Pattern Recognition 121 (2022) 108248

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Preference prediction based on a photo gallery analysis with scene recognition and object detection

A.V. Savchenko[a,*], K.V. Demochkin[b], I.S. Grechikhin[b]

[a] HSE University, Laboratory of Algorithms and Technologies for Network Analysis, Nizhny Novgorod, Russia
[b] St. Petersburg Department of Steklov Institute of Mathematics, Samsung-PDMI Joint AI Center, St. Petersburg, Russia

ARTICLE INFO

Article history:
Received 28 October 2019
Revised 5 August 2021
Accepted 11 August 2021
Available online 13 August 2021

Keywords:
Scene recognition
Event recognition
Object detection
Recognition of a set of images
CNN (Convolutional neural network)
Mobile device

ABSTRACT

In this paper, a user modeling task is examined by processing mobile device gallery of photos and videos. We propose a novel engine for preferences prediction based on scene recognition, object detection and facial analysis. At first, all faces in a gallery are clustered, and all private photos and videos with faces from large clusters are processed on the embedded system in offline mode. Other photos may be sent to the remote server to be analyzed by very deep sophisticated neural networks. The visual features of each photo are obtained from scene recognition and object detection models. These features are aggregated into a single descriptor in the neural attention unit. The proposed pipeline is implemented in mobile Android application. Experimental results for the Photo Event Collection, Web Image Dataset for Event Recognition and Amazon Fashion data demonstrate the possibility to efficiently process images without significant accuracy degradation.

1. Introduction

Important features of today's mobile devices are personalized services that adapt to individual users and collect user-specific data. Recommendations are developed to assist customers in finding relevant things within large item collections. The design of such systems requires the careful consideration of a user modeling algorithm, which defines user's interests in his or her profile. Conventional content-based recommender systems use only structured and textual data [1]. However, a large number of photos is available on a mobile device, which can be also used for understanding of such interests as sport, gadgets, fitness, cloth, cars, food, pets, etc. Indeed, the usage of contemporary pattern recognition methods for photos from the gallery has many advantages for improving the quality of recommender system [2].

It is important to emphasize that processing of photos has a requirement to protect users' privacy [3]. The mobile phone contains much more photos when compared to the number of photos in their publicly available profiles in social networks. Most photos contain private information, for which the user may not permit processing on the remote server. Hence, the analytics engines should be preferably run on the embedded system. As a result, the state-of-the-art very deep convolutional neural networks (CNNs) [4] cannot be directly applied due to their enormous inference time and energy consumption. Thus, in this paper we consider several directions for faster prediction of the user preferences [5], namely, scene recognition [6,7] to extract such interests as art and theaters, nightlife, sport, etc.; detection of objects [8,9] including food, pets, musical instruments, vehicles, brand logos [10]; analysis of demography and sociality by processing facial images in photos and videos including facial clustering [11].

This paper makes two main contributions to efficient recognition of photos and videos. Firstly, we propose a technological framework to predict user's preferences given photos that exploits as much as possible offline processing on the personal device. It is assumed that the preferences can be represented as a discrete distribution over $C > 1$ pre-defined categories of interest, where C is the total number of categories. This distribution is proposed to be estimated based on a mobile device photo gallery in a form of a histogram, i.e., a set of (category, count) or (category, frequency) tuples. Secondly, we describe a representation of image by using scores and embeddings from scene recognition neural network combined with the scores of object detector. We demonstrate that such a representation is suitable for several different pattern recognition tasks including visual product recommendation and event recognition [12]. In particular, we propose to combine obtained representations of all photos from a gallery into a single descriptor of a user by modifying the neural aggregation module with an

[*] Corresponding author.
E-mail addresses: avsavchenko@hse.ru (A.V. Savchenko), kdemochkin@gmail.com (K.V. Demochkin), gis1093@mail.ru (I.S. Grechikhin).

https://doi.org/10.1016/j.patcog.2021.108248
0031-3203/© 2021 Elsevier Ltd. All rights reserved.

attention mechanism [13] originally used in the video-based face recognition tasks [14] in order to decrease its running time without degradation in accuracy.

The rest part of the paper is organized as follows. In Section 2, we review recent articles related to our task. In Section 3, we present the proposed pipeline for inferring user's profile by processing a set of photos. In Section 4, the trade-off between accuracy and complexity of various CNN-based models is experimentally studied. Finally, concluding comments are given in Section 5.

2. Literature survey

2.1. Image, scene and event recognition

The majority of computer vision literature is focused on the problems of object recognition or scene classification [6], partially due to the simplicity of object and scene concepts [12] and the availability of large-scale datasets, such as ImageNet and Places. However, new tasks has been recently appeared, for example, recognition of complex images with large variations in visual appearance and structure, such as events [15–17] that captures the complex behavior of a group of people, interacting with multiple objects, and taking place in a specific environment (holidays, sport events, wedding, activity, etc.) [12].

Nowadays deep CNNs provide an excellent accuracy in many above-mentioned large-scale image classification problems [18]. If the training sample is rather small to train a deep CNN from scratch, two techniques are mainly applied to represent visual data. In both methods a large external dataset (e.g., ImageNet-1000), is used to pre-train a deep CNN. The first one is a transfer learning [4], in which the last logistic regression layer of the pre-trained CNN is replaced to the new layer with Softmax activations and C outputs. The final step in transfer learning is fine-tuning of this neural network using the training set from the limited sample of instances.

This procedure can be modified by replacing the logistic regression in the last layer to more complex classifier [19]. In this case the off-the-shelf features are extracted using the outputs of one of the last layers of pre-trained or fine-tuned CNN. It is especially suitable for small training samples when the results of the fine-tuning is not too accurate. Namely, the images are fed to the CNN, and the outputs of the one-but-last layer are used as the feature vectors. Such deep learning-based feature extractors allow training a general classifier, such as random forest (RF), support vector machine (SVM), multi-layered perceptron or gradient boosting, that performs nearly as well as if a large training dataset of images from these C classes is available [4].

Various recent studies have shown that the increase of the number of layers in a CNN leads to increase of accuracy and the running time [8,18]. As a result, this time may be too high for real-time processing especially if expensive GPUs are unavailable [20]. The most remarkable research direction in improving the speed of CNNs is the optimization of algorithms and neural network architectures. Conventional compression includes the usage of pruning in which connections between neurons with low weights are removed and the network is fine-tuned, until achieving the allowable decrease in accuracy. Such pruning reduces the model size but does not lead to significant inference speed-up. However, there exist various structural pruning methods [21,22], which remove entire convolutional channels.

In order to recognize complex images, several CNNs may be combined. For instance, the paper [12] introduces an ensemble of two CNNs trained to classify objects from ImageNet and scenes from Places dataset. Four different layers of fine-tuned CNN were used to extract features and perform Linear Discriminant Analysis to obtain the top entry in the ChaLearn LAP 2015 cultural event recognition challenge [23].

2.2. Object detection

A special interest for user modeling is involved in the CNN-based object detection, which can discover particular categories of interests, such as interior objects, food, transport, sports equipment, animals, etc. It is important to emphasize that there exist several papers devoted to application of object detection in scene analysis. For example, detected bounding boxes are projected onto multi-scale spatial maps for increasing the accuracy of event recognition [16]. The neural detectors are much more computationally difficult than the above-mentioned CNNs used for image recognition, because they contain additional structures to transfer visual representation obtained by the CNN into predictions of multiple object positions and confidence scores. That is why there is a significant demand for developing efficient architectures of CNNs [8], which can be implemented directly on a mobile device. There exist a number of computationally efficient architectures that have good accuracy: YOLO (You Only Look Once), SSD (Single shot detector) and/or SSDLite with different variations of MobileNet [24] in a backbone CNN. Unfortunately, if it is necessary to detect small objects (road signs, food, fashion accessories, etc.), the accuracy of such one-shot detectors is usually much lower when compared to Faster R-CNN with very deep backbone CNN, such as ResNet or InceptionResNet.

2.3. Visual recommender systems

Development of visual recommender systems has become all the more important in the last few years [2,25]. A probabilistic matrix factorization model helped to unify visual and functional aspects pf products to estimate user preferences on products [26]. The clothing, shoes and jewelry from Amazon product dataset are recognized in [27] by an extraction of ResNet-based visual features and a special shallow network. A deep CNN and Attractiveness Visual Features have been applied to develop explanations for a visual recommender system of artistic images [28]. Visual search and recommendations have been implemented on Pinterest using Web-scale object detection and indexing of a multi-task metric learning network has been developed [29]. Moreover, the PinnerSage end-to-end recommender system has been recently introduced to represent each user of PInterest via multi-modal embeddings to provide high quality personalized recommendations by clustering users actions [30].

3. Materials and methods

3.1. Proposed approach

The main task of this paper is to predict a probability distribution $\{\pi_c \in [0,1] | c \in \{1,\dots,C\}\}$, over C categories as the interest distribution for a new user given a *set* of his or her M photos $X(m), m \in \{1,\dots,M\}$ [5,25]. This distribution can be used either directly to make recommendations or as initial information to solve the cold start problem. We assume that users will grant access to their media files for modeling of their preferences.

