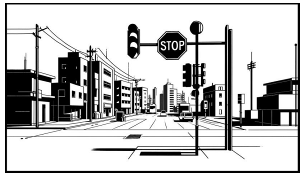
GEO-LOCATOR



ROME

THE ERDŐS INSTITUTE
Helping PhDs get jobs they love.
Helping you hire the PhDs you need.

FRANCESCA BALESTRIERI
ZACK BEZEMEK
DANTE BONOLIS
LEONHARD HOCHFILZER
AASHRAYA JHA

MOTIVATING PROBLEM

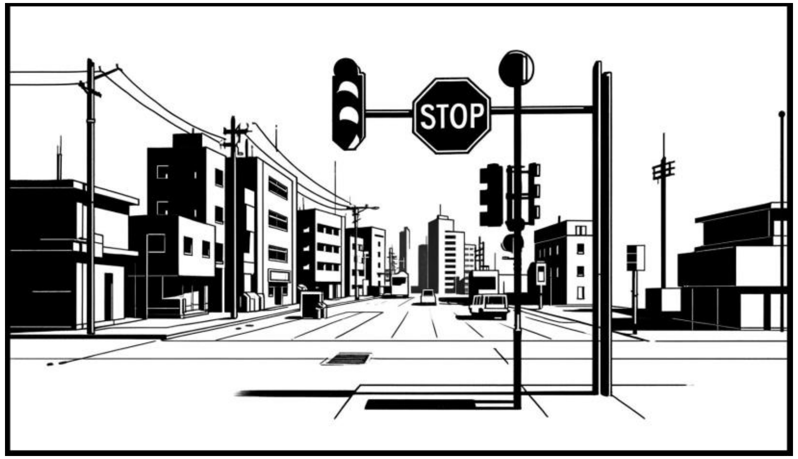
ROME OR MADRID?

MADRID OR ROME?

MADRID OR ROME?

ROME OR MADRID?

OUR APPROACH



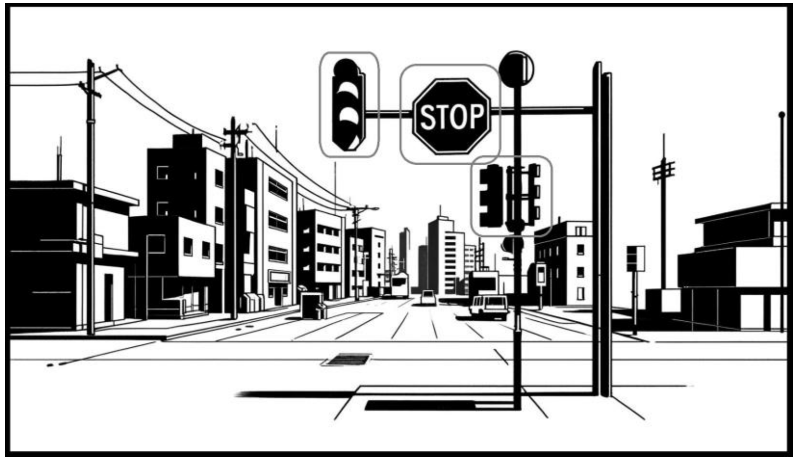
AN IMAGE IS QUITE
UNSTRUCTURED

ROME ???

EXPERT DOMAIN KNOWLEDGE

INSPIRED BY GEOGUESSR PROS:
they use **man-made** homogeneous/persistent **features highly specific to each country** as discriminants

BY ALSO CONSIDERING SPECIFIC FEATURES (IF AVAILABLE), WE IMPOSE **EXTRA STRUCTURE** ON THE IMAGE



THE DATASET

GSV-CITIES arxiv:2210.10239 / Neurocomputing 2022

- It contains ~530k images, across 23 different cities
- There are more than 62k different places, spread across multiple cities
- Each place is depicted by at least 4 images (up to 20 images)
- All places are physically distant (at least 100 meters between any pair of places)

EXAMPLE OF IMAGE METADATA:

	place_id	year	month	northdeg	city_id	lat	lon	panoid
0	1678	2014	10	370	Barcelona	41.402066	2.198988	DB4DzlzCRq4lyE9FMx_9Ow

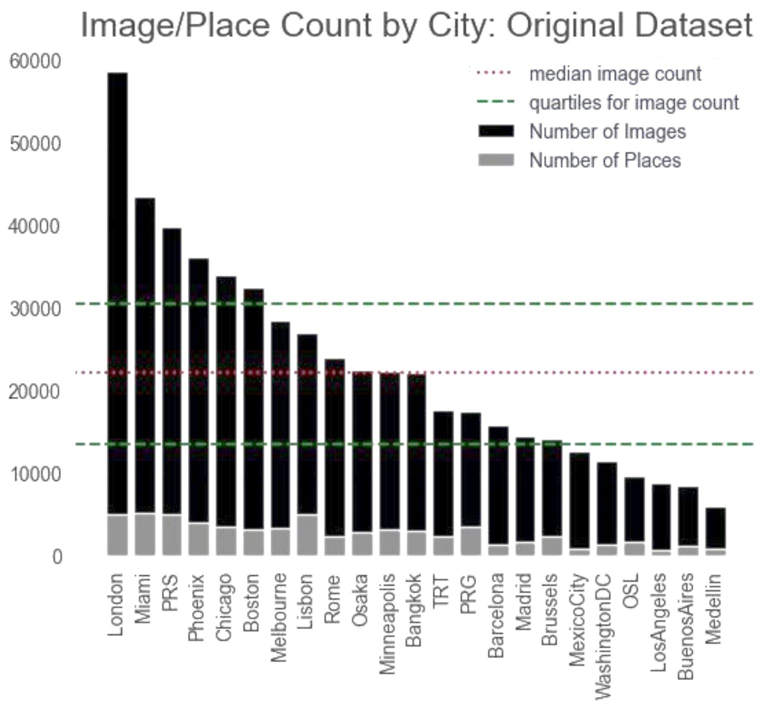
EXAMPLE OF A PLACE (BARCELONA, PLACE ID 17801):

