Deep Representations to Model User 'Likes'

Sharath Chandra Guntuku*[1], Joey Tianyi Zhou*[1],
Sujoy Roy[2], Lin Weisi[1], Ivor W. Tsang[3]

[1] School of Computer Engineering, Nanyang Technological University, Singapore
sharathc001@e.ntu.edu.sg, tzhou1@ntu.edu.sg, wslin@ntu.edu.sg
[2] Institute for Infocomm Research, Singapore
sujoy@i2r.a-star.edu.sg
[3] QCIS, University of Technology, Sydney
ivor.tsang@uts.edu.au

Abstract. Automatically understanding and modeling a user's liking for an image is a challenging problem. This is because the relationship between the images features (even semantic ones extracted by existing tools, viz. faces, objects etc) and users' 'likes' is non-linear, influenced by several subtle factors. This work presents a deep bi-modal knowledge representation of images based on their visual content and associated tags (text). A mapping step between the different levels of visual and textual representations allows for the transfer of semantic knowledge between the two modalities. It also includes feature selection before learning deep representation to identify the important features for a user to like an image. Then the proposed representation is shown to be effective in learning a model of users image 'likes' based on a collection of images 'liked' by him. On a collection of images 'liked' by users (from Flickr) the proposed deep representation is shown to better state-of-art low-level features used for modeling user 'likes' by around 15-20%.

1 Introduction

This work investigates the answer to the question - "Why did a user like an image?". If there was only one image and we had no clue about the user, it would be a very hard question to answer. But, given a collection of images that the user has 'liked', it should be possible to model the user's taste and infer more on - "What 'kind' of images the user 'likes'?". Based on this knowledge, images that the user has not seen and may probably 'like' can be recommended. Conversely, given an image and a set of users, we should be able to predict the user(s) who will 'like' the image.

The notion of 'like' is very subjective and subtle, and hence hard to describe by methods or processes. This makes the process of modeling user's taste hard. If we know a set of pictures that a user has 'liked', then computationally we can consider several factors that can contribute towards the user liking the images, namely, the affective factors,the objects in the image and their perspectives/poses, the setting of the image, colors or no colors, the context of the ima-

* denotes equal contribution.

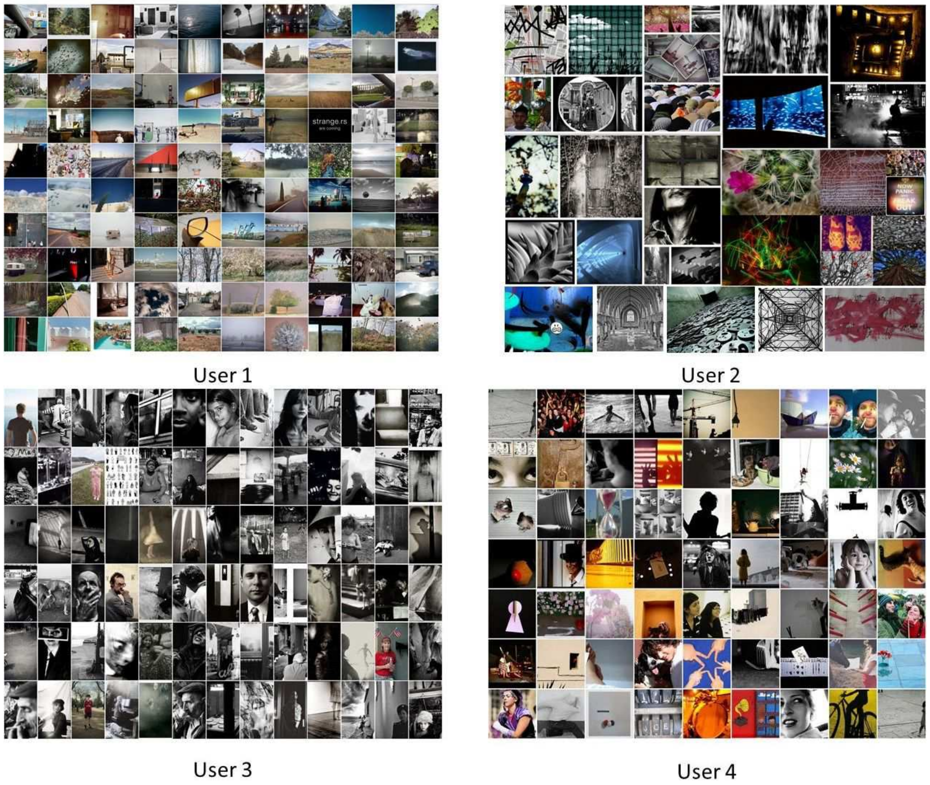
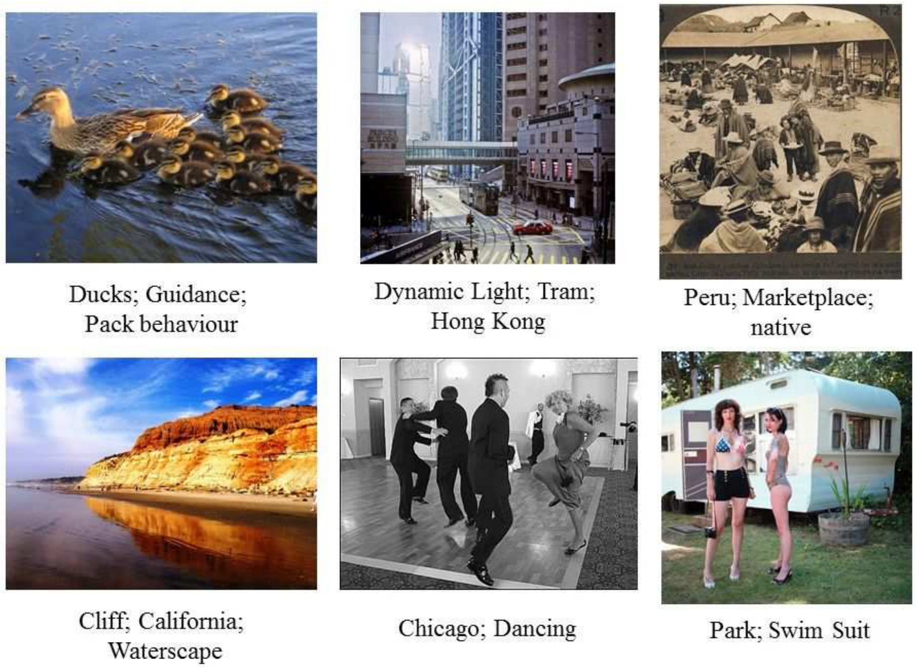


