Beyond the Google StreetView: learning predictors for architecture style, graffiti and vegetation

Anonymous ECCV submission

Paper ID ***

Abstract. Given a large database of geotagged imagery of a whole city the goal is to evaluate a distribution of different architecture styles across the city and to detect areas with high occurrence of graffiti. We also aim on detecting areas with open or close view or areas with dense or loose vegetation. We first download 180,000 panoramas of the city of Madrid from the Google street view, then we generate a random set of 7,000 perspective images and label them. We use the labeled images to train a set of linear SVM predictors for each class. Finally, we uniformly sample 120,000 random images across the city of Madrid covering roughly area of $32 \times 36km$ and generate a set of heatmaps showing response of the learned predictors for different classes. The contribution of the paper is two fold: (i) We propose a simple method for detection of architecture style, graffiti, vegetation and view. We show that response of the classifier is semantically correct. (ii) We have created a labeled set of images that is going to be publicly available.

Keywords: We would like to encourage you to list your keywords within the abstract section

1 Introduction

It's difficult to make any one sweeping statement about Madrid architecture. As Spain's monarchical dynasties shifted from Flanders to Austria to France, so did the principal styles that shaped every period. Madrid was rarely a trendsetter; rather the city tended to absorb foreign influences and adapt them, more often than not, to a somewhat austere Catholic aesthetic.

2 Related work

3 Approach overview

Unlike [**?**] our goal is not to learn discriminative patches for different architecture styles and visual attributes. Instead we aim on learning our linear SVM on bag-of-words-like representation since we want to classify the whole appearance of

the image. Moreover, our goal is to learn a cheap and scalable pipeline that eventually allows to process millions of images.

In this section we first describe a collection of the data followed by description of our annotated database, then we describe how we learn a set of linear predictors using SVM and finally our qualitative evaluation of learned predictors.

3.1 Building a database

Similar to other works [**?**,**?**] we follow [**?**] and build our database from a collection of Google StreetView images. We have downloaded about 180,000 panoramic image of the city of Madrid. For a subset of panoramas we have generated four perspective images that capture both building facades and street level scene (details are given next in section 4). In the same manner we have generated a disjunctive set of images that are be subsequently annotated by humans.

3.2 Annotated database

We have designed a simple web-based annotation interface in order to collect image labels for a random set of Google streetview images described above. We have asked nine annotaters to label a given subset of images. Rather then classifying each of many Madrid architecture styles (Castilian baroque, Gothic, Romanesque, Neoclassical, Baroque, Bourbon rococo etc.) we asked annotaters to classify the architecture appearance in one of four classes: *Classical residential*, *Contemporary-residential* (modern buildings, second half of 20th century), *Non-residential* (office buildings, factories, shopping centers) and *Monumental*.

In addition we have asked them to also label amount of vegetation appearing in the image (*dense/loose* vegetation), a type of view (*close/open/partially-open* view) and presence of *graffiti*. A number of collected labels for each category is summarized in table 1. It is worth noting that non of the annotaters was an urbanist or an expert in architecture. Hence, our database is expected to contain some amount of human-induced noise which makes our task even more interesting.

3.3 Learning SVMs

Each image j is represented by its feature vector x_j. In this work we represent images by a pyramid-of-bag-of-words (PHOW) features (details are given next in section 4). For each class k we train a linear SVM by minimizing objective

$$||w_k||^2 + C_1 \sum_{x_j \in \mathcal{P}_k} h\left(w_k^T x_j + b_k\right) + C_2 \sum_{x_j \in \mathcal{N}_k} h\left(-w_k^T x_j - b_k\right), \qquad (1)$$

where $\mathcal{P}_k$ and $\mathcal{N}_k$ are positive and negative training sets for class k, h is the squared hinge loss and C_1, C_2 are penalty weights. While positive set $\mathcal{P}_k$ consist of all labeled images in class k the negative set N_k contains rest of the images within the same category. For instance, for class *Classical residential* form table 1 the negative set $\mathcal{N}_k$ contain images labeled as *Contemporary residential*,

Category	# of labeled images	E[AP] [%]
Architecture style:		
Classical residential	2702	65.0
Contemporary residential	1431	69.0
Contemporary non-residential	1032	39.0
Monumental	236	28.2
Vegetation:		
Dense	1219	81.1
Loose	2567	94.9
View:		
Open view	911	71.5
Close view	3767	94.4
Partially open	2509	71.7
Other:		
Graffiti	605	50.6
Not-graffiti		

Table 1. A number of labeled images in each category. The right column shows an expected average precision (AP) for each class within given category. The expected AP have been estimated by 6-fold cross-validation during training. The values indicate which class has a potential to be detected by learned predictor.

