A Novel Transformer-based Two Stage Framework for Multi-Label Image Classification

Zhi-Jie Wang
College of Computer Science
Chongqing University
Chongqing, China
Email: cszjwang@cqu.edu.cn

Nai-Kang Zhong
Department of Computer Science
Shanghai Normal University
Shanghai, China
Email: naikangzhong@stu.snu.edu.cn

Xiao Lin
Department of Computer Science
Shanghai Normal University
Shanghai, China
Email: linxiao6008@snu.edu.cn

***Abstract*—Multi-label image classification (MLIC) has received much attention due to its wide applications. Recently, the Transformer-based methods have exhibited excellent performance in handing MLIC tasks. Existing Transformer-based methods either used standard encoders that employ traditional relative positional encoding, or discarded encoders by focusing on the decoders to establish relations between labels and the regions of interest (RoI) in images. Yet, researchers pay less attention on enhancing the encoders. Moreover, the self-attention layer in the decoder is often employed in existing works, and is considered a component that could potentially enhance the internal relationships of label embeddings. Yet, it is unclear whether the self-attention layer is really helpful in establishing label correlation. Last but not least, existing Transformer-based methods use Asymmetric Loss to alleviate the problem of positive and negative sample imbalance; however, Asymmetric Loss may lead to the exclusion of some negative samples that are actually helpful for model training. To address these issues, this paper presents a novel Transformer-based two stage framework, which can be viewed as a fusion of the RoI based technique and an adapted Transformer. Our framework captures global and local features in model training. It is simple and easy-to-implement, but it can achieve excellent performance. We have conducted extensive experiments on several widely used multi-label image classification datasets. The results consistently show us that our proposed framework is feasible and also competitive, compared against state-of-the-art models.**

***Index Terms*—Transformer, multi-label image classification, deep learning.**

I. Introduction

Image classification [1, 2, 3, 4, 5] is to analyze input images and categorize them into predefined classes based on their content. Specifically, the objective of image classification is to assign one or more category labels to an image according to the objects, scenes, or other features. Image classification can be viewed as a branch of data classification, it has been extensively studied in data mining [6, 7, 8] and computer vision [9, 10, 11, 12] community. With the development of deep learning, image classification tasks have achieved significant advancements [13, 14, 15, 16, 17] and are now widely used in various real-world domains, such as image retrieval [18], e-commerce [19], and social media [20].

Image classification contains two types of cases: single-label image classification (SLIC) [21, 22] and multi-label image classification (MLIC) [23, 24, 25, 26, 22, 27]. Single-label image classification refers to the task of assigning an image to a single category or label. In this type of task, the image is assigned to only one most relevant label. For instance, categorizing an image as a single class like a cat, dog, or airplane. In contrast, multi-label image classification often assigns an image to multiple labels or categories. It is more challenging than SLIC due to various reasons (e.g., a greater variety of objects within the images).

Early methods for multi-label image classification often transform it into single-label problem, and then use region of interest (RoI) based techniques [28, 29]. Owing to the rapid advancement of convolutional neural networks (CNNs) and their excellent feature extraction capabilities, CNNs often utilizes convolutional layers to extract features from input images, and the resultant feature maps are utilized to predict and determine the presence of specific target categories. A group of researchers has delved into the investigation of attention mechanisms [30, 25]. Methods combining attention mechanisms with convolutional neural networks (CNNs) can further focus on more important local information. However, they often incur issues such as incomplete feature representation, insufficient utilization of inter-label correlations and dependencies, and relatively poor generalization ability. Later, with the rise of graph convolutional networks (GCNs), another group of researchers has focused on developing graph convolution based methods for multi-label image classification [24, 3]. They can establish correlations and dependencies between labels, and exhibit strong generalization capabilities. However, these methods are computationally intensive, requiring substantial computational resources. Additionally, they carry a risk of overfitting in cases of unreasonable graph structures or significant noise.

Recently, Transformer-based two-stage framework has attracted much attention [30, 5, 31, 27, 32]. These approaches effectively integrate spatial information with semantic information. Particularly, it is much more flexible in modeling the relationship between image features and labels. These characters are beneficial to enhance the model's classification performance and generalization capabilities. Yet, there remains several issues: (i) We observe that existing works either used standard encoders that employ traditional relative positional encoding, or focused on leveraging the decoder to establish

relations between labels and the RoI, i.e., depriving encoders directly. As for the former, traditional relative positional encoding is originally designed for one dimensional sequence data, which might be improper for image data. As for the latter, the utility and effectiveness of the Transformer encoder is ignored, which might harm the performance. (ii) The self-attention layer in the decoder is often employed in existing works, and is considered a component that could potentially enhance the internal relationships of label embeddings. We realize that, in multi-label image classification tasks, the input to the decoder consists of learnable label embeddings, where the labels themselves are not inherently correlated, since they are independent texts. If one conducts self-attention calculations on these learnable label embeddings, it actually forces the labels to learn relationships with each other; this may lead to spurious/fake label correlations. In other words, it is unclear whether the self-attention layer is really helpful in establishing label correlation. (iii) Existing Transformer-based methods use Asymmetric Loss to alleviate the problem of positive and negative sample imbalance; however, Asymmetric Loss mainly excludes easy negative samples through a hard threshold mechanism, which may lead to the exclusion of some negative samples that are actually helpful for model training.

To alleviate the above issues, we propose a new Transformer-based two stage framework for multi-label image classification. Our model consists of four major components: (i) Feature extracting, which integrates data augmentation and CNNs together to extract local image features. (ii) Feature reshaping, which utilizes a 1×1 convolution and learned positional encoding together to perform dimension reduction and features serialization. (iii) Feature label correlation, which uses an enhanced Transformer structure to learn global feature information, and performs cross-attention computations between learnable label embeddings and the globally extracted image features. (iv) Final classification and Loss, which obtains the classification score based on a linear projection layer and sigmoid function. To summarize, our main contributions can be listed as follows:

- We proposed a novel Transformer-based two stage framework for multi-label image classification. To our knowledge, for MLIC tasks, it is the first attempt to introduce image relative positional encoding (IRPE) into the Transformer encoder. Our method is simple, easy-to-implement but without loss of efficiency.
- We conducted extensive experiments on several widely-used datasets including MS-COCO, PASCAL VOC, and NUS-WIDE. The experimental results consistently show us that the proposed model can achieve competitive performance, compared against a set of state-of-the-art models.

II. Related Work

In recent years, a lot of methods have been proposed for multi-label image classification. One type of approach is based on Region of Interest, another type of approach is based on Transformer. Our method combines these types of techniques, and thus is highly related to them. In what follows, we review previous works most relevant to ours.

