IML-GCN: Improved Multi-Label Graph Convolutional Network for Efficient yet Precise Image Classification

Inder Pal Singh, Oyebade Oyedotun, Enjie Ghorbel, Djamila Aouada

University of Luxembourg *
Interdisciplinary Centre for Security, Reliability and Trust
29, Avenue J.F Kennedy
L-1855 Luxembourg
inder.singh@uni.lu, oyebade.oyedotun@uni.lu, enjie.ghorbel@uni.lu, djamila.aouada@uni.lu

Abstract

In this paper, we propose the Improved Multi-Label Graph Convolutional Network (IML-GCN) as a precise and efficient framework for multi-label image classification. Although previous approaches have shown great performance, they usually make use of very large architectures. To handle this, we propose to combine the small version of a newly introduced network called TResNet with an extended version of Multi-label Graph Convolution Networks (ML-GCN); therefore ensuring the learning of label correlation while reducing the size of the overall network. The proposed approach considers a novel image feature embedding instead of using word embeddings. In fact, the latter are learned from words and not images making them inadequate for the task of multi-label image classification. Experimental results show that our framework competes with the state-of-the-art on two multi-label image benchmarks in terms of both precision and memory requirements.

Introduction

Multi-label image classification is the task of predicting a set of labels corresponding to objects, attributes or other entities present in an image. It is an active research topic in computer vision mainly due to its numerous fields of application such as human attribute recognition (Li et al. 2016), scene recognition (Shao et al. 2015, 2016) and multi-object recognition (Kang et al. 2016; Bell et al. 2016). While classical image classification methods predict only one label per image, multi-label image classification aims at predicting a set of objects (or labels) present in a given image.

In the literature, multi-label prediction approaches can be classified into two main categories. The first class of methods generally learns a one-stream Deep Neural Network (DNN) for multiple binary classification tasks, without integrating any prior knowledge in the architecture design as in (He et al. 2016; Huang et al. 2017; Ridnik et al. 2021b). We refer to these approaches as *direct methods*. Although direct methods have been shown to achieve high performance as in (Ridnik et al. 2021b), they generally necessitate the use of multiple layers to work effectively. This leads to a high memory consumption; therefore restricting their applicability in a memory-constrained context. In contrast to the latter, the second category of approaches, that we call *indirect methods*, takes advantage of the prior knowledge related to the correlations existing among different objects present in an image (Chen et al. 2019b; Lanchantin et al. 2021; Zhu et al. 2017a; Wang et al. 2016). This is intuitive when one considers that in real-life some combinations of objects are more likely to appear together than others. For instance, it is extremely likely for a racket and person to appear together, than a racket and a dog. Indirect methods usually extend direct approaches by adding a subnetwork that models the dif-

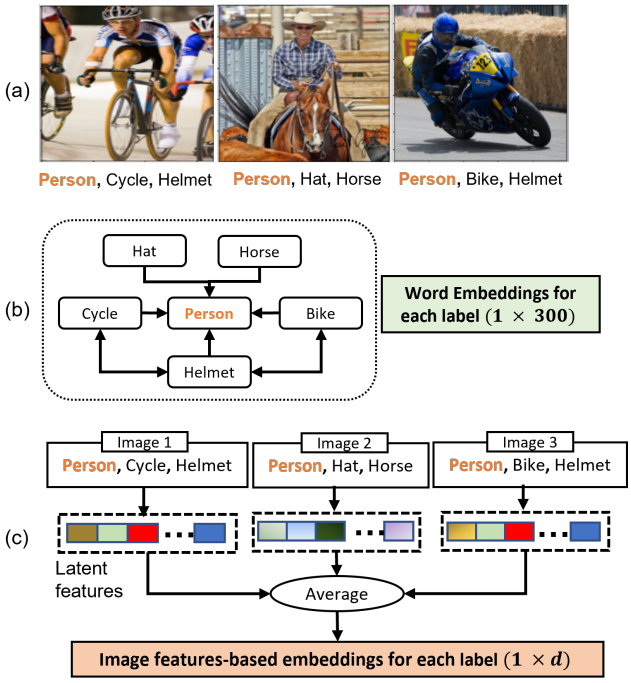

Figure 1: a) Image samples with multiple labels, b) representation of graph nodes using word embeddings and, c) our proposed embeddings using learned image-based latent representation.

