Group

- Filippo Casini
- Giovanni Zara

Computer Vision Project

Outline

2 Tasks →

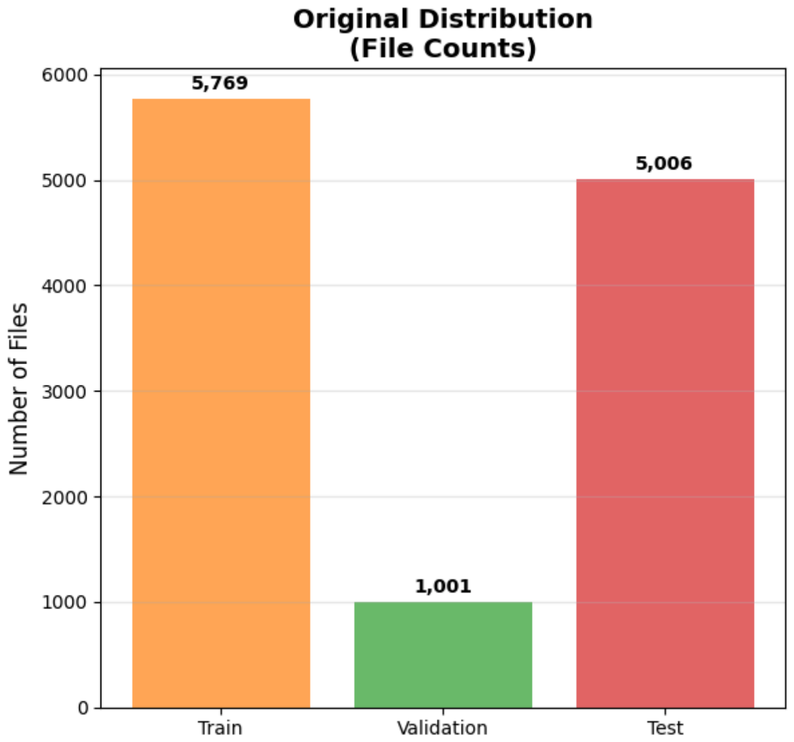
- Baselines
  1. Ground truth + Custom Recognition Model
  2. Custom Detection Model + PDLPR
  3. Custom Detection Model + Custom Recognition Model
- Paper Implementation (Yolo + PDLPR)

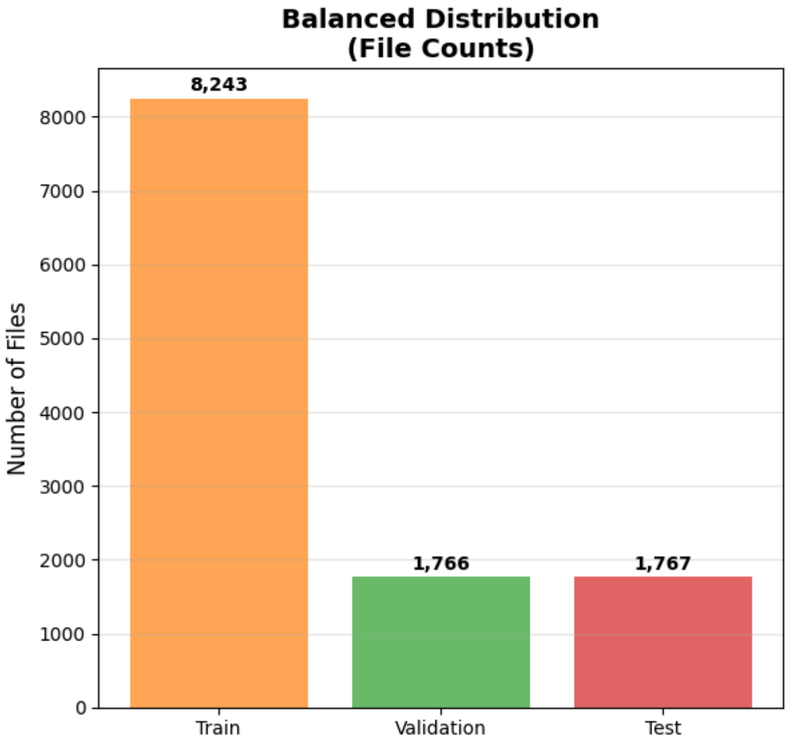
Baselines:

