Training Efficient Saliency Prediction Models with Knowledge Distillation

Peng Zhang^{1,2}, Li Su^{1,2,3*}

Liang Li³, BingKun Bao⁴, Pamela Cosman⁵, GuoRong Li^{1,2,3}, Qingming Huang^{1,2,3}

¹School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China

²Key Lab of Big Data Mining and Knowledge Management, University of Chinese Academy of Sciences, Beijing, China

³Key Lab of Intelligent Information Processing (IIP), Institute of Computing Technology, Beijing, China

⁴Nanjing University of Posts and Telecommunications, Nanjing, China

⁵University of California, San Diego, United States

zhangpeng172@mails.ucas.ac.cn, sulil@ucas.ac.cn, liang.li@vpl.ict.ac.cn

bingkunbao@njupt.edu.cn, pcosman@eng.ucsd.edu, liguorong@ucas.ac.cn, qmhuang@ucas.ac.cn

ABSTRACT

Recently, deep learning-based saliency prediction methods have achieved significant accuracy improvements. However, they are hard to embed in practical multimedia applications due to large memory consumption and running time caused by complicated architectures. In addition, most methods are fine-tuned from pre-trained models for classification tasks, and networks cannot flexibly be transferred for a new task. In this paper, a condensed and randomly initialized student network is employed to achieve higher efficiency by transferring knowledge from complicated and well-trained teacher networks. This is the first use of knowledge distillation for efficient pixel-wise saliency prediction. Instead of directly minimizing Euclidean distance between feature maps, we propose two statistical representations of feature maps (i.e., first-order and second-order statistics) as knowledge. We conduct experiments on three kinds of teacher networks and four benchmark datasets to verify the effectiveness of the proposed method. Compared with the teacher networks, the student networks achieve an acceleration ratio of 4.56~4.73. Compared with state-of-the-art approaches, the proposed model achieves competitive accuracy with faster running speed (up to 4.38 times) and smaller model size (up to 93.27% reduction). We further embedded the proposed saliency prediction model into a video captioning application. The saliency-embedded approaches improve video captioning on all test metrics with a small complexity cost. The student-model embedded approach achieves 25% time saving with similar performance to the teacher embedded one.

CCS CONCEPTS

• **Human-centered computing** → *Human computer interaction (HCI)*; • **Computing methodologies** → **Computer vision**.

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '19, October 21–25, 2019, Nice, France

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6889-6/19/10...\$15.00

<https://doi.org/10.1145/3343031.3351089>

KEYWORDS

saliency prediction, knowledge transfer, first-order statistics, second-order statistics, video caption

ACM Reference Format:

Peng Zhang, Li Su, Liang Li, BingKun Bao, Pamela Cosman, GuoRong Li, and Qingming Huang. 2019. Training Efficient Saliency Prediction Models with Knowledge Distillation. In *Proceedings of the 27th ACM International Conference on Multimedia (MM'19)*, Oct. 21–25, 2019, Nice, France. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3343031.3351089>

1 INTRODUCTION

Saliency prediction aims to identify the most salient objects or regions. The goal is to simulate the astonishing capability of human attention in prioritizing objects for high-level processing [1, 2]. As shown in Fig. 1, when the HVS (human visual system) perceives a scene, objects or regions in the scene are not perceived with the same importance. Saliency prediction is embedded in many practical applications such as image captioning [3–5], visual question answering [6], video compression [7], image classification [5], image retargeting [8, 9], object detection [10–12], and video captioning [13, 14]. In Fig. 1, saliency prediction is used to predict which parts of the frames are salient. The visually salient region usually contains the main information of the content. Currently, some methods introduce attention modules [15–17] and related handcrafted features to improve multimedia tasks. However, the accuracy of methods with handcrafted features is often inadequate. Saliency prediction models based on deep learning might achieve significant accuracy improvement, but most of them require a large amount of memory and time consumption due to complicated architectures [18, 19]. Fig. 2 shows four types of saliency prediction models which represent four categories: hyper feature learning [20], fully convolutional network (FCN) [21], HED [22] and recurrent refinement network [23]. Most recent saliency models fall into these categories. They apply different fusion strategies, which lead to complex structures. Therefore, this paper proposes to train a smaller saliency prediction model to reduce the complexity, aimed at multimedia applications.

First, one must choose a small model. Saliency prediction is different from object detection. The output is a binarized segmentation image with the same size as the original image. In order to obtain more refined results, it is necessary to combine low-level features

Figure 1: Saliency-based video captioning. Saliency prediction aims to simulate the capability of human attention and identify the most salient regions.

to capture fine-grained details. Traditional semantic segmentation does not distinguish between foreground and background. But saliency prediction needs to locate salient regions and only segment salient objects. Therefore, in order to better combine high-level and low-level features, we consider reducing the number of network channels to get a smaller model, rather than making changes in the network structure.

Secondly, nearly all existing saliency prediction frameworks rely on a pre-trained model on an annotated dataset such as ImageNet for initialization. If we reduce the number of channels, we need to pre-train the model for the classification task and then fine-tune it for the saliency prediction task. Unfortunately, training a classification model on ImageNet is highly resource intensive [24]. If we train such a saliency prediction model from scratch, it leads to poor performance. Inspired by the learning process of human beings, we adopt a teacher-student strategy (i.e., knowledge distillation) to ease the training of the small model. The small student models will learn better and faster with the guidance of teacher models.