In order to extract specific interests from each photo or video, object detection and scene recognition methods are used. At first, we analyzed large object detection datasets, such as MS COCO, OID (Open Image Dataset v4) and ImageNet object datasets, and selected $O = 145$ categories from them related to user interests [9]. All categories were organized into several high-level groups: outdoors, indoors, sports, food, activity, fashion, musical instruments,

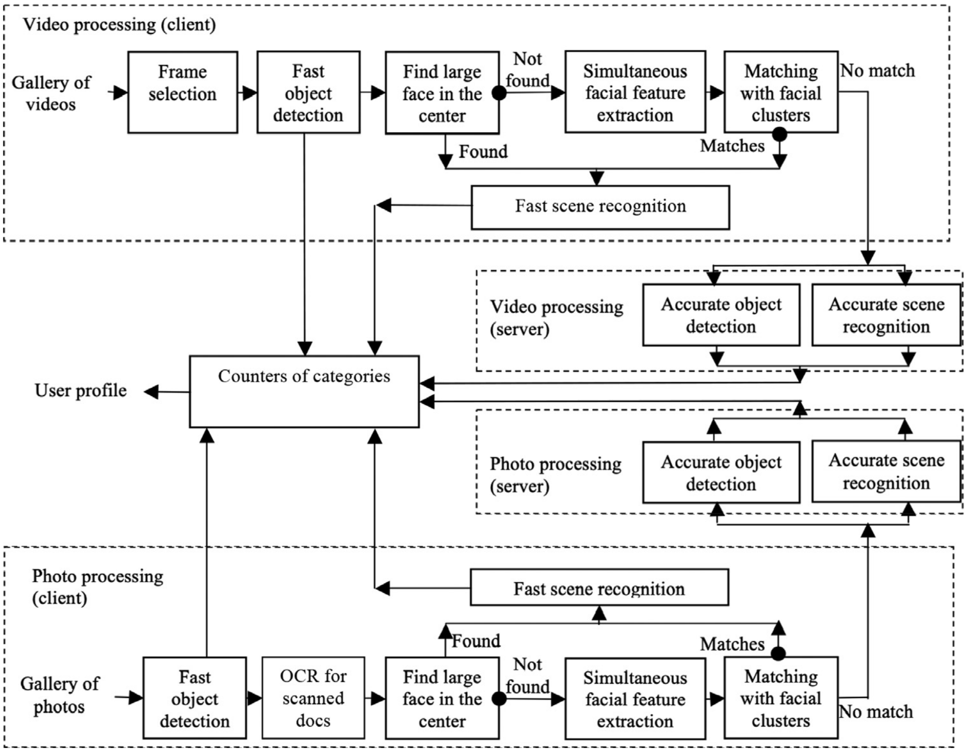

Video processing (client)
Gallery of videos
Frame selection
Fast object detection
Find large face in the center
Not found
Simultaneous facial feature extraction
Matching with facial clusters
No match
Found
Matches
Fast scene recognition
Video processing (server)
Accurate object detection
Accurate scene recognition
User profile
Counters of categories
Photo processing (server)
Accurate object detection
Accurate scene recognition
Photo processing (client)
Fast scene recognition
Found
Matches
Gallery of photos
Fast object detection
OCR for scanned docs
Find large face in the center
Not found
Simultaneous facial feature extraction
Matching with facial clusters
No match

Fig. 1. Proposed visual preferences prediction pipeline.

transport, services, appliances and toys, so that it is actually necessary to estimate probability distribution of interests in each group separately. In addition, we examined the Places365 and Places-Extra69 datasets [7] and chose $S = 337$ scenes related to abovementioned groups of interests.

The proposed pipeline is presented in Fig. 1. It is an extension of approach from [5,9], which does not contain significant demography analysis. Here, firstly, objects (including faces) on all photos from the gallery are detected in the “Fast object detection” block in an offline mode using efficient CNNs, such as MobileNet and SSDLite [24]. Secondly, it is predicted, whether a photo may be loaded to remote server. At first, scanned documents are considered. The text is detected in a photo using existing OCR (optical character recognition) engines, such as Firebase Machine learning kit (ML Kit). Next, the detected text is fed into the fully-connected neural network with two hidden layers from our previous paper [31]. It was demonstrated that it improves the accuracy to more than 97.2% for specially gathered balanced dataset of sensitive scanned documents. If the detected text is classified as sensitive, then this photo is considered as a private one [32] and must be further processed on mobile device. In other case, detected facial regions are analyzed. If the photo is a portrait, i.e., its central part contains at least one face with width greater than a predefined threshold multiplied by a width of the photo, then it is also considered as a private image. In order to estimate this threshold, we gathered 4 galleries of different users with at least 200 portrait images in each gallery and manually labeled all personal photos. After that we computed the minimal width of the face detected by Faster R-CNN with InceptionResNet in a central 2/3 part divided by the width of the photo. As a result, we obtained threshold 0.05 for this ratio, which helps to identify 98.5% portrait photos.

Thirdly, the following heuristic is used: the photo is considered to be private if it contains a face appeared at several photos made in at least 2 days. Such person is potentially important to the owner of the mobile device so he or she can be supposed to represent a family member or a closed friend. In order to apply such an heuristic, all $R \geq 0$ faces detected in the photos are fed into a CNN trained for face identification [19] to simultaneously extract numerical feature vectors of faces and predict age and gender [11] in the “Simultaneous facial feature extraction” unit. As the faces are observed in unconstrained conditions, modern transfer learning and domain adaptation techniques are used for this purpose [4]. According to these methods, the large external dataset of celebrities is used to train a deep CNN. The outputs of one of the last layers of this CNN form the D-dimensional ($D \gg 1$) embeddings $\mathbf{x}_r = [x_{r;1}, \ldots, x_{r;D}]$ of the rth facial image from the photo gallery. These feature vectors are L_2-normed to provide additional robustness to variability of observation conditions [33]. Fourthly, as the facial images do not contain labels of particular subjects on the photos, the problem of extracting people from the gallery should be solved by clustering methods in the “Matching with facial clusters” block. Namely, every face on image should be assigned to one of the labels $1, \ldots, \tilde{R}$, where $\tilde{R}$ is a number of people on images in the photo gallery [33]. As $\tilde{R}$ is unknown, all resulted facial identity feature vectors are grouped by hierarchical agglomerative or density based spatial clustering methods, such as DBSCAN. The gender and the birth year of a person in each cluster are estimated by appropriate fusion technique. Selfies are automatically detected using EXIF information about camera model and focal length. An owner of the device is associated with the facial cluster with the largest number of selfies. By using gender and birth year predictions, other relations (“girl-friend” or “boy-friend”, “father”, “mother” and “child”) are detected during this matching with facial clusters. The demography analysis procedure is presented in Algorithm 1. It has two hyper-parameters, namely, the minimal number of faces in a cluster and the maximal distance ϵ between photos of the same person. The latter is estimated using the datasets suitable for face verification, such as Labeled Faces in-the-Wild.

Algorithm 1 Demography analysis in user visual preference prediction.

0

```
Input: gallery of photos $X(m), m \in \{1,...,M\}$
Initialize the number of found faces $R := 0$
for each photo $m \in \{1,...,M\}$ do
    Feed image $X(m)$ into a fast object detector and detect facial regions
    for each detected face do
        Assign $R := R+1$
        Feed facial image into the multi-output CNN~[11] to extract embeddings $\mathbf{x}_R$ and predict scores of ages $\mathbf{a}_R$, genders $\mathbf{g}_R$ and ethnicities $\mathbf{e}_R$
    end for
end for
Perform clustering of a set of facial descriptors $\{\mathbf{x}_r\}, r \in \{1,...,R\}$ to obtain $\tilde{R}$ clusters
for each cluster $\tilde{r} \in \{1,...,\tilde{R}\}$ do
    Estimate born year, gender and ethnicity by averaging of $\mathbf{a}_r, \mathbf{g}_r, \mathbf{e}_r$ in this cluster
end for
Compute *Histogram* of the number of clusters per gender/age range
if there exist only one cluster with maximal number of selfies greater than 0 then
    Obtain cluster $\tilde{r}_{owner} \in \{1,...,\tilde{R}\}$ that contains the maximal number of photos
    return *Histogram* and age, gender and ethnicity of the estimated owner $\tilde{r}_{owner}$ of the mobile device
end if
return *Histogram* and the following status: "The number of photos in the gallery is not enough to perform demography analysis"
```

Fifthly, the scenes on the private photos are recognized on the mobile device in the "Fast scene recognition" unit by using efficient CNNs, such as MobileNets. Sixthly, other ("public") photos are sent to remote Flask server to be processed in the "Accurate object detection" unit by complex but accurate object detector, such as Faster R-CNN with Inception or InceptionResNet backbone, and scene classifier, such as Inception or EfficientNet, in the "Accurate scene recognition" unit. After that the photos are immediately removed at the remote server to prevent the user's privacy. Seventhly, the detected objects and recognized scenes are mapped to the predefined list of categories and the resulted categories are combined into the user's profile in the "Counters of categories" unit by computing the histogram of categories recognized by scene classifier and object detector. The videos are processed similarly: each of kth frames ($k = 3\ldots5$) in each video are selected in the "Frame selection" unit, and the same procedure is repeated, though a video is considered as a public one only if all its frames are marked as public.

3.2. Representation of visual data using scene recognition and object detection models

In this subsection we demonstrate how to solve several different complex image recognition tasks with high accuracy using only results of image processing in the pipeline (Fig. 1). Let us consider details about proposed representation of photos and videos suitable for complex analysis, for instance, in scene and event classification tasks. In this case a training set of N users should be available, so that each nth user ($n \in \{1,\ldots,N\}$) is associated with a collection (gallery) of his or her M_n photos $\{X_n(m)\}, m \in \{1,\ldots,M_n\}$. It is assumed that the preferences of every nth user over all C categories are known.

In this paper we modify the traditional approach (a feature extractor CNN and a fine-tuned classifier CNN) with additional features [5]. As we pay special attention to offline recognition on mobile devices, it is reasonable to use such CNNs as MobileNet v1/v2 [24]. Namely, the mth photo of the nth user $X_n(m)$ is represented by:

1. D-dimensional vector $\mathbf{f}_n(m) = [f_{n;1}(m),\ldots,f_{n;D}(m)]$ of the off-the-shelf CNN *features (embeddings)* extracted at the penultimate layer of the CNN trained for scene recognition;
2. *scores* (predictions at the last layer/estimates of scene posterior probability) $\mathbf{p}_n(m) = [p_{n;1}(m),\ldots,p_{n;S}(m)]$ from the last layer of the same CNN ($\sum\limits_{s=1}^{S} p_{n;s}(m) = 1$).