A. Regions of Interest

It is crucial to locate regions of interest (RoI) in computer vision tasks [28, 29, 27]. Early methods for multi-label image classification often transform it into single-label problem, and then use region based techniques [28, 29]. For example, Wei *et al.* [28] proposed a flexible deep CNN infrastructure to produce the multi-label predictions. Afterwards, many powerful region-based methods were developed [33, 25, 27]. For example, Wang *et al.* [33] suggested to locate attentional regions by using a spatial transform layer. Recently, the work [30] defined spatial class-aware attention, which is used to reinforce the correct causal relationships. The afore-mentioned methods can efficiently focus on important local information, they however often suffer from incomplete feature representation and relatively weak ability to extract global features. To address this issue, researchers developed methods considering global and local features together [26, 31]. For example, Gao *et al.* [26] proposed a global-to-local feature mechanism to find proper regions. Our method also considers both global and local feature information, but is different from theirs. In brief, our method enhances information interconnections among different regions, by introducing a two-dimensional relative positional encoding into Transformer encoders.

B. Transformer

The Transformer [34] was initially been applied to natural language processing tasks. Recently, the Transformer architecture has been applied in the field of computer vision [35, 36]. Particularly, there are some studies [30, 5, 31, 27, 32] that have employed Transformers for multi-label image classification. In the literature, early works [30, 5, 31] mainly utilized Transformer encoders to extract global image features. For example, Zhao *et al.* [5] used a Transformer encoder to capture contextual information, obtaining global information from feature maps of different scales. Nevertheless, these works often use standard Transformer encoders that employ traditional relative position encoding or do not employ relative position encoding at all. Unlike these works, our paper introduces the Image Relative Position Encoding (IRPE), which is specifically designed for 2D images, into the Transformer encoder. This idea addresses the shortcomings of traditional relative position encoding which is originally designed for language sequences.

Recently, some studies [27, 32] have focused on using Transformer decoders to establish relationships between labels and image features. For example, Ridnik *et al.* [32] redesigned the decoder architecture and introduced a novel grouped decoding approach. Liu *et al.* [27] used the cross-attention mechanism built in the Transformer decoder to establish relationships between labels and image features. Among these works, our work is most similar to [27], as our method bears many similarities with it (e.g., utilizing a Transformer decoder

to query the presence of category label). However, these methods often overlook the importance of the Transformer encoder. Instead, we claim that the Transformer encoder is also useful, as it plays a crucial role in global information extraction and spatial structure modeling. Compared to [27], there are several obvious differences: (i) We use the Transformer encoders to help the model acquire global image feature information, and we further employ IRPE to enhance the performance; (ii) We analyze the self-attention layers in the decoder, and point out that they are often unhelpful for multi-label image classification tasks; and (iii) we introduced the Asymmetric Polynomial Loss which is not covered in their paper.

C. Others

Previous works considered also label correlations [9, 24, 3, 37]. For example, Wang *et al.* [9] integrated RNNs with CNNs to capture semantic label dependencies and predicted labels in a predefined sequence. Chen *et al.* [24] utilized graph convolutional networks (GCNs) to construct directed graphs on labels. Ye *et al.* [3] proposed an attention-driven dynamic graph convolutional network. These methods establish correlations and dependencies between labels. They, however, often rely on graph structures that require substantial computational resources, and sometimes they may get false correlations, harming the performance [27, 22]. In contrast, our method does not require the construction of complex graph models, it instead directly detects image regions relevant to categories.

Loss functions are also obviously important to classification quality. The binary cross-entropy loss can be used in multi-label image classification, but such a loss function often fails to address class imbalance issue [38, 39], which is crucial for the model's performance. To this, many researchers [40, 4, 12] have focused on improving the loss functions. For example, Lin *et al.* [40] proposed focal loss. Ridnik *et al.* introduced the asymmetric loss [4]. Huang *et al.* [12] proposed the Asymmetric Polynomial Loss, which leverages asymmetric attention mechanisms to alleviate the imbalance problem of positive-negative samples. Inspired by its excellent performance, our work adopts the Asymmetric Polynomial Loss.

III. Method

In this section, we first present an overview of our proposed model, and then cover the details of the model.

A. Overview of Our Model

Fig. 1 illustrates the architecture of our model. The first step is feature extraction. Specifically, for a given input image, we first apply data augmentation and then feed it into a convolutional-based model to extract preliminary image features. The second stage is feature reshaping. Generally speaking, for the preliminary image feature, we first conduct the dimensionality reduction by using a 1x1 convolution, and then combine it with the learned positional embedding, by executing a "+" operator between feature vectors. Afterwards, we obtain the serialized feature information by "flattening" the combined feature vectors. Later, the serialized feature information is to be passed into the next stage. The third stage is feature label interaction. The essence of this step is to extract global image features and to locate regions of interest according to specific categories. More specifically, in this stage, we develop a variant of the standard Transformer structure. In our structure, we introduce IRPE into the encoder, which is beneficial to capture richer global features. Meanwhile, in the decoder part, we, based on empirical study, remove the self-attention layer that was used in prior studies. The final stage is to obtain the multi-label prediction scores. Specifically, the feature vectors from the decoder are to be transformed into logits through a linear projection layer, the multi-label prediction scores can be obtained by the sigmoid function.

B. Feature Extracting and Reshaping

To better understand this stage, we first introduce the concepts of Cutout and learned positional encoding.

1) Cutout: Cutout is a data augmentation technique involving the following two steps: (i) Random Region Selection. Randomly select a rectangular region within the image. (ii) Masking the Region. Set the pixel values in the selected rectangular region to zero or apply random noise. This masks the chosen region within the image. The resulting image with the masked region serves as a new training sample, contributing to increasing data diversity and model robustness, ultimately improving generalization.

2) Learned positional encoding: The learned positional encoding is actually a technique for modeling positional information in deep learning models. Unlike traditional absolute positional encodings based on sine and cosine functions, the learned positional encoding does not rely on a predefined mapping from positions to vectors, it can be implemented with a simple embedding layer. During model training, the weights of these embedding layers are adjusted to minimize task-specific loss functions, enabling the model to learn optimal positional representations. In other words, it learns the vector representation for each position during training, providing the model with more flexible and adaptive positional information.

With the above concepts in mind, we proceed to explain the details of "Feature Extracting and Reshaping". Given an image $x \in \mathbb{R}^{H\times W\times 3}$ as an input, we first implement the Cutout, and then extract local spatial feature $F \in \mathbb{R}^{H'\times W'\times d_0}$ using CNN-based backbone models (e.g., ResNet, TresNet, ConvNext), where H and W respectively denote the height and weights of the original image, H' and W' respectively denote the height and weights of the feature map F, and d_0 represents the dimensionality of feature map F. Subsequently, we need to reshape the features to match the input expected by the Transformer. Specifically, we divide the reshaping into the following two steps:

(i) Dimension Reduction to Hidden Dimension: We perform dimension reduction using a 1x1 convolution to transform the features from $H'\times W'\times d_0$ to $H'\times W'\times d_{model}$, where d_{model}