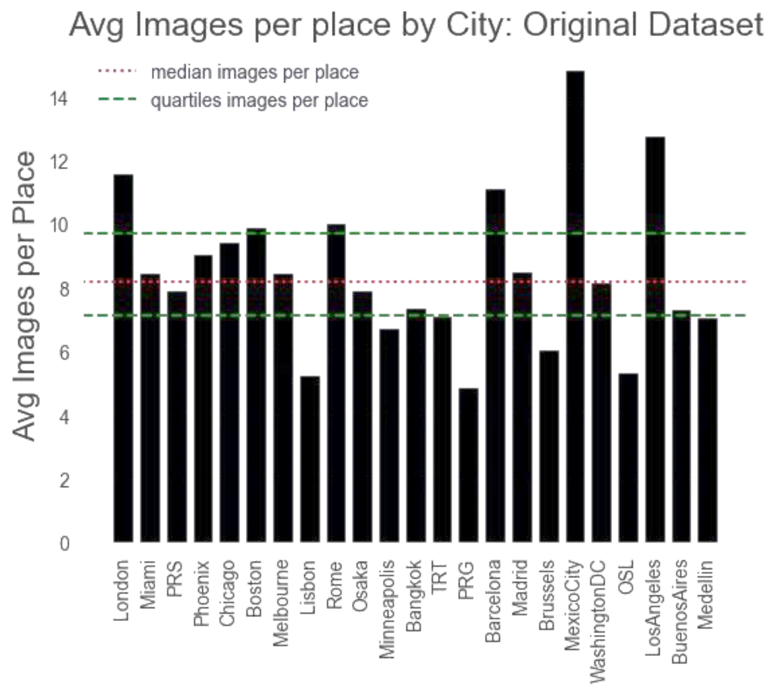
ORIGINAL DATASET

23 cities

Total number of images = **529506**

Total number of places = **64394**





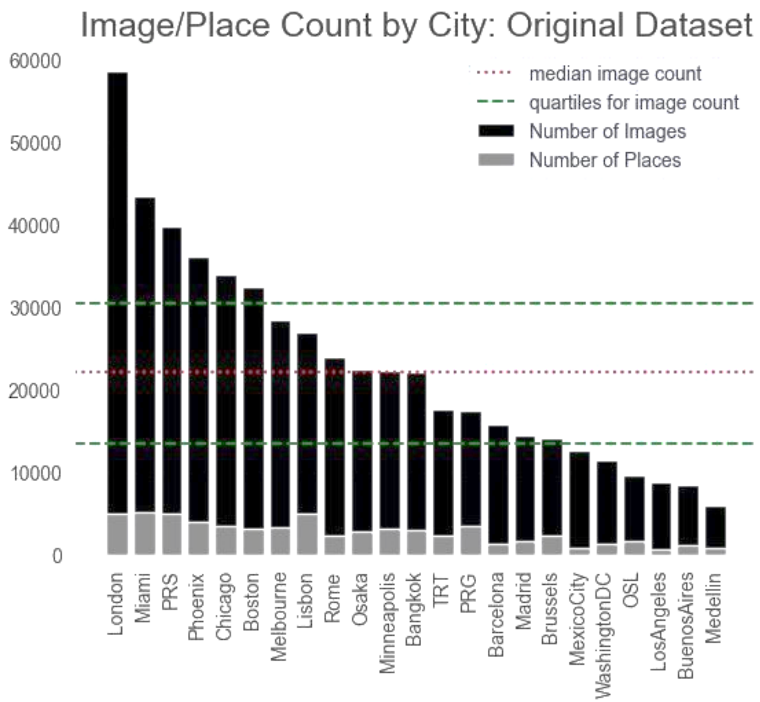
ORIGINAL DATASET

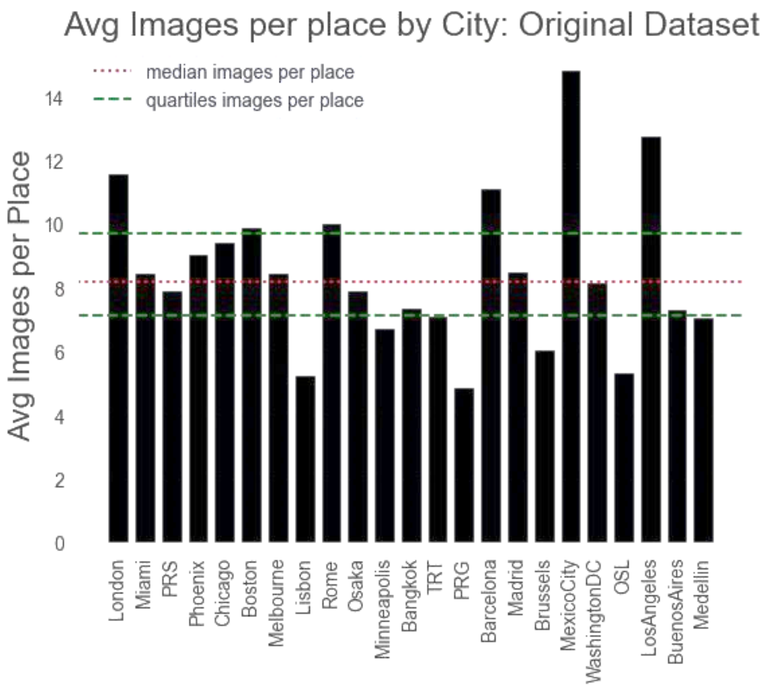
23 cities

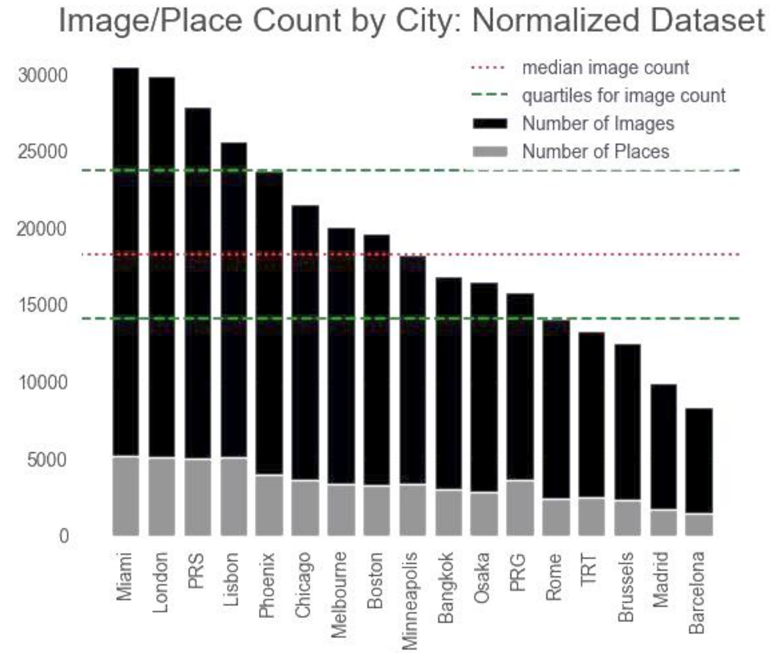
Total number of images = **529506**

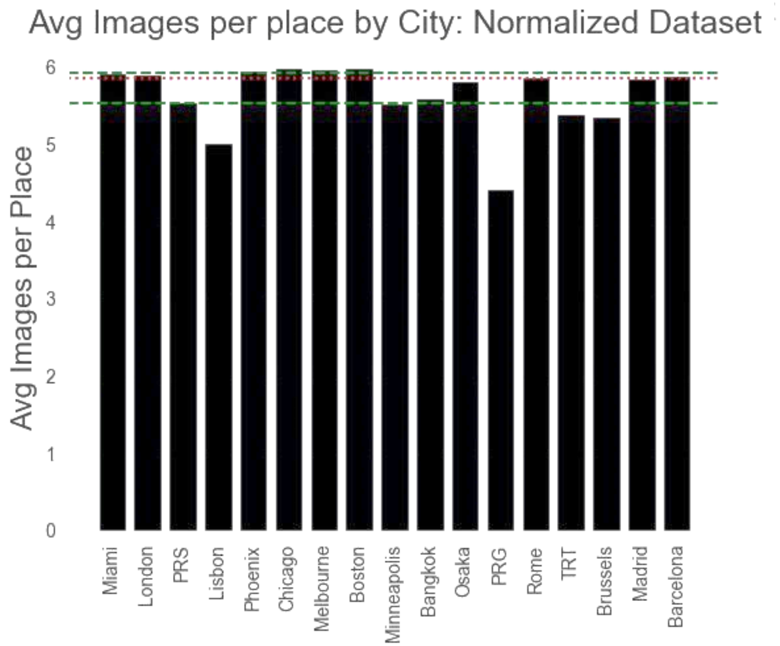
Total number of places = **64394**

CLEANING AND NORMALIZATION









BALANCED DATASET

