Event Recognition with Automatic Album Detection based on Sequential Grouping of Confidence Scores and Neural Attention

Andrey Savchenko*†
*Samsung-PDMI Joint AI Center
St. Petersburg Department of Steklov Institute of Mathematics
†Laboratory of Algorithms and Technologies for Network Analysis
National Research University Higher School of Economics
Nizhny Novgorod, Russia
Email: avsavchenko@hse.ru

***Abstract*—In this paper a new formulation of event recognition task is examined: it is required to predict event categories given a gallery of images, for which albums (groups of photos corresponding to a single event) are unknown. The novel two-stage approach is proposed. At first, features are extracted in each photo using the pre-trained convolutional neural network (CNN). These features are classified individually. The normalized scores of the classifier are used to group sequential photos into several clusters. Finally, the features of photos in each group are aggregated into a single descriptor using neural attention mechanism. This algorithm is implemented in Android mobile application. Experimental study with features extracted by contemporary convolutional neural networks including EfficientNets for Photo Event Collection and Multi-Label Curation of Flickr Events Dataset demonstrates that the proposed approach is 9-23% more accurate than conventional event recognition on single photos. Moreover, proposed method has 13-16% lower error rate when compared to classification of groups of photos obtained with hierarchical clustering of CNN-based embeddings.**

***Index Terms*—image recognition, convolutional neural network, event recognition, attention network**

I. Introduction

People are taking more photos than ever before in recent years [1] due to the rapid growth of social networks, cloud services and mobile technologies. To organize a personal collection, the photos are usually assigned to albums according to some events. The photo organizing systems (Apple and Google Photos, etc.) allow the user to rapidly search for required photo, and also to increase the efficiency of work with a gallery [2]. Nowadays, these systems usually include content-based image analysis and automatic association of each photo with different tags (scene description, persons, objects, locations, etc.). Such analysis can be used not only to selectively retrieve photos for particular tag in order to keep nice memories of some episodes of user's live [3], but to make personalized recommendations that assist customers in finding relevant items within large collections. A large gallery of photos on a mobile device can be used for analysis of the user's demography [4] and understanding of such interests as sport, gadgets, fitness, cloth, cars, food, traveling, etc. [5], [6].

In this paper I focus on one of the most challenging parts of photo organizing engine, namely, image-based event recognition [7], in order to extract such events as holidays, sport events, weddings, various activities, etc. An event can be defined as a category that captures the "complex behavior of a group of people, interacting with multiple objects, and taking place in a specific environment" [3]. There exist two different tasks of event recognition. The first one is focused on processing of single photos, i.e. event is considered as a complex scene with large variations in visual appearance and structure [3]. The second task aims at predicting the event categories of a group of photos (album) [8]. In the latter case it is assumed that all photos in an album are weakly labeled [9], though importance of each image may differ [10]. However, in practice only a gallery of photos is available so that the latter approach requires a user to manually choose the albums. Another option includes location-based album creation if the GPS tags are switched on. In both cases the usage of album-based event recognition is limited or even impossible.

Let me summarize the main contribution of this paper:

1) I consider the new task of event recognition, in which a gallery of photos is given and it is known that it contains ordered albums with unknown borders. This problem is much more practically-oriented than conventional album-based recognition [8]. A testing protocol is provided for this task which can be used with existing datasets suitable for album-based recognition [10], [11]. It is experimentally demonstrated that most obvious approach, namely, hierarchical or sequential clustering of CNN-based embeddings, does not lead to much more accurate decision when compared to recognition of single photos.
2) It is proposed to automatically assign these borders based on the visual content of consecutive photos in a

This research is based on the work supported by Samsung Research, Samsung Electronics. The work in the Section IV was prepared within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE).

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

gallery. I highlight the need to match the normalized confidence scores of classifiers for individual photos instead of conventional embeddings extracted by convolutional neural network (CNN). Consecutive photos are grouped and a final decision is made for a single descriptor of each group computed with an attention mechanism [12]. The developed engine is made as simple as possible in order to be considered as a suitable baseline for future studies of the new task from the previous item.

The rest of the paper is organized as follows. Existing deep learning-based event recognition techniques are briefly reviewed in Section II. The proposed approach is discussed in Section III. Experimental results for the Photo Event Collection (PEC) [11] and the Multi-Label Curation of Flickr Events Dataset (ML-CUFED) [10] are presented in Section IV. Finally, concluding comments are discussed in Section V.

II. Literature Survey

Annotating personal photo albums is an emerging trend in photo organizing services. A method for hierarchical photo organization into topics and topic-related categories on a smartphone is proposed in [13] based on integration of CNN and topic modeling for image classification. Organizing photo albums for user preference prediction is considered in [14].

Speaking about event recognition, there are two tasks studied in literature [7]. The first one predicts event type of a single photo [3] by using existing methods of image and scene recognition [6]. The second task includes event recognition in the whole album (a sequence of photos). There exist many techniques to solve the latter task. For example, the Stopwatch Hidden Markov Models (HMM) were applied in [11] by treating the photos in an album as sequential data. The paper [9] tackles the presence of irrelevant images in an album with active learning. An iterative updating procedure for event type and image importance score prediction in a siamese network is presented in [10]. The authors of this paper used a CNN that recognizes the event type, and a Long Short-Term Memory (LSTM)-based sequence level event recognizer in a whole album. Moreover, they successfully applied the method for learning representative deep features for image set analysis [15]. The latter approach focuses on capturing the co-occurrences and frequencies of features so that the temporal coherence of photos in an album is not required. A model to recognize events from coarse to fine hierarchical level using multi-granular features is proposed in [1] based on an attention network that learns the representations of photo albums. The efficiency of re-finding expected photos in mobile phones was improved by a method to classify personal photos based on relationship of shooting time/location to specific events [16].

