An Analysis of Human-centered Geolocation

Kaili Wang, Yu-Hui Huang, Luc Van Gool, José Oramas M., Tinne Tuytelaars
KU Leuven, ESAT-PSI, IMEC

ABSTRACT
Online social networks contain a constantly increasing amount of images - most of them focusing on people. Due to cultural and climate factors, fashion trends and physical appearance of individuals differ from city to city. In this paper we investigate to what extent such cues can be exploited in order to infer the geographic location, i.e. the city, where a picture was taken. We conduct a user study, as well as an evaluation of automatic methods based on convolutional neural networks. Experiments on the Fashion 144k and a Pinterest-based dataset show that the automatic methods succeed at this task to a reasonable extent. As a matter of fact, our empirical results suggest that automatic methods can surpass human performance by a large margin. Further inspection of the trained models shows that human-centered characteristics, like clothing style, physical features, and accessories, are informative for the task at hand. Moreover, it reveals that also contextual features, e.g. wall type, natural environment, etc., are taken into account by the automatic methods.

KEYWORDS
Geolocation, Recognition, Classification, Neural Networks

1 INTRODUCTION
The increasing amount of low-cost camera-capable devices released on the market and the popularity of online social networks have produced an almost exponential increase in the amount of visual data uploaded to the Web. A large subset of this data consists of "human-centered images", i.e. images whose content is mostly focused on a single individual. A side effect of this human-centered characteristic is that the amount of background information is reduced, thus, limiting the possibilities of inferring the location where the image was taken in a direct manner, i.e. by recognizing the place. Two questions then arise: Is it still possible to geolocate the image?, and if so, what are the useful visual cues for this task?

Here we start from the hypothesis that cultural and climate factors have an influence on the fashion trends and physical appearance of individuals of different countries. For example, individuals from tropical locations are more likely to have a tanned skin color than those living in polar regions. As to their clothing, people near the poles are more likely to wear warm clothing than individuals living near the Equator. Likewise, people at seaside towns may dress differently than those in dense urban areas. We investigate to what extent such cues can be exploited to predict the geographic location, i.e. the city, where a picture was taken.

We formulate the geolocation problem as a classification task where the city names are the classes to be inferred from human-centered images. Firstly, we let people guess, providing an idea about human performance, and illustrating the difficulty of the task. Secondly, we conduct a series of experiments with automatic classification methods based on Convolutional Neural Networks (ConvNet). Finally, we analyse the visual features that the ConvNet-based methods take into account to make a decision (Fig. 1). Our automatic methods surpass human performance by a large margin for this task. Moreover, our analysis suggests that human-centered features are being used. In addition, despite their less dominant nature, contextual features, e.g. wall type, natural environment, etc., are taken into account by the automatic methods.

Being able to geolocate human-centered images is not just of academic interest. Knowing the relation between geographic location and clothing is of commercial importance as well. For example, online shops can leverage this type of information to provide geography-based recommendations. Likewise, multinational retailers can use it to decide which type of products to put on their shelves.

This paper is organized as follows: in Section 2 we position our work with respect to earlier work. Section 3 presents the methodology followed in our analysis. In Section 4, we conduct a series of experiments and discuss the observations and findings made throughout our evaluation. Finally, in Section 5, we draw conclusions from the analysis.

2 RELATED WORK
Here we present related works on photo geolocation, on fashion analysis, and on ConvNets understanding.
**Photo geolocation** Photo-based geolocation has been studied from different cues. Some authors focus on landmarks of cities [2, 18, 32], some on street view images [5, 11, 30], and some on arbitrary photos [9, 10, 27]. These contributions can be divided into two groups: a first coming from retrieval [2, 5, 9, 10], and a second from classification [27]. We treat our problem as a classification task and employ a ConvNet instead of handcrafted features. Differently from [27], we target human-centered photos.

**Fashion analysis** Simo-Serra et al. [21] exploit a large amount of data from chicthopia.com, a large social network for fashion style sharing. Based on the user posts and tags, they train a model to predict how fashionable a person looks from a photo and suggest a way to dress better. Similarly, Bossard et al. [3] train a random forest to classify the clothing style of people in natural images. Murillo et al. [17] and Wang et al. [26] predict a person’s occupation based on the clothing cues and contexts from a photo. Wang et al. apply hand-crafted features to represent human body parts. Using sparse coding [14, 29], they learn representative patterns for each occupation. Similar to us, they also consider foreground and background information for the classification.

