Personalized Deep Learning for Tag Recommendation

Hanh T.H. Nguyen, Martin Wistuba, Josif Grabocka,
Lucas Rego Drumond, and Lars Schmidt-Thieme

Information Systems and Machine Learning Lab,
University of Hildesheim, Universitätsplatz 1, 31141 Hildesheim, Germany
{nthhanh,wistuba,josif,ldrumond,schmidt-thieme}@ismll.de

Abstract. Social media services deploy tag recommendation systems to facilitate the process of tagging objects which depends on the information of both the user's preferences and the tagged object. However, most image tag recommender systems do not consider the additional information provided by the uploaded image but rely only on textual information, or make use of simple low-level image features. In this paper, we propose a personalized deep learning approach for the image tag recommendation that considers the user's preferences, as well as visual information. We employ Convolutional Neural Networks (CNNs), which already provide excellent performance for image classification and recognition, to obtain visual features from images in a supervised way. We provide empirical evidence that features selected in this fashion improve the capability of tag recommender systems, compared to the current state of the art that is using hand-crafted visual features, or is solely based on the tagging history information. The proposed method yields up to at least two percent accuracy improvement in two real world datasets, namely NUS-WIDE and Flickr-PTR.

Keywords: Image tagging; convolutional neural networks; personalized tag recommendation

1 Introduction

Tags assigned freely by users can be used to support users organizing or searching resources of social media systems [1]. However, a considerable number of shared resources has few or no tags because of the time-consuming aspect of the tagging task. For example, between February 2004 to June 2007, around 64 percent of Flickr uploaded photos had 1 to 3 tags and around 20 percent had no tags [20]. To encourage users annotating their resources, tag recommendation systems are used to facilitate the tagging task by suggesting relevant tags for them. These systems can be personalized systems that recommend different tags depending on the users' preferences, or non-personalized ones that omit the users' interests. Because the tags represent the user's view to his resource, the recommended tag list for a user is practically a personalized list containing his "favorite" keywords.

The personalized models can be based on the relation between users, items and tags, or otherwise on the correlation information of tags [4,19,16].

The personalized approaches are not efficient for new images with no historical information. As Sigurbjörnsson and Van Zwol mentioned [20], people usually choose the words related to the contents or contexts such as location or time to annotate images. The visual information can be considered to be used in the personalized recommendation models in order to enhance the prediction quality. The recommended tags of a personalized content-aware tag recommendation express personal and content-aware characteristics as in Figure 1.

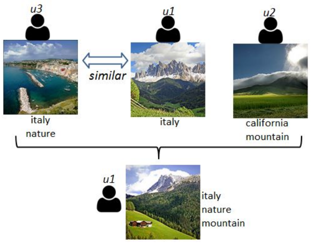


Fig. 1: The tags recommended for *u1* contain his favorite word *italy*, a word *mountain* related to the content of the image and a word *nature* from *u3* being similar to *u1*.

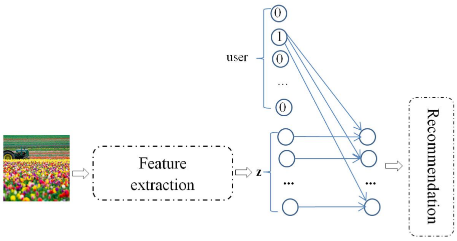


Fig. 2: The architecture of CNN-PerMLP

In this work, we show how a deep learning approach can be adapted to solve a personalized image tag recommendation. For a personalized problem, the features used in a deep learning model have to include the information of a user and an associated image. We propose a new way to add the user's information into the CNN models. A new layer that captures the interaction between users and visual features plays a bridged role between a CNN image feature extractor and a multilayer perceptron as in Figure 2. In addition, we adapt the Bayesian Personalized Ranking optimization [18] in a different way to apply for the model.

Empirically, our experiments obtained in two real datasets, namely NUS-WIDE and Flickr-PTR, show that the proposed model outperforms the state-of-the-art personalized tag recommendation models, which are purely based on tagging history, up to at least four percent. The experiments also indicate the stronger support of the supervised features to increase the prediction quality up to at least two percent compared to low-level features.