The album information is not always available so that a gallery contains unstructured list of photos ordered by their creation time. In such case it is possible to use existing methods of event recognition on single photos [7]. Similar to other computer vision domains, it is typical to apply CNN-based architectures [17]. For example, four different layers of fine-tuned CNN were used for feature extraction to obtain the top entry in the ChaLearn LAP 2015 cultural event recognition challenge [18]. The bounding boxes of detected objects are projected onto multi-scale spatial maps for increasing the accuracy of event recognition [19]. The novel iterative selection method is introduced in [3] to identify a subset of classes that are most relevant for transferring deep representations learned from object (ImageNet) and scene (Places2) datasets.

Unfortunately, the accuracy of event classification on still photos [3] is in general much lower than the accuracy of album-based recognition [10]. Thus, an important task studied in this paper is automatic extraction of albums from a personal gallery based on a visual content of photos.

III. Materials and Methods

A. Event recognition in a gallery of photos

The main task can be formulated as follows. It is required to assign each photo $X_t, t \in \{1, ..., T\}$ from a gallery of an input user to one of $C > 1$ event categories (classes). Here $T \geq 1$ is the total number of photos in a gallery. I assume that the training set of $N \geq 1$ albums is available for learning of event classifier. The n-th reference album is defined by L_n images $\{X_n(1), ..., X_n(L_n)\}$. The set of class labels $c_n \subset \{1, ..., C\}$ of each n-th album is supposed to be given, i.e., an album may be associated with several event types [10].

Conventional event recognition on single photos [3] is the special case of above-formulated problem if $T = 1$. The main difference is the following assumption. The gallery $\{X_t\}$ is not a random collection of photos but can be represented as a *sequence of disjoint albums*. Each image in an album is associated with the same event. In contrast to the album-based event recognition, the borders of each album are unknown. This task possesses several characteristics that makes it extremely challenging compared to previously studied problems. One of these characteristics is the presence of irrelevant images or unimportant photos that can be in principle associated to any event [7]. These images are easily detected in attention-based models [1], [12], but may have a significant impact on a quality of automatic album selection.

The baseline approach here is to classify all T photos independently. The training albums may be unfolded into a set $\mathbf{X} = \{X_1(1), ..., X_1(L_1), X_2(1), ..., X_2(L_2), ..., X_N(L_N)\}$ of $L = L_1 + ... + L_N$ photos so that the collection-level label c_n of the n-th album is assigned to labels of each l-th photo ($l \in \{1, ..., L_n\}$). Next, it is possible to train an arbitrary event classifier. If L is rather small to train a deep CNN from scratch, the transfer learning or domain adaptation can be applied [17]. In these methods a large external dataset, e.g. ImageNet-1000 or Places2, is used to pre-train a deep CNN. As a special attention is paid to offline recognition on mobile devices, it is reasonable to use such CNNs as MobileNet v1/v2. The final step in transfer learning is fine-tuning of this neural network on $\mathbf{X}$. This step includes replacement of the last layer of the pre-trained CNN to the new layer with sigmoid (for multiple labels) or Softmax (for exactly one event label per album) activations and C outputs. During the classification

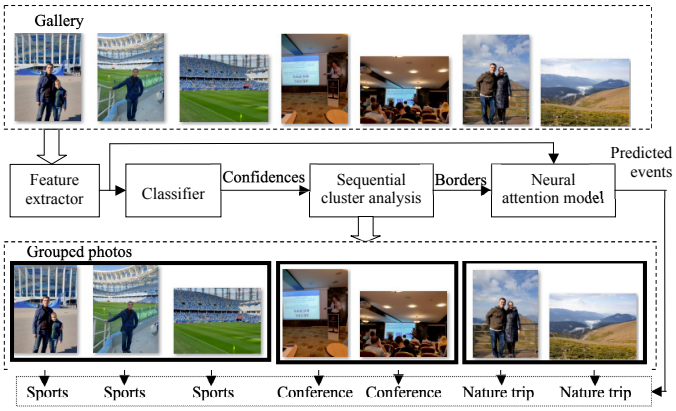


Fig. 1. Proposed gallery-based event recognition pipeline

process, each input image X_t is fed to the fine-tuned CNN to compute the scores (predictions at the last layer) $\mathbf{p}_t = [p_{1;t}, ..., p_{C;t}]$. This procedure can be modified by replacing C logistic regressions in the last layer to more complex classifier, e.g., random forest (RF), support vector machine (SVM) or gradient boosting. In this case the features (embeddings) [20] are extracted using the outputs of one of the last layers of pre-trained CNN. Namely, the images X_t and $X_n(l)$ are fed to the CNN, and the outputs of one of the last layers are used as the D-dimensional feature vectors $\mathbf{x}_t = [x_{1;t}, ..., x_{D;t}]$ and $\mathbf{x}_n(l) = [x_{n;1}(l), ..., x_{n;D}(l)]$, respectively. Such deep learning-based feature extractors allow training of a general classifier $\mathcal{C}$. The t-th photo is fed into this classifier to obtain C-dimensional confidence scores $\mathbf{p}_t$. Finally, the confidences $\mathbf{p}_t$ computed by any of above-mentioned ways are used to make a decision:

$$c^*(t) = \{\, c |\, c \in \{1, ..., C\}, p_{c;t} > p_0 \},\tag{1}$$

where p_0 is a fixed threshold for a minimal confidence score. This threshold should be estimated in such a way that the result set (1) will be empty if a photo describe some unseen (out-of-class) events, one can expect that all the confidence scores are rather low. If it is required to return exactly one event label, the class with the maximal confidence is returned:

$$c^*(t) = \underset{c \in \{1,...,C\}}{\operatorname{argmax}}\ p_{c;t}.\tag{2}$$

