This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2921861, IEEE Transactions on Image Processing

Image Recognition by Predicted User Click Feature with Multidomain Multitask Transfer Deep Network

Min Tan*, *Member, IEEE*, Jun Yu*, *Member, IEEE*, Hongyuan Zhang*, Yong Rui†, *Fellow, IEEE*, and Dacheng Tao‡, *Fellow, IEEE*

***Abstract*—The click feature of an image, defined as a user click count vector based on click data, has been demonstrated to be effective for reducing the semantic gap for image recognition. Unfortunately, most of the traditional image recognition datasets do not contain click data. To address this problem, researchers have begun to develop a click prediction model using assistant datasets containing click information and have adapted this predictor to a common click-free dataset for different tasks. This method can be customized to our problem, but it has two main limitations: 1) the predicted click feature often performs badly in the recognition task since the prediction model is constructed independently of the subsequent recognition problem; 2) transferring the predictor from one dataset to another is challenging due to the large cross-domain diversity. In this paper, we devise a multitask and multidomain deep network with varied modals (MTMDD-VM) to formulate image recognition and click prediction tasks in a unified framework. Datasets with and without click information are integrated in the training. Furthermore, a nonlinear word embedding with a position-sensitive loss function is designed to discover the visual click correlation. We evaluate the proposed method on three public dog breed image datasets, and we utilize the Clickture-Dog dataset as the auxiliary dataset that provides click data. The experimental results show that 1) the nonlinear word embedding and position-sensitive loss function largely enhance the predicted click feature in the recognition task, realizing a 32% improvement in accuracy; 2) the multitask learning framework improves accuracies in both image recognition and click prediction; and 3) the unified training using the combined dataset with and without click data further improves the performance. Compared with state-of-the-art methods, the proposed approach not only performs much better in accuracy but also achieves good scalability and one-shot learning ability.**

***Index Terms*—Image recognition, Click prediction, Transfer deep learning, Multitask learning, Word embedding.**

I. Introduction

FINE-GRAINED image recognition serves as a core problem in the computer vision community. Though many efforts have been made to improve the performance, the high visual similarities among different categories still challenge this task [1]–[3]. To address this issue, some researchers have been focusing on utilizing user click data [4] for improved image recognition.

* Key Laboratory of Complex Systems Modeling and Simulation, School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China;
† Lenovo, No.6 Shang Di West Road, Haidian, Beijing, 100085, China;
‡ UBTECH Sydney Artificial Intelligence Centre and the School of Computer Science, in the Faculty of Engineering and Information Technologies, at the University of Sydney, 6 Cleveland St, Darlington, NSW 2008, Australia;
Jun Yu is the corresponding author. Email: yujun@hdu.edu.cn.

User click data, which is the human annotation for the correlation of a query-image pair, has been demonstrated to be effective for reducing the semantic gap in fine-grained image recognition [4]–[6]. It consists of three parts: queries, images, and the corresponding click count. Fig. 1 has visualized user click data, wherein some clicked queries with corresponding click count for three dog (top row) and bird (bottom row) samples are shown. The correlation between images and associated queries can be quantified by click count [5].

With click data, each image can be represented as a click-count-feature-vector based on its clicked query/word set [3], [7], e.g., term frequency-inverse document frequency (TF-IDF) vector shown in Fig. 1. The nonzero elements in TF-IDF vector can capture valuable image attributes. Compared with visual features, the click feature not only easily describes semantics but is also invariant to the changes of image conditions.

A. Motivation

Though the click feature is powerful, most existing datasets do not contain user click data. Recently, Bai et al. proposed a deep word embedding model to learn the visual click correlation from an assistant user click data, and adapted it to dataset construction tasks [7]. The word embedding model is used for click feature regression from the visual feature (shown in Fig. 3(a)), and is learned by minimizing the distance between the regressed click feature and ground truth one.

One can consider this word embedding model as a click predictor, and directly use it for image recognition tasks on a common click-free dataset (refer to Fig. 2). However, it may be problematical since the learning of click predictor and image classifier are independent. Note that the prediction model is learned only by minimizing the prediction error without any supervised information for recognition, i.e., ground truth category labels. This unsupervised learning framework decreases the discriminant of the predicted click feature in recognition tasks and may result in unsatisfactory accuracies.

B. Idea of Our Approach

To address these issues, we formulate the click prediction and the image recognition tasks in a unified framework. A click predictor is learned via jointly minimizing the prediction and recognition error (refer to Fig. 3(b)), wherein the combined softmax and ℓ_2 distance loss function is designed. Note that softmax and ℓ_2 loss functions are utilized for image recognition (plotted in purple-dotted box) and click prediction (plotted in red-dotted box) tasks, respectively. We call the click

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2921861, IEEE Transactions on Image Processing

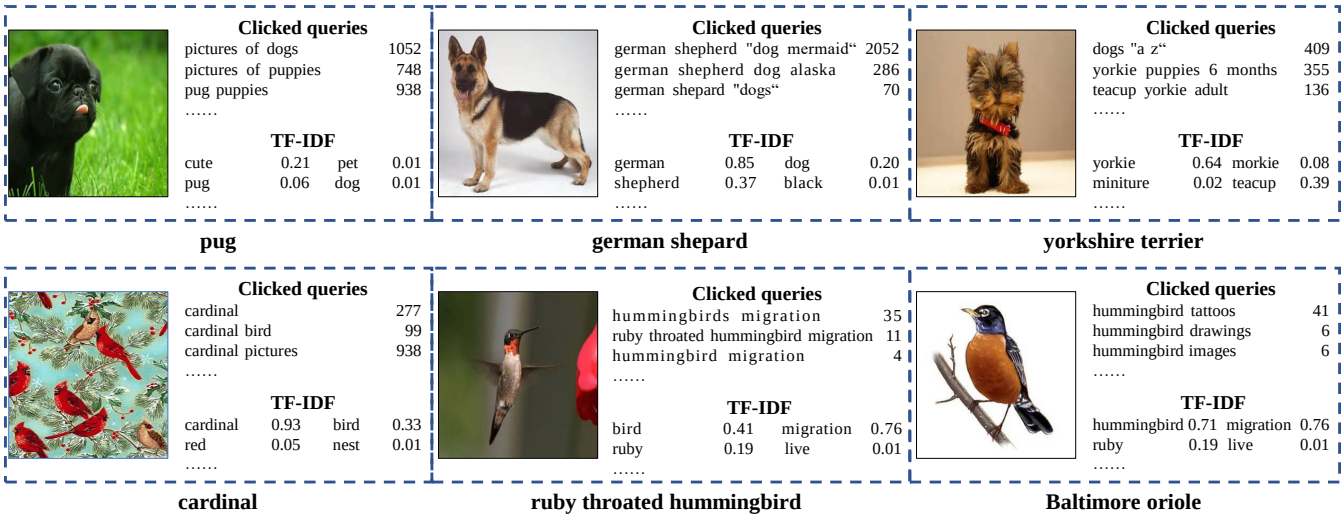


Fig. 1. Visualization of some clicked queries and the corresponding click feature, namely TF-IDF vector, for three dog (top row) and bird (bottom row) samples. The category name of each sample is listed below. For each TF-IDF vector, only a subset feature vector with respect to four word-items are shown.

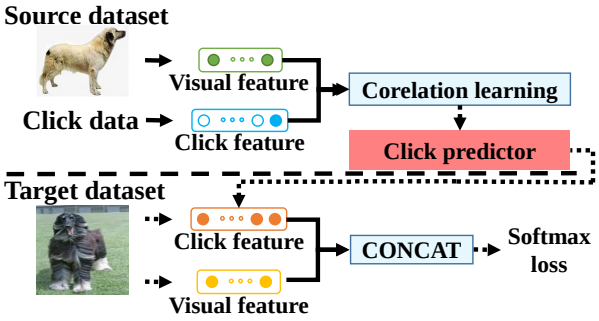


Fig. 2. Traditional pipeline of image recognition with predicted click feature. It first learns a click predictor from an assistant dataset with click information, then adapts the predictor to another click-free dataset for the recognition task.

predictor learned with only the ℓ_2 distance loss function (refer to Fig. 3(a)) and the combined loss (refer to Fig. 3(b)) task-independent and task-related learning, respectively. Compared with the task-independent learner, the task-related learner helps generate more discriminative click features since 1) it provides additional supervised information for recognition, i.e., the image category labels in softmax loss and 2) it is more robust to noise. This is because the task-independent learners try to enhance the click prediction accuracy on all image samples even if they are noises, while for the task-related learners, noises will likely be penalized with the softmax loss (the noisy sample always has large recognition error). In summary, we propose a novel image recognition approach with a combined deep visual and predicted click feature, and propose a multitask and multidomain deep network with varied modals (MTMDD-VM) to learn both the deep visual feature and click predictor. A multitask learning framework is devised by minimizing the prediction and recognition error simultaneously. Additionally, to address the cross-domain diversity, we combine both the datasets with (source) and without (target) click data in a unified training framework. We call our model MTMDD-VM since both the click prediction and recognition tasks are simultaneously formulated and since the input images are sampled from different domains with varied modals (one containing click data while another being a click-free one).

Furthermore, to better model the visual click similarity, a nonlinear word embedding with position-sensitive loss function is developed. The nonlinear word embedding handles the isomerism of visual click feature spaces, while the position-sensitive loss function improves the robustness of click prediction against the heavy noise in the ground truth click-count-vector. In MTMDD-VM, different loss functions are developed for different subtasks. For the image recognition task, a categorical cross-entropy loss is designed on the combined visual and click feature; for the click prediction task, both the count-based and position-sensitive loss functions are developed on the click feature space.

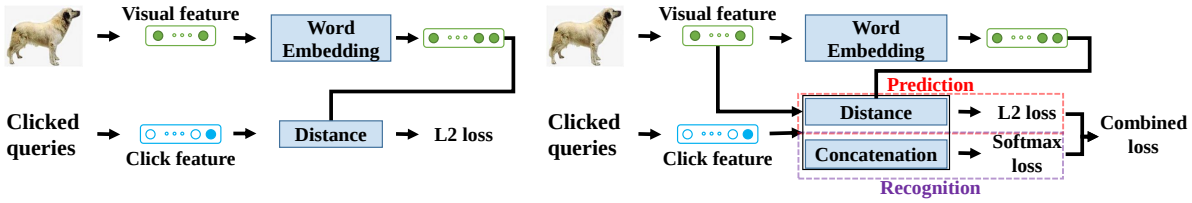
We conduct extensive comparisons and validations on three challenging public datasets, which demonstrate the advantages of our method.

C. Contributions

The contributions of this work are threefold:

- It is the FIRST paper to address the image recognition problem using the combined deep visual and predicted click feature. A click feature predictor is learned with the help of an assistant dataset that contains user click data. The proposed method has good scalability and one-shot learning ability even when the source and target data originate from different objects.
- We devise a novel multitask multidomain deep neural network with varied modals to simultaneously learn the deep visual feature and click feature predictor. Both the click prediction and image recognition tasks are formulated into the problem. Datasets with (source) and without (target) click information are combined in a unified training framework, wherein the multiple domains are with different input modals.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2921861, IEEE Transactions on Image Processing



(a) Task-independent (b) Task-related

Fig. 3. Click predictors trained with task-independent and task-related learners. The ℓ_2 loss and softmax loss are designed for click prediction and image recognition tasks, respectively. The softmax loss is computed on the concatenated predicted click and visual feature.

- We propose a nonlinear word embedding with a position-sensitive loss function to improve the robustness of click prediction model. The nonlinear word embedding can better handle the isomerism of visual click feature spaces. The position-sensitive loss function helps to predict the click position in addition to the actual click count for images, which improves recognition accuracy as well as generating a better click feature that is much closer to the ground truth one.

The rest of this paper is organized as follows. Section II discusses some related work. Section III describes our model, including the formulation and training framework. Section IV presents the experimental results. Finally, Section V concludes our work and sketches several directions of future work.