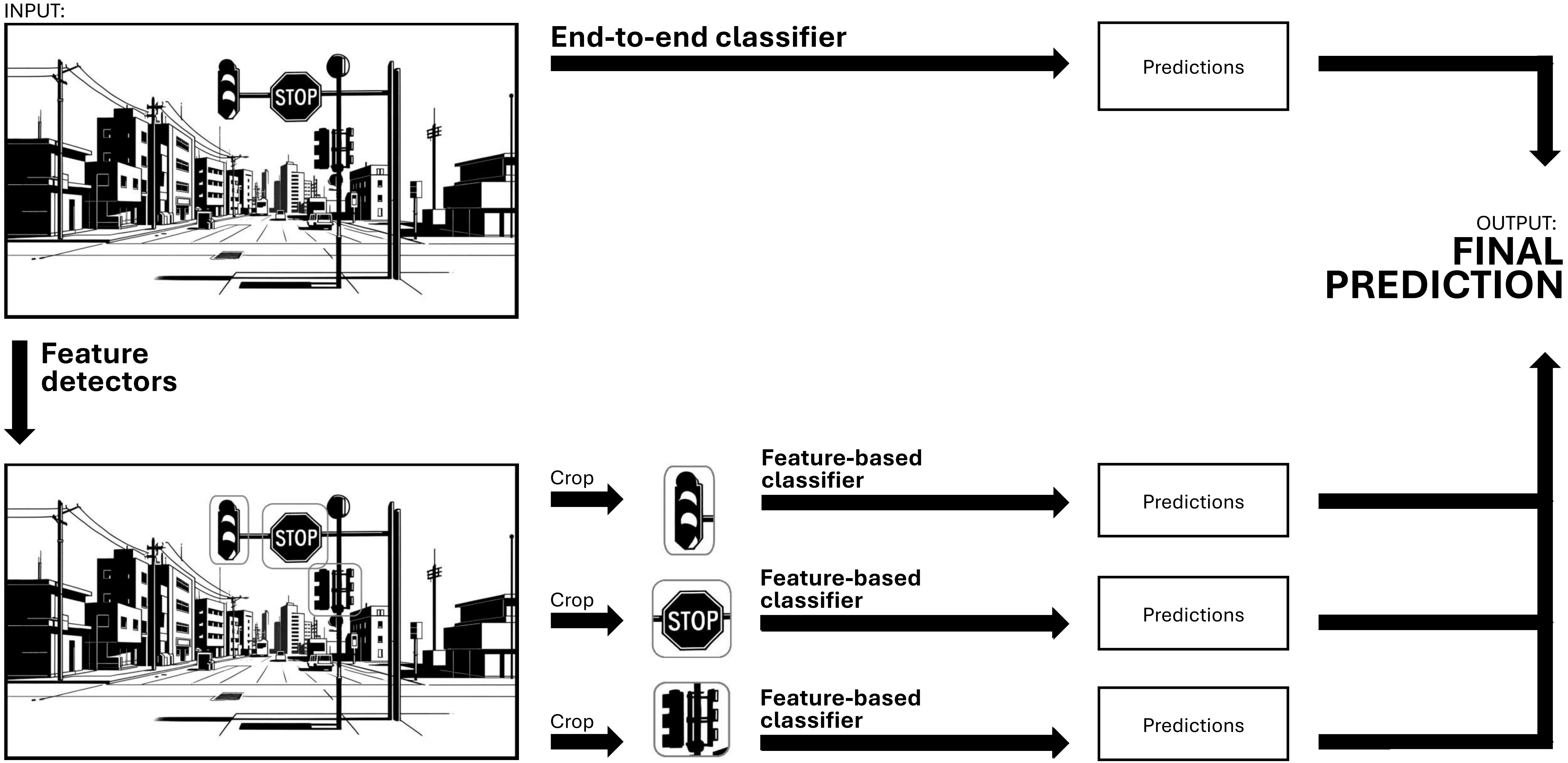
17 cities

Total number of images = **324697**

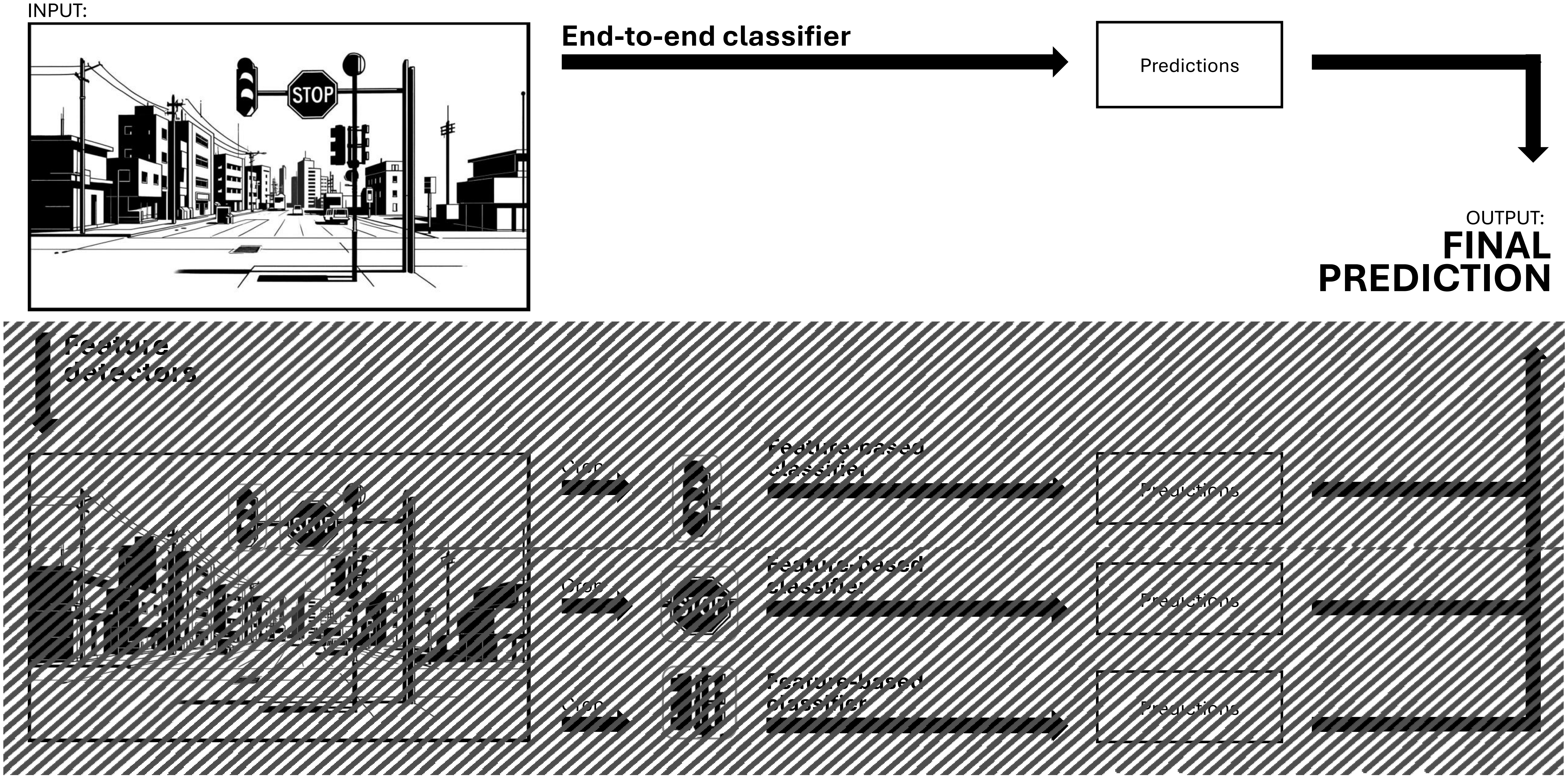
Total number of places = **57618**

THE WORKFLOW

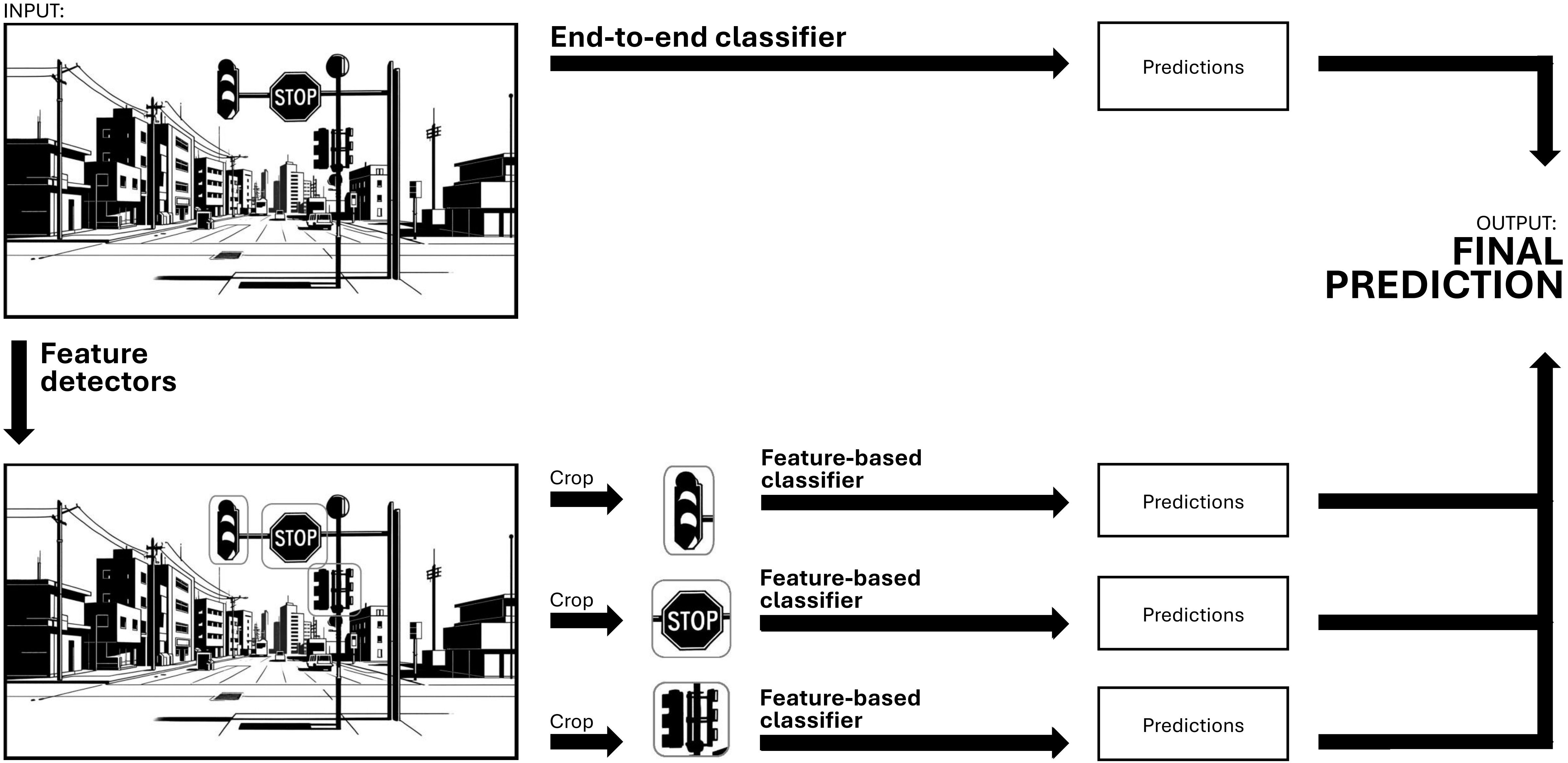
WORKFLOW


INPUT:
End-to-end classifier
Predictions
OUTPUT:
FINAL PREDICTION
Feature detectors
STOP
Crop
Feature-based classifier
Predictions
Crop
STOP
Feature-based classifier
Predictions
Crop
Feature-based classifier
Predictions