**Understanding ConvNets** ConvNets have been shown to be powerful tools for feature representation. However, without further analysis they remain somewhat of a black box. Simonyan et al. [22] tackled this problem by finding the images that activate specific ConvNet nodes. Similarly, Zeiler et al. [31] proposed a DeconvNet-based network to visualize activations. Later, Springenberg et al. [23] introduced a new variant of DeconvNet-based approach called guided backpropagation for feature visualization. Compared with the DeconvNet, the guided backpropagation provides a sharper and cleaner visualization for higher layers of the network. In order to have a more clear view on designing and training networks, Agrawal et al. [1] investigated several aspects of ConvNet feature learning.

Visualization is an intuitive way to understand a ConvNet; however, when the networks grow large, this technique gets infeasible. To deal with that, Escorcia et al. [7] exploit mid-level representations to learn the relation between visual attributes and pre-trained ConvNets. They investigate the impact of attributes on object recognition with data mining techniques. In our work, we apply a similar method to determine the important filters for the classification, then visualize and analyze those filters.

### 3 METHODOLOGY

As said, we formulate the geolocation problem as a classification problem, with the goal of assigning to a given image $I$ a city class label, from a fixed set of cities $C$. Following the landmark work of [12], we address this problem through Convolutional Neural Networks (ConvNets). In the following sections we describe several ConvNet-based image classification schemes.

#### 3.1 Image Classification with ConvNets

A deep ConvNet is a powerful mechanism for learning representations. Standard ConvNet architectures are usually composed of a set of feed-forward operations, with convolutional layers followed by fully connected layers. The first convolutional layers capture some basic features like color, gradient strength, edge orientation, etc., while the fully connected layers extract more abstract, complex semantic features [19]. In addition, [19] indicates that ConvNet extracted features outperform hand-crafted ones.

We investigate three methods to geolocate our images automatically. To this end we evaluate three ConvNets variants [4, 12] as described below. We focus on the case when the available data is not sufficient to train a deep ConvNet from scratch.

3.1.1 **Pre-trained ConvNet + SVM (Pretrained+SVM).** This method is inspired by [6]. The main idea is that given an input image $I$ and a pre-trained ConvNet $f$, when $I$ is pushed through $f$, every layer of $f$ produces an activation response. The activations at a specific layer(s) are regarded as the features $x$ of an input image, $x = f(I)$. Then, having a set of image - label pairs $(I_i, y_i)$, we extract the features $x_i$ from each image. Using the feature - label pairs $(x_i, y_i)$, we train a multiclass classifier $g$, i.e. a Multiclass Support Vector Machine (SVM), used at test time to predict the class label $\hat{y}_i$.

Following this methodology, the pre-trained network $f$ becomes a feature extraction mechanism, on top of which a SVM classifier is used to assign city labels to human-centered images. Hence, $\hat{y}_i = g(x_i) = g(f(I_i))$.

3.1.2 **Fine-tuning a pre-trained ConvNet (Finetuned).** A deep neural network has millions of parameters to tune, which means that it will need a huge dataset in order to set these parameters properly. For instance, for “VGG-F” [4] 138GB of image data were used for training, 6.3GB of image data for validation and 13GB of image data for testing.

Fine-tuning is an alternative for situations when a large dataset is not available to train a network from scratch. Moreover, it has already been proved to yield a better performance than when training a network from scratch with insufficient data[13]. It follows the same architecture as the pre-trained model $f$, but changes the last layers to satisfy the new classification task, thus producing a new network $f'$. Different from the previous method, given an image $I_i$, we focus on the output of $f'$ (over the set of classes of interest) as the predicted class label $\hat{y}_i = f'(I_i)$.

3.1.3 **Fine-tuned ConvNet + SVM (Finetuned+SVM).** This method is a combination of the previous two methods. It follows a similar procedure as Pretrained+SVM, i.e. the class labels are predicted as $\hat{y}_i = g(x_i) = g(f(I_i))$. However, different from Pretrained+SVM, here we replace $f$ by its fine-tunned counterpart $f'$ (Finetuned) and use activation responses from $f'$ as features. Hence, $\hat{y}_i = g(x_i) = g(f'(I_i))$.