2 Related Work

A large number of tag recommendation approaches focus on various features of objects, such as the contents of media objects, the relation between users and images, or the objects' contexts. The neighbor voting model [10] assembles the votes of similar images to retrieve the relevant tags. The collective knowledge approach [20] recommends the correlated tags with the user-provided tags based on the co-occurrence metric. The metric is also used for the personalized tag recommendation [4]. The model predicts the relevant tags for users based on the global and personal correlated scores of tags.

The correlated scores of tags achieved from different contexts are aggregated to look for the relevant tags [15]. The contexts include the information of the whole system, the social contacts of a user and the attending groups. In another approach, both the content and context information are used to find the neighbors of a given image from the historical tagging collection of the owner's images. The most frequent tags selected from its neighbors are recommended for the image [14]. In the model proposed by Chen and Shin [2], textual and social features that are extracted from tags, titles, contents, comments or users' social activities are combined to represent tags. Then, logistic regression or Naïve Bayes is employed as the recommender.

Factorization models are widely applied and show a good performance for tag recommendation. One of the state-of-the-art models is the Pairwise Interaction Tensor Factorization (PITF). It models all interactions between different pairs of users, items and tags, and accumulates all pairwise scores to the tags' scores [19]. Factorization Machine (FM) [16] is an approach that takes advantage of feature engineering and factorization. It can be applied to solve different tasks, such as regression, classification or ranking.

Tag recommendation based on the visual information of items only can be viewed as a multilabel classification or an image annotation task. A Convolutional Neural Network (CNN), a strong model for image classification and recognition [9,8,22], is applied to solve image annotation [5,23]. The approach can learn the predictor by optimizing different losses, such as pairwise, or Weighted Approximate Ranking (WARP), either to deal with the ranking problem [5], or to predict labels from arbitrary trained objects [23].

Because the factorization models merely depend on the relation between users, images and tags, they perform worse when predict new images. Our proposed model relied on users and visual features of images overcomes the limitation of recommending tags for new images.

The image annotation models do not contain the user's information so they work poorly in a personalized scenario. The proposed model has a personalized layer that captures the the user-aware features so that the deep learning model can be adapted into a personalized tag recommendation.

3 The proposed model

3.1 Problem Formulation

The personalized tag recommender suggests a ranked list of relevant tags to a user annotating a specific image. The set of tag assignments $\mathcal{A}$ can be represented as a combination of users U, images I and tags T. It is denoted as $\mathcal{A} = (a_{u,i,t}) \in \mathbb{R}^{|U|\times|I|\times|T|}$ [11] where $a_{u,i,t} = 1$ if the user u assigns the tag t to the image i, or otherwise $a_{u,i,t} = 0$. The observed tagging set is defined as $S := \{(u,i,t)|a_{u,i,t} \in \mathcal{A} \wedge a_{u,i,t} = 1\}$. The set of relevant tags of a user-image tuple (u,i) is denoted as $T_{u,i} := \{t \in T \mid (u,i,t) \in S\}$. Let $P_S := \{(u,i) \mid \exists t \in T : (u,i,t) \in S\}$ be all observed posts [17].

In addition, the collection of all RGB squared images is defined as R. The visual features of the i-th image R_i is a vector $\mathrm{z}_i \in \mathbb{R}^m$. In this paper, we crop each image into Q patches to enhance the value of extracted features so we can define the collection of images $R = \{R_{i,q}|R_{i,q} \in \mathbb{R}^{d\times d\times 3} \wedge i \in I \wedge q \in Q\}$.

