PAPER

Finding Important People in a Video Using Deep Neural Networks with Conditional Random Fields

Mayu OTANI[†a)], **Atsushi NISHIDA**[††], ***Nonmembers***, **Yuta NAKASHIMA**[†††], **Tomokazu SATO**[††††], ***Members***, ***and*** **Naokazu YOKOYA**[†], ***Fellow***

SUMMARY Finding important regions is essential for applications, such as content-aware video compression and video retargeting to automatically crop a region in a video for small screens. Since people are one of main subjects when taking a video, some methods for finding important regions use a visual attention model based on face/pedestrian detection to incorporate the knowledge that *people are important*. However, such methods usually do not distinguish important people from passers-by and bystanders, which results in false positives. In this paper, we propose a deep neural network (DNN)-based method, which classifies a person into important or unimportant, given a video containing multiple people in a single frame and captured with a hand-held camera. Intuitively, important/unimportant labels are highly correlated given that corresponding people's spatial motions are similar. Based on this assumption, we propose to boost the performance of our important/unimportant classification by using conditional random fields (CRFs) built upon the DNN, which can be trained in an end-to-end manner. Our experimental results show that our method successfully classifies important people and the use of a DNN with CRFs improves the accuracy.

key words: *neural network, conditional random field, important people classification*

1. Introduction

Some applications related to videos require finding important regions in video frames in order for semantically meaningful video handling. Video retargeting is one of such applications, which automatically crops a video that is originally for big screens to make it fit to smaller screens [1]. Another example application is content-aware video compression, which assigns more bits to important regions [2].

Most of such applications rely on visual attention models. These models mainly based on visual saliency to find prominent regions in images or videos. One of the most well-known approaches was proposed by Itti *et al.* [3] that mimics the vision system of primates. This type of models well suit to finding regions that may inherently draw attention (*e.g.*, a red ball on grass fields or rapidly moving objects).

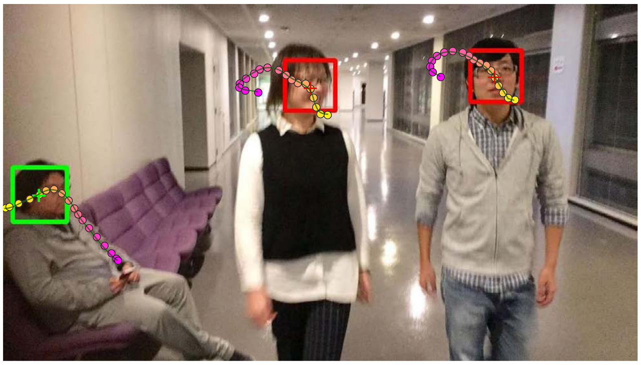

Fig. 1 Example of important people (red) and an unimportant person (green). The important people are walking together, so their trajectories are highly correlated, while the unimportant person makes completely different trajectories.

Some of more recent techniques integrate a higher-level cues into visual attention models [4]. One of the powerful and convincing approaches leverages face detection results, along with other types of cues, based on the observation that facial or human body regions attract the humans' attention. Their approach may be also motivated by the fact that people are one of major content of images or videos.

One possible criticism on such techniques that use face or human body detection results can be that all people in a video frame are not always important for the videos (Fig. 1). Taking the video retargeting application as an example and supposing only a subset of people in a video are important, the video might be spoiled if the important people are removed in the course of applying video retargeting; however, other people might be just passers-by and bystanders who are not necessarily in a cropped video. This can be a critical problem for consumer generated videos, most of which are captured by people who do not have any specific experience nor training in video shooting with hand-held cameras.

In order to address this problem, we need an automatic technique to classify a person into important/unimportant one, where there are an arbitrary number of important people in a single video frame. The problem is that there is no obvious gold standard in distinguishing important/uninportant people, and the person who is important in a given video might be different viewer to viewer. For example, parents deem their kids are important in most cases, while the kids may not be important for other people. One possible way to disambiguate important people is adopting

Manuscript received January 19, 2018.
Manuscript revised June 23, 2018.
Manuscript publicized July 20, 2018.

[†] The authors are with Nara Institute of Science and Technology, Ikoma-shi, 630–0192 Japan.

[††] The author is with Dai Nippon Printing Co., Ltd., Tokyo, 162–8001 Japan.

[†††] The author is with Osaka University, Suita-shi, 565–0871 Japan.

[††††] The author is with Shiga University, Hikone-shi, 522–0069 Japan.

a) E-mail: otani_mayu@cyberagent.co.jp

DOI: 10.1587/transinf.2018EDP7029

the videographer's viewpoint: Since the videographer usually has something that she/he intends to capture, important/unimportant people can be obvious.

Some research efforts have been made in this direction *e.g.* [5], which is based on the observation that videographers move or operate their cameras differently when capturing important people and unimportant people. For example, a videographer may move the camera to follow an important person if the person is moving, while the motion of an unimportant person does not affect the videographer's behavior. Taking this into account, these methods use trajectories of detected people as features and classify people in each frame into important/unimportant. The work also proposed to incorporate the prior that a group of important people tend to have similar trajectories as in Fig. 1. Their results demonstrated that the classification performance can be improved by modeling correlation among trajectories of two or more important people using conditional random fields (CRFs).

