

A Project Report
ON
Image Annotation

Submitted By
Rajat Sharma B18CSE043
Priyanshi Jain B18CSE042

Under the Supervision of
Dr. Yashaswi Verma



॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Bachelor of Technology
Department of Computer Science and Engineering
Indian Institute of Technology, Jodhpur

2020

ACKNOWLEDGEMENT

We express our sincere thanks and deep gratitude to our guide and supervisor, Dr. Yashaswi Verma, Professor, Indian Institute of Technology, Jodhpur, for his unwavering support, suggestions, encouragement, guidance and patience for accomplishing this project work. With great respect, we express our gratitude to him for his contribution during the progress and development of this project work. We have benefitted a lot from and inspired greatly from the breadth and depth of his knowledge and his immense talent. He has taught us, not only how to undertake project work, but also how to communicate and present. We are fortunate to be associated with him, his active participation, untiring efforts, affection, guidance and approach has brought this work to present stage.

We would like to take the opportunity to thank all our faculty members and class mates for their constant encouragement and frequent lively discussions, which helped us in understanding and completion of the work.

Rajat Sharma

Priyanshi Jain

ABSTRACT

Image annotation refers to a process by which the computer assigns labels/captions to an image, which has many applications in today's world and hence is a topic of great interest. In this project, we have studied and implemented four different approaches to image annotation, namely *KNN based Label Transfer* ^[1], *Multi Label Image Recognition with Graph Convolutional Networks (ML-GCN)* ^[2], *Incorporating Semantic Information in ML-GCN* ^[3], and *Training GCN based Binary Classifiers* to perform Image Annotation. The techniques provide different approaches towards the problem, and hence have their parameters and values to tune. A comparative analysis is done for all the methods on the basis of 5 metrics, i.e. *Precision(P)*, *Recall(R)*, *F1 score(F1)*, *Semantic Precision(SP)*, *Semantic Recall(SR)* and *Semantic F1 score(SF1)*. In the end, we provide a conclusion to the report in the form of explanation of results, and improvements that can be made in the methods.

TABLE OF CONTENTS

Contents	Page No.
ACKNOWLEDGMENT	I
ABSTRACT	II
Chapter-1: Introduction	1-2
1.1 What is Image Annotation?	1
1.2 Image Annotation Methods	1-2
1.2.1 Label Transfer Using KNN	1
1.2.2 Multi Label Classification using GCN	2
1.2.3 Binary Classification using GCN	2
Chapter-2: Methodology	3-10
2.1 KNN Based Label Transfer	3-4
2.1.1 Label Transfer Using KNN	3
2.1.2 Implementation	4
2.2 Multi-label Classification using GCN	5-7
2.2.1 Ideology	5
2.2.1.1 Graph Convolutional Networks (GCN)	5
2.2.1.2 GCN for Multi Label Recognition	5
2.2.1.3 Correlation Matrix of ML-GCN	6

2.2.2	Implementation	7
2.3	ML-GCN with Semantic Information	7-9
2.3.1	Ideology	7
2.3.1.1	Semantic Hierarchy(SH)	7
2.3.1.2	Synonyms	8
2.3.1.4	Weighted Semantic path(SP)	8
2.3.2	Implementation	9
2.4	GCN Based Binary Classifiers	9-10
2.4.1	Ideology	9
2.4.1.1	Sampling Data for Training	9
2.4.1.2	GCN based Binary Classifier	10
2.4.2	Implementation	10
Chapter-3:	Results and Discussion	11-13
3.1	Metrics Considered	11-12
3.1.1	Precision	11
3.1.2	Recall	11
3.1.3	F1 Score	12
3.1.4	Sematic Metrics	12
3.2	Results	12
3.3	Analysis	13
Chapter-4:	Conclusion and Suggestions	14-15
4.1	Conclusion	14
4.2	Suggestions	14-15
REFERENCES		16

CHAPTER-1

INTRODUCTION

1.1 What is Image Annotation?

Image Annotation is the process by which a computer assigns labels to an image sample, which can be used in image retrieval systems to organize and locate images of interest from a database.

This form of classification is done by analysing the features of a given image and comparing them with already labelled images to assign captions to them. The methods used for this analysis may vary, and this is what we try to study in this report.