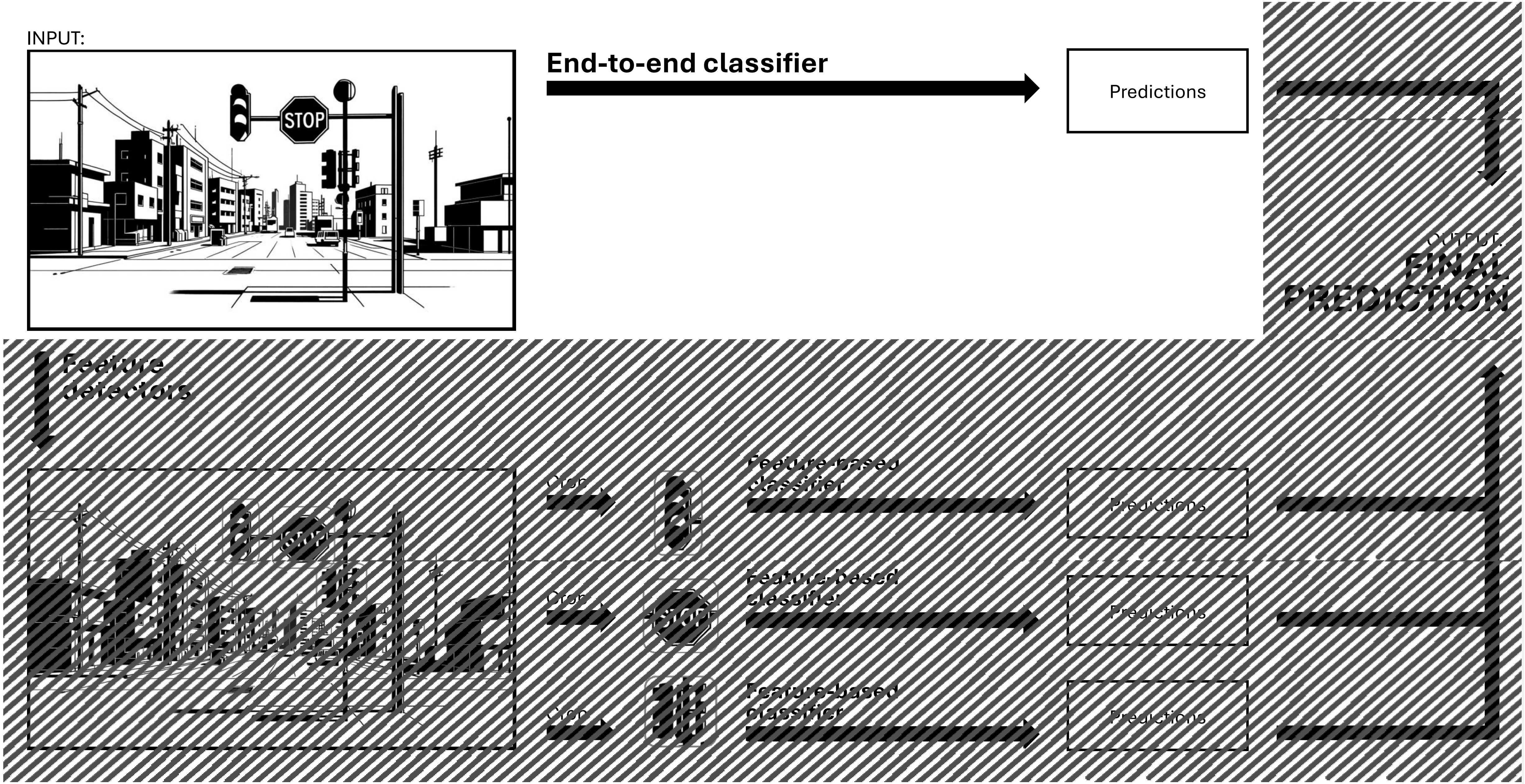
WORKFLOW (NO FEATURES DETECTED)

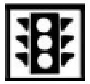
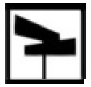


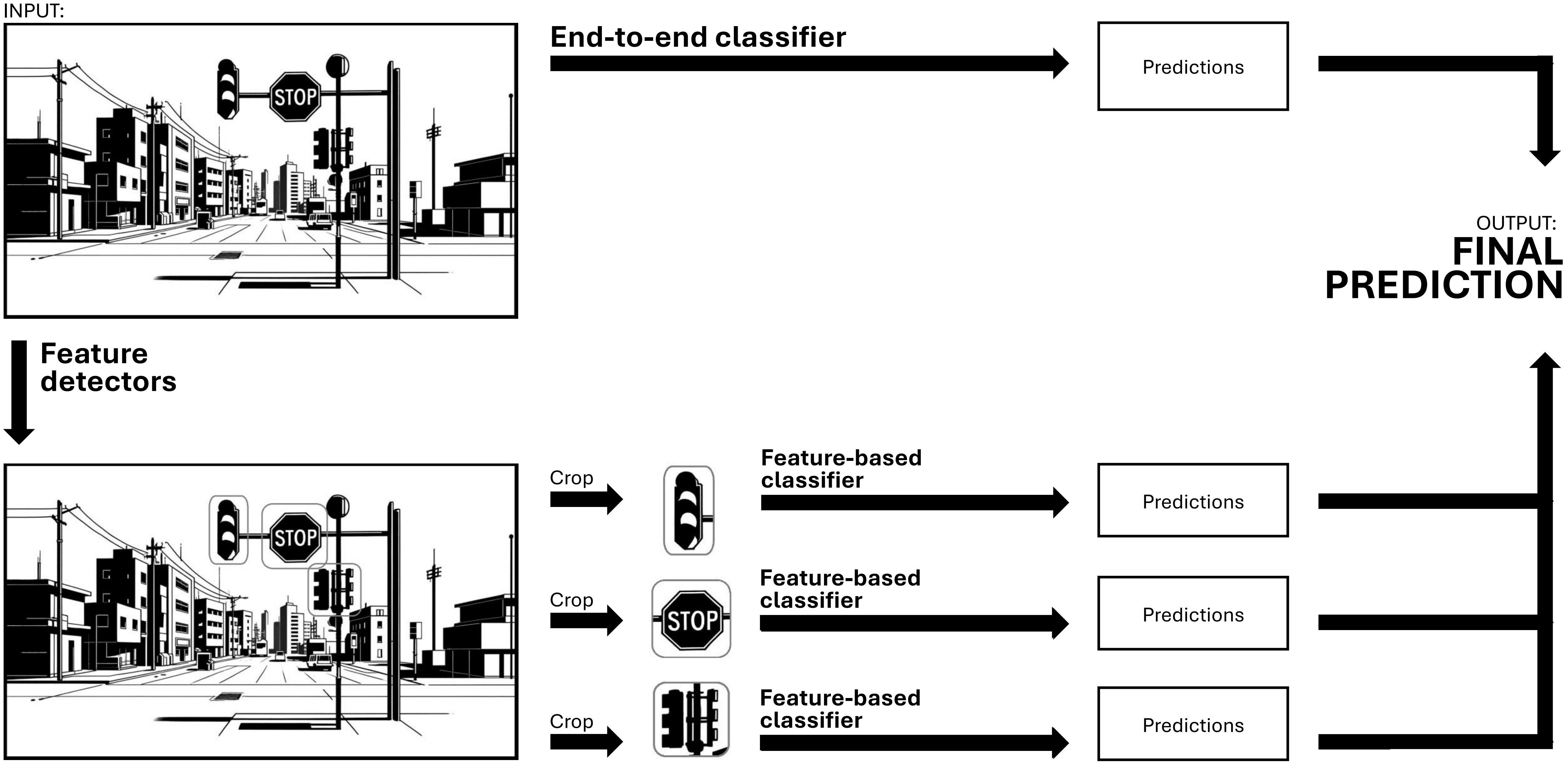
THE PIPELINE


INPUT:
End-to-end classifier
Predictions
STOP
Feature detectors
Crop
Feature-based classifier
Predictions
Crop
Feature-based classifier
Predictions
Crop
Feature-based classifier
Predictions
OUTPUT:
FINAL PREDICTION

END-TO-END CLASSIFIER

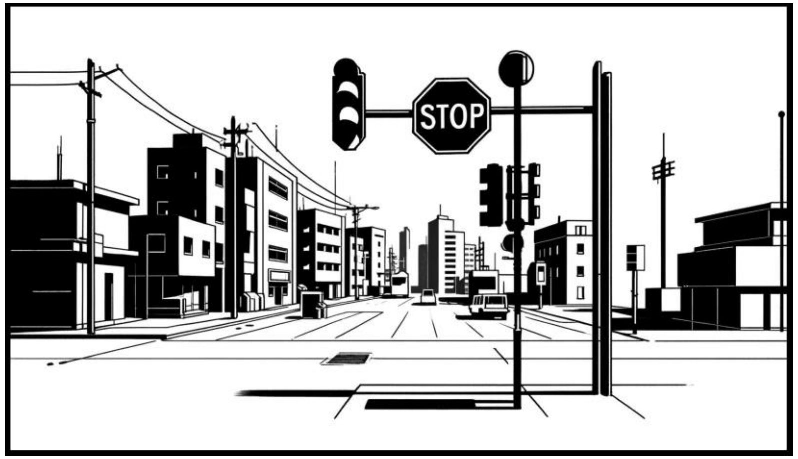


	BASELINES				PERFORMANCES (ACCURACY)		
Feature	**Baseline (top k)**	**Top 1**	**Top 2**	**Top 3**	**Top 1**	**Top 2**	**Top 3**
$\varnothing$	$\max_{i_1<\ldots<i_k}\left\{\sum_{r=1}^{k}\mathbb{P}\left(\mathrm{CITY}_{i_r}\right)\right\}$	0.094	0.185	0.271	0.634	0.789	0.865
	$\max_{i_1<\ldots<i_k}\left\{\sum_{r=1}^{k}\mathbb{P}\left(\mathrm{CITY}_{i_r}\mid\right.\right.$  $\left.\left.\right)\right\}$	0.169	0.333	0.484	0.595	0.762	0.866
	$\max_{i_1<\ldots<i_k}\left\{\sum_{r=1}^{k}\mathbb{P}\left(\mathrm{CITY}_{i_r}\mid\right.\right.$  $\left.\left.\right)\right\}$	0.263	0.391	0.495	0.536	0.643	0.821
	$\max_{i_1<\ldots<i_k}\left\{\sum_{r=1}^{k}\mathbb{P}\left(\mathrm{CITY}_{i_r}\mid\right.\right.$  $\left.\left.\right)\right\}$	0.325	0.486	0.601	0.750	0.851	0.895
	$\max_{i_1<\ldots<i_k}\left\{\sum_{r=1}^{k}\mathbb{P}\left(\mathrm{CITY}_{i_r}\mid\right.\right.$  $\left.\left.\right)\right\}$	0.128	0.253	0.360	0.636	0.802	0.877
	$\max_{i_1<\ldots<i_k}\left\{\sum_{r=1}^{k}\mathbb{P}\left(\mathrm{CITY}_{i_r}\mid\right.\right.$  $\left.\left.\right)\right\}$	0.349	0.425	0.494	0.632	0.743	0.827
	$\max_{i_1<\ldots<i_k}\left\{\sum_{r=1}^{k}\mathbb{P}\left(\mathrm{CITY}_{i_r}\mid\right.\right.$  $\left.\left.\right)\right\}$	0.142	0.241	0.321	0.630	0.783	0.865