Fig. 1: Users have innate preferences which draw them to different 'kinds' of images.

ge, etc., all of which induce the user to have some emotional connection with the image. The question this work investigates is "How do these factors influence the user's liking for an image?" For example, Fig. 1 shows the set of images liked by four different users. The difference between Users 1,2 and 3,4 is that while one group (Users 3 & 4) tends to like images with a lot of people and faces , the other group (Users 1 & 2) likes images without people. Even within Users 3 & 4 we can see that User 3 tends to like black and white images whereas User 4 tends to like color images. Similarly User 1 tends to like images which are spread out and natural whereas User 2 tends to like images with high clutter and which have some tinge of graphics in them.

While visual content does play a significant role in capturing the users' preferences, we note that tags associated with images often enhance the information conveyed by visual features or sometimes compliment them by conveying information about the image which visual features fail to convey [1]. Users tend to like an image because it is a place that they have visited before or because they relate to the person in the image or the scene in the image triggers a memory from their past and so on. This 'contextual' value associated with images is often



Ducks; Guidance; Pack behaviour

Dynamic Light; Tram; Hong Kong

Peru; Marketplace; native

Cliff; California; Waterscape

Chicago; Dancing

Park; Swim Suit

Fig. 2: Tags can show perspectives which Visual features can fail to convey.

captured by the tags associated with an image [2]. While tags primarily convey the concepts in an image which visual detectors can be trained to detect (for example the objects in the image etc.), they many a time convey semantic concepts that go beyond what traditional visual detectors can be trained to detect. We can also attribute the varieties in 'likes' to the users psychophysical nature or personality. The role of personality in influencing preferences has been studied in several domains (viz. music [3, 4], images [5, 6]). However, in this work we are not trying to learn the users' personality. Rather we aim to learn deep representations for the image based on all the factors we can think of and how they combine to induce a user to 'like' the image.

Difference from Deep Learning for Concepts Detection: The images we consider are accompanied with associated textual information (tags). Visual and textual information have been used in the computer vision literature to model contents for tasks like categorization, image search and so on. We wish to highlight that predicting 'likes' is different from visual concept detection as the supervisory information for modeling 'likes' is subtle and cannot be visually verified. Modeling such subtle response would intuitively involve a combination of several higher level semantic factors viz. context, affective and aesthetic. The goal is to look for mid-level representation that can help model user 'likes'.

Contributions (1) Unlike [7], this work investigates the user 'like' modeling problem based on a larger collection of semantic, syntactic, aesthetic and contextual features. (2) A deep bimodal knowledge representation from the initial collection of features is proposed that allows for knowledge transfer between the modalities (visual and text). These features are used to learn a model for predicting user 'likes' for images. The efficacy of the proposed approach (usage of high-level features and the learning process of deep bimodal representations) is evaluated in an image recommendation scenario. Throughout the text, we use the phrase 'high-level features' to denote the features mentioned in Section 2 and the phrase 'low-level features' to denote the features mentioned in [7]. The proposed deep representation is denoted as 'deep bimodal feature representation'.

2 Semantic Feature Representation

In this section We give a short description and motivation for using the mentioned semantic features. Table 1

1. **Head and Upperbody recognition**: Presence of people plays a significant role in the way images are perceived [8]. And most images have persons who are only partially visible. The field of view typically shows only their upper body. For this situation, we used an upper-body detector for near frontal viewpoints [9]. And for images where the field of view consists only the head, we employed the head detector [10].
2. **Face and Pose recognition**: We find that some people tend to like images where people are looking away from the camera, and others tend to like images where people are looking at the camera. We extracted the relative area of the bounding boxes returned by the algorithm [11], along with the pose angle to categorise a face as a profile/frontal face.
3. **Gender identification**: Images with people are found to be liked and commented upon heavily by members of the opposite gender [12]. Output from the face detection was used as input for the IntraFace [13] to enable it to detect distribution of gender in the images users liked.
4. **Scene Features**: What is in an image can be 'summarised' by its scene. We selected the PRICoLBP feature [14] which was shown to be effective for scene classification task.
5. **Computer Graphics image**: With the advancement of computer graphics, very attractive images can be rendered, which capture the attention of people. Flickr even has groups [15] where computer graphics images are exclusively posted and discussed about. At the same time there are users who like natural images. They can be distinguished using geometry features [16].
6. **Saliency: threshold count**: Saliency gives a possible point of attention of a viewer. We find that saliency also gives an idea of the scale of the image. Zoomed-in images usually have spread out saliency, which tells that there is nothing explicit in the image grabbing the attention of the user. This is also the case with images which have a scenic backdrop. But iconic images or images with clear objects show sharp saliency spikes. To capture if users

like iconic images or otherwise, we used saliency [17] to compute the average number of maximal intensity points in an image relative to a threshold.

