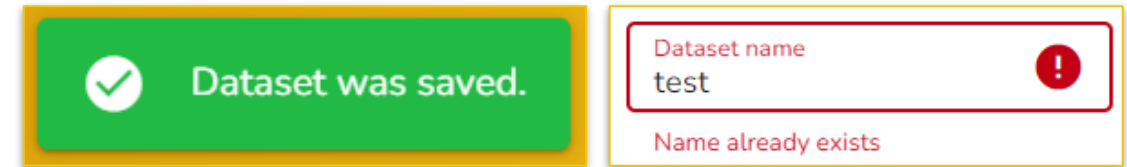
Label amount			
Model	Learn more btn	Logo	Cookies btn
YOLOv9	4	1	1
RT-DETR	4	1	1

- Nieuwe run of verder gaan
- Selecteer training run
- Selecteer screenshots of open website a.d.h.v. Selenium en neem screenshots
- Bekijk de detectie output





- Importeren & exporteren
- Positieve & negatieve feedback →
- Zoomen & slepen canvas
- Bevestigingsvensters →
- Instellingen & FAQ
- Hints →
- Training via Google Colab
- Packaged als executable





- Unit testen met PyTest
 - Testen van de functionaliteit van klassen, functies, ...
- 20 testcases
 - Functionaliteit en prestaties van software valideren en verifiëren
- Handmatig testen
 - Software interactief gebruiken om te controleren of deze voldoet aan de vereisten en specificaties



- Maak automatisch screenshot van website
- Detecteer op screenshot a.d.h.v BrightSight (code) API
 - Pad naar (getrainde) model file
 - Pad naar afbeelding
 - Label filter (optioneel)
 - → Label, positie, zekerheid van gedetecteerde elementen
- Element niet gedetecteerd → test faalt
- Element gedetecteerd → test slaagt

Samenvatting &
terugkoppeling



Samenvatting resultaten

- Modellen onderzoek
- BrightSight applicatie faciliteert volledig process
 - Dataset creatie
 - Training
 - Testing
- Getrainde modellen gebruiken in test automatisatie



- Modellen onderzoek
- BrightSight applicatie faciliteert volledig process
 - Dataset creatie
 - Training
 - Testing
- Getrainde modellen gebruiken in test automatisatie

- Gebruik maken van machine learning modellen & frameworks om patronen te detecteren op websites
- Ontwikkelen van een proof-of-concept om te testen of een bepaald patroon correct wordt weergegeven op een pagina



Hoe kan image processing, en meer specifiek object detection, gebruikt worden in het geautomatiseerd software testing proces?



- Correctheid van weergave van elementen testen
- Model training process is niet vanzelfsprekend
→ BrightSight als hulpmiddel
- Automatisatie a.d.h.v BrightSight API
- Negatief:
 - Langere testduur
 - Geen 100% zekerheid
- Positief:
 - Nog geen andere bestaande oplossing voor testen van correctheid van weergave

Bijdrage

- Grotendeels gedeeld werk met Jan Haegdorens (applicatieontwikkeling)
- Bijdrage enkel Jan
 - Unit testing
 - Setup YOLOv9 en RT-DETR
- Bijdrage enkel ikzelf
 - Vergelijking modellen
 - Setup Co-DINO
 - Dataset creator & augmentatie

Vragen?



brightest
AIM HIGH

Brightest East

Thor Park 8300 bus 6

André Dumontlaan 67, 3600 Genk

Tel: +32 89 39 59 79
info@brightest.be



BRIGHTTEST LOGO 0.99

BrightSight demo



High-level planning														
	Intro	Sprint 1		Sprint 2		Sprint 3		Sprint 4		Sprint 5		Sprint 6		Sprint 7
Week	26/2 - 03/03	04/03 - 10/03	11/03 - 17/03	18/03 - 24/03	25/03 - 31/03	01/04 - 07/04	08/04 - 14/04	15/04 - 21/04	22/04 - 28/04	29/04 - 05/05	06/05 - 12/05	13/05 - 19/05	20/05 - 26/05	27/05 - 31/05
Planning														
Research														
Implementing models														
Implementing dataset creation app														
Implementing POC														
Testing														
Analysis / Reflection / Paper														
Courses														