1.2 Image Annotation Methods

Image Captioning can be accomplished using several different methods. In this report we explore them and try to modify and appropriate them for our given task. The main methods that we analyse are:

1.2.1 Label Transfer Using KNN

The KNN method forms a baseline for our experiments and uses greedy heuristics for performing label transfer for K nearest neighbours of an image

1.2.2 Multi Label Classification using GCN

The given method performs aggregates information for labels using GCN operations performed on its correlation matrix, in order to generate label classifiers, which are then used for annotating image samples.

1.2.3 Binary Classification using GCN

In this method we create a GCN based classifier for each of the given labels, and then assign a label score for each image. By selecting the K labels by highest score for each image, we perform image annotation.

CHAPTER-2

METHODOLOGY

2.1 KNN Based Label Transfer

2.1.1 Ideology

In this treat image annotation problems as that of the image retrieval problem using the KNN algorithm.

We calculate the k nearest neighbours using standard KNN algorithm and then transfer labels to images from the nearest neighbours.

The different distance metrics mentioned in [1] are

$$L_1 = \sum (x_i - y_i)$$
$$L_2 = \sqrt{\sum (x_i - y_i)^2}$$
$$\chi^2 = \frac{1}{2} \sum \frac{(x_i - y_i)^2}{(x_i + y_i)}$$

For the DIA features, we have used L_2 distance on the ESPGAME dataset.

2.1.2 Implementation

The given technique uses L_2 normalized features for each image, and finds L_2 distances between the test images and the train images. For each test image it takes the nearest $K = 5$ neighbours and then performs a label transfer using a greedy algorithm described below [1].