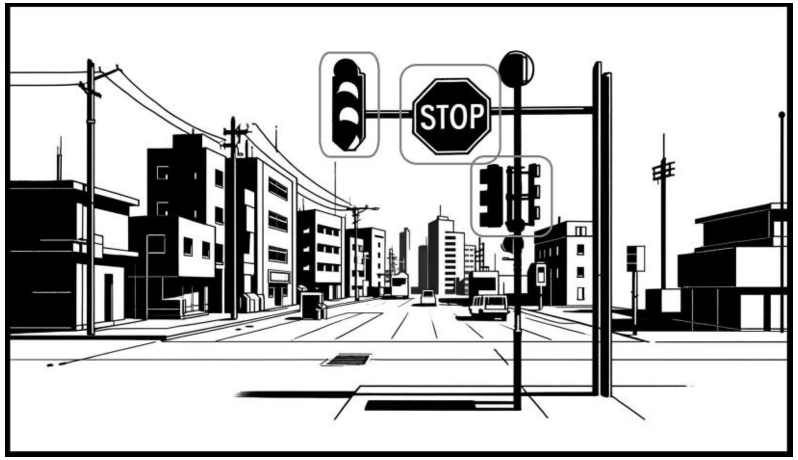

INPUT:
End-to-end classifier
Predictions
OUTPUT:
FINAL PREDICTION
Feature detectors
STOP
Crop
Feature-based classifier
Predictions
Crop
Feature-based classifier
Predictions
Crop
Feature-based classifier
Predictions

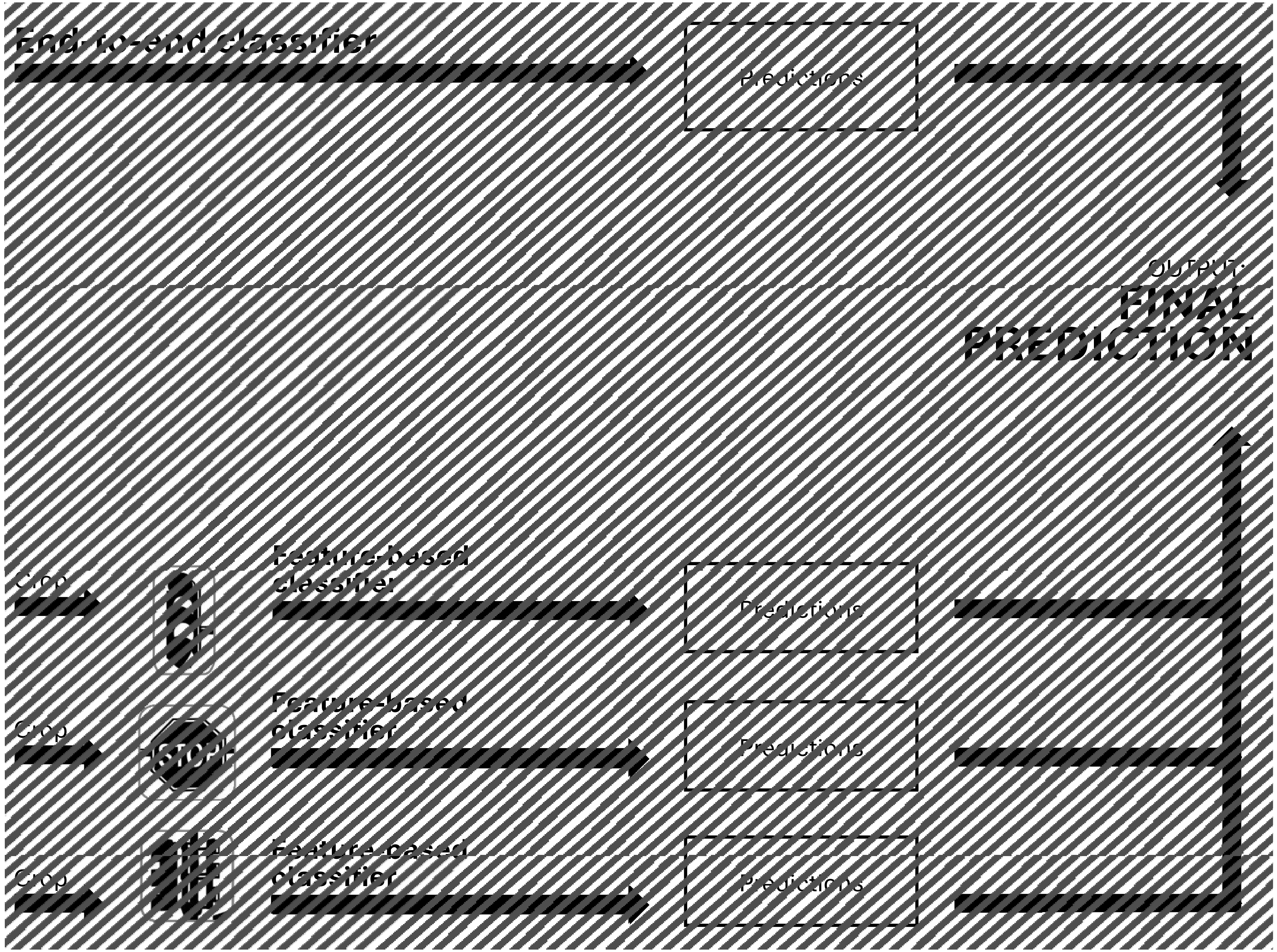
FEATURE DETECTORS

INPUT:



Feature detectors



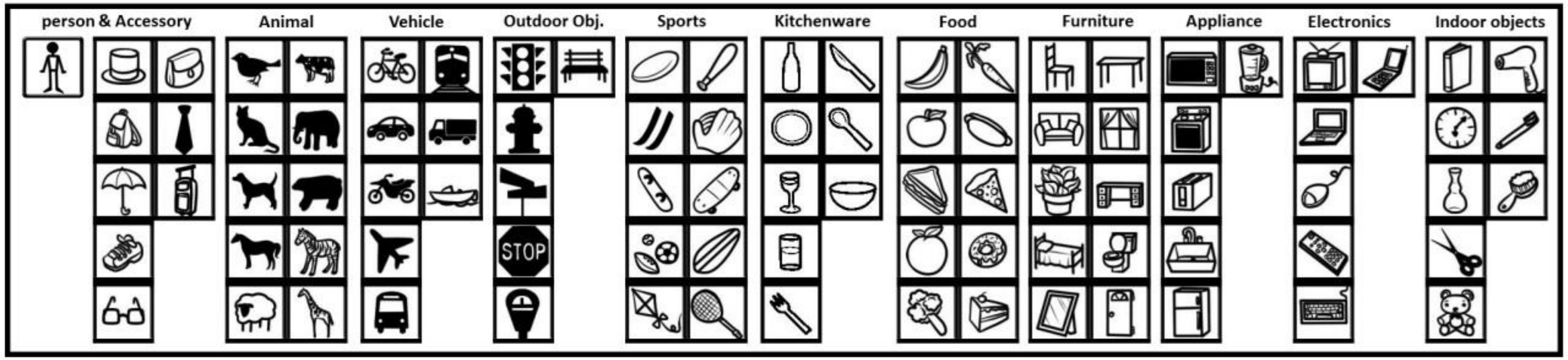


■ We used a neural network model (**ssd_mobilenet_v1_coco_11_06_2017**) pre-trained on

COCO

COMMON **O**BJECTS IN **CO**NTEXT

arxiv:1405.0312

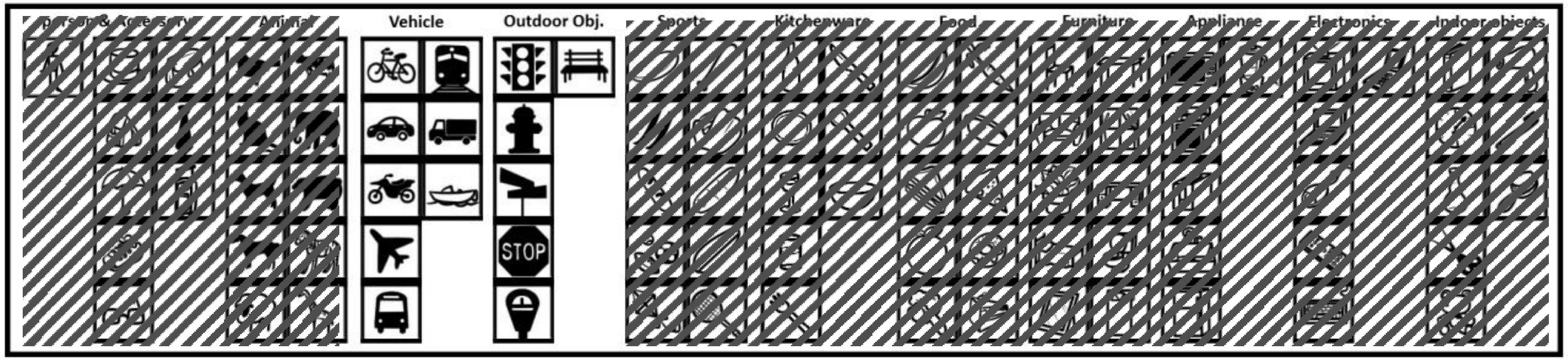


■ We used a neural network model (**ssd_mobilenet_v1_coco_11_06_2017**) pre-trained on