In addition to this baseline approach, hierarchical agglomerative clustering of embeddings $\mathbf{x}_t$ extracted by pre-trained CNN and confidence scores $\mathbf{p}_t$ with appropriate dissimilarity measure may be used for entire gallery. Next, the embeddings or confidence scores in each cluster are averaged in a single descriptor. The latter is classified to predict event classes for the whole cluster. Finally, these classes are assigned to each photo in a cluster. However, sequential nature of photos in a gallery is not used in this approach, so that potential accuracy increase for such a clustering is not high.

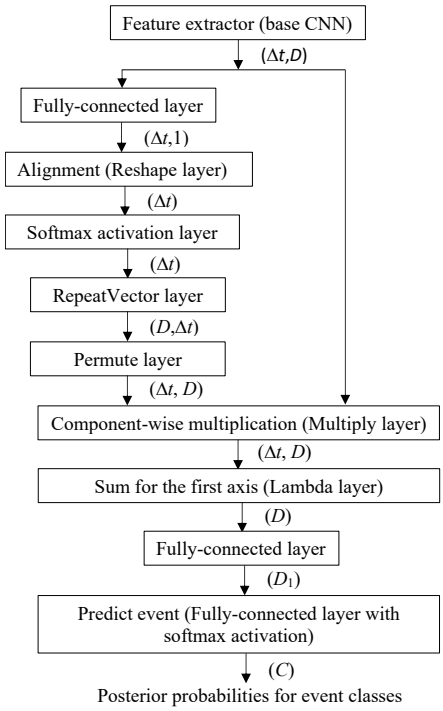


Fig. 2. Attention-based neural network. The shape of outputs is shown in brackets after each layer.

B. Proposed approach

The proposed pipeline is presented in Fig. 1. Here, firstly, the "Feature extractor" unit computes embeddings $\mathbf{x}_t$ of every t-th individual photo as described in previous Subsection III-A. The classifier confidences $\mathbf{p}_t$ are estimated in the "Classifier" unit. Next, sequential analysis is used from the Basic Sequential Algorithmic Scheme (BSAS) clustering [21] in the "Sequential cluster analysis" unit for a sequence of *confidences* $\{\mathbf{p}_t\}$ in order to obtain the borders of albums. Namely, the distances between neighbor photos $\rho(\mathbf{p}_t, \mathbf{p}_{t-1})$ are computed. The confidences of some classifiers $\mathcal{C}$, e.g., SVM returned by decision_function, i.e., distances to the separating hyper-plane, may be L_2-normalized before computation of this distance. If a distance does not exceed a certain threshold ρ_0 then it is assumed that both photos are included in the same album. Otherwise, the border between two albums is established at the t-th position. As a result, the borders $1 \le t_1 < ... < t_K = T$ of $K \in \{1, ..., T\}$ albums are obtained, so that the k-th album contains photos $X(t), t \in \{t_{k-1} + 1, ..., t_k\}$, where $t_0 = 0$.

At the second stage, the final descriptor of the k-th album is produced as a weighted sum of individual features $\mathbf{x}_t$:

$$\mathbf{x}(k) = \sum_{t=t_{k-1}+1}^{t_k} w(\mathbf{x}_t)\mathbf{x}_t,\tag{3}$$

where the weights w may depend on the features $\mathbf{x}_t$. It is typical to use here average pooling (AvgPool) with equal weights, so that the mean feature vector is computed. How-

Require: input gallery $X(t), t \in \{1, ..., T\}$
Ensure: sets of events $c^*(t) \subset \{1, ..., C\}$ for every image
1: Assign $K := 0$, initialize list of borders $B := []$
2: **for** each input image $t \in \{1, ..., T\}$ **do**
3: Feed the t-th image into a CNN and extract features $\mathbf{x}_t$
4: Use classifier $\mathcal{C}$ to compute confidences $\mathbf{p}_t$ and normalize them
5: **if** $t = 1$ or $\rho(\mathbf{x}_t, \mathbf{x}_{t-1}) > \rho_0$\{sequential cluster analysis\} **then**
6: Assign $K := K + 1$, append $t - 1$ to the list B
7: **end if**
8: **end for**
9: Append T to the list B
10: **for** each extracted album $k \in \{1, ..., K\}$ **do**
11: Feed input images $\{X_{B[k-1]+1}, X_{B[k-1]+2}, ..., X_{B[k]}\}$ into neural attention network (3)-(4) to obtain labels c^*
12: Assign $c^*(t) := c^*$ for all $t \in \{B[k-1] + 1, ..., B[k]\}$
13: **end for**
14: **return** set of event labels $c^*(t), t \in \{1, ..., T\}$

