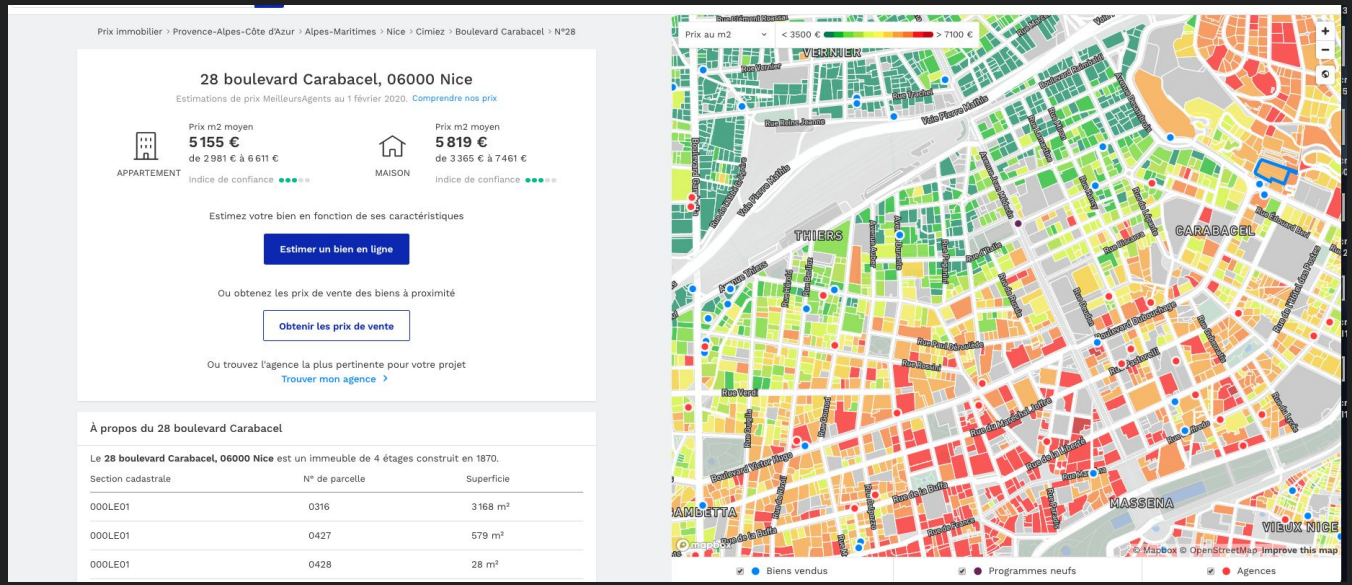
Detecting buildings types from raster images

Problem: building or house?

Motivations

- Meilleurs Agents parcel price maps (hybridation house price/apartment price)

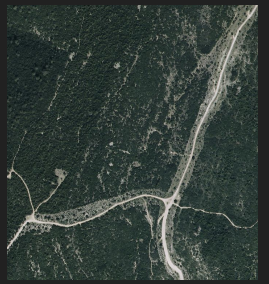


Training data: sources

1. geometries (50M, 27M > 50m², 32M many within a parcel)
    a. cadastre (parcels, building, blocks...)
2. building classes house/building (~5M??)
    a. meilleurs agents geolocated platform data
        i. listings
        ii. past sales
    b. siret, osm to exclude commercial buildings
3. local social statistics
    a. insee (iris/commune) => number of apartment/house dwelling units within iris
    b. carroyé (200m/1km)

Training data: images

- IGN data orthohr over 50 departements (aerial shoots) ~1T images, not idf
- Resolution 1 pixel = ~ 20/50 cm => image quality/scale differs a lot
- shoots can have varying color scale/distribution



