Image Tagging

Ranjani K
U.G Scholor
SRM Institute of Science and Technology
Vadapalani,
Chennai,
Tamil nadu,India

Gokul V
U.G Scholar
SRM Institute of Science and Technology
Vadapalani,
Chennai,
Tamil nadu,India

Sridevi R
U.G Scholor
SRM Institute of Science and Technology
Vadapalani,
Chenai,
Tamil nadu,India

Mrs.M.Poonkodi
Assistant Professor
SRM Institute of Science and Technology
Vadapalani,
Chennai,
Tamil nadu,India.

***Abstract*— In this current world, the need for tagging an image has been increased. Social media plays a major role in people's lives, which created the need for image tagging. People's phones and systems are dumped with images without proper tags and it is hard to find an image. Previous image tagging models have a limitation such that it could only tag personal images and the web images will not be tagged. Image tagging is based on labelling or tagging images. The technology enables the automatic assignment of tags or relevant keywords to a vast collection of images using Transfer learning. The features extracted from the image using CNN are fed into an artificial neural network. The tags predicted by neural network is mapped to word vector model, which are then added along with the Zero shot tagging that has the ability to assign tags to images outside its own training class examples. We experiment with a large scale MLflickr-25k dataset.**

Keywords—ImageTagging,Transferlearning,Convolutional neural network.

I. INTRODUCTION

It is important to tag images and photos in an efficient way. Image tagging is the process of labelling or tagging images. New techniques can be adopted which can assign tags to images beyond limited tags that are present during training. Therefore, the obvious solution would be Zero shot tagging because it assigns unseen tags during inference.

In the existing method of the Zero shot classification, only one unseen tag/label is assigned to an image/picture. There is another challenge in the learning approach which is weakly supervised owing to its capability to seek out a Bounding box for every tag without requiring any box annotation throughout the training. To overcome the above challenge, zero shot tagging technique along with GloVe model (word embedding) to tag multiple tags per image.

Jio Lang fu[1] in their paper proposed a deep learning approach to tag personal photos. This approach solves semantic distribution gap. Visual appearance gap can be solved by bottom up and top down approaches. In addition, two modes-single and batch mode, that is employed to effectively tag personal photos where the batch mode is highly effective. They conducted experimental analysis on 7000 real photos and Adobe 5k photo dataset. .

The effective image tagging method in Young Rui's [3] paper generally consists of two stages, which involve initial image tagging and ulterior Tag refinement. Image Tagging aims to Tag an image with one or a lot of Human-friendly ideas to project the visual content of an image. In Image tag refinement, the tactic aims to get rid of indefinite and incomplete tags. The prevailing works principally concentrates on cloud-based solutions, which comprises of data of image transmitting from client to cloud and online searching images on cloud.

Multiple instance learning (MIL) and Deep neural network in Shafin Rahman's[4] paper, are together used for multi label classification tagging, Image captioning and text analysis. In many cases MIL uses max or min pooling. In recent years Zero shot learning (ZSL) has provided exciting progress to classify an image to an unseen class from an untrained dataset. In this system, they also have introduced the first unified network for zero shot image tagging known as Deep0tag. Deep0tag can by itself locate similar image patches. This paper is trained on MSCOCO datasets.

Hanh T.H Nguyen[2] in their paper, provide content aware image tag suggestion that warps the image-based feature and historical tagging information in a factorization model. It applies Art deep learning image classification and object detection method by using transfer learning to obtain features from the images. It will feed both tagging history and image information into factorization model for recommending tags. The visual and the object-based features can improve the performance up to 1.5%.

In Image Tagging Classification [5], it is a software that will take images as inputs and tag them based on the features using auto tagging. To tag images based on auto tagging algorithm, the training data set is created and then when a user uploads an image, the software will automatically tag the image using the features of the image. The images are also classified using the algorithm based on neural networks. The clustering algorithm will classify the images based on pixels, shape, size, texture, geometry and context.