As complex photos of events and scenes are characterized by high variability, they are usually composed of parts, some of those parts can be named and correspond to objects [7]. Moreover, many photos from the one interest category contains identical objects (e.g., ball in the football), which can be detected by contemporary methods [34], such as SSDLite or Faster R-CNN. These methods detect the positions of several objects in the input image and predict the scores of each class from the predefined set of $O > 1$ types. We completely ignore bounding boxes and propose to extract only the sparse vector of confidences $\mathbf{o}_n(m) = [o_{n;1}(m),\ldots,o_{n;O}(m)]$. If there are several objects of the same type, the maximal score is stored in this feature vector [5].

During the recognition process, these three vectors may be combined into a single $(D+S+O)$-dimensional representation, or all of them are classified independently and then the classification results are combined using the simple voting. In this paper, we use the latter approach and implemented the classifier fusion technique, which consists of the features from the scene recognition model, scores from the fine-tuned CNN and predictions of the object detection model trained on large dataset. The outputs of individual classifiers are combined with soft aggregation. For example, the decision is taken in favor of the class with the highest weighted sum of outputs of individual classifier. The weights can be chosen using special validation subset (Algorithm 2). It is important to note that this algorithm does not have hyper-parameters which should be tuned. A new image is classified using steps 7–9, 15–16 of this algorithm, in which w_f, w_s, w_o are replaced by the best weights w_f^*, w_s^*, w_o^*.

3.3. Recognition of a set of photos

In this subsection we describe how to use representation of images from previous Subsection for classification of a *set* of photos $X(m), m \in \{1,\ldots,M\}$ of the nth user. For simplicity, we represent the mth photo of the nth user as a K-dimensional feature vector $\mathbf{x}_n(m)$, which can be either any of the above-mentioned features ($K \in \{D,S,O\}$) or their combination ($K = D+S+O$). Similarly, the mth photo $X(m)$ of the input user is represented with the K-dimensional feature vector $\mathbf{x}(m)$. At the second stage, his or her final descriptor $\mathbf{x}$ is produced as a weighted sum of features $\mathbf{x}(m)$, where the weights $w(\mathbf{x}(m))$ may depend on these features [13]. If there is no training data ($N = 0$), then the equal weights will be used, so that conventional averaging with computation of mean feature vector is implemented. However, in this paper we propose to learn the weights $w(\mathbf{x}(m))$ by using a modification of an attention mechanism with a learnable K-dimensional vector of weights $\mathbf{q}$ [14]. Its authors sequentially combined two attention blocks so that the first aggregated vector is fed into fully-connected layer with tanh activation and matrix of weights $W \in \mathbb{R}^{K\times K}, \mathbf{b} \in \mathbb{R}^{K\times 1}$ in order to compute K-dimensional vector of attention weights $\mathbf{q}^1$.

Algorithm 2 Training of classifiers based on proposed image representation.

```
Input: training and validation sets for $C$ classes
Output: ensemble of classifiers
for each training image do
    Feed the image into a scene CNN and compute the embeddings $\mathbf{f} := [f_1, \ldots, f_D]$ and scores $\mathbf{p} := [p_1, \ldots, p_S]$ at the outputs of penultimate and last layers
    Feed the image into an object detector and extract vector $\mathbf{o} := [o_1, \ldots, o_O]$ of maximal confidences for each type of object
end for
Train classifiers $\mathcal{C}_f, \mathcal{C}_p, \mathcal{C}_o$ using training sets of embeddings, scene scores and detector confidences, respectively
for each validation image do
    Extract scene embeddings $\mathbf{f}$ and predict $C$-dimensional confidence scores $\mathbf{cs}_f$ using classifier $\mathcal{C}_f$
    Extract scene scores $\mathbf{p}$ and predict $C$-dimensional confidences $\mathbf{cs}_p$ using $\mathcal{C}_p$
    Compute maximal confidences of each detected object $\mathbf{o}$ and predict $C$-dimensional confidences $\mathbf{cs}_o$ using $\mathcal{C}_o$
end for
Assign $\alpha^* := 0$
for all possible weights $w_f, w_s$ do
    Assign $w_o := 1 - (w_f + w_s)$
    for each validation image do
        Compute confidences $[cs_1, \ldots, cs_C] := w_f \mathbf{cs}_f + w_p \mathbf{cs}_p + w_p \mathbf{cs}_o$
        Obtain class with the maximal confidence $c^* := \operatorname{argmax}_{c \in \{1,\ldots,C\}} cs_c$
    end for
    Compute accuracy $\alpha$ using predictions $c^*$ of all validation images
    if $\alpha^* < \alpha$ then
        Assign $\alpha^* := \alpha, w_f^* := w_f, w_s^* := w_s, w_o^* := w_o$
    end if
end for
return classifiers $\mathcal{C}_f, \mathcal{C}_p, \mathcal{C}_o$ and their weights $w_f^*, w_s^*, w_o^*$
```

After that the second attention block is used to aggregate the input features. Its usage leads to $K(K+1)$ additional parameters and slower decision-making. In order to improve the run-time complexity, we proposed to reduce the dimensionality of the visual features $\mathbf{x}_n(m)$ by learning the matrix $W_s \in \mathbb{R}^{K \times \tilde{K}}$ [13]:

$$\mathbf{s}(m) = W_s \mathbf{x}(m), \tag{1}$$

where $\tilde{K} < K$. The final $\tilde{K}$-dimensional descriptor is computed as follows

$$\mathbf{x} = \sum_{m=1}^{M} w(\mathbf{x}(m)) \mathbf{s}(m), \tag{2}$$

where the weights

$$w(\mathbf{x}(m)) = \frac{\exp(\mathbf{q}^T \mathbf{s}(m))}{\sum_{j=1}^{M} \exp(\mathbf{q}^T \mathbf{s}(j))} \tag{3}$$

are estimated using the squeezed features (1) and an attention mechanism [13,14]. As a result, the proposed approach (1)–(3) introduces only $(K+1) \cdot \tilde{K}$ parameters to the traditional computation of average features. In such case, we obtain very fast classifier based only on features from scene recognition and object detection models.

Finally, a fully-connected (FC) layer with C multi-label classifiers (logistic regressions) is added to the outputs of attention block (1) and the entire model including the $\tilde{K}$-dimensional vector $\mathbf{q}$ is learned in end-to-end fashion to predict distribution $\{\pi_c | c \in \{1, \ldots, C\}\}$ using the available training set. The profile of a user is predicted for the feature vector $\mathbf{x}$ and top-k categories with the highest scores are used to make relevant recommendations.

3.4. Mobile application

We implemented the entire pipeline (Fig. 1) in the Android demo application[1] (Fig. 2), which sequentially processes all photos from the gallery in the background thread. The source code of this application is publicly available.[2] An example of the main user window is shown in Fig. 2a. It is possible to tap any bar in this histogram to show a new form with a histogram of detailed categories (Fig. 2b). We support two special high-level categories: “Locations” with most popular cities obtained from geo tags and “Demography” with the stacked histograms (Fig. 2c) of age and gender of the closed persons. If a concrete category is tapped, a “display” form appears, which contains a list of all photos from the gallery with this category (Fig. 3a). Tapping the bar in the Demography page (Fig. 2c) will display the list of all photos of particular subject (Fig. 3b).

The gathered profile for each high-level category (Fig. 2b) can be used for recommending locations of shops, restaurants and other places near his or her current geographic location. We added Google maps/Google places and Amazon search functionality so that a request is made to the Google Places API for each identified category and all of the results are consolidated into a single list of store names and markers are placed on an interactive map for each identified store (Fig. 4). Amazon search for now just opens a search page on the Amazon website with a query for all found categories.

4. Experimental study

4.1. Scene recognition for user modeling

In this subsection we explore several known ways to implement fast object detection and scene recognition models used in the proposed pipeline (Fig. 1). The best scene recognition model was chosen using the above-mentioned subset of the united Places2 dataset. The scene dataset was split into the training and test subsets with 8M and 40K images, respectively. This dataset was used to train several CNNs including MobileNet v1/v2 [24], Inception v3 [35] and EfficientNet-B3 [18]. We applied *structural pruning* methods to speed up offline classification on mobile side. It was found in the preliminary experiments that the best convergence is demonstrated by the pruning with the Taylor expansion [22]. Hence, this method was used to prune approximately 25% and 40% of all channels in each convolutional layer. The recognition results are presented in Table 1. In addition, inference time (Table 2) was measured on MacBook and two Samsung devices (Galaxy Tab S4 tablet PC and Galaxy S9+ phone). The best values in both tables are marked in bold.

Though EfficientNet-B3 model is the most accurate one, its implementation on mobile platforms can be inappropriate due to the large running time. Hence, this model is an ideal candidate for the server-side scene recognition only (Fig. 1). The usage of simplified MobileNet model ($\alpha = 1.0$) caused better performance with a very low increase of the error rate. Our implementation of structural pruning improves both running time and memory space, so we decided to use the 25% pruned MobileNet v2 ($\alpha = 1.0$) for scene classification on mobile devices.