7. **Black and white vs. color image**: People who like black & white images are believed to be attracted by the focus, subtlety of tones and versatility in those images [18]. [19, 20].
8. **Visual Clutter**: People characterized by anxious and tense behavior tend to have different tastes when compared with people with peaceful and easygoing outlook. To capture this cue, we use visual clutter which can be a good descriptor to measure the busy-ness of an image.
   It can also be used as an alternative to the number of objects in the scene as the latter is difficult to measure for natural scenes. We used the feature congestion measure [21] for measuring clutter.
9. **Tags**: We construct a dictionary of tags (with an average of 9.1 tags per image). We removed some stop words like camera names ('kodak','fujifilm','canon' etc), lens characteristics ('70mm','eos' etc), and 'generic' words like 'image' and 'photo', just having a single letter ('a-Z') or number ('0-9'). The tag feature matrix is binary and sparse unlike the visual feature matrix. We then apply sparse SVD to convert the sparse tags feature matrix into a compressed representation (refer [2]).

Table 1: List of Features (with newly added features on the right-hand column)

List of Features used	
Color: avg. intensity of RGB	**Head and Upperbody recognition**
#Edges: Canny edge detection	**Face and Pose recognition**
Texture index	**Gender identification**
Regions: using mean shift segmentation algorithm	**Scene Classification**
#Objects: using Deformable Parts Models	**Computer Graphics vs. Natural image**
#Number of Faces: using Voila Jones detector	**Saliency**
GIST Descriptors	**Black & White vs. Color image**
Entropy	**Visual Clutter**
	Tags

3 Proposed Approach

The framework for modeling user 'likes' is depicted in Fig. 3. This includes the design steps for bimodal deep knowledge representation and the process of training a model for predicting user 'likes'. There are two separate parts in this framework. (a) Identifying the features in the initial feature collection which influence a user the most in liking the image (Section 3.1). This gives us an idea of the features among the initial collection of features that would be considered relevant if we did not have mid-level representations. (b) Learning the deep bimodal representation of images (described in Section 3.2).

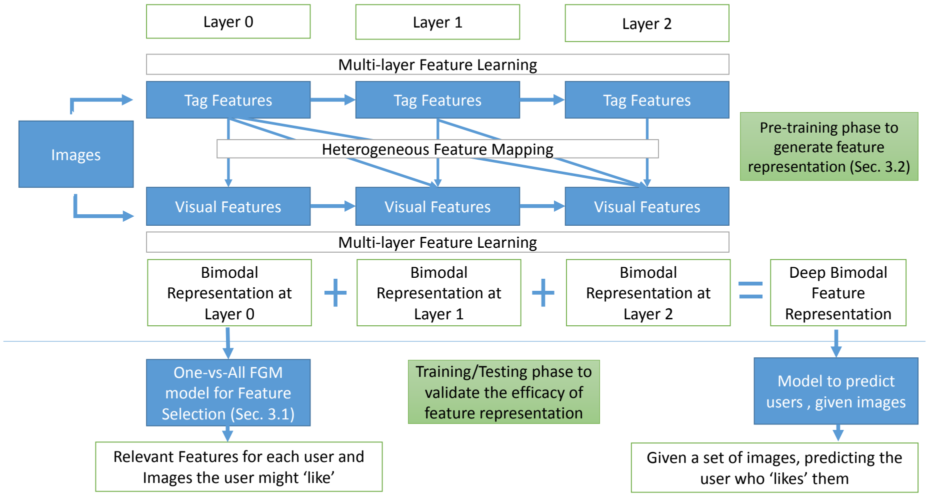

Layer 0
Layer 1
Layer 2
Multi-layer Feature Learning
Tag Features
Tag Features
Tag Features
Images
Heterogeneous Feature Mapping
Pre-training phase to generate feature representation (Sec. 3.2)
Visual Features
Visual Features
Visual Features
Multi-layer Feature Learning
Bimodal Representation at Layer 0
Bimodal Representation at Layer 1
Bimodal Representation at Layer 2
Deep Bimodal Feature Representation
One-vs-All FGM model for Feature Selection (Sec. 3.1)
Training/Testing phase to validate the efficacy of feature representation
Model to predict users , given images
Relevant Features for each user and Images the user might 'like'
Given a set of images, predicting the user who 'likes' them

Fig. 3: Framework for modeling user 'likes'.

3.1 User-specific Feature Selection

The features described in Section 2 are semantic concepts to represent an image. However it is not clear what influences a user to 'like' an image as intuitively, only a specific set of features might contribute for each user to like an image. Formally, the selection of most informative features for users can be expressed as solving a sparse SVM formulation as follows (similar to [22]):

$$\min_{\mathbf{d}} \min_{\mathbf{w},\epsilon,\rho} \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{C}{2}\sum_{i=1}^{n} {\epsilon_i}^2 - \rho \tag{1}$$

$$s.t. \quad y_i \mathbf{w}^\top (\mathbf{x}_i \odot \mathbf{d}) \geq \rho - \epsilon_i, \tag{2}$$

where $\mathbf{d} \in \mathcal{D} = \{\|\mathbf{d}\|_0 \leq B, \mathbf{d}_j \in \{0,1\}\}$ and 1, 0 indicate that the feature is selected or not respectively. $\mathbf{x}$ denotes the feature vector, $\mathbf{w}$ denotes the weight vector, ρ is the bias, ϵ_i is the ith instances loss incurred by classifiers, B is the number of features to be selected and C is the trade-off parameter. Solving the above optimization is a NP-hard problem with lots of possible combinations of features. To reduce the computation cost, we adopt the Feature Generation Machine (FGM) [22] method to learn $\mathbf{d}$ and $\mathbf{w}$ simultaneously, which represent the index of selected features and their corresponding weight. For more details of solving this problem, please refer to [22].