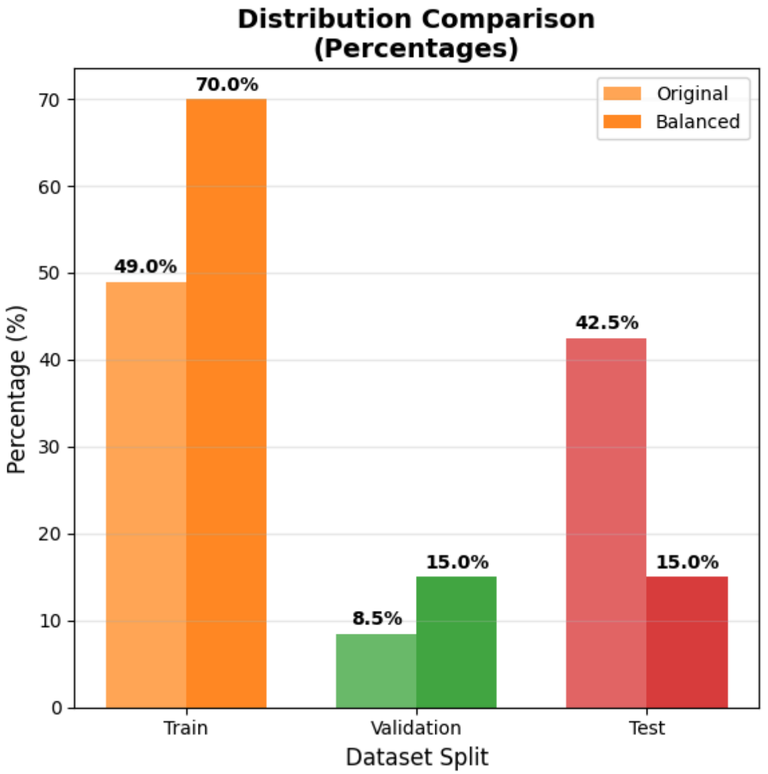
1. State of the art + Custom Model
2. Custom Model + Sate of the art
3. Full custom Pipeline

Data

We used the CCPD_green dataset, containing about 11k samples rebalanced in the following way:



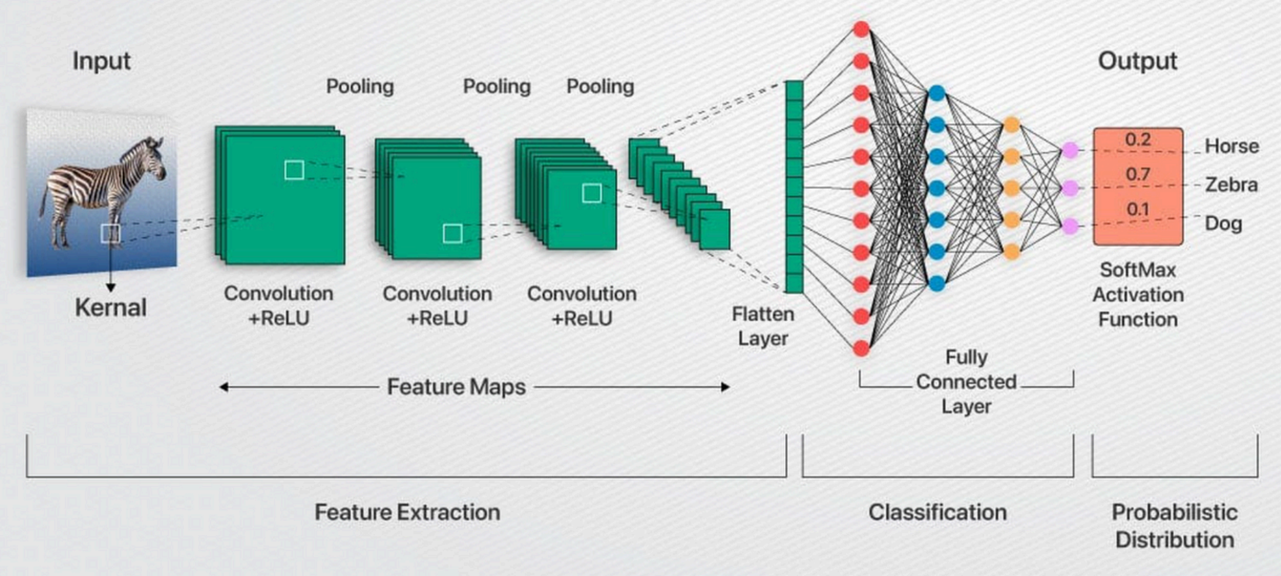




Custom Detection Model

Fine tuning of fasterrcnn_resnet50_fpn with:

- 500 proposed regions
- 2 classes (1 for the car plate and 1 for the background)
- Intersection over Union to evaluate the results
- 1 training epoch