Contemporary non-residential and *Monumental*. Hence, we are training a set of one-versus-rest (OVR) predictors.

3.4 Evaluation

Rather then quantitative evaluation our goal is a qualitative analysis. Particularly, we aim on generating a set of heatmaps, one for each predictor, showing its response across the city on the map. These heatmaps can be interpreted as a spatial density of each class. What we expect to see is a semantics and complementarity of the maps. For instance, is the classical residential style located in the center, are there open views in the center, is moder style and office buildings located out of the city center, are there office buildings and at the same place as old castilian baroque houses, is there a lot of vegetation in the city center?

Indeed, we know the answers to the questions above because of a prior knowledge: The city center is typically the oldest part of the city, hence, we expect there is a majority of the classical residential style buildings. As the city expands a more modern buildings are being constructed at the peripheries as well as the office buildings and factories. There is probably not much open views in the center since it consists of lot of narrow streets and close spaces but in the suburbs we may expect a lot of open views since a density of buildings is significantly lower.

However, answering some questions may not be that straight forward. For instance, where is the majority of the graffiti in Madrid? Is it gonna be in the

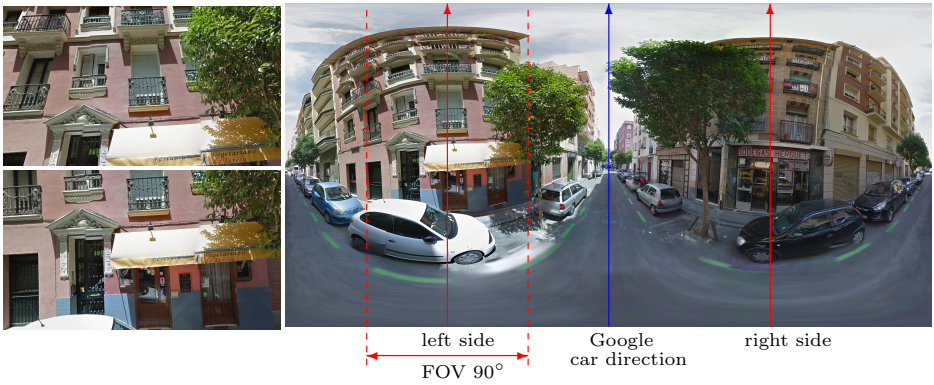


Fig. 1. A center of each panorama *(right)* corresponds to the Google car motion. For each side of the panorama we generate two perspective images *(left)* with horizontal field of view 90° in two different elevation pitches in order to capture both street view level and building facades.

center, in suburbs, somewhere else, or is it uniformly distributed across the city? Beside of the class density heatmaps we are also interested in visual appearance of the top ranked images in each class and see whether or not these are coherent with the classes. In section **??** we present a set of heatmaps along with several image examples for each class and we briefly analyze the qualitative results and its semantics.

3.5 Visualization

The idea is to visualize a response of each class on a map. Since each image in the database has it's associated GPS location we aim on utilizing the Google StreetVoew API [**?**] to plot a density of each class over a map of the city. To do so, for each image j in turn we compute its thresholded SVM score

$$s_{jk} = \max\left(0,\ w_k^T \cdot x_j + b_k - t_k\right) \tag{2}$$

where t_k is a threshold for given class k, and than we use this score as a voting weight in a accumulation space. Hence, the score is either zero or a positive distance from the threshold t_k. The accumulation space for plotting a heatmap is implemented in the Java Script API in `google.maps.visualization.HeatmapLayer` object [**?**]. Details will be given later in the section 4.

4 Experimental details

Database: We have downloaded set of 180,000 panoramic images [**?**] roughly covering the area of $30km \times 20km$. We then split this area by a regular grid with spacing of $1km$, hence we have spit the area by 600 uniform cells. Then we have randomly selected a set of 30,000 panoramas in such a way that for each in turn

we (i) have randomly picked a grid cell and (ii) from this cell we have randomly picked the panorama. In this manner we have achieved almost uniform sampling across the whole area.

Perspective images: For each panorama in turn we generate four perspective images. As shown in figure 1 the center of each panorama corresponds to the Google car direction of motion. Since our goal is to capture building facades we generate views to the right and left w.r.t. the car motion. In order to capture both street-level view and building facades, for each side we generate perspective images with two different elevation pitches, namely 4° and 28°. All the images have resolution of $960x768$ pixels and horizontal field of view 90°. Finally, our database consists of 120,000 perspective images.