3.2 Learning Deep Bimodal Feature Representation

Multi-view fusion has been widely studied in machine learning to improve performance of various tasks. [23–25]. Especially, several methods for fusing visual and textual information related to images have been proposed (eg: CCA [2]). These methods essentially find linear projections of two random vectors that are maximally correlated. While these approaches have been very successful in applications like image search and so on, it must be noted that mapping features to user 'likes' is by nature non-linear and hence such linear approaches could fail. Although kernel versions of such linear projection based methods exist (KCCA [26]), the time required to train (i.e, to compute the joint representations of two domains) scales poorly with training data size. We also note that while kernel methods are aimed at finding higher dimensional space where data can be linearly separable, the motivation of this problem necessitates finding better feature representations (which are not necessarily separable in higher dimensions).

Recently, there have also been works [27] which learn multimodal representations of images using deep learning methods. However, we use deep networks to learn multi-layer nonlinear representations of both tag and visual features individually. At every layer, we "translate" the features from tag domain into visual domain. We concatenate the visual features and the "translated" tag features at every layer and use this as the representation for every image. The entire process is described formally in Algorithm 1. The initial idea of feature translation is inspired from [28] where heterogenous feature mapping was done on text domain of different languages. However, our proposed method is different from [28] in two aspects:

- The algorithm in [28] is for heterogeneous transfer learning [29], while we focus on learning multimodal representations by augmenting features for classification.
- The deep structure in [28] learns feature representation in the same layer across domains, however we propose a new architecture for multi-modal representations shown in Figure 4 based on the intuitions presented in Section 3.2. The translators learnt across layers are expected to aid in capturing more information about user likes, thereby increasing the model performance.

Multi-layer Homogeneous Feature Learning Marginalized Stacked Denoising Autoencoders (mSDA) [30] are used to form a deep network wherein layer-wise nonlinear transformations of the Visual and Textual features of an image are learned. We choose mSDA because of its advantages like faster training time and implementation simplicity, when compared to other forms of Autoencoders. The mid-level representations thus learned by the deep network are found to be very effective features for classification with SVMs [28, 27].

In presenting the formulation, we follow the notations used in [30] for simplicity. We absorb a constant feature into the selected feature vector as $\mathbf{x}_V = [\mathbf{x}_V^\top \; 1]^\top$ or $\mathbf{x}_T = [\mathbf{x}_T^\top \; 1]^\top$, and incorporate a bias term $\mathbf{b}$ within the weight matrix as $\mathbf{W}_V = [\mathbf{W}_V \; \mathbf{b}_V]$. We further denote $\mathbf{X}_V$ as Visual Domain data, and $\mathbf{X}_V$ the union tag domain data.

Firstly, for the Visual Domain data, we apply mSDA on $\mathbf{X}_V$ to learn a weight matrix $\mathbf{W}_V \in \mathbb{R}^{(d_V+1)\times(d_V+1)}$ by minimizing the squared reconstruction loss as

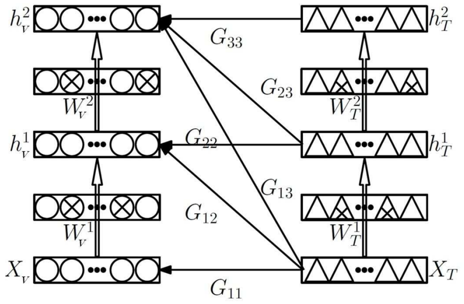


Fig. 4: Learning Deep Feature Representation.

follows,

$$\sum_{i=1}^{m} \left\| \mathbf{X}_V - \mathbf{W}_V \mathbf{X}_V^{(i)} \right\|_F^2 , \tag{3}$$

where $\mathbf{X}_V^{(i)}$ denotes the i-th corrupted version of $\mathbf{X}_V$. The solution to (3) depends on how the original features are corrupted which can be explicitly expressed as follows,

$$\mathbf{W}_V = \mathbf{P}\mathbf{Q}^{-1} \quad \text{with} \quad \mathbf{Q} = \widetilde{\mathbf{X}}_V \widetilde{\mathbf{X}}_V^\top \quad \text{and} \quad \mathbf{P} = \widehat{\mathbf{X}}_V \widetilde{\mathbf{X}}_V^\top , \tag{4}$$

where $\widehat{\mathbf{X}}_V = [\mathbf{X}_V \ \mathbf{X}_V \ \cdots \ \mathbf{X}_V]$ denotes the m-times repeated version of $\mathbf{X}_V$, and $\widetilde{\mathbf{X}}_V$ is the corrupted version of $\widehat{\mathbf{X}}_V$. In general, to alleviate bias in estimation, a large number of m over the training data with random corruptions are required, which is computationally expensive. To address this issue, mSDA introduces a corruption probability p to model infinite corruptions, i.e., $m \longrightarrow \infty$. Then a feature vector $\mathbf{q} = [1-p, \cdots, 1-p, 1]^\top \in \mathbb{R}^{d_V+1}$ is defined, where $\mathbf{q}_i$ represents the probability of a feature indexed by i "surviving" after the corruption. Thus, we can obtain the expectation of (3), and its solution can be written analytically as

$$\mathbf{W}_V = \mathbb{E}[\mathbf{P}]\mathbb{E}[\mathbf{Q}]^{-1}, \tag{5}$$

where $\mathbb{E}[\mathbf{P}]_{ij} = \mathbf{S}_{ij}\mathbf{q}_j$, $\mathbf{S} = \mathbf{X}_V \mathbf{X}_V^\top$, and

$$\mathbb{E}[\mathbf{P}]_{ij} = \begin{cases} \mathbf{S}_{ij}\mathbf{q}_i\mathbf{q}_j, & \text{if } i \neq j, \\ \mathbf{S}_{ij}\mathbf{q}_i, & \text{otherwise.} \end{cases} \tag{6}$$

After $\mathbf{W}_V$ is learned, the nonlinearity of features is injected through the nonlinear encoder function $h(\cdot)$ that is learned together with the reconstruction weights $\mathbf{W}_V$, mSDA applies a nonlinear squashing-function, e.g., the hyperbolic tangent function $\tanh(\cdot)$, on the outputs of mSDA, $\mathbf{h}_V = \tanh(\mathbf{W}_V \mathbf{X}_V)$, to generate nonlinear features.