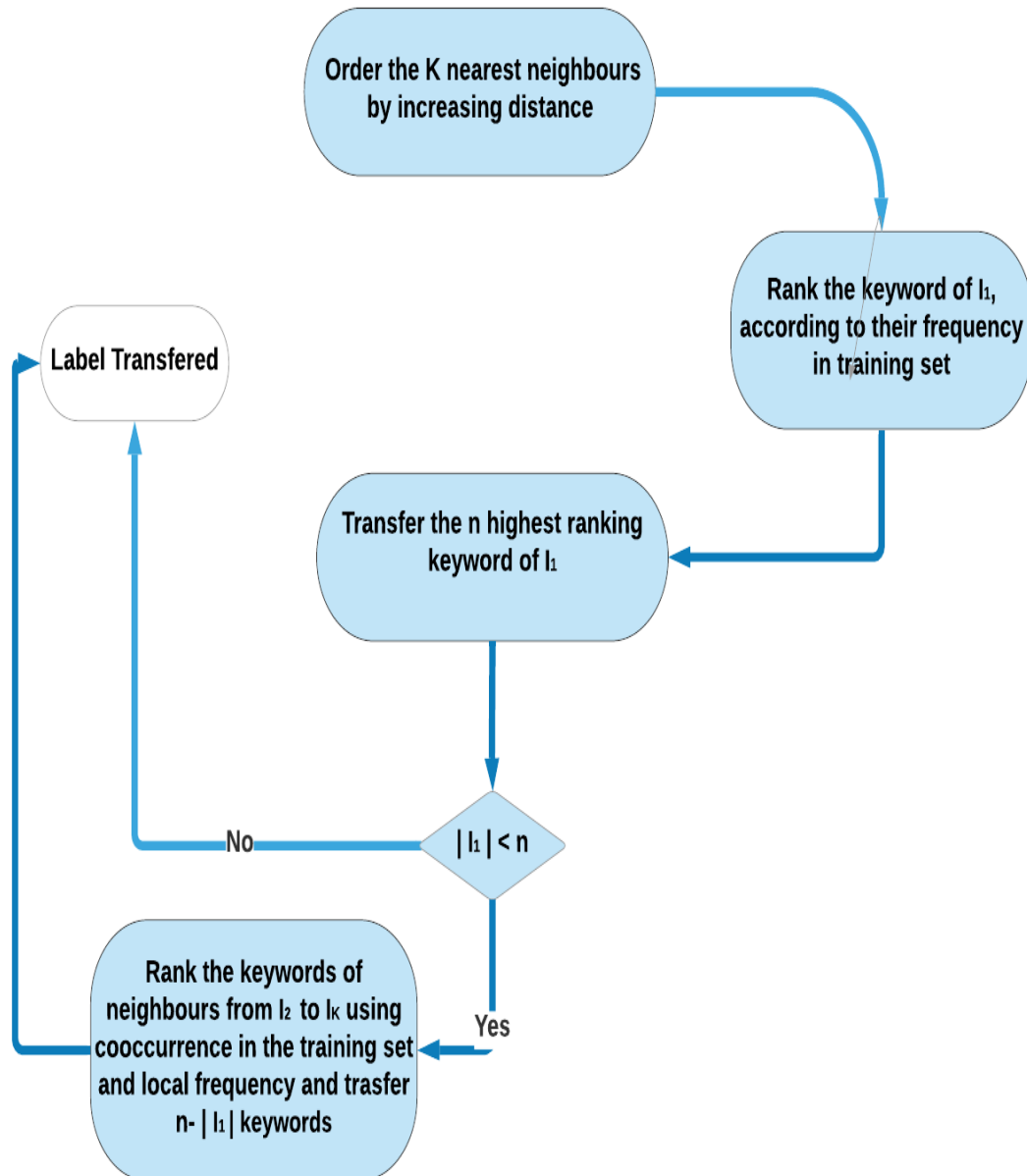


Figure-1: Image Label Transfer Algorithm

2.2 Multi Label Classification using GCN

2.2.1 Ideology

2.2.1.1 Graph Convolutional Networks (GCN)

The goal of GCN [4] is to learn a function f of features on a graph $\mathcal{G} = (\mathcal{V}, E)$, which takes a feature description matrix \mathcal{X}^ℓ of dimensions $N \times D$ and the corresponding correlation matrix \mathcal{A} of dimensions $N \times N$, where N denotes the number of nodes, and D the dimensions of the node feature vectors. Every GCN layer can be written as:

$$\mathcal{X}^{\ell+1} = f(\mathcal{X}^\ell, \mathcal{A})$$

where $\mathcal{X}^{\ell+1}$ is a $N \times D'$ matrix. The layer wise propagation rule is as follows:

$$\mathcal{X}^{\ell+1} = \sigma(\mathcal{A}' \mathcal{X}^\ell \mathcal{W}^\ell),$$

where \mathcal{W} is a $D \times D'$ weight matrix, \mathcal{A}' is the normalized correlation matrix, and $\sigma(\cdot)$ is a nonlinear function, here LeakyReLU.

2.2.1.2 GCN for Multi Label Recognition

The method for ML-GCN consists of two parts, i.e. extracting image feature representations and performing GCN on the label representations. The image features are taken from the ResNet-101 [9] which is pre-trained on ImageNet [10], with each image represented as a 2048-dimension vector. For the label representations, we use a stacked GCN consisting of two layers, with output dimensions of 1024 and 2048 respectively. Thus the GCN operation takes in a label representation \mathcal{Z} of dimension $N \times D$ and returns output \mathcal{W} of dimensions $N \times D'$ where N is the number of labels and D and D' are dimensions of feature

representations. The output classifier \mathcal{W} is then applied to image representations \mathbf{x} to get predicted scores as

$$\mathbf{y} = \mathcal{W}\mathbf{x}$$

The matrix \mathbf{y} of dimensions $N \times I$ gives the predicted scores of each label for an image. The entire network is trained using the multi-label classification loss function.

2.2.1.3 Correlation Matrix of ML-GCN

The correlation matrix, as defined in [2], relates the nodes i.e. the labels with each other, hence serving as an adjacency matrix for the graph formed by these nodes. The correlation values indicate the probability $P(L_j|L_i)$ which denotes the probability of occurrence of label L_j given that label L_i already occurs. Thus to construct the correlation matrix, we first the co-frequency matrix \mathcal{M} , where \mathcal{M}_{ij} indicates the number of times label L_i and L_j occur together. Thus we can construct the probability matrix \mathcal{P} as

$$\mathcal{P}_{ij} = \frac{\mathcal{M}_{ij}}{N_i}$$

So that \mathcal{P}_{ij} denotes the probability that label L_j appears given that label L_i is already present, and N_i is the number of occurrences of label L_i .

Using the matrix \mathcal{P} , we can construct the binary correlation matrix \mathcal{A} as

$$\mathcal{A}_{ij} = \begin{cases} 0 & \text{if } \mathcal{P}_{ij} < \Gamma \\ 1 & \text{if } \mathcal{P}_{ij} \geq \Gamma \end{cases}$$

where a threshold Γ is used to filter noisy edges.

