Multilabel Image Annotation Using Deep Neural Networks

Ajitesh Singh, Goutham Kalikrishna Reddy Kuncham, K. Srikar

Abstract – Multilabel Image Tagging is one of the most important challenges in computer vision with many real world applications and thus we have used Deep Neural Networks for Image Annotation to boost performance. This experiment is performed on NUS-WIDE Dataset with 1K Tags.

Keywords: Multilabel; Deep Neural Networks; Image Annotation;

I. INTRODUCTION

Multilabel image annotation is an important and challenging problem in computer vision. Most existing work focus on single-label classification problems, where each image is assumed to have only one class label. However, this is not necessarily true for real world applications, as an image may be associated with multiple tags. As a practical example, images from Flickr are used as an example which has multiple tags, such as objects, activities, and scene descriptions. Images on the Internet, in general, are usually associated with sentences or descriptions, instead of a single class label, which is a type of multitagging. Therefore, it is a practical and important problem to accurately assign multiple labels to one image. Single-label image classification has been extensively studied in the vision community, the most recent advances reported on the large-scale ImageNet Architecture. Most existing work focus on designing visual features for improving recognition accuracy. For example, sparse coding, Fisher vectors, and VLAD have been proposed to reduce the quantization error of “bag of words”-type features. Very recently, deep convolutional neural networks (CNN) have demonstrated promising results for single-label image classification. Such algorithms have focused on one vs all, but one have worked on the multilabel image annotation problem. In this work, we used the highly expressive convolutional network for the problem of multilabel image annotation. We employed a similar network structure to as used in Image Net, which contains several convolutional and dense connected layers as the basic architecture. We studied and compared several other popular multilabel losses, such as the ran-king loss that optimizes the area under ROC curve (AUC), and the cross-entropy loss used in Tagprop. Specifically, we propose to use the top-k ranking loss, inspired by, for embedding to train the network. Using the largest publicly available multilabel dataset NUS-WIDE, we observe a significant performance boost over conventional features, reporting the best retrieval performance. We even performed test on Triplet Loss giving much better results.

II. MOTIVATION

Multilabel image annotation is an important and challenging problem in computer vision. Most existing work focus on single-label classification problems, where each image is assumed to have only one class label. However, this is not necessarily true for real world applications, as an image may be associated with multiple tags. As a practical example, images from Flickr are used as an example which has multiple tags, such as objects, activities, and scene descriptions. Images on the Internet, in general, are usually associated with sentences or descriptions, instead of a single class label, which is a type of multitagging. Therefore, it is a practical and important problem to accurately assign multiple labels to one image.
III. LITERATURE SURVEY

Various research works has already been done on the images over the Internet by the Computer Vision Community. The major research topic has been done in the field of Image Classification, Images with Captions and other such related field. In this project we have worked on Image Annotation problem. There have been researchers who have tried to establish a mapping between text and image features. Image Tagging is now a classification problem. There have been models which have worked upon the generative model based tagging and learning from parametric models. As we can see that Image Annotation is a highly non-linear problem we had to make sure that we are able to capture the complex distribution of data. There have been works done non-parametric nearest-neighbours of the data. Makadia.et.al works, which proposed a simple nearest-neighbor-based tag transfer approach, achieved significant improvement over previous model-based methods. Tagprop is also a recent work which learns a discriminative metric for nearest neighbors to improve tagging. In this work we proposed a Convolutional Neural Network based approach, which utilizes specific network structures, such as convolutions and spatial pooling, and have exhibited good generalization power in image-related applications.

We have used a Dropout technique used in CNN for better results. There have been different pooling methods for training CNNs, and several different regularization methods, such as Dropout, DropConnect, and Maxout have been proposed to improve the robustness and representation power of the networks. Our work focuses on how to train a deep network from raw pixels, using multilabel ranking loss, to address the multilabel annotation problem. More exact, all (or most) the training data is needed during the testing phase. KNN Algorithm is based on feature similarity: How closely out-of-sample features resemble our training set determines how we classify a given data point. We used the K-Nearest Neighbour for classification of the model. K Nearest Neighbors - Classification K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure. KNN is a non-parametric, lazy learning algorithm. When we say a technique is non-parametric, it means that it does not make any assumptions on the underlying data distribution. In other words, the model structure is determined from the data. If you think about it, it's pretty useful, because in the “real world”, most of the data does not obey the typical theoretical assumptions made (as in linear regression models, for example). Therefore, KNN could and probably should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution data.

Algorithm:

Let m be the number of training data samples. Let p be an unknown point.
Store the training samples in an array of data points arr[]. This means each element of this array represents a tuple (x, y).
for i=0 to m:
Calculate Euclidean distance d(arr[i], p).
Make set S of K smallest distances obtained. Each of these distances corresponds to an already classified data point.
Return the majority label among S.