#### 3.2 Identification of Relevant Features within the Network

One of the important strengths of deep models is their ability to learn abstract concepts from simpler ones. That is, when an example is pushed into the model, a conclusion concerning a specific task can be reached as a function of the results (activations) of intermediate operations at different levels (layers) of the model. In a classification task, these intermediate results may hint at the “semantic” concepts that the model is taking into account when making a decision. This will allow us to reach some understanding of what the model is actually looking at when classifying our human-centered images. To this end, we follow a procedure similar to [7], aiming to reconstruct...
With these terms in place, we resort to solving the equation: with guided backpropagation \cite{23} from \cite{8} to visualize the important features as defined by the layer-filter pairs \((p, q)\). Following the visualization method from \cite{8}, given a filter \(p\) from layer \(q\) and an input image, we first push forward the input image through the network, storing the activations from each filter at each layer, until reaching the layer \(p\). Then, we backpropagate the activations from filter \(q\) at layer \(p\) with inverse operations until reaching back to the input image space. Please refer to \cite{8, 23} for further details.

4 EVALUATION

In the following section we present the protocol followed to investigate the problem at hand. Then, we present two directions to address this problem, i.e. a user study (Section 4.1) and a series of experiments based on automatic methods (Section 4.2 & 4.3). We conclude this section with an inspection of the relevant features learned by the networks (Section 4.4).

Datasets: We use three different datasets in our experiments. The first one is the Fashion 144k dataset \cite{21} which contains 144,169 images posted from the largest fashion website chictopia.com. These images are "human-centered": a photograph of the person who posted it wearing an outfit, with an outdoor or indoor background. There are 3,443 different photographing locations but the number of photographs of each location varies considerably: some locations like Los Angeles have thousands of posts while most locations only contain less than 100 images. To keep the data balanced, we chose 12 different locations with more than 1,000 images. This produced a dataset composed by 12,448 images covering 12 city classes, i.e. 'LA', 'London', 'Paris', 'Madrid', 'Melbourne', 'Miami', 'Montreal', 'Moscow', 'North Europe region', 'NYC', 'Paris', 'San Francisco' and 'Vancouver'. The second and third datasets are quite similar. We collected them ourselves from chictopia.com and pinterest.com. Both contain the same 12 locations, with 13,332 and 12,671 images in total, respectively. In terms of photographing style, Pinterest images represent "instant" photos taken on the fly, while images from Chictopia are more planned, i.e. a user is posing (like a model) for the photo. We divide each dataset into three independent parts: 70% for training, 15% for validation and 15% for testing.

Performance metric: After training the classifiers, we evaluate their performance on the testing dataset. The mean accuracy for the 12 locations was calculated and we use this mean class accuracy (mCA) as the performance metric.
We start our evaluation by performing a study on how well people with an average of will also provide an indication for the difficulty of the geolocation.

Table 1: Quantitative results for user study. Mean class accuracy (mCA) on the Chictopia and Pinterest based datasets.

|                  | Chictopia | Pinterest | Mix |
|------------------|-----------|-----------|-----|
| Human            | 11.52     | 12.00     | 11.76 |

Figure 3: Cumulated score distribution over all participants.

4.1 Exp.1: Human Performance Study

We start our evaluation by performing a study on how well people perform the task of geolocation from human-centered images. It will also provide an indication for the difficulty of the geolocation task.

We conducted a survey asking people to determine where a given photo was taken. We randomly select 24 images from each city from our self-collected datasets (Chictopia and Pinterest), producing a total of 288 images for our online questionnaire. Each time, one image is presented to the participant and the participant is asked to select one city from the list of 12 possibilities. In total, we received 5,514 responses from 123 participants with ages between 20 - 30, with an average of ~20 votes given for every image. Among these responses, 2,763 are for the images from the Chictopia dataset while 2,751 are for the Pinterest dataset. We calculated the mean class accuracy (mCA) from those responses. The quantitative results can be found in Table 1. For reference, we present in Figure 3, the cumulative distribution of the accuracy obtained by the participants of the survey.

Discussion: From Table 1, we can see that the human performance on Pinterest dataset is a little bit higher than on Chictopia. The reason might be because the photos of Pinterest tend to offer a bit more background context. Users may gain more cues from the background to determine the geographic location. However, it is remarkable that for both datasets human performance is only 84%. Moreover, we can notice a very reduced number of participants reaching accuracy scores above 0.5.

4.2 Exp.2: Automatic Geolocation via ConvNets

In this section we evaluate the performance of the automatic methods presented in Section 3.1. We select “VGG-F” [4] as architecture for the implementation of our automatic methods. VGG-F is a feed-forward 21-layer ConvNet with 16 convolutional layers and 4 fully connected layers. For the case of Pretrained+SVM (Section 3.1.1), considering the activation from internal layers as a feature, we take the activations from the last fully connected layer (fc7) of a VGG-F network pretrained on ImageNet. This produces a feature vector with length 4,096 per image that we use to train a multiclass SVM. Cross-validation is adopted to get the best parameters for the SVM.

