LCNet: Learnable Label Correlation Network for Multi-Label Image Classification

Author: M.H.D. Scholten, Supervisor: dr. D.V. Le Viet Duc
dept. of Electrical Engineering, Mathematics and Computer Science
University of Twente, Enschede, Netherlands
m.h.d.scholten@student.utwente.nl
v.d.le@utwente.nl

Abstract

In computer vision, multi-label image classification is a well known topic which can be applied in many real-world applications. In classical multi-label image classification, only the spatial features of images are used as input of classification models. In this paper we introduce LCNet: A Multi-Label Classification model, which includes a learnable semantic graph followed by a novel decoder to fuse the multiple modalities. Our decoder is stack-able which allows the model to understand deeper relations between the modalities. In addition, we design a feature to reduce spatial resolution loss called Crop-Forwarding. Crop-Forwarding allows higher resolution images to be forwarded without using higher resolution spatial feature extractors. Extensive experiments conducted on several multi-label classification benchmarks, Pascal-VOC and MS-COCO, demonstrate that our solution significantly improved the state-of-the-art results. Our proposed method achieves mAP 91.5% *on MS-COCO and* 97.7% *on Pascal-VOC. Especially, our model outperforms the state-of-the-art models on a small datasets such as the synthetic-fiber rope damage dataset, resulting in a new top score of* 88.2%.

1. Introduction

Multi-label image classification is prominent and challenging topic in computer-vision in which an image may be assigned with multiple labels. As opposed to multi-class classification[15, 24], which is the process of assigning a single label to an image out of set of labels, multi-label image classification enables the simultaneous assignment of multiple labels on a single image, capturing more detailed and nuanced information per image. This capability allows multi-label classification to provide benefits in many real-life problems and tasks for autonomous systems where an understanding of multiple features of the image is required

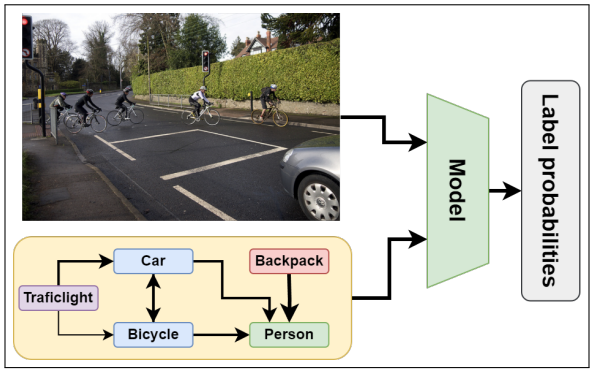


Figure 1. LCNet is a versatile classification method that provides a solution to use learnable label correlation graphs and a novel multi-modal decoder in multi-label image classification.

such as: fault detection, multi-object classification, image and video captioning, and medical diagnosis [6, 8, 12].

Multi-label classification poses some unique challenges that distinguish it from traditional classification methods. One significant challenge is the need to classify a large number of classes (>50), forcing the model to be scalable and capable to separate all the different, often subtle, features of these classes. Another key issue is that often a limited amount of (labeled) training-data is available to train the model. The insufficient amount of data could result in the model over-fitting to the dataset, resulting in lower accuracies on unseen data. Additionally, datasets often suffer from label imbalance, where the probability of the labels appearing in the images could vary a lot between the labels. This could lead the model to prioritize the most common labels at the expense of rarer labels, resulting in a biased prediction, skewing the probability output to the most common labels.

A common approach to multi-label image classification is by converting the problem into a set of binary classification problems to predict whether or not a label is

present. These binary classifiers are typically implemented using the following two strategies: One-vs-One classification and One-vs-Rest classification. The one-vs-One strategy compares each label against every other label in the total set of labels, while the one-vs-rest strategy compares each label against the combined set of all other labels. Modern approaches to multi-label image classification typically enhance their classification performance using a backbone to extract the spatial features from the images, such as convolutional-neural-networks (CNN) [7, 21, 28] and vision-transformers (ViT) [13, 26]. These backbones are paired with a classification-head [11, 17, 27, 33], such as a multi-layered perceptron or transformer[25], creating a two-stage classification framework that efficiently maps extracted spatial features to label predictions logits.

Existing classification-heads using attention-based classification[11, 17, 27, 33] are simple and efficient to implement, however, these approaches often treat the challenges in multi-label classification independently and fail to counter multiple challenges effectively within the classification-head architecture. Additionally, these classification-heads often forget about the semantics of the label, which could provide the model with a understanding of the relations between the labels. Some recent solutions attempted to address these issues by incorporating label semantic graphs [3, 32] to learn the relations between labels to enhance the understanding of the label relations while also improving label imbalance. While effective at capturing the basic label relations, these solutions typically use the graph network only for the first, and initial, classification depth, lacking the learning about the deeper relations. This results in lost opportunities to refine the label interactions and dependencies with a deeper understanding of the relations.