To prevent over smoothing after GCN, a re-weighted correlation matrix \mathcal{A}' is constructed as:

$$\mathcal{A}'_{ij} = \begin{cases} \frac{p}{\sum_{\substack{j=1 \\ i \neq j}}^C \mathcal{A}_{ij}} & \text{if } i \neq j \\ 1 - p & \text{if } i = j \end{cases}$$

where p is a hyper parameter which determines the weight assigned to a node itself and other correlated nodes.

2.2.2 Implementation

The implementation for this method in the project is nearly identical to that of [2], the results being measured on the ESPGAME dataset, with images being taken from [5] and train/test splits from [3]. The label representations come from the 300-dim GLoVe [7] trained on the Wikipedia dataset. For the correlation matrix, we have set $\Gamma = 0.4$ and $p = 0.2$. For the activation function, we have chosen LeakyReLU [8] with a negative slope of 0.2. ResNet-101 [9] is used for feature extraction, which is pre-trained on ImageNet [10]. For network optimization, SGD is used as the optimizer, with momentum set to 0.9. The Weight Decay is 10^{-4} . The model is trained for 100 epochs and is implemented in PyTorch.

2.3 ML-GCN with Semantic Information

2.3.1 Ideology

2.3.1.1 Semantic Hierarchy(SH)

Semantic Hierarchy is a way to explore semantic dependencies between tags. The Primary link between semantic types is the “is-a” link. Let’s consider two tags: apple and fruit. Here Apple “is a” fruit. Therefore, in

a semantic hierarchy between these two tags, the fruit is the ancestor of apple, and apple conveys more information about image than fruit.

2.3.1.2 Synonyms

Synonyms in the pair of tags that share the same meaning. Consider two tags, women and lady, they have the same meaning thus they are synonyms.

2.3.1.4 Weighted Semantic path(SP)

We now combine the idea of Semantic Hierarchy and synonyms. In some cases, two paths may represent a similar meaning as their tags are synonyms, so we merge those paths that are different only at anonymous tags. All semantic paths corresponding to the whole tag set T is denoted as

$$SP_T = \{sp_1, \dots, sp_r\}$$

We take two important observations into consideration

1. The weight of the descendant tag should be higher than the weights of its ancestor tags.
2. The more descendent of a tag implies that the tag is less discriminative and vice versa

Combining both observations

$$\omega_{ij} = \frac{\tau^{l_{ij}}}{|d_i|}$$

where $|d_i|$ indicates the number of descendants of y_i , l_{ij} represents the layer of y_i in sp_j , and $\tau \in (0, 1)$. Here we set $\tau = 0.7$.

The weight of tag y_i in the whole set of semantic paths is defined as the sum of its weights in all semantic paths, i.e.,

$$\omega_i = \sum_j^{|SP|} \omega_{ij}$$

The weights of all tags can be concatenated into one vector

$$\omega = (\omega_1, \dots, \omega_m)$$

2.3.2 Implementation

We generate a new correlation matrix.

$$\mu_{ij} = \frac{\omega_i}{\omega_j} \mathcal{M}_{ij}$$

where \mathcal{M}_{ij} is the cofrequency matrix for the labels. This matrix is now used to create the correlation matrix in ML-GCN instead of \mathcal{M}_{ij} .

2.4 GCN Based Binary Classifiers

2.4.1 Ideology

2.4.1.1 Sampling Data for Training

When considered for a single label, on average the training dataset is quite large when compared to the actual number of images to which the label is assigned. If such values are considered for training the classifier, it can create a bias in favour of negative labelling. Thus a need for sampling the data is created so as to have an equal number of positive and negative samples.

This is accomplished by using the cofrequency matrix between labels, from where we assign negative samples to a label on the basis of its cofrequency with other labels. Suppose \mathcal{N}_{ij} is the number of images that contain label L_j but don't contain label L_i . Then,

$$\mathcal{N}_{ij} = \frac{\mathcal{M}_{ij}}{\sum_k \mathcal{M}_{ik}} \mathcal{C}_i$$

Where \mathcal{M}_{ij} is the cofrequency of labels L_i and L_j and \mathcal{C}_i is the total number of positive samples of label L_i .

2.4.1.2 GCN based Binary Classifier

The binary classifier that we use to classify images for a single label uses a single layer GCN model to operate on the neighbourhood aggregation of train data for each label.

The first part consists of performing a neighbourhood feature aggregation for each image, which consists of the mean of the features of the top 5 nearest images (considered using L2 distance) for each image.