Challenges

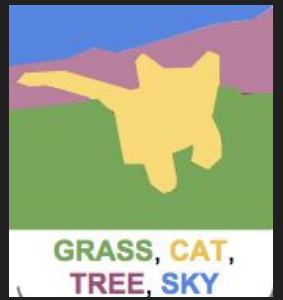
Classical ML vision problems

Classification/tagging

CAT

Classification + Localization

Instance segmentation



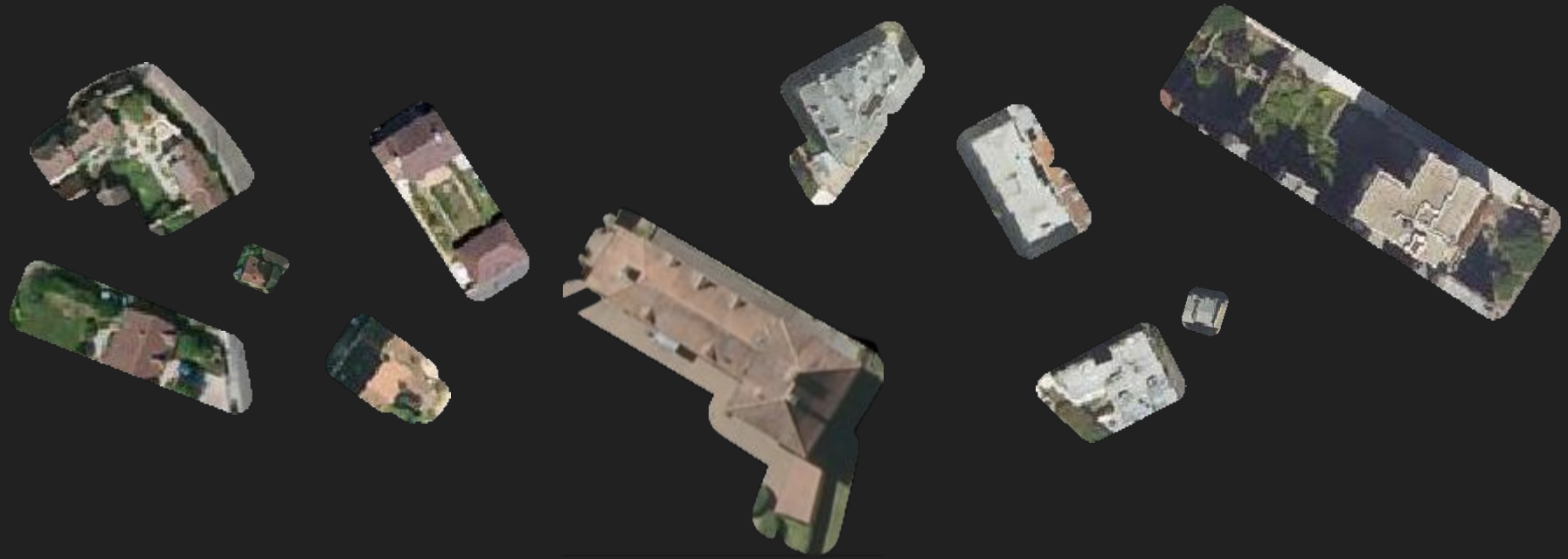
Learning datasets for localization and segmentation algorithms are harder to build since they need complete examples (negative classes)

Less treated problems

- Enrich classification models with non images features (geolocation…)
- Classify a specific region of interest within an image
- Influence of geographic location
- Low resolution images / raster images

Easy task?

- asymmetric classes (idf 90 % houses)
- distribution varying with location/density etc



Easy task? ambiguous cases

Issues with training data

- platform data quality and bias
- cases of multiple dwelling units within one house
- many buildings are small (23M < 50m2) and are likely to be just artefacts
- multiple buildings per parcel

Building/house modelisation

Image model

- images: cropped parcels (having at least one building > 50m2)
- image classification model:
  - tested SOA keras models resnet, inception, mobilenet, inception v4 chosen
  - params optimizer ADAM, batch size 32, nb epoch 10
- images rescaled to 200 * 200
- data normalization + augmentation (blur) to better generalize to full set of images
- training set balanced (70.000 house, 10.000 buildings)
- building an unbiased validation/test set
  - matching multiple data sources at same address
  - having full coverage over cities
  - manual validation
  - discarding ambiguous cases from validation

Meta-model

- 3 possible modelisations:
  a. features within dense layers of CNN
     - ++ a unique model that well tuned can outperform meta models
     - -- less control on features impact, much longer to learn and iterate
  b. meta model given image model predictions + normalized features
  c. independent model with non image features only + combine classifier (majority vote)
- models very sensitive to learning data bias and image model:
  a. generalization of image-model varies a lot with localisation, meaning meta-model sometimes should just stick to image-model predictions and sometimes discard them to make its own predictions
- multiple choices as meta - model classifiers: random forest, SVM...