Thirdly, although the spirit of knowledge distillation has been shared in many other computer vision tasks (e.g. object detection [25] and pedestrian detection [26]), these non-pixel-wise tasks typically directly minimize Euclidean distance of feature maps between teacher and student models. There are two limitations: (1) To calculate Euclidean distance, transformation modules are needed to resolve the inconsistent dimensions of feature maps for teacher and student networks. This increases the training complexity. (2) Optimizing CNNs directly in a high dimensional feature space is a hard problem for randomly initialized models [25]. In this paper, we consider two statistical representations of feature maps as knowledge. Specifically, to learn better representations for student models, we exploit first-order and second-order statistics, which represent, respectively, pixel-wise information of salient regions in each layer and correlation between pixels. For saliency prediction, pixel-wise information will help the model predict the salient region, and the correlation between pixels will help the model predict details. Our contributions in this paper are summarized as follows:

- (1) We introduce knowledge distillation to train an efficient student network for saliency prediction. Instead of being initialized by fine-tuning, the proposed model is small and randomly initialized. This increases the flexibility of the model. To the best of our knowledge, this is the first usage of knowledge distillation for the task of efficient pixel-wise saliency prediction.
- (2) We combine first-order and second-order statistics transfer to learn better representations for student networks. First-order statistics represent the pixel-wise information of salient regions in each layer. Second-order statistics represent the correlation between pixels. These two types of knowledge work as complementary support.
- (3) We conduct experiments on different teacher networks to verify the effectiveness of the proposed methods. Compared with the teacher networks, the student networks achieve an acceleration ratio of 4.56~4.73. Compared with state-of-the-art approaches, the proposed model achieves competitive accuracy with faster running speed (up to 4.38 times) and smaller model size (up to 93.27% reduction).
- (4) We further embedded the proposed saliency prediction model into a video caption application. The saliency-embedded approaches improve video captioning on all test metrics with acceptable over cost.

2 RELATED WORK

2.1 Saliency Prediction

In the past decade, there are a number of approaches for saliency prediction. Early approaches estimate salient values based on hand-crafted features [27]. However, hand-crafted features and priors have limited ability to represent features, which makes it difficult to capture high-level semantic meanings of salient objects.

In recent years, CNNs [24] have raised the bar in performance for many multimedia applications such as salient object prediction. Some exploit superpixel and object region-based models to achieve accurate salient object prediction. Such methods follow a multi-branch architecture where a CNN is used to extract semantic information across different levels of abstraction to generate an initial saliency prediction.

Fully CNN-based algorithms [21] have achieved remarkable performance on saliency prediction due to strong feature representation ability. For example, Zhao, et al. [28] used FCN to get the features of a full image to model the global context and the features of a part of an image to model the local context. Luo, et al. [29] proposed a multi-resolution grid structure to capture both local and global cues. Zhang, et al. [30] learned deep uncertain convolutional features by a dropout technique that randomly drops units of the deep network during training.

Other methods use an end-to-end encoder-decoder architecture that produces an initial coarse saliency map and then refines it stage-by-stage to provide better localization of salient objects [31, 32]. Deng, et al. [23] propose a novel recurrent residual refinement network (R³Net) equipped with residual refinement blocks (RRBs) to more accurately predict salient regions of an input image. Chen, et al. [33] employ residual learning to learn side-output residual features for saliency refinement and propose reverse attention to

Figure 2: Illustration of deep learning architectures: (a) hyper feature learning; (b) FCN style; (c) HED style; (d) recurrent refinement network

guide such side-output residual learning in a top-down manner. All of them aim to explore different combinations of features at multiple layers to integrate salient visual cues from multiple views. However, most of them require a large amount of memory and time due to complicated architectures.

2.2 Knowledge Distillation

CNNs are expensive in terms of computation and memory costs. But efficient models are particularly important when running on resource-limited embedded devices such as mobile phones and digital cameras. To reduce run-time memory costs and execution time for each single inference, several approaches have been proposed for model compression and acceleration. Knowledge Distillation (KD) [34] is one of the effective methods. KD trains a student model using a teacher model. Predictions from the teacher network, called softened outputs, are often similar to one-hot hard targets. The softened outputs provide guidance in training student networks. FitNets [35] adds additional regression layers to make the student model mimic the intermediate feature maps from the teacher model. To avoid mimicking the full feature map directly, [36] proposed to mimic the attention map.

In this paper, we use “knowledge distillation” to refer to the above process. Unlike early KD and FitNets, we do not add additional new layers to mimic the intermediate feature maps directly. We exploit first-order statistics to learn better representations of pixel-wise knowledge. Considering that the output of saliency prediction is a binarized segmentation image, we also exploit second-order statistics to learn better representations of knowledge between pixels. This is the first usage of KD for efficient pixel-wise saliency prediction.

3 PROPOSED METHOD

We propose an efficient saliency prediction framework that makes use of KD to help a randomly initialized student model mimic a well-trained teacher model. We introduce the framework in Section 3.1, and describe two feature map KD methods that facilitate better

and faster learning of student networks in Section 3.2. A detailed description of optimization is provided in Section 3.3.