II. Related Work

A. Fine-grained Image Recognition

In terms of input data, research on fine-grained image recognition can be categorized into two groups: 1) visual feature-based recognition. It is challenging due to illumination variations, view changes, occlusion, etc. In addition, semantic gaps appear in visual feature-based recognition; 2) recognition from combined visual feature and image attributes. The attributes are either distinctive object parts [8] or property descriptors [9]. Recently, user click data are increasingly used as the attribute annotations [4], [6]. Note that in most existing research on click data-based recognition, image datasets already contain user click information [4], [6], [10]. Unfortunately, in practice, datasets that contain click data are extremely rare, resulting in a large limitation for click data-based recognition. Recently, some researchers started to construct click prediction models and customize them to a common click-free dataset for different tasks, e.g., retrieval, dataset construction [3], [7], etc.

B. Click Prediction

Research on click prediction can be summarized into at least two categories: 1) **Click completion**. Predict the zero items using the nonzero items in an incomplete click feature vector. As shown in Fig. 4(a), for the input image $\mathbf{x}$ and its corresponding "rough" click feature $\mathbf{v}$, the predictor helps to generate a dense click vector $\widehat{\mathbf{v}}$ by estimating the zero items. For example, Yu et al. proposed a multimodal sparse coding to predict these zero items [10]; Tan et al. proposed a similarity-based propagation method for this task [6]. One large limitation of these methods is that a "rough" sparse click feature is required in advance. 2) **Click generation**. Predict the whole query click feature vector for each image. As shown in Fig. 4(b), for the input image $\mathbf{x}$, the predictor is used to generate a click vector $\widehat{\mathbf{v}}$ without any prior information. It is a challenging task since the visual click feature correlation is required to be learned. This problem is RARELY studied but of great help for image recognition, since most of existing recognition datasets contain NO click data.

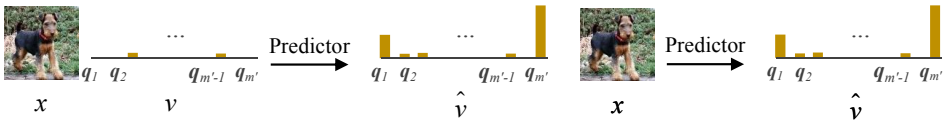
Closely related techniques for click generation/prediction can be roughly summarized into the following groups:

1) Transfer Learning. With this method, a click predictor is first learned from one dataset and then adapted to other datasets for different tasks, such as recognition, retrieval, etc [11]. Due to the powerful transferability of the convolutional neural network, many researchers focus on transferring the learned convolutional features to different tasks. For example, Bai et al. proposed a deep neural network to learn the image query correlation using click data, and transfer it to a dataset construction task with query-dependent categories [7]. Note that, the transferability of features in different layers varies considerably [12], such that the performance of model transfer is greatly influenced by the used feature layers.

2) Multitask/multidomain Learning. It differs from conventional transfer learning approach in handling different domains/tasks simultaneously. Early researchers focus on multitask learning in a single domain, wherein different tasks are formulated in a unified framework [13], [14]. Others employed domain adaptation schemes to construct a transferable model on the combined multidomain dataset, including the adversarial network [15]–[18], domain alignment [19]–[23], and weight sharing [24]–[26], etc. Recently, inspired by advantages of deep learning in addressing large-scale and multisource datasets, there is an increasing number of learning of multitask and multidomain deep learning (MTMDD) methods [27]–[29]. In MTMDD, a combined dataset from several domains was utilized to optimize multiple tasks simultaneously. Note that in existing MTMDD methods, different domains share the same input modals, but differ in feature distribution or annotation degree (well-labeled data in the source while partially labeled in the target). However, in our problem, datasets are all well-labeled in different domains but with different input modals, i.e., one containing both visual and click features while another being a click-free one.

To the best of our knowledge, this is the FIRST paper to jointly use both the deep visual and predicted click feature for image recognition, wherein a multitask multidomain deep

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2921861, IEEE Transactions on Image Processing



(a) Click completion (b) Click generation

Fig. 4. Two kinds of click prediction models. The former predicts the zero items in an incomplete click feature vector, while the later predicts the whole click feature vector without any prior information

neural network is constructed to learn both the deep visual model and click predictor. Datasets with multiple domains and multiple modals are combined in a unified learning framework.

III. Image Recognition with Predicted User Clicks

As aforementioned, we utilize the combined deep visual and predicted user click feature for image recognition, wherein a visual click correlation model is learned for click prediction from the visual feature. Recently, Bai et al. utilized an assistant dataset with user click data to learn this visual click correlation, and adapted it to dataset construction tasks [7]. One can consider this visual click correlation as a click predictor, and customize it to the current click-free dataset for recognition.

It is worth noting that directly applying these predictors in another dataset may be problematic. This occurs because of the cross-domain and cross-task variance, where the predictor learned for prediction tasks on the source domain may be inapplicable for recognition tasks on the target domain. Additionally, the input from different domains is with different modals, i.e., input for the source domain is image-click pairs, while only images serve as the target domain input. To address these problems, we propose a novel multidomain and multitask deep learning framework to learn both the deep visual feature and click predictor. In addition, to better learn the visual click correlation, we design an improved word embedding model by integrating a nonlinear transformation and position-sensitive loss function. In this subsection, we first describe the user click feature of images, then introduce image recognition with conventional transfer deep learning, and finally introduce our MTMDD-VM approach in detail. The major notations used in this paper are listed in Table I.

A. User Click Feature of Images

The user click feature captures semantic content in images, and it is generated from user click data [5]. We describe the generation of the click feature from click data below.

Similar to [5], we represent each image as a query click feature vector based on query click frequency. To address the high dimensionality and semantic redundancy in queries, we follow [7] and generate the click feature of images using keyword click frequency instead of query click frequency. The user click feature helps to handle the subtle visual differences among categories in fine-grained recognition tasks, which is an important issue in traditional visual-based approaches.

In particular, we denote the query click matrix by $\mathbf{C}^s$, with each row as the query click frequency vector for one image. $\mathcal{Q}^s$ denotes all the clicked query sets in the source domain. Inspired by [7], we construct the ground truth click feature $\mathbf{v}$ for source images with $\mathbf{C}^s$ and $\mathcal{Q}^s$. It is generated by following steps via the TF-IDF algorithm:

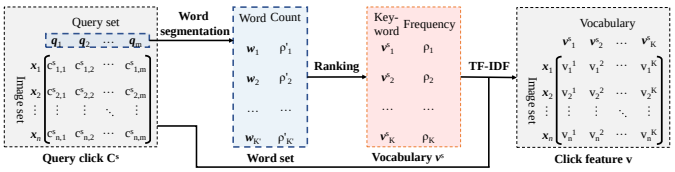


Fig. 5. Generation of click feature from user click data via the TF-IDF algorithm, wheree K' denotes the number of all segmented words.

- Obtain the word set $\{(\mathbf{w}_i,\rho'_i)|\mathrm{i}=1,2,...,\mathrm{K}'\}$[1] by query formation on $\mathcal{Q}^s$, and create a vocabulary $\mathcal{V}^s$ with K top clicked words. Additionally, each query set ζ_j containing word $\mathcal{V}^s_j$ is obtained.
- Generate a word click matrix $\mathbf{C}'^s$ using $\mathbf{C}^s$ and ζ_j, with its element $c'^s_{i,j} = \sum_{k\in\zeta_j} c^s_{i,k}$.
- Compute the ground truth click feature $\mathbf{v}$ via the TF-IDF algorithm [30]. The j-th element in the ground truth click feature for image i, i.e., $\mathbf{v}^j_i$, is defined as follows:

$$\mathbf{v}^j_i = \frac{c'^{\mathrm{s}}_{i,j}}{\sum_j c'^{\mathrm{s}}_{i,j}} \log\frac{1}{\rho_j}, \rho_j = \frac{\sum_i I(c'^{\mathrm{s}}_{i,j})}{n}, \qquad (1)$$

where ρ_j denotes the percentage of images with a nonzero click count on $\mathcal{V}^s_j$, $\mathrm{I}(x)$ is 1 (0) when $x \neq 0$ ($x = 0$), and $\sum_i I(c'^{\mathrm{s}}_{i,j})$ is the number of images clicked under $\mathcal{V}^s_j$.

The generation of the click feature from user click data is illustrated in Fig. 5, wherein a vocabulary $\mathcal{V}^s$ with the top click frequencies is created.

B. The Conventional Transfer Deep Learning (TDL)

Given a target image dataset $\mathcal{I}^t$ with n' images $\mathbf{x}$ and their category labels $y \in \mathcal{Y}$, our task is to learn the deep visual feature $\boldsymbol{\varphi}$ and the predicted click feature $\hat{\mathbf{v}}$ for image recognition. We introduce an assistant dataset $\mathcal{I}^s$ (source dataset) with click information to learn the visual click correlation via a word embedding [7] so that we can utilize it to predict the click feature from the visual feature for target images. The transfer deep learning-based recognition is divided into two successive subproblems, i.e., click prediction and image recognition.

[1] ρ'_i summarizes the click count under all queries containing $\mathbf{w}_i$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2921861, IEEE Transactions on Image Processing

TABLE I
MAJOR NOTATIONS AND DEFINITIONS.

Notation	Definition
$\mathcal{Y}^s, \mathcal{Y}^t, \mathcal{Y}$	Category label set for source, target, and the combined domain, respectively.
$\mathcal{I}^s = \{(\mathbf{x}_i, y_i, \mathbf{u}_i) \vert 1 \le i \le n\}$	Source domain dataset containing n images $\mathbf{x}$ with labels $y \in \mathcal{Y}$ and query-click vector $\mathbf{u} \in \mathbb{R}^m$.
$\mathcal{Q}^s = \{(\mathbf{q}_i) \vert 1 \le i \le m\}$	Associated query set with m queries $\mathbf{q}$ in source domain.
$\mathcal{V}^s = \{(\mathbf{w}_i) \vert 1 \le i \le K\}$	Vocabulary with K frequently clicked words generated from $\mathcal{Q}^s$ by query formation.
$\mathbf{C^s} \in \mathbb{R}^{n \times m}, \mathbf{C'}^s \in \mathbb{R}^{n \times K}$	Click matrix in source domain, where $c^s_{i,j}$ ($c'^s_{i,j}$) is the click count of $\mathbf{x}_i$ under query $\mathbf{q}_j$ (word $\mathcal{V}^s_j$).
$\mathcal{I}^t = \{(\mathbf{x}_i, y_i) \vert 1 \le i \le n'\}$	Target domain dataset containing n' images $\mathbf{x}$ with labels $y \in \mathcal{Y}$.
$\boldsymbol{\theta}_1, \mathbf{E}$	Visual feature functions and linear word embedding matrix trained on source domain with TDL.
$\boldsymbol{\theta}_2$	Parameter for the whole deep network fine-tuned on the target domain with TDL.
$\boldsymbol{\theta}$	The whole deep network parameter trained on combined source and target domains with MTMDD-VM.
α, λ	Parameter for weight decay and tradeoff between prediction and recognition error.
τ	Tradeoff between click count-based loss function and position-sensitive one.
μ	Weight for loss functions on target domain with respect to source domain.
$\mathbf{E}_1$	Convolutional transformation in the nonlinear word embedding function.
$\mathbf{E}_2, \mathbf{E}_3$	Two fully connected mapping matrixes in the nonlinear word embedding function.
$\mathbf{v}, \widehat{\mathbf{v}}$	The ground truth and predicted click count feature vector.
$\mathbf{v}^b, \widehat{\mathbf{v}}^b$	The ground truth and predicted click position feature vector.
B, T	Bandwidth and error threshold in converting the click count vector to a click position one.
$\ell^p(\mathbf{v}, \widehat{\mathbf{v}})$	Distance error of predicted click feature $\widehat{\mathbf{v}}$ with respect to the ground truth one $\mathbf{v}$.
$\ell^c(y, \mathbf{o})$	Softmax loss with output $\mathbf{o}$ when ground truth category is y.
$\boldsymbol{\omega}, \boldsymbol{\varphi}$	The final deep visual feature and deep visual feature output from "ConvNet" in Fig. 8.
$\boldsymbol{\phi}$	The combined deep visual and click feature vector.
$\mathcal{N}, \mathcal{N}'$	Batch normalization and ℓ_2 normalization operator.