Model stacking

parcel GIS geometry

Image classifier model (inception v4)

prediction
building: 0.1
house: 0.9

prediction
building: 0.1
house: 0.9

features

- geometric features
  a. area
  b. ratio area parcel/building
  c. convex hull...
- insee features
  a. population, density
  b. nb house
- including bloc information
  a. bloc predictions
  b. bloc mixity

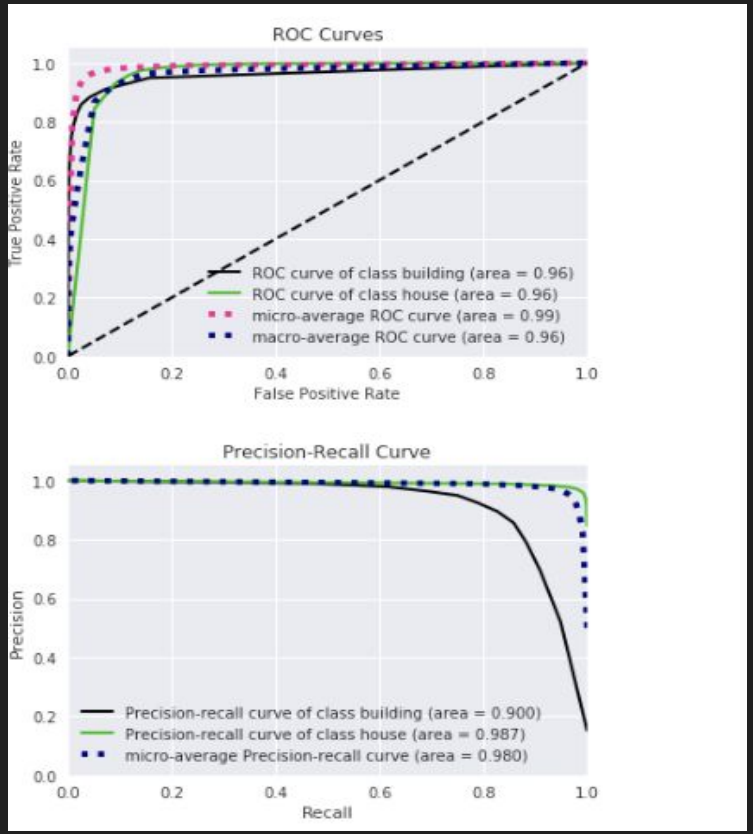
Meta learning (random forest)

prediction
building: 0.03
house: 0.97

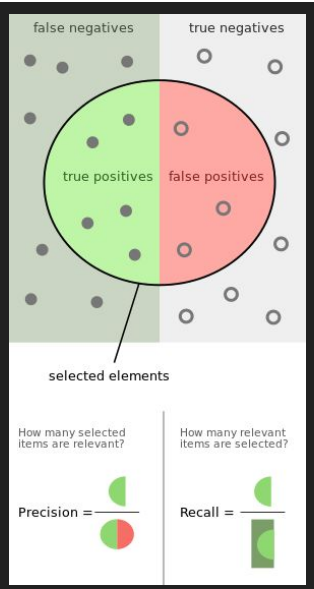
Meta model: features importance

```
[('image_model_prediction_house_score', 0.28901156059132416),
 ('image_model_prediction_building_score', 0.2395851146982416),
 ('area', 0.12493342189574272),
 ('INSEE.LOG.MAISON', 0.06413477733941705),
 ('ratio_parcel_building', 0.051377040956000486),
 ('INSEE.LOG.LOG', 0.033434756311099184),
 ('iris_area', 0.02908829982456862),
 ('parcel_area', 0.02753874034443895),
 ('INSEE.POP.DENS', 0.022069514434033013),
 ('INSEE.POP.POP65P', 0.019023065530480048),
 ('INSEE.REV.NBMEN', 0.0169781680347407),
 ('INSEE.POP.POP1824', 0.01666985494527238),
 ('ratio_bounds', 0.014030923564473973),
 ('ratio_convex_hull', 0.013363574015855825),
 ('parcel_ratio_convex_hull', 0.01321461425286768),
 ('ratio_height_width', 0.013130924535539278),
 ('parel_ratio_height_width', 0.012415648725904352)]
```