Cross-layer Heterogeneous Feature Mapping So far, in a specific layer k of feature learning, we have learned a pair of reconstruction weights $\mathbf{W}_V^k$ and $\mathbf{W}_T^k$, and higher-level feature representations $\mathbf{h}_V^k$ and $\mathbf{h}_T^k$ for the visual and tags domain data respectively (architecture shown in Fig. 4). By denoting $\mathbf{h}_V^k$ and $\mathbf{h}_T^j$ the layer k and layer j feature representations of the cross-domain corresponding instances in the visual and tag domains respectively, we now introduce how to learn a feature mapping across heterogeneous features $\mathbf{h}_V^k$ and $\mathbf{h}_T^j$.

Algorithm 1 Deep Heterogenous Feature Mapping.

```
Input: tag domain data $\mathbf{D}_T = \{\mathbf{x}_{T_i}\}_{i=1}^{n_1}$, visual domain data $\mathbf{D}_V = \{(\mathbf{x}_{V_i}, y_{V_i})\}_{i=1}^{n_2}$,
a feature corruption probability $p$ in mSDA, a trade-off parameter $\lambda$, and the number
of layers $K$.
Initializations: $\mathbf{X}_V$, $\mathbf{X}_T$, $\mathbf{h}_V^1 = \mathbf{X}_V$, $\mathbf{h}_T^1 = \mathbf{X}_T$, and learn $\mathbf{G}_1$ by solving
    $\min_{\mathbf{G}_{1,1}} \|\mathbf{h}_V^1 - \mathbf{G}_{1,1}\mathbf{h}_{T,1}\|_F^2 + \lambda\|\mathbf{G}_{1,1}\|_F^2.$
Run FGM to select relevant features
for $i = 2, ..., K$ do
  1: Apply mSDA on $\mathbf{h}_V^{i-1}$ and $\mathbf{h}_T^{i-1}$:
      $\{\mathbf{W}_V^i, \mathbf{h}_V^i\} = \mathrm{mSDA}(\mathbf{h}_V^{i-1}, p),$
      $\{\mathbf{W}_T^i, \mathbf{h}_T^i\} = \mathrm{mSDA}(\mathbf{h}_T^{i-1}, p).$
end for
for $j = 1, ..., K$ do
  for $k = j + 1, ..., K$ do
    2: Learn heterogeneous feature mapping $\mathbf{G}_{k,j}$:
        $\min_{\mathbf{G}_{k,j}} \|\mathbf{h}_V^k - \mathbf{G}_{k,j}\mathbf{h}_T^j\|_F^2 + \lambda\|\mathbf{G}_{k,j}\|_F^2.$
  end for
end for
Do feature augmentation on visual domain data and mapped tag domain data
    $\mathbf{Z}_V = [{\mathbf{h}_V^k}^\top \; \cdots \; \mathbf{G}_{k,j}{\mathbf{h}_T^K}^\top]^\top,$
and train a classifier $f$ with $\{\mathbf{Z}_V, \mathbf{Y}_V\}$.
Output: $f$, $\{\mathbf{G}_{k,j}\}_{k,j=1}^K$, $\{\mathbf{W}_V^k\}_{k=2}^K$, and $\{\mathbf{W}_T^k\}_{k=2}^K$.
```

Specifically, from layer j of text domain to layer k of visual, we aim to learn a feature transformation $\mathbf{G}_{k,j} \in \mathbb{R}^{(d_V+1)\times(d_T+1)}$, where a bias term is incorporated by minimizing the following objective,

$$\|\mathbf{h}_V^k - \mathbf{G}_{k,j}\mathbf{h}_T^j\|_F^2 + \lambda\|\mathbf{G}_{k,j}\|_F^2, \tag{7}$$

where $\lambda > 0$ is a parameter of the regularization term on $\mathbf{G}_{k,j}$, which controls the tradeoff between the alignment of heterogeneous features and the complexity of $\mathbf{G}_{k,j}$. It can be shown that the optimization problem (7) has a closed form solution which can be written as follows,

$$\mathbf{G}_{k,j} = (\mathbf{h}_V^k \mathbf{h}_T^{j\top})(\mathbf{h}_T^j \mathbf{h}_T^{j\top} + \lambda \mathbf{I})^{-1}, \tag{8}$$

where $\mathbf{I}$ is the identity matrix of the dimensionality $d_T + 1$.

We note that learning the feature mapping between a layer of tags and all the corresponding higher layers of visual features might probably capture the semantics of the image better than mapping between the same layers. For example, given a picture of sunset - tag associated with it might be 'sunset', whereas the visual features will depict the colors in the image like 'red', 'yellow' and low clutter in the image and so on. We can see that the tag features are in this case are at a higher level of semantics when compared to visual features, therefore making the mapping between this layer of tags and higher layers of visual features more meaningful.