Average Intersection over Union on Test Set = 0.87

Custom Detection Model

Example of the Model

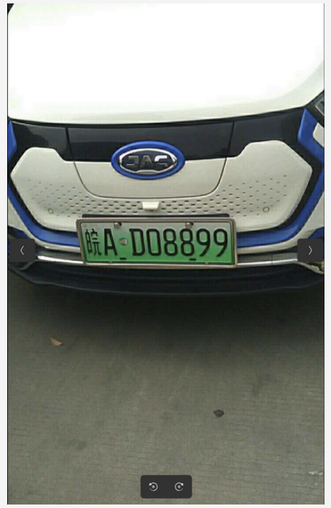
Ground Truth Bounding Box

Predicted Bounding Box

Custom Detection Model

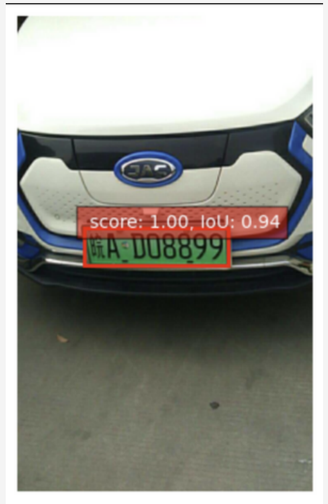
Pipeline

Original Image



Detection Model

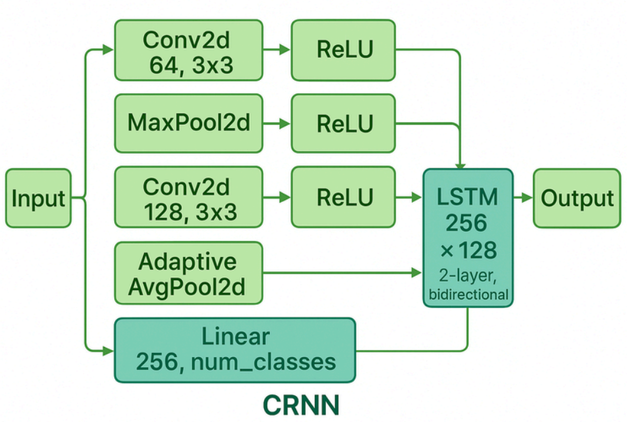
Bounding Box

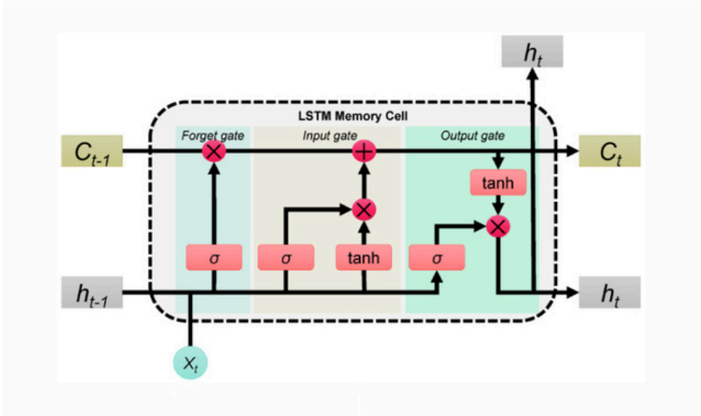


Cropped Image

Custom Recognition Model

Architecture and Pipeline





LSTM architecture

Training Parameters: 1.177.925

Cropped Image

Recognition Model

Predicted Text

皖AD08899

First Baseline

Ground truth + Custom Recognition Model

For this baseline we simply cropped the images using the ground truth and applied the Custom Recognition Model.
The goal of this Baseline is to show how important is the recognition part with respect to the detection one.

Cropped image with GT

Recognition Model

Predicted Text

皖AD35199

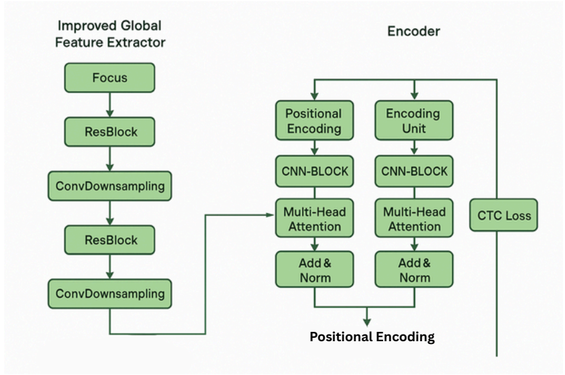
Second Baseline

Custom detection Model + PDLPR

For this baseline we simply cropped the images using the detection model and applied PDLPR, the state of the art recognition model introduced in the paper.
The goal of this Baseline is to show how important is the detection part with respect to the recognition one.

Cropped Image with detection Model

PDLPR



Predicted Text

皖AD08899

Third Baseline

Custom Detection Model + Custom Recognition Model

For this baseline we simply cropped the images using the detection model and applied the Custom Recognition Model.
The goal of this Baseline is to test how well our models were performing with respect to the paper implementation.