The scoring function of the recommendation model computes the scores of tags for a given post $p_{u,i}$ which are used to rank tags. The score of a tag to a given post is represented as $\hat{y}(u,i,t) : U \times I \times T \to \mathbb{R}$. If the score $\hat{y}_{u,i,t_a}$ is larger than the score $\hat{y}_{u,i,t_b}$, the tag t_a is more relevant to the post $p_{u,i}$ than the tag t_b. The tag recommendation model is expected providing a top-K tag list $\hat{T}_{u,i}$ that is ranked in descending order of tags' scores for a post $p_{u,i}$.

$$\hat{T}_{u,i} := \underset{t\in T, |\hat{T}_{u,i}|=K}{\operatorname{argmax}} \; \hat{y}(u,i,t) \tag{1}$$

3.2 Personalized Content-aware Tag Recommendation

The architecture of the proposed model called **CNN-PerMLP** based on the relation between the user and the visual features of the given image is illustrated in Figure 2. The supervised visual features are achieved by passing a patch q of the image i through the CNN feature extractor.

To personalize the visual features, a proposed specific layer called the personalized fully-connected layer is obtained following the extractor. The layer captures the interaction between the user and each visual feature to generate the latent features for the post $p_{u,i}$.

A neural network is deployed as a predictor to compute the relevant probabilities of tags. The network receives the user-image features as the input and its outputs are used to derive a ranking of recommended tags.

In this paper, we divide images into several patches and the final scores of tags are the average scores computed from different patches. If the score of a tag to a given post $p_{u,i}$ and a patch q is represented as $\hat{y}' : U \times R \times T \to \mathbb{R}$, the final tag's score is

$$\hat{y}(u,i,t) = \underset{R_{i,q}, q\in Q}{\operatorname{avg}} \; \hat{y}'(u, R_{i,q}, t) \tag{2}$$

Convolution Neural Networks. The CNN is obtained to represent high-level abstraction of image features. One or more convolutional layers are employed to generate several feature maps by moving kernel windows smoothly across images. The k-th feature map of a given layer is denoted as τ^k, the weights and the biases of the filters for τ^k are $W^k \in \mathbb{R}^{p_1} \times \mathbb{R}^{p_2} \times \mathbb{R}^{p_2}$ and b_k where p_1 is the number of the previous layer's feature maps and p_2 is the dimension of kernel windows. The element at the position (i, j) of τ^k is acquired as

$$\tau_{ij}^k = \varphi\left(b_k + \sum_{a=1}^{p_1} (W_a^k * \xi^a)_{ij} \right) \tag{3}$$

with $*$ being the convolution operator, ξ^a being the a-th feature map of the previous layer and φ being the activation function. The subsampling layer, which pools a rectangular block of the previous layer to generate an element for the current feature map, follows the convolutional layer. If the max pooling operator is used, the element at the position (i, j) of the k-th feature map τ^k is denoted as follows

$$\tau_{ij}^k = \max_{a,b} (\xi^k)_{a,b} \tag{4}$$

where a and b are the positions of the element associating to the pooled block. The output of the CNN feature extractor is a dense feature vector representing the image. We define the extraction process of the patch q of the i-th image by

$$\mathrm{z}_i^q = f_{cnn}(R_i^q) : \mathbb{R}^{d \times d \times 3} \rightarrow \mathbb{R}^m \tag{5}$$

Personalized Fully-connected Layer. The extracted features from the CNN only contain the information of the image i. To personalize visual features of an image, the user's information has to be added or combined with these features. For this reason, a layer stood between the feature extractor and the predictor is employed to generate the user-aware features that are used as the input of the predictor.

If the model uses only the user's id as the personalized information, the user's features u are described as a sparse vector represented $\kappa_u := \{0, 1\}^{|U|}$. Both the visual feature vector z_i^q and the sparse vector κ_u are the input of this layer. The layer is responsible to capture the interaction between the user and to each visual feature. If the output of this layer is denoted by $\psi \in \mathbb{R}^m$, it is obtained as follows

$$\psi_j(u, \mathrm{z}_i^q) = \varphi(b_j + w_j^{per} \cdot (z_i^q)_j + V_j \kappa_u) \tag{6}$$

where $\mathrm{w}^{per} \in \mathbb{R}^m$ is the weights of the visual features and $V \in \mathbb{R}^{m \times |U|}$ is the weights of the user features. As in the convolutional layer, the elementwise activation function φ is used after combining the weighting visual feature and the user's features.