Fig. 3. Proposed gallery-based event recognition

1: **for** each album $n \in \{1, ..., N\}$ **do**
2: **for** each image $l \in \{1, ..., L_n\}$ **do**
3: Feed image $X_n(l)$ into a CNN and compute embeddings $\mathbf{x}_n(l)$
4: **end for**
5: **end for**
6: Train classifier $\mathcal{C}$ using unfolded training set $\mathbf{X}$ of embeddings
7: Train attention network (3)-(4) using subsets with fixed size S of all training sets of features $\{\mathbf{x}_n(l)\}$
8: **for** each album $n \in \{1, ..., N\}$ **do**
9: **for** each image $l \in \{1, ..., L_n\}$ **do**
10: Feed embeddings $\mathbf{x}_n(l)$ into classifier $\mathcal{C}$ to predict confidence scores $\mathbf{p}_n(l)$ and normalize them
11: **end for**
12: **end for**
13: Randomly permute all indices $\{1, ..., N\}$ to obtain sequence $(n_1, ..., n_N)$
14: Unfold training embeddings using this permutation: $\tilde{\mathbf{X}} = \{X_{n_1}(1), ..., X_1(L_{n_1}), ..., X_{n_N}(1), ..., X_{n_N}(L_{n_N})\}$
15: Assign $\rho := 0, \alpha^* := 0$
16: **for** each potential threshold ρ **do**
17: Call Algorithm (Fig. 3) with parameters $\tilde{\mathbf{X}}$, $\mathcal{C}$ and threshold ρ
18: Compute accuracy α using predictions for all training images
19: **if** $\alpha^* < \alpha$ **then**
20: Assign $\alpha^* := \alpha, \rho_0 := \rho$
21: **end if**
22: **end for**
23: **return** classifier $\mathcal{C}$, attention network, threshold ρ_0

Fig. 4. Learning procedure in the proposed approach

ever, I propose to learn the weights $w(\mathbf{x}_t)$ with an attention mechanism in this paper:

$$w(\mathbf{x}_t) = \frac{\exp(\mathbf{q}^T \mathbf{x}_t)}{\sum\limits_{j=t_{k-1}+1}^{t_k} \exp(\mathbf{q}^T \mathbf{x}_j)}. \quad (4)$$

Here $\mathbf{q}$ is the learnable D-dimensional vector of weights. The dense (fully connected) layer is attached to the resulted descriptor $\mathbf{x}(k)$, and the whole neural network (Fig. 2) is trained in end-to-end manner using given training set of $N \geq 1$ albums. Here Δt is the number of images in the input set ($t_k - t_{k-1} + 1$) and D_1 is an arbitrary width of the hidden layer ($D_1 = 128$ will be used further). The event class predicted by this network in the "Neural attention model" unit (Fig. 1) is assigned to all photos $X(t), t \in \{t_{k-1} + 1, ..., t_k\}$.

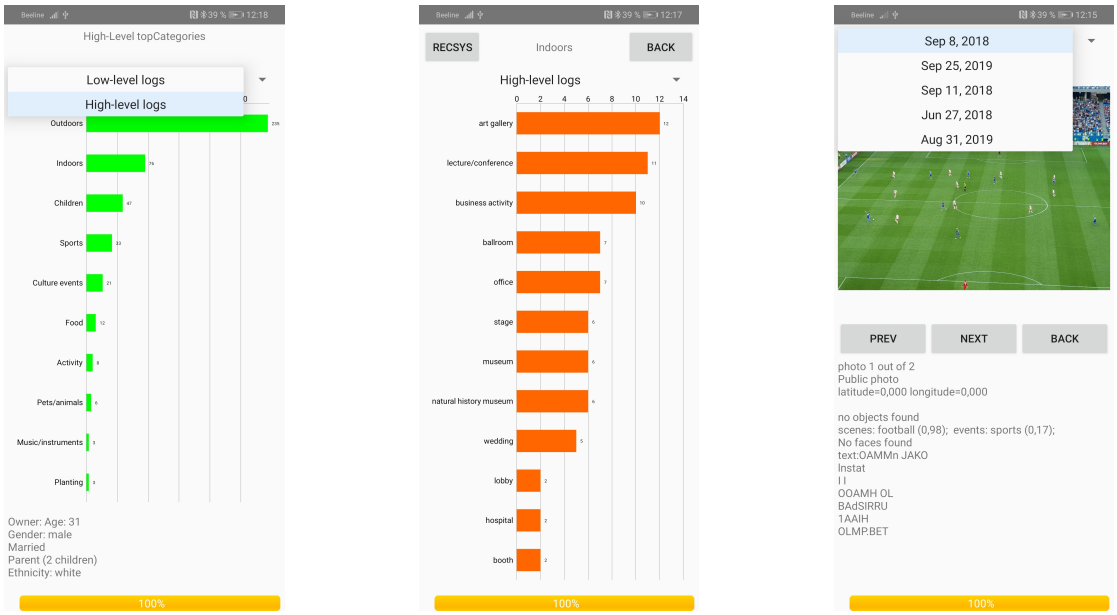
Complete algorithm for our pipeline (Fig. 1) is presented in Fig. 3. The learning procedure is shown in Fig. 4. It permutes the training set (steps 13,14) in order to create a gallery, in which the sequence of albums and their positions are random, though photos from the same album are located close to each other. Moreover, it calls the event prediction at step 17 for simplicity. However, to speed-up computations it is recommended to pre-compute the pair-wise distance matrix between confidence scores of all training images so that feature extraction (steps 3-4 in Algorithm (Fig. 3)) and distance calculation are not needed during the learning stage. New photos are processed very efficiently as there is no need to repeat the processing of the whole gallery. If a new photo is entered into the gallery it may be grouped with the last K-th cluster if the contents of the last photo and the new one are similar or be treated as a new cluster.