4 Experiments

4.1 Dataset

For the experiments, from Flickr we crawled 200 images (and tags associated with each image) each marked as favorite by 20 random users (i.e., a total of 4000 images). For every image, we extract the features mentioned in Section 2 using default-parameters of the of the softwares provided by the cited works along with low-level features mentioned in [7].

4.2 Results and Analysis

We attempt to investigate the following questions in this section:

1. How do high-level features compare with low-level features in representing user 'likes'? Can we represent a user's 'likes' more efficiently with high-level visual features?
2. What features influence a user the most in 'liking' an image?
3. How well do the bimodal deep representations perform in modeling users' preferences when compared to individual modalities?

While the first two questions deal with the efficacy of using the proposed high-level features (without any deep learning involved) in representing user 'likes', the third question deals with the efficacy of using the learned bimodal deep representations in representing user 'likes'.

Comparing high-level features with low-level features: For answering the first question, we compare the performance of high-level features and low-level features in discriminating users' 'likes'. For each user, we divide the data (images) into training and test sets (with three splits a) 25%/75% b) 50%/50%

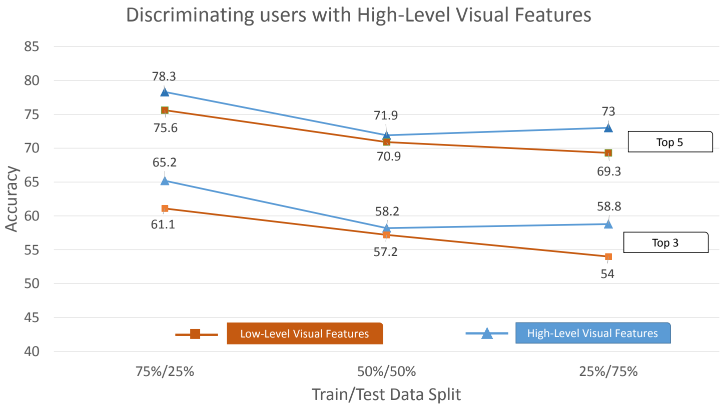


Fig. 5: High-Level features require lesser amount of training data to capture users' preferences when compared to low-level features.

and c) 75%/25%) and build a one-vs-all user classification module using FGM (described in 3.1). For each split we note the accuracy of user being correctly predicted for the test images (shown in Fig. 5). When low-level features are used, the model's performance varies based on the amount of training data, but when high-level features are used, the model performs well with less amount of training data. The reason for this can be ascertained to the amount of information the model is able to gather with each image based on the features. While low-level features would need a lot of examples to identify the patterns in users' preferences, this would not be required in case of high-level features.

Using the above one-vs-all models, we ranked the test images for each user based on the score given by the classifiers and we obtained the mean recall curves with high-level features and low-level features (shown in Fig. 6). For many users, the model also retrieves images in top-k which were not actually tagged as favorite by the user, but upon visual inspection we found that they had very similar characteristics to that of the images that the user tagged as favorite. These so-called 'false-positives' are expected because the user is not expected to have seen all the images that other users have tagged as favorite and therefore we cannot rule out the possibility that the user would like the image, provided the user sees it. As we know only the relevant images (user 'likes'), we use recall as the metric to measure the model's performance. For every user a total of 3800 images are provided as test images and within the images ranked as top 100, high-level features can retrieve about 30% of the images ‘liked’ by the user while low-level features can retrieve only 15%. Also high-level features are able to retrieve all the images 'liked' by the user around 15% faster than low-level features. This again confirms the results shown in the previous experiment (in Fig. 5).

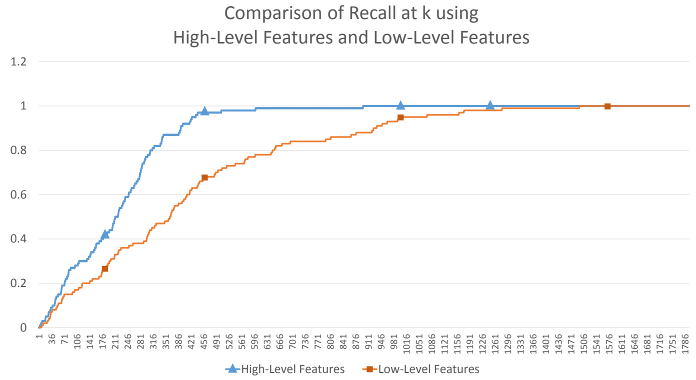


Fig. 6: Image retrieval performance to users is higher with high-level features than by using low-level features.

Features Relevant to each User: While conducting the above experiment, we also note the features selected by FGM to answer the second question. As mentioned in Section 1, we find that most users have preferences which can be captured by high-level semantic features. The weights learnt, along with random images liked by three users, are depicted in Fig. 7.

We observe that User C likes images which have pictures with people (predominantly women), User B likes colorful images and User A likes cluttered images which have a tinge of graphics in them. Also we note that there are users who have certain common preferences - for example both Users A and B prefer high clutter in images. Similarly there were other instances in the data set where more than one users had similar preferences for the presence of people in the image (for example User's 3 and 4 in Fig. 1). This observation can be used to build many interesting applications of which one is providing recommendations to a group a people (by clustering them according to personality, culture etc.).

Comparing bimodal deep representation with individual modalities: We then train multiple layers of features for each modality using the model and use heterogenous feature mapping described in Section 3.2 to learn mapping from the tag features to visual features. The intuition behind using stacked auto-encoder is that pre-training multiple layers of features can help in capturing the characteristics of an image better (due to the non-linearities). Then, we concatenate features in all the layers to form the 'deep-feature' vector. To verify the intuition behind multi-layer pre-training and also to examine the efficacy of the trained bimodal (visual+tag) representation, we compare the performance of every layer of individual modalities in discriminating the users' 'likes' with that of the combined bimodal representation. This is shown in Fig. 8.