[1] https://drive.google.com/drive/folders/1rQkJZifq_89pu0sT_UnYXziuxutTpNEN

[2] https://github.com/HSE-asavchenko/mobile-visual-preferences

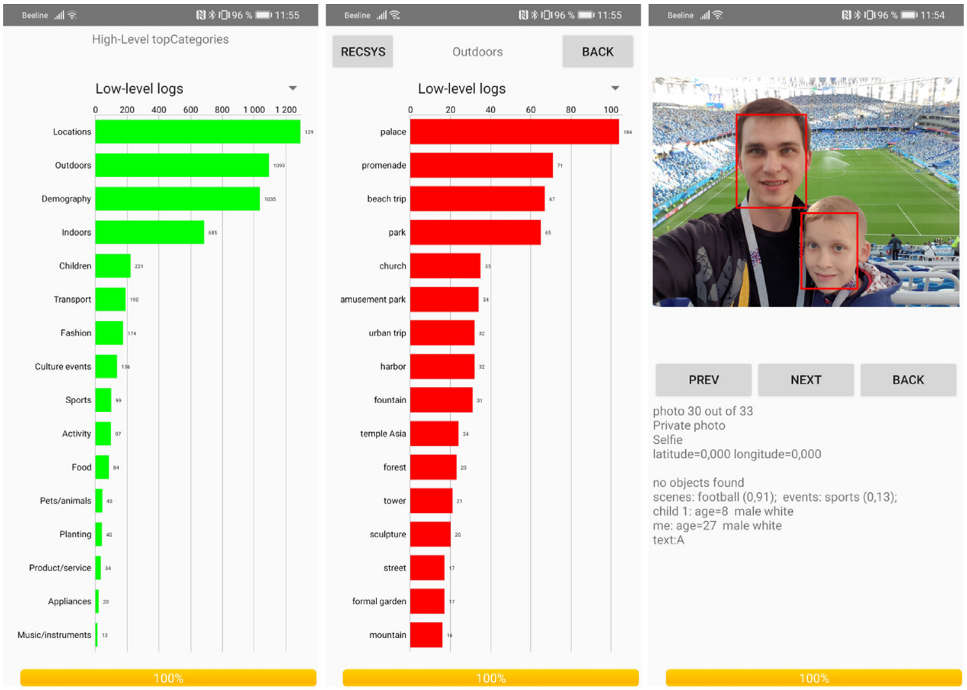


photo 30 out of 33
Private photo
Selfie
latitude=0,000 longitude=0,000

no objects found
scenes: football (0,91); events: sports (0,13);
child 1: age=8 male white
me: age=27 male white
text:A

(a) (b) (c)

Fig. 2. User's profile in the mobile demo GUI.

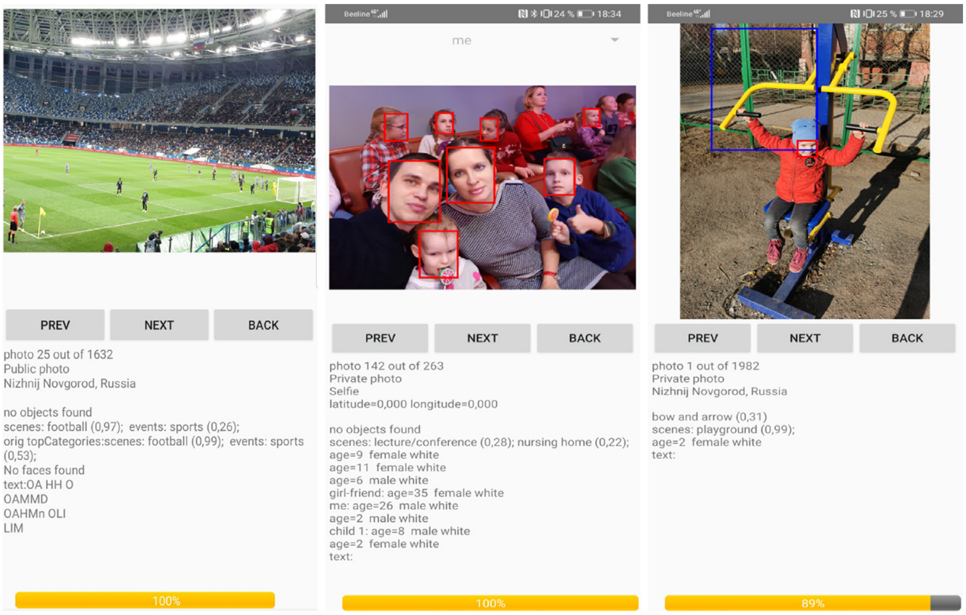


photo 25 out of 1632
Public photo
Nizhnij Novgorod, Russia

no objects found
scenes: football (0,97); events: sports (0,26);
orig topCategories:scenes: football (0,99); events: sports (0,53);
No faces found
text:OA HH O
OAMMD
OAHMn OLI
LIM

photo 142 out of 263
Private photo
Selfie
latitude=0,000 longitude=0,000

no objects found
scenes: lecture/conference (0,28); nursing home (0,22);
age=9 female white
age=11 female white
age=6 male white
girl-friend: age=35 female white
me: age=26 male white
age=2 male white
child 1: age=8 male white
age=2 female white
text:

photo 1 out of 1982
Private photo
Nizhnij Novgorod, Russia

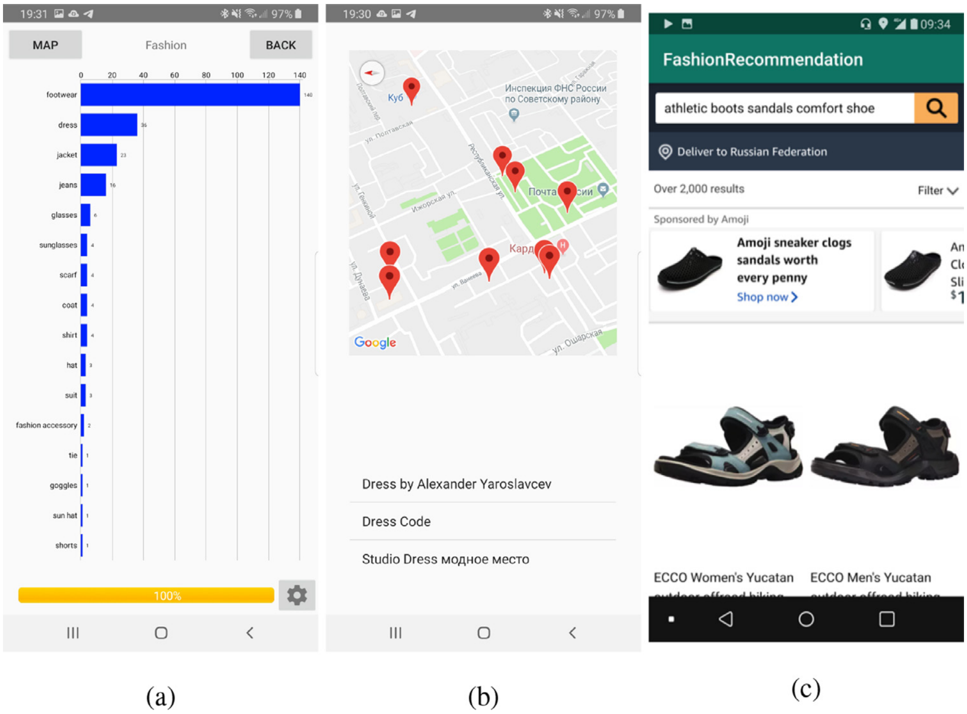
bow and arrow (0,31)
scenes: playground (0,99);
age=2 female white
text:

(a) (b) (c)

Fig. 3. Detailed photo analysis in the mobile demo GUI.

Table 1
Performance analysis of scene recognition models, subset of Places2 dataset.

CNN	Pruning	Top-1 accuracy, %	Top-5 accuracy, %	Precision, %	Recall, %	Size, Mb
MobileNet v2	Original	50.7	80.4	57.5	46.7	11.1
($\alpha = 1.0$)	Pruning (25%)	49.8	79.8	56.2	46.2	8.3
	Pruning (40%)	48.7	79.0	54.9	45.4	**6.7**
MobileNet v2	Original	51.3	80.7	58.0	47.1	20.3
($\alpha = 1.4$)	Pruning (40%)	49.5	79.3	56.1	45.6	12.2
Inception v3	Original	53.5	83.0	60.7	48.7	91.1
EfficientNet-B3	Original	**55.2**	**83.9**	**62.6**	**49.1**	44.7



(a) (b) (c)

Fig. 4. Fashion recommendation: (a) fashion profile; (b) nearby shops; (c) search.

Table 2
Inference time (ms) of scene recognition models.

CNN	Pruning	MacBook Pro 2015	Galaxy Tab S4	Galaxy S9+
MobileNet v2	Original	18	95	80
($\alpha = 1.0$)	Pruning (25%)	14	80	68
	Pruning (40%)	**12**	**70**	**60**
MobileNet v2	Original	31	165	135
($\alpha = 1.4$)	Pruning (40%)	29	140	110
Inception v3	Original	99	440	350
EfficientNet-B3	Original	151	610	480

Table 3
Results of object detection testing.

Detector	CNN	Model size, MB	Inference time, ms.	Recall, %	mAP, %
SSDLite-500	MobileNet v2	**23**	**35**	16.6	52.5
RetinaNet	ResNet-50	51	432	32.9	69.2
Faster	Inception v3	54	223	41.4	59.3
R-	ResNet-50	116	318	35.0	**63.6**
CNN	ResNet-101	189	847	44.8	53.4
	InceptionResNet	251	6570	**48.5**	61.8

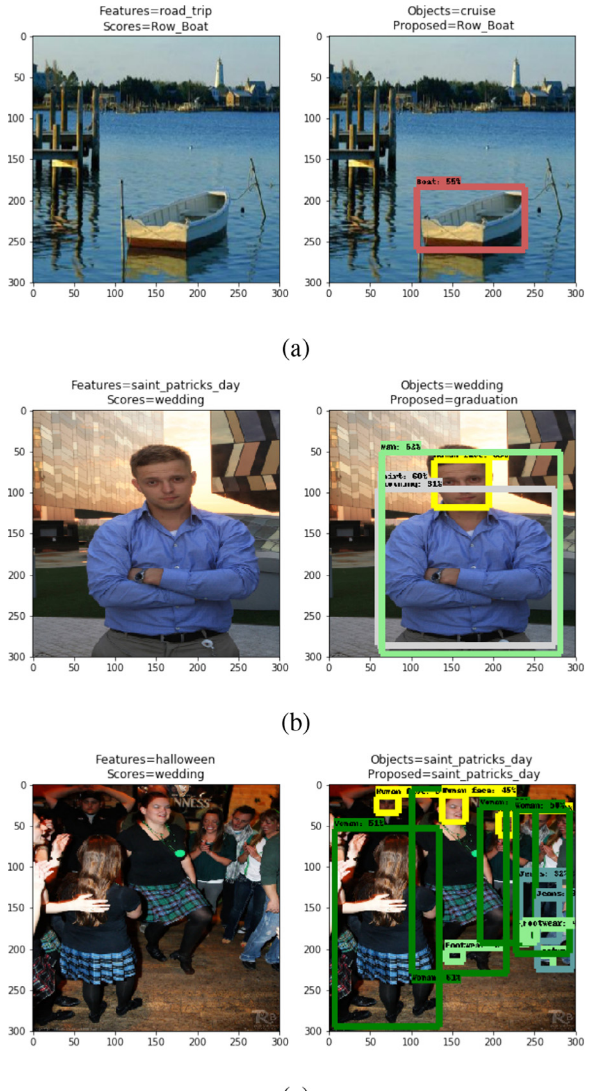
4.2. Object detection for user modeling

In this subsection we compared existing detectors using the training/testing dataset described in Section 3. By using the balanced training set with no more than 5000 images per category [9], we trained such detectors as Faster R-CNN and SSDLite [24] using Tensorflow Object Detection API. The testing set contains 5000 images for each category. Some of the categories are considered as a family of similar categories, such as “animal” and “cat” or “dog” categories or “building” and “skyscraper” [9]. We took such categories into account while estimating recall (an average rate of detected objects from one class) and mAP (average rate of correctly detected object for particular category to all detected objects of this category).