1) Learning the Click Predictor. As shown in Fig. 3(a), using the source dataset $\mathcal{I}^s$, we learn a predictor with a deep network and word embedding [7]:

$$\begin{aligned} \{\boldsymbol{\theta}_1^*, \mathbf{E}^*\} = & \underset{\boldsymbol{\theta}_1, \mathbf{E}}{argmin} \{\frac{\alpha}{2}(\|\boldsymbol{\theta}_1\|_2^2 + \|\mathbf{E}\|_F) + \frac{1}{n}\sum_{\mathbf{x}_i \in \mathcal{I}^s} \ell^p(\mathbf{v}_i, \widehat{\mathbf{v}}_i)\}, \\ & s.t. \begin{cases} \widehat{\mathbf{v}}_i = \mathbf{E} \cdot \boldsymbol{\theta}_1(\mathbf{x}_i), \\ \ell^p(\mathbf{v}_i, \widehat{\mathbf{v}}_i) = \|\mathcal{N}'(\mathbf{v}_i) - \mathcal{N}'(\widehat{\mathbf{v}}_i)\|_2^2 . \end{cases} \end{aligned} \tag{2}$$

where ℓ^p is prediction loss computed by the ℓ_2 distance on ℓ_2-normalized click count vector, $\boldsymbol{\theta}_1$ are feature functions based on the deep convolutional neural network, $\mathbf{E}$ is the matrix encoding the word embedding function, and $\|\mathbf{E}\|_F$ denotes the Frobenius norm of $\mathbf{E}$. In (2), the prediction error of predicted click feature against the ground truth one is minimized.

Let $\mathbf{h}(\cdot)$ be the click feature predictor composed of $\boldsymbol{\theta}_1^*$ and the word embedding matrix $\mathbf{E}^*$. The predicted click feature $\widehat{\mathbf{v}}$ is obtained as follows:

$$\widehat{\mathbf{v}}_i = \mathbf{h}(\mathbf{x}_i) = \mathbf{E}^* \cdot \boldsymbol{\theta}_1^*(\mathbf{x}_i). \tag{3}$$

Similar to [7], the word embedding layer $\mathbf{E}^*$ can be designed after convolutional layers of VGGNet [31] by performing a linear transformation on visual features.

2) Learning Recognizer. With the learned click predictor $\mathbf{h}(\cdot)$, we learn a deep neural network-based classifier in target domain $\mathcal{I}^t$. The deep learning problem is formulated as follows:

$$\begin{aligned} \boldsymbol{\theta}_2^* = & \underset{\boldsymbol{\theta}_2}{argmin} \{\frac{\alpha}{2}\|\boldsymbol{\theta}_2\|_2^2 + \frac{1}{n'}\sum_{\mathbf{x}_i \in \mathcal{I}^t} \ell^c(y_i, \mathbf{o}_i)\}, \\ & s.t.\ \ell^c(y_i, \mathbf{o}_i) = -\log(\frac{e^{\mathbf{o}_{y_i}}}{\sum_{j \in \mathcal{Y}} e^{\mathbf{o}_j}}), \end{aligned} \tag{4}$$

where $\boldsymbol{\theta}_2$ encodes the whole deep network parameter fine-tuned in the target domain, $\ell^c(y_i, \mathbf{o}_i)$ is softmax loss for image i, and $\mathbf{o}_i$ is the output probability distribution over each category computed on image representation $\boldsymbol{\phi}_i$. Note that $\boldsymbol{\phi}_i$ can be either the predicted click feature $\widehat{\mathbf{v}}_i$ or its combination with deep visual feature $\boldsymbol{\varphi}_i$ as follows:

$$\boldsymbol{\phi}_i = [\boldsymbol{\varphi}_i, \widehat{\mathbf{v}}_i] = [\boldsymbol{\varphi}_i, \mathbf{E}^* \cdot \boldsymbol{\theta}_1^*(\mathbf{x}_i)]. \tag{5}$$

Note that $\boldsymbol{\theta}_1^*$ and $\mathbf{E}^*$ are the learned click predictors on the source dataset (refer to (2)).

C. The Proposed Model: MTMDD-VM

The TDL model presented in Section III-B has difficulties addressing multitask problems with datasets that are multidomain and multimodal; therefore, we propose the following multidomain multitask deep model framework for our problem. Different from TDL, we integrate the training images in both source $\mathcal{I}^s$ and target $\mathcal{I}^t$ domains and learn a unified deep neural network $\boldsymbol{\theta}$ for both the click prediction ($\boldsymbol{\theta}_1$ in (2)) and the image recognition ($\boldsymbol{\theta}_2$ in (4)) part.

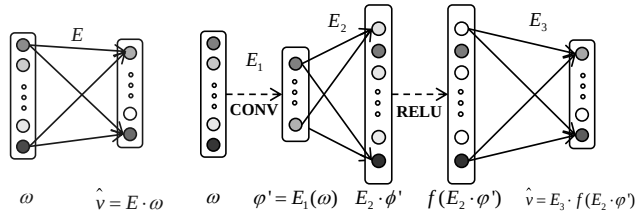
We formulate our problem as follows:

$$\begin{aligned} \boldsymbol{\theta}^* &= \underset{\boldsymbol{\theta}}{argmin}\, J(\boldsymbol{\theta}) \\ &= \underset{\boldsymbol{\theta}}{argmin} \{\frac{\alpha}{2}\|\boldsymbol{\theta}\|_2^2 + \frac{1}{n+n'}\sum_{\mathbf{x}_i \in \mathcal{I}} \ell(y_i, \mathbf{o}_i, \mathbf{v}_i, \widehat{\mathbf{v}}_i)\}, \end{aligned} \tag{6}$$

where $\mathcal{I}$ is the union of the source $\mathcal{I}^s$ and target $\mathcal{I}^t$ domain, $\ell(y_i, \mathbf{o}_i, \mathbf{v}_i, \widehat{\mathbf{v}}_i)$ is the loss value for image $\mathbf{x}_i$ with output $\mathbf{o}_i$ and predicted click feature $\widehat{\mathbf{v}}_i$, when the ground truth category label and click feature are y_i and $\mathbf{v}_i$, respectively. $\boldsymbol{\theta}$ encodes the whole network parameter, which summarizes the deep convolutional neural network, the word embedding functions, and the related layers in the feature combination, e.g., normalization and RELU operators.

Note that the first term in (6) is the regularization term, which trades an ℓ_2 penalty on the parameters of the whole

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2921861, IEEE Transactions on Image Processing



(a) Linear (b) Nonlinear

Fig. 6. The proposed nonlinear word embedding model compared with the traditional linear one. In nonlinear word embedding, the deep visual feature $\boldsymbol{\phi}$ is transformed to a click count feature $\widehat{\mathbf{v}}$ by a convolutional transformation $\mathbf{E}_1$, two word embedding matrixes $\mathbf{E}_2$, $\mathbf{E}_3$, and a nonlinear operator $f(\cdot)$.

network; the second term tries to minimize both the prediction and recognition error. For $\mathbf{x} \in \mathcal{I}^s$, $\mathbf{v}$ is defined exactly as in Section III-A, while for $\mathbf{x} \in \mathcal{I}^t$, we define the ground truth $\mathbf{v} = \mathbf{0}$, since no ground truth click feature $\mathbf{v}$ is given.

The final image representation $\boldsymbol{\phi}$ is generated by concatenating the deep visual feature $\boldsymbol{\varphi}$ and predicted click feature $\widehat{\mathbf{v}}$:

$$\boldsymbol{\phi} = [\mathcal{N}(\boldsymbol{\varphi}), \mathcal{N}(\widehat{\mathbf{v}})], \tag{7}$$

where $\mathcal{N}(\cdot)$ is the batch normalization operator to address the scale variance between the visual feature and click one.

The main differences between our model and the traditional TDL (refer to Section III-B) lie in two aspects: the multitask learning framework and the unified training with cross-domain cross-modal datasets. In additional to the transfer framework, we propose an improved click prediction model.

1) Improved Click Prediction Model. We propose an improved click prediction model by integrating a nonlinear word embedding structure and position-sensitive loss function.

a) Nonlinear Word Embedding. Note that the linear word embedding function (refer to Fig. 6(a)) in (3) may have difficulty addressing the isomerism of the image visual and click feature spaces. To address this issue, we propose a nonlinear word embedding by adding several nonlinear transformation layers in the network. As shown in Fig. 6(b), we design an additional "convolutional" transformation (CONV) and two fully connected transformations (biFC) in the nonlinear word embedding function, and the corresponding mapping matrixes are denoted by $\mathbf{E}_1$, $\mathbf{E}_2$, and $\mathbf{E}_3$. In addition, a nonlinear "RELU" operator $f(\cdot)$ is employed to filter out the negative word frequencies. Using nonlinear word embedding functions, the click feature $\widehat{\mathbf{v}}$ is predicted as follows:

$$\widehat{\mathbf{v}} = \mathbf{E}_3 \cdot f(\mathbf{E}_2 \cdot \boldsymbol{\varphi}'), \boldsymbol{\varphi}' = \mathbf{E}_1(\boldsymbol{\omega}), \tag{8}$$

where $\boldsymbol{\omega}$ denotes the output of the deep convolutional network ("ConvNet" in Fig. 8). Note that $\mathbf{E}_1$ encodes the "convolutional" transformation, which summarizes the convolution, pooling, and "RELU" operator.

b) Position-sensitive Loss Function. Similar to (2), the click prediction error can be evaluated by the Euclidean distance between the predicted and ground truth click feature in terms of click count. However, for each image-query pair, the click counts collected from the search engine often have too much noise, making training the click predictor by minimizing the prediction error in the click count sensitive to noise. Additionally, in practice, predicting the word set for an image with a nonzero click-count is sometimes more meaningful than the actual click count vector. Therefore, in addition to the distance error function on the click count, we propose a relaxed loss function to measure the prediction error on the word set with a nonzero click count, namely, the click position:

$$\begin{aligned} \widetilde{\ell^p}(\mathbf{v}_i, \widehat{\mathbf{v}}_i) &= \left| \{\mathcal{A} \cup \widehat{\mathcal{A}}\} \backslash \{\mathcal{A} \cap \widehat{\mathcal{A}}\} \right|, \\ & s.t.\ \mathcal{A} = \{j | (\mathbf{v}_i)_j \neq 0\}, \widehat{\mathcal{A}} = \{j | (\widehat{\mathbf{v}}_i)_j \neq 0\}, \end{aligned} \tag{9}$$

where $|\mathcal{A}|$ denotes the cardinal number for set $\mathcal{A}$, and $\mathcal{A}\backslash\mathcal{B}$ is the difference set of $\mathcal{A}$ to $\mathcal{B}$. Since (9) measures the error on the position of nonzero items in the predicted click count vector, we call it the position-sensitive loss function.

By introducing an ℓ_2-normalized binarized function $S(\cdot)$, we re-write (9) as the approximate click position loss function:

$$\begin{aligned} \widetilde{\ell^p}(\mathbf{v}_i, \widehat{\mathbf{v}}_i) &= \|S(\mathbf{v}_i) - S(\widehat{\mathbf{v}}_i)\|_2^2, \\ & s.t.\ S(\mathbf{x}) = \mathcal{N}'([\mathbf{x}_1^b, ..., \mathbf{x}_K^b]), \mathbf{x}_j^b = \left\{ \begin{array}{l} 1, \mathbf{x}_j \neq 0 \\ 0, \mathbf{x}_j = 0 \end{array} \right. , \end{aligned} \tag{10}$$

let $\mathbf{v}_i^b = S(\mathbf{v}_i)$ and $\widehat{\mathbf{v}}_i^b = S(\widehat{\mathbf{v}}_i)$ be the click position feature vectors corresponding to $\mathbf{v}_i$ and $\widehat{\mathbf{v}}_i$, respectively. Note that $S(\cdot)$ in (10) is a piecewise discontinuous function (refer to Fig. 7(a)), which is not differentiable everywhere. To facilitate the backpropagation for $\widetilde{\ell^p}$, we replace $S(\cdot)$ in (10) by a smooth normalized binarized function $\widehat{S}(\cdot)$ using the following sigmoid function (refer to Fig. 7(b)):

$$\widehat{S}(\mathbf{x}) = \mathcal{N}'(\frac{1}{1 + exp(-a(\mathbf{x} - \frac{B}{2}))}), a = \frac{2}{B} ln(\frac{1-T}{T}). \tag{11}$$

Compared with $S(\cdot)$, the differentiable function $\widehat{S}(\cdot)$ not only facilitates backpropagation during training but is also robust to precision-related errors. Note that the predicted click feature is obtained by word embedding, which involves many multiplication operations on float values. Therefore, element values in the predicted click vector may be extremely small but not exactly zero. Such precision-related error can be handled by the two slack variables B and T in (11).