Cropped Image with detection Model

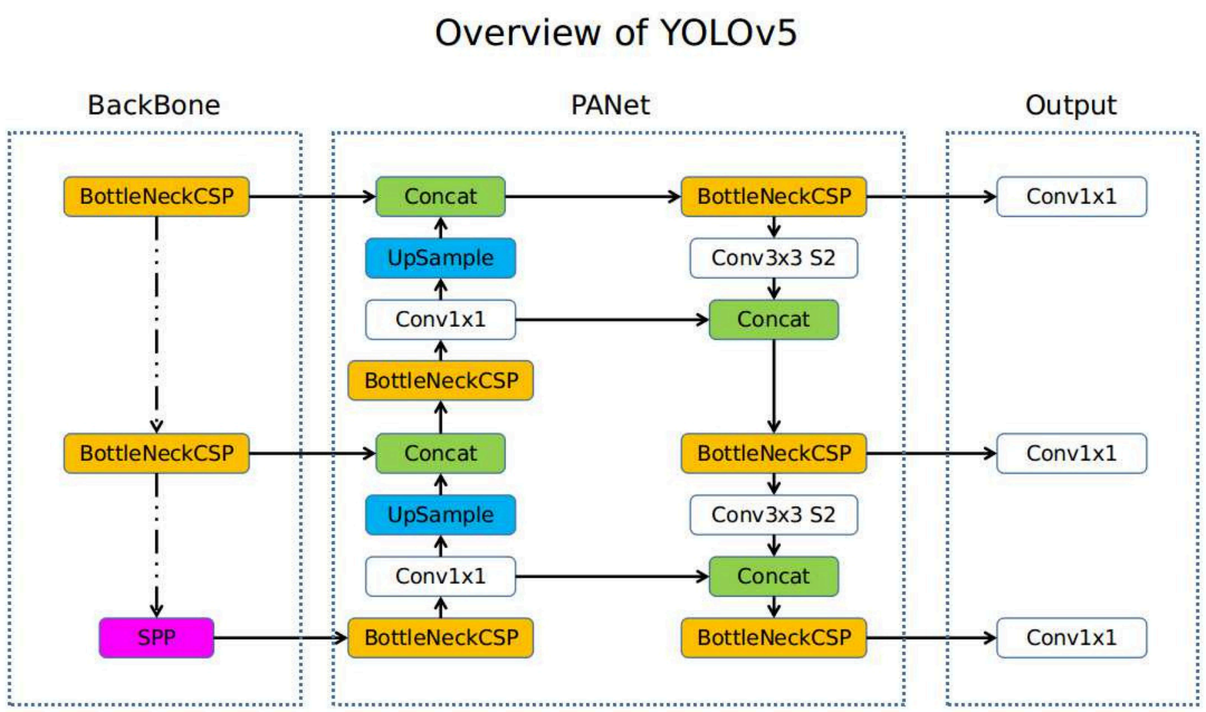
Recognition Model

Predicted Text

皖AD08899

Yolo
Finetuning

We finetuned Ultralytics YoloV5 to distinguish between car plate and background , applying a light data pre-processing



Pretrained Checkpoints

Model	AP^{val}	AP^{test}	AP_{50}	$Speed_{GPU}$	FPS_{GPU}	params	FLOPs
YOLOv5-s (ckpt)	35.5	35.5	55.0	**2.1ms**	**476**	7.1M	12.6B
YOLOv5-m (ckpt)	42.7	42.7	62.4	3.2ms	312	22.0M	39.0B
YOLOv5-l (ckpt)	45.7	45.9	65.1	4.1ms	243	50.3M	89.0B
YOLOv5-x (ckpt)	**47.2**	**47.3**	**66.6**	6.5ms	153	95.9M	170.3B
YOLOv3-SPP (ckpt)	45.6	45.5	65.2	4.8ms	208	63.0M	118.0B