This paper also presents a model to classify detected people in a video captured by a hand-held camera into important/unimportant ones for higher-level visual attention models, which can handle multiple important people in a single frame. To further boost the performance of a CRF-based classifier, we construct a deep model with CRFs that can be trained in the end-to-end fashion. By doing this, we can expect that the trained model is fully optimized to our problem of finding important people. Assuming that the number of people in a single frame is small (*e.g.*, less than 10), we can evaluate the partition function in the negative log-likelihood of our DNN-CRF in the exhaustive manner without approximation. The contribution of this paper is summarized as follows:

- We propose a model for important/unimportant people classification, which can improve the classification performance thanks to end-to-end training of our deep model. This allows additional tuning on features for our classification task.
- We develop an easy-to-implement deep neural network (DNN) model with a CRF layer that can be implemented using off-the-shelf tools for DNNs. Yet, our model might be applied to such problems as social role discovery [6] and articulated pose estimation [7].
- We experimentally demonstrate that our method outperforms vanilla DNN-based and support vector machine classifiers.

2. Related Work

Visual attention models have been used in various applications that exploit detection of important regions in images/videos. Such applications include video summarization [4], video retargeting [1], virtual cinematography [8], and content-aware video compression [2].

The early work on visual attention model is Itti *et al.*'s visual saliency [3], which is well-known and adopted in later works. Itti's group also proposed to model "surprise" to find visually prominent regions [9]. An interesting extension of such visual attention models is to integrate higher-level cues [10]. For example, humans' vision system tends to focus on people's faces, and people are one of the most important contents in images/videos. This observation can be leveraged in the model by using detected faces or human bodies for candidate salient regions as in Ma *et al.*'s model [4].

Considering that people are not equally important as mentioned in the previous section, classifying people in video frames into important vs. unimportant can be beneficial for such applications as video summarization, video retargeting, *etc*. Nakashima *et al.* proposed to classify people in this way from videographers' perspective [5]. Observing that important people in the same video frame tend to have highly correlated features (*i.e.*, trajectories and sizes of bounding boxes), they employed a CRF-based model to leverage this observation into classification.

Recent progress in large scale datasets [11], [12] and DNN techniques have significantly improved the performance of various vision tasks, such as object classification [13]–[15] and semantic segmentation [16]–[19]. In this work, we also develop a deep model to classify people into important or unimportant ones, which is an extension of [5], [20]. As in these work, we uses a CRF built upon a deep model. Incorporating CRFs into DNNs is one of the main research directions to improve structured output modeling. Ma *et al.* built an LSTM language model with a CRF layer for part-of-speech tagging [21]. In the domain of computer vision, recent works have shown that the use of CRFs boosts the performance of semantic segmentation [16]–[19] and human pose estimation [22]. Chandra *et al.* proposed an end-to-end training of a deep architecture with CRFs by designing a quadratic CRF layer which can be efficiently optimized [19].

Our problem of classifying people in a video frame into important/unimportant is small: there usually are a few people in a video, and the CRFs need only a few nodes corresponding to detected people to model the correlation among them. This means that we can exhaustively evaluate the partition function in the negative log-likelihood that is minimized during the training session, without any sampling steps to approximate the partitioning function. Our method can be easily implemented using off-the-shelf tools for DNNs.

3. Overview

In order to classify people in a video frame in a supervised fashion, we need a ground truth label $t_i \in \{0, 1\}$ for person i in a certain frame, where $t_i = 1$ means the person is important. However, who is important in a given video might vary for different viewers. To determine ground truth importance labels of people, we take the videographers' perspective, following [5], which is advantageous in two ways: Firstly the ground truth labels assigned to each detected person is not ambiguous because they solely depend on the videographer's intention. Secondly the videographer moves or operates their cameras (*e.g.*, following a moving person) to capture what they want to show to others. Each person's two-dimensional trajectory in a video frame is a combination of the person's motion in the scene and the camera motion in response to the person's motion, and thus who is important for the videographer should be reflected in the person's trajectory.

Given this observation, we use the trajectory of a detected person i obtained by tracking the person for L frames in both temporal directions centered at the frame. Tracking of a person gives us bounding boxes, whose edges have the same length, in successive L frames, and we use the center positions (horizontal and vertical positions) and the length of edges of the bounding boxes as our spatial features, which forms spatial feature vector $x_i \in \mathbb{R}^{3L}$. Visibility of faces is also helpful for this classification task because videographers seem to capture people's frontal faces or at least profile faces. We thus use either color histograms or CNN-based features pretrained for a face recognition task (*e.g.* [23]). The color histogram or CNN-based features form a facial feature vector y_i whose size is K_{H} or K_{F}, respectively.

An interesting insight is presented in [5] that the trajectories of important people in a single video frame are highly correlated as mentioned above (for example, if there are two people walking together as shown in Fig. 1, the trajectories of these people should be similar, and it may not be likely that only one of them is important). To encode this insight, we also use CRFs to model such spatial relationships among people in a frame. CRFs takes pairs of input features, *e.g.*, trajectories and facial feature vectors, as input and compute energy functions over the pairs to predict a set of labels for people in a video frame. We can expect that the CRF layer is trained to capture relations between people in a video frame.

Figure 2 shows an overview of our method. Since our features are not well designed for this task, we use a simple neural network in the lower part of our model to transform the original spatial and facial feature vectors. They are then handled by a CRF network to compute the posterior probability of a given label combination. In the following sections, we detail our DNN-CRF model.