In this paper, we propose LCNet: A classifier model which extends the usage of label semantics together with label embeddings in a new decoder architecture. The model follows the existing two-stage principle of using a backbone to extract the features from the image, and a classifier to determine the prediction logits. The proposed models backbone could be either a conventional CNN or a state-of-the-art ViT. The second-stage leverages its classification performance on a new decoder: the Learnable Label Correlation Decoder (LLCD). The LLCD architecture combines the strengths of attention-based classification methods and label semantic graph learning. As opposed to existing methods, the LLCD is stack-able while fusing the semantics of labels using a graph network with the label embeddings and the spatial features from the image, allowing deeper networks and learning of deeper relations.

The main contributions of this paper are as follows:

- We propose LCNet, a novel trainable multi-label classification framework, which fuses label semantic relations, label embeddings, and spatial features of the image, to predict the probability of logits in the image.
- We introduce a new decoder model which takes in three different modalities: Label relation graph, Label embeddings, and Spatial features of an image. This decoder provides a method to combine these different modalities while allowing decoder stacking to enhance future image classification using deeper networks.
- We evaluated the model on two multi-label benchmark datasets (COCO2014[10], PASCAL-VOC2007[5]) and one dataset containing deeper relations between labels: The Imagery Dataset for Condition Monitoring of Synthetic Fibre Ropes[20], to compare the performance of our proposed method to state-of-the-art methods.

2. Related Work

2.1. Multi-label classification

Multi-label image classification is a well known topic in computer-vision where an image may be assigned with multiple labels. As opposed to multi-class classification this allows for the classification of multiple labels on a single image. Multi-label image classification has increased in popularity recently where new, and improved, methods have focused a lot on three main directions:

Improving loss functions

A key challenge in multi-label classification is the issue of class imbalance, as most images contain only a few present classes while the majority are absent. This imbalance complicates training, as frequent negative examples can bias the model toward overfitting on abundant classes, ignoring less common ones. Solutions like Focal-loss[23] address class-imbalance by down-weighting easy, frequent classes to focus on harder, minority classes. Building on this, Asymmetric-loss[1] further differentiates weighting classes by implementing a positive and negative down-weighting factors, improving model performance by emphasizing challenging examples in both positive and negative classes.

Despite these advancements, current solutions do not fully resolve accuracy dispersion, the variability in per-class accuracies, often resulting in skewed performance across classes. Moreover, they primarily focus on individual class predictions, overlooking label relationships that are crucial in multi-label tasks. This limitation suggests a need for a solution that enhances the classification performance by learning the label relations.

Transformer-based classification

Since the introduction of the transformer architecture[25], a lot of new approaches were found in multi-label classification of images using transformers. A lot of focus in state-of-the-art solutions was on the decoder part of the transformer. A solution presented in the Q2L paper[11] inte-

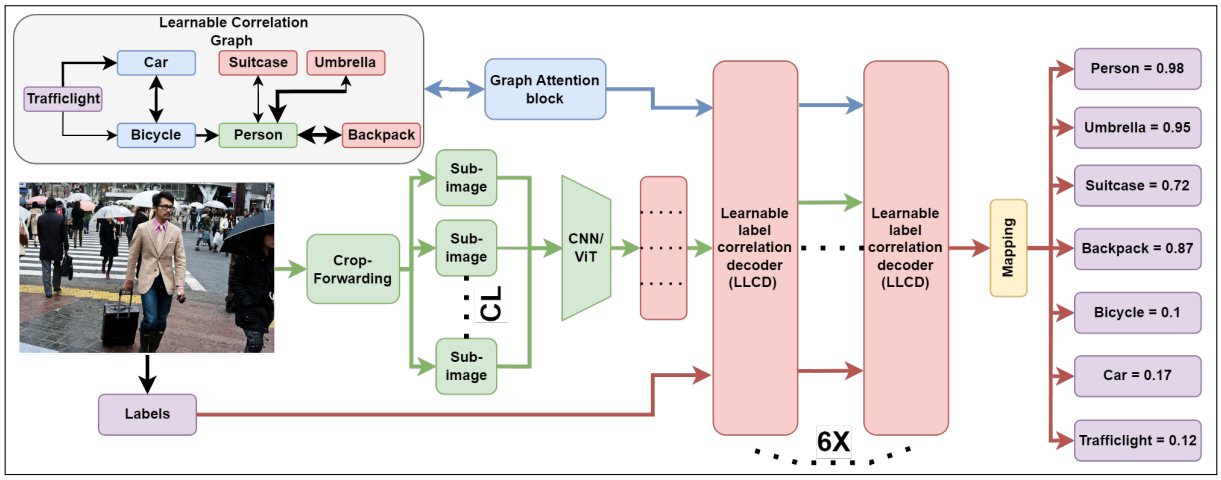


Figure 2. An overview of the model, LCNet, consisting out of three main features: The label correlation graph, The Learnable Label Correlation Decoder (LLCD), and the Crop-Forwarding feature.