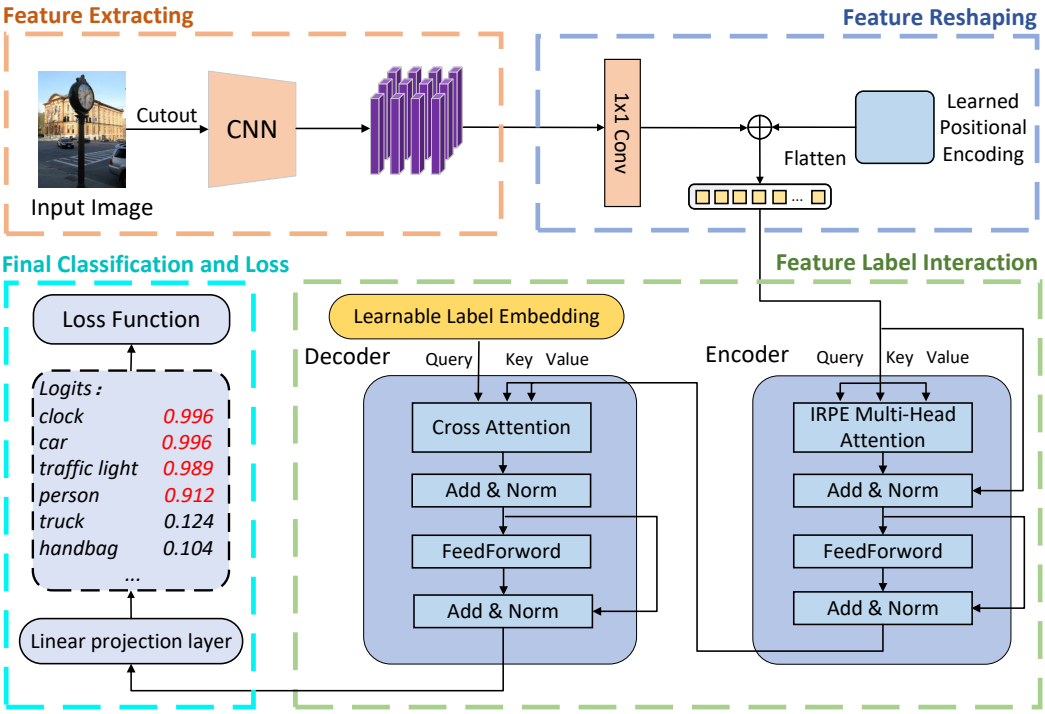

Fig. 1: The architecture of our model.

is the desired hidden dimension. This can be implemented with the following formula:

$$Y = \text{Conv}_{1\times1}(F), \quad Y \in \mathbb{R}^{H'\times W'\times d_{model}} \tag{1}$$

where, $\text{Conv}_{1\times1}$ denotes the 1x1 convolution operation, converting d_0 channel feature maps into d_{model} channel feature maps.

(ii) Flattening into Sequence Data: The reduced features Y are reshaped from a 3D tensor ($H' \times W' \times d_{model}$) into a 2D tensor representing a sequence of data, with dimensions $d_{model} \times L$, where $L = H' \times W'$ is the sequence length. This can be implemented with the following formula:

$$Z = \text{Reshape}(Y), \quad Z \in \mathbb{R}^{d_{model}\times L} \tag{2}$$

where Z represents the reshaped sequence features, Reshape involves the Flatten operation followed by transposition of certain dimensions.

C. Feature-Label Interaction

There are two major components: the enhanced encoder and decoder, respectively (recall Figure 1). The core element of our enhanced encoder is the self-attention mechanism that incorporates IRPE. This mechanism aims to provide directional information for the sequential image features in a two-dimensional encoding, enabling the model to extract better global features. The core element of the enhanced decoder is the cross-attention mechanism. We use learnable label embeddings as queries and the global image features obtained from the encoder as keys and values for cross-attention calculation. In what follows, we first cover the details of IRPE, and then discuss the enhanced encoder and decoder, respectively.

1) Image Relative Positional Encoding: Inspired by [41], we attempt to incorporate IRPE into our model. One can incorporate the dot-product and context based IRPE. To understand how IRPE is incorporated into the encoder, we briefly introduce the process of applying IRPE simultaneously to query, key, and value, for example. First, we can apply IRPE to both query and key. The dot-product attention weights e_{ij} is computed as:

$$e_{ij} = \frac{\left(x_i W^Q\right)\left(x_j W^K\right)^T + b_{ij}}{\sqrt{d_z}} \tag{3}$$

where $b_{ij} \in \mathbb{R}$ is the 2D relative positional encoding. For the contextual mode,

$$b_{ij} = \left(x_i W^Q\right)\left(r_{ij}^K\right)^T + \left(x_j W^K\right)\left(r_{ij}^Q\right)^T \tag{4}$$

where r_{ij}^K, $r_{ij}^Q \in \mathbb{R}^d$ are both learnable vectors. Second, we incorporate IRPE into the value. Consequently, the output after the self-attention layer can be represented as:

$$z_i = \sum_{j=1}^{n} \text{softmax}\left(e_{ij}\right)\left(x_j W^V + r_{ij}^V\right) \tag{5}$$

2) Encoder: Recall Section III-B, we have obtained local image features. To obtain the global features, we need to perform a second round of feature extraction on the feature maps. We achieve it by using the self-attention mechanism within the Transformer. Specifically, the reshaped serialized image features undergo a linear transformation to map them into q,

k, and v inputs for the self-attention layer. By combining the self-attention layer with the applied IRPE, we capture more comprehensive global image features. The self-attention layers in the enhanced encoder enable each image feature vector to interact with all other feature vectors, facilitating the global contextual information. Formally, the computational process of the self-attention layer is as follows:

$$F_i = MultiHead\left(Z_i, \tilde{Z}_i, Z_i\right) \tag{6}$$

where Z_i represents image sequence feature, $\tilde{Z}_i$ represents image sequence feature augmented with image relative positional encoding, F_i represents the new feature representation obtained from image sequences after undergoing multi-head attention. Note that IRPE can also be embedded into the Query and Value; but adding IRPE to Key can already achieve highly desirable results, which will be shown in Section IV.

3) Decoder: We utilize learnable label embeddings as queries, image features from the encoder as keys and values; and we employ multi-layer Transformer decoders to perform cross-attention computation, aiming to identify regions of interest. Specifically, the cross-attention computation is as follows:

$$Q_j = MultiHead\left(Q_{j-1}, F_i, F_i\right) \tag{7}$$

where Q_j denotes the query input; when $j = 1$, Q_0 represents the learnable label embedding; F_i denotes the sequence of image features obtained from the encoder, and it is actually identical to the F_i in Equation 6.

It is important to note that the decoder structure we used excludes the self-attention layer. In prior studies, the self-attention layer within the decoder was considered a component that could potentially enhance the internal relationships of label embeddings. However, we realize that, in multi-label image classification tasks, the input to the decoder consists of learnable label embeddings, where the labels themselves are not inherently correlated, since they are independent texts. If one conducts self-attention calculations on these learnable label embeddings, it actually forces the labels to learn relationships with each other, potentially leading to spurious/fake label correlations.

D. Final Classification and Loss

1) Final Classification: After the feature-label interaction is completed, we use a linear projection layer to assign a corresponding value to each class, and obtain the corresponding prediction score using a sigmoid function.