3.1 Overall Framework

The training framework is shown in Fig. 3. Saliency prediction models are based on an encoder-decoder structure to generate a saliency map of the same size as the original image. We use the encoder-decoder structure to represent the teacher and student network. The proposed method consists of two stages. The first stage is knowledge distillation. We use a fine-tuned teacher network and randomly initialized small student network to extract image feature maps. Then we use the KD module to make the features maps of each block between the two networks approach each other. We use the output of the teacher network as a soft target, rather than the ground-truth. This helps the student network mimic a well-trained teacher model better. The number of channels of the student’s feature maps are half that of the teacher’s. So the number of parameters of the student network is lower and the calculation cost is reduced. Details are in the following section. In the second stage, we make the student network learn by itself, called self-learning. Although the student network is thinner than the teacher network, it can learn some knowledge by itself.

3.2 Knowledge Distillation Module

3.2.1 First-order Statistics Transfer. As mentioned in Section 1, instead of directly minimizing Euclidean distance, we seek statistical representations of feature maps for two reasons. First, we set up a student network with fewer channels (such as 1/2 or 1/4) in each layer compared with the teacher network. To calculate Euclidean distance, the dimensions of feature maps for teacher and student networks need to be consistent. We inevitably need to add new layers to adjust the number of channels for the computation of Euclidean distance [25], which increases the difficulty of learning. Secondly, it is difficult to perform regression on two high-dimensional feature maps directly.

In this paper, we squeeze global channel information into a spatial descriptor to obtain the first-order statistics. This is achieved

Figure 3: Our framework consists of two stages. In the knowledge distillation stage, we fix the fine-tuned teacher model and train the randomly initialized small student network with transfer loss along with binary cross entropy loss. The transfer loss is generated in the KD module. The student’s feature maps have half as many channels as the teacher’s. In the self-learning stage, we train the student network with ground-truth samples with manual annotation. ‘KD’ means the knowledge distillation module shown in Fig. 5.

by using global max pooling to generate the first-order statistics maps as the knowledge to transfer from the teacher network to the student network for saliency prediction. As illustrated in Fig. 4, different layers present different statistical information.

The feature maps of student network and teacher network are denoted as $F_S \in R^{c_s \times (h \times w)}$ and $F_T \in R^{c_t \times (h \times w)}$, where c_s/c_t , h and w are the numbers of channels (student/teacher), and height and width of the feature map, respectively. As mentioned above, $c_s < c_t$. We adopt global max pooling along the channel dimension as the mapping function to compute $A_S \in R^{h \times w}$ and $A_T \in R^{h \times w}$:

$$A_S(i, j) = \max_{c=0}^{c_s} F_S(c, i, j) \quad (1)$$

$$A_T(i, j) = \max_{c=0}^{c_t} F_T(c, i, j) \quad (2)$$

where $F_S(c, i, j)$ and $F_T(c, i, j)$ are the pixel values of the c -th channel at (i, j) . As visualized in Fig. 4, low-level feature maps tend to have higher activations for the edge of both foreground and background regions, while high-level feature maps attend to the discriminative regions around foreground objects. Then the first-order statistics transfer loss is defined as:

$$\mathcal{L}_{ft}(W_S) = \sum_{k=1}^M \frac{1}{hw} \left\| (\text{vec}(A_S^k))^{\text{norm}} - (\text{vec}(A_T^k))^{\text{norm}} \right\|_2 \quad (3)$$

where M is the number of first-order statistics map pairs and $(\text{vec}(A_S^k))^{\text{norm}}$ is the normalized $\text{vec}(A_S^k)$. We use l_2 -normalization before calculating loss.

3.2.2 Second-order Statistics Transfer. For saliency prediction, the correlation information between pixels is important. Therefore, we model the knowledge to be transferred with a correlation matrix to capture the second-order statistical information.

Figure 4: First-order statistics maps of two example images. Different layers show different statistics maps. The student network mimics the statistics maps of the teacher network.

Following the notation above, we assume it is the same spatial size for the student and teacher networks. The parameter-free interpolation module can be used if the spatial size of student and teacher networks are different. Then the correlation matrices for student and teacher networks are defined as:

$$C_S = \frac{1}{c_s hw} (F_S^{\text{norm}})^{\text{transpose}} \cdot F_S^{\text{norm}} \quad (4)$$

$$C_T = \frac{1}{c_t hw} (F_T^{\text{norm}})^{\text{transpose}} \cdot F_T^{\text{norm}} \quad (5)$$

where F_S^{norm} (F_T^{norm}) is the normalized F_S (F_T). Note the normalization is important to make the training process stable due to the different scales of feature maps of student and teacher networks. We adopt l_2 -normalization along the channel dimension. Given $C_S \in R^{(h \times w) \times (h \times w)}$ and $C_T \in R^{(h \times w) \times (h \times w)}$, the transfer loss is

calculated as

$$\mathcal{L}_{st}(W_S) = \sum_{k=1}^M \frac{1}{hw \times hw} \|C_S^k - C_T^k\|_2 \quad (6)$$

Figure 5: An illustration of the proposed KD methods. This module consists of two parts: (a) First-order statistic transfer (b) Second-order statistic transfer. As shown in Fig. 3, we generate the feature maps of teacher and student networks in each block as the input of the KD module. The two parts of the module transfer knowledge from teacher’s feature maps to student’s feature maps.