grates the label-embeddings as the queries into the decoder to leverage the build-in cross-attention within the decoder. This allows the decoder to learn the relations between the label-embeddings and the images provides as the key/value pairs. The ML-Decoder solution[17] build on this by reducing computational overhead by creating a more scalable classification head, while the ADDS paper[27] introduces a dual-modal decoder that fuses visual and label embeddings to improve context aware classification using deeper decoder layers. The ADDS paper showed that fusing multiple modalities, such as spatial features and textual labels, on deeper layers could increase the performance of the model. Therefore, the solution presented in M3TR[33] attempts in fusing three modalities: spatial visual features, semantic visual features, and the label embeddings into a decoder to create a context aware model.

Graph Networks

As the semantics of labels could be used to learn about the co-occurrence, a new method of multi-label classification was proposed using Graph Convolutional Networks[3, 29] (GCN) and Graph Attention Transformers (GAT)[4]. Unfortunately, integrating these graphs with images features in multi-label classifiers remains challenging. Techniques like those in GATMetric Learning[18] and GATN [32] combine graph and image outputs for classification, but often struggle with learning complex, deeper, label relations. Solutions like Multi-Layered Semantics[19] and Graph-embedding[31] use additional modules such the Semantic Guided Attention (SGA) module and Cross-Modality Transformer(CMT). The SGA module uses the graph to produce correlations scores to extract the prediction logits while the CMT transforms the spatial information from the image into the same semantic space as the label embeddings. While these graph-based approaches advance multi-label classification by embedding label semantics, they often lack the depth and flexibility of transformers, which can stack layers to capture deeper relationships. This gap indicates a need for graph-based networks that, like transformers, can scale in depth to better represent complex, multi-layered label dependencies.

3. Methods

Our proposed framework, as shown in Fig. 2, is a two-stage framework for multi-label classification. The first stage consists out of a backbone for spatial feature extraction, such as a CNN or ViT, coupled with a Graph Attention Transformer (GAT) to learn a label correlation graph. The classification-head, composed of the Learnable Label Correlation Decoder (LLCD) with the linear mapping, forms the second-stage which has the task to determine the probabilities of the logits.

This framework is designed to enhance the semantic understanding among labels and between labels and the features within the image. By learning meaningful relations between the modalities, it improves the models ability to identify related labels and adjust predictions accordingly. Additionally, by incorporating decoder layer stacking, allowing deeper decoder structures , the LLCD enables learning a deeper understanding of the relations between the modalities: correlation graph, the spatial features, and the label embeddings.

3.1. Overview

In general, multi-label classification is to predict whether a label/class is present. The label, or class, can be of several types such as objects (e.g. car, human, chair) or categories (e.g. crumbled, cracked, folded). Formally, multi-label classification can be described as follows: given an input image I and a set of labels L consisting out of N types, the corresponding labels to the image I can be de-

scribed as $L = [l_1, \ldots, l_N]$. Each individual label l_n in the set L indicates wether a specific label n is present, where $l_n \in \{0, 1\}$. The multi-label classification model predicts the probability of the label being present in the image. Therefore, the multi-label classification method can be generalized as $L = [[p(l_n > T_h|I)]] \in \{0, 1\}$, where p is the probability function and T_h is the threshold function.

Our method, as illustrated in Fig. 2, integrates a learnable correlation graph structure with a spatial feature extractor, such as a CNN/ViT, using our Learnable Label Correlation Decoder (LLCD) architecture to learn the relations between the labels, images, and the semantics of the labels. The model takes in an image and extracts the features using a existing CNN feature extractor such as ResNet and TResNet [7, 21] or a ViT such as CvT[26]. The learnable graph together with the Graph Attention Transformer(GAT) block is responsible for learning the semantic relations of the labels. The GAT implemented in this model is inspired by the GAT described in the GATN paper[32] as the correlation graph is initialized by using the word-embedding created by either Word2Vec[14], Glove[16], or One-Hot-Encoding and cosine-similarity. However, in our solution the correlation graph is used directly from the GAT layer without transforming the graph into a CNN layer. The LLCD ensures that encoded results of the different modalities are combined and learned. The main feature of the LLCD is that it allows stacking to improve deeper learning of the relations between the modalities. Additionally, before the CNN/ViT block, the model performs a method called Crop Forwarding (CF) to let the model learn on higher resolution images without re-training the backbone on a higher resolution and minimizing the loss of features within the image.