The results including the model size and inference time for one image on MacBook Pro 2015 laptop are shown in Table 3. They are similar to existing studies of object detectors [8,34]. The highest mAP/recall are obtained by Faster R-CNN with InceptionResNet and ResNet-101 backbones, which are the best candidates for the server-side processing in our pipeline (Fig. 1). The choice for the client side is more difficult, because the SSDLite models are rather fast (200–300 ms per photo on Samsung S9+ mobile phone), but their recall is too low due to the small size of most objects from our dataset. The Faster R-CNN is too slow for offline detection on mobile devices (1.2 sec per image on Samsung S9+ for Inception v2 backbone and more than 30 sec per photo for InceptionResNet).

4.3. Event recognition

In this subsection we examine the representation of visual data (Section 3.2) with scores and embeddings from scene recognition model and score from object detection model in event recognition task. Two datasets were used, namely, 1) PEC (Photo Event Collection) [15] with 61,364 images from 807 collections of 14 social



(a)

(b)

(c)

Fig. 5. Sample results of event recognition.

Table 4
Event recognition accuracy, %.

Features	Classifier	PEC	WIDER
MobileNet, scores	Random Forest	55.76	40.18
	Linear SVM	51.66	36.29
	Fine-tuned	61.11	40.49
MobileNet, embeddings	Random Forest	57.25	42.32
	Linear SVM	59.72	48.31
	Fine-tuned	62.13	49.48
SSD + MobileNet	Random Forest	30.84	14.69
	Linear SVM	42.18	19.91
	Fine-tuned (new FC layer)	40.16	12.91
Proposed representation (client-side classifiers)	Random Forest	57.60	43.55
	Linear SVM	60.92	48.91
	Fine-tuned	63.34	49.80
EfficientNet-B3, scores	Random Forest	58.37	39.50
	Linear SVM	52.56	29.58
	Fine-tuned	63.44	45.59
EfficientNet-B3, embeddings	Random Forest	58.97	43.11
	Linear SVM	62.66	51.62
	Fine-tuned	63.58	53.31
Faster R-CNN + InceptionResNet	Random Forest	43.32	27.23
	Linear SVM	48.83	28.66
	Fine-tuned (new FC layer)	47.45	21.27
Proposed representation (server-side classifiers)	Random Forest	59.64	45.28
	Linear SVM	63.82	52.61
	Fine-tuned	66.26	54.61
Complete pipeline	Random Forest	59.06	45.01
	Linear SVM	62.72	52.26
	Fine-tuned	65.69	54.04

event classes (birthday, wedding, graduation, etc.); and 2) WIDER (Web Image Dataset for Event Recognition) [16] with 50,574 images and 61 event categories (parade, dancing, meeting, press conference, etc).

We used standard train/test split for both datasets proposed by their creators. We directly assigned the collection-level label to each image contained in the PEC and simply use the image itself for event recognition, without any metadata such as temporal information similarly to recent paper [12]. The results of the client-side models (MobileNet scene classifier and SSDLite object detector) and server-side models (EfficientNet-B3 scene classifier and Faster R-CNN detector with InceptionResNet backbone) are shown in Table 4. In addition to client-side and server-side models, we used the entire pipeline (Fig. 1) here. It automatically selects private photos and recognizes them with the client-side models. All other photos are processed by the server-side models.

If the fine-tuned CNNs are used in an ensemble, the previous state-of-the-art for the PEC was improved from 62.2% [12] to 63.34% even for the lightweight MobileNet-based models. Our server-side model is even better (accuracy 66.26%). Moreover, we achieve an excellent performance by using linear SVM for features and scores extracted by pre-trained models from previous subsections. For example, if only pre-trained scene recognition and object detection models are used in the proposed approach, the resulted accuracy for the WIDER is only 0.17-0.89% lower for lightweight architectures. As a result, additional inference in fine-tuned models is not required, so that it is possible to classify events by using the output of user preferences prediction engine (Fig. 1). The accuracy of our MobileNet-based classifiers for the WIDER is 7–12% higher when compared to the best results (42.4%) from original paper [16]. Moreover, the modern EfficientNet model has 1.6% higher accuracy the previously known state-of-the-art results (53%) [12]. In contrast to the PEC dataset, the WIDER does not contain many facial images of the same subjects and scanned sensitive documents. As a result, most images are considered to be public and the overall accuracy of the complete pipeline is very close to the server-side models.

4.4. Recognition of a set of images based on a user's profile

In the next experiment the proposed aggregation of image features into a single user descriptor (1)-(3) for our representation of images is studied [13]. As there is no publicly available labeled datasets of the photo galleries for different users, we used the Amazon Fashion [36] dataset that contains 500,000 entries of $N = 16,000$ unique users interacting with 40,000 products from $C = 75$ fashion categories. There is a single unique item on each picture that belongs to one or more classes. The number M_n of items purchased by a user varies from 5 to 40; an average user has interacted with 8 unique items. The images are grouped by user, and each user is associated with a C-dimensional target vector, so that the cth component is equal to 1 if the user has interacted with at least one item from the cth category, and 0, otherwise. The united feature vector obtained by merging the scores and embeddings of the scene recognition MobileNet model with the output of SSDLite object detector was used.

We implemented the following aggregation techniques: 1) average pooling of fine-tuned features; 2) pooling of all features with one attention block (2) and additional context gating (CG), which applies a scaling mask to the resulting aggregated vector [37]; 3) proposed usage of 1-layer attention with reduced ($\tilde{K} = 128$) fea-

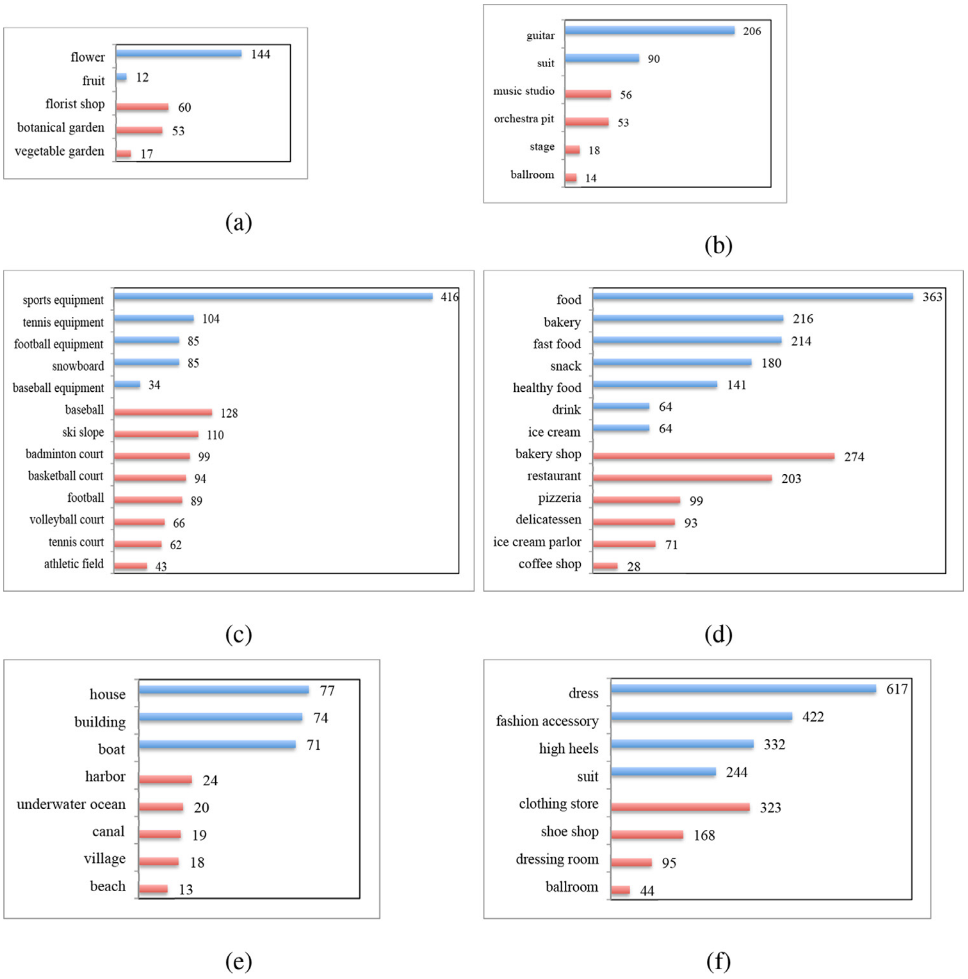
Table 5
Comparison of aggregation techniques, Amazon Fashion dataset.