*This work was funded by the Luxembourg National Research Fund (FNR), under the project reference BRIDGES2020/IS/14755859/MEET-A/Aouada.

ferent label relationships. Intuitively, one would think that using these data-driven approaches would allow obtaining a model with a reduced number of parameters. Nonetheless, it has been noted that most of these methods present a high number of parameters or reduce the memory requirements at the cost of a decrease in terms of precision. In this paper, our assumption is that finding the good combination between direct and indirect architectures will enable us achieving competitive performance, while reducing the size of the model.

For this reason, we design a new framework termed the *Improved Multi-label Graph Convolutional Network (IML-GCN)* that simultaneously considers the newly introduced direct approach called *TResNet* (Ridnik et al. 2021b) and an indirect model termed the *Image Feature Embeddings-based Graph Convolutional Network (IFE-GCN)*, which extends the graph subnetwork of the *Multi-label Graph Convolutional Network (ML-GCN)* introduced in (Chen et al. 2019b). TResNet is chosen given its impressive performance in terms of precision even when reducing the number of layers, while an improved version of ML-GCN is considered given its relatively low memory consumption. ML-GCN is one of the most popular works using graphs for modeling the label dependencies. Each label is represented by a node while the relationship between labels is modeled using weighted edges. Then, GLOVE word embeddings (Pennington, Socher, and Manning 2014) are used as node features. Unfortunately, this might lead to an inconstancy since the GCN is used to create binary classifiers that takes image features as input that are extracted from a second network. In fact, we recall that GLOVE has been initially designed to represent words with vectors in the field of Natural Language Processing (NLP), while visual object features are by nature different.

To overcome that, we propose to replace the word embeddings by novel image embeddings which are more meaningful in this problem of multi-label image classification, as illustrated in Figure 1. More specifically, our image embeddings are computed using label-wise image representations that are extracted by a state-of-the-art image feature extractor. Figure 2 shows an overview of the proposed framework. Its relevance in terms of precision and number of parameters with respect to the state-of-the-art is shown by performing experiments on two well-known datasets.

The organization of the remaining sections of this paper is as follows. **Background and Problem Statement** introduces the concept of GCN with an overview of the GCN-based approach ML-GCN (Chen et al. 2019b) followed by the problem formulation and motivation. **Proposed Approach** depicts the proposed framework of IML-GCN and details the methodology for generating the image-based embeddings. **Experiments** details the different experiments, and results discussion followed by an extensive study on model performance. The paper is finally concluded in **Conclusion** section, which summarizes the major findings in this work.

Background and Problem Statement

In this section, we review the concept of Graph Convolutional Networks (GCN), then present an overview of the GCN-based indirect method called ML-GCN (Chen et al. 2019b). Finally, we formulate the problem which leads to the motivation of our proposed approach.

Graph Convolution Networks (GCN): Graph convolution networks (GCN), initially introduced in (Kipf and Welling 2016), are the natural extension of Convolution Neural Networks (CNNs) to graphs. In fact, classical CNNs are designed for Euclidean structures and consequently applying them to graphs that are non-linear is not straightforward. Let us consider a graph $\mathcal{G} = (V, E, F^0)$ with $V = \{v_1, v_2, ..., v_n\}$ the set of nodes, n the number of nodes, $E = \{e_1, e_2, ..., e_m\}$ the set of edges connecting the nodes, m the number of edges and $F^0 \in \mathbb{R}^{n \times d}$ the input d-dimensional node features. Let A be the adjacency matrix defining the weighted connectivity of nodes.

Considering F^l the input features of the l^{th} layer, the aim of GCN is to learn a non-linear function $f(.)$ in order to update the node features of the next layer denoted as $F^{l+1} \in \mathbb{R}^{n \times d'}$ which can be written as,

$$F^{l+1} = f(F^l, A). \tag{1}$$

Using the same approach for convolution as (Kipf and Welling 2016), we can re-write Eq. 1 as:

$$F^{l+1} = h(\hat{A} F^l W^l), \tag{2}$$

where $W^l \in \mathbb{R}^{d \times d'}$ is the weight matrix to be learned and $\hat{A} \in \mathbb{R}^{n \times n}$ is the normalized version of A.