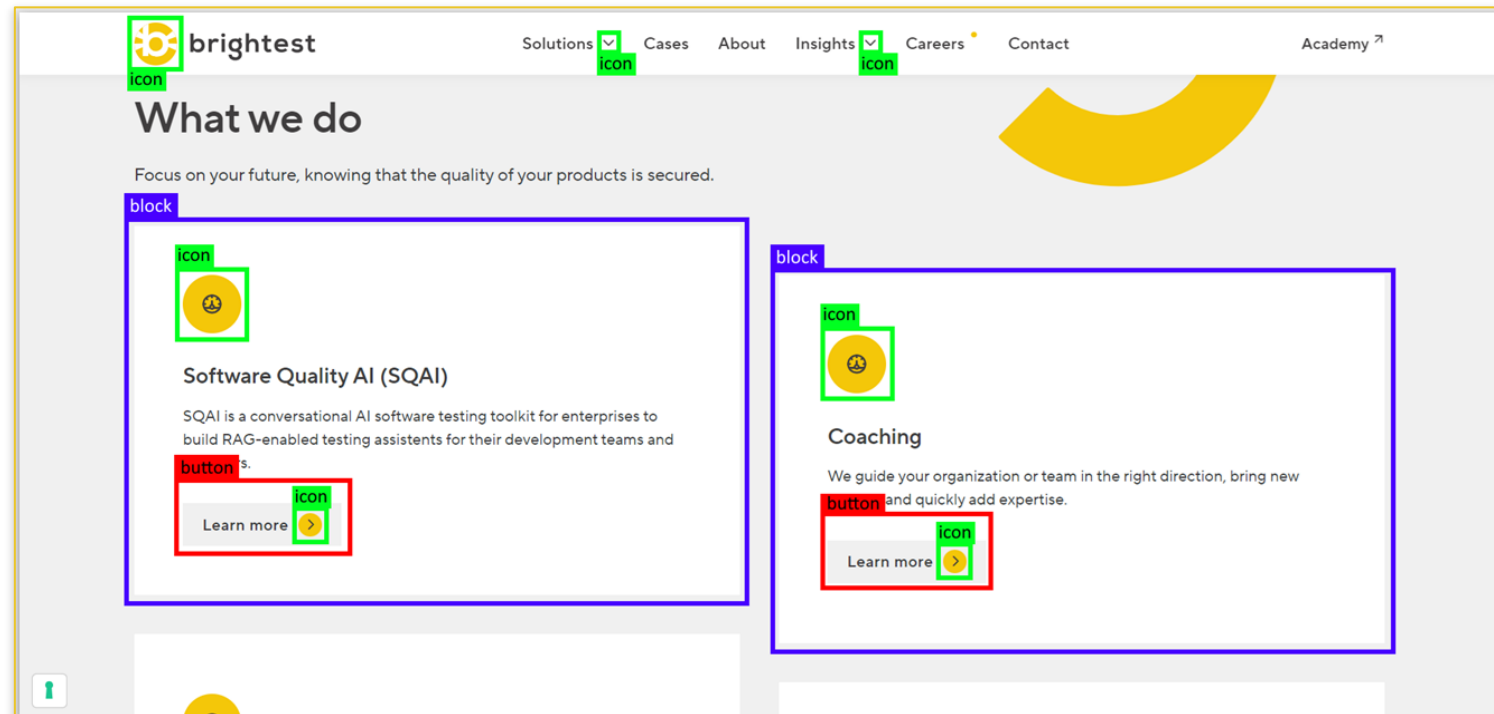


- Agile
- Twee weken per sprint
- Totaal van vijf sprints
- Schatting van tijd per topic
- Stand-ups, meetings, presentaties, en vrije dagen zijn mee gerekend

Sprint 3 (Dataset creation app)		
01/04 - 14/04		
	Duration Jan	Duration Brent
	72.00	75.00
Total (h)	147.00	
Other		
Work on school assignments	480	480
Easter Holiday	2880	960
Easter Monday	480	480
Meetings / other assignments		
Daily stand-up	60	150
Discuss/process stand-up	60	150
Technical meeting	60	120
Discuss/process technical meeting	60	60
Sprint demo / review / retrospective	60	60
Models		
Co-DINO: documentation		60
YOLO: review		120
RT-DETR: review		120
Co-DINO: review	120	
Cleanup and refactoring	60	60
Dataset creation app		
Dataset creation functionality		720
Error handling		480
Manual testing		180
Cleanup and refactoring		120
Documentation		180