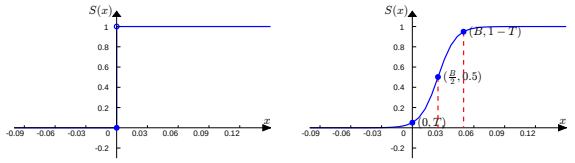
Using the position-sensitive loss function $\widetilde{\ell^p}$, we replace the count-based loss function ℓ^p in (2) with the following combined loss function $\widehat{\ell^p}$:

$$\widehat{\ell^p}(\mathbf{v}_i, \widehat{\mathbf{v}}_i) = (1-\tau) \left\| \mathcal{N}'(\mathbf{v}_i) - \mathcal{N}'(\widehat{\mathbf{v}}_i) \right\|_2^2 + \tau \left\| \widehat{S}(\mathbf{v}_i) - \widehat{S}(\widehat{\mathbf{v}}_i) \right\|_2^2, \tag{12}$$

where τ is the combination weight. In (12), prediction error on both the click count (first term) and the click position (second term) are minimized, whereas in (2), only the prediction error on the click count is constrained.

To ensure that the gradient trend is unchanged after integrating the position-sensitive loss function, τ should be controlled so that gradient magnitude of the weighted count-based loss function (first term in (12)) is similar to that of the weighted position-sensitive one (second term in (12)).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2921861, IEEE Transactions on Image Processing



(a) Indicator (b) Sigmoid

Fig. 7. Indicator function and its approximated sigmoid function in position-sensitive loss function. B and T, respectively, control the bandwidth and truncation threshold in converting the click count vector to a click position one.

To overcome the vanishing gradient problems, we perform batch normalization on the input of the sigmoid transformation ("SIGMOID" in Fig. 8), since batch normalization helps training a network with a sigmoid function efficiently [32].

2) Multitask Learning Framework. As aforementioned, the predictor learned by minimizing the prediction error may not be optimal for the recognition task; thus the prediction and recognition tasks should be considered simultaneously in learning click predictors. To address this issue, we employ the multitask learning framework and design a multitask loss function to minimize the prediction and recognition error simultaneously. The loss function is designed specifically for the source dataset with click information and is defined as the combination of the ℓ_2 distance and softmax loss as follows:

$$\ell(y_i, \mathbf{o}_i, \mathbf{v}_i, \widehat{\mathbf{v}}_i) = -(1-\lambda)\log(\frac{e^{\mathbf{o}_{y_i}}}{\sum_{j\in\mathcal{Y}} e^{\mathbf{o}_j}}) + \lambda\widehat{\ell^p}(\mathbf{v}_i, \widehat{\mathbf{v}}_i), \quad s.t.\ \mathbf{x}_i \in \mathcal{I}^s, \tag{13}$$

where λ is the tradeoff weight between prediction and recognition error.

Note that the multitask learning framework can also be introduced into (2). By slightly revising the loss function $\ell^p(\mathbf{v}_i, \widehat{\mathbf{v}}_i)$ in (2), we can construct an improved multitask transfer deep learning (MTTDL) model as follows:

$$\ell^p(\mathbf{v}_i, \widehat{\mathbf{v}}_i) = -(1-\lambda)\log(\frac{e^{\mathbf{o}_{y_i}}}{\sum_{j\in\mathcal{Y}} e^{\mathbf{o}_j}}) + \lambda\left\|\mathcal{N}'(\mathbf{v}_i) - \mathcal{N}'(\widehat{\mathbf{v}}_i)\right\|_2^2. \tag{14}$$

3) Multidomain Multimodal Training. In our problem, samples from the source $\mathcal{I}^s$ and target domain $\mathcal{I}^t$ differ in the input modal. The click data are contained only in $\mathcal{I}^s$, but absent in $\mathcal{I}^t$. We design a unified training framework with the cross-domain and cross-modal samples, and the loss function for samples in different domains is defined as:

$$\ell(y_i, \mathbf{o}_i, \mathbf{v}_i, \widehat{\mathbf{v}}_i) = \begin{cases} \lambda\widehat{\ell^p} - (1-\lambda)\log(\frac{e^{\mathbf{o}_{y_i}}}{\sum_{j\in\mathcal{Y}} e^{\mathbf{o}_j}}), & \mathbf{x}_i \in \mathcal{I}^s \\ -\mu\log(\frac{e^{\mathbf{o}_{y_i}}}{\sum_{j\in\mathcal{Y}} e^{\mathbf{o}_j}}), & \mathbf{x}_i \in \mathcal{I}^t. \end{cases} \tag{15}$$

D. The Proposed Model Structure

Using the improved click feature predictor and unified training strategy, we design our multitask and multidomain deep network as shown in Fig. 8 It learns a click prediction model and deep visual feature simultaneously. Datasets from multiple domains and modals are combined in the training. A nonlinear word embedding with a position-sensitive loss function is integrated to better learn the visual click correlation.

In Fig. 8, "ConvNet", "Nonlinear Word Embedding", and "FC" are particularly designed for visual feature extraction, click prediction, and recognition tasks, respectively. As the low-level feature space and visual click correlation are relatively stable in different domains, we employ the shared "ConvNet" and "Nonlinear Word Embedding" models across different domains, i.e., the two word embedding modules printed in red in Fig. 8 share the same parameters.

For "FC" layers, they can either have the same or different parameters in the two domains. Intuitively, sharing the "FC" layers seems to better capture the domain variance in transferring. However, when the recognition tasks differ considerably in multiple domains (the category sets differ considerably), constructing the same "FC" layers across different domains may be improper. For a better explanation, we denote the proposed model with different/shared "FC" layers in multiple domains as MTMDD-VM/MTMDD-VM*.

E. Extension for Unseen Category Recognition

In practice, a click dataset that contains the same categories as target one may not exist. Therefore, we present an extension of our method, namely, unseen category recognition. In the unseen category recognition scenario, an assistant source click dataset containing a different category set is employed to learn the visual click relationship, then adapted to the recognition task in the target one. For this task, we construct a simple yet efficient loss function by replacing the softmax function in (15) with the following one:

$$\ell(y_i, \mathbf{o}_i, \mathbf{v}_i, \widehat{\mathbf{v}}_i) = \begin{cases} \lambda\widehat{\ell^p}(\mathbf{v}_i, \widehat{\mathbf{v}}_i) - (1-\lambda)\log(\frac{e^{\mathbf{o}_{y_i}}}{\sum_{j\in\mathcal{Y}^s} e^{\mathbf{o}_j}}), & \mathbf{x}_i \in \mathcal{I}^s \\ -\mu\log(\frac{e^{\mathbf{o}_{y_i}}}{\sum_{j\in\mathcal{Y}^t} e^{\mathbf{o}_j}}), & \mathbf{x}_i \in \mathcal{I}^t, \end{cases} \tag{16}$$

where μ is the weight for loss functions on target domain with respect to the source domain. $\mathcal{Y}^s$ and $\mathcal{Y}^t$ ($\mathcal{Y}^t \neq \mathcal{Y}^s$) denote the category set in the source and target domain, respectively. For unseen category recognition, the source and target data can originate from the same (with different classes) or different objects, and we call them out-of-class and cross-object recognition, respectively.

IV. Experiment

We conduct extensive comparisons and validations in this section[2]. First, we show the experimental settings, including data preprocessing and feature generation. Second, we evaluate three main components in MTMDD-VM, i.e., the improved click prediction model, the multitask learning strategy, and the multidomain multimodal training. Both the nonlinear word embedding and position-sensitive loss function are evaluated. Third, we compare MTMDD-VM with many state-of-the-art methods. Finally, we demonstrate the scalability and one-shot ability of the proposed method. Specifically, to better evaluate

[2]The code can be downloaded from https://github.com/GYxiaOH/caffe.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2921861, IEEE Transactions on Image Processing

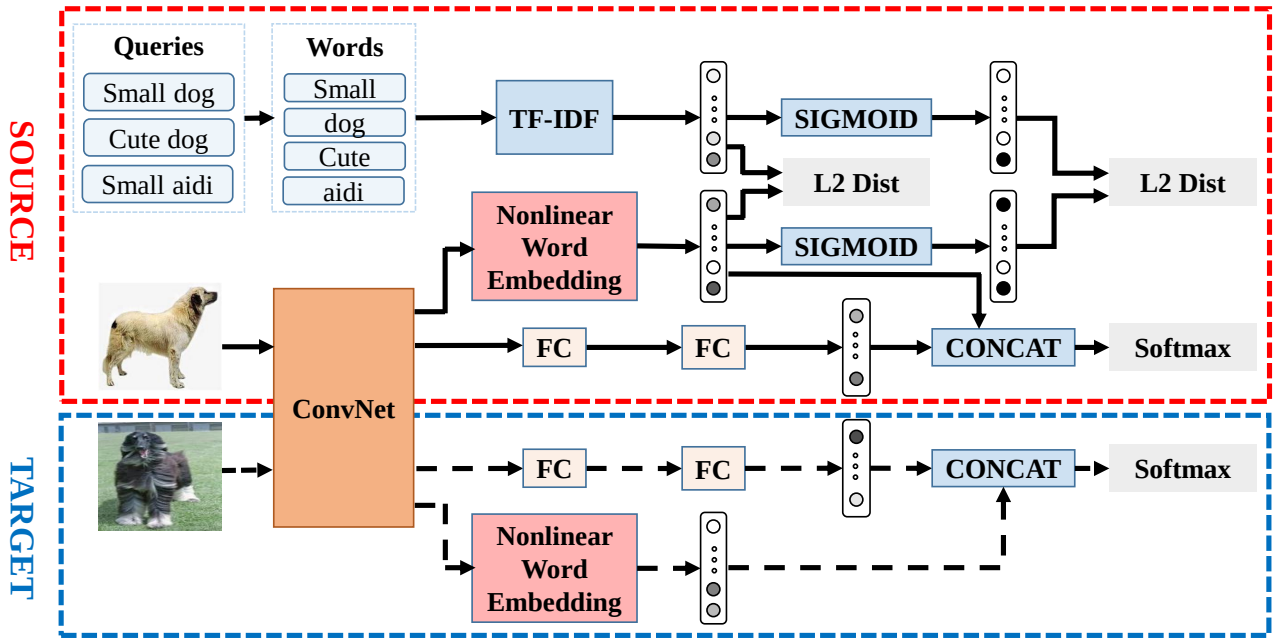


Fig. 8. Pipeline of our image recognition with the deep visual and predicted click feature. With the help of an assistant dataset containing click information, a deep visual model and click feature predictor are jointly learned. To address the cross-domain variance, datasets with (source) and without (target) click information are combined in the training. The prediction and recognition errors are minimized simultaneously.

the scalability to unseen categories, source and target data that originate from the same (out-of-class recognition) and different objects (cross-object recognition) are both tested.

Similar to [33], we combine three frequently used dog breed datasets, i.e., Stanford-Dog [34], the Columbia-Dog[3], and the ImageNet-Dog validation set [35] to evaluate the proposed approach. The three datasets serve as target domain dataset since none of them contain click data. We call the target dataset a "click-free" dataset in the following part. Specifically, we utilize the Clickture-Dog [4] as the source domain dataset, which is used to learn the click predictor.

For image recognition tasks, the recognition accuracies on the test set of both the source and target domains are evaluated, which are denoted by "Acc-S" and "Acc-T", respectively; for click prediction task, we test the root-mean-square prediction error (denoted as "RMSE-S") on the test set of the source dataset, which contains ground truth click information.

A. Experimental Settings

In this section, we introduce the used preprocessing for datasets in both domains and the ground truth click feature generation on the source dataset.