We train a SVM model (using a 5-fold cross validation setting) on the combined (visual+tag) features. We randomly divide the data set into 25%/75% training and test sets. To see if the features are able to distinguish between

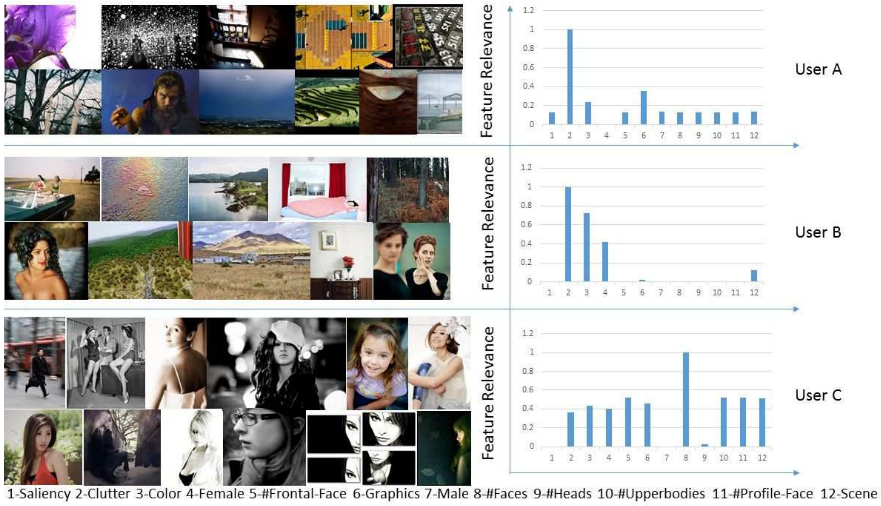


Fig. 7: Features weighted for three users and a sample of images they tagged as favorite. Users’ preferences can be represented using high-level visual features

users’ preferences, we test the accuracy of predicting the right user, given an image. Modeling user likes is a personalization problem which is user centric where the number of images liked by each user might play a more important role than the number of users themselves. Keeping this issue in mind, we considered 200 images for each user, which is similar to that used by [5, 7]. And we test the method used in [7] on our dataset and use that as the baseline for comparing the performance of our method.

The results (in Fig. 8) show a comparison of the performance of the above model under different settings (i.e., using visual features alone, using tag features alone, using bimodal features). The following observations can be made based on the Fig. 8:

– 15%-20% increase in performance is achieved at each top-k setting by using the deep bimodal representation when compared to baseline performance.
– Hierarchical representations give to 5-10% increase in performance when compared to shallow models trained on corresponding individual modalities.
– From top-1 to top-5 there is an improvement of over 35-40% using bimodal deep representation.
– Visual features model user ’likes’ better than tag features (by 20-30%) which is different from the applications like image search etc where tag features are more important.
– When compared with individual modalities, the combined representation improves the performance by 10-30% which shows that a combination of multiple modalities can model user ‘likes’ better than individual modalities.

We also found in our experiments that simple concatenation of visual and tag features give inferior performance (5%-10% at each the layer) when compared

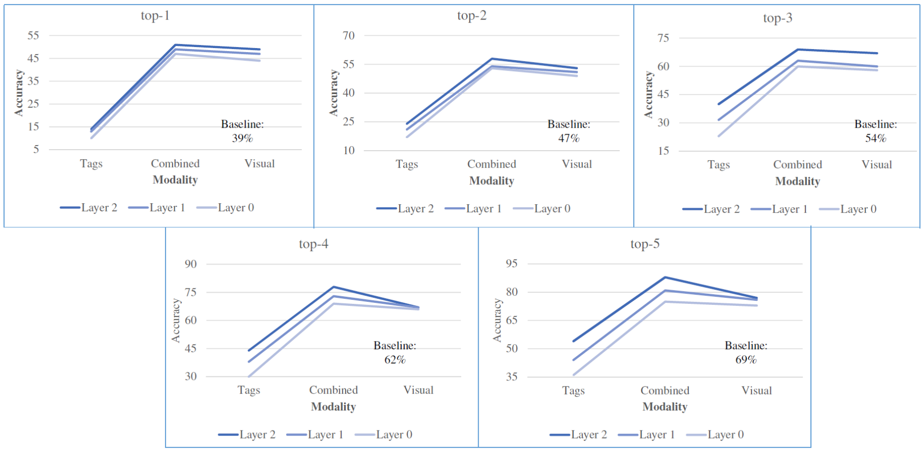


Fig. 8: **Comparison of bimodal deep representation with individual modalities.** Each graph shows the accuracy of predicting the user who tagged test images as favorite in top-k attempts. The baseline performance (with features used in [7]) is mentioned for each top-k setting. Learning multi-layer bimodal representation gives better performance than using shallow individual modalities. Layer 0 indicates the original features and Layers 1 & 2 the deep features for each modalities. 'Combined' indicates the performance of the bimodal representation at each layer.

with learning a bimodal representation using feature mapping from tag domain to visual domain. This confirms that simple concatenation of features from different domains does not lead to a good representation [2] and feature mapping is necessary to learn effective bimodal representations to model user 'likes'.

5 Conclusion

In this paper, we attempted to model users 'likes' using bimodal deep representation of images on a Flickr data set. Several syntactic, semantic, aesthetic and contextual features were used to build a deep knowledge representation for images (using visual and textual information). A feature selection strategy was applied to learn the most influential features for a user to like an image. Deep bimodal representation was learnt using novel approach for knowledge transfer between tag domain and visual domain of images to model user 'likes'. A 15%-20% increase in performance was achieved when compared to shallow models trained on low-level features (and a 5%-10% increase when compared to shallow models trained on high-level features). Further work on understanding what the mid-level representations mean and testing our method on large scale dataset is under progress.

