User Modeling on Mobile Device Based on Facial Clustering and Object Detection in Photos and Videos

Ivan Grechikhin[1,2](✉) and Andrey V. Savchenko[1,2]

[1] National Research University Higher School of Economics, Laboratory of Algorithms and Technologies for Network Analysis, Nizhny Novgorod, Russia
{igrechikhin,avsavchenko}@hse.ru

[2] Samsung-PDMI Joint AI Center, St. Petersburg Department of Steklov Institute of Mathematics, Fontanka Str., St. Petersburg, Russia

Abstract. The article describes an approach for extraction of user preferences based on the analysis of a gallery of photos and videos on mobile device. It is proposed to firstly use fast SSD-based methods in order to detect objects of interests in offline mode directly on mobile device. Next we perform facial analysis of all visual data: extract feature vectors from detected facial regions, cluster them and select public photos and videos which do not contain faces from the large clusters of an owner of mobile device and his or her friends and relatives. At the second stage, these public images are processed on the remote server using very accurate but rather slow object detectors. Experimental study of several contemporary detectors is presented with the specially designed subset of MS COCO, ImageNet and Open Images datasets.

Keywords: User modelling · Object detection · Mobile systems · Facial clustering · Convolutional neural network

1 Introduction

Nowadays the world lives through time, when social networks and mobile devices create a vast stream of multimedia data including photos and videos [1]. Such visual data contains specific information about a user, which might be used to improve quality of user modeling and, consequently, accuracy of recommender systems. Recently, deep learning techniques including convolutional neural networks (CNNs) are applied in many image processing tasks [2]. In particular, CNN-based object detection for discovering of particular categories of interests, e.g., interior objects, food, transport, sports equipment, animals, etc., can be used to extract information from the user's photos and videos about his or her preferences [3]. Applications of computer vision in recommender systems becomes all the more attractive. For example, visual recommender systems are used in PInterest services in order to provide relevant photos based on their content [4] with Web-scale object detection and indexing. Visual search and recommendation of clothing, shoes and jewelry with the VisNet architecture [5] and its modification [6] using extraction of visual features and a parallel shallow net.

However, as the photos and videos on mobile devices are created by a user, they potentially contain sensitive personal information, and there is a need for protections of users' privacy. As a result, these multimedia data should not be transferred to remote servers for analysis with the state-of-the-art complex methods [2]. That is why there is a significant demand for developing efficient architectures of CNNs [3,7], which can be implemented directly on a mobile device. There exist a number of architectures that computationally effective and at the same time have good accuracy [8]: SSD (Single shot detector) [9], SSDLite [10], YOLO [11], with different variations of MobileNet [10,12] in a backbone CNN. Unfortunately, if it is necessary to detect small objects (road signs, food, fashion accessories, etc.), the accuracy of such computationally efficient detectors is usually much lower when compared to Faster R-CNN [13] with very deep backbone CNN, such as ResNet [14] or InceptionResNet [15].

In this paper we exploit the fact that not all images can be considered as personal data. For example, panoramic photos, images of food, interiors and showplaces may be sent to remote server for object detection. These types of photos usually contain important information about user's preferences. Hence, we propose here to automatically select private and public visual data. All public photos and videos can be processed remotely with more accurate but slow models. At the same time, private data must be processed offline on mobile device with less precise but lightweight detectors. Here we assume that private image data contains faces of a user (owner of a mobile device) and his or her relatives and friends. Such private photos are chosen using known face recognition [16,17] and clustering techniques [18,19].

The rest of the paper is organized as follows. In Sect. 2 we introduce the proposed pipeline for user preference prediction based on object detection and facial clustering. Experimental results for subsets of MS COCO, ImageNet and Open Images datasets are presented in Sect. 3. Concluding comments and future works are discussed in Sect. 4.

2 Materials and Methods

The quality of recommender systems significantly depends on the solution of the user modeling problem [20]. In this task it is required to develop the user's profile, i.e., predict his or her interests in $C > 1$ categories of preferences, e.g., food, animals, sports equipment, etc. In this paper we assume that a collection of photos and videos is given, and the profile can be defined as a histogram, i.e., frequencies of C categories, which are observed in the gallery of mobile device.