Metric	Average	1-Attention + CG	Proposed 1-Attention (1)-(3)	Proposed 2-Layers Attention + FC
Size, MB	**1.3**	4.5	2.18	3.5
Decision time, ms	**40**	43	41	47
Precision@k=5	0.29	0.44	**0.51**	0.50
Recall@k=5	0.56	0.61	0.66	**0.68**
F1-score@k=5	0.38	0.51	**0.58**	**0.58**
AUC@k=5	0.50	0.60	**0.62**	0.61
Precision@k=10	0.18	0.43	**0.51**	0.50
Recall@k=10	0.70	0.63	0.67	**0.68**
F1-score@k=10	0.29	0.51	**0.58**	**0.58**
AUC@k=10	0.52	0.51	**0.71**	**0.71**

tures (1)–(3); and 4) proposed usage of reduced ($\tilde{K} = 128$) features with two attention blocks recommended in the original paper [14]. The resulted descriptor were fed into FC layer in order to obtain the final profile as described in the last paragraph of Section 3.3. The aggregation models were trained on the 70% of the dataset. The algorithms were tested on the for remaining 30% users in order to predict C interest probabilities, among which top k interests were recommended. The model size (excluding the size of the feature extractor) and the dependence of precision, recall, F1-score and AUC (area under the ROC curve) on the number k of top categories (with the highest probabilities) for each aggregation technique and the best configuration are shown in Table 5.

Here the F1-score of traditional feature aggregation is 13–31% lower when compared to learnable pooling techniques. The proposed approach (1)–(3) with one attention layer is the most appropriate for mobile applications due to the lowest number of parameters and quality metrics higher than almost all other aggregation methods.

In the next experiment event recognition task was examined using the PEC [15]. In contrast to previous Subsection, we focus here on a image-set classification problem, in which a *set* of images from an album is observed [17]. We used conventional split into the training set with 667 albums and testing set with 140 albums. Two techniques were applied to obtain a final descriptor of a set



(a)

(b)

(c)

(d)

(e)

(f)

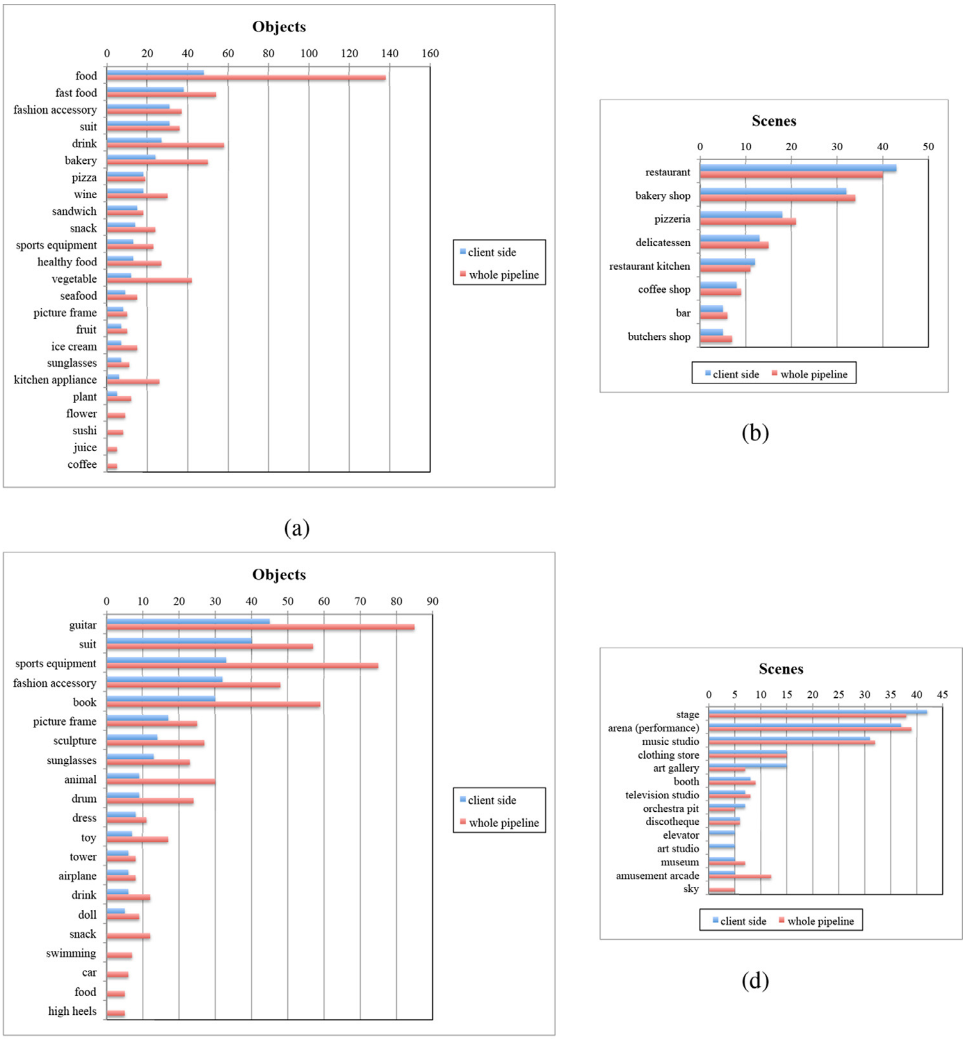
Fig. 6. Profiles gathered for images from the web using the client-side models (blue bars - objects, red bars - scenes): (a) Nature; (b) Music; (c) Sport; (d) Food; (e) Traveling; (f) Fashion. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 6
Event recognition in a set of images (album), PEC dataset.

Features	Aggregation	Accuracy, %
MobileNet v2 ($\alpha = 1.0$)+SSDLite	Average	86.42
	Proposed 1-Attention (1)–(3)	**89.29**
MobileNet v2 ($\alpha = 1.4$)+SSDLite	Average	86.44
	Proposed 1-Attention (1)–(3)	87.36
Inception v3+Faster R-CNN	Average	86.43
	Proposed 1-Attention (1)–(3)	87.86
EfficientNet+Faster R-CNN	Average	88.57
	Proposed 1-Attention (1)–(3)	**89.29**
AlexNet	CNN-LSTM-Iterative [38]	84.5
	Aggregation of representative features [39]	87.9
ResNet-101	CNN-LSTM-Iterative [38]	84.5
	Aggregation of representative features [39]	89.1

of images, namely, 1) simple averaging of features of individual images in a set; and 2) proposed implementation of neural attention mechanism (1)–(3) for L_2-normed features [17]. In the latter case the attention weights are learned using the sets with 10 randomly chosen images from all albums in order to make identical shape of input tensors. As a result, 667 training subsets with 10 images were obtained. The recognition accuracies are presented in Table 6. Here we provided the best-known results for this dataset [38,39].

The attention-based aggregation is 1–3% more accurate when compared to classification of average features. As one can notice, the proposed implementation of attention mechanism achieves the state-of-the-art results, though we used lightweight CNNs (MobileNet and EfficientNet). The most remarkable fact here is that the best results are achieved for the most simple model (MobileNet v2 $\alpha = 1.0$), which can be explained by the lack of training data. What is more important, we do not need to fine-tune existing



(a)

(b)

(c)

(d)

Fig. 7. Profiles gathered for Instagram photos. (a)-(b) Gordon Ramsay (chef); (c)-(d) The Rolling Stones; (a), (c) object detection, (b), (d) scene recognition.

scene recognition models, so the implementation of event recognition in an album will be very fast if the user preferences have already been predicted (Fig. 1).

4.5. Examples

In the last subsection we provide several qualitative examples for the usage of our pipeline. Firstly, the results of correct event recognition using the proposed ensemble (Algorithm 2) are demonstrated in Fig. 5. Here the title of each left image displays the result of event recognition using only features $\mathbf{x}_n$ or scores $\mathbf{p}_n$. Each right image shows the object detection results. Its title contains the event prediction based on object scores and our ensemble. As one can notice, the proposed representation manage to obtain reliable solution even when individual classifiers make wrong decisions.

Secondly, we present several profiles obtained with the proposed approach (Fig. 1) by processing real galleries of photos. At first, we took six keywords and automatically retrieved images from the Web by feeding these keywords into the visual search engine. Fig. 6 contains the scene (red bars) and object (blue bars) categories for top predictions of interests by the lightweight neural networks. Here the set of images gathered by keyword is processed very accurately. Though several detected objects are still too general, such as building/house for traveling and suit for music, the red bar of top scenes (Fig. 6) contains the categories which characterize the keyword perfectly well.

Next, we gathered publicly available subsets of first 500 photos ordered by date of several celebrities and companies from Instagram accounts. At first, we compare the results of the client-side models with the proposed pipeline (Fig. 7). As one expects, the remote processing of public photos increases the number of detected objects. As a result, the red bars in Fig. 7(a),(c) are in most cases larger than the blue bars. However, the entire structure of the gathered profiles is very similar. Thus, if the number of photos in a gallery is not too small, even secure offline processing with lightweight client-side models may be appropriate.

Finally, other results of client-side models are shown in Figs. 8 and 9. They are even more attractive when compared to first examples (Fig. 6). Again, scene recognition characterizes the