3.2. Learnable Label Correlation Decoder

Traditionally, transformer-based multi-label image classifiers rely on textual and visual embeddings[17, 27] to identify the specific labels present in the image. These classifiers primarily focus on the correlation between the individual label and spatial features, often overlooking inter-label relations that could provide critical contextual information. To address this limitation, multi-label classifiers which make use of graphs networks try to improve their results by learning about the semantics of the labels and image contents. State-of-the-art graph multi-label classifiers attempt to merge the label semantics and visual features either through a single linear layer[32] or by projecting the graph to an semantic space to combine the embeddings of the visual features with the semantics of the graph[31]. However, these approaches often fall short in capturing complex, deeper, relationships between the modalities.

Our proposed decoder introduces a novel architecture that integrates correlation graphs networks with image features and label queries, while also supporting layer-stacking to capture the deeper relations and determine the per-class probability. The decoder is inspired by the dual-modal decoder[27], which provides a method to integrate image features into stack-able decoders, and the ML-decoder[17], which provides a method to efficiently use the label embeddings in a decoder. However, the dual-modal decoder does not learn inter-label relations to create additional context for the classifier. To overcome the issues of missing contextual information of label relations, our decoder design, as shown in Fig. 3, encapsulates the pros from transformer-based classifiers, such as a layer-stacking, enabling deeper models to capture deeper features, with an additional section to capture the inter-label relations using a correlation graph.

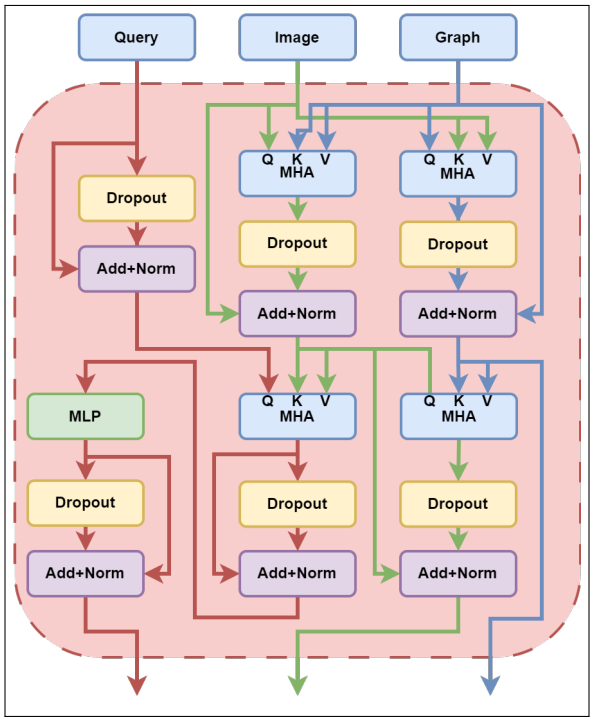


Figure 3. The architecture of the Learnable Label Correlation Decoder (LLCD) consisting out of several connected multi-headed attention blocks and residual networks.

Each layer of the decoder has the following core components:

- **Cross-attention Image-to-Graph:** This component allows for the extraction of relations between the image features space to the graph label adjacency.
- **Cross-attention Graph-to-Image:** This component allows for the extraction of relations between the graph label adjacency matrix and the feature space of the image. These complementary components provide the model with bidirectional insight between the graph (representing label adjacency relations) and image features, enhancing

multi-modal interaction and label semantic understanding.

- **Cross-attention label-Image-Graph features:** This component attends to the image-graph feature space with the label encoding as the query. This is the primary feature which allows the model to learn the spatial features and correlation relations with the label embeddings.
- **Linear Layer label-graph-image** This multi-layer perceptron creates additional relations between the graph-image relations and the labels while also reducing the dimensions of the output.
- **Residual networks and normalizations** The residual and normalization networks help with the gradient flow through the structure.

Formally, we denote inputs of the decoder as "Q_{lbl}", "KV_{img}", and "KV_{graph}". For additional decoder layers, the output of the previous layer ("Q'_{lbl}", "KV'_{img}", and "KV'_{graph}") are connected directly to the input of the next layer. "MHA" represents the multi-headed attention block, "DO" denotes the dropout layer, and "$AddNorm$" as the residual network addition and normalization layer. Then, each LLCD layer can be formulated as:

$$
\begin{aligned}
INT_1 &= AddNorm(DO(Q_{lbl}) + Q_{lbl}),\\
MHA_1 &= MHA(KV_{img}, KV_{graph}, KV_{graph}),\\
INT_2 &= AddNorm(DO(MHA_1) + KV_{img}),\\
MHA_2 &= MHA(INT_1, INT_2, INT_2),\\
INT_3 &= AddNorm(DO(MHA_2) + MHA_2),\\
INT_4 &= MLP(INT_3),\\
MHA_3 &= MHA(KV_{graph}, KV_{img}, KV_{img}),\\
INT_5 &= AddNorm(DO(MHA_3) + KV_{graph}),\\
MHA_4 &= MHA(INT_2, INT_5, INT_5),\\
INT_6 &= AddNorm(DO(MHA_4) + INT_2),\\
Q'_{lbl} &= AddNorm(DO(INT_4) + INT_4),\\
KV'_{img} &= INT_6,\\
KV'_{graph} &= INT_5.
\end{aligned}
\tag{1}
$$

In summary, the LLCD is designed to maximize the interaction between labels, image features, and graph-label relations. Each decoder layer processes and transfers outputs to the next layer, progressively enriching deeper label relations. This results in a new method to improve multi-label classification tasks by capturing and integrating complex multi-modal dependencies.

3.3. Crop Forwarding

To limit computational resources necessary, models are frequently trained on low-resolution images(e.g. 448x448-384x384). Therefore, when high-resolution images are resized to be extracted to spatial features in the backbone, the original spatial resolution is increased, and thus, losing a lot of the smaller details visible in the original image. These lost details could provide the multi-label classifier with more information about the labels or could even contain an entire label. For example, a high-resolution image may contain subtle textures, small objects, or fine patterns, which may become blurred or completely lost when downscaled. This could therefore result in lower accuracy due to the classification head missing some important details.

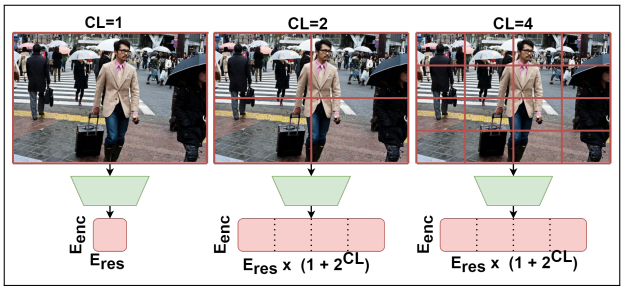


Figure 4. Crop Forwarding allows for lower loss in spatial resolution by using cropped higher resolution images

While training on higher resolutions would solve this issue, it would require more computational cost and memory requirements, especially for large datasets. To address these issues, we propose a new solution: Crop Forwarding (CF) as shown in Fig. 4. CF can be applied to a lot of different image feature extractors such as ViT and CNN, because it only applies on the image. Specifically, given a pre-trained model with a resolution of $M_{img} \times M_{img}$, where $M_{img} \in \mathbb{N}$. Let the input image be $I_{inp} \in \mathbb{R}^{S_{img} * S_{img} * D_{img}}$ where $S_{img} = M_{img} * C_{Layer}$ and where the Crop-Layer $C_{Layer} \in \mathbb{N} > 0$.

As visualized in Fig. 4, an input image is separated into several sub-images, or "crops", of size $S_{img}/C_{Layer} \times S_{img}/C_{Layer}$ depending on the CropLayer (CL). Each of these sub-images, as well as the original input image, are resized to the dimension of the pre-trained model and individually processed by the feature extractor resulting in the spatial features of the images where E_{enc} is the encoding dimension and E_{res} is the spatial feature resolution. Finally, the individual spatial features are then stacked along the spatial resolution axis to be processed by the LLCD.

In summary, the following steps are proceeded in the Crop Forwarding approach:

- **Divide the input image into sub-images:** The input image (I_{inp}) is divided into several sub-images or "crops". The number of sub-images is depended on the Crop-Level (CL).
- **Resize the sub-images:** Resize the images to the resolution of the pre-trained model(backbone).
- **Extract individual image features:** For each sub-image and the input image, the spatial features are extracted us-