I implemented the whole pipeline (Fig. 1) in the publicly-available demo application for Android (https://drive.google.com/open?id=1rThhcKReOb5A9LBIH6jkP8tTiYjoVNWH) (Fig. 5), that was previously developed to extract user preferences by processing all photos from the gallery [14]. The similar events found in photos made in one day were united into High-level logs for the most important events. Only those scenes/events are displayed for which there exist at least 2 photos and the average score of scene/event predictions for all photos of the day exceeds a certain threshold. The sample screenshot of the main user interface is shown in Fig. 5a. It is possible to tap any bar in this histogram to show a new form with detailed categories (Fig. 5b). If a concrete category is tapped, a "display" form appears, which contains a list of all photos from the gallery with this category (Fig. 5c). Here events are grouped by date and provide a possibility to choose concrete day.

IV. Experimental Results

Two main datasets in event recognition in personal photo-collections [7] are examined, namely:

1) PEC [11] with 61,364 images from 807 collections of 14 social event classes (birthday, wedding, graduation, etc.). I used its split provided by authors: the training set



(a) (b) (c)

Fig. 5. Mobile demo GUI

with 667 albums (50,279 images) and testing set with 140 albums (11,085 images).

2) ML-CUFED [10] contains 23 common event types. Each album is associated with several events, i.e., it is a multi-label classification task. Conventional split into the training set (75,377 photos, 1507 albums) and test set (376 albums with 19,420 photos) was used.

The features were extracted at the outputs of penultimate (global average pooling) layers of scene recognition models (Inception v3, ResNet-101 and MobileNet v2 with $\alpha = 1$ and $\alpha = 1.4$) pre-trained on the Places2 dataset. In addition, I took the standard versions of EfficientNet v5 (Rand-aug) and v7 (Rand-aug) [22] pre-trained on the ImageNet dataset.

I used two techniques discussed in Subsection III-B to obtain a final descriptor of a set of images, namely, AvgPool and the neural attention mechanism (3)-(4) for L_2-normed features. In the former case the average descriptors are classified with the feed-forward neural network identical to last two layers from Fig. 2 in order to use the same classifiers as in the attention-based model. In the latter case its weights are learned using the sets with $S = 10$ randomly chosen images from all albums in order to make identical shape of input tensors. As a result, 667 training subsets and 1507 subsets with $S = 10$ images were obtained for PEC and ML-CUFED, respectively. As the ML-CUFED contains multiple labels per each album, sigmoid activations and binary cross-entropy loss were used. Conventional Softmax activations and categorical cross-entropy are applied for the PEC. The model was learned using ADAM optimizer (learning rate 0.001) for 10 epochs with early stop in Keras 2.3 framework with TensorFlow 1.15 backend. The linear SVM classifier from scikit-learn library was used as $\mathcal{C}$ to combine sequential photos in Step 4 (Fig. 3), because it has higher accuracy than RF, k-NN and RBF SVM.

TABLE I
ACCURACY (%) OF EVENT RECOGNITION IN A SET OF IMAGES (ALBUM).

CNN	Aggregation	PEC	ML-CUFED
MobileNet2, $\alpha = 1.0$	AvgPool	86.42	81.38
	Attention	89.29	84.04
MobileNet2, $\alpha = 1.4$	AvgPool	87.14	81.91
	Attention	87.86	84.31
Inception v3	AvgPool	86.43	82.45
	Attention	87.86	84.84
EfficientNet v5	AvgPool	88.57	86.70
	Attention	89.29	88.56
EfficientNet v7	AvgPool	89.29	88.83
	Attention	90.0	90.42
ResNet-101	AvgPool	86.43	82.18
	Attention	88.57	85.37
	CNN-LSTM-Iterative [10]	84.5	71.7
	Feature learning [15]	89.1	83.4
AlexNet	CNN-LSTM-Iterative [10]	84.5	79.3
	Feature learning [15]	87.9	84.5
-	Stopwatch HMM [11]	55.71	-
A-net	Feature-learning [15]	73.43	-
GoogLeNet	R-OS-PGM [8]	74.28	-
VGGNet	Active learning [9]	85.0	-
Ensemble of 3 VGGNets	Hierarchical attention [1]	87.86	-

A. Album-based event recognition

The recognition accuracies of the pre-trained CNN for album-based event recognition in conventional testing proto-

cols of these datasets together with the best-known results from literature [10], [15] are presented in Table I. The multi-label accuracy is computed for ML-CUFED so that prediction is assumed to be correct if it corresponds to any label associated with an album.

Here the modern EfficientNets provide 4-6% higher accuracy for the ML-CUFED dataset when compared to the state-of-the-art method [10]. However, their improvements over existing CNNs for the PEC are not so huge. Anyway, they outperform the known best accuracy for this dataset on 0.9%. The most remarkable fact here is that one of the best results for the PEC are achieved for the most simple model (MobileNet v2, $\alpha = 1.0$), which can be explained by the lack of training data for this particular dataset. Finally, the attention-based aggregation is 1-3% more accurate when compared to classification of average features in all cases. As one can notice, the proposed implementation of attention mechanism achieves the known state-of-the-art results, though I used much faster CNNs (MobileNet and Inception rather than AlexNet and ResNet-101) and do not consider sequential nature of photos in an album in the attention-based network (Fig. 2). Anyway, slight improvement of the best results for album-based event recognition is auxiliary part of this study. It just demonstrates that the quality of attention model is comparable with other techniques.