II. RELATED WORKS

The above-mentioned existing system [1], has dominantly focused on tagging personal photos or images. Moreover, the system [1], uses two methods to tag images namely single and batch mode to discover immediate image representation, but the single mode is less efficient as it examines the content of only a single photo/image. To overcome these challenges, this paper focuses on tagging multi class and multi label objects and it also tags images and objects from an untrained dataset using word embedding.

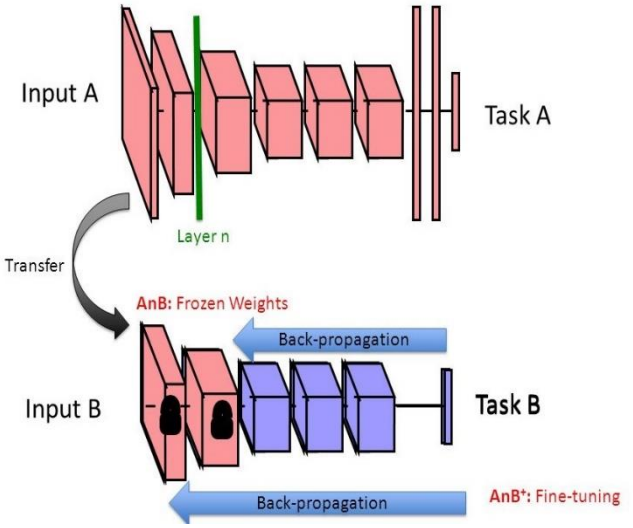

Fig 1: Transfer learning Architecture

A. *Zero shot tagging:*

Zhang.at.el [7], on their paper, proposed a method of zero shot tagging based on associating an image to a tag and principle direction on a word vector plane but the word vector plane and unseen tag sets used by them is limited. They assign tags to the image based on the principle direction of the image features and rank the words in that direction. This illustrates the following equation:

$$f(X_m) \cong W_m \qquad (1)$$

Where $f(X_m)$ is the principle direction, X_m is the set of visual features and W_m is the word along the principle direction ranked linearly. By approximating this function f, the zero-shot tagging is achieved. Zero shot taggers learn from the training classes but ultimately it tries to classify test images into unseen classes. This is made possible by the usage of word embedding, which ultimately helps in zero shot tagging. Our proposed method uses the glove vector model to provide the word embeddings and based on the tags assigned by the neural net to assign unseen tags to the images. So, the problem is reduced to training a neural network to tag the images and the word vector model to assign unseen tags to them.

B. *Word embedding*

Word embedding or word vectorization is a process of converting word into numbers/vectors for the usage in classification and clustering tags. The proposed method utilizes GloVe model to convert words into vectors. GloVe is an unsupervised algorithm, which is used to obtained word embedding for word. Figure 1. shows the GloVe vector based on co-occurrence of the words. This co-occurrence of the words is modelled by the co-occurrence matrix X, where each element X_{ij} calculates the number of times the word *i* appears in the context of the word *j*. The GloVe vector is generally modeled by the following equation 2:

$$F(w_i, w_j, w_k) = \frac{P_{ij}}{P_{jk}} \qquad (2)$$

where w_i, w_j, w_k are the word vectors and $P_{ij} = \frac{X_{ij}}{X_i}$ be the probability that word j appear in the context of word i.

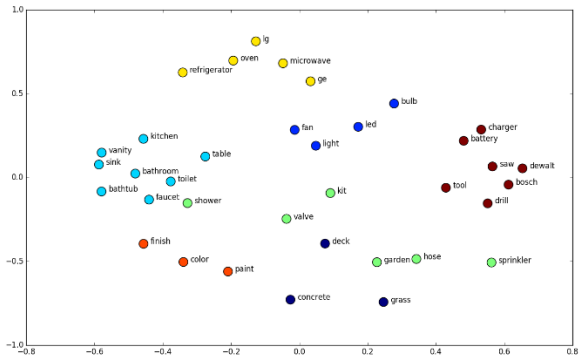

Fig 2: Representation of word embedding in vector space

III. PROPOSED METHODOLOGY

The Existing models lack the capacity to assign totally unseen tag sets to images. To overcome this problem, we propose to utilize "aa" word vector model in combination with a pre-trained CNN and a neural network to assign tags that are unseen by the neural network classifier.