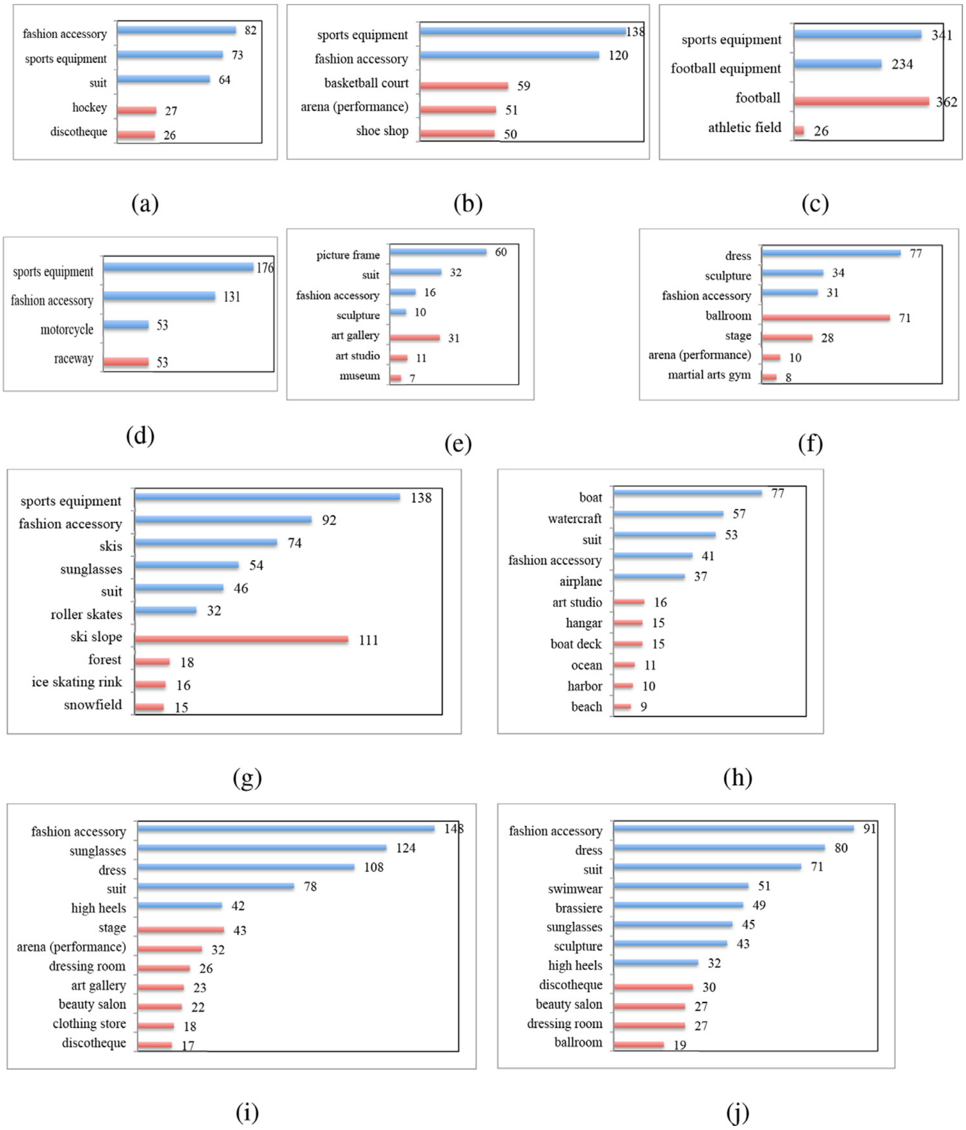


Fig. 8. Profiles gathered for Instagram photos using the client-side models (blue bars - objects, red bars - scenes): (a) Alex Ovechkin (hockey player); (b) LeBron James (basketball player); (c) Leo Messi (football player); (d) Max Verstappen (F1 driver); (e) Kehinde Wiley (artist); (g) Svetlana Zakharova (ballet dancer); (h) Johannes Klaebo (skier); (i) Fedor Konyukhov (traveler); (j) Beyonce (singer); (g) Kim Kardashian (media personality). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

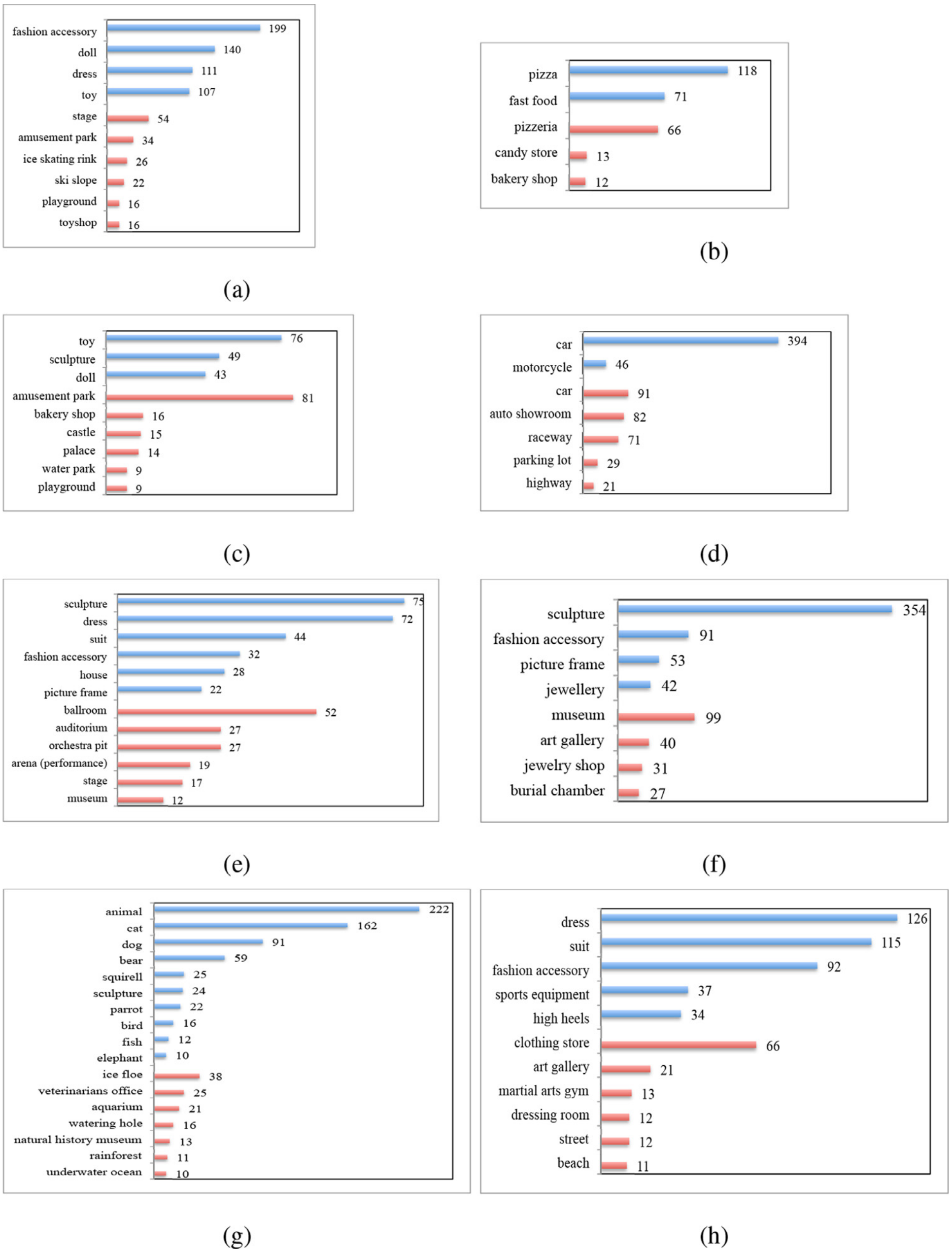


Fig. 9. Profiles gathered for Instagram photos using client-side models (blue bars - objects, red bars - scenes): (a) Ded Moroz (Santa); (b) Mir Pizzy (pizzeria); (c) Disneyland; (d) Ferrari; (e) Bolshoi theater; (f) Louvre; (g) Moscow zoo; (h) Uniqlo. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

subject almost perfectly, and the detected objects contain important information for scene and event processing. It is necessary to highlight that the largest facial cluster obtained in our pipeline for all cases of single person account contains the photo of this person. The second largest clusters contain the main relatives (wife/boyfriend, children, etc.). The face clustering [11] of the Rolling stones' gallery resulted to 4 large clusters of the main musicians of the rock band.

4.6. Discussion

The usage of the proposed pipeline (Fig. 1) in various image recognition tasks leads to the following main results:

1. Developed structurally pruned MobileNet-based scene recognition model has near state-of-the-art accuracy for classification of more than 300 different scenes and requires only 60–85 ms to process an image on mobile device (Table 2).
2. The previous state-of-the-art for PEC was improved from 62.2% [12] even for client-side model (63.34%). Our server-side model is much better (66.26%).
3. The accuracy of our approach for the WIDER dataset is 7–12% higher when compared to the best results (42.4%) from original paper [16]. The results of our server-side model is 1.6% more accurate when compared to the known best result for this dataset [12].
4. The proposed aggregation of visual features in an album improves the state-of-the-art results for the original testing protocol for the PEC dataset [15] by using the client-side models (Table 6).

Our approach overcomes some key privacy issues that are implicit in computations performed on remote servers [3]. Despite typical applications of private (sensitive) image detection [32], our server-side processing does not need to store the images in any database as it immediately removes an input photo right after processing in “Accurate scene classification” and “Accurate object detection” units. Thus, it is vulnerable only to network sniffing attacks. In practical applications it is important to add traffic encryption and/or implement a server as a part of cloud storages (Samsung Cloud, Apple iCloud, etc.). However, we should emphasize that any above-mentioned heuristic (detection of faces and sensitive words) may fail in some specific cases because it is impossible to *automatically* detect *all* private images with 100% true positive rate and high true negative rate. Hence, in order to improve the generalizability of the proposed method, we provide several opportunities for a user. If the above-mentioned guarantees with the prohibition of storage of all photos in an input server and our privacy detection engine are appropriate, the entire pipeline (Fig. 1) is used. However, if the privacy is extremely important for a particular user, we let him to explicitly prevent privacy detection. In such case all photos are marked as private in order to process them in a mobile device in offline mode. This option is chosen by default. If the user verifies that our engine predicts private/public status correctly, he could turn on the option of remote processing of public photos.

Certainly, though the client-side models let us obtain very accurate profile for many examples of Instagram accounts (Figs. 7, 8 and 9), predicted profile will be less accurate when compared to the entire pipeline. However, our experimental study demonstrated that the scene recognition results of MobileNet scores and features for both event datasets from Section 4.3 are only 0.2–3% less accurate when compared to EfficientNet model. In this case there is no practical need for scene classification on remote server. However, the situation is different for object detection task due to the limitations of SSD-based models to detect small objects (food, pets, fashion accessories, etc.). Indeed, though the processing in a client mode is better than the random guess with accuracy $100\%/14 \approx 7.14\%$ for PEC and $100\%/61 \approx 1.64\%$ for WIDER, the accuracy of SSDLite are much (6-13%) lower than the accuracy of Faster R-CNN. In this particular example, the proposed representations hides this disadvantage and the best ensemble for the client-side models is not too worth than the server-side models (60.92% vs 63.82% for PEC and 48.91% vs 52.61% for WIDER). However, in general, the user’s profile gathered with Faster R-CNN detectors has much more objects so that the remote processing of public photos is worth implementing.

