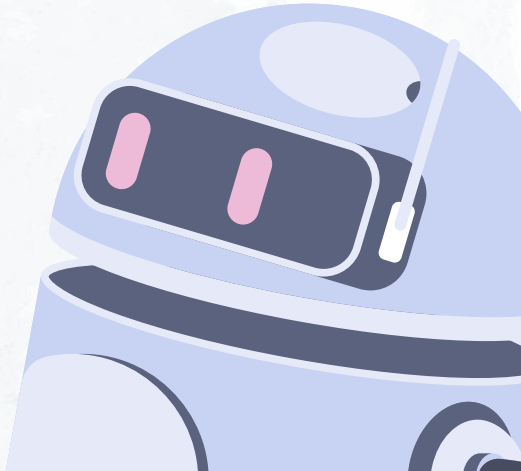
A Dataset Is All You Need

Bastien Pouëssel - Quentin Fisch - Arnaud Baradat - Théo Ripoll - Tom Genlis - Nicolas Fidel



EPITA - Deep Neural Networks

Table of contents

01 →

The Dataset

Visible Watermarks detection datasets

- A very underrepresented task
- Few datasets exists, most are very bad for many different reasons
- There is a need to create a well generated dataset for this task
- Other datasets main issues
    - **Unrealistic logo placement**
    - **Very few images**
    - **No text, only logos**
    - **Made for segmentation (no boxes)**
    - **Hosted on weird chinese websites**

=> Our goal = create reference dataset for this task

Our dataset → PITA dataset

(a) Logos + text watermarks ⟶

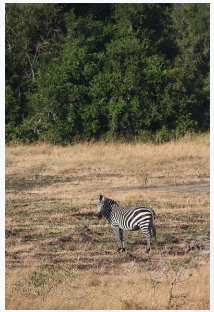
- Pool of 350 logos from a Logo dataset
- Infinite text generation with different fonts, size, color and opacity
- Classification task on top of the detection task

(b) 2 formats supported ⟶

- Coco format
- YOLO (Ultralytics) format
- Based on COCO images
- Hosted on Hugging Face = easier download
- Python package to generate images locally or download from HF
- 20,000 total images
- **CLI Tool** for dataset generation and downloading