If each category corresponds to some specific objects, this problem can be solved with existing object detection techniques. In order to train the detection models, a balanced dataset with 146 categories of objects with known bounding boxes was created. The balance is chosen at a level of 5000 images per category.

If a category has less pictures, all of them are used, otherwise the images are randomly selected. The list of categories was split into the following high-level interests: activity, appliances, children, indoors, fashion, food, musical instruments, pets/animals, planting, product/services, outdoors, sport and transport. Each high-level group contains 2–15 different types of particular objects from MS COCO [22], ImageNet [21] and Open Images Dataset (OID) [3]. Sample images are given in Fig. 1.

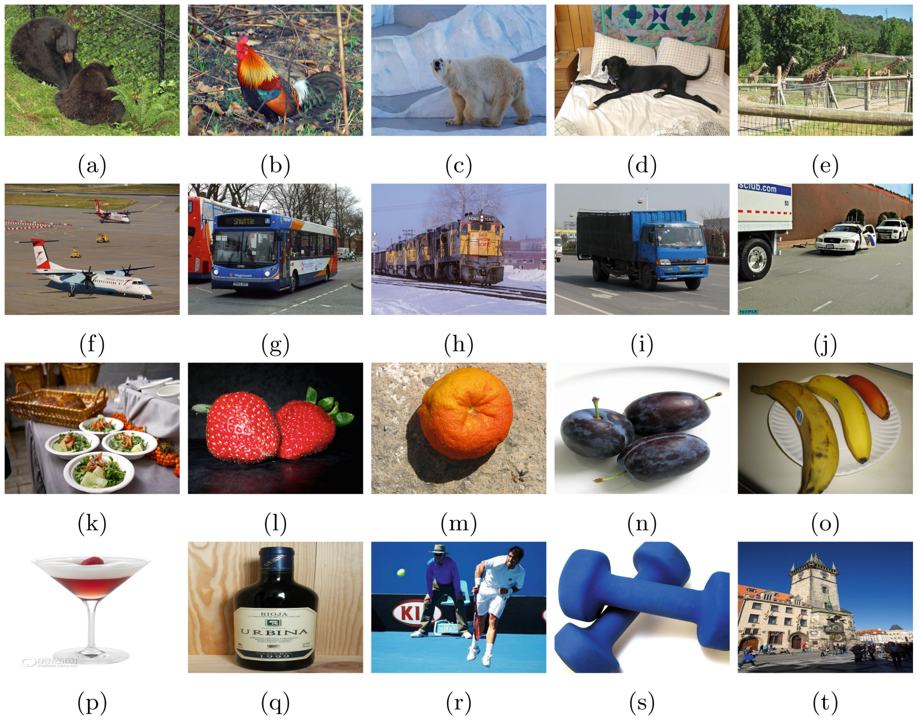


Fig. 1. Example of images from different categories used for training: (a)–(e) animals (bear, cock, polar bear, dog, giraffe), (f)–(j) transport (plane, bus, train, truck, car), (k)–(o) fresh food (salad, strawberry, orange, plum, banana), (p)–(t) mixture from various high-level groups (drink, wine, tennis, dumbbell, tower).

Computational complexity and memory costs of the most accurate object detectors [13], e.g., Faster R-CNN [13] with ResNet or InceptionResNet backbones, do not allow them to be implemented even in the most contemporary mobile devices. At the same time, the usage of remote server to process *all* multimedia data of an user is not permitted because of the privacy constraints. That is why development of fast object detection models is so important [8,9,13]. Unfortunately, fast SSD-based methods are known to be much less accurate when compared to Faster R-CNN, if the small objects should be detected [13]. However, namely such small objects usually defines the user preferences on his or her photos and videos. Hence, in this paper we propose to implement the multi-stage procedure of user modeling (Fig. 2).