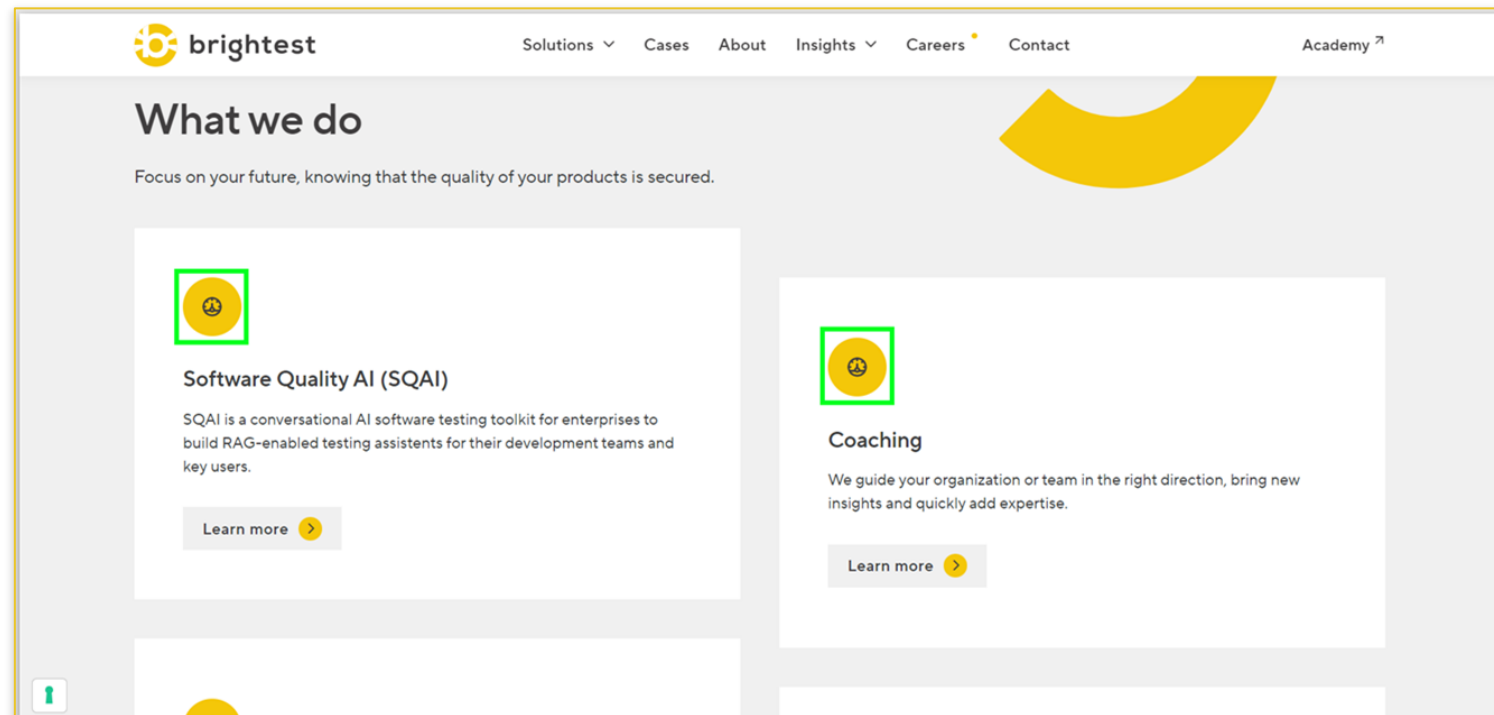


- Bestaande ML oplossingen: detecteren van categorieën van UI elementen





- Bestaande ML oplossingen: detecteren van categorieën van UI elementen
- Vereist: detecteren van (alle instanties) van een specifiek element