Multilayer Perceptron as the Predictor. To compute the scores of the tags, a multilayer perceptron is adopted as a predictor and its input is the output of the personalized fully-connected layer ψ. The output of the network is the relevant scores of tags associated with the post (u, i) and the patch q of the image i. Because the network in the proposed model has one hidden layer, we denote the neural network score function as follows

$$\hat{y'}(u, R_i^q, t_j) = \varphi\left(w_j^{out} \cdot \varphi(W^{hidden} \psi + b^{hidden}) + b_j^{out} \right) \tag{7}$$

where W^{hidden} and b^{hidden} are the weights and the biases of the hidden layer; $w_j^{out} \in W^{out}$ and b^{out} are the weights and the biases of the output layer.

3.3 Optimization

We adapt the Bayesian Personalized Ranking (BPR) optimization criterion [18] in a different way so that the algorithm can be applied to learn the deep learning personalized image tag recommendation.

The optimization based on BPR finds the model's parameters that maximize the difference between the relevant and irrelevant tags. In addition, the stochastic gradient descent applied for BPR is in respect of the quadruple (u, i, t^+, t^-); i.e, for each $(u, i, t^+) \in S_{train}$ and an unobserved tag of $p_{u,i}$ drawn at random, the loss is computed and is used to update the model's parameters. The aforementioned BPR is not efficient to be used to learn the proposed model. The BPR criterion with respect to the posts is proposed to use and it is defined as

Algorithm 1 Learning BPR

```
Input: $P_S$, $S$, $R$, $N$, $\alpha$
Output: $\Theta$

Initialize $\Theta \leftarrow \mathcal{N}(0, 0.1)$
repeat
    Pick $(u, i) \in P_{S_{train}}$ and $R_i^q \in R$ randomly
    Get $T_{u,i}^+ := \{t \in T \mid (u, i, t) \in S\}$
    Pick $T_{u,i}^- := \{t \in T \mid (u, i, t) \notin S\}$ randomly where $\mid T_{u,i}^- \mid= N$
    Compute $\mathrm{z}_i^q = f_{cnn}(R_i^q)$ and $\psi(u, \mathrm{z}_i^q)$
    for $t \in 1, \ldots, |T|$ do
        if $t \in T_{u,i}^+ \vee t \in T_{u,i}^-$ then
            Compute $\hat{y}'(u, R_i^q, t)$
        end if
    end for
    Update $\Theta \leftarrow \Theta + \alpha\left(\frac{\partial \,\mathrm{BPR}(u,i)}{\Theta}\right)$
until convergence
return $\Theta$
```

$$\mathrm{BPR}(u, R_i^q) := \frac{1}{\mid T_{u,i}^+ \mid\mid T_{u,i}^- \mid} \sum_{t^+ \in T_{u,i}^+, t^- \in T_{u,i}^-} \ln \sigma(\hat{y}'(u, R_i^q, t^+, t^-)) \qquad (8)$$

where $T_{u,i}^+ := \{t \in T \mid (u, i, t) \in S_{train}\}$ is the set of tags selected by the user u for the image i. The rest of tags is the unobserved tag set denoted as $T_{u,i}^- := \{t \in T \mid (u, i) \in P_{S_{train}} \wedge (u, i, t) \notin S_{train}\}$. The function $\sigma(x)$ is described as $\sigma(x) = \frac{1}{1+e^{-x}}$. The difference between the score of relevant tags and irrelevant tags is defined as $\hat{y}'(u, R_i^q, t^+, t^-) = \hat{y}'(u, R_i^q, t^+) - \hat{y}'(u, R_i^q, t^-)$.