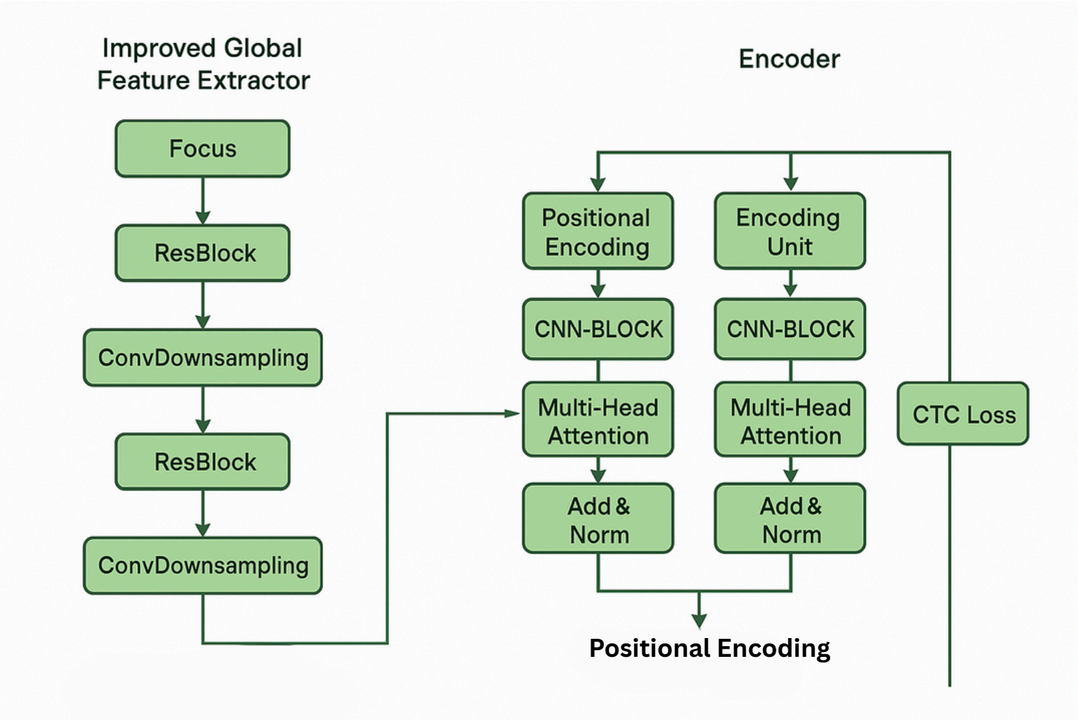
Yolo
Metrics

Yolo FineTuned	Metrics
Number of Parameters	7.012.822
GFLOPs	15.8
Maximum Average Precision (IoU>0.5)	0.995
Maximum Average Precision (0.9<IoU<0.95)	0.902
Instances per Image	1

PDLPR

Implementation

We reproduced the architecture described in the paper simplifying some over complex passages while trying to maintain an accurate text recognition pipeline.

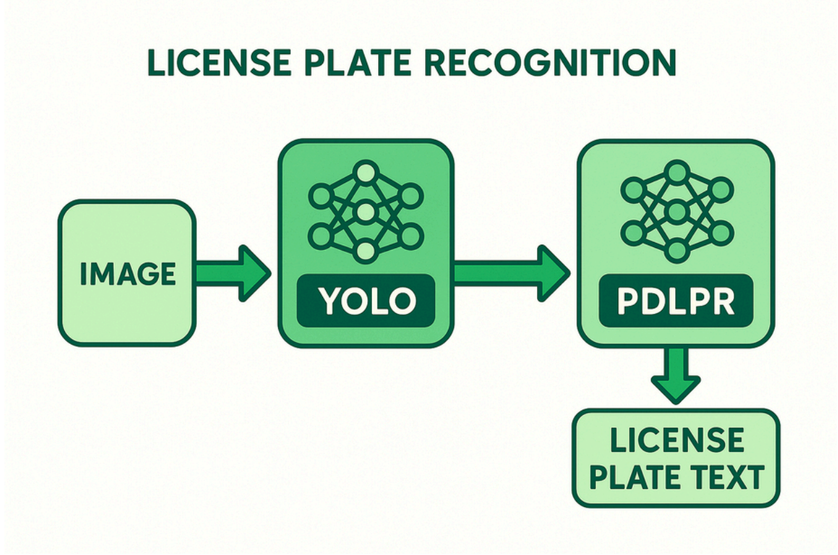


Paper Implementation

As said earlier, the detection part has been carried over by using a lightweight Yolo version, while the recognition part has been simplified by using the CTC Loss instead of a complex Auto Regressive Decoder.
We have been able to do so due to a limited-domain dataset and a less flexible approach with respect to the one described in the paper.

Online Implementation

Since this model should be deployed in real time applications we also created an online version that takes the original image and extracts the text in a direct way.



Results

Models	Plate Accuracy on Test Set	Character Accuracy on Test Set
First Baseline	0.8970	0.9680
Second Baseline	0.8273	0.9670
Third Baseline	0.8975	0.9669
Paper Implementation	0.9184	0.9778

More Metrics

Models	Province Accuracy	Alphabet Accuracy	Length Error Rate
First Baseline	0.9904	0.9904	0.0424
Second Baseline	0.9853	0.9836	0.0102
Third Baseline	0.9987	0.9881	0.0470
Paper Implementation	0.9911	0.9916	0.0006

Paper Vs Our Implementation

Metrics	Paper
Plate Accuracy on Test Set	0.9940
Speed (FPS)	159.8

Metrics	Our Paper Implementation
Plate Accuracy on Test Set	0.9184
Speed (FPS)	110

Thanks For Attention