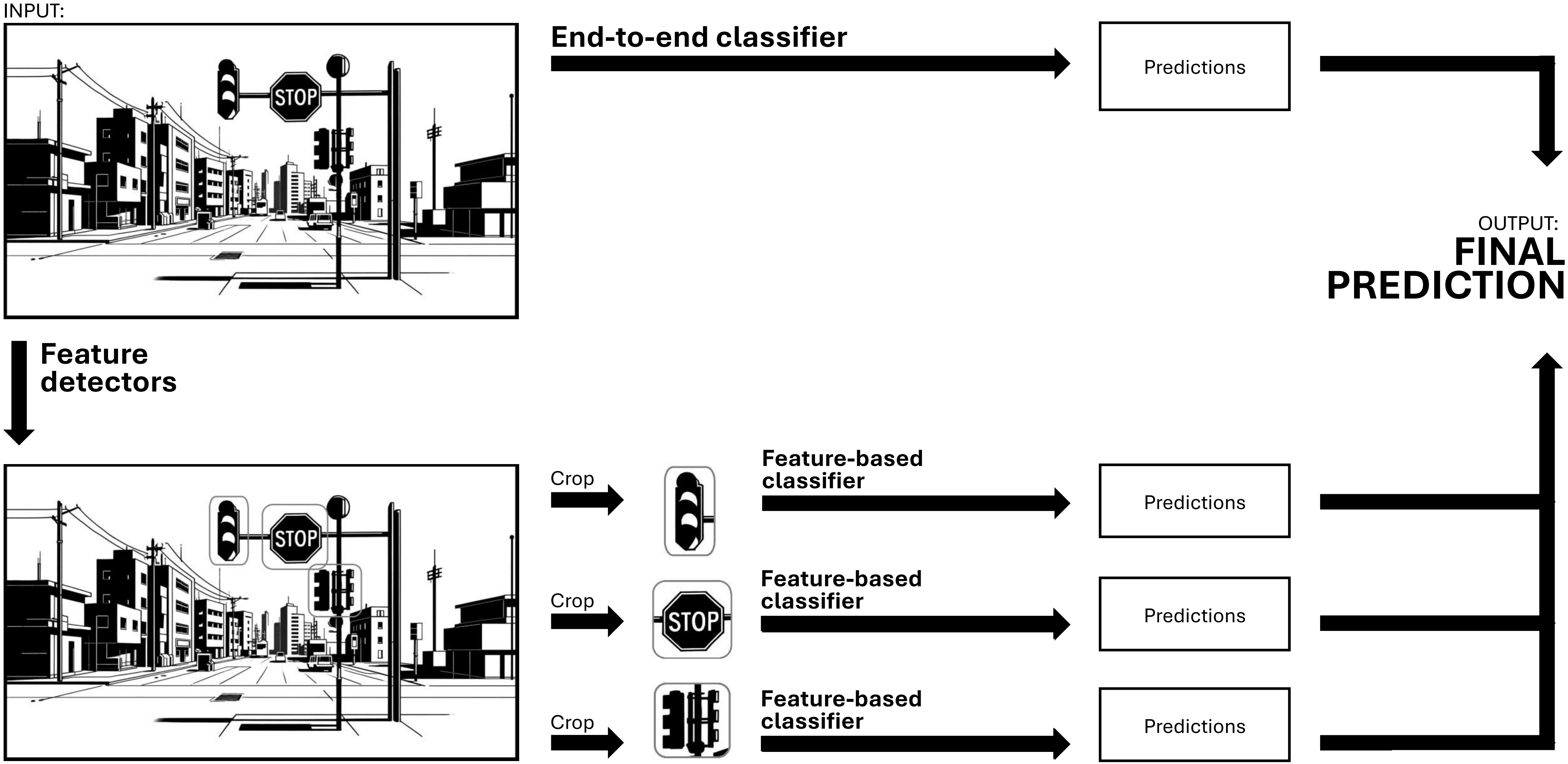
COCO

COMMON **O**BJECTS IN **CO**NTEXT

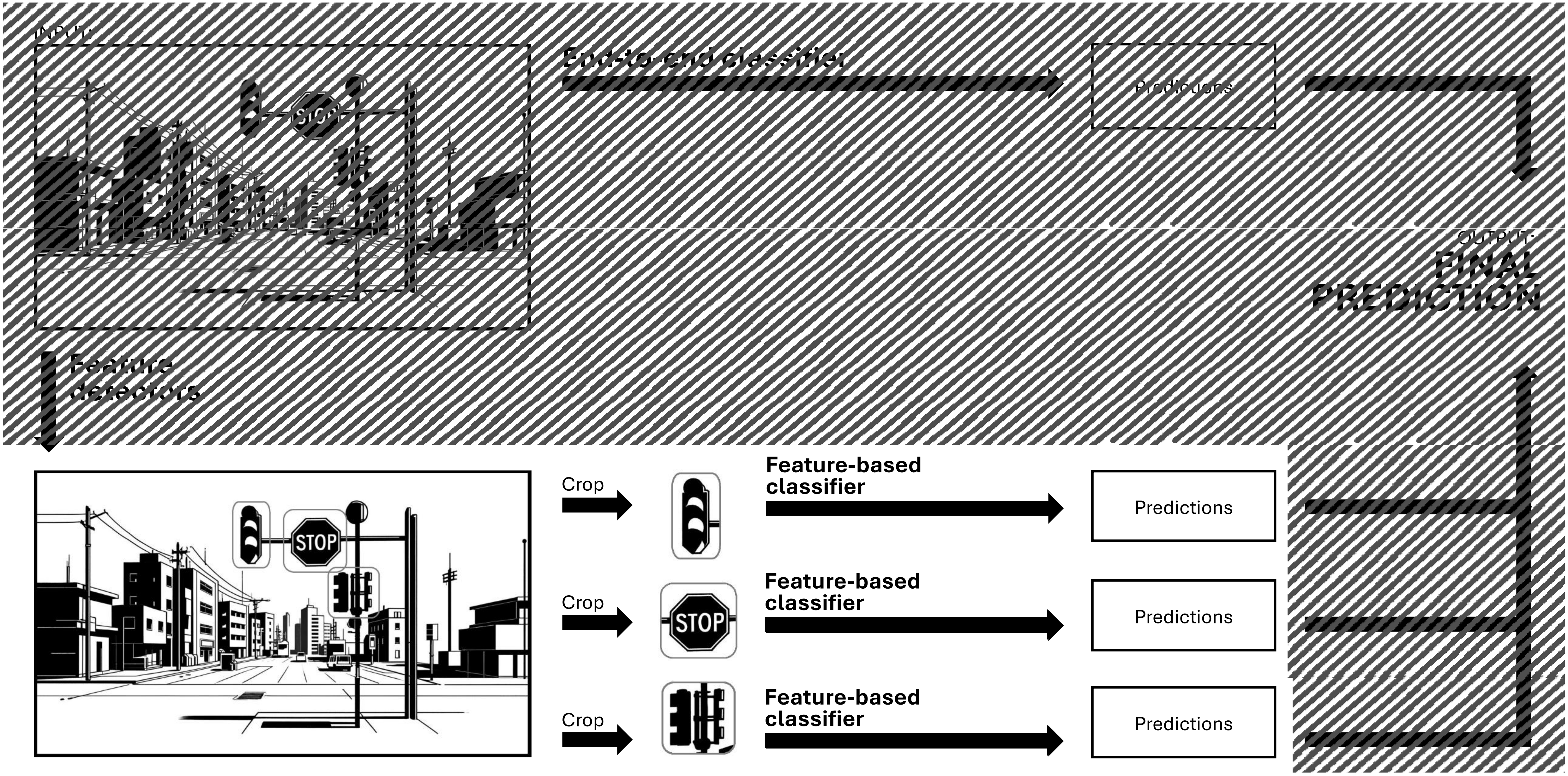
arxiv:1405.0312

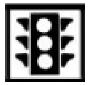
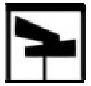