The learning model's parameters process is described in Algorithm 1. For each random post, a random patch of the associated image is chosen to extract the visual features. An irrelevant set having N tags is selected at random from the unobserved tags of the post. The system computes the scores of all relevant tags and the drawn irrelevant tags. From Equation (8), the gradient of BPR with respect to the model parameters is obtained as follows:

$$\frac{\partial\,\mathrm{BPR}}{\partial \Theta} = \Omega \sum_{t^+ \in T^+_{u,i}} \sum_{t^- \in T^-_{u,i}} \Psi_{t^+,t^-} \frac{\partial \hat{y}'(u, R_i^q, t^+, t^-)}{\partial \Theta} \tag{9}$$

where

$$\Omega = \frac{1}{\mid T^+_{u,i} \mid\mid T^-_{u,i} \mid} \qquad \Psi_{t^+,t^-} = \frac{e^{-\hat{y}'(u,R_i^q,t^+,t^-)}}{1 + e^{-\hat{y}'(u,R_i^q,t^+,t^-)}}$$

To learn the model, the gradients $\frac{\partial \hat{y}'(u,R_i^q,t^+)}{\partial \Theta}$ and $\frac{\partial \hat{y}'(u,R_i^q,t^-)}{\partial \Theta}$ have to be computed. Depending on the weights in the different layers, one or both the gradients are computed. For example, if the parameter θ_j depends on the relevant tags t_j^+, Equation (9) becomes

$$\frac{\partial\,\mathrm{BPR}}{\partial \theta_j} = \frac{\partial\,\mathrm{BPR}}{\partial \hat{y}'(u, i, t_j^+)} \times \frac{\partial \hat{y}'(u, R_i^q, t_j^+)}{\partial \theta_j} = \frac{\partial \hat{y}'(u, R_i^q, t_j^+)}{\partial \theta_j} \cdot \Omega \sum_{t^- \in T^-_{u,i}} \Psi_{t_j^+,t^-} \tag{10}$$

To find the gradients of the CNN parameters, the derivatives with respect to the visual features are propagated backward to CNN. From the Equation (6) and (9), the derivatives are computed as

$$\frac{\partial\,\mathrm{BPR}}{\partial (z_i^q)_j} = \Omega \sum_{t^+ \in T^+_{u,i}} \sum_{t^- \in T^-_{u,i}} \Psi_{t^+,t^-} \frac{\partial \hat{y}'(u, R_i^q, t^+, t^-)}{\partial \psi_j} \cdot w_j^{per} \tag{11}$$

4 Evaluation

In the evaluation, we performed experiments addressing the impact of supervised visual features and the personal factor on the tag recommendation process.

4.1 Dataset

We obtained experiments on subsets of the publicly available multilabel dataset NUS-WIDE [3] that contains 269,648 images and Flickr-PTR [12] that was created by crawling around 2 million Flickr images. We preprocessed the NUS-WIDE dataset as follows: keeping available images annotating by the 100 most popular tags, sampling 1.000 users, refining to get 10-core dataset referring to users and tags where each user or tag occurs at least in 10 posts [6] and removing tags assigning more than 50 percent of images by one user to avoid the case that users tag all their images by the same words. Similarly, the Flickr-PTR dataset

Table 1: Dataset characteristics

Dataset	**Users**	**Images**	**Tags**	**Triples**	**Posts**	**Training posts**	**Test posts**
	$\|U\|$	$\|I\|$	$\|T\|$	$\|S\|$	$\|P_S\|$	$\|P_{S_{train}}\|$	$\|P_{S_{test}}\|$
NUS-WIDE	1000	27.662	100	81.263	27.858	25.858	2.000
Flickr-PTR	323	29.095	133	94.387	29.096	23.402	5.694

Table 2: Layer characteristics of the convolutional architectures

Layer	**NUS-WIDE**	**Flickr-PTR**
The 1st ConvL	6 × 6 × 3 (stride: 3)	5 × 5 × 3 (stride: 2)
The 1st MaxPoolL	2 × 2	2 × 2
The 2nd ConvL	6 × 6 × 10 (stride: 2)	5 × 5 × 10 (stride: 1)
The 2nd MaxPoolL	2 × 2	3 × 3
The 3rd ConvL	2 × 2 × 30 (stride: 1)	3 × 3 × 30 (stride: 1)

is preprocessed by mapping all tags to WordNet [13], refining dataset to get the 40-core regarding to users and 400-core to tags dataset, sampling 500 users and removing tags assigning more than 50 percent of images by a user.