For the case of Finetuned (Section 3.1.2), the dimension of the output layer is modified to 12 instead of the original 1,000 in order to produce an output focused on our classes of interest. The weights of the last layer are initialized with random values from a Gaussian distribution. Additionally, two more dropout layers are added between fc6 and fc7 and between fc7 and fc8. We tested two fixed learning rates (1e-4 and 1e-5) and a range of adaptive rates (1e-8 to 1e-4) to find the best one. For the case of Finetuned+SVM (Section 3.1.3), we first fine-tune the network, as done for Finetuned, and use the fine-tuned network as feature extraction mechanism (Finetuned+SVM).

During training, for the case of our self-collected Chictopia and Pinterest datasets, we use the Mix dataset to train ConvNets/classifiers by adopting the above three methodologies and evaluate the ConvNets/classifiers on each of the subsets, i.e. Chictopia, Pinterest and the Mix dataset, respectively. For the case of the Fashion 144k dataset [21], we use the pre-defined image sets for training and testing. All our models are trained using the MatConvNet framework [25]. Table 2 shows the quantitative performances for this experiment.

Discussion: From Table 2, we can observe that the automatic methods perform best on the Chictopia dataset. Moreover, their performance is substantially higher than human performance, and this for all three datasets. On the Chictopia dataset, the automatic methods have a much higher mCA than human whose mCA is only ~12 percentage points (pp). Finetuned has the best performance among the three automatic methods with its highest mCA 40.75%. Finetuned+SVM follows with 35.5% mCA while Pretrained+SVM has the lowest mCA (33.97%). Therefore, the features extracted from the fine-tuned model yield a better performance than the pre-trained model when using the same classification technique. Overall these results show that predicting the geographic location where the analyzed “human-centered” photos are taken is - to some extent - possible.

4.3 Exp.3: Human-based Feature Pooling

In the previous experiment we followed an image-based feature pooling approach, i.e. features were extracted by considering the

|                  | Chictopia | Pinterest | Mix | Fashion 144k |
|------------------|-----------|-----------|-----|--------------|
| Pretrained+SVM   | 33.97     | 25.22     | 29.94 | 37.16        |
| Finetuned        | 40.75     | 28.13     | 34.75 | 39.15        |
| Finetuned+SVM    | 35.45     | 24.25     | 30.26 | 35.60        |
| Human            | 11.52     | 12.00     | 11.76 | -           |

Table 2: Mean class accuracy (mCA) of image-based feature pooling in percentage points.
Table 3: Mean class accuracy (mCA) of human-based feature pooling in percentage points

|                | Chictopia | Pinterest | Mix   | Fashion 144k |
|----------------|-----------|-----------|-------|--------------|
| Pretrained+SVM | 28.07     | 19.54     | 24.06 | 33.28        |
| Finetuned      | 35.00     | 22.00     | 28.79 | 35.20        |
| Finetuned+SVM  | 29.66     | 19.26     | 24.69 | 30.44        |
| Human          | 11.52     | 12.00     | 11.76 | —            |

Figure 4: Quantitative Performance Comparison.

Discussion: From Table 3, we again see that Finetuned performs best on the Fashion 144k (35.20%) and Chictopia (35%) datasets, but also that the accuracy drops about four percent compared to Exp.2. By comparing the quantitative performance between human-based feature pooling and image-based feature pooling (Section 4.3), we can see a general trend of performance decreases by four to six percent (Figure 4). This shows that the context information does play a role in determining the final decision, but that it is not critical since the performance after removing most background information is still two times better than that of humans.

4.4 Exp.4: Inspection of Relevant Features within the Network

In this section we take a look at the visual patterns that the automatic models take into account when predicting geographic locations. More specifically, we focus on the network from Finetuned trained on the Fashion 144k dataset based on image-based feature pooling (Section 4.2).

Figure 5: Mean area under the ROC curve (mAUC) for location class reconstruction on the Fashion 144k dataset. Lower \( \mu \) values indicate higher sparsity of identified relevant filters.

We identify the relevant features within the network by following the procedure presented in Section 3.2. We solve the \( \mu \)-Lasso problem from Eq. 1 with the SPArse Modeling Software (SPAMS) [16]. We verify the effect of the sparsity parameter \( \mu \) by performing this experiment with different values of \( \mu = \{10, 50, 100, 500, 1000, 5000, 10000, 15000\} \). In Figure 5, we can observe the mean area under the ROC curve of the classification task over the classes of interest as a function of the sparsity parameter \( \mu \) (Section 3.2).