Meta model results



	precision	recall	f1-score	support
building	0.90	0.83	0.86	1354
house	0.97	0.98	0.98	7570
avg / total	0.96	0.96	0.96	8924

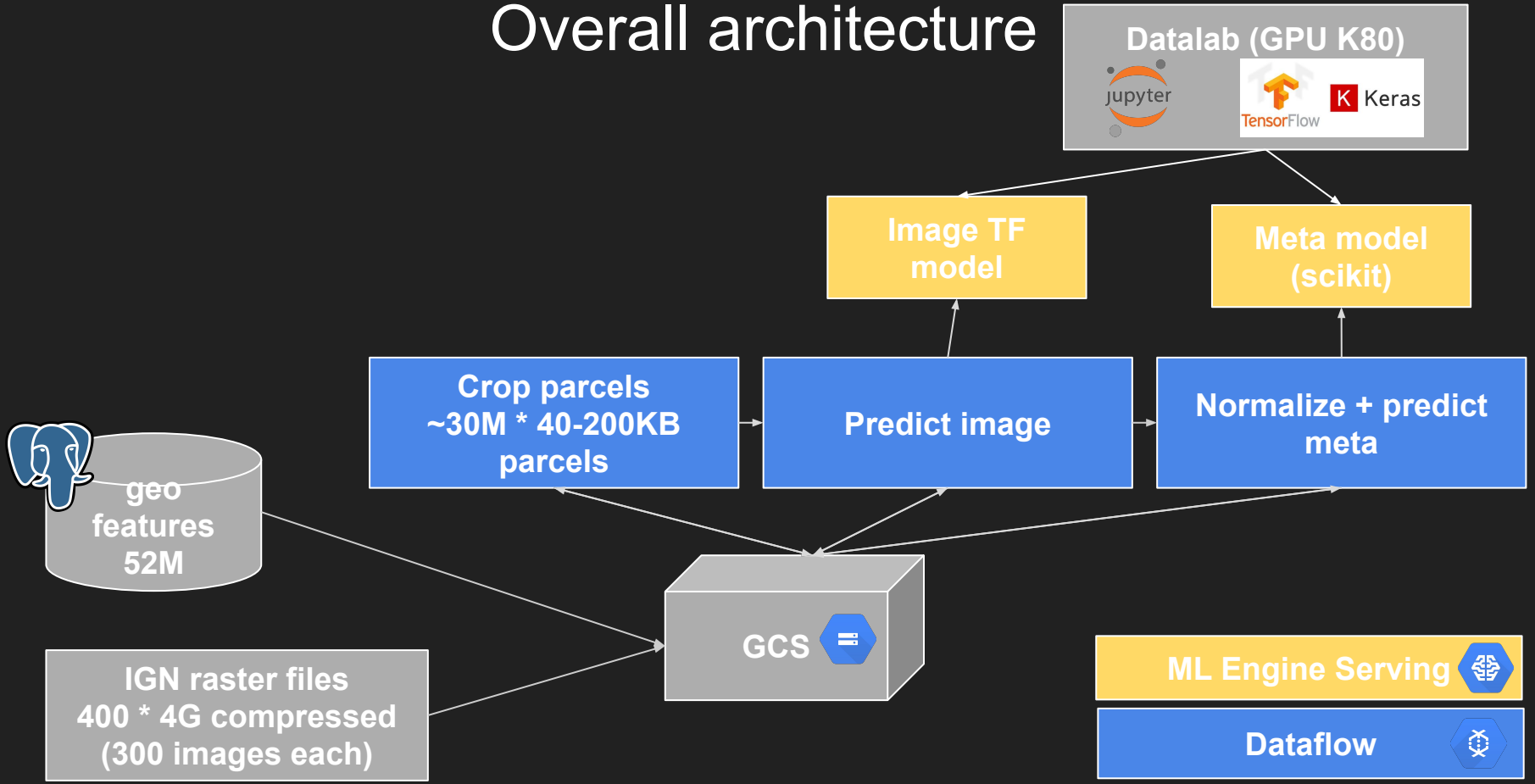


Room for improvement

- detecting region of interest classes within an image
- including locality in image model?
- inclusion of blocs in meta model
- more data for more generalization (DVF \o/)