1) Data Preprocessing. The source Clickture-Dog dataset consists of dog images of 344 categories. To ensure a valid training/testing split, we filter out the categories that contain less than 3 images, and obtain a dog breed dataset with $95,041$ dog images of 283 categories. For the collection of the target dataset, we combine three frequently used dog datasets, namely Stanford-Dog, Columbia-Dog, and ImageNet-Dog. Finally, we collect $12,358$ dog images of 129 categories.

[3]http://faceserv.cs.columbia.edu/DogData/

We conduct data cleaning on the target Clickture-Dog dataset to adress the heavy noises in its associated click data. Additionally, with no publicly available training/testing split, we split both the source and domain datasets ourselves.

The data cleaning procedure is similar to [33]. First, using the original Clickture-Dog dataset, we train a dog classifier by fine-tuning the VGGNet-19, which is pretrained on ImageNet. Second, we filter out the noisy images based on both the click count and the probability in the dog classifier. Due to the large unbalance in data frequency among different categories, we slightly adjust the probability threshold used in [33] in filtering out noisy images. The used probability thresholds in different categories are listed in Table II. Note that "First"/"Second" in the table denotes the images with highest/lowest 50% click count (refer to [33]). After image filtering, we obtain $32,691$ dog images for the source Clickture-Dog dataset.

In Clickture-Dog, we filter out the dog images in categories that are absent in the target data since: 1) category differences in multiple domains may result in a large data distribution difference, therefore, the visual feature and visual click correlation models trained with all source categories may be inapplicable for the target domain; and 2) utilizing all categories in Clickture-Dog poses difficulties in generating a word vocabulary suitable for both domains. The vocabulary built in the source domain may be improper to describe target images, while that built in the target domain makes the ground truth click feature of some source images be zero vectors.

Furthermore, to ensure data balance, we randomly select 300 images for a category that contains more than 300 samples. Finally, we obtain $19,833$ dog images on source dataset. With no available training/testing sets for the used

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2921861, IEEE Transactions on Image Processing

TABLE II
PROBABILITY THRESHOLD USED IN FILTERING OUT NOISY IMAGES FOR DIFFERENT CATEGORIES. n IS THE IMAGE NUMBER IN A CATEGORY.

Category size	$n < 8$	$8 \leq n < 100$		$100 \leq n < 300$		$n \geq 300$	
Click count	All	First	Second	First	Second	First	Second
Probability	All	< 0.2	< 0.4	< 0.3	< 0.5	< 0.4	< 0.6

dataset[4], we randomly split both the source and target datasets into three parts: 50% for training, 25% for validation, and 25% for testing. The final reduced datasets and the corresponding splits are publicly available[5].

2) Click Feature Generation. Similar to [7], we generate the ground truth click feature via the TF-IDF algorithm. First, using all queries with a nonzero click count in the source dataset, we obtain $39,482$ words by word segmentation on the queries. Second, we select $1,000$ word items with the largest click count to generate a word vocabulary. Finally, we generate a $1,000$-D ground truth click feature for each image using its clicked words and vocabulary with TF-IDF algorithm.

B. Components of MTMDD-VM

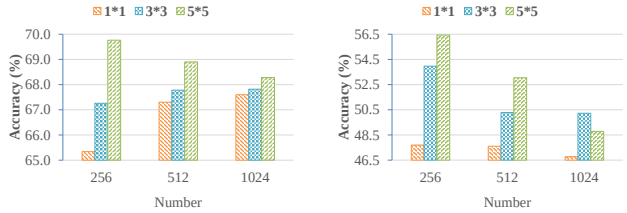
There are three important components in MTMDD-VM: improved click predictor, multitask learning framework, and the multidomain multimodal training. We evaluate the effect of each component in this section.

1) Improved Click Predictor. Note that in MTMDD-VM, the predicted click feature is combined with the deep visual one in recognition. To better illustrate the benefit of the improved click prediction model, we first test the recognition accuracies only with predicted click feature. Two components in the improved click predictor, i.e., the nonlinear word embedding and the position-sensitive loss function, are evaluated. Afterwards, we combine the predicted click feature with the deep visual feature in the recognition task. In this subsection, the conventional transferred deep learning framework (TDL in Section III-B) is employed, wherein a click predictor is first trained on source data and then applied on the target one.

a) Nonlinear Word Embedding. Compared to [7], in the word embedding model, we add a convolutional layer and fully connected layer before the original linear word embedding layer to increase nonlinearity. We first investigate different kernel sizes and numbers in the convolutional layer. The recognition accuracies with varied convolutional kernels are shown in Fig. 9, wherein both source and target domains are tested. It can be observed that a larger kernel size often yields better results. A reasonable explanation is that the click feature captures high-level semantical information, which should be better transformed from a visual feature with more global information. Intuitively, a larger convolutional kernel size often introduces relatively global features. To maximize the performance, in the following experiments we set the kernel size and number as 3×3 and $1,024$, respectively.

[4]The training/testing split in [33] is not publicly available.

[5]The used dog dataset can be downloaded from https://pan.baidu.com/s/1vUAvlpB-8ZcafN0Gh1scDw.



(a) Source domain (b) Target domain

Fig. 9. Recognition accuracies using different convolutional transformations with varying kernel size and number. Both source and target domains are tested.

TABLE III
RECOGNITION ACCURACIES USING TRANSFERRED DEEP NETWORKS (REFER TO FIG. 3(A)) WITH DIFFERENT WORD EMBEDDING MODELS. MULTIPLE NONLINEAR OPERATORS ARE EVALUATED. BOTH SOURCE AND TARGET DOMAINS ARE TESTED.

Model	Linear	Nonlinear		
	FC	biFC	CONV+FC	CONV+biFC
Acc-S (%)	63.0	68.7	68.9	70.3
Acc-T (%)	44.2	54.0	51.6	**57.3**

In addition, we also test the performance of the biFC structure (two fully connected layers with the RELU operator) in nonlinear word embedding. The result is shown in Table III. It is found that both the convolutional and biFC layers improve the performance of the traditional linear word embedding model, i.e., using a single "FC" layer. In the following experiments, we employ the nonlinear word embedding with the combined convolutional and biFC layers.

b) Position-sensitive Loss Function. There are three parameters in the position-sensitive loss function (12): bandwidth B, truncated threshold T, and the combination weight τ. It is found that T have little impact on recognition accuracy.

With fixed $\tau = 0.1$ and $T = 0.1$, we investigate the effect of B, and the result is shown in Table IV. In addition to the recognition accuracies in both domains, we also list the ratio of input elements to the sigmoid transformation that falls into range [0,B] (denoted by "Ratio" in Table IV). Note that the optimal B "seems" small, and the reason is twofold: 1) a large B easily brings in a click position vector far away from the ideal one. Note that elements of an ideal click position vector should be 0 or 1, which can be converted from the click count one with an indicator function shown in Fig. 7(a). The sigmoid transformation is designed to replace the indicator function in order to address its discontinuity. Therefore, to generate an ideal click position vector, B should be controlled to be small so that the sigmoid transformation can better approximate the distribution of the indicator function; and 2) the value of B should be set based on the input distribution of the sigmoid transformation. Owing to the batch normalization operation, the input elements of the sigmoid function have already been scaled to a small value. Compared with the input of the sigmoid function, B is not very small when $B = 0.01$, and the recognition accuracy is damaged if B further decreases. As shown in Table IV, the recognition accuracy decreases when reducing B to 0.001, and it is unacceptable when B is reduced

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2921861, IEEE Transactions on Image Processing

to 0.0001. In addition, due to these small input elements, a large B brings in more input elements falling into the range [0,B], resulting in an undesirable click position vector (refer to "Ratio" in Table IV). To maximize the performance in both domains, we set $B = 0.01$ and $T = 0.1$[6], respectively.

With the optimal B and T, we test different τ and the result is shown in Table V. It shows that: 1) though the single position-sensitive loss function performs badly ($\tau = 1$), integrating the position-sensitive loss function into the count-based one ($0 < \tau < 1$) performs better than using a single count-based loss function ($\tau = 0$), implying that position-sensitive loss provides valuable complementary information for the count-based one; and 2) a small τ often results in better result. A reasonable explanation is that compared with the count-based loss function, the position-sensitive one may introduce vanishing gradient problems due to the sigmoid function; a large τ makes the position-sensitive loss function dominate the gradient of the combined loss. Therefore, to restrain the gradient influence of the position-sensitive loss function on the combined loss function, we employ a relatively small τ. As shown in Table V, when $\tau = 0.1$, the gradient magnitude ratio[7] of the weighted position-sensitive loss function (second term in (12)) with respect to weighted count-based one (first term in (12)), i.e., "gd-ratio", is close to 1. It ensures that the backpropagating trend is unchanged after integrating the position-sensitive loss function. In the following experiments, we use the optimal $\tau = 0.1$ for both higher recognition accuracy and a safer backpropagating gradient.

We also show the backpropagation of the gradient magnitude for different layers in the click prediction model using loss functions based on the click count (first item in (12)), the click position (second item in (12)), and their combination (both items in (12)). They are denoted as "Count", "Position", and "Combined" in Fig. 11, respectively. It demonstrates that, compared with the original count-based loss function, integrating the position-sensitive loss function does not change the gradient trend during training. As shown in Fig. 11, though the gradient of the position-sensitive loss function fluctuates more wildly than that of the count-based loss function in early iterations, the gradient magnitudes of the count-based loss, the position-sensitive loss, and their combination are prone to become the same in the late iterations (after $3,000$th iteration). The noticeable fluctuation of gradient of the position-sensitive loss function in early iterations is mainly caused by the sigmoid function, resulting in a gradient fluctuation.

In addition, to better demonstrate the advantage of the position-sensitive loss function in click prediction, we visualize the predicted click features for some samples. Both the ground truth and click features predicted from models with and without position-sensitive loss functions are illustrated. As shown in Fig. 10, compared with models without position-sensitive loss functions, models with a position-sensitive loss function generate a better click feature that is much closer to the ground truth click feature, owing to the additional penalty on click position constraints. Those feature items that are mispredicted by models without position-sensitive loss functions can be corrected by the position loss in the proposed model (plotted in the red circle in Fig. 10).

[6] B is selected from the range exponentially growing from 0.0001 to 1.

[7] The gradient magnitude is computed as the ℓ_2 norm of the gradient vector of the loss functions to all parameters at iteration $1,500$.

TABLE IV
RECOGNITION ACCURACIES USING THE POSITION-SENSITIVE LOSS FUNCTION WITH DIFFERENT B. "RATIO" DENOTES THE AVERAGE RATIO OF FEATURE ELEMENTS FALLING INTO THE RANGE [0,B]. BOTH SOURCE AND TARGET DOMAINS ARE TESTED.

B	10	1	0.1	0.01	0.001	0.0001
Acc-S (%)	59.7	70.7	71.4	71.5	69.5	0.7
Acc-T (%)	43.6	58.3	58.4	58.9	57.3	1.5
Ratio (%)	57.53	34.36	3.90	0.27	0.04	0.01

TABLE V
COMPARISON OF MODELS WITH COUNT-BASED LOSS FUNCTION ($\tau = 0$), POSITION-SENSITIVE LOSS FUNCTION ($\tau = 1$), AND THEIR COMBINATION ($0 < \tau < 1$). RECOGNITION ACCURACIES ON BOTH DOMAINS ARE TESTED. "GD-RATIO" DENOTES GRADIENT-MAGNITUDE-RATIO OF WEIGHTED POSITION-SENSITIVE LOSS (SECOND TERM IN (12)) WITH RESPECT TO WEIGHTED COUNT-BASED ONE (FIRST TERM IN (12)).