Our dataset → PITA dataset

(a) Coco Dataset

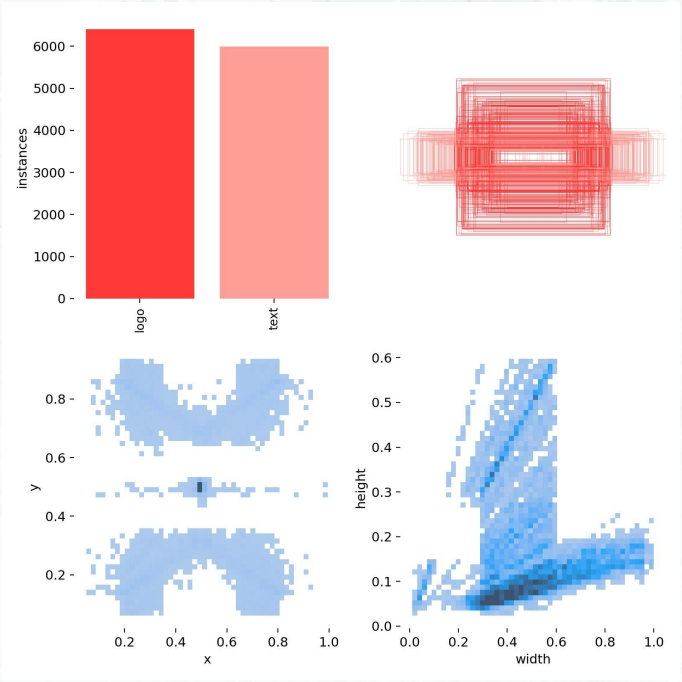


(b) Watermark

512x512

Our dataset → PITA dataset

- See on Hugging Face at **bastienp/visible-watermark-pita**

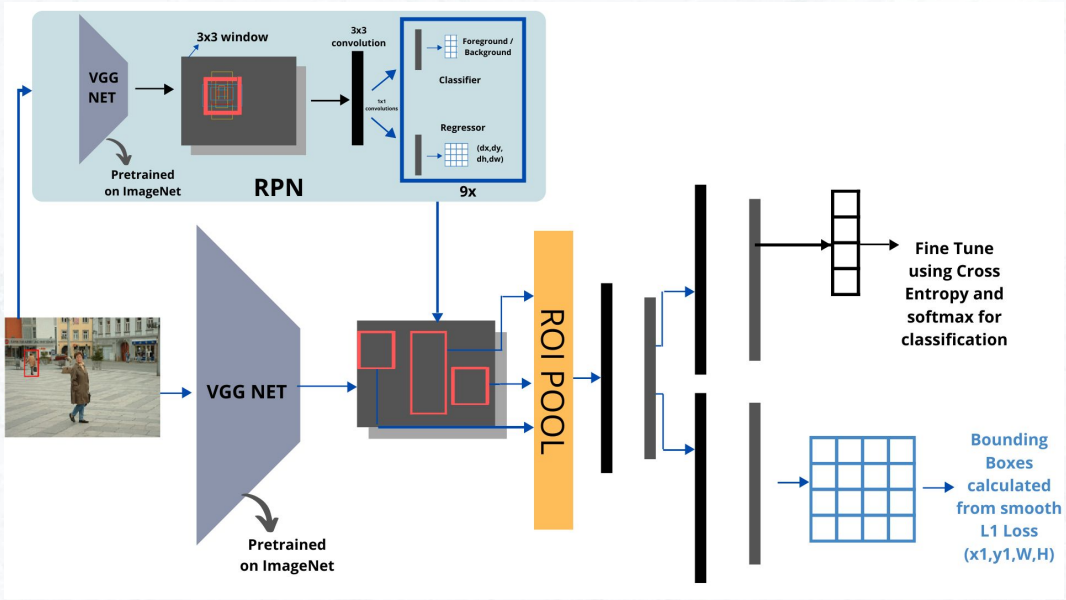


02 →

Models

Faster R-CNN

Faster R-CNN is a two-stage object detection model composed of a Region Proposal Network (RPN) and an Object Detection Network



(a) Faster RCNN V2

- 43.3 M Total params
- Fine-tuned on 5 epochs with Pytorch Lightning

https://towardsdatascience.com/understanding-fast-r-cnn-and-faster-r-cnn-for-object-detection-adbb55653d97

YOLO

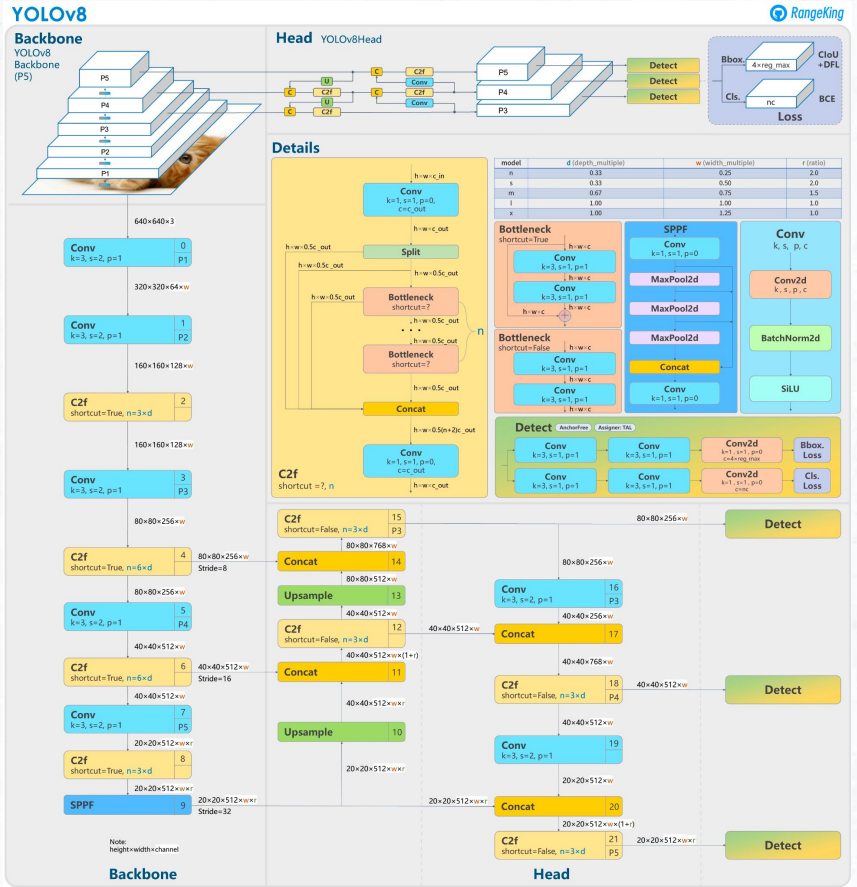
We used Ultralytics models to fine-tune YOLO. PITA dataset's YOLO format is designed for Ultralytics train format