Method	Backbone	Resolution	mAP	All						Top 3					
				CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
SRN [34]	ResNet101[7]	224×224	77.1	81.6	65.4	71.2	82.7	69.9	75.8	85.2	58.8	67.4	87.4	62.5	72.9
ResNet-101 [7]	ResNet101[7]	224×224	78.3	80.2	66.7	72.8	83.9	70.8	76.8	84.1	59.4	69.7	89.1	62.8	73.6
CAMD [2]	ResNet101[7]	448×448	82.3	82.5	72.2	77.0	84.0	75.6	79.6	87.1	63.6	73.5	89.4	66.0	76.0
ML-GCN [3]	ResNet101[7]	448×448	83.0	85.1	72.0	78.0	85.8	75.4	80.3	87.2	64.6	74.2	89.1	66.7	76.3
MS-CMA [31]	ResNet101[7]	448×448	83.8	82.9	74.4	78.4	84.4	77.9	81.0	86.7	64.9	74.3	**90.9**	67.2	77.2
ADD-GCN [30]	ResNet101[7]	576×576	**85.2**	84.7	**75.9**	**80.1**	84.9	**79.4**	**82.0**	**88.8**	66.2	75.8	90.3	68.5	**77.9**
Q2L[11]	ResNet101[7]	448×448	**84.9**	84.8	74.5	79.3	**86.6**	76.9	81.5	78.0	69.1	73.3	80.7	**70.8**	75.4
LCNet (Ours)	ResNet101[7]	448x448	**84.9**	**85.2**	72.9	77.8	85.9	76.2	80.8	86.4	**74.1**	**76.5**	89.4	65.6	75.7
ASL [1]	TResNetL[21]	448×448	86.6	87.2	76.4	81.4	88.2	79.2	81.8	91.8	63.4	75.1	92.9	66.4	77.4
GKGNet[29]	ViG-S (1k)[9]	448×448	86.7	86.4	77.1	81.5	87.3	79.7	83.3	-	-	77.0	-	-	78.8
GKGNet[29]	ViG-S (1k)[9]	576×576	87.7	87.0	78.5	82.5	87.6	81.0	84.2	-	-	77.6	-	-	79.3
TResNetL[21]	TResNetL(22k)[21]	448×448	88.4	-	-	-	-	-	-	-	-	-	-	-	-
Q2L[11]	TResNetL(22k)[21]	448×448	89.2	86.3	81.4	83.8	86.5	83.3	84.9	91.6	69.4	79.0	92.9	70.5	80.2
GATN[32]	ResNeXt-101[28]	448x448	89.3	85.1	**89.1**	79.9	84.3	**89.6**	82.0	-	-	-	-	-	-
ML-Decoder[17]	TResNetL(22k)[21]	448×448	90.0	-	-	-	-	-	-	-	-	-	-	-	-
LCNet (Ours)	TResNetL (22k)[21]	448x448	**90.2**	**88.2**	80.9	**84.0**	**90.0**	83.2	**86.0**	**93.8**	**77.7**	**81.9**	**95.1**	**71.9**	**81.9**
Swin-L [13]	Swin-L(22k)[13]	384×384	89.6	**89.9**	80.2	84.8	90.4	82.1	86.1	93.6	69.9	80.0	94.3	71.1	81.1
CvT-w24 [26]	CvT-w24(22k)[26]	384×384	90.5	89.4	81.7	85.4	89.6	83.8	86.6	93.3	70.5	80.3	94.1	71.5	81.3
Q2L[11]	CvT-w24(22k)[26]	384×384	91.3	88.8	**83.2**	**85.9**	89.2	**84.6**	**86.8**	92.8	71.6	80.8	93.9	**72.1**	81.6
LCNet (Ours)	CvT-w24(22k)[26]	384×384	**91.5**	89.6	81.3	84.8	**89.9**	83.2	86.5	**95.1**	**77.1**	**81.4**	**96.1**	71.2	**81.8**

Table 1. Comparison of our method with state-of-the-art models on the MS-COCO2014 dataset, showing performance metrics across different backbones and resolutions.

ing the backbone.

- **Stacking of spatial features:** Once the spatial features are extracted, they are stacked along the spatial resolution axis.
- **Process the image stack:** Process the stacked images in the decoder structure to retrieve the logits.

4. Experiments

Datasets.

The model has been evaluated on three datasets: MS-COCO2014[10], PASCAL VOC2007[5], and the Synthetic Fiber Dataset[20]. The MS-COCO dataset consists out of 80 classes and is regarded as a challenging dataset for multi-label classification. This dataset contains a training-set, including 82081 images, and a verification-set, containing 40137 images. The VOC2007 dataset consists out of 20 different classes and has 3 sets: Training, Validation, and Test. Where the sets are containing 5011, 2476, and 2476 images respectively. The Synthetic Fiber dataset is a dataset which contains 16 classes containing different types of damages to synthetic fiber ropes. This dataset is included in the experiments to show the performance of the model when relations between labels are more available. This dataset contains 6942 images which, for the training, is split into three batches with 60%-20%-20% for the training, validation, and test-sets respectively.

Hardware and implementation details

Following existing papers, the proposed model is trained with different backbones such as ResNet-101 [7] and TResNet-L[21]. The TResNet-L backbone can also use the pre-trained model which has been trained on the ImageNet-21K dataset[22]. If the pre-trained model is used, the type of pre-trained dataset is indicated.