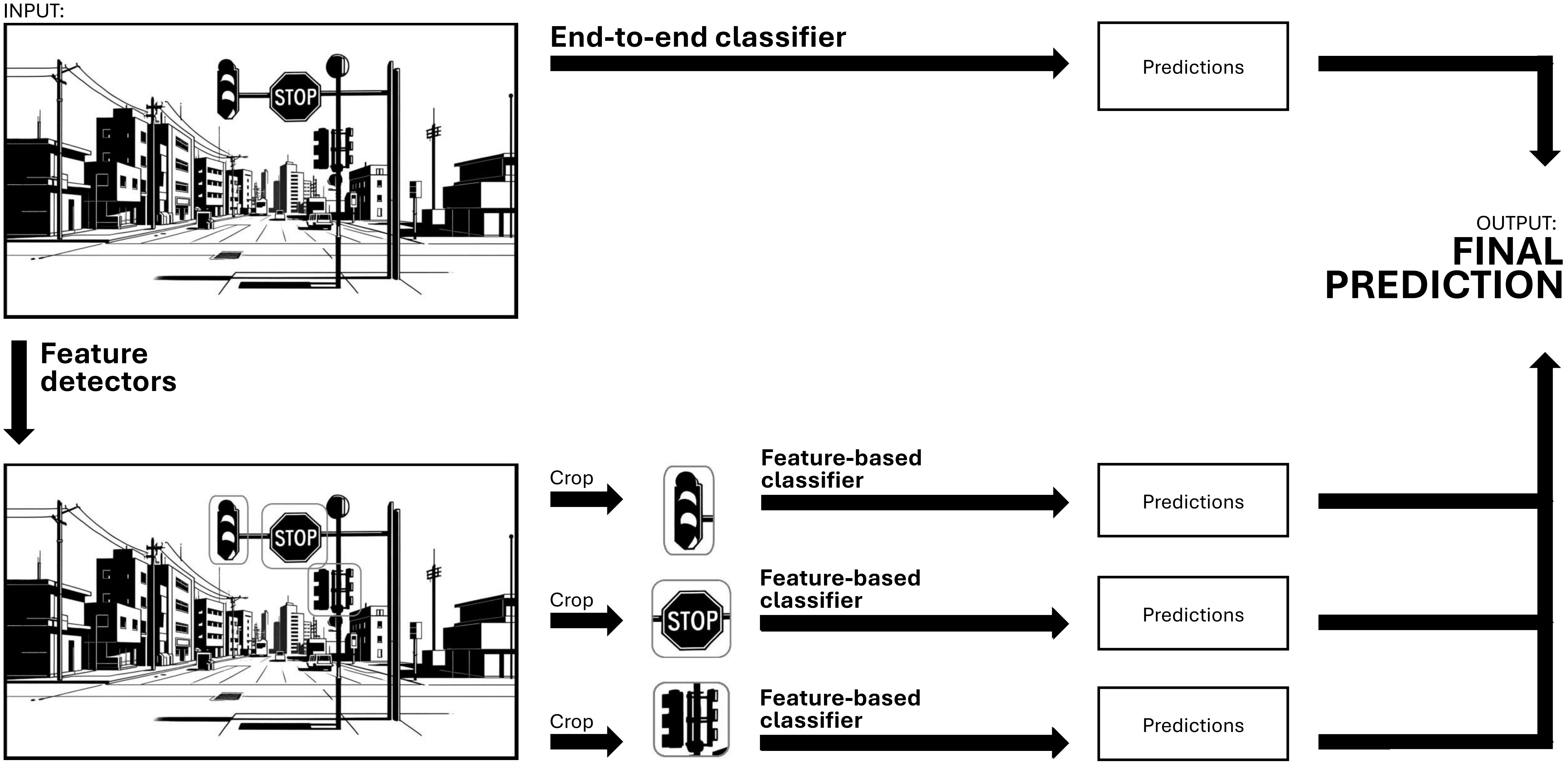
FEATURE	FREQUENCY	TOP 3 CITIES BY FREQUENCY		
	P( ) ≈ 0.01	(1) CHICAGO	(2) LONDON	(3) PHOENIX
STOP	P(STOP) ≈ 0.0025	(1) MIAMI	(2) CHICAGO	(3) BOSTON
	P( ) ≈ 0.005	(1) BANGKOK	(2) ROME	(3) LONDON
	P( ) ≈ 0.418*	(1) LONDON	(2) LISBON	(3) ROME
	P( ) ≈ 0.009	(1) LONDON	(2) ROME	(3) PRS


INPUT:
End-to-end classifier
Predictions
OUTPUT:
FINAL PREDICTION
Feature detectors
Crop
Feature-based classifier
Predictions
Crop
Feature-based classifier
Predictions
Crop
Feature-based classifier
Predictions

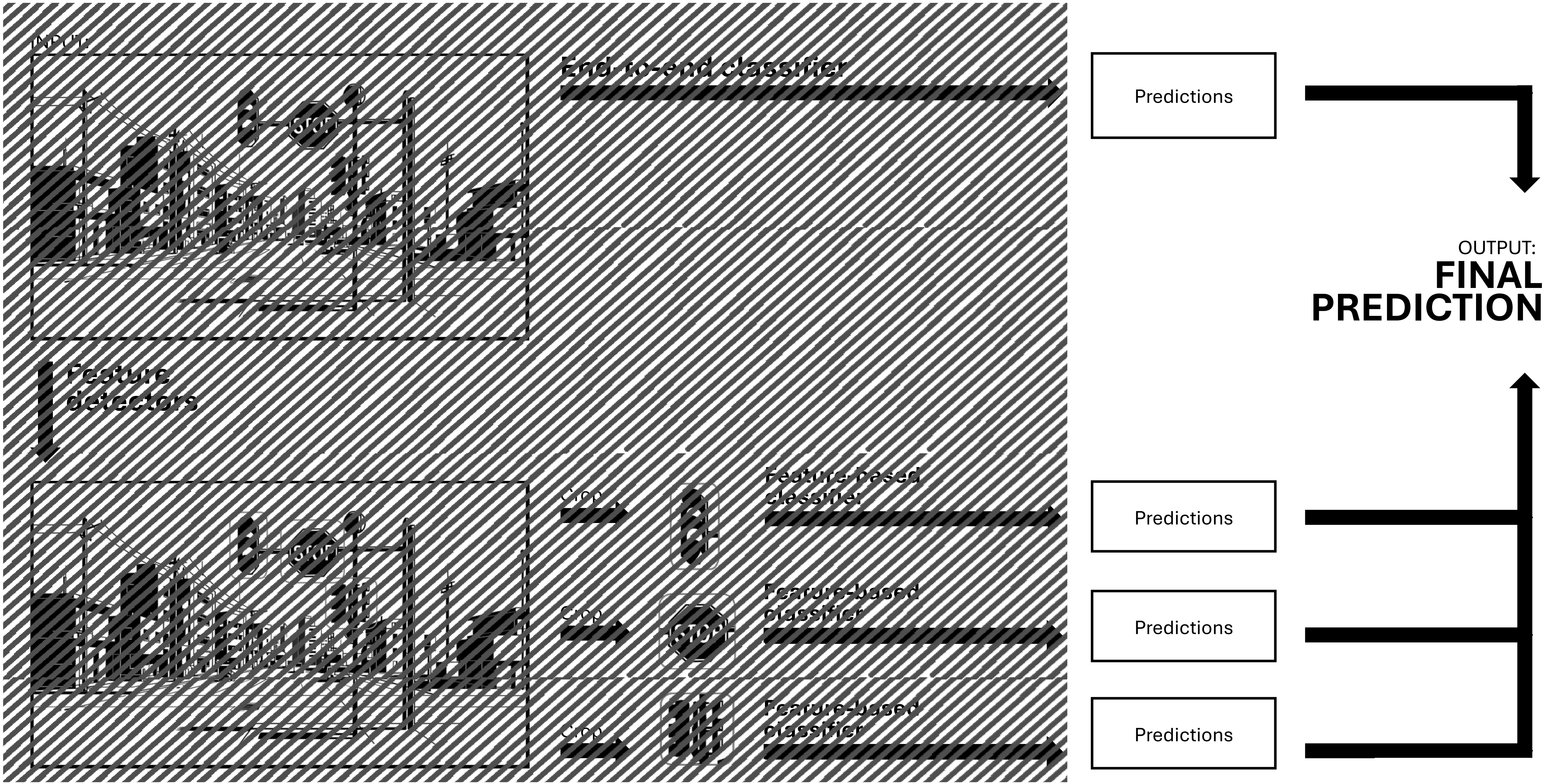
FEATURE-BASED CLASSIFIERS



	BASELINES				PERFORMANCES (ACCURACY)
Feature	**Baseline (top k)**	**Top 1**	**Top 2**	**Top 3**	**Top 1**
	$\max_{i_1<\ldots<i_k}\{\sum_{r=1}^{k}\mathbb{P}(\mathrm{CITY}_{i_r} \mid$  $)\}$	0.169	0.333	0.484	0.264
	$\max_{i_1<\ldots<i_k}\{\sum_{r=1}^{k}\mathbb{P}(\mathrm{CITY}_{i_r} \mid$  $)\}$	0.263	0.391	0.495	0.363
	$\max_{i_1<\ldots<i_k}\{\sum_{r=1}^{k}\mathbb{P}(\mathrm{CITY}_{i_r} \mid$  $)\}$	0.325	0.486	0.601	0.444
	$\max_{i_1<\ldots<i_k}\{\sum_{r=1}^{k}\mathbb{P}(\mathrm{CITY}_{i_r} \mid$  $)\}$	0.128	0.253	0.360	0.208
	$\max_{i_1<\ldots<i_k}\{\sum_{r=1}^{k}\mathbb{P}(\mathrm{CITY}_{i_r} \mid$  $)\}$	0.349	0.425	0.494	0.416
	$\max_{i_1<\ldots<i_k}\{\sum_{r=1}^{k}\mathbb{P}(\mathrm{CITY}_{i_r} \mid$  $)\}$	0.142	0.241	0.321	0.208


INPUT:
End-to-end classifier
Predictions
OUTPUT:
FINAL PREDICTION
Feature detectors
STOP
Crop
Feature-based classifier
Predictions
Crop
Feature-based classifier
Predictions
Crop
Feature-based classifier
Predictions