A. *Architecture of the proposed system :*

The architecture of the proposed model is explained in figure3. The tagger consists of three parts namely, tagger net, object detection net and a GloVe model. The Tag Net does the initial tagging, after which object detection net and GloVe net appends the tags given by TagNet.

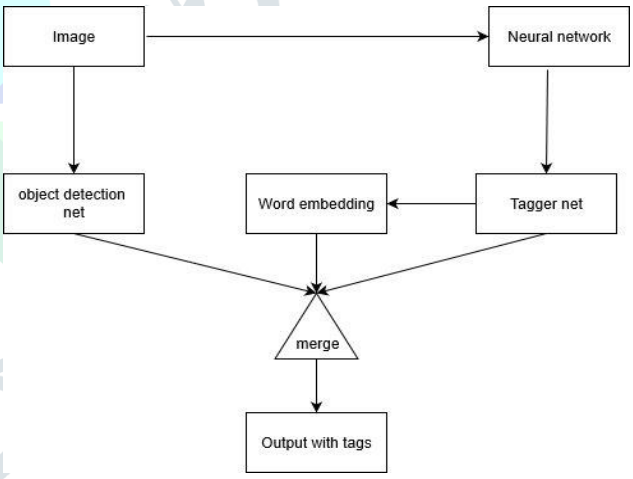

Fig 3: Architecture Diagram

The first component is the TaggerNet which is the collection of pre-trained convolutional neural network combined with a neural network to assign tags to images. The ResNet-50 is the pre-trained neural network that is used to extract features from the training and testing images. The extracted features are then fed into the neural network. The TaggerNet can be thought of a multi label multi-class classifier that tags images based on its training experiences. The main work of TaggerNet is to assign initial tags to a given image. The equation below illustrates the working of the TaggerNet.

$$f(V_i) = \{T_i\} \qquad (3)$$

The object detection net is a neural net that has been trained on the MSCOCO dataset which detects object in the given image. It removes duplicate tags if any assigned by the object detection neural net.

The GloVe model is used to generate word embedding which are then used to predict similar words. The GloVe model works on the output provided by TaggerNet. The tags returned by TaggerNet is converted into vectors using GloVe model. Then the GloVe model is compared with the tags returned the TaggerNet to find most similar tags. The cosine similarity is used to measure the comparison similarity. Thereby, the most similar tags are returned for each tag that is assigned by the TagNet. Thus, by combining the output returned by each component, we have a set of tags that can be assigned to each image based on its content.

Algorithm: For Zero Shot tagging

Initialize: For resnet-50 images with weights
Img= Resize (Img)
Img_array= im_to_array (Img)
Features=Resnet-50. predict(img_array)

Initialize: Tagger net with trained weights
Tags= TaggerNet .Predict (Features)
For tag in Tags:
do:
vector_tag=vectorize(tag)
similar_tag= most_similar(tag)
End for

Initialize: object-detection Net
Top_tags= object_detectionNet.predict(Img)
Top_tags=[ ]
Total_tag.append(Tags)
Total_tag.append(similar_tag)
Total_tag.append(Top_tags)
Return: Total_tag

B. Training

The learning process of proposed Tagger consists of 3 stages that depend on each other. The first stage is to train the TaggerNet, the second stage is to train the word embedding model and the final stage is to find out the neighboring tags in the word embedding model to the tags assigned by the TaggerNet.

In the architecture of the TaggerNet, we initialize ResNet-50 neural net for feature extraction. A neural net is created to learn the tags from the image features. The Adam optimizer is used by the neural net to attain the learning rate of 0.001.

$$W_t = W_{t-1} - \eta \frac{m_t}{\sqrt{v_t + \varepsilon}} \quad (4)$$

Based on the formula given in equation(4) the weights are changed by the Adams optimizer, where W_t is the weight of the current step, W_{t-1} is the weight of the previous step, where η is the size of the step, $m_t, and\ v_t$ are the corrected first and second moment values.