3.3 Optimization

The training process is divided into two stages, as shown in Fig. 3. **Stage 1: Knowledge Distillation.** The first stage is trained using a weighted combination of the transfer loss and the binary cross entropy loss. We define the loss $\mathcal{L}(W_S)$ for the first stage as follows:

$$\mathcal{L}(W_S) = \alpha \mathcal{L}_{ce}(W_S) + \beta_{ft} \mathcal{L}_{ft}(W_S) + \beta_{st} \mathcal{L}_{st}(W_S) \quad (7)$$

where W_S denotes the weights of the student network. $\mathcal{L}_{ce}(W_S)$, $\mathcal{L}_{ft}(W_S)$ and $\mathcal{L}_{st}(W_S)$ represent the binary cross entropy loss, first-order transfer loss, and second-order transfer loss, respectively. As mentioned above, $\mathcal{L}_{ft}(W_S)$ and $\mathcal{L}_{st}(W_S)$ encourage the student network to mimic the teacher’s feature maps. α , β_{ft} and β_{st} are hyper-parameters to control the losses to be of the same order of magnitude. $\mathcal{L}_{se}(W_S)$ indicates the loss between the predictions of the student and teacher networks as follows:

$$\mathcal{L}_{ce}(W_S) = -\frac{1}{N} \sum_{i=1}^N \sum_{c=0}^1 \mathbf{1}(y_i = c) \log p(\hat{y}_i = c | W_S) \quad (8)$$

where N is the number of pixels contributed to the loss, $\mathbf{1}$ is the indicator function, and y_i and $\hat{y}_i$ denote the sigmoid activation of the student’s output.

Stage 2: Self-learning. Although the student network is thinner than the teacher network, it can also learn some knowledge by itself. So to create a better student network, we will use the binary cross entropy loss to define the loss $\mathcal{L}_{sl}(W_S)$ for self-learning:

$$\mathcal{L}_{sl}(W_S) = -\frac{1}{N} \sum_{i=1}^N \sum_{c=0}^1 \mathbf{1}(g_i = c) \log p(\hat{y}_i = c | W_S) \quad (9)$$

where g_i represents the ground-truth of the i -th image.

4 EXPERIMENTS AND RESULTS

We use FCN and U-shape networks to verify the effectiveness of the proposed methods. The structures are shown in Fig. 6. For FCN, we use dilated convolution [37] in block conv5 to augment the receptive field size without losing resolution. We obtain the output with spatial reduction of eight in total, followed by a bilinear up-sample module to recover to the input image’s size. For the U-shape network, we first reduce the number of feature map channels in each block to the same number (e.g. 32) through 1×1 convolution, which substantially reduces computational requirements. Secondly, we insert 1×1 convolution before each element-wise summation procedure, which allows feature maps from different levels to aggregate more suitably.

For the FCN and U-shape networks, training is performed using standard Stochastic Gradient Descent (SGD) with momentum 0.9 and base learning rate 0.01. The input images are resized to 321×321 . The learning rate is reduced by a ratio of 0.1 after every 20 epochs. The batch size is 4 at the transfer stage and 8 at the self-learning stage. In Equation (1), β_{ft} and β_{st} are set to 4 and 0.001, respectively. α is fixed to 1. For R^3 Net, the parameter is the same as [23]. We use VGG16 as the basic network instead of ResNeXt. CRF [23] is used as the post-processing module. Since the second-order statistics transfer experiment needs more memory at the training stage, we adopt conv2, conv3, conv4 and conv5 layers to transfer for FCN and R^3 Net. We adopt conv3, conv4, decoder block 3 and decoder block 4 to transfer for the U-shape network in our experiments. Experiments are conducted in the PyTorch framework with a GTX 1080Ti GPU and an Intel i7-6700 CPU (3.4 GHz).

4.1 Datasets and Evaluation Metrics

We train the FCN [21] and U-shape networks [38] on THUS10K [39], which contains 10,000 images. The training dataset for R^3 Net [37] is the same as the compared methods: MSRA-B [28], which contains 5,000 images. We test on four benchmark datasets. ECSSD [40] contains 1,000 semantically meaningful but structurally complex natural images. HKU-IS [41] contains more than 4000 challenging images. PASCAL-S [42] with 850 images and DUT-OMRON [43] with 5,168 images are popular datasets as well.

We adopt Frames per Second (FPS), F-measure (F_β) and mean absolute error (MAE) as our evaluation metrics [27]. MAE is calculated as the average absolute difference between ground-truth and predicted saliency map. F-measure is a weighted harmonic mean of Precision and Recall with a non-negative weight. FPS is calculated with 256×256 resolution. A better saliency detector should have a larger FPS, a larger F-measure (F_β), and a smaller MAE.

Figure 6: Illustration of FCN and U-shape networks. (a) FCN with dilated convolution. (b) U-shape network with bottom-up and top-down branches.