2) Loss: To address the issue of sample imbalance, our framework generally employs an asymmetric polynomial loss (APL) function, although traditional binary cross-entropy and focal loss can also be utilized. Here, APL is built upon the foundation of asymmetric loss, it employs a Taylor expansion to enable the function to fit more complex computations. The specific computation is detailed as follows:

$$\begin{aligned} L_{APL} = \sum_{i=0}^{C} \Bigg\{ & y_i (1-p_i)^{\gamma^+} \Big[-\log p_i + (\alpha_1 - 1)(1-p_i) \\ & + \left(\alpha_2 - \frac{1}{2}\right)(1-p_i)^2 \Big] + (1-y_i) p_{res}^{\gamma^-} \\ & \Big[-\log(1-p_i) + (\beta_1 - 1) p_{res} \Big] \Bigg\} / C \end{aligned} \tag{8}$$

where y_i represents the true label of the ith sample, p_i represents the predicted probability of the ith sample being in the positive class, C are coefficients, and p_{res} denotes $\max(p_i - p_{th}, 0)$. In our experiments, we set $\gamma^+ = 0$ and $\gamma^- = 2$ by default.

IV. Experiments

A. Datasets, Evaluation Metrics, and Competitors

To assess our method, we conducted experiments on several large public datasets, namely, MS-COCO [44], PASCAL VOC [45], and NUS-WIDE [46]. There details are as follows.

- **MS-COCO**[1]: It is a large-scale dataset designed for object detection and segmentation, widely used in recent years for evaluating multi-label image classification. It comprises 82,783 training images, 40,504 validation images, covering 80 common objects, with an average of 2.9 labels per image.
- **PASCAL VOC**[2]: It contains two versions: VOC 2007 and VOC 2012. VOC 2007 is utilized for object detection and image classification tasks. This dataset comprises 5011 images as the ***train-val*** set and 4952 images as the ***test*** set, covering 20 common object categories. On average, each image contains annotations for 1.6 object instances. The object categories within VOC 2007 encompass, but are not limited to, humans, dogs, cats, airplanes, ships, bicycles, and more. Each object instance is annotated with its corresponding category label and bounding box delineating its spatial extent. VOC 2012 is a classic computer vision dataset used for object detection and image segmentation tasks. It comprises bounding boxes for each object and pixel-level segmentation annotations. VOC 2012 consists of 11,540 images for the ***train-val*** set and 10,991 images for the **test** set.
- **NUS-WIDE**[3]: It is a large-scale dataset used for image annotation and image retrieval. It consists of 125449 images for the ***train*** set and 83898 images for the **test** set, manually annotated with 81 visual concepts.

In terms of evaluation metrics, we utilized the Average Precision (AP) for each individual class; and we use the mean Average Precision (mAP) across all classes, which is computed

[1]https://cocodataset.org

[2]http://host.robots.ox.ac.uk/pascal/VOC/

[3]https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html

TABLE I: Comparison results on the MS-COCO dataset. The backbones marked with 22k are pre-trained on the ImageNet-22k dataset. In the comparison, our primary focus lies in assessing more comprehensive metrics, including mAP, CF1, and OF1. Other metrics can be used for reference.

Method	Backbone	Resolution	mAP	CP	CR	CF1	OP	OR	OF1
SRN[23]	ResNet101	224x224	77.1	81.6	65.4	71.2	82.7	69.9	75.8
ResNet-101[21]	ResNet101	224x224	78.3	80.2	66.7	72.8	83.9	70.8	76.8
ML-GCN[24]	ResNet101	448x448	83.0	85.1	72.0	78.0	85.8	75.4	80.3
MS-CMA[25]	ResNet101	448x448	83.8	82.9	84.4	78.4	84.4	77.9	81.0
MCAR[26]	ResNet101	448x448	83.8	85.0	72.1	78.0	88.0	73.9	80.3
CCD[22]	ResNet101	448x448	84.0	87.2	70.9	77.3	88.8	74.6	81.1
TDRG[5]	ResNet101	448x448	84.6	86.0	73.1	79.0	86.6	76.4	81.2
IDA[30]	ResNet101	448x448	84.8	-	-	78.7	-	-	80.9
Q2L[27]	ResNet101	448x448	84.9	84.8	74.5	79.3	86.6	76.9	81.5
SSGRL[11]	ResNet101	576x576	83.8	89.9	68.5	76.8	91.3	70.8	79.7
C-Trans[10]	ResNet101	576x576	85.1	86.3	74.3	79.9	87.7	76.5	81.7
ADD-GCN[3]	ResNet101	576x576	85.2	84.7	75.9	80.1	84.9	79.4	82.0
TDRG[5]	ResNet101	576x576	86.0	87.0	74.7	80.4	87.5	77.9	82.4
IDA[30]	ResNet101	576x576	86.3	-	-	80.4	-	-	82.5
Q2L[27]	ResNet101	576x576	86.5	85.8	76.7	81.0	87.0	78.9	82.8
Ours	ResNet101	448x448	85.2	84.9	74.7	79.5	86.8	76.9	81.5
Ours	ResNet101	576x576	**86.8**	85.9	77.0	**81.2**	87.3	79.0	**83.0**
ASL[4]	TResNetL	448x448	86.6	87.2	76.4	81.4	88.2	79.2	81.8
TResNetL[42]	TResNetL(22k)	448x448	88.4	-	-	-	-	-	-
Q2L[27]	TResNetL(22k)	448x448	89.2	86.3	81.4	**83.8**	86.5	83.3	**84.9**
Ours	TResNetL(22k)	448x448	**89.3**	86.2	81.5	**83.8**	86.5	83.2	**84.9**
Convnextv2-B[43]	ConvNextv2-B(22k)	384x384	89.6	89.3	80.5	84.7	89.5	82.1	85.7
Swin-L[35]	Swim-L(22k)	384x384	89.6	89.9	80.2	84.8	90.4	82.1	86.1
CCD[22]	Swim-L(22k)	384x384	90.3	85.9	84.0	84.6	85.1	86.4	85.7
IDA[30]	Swim-L(22K)	384x384	90.3	-	-	84.7	-	-	85.9
Q2L[27]	Swim-L(22k)	384x384	90.5	89.4	81.7	85.4	89.8	83.2	86.4
Ours	ConvNextv2-B(22k)	384x384	**91.1**	89.8	82.3	**85.9**	90.0	83.8	**86.8**

as follows: $mAP = \frac{\sum_{c=1}^{C} \text{AP}_c}{C}$, where c, and C denote current specific class, and number of all classes respectively. Meanwhile, we employed Overall Precision (OP), and Overall Recall (OR), Overall F1-Score (OF1), they are computed as: $OP = \frac{\sum_i M_c^i}{\sum_i M_p^i}$, $OR = \frac{\sum_i M_c^i}{\sum_i M_g^i}$, $OF1 = \frac{2 \times OP \times OR}{OP+OR}$. Also, we utilized per-Category Precision (CP), per-Category Recall (CR), and per-Category F1-Score (CF1), they are computed as: $CP = \frac{1}{C}\sum_i \frac{M_c^i}{M_p^i}$, $CR = \frac{1}{C}\sum_i \frac{M_c^i}{M_g^i}$, $CF1 = \frac{2 \times CP \times CR}{CP+CR}$.