Multi-Label Graph Convolutional Networks (ML-GCN) ML-GCN (Chen et al. 2019b) were among the first to use graph Convolutional Networks in the context of multi-label image classification for modeling the label correlations. This architecture is composed of two main branches. The first branch consists of a classical image representation learning network. More precisely, the authors made use of ResNet-101 (He et al. 2016) to generate discriminative image features from the input image. On the other hand, the second subnetwork consisting in a GCN attempts to model the label correlations to generate C learned binary classifiers, with C the number of classes. In this context, each node of the graph represents a label. Then, the probability that two labels appear together in an image is used for encoding A. Hence, the aim of the GCN becomes to learn label features by aggregating neighbouring features. ML-GCN (Chen et al. 2019b) use word embedding representations as input node features denoted by F_W. These node features are generated by Glove (Pennington, Socher, and Manning 2014). Thus, in this case, we can say that $F^0 = F_W$. Furthermore, a re-weighted scheme is proposed where firstly a threshold τ has been used to filter the noisy edges resulting on:

$$A_{ij} = \begin{cases} 0, if P_{ij} < \tau, \\ 1, if P_{ij} \geq \tau \end{cases}, \tag{3}$$

where $P_{ij} = P(L_j|L_i)$ is the probability of the occurrence of an object label L_j in an image provided that the label L_i is already present. Secondly, in order to avoid over-smoothing, the following re-weighted scheme is used (Chen

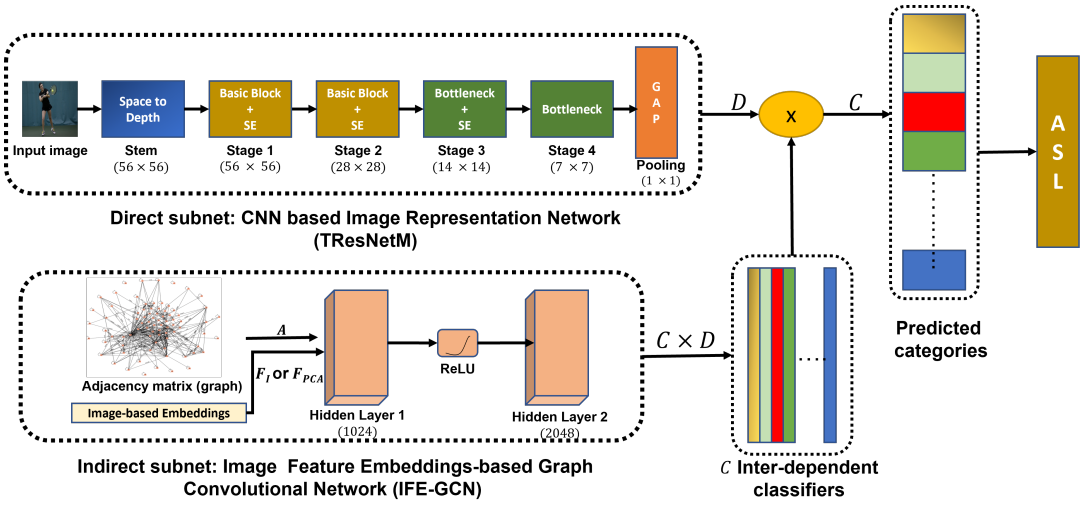

Figure 2: Architecture of our IML-GCN approach for multi-label image classification.

et al. 2019b):

$$A'_{ij} = \begin{cases} p / \sum_{j=1}^{n} A_{ij}, if i \neq j, \\ 1 - p, if i = j \end{cases}, \quad (4)$$

where p determines the weights assigned to a node itself and other correlated nodes. This means that when $p \to 1$, the features of the node itself are not considered. Inversely, when $p \to 0$, neighbouring information tends to be ignored.
This architecture aims to learn a set of inter-dependent object classifiers which are applied to the image representations extracted from a ResNet-101. This allows to determine whether the associated object is present in the image or not.