References

1. Kennedy, L., Naaman, M., Ahern, S., Nair, R., Rattenbury, T.: How flickr helps us make sense of the world: Context and content in community-contributed media collections. In: Proceedings of the 15th International Conference on Multimedia. MULTIMEDIA '07, New York, NY, USA, ACM (2007) 631–640
2. Gong, Y., Ke, Q., Isard, M., Lazebnik, S.: A multi-view embedding space for modeling internet images, tags, and their semantics. International Journal of Computer Vision **106** (2014) 210–233
3. Lampropoulos, A.S., Lampropoulou, P.S., Tsihrintzis, G.A.: A cascade-hybrid music recommender system for mobile services based on musical genre classification and personality diagnosis. Multimedia Tools and Applications **59** (2012) 241–258
4. Mairesse, F., Walker, M.A., Mehl, M.R., Moore, R.K.: Using linguistic cues for the automatic recognition of personality in conversation and text. J. Artif. Intell. Res.(JAIR) **30** (2007) 457–500
5. Cristani, M., Vinciarelli, A., Segalin, C., Perina, A.: Unveiling the multimedia unconscious: Implicit cognitive processes and multimedia content analysis. In: Proceedings of the 21st ACM International Conference on Multimedia. MM '13, New York, NY, USA, ACM (2013) 213–222
6. Guntuku, S.C., Roy, S., Lin, W.: Personality modeling based image recommendation. In: MultiMedia Modeling, Springer (in press 2015)
7. Lovato, P., Perina, A., Sebe, N., Zandonà, O., Montagnini, A., Bicego, M., Cristani, M.: Tell me what you like and il tell you what you are: discriminating visual preferences on flickr data. In: Computer Vision–ACCV 2012. Springer (2013) 45–56
8. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: Computer Vision, 2009 IEEE 12th international conference on, IEEE (2009) 2106–2113
9. Eichner, M., Marin-Jimenez, M., Zisserman, A., Ferrari, V.: 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. International Journal of Computer Vision **99** (2012) 190–214
10. Marin-Jimenez, M., Zisserman, A., Eichner, M., Ferrari, V.: Detecting people looking at each other in videos. International Journal of Computer Vision **106** (2014) 282–296
11. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. (2012) 2879–2886
12. Ploderer, B., Howard, S., Thomas, P., Reitberger, W.: ey world, take a look at me! appreciating the human body on social network sites. In: Persuasive Technology. Springer (2008) 245–248
13. Xiong, X., De la Torre, F.: Supervised descent method and its applications to face alignment. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2013)
14. Qi, X., Xiao, R., Guo, J., Zhang, L.: Pairwise rotation invariant co-occurrence local binary pattern. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C., eds.: Computer Vision ECCV 2012. Volume 7577 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 158–171
15. Flickr, h.: Freecg (2014)
16. Ng, T.T., Chang, S.F.: Classifying Photographic and Photorealistic Computer Graphic Images using Natural Image Statistics. Technical report, ADVENT Technical Report, No. 220-2006-6, Columbia University (2004)

17. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: IEEE International Conference on Computer Vision (ICCV). (2009)
18. Rowse, D.: Why black and white photography, http://digital-photography-school.com/why-black-and-white-photography/ (2014) Retrived.
19. Machajdik, J., Hanbury, A.: Affective image classification using features inspired by psychology and art theory. In: Proceedings of the International Conference on Multimedia. MM '10, New York, NY, USA, ACM (2010) 83–92
20. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Studying aesthetics in photographic images using a computational approach. In: Computer Vision–ECCV 2006. Springer (2006) 288–301
21. Rosenholtz, R., Li, Y., Mansfield, J., Jin, Z.: Feature congestion: a measure of display clutter. In: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM (2005) 761–770
22. Tan, M., Wang, L., Tsang, I.W.: Learning sparse svm for feature selection on very high dimensional datasets. In: ICML. (2010) 1047–1054
23. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th annual conference on Computational learning theory. (1998) 92–100
24. Sindhwani, V., Niyogi, P.: A co-regularized approach to semi-supervised learning with multiple views. In: Proceedings of the ICML Workshop on Learning with Multiple Views. (2005)
25. Zhou, J.T., Pan, S.J., Qi, M., W Tsang, I.: Multi-view positive and unlabeled learning. In: Proceedings of the 4th Asian Conference on Machine Learning, ACML 2012, Singapore, Singapore, November 4-6, 2012. (2012) 555–570
26. Zheng, W., Zhou, X., Zou, C., Zhao, L.: Facial expression recognition using kernel canonical correlation analysis (kcca). Neural Networks, IEEE Transactions on **17** (2006) 233–238
27. Srivastava, N., Salakhutdinov, R.: Multimodal learning with deep boltzmann machines. In: Advances in neural information processing systems. (2012) 2222–2230
28. Zhou, J.T., Pan, S.J., Tsang, I.W., Yan, Y.: Hybrid heterogeneous transfer learning through deep learning. In: Twenty-Eighth AAAI Conference on Artificial Intelligence. (2014)
29. Zhou, J.T., W Tsang, I., Pan, S.J., Tan, M.: Heterogeneous domain adaptation for multiple classes. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics. (2014) 1095–1103
30. Chen, M., Xu, Z.E., Weinberger, K.Q., Sha, F.: Marginalized denoising autoencoders for domain adaptation. In: ICML. (2012)