Machine Learning For a House pricing Prediction Web Application

Alex Soudant

Ynov Ingésup M1,

20 Boulevard Général de Gaulle, 44200 Nantes

Correspondence: Alex Soudant. E-mail: alex.soudant@ynov.com

report on the 10th of February 2017

Our planned objectives for Week 2 were "collection of images to obtain the dataset by geographical localisation".

I chose to build the scraper on the explorimmo.com website that contains few javascript scripts in it which ease the navigation through the different pages and information that I need to collect in order to build the dataset wanted to train and test the neural network on.
I also chose to build a basic javascript scraper instead of using the scrapy library. This choice permits to have a faster scraper despite loosing the avantages of the functionalities from the python library. I developed the code necessary to perform the scraping by using functional programing instead of an object oriented programmation approach. This makes scraping even more efficient as it regroups all scraping steps into tables of actions to perform which then can get executed by batches. However, I lost some time with the downloading of the estate images because batch execution of many http requests and picture files saving into the hard disk in a synchronous way provokes the loss of images in the process and sometimes even swapping pictures between the different advert folders created on the hard disk.
This issue was resolved by using the 'request' library in javascript. However, I still see that some pictures are saved with a bad file extention for no explanable reason to me instead of the usual .png or .jpg extentions for most pictures. These pictures are still openable so solving this problem is not of high priority.
After having the scraper correctly scripted, I ran it through 4 cities in France which represents the following adverts quantities :

1. Nantes : 1271 ;
2. Angers : 694 ;
3. Rennes : 852 ;
4. Brest : 1178 ;

Each advert information is saved in a different json file format with images saved in the same folder so I can keep track of which pictures belongs to which estate information file.
While the scraper was running I started looking at the size of the pictures files as they differ from an advert to another. In OpenCv there is a function named cv.Resize that will allow me to convert all images to the same image size.
Another interogation concerns the fact that some images in the adverts are actually not pictures taken of the advertised estate but pictures of 3D modelisation from the inside of the estate. Therefore a solution to avoid filtering by hand each image and discard the unwanted pictures is to train a first node of the neural network to recognize a picture of

an estate from an unwanted picture and automatically remove them for the next steps of computer vision analysis.