Problem statement

Despite the performance of ML-GCN, two main limitations can be noted. First, the backbone network for image representation is very deep (ResNet-101) and therefore naturally induces a heavy architecture leading to high memory computation and consumption. Second, it uses word embeddings generated by Glove (Pennington, Socher, and Manning 2014) to represent each node (label). Unfortunately, they might not be optimal in the context of image classification. In fact, these embeddings have been generated for representing words using unique vectors in the field of NLP, while the latter are used in ML-GCN for generating binary classifiers that take image features as input. Given the difference in nature of the two modalities (words and images), it seems not suitable to consider word embeddings as node features. Based on these two observations, two interesting ideas are driving this work. (1) It would be interesting to investigate if finding an appropriate combination between direct and indirect networks could be a way to achieve state-of-the-art results, while reducing the size of the network. (2) Replacing word embeddings with meaningful image embeddings could help improving the results without an increase of the parameter number.

Proposed Approach

In this section, we depict our framework called IML-GCN for multi-label image classification. The two subnetworks of our framework are detailed below.

Direct subnetwork: TResNet

Our first subnetwork, as shown in Figure 2, is a CNN-based image-representation network whose aim is to extract image features by using a set of convolutional blocks. More specifically, we use TResNetM, a smaller version of TResNet (Ridnik et al. 2021b), which was designed to boost neural networks accuracy while retaining their GPU training and inference efficiency. This is in line with our objective of reducing the size of the final network. It has already demonstrated state-of-the-art results on single and multi-label datasets (Ridnik et al. 2021a) while maintaining a balanced trade-off between speed and accuracy. However, it has been noted that the precision drops considerably when employing the small version TResNetM, compared to TResNetL. In general, the refinements on top of plain ResNet architecture include: SpaceToDepth Stem (Sandler et al. 2019), Anti-Alias Downsampling (Lee et al. 2020), In-Place Activated BatchNorm (Rota Bulò, Porzi, and Kontschieder 2018), Novel Block-Type Selection (Ridnik et al. 2021b) and Optimized SE Layers (Hu, Shen, and Sun 2018).

For any given input image I, the output of this subnetwork is a d-dimensional latent representation of the image which is denoted by $F_{GAP} \in \mathbb{R}^{1 \times d}$. For TResNetM specifically, the output dimension $d = 2048$.

Indirect subnetwork: Image Feature Embeddings-based Graph Convolutional Network (IFE-GCN)

The second subnetwork consists of an improved version of the graph subnetwork introduced in the indirect method ML-GCN (Chen et al. 2019b). The overall architecture remains

the same as the GCN branch of the original ML-GCN, except that we replace the input word embedding based node representation by our proposed image-feature based embeddings.

As we can see in Figure 2, the output of the proposed GCN network creates C interdependent binary classifiers incorporating the information of label correlations. However, as stated earlier, word embeddings are not adapted for multi-label image classification. Thus, the idea would be to replace these word embeddings by relevant image embeddings that could be sufficiently discriminative to design effective classifiers. Intuitively, the idea would be to generate a vector per object label including relevant image features related to the corresponding object. Below, we depict in details how these novel image embeddings are computed.

Image feature embeddings: Assuming N is the total number of training samples in a particular dataset, we initialize the CNN model, i.e. TResNetM, using the weights pre-trained on the ImageNet dataset. We first train the CNN model to convergence. Once we obtain the fully-trained weights, we make one forward pass for the N images. More specifically, the output of the penultimate layer (GAP) $F_{GAP} \in \mathbb{R}^{N \times d}$, provides d dimensional vector as learned image-level features for each input image.

Then, using the ground-truth, we gather for each label the set of generated features S_i such that the associated object is visible in the corresponding image. Note that $i \in \{1, ..., n\}$ and n is the total number of nodes or object labels. Finally, for each label i, we compute the average of the corresponding set of features F_I given as:

$$(F_i)_I = mean(S_i) \tag{5}$$

with $F_I = [(F_1)_I, (F_2)_I, ..., (F_n)_I] \in \mathbb{R}^{n \times d}$.

Furthermore, since we employ the image-feature embeddings as inputs to the GCN, improving the signal-to-noise of the input can facilitate the learning of robust representations by the GCN. Therefore, we use Principal Component Analysis (PCA), which simultaneously reduces the dimension of the image-feature embeddings from d to n such that the new input feature matrix $F_{PCA} \in \mathbb{R}^{n \times n}$. Thus, these features are used as input to the first layer such as,

$$F_I = F_{PCA} \tag{6}$$

Experiments

In this section, we start by presenting the implementation details. Subsequently, we present the results and discussion on two benchmarking multi-label image recognition datasets, which include the MS-COCO (Bell et al. 2016) and VG-500 (Krishna et al. 2017).