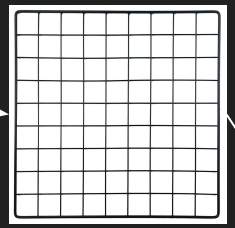
Architecture

Overall architecture



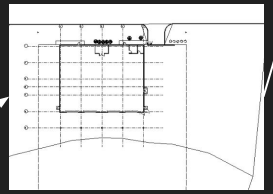
Focus building cropped parcels task

raster image coordinates grid



building geom

building coordinates



co group key image's buildings

for each image's buildings mask + crop

Geospatial transformations

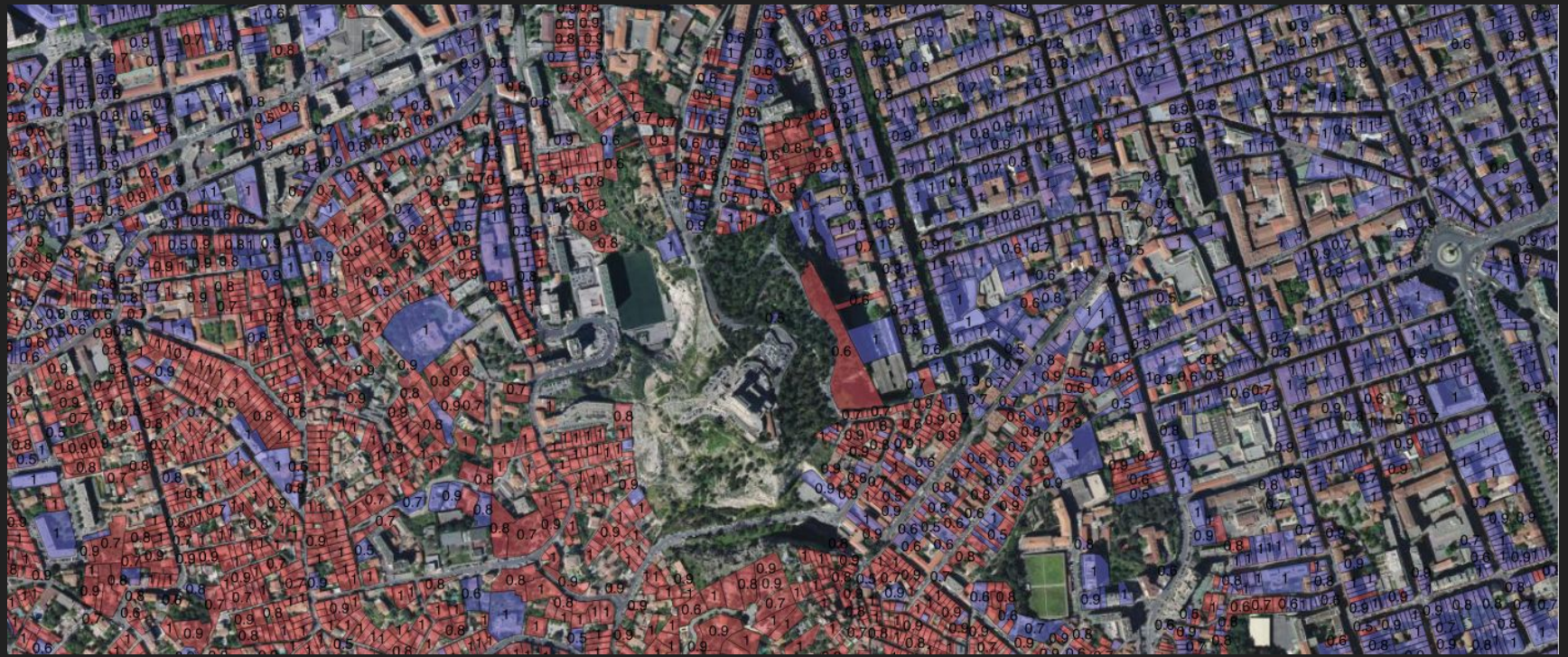
- postgis
- gdal/osgeo python bindings
- rasterio (transformations from raster images powered by gdal)
- shapely (geoms as python objects)
- pyproj (coordinates transform)

Serving keras models with ML engine

https://medium.com/meilleursagents-engineering/scaling-up-running-keras-models-on-ml-engine-40ddab58871

Maps!

Marseille



Angers