To examine the competitiveness of our proposed method, we compared it with state-of-the-art methods (e.g., published in ICLR 2023 [30] and CVPR 2023 [43]). These competitors can be categorized into three classes (recall Section II):

- **Region of Interest based methods**: ResNet101 [21], Fev+lv [29], HCP [28], RDAL [33], MCAR [26], TResNetL [42], Convnextv2-B [43], Swim-L [35].
- **Transformer-based methods**: TDRG [5], IDA [30], Q2L [27], C-Trans [10].
- **Other methods**: CNN-RNN [9], Focal Loss [40], SRN [23], ML-GCN [24], MS-CMA [25], CCD [22], SSGRL [47], ADD-GCN [3], ASL [4].

B. Experimental Settings

We adopted the following experimental settings by default, unless stated otherwise. As for the image enhancement, we used cutout; and for regularization we used RandAugment. Considering GPU efficiency and ensuring fairness and convenience, we selected ResNet101, Tresnet-L, and Convnextv2-Base as the backbone models when comparing various methods. Regarding resolution, we utilized three resolutions: 384, 448, and 576. In what follows, when we use ResNet101 and 448 resolution, the resulting output features of backbone are denoted as $H \times W \times d_0 = 14 \times 14 \times 2048$. Subsequently, the obtained image features are reshaped to match the shape required by the Transformer. Specifically, in our experiments, we employed one encoder layer and two decoder layers (which can be adjustable as needed) to update the label features. Finally, we accomplished the classification prediction through a linear projection layer.

We trained the model for 80 epochs using an AdamW optimizer with a weight decay of 5e-3, a momentum of 0.9; β_1 and β_2 are set to 0.9 and 0.9999, respectively. Additionally, we employed the one-cycle learning rate policy with a maximum learning rate of 5e-5. The experimental platform is an Ubuntu 18.04 system, equipped with an Intel Xeon Gold 6330 processor @ 2.0 GHz with 60 cores, 240 GB of RAM, and 6 Nvidia GeForce RTX 3090 GPUs.

C. Comparison with State-Of-The-Art Methods

1) Performance on MS-COCO: Table I presents the comparative results. We can see that, when Resnet-101 is used as the backbone, our model consistently outperforms other state-of-the-art methods in terms of mAP, CF1, OF1, and some other reference metrics. Particularly, although Q2L also employs a Transformer based two-stage approach, our model beats it in all metrics at both resolutions. Also, our model beats the IDA[30] (ICLR'2023) by 0.5% at 576X576 resolution. Remark that, an increase of 0.5% is non-trivial in MLIC tasks.

TABLE II: Comparison results on PASCAL VOC 2007, in terms of AP and mAP in percentages. The resolution of all results are 448×448, with the exception of ADD-GCN and SSGRL, which have resolution of 576x576.

Methods	*aero*	*bike*	*bird*	*boat*	*bottle*	*bus*	*car*	*cat*	*chair*	*cow*	*table*	*dog*	*horse*	*mbike*	*person*	*plant*	*sheep*	*sofa*	*train*	*tv*	mAP
CNN-RNN[9]	96.7	83.1	94.2	92.8	61.2	82.1	89.1	94.2	64.2	83.6	70.0	92.4	91.7	84.2	93.7	59.8	93.2	75.3	**99.7**	78.6	84.0
Fev+Lv[29]	97.9	97.0	96.6	94.6	73.6	93.9	96.5	95.5	73.7	90.3	82.8	95.4	97.7	95.9	98.6	77.6	88.7	78.0	98.3	89.0	90.6
HCP[28]	98.6	97.1	98.0	95.6	75.3	94.7	95.8	97.3	73.1	90.2	80.0	97.3	96.1	94.9	96.3	78.3	94.7	76.2	97.9	91.5	90.9
RDAL[33]	98.6	97.4	96.3	96.2	75.2	92.4	96.5	97.1	76.5	92.0	87.7	96.8	97.5	93.8	98.5	81.6	93.7	82.8	98.6	89.3	91.9
MCAR[26]	99.7	99.0	98.5	98.2	**85.4**	96.9	97.4	**98.9**	83.7	95.5	88.8	**99.1**	98.2	95.1	99.1	84.8	97.1	**87.8**	98.3	94.8	94.8
TDRG[5]	**99.9**	98.9	98.4	**98.7**	81.9	95.8	97.8	98.0	**85.2**	95.6	**89.5**	98.8	98.6	97.1	99.1	86.2	97.7	87.2	99.1	**95.3**	95.0
Ours	**99.9**	**99.3**	**98.9**	98.1	84.8	**97.6**	**98.6**	**98.9**	82.7	**98.3**	88.0	98.8	**98.7**	**97.3**	**99.2**	**88.8**	**98.6**	85.3	99.4	94.9	**95.3**
SSGRL[11]	99.7	98.4	98.0	97.6	85.7	96.2	98.2	98.8	82.0	98.1	89.7	98.8	98.7	97.0	99.0	86.9	98.1	85.8	99.0	93.7	95.0
ADD-GCN[3]	99.8	99.0	98.4	**99.0**	86.7	98.1	98.5	98.3	**85.8**	98.3	88.9	98.8	99.0	97.4	99.2	88.3	98.7	**90.7**	99.5	**97.0**	96.0
TResNet[42]	**99.9**	98.4	98.9	98.7	86.8	98.2	98.7	98.5	83.1	98.3	89.5	98.8	**99.2**	98.6	99.3	89.5	99.4	86.8	99.6	95.2	95.8
Q2L[27]	**99.9**	98.9	99.0	98.4	87.7	**98.6**	**98.8**	99.1	84.5	98.3	89.2	**99.2**	**99.2**	**99.2**	99.3	90.2	98.8	88.3	99.5	95.5	96.1
Ours	**99.9**	**99.1**	**99.2**	98.8	**87.9**	**98.6**	98.7	**99.2**	83.4	**98.4**	**90.6**	99.1	98.9	98.9	**99.4**	**90.5**	**99.7**	88.6	**99.8**	95.7	**96.2**

TABLE III: Comparison results on PASCAL VOC 2012. The resolution of all results are 448×448, except that ADD-GCN and SSGRL have resolution of 576×576. Differing from the VOC2007 dataset, for the sake of impartiality, we conducted tests utilizing the official evaluation server, all results obtained are provided by the VOC2012 official evaluation server.