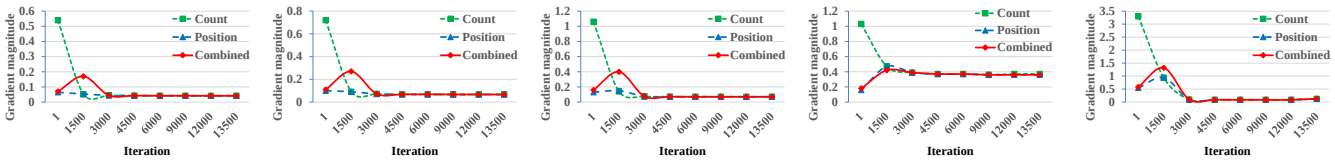
τ	Count	Combined					Position
	0	0.05	0.1	0.15	0.2	0.4	1
Acc-S (%)	70.3	71.5	71.5	70.5	69.0	38.6	11.8
Acc-T (%)	57.3	58.8	**58.9**	57.9	56.3	29.3	7.9
gd-ratio	-	0.31	0.99	2.21	3.63	19.84	-

c) Combination with the Visual Feature. Recognition accuracies with the combined visual and predicted click feature are illustrated in Table VI. It can be found that: 1) the combined feature performs better than the single deep visual or click feature; and 2) the click feature generated from the proposed click prediction model (C) performs better than the traditional linear one (C [7]), and both the nonlinear word embedding and position-sensitive loss function are beneficial to performance. Note that C_E and C performs better than C [7] and C_E, respectively, regardless of whether the visual feature is integrated.

2) Multitask Learning Framework. As aforementioned, in the multitask learning framework, both the recognition and prediction errors are minimized simultaneously. We evaluate the multitask learning framework by comparing the performances on deep transfer models using multiple loss with those using single loss. The performance is evaluated by both the image recognition and click prediction accuracies. The root mean square error on the source click data ("RMSE-S" in Table VII) is employed to measure the prediction error.

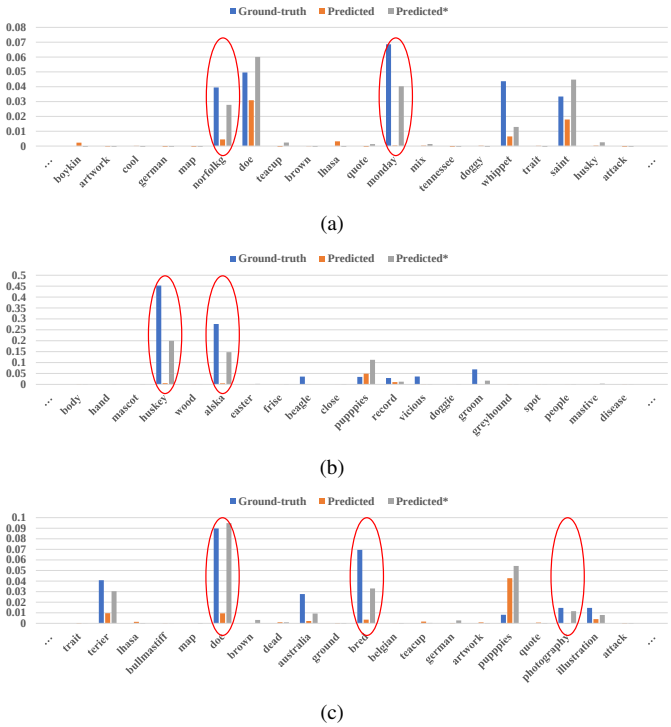
The multitask learning is tested on models with both unshared/shared "FC" layers (refer to Section III-D), i.e., MTMDD-VM/MTMDD-VM*, wherein μ is set to be 1. Different λ in (14) that balance the recognition and prediction loss are tested. To make a more comprehensive comparison, for the single-task learning framework, we test both the models with only prediction ($\lambda = 1$) and recognition ($\lambda = 0$) error. The result is shown in Table VII, and it can be found that: 1) λ seems to have little impact on the performance, implying the high correlation between the prediction and recognition tasks; 2) both the improvements of MTMDD-VM and MTMDD-VM* over MDD-VM and MDD-VM* demonstrate the benefit

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2921861, IEEE Transactions on Image Processing



(a) CONV1 (b) CONV3 (c) CONV5 (d) FC1 (e) FC2

Fig. 11. Backpropagating the gradient-magnitude for different layers in the click prediction model with loss functions based on click count (first item in (12)), click position (second item in (12)), and their combination (both items in (12)). For each layer, the gradient magnitude is computed as the ℓ_2 norm of the gradient vector of the loss function with respect to all parameters.



(a) (b) (c)

Fig. 10. The predicted click features for some image samples. The click feature vector predicted by models without and with the position-sensitive loss function are denoted as "Predicted" and "Predicted*", respectively. 20 words are randomly selected as the basis.

TABLE VI
RECOGNITION ACCURACIES WITH THE IMPROVED CLICK PREDICTION MODEL COMPARED WITH THE TRADITIONAL ONE (C [7]). THE PREDICTED CLICK FEATURE C AND ITS COMBINATION WITH DEEP VISUAL FEATURE V ARE TESTED. BESIDES, THE NONLINEAR WORD EMBEDDING (C_E) AND POSITION-SENSITIVE LOSS FUNCTION (C) ARE TESTED.

Features	V	C [7]	C_E	C	$V+C$ [7]	$V+C_E$	$V+C$
Acc-S (%)	76.5	63.0	70.3	71.5	76.4	76.6	**77.5**
Acc-T (%)	70.7	44.2	57.3	58.9	71.3	71.6	**72.2**

of the multitask learning framework. Though the click feature seems to be predicted more accurately (smaller RMSE) when only using the prediction loss function ($\lambda = 1$), it fails in recognition tasks (with lower recognition accuracies in both source and target domains); and 3) the accuracy improvement of the multitask loss model over single loss (denoted by "ΔAcc-T" in Table VII) is more noticeable for the shared model MTMDD-VM* compared with un-shared one MTMDD-VM. To maximize the performance, we set $\lambda = 0.4$ in the following experiments.

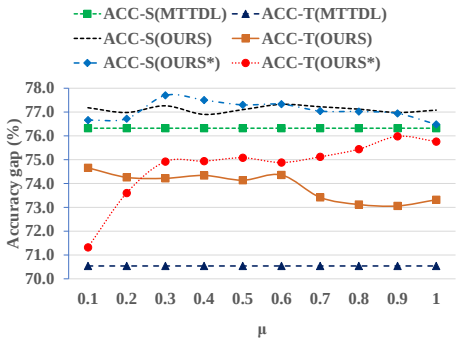


Fig. 12. Comparison between MTMDD-VM (denoted by "OURS") and MTTDL under different softmax loss weight μ. Both domains are tested.

3) Multidomain Multimodal Training. We demonstrate the advantage of the unified training scheme by testing the accuracy improvement of the multidomain multimodal training (MTMDD-VM/MTMDD-VM*) over the individual training (MTTDL) strategy. Note that MTTDL is the multitask transfer learning framework that trains datasets of different domains separately (refer to Section III-B).

We test different μ, controlling the softmax loss weight for loss functions in the target domain respect to the source domain, and the result is shown in Fig. 12. Models with both the un-shared/shared "FC" layers (refer to Section III-D) are tested. It can be found that: 1) training a unified network with the combined dataset of both domains (MTMDD-VM/MTMDD-VM*) performs much better than that with a single-domain dataset (MTTDL); 2) smaller μ may result in better performance[8], indicating the source dataset may be more reliable than the target one for training. A reasonable explanation is that the predicted click feature is more accurate in the source dataset compared with that in the target one, since the ground truth click for constraining the prediction error is provided only in the source domain; and 3) MTMDD-VM* performs slightly better than MTMDD-VM, implying that sharing the "FC" layers better captures the cross-domain variance during transferring. To maximize the performance on the target dataset, in the following parts we set $\mu = 0.1$ and $\mu = 0.9$ for MTMDD-VM and MTMDD-VM*, respectively.

[8]The recognition accuracy is very low when $\mu > 1$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2921861, IEEE Transactions on Image Processing

TABLE VII
COMPARISON BETWEEN MULTITASK LOSS AND SINGLE-LOSS DEEP LEARNING MODELS. DIFFERENT λ IN THE LOSS COMBINATIONS ARE TESTED. "ΔACC-T" DENOTES THE ACCURACY GAP OF MTMDD-VM (MTMDD-VM*) WITH RESPECT TO MDD-VM (MDD-VM*) WITH ONLY PREDICTION LOSS ($\lambda = 1$). BOTH SOURCE AND TARGET DOMAINS ARE TESTED.

Model	MTMDD-VM				MDD-VM		MTMDD-VM*				MDD-VM*	
λ	0.8	0.6	0.4	0.2	0	1	0.8	0.6	0.4	0.2	0	1
Acc-S (%)	77.0	77.2	77.1	76.7	77.0	0.1	73.9	75.5	76.6	76.4	77.5	40.1
RMSE-S (10^{-2})	81.8	81.9	81.9	82.2	100.0	**81.7**	82.0	84.0	87.0	85.0	101.0	**81.0**
Acc-T (%)	73.1	73.1	**73.3**	72.3	71.9	71.5	75.2	75.2	**75.6**	74.4	75.4	63.5
ΔAcc-T (%)	1.6	1.6	**1.8**	0.6	0.4	-	11.7	11.7	**12.1**	10.9	11.9	-

C. Comparison with State-of-the-art Methods

We compare MTMDD-VM with some state-of-the-art methods. Note that in existing works, only visual features are used for the recognition task on these click-free dog datasets, and this is the first paper to use the predicted click feature for image recognition. Therefore, we compare MTMDD-VM with several recent visual feature-based models. In addition, as a word embedding model is adopted in the proposed model, we also compare other word embedding approaches for click prediction. For each algorithm, we use the optimal parameter. The compared methods can be categorized as follows:

1) Visual Feature-based Approaches. We compare two visual features with different multidomain training strategies: 1) combining source and target images in the training, i.e., unified training. They are the domain adaptation network (DAN) [36] and one domain-adversarial neural network (DANN)[9] [19]; 2) for training on the source data and adapting to target data, i.e., fine-tuning. Two VGGNet-based models[10] and two state-of-the-art fine-grained image recognition models, namely the recurrent attention convolutional neural network (RACNN) [37][11] and the selective convolutional descriptor aggregation (SCDA) model [38][12], are tested.

2) Approaches based on Combined Visual and Predicted Click Feature. Two schemes are tested: 1) score fusing (denoted by "SF" in Table VIII) on two pre-trained recognizers, e.g., the visual feature-based and predicted click feature-based one. Specifically, we employ a weighted combination[13] for the two recognizers. 2) training a unified recognizer with a combined visual and predicted click feature. We customize two word embedding models for click feature prediction, and they are the transferred deep model (TDL) proposed in [7] and the multimodal embedding model with a hard mining strategy VSE++ [39]. For TDL and VSE++[14], using word embedding models learned on source click data, we predict click features of images in the target data, which are integrated with their original visual features. A softmax loss-based network is trained with the combined features on target data[15]. For TDL, we also test its multitask (MTTDL) and multidomain (the proposed MTMDD-VM/MTMDD-VM*) versions.

The results are reported in Table VIII. Note that "TDL [7]" and "TDL" in Table VIII differ in the click prediction model structure. In "TDL [7]", the traditional linear word embedding proposed in [7] is used, while in "TDL", the proposed nonlinear word embedding with position-sensitive loss function is employed. It can be found that:

- Integrating the predicted click feature into the visual one enhances the performance, implying that the predicted click feature provides complementary semantical information for visual features in recognition tasks.
- For visual feature-based approaches, employing domain adaptation or designing powerful fine-grained image features results in satisfactory performances. The improvement of VGG_{src} over VGG_{img} indicates the importance of model initialization in learning deep networks.
- For recognition with the predicted click feature: 1) training a recognizer with the combined visual and predicted click feature outperforms the direct score fusion on multiple pre-trained recognizers since it can better discover inter-domain feature correlation; 2) when transferring the predictor from the source click data to a click-free target one, both multitask (MTTDL) and unified training (MTMDD-VM/MTMDD-VM*) are helpful in learning a domain-invariant click feature. The improvements of MTTDL and MTMDD-VM over TDL and MTTDL demonstrate the benefit of the multitask and multidomain learning framework, respectively; and 3) MTMDD-VM performs better than TDL/MTTDL and VSE++ since it learns a better visual semantic embedding model.
- MTMDD-VM outperforms other state-of-the-art methods in recognition on the target click-free dataset, even compared with the recent sophisticated feature proposed in RACNN/SCDA. The improvements of MTMDD-VM over some visual feature-based methods are not noticeable, since in MTMDD-VM, the click feature is predicted from the visual feature learned by VGGNet-19. We believe that if we construct a word embedding model based on RACNN, SCDA, or other state-of-the-art visual models, we can achieve a further improvement

[9]For DAN and DANN, the moment, and weight-decay are set to be 0.9 and 0.0005, and the initial learning rate is set to be 0.001 (0.0001) for DAN (DANN), respectively.