Features: We first extract dense SIFT features with a step of 16 pixels. Then we represent each image by a pyramid-of-histogram-of-bag-of-words (PHOW) [] quantized into a $20k$ visual word dictionary. The dictionary has been learned by k-means from SIFT descriptors of $5,000$ randomly elected images. We have found that PHOW representation performs better then BOW. This can be explained by the fact that rather then representing a place-specific features of the particular image we aim on representing it's global appearance.

Learning SVMs: For each category we learn one-versus-rest linear SVM using [**?**]. To select optimal parameters C_1, C_2 we perform a grid search with $6-fold$ cross-validation and for each coupe of parameters we compute a expected average precision (see table 1). Then we pick a set of parameters that maximizes the expected AP and re-train the SVM with all available data. Table 1 shows a maximal expected AP reached during cross-validation. The values can indicate how reliable the classifier will be. For instance, the $E[AP]$ of the class *Monumental* is only 28%, considering that *Architecture style* category has only four classes, for a random performance would be 25%.Thus a predictor of this class will be very noisy.

5 Results and analysis

6 Qualitative analysis

7 Conclusion

(a) Classical residential

(b) Contemporary residential

Fig. 2. Architecture style: Heatmaps *(right)* showing a density of different architecture styles across the city of Madrid. Notice that while *classical residential* style (a) is mostly concentrated in the city center the *contemporary residential* style (b) is detected away from the city center. On the *left* there are examples of several top-ranked images for given style.

(a) Classical residential

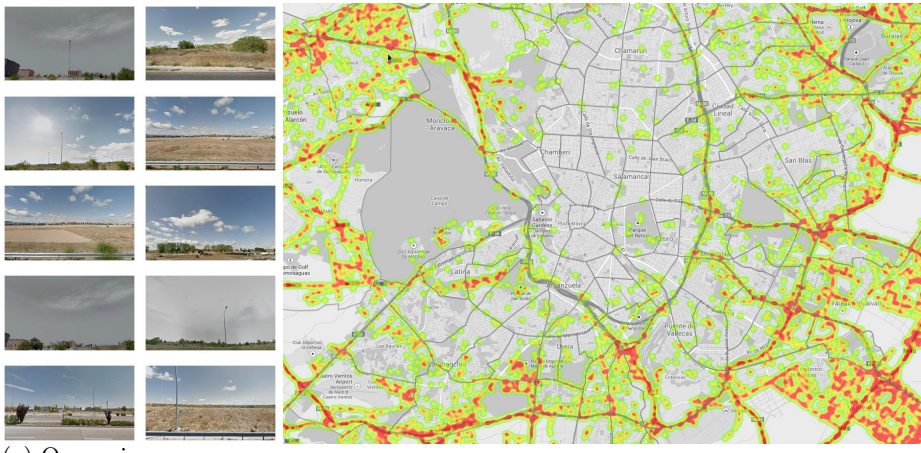
(b) Contemporary residential

Fig. 3. Architecture style: Heatmaps *(right)* showing a density of different architecture styles across the city of Madrid. Notice that while *classical residential* style (a) is mostly concentrated in the city center the *contemporary residential* style (b) is detected away from the city center. On the *left* there are examples of several top-ranked images for given style.

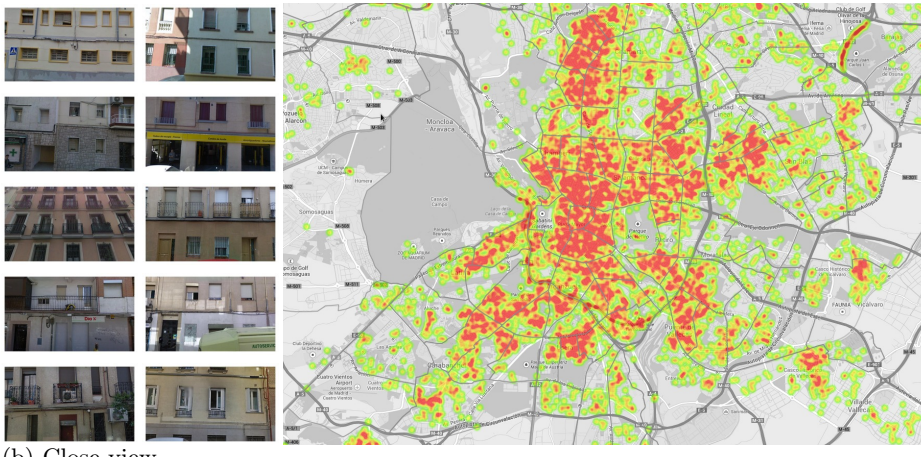
(a) Dense vegetation

(b) Loose vegetation

Fig. 4. Vegetation: Heatmaps *(right)* showing a density of vegetation across the city of Madrid. Notice a complemetarity of the heatmaps. The *column* shows several top-ranked images by learned predictor.