5. Conclusion

In this paper, we have proposed the novel pipeline (Fig. 1) for recognition of user’s preferences for hobbies and lifestyle in visual data based on the representation of the photos with scores and/or embeddings of scene classifier and outputs of object detectors. It was demonstrated how to efficiently combine the same features of each photo in a given input set into a single descriptor of particular user (1)–(3) based on the learnable pooling used previously only for video recognition [14,37]. We achieved the state-of-the-art results for Photo Event Collection [15] even by using the client-side models (Tables 4, 6).

Our engine was implemented in the publicly available Android application (Figs. 2, 3). It is applicable for various personalized mobile services such as target advertisements, marketing and recommender systems. For example, users can get better personalization while traveling. Depending on the user’s profile, it is possible to recommend suitable products, shops, content. The cold-start problem could be alleviated if the companies can get access to reliable preference information without the need of rating elicitation of monitoring users closely over time.

One of the main limitations of the proposed Algorithm 2 is the need to perform rather slow object detection. If only scenes/events are required in a user’s profile, preliminarily object detection can significantly increase the processing time. Moreover, there is a possibility of having some images to be misclassified as public and sent to the remote server, which may be undesirable for very secure applications even if it is guaranteed that the channel is encrypted and the images will be removed there right after scene classification and object detection. Though a user can turn off the remote processing completely, it may reduce the total number of detected objects in his or her profile (Fig. 7).

In future it is important to improve the accuracy of private photo detection [3,32] by gathering large datasets and training classifiers with manageable false positive rate. Secondly, it is necessary to apply the transfer learning techniques from [12] to our representations in order to increase the accuracy for the WIDER dataset. Finally, it is desirable to extend the number of categories by estimating concrete characteristics of extracted objects, such as pet breeds, car models, sport teams, logos [10], etc.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research is based on the work supported by Samsung Research, Samsung Electronics. The work of A.V. Savchenko is supported by the National Research University Higher School of Economics (HSE University) within the framework of Basic Research Program.

References

[1] X. Yu, F. Jiang, J. Du, D. Gong, A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains, Pattern Recognit. 94 (2019) 96–109.
[2] Y. Deldjoo, M. Schedl, P. Cremonesi, G. Pasi, Recommender systems leveraging multimedia content, ACM Comput. Surv. 53 (5) (2020) 1–38.
[3] G. Yang, J. Cao, Z. Chen, J. Guo, J. Li, Graph-based neural networks for explainable image privacy inference, Pattern Recognit. 105 (2020) 107360.
[4] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT press, 2016.
[5] A.V. Savchenko, User preference prediction in visual data on mobile devices, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–7.
[6] L. Xie, F. Lee, L. Liu, K. Kotani, Q. Chen, Scene recognition: a comprehensive survey, Pattern Recognit. 102 (2020) 107205.
[7] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: a 10 million image database for scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (6) (2018) 1452–1464.
[8] T. Chu, Q. Luo, J. Yang, X. Huang, Mixed-precision quantized neural networks with progressively decreasing bitwidth, Pattern Recognit. 111 (2021) 107647.
[9] I. Grechikhin, A.V. Savchenko, User modeling on mobile device based on facial clustering and object detection in photos and videos, in: Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), Springer, 2019, pp. 429–440.
[10] H. Su, S. Gong, X. Zhu, Scalable logo detection by self co-learning, Pattern Recognit. 97 (2020) 107003.
[11] A.V. Savchenko, Efficient facial representations for age, gender and identity recognition in organizing photo albums using multi-output ConvNet, PeerJ Comput. Sci. 5 (e197) (2019), doi:10.7717/peerj-cs.197.
[12] L. Wang, Z. Wang, Y. Qiao, L. Van Gool, Transferring deep object and scene representations for event recognition in still images, Int. J. Comput. Vis. 126 (2–4) (2018) 390–409.
[13] A.V. Savchenko, K.V. Demochkin, L.V. Savchenko, Neural attention mechanism and linear squeezing of descriptors in image classification for visual recommender systems, Opt. Memory Neural Netw. 29 (4) (2020) 297–304.
[14] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, G. Hua, Neural aggregation network for video face recognition, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 5216–5225.
[15] L. Bossard, M. Guillaumin, L. Van Gool, Event recognition in photo collections with a stopwatch HMM, in: Proceedings of the International Conference on Computer Vision (ICCV), IEEE, 2013, pp. 1193–1200.

[16] Y. Xiong, K. Zhu, D. Lin, X. Tang, Recognize complex events from static images by fusing deep channels, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1600–1609.

[17] A.V. Savchenko, Event recognition with automatic album detection based on sequential grouping of confidence scores and neural attention, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–8.

[18] M. Tan, Q. Le, EfficientNet: rethinking model scaling for convolutional neural networks, in: Proceedings of the International Conference on Machine Learning (ICML), 2019, pp. 6105–6114.

[19] A.V. Savchenko, Maximum-likelihood approximate nearest neighbor method in real-time image recognition, Pattern Recognit. 61 (2017) 459–469.

[20] A.V. Savchenko, Fast inference in convolutional neural networks based on sequential three-way decisions, Inf. Sci. 560 (2021) 370–385.

[21] D. Mittal, S. Bhardwaj, M.M. Khapra, B. Ravindran, Recovering from random pruning: on the plasticity of deep convolutional neural networks, in: Proceedings of Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 848–857.

[22] P. Molchanov, S. Tyree, T. Karras, T. Aila, J. Kautz, Pruning convolutional neural networks for resource efficient inference, in: Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[23] R. Rothe, R. Timofte, L. Van Gool, DLDR: deep linear discriminative retrieval for cultural event classification from a single image, in: Proceedings of the International Conference on Computer Vision Workshops (ICCVW), 2015, pp. 53–60.

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobilenetV2: inverted residuals and linear bottlenecks, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510–4520.

[25] Q. You, S. Bhatia, J. Luo, A picture tells a thousand words about you! User interest profiling from user generated visual content, Signal Process. 124 (2016) 45–53.

[26] B. Wu, X. He, Y. Chen, L. Nie, K. Zheng, Y. Ye, Modeling product's visual and functional characteristics for recommender systems, IEEE Trans. Knowl. Data Eng. (2020). 1–1

[27] E. Andreeva, D.I. Ignatov, A. Grachev, A.V. Savchenko, Extraction of visual features for recommendation of products via deep learning, in: Proceedings of International Conference on Analysis of Images, Social Networks and Texts (AIST), Springer, 2018, pp. 201–210.

[28] V. Dominguez, P. Messina, I. Donoso-Guzmán, D. Parra, The effect of explanations and algorithmic accuracy on visual recommender systems of artistic images, in: Proceedings of the 24th International Conference on Intelligent User Interfaces, 2019, pp. 408–416.

[29] A. Zhai, H.-Y. Wu, E. Tzeng, D.H. Park, C. Rosenberg, Learning a unified embedding for visual search at Pinterest, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2412–2420.

[30] A. Pal, C. Eksombatchai, Y. Zhou, B. Zhao, C. Rosenberg, J. Leskovec, PinnerSage: multi-modal user embedding framework for recommendations at PInterest, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 2311–2320.

[31] L. Kopeykina, A. Savchenko, Photo privacy detection based on text classification and face clustering, in: Proceedings of the VI International Conference Information Technology and Nanotechnology. Image Processing and Earth Remote Sensing (ITNT-IPERS 2020), CEUR-WS, vol. 2665, 2020, pp. 171–176.

[32] L. Tran, D. Kong, H. Jin, J. Liu, Privacy-CNH: a framework to detect photo privacy with convolutional neural network using hierarchical features, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2016.

[33] A.V. Savchenko, Efficient statistical face recognition using trigonometric series and CNN features, in: Proceedings of the 24th International Conference on Pattern Recognition (ICPR), IEEE, 2018, pp. 3262–3267.

[34] L. Zhu, Z. Xie, L. Liu, B. Tao, W. Tao, IOU-uniform R-CNN: breaking through the limitations of RPN, Pattern Recognit. 112 (2021) 107816.

[35] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, Inception-ResNet and the impact of residual connections on learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 4278–4284.

[36] W.-C. Kang, C. Fang, Z. Wang, J. McAuley, Visually-aware fashion recommendation and design with generative image models, in: Proceedings of International Conference on Data Mining (ICDM), IEEE, 2017, pp. 207–216.

[37] A. Miech, I. Laptev, J. Sivic, Learnable pooling with context gating for video classification, arXiv:1706.06905(2017).

[38] Y. Wang, Z. Lin, X. Shen, R. Mech, G. Miller, G.W. Cottrell, Recognizing and curating photo albums via event-specific image importance, in: Proceedings of British Conference on Machine Vision (BMVC), 2017.

[39] Z. Wu, Y. Huang, L. Wang, Learning representative deep features for image set analysis, IEEE Trans. Multimedia 17 (11) (2015) 1960–1968.

Andrey V. Savchenko received Doctor of Science degree in System analysis, control and information processing (2016) in Nizhniy Novgorod State Technical University and PhD in Mathematical Modeling (2010) in Higher School of Economics, Moscow. Currently, he is a full professor and leading research fellow of Laboratory of Algorithms and Technologies for Network Analysis in HSE University, Nizhny Novgorod, Russia.

Kirill V. Demochkin received B.Sc. degree at the National Research University Higher School of Economics in 2019 and M.Sc. degree at Skolkovo Institute of Science and Technology (Skoltech) in 2021. In 2018–2019, he worked as a junior researcher in Samsung-PDMI Joint AI Center, St. Petersburg Department of Steklov Institute of Mathematics. He is currently a researcher at Samsung AI Center, Moscow, Russia.

Ivan S. Grechikhin received M.Sc. degree at the National Research University Higher School of Economics in 2016. He is a researcher of Samsung-PDMI Joint AI Center in St. Petersburg Department of Steklov Institute of Mathematics and a research assistant of Laboratory of Algorithms and Technologies for Network Analysis at HSE University, Nizhny Novgorod.