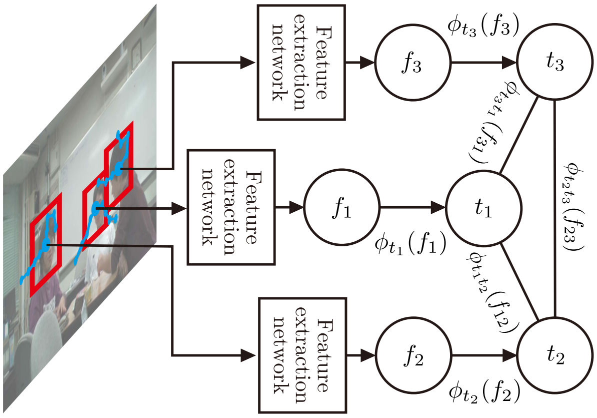

Fig. 2 An overview of our model. Feature vectors extracted from each face regions are fed to the feature extraction network, which fuses spatial features and facial features. The CRF layer takes the outputs of the feature extraction network and computes the posterior probability for a set of labels. Our method employs a label combination with a highest probability.

4. DNN-CRF Model for Classification

4.1 Feature Extraction Networks

Figure 3 illustrates the lower part of our model. It takes a spatial feature vector x_i and a facial feature vector y_i separately as input. After fully connected (FC) layers with the ReLU non-linearity, we concatenate the activations from x_i and y_i and feed the output into an FC layer to obtain $f_i \in \mathbb{R}^N$.

4.2 CRF Network

On top of the feature extraction networks for each person in a frame, we build a CRF network, which is illustrated in Fig. 2. The CRF network has a data term ϕ for each person and a pairwise term ψ for each pair of the people.

Provided the output of feature extraction network f_i and corresponding label t_i, the data term is basically given by

$$\phi_{t_i}(f_i) = v_{t_i}^{\top} f_i + b_{t_i}, \tag{1}$$

where v_{t_i} is a vector in $\mathbb{R}^N$ and b_{t_i} a scalar, both of which are trainable. This term gives a larger value when $t_i = 1$ and the person f_i is likely to be important. For $t_i = 0$, it again gives a larger value when f_i is *not* likely to be important.

In order to model the correlation among people in a video frame, we concatenate the transformed features f_i and f_j to f_{ij}, and compute a pairwise term given by

$$\psi_{t_it_j}(f_{ij}) = w_{t_it_j}^{\top} f_{ij} + c_{t_it_j}, \tag{2}$$

where $w_{t_it_j}$ is a vector in $\mathbb{R}^{2N}$ and $c_{t_it_j}$ be a scalar. Various types of CRF applications like ours usually use a pairwise term that only cares if labels agree or not. In contrast, our

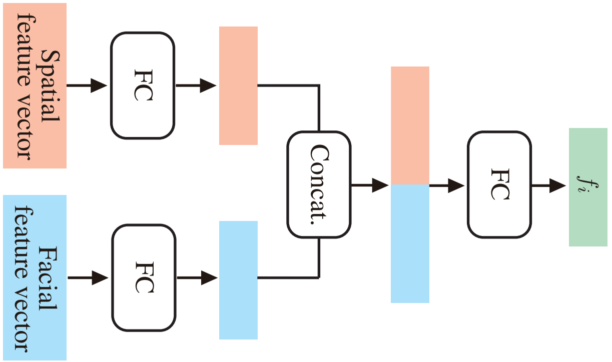

Fig. 3 Illustration of feature extraction network.

model gives different values for all of four possible combinations of labels by giving a larger value if a certain combination is likely based on the feature. Note that this term is dependent on the order of f_i and f_j; therefore, exchanging f_i and f_j as well as corresponding labels can alter the resulting value. We consider that with a sufficient number of training data, the negative effect due to this dependency is not severe.

We define an energy function $E(T, F)$ of the set of labels $T = \{t_i | i = 1, \dots, I\}$ and features $F = \{f_i | i = 1, \dots, I\}$, where I is the number of people in the frame, using the data and pairwise terms as

$$E(T, F) = \sum_i \phi_{t_i}(f_i) + \sum_{ij} \psi_{t_i t_j}(f_{ij}), \tag{3}$$

where the summations for the first and second terms are computed over all people and all combinations of people in the frame, respectively. The probabilistic interpretation can be given by

$$p(T|F) = \frac{1}{Z} e^{-E(T,F)}, \tag{4}$$

where Z is the partition function computed by

$$Z = \sum_T e^{-E(T,F)}. \tag{5}$$

The summation is calculated over all possible combinations of t_i's values. We can evaluate a certain combination of T using Eq. (3), and the important/unimportant classification is done by finding the combination that maximizes Eq. (4).

4.3 Training

The training of this network is done by minimizing the negative log-likelihood

$$-\sum_m \log p(T_m|F_m), \tag{6}$$

where T_m and F_m are the sets of labels and features in the m-th frame in the training dataset. Generally, this minimization problem is intractable and various approximation techniques, such as contrastive divergence (CD) [24] or its variant [25], are employed in existing DNN-CRF approaches.

Our problem of important/unimportant people classification, however, basically is small. Figure 4 (top) shows the distribution of the number of people in a certain frame in our training dataset. There are 16 people in a frame at most, and its mode is 1. In addition, there are only two possible labels (*i.e.*, $t_i \in \{0, 1\}$). This characteristics of our problem makes the minimization much easier because we can directly and exhaustively evaluate the partition function in Eq. (5) during training without using any approximation techniques (this also applies classification of test data because we can exhaustively evaluate Eq. (4) for all possible combinations of labels to find the best one). In the training session, we discard frames with many people (*e.g.*, more than 10). Since in our network, the pairwise term uses the same parameters regardless of the number of people, discarding such frames does not much affect the training results.