We adapted leave-one-post-out [11] for users to split the dataset. For each user, 20 percent of Flickr-PTR posts and 2 NUS-WIDE posts are randomly picked and put into the test sets. These subdivided dataset can be described with respect to users, images, tags, triples and posts as in Table 1. Images crawled by Flickr API[1] were cropped from the aspect ratio retained 75 × 75 for NUS-WIDE or 50 × 50 for Flickr-PTR into 3 pieces at 3 positions top-left, center and bottom-right to be used as the input patches for training and predicting.

4.2 Experimental Setup

The architectures used for both datasets contain 3 convolutional layers (*ConvL*) alternated with 2 max-pooling layers (*MaxPoolL*). *ConvL*s in these architectures have the same number of kernels that are 10 for the first, 30 for the second and 128 for the third. Because of the difference of the image size, the dimensions of convolutional kernels and pooling blocks in these architectures are different shown in Table 2. The hidden layers of the predictor have the dimension 128 for both architectures and the rectifier function $max(0, x)$ is used as the activation function. The evaluation metric used in this paper is F1-measure in top K tag lists [19].

$$\text{F1@K} = \frac{2 \cdot \text{Prec@K} \cdot \text{Recall@K}}{\text{Prec@K} + \text{Recall@K}} \tag{12}$$

where

$$\text{Prec@K} = \underset{(u,i) \in S_{test}}{\text{avg}} \frac{|\hat{T}_{u,i} \cap T_{u,i}|}{K} \qquad \text{Recall@K} = \underset{(u,i) \in S_{test}}{\text{avg}} \frac{|\hat{T}_{u,i} \cap T_{u,i}|}{|T_{u,i}|}$$

[1] https://www.flickr.com/services/api/

$$\hat{T}_{u,i} = \text{Top}(u, i, K) = \underset{t \in T, |\hat{T}_{u,i}| = K}{\text{argmax}} \; \hat{y}(u, i, t)$$

The grid search mechanism was used to find the best learning rate α among the range $\{0.001, 0.0001, 0.00001\}$ for all *ConvL*s and $\{0.01, 0.0001, 0.0001\}$ for all fully-connected layers, the best L2-regularization λ from the range $\lambda \in \{0.0, 0.0001, 0.00001\}$ while the momentum value μ was fixed to 0.9. The 64-dimension color histogram (CH) and 225-dimension block-wise color moments (CM55) provided by NUS-WIDE's authors[3] and the 64-dimension color histogram (CH) of Flickr-PTR images are used for comparison.

The proposed model **CNN-PerMLP** is compared to following personalized tag recommendation methods that use only the users' preference information and do not consider the visual features: Pairwise Interaction Tensor Factorization (**PITF**) [19], Factorization Machine (**FM**) [16], most popular tags by users (**MP-u**) [6].

It is also compared to the non-personalized models including most popluar tags(**MP**) [6], the multilabel neural networks (BP-MLLs) [24] that have low-level visual features as the input (**CH-BPMLL**, **CM55-BPMLL**), **CNNs** obtained for image annotation which optimizes the pairwise ranking loss to learn the parameters as the loss used by Zhang and Zhou [24]. The reimplemented CNN is similar to the proposed model of Gong et al. [5] with respect to optimizing the loss under the ROC curve (AUC).

The adjusted models (**CH-PerMLP** and **CM55-PerMLP**) of the proposed model using low-level features were obtained for the comparison. We used the Tagrec framework [7] to learn MP and MP-u, and the Mulan library [21] to learn CH-BPMLL and CM55-BPMLL.

4.3 Results

As shown in Figure 3, the non-personalized models cannot capture the user's interests and they just recommend tags related to the content. The prediction quality of these models is lower than that of the personalized model. However, the model with CNN supervised features captures more information than the models using low-level features, leading to a boosted performance around 2 percent.