B. Event recognition in single images

The task considered in this paper (Subsection III-A) is new so that there are no results for it in existing literature. It is a generalization of event recognition in still images, because in practice there is no information about albums in a gallery. Moreover, class labels are assigned to each photo from the gallery individually (Step 12 in my Fig. 3). Thus, it is necessary to compare proposed approach with existing state-of-the-art for recognition in still images. In the next experiment the collection-level first label is directly assigned to each image contained in both datasets and simply use the image itself for event recognition, without any meta information. In addition to baseline approach (Subsection III-A), the average linkage clustering was used, which achieved the best accuracy when compared to other hierarchical agglomerative clustering techniques fro scikit-learn. Euclidean (L_2) distance between embeddings and confidence scores is implemented in all cases. In addition, I used chi-squared (χ^2) distance to match non-negative embeddings from several CNNs. It is impossible to use chi-squared (χ^2) distance for the confidence scores returned by decision_function for LinearSVC, because they are not always non-negative. The results are shown in Table II.

Here, firstly, the accuracy of event recognition in single images is 25-30% lower than the accuracy of the album-based classification (Table I). Secondly, the best known accuracy (62.2%) [3] of image-level event recognition for the PEC is 2.6% lower when compared to the usage of the best EfficientNet [22]. Thirdly, clustering of the confidence scores at the output of the best classifier does not significantly influence the overall accuracy. Fourthly, hierarchical clustering with the χ^2 distance leads to slightly more accurate results than conventional Euclidean metric. Finally, preliminarily clustering decreases the error rate of the baseline in only 1.2-2% even if the distance threshold in clustering is carefully chosen.

TABLE II
ACCURACY (%) OF EVENT RECOGNITION IN A SINGLE IMAGE.

Dataset	CNN	Baseline	Average linkage clustering		
			Embeddings		Scores
			L_2	χ^2	L_2
PEC	MobileNet2, $\alpha = 1.0$	58.32	60.42	60.69	58.44
	MobileNet2, $\alpha = 1.4$	60.34	61.25	61.92	60.58
	Inception v3	61.82	64.19	64.22	61.97
	ResNet-101	61.56	64.19	63.67	61.78
	EfficientNet v5	63.25	65.15	-	63.37
	EfficientNet v7	64.81	66.00	-	64.91
	OS2E-CNN (two BN-Inceptions) [3]	60.6	-	-	-
	OS2E-CNN (four data+knowledge BN-Inceptions) [3]	62.2	-	-	-
ML-CUFED	MobileNet2, $\alpha = 1.0$	54.41	57.03	57.45	54.56
	MobileNet2, $\alpha = 1.4$	53.54	54.97	55.98	54.03
	Inception v3	57.26	59.19	60.12	57.87
	ResNet-101	55.56	58.09	58.59	65.72
	EfficientNet v5	59.78	62.13	-	59.93
	EfficientNet v7	61.58	64.36	-	61.67

C. Event recognition in a gallery of photos

Let me demonstrate how the assumption about sequentially ordered photos in an album can increase the accuracy of event recognition. I propose the following protocol for existing event datasets with known album labels to make the task more complex. The sequence of albums is randomly shuffled. The photos in each album are also shuffled. This transformation of the order of testing photos was performed 10 times, and average accuracy and its standard deviation are evaluated.

In addition to pre-trained CNNs, I fine-tuned several CNNs using the unfolded training set $\mathbf{X}$ as follows. At first, the weights in the base part of the CNN were frozen and the new head (fully connected layer with C outputs and Softmax activation) was learned during 10 epochs. Next, the weights in the whole CNN were learned during 3 epochs with 10-times lower learning rate. The scores of fine-tuned CNN are L_1-normalized, so that additional normalization is not required.

The results (mean accuracy $\pm$ its standard deviation) of the proposed Algorithms 3, 4 for the PEC and the ML-CUFED using my testing protocol are presented in Table III and Table IV, respectively. Here the attention mechanism provides up to 8% lower error rates in most cases. This increase is much higher when compared to 1-3% gain of attention mechanism in traditional album-based event recognition (Table I). It is remarkable that the matching of distances between L_2-normed confidences significantly improves the overall accuracy of attention model for the PEC (Table III), though my experiments did not show any improvements in conventional clustering

TABLE III
ACCURACY (%) OF THE PROPOSED APPROACH, PEC.