Unless otherwise stated, the settings described below are used for all experiments, For the training of the model an asymmetric focal loss function[1] with $\gamma+ = 0$ and $\gamma- = 4$ and a learning rate of $5e - 5$ is used. Therefore the loss function will favor less false positives/negatives over more true positives/negatives. The image crop size used is $448 * 448$ and the Crop-Level is set to 3. The batch-size is 56 and the maximum epochs allowed 80. All the training was performed on a single RTX4090 GPU.

4.1. Dataset experiments

4.1.1. MS-COCO 2014

The MS-COCO-2014[10] dataset is a commonly-used dataset to benchmark multi-label image classification models. In Tab. 1, a comparison of the performance on the COCO2014 dataset between our LCNet solution and other state-of-the-art classifiers is shown. We have divided the results into 3 different classes related to the backbones used. It should also be noted that the results such as mAP and F1-score are highly influenced by the input resolution and backbone type. Therefore, we have decided to only include state-of-the-art (SOTA) solutions with similar resolutions and backbones and train our model also on these resolu-

Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP	std
ASL[1]	**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	5.333
ADD-GCN(576)[30]	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	4.798
Q2L-TResL [11]	**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	4.974
GATN [32]	99.8	98.9	99.1	98.8	89.5	97.6	97.6	99.3	87.3	98.4	90.7	99.1	99.2	98.5	99.1	88.7	98.4	90.8	99.1	96.3	96.3	4.216
GKGNet[29]	**99.9**	**99.4**	**99.2**	**99.4**	87.0	98.2	**99.1**	**99.6**	88.4	99.5	92.6	**99.5**	**99.5**	98.7	**99.5**	89.5	99.3	90.4	**99.7**	97.2	96.8	4.413
ML-Decoder[17]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	96.6	-
LCNet (ours)	**99.9**	97.1	97.3	99.0	**91.1**	97.6	96.3	97.8	**97.7**	**98.6**	**99.9**	99.1	97.4	96.8	96.8	**97.7**	**99.9**	**95.7**	98.5	**99.9**	**97.7**	**2.003**

Table 2. Comparison of our method with known state-of-the-art models on the PASCAL VOC-2007 dataset. Tested with the TResnet-L backbone and a image resolution of 448x448

tions. As shown in Tab. 1, we can see that LCNet performs the best on all the mAP scores with the different backbones. The model also shows high OF1 and CF1 scores. In particular in comparison with other SOTA with the TResNet-L backbone. There it outperforms the GATN[32], Q2L[11], and ML-decoder[17] with 1.13%, 1.0%, and 0.23% respectively.

4.1.2. Pascal-VOC2007

Similary to MS-COCO, Pascal-VOC2007 forms a major benchmarking dataset for multi-label image classification. Each image in VOC contains up to 20 different object categories. The results on VOC2007 are shown in Tab. 2. The results shown are created on the TResNet-L backbone pretrained on 22K image-net [22] with the standard image resolution of 448x448. Using the CvT ViT backbone resulted in reduced result due to dataset having less complexity and not enough training-data for a ViT to have an advantage. We can see that our method achieves the best mAP performance along with the lowest class accuracy deviation. We can observe that we have the largest margin between the results, that could indicate that the label relations are easier identifiable in the VOC2007 dataset. In general, our model outperforms GATN[32], GKGNet[29], and ML-decoder[17] with 1.5%, 0.9%, and 1.1% respectively.

4.1.3. Synthetic Fiber dataset

The Synthetic Fiber dataset[20] is a dataset collected to use AI to classify damages of synthetic fiber ropes. This dataset is included in this study to show the performance of our model on closely related classes. Due to the graph structure in our model, the relations between damage classifications should overall result in better classification of the damages. Due to time constraints, we could only test with two of the best performing SOTA models: Q2L and MLDecoder. All models are tested with the same backbone (TResNet-L) and resolution (448x448) to ensure a even comparison. For this dataset, a Croplevel of 4 is chosen because the dataset images have higher resolutions and the spatial features indicating the damages are relatively small. The results shown in Tab. 3, show that our model performs the best with an mAP improvement of 5.4% and 4.3% on MLdecoder and Q2L respectively. The F1-scores show also a great improvement over the other SOTA models resulting, in combination with a higher mAP, in less incorrect classifications.

Model	mAP	F1	F1 (top3)
MLDecoder[17]	83.7	80.7	81.5
Q2L[11]	84.6	83.9	85.2
LCNet (Ours)	**88.2**	**86.9**	**88.4**

Table 3. Comparison of our method with known state-of-the-art models on the Synthetic Fiber Dataset. Tested with the TResnet-L backbone and a image resolution of 448x448

4.2. Ablation study: Effectiveness of the Learnable Label Correlation Decoder

To evaluate the effectiveness of the number of decoder layers, the model is trained using different decoder depths. To ensure a fair comparison, we standardized the backbone to TResnet-L[21] with a resolution of 448x448 pretrained on imagenet-22k[22]. The Croplevel for this test is set to 1 to reduce the influence of this parameter in the result. We selected the MS-COCO2014[10] dataset for this ablation-study. This dataset was chosen because it will result in the largest number of classes and therefore it would require the most interactions between the graph-network and the image features.