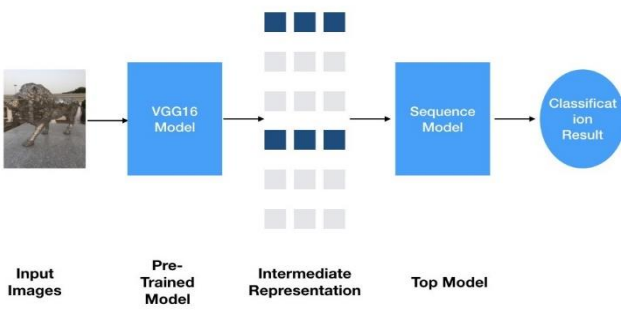

Fig 4: VGG-16 Convolutional neural network for classification

The second stage involves training a word embedded model. The model is trained on large corpus of data. The preprocessing involves in training the model includes, removal of stop words, lemmatization of the words. Then the model is trained for set number of iterations, with a fixed embedding size and context window size.

When the TaggerNet is trained the tags returned by its vectorized using the word embedding model to find two nearest neighbor words. The nearest word to a given tag is calculated by the cosine distance between the tag vector and the similar word vector. By, this way we extract the two most similar words to the given tags.

$$D = \frac{T.E}{\|T\| .\|E\|} \quad (5)$$

The above equation 5 specifies the cosine difference D where, T is the vector of tag returned by the TaggerNet and E is the vector of similar word in the vector space. The vector E will increase the value of D, which means the two vectors are more similar to each other. The given set of tags T_i we find set of tags W_i that consist of most similar words returned by the model utilizing the cosine distance as specified in equation 6.

$$\{ W_t \} = \text{Similar} (T_i) \quad (6)$$

IV. EXPERIMENTAL ANALYSIS

The dataset used to develop this project is the MIRFLICKR-25000 dataset which consists of 25000 images with their corresponding tags and ImageNet-50 dataset. The ImageNet-50 is an image subset selected from the large scale ImageNet data corpus covering many visual concepts includes vehicle, plant, animal, food etc.. Each category in ImageNet-50 contains more than 500 images.

There are 24 ground truth classes and 1386 tags in MIRFLICKR dataset. The tags have semantic overlapping i.e., an image of tree will be tagged as (tree, plant life).

The neural net is trained for 15 epochs with the 20000-image data with a train and validation split of 80 and 20 respectively. After the neural net is trained, the remaining 5000 images is used for testing.

Tag	# Images	Tag	# Images
sky	845	people	330
water	641	city/urban	308/247
portrait	623	sea	301
night	621	sun	290
nature	596	girl	262
sunset	585	snow	256
clouds	558	food	225
flower/flowers	510/351	bird	218
beach	407	sign	214
landscape	385	car	212
street	383	lake	199
dog	372	building	188
architecture	354	river	175
graffiti/streetart	335/184	baby	167
tree/trees	331/245	animal	164

Fig 5: Description of MLLICKR-25K dataset

V. RESULT

The averaged precision over all the classes is found out to be 0.81. we use cross entropy as the loss function which is combined with sigmoid function as the activation function. The figure 6 represents the variation of the loss and accuracy during each epoch.

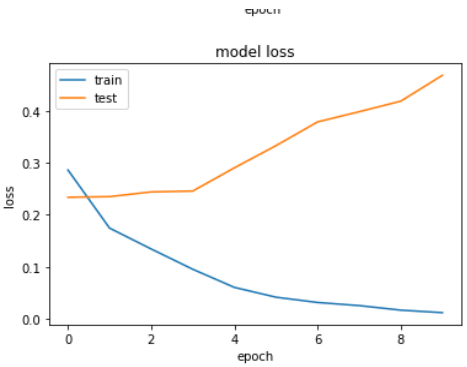

Fig 6. Loss Curve for MIRFLICKR-25k dataset

The multi-label classification model is evaluated by the *top-k* accuracy performance evaluator. The *top-k* accuracy takes into account the number of guesses that is taken by the model to correctly classify an image into set of classes. *top-k* accuracy is defined as the correct class in the *top-k* predicted probabilities by the model for it to count as correct.