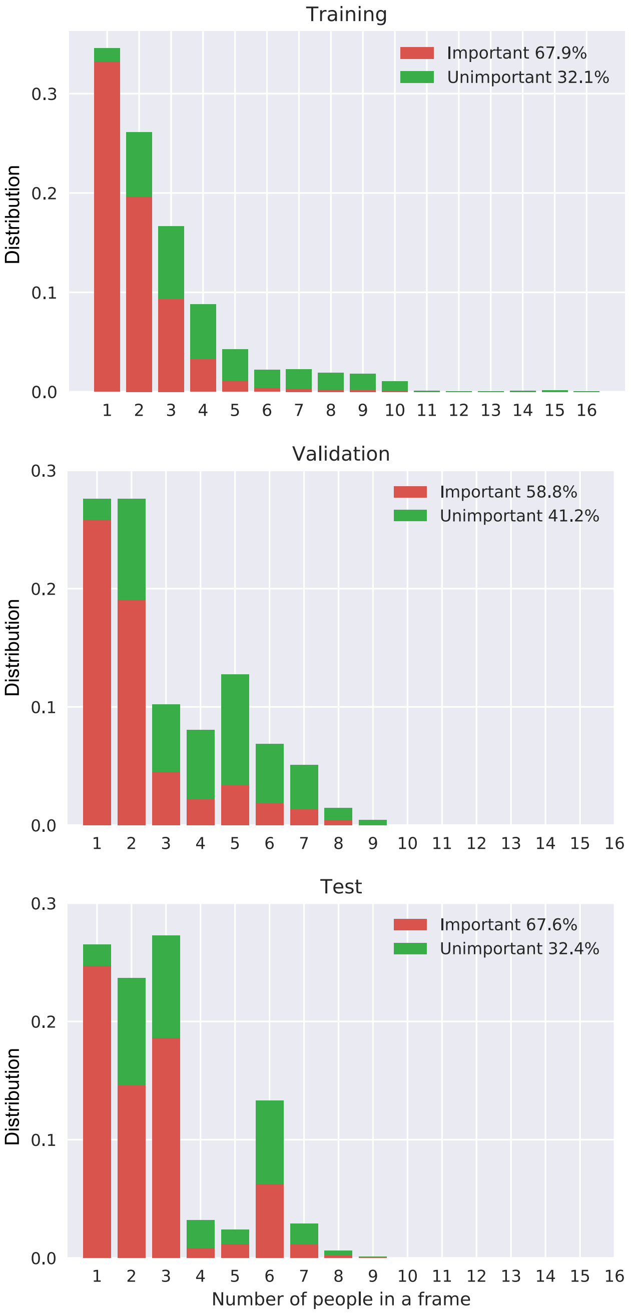

Fig. 4 The distributions of the numbers of people in a frame obtained from our (top) training, (middle) validation, and (bottom) test datasets. The number of people labeled as important is represented by red bars and unimportant by green bars.

For training, we use a variant of the stochastic gradient descent algorithm (*i.e.*, Adam [26]) and apply dropout [27] and weight decay for regularization. The evaluation of the partition function in Eq. (5) requires to compute the data and pairwise terms multiple times for the same features; therefore, instead of actually evaluating them, we store $\phi(f_i) \in \mathbb{R}^2$ and $\psi(f_{ij}) \in \mathbb{R}^4$ given by

$$\phi(f_i) = V f_i + b, \tag{7}$$

Algorithm 1 Evaluation of $p(T|F)$ in Eq. (4).

```
Input: Features $F$ and labels $T$
  Fill all entries in $\Phi$ and $\Psi$ using Eqs. (7) and (8)
  $z \leftarrow 0$
  for $T' = (t'_1, \dots, t'_I)$ in all possible combinations of $t'_i$'s do
      $e'_{\text{data}} \leftarrow \sum_i \Phi_i(t'_i)$
      $e'_{\text{pairwise}} \leftarrow \sum_{ij} \Psi_{ij}(t'_i, t'_j)$
      $e' \leftarrow e'_{\text{data}} + e'_{\text{pairwise}}$
      $z \leftarrow z + \exp(-e')$
  end for
  $e_{\text{data}} \leftarrow \sum_i \Phi_i(t_i)$
  $e_{\text{pairwise}} \leftarrow \sum_{ij} \Psi_{ij}(t_i, t_j)$
  $e \leftarrow e_{\text{data}} + e_{\text{pairwise}}$
Output: $\frac{1}{z}\exp(-e)$
```

$$\psi(f_{ij}) = W f_{ij} + c, \tag{8}$$

where $V = (v_0\ v_1)^\top$, $W = (w_{00}\ w_{01}\ w_{10}\ w_{11})^\top$, $b = (b_0\ b_1)^\top$, and $c = (c_{00}\ c_{01}\ c_{10}\ c_{11})^\top$. When calculating the data and pairwise terms, we only need to get the corresponding entries of $\phi(f_i)$ and $\psi(f_{ij})$ according to the labels.