(a) Open view



(b) Close view

Fig. 5. View: Heatmaps *(right)* showing a density of ..... TODO

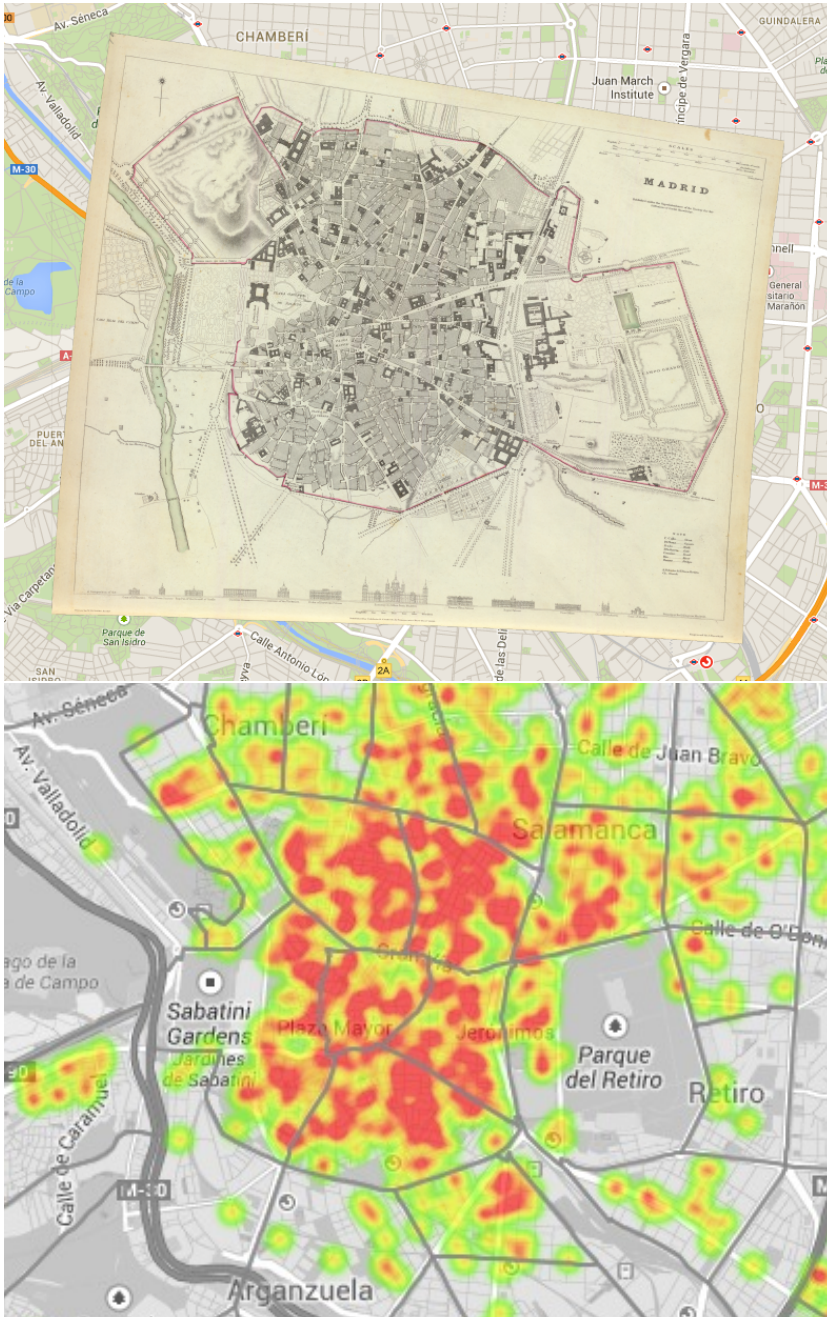


Fig. 6. An ancient map of the city of Madrid from 1834 aligned with a Google map. The map is overlayed with our heatmap showing a response of the predictor for *Classical residential* class. Notice how correlated is ..... lorem ipsum matriculum ipsum.