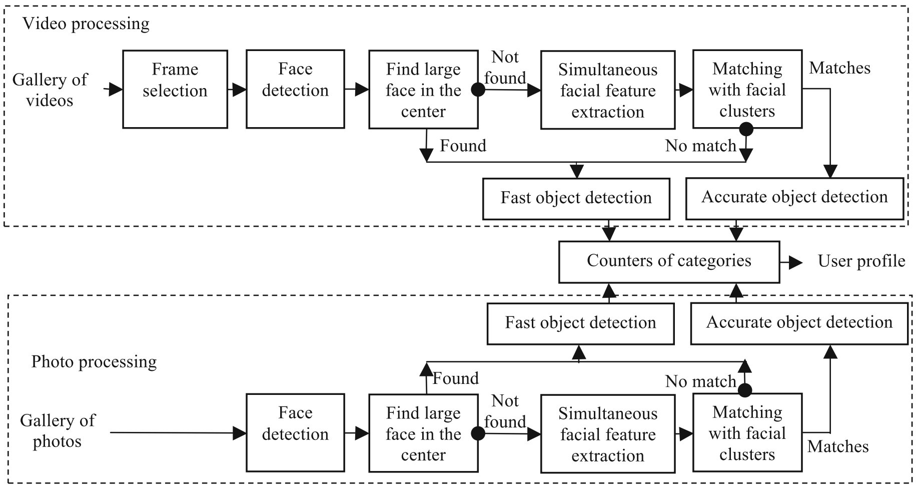
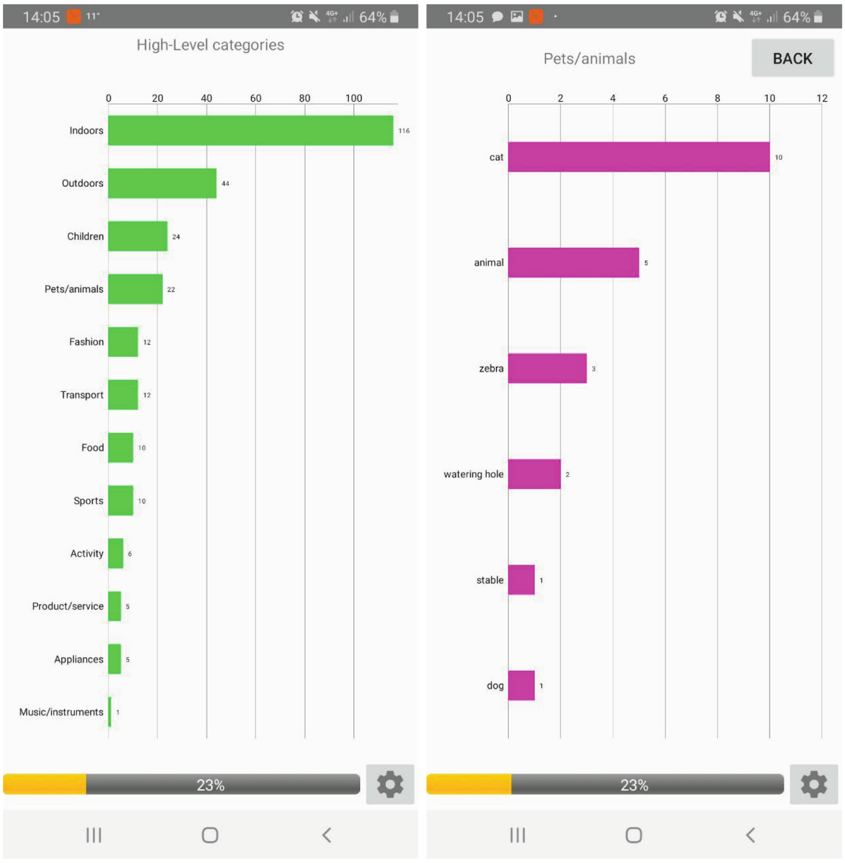


Fig. 2. Proposed pipeline for visual user modeling based on object detection and facial clustering.

Here public data is extracted from the whole collection of images and videos using facial analytic techniques implemented in offline mode on a mobile device. Namely, private photos and videos are assumed to contain faces of an owner of mobile device and his or her friends, relatives, etc. At first, $N \geq 0$ faces are detected on all photos using, e.g., lightweight SSD-based object detectors.