The results, shown in Tab. 4, demonstrate that using more decoder layers progressively increase the performance. Specifically, the model with 6 decoder layers achieved the highest mAP of 89.8, alongside an F1-score of 86.0 and a top3 F1-score of 81.4. Notably even a modest increase from 1 layer to 2 decoder layers resulted in a noticeable performance boost. Overall, the model had an maximum mAP increase of 0.79% with 6 decoder layers over the default single layer.

Model	mAP	F1	F1 (top3)
Decoder 1x	89.1	85.4	81.0
Decoder 2x	89.5	85.7	81.2
Decoder 4x	89.7	85.9	81.2
Decoder 6x	89.8	86.0	81.4
Decoder 8x	89.8	85.9	81.3

Table 4. Ablation study with comparisons of different amount of LLCD layers

However, increasing the decoder even more did not result in noticeable performance gains. This could be a result

of the model starting to overfit on the data, or the possibility that the maximum label-relations are modeled and have reached an optimal level for the dataset.

4.3. Ablation study: Effectiveness of the Correlation/Semantic Graph

We conducted an ablation study to assess the contribution of the correlation graph in modeling label semantics and interactions. This was tested by evaluating the LLCD with and without the label semantic graph approach across multiple decoder depths. In these experiments the MS-COCO dataset was used with the TResNet-L backbone and with a CropLevel of 1.

The results without the semantic graph, presented in Tab. 5, compared with the results including the semantic graph, presented in Tab. 4, indicate a significant improvement in all metrics when the correlation graph is included. For example, the 6-layer configuration without the graph achieved an mAP of 87.9, compared to 89.8 with the graph included, underscoring the benefit of leveraging inter-label relationships in capturing label semantics. These findings highlight that the correlation graph enhances the model's contextual awareness, resulting in improved classification performance.

Model	mAP	F1	F1 (top3)
Normal	**90.2**	**86.0**	**81.9**
Decoder 1x	85.3	81.5	80.6
Decoder 2x	86.1	81.9	81.0
Decoder 4x	86.7	82.4	81.7
Decoder 6x	87.9	83.9	82.6

Table 5. Ablation study with comparisons of different amount of LLCD layers without a graph

4.4. Ablation study: Effects of mirroring image and graph inputs

To examine the impact of mirroring image and graph inputs, we conducted experiments in which both inputs were mirrored, analyzing the effect on classification performance. Mirroring inputs can alter spatial features and potentially disrupt the alignment between image content and label semantics, thus affecting model predictions. In these experiments the MS-COCO dataset was used with the TResNet-L backbone and with a CropLevel of 1.

Model	mAP	F1	F1 (top3)
Normal	**90.2**	**86.0**	**81.9**
Decoder 1x	37.5	24.1	17.4
Decoder 2x	38.9	23.8	17.6
Decoder 4x	39.0	25.7	24.3
Decoder 6x	42.6	27.4	26.2
Decoder 8x	37.8	24.8	18.3

Table 6. Ablation study with comparisons of mirrored image and graph inputs

As shown in Tab. 6, mirrored inputs yielded lower performance across all metrics compared to the standard configuration. While increasing the decoder layers increased performances, it did not result in reaching the accuracy levels of non-mirrored inputs.

4.5. Ablation study: Effectiveness of Crop-Forwarding

To evaluate the effectiveness of Crop-Forwarding, the model is trained using different values of Crop-levels. To ensure a fair comparison, we standardized the backbone to TResnet-L[21] with a resolution of 448x448 pretrained on imagenet-22k[22]. The decoder depth is set to 1 layer to minimize the influence of the LLCD on the number of Crop-layers. We selected the Pascal-VOC2007[5] dataset for this ablation-study.

The results, shown in Tab. 7, demonstrate that using Crop-forwarding enhanced the model performance. As the Croplevel increases, the model exhibits improved accuracy metrics, specifically in the mAP and F1-scores, particularly at Croplevels 2 and 3. In comparrison to a Croplevel 1, levels 2 and 3 improved the result with 0.5% and 0.72% respectively. Increasing the Croplevels further had diminishing results due to potential loss of context for this particular dataset. This is because the images used in pascal-VOC have a limited resolution. Higher Croplevels ($\geq$ 4) have provided better results in datasets with higher-resolution images such as the synthetic fiber dataset[20].

Model	mAP	F1	F1 (top3)
Croplevel 1