CNN	Aggregation	Baseline	Embeddings		Scores	Scores (normalized)	
			L_2	χ^2	L_2	L_2	χ^2
MobileNet2, $\alpha = 1.0$	AvgPool	58.32	66.85 ± 0.59	68.52 ± 0.89	71.08 ± 0.59	72.68 ± 0.56	-
(pre-trained), embeddings	Attention	54.43	68.51 ± 0.41	70.65 ± 1.20	74.49 ± 0.70	80.48 ± 1.01	-
MobileNet2, $\alpha = 1.4$	AvgPool	60.34	68.85 ± 0.59	69.57 ± 0.57	72.59 ± 1.49	73.49 ± 0.86	-
(pre-trained), embeddings	Attention	55.36	70.53 ± 0.79	71.16 ± 0.72	78.20 ± 1.47	81.27 ± 0.81	-
MobileNet2, $\alpha = 1.4$	AvgPool	61.89	-	-	75.66 ± 0.55		76.96 ± 0.97
(fine-tuned), scores	Attention	61.55	-	-	78.77 ± 0.49		81.33 ± 0.69
Inception v3	AvgPool	61.82	72.29 ± 1.28	72.32 ± 1.54	74.54 ± 1.04	76.48 ± 0.47	-
(pre-trained), embeddings	Attention	56.94	72.38 ± 1.13	71.96 ± 0.67	76.76 ± 0.70	80.17 ± 1.14	-
Inception v3	AvgPool	63.56	-	-	78.87 ± 0.67		79.92 ± 0.65
(fine-tuned), scores	Attention	62.91	-	-	81.03 ± 0.77		81.95 ± 1.11
ResNet-101	AvgPool	61.56	72.45 ± 0.75	72.29 ± 1.09	76.35 ± 0.94	76.10 ± 0.47	-
(pre-trained), embeddings	Attention	58.26	73.14 ± 0.97	72.69 ± 0.95	78.95 ± 1.01	80.74 ± 0.57	-
EfficientNet v5	AvgPool	63.25	75.00 ± 0.95	-	76.99 ± 0.87	77.38 ± 0.73	-
(pre-trained), embeddings	Attention	59.58	77.45 ± 1.34	-	80.58 ± 0.92	82.08 ± 0.60	-
EfficientNet v7	AvgPool	64.81	77.11 ± 0.53	-	79.58 ± 0.43	80.37 ± 0.58	-
(pre-trained), embeddings	Attention	60.19	80.12 ± 0.48	-	81.53 ± 0.78	83.42 ± 0.76	-

TABLE IV
ACCURACY (%) OF THE PROPOSED APPROACH, ML-CUFED.

CNN	Aggregation	Baseline	Embeddings		Scores	Scores (normalized)	
			L_2	χ^2	L_2	L_2	χ^2
MobileNet2, $\alpha = 1.0$	AvgPool	54.41	67.54 ± 0.76	67.42 ± 0.93	69.83 ± 0.74	70.42 ± 0.41	-
(pre-trained), embeddings	Attention	51.05	68.71 ± 0.71	68.55 ± 0.61	71.44 ± 0.82	71.61 ± 0.69	-
MobileNet2, $\alpha = 1.4$	AvgPool	53.54	66.93 ± 0.60	67.21 ± 0.55	68.56 ± 0.73	69.47 ± 0.36	-
(pre-trained), embeddings	Attention	51.12	68.34 ± 0.68	68.62 ± 0.50	70.79 ± 0.75	71.78 ± 0.74	-
MobileNet2, $\alpha = 1.4$	AvgPool	56.01	-	-	70.57 ± 0.48		71.61 ± 0.28
(fine-tuned), scores	Attention	56.09	-	-	72.90 ± 0.59		73.46 ± 0.58
Inception v3	AvgPool	57.26	69.91 ± 0.58	70.01 ± 0.62	72.25 ± 0.61	72.78 ± 0.71	-
(pre-trained), embeddings	Attention	50.89	69.30 ± 0.47	68.52 ± 0.89	72.73 ± 0.72	73.00 ± 0.65	-
Inception v3	AvgPool	57.12	-	-	72.18 ± 0.63		73.20 ± 0.74
(fine-tuned), scores	Attention	57.29	-	-	73.06 ± 0.74		73.92 ± 0.81
ResNet-101	AvgPool	55.56	69.90 ± 0.73	69.06 ± 0.71	72.14 ± 0.60	71.87 ± 0.62	-
(pre-trained), embeddings	Attention	51.80	70.25 ± 0.66	68.14 ± 0.70	72.80 ± 0.91	73.58 ± 0.82	-
EfficientNet v5	AvgPool	59.78	73.05 ± 0.48	-	74.36 ± 0.57	75.28 ± 0.41	-
(pre-trained), embeddings	Attention	56.31	73.88 ± 0.44	-	76.02 ± 0.59	77.08 ± 0.38	-
EfficientNet v7	AvgPool	61.58	75.20 ± 0.41	-	75.92 ± 0.43	76.82 ± 0.71	-
(pre-trained), embeddings	Attention	57.79	76.10 ± 0.53	-	76.63 ± 0.45	78.31 ± 0.75	-

from the previous experiment (Table II). The fine-tuned CNNs obviously lead to the most accurate decision, but the difference (0.1-1.6%) with the best results of the pre-trained models is rather small. However, the latter do not require additional inference in existing scene recognition models, so the implementation of event recognition in an album will be very fast if the scenes should be additionally classified, e.g., for more detailed user modeling [14]. Surprisingly, computing the distance between confidence scores of classifiers ($\rho(\mathbf{p}_t, \mathbf{p}_{t-1})$) reduces the error rate of conventional matching of embeddings ($\rho(\mathbf{x}_t, \mathbf{x}_{t-1})$) on 2-7%. Let me recall that conventional clustering of embeddings was 1-2% more accurate when compared to the classifier's scores (Table II). It seems that the threshold ρ_0 can be estimated (Fig. 4) more reliably in this particular case when most images from the same event are matched in the prediction procedure (Fig. 3). Finally, the most important conclusion is that the proposed approach has 9-23% higher accuracies when compared to baseline state-of-the-art image-level event recognition. Moreover, my algorithm is 13-16% more accurate than classification of groups of photos obtained with hierarchical clustering (Table II).