As in Eqs. (7) and (8), $\phi_{t_i}(f_i)$ and $\psi_{t_i,t_j}(f_{ij})$ can be implemented using fully-connected layers. After computing and storing them, we can evaluate $p(T|F)$ in Eq. (4), including the partition function Z only using standard functions, which is implemented in most deep learning frameworks. Let $\Phi_i(t_i)$ and $\Psi_{ij}(t_i, t_j)$ be stored values of $\phi_{t_i}(f_i)$ and $\psi_{t_i,t_j}(f_{ij})$, respectively. That is, $\Phi_i(t_i) = \phi_{t_i}(f_i)$ and $\Psi_{ij}(t_i, t_j) = \psi_{t_i,t_j}(f_{ij})$, which emphasize that they are functions of labels for a single video frame. We can preliminarily compute all entries in Φ and Ψ for all combination of i and j. The partition function can be computed by summing up corresponding entries in Φ and Ψ with taking their exponential for a label combination, and then taking their summation for all possible combinations of labels. To compute the loss function in Eq. (6), the entries corresponding to the ground truth label T are picked up and summed up. Algorithm 1 illustrates this process to evaluate $p(T|F)$. To pick up the corresponding entries in Φ and Ψ, we can use `select_item()` in Chainer [28], for example. The other operations in the process are very common in any deep learning framework. The same process for evaluating $p(T|F)$ can be used for inferring important people as well.

5. Experimental Results

We experimentally demonstrate the merits of our model with an implementation using the Chainer framework [28]. To track facial regions for spatial features, we used the KCF tracker [29]. Each facial region in a certain frame was tracked for 100 frames in forward and backward temporal directions (*i.e.*, $L = 200$), which makes a 600-D spatial feature. As facial features, FaceNet features were adopted [23], where $K_{\text{F}} = 128$ in this case. The parameters of FaceNet are not fine-tuned for this task. For color histogram-based features, we separately generated a 50-D histogram from a facial region for each color channel (RGB) and concatenated them into a single 150-D vector (*i.e.*, $K_{\text{H}} = 150$).

5.1 Datasets

For training and testing, we used the datasets that are used in [5]. Their datasets consist of (i) 99 YouTube videos with ground truth labels assigned by multiple human annotators (each frame has several people and each of them are labeled by six annotators) and (ii) 20 videos captured by videographers and the videographers assigned the ground truth labels by themselves. All videos in both datasets were resized so that each of them has 854 pixels and 480 pixels in width and height, respectively. All videos are in approximately 30 fps. The facial regions are manually specified in all videos. Since each facial region in dataset (i) has multiple labels by different annotators, we employed majority voting to make ground truth label. The annotators were asked to infer what the videographer wanted to show. In [5], it is reported that the human annotators were able to infer the important people in a video accurately. In order to demonstrate the generalization performance of our approach, instead of cross-validation, we divided dataset (i) into two parts; one for training (66 videos) and the other for validation (33 videos). We used dataset (ii) for testing. The ground truth labels for the training and validation datasets were not assigned by the corresponding videographers; however, we consider that this is still fair since they were used solely for training and the evaluation was done by the dataset (ii).

Figure 4 shows the distributions of numbers of people in each frame. The training dataset contains frames with over 10 people, while the other datasets do not. The training dataset has 120,955 people, among which 82,079 are important (67.9%). In the test dataset, there are 55,336 people and the number of important people are 37,431 (67.6%). The numbers are not consistent with [5] because we discarded some frames to ensure all data have complete spatial features. Important and unimportant people are represented by red and green bars in the figure. We can see that most people are likely to be labeled as important when a frame shows only one person. On the other hand, a small number of people are important in frames with a lot of people. Note that the videographers who took the training dataset and test dataset are completely disjoint.

5.2 Results

To demonstrate the performance boost by our CRF-based approach, we developed variants of our full-model. We compared our DNN-CRF model with models of only DNN and DNN-CRF without pairwise term ψ. The DNN-based model consists of our feature extraction network and an FC classification layer. We train this model using the softmax cross-entropy loss. We also trained the DNN model using Eq. (6) but without the pairwise term (*i.e.*, the pairwise term is changed to always give $\psi = 0$) to examine effects of different losses. To show the direct effects of the pairwise term, we evaluated our DNN-CRF with removing the pairwise term (*i.e.*, ψ always gives 0) *after training*. We also com-

Table 1 Results on the classification of people into important or unimportant ones. We report accuracy (ACC), recall (REC), precision (PRE), false positive rate (FPR), and F1-measure (F1) for each method.

	ACC (%)	REC (%)	PRE (%)	FPR (%)	F1 (%)
Trajectories (spatial features) only					
(a) SVM	74.4	65.3	95.4	6.5	77.5
(b) SVM-CRF	77.9	76.9	88.9	20.0	82.5
(c) DNN (softmax cross-entropy loss)	80.7	80.8	89.7	19.3	85.0
(d) DNN (loss in Eq. (6))	78.6	77.4	89.6	18.8	83.0
(e) DNN-CRF (ψ removed)	74.0	98.5	72.7	77.2	83.7
(f) DNN-CRF	75.8	77.5	85.3	27.8	81.2
Trajectories with color histograms					
(g) SVM	65.3	51.3	95.1	5.6	66.7
(h) SVM-CRF	76.4	72.0	91.3	14.4	80.5
(i) DNN (softmax cross-entropy loss)	75.5	68.2	93.9	9.3	79.0
(j) DNN (loss in Eq. (6))	79.3	75.5	92.5	12.7	83.1
(k) DNN-CRF (ψ removed)	74.3	98.4	73.0	76.0	83.8
(l) DNN-CRF	**83.6**	87.8	87.9	25.3	**87.8**
Trajectories with FaceNet features					
(m) SVM	67.0	53.9	95.3	5.6	68.9
(n) SVM-CRF	78.2	77.3	89.1	19.8	82.8
(o) DNN (softmax cross-entropy loss)	77.9	73.0	92.8	11.9	81.7
(p) DNN (loss in Eq. (6))	78.8	79.5	88.1	22.4	83.6
(q) DNN-CRF (ψ removed)	76.9	96.9	75.7	65.1	85.0
(r) DNN-CRF	81.0	82.1	89.0	21.3	85.4
random sampling	63.3	69.8	74.4	50.2	72.0