In addition, we analyze the proportion of activations of the identified relevant features within the bounding boxes predicted by the Faster R-CNN detector [20]. Towards this objective, we compute the proportion between the feature response within the bounding box with respect to the response on the whole image. Then we estimate the probability density function per filter. In Figure 6, we show the probability density of this proportion color coded in jet scale. For clarity, the identified features are sorted in the order of decreasing proportion from left to right along the x-axis. In the y-axis we indicate the proportion of activations occurring within bounding box.

Finally, Figure 7 shows qualitative visualizations of the relevant features for \( \mu = 10 \). For clarity, we present the response of the filters with clearer "semantic" interpretation (added in green text) per city. For each filter, we show images, from left to right, sorted by decreasing magnitude of response on the test set.

Discussion: In Figure 5 we can observe that already with a small value of \( \mu = 10 \), i.e. with a small number of filters it is possible to achieve an initial mAUC of \( \sim 0.64 \). This suggests that there is a relatively low number of features that Finetuned exploits in order to make accurate predictions. We can notice that as the sparsity decreases, i.e. for higher values of \( \mu \), the mAUC tends to reach an asymptote.

To complement the observations made previously, in Figure 7, we show visualizations from relevant features estimated for \( \mu = 10 \). As we can see, some of these features are related to the city labels. For instance, among the 'Los Angeles' features, human limbs are
of importance. A similar trend is visible for ‘Miami’ where, in addition, features related to skin color are of importance. These two concepts, i.e. visible limbs and color skin, imply the fact that these two locations are relatively warmer. We can also observe some context-related features. For instance, for ‘Melbourne’, vegetation seems to be quite common. Consequently, it is highlighted by filters focused on the color green. For the ‘North Europe region’, the color white seems to be a strong feature, present as it is on walls and in natural landscapes (mostly due to snow). Similarly, for ‘Paris’ there seems to be a high occurrence of walls with beige color. Furthermore, for ‘Paris’ this color is also popular for clothing. Likewise, for ‘Moscow’ wearing blue colors seems to be common. This shows that indeed, there are some features on human-centered images that can be informative for geographic localization.

Finally, regarding the question of whether these relevant features lie on the background (context) or on the foreground (human-region), Figure 6 shows that these relevant features are found grouped in three clusters. On the top-left corner, we can note a group of activations for relevant filters that occur mostly in the foreground. We can notice group of features (filter id: 1-20) with more than 80% occurrence on the foreground. In the bottom-right, we can see a spread-out group (filter id: 70-108) these are activations of filters with middle to high relevance that are shared with the background. Finally, on the center (filter id: 40-70), we can notice a set of activations from filters that are almost equally shared between foreground and background. It is visible that these shared features have a lower intensity that those located either on the foreground (top-left) or in the background (bottom-right).

These observations confirm that indeed there are informative features (bottom-right) on the background. Yet, their presence is not as prevalent as for the case of those (top-left) that occur on the foreground, i.e. in regions where persons appear (Section 4.3). In future work, we will take a deeper look and analyze where these relevant features come from, either from the physical characteristics of the persons or from the clothing they wear. In this regard, methods of clothing parsing/segmentation [15, 28] might be a possible direction to achieve “clothing-based” feature pooling.

5 CONCLUSION
We have investigated the problem of predicting the geographic location where a human-centered photo was taken. We have conducted an analysis of several aspects to this challenge. Our results suggest that it can be resolved successfully to some extent in an automatic fashion, which even surpasses human performance. A close inspection of the trained models shows that indeed, there are some human-centered characteristics, e.g. clothing style, physical features, accessories, which are informative for the task. Moreover, it reveals that, despite their apparent low occurrence, contextual features, e.g. wall type, natural environment, etc., are also taken into account by the automatic methods.

Acknowledgments: This work was partially supported by the KU Leuven PDM Grant PDM/16/131, and a NVIDIA Academic Hardware Grant.