As the facial images $X_n, n \in \{1, ..., N\}$ do not contain labels of concrete subjects on the photos, the problem of extracting people from the gallery should be solved by clustering methods. Namely, every face on image should be assigned to one of the labels $1...K$, where K is a number of people on images in the user's gallery. As K is usually unknown [23], hierarchical agglomerative or density based spatial clustering methods should be used [24]. In order to apply these methods, numerical feature vector should be extracted from each detected facial image. As the faces are observed in *unconstrained* conditions, modern transfer learning and domain adaptation techniques can be used for this purpose [2]. According to these methods, the large external dataset of celebrities is used to train a deep CNN [25,26]. The outputs of one of the last layers of this CNN form D-dimensional ($D \gg 1$) off-the-shelf features [27] $\mathbf{x}_n = [x_{n;1}, ..., x_{n;D}]$ of the photos X_n from the gallery [17]. These feature vectors are L_2-normed in order to provide additional robustness to variability of observation conditions.

The same procedure is implemented for a set of frames from each *video* from a gallery (Fig. 2), and features of the obtained cluster centers are appended to the set of all facial feature vectors. After that, the final clustering is conducted to detect clusters with sufficient number of faces in the gallery. We assume that these faces belong to owner of the device and to his relatives and friends, and all information on corresponding visual data is personal. A user may also manually choose additional private photos.



(a) (b)

Fig. 3. Example of user profile extracted by our mobile demo application: (a) high-level groups, (b) detailed object categories

We try to predict if each photo can be loaded to external server or it contains private information to see if the processing should be implemented on a mobile device in the offline mode. The photo is considered to be private if it contains faces either in the middle (portrait photo) or a face from rather large facial clusters (family members or close friends). After that the "private" photos are fed to the input of simple TensorFlow object detector, e.g., SSDLite with MobileNet backbone. The "public" photos are processed by rather complex and accurate

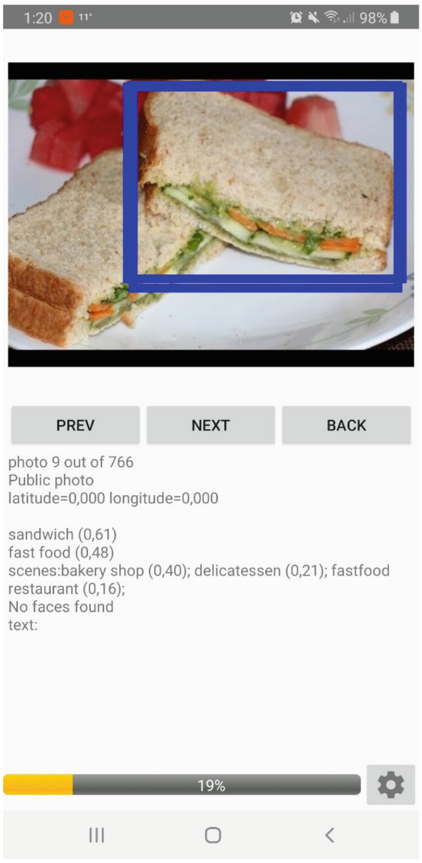


photo 9 out of 766
Public photo
latitude=0,000 longitude=0,000

sandwich (0,61)
fast food (0,48)
scenes:bakery shop (0,40); delicatessen (0,21); fastfood restaurant (0,16);
No faces found
text:

(a)

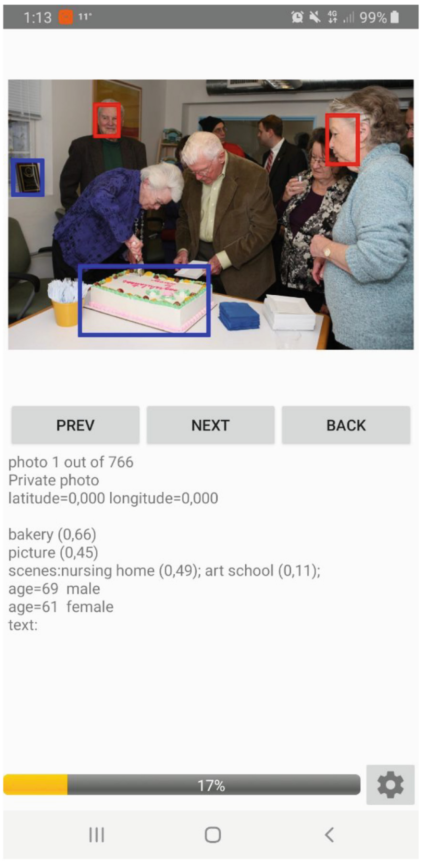


photo 1 out of 766
Private photo
latitude=0,000 longitude=0,000

bakery (0,66)
picture (0,45)
scenes:nursing home (0,49); art school (0,11);
age=69 male
age=61 female
text:

(b)

Fig. 4. Example of detections on mobile devices: (a) public data, (b) private data.

detector, e.g., the Faster R-CNN with Inception or InceptionResNet. We map the detected objects on the predefined list of categories and obtain the most frequent categories filling the user profile. The videos are processed similarly: each of 3–5-th frames in each video is selected and the same procedure is repeated, though the decision about private/public status is made for a whole video by considering the video public only if all frames are marked as "public". Then the list of obtained user preferences from public photos and videos is sent back to the mobile device, where it is combined with the results of offline detector in order to obtain the final user profile, i.e., the total preferences histogram.

The proposed pipeline (Fig. 2) is implemented in a demo mobile application for Android. It can work in offline mode but supports object detection with Faster R-CNN with InceptionResNet v2 backbone on a remote Flask server. This application sequentially processes all photos from the gallery in the background thread. However, the intermediate results are available, so it is not necessary to wait for a long time. The resulted profile (histogram of detected objects for each high-level group) is displayed in the main window (Fig. 3a). It is possible to tap any bar in this histogram, and a new form with detailed categories is shown (Fig. 3b). If the user taps a concrete category, a special form appears, which contains a list of all photos from the gallery, in which the corresponding object was found. Sample processing of public and private photos are given in a Fig. 4.

3 Experimental Results

In the first experiment we examined facial clustering results [18] using the GFW (Grouping Faces in the Wild) dataset [28]. It contains facial images from albums of 60 real users from a Chinese social network portal. The size of an album varies from 120 to 3600 faces, with average number of subjects $C = 46$. Three CNNs were used for feature extraction: VGGFace [25], ResNet-50 fine-tuned on VGGFace2 [26] and MobileNet [18,29], trained by us on the same VGGFace2 dataset [26]. The VGGFace, VGGFace2 and MobileNet extract $D = 4096$, $D = 2048$ and $D = 1024$ non-negative features in the output of "fc7", "pool5_7x7_s1" and "reshape_1/Mean" layers from 224×224 RGB images, respectively. The estimates of relation of the number of clusters to the number of different people in the dataset K/C, ARI (Adjusted Rand Index), AMI (Adjusted Mutual Information), homogeneity, completeness and BCubed F-measure for different cluster linkages are shown in Table 1. The best results are marked by bold.

Table 1. Facial clustering results, GFW dataset.

Method	Features	K/C	ARI	AMI	Homogeneity	Completeness	F-measure
Single	VGGFace	4.10	0.440	0.419	0.912	0.647	0.616
	VGGFace2	3.21	0.580	0.544	0.942	0.709	0.707
	MobileNet	4.19	0.492	0.441	0.961	0.655	0.636
Average	VGGFace	1.42	0.565	0.632	0.860	0.751	0.713
	VGGFace2	1.59	0.603	**0.663**	0.934	0.761	0.746
	MobileNet	1.59	**0.609**	0.658	0.917	**0.762**	**0.751**
Complete	VGGFace	**0.95**	0.376	0.553	0.811	0.690	0.595
	VGGFace2	1.44	0.392	0.570	0.916	0.696	0.641
	MobileNet	1.28	0.381	0.564	0.886	0.693	0.626
Weighted	VGGFace	1.20	0.464	0.597	0.839	0.726	0.662
	VGGFace2	1.05	0.536	0.656	0.867	**0.762**	0.710
	MobileNet	1.57	0.487	0.612	0.915	0.727	0.697
Median	VGGFace	5.30	0.309	0.307	0.929	0.587	0.516
	VGGFace2	4.20	0.412	0.422	0.929	0.639	0.742
	MobileNet	6.86	0.220	0.222	**0.994**	0.552	0.411

Here the specially-trained MobileNet [18] is in most cases more accurate than the widely-used VGGFace. As expected, the quality of this model is slightly lower when compared to much deeper ResNet-50 trained on the same VGGFace2 dataset. However, this MobileNet with average-linkage clustering is characterized by the highest BCubed F-measure, which is slightly higher than the best BCubed F-measure (0.745) reported by the authors [28]. The most important advantage of MobileNet model is an excellent speed (5–10-times than VGGFace and VGGFace2) appropriate for offline mobile processing. Moreover, the dimensionality of the feature vector is 2–4-times lower leading to the faster computation of distances during clustering.