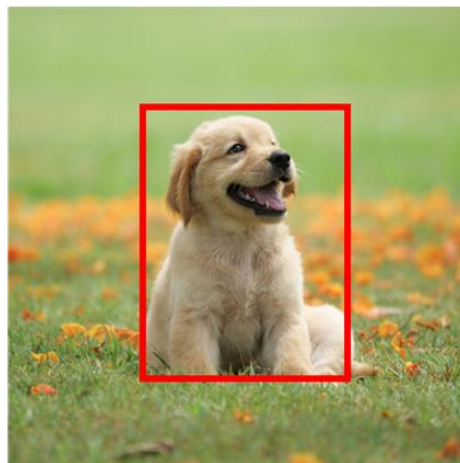


- Detecteren van bepaalde patronen in een afbeelding
- Classification vs. localization vs. detection
 - Classification: “deze afbeelding is een hond, geen kat” (multi-categorie)
 - Localization: “er is een hond op (20, 60)” (multi-positie)
 - Detection: “er is een hond op (20, 60), nog een hond op (120, 80), en een kat op (10, 30)” (multi-categorie en multi-positie)

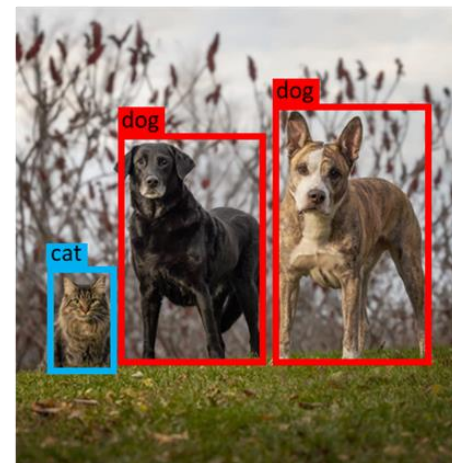
Classification



Localization



Detection





- 12 object detectie modellen (37 varianten, afbeelding niet volledig)
- Rang op basis van 3 factoren
 - Accuraatheid (op COCO set)

Lineaire mapping

	Slechtste	Beste
Accuraatheid (%)	23.2	64.1
Score	0	100

Accuracy	
Score	mAP (%)
34.47	37.30
53.06	44.90
66.01	50.20
72.62	52.90
75.06	53.90
57.70	46.80
68.95	51.40
72.86	53.00
79.22	55.60
72.86	53.00
77.26	54.80
100.00	64.10
86.31	58.50
33.74	37.00
41.56	40.20
45.97	42.00
48.41	43.00



- 12 object detectie modellen (37 varianten, afbeelding niet volledig)
- Rang op basis van 3 factoren
 - Accuraatheid (op COCO set)
 - Snelheid

Lineaire mapping

	Slechtste	Beste
Snelheid (miljarden FLOPs)	1782	8.4
Snelheid backup (FPS)	5.88	161
Score	0	100

Speed		
Score	FLOPs (B)	FPS
100.00	8.70	
98.88	28.60	
96.06	78.60	
91.17	165.20	
85.95	257.80	
98.98	26.70	
96.16	76.80	
94.69	102.80	
89.64	192.50	
94.29	110.00	
87.29	234.00	
84.76	279.00	
51.99	860.00	
75.28	447.00	10.20
13.17		26.32
8.85		19.61
2.79		10.20



- 3 categorieën

Availability	
Score	How
100.00	lib, no data loading
100.00	lib, no data loading
100.00	lib, no data loading
100.00	lib, no data loading
100.00	lib, no data loading

100.00	lib, no data loading
100.00	lib, no data loading
100.00	lib, no data loading
100.00	lib, no data loading