REFERENCES

[1] P. Agrawal, R. B. Girshick, and J. Malik. 2014. Analyzing the performance of multilayer neural networks for object recognition. In ECCV.
[2] Y. Avrithis, T. Kalantidis, G. Tolias, and E.Spyrou. 2010. Retrieving landmark and non-landmark images from community photo collections. In ACM MAI.
[3] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. Van Gool. 2012. Apparel classification with style. In ACCV.
[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. 2014. Return of the devil in the details: delving deep into convolutional nets. In BMVC.
[5] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pfurtscheller, B. Giroud, and R. Grzeszczuk. 2011. City-scale landmark identification on mobile devices. In CVPR.
[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. 2014. DeCAF: a deep convolutional activation feature for generic visual recognition. In ICML.
[7] V. Escorcia, J. C. Niebles, and B. Ghamen. 2015. On the relationship between visual attributes and convolutional networks. In CVPR.
[8] F. Grün, C. Rupprecht, N. Navab, and F. Tombari. 2016. A taxonomy and library for visualizing learned features in convolutional neural networks. In ICML Visualization for Deep Learning Workshop.
[9] J. Hays and A. A. Efros. 2008. im2gps: estimating geographic information from a single image. In CVPR.
[10] J. Hays and A. A. Efros. 2015. Large-scale image geolocalization. Springer International Publishing, 41–62.
[11] H. J. Kim, E. Duna, and J.-M. Frahm. 2016. Predicting good features for image geo-localization using per-bundle VLAD. In ICCV.
[12] A. Kuznetsova, I. Sutskever, and G. E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In NIPS.
[13] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays. 2015. Learning deep representations for ground-to-aerial geolocalization. In CVPR.
[14] H. Liu, M. Palatucci, and J. Zhang. 2009. Blockwise coordinate descent procedures for the multi-task Lasso, with applications to neural semantic basis discovery. In ICML.
[15] S. Liu, X. Liang, L. Liu, X. Shen, J. Yang, C. Xu, L. Lin, X. Cao, and S. Yan. 2015. Matching-CNN meets KNN: quasi-parametric human parsing. In CVPR.
[16] Juden Mairal, Francis Bach, and Jean Ponce. 2014. Sparse Modeling for Image and Vision Processing. Found. Trends. Comput. Graph. Vis. 8, 2-3 (2014), 85–283.
[17] A. C. Murillo, I. S. Kwak, L. Bourdev, D. Kriegman, and S. Belongie. 2012. Urban tribes: analyzing group photos from a social perspective. In CVPR Workshop on Socially Intelligent Surveillance and Monitoring (SISM).
[18] T. Quack, B. Leibe, and L. Van Gool. 2008. World-scale mining of objects and non-landmark images from community photo collections. In CVPR.
[19] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. 2014. CNN features off-the-shelf: an astounding baseline for recognition. CoRR abs/1403.6852 (2014).
[20] S. Ren, K. He, R. B. Girshick, and J. Sun. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. In NIPS.
[21] E. Samo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun. 2015. Neueroaesthetics in fashion: modeling the perception of fashionability. In CVPR.
[22] K. Simonyan, A. Vedaldi, and A. Zisserman. 2013. Deep inside convolutional networks: visualising image classification models and saliency maps. CoRR abs/1312.6034 (2013).
[23] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. 2014. Striving for simplicity: the all convolutional net. CoRR abs/1412.6806 (2014).
Figure 7: Identified relevant features visualizations for $\mu = 10$ generated with the method from [8].
[24] E. van den Berg and M. P. Friedlander. 2008. Probing the pareto frontier for basis pursuit solutions. SIAM J. Sci. Comput. 31, 2 (2008), 890–912.

[25] A. Vedaldi and K. Lenc. 2015. MatConvNet: Convolutional neural networks for MATLAB. In ACM MM.

[26] M. Wang, S. Yan, Z. Song, and X.-S. Hua. 2011. Predicting occupation via human clothing and contexts. In ICCV 2011.

[27] T. Weyand, I. Kostrikov, and J. Philbin. 2016. PlaNet - photo geolocation with convolutional neural networks. In ECCV.

[28] K. Yamaguchi, H. Kiapour, L. E. Ortiz, and T. L. Berg. 2012. Parsing clothing in fashion photographs. In CVPR.

[29] X.-T. Yuan, X. Liu, and S. Yan. 2012. Visual classification with multitask joint sparse representation. IEEE Trans. Image Processing 21, 10 (2012), 4349–4360.

[30] A. R. Zamir and M. Shah. 2014. Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. IEEE Trans. Pattern Anal. Mach. Intell. 36, 8 (2014), 1546–1558.

[31] M. D. Zeiler and R. Fergus. 2014. Visualizing and understanding convolutional networks. In ECCV.

[32] Y.-T. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven. 2009. Tour the world: building a web-scale landmark recognition engine. In CVPR.