Methods	*aero*	*bike*	*bird*	*boat*	*bottle*	*bus*	*car*	*cat*	*chair*	*cow*	*table*	*dog*	*horse*	*mbike*	*person*	*plant*	*sheep*	*sofa*	*train*	*tv*	mAP
Fev+Lv[29]	98.4	92.8	93.4	90.7	74.9	93.2	90.2	96.1	78.2	89.8	80.6	95.7	96.1	95.3	97.5	73.1	91.2	75.4	97.0	88.2	89.4
HCP[28]	99.1	92.8	97.4	94.4	79.9	93.6	89.8	98.2	78.2	94.9	79.8	97.8	97.0	93.8	96.4	74.3	94.7	71.9	96.7	88.6	90.5
MCAR[26]	99.6	97.1	98.3	96.6	87.0	95.5	94.4	98.8	87.0	96.9	85.0	98.7	98.3	97.3	99.0	83.8	96.8	83.7	98.3	93.5	94.3
SSGRL[11]	99.7	96.1	97.7	96.5	86.9	95.8	95.0	98.9	88.3	97.6	87.4	99.1	99.2	97.3	99.0	84.8	98.3	85.8	99.2	94.1	94.8
ADD-GCN[3]	99.8	97.1	98.6	96.8	89.4	97.1	96.5	99.3	89.0	97.7	87.5	99.2	99.1	97.7	99.1	86.3	**98.8**	87.0	99.3	95.4	95.5
CCD[22]	99.8	98.2	98.3	98.0	88.6	97.4	96.9	99.1	90.8	**98.9**	**90.2**	99.2	**99.6**	98.4	99.0	87.7	98.4	88.8	99.7	96.4	96.1
Q2L[27]	99.9	98.2	**99.3**	98.1	90.4	**97.7**	97.4	99.4	**92.7**	98.7	89.9	**99.4**	99.5	**99.0**	**99.4**	88.4	**98.8**	89.3	99.6	96.8	96.6
Ours	**100.0**	**98.5**	99.1	**98.5**	**90.6**	97.5	**97.6**	**99.5**	91.8	98.6	89.6	**99.4**	99.3	98.9	**99.4**	**89.2**	98.7	**90.5**	**99.8**	**97.0**	**96.7**

TABLE IV: Comparison of our methods to known state-of-the-art models on NUS-WIDE. All results are reported at the input resolution of 448 × 448.

Method	Backbone	mAP	CF1	OF1
MS-CMA[25]	ResNet101	61.4	60.5	73.8
SRN[23]	ResNet101	62.0	58.5	73.4
Q2L[27]	ResNet101	65.0	63.1	75.0
CCD[22]	ResNet101	65.1	61.3	75.0
Ours	ResNet101	**65.4**	**63.3**	**75.1**
CE	TresNetL	63.1	61.7	74.6
Focal loss[40]	TresNetL	64.0	62.9	74.7
ASL[4]	TresNetL	65.2	63.6	75.0
Q2L[27]	TresNetL	66.3	64.0	75.0
Ours	TresNetL	**67.1**	**64.4**	**75.2**

All these results consistently demonstrate the competitiveness and effectiveness of our proposed approach. Moreover, we can see that, when Tresnet-L is used as the backbone, our model achieves the best performance on all core evaluation metrics including mAP, CF1 and OF1. This indicates the versatility of our proposed framework across various backbones.

Furthermore, we have observed that our proposed framework is highly compatible with the ConvNeXt V2 backbone. The unique Global Response Normalization (GRN) layer of ConvNeXt V2 effectively addresses the issue of feature collapse, preventing label embeddings from activating similar but incorrect image regions, thereby enhancing the model's comprehension of input data. ConvNeXt V2 is adept at extracting rich local image features while avoiding redundant activations across different feature channels. Meanwhile, the Transformer component focuses on processing global image features and precise activation of label-specific features. Particularly, after the incorporation of the IRPE, the Transformer encoder in conjunction with the ConvNeXt V2 can leverage the best global feature extraction capabilities, thereby improving the model's accuracy. Experimental results indicate that our method can beat the baseline (ConvNeXt V2-Base) by 1.5%, achieving an ultra-high precision of 91.1% on the MS-COCO dataset. This validates the effectiveness of the integration of our framework and ConvNeXt V2, as well as the superiority of our framework itself. We have to mention that, it is non-trivial to obtain an increase of 1.5% to the competitor, Convnext2-B [43], published in 2023.

2) Performance on PASCAL VOC: (i) Results on VOC 07. Table II presents the results. In the first part of Table II, we utilized the standard Resnet101 as the backbone with a 448 resolution. The reported metrics include the AP values of 20 categories and the overall mAP. We can see that our method achieves the best results in most categories. In terms of mAP, we have surpassed all these methods. More specifically, comparing to region-based approaches (e.g., Fev+Lv, HCP, RDAL), it can be observed that our result is obviously better than them. This is mainly because our method utilizes the Transformer decoder to obtain class-specific representations, resulting in more accurate ROI (i.e., region of interest). On the other hand, our method also beats the Transformer-based methods (e.g., TDRG), this is mainly because our method introduces IRPE into the encoder, providing us the spatial relationships between pixels. In the second part of Table II, except for SSGRL and ADD-GCN, which use Resnet101, all other methods use Tresnet-L. We can see that our model also achieves the best results in most categories. Particularly, we achieved a mAP value of 96.2% on this dataset. Considering the experimental results in Table II as a whole, our method beats all these competitors in terms of mAP.

(ii) Results on VOC 12. Since complete labels for the *test* set are not provided officially, we conducted testing on the official evaluation server. Table III presents the comparative results

(the reader can also visit the anonymous link[4]). The training strategy used in this experiment is the same as that for the second part of VOC 2007. Similar to the results demonstrated on the VOC 2007 dataset, we also achieved the state-of-the-art performance on the VOC 2012 dataset. Particularly, we observe that, even if SSGRL and ADDGCN used a higher resolution (576x576), we still achieve higher mAP values than theirs (i.e., demonstrating the competitiveness of our model). The possible reasons are that, SSGRL and ADD-GCN used graph structures to establish label correlations, primarily focusing on local features, their ability to capture global features is weak. In contrast, our method utilized cross-attention to compute the attention weights of each label for different image regions, it implicitly captured the relationships between labels. Furthermore, our encoder with IRPE can acquire global information efficiently.

3) Performance on NUS-WIDE: We followed the steps outlined in [4] for evaluation. We conducted experiments using ResNet101 and TResNetL as backbones, respectively. The reported results include mAP, CF1, and OF1. We can see from Table IV that our model achieves the best performance across all these evaluation metrics. Particularly, compared to Q2L, our model outperformed it across two backbones in terms of all these metrics. To some extent, it implies that incorporating IRPE into the encoder should be helpful. We remark that NUS-WIDE is a highly challenging dataset characterized by its large scale, diverse samples with noise, imbalanced labels, and complex visual features. These factors incur that methods on the NUS-WIDE dataset have not achieved very good performance (e.g., about 60+% in terms of mAP value), and the accuracy gains achieved by these methods are typically a gradual process. With this in mind, one can easily understand that our model has achieved a considerable improvement (e.g., it increased 0.3% and 0.8% across two backbones in terms of mAP values).

D. Ablation study

1) Encoder: Previous works based on the Transformer architecture either overlook the importance of the encoder, or only use standard encoders. Table V confirms the necessity of the encoder in multi-label image classification model. We can see that, on the MS-COCO dataset, employing a standard Transformer yields a performance increase, compared to the Transformer without encoders. Particularly, when our enhanced encoder is employed, the performance is enhanced further. As for the NUS-WIDE dataset, we obtain similar findings. All these evidences imply that using encoder (especially our enhanced encoder) is useful to achieve better results. Remark that, MLIC tasks are challenging, an increase of 0.5% is also rather considerable.