Implementation details:

The Asymmetric Loss (ASL) (Ridnik et al. 2021a) is used as our loss function. The adjacency matrix for the GCN is computed using the same approach depicted in ML-GCN.

The hyper-parameters are empirically fixed. More specifically, we set the threshold to $\tau = 0.1$ in Eq. 3. We train the model for 40 epochs using a multi-step learning rate scheduler initialized with a learning rate of 10^{-3} and decayed by a factor of 0.1 at the 10, 20, and 30th epochs. For data augmentation, we use the same Randaugment technique as the baseline (Ridnik et al. 2021a) during the training. Adam is used as optimizer (Kingma and Ba 2015) with a weight decay of $5e^{-4}$.

Experimental results:

In this part, we start by comparing our approach to state-of-the-art methods using MS-COCO and VG-500 datasets. Subsequently, we conduct an ablation study to evaluate the interest of the proposed contributions.

Performance on MS-COCO: The MS-COCO (Bell et al. 2016) dataset is a well-known large-scale multi-label image dataset. It contains 122,218 images and covers 80 common objects. Following the conventional training and evaluation protocols for the MS-COCO dataset (Wang et al. 2020; Ge, Yang, and Yu 2018), we report the following statistics: mean Average Precision (mAP), average per-Class Precision (CP), average per-Class Recall (CR), average per-Class F1-score (CF1), the average Overall Precision (OP), average overall recall (OR) and average Overall F1-score (OF1). We report the results obtained for our approach using two types of settings; that is, **(IML-GCN with F_I)** and **(IML-GCN with F_{PCA})** using image embeddings without and with PCA respectively.

Table 1 reports the quantitative results obtained on the MS-COCO dataset. It can be clearly seen that although our models are noticeably smaller than others, they outperform state-of-the-art methods in terms of the mAP. Specifically, our approach achieves an mAP of 86.62% using only 31.5M parameters. Thus, it outperforms ML-GCN by 3.62% and requires around 30% less parameters. Similarly, our approach slightly registers higher mAP than ASL with 0.2% of increase, while requesting 42% less parameters. Also, it is observed that the model using the IML-GCN with PCA performs better than the model without PCA in terms of mAP. We can see an improvement of 0.8%. Moreover, it can be noted than 2M less parameters are needed when using PCA. Therefore, the obtained results show the interest of applying PCA to the image feature embeddings.

Performance on VG-500: The Visual Genome dataset (Krishna et al. 2017) is another large-scale multi-label image dataset that contains a total of 108,077 images, which covers over thousands of categories. Given that the distribution of the labels is quite sparse, the VG-500 subset (Chen et al. 2020) that consists of 500 most frequent objects as categories is used. It is divided into a training set of 98,249 training images and 10,000 test images.

In Table 2, we compare our model with recent approaches. It can be seen that we achieve an mAP of 34.5% which is higher than the score reported for ResNet-101 (He et al. 2016), ML-GCN (Chen et al. 2019b) and ASL (TResNetM) (Ridnik et al. 2021a). We also note that ResNet101 and ML-GCN employed a larger backbone CNN network, ResNet-101 leading to a higher number of parameters. Only ASL uses fewer number of parameters, which is fair since this network represents the direct backbone of our model. It can be noted that **C-Tran** (Lanchantin et al. 2021) is the

Table 1: Comparisons with state-of-the-art methods on the MS-COCO dataset with n_components=80 the number of the components fixed for computing F_{PCA}.