In the next experiments the gathered dataset (Fig. 1) was used to train several object detection architectures using the Tensorflow Object Detection API, namely, SSDLite (with image size 300×300 and 512×512) with MobileNet backbone, Faster R-CNN with Inception v2, ResNet-50/101 and InceptionResNet v2 with Atrous convolutions. The size of the model files and average inference time on a laptop (MacBook Pro 2015) and mobile phone (Samsung S9+) are presented in Table 2. Our SSDLite models are rather fast (200 ms and 500 ms per photo on Samsung S9+ for SSDLite-300 and SSDLite-512, respectively). However, Faster R-CNN models are inappropriate for offline detection on mobile devices (1.2 s per image for simple Inception v2 model).

Table 2. Performance analysis of object detection models.

CNN	Model size, MB	Average inference time, s.	
		Laptop	Mobile phone
SSDLite-300 (MobileNet v2)	31.83	0.16	0.30
SSDLite-512 (MobileNet v2)	31.83	0.21	0.52
Faster R-CNN (Inception v2)	64.91	0.40	1.25
Faster R-CNN (InceptionResNet v2)	204.34	1.01	2.39

The detection results for several testing images are shown in Fig. 5. As expected, a simple SSD-based model here misses many small objects relevant for user modeling. For example, this method does not detect anything on Fig. 5a.

The quantitative results of object detection models were obtained using 5000 testing images in each of 146 categories. In particular, we computed recall (an average rate of detected objects from one class) and precision (average rate among categories of correctly detected object for one category to all detected objects of this category). Additionally, some of the categories are considered as “family” categories to each other. For example, sometimes it is not mistake if a category “animal” is detected instead of concrete objects, e.g., “cat” or “dog”, detection of “building” instead of “skyscraper” is also suitable, etc. Because of this, precision and recall were calculated “as-is” (exact matching) and with taking into account these family categories. The results are presented in Table 3.