The claim that visual features improve the prediction quality is more serious in Figure 4. In the test having most new images, the weights associated with these images are not learned in the training process. So the prediction of the personalized content-ignored models like FM and PITF solely depends on users and their results are clearly comparable to the prediction of MP-u. The personalized content-aware models work better than them in this case and recommended tags rely on both users and visual image information. The visual features help increasing the prediction quality around 4 percent. The supervised features also prove their strength in the recommendation quality compared to the low-level features. The performance is improved around 2 to 3 percent as a result of using the learned visual features.

Examples in Table 3 show that the proposed model can predict both personal tags and content tags compared to MP-u that purely predicts personal tags and

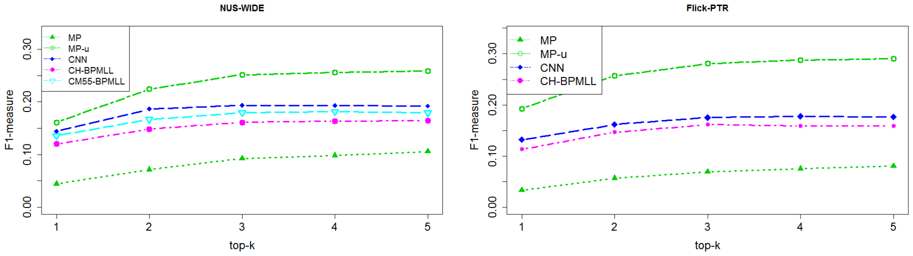


Fig. 3: The results of the non-personalized models are not as good as the personalized models but the model using supervised features outperforms the model using low-level features.

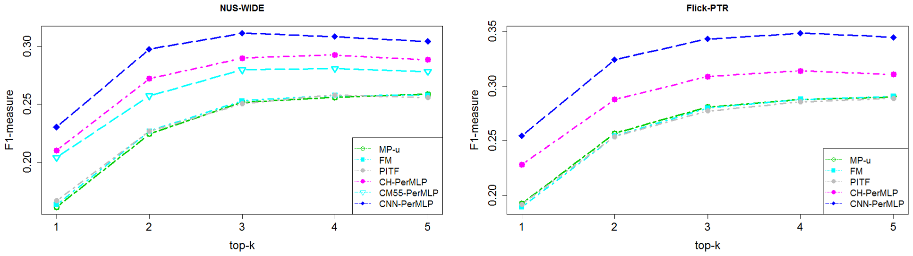


Fig. 4: CNN-PerMLP outperforms the personalized models based purely on tagging history and the personalized models using low-level visual features.

CNN recommending content tags. As a result, the CNN-PerMLP suggests more relevant tags to the image. For example, in the first photo, the recommender can catch personal tags as ”flowers”, ”flower”, ”orange” and content tags as ”white” or ”red”. Through Table 4 and 5, CNN-PerMLP works well in the case that people use their frequent tags or tags related to the image’s content to annotate a new image. However, the prediction quality of the model is poor if the users assign tags that are new and do not relate to the content.

5 Conclusion

In this paper, we propose a deep learning model using supervised visual features for personalized image tag recommendation. The experiments show that the proposed method has advantages over the state-of-the-art personalized tag recommendation purely based on tagging history, like PITF or FM in the narrow folksonomy scenarios. Moreover, the learnable features strongly influence the recommendation quality compared to the low-level features. The information of users used in the proposed models is plainly the users’ id and it does not really represent the characteristic of the users, such as favorite words or favorite images. In the future, we plan to investigate how to use the textual features of users, in

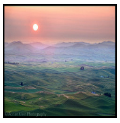
Table 3: Examples of top 5 recommended tags of CNN-PerMLP, CNN and MP-u

Image	**Ground truth**	**CNN-PerMLP**	**CNN**	**MP-u**	**Image**	**Ground truth**	**CNN-PerMLP**	**CNN**	**MP-u**
	flower red orange white	red white flowers orange flower	red woman girl white people	flowers white orange pink flower		green grass landscape park	landscape green sky park grass	green bravo blue nature flowers	beautiful park landscape color animal

Table 4: Example having the highest accuracy of top 5 recommended tags

Image	**Ground truth**	**Prediction**
	lake sunset water blue sun	sunset water lake blue sun

Table 5: Example having the lowest accuracy of top 5 recommended tags

Image	**Ground truth**	**Prediction**
	green	sea beach sunset clouds ocean

combination with the visual features of images to enhance the recommendation quality.