Method	#Parameters	mAP	CP	CR	CF1	OP	OR	OF1
CNN-RNN (Wang et al. 2016)	66.2 M	61.2	-	-	-	-	-	-
SRN (Zhu et al. 2017b)	∼48M	77.1	81.6	65.4	71.2	82.7	69.9	75.8
ResNet101 (He et al. 2016)	44.5M	77.3	80.2	66.7	72.8	83.9	70.8	76.8
Multi-Evidence (Ge, Yang, and Yu 2018)	∼47M	-	80.4	70.2	74.9	85.2	72.5	78.4
ML-GCN (Chen et al. 2019b)	44.9M	83	85.1	72	78	85.8	75.4	80.3
SSGRL (Chen et al. 2019a)	92.2M	83.8	**89.9**	68.5	76.8	**91.3**	70.8	79.7
KGGR (Chen et al. 2020)	∼45M	84.3	85.6	72.7	78.6	87.1	75.6	80.9
C-Tran (Lanchantin et al. 2021)	120M	85.1	86.3	74.3	79.9	87.7	76.5	81.7
ASL (TResNetM) (Ridnik et al. 2021a)	**29.5M**	81.8	82.1	72.6	76.4	83.1	76.1	79.4
ASL (TResNetL) (Ridnik et al. 2021a)	53.8M	86.6	87.4	76.4	**81.4**	88.1	79.2	81.8
Ours (IML-GCN with F_I)	33.5M	85.9	82.7	78.9	80.5	84.6	82.1	**83.3**
Ours (IML-GCN with F_{PCA})	31.5M	**86.6**	78.8	**82.6**	80.2	79.0	**85.1**	81.9

Table 2: Comparisons with state-of-the-art methods on the VG-500 dataset with n_components=500 for F_{PCA}.

Method	# Parameters	mAP (%)
ResNet-101 (He et al. 2016)	44.5M	30.9
ML-GCN (Chen et al. 2019b)	44.9M	32.6
ASL (TResNetM) (Ridnik et al. 2021a)	**29.5M**	33.6
C-Tran (Lanchantin et al. 2021)*	120M	38.4*
Ours (IML-GCN with F_I)	33.5M	**34.0**
Ours (IML-GCN with F_{PCA})	32.1M	**34.5**

*The model is roughly 273% larger than our proposal

only approach that outperforms our method in terms of mAP. However, they rely on extremely large models. Indeed, the mAP result of 38.4% obtained in (Lanchantin et al. 2021) used a model which is roughly 273% larger than the model that we propose in this paper, as it can deduced from Table 2. The extremely large size of the model in (Lanchantin et al. 2021) places a limitation on its practical usefulness when considering the high computational resource and latency it incurs. Importantly, our proposed model that requires modest computational resources and gives interesting results.

Impact of GCN input features:

This section reports the results of experiments, which were performed to study the performance improvements obtained using the proposed image-feature embeddings as input features for the GCN in comparison to word embeddings. For these experiments, the proposed framework (CNN-GCN architecture) remains the same except that the GCN of IML-GCN is replaced with the graph subnetwork of ML-GCN. We report the performance improvements for three different settings in Table 3.

Word embeddings: We use the same GCN subnetwork proposed for ML-GCN (Chen et al. 2019b). Table 3 shows that using the original GCN which incorporates word embeddings as node features lead to a visible decrease of 5% and 1.9% in mAP on MS-COCO and VG-500, respectively. This is expected as the used word embeddings are not relevant to the task of image classification and confirms our assumption.

Table 3: Impact of GCN input features.

Dataset	Refinements	mAP (%)
COCO	Word embeddings (F_W)	81.6
	(+) Image based-feature embeddings (F_I)	85.9 (**+4.3**)
	(+) Image based-feature embeddings PCA (F_{PCA})	86.62 (**+0.7**)
VG-500	Word embeddings (F_W) ML-GCN (Chen et al. 2019b)	32.6
	(+) Image based-feature embeddings (F_I)	33.39 (**+0.8**)
	(+) Image based-feature embeddings PCA (F_{PCA})	34.47 (**+1.1**)

Image embeddings: When the word embeddings are replaced by the d-dimensional embeddings generated using latent image-representations, there is a significant improvement in the accuracy for the two benchmarks as reported in Table 3. This shows that the proposed image-feature embeddings can provide more robust representations in comparison to word embeddings.

Image based-feature embeddings PCA: As discussed earlier, PCA is applied to the generated d-dimensional embeddings to obtain C-dimensional feature embeddings with improved signal-to-noise ratio. The results given in Table 3 shows that applying PCA to the image embeddings improves the performance of the proposed model by 0.7% and 1.1% on MS-COCO and VG-500, respectively.