This aggregate image data is then fed to the model to train, and serves as a combination of the correlation matrix and feature matrix in the original GCN. For the layer forward pass, a sigmoid function is used as the nonlinear activation function, and MSE loss is considered for optimization.

2.4.2 Implementation

The image feature representations are taken from the VGGF features from [3] for the ESPGAME dataset, where each image is represented as a 597-dimension feature vector. Training is done for 2500 epochs for each label, with 0.1 as the learning rate. The weights and aggregated features returned from the classifier are then multiplied to get test scores, which are stored in a matrix of images and labels. For each image, we pick the top 5 labels with the highest score, and assign them to it.

CHAPTER 3:

RESULTS AND DISCUSSION

3.1 Metrics Considered

3.1.1 Precision

Precision measures the probability that an assigned label is correct, i.e. the accuracy of the model. We define P_i for label L_i as:

$$P_i = \frac{\text{Number of correctly captioned samples}}{\text{Number of times the label is predicted}}$$

3.1.2 Recall

Recall is defined as the ratio of correctly assigned samples to that of the number of samples in the ground truth, i.e. R_i for label L_i is:

$$R_i = \frac{\text{Number of correctly captioned samples}}{\text{Number of times it appears in the ground truth}}$$

3.1.3 F1 Score

F1 score is the harmonic mean of mean precision(P) and mean recall(R), i.e.

$$F = \frac{2 * P * R}{(P + R)}$$

3.1.4 Semantic Metrics

Suppose y is the ground truth label subset, and y' is the predicted label subset for label L_i . Then semantic paths SP_y and $SP_{y'}$ are constructed, and algorithm-2 from [3] is used to calculate semantic precision, recall and F1 score.

3.2 Results

Method	P	R	F1	SP	SR	SF1
KNN	28.74	23.6	26			
ML-GCN	54.7	34.8	42.5			
ML-GCN with Semantic Information	57.81	16.7	25.9			
GCN based Binary Classifier	12.5	8.3	9.98			

Table-1: Results Obtained (in %)

3.3 Analysis

The results obtained from application of the methods to the ESPGAME dataset are shown in the above table. The first method KNN, which we considered as a baseline method performs well enough, since its metrics are similar to those obtained in [1] for the same dataset. ML-GCN outperforms the rest with great results, and is the best performing method in our project. From the table we can theorise that incorporating semantic information into ML-GCN disturbs the graph structure, which could account for the disparity between precision and recall. GCN based Binary Classifiers method performs the worst as can be seen, which can perhaps be improved by normalizing the CNN features beforehand.

CHAPTER 4:

CONCLUSION AND SUGGESTIONS

4.1 Conclusion

In this report we have presented work done in B.Tech. Project CS299. We have explored the basic concepts of image annotation and implemented the methods to perform the same. We modified the method of ML-GCN to incorporate semantic information, applied KNN baseline algorithm to VGGF CNN features, and created a method to use binary classifiers for multi-label classification. We also created a new train data sampling algorithm for binary classifiers. The learning outcome is that we got an insight into the world of machine learning, and we learnt several new and important concepts.

4.2 Suggestions

After a thorough investigation

- 1 ML-GCN results can be improved by decreasing the learning rate after a fixed number of epochs, in order to decrease the chances of overshooting of minima.

- 2 Better constructs can be explored for incorporating semantic information into ML-GCN.
- 3 The label transfer algorithm can be explored further in hopes of getting better results.
- 4 Performing normalization of features beforehand can perhaps get better results in the binary classifier method.

REFERENCES

- [1] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *ECCV*. 2008.
- [2] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, Yanwen Guo. Multi-Label Image Recognition with Graph Convolutional Networks. *arXiv preprint arXiv:1904.03582*, 2019
- [3] B. Wu, F. Jia, W. Liu and B. Ghanem, "Diverse Image Annotation," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 6194-6202, doi: 10.1109/CVPR.2017.656.
- [4] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, pages1–10, 2017.
- [5] lear.inrialpes.fr/people/guillaumin/data.php
- [6] Verma, Y., Jawahar, C.V. Image Annotation by Propagating Labels from Semantic Neighbourhoods. *Int J Comput Vis* **121**, 126–148 (2017). <https://doi.org/10.1007/s11263-016-0927-0>
- [7] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014
- [8] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, pages 1–6, 2013
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages770–778, 2016
- [10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and LiFei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009