There were many visual features which we could have used but we decided on keeping two visual features for our model. Although such a set of features might not have been the best possible ones we could obtain, they already serve as a very strong visual representation, and the computation of such features is nontrivial.

SIFT - We used two different sampling methods and three different local descriptors to extract texture features, which gave us a total of 6 different features. We used dense sampling and a Harris corner detector as our patch-sampling methods. For local descriptors, we extracted SIFT, CSIFT, and RGBSIFT, and formed a codebook of size 1000 using kmeans clustering; then built a two level spatial pyramid that resulted in a 5000-dimensional vector for each image. We will refer to these six features as D-SIFT, D-CSIFT, D-RGBSIFT, H-SIFT, H-CSIFT, and H-RGBSIFT.

HOG - To represent texture information at a larger scale, we used 22 overlapping HOG as described in Large-scale scene recognition from abbey to zoo paper. We quantized the HOG features to a codebook of size 1000 and used the same spatial pyramid scheme as above, which resulted in 5000-dimensional feature vectors. The same sets of features were used in A multi-view embedding space for internet images, tags, and their semantics. IJCV, 2013. and achieved state-of-the-art performance for image retrieval and annotation. The combination of this set of features has a total dimensionality of 36,472, which makes learning very expensive.

IV. MATHEMATICAL MODELING

We have used the Convolutional Neural Network for the Multi Label Image Annotation problem. The image below shows how the Convolution operation takes place in the network. The architecture used is similar to used in ImageNet. The activation function used is the RELU activation function. The mathematical function for the same is

\[ f(x) = x^+ = \max(0, x) \]
Next we used the asynchronized stochastic gradient descent to optimize the whole network for better accuracy. Stochastic Gradient descent is using the cost gradient of 1 example at each iteration, instead of using the sum of the cost gradient of all examples which makes it better than other optimizer function. Asynchronous SGD (ASGD) is adopted and with which no barrier is imposed, and each local worker continues its training process right after its gradient is added to the global model. Although ASGD can achieve faster speed due to no waiting overhead, it suffers from another problem which we call delayed gradient. That is, before a worker wants to add its gradient g(wt) (calculated based on the model snapshot wt) to the global model, several other workers may have already added their gradients and the global model has been updated to wt+ (here is called the delay factor). Adding gradient of model wt to another model wt + does not make a mathematical sense, and the training trajectory may suffer from unexpected turbulence.

The loss function used in the whole network is the Softmax Function (or Cross Entropy Loss Function). This loss function was also used in single label image classification. The posterior probability of an image \( x_i \) and class \( j \) and \( k \) be expressed as

\[
p_{ij} = \frac{\exp(f_i(x_i))}{\sum_{k=1}^{c} \exp(f_k(x_i))}
\]

**V. RESULTS AND DISCUSSION**

We performed experiments on the largest publicly available multilabel dataset, NUS-WIDE. This dataset contains 269,648 images downloaded from Flickr that have been manually annotated, with several tags (2-5 on average) per image. After ignoring the small subset of the images that are not annotated by any tag, we had a total of 209,347 images for training and testing. We used a subset of 150,000 images for training and used the rest of the images for testing. The tag dictionary for the images contains 81 different tags. We followed previous research in our use of the following protocols to evaluate different methods. For each image, we assigned \( k \) (e.g., \( k = 3, 5 \)) highest-ranked tags to the image and compared the assigned tags to the ground-truth tags. There have been two metrics for the evaluation of the accuracy of the model.

![Figure 4: Qualitative image annotation results obtained with WARP](image)

No. of similar items

Semantics between the tag mean and the tag and distance between the two using NLTK. This helped us know how good our model is performing.
In the dataset, each image had different numbers of ground-truth tags, which made it hard for us to precisely compute an upper bound for performance with different k. For each image, when the number of ground-truth tags was larger than k, we randomly chose k ground-truth tags and assigned them to that image; when the number of ground-truth tags was smaller than k, we assigned all ground-truth tags to that image and randomly chose other tags for that image. We believe this baseline represents the best possible performance when the ground truth is known. The results for assigning 3 keywords per image are reported in Table 1. The results indicate that the deep network achieves a substantial improvement over existing visual-feature-based annotation methods. CNN network with different loss functions, results show that softmax already gives a very powerful baseline.

VI. CONCLUSION

This model was an minimized implementation of the previous work done on Multilabel Image Annotation by Google Researchers. We proposed to use ranking to train deep convolutional neural networks for multilabel image annotation problems. We investigated several different ranking-based loss functions for training the CNN, and found that the Softmax loss works particularly well for multilabel annotation problems. We performed experiments on the largest publicly available multilabel image dataset NUS-WIDE, and demonstrated the effectiveness of using top-k ranking to train the network. In the future, we would like to use very large amount of noisy-labeled multilabel images from the Internet (e.g., from Flickr or image searches) to train the network.

REFERENCES

2. Deep Convolutional Ranking for Multilabel Image Annotation, Arxiv, 2014