Conclusion

Multi-label image classification problems can be tackled using CNN-GCN frameworks, where the GCN employs word embeddings as input features. However, word embeddings schemes might not be optimal for allowing the GCN to learn robust representations that encode label dependencies; word embeddings are more suited for NLP tasks. Furthermore, existing models, including CNN-GCN are considerably large, and thus their practical usefulness is limited in applications that require low latency and/or memory. As such, this paper proposes a new framework called IML-GCN that achieves high precision while reducing the size of the network. It takes advantage of the latest advancements in direct (TResNet) and indirect methods (ML-GCN). Moreover, instead of employing word embeddings, we use image-feature embeddings, which are more adapted in an image classification context. We show that better classification results can be obtained compared to previous methods including CNN-GCN based approaches, while reducing the number of parameters.

References

Bell, S.; Zitnick, C. L.; Bala, K.; and Girshick, R. 2016. Inside-Outside Net: Detecting Objects in Context With Skip Pooling and Recurrent Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, T.; Lin, L.; Hui, X.; Chen, R.; and Wu, H. 2020. Knowledge-Guided Multi-Label Few-Shot Learning for General Image Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

Chen, T.; Xu, M.; Hui, X.; Wu, H.; and Lin, L. 2019a. Learning Semantic-Specific Graph Representation for Multi-Label Image Recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 522–531.

Chen, Z.-M.; Wei, X.-S.; Wang, P.; and Guo, Y. 2019b. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5177–5186.

Ge, W.; Yang, S.; and Yu, Y. 2018. Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1277–1286.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-Excitation Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7132–7141.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Kang, K.; Ouyang, W.; Li, H.; and Wang, X. 2016. Object Detection from Video Tubelets with Convolutional Neural Networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 1–15.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D.; Bernstein, M.; and Li, F.-F. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision*, 123.

Lanchantin, J.; Wang, T.; Ordonez, V.; and Qi, Y. 2021. General Multi-label Image Classification with Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16478–16488.

Lee, J.; Won, T.; Lee, T. K.; Lee, H.; Gu, G.; and Hong, K. 2020. Compounding the Performance Improvements of Assembled Techniques in a Convolutional Neural Network. arXiv:2001.06268.

Li, Y.; Huang, C.; Loy, C. C.; and Tang, X. 2016. Human Attribute Recognition by Deep Hierarchical Contexts. volume 9910, 684–700. ISBN 978-3-319-46465-7.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Ridnik, T.; Ben-Baruch, E.; Zamir, N.; Noy, A.; Friedman, I.; Protter, M.; and Zelnik-Manor, L. 2021a. Asymmetric Loss for Multi-Label Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 82–91.

Ridnik, T.; Lawen, H.; Noy, A.; Ben Baruch, E.; Sharir, G.; and Friedman, I. 2021b. TResNet: High Performance GPU-Dedicated Architecture. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 1400–1409.

Rota Bulò, S.; Porzi, L.; and Kontschieder, P. 2018. In-place Activated BatchNorm for Memory-Optimized Training of DNNs. 5639–5647.

Sandler, M.; Baccash, J.; Zhmoginov, A.; and Howard, A. 2019. Non-Discriminative Data or Weak Model? On the Relative Importance of Data and Model Resolution. 1036–1044.

Shao, J.; Kang, K.; Loy, C. C.; and Wang, X. 2015. Deeply learned attributes for crowded scene understanding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4657–4666.

Shao, J.; Loy, C. C.; Kang, K.; and Wang, X. 2016. Slicing Convolutional Neural Network for Crowd Video Understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5620–5628.

Wang, J.; Yang, Y.; Mao, J.; Huang, Z.; Huang, C.; and Xu, W. 2016. CNN-RNN: A Unified Framework for Multi-label Image Classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2285–2294.

Wang, Y.; He, D.; Li, F.; Long, X.; Zhou, Z.; Ma, J.; and Wen, S. 2020. Multi-Label Classification with Label Graph Superimposing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34: 12265–12272.

Zhu, F.; Li, H.; Ouyang, W.; Yu, N.; and Wang, X. 2017a. Learning spatial regularization with image-level supervisions for multi-label image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5513–5522.

Zhu, F.; Li, H.; Ouyang, W.; Yu, N.; and Wang, X. 2017b. Learning Spatial Regularization with Image-Level Supervisions for Multi-label Image Classification. 2027–2036.