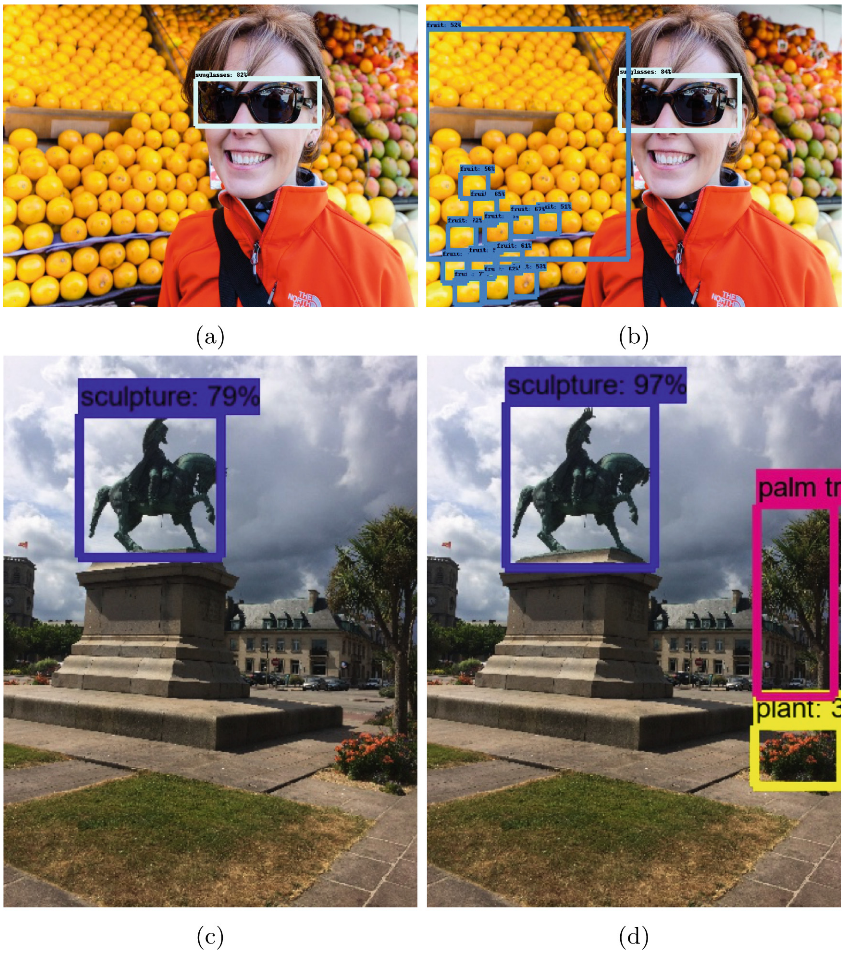


Fig. 5. Sample detection results: (a), (c) SSDLite (MobileNet v2); (b), (d) Faster R-CNN (InceptionResNet).

Here there exist three models with the highest quality: Faster R-CNN with both versions of InceptionResNet (original and quantized by standard procedure from TensorFlow) and ResNet-101. Overall, InceptionResNet returns more detections than the model with Inception or ResNet backbones.

In addition, the most reliable categories were selected for each model so their average recall was more than 0.75 and their precision was calculated (Table 4). The asterisks here mean that some of the important categories were not included in selected categories e.g., faces and buildings. Here the main quality criterion is the number of selected categories, which characterizes how stable is the model for a variety of categories. ResNet-101 on average has higher recall and precision in every taken measurement, however, the number of selected categories is smaller.

Table 3. Estimates of object detection precision/recall.

CNN	Exact match		Family match	
	Recall	Precision	Recall	Precision
Faster R-CNN (InceptionResNet v2)	0.425	0.477	0.448	0.534
Faster R-CNN (InceptionResNet v2 quantized)	0.425	0.471	0.448	0.528
Faster R-CNN (Inception v2)	0.393	0.537	0.414	0.593
Faster R-CNN (ResNet-50)	0.332	0.583	0.35	0.636
Faster R-CNN (ResNet-101)	0.465	0.562	0.485	0.618
SSDLite (MobileNet v2)	0.149	0.465	0.166	0.525
SSDLite (MobileNet v2 quantized)	0.149	0.463	0.166	0.524

Table 4. Results of object detection with selection of reliable classes with recall ≥ 0.75.

CNN	Number of selected categories	Precision
Faster R-CNN (InceptionResNet v2)	78	0.662
Faster R-CNN (InceptionResNet v2 quantized)	79	0.663
Faster R-CNN (Inception v2)	44	0.762
Faster R-CNN (ResNet-50)*	30	0.838
Faster R-CNN (ResNet-101)	67	0.760
SSDLite (MobileNet v2)*	3	0.773
SSDLite (MobileNet v2 quantized)*	3	0.768

This is due to lower precision/recall for categories that are considered to be important. The lowest precision is for such categories as house, animal, car and, especially, face.

4 Conclusion

In this paper we proposed the novel algorithm for user modeling (Fig. 2) based on detection of special objects in photos and videos in a gallery of mobile device. It is known that lightweight SSD-based models cannot be used for small object detection (Fig. 5), so the processing on a remote server with contemporary GPU is needed. However, transfer of *all* photos to the server is usually undesirable due to the presence of personal information in many photos. Hence, we implemented the heuristical rule-based classification of private and public photos/videos based on the presence of faces of relatives/friends on private images. We implemented an efficient facial analysis in offline mode using specially trained MobileNet [18], which was practically as accurate as the state-of-the-art facial feature extractors (Table 1).

In order to train various object detectors and compare their performance (Tables 2, 3 and 4), we gathered a special dataset (Fig. 1). As expected, lightweight models, e.g, SSDLite, are faster and take less memory than Faster R-CNN with different backbones, however their accuracy is also much lower (Table 3). The proposed approach was implemented in a special mobile application. In future, it is important to detect private photos more accurately by, e.g., using text recognition techniques for processing of scanned documents (tickets, passports, etc.). Moreover, other backbones for SSD should be also examined as current MobileNet v2 leads to rather low quality (Fig. 5).

Acknowledgements. This research is based on the work supported by Samsung Research, Samsung Electronics and was prepared within the framework of the Academic Fund Program at the National Research University Higher School of Economics (HSE University) in 2019 (grant No. 19-04-004) and by the Russian Academic Excellence Project "5-100".