4.2 Analysis of Knowledge Distillation.

Effectiveness of knowledge distillation: To validate the effectiveness of KD, we perform experiments on ECSSD using learning from scratch and KD strategies. The results are shown in Fig. 7. This comparison is based on FCN. The student model trained by KD achieves more similar performance to the teacher network than the model trained from scratch. In addition, the former speeds up the convergence during the beginning of training. The comparative experiments indicate that knowledge from a teacher network can effectively help the small student network converge to a better local minimum.

Complementary advantages of the first-order transfer and the second-order transfer: Results in Table 2 demonstrate the effectiveness of KD for both of these architectures. We adopt 1/2 and 1/4 width of the teacher network, where 1/2 (1/4) represents the student network that has same depth but each convolutional layer only has 1/2 (1/4) of the number of filters of the teacher network. From the results, combining the first-order statistics transfer and the second-order statistics transfer, we obtain the best performance of the student models. These two types of knowledge work as complementary support. The 1/2 width of FCN gets a higher accuracy and faster speed than FCN. The student networks of the U-shape that only transferred high-level features are faster than the teacher network with only a slight decrease in performance. Overall, large well-trained teacher models perform a little better than smaller student models with random initialization, while student models run significantly faster.

Speed-up of student network with different widths: Not surprisingly, 1/2-VGG, which contains 1/2 as many parameters as VGG, runs faster with a 2x speed-up. The 1/4-VGG, which has only about 1/4 of the parameters of VGG, runs at 4x speed-up with slight worse performance than 1/2-VGG.

4.3 Ablation Study

The effectiveness of training schemes: We study three schemes to train the models. In the first and second schemes, we train the student model in one stage without self-learning. In the first stage, we use ground-truth to calculate the loss $\mathcal{L}_{st}(W_S)$ instead of the teacher's output. In the third scheme, we train the student model in two stages using teacher's output and self-learning. The results are

Figure 7: F-measure versus number of epochs on the ECSSD dataset: Training small model from scratch (bottom curve); Transferring knowledge from teacher to student network (middle curve); Fine tuning the teacher network from ImageNet pre-trained model (top curve).

Table 1: Evaluation of different components of the proposed method on two benchmark datasets. F, U, FT and ST stand for FCN, U-shape network, first-order statistics transfer and second-order statistics transfer respectively. FS means using KD and self-learning.

Methods		FPS	ECSSD		HKU-IS	
			MAE↓	F_{β} ↑	MAE↓	F_{β} ↑
F	Teacher	97	0.0788	0.8199	0.0622	0.8073
	1/2-FT	218	0.0824	0.8144	0.0643	0.7943
	1/2-ST	217	0.0818	0.8181	0.0655	0.7988
	1/2-FS	217	0.0775	0.8269	0.0628	0.8056
	1/4-FT	443	0.0942	0.7931	0.0762	0.7677
	1/4-ST	442	0.0905	0.7972	0.0757	0.7696
	1/4-FS	442	0.0868	0.8066	0.0726	0.7783
U	Teacher	80	0.0637	0.8586	0.0501	0.8436
	1/2-FT	182	0.0765	0.8306	0.0638	0.8075
	1/2-ST	183	0.0779	0.8279	0.0651	0.8027
	1/2-FS	182	0.0722	0.8402	0.0604	0.8155
	1/4-FT	378	0.0874	0.8041	0.0751	0.7766
	1/4-ST	378	0.0902	0.7975	0.0764	0.7735
	1/4-FS	378	0.0849	0.8091	0.0724	0.7832

reported in Table 3. The student models show better performance with the third scheme, so we choose this scheme in our experiments. **The effectiveness of different layers to transfer:** We study transfer of different layers of FCN on the ECSSD and HKU-IS datasets. The results are shown in Fig. 9. The conv1-5 presents that conv1, conv2, conv3, conv4 and conv5 were transferred. Results imply that better performance is obtained with more layers transferred, but transferring low-level layers has little effect on accuracy. As we can see, the computation cost of second-order statistic transfer depends on the resolution of feature maps when training. Therefore, a solution would be transferring high-level layers with smaller resolution than low-level layers.

Table 2: Quantitative comparison of Size (smaller is better), FPS (larger is better), F-measure (larger is better) and MAE (smaller is better) on four benchmark datasets. The superscript ⁺ denotes CRF is used as the post-processing module.

Methods	Size(Mb)	FPS	ECSSD		HKU-IS		PASCAL-S		DUT-OMRON	
			MAE↓	F_β ↑	MAE↓	F_β ↑	MAE↓	F_β ↑	MAE↓	F_β ↑
UCF	112.5	22	0.071	0.902	0.061	0.888	0.116	0.818	0.120	0.730
NLDF	406	20.8	0.065	0.903	0.048	0.902	0.099	0.829	0.080	0.753
DSS	237	21	0.063	0.908	0.050	0.900	0.102	0.826	0.072	0.764
Amulet	126.5	16	0.061	0.914	0.050	0.899	0.100	0.832	0.098	0.743
RAS ⁺	81	43	0.059	0.918	0.045	0.913	0.104	0.834	0.062	0.786
R ³ Net ⁺	92.6	40	0.054	0.919	0.043	0.906	0.097	0.828	0.063	0.793
R³Net-1/2-FS⁺ (proposed)	41.7	60	0.064	0.903	0.053	0.887	0.113	0.805	0.066	0.781
R³Net-1/4-FS⁺ (proposed)	27.3	70	0.070	0.890	0.057	0.875	0.121	0.787	0.074	0.755

Figure 8: Visual comparison of saliency maps produced by different saliency models. Ground-truth is denoted 'GT'. The proposed model (c) achieves competitive accuracy with up to 4.38 times faster running and 93.27% size saving.

