Image Annotation by Exploiting Image Metadata

`Image annotation/tagging`

1. Some differences

1. **Introduce the neighborhoods of related images using Jaccard similarities**
2. **Use a deep neural network to blend visual information from the images and its neighbors**
3. **Allow the vocabulary of metadata changes between training and testing.**

2. Model

1. **What is Metadata ?:** three types of image metadata are user tags, image photo-sets, and image groups
2. **How to generate candidate neighborhoods?:** Vocabulary T, a subset $t_x \subseteq T$. each type of metadata has different means of T and t_x. For tags, T is the set of all possible user tags and t_x are the tags for image x; for groups (and sets), T is the set of all groups (sets), and t_x are the groups (sets) to which x belongs. For sets and groups, we use the entire vocabu- lary T. **Compute the distance between images using the Jaccard similarity between image metadata.**

$$d(x, x') = 1 - \frac{|t_x \cap t_{x'}|}{|t_x \cup t_{x'}|}$$

3. **How to use the neighborhoods?:**

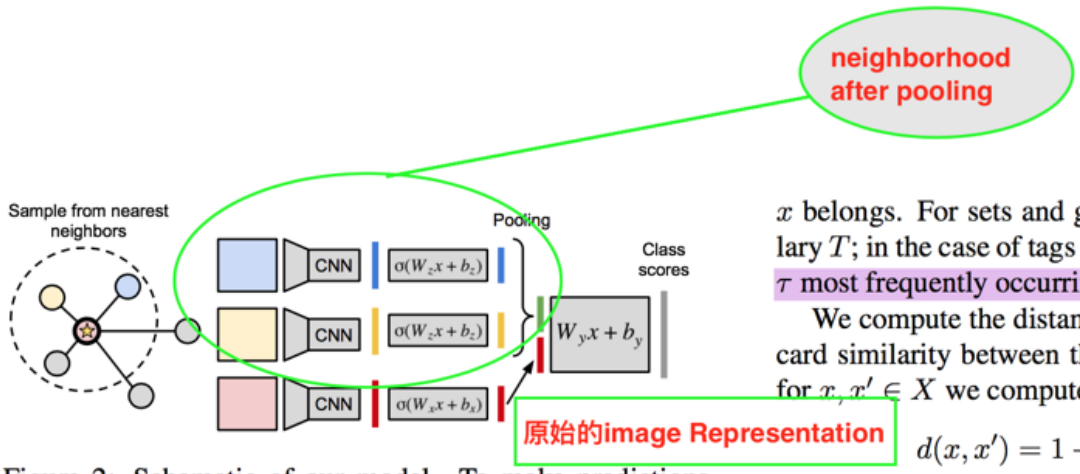

Figure 2: Schematic of our model. To make predictions for an image, we sample several of its nearest neighbors to form a *neighborhood* and we use a CNN to extract visual features. We compute hidden state representations for the image and its neighbors, then operate on the concatenation of these two representations to compute class scores.

x belongs. For sets and gr
lary T; in the case of tags v
τ most frequently occurrin
We compute the distanc
card similarity between th
for $x, x' \in X$ we compute
$d(x, x') = 1 -$
To prevent an image from
hoods, we set $d(x, x) = 0$
Generating candidate n
hyperparameters, namely t
rank M, the type of meta
and the tag vocabulary siz

For an image $x \in X$ and neighborhood $z \in Z_x$, we average these scores over all neighborhoods for x,

$$s(x; w) \frac{1}{Z_x} \sum_{z \in Z_x} f(x, z; w)$$

We choose a loss and optimize

$$w^* = arg\ min_w \sum_{(x,y) \in D} l(s(x; w), y)$$

.

In practice:, The set Z_x may be large, so for computational efficiency we approximate $s(x; w)$ by sampling from Z_x. During training, we draw a single sample during each forward pass and at test time we use ten samples.

4. **How to generate the image's label?:** present a fully-connected two layer neural network applied to features from the image and its neighborhood, **Schematic as above**. There are several matrices parameterized by $W_x \in \mathcal{R}^{dxh}. b_x \in \mathcal{R}^h, W_z \in \mathcal{R}^{dxh}, b_z \in \mathcal{R}^h$. We use following equations to generate the final score.

$$v_x = \delta(W_x \phi(x) + b_x)$$
$$v_z = max_{i=1,\ldots,m}(\delta(W_x \phi(z_i) + b_z))$$
$$f(x, w; z) = W_y[v_x, v_z] + b_y$$

PS, the $y = w * x + b$ is a kind of affine transform.

5. **Which kind of Loss in the model?:** Our loss function $\mathcal{L}$ is a sum of

independent one-vs-all logistic calssifiers. Other popular Loss for multitagging is multilabel ranking losses.

3. The model's resilience

1. **our model could be applied in cases where some types of metadata are unavailable during testing.**
2. **We see that the performance of our model degrades as we decrease the overlap between the training and testing tags; however even in the case of 0% overlap our model is able to outperform both the visual-only model**

4. Some details

1. **The best results computes neighborhoods using tags with a vocab- ulary size of τ = 5000, neighborhood size m = 3 and max-rank M = 6** and **Preliminary experiments at combining all types of metadata did not show improvements over us- ing tags alone**
2. We apply $L2$ regularization to the matrices W_x,W_z,and W_y and apply dropout with p = 0.5 to the hidden layers h_x and h_z
3. **For all experiments we use a learning rate of 1×10^{-4} , L2 regularization strength 3×10^{-3} and hidden dimension h = 500; these values were chosen using grid search.**
4. **Our image feature function ϕ returns the activations of the last fully-connected layer of the BLVC Reference CaffeNet**
5. some results better than only visual