References

1. Harrison, G.: Next Generation Databases: NoSQL, NewSQL, and Big Data. Springer, Heidelberg (2015). https://doi.org/10.1007/978-1-4842-1329-2
2. Goodfellow, I.: Deep Learning (Adaptive Computation and Machine Learning series). MIT Press, Cambridge (2016)
3. Kuznetsova, A., et al.: The open images dataset v4: unified image classification, object detection, and visual relationship detection at scale. arXiv preprint arXiv:1811.00982 (2018)
4. Zhai, A., et al.: Visual discovery at Pinterest. In: Proceedings of the 26th International Conference on World Wide Web Companion, pp. 515–524 (2017)
5. Shankar, D., Narumanchi, S., Ananya, H.A., Kompalli, P., Chaudhury, K.: Deep learning based large scale visual recommendation and search for e-commerce. arXiv preprint arXiv:1703.02344 (2017)
6. Andreeva, E., Ignatov, D.I., Grachev, A., Savchenko, A.V.: Extraction of visual features for recommendation of products via deep learning. In: van der Aalst, W.M.P. (ed.) AIST 2018. LNCS, vol. 11179, pp. 201–210. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-11027-7_20
7. Huang, J., et al.: Speed accuracy trade-offs for modern convolutional object detectors. arXiv preprint arXiv:1611.10012 (2016)
8. Qin, Z., Zhang, Z., Chen, X., Wang, C., Peng, Y.: FD-MobileNet: improved MobileNet with a fast downsampling strategy. In: Proceedings of the 25th International Conference on Image Processing (ICIP), pp. 1363–1367. IEEE (2018)
9. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
10. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Inverted residuals and linear bottlenecks: mobile networks for classification, detection and segmentation. arXiv preprint arXiv:1801.04381 (2018)
11. Redmon, J., Farhadi, A.: YoloV3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
12. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)

13. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS), pp. 91–99 (2015)
14. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38
15. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, Inception-ResNet and the impact of residual connections on learning. In: Proceedings of Thirty-First AAAI Conference on Artificial Intelligence, vol. 4, p. 12 (2017)
16. Savchenko, A.V., Belova, N.S.: Unconstrained face identification using maximum likelihood of distances between deep off-the-shelf features. Expert Syst. Appl. **108**, 170–182 (2018)
17. Savchenko, A.V.: Efficient statistical face recognition using trigonometric series and CNN features. In: Proceedings of the 24th International Conference on Pattern Recognition (ICPR), pp. 3262–3267. IEEE (2018)
18. Savchenko, A.V.: Efficient facial representations for age, gender and identity recognition in organizing photo albums using multi-output ConvNet. PeerJ Comput. Sci. **5**, e197 (2019). https://doi.org/10.7717/peerj-cs.197
19. Pan, S.J.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
20. Lakiotaki, K., Matsatsinis, N.F., Tsoukias, A.: Multicriteria user modeling in recommender systems. IEEE Intell. Syst. **26**(2), 64–76 (2011)
21. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255. IEEE (2009)
22. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. arXiv preprint arXiv:1405.0312 (2014)
23. Sokolova, A.D., Kharchevnikova, A.S., Savchenko, A.V.: Organizing multimedia data in video surveillance systems based on face verification with convolutional neural networks. In: van der Aalst, W.M.P., et al. (eds.) AIST 2017. LNCS, vol. 10716, pp. 223–230. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73013-4_20
24. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier, Burlington (2011)
25. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: Proceedings of the British Conference on Machine Vision (BMVC), vol. 1, p. 6 (2015)
26. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman. A.: VGGFace2: a dataset for recognizing faces across pose and age. In: Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG 2018), pp. 67–74 (2018)
27. Sharif, R.: CNN features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, pp. 806–813. IEEE (2014)
28. Yue, H., Kaidi, C., Cheng, L., Chen, C.L.: Merge or not? Learning to group faces via imitation learning. arXiv preprint arXiv:1707.03986 (2017)
29. Kharchevnikova, A.S., Savchenko, A.V.: Neural networks in video-based age and gender recognition on mobile platforms. Opt. Mem. Neural Netw. **27**(4), 246–259 (2018)