Table 3: Comparison of different training schemes' effectiveness. T presents teachers' output. S presents self-learning.

training schemes	ECSSD		HKUIS	
	MAE↓	F_β ↑	MAE↓	F_β ↑
Teacher	0.0788	0.8199	0.0622	0.8073
without T and S	0.0856	0.8072	0.1031	0.7223
without S	0.0820	0.8155	0.0902	0.7452
with T and S	0.0775	0.8269	0.0628	0.8056

Figure 9: Accuracy of different layers to transfer.

4.4 Comparison with State-of-the-art

We experiment with the R³Net [23] to compare with five state-of-the-art saliency prediction methods, including UCF [30], NLDF [29], DSS [22], Amulet [44], RA [33]. For fair comparison, we use the

saliency maps of these methods provided by the authors. As shown in Table 2, the small model transferred from R³Net, which is randomly initialized, obtains impressive speedups. The R³Net-1/4-FS model runs nearly 2 times faster than R³Net and is faster than the existing methods. As we can see, other models have large model size, which is too heavy for a pre-processing step to apply in subsequent high-level tasks. The size of the R³Net-1/4-FS model is much lower than others. In terms of F-measure, the gaps between the R³Net-1/4-FS model and the large teacher model are 2.9%, 3.1%, 4.1% and 3.8% on the four datasets. The gap is closer in terms of MAE. The R³Net-1/2-FS model achieves better performance than the R³Net-1/4-FS. We also show some visual results of different models in Fig. 8.

5 EMBEDDED APPLICATION IN VIDEO CAPTIONING

Saliency prediction is often embedded in other multimedia tasks, to provide better understanding of complex scenes. For example, in video captioning, a CNN such as VGG [24] or ResNet [45] is a commonly used model to extract visual features. The network is usually pre-trained on a large-scale image classification dataset [24], so it can discover many visual concepts from the input image. However, some concepts are more important than others when we try to interpret video content. It is necessary to design a method to discover the dominant visual concepts, which can be helped to generate correct and concise descriptions. Thus, we further embed the proposed saliency model into video captioning to verify its effect and efficiency.

Figure 10: The proposed saliency based S2VT approach. The input visual sequence to the model is comprised of the original images and saliency region images.

5.1 Datasets and Architecture

The MSR Video to Text [46] dataset is a popular large-scale benchmark for video captioning. It consists of 7,180 videos that are transformed into 10,000 clips. Following the original paper [46], we split videos by index number: 7,000 for training, 1,000 for validation and 2,000 for the test.

We use S2VT model [47] as the baseline model. The S2VT approach performs video description using a sequence to sequence model. It incorporates a stacked LSTM which first reads the sequence of frames and then generates a sequence of words. The visual sequence input to the model in [47] is comprised of RGB and optical flow CNN outputs. Due to large computation cost of optical flow, here we do not use optical flow images. We sample 40 frames for one video. As shown in Fig. 10, for each frame, we use the saliency prediction model to generate saliency maps. Then, we multiply the original images with the corresponding saliency maps, and the salient region can be obtained.

Since the salient region usually provides the dominant visual concepts and the non-salient region provides context information, a good model to extract visual features from video frames should pay more attention to the salient content, while still properly extracting information from non-salient regions at the same time. So we employ ResNet152 [45] to extract feature v_i from the complete image and feature v_s from the salient region respectively. The combination of these two features is regarded as the video representation. Thus a stacked LSTM can be used to generate a sequence of words.

5.2 Performance

The metrics for evaluating video captioning are the same as for image captioning, including BLEU [48], METEOR [49], and CIDEr [50]. The R³Net-1/4-FS model and R³Net model are used to predict saliency maps. As shown in Table 4, the proposed saliency-based S2VT approach achieves improvement on all the metrics. The gaps between SV2T-SS and SV2T-TS models are very small, but SV2T-SS is faster than SV2T-TS. If the number of video frames is larger, the advantage of SV2T-SS will be more significant. The result shows that saliency prediction is useful for video captioning to understand what is happening in the video. It provides the dominant visual

Table 4: Performance Comparisons on the MSR-VTT. B, M and CR represent BLEU, METEOR and CIDEr, respectively. Time is used to extract both CNN features and saliency features of one video. The teacher-model (R³Net) is embedded in SV2T-TS. The student-model (R³Net-1/4-FS) is embedded in SV2T-SS.

Model	B1	B2	B3	B4	M	CR	Time
SV2T	74.65	57.22	40.40	28.60	25.20	34.81	1.76
SV2T-TS	75.78	59.09	44.39	32.35	26.00	37.31	3.29
SV2T-SS	75.37	59.14	44.95	32.98	25.91	36.74	2.46

Figure 11: Visualization of some video captioning results. (Please find more results in the supplementary materials).

concept. Meanwhile, the result validates the effectiveness of KD. We also show some visual results in Fig. 11.