[10]For both VGG_{img} and VGG_{src}, the moment and weight-decay are set to be 0.9 and 0.0005, respectively, and the initial learning rate is set to be 0.1 (0.01) for VGG_{img} (VGG_{src}).

[11]For RACNN, the initial learning rate, moment and weight-decay are set to be 0.1, 0.9 and 0.0005, respectively.

[12]The penalty parameter in the linear SVM classifier is set to 1.

[13]The weight of the predicted click feature-based recognizer is selected from $[0.2, 0.4, 0.5, 0.6, 0.8]$, and the optimal weight is set to be 0.2.

[14]For VSE++, we use the optimal parameters as employed in constructing the word embedding model.

[15]For training the softmax-loss-based network, we set the moment and weight-decay as 0.9 and 0.0005, and the initial learning weight as 0.3 (0.1) for the visual (click) feature.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2921861, IEEE Transactions on Image Processing

in recognition accuracy.

D. Additional Advantages of MTMDD-VM

In this section, we show two Additional Advantages of MTMDD-VM, i.e., the scalability to unseen categories and one-shot learning ability.

1) Unseen Categories. We show the scalability to unseen categories of MTMDD-VM by testing the recognition accuracies on the target dataset with some categories absent in the source dataset. Both the out-of-class and cross-domain scenarios are evaluated.

a) Out-of-class Recognition. We show the scalability of *out-of-class* recognition, i.e., using limited source data to learn knowledge and recognizing on target dataset with new categories that are absent in the source data. We test the performance with increasing category numbers in the source domain[16]. More specifically, we randomly select a subset of 30 categories and then increase the dataset by randomly selecting subsets from the remaining categories. Finally, we obtain a set of datasets with $30, 60, 90, 120$, and 129 categories successively. In addition, we have also tested recognition accuracies when using dog images in all source categories (283 categories in total), including those categories absent in the target domain. Note that when using all 283 categories in the source Clickture-Dog, we did not test MTMDD-VM* since the dimension of matrix encoding the FC layers in source domain differs from that in target domain, making it impossible to train a shared FC layer in the two domains.

The out-of-class recognition of the proposed method is shown in Fig. 13, wherein each curve denotes the recognition accuracies on the target dataset by transferring from the source data with different category numbers. We compare the scalability of the proposed method with "VGG$_{img}$", "VGG$_{src}$", and "TDL". Note that only the deep visual model is transferred for "VGG$_{img}$" and "VGG$_{src}$", while for "TDL" and our method, both the visual feature and click prediction models are transferred across domains simultaneously.

It can be found that: 1) the performance decreases as the category number is reduced in the source domain for each algorithm, indicating that the transfer is more difficult when more unseen categories need to be recognized in the target data; 2) the proposed method still performs better than others with reduced categories, implying its powerful scalability to unseen categories. A reasonable explanation is that with the unified training using samples from different domains, the cross-domain correlation can be better learned. Note that the performance is still satisfactory for testing on all 129 target categories when only samples of 30 source categories are available in the source domain; 3) compared with MTMDD-VM, MTMDD-VM* decreases more sharply with reducing the category numbers, implying that constructing the same "FC" layers across domains with large differences in the category set may be improper; and 4) when using dog images in all source categories (283 categories in total), the top-1 accuracies

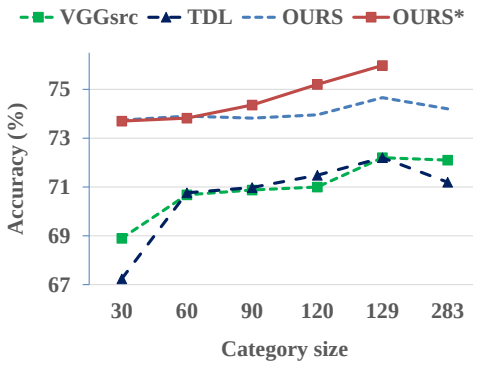


Fig. 13. Comparison of scalability between the proposed MTMDD-VM (denoted by "OURS") and other methods. It is shown by the accuracy comparison using subsets of source data with increasing categories. The accuracies are tested on the target domain.

are even slightly worse than the accuracies when using dog images in shared categories (129 categories) in both domains. The performance decrease is mainly caused by both the data distribution gap of multiple domains and the difficulty in generating word vocabulary as mentioned Section IV-A1.

b) Cross-object Recognition. We show the cross-object recognition ability of MTMDD-VM by testing the recognition accuracy in the dog (bird) target data using the source bird (dog) data with click information. A comparison between the cross-object recognition and the single-object one is conducted. In the single-object recognition scenario, both source and target data are of the same object, i.e., dog or bird data.

The source and target dog data are exactly the same as those used in previous experiments. For bird breed recognition, we use CUB-200-2011 [40] as the target "click-free" dataset, and employ a recent "Clickture-Bird" dataset established in [5] as the source domain dataset. Clickture-Bird is denoised using a data cleaning procedure similar to [33]. Similar to Section IV-A1, we only select images of categories that are present in both the source and target bird data to make the source and target bird data share the same category set; thus, we obtain a subset of bird datasets with 71 subcategories. In addition, categories that contain less than 3 images are discarded to ensure a valid training/testing split[17]. Similar to Section IV-A2, we generate a $1,000$-D ground truth click feature vector for each image in Clickture-Bird via the TF-IDF algorithm, wherein a bird word vocabulary with $1,000$ items is created.

The result is shown in Table IX, wherein "Single" and "Cross" denote single-object and cross-object recognition accuracies, respectively. It can be found that: 1) when adapting the visual click correlation from the source data to the target one, the recognition accuracies are similar in both single- and cross-object scenarios. It is because the visual click correlation is relatively stable and universal for different objects, which is the major rationale behind the proposed approach; 2) compared with other approaches, the proposed MTMDD-VM

[16]The categories are randomly selected from the whole dataset (129 categories in total).

[17]The final used bird data can be downloaded from https://pan.baidu.com/s/1Znn3q3wHyL-PnQY4m_Ec2Q.

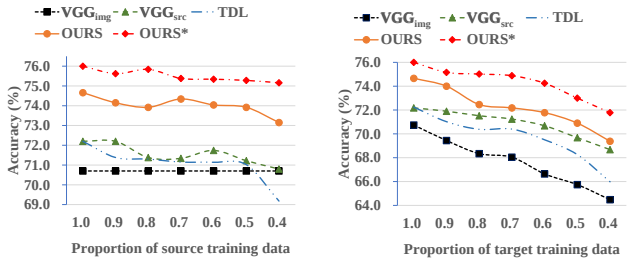
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2921861, IEEE Transactions on Image Processing

TABLE VIII
COMPARISON BETWEEN MTMDD-VM (DENOTED BY "OURS") AND SOME STATE-OF-THE-ART METHODS. THE CLICK FEATURE IS PREDICTED BY DIFFERENT TRANSFER METHODS. BOTH SOURCE AND TARGET DOMAINS ARE TESTED.

Feature	Visual						Combined Visual and Click						
Method	DAN	DANN	VGG_{img}	VGG_{src}	RACNN	SCDA	SF	VSE++	TDL [7]	TDL	MTTDL	OURS	OURS*
Acc-S (%)	77.1	67.9	76.5	76.5	75.9	79.0	75.2	76.5	76.4	77.5	77.1	77.3	**77.7**
Acc-T (%)	74.9	45.8	70.7	72.2	71.6	74.8	70.7	74.3	71.3	72.2	73.3	74.7	**76.0**

TABLE IX
CROSS-OBJECT RECOGNITION OF MTMDD-VM COMPARED WITH OTHER METHODS. IN THE CROSS-OBJECT SCENARIO, CLICKTURE-BIRD (CLICKTURE-DOG) IS USED AS SOURCE DATA FOR RECOGNITION IN THE DOG (BIRD) TARGET DATASET. RECOGNITION ACCURACIES (%) ON THE TARGET CLICK-FREE DATA ARE TESTED.

Dataset	VGG_{src}		TDL		MTMDD-VM	
	Single	Cross	Single	Cross	Single	Cross
Dog	72.2	71.4	71.3	70.2	74.7	73.5
Bird	71.7	70.3	66.9	66.9	73.8	70.7



(a) Source domain (b) Target domain

Fig. 14. Comparison of one-shot learning capability between the proposed method MTMDD-VM (denoted by "OURS") and some state-of-the-art methods. It is shown by the comparison of different methods with decreasing samples on both domains. The accuracies are tested on the target domain.

always preforms better under both single- and cross-object scenarios, showing its advantages in cross-object recognition.

2) One-shot Learning. We evaluate the one-shot capability by testing the performances via decreasing the proportion of training samples α in both source and target domains[18].

The results with reduction of source and target training samples are shown in Fig. 14(a) and Fig. 14(b), respectively. It can be found that: 1) the performance decreases when reducing training samples either in the source or target domain. The reason lies in that reducing source and target samples, respectively, affects the click prediction and recognizer learning procedure; 2) with reduced training samples, the performance of "TDL"/"VGG" exhibits a sharper decrease compared with ours. A reasonable explanation is that "TDL"/"VGG" are trained individually in different domains, and reducing samples on one domain largely affects the model initialization or training procedure; in our method, data in both domains are trained simultaneously, resulting in a smaller effect when decreasing samples in one domain.

[18]For a specific proportion α, we select $\lceil \alpha \times n_k \rceil$ training samples for each category k (n_k is the number of samples in category $k \in [1, 2, \ldots, 129]$) in source and target domains. We set $\alpha = [1, 0.9, \ldots, 0.4]$.

V. CONCLUSIONS AND DISCUSSIONS

A. Conclusion

We address the problem of image recognition with user click data, a kind of feedback information. As click data are absent in many image datasets, we employ word embedding to learn a click predictor from an assistant dataset containing click information (source dataset), then adapt it to the target click-free dataset. Specifically, we present a novel multitask multidomain transfer learning framework for the adaptation, wherein the predictor and recognizer are simultaneously learned. A nonlinear word embedding and position-sensitive loss function are integrated to handle the isomerism of the visual click spaces and the sparsity in the click feature. Additionally, a multitask deep learning framework is designed to ensure that the predicted click achieves higher accuracy in both prediction and recognition tasks, and datasets from multiple domains are combined during training.

We evaluate our approach on three public dog datasets (target dataset), and employ the Clickture-Dog dataset as the source dataset. It shows that both the nonlinear embedding structure and position-sensitive loss function can improve recognition accuracy in both domains. Moreover, the multitask learning framework helps to learn a better click feature with higher accuracy in both click prediction and image recognition tasks, and the unified training strategy with different domains can further improve the performance. The proposed model outperforms many state-of-the-art approaches. In addition, we show the scalability and one-shot learning ability of our method, which are demonstrated by the stable improvement of our method over others as unseen categories are increased and training samples are decreased.

B. Limitations

Despite the promising results, our current system suffers from some limitations in several domain transfer settings.

First, the proposed method still has some difficulties addressing adapting across multiple domains originated from different objects. As shown in Table IX, adapting the visual click correlation across domains in cross-object scenarios performs worse than that in single-object scenarios. Fig. 15 shows some samples mis-classified in cross-object scenario but corrected classified in single-object scenario. It can be observed that most of the mis-classified dog images are with complex background or huge appearance differences from birds, implying the vocabulary built on a specific object is somewhat inapplicable for images of a different object.

Second, in the proposed method, both domains are well-labeled, such that we can design softmax loss on both domains

Fig. 15. Some dog samples mis-classified in cross-object scenario but corrected classified in single-object scenario.

to bridge the cross-domain gaps. However, our current system may suffer from several problems when different domains are with different annotation degree, i.e., well-labeled data in the source domain while partially labeled/unlabeled in the target domain.