V. Conclusion

Existing studies of event recognition cannot be directly used for processing of a gallery of mobile device because the albums of photos corresponding to the same event may be unavailable. The usage of event recognition in single images is possible but is very inaccurate even if similar photos are combined with a clustering of visual features (Table II). I have demonstrated that grouping of consecutive photos and attention-based recognition of resulted image sets (Fig. 3) can drastically (up to 23%) improve the accuracy (Tables III, IV) of the baseline state-of-the-art methods of image-level event recognition (Table II). The consecutive photos from the same album are discovered much better if the confidence scores of classifier are matched. The classifier has been learned on unfolded training set $\mathbf{X}$. It was unexpected that L_2-normed confidence scores of SVM classifier (distances to the hyperplane) are grouped so efficiently. Moreover, the χ^2 distance for the CNN-based scores works much better than conventional L_2 distance. It is important that the usage of the same training set is enough to automatically estimate the most important parameter, namely, distance threshold ρ_0, in the learning procedure (Fig. 4). Finally, the proposed implementation of attention-based network and its training procedure slightly improved the known state-of-the-art for two widely-used datasets (Table I) if EfficientNet v7 is used for feature extraction [22].

Proposed engine has been implemented in the publicly available Android application (Fig. 5) that extracts the profile of user's interests. It is applicable for such personalized mobile services as recommender systems and target advertisements.

The main disadvantage of the proposed approach is its rather lower accuracy (up to 8-11%) when compared to the best models for the case of known borders of albums (Table I). In future it is necessary to replace the pre-defined distance $\rho(\mathbf{p}_t, \mathbf{p}_{t-1})$ to a metric learned on a given training set [17]. Moreover, the issue of irrelevant photos [10] should be thoroughly studied. The proposed approach with matching of the distances between neighbor images simply splits an album into several chunks so that unimportant photos will be assigned to very small groups. However, it was experimentally noticed that it is better to match neighbor images than find the minimal distance of the t-th photo with several previous photos or even all images in the current group. Finally, it is desirable to improve the attention model, which does not work well now for single photos: its accuracy for the baseline with pre-trained CNNs is 4-5% worth than the accuracy of linear SVM (row "AvgPool" in Tables III, IV).

References

[1] C. Guo, X. Tian, and T. Mei, "Multigranular event recognition of personal photo albums," *IEEE Transactions on Multimedia*, vol. 20, no. 7, pp. 1837–1847, 2017.

[2] A. D. Sokolova, A. S. Kharchevnikova, and A. V. Savchenko, "Organizing multimedia data in video surveillance systems based on face verification with convolutional neural networks," in *Proceedings of the International Conference on Analysis of Images, Social Networks and Texts (AIST)*. Springer, 2017, pp. 223–230.

[3] L. Wang, Z. Wang, Y. Qiao, and L. Van Gool, "Transferring deep object and scene representations for event recognition in still images," *International Journal of Computer Vision*, vol. 126, no. 2-4, pp. 390–409, 2018.

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

[5] I. Grechikhin and A. V. Savchenko, "User modeling on mobile device based on facial clustering and object detection in photos and videos," in *Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2019, pp. 429–440.

[6] A. Rassadin and A. Savchenko, "Scene recognition in user preference prediction based on classification of deep embeddings and object detection," in *Proceedings of the International Symposium on Neural Networks (ISNN)*, vol. 11555. Springer, 2019, pp. 422–430.

[7] K. Ahmad and N. Conci, "How deep features have improved event recognition in multimedia: A survey," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 15, no. 2, p. 39, 2019.

[8] S. Bacha, M. S. Allili, and N. Benblidia, "Event recognition in photo albums using probabilistic graphical models and feature relevance," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 546–558, 2016.

[9] K. Ahmad, M. L. Mekhalfi, and N. Conci, "Event recognition in personal photo collections: An active learning approach," *Electronic Imaging*, vol. 2018, no. 2, pp. 173–1, 2018.

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

[11] L. Bossard, M. Guillaumin, and 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.

[12] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and 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.

[13] S. Lonn, P. Radeva, and M. Dimiccoli, "Smartphone picture organization: A hierarchical approach," *Computer Vision and Image Understanding*, vol. 187, p. 102789, 2019.

[14] A. V. Savchenko, K. V. Demochkin, and I. S. Grechikhin, "User preference prediction in visual data on mobile devices," *arXiv preprint arXiv:1907.04519*, 2019.

[15] Z. Wu, Y. Huang, and L. Wang, "Learning representative deep features for image set analysis," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1960–1968, 2015.

[16] M. Geng, Y. Li, and F. Liu, "Classifying personal photo collections: an event-based approach," in *Proceedings of the Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*. Springer, 2018, pp. 201–215.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[18] S. Escalera, J. Fabian, P. Pardo, X. Baro, J. Gonzalez, H. J. Escalante, D. Misevic, U. Steiner, and I. Guyon, "ChaLearn looking at people 2015: Apparent age and cultural event recognition datasets and results," in *Proceedings of the International Conference on Computer Vision Workshops (ICCVW)*, 2015, pp. 1–9.

[19] Y. Xiong, K. Zhu, D. Lin, and 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.

[20] A. V. Savchenko, "Sequential three-way decisions in multi-category image recognition with deep features based on distance factor," *Information Sciences*, vol. 489, pp. 18–36, 2019.

[21] W. M. Ashour, R. Z. Muqat, A. B. AlQazzaz, and S. R. AbdElnabi, "Improve basic sequential algorithm scheme using ant colony algorithm," in *Proceedings of the 7th Palestinian International Conference on Electrical and Computer Engineering*. IEEE, 2019, pp. 1–6.

[22] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.