References

1. Ames, M., Naaman, M.: Why we tag: motivations for annotation in mobile and online media. In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 971–980 (2007)
2. Chen, X., Shin, H.: Tag recommendation by machine learning with textual and social features. Journal of Intelligent Information Systems 40, 261–282 (2013)
3. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: a real-world web image database from national university of singapore. In: Proceedings of the ACM international conference on image and video retrieval. p. 48 (2009)
4. Garg, N., Weber, I.: Personalized, interactive tag recommendation for flickr. In: Proceedings of the 2008 ACM conference on Recommender systems. pp. 67–74 (2008)
5. Gong, Y., Jia, Y., Leung, T., Toshev, A., Ioffe, S.: Deep convolutional ranking for multilabel image annotation. arXiv preprint arXiv:1312.4894 (2013)
6. Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in folksonomies. In: Knowledge Discovery in Databases: PKDD 2007. pp. 506–514 (2007)
7. Kowald, D., Lacic, E., Trattner, C.: Tagrec: Towards a standardized tag recommender benchmarking framework. In: Proceedings of the 25th ACM conference on Hypertext and social media. pp. 305–307 (2014)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

9. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324 (1998)
10. Li, X., Snoek, C.G., Worring, M.: Learning tag relevance by neighbor voting for social image retrieval. In: Proceedings of the 1st ACM international conference on Multimedia information retrieval. pp. 180–187 (2008)
11. Marinho, L.B., Hotho, A., Jäschke, R., Nanopoulos, A., Rendle, S., Schmidt-Thieme, L., Stumme, G., Symeonidis, P.: Recommender systems for social tagging systems. Springer Science & Business Media (2012)
12. McParlane, P.J., Moshfeghi, Y., Jose, J.M.: Collections for automatic image annotation and photo tag recommendation. In: International Conference on Multimedia Modeling. pp. 133–145 (2014)
13. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM 38, 39–41 (1995)
14. Qian, X., Liu, X., Zheng, C., Du, Y., Hou, X.: Tagging photos using users' vocabularies. Neurocomputing 111, 144–153 (2013)
15. Rae, A., Sigurbjörnsson, B., van Zwol, R.: Improving tag recommendation using social networks. In: Adaptivity, Personalization and Fusion of Heterogeneous Information. pp. 92–99 (2010)
16. Rendle, S.: Factorization machines. In: Data Mining (ICDM), 2010 IEEE 10th International Conference on. pp. 995–1000 (2010)
17. Rendle, S., Balby Marinho, L., Nanopoulos, A., Schmidt-Thieme, L.: Learning optimal ranking with tensor factorization for tag recommendation. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 727–736 (2009)
18. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. pp. 452–461 (2009)
19. Rendle, S., Schmidt-Thieme, L.: Pairwise interaction tensor factorization for personalized tag recommendation. In: Proceedings of the third ACM international conference on Web search and data mining. pp. 81–90 (2010)
20. Sigurbjörnsson, B., Van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: Proceedings of the 17th international conference on World Wide Web. pp. 327–336 (2008)
21. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Data mining and knowledge discovery handbook, pp. 667–685. Springer (2009)
22. Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R.: Regularization of neural networks using dropconnect. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13). pp. 1058–1066 (2013)
23. Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., Yan, S.: Cnn: Single-label to multi-label. arXiv preprint arXiv:1406.5726 (2014)
24. Zhang, M.L., Zhou, Z.H.: Multilabel neural networks with applications to functional genomics and text categorization. IEEE transactions on Knowledge and Data Engineering 18(10), 1338–1351 (2006)