2) IRPE: As IRPE can be simultaneously embedded in the Query, Key, and Value within the self-attention layers, we apply different embedding combinations to the self-attention layers of the encoder. This allows us to observe the contributions of positional encoding at different embedding positions to the model. As the same as that in [41], we mainly consider three cases (key, query-key, query-key-value) . The results are shown in Table VI. We can see that applying IRPE on the Query-Key (all backbones), or solely on the Key (see the Resnet101 backbone) yields the best results. Particularly, we can see that applying IRPE solely on the Key achieves a large improvement. However, when applying IRPE on the Query-Key or Query-Key-Value further, the improvement is marginal or even no improvement (e.g., the case of Resnet101). These results show that adding IRPE to Key is much more important than to Query and Value. We guess that the Key plays a crucial role in determining attention distribution. By incorporating two-dimensional positional encoding solely on the Key, the model may grasp most of the positional relationships, and this is particularly suitable for spatial information in image data.

[4]http://host.robots.ox.ac.uk:8080/anonymous/XFI9GJ.html

TABLE V: Comparisons of our encoders, standard encoders, and no encoders. Our encoder contains IRPE. For MS-COCO and NUSWIDE datasets, a resolution of 448x448 is utilized.

Dataset	Setting	mAP
MS-COCO	w/ our encoder	**85.22**
	w/ standard encoder	84.90
	w/o encoder	84.72
NUSWIDE	w/ our encoder	**65.44**
	w/ standard encoder	65.14
	w/o encoder	64.85

TABLE VI: Comparison of IRPE applying to the Query, Key, and Value. For the ResNet101 and TResNet-L backbones, a resolution of 448x448 is utilized. For the Convnextv2-Base backbone, a resolution of 384x384 is employed.

Method	Backbone	mAP
Ours-without IRPE	Resnet101	84.90
Ours-IRPE-K	Resnet101	**85.20**
Ours-IRPE-QK	Resnet101	**85.20**
Ours-IRPE-QKV	Resnet101	85.17
Ours-without IRPE	TresNetL	89.05
Ours-IRPE-K	TresNetL	89.21
Ours-IRPE-QK	TresNetL	**89.27**
Ours-IRPE-QKV	TresNetL	89.22
Ours-without IRPE	Convnextv2-B	90.88
Ours-IRPE-K	Convnextv2-B	91.07
Ours-IRPE-QK	Convnextv2-B	**91.09**
Ours-IRPE-QKV	Convnextv2-B	91.06

TABLE VII: Comparison results between using self-attention layers and without self-attention layers. "*" denotes that self-attention layer is only used in the second decoder.

Dataset	Setting	mAP
MS-COCO	w/ self-attention	85.14
	w/ self-attention*	85.07
	w/o self-attention	**85.20**
NUSWIDE	w/ self-attention	65.41
	w/ self-attention*	65.43
	w/o self-attention	**65.44**

3) Self-Attention: The self-attention layer in the Transformer decoder is often thought to enable better embedding representations of labels, and previous works that used the standard decoder often employ the self-attention layer. In Section III-C3, we argue that the self-attention layer in the

TABLE VIII: Comparison between different loss functions.

Loss Function	mAP
Binary Cross-Entropy	84.92
Focal Loss[40]	85.11
Asymmetric Loss[4]	85.15
Asymmetric Polynomial Loss[12]	**85.20**

decoder may be not much helpful. To validate this, we conducted experiments on two large datasets. Table VII shows the comparison results. From this table, we can see that removing self-attention layers from the decoder consistently achieves better performance on these datasets. One of possible reasons is that, applying self-attention to these learnable label embeddings forces the labels to learn inter-relationships, which may result in spurious label correlations, as mentioned in Section III-C3. Another possible reason is that, before computing cross-attention (recall Fig. 1), the input label embeddings undergo a linear transformation, which enables the input to fit adaptively the cross-attention module's expected input; in other words, removing self-attention layer in decoders makes less or even no negative impact.

4) Loss Function: The Asymmetric Polynomial Loss (APL) utilizes a Taylor expansion relative to the Asymmetric Loss (AL), allowing it to handle more intricate computations. On the other hand, compared to Binary Cross-Entropy (BCE) and Focal Loss (FL), it possesses better capabilities in addressing sample imbalance issues. To validate their performance when they are integrated into our framework, we conducted comparisons. For Asymmetric Loss and Asymmetric Polynomial Loss, we configured $\gamma+ = 0$ and $\gamma- = 2$ respectively. The experimental results are shown in Table VIII. We can see that the best performance is achieved when loss function is Asymmetric Polynomial Loss. The reason could be that, APL can provide a more refined ability of handling positive and negative samples than other competitors.

V. Conclusion

In this paper, we have proposed a novel Transformer based two stage framework for multi-label image classification. Our framework incorporates RoI-based technique and an enhanced Transformer structure. We have validated its effectiveness and competitiveness on several widely used multi-label image classification datasets. Our study yields several important findings for MLIC tasks: (i) incorporating IRPE into Transformer decoders is useful; (ii) removing self-attention layers in the Transformer decoder has no negative impact, or even improves the performance; and (iii) the APL function could be more compatible with Transformer-based frameworks, compared against BCE, FL, and AL.

References

[1] M. Shah, K. Viswanathan, C.-T. Lu, A. Fuxman, Z. Li, A. Timofeev, C. Jia, and C. Sun, "Inferring context from pixels for multimodal image classification," in *Proceedings of CIKM*, pp. 189–198, 2019.

[2] A. Balayn, P. Soilis, C. Lofi, J. Yang, and A. Bozzon, "What do you mean? interpreting image classification with crowdsourced concept extraction and analysis," in *Proceedings of WWW*, pp. 1937–1948, 2021.

[3] J. Ye, J. He, X. Peng, W. Wu, and Y. Qiao, "Attention-driven dynamic graph convolutional network for multi-label image recognition," in *Proceedings of ECCV*, pp. 649–665, 2020.

[4] T. Ridnik, E. Ben-Baruch, N. Zamir, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor, "Asymmetric loss for multi-label classification," in *Proceedings of ICCV*, pp. 82–91, 2021.

[5] J. Zhao, K. Yan, Y. Zhao, X. Guo, F. Huang, and J. Li, "Transformer-based dual relation graph for multi-label image recognition," in *Proceedings of ICCV*, pp. 163–172, 2021.

[6] S. Gilhuber, R. Hvingelby, M. L. A. Fok, and T. Seidl, "How to overcome confirmation bias in semi-supervised image classification by active learning," in *Proceedings of ECML/PKDD*, pp. 330–347, 2023.

[7] D. Rymarczyk, Ł. Struski, J. Tabor, and B. Zieliński, "Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification," in *Proceedings of SIGKDD*, pp. 1420–1430, 2021.

[8] X. Yang, H. Ye, Y. Ye, X. Li, and S. Ji, "Generative max-mahalanobis classifiers for image classification, generation and more," in *Proceedings of ECML/PKDD*, pp. 67–83, 2021.

[9] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proceedings of CVPR*, pp. 2285–2294, 2016.

[10] J. Lanchantin, T. Wang, V. Ordonez, and Y. Qi, "General multi-label image classification with transformers," in *Proceedings of CVPR*, pp. 16478–16488, 2021.