ENSEMBLE

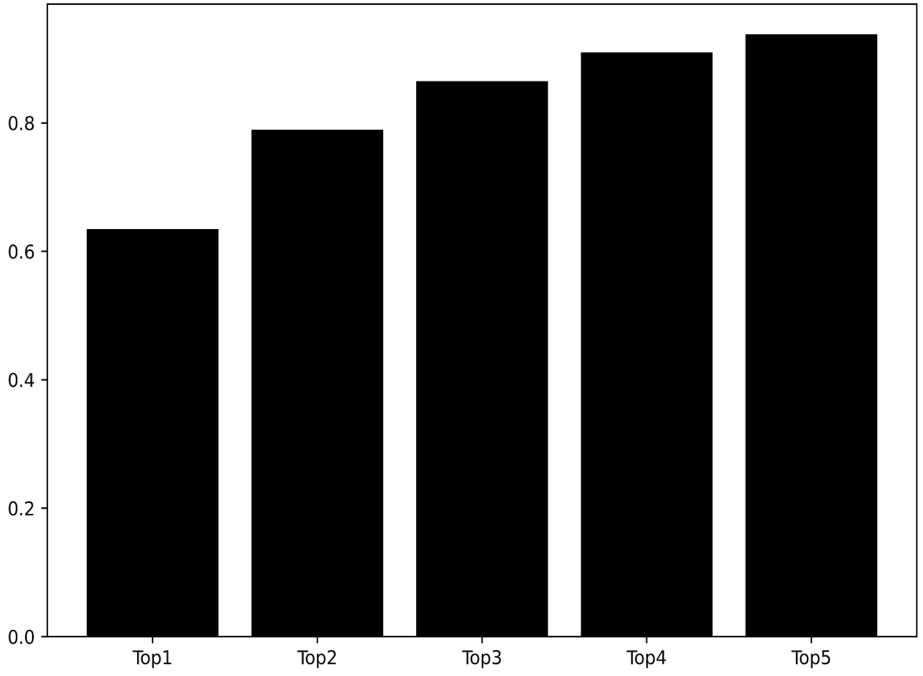

INPUT:
End-to-end classifier
Predictions
Feature detectors
Crop
Feature-based classifier
Predictions
Crop
Feature-based classifier
Predictions
Crop
Feature-based classifier
Predictions
OUTPUT:
FINAL PREDICTION

RESULTS

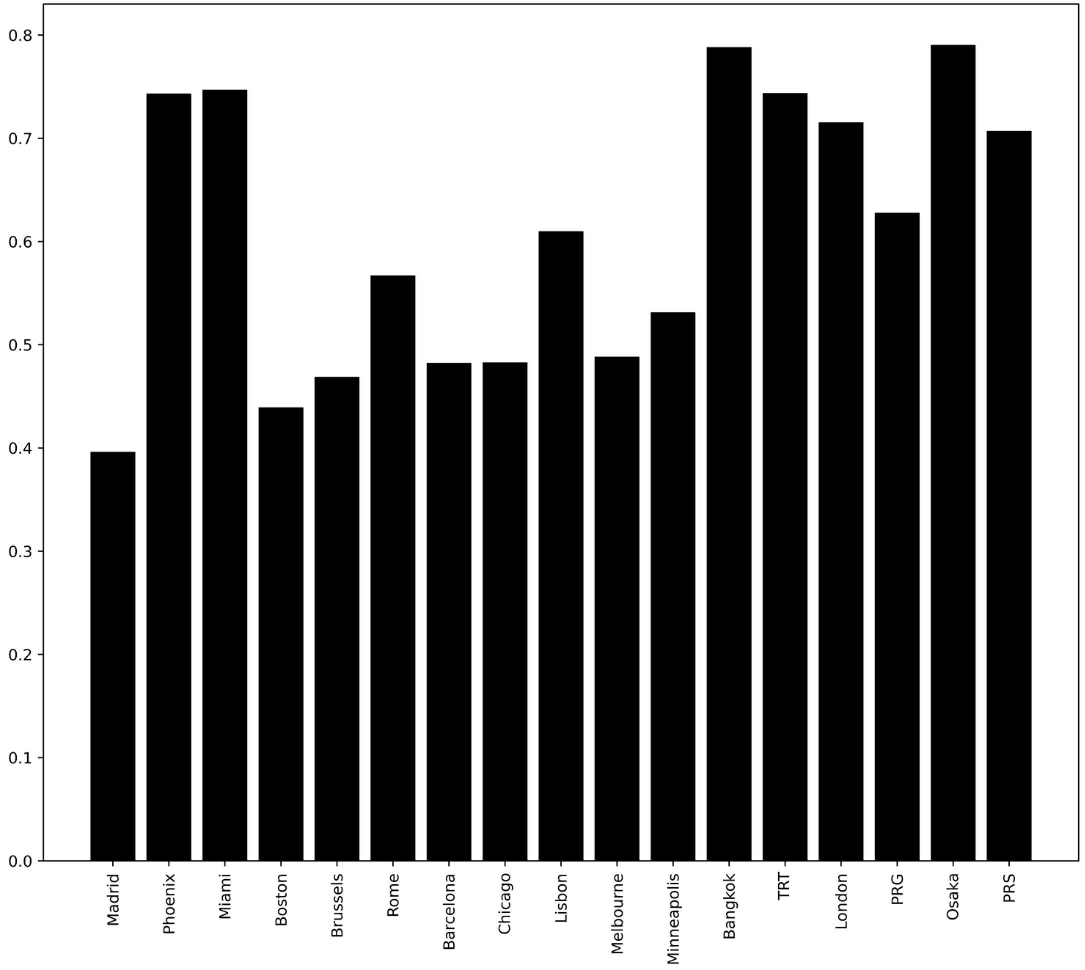
PERFORMANCE ANALYSIS

- Performance results of the complete pipeline according to various metrics

ACCURACY



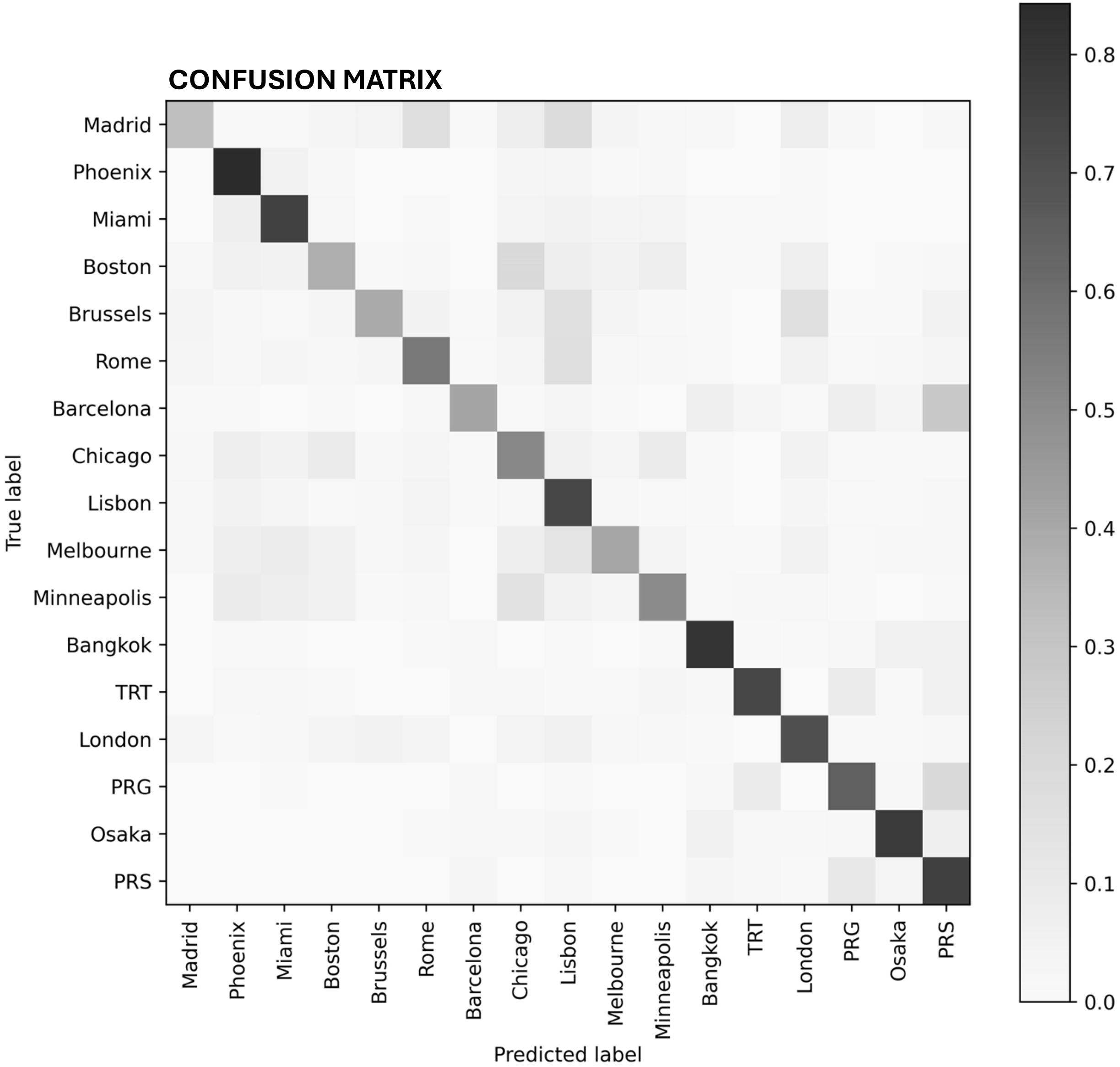
F1 SCORES



FINAL ACCURACY: 0.635

(essentially the end-to-end model)

- The model seems to mix up cities from similar geographic areas, but is able to distinguish between different geographic regions fairly well!



FUTURE IMPROVEMENTS

- Improve the feature-based classifiers by getting more quality data for the training, so to be able to also explore more complex models.
- Add more features (a starting point could be to add all the “COCO outdoors objects” features).
- Include rural areas and use texture-based features (such as GCLM).
- Improve the end-to-end model by experimenting with other architectures.
- Optimise the final model’s ensemble weighting and explore other ways to aggregate and combine the predictions from the classifiers and the end-to-end model.

THANK YOU

DEMO AVAILABLE AT

https://github.com/hochfilzer/geo-locator