The *top-k* validation Accuracy increases till five epochs, after which it starts reducing, indicating that the neural network is starting to over-fit. Therefore, the training is stopped at five epochs.

Therefore, by comparing the results obtained by the neural net augmented with GloVe and object detection model, it is found that the augmented model identifies more content the image compared to the TaggerNet.

VI. CONCLUSION

In this paper, the model is trained to perform image tagging based on multi-label and multi-class classification approach. The tagging is further improved by adding a GloVe model to the zero-shot tagging and object detection neural net which helps the tagger to tag the images on unseen tags also. By adding additional layers of word vectors to zero shot tagging, which helps in augmenting the tags to the content of the image effectively.

VII. REFERENCES

[1] Jialong Fu, Tao Mei Kuiyuan. Tagging Personal Photos With Transfer Deep Learning . In International world wide web committee, 2015 .

[2] Hanh T.H. Nguyen, Martin Wistuba. Personalized tag recommendation for images using deep transfer learning. Information systems and machine learning lab, University of Hieldeshiem, Germany, 2016.

[3] Jialong Fu, Yong Rui. Advances in dep learning approaches for image tagging, International conference in Cambridge University, 2017.

[4] Shafin Rahman. Deep multiple instance learning for zero shot image tagging. IEEE Transactions on multimedia, 2019.

[5] Dishant Mohite, Avinash Monde. Image tagging in classification, International journal of engineering research and technology, 2019.

[6] Jeffrey Pennington, Richard Socher, Christopher Manning. GloVe: Global vectors for word representation, Conference on Emperical methods in Natural Language Processing(EMNLP), Pages 1532-1543, 2014.

[7] Yang Zhang, Boqing Gong, Mubarak Shah. Fast zero shot image tagging in 2016 IEEE conference on computer vision and patter recognition(CVPR), Pages 5985-5994, IEEE 2016.

[8] Tesung Yi Lin, Michael Marie, Serge Belongie, James Hays, Deva Ramanan. Microsoft COCO: Common objects in context. In Europian conference on computer vision, 2014.

[9] Diederik P Kingama, Jimmy Ba Adam. The method stochastic optimization. arXiv preprint arXiv: 1412.6980, 2014.

[10] Anmesh Makida, Daldimer Pavlovic, Sanjiv kumar. Baselines for Image annotation. International journal of computer vision, 90(1): 88-105,2010.

[11] Ameesh Makadia, Dladimir Pavlovic, Snajiv Kumar. Baselines for Image annotation. International journal of computer vision, 90(1): 88-105,2010.

[12] Branson, S., Van Horn, G., Belongie, S., & Perona, P. (2014). Bird species categorization using pose normalized deep convolutional nets. arXiv preprint arXiv:1406.2952.

[13] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[14] Mark J Huiskes and Michael S Lew. The mir flickr retrieval evaluation. In Proceedings of the 1st ACM international conference on Multimedia information retrieval, pages 39–43. ACM, 2008.

[15] Xirong Li, Cees GM Snoek, and Marcel Worring. Learning tag relevance by neighbor voting for social image retrieval. In Proceedings of the 1st ACM international conference on Multimedia information retrieval, pages 180–187. ACM, 2008

[16] Akata Z, Malinowski M, Fritz M, Schiele B. Multi-cue zero-shot learning with strong supervision. InProceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016 (pp. 59-68).

[17] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS. (2013) 3111–311

[18] Li, Y., Song, Y., Luo, J.: Improving pairwise ranking for multilabel image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 3617–3625