(a) YoloV8 Nano

- 3.2M parameters
- Fine-tuned on 15 epochs

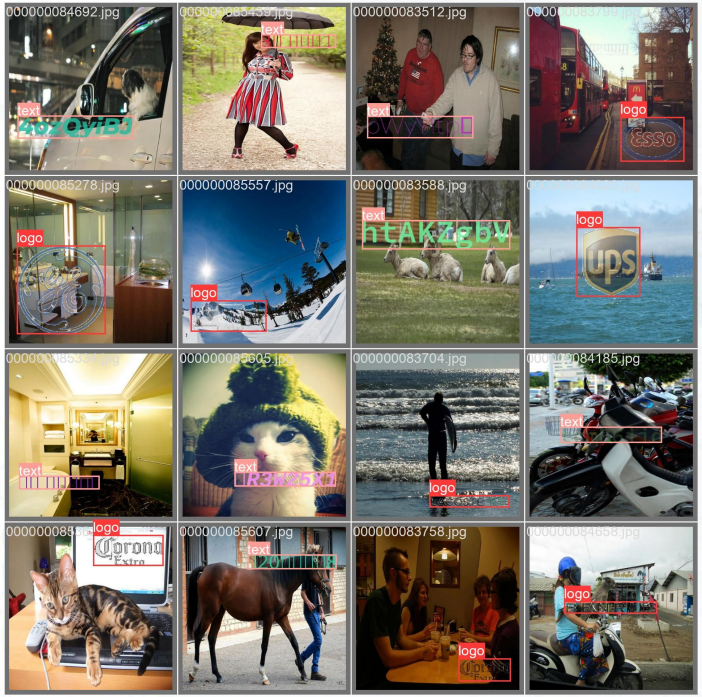
(b) YoloV8 Large

- 43M parameters
- Fine-tuned on 15 epochs

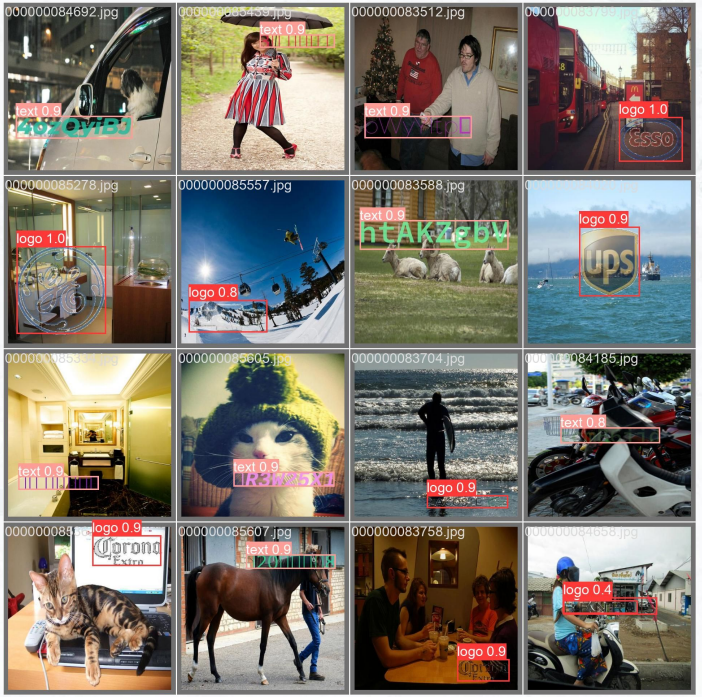


https://blog.roboflow.com/whats-new-in-yolov8/

YOLO - results



Labels



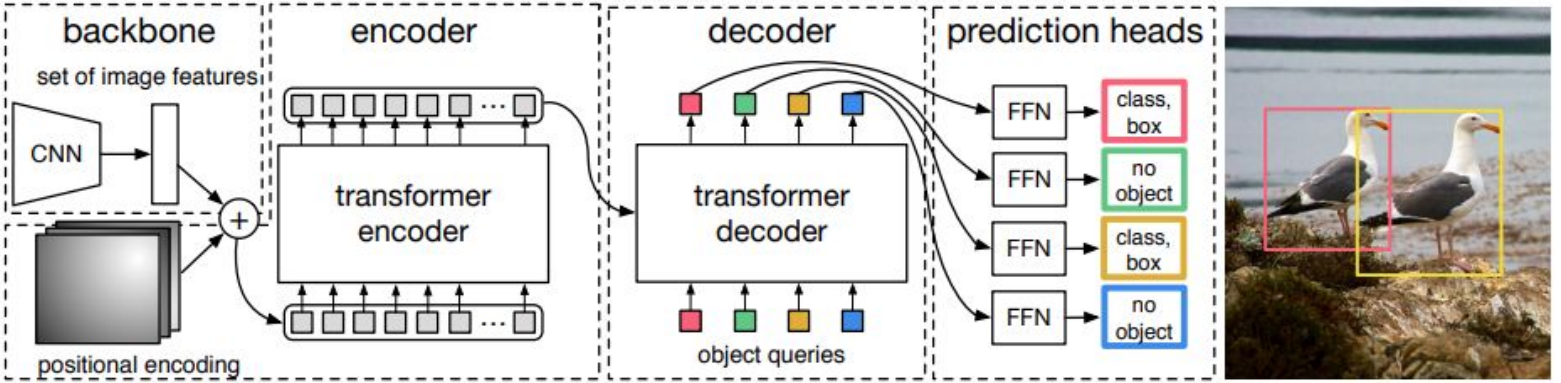
Predictions

DeTR

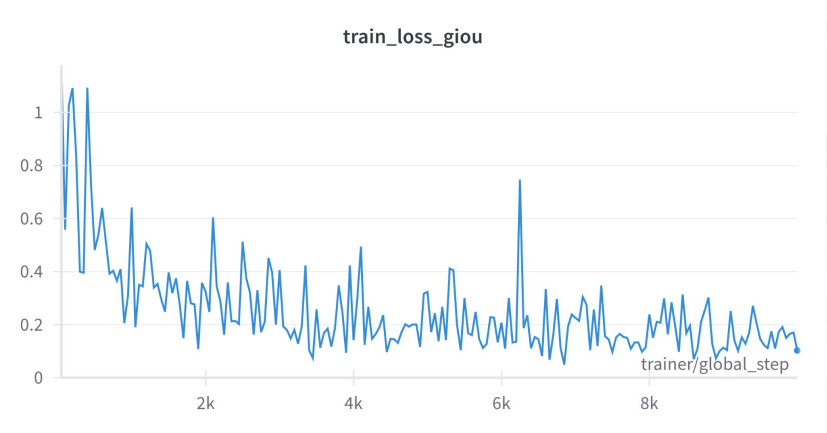
- Detr uses a transformer to reason between the image features and the prediction, and typically a ResNet50 backbone for low dimensional representation of the image.

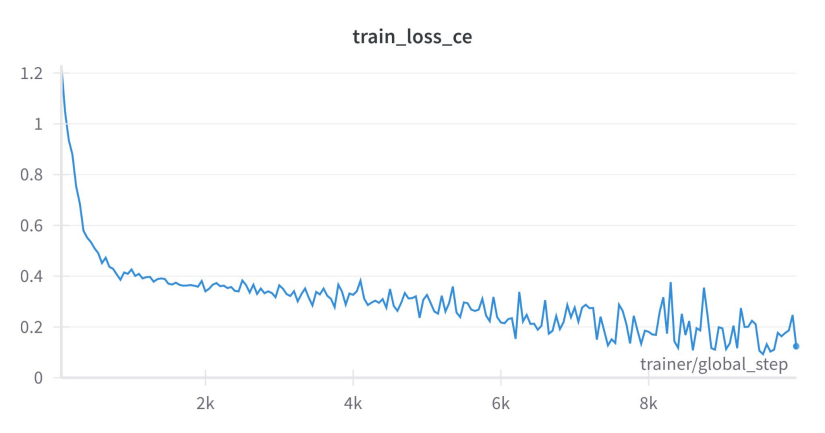
- Transformer (six layers): ~66M parameters
- Backbone (ResNet-50): ~23.6M parameters

Fine-tuned on 5 epochs



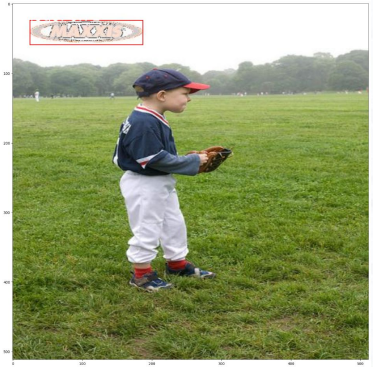
DeTR - Results





We managed to make the network converge, however we couldn’t achieve any results when we benchmarked the model.

DeTR - Results



03 →

Benchmark and results

CLWD Dataset

- Initially created for watermark segmentation / removal
- Contains only logo watermarks
- Samples at the frontier of visible watermarks
- Samples with ultra stretched logos, far from real life use cases

Results

- **MAP**: Global Mean Average Precision (across all MAP metrics)
- **MAP 50**: MAP with an IoU threshold of 50%
- **MAP 75**: MAP with an IoU threshold of 75%
- **MAP per class**: MAP separated for logos and texts

Table 1: Benchmark on OUR dataset VS CLWD dataset

Metric	map	map_50	map_75	map_per_class
PITA Dataset				
DeTR	–	–	–	–
FasterR-CNN	**0.9005**	**0.9839**	**0.9644**	[**0.9108**, 0.8798]
YoloV8n	0.8750	0.9690	0.9415	[0.8643, 0.8858]
YoloV8l	0.8900	0.9741	0.9473	[0.8783, **0.9018**]
CLWD Dataset				
DeTR	–	–	–	—
FasterR-CNN	0.0005	0.0019	0.0001	—
YoloV8n	**0.0190**	**0.0394**	**0.0160**	—
YoloV8l	0.0082	0.0130	0.0063	—

04 →

Demo