100.00	lib, no data loading
100.00	lib, no data loading

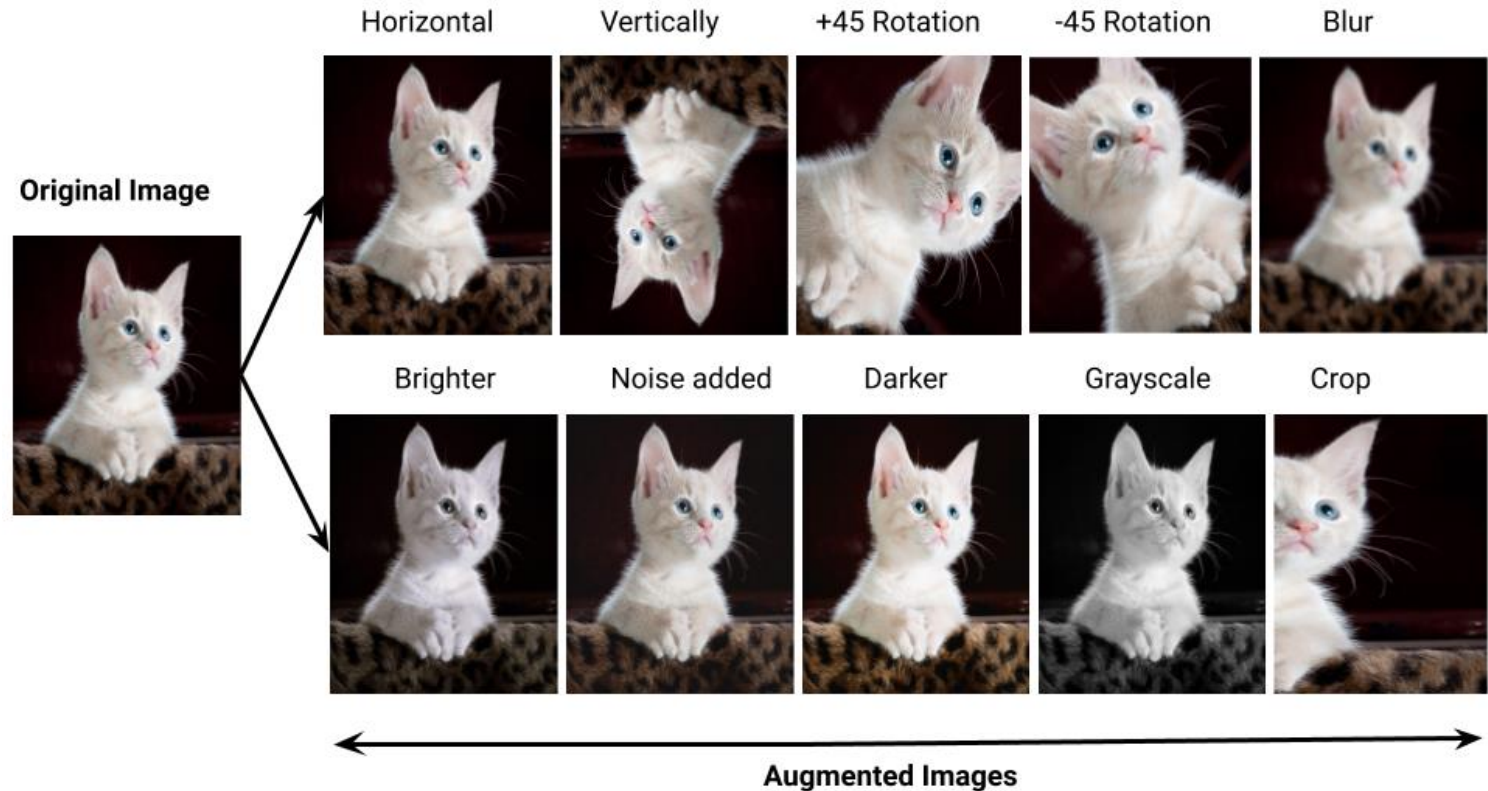
0.00	download
0.00	download

50.00	lib
50.00	lib
50.00	lib
50.00	lib



Data augmentatie

- Vergroten van dataset
- Variatie introduceren
- Manieren:
 - Schaal aanpassen
 - Bijsnijden
 - Verplaatsen
 - Noise toevoegen
 - Kleur aanpassen
 - Spiegelen
 - Roteren
 - ...





Data augmentatie

- Vergroten van dataset
- Variatie introduceren
- Manieren:
 - Schaal aanpassen
 - Bijsnijden
 - Verplaatsen
 - Noise toevoegen
 - ~~Kleur aanpassen~~
 - ~~Spiegelen~~
 - ~~Roteren~~
 - ...