6 CONCLUSION

In this paper, we delve into the training of efficient saliency prediction models with knowledge distillation. Instead of fine-tuning from models pre-trained on ImageNet, we train a small and randomly initialized student model with the guidance of a teacher model. We employ first-order and second-order statistics transfers which are two complementary methods to help the student model to learn intermediate representations from the teacher model. We conduct experiments on three different networks (FCN, U-shape network and R³Net) and four benchmark datasets. The proposed model achieves competitive performance with state-of-the-art models with faster running speed due to smaller model size. The embedded experiments on video captioning show the proposed saliency prediction model is helpful to understand what is happening in the video and the proposed knowledge distillation mechanism improves the efficiency of student saliency prediction models. In the future, we plan to expand the transfer method to more architectures and pixel-wise tasks.

ACKNOWLEDGMENTS

This work was supported by the University of Chinese Academy of Sciences, in part of National Natural Science Foundation of China: 61472389, 61771457, 61732007, 61572503, 61872424, 61620106009, U1636214 and 61836002, in part by Youth Innovation Promotion Association, CAS: 61772494, 61836002, in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013. This research was supported in part by the National Science Foundation under grant IIS-1522125.

REFERENCES

- [1] B. Kandemir, Z.H. Zhou, J. Li, and J.Z. Wang. Beyond saliency: Assessing visual balance with high-level cues. In *ACM Multimedia Conference on Multimedia Conference*, pages 26–34. ACM, 2017.
- [2] L. Chukoskie, S. Guo, E. Ho, Y. Zhang, Q. Chen, V. Meng, J. Cao, N. Devgan, S. Wu, and P.C. Cosman. Quantifying gaze behavior during real world interactions using automated object, face, and fixation detection. *IEEE Transactions on Cognitive and Developmental Systems*, 10(4):1143–1152, 2018.
- [3] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. Paying more attention to saliency: Image captioning with saliency and context attention. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(2):48:1–48:21, 2018.
- [4] D.Q. Liu, Z.J. Zha, H.W. Zhang, Y.D. Zhang, and F. Wu. Context-aware visual policy network for sequence-level image captioning. In *ACM Multimedia Conference on Multimedia Conference*, pages 1416–1424. ACM, 2018.
- [5] K. Xu, L.Y. Wen, G.R. Li, L.F. Bo, and Q.M. Huang. Attentive recurrent neural network weak-supervised multi-label image classification. In *ACM Multimedia Conference on Multimedia Conference*, pages 22–28, 2018.
- [6] Y.T. Lin, Z.Y. Pang, D.H. Wang, and Y.T. Zhuang. Task-driven visual saliency and attention-based visual question answering. *CoRR*, abs/1702.06700, 2017.
- [7] Y.M. Fang, W.S. Lin, Z.Z. Chen, C.M. Tsai, and C.W. Lin. Video saliency detection in the compressed domain. In *ACM Multimedia Conference on Multimedia Conference*, pages 697–700. ACM, 2012.
- [8] Y.M. Fang, Z.Z. Chen, W.S. Lin, and C.W. Lin. Saliency detection in the compressed domain for adaptive image retargeting. *TIP*, 21(9):3888–3901, 2012.
- [9] Y.M. Fang, Z.Z. Chen, W.S. Lin, and C.W. Lin. Saliency-based image retargeting in the compressed domain. In *ACM Multimedia Conference on Multimedia Conference*, pages 1049–1052. ACM, 2011.
- [10] X.T. He, Y.X. Peng, and J.J. Zhao. Fine-grained discriminative localization via saliency-guided faster R-CNN. In *ACM Multimedia Conference on Multimedia Conference*, pages 627–635. ACM, 2017.
- [11] B.K. Bao, C.S. Xu, W.Q. Min, and M.S. Hossain. Cross-platform emerging topic detection and elaboration from multimedia streams. *TOMCCAP*, 11(4):54, 2015.
- [12] K. Xu, L.Y. Wen, G.Y. Li, L.F. Bo, and Q.M. Huang. Spatiotemporal cnn for video object segmentation. In *CVPR*, pages 1379–1388, 2019.
- [13] H.Y. Wang, Y.J. Xu, and Y.H. Han. Spotting and aggregating salient regions for video captioning. In *ACM Multimedia Conference on Multimedia Conference*, pages 1519–1526. ACM, 2018.
- [14] Y.Y. Chen, W.G. Zhang, S.H. Wang, L. Li, and Q.M. Huang. Saliency-based spatiotemporal attention for video captioning. In *Fourth International Conference on Multimedia Big Data (BigMM)*, pages 1–8, 2018.
- [15] Z. Guo, L.L. Gao, J.K. Song, X. Xu, J. Shao, and H.T. Shen. Attention-based LSTM with semantic consistency for videos captioning. In *ACM Multimedia Conference on Multimedia Conference*, pages 357–361. ACM, 2016.
- [16] Y. Wang, J.L. Liu, and X.J. Wang. Image caption with synchronous cross-attention. In *ACM Multimedia Conference on Multimedia Conference*, pages 433–441. ACM, 2017.