C. Future Work

Future work will concentrate on several open problems: 1) building a larger image recognition dataset containing click data to compressively evaluate the robustness of the proposed method for general transfer learning settings. These datasets are required to cover more object categories (e.g., cars, plants); 2) constructing a hierarchical click prediction model to generate the click feature with structured semantics; 3) integrating Maximum Mean Discrepancy (MMD) loss [36] into the transfer framework to address multiple domains with vast differences, i.e., transferring from a dog to a plant dataset; 4) extending the model to conduct adaptation across more than two domains; and 5) integrating a weakly supervised training procedure [41], [42] to deal with different domains with varied annotation degrees. Also, integrating this technique to automatically select better samples and learn recognition models simultaneously.

VI. Acknowledgments

This work was supported by Zhejiang Provincial Natural Science Foundation of China (No.LY19F020038), National Natural Science Foundation of China (No.61602136, No.61622205, No.61472110, No.61702143, and No.61601158), Australian Research Council Projects (FL-170100117, DP-180103424, and IH-180100002), and Zhejiang Provincial Key Science and Technology Project Foundation (No.2018C01012).

References

[1] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur, "Birdsnap: Large-scale fine-grained visual categorization of birds," in *Proc. CVPR*, 2014, pp. 2019–2026.

[2] A. Iscen, G. Tolias, P. H. Gosselin, and H. Jegou, "A comparison of dense region detectors for image search and fine-grained classification," *IEEE Trans. Image Processing*, vol. 24, no. 8, pp. 2369–81, 2015.

[3] G. Zheng, M. Tan, J. Yu, Q. Wu, and J. Fan, "Fine-grained image recognition via weakly supervised click data guided bilinear cnn model," in *Proc. ICME*, 2017, pp. 661–666.

[4] X.-S. Hua, L. Yang, J. Wang, J. Wang, M. Ye, K. Wang, Y. Rui, and J. Li, "Clickage: Towards bridging semantic and intent gaps via mining click logs of search engines," in *Proc. ACMMM*. ACM, 2013, pp. 243–252.

[5] M. Tan, J. Yu, Z. Yu, F. Gao, Y. Rui, and D. Tao, "User-click-data-based fine-grained image recognition via weakly supervised metric learning," *ACM Trans. Multimedia Computing, Communications, and Applications*, vol. 14, no. 3, p. 70, 2018.

[6] M. Tan, J. Yu, Q. Huang, and W. Wu, "Click data guided query modeling with click propagation and sparse coding," *Multimedia Tools and Applications*, vol. 77, no. 17, pp. 22 145–22 158, 2018.

[7] Y. Bai, K. Yang, W. Yu, C. Xu, W.-Y. Ma, and T. Zhao, "Automatic image dataset construction from click-through logs using deep neural network," in *Proc. ACMMM*. ACM, 2015, pp. 441–450.

[8] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev, "Panda: Pose aligned networks for deep attribute modeling," in *Proc. CVPR*, 2014, pp. 1637–1644.

[9] A. Vedaldi, S. Mahendran, S. Tsogkas, S. Maji, R. Girshick, J. Kannala, E. Rahtu, I. Kokkinos, M. B. Blaschko, D. Weiss, B. Taskar, K. Simonyan, N. Saphra, and S. Mohamed, "Understanding objects in detail with fine-grained attributes," in *Proc. CVPR*, 2014, pp. 3622–3629.

[10] J. Yu, Y. Rui, and D. Tao, "Click prediction for web image reranking using multimodal sparse coding," *IEEE Trans. Image Processing*, vol. 23, no. 5, pp. 2019–2032, 2014.

[11] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[12] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. NIPS*, 2014, pp. 3320–3328.

[13] X. Yin and X. Liu, "Multi-task convolutional neural network for pose-invariant face recognition," *IEEE Trans. Image Processing*, vol. 27, no. 2, pp. 964–975, 2018.

[14] N. Sarafianos, T. Giannakopoulos, C. Nikou, and I. A. Kakadiaris, "Curriculum learning of visual attribute clusters for multi-task classification," *Pattern Recognition*, vol. 80, pp. 94–108, 2018.

[15] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *Proc. CVPR*, 2018, pp. 3801–3809.

[16] W. Hong, Z. Wang, M. Yang, and J. Yuan, "Conditional generative adversarial network for structured domain adaptation," in *Proc. CVPR*, 2018, pp. 1335–1344.

[17] L. Hu, M. Kan, S. Shan, and X. Chen, "Duplex generative adversarial network for unsupervised domain adaptation," in *Proc. CVPR*, 2018, pp. 1498–1507.

[18] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proc. CVPR*, 2018, pp. 8503–8512.

[19] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by back-propagation," *arXiv preprint arXiv:1409.7495*, 2014.

[20] X. Zhang, F. X. Yu, S. F. Chang, and S. Wang, "Deep transfer network: Unsupervised domain adaptation," *arXiv preprint arXiv:1503.00591*, 2015.

[21] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," *arXiv preprint arXiv:1502.02791*, 2015.

[22] M. Long, J. Wang, Y. Cao, J. Sun, and P. S. Yu, "Deep learning of transferable representation for scalable domain adaptation," *IEEE Trans. Knowledge and Data Engineering*, vol. 28, no. 8, pp. 2027–2040, 2016.

[23] T. Gebru, J. Hoffman, and F. F. Li, "Fine-grained recognition in the wild: A multi-task domain adaptation approach," in *Proc. ICCV*, 2017, pp. 1358–1367.

[24] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 41, no. 4, pp. 801–814, 2019.

[25] ——, "Residual parameter transfer for deep domain adaptation," in *Proc. CVPR*, 2018, pp. 4339–4348.

[26] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Efficient parametrization of multi-domain deep neural networks," in *Proc. CVPR*, 2018, pp. 8119–8127.

[27] N. Peng and M. Dredze, "Multi-task multi-domain representation learning for sequence tagging," *CoRR*, vol. abs/1608.02689, 2016. [Online]. Available: http://arxiv.org/abs/1608.02689

[28] Y. Luo, Y. Wen, and D. Tao, "Heterogeneous multitask metric learning across multiple domains," *IEEE Trans. Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–14, 2018.

[29] G. Pons and D. Masip, "Multi-task, multi-label and multi-domain learning with residual convolutional networks for emotion recognition," *arXiv preprint arXiv:1802.06664*, 2018.

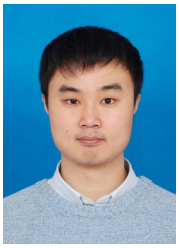
[30] C. H. Chen, "Improved tfidf in big news retrieval: An empirical study," *Pattern Recognition Letters*, vol. 93, 2016.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[32] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
[33] W. Feng and D. Liu, "Fine-grained image recognition from click-through logs using deep siamese network," in *International Conference on Multimedia Modeling*, 2017, pp. 127–138.
[34] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," in *First Workshop on Proc. CVPR*, June 2011.
[35] C. Goring, A. Freytag, E. Rodner, and J. Denzler, "Fine-grained categorization – short summary of our entry for the imagenet challenge 2012," *Computer Science*, 2013.
[36] M. Long and J. Wang, "Learning transferable features with deep adaptation networks," *CoRR*, vol. abs/1502.02791, 2015. [Online]. Available: http://arxiv.org/abs/1502.02791
[37] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *Proc. CVPR*, 2017, pp. 4476–4484.
[38] X. S. Wei, J. H. Luo, J. Wu, and Z. H. Zhou, "Selective convolutional descriptor aggregation for fine-grained image retrieval," *IEEE Trans. Image Processing*, vol. PP, no. 99, pp. 1–1, 2017.
[39] F. Faghri, D. J. Fleet, R. Kiros, and S. Fidler, "VSE++: improved visual-semantic embeddings," *CoRR*, vol. abs/1707.05612, 2017. [Online]. Available: http://arxiv.org/abs/1707.05612
[40] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds200-2011 dataset," *California Institute of Technology*, 2011.
[41] M. Tan, B. Wang, Z. Wu, J. Wang, and G. Pan, "Weakly supervised metric learning for traffic sign recognition in a lidar-equipped vehicle," *IEEE Trans. Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1415–1427, 2016.
[42] M. Tan, Z. Hu, B. Wang, J. Zhao, and Y. Wang, "Robust object recognition via weakly supervised metric and template learning," *Neurocomputing*, vol. 101, pp. 96–107, 2016.

Min Tan is currently an Associate Professor with the School of Computer Science and Technology, Hangzhou Dianzi University. She received the B.S. Degree in School of Mathematical Science and Computing Technology from Central South University, Changsha, China, in 2009, and received Ph.D. Degree in College of Compute Science and Technology from Zhejiang University, Hangzhou, China, in 2015. From Jun. 2013 to Feb. 2014, she was an Intern in Visual Computing group in MSRA. Her research interests include computer vision, pattern recognition and machine learning.

Jun Yu received his BEng and PhD from Zhejiang University, Zhejiang, China. He is currently a Professor with the School of Computer Science and Technology, Hangzhou Dianzi University. He was an Associate Professor with School of Information Science and Technology, Xiamen University. From 2009 to 2011, he worked in Singapore Nanyang Technological University. From 2012-2013, he was a visiting researcher in Microsoft Research Asia (MSRA). Over the past years, his research interests include multimedia analysis, machine learning and image processing. He has authored and co-authored more than 70 scientific articles. He has (co-)chaired for several special sessions, invited sessions, and workshops. He served as a program committee member or reviewer top conferences and prestigious journals. He is a Professional Member of the IEEE, ACM and CCF.



Hongyuan Zhang received the B.E. Degree is a S-tudent from Hangzhou Dianzi University, Hangzhou, China，in 2016. He is currently a master student in computer science from Hangzhou Dianzi University, Hangzhou, China. His research interests include computer vision, image recognition and machine learning (deep learning).

Yong Rui is currently Deputy Managing Director of Microsoft Research Asia (MSRA). A Fellow of IEEE, IAPR and SPIE, and a Distinguished Scientist of ACM, Rui is recognized as a leading expert in his research areas. He is the recipient of the IEEE Computer Society 2016 Technical Achievement Award, IEEE Trans. Multimedia 2015 Best Paper Award, and ACM Multimedia 2009 Best Paper Award. He holds 60 US and international patents. He has published 16 books and book chapters, and 200+ referred journal and conference papers. Rui's publications are among the most cited –17,000+ citations and his h-Index = 55. Dr. Rui is the Editor-in-Chief of IEEE Multimedia Magazine, an Associate Editor of ACM Trans. on Multimedia Computing, Communication and Applications (TOMM), and a founding Editor of International Journal of Multimedia Information Retrieval (IJMIR). He was an Associate Editor of IEEE Trans. on Multimedia (2004-2008), IEEE Trans. on Circuits and Systems for Video Technologies (2006-2010), ACM/Springer Multimedia Systems Journal (2004-2006), and International Journal of Multimedia Tools and Applications (2004-2006). He also serves on the Advisory Board of IEEE Trans. on Automation Science and Engineering. He is an Executive Member of ACM SIGMM, and the founding Chair of its China Chapter. Dr. Rui received his BS from Southeast University, his MS from Tsinghua University, and his PhD from University of Illinois at Urbana-Champaign (UIUC).

Dacheng Tao (F'15) is Professor of Computer Science and ARC Laureate Fellow in the School of Computer Science and the Faculty of Engineering and Information Technologies, and the Inaugural Director of the UBTECH Sydney Artificial Intelligence Centre, at the University of Sydney. He mainly applies statistics and mathematics to Artificial Intelligence and Data Science. His research results have expounded in one monograph and 200+ publications at prestigious journals and prominent conferences, such as IEEE T-PAMI, T-IP, T-NNLS,T-CYB, IJCV, JMLR, NIPS, ICML, CVPR, ICCV, ECCV, ICDM; and ACM SIGKDD, with several best paper awards, such as the best theory/algorithm paper runner up award in IEEE ICDM' 07, the best student paper award in IEEE ICDM'13, the 2014 ICDM 10-year highest-impact paper award, the 2017 IEEE Signal Processing Society Best Paper Award, and the distinguished paper award in the 2018 IJCAI. He received the 2015 Austrlian Scopus-Eureka Prize and the 2018 IEEE ICDM Research Contributions Award. He is a Fellow of the Australian Academy of Science, AAAS, IEEE, IAPR, OSA and SPIE.