[11] T. Chen, M. Xu, X. Hui, H. Wu, and L. Lin, "Learning semantic-specific graph representation for multi-label image recognition," in *Proceedings of ICCV*, pp. 522–531, 2019.

[12] Y. Huang, J. Qi, X. Wang, and Z. Lin, "Asymmetric polynomial loss for multi-label classification," in *Proceedings of ICASSP*, pp. 1–5, 2023.

[13] D. Rymarczyk, A. Pardyl, J. Kraus, A. Kaczyńska, M. Skomorowski, and B. Zieliński, "Protomil: multiple instance learning with prototypical parts for whole-slide image classification," in *Proceedings of ECML/PKDD*, pp. 421–436, 2022.

[14] D. Yu, X. Liu, and B. Yang, "Zero-shot image classification with logic adapter and rule prompt," in *Proceedings of WWW*, pp. 2075–2084, 2024.

[15] K. K. Gadiraju, B. Ramachandra, Z. Chen, and R. R. Vatsavai, "Multimodal deep learning based crop classification using multispectral and multitemporal satellite imagery," in *Proceedings of SIGKDD*, pp. 3234–3242, 2020.

[16] M. Cao, X. Zhou, Y. Xu, Y. Pang, and B. Yao, "Adversarial domain adaptation with semantic consistency for cross-domain image classification," in *Proceedings of CIKM*, pp. 259–268, 2019.

[17] T. Boone-Sifuentes, A. Nazari, I. Razzak, M. R. Bouadjenek, A. Robles-Kelly, D. Ierodiaconou, and E. S. Oh, "Marine-tree: A large-scale marine organisms dataset for hierarchical image classification," in *Proceedings of CIKM*, pp. 3838–3842, 2022.

[18] M. Zeng, B. Yao, Z.-J. Wang, Y. Shen, F. Li, J. Zhang, H. Lin, and M. Guo, "Catiri: An efficient method for content-and-text based image retrieval," *Journal of Computer Science and Technology*, vol. 34, pp. 287–304, 2019.

[19] Y. Tang, F. Borisyuk, S. Malreddy, Y. Li, Y. Liu, and S. Kirshner, "Msuru: Large scale e-commerce image classification with weakly supervised search data," in *Proceedings of SIGKDD*, pp. 2518–2526, 2019.

[20] E. Moons, T. Tuytelaars, and M.-F. Moens, "Text-enriched representations for news image classification," in *Proceedings of WWW*, pp. 99–100, 2018.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of CVPR*, pp. 770–778, 2016.

[22] R. Liu, H. Liu, G. Li, H. Hou, T. Yu, and T. Yang, "Contextual debiasing for visual recognition with causal mechanisms," in *Proceedings of CVPR*, pp. 12755–12765, 2022.

[23] F. Zhu, H. Li, W. Ouyang, N. Yu, and X. Wang, "Learning spatial regularization with image-level supervisions for multi-label image classification," in *Proceedings of CVPR*, pp. 5513–5522, 2017.

[24] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proceedings of CVPR*, pp. 5177–5186, 2019.

[25] R. You, Z. Guo, L. Cui, X. Long, Y. Bao, and S. Wen, "Cross-modality attention with semantic graph embedding for multi-label classification," in *Proceedings of AAAI*, pp. 12709–12716, 2020.

[26] B.-B. Gao and H.-Y. Zhou, "Learning to discover multi-class attentional regions for multi-label image recognition," *IEEE Transactions on Image Processing*, vol. 30, pp. 5920–5932, 2021.

[27] S. Liu, L. Zhang, X. Yang, H. Su, and J. Zhu, "Query2label: A simple transformer way to multi-label classification," *arXiv preprint arXiv:2107.10834*, 2021.

[28] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "Hcp: A flexible cnn framework for multi-label image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1901–1907, 2015.

[29] H. Yang, J. Tianyi Zhou, Y. Zhang, B.-B. Gao, J. Wu, and J. Cai, "Exploit bounding box annotations for multi-label object recognition," in *Proceedings of CVPR*, pp. 280–288, 2016.

[30] R. Liu, J. Huang, T. H. Li, and G. Li, "Causality compensated attention for contextual biased visual recognition," in *Proceedings of ICLR*, 2023.

[31] J. Zhan, J. Liu, W. Tang, G. Jiang, X. Wang, B.-B. Gao, T. Zhang, W. Wu, W. Zhang, C. Wang, *et al.*, "Global meets local: Effective multi-label image classification via category-aware weak supervision," in *Proceedings of ACM Multimedia*, pp. 6318–6326, 2022.

[32] T. Ridnik, G. Sharir, A. Ben-Cohen, E. Ben-Baruch, and A. Noy, "Ml-decoder: Scalable and versatile classification head," in *Proceedings of WACV*, pp. 32–41, 2023.

[33] Z. Wang, T. Chen, G. Li, R. Xu, and L. Lin, "Multi-label image recognition by recurrently discovering attentional regions," in *Proceedings of ICCV*, pp. 464–472, 2017.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Proceedings of NIPS*, vol. 30, 2017.

[35] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of ICCV*, pp. 10012–10022, 2021.

[36] M. Sun, W. Ma, and Y. Liu, "Global and local feature interaction with vision transformer for few-shot image classification," in *Proceedings of CIKM*, pp. 4530–4534, 2022.

[37] W. Kuang, Q. Zhu, and Z. Li, "Multi-label image classification with multi-scale global-local semantic graph network," in *Proceedings of ECML/PKDD*, pp. 53–69, Springer, 2023.

[38] W. Yan, R. Li, J. Wang, Y. Li, J. Wang, P. Zhou, and X. Gu, "Imbalance rectification in deep logistic regression for multi-label image classification using random noise samples," in *Proceedings of CIKM*, pp. 1131–1140, 2019.

[39] C. Zhang, W. Tavanapong, G. Kijkul, J. Wong, P. C. De Groen, and J. Oh, "Similarity-based active learning for image classification under class imbalance," in *Proceedings of ICDM*, pp. 1422–1427, 2018.

[40] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of ICCV*, pp. 2980–2988, 2017.

[41] K. Wu, H. Peng, M. Chen, J. Fu, and H. Chao, "Rethinking and improving relative position encoding for vision transformer," in *Proceedings of ICCV*, pp. 10033–10041, 2021.

[42] T. Ridnik, H. Lawen, A. Noy, E. Ben Baruch, G. Sharir, and I. Friedman, "Tresnet: High performance gpu-dedicated architecture," in *Proceedings of WACV*, pp. 1400–1409, 2021.

[43] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "Convnext v2: Co-designing and scaling convnets with masked autoencoders," in *Proceedings of CVPR*, pp. 16133–16142, 2023.

[44] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of ECCV*, pp. 740–755, 2014.

[45] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, pp. 98–136, 2015.

[46] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "Nus-wide: a real-world web image database from national university of singapore," in *Proceedings of CIVR*, pp. 1–9, 2009.

[47] Z.-M. Chen, X.-S. Wei, X. Jin, and Y. Guo, "Multi-label image recognition with joint class-aware map disentangling and label correlation embedding," in *Proceedings of ICME*, pp. 622–627, 2019.