- [17] S.C. Fang, H.T. Xie, Z.J. Zha, N.N. Sun, J.L. Tan, and Y.D. Zhang. Attention and language ensemble for scene text recognition with convolutional sequence modeling. In *ACM Multimedia Conference on Multimedia Conference*, pages 248–256. ACM, 2018.
- [18] Y.B. Tang, X.Q. Wu, and W. Bu. Deeply-supervised recurrent convolutional neural network for saliency detection. In *ACM Multimedia Conference on Multimedia Conference*, pages 397–401. ACM, 2016.
- [19] S. He, J.W. Han, X.T. Hu, M. Xu, L. Guo, and T.M. Liu. A biologically inspired computational model for image saliency detection. In *ACM Multimedia Conference on Multimedia Conference*, pages 1465–1468. ACM, 2011.
- [20] B. Hariharan, P.A. Arbeláez, R.B. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, pages 447–456, 2015.
- [21] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *TPAMI*, 39(4):640–651, 2017.
- [22] Q.B. Hou, M.M. Cheng, X.W. Hu, A. Borji, Z.W. Tu, and P.H.S. Torr. Deeply supervised salient object detection with short connections. *TPAMI*, 41(4):815–828, 2019.
- [23] Z.J. Deng, X.W. Hu, L. Zhu, Xu X.M., J. Qin, G.Q. Han, and P.A. Heng. R³net: Recurrent residual refinement network for saliency detection. In *IJCAI*, pages 684–690, 2018.
- [24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- [25] Q.Q. Li, S.Y. Jin, and J.J. Yan. Mimicking very efficient network for object detection. In *CVPR*, pages 7341–7349, 2017.
- [26] J. Shen, N. Vedapant, V.N. Boddeti, and K.M. Kitani. In teacher we trust: Learning compressed models for pedestrian detection. *CoRR*, abs/1612.00478, 2016.
- [27] A. Borji, M.M. Cheng, H.Z. Jiang, and J. Li. Salient object detection: A benchmark. *TIP*, 24(12):5706–5722, 2015.
- [28] R. Zhao, W.L. Ouyang, H.S. Li, and X.G. Wang. Saliency detection by multi-context deep learning. In *CVPR*, pages 1265–1274, 2015.
- [29] Z.M. Luo, A.K. Mishra, A. Achkar, J.A. Eichel, S.Z. Li, and P.M. Jodoin. Non-local deep features for salient object detection. In *CVPR*, pages 6593–6601, 2017.
- [30] P.P. Zhang, D. Wang, H.C. Lu, H.Y. Wang, and B.C. Yin. Learning uncertain convolutional features for salient object detection. In *ICCV*, pages 212–221, 2017.
- [31] Z. Wu, L. Su, and Q.M. Huang. Cascaded partial decoder for fast and accurate salient object detection. In *CVPR*, pages 3907–3916, 2019.
- [32] Z. Wu, L. Su, and Q.M. Huang. Learning coupled convolutional networks fusion for video saliency prediction. In *IEEE Transactions on Circuits and Systems for Video Technology*, page Early Access DOI: 10.1109/TCSVT, 2018.
- [33] S.H. Chen, X.L. Tan, B. Wang, and X.L. Hu. Reverse attention for salient object detection. In *ECCV*, volume 11213 of *Lecture Notes in Computer Science*, pages 236–252. Springer, 2018.
- [34] G.E. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.
- [35] A. Romero, N. Ballas, S.E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.
- [36] S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.
- [37] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.
- [38] O. Ronneberger. Invited talk: U-net convolutional networks for biomedical image segmentation. In *Bildverarbeitung für die Medizin, Informatik Aktuell*, page 3. Springer, 2017.
- [39] M.M. Cheng, N.J. Mitra, X.L. Huang, P.H.S. Torr, and S.M. Hu. Global contrast based salient region detection. *TPAMI*, 37(3):569–582, 2015.
- [40] J.P. Shi, Q. Yan, L. Xu, and J.Y. Jia. Hierarchical image saliency detection on extended CSSD. *TPAMI*, 38(4):717–729, 2016.
- [41] G.B. Li and Y.Z. Yu. Visual saliency detection based on multiscale deep CNN features. *TIP*, 25(11):5012–5024, 2016.
- [42] Y. Li, X.D. Hou, C. Koch, J.M. Rehg, and A.L. Yuille. The secrets of salient object segmentation. In *CVPR*, pages 280–287, 2014.
- [43] C. Yang, L.H. Zhang, H.C. Lu, X. Ruan, and M.H. Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013.
- [44] P.P. Zhang, D. Wang, H.C. Lu, H.Y. Wang, and X. Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *ICCV*, pages 202–211, 2017.
- [45] K.M. He, X.Y. Zhang, S.Q. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [46] S. Liu, Z. Ren, and J.S. Yuan. Sibnet: Sibling convolutional encoder for video captioning. In *ACM Multimedia Conference on Multimedia Conference*, pages 1425–1434. ACM, 2018.
- [47] S. Venugopalan, M. Rohrbach, J. Donahue, R.J. Mooney, T. Darrell, and K. Saenko. Sequence to sequence - video to text. In *ICCV*, pages 4534–4542, 2015.
- [48] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002.
- [49] M.J. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *EACL*, pages 376–380, 2014.
- [50] S.Z. Chen, J. Chen, Q. Jin, and A.G. Hauptmann. Video captioning with guidance of multimodal latent topics. In *ACM Multimedia Conference on Multimedia Conference*, pages 1838–1846, 2017.