pared our approach to raw support vector machine decisions (SVM) and the previous work in [20] (SVM-CRF), which uses a support vector machine to obtain decision values and applies CRF to them together with features. Since the results in [20] shows that the improvement by the temporal consistency term in their model is not very large, we employed an SVM-CRF model simplified by removing the temporal consistency term. We tuned the hyperparameters of our DNN-based models and SVM-based models (*i.e.*, learning rate, dropout ratio, weight decay ratio, and unit size of hidden layer N for DNN-based models, and γ and C of SVM with the radial basis function) with Bayesian optimization. For each approach, we evaluated up to 100 combinations of hyperparameters and picked the best model.

As shown in Fig. 4, the numbers of important and unimportant people in a frame is biased, which can be a strong prior about the importance of people. People are likely to be important when there are few people in a frame. Therefore, as a baseline, we also report the performance of randomly sampled labels from the distributions of important and unimportant people in the frame. We computed the distributions of important and unimportant people over the number of people in a frame on the training set. To evaluate the performance, given a frame, we sample labels from the marginal distribution given the number of people in the frame. For example, the probability of the label being "important" is 96% if there is only one person in a frame and 26% if there are five.

Table 1 shows the results in accuracy, recall (or true positive rate), precision, false positive rate, and F1-measure. The baseline method, which randomly produces a label based on the number of people in a frame, achieved 63.3% of accuracy and 72.0% of F1-measure. Regarding both accuracy and F1-measure, the DNN-CRF model using trajectories and color histograms (l) performed the best among all. Among models using trajectories with the FaceNet features, our full-model, DNN-CRF, also achieved high scores in accuracy and F1-measure. Comparing the results of DNN models using different losses, we did not observe consistent effects. For color histograms and FaceNet features, the comparison between DNN-CRF with and without pairwise term ψ (k)–(l) and (q)–(r) demonstrated that the pairwise term improved the performance.

Regarding the FPR scores, the data term looks to be biased towards assigning label "important," and the pairwise term is learned to prevent this. For the accuracy and F1-measure, although the use of the pairwise term showed positive effects in models using color histograms and FaceNet features ((j) and (p) outperformed (i) and (o)), the model using only trajectories did not show similar effects ((d) did not outperform (c)). The best model among ones that uses only trajectories is the naive DNN model (c), which achieved 80.7% of accuracy. Moreover, even the simple SVM model (a) still worked well. These results suggest that the input trajectories, which is a 600-D feature vector, contain rich information and do not require complicated models to predict the importance of people. Although (c) DNN model using only trajectories worked well, incorporating additional input features and CRFs demonstrated further improvement on this task.

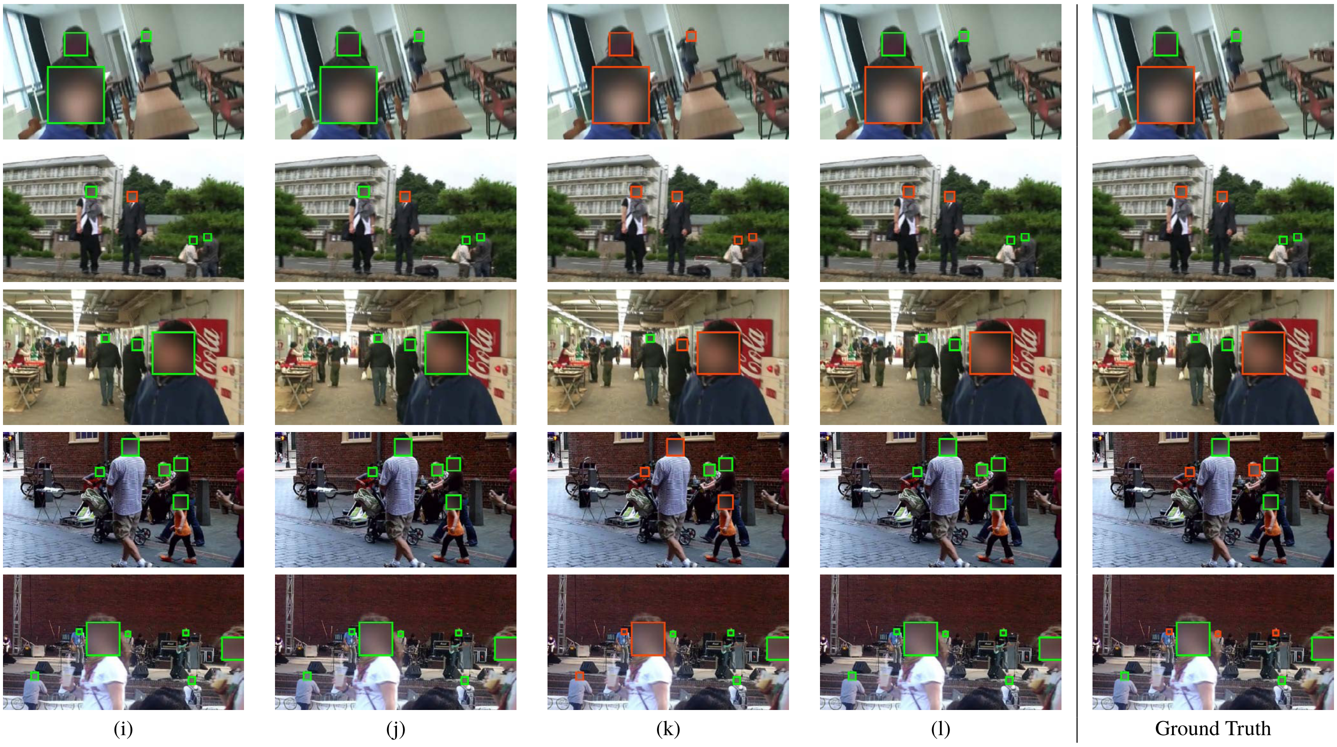

Fig. 6 Examples of classification results. Each column is results of a DNN-based approaches using trajectories and color histograms in Table 1 (i.e., (i), (j), (k), and (l)) and ground truth (right). Red and green indicate important and unimportant person, respectively. The facial regions are blurred for privacy reasons.

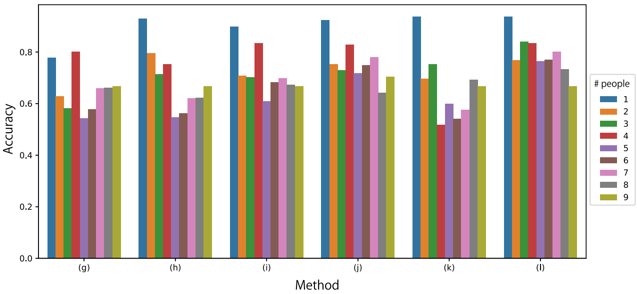

Fig. 5 Accuracy with respect to the number of people in a video frame.

Figure 5 shows the accuracy with respect to the number of people, evaluated over the test dataset for the methods that use trajectories and color histograms as features (*i.e.*, methods (g)–(l)) since one of them performed the best. According to the figure, the accuracy is particularly high when the number of people is one. This is because our model is biased towards giving label "important" and the CRF suppress it when there are more than one people. Also, if there are only one person in a video frame, the person is important in most cases in our datasets. We may see slight drop in accuracy for (l) when the number of people is nine; however, the accuracy value may not be stable for because the number of frames with nine people in them are relatively rare.

Figure 6 shows some success (the first, second, and third rows) and failure (the fourth and fifth row) examples. For the top three examples, our DNN-CRF model successfully distinguished the important people from unimportant ones. Comparing the results by (k) DNN-CRF (ψ removed) and (l) DNN-CRF, we can observe that our model without the pairwise term is biased towards predicting people as important, and the pairwise term is helpful to correct the prediction by assigning different labels to people whose trajectories or facial features are significantly different to each other.

However, our model failed in the fourth row, which shows two men playing the guitars on the street and some people crossing in front of them. We can see both important and unimportant person in the result of the CRF-DNN model without the pairwise term, but our full-model failed and assigned "unimportant" labels for all people. This frame has two group of people (the pedestrians and the musicians). We consider that our full-model learned that the group of people whose trajectories tend to be similar have the same label, and thus it tried to assign the same label to these groups of people. Due to small face sizes of musicians, our model is not very confident about the "important" label assigned to one of the musicians, and this resulted in failure. In the bottom row, ours failed to assign the "important" label for a band on the stage. We consider that the frame is a hard example: The faces of important people are very small in the frame and we assume that these small face regions make it difficult to correctly classify people as important since most important people in the dataset are shot to occupy a larger region.

6. Conclusion

We have presented a model for classifying people in a frame into important and unimportant using a CRF model built on top of DNNs. Based on the observation that spatial trajectories of face regions in a video, which represent the videographers intension, provides strong cues to estimate the importance of the people, we use the trajectories as input for our model. Since this classification problem is not big (*i.e.* only a few people appear in a frame and there are only two possible labels), we can directly minimize the negative log-likelihood during the training process, in which the partition function can be exhaustively evaluated. We tested our DNN-CRF model with using either color histograms and face recognition features based on FaceNet as additional features, and one with color histogram outperformed other baselines in accuracy and F1-measure. Our ablation study demonstrated that considering relations between people in a frame improves the classification accuracy. We expect that face orientation and facial expressions are one of informative cues for this task. Therefore, adopting other features related to face orientations or other deep facial features would be an interesting future direction.

Acknowledgements

This work was partly supported by JSPS KAKENHI No. 16K16086 and No. 16H06302.

References

[1] F. Liu and M. Gleicher, “Video retargeting: Automating pan and scan,” ACM International Conference on Multimedia (MM), pp.241–250, 2006.

[2] L. Itti, “Automatic foveation for video compression using a neurobiological model of visual attention,” IEEE Trans. Image Process., vol.13, no.10, pp.1304–1318, 2004.

[3] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” IEEE Trans. Pattern Anal. Mach. Intell., vol.20, no.11, pp.1254–1259, 1998.

[4] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhang, “A generic framework of user attention model and its application in video summarization,” IEEE Trans. Multimedia, vol.7, no.5, pp.907–919, 2005.

[5] Y. Nakashima, N. Babaguchi, and J. Fan, “Intended human object detection for automatically protecting privacy in mobile video surveillance,” Multimedia Systems, vol.18, no.2, pp.157–173, 2012.

[6] V. Ramanathan, B. Yao, and L. Fei-Fei, “Social role discovery in human events,” IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp.2475–2482, 2013.

[7] M. Andriluka, S. Roth, and B. Schiele, “Pictorial structures revisited: People detection and articulated pose estimation,” IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp.1014–1021, 2009.

[8] Y.-C. Su, D. Jayaraman, and K. Grauman, “Pano2Vid: Automatic cinematography for watching 360° videos,” Asian Conference on Computer Vision (ACCV), pp.154–171, 2016.

[9] L. Itti and P. Baldi, “Bayesian surprise attracts human attention,” Vision. Res., vol.49, no.10, pp.1295–1306, 2009.

[10] S. Frintrop, G. Backer, and E. Rome, “Goal-directed search with a top-down modulated computational attention system,” in DAGM Conference on Pattern Recognition, pp.117–124, 2005.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, and L. Fei-Fei, “ImageNet large scale visual recognition challenge,” International Journal of Computer Vision, vol.115, no.3, pp.211–252, 2015.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick, “Microsoft COCO: Common objects in context,” European Conference on Computer Vision (ECCV), pp.740–755, 2014.

[13] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recoginition,” International Conference on on Learning Representations (ICLR), pp.1–14, 2015.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1–9, 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.770–778, 2016.

[16] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan, “Semantic object parsing with graph LSTM,” European Conference on Computer Vision (ECCV), pp.125–143, 2016.

[17] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P.H.S. Torr, “Conditional random fields as recurrent neural networks,” IEEE International Conference on Computer Vision (ICCV), pp.1529–1537, 2015.

[18] A. Arnab, S. Jayasumana, S. Zheng, and P.H.S. Torr, “Higher order conditional random fields in deep neural networks,” European Conference on Computer Vision (ECCV), pp.524–540, 2016.

[19] S. Chandra and I. Kokkinos, “Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian CRFs,” European Conference on Computer Vision (ECCV), pp.402–418, 2016.

[20] Y. Nakashima, N. Babaguchi, and J. Fan, “Privacy protection for social video via background estimation and CRF-based videographer’s intention modeling,” IEICE Transactions on on Information and Systems, vol.E99-D, no.4, pp.1221–1233, 2016.

[21] X. Ma and E. Hovy, “End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF,” Annual Meeting of the Association for Computational Linguistics (ACL), pp.1064–1074, 2016.

[22] X. Chu, W. Ouyang, H. Li, and X. Wang, “CRF-CNN: Modeling structured information in human pose estimation,” Conference on Neural Information Processing Systems (NIPS), pp.316–324, 2016.

[23] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.815–823, 2015.

[24] G.E. Hinton, “Training products of experts by minimizing contrastive divergence,” Neural Computation, vol.14, no.8, pp.1771–1800, 2002.

[25] A. Kirillov, D. Schlesinger, S. Zheng, B. Savchynskyy, P.H.S. Torr, and C. Rother, “Joint training of generic CNN-CRF models with stochastic optimization,” Asian Conference on Computer Vision (ACCV), pp.221–236, 2016.

[26] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” International Conference on Learning Representations (ICLR), 13 pages, 2015.

[27] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” arXiv preprint arXiv:1207.0580, 18 pages, 2012.

[28] S. Tokui, K. Oono, S. Hido, and J. Clayton, “Chainer: A next-generation open source framework for deep learning,” Conference on Neural Information Processing Systems (NIPS), 6 pages, 2015.

[29] J.F. Henriques, R. Caseiro, P. Martins, and J. Batista, “Exploiting the circulant structure of tracking-by-detection with kernels,” European Conference on Computer Vision (ECCV), pp.702–715, 2012.

Mayu Otani received her B.E. from Kyoto University in 2013 and the M.E. from Nara Institute of Science and Technology in 2015. She was a visiting scholar at the University of Oulu in 2015. She received her Ph.D. from Nara Institute of Science and Technology in 2018. She is a research scientist at CyberAgent, Inc. since 2018.

Atushi Nishida received a B.E in Information Engineering from Okayama Prefectural University, Okayama, Japan in 2015 and his M.E. in information sciences from Nara Institute of Science and Technology, Nara, Japan in 2017, respectively. He is currently with Dai Nippon Printing Co., Ltd.

Yuta Nakashima received a B.E. and a M.E. and a Ph.D. from Osaka University, Osaka, Japan in 2006, 2008, and 2012, respectively. He was an assistant professor at Nara Institute of Science and Technology from 2012 to 2016 and is currently an associate professor at the Institute for Datability Science, Osaka University. He was a visiting scholar at UNCC in 2012 and at CMU from 2015–2016. His main research focus includes video content analysis using machine learning approaches. He is a member of IEEE, ACM, IEICE, and IPSJ.

Tomokazu Sato received a B.E. in computer and system science from Osaka Prefecture University in 1999. He received a M.E. and a Ph.D. in information sciences from Nara Institute of Science and Technology (NAIST) in 2001 and 2003, respectively. He was an assistant professor at NAIST during 2003–2011. He served as a visiting researcher at Czech Technical University in Prague from 2010–2011. He was an associate professor at NAIST from 2011 to 2017. He is a professor at Shiga University since 2018. His research interests include computer vision and mixed reality.

Naokazu Yokoya received a B.E., a M.E., and a Ph.D. in information and computer sciences from Osaka University in 1974, 1976, and 1979, respectively. He joined the Electrotechnical Laboratory (ETL) at the Ministry of International Trade and Industry in 1979. He was a visiting professor at McGill University in 1986–1987 and was a professor at Nara Institute of Science and Technology (NAIST) during 1992–2017. He is the president at NAIST since April 2017. His research interests include image processing, computer vision, and mixed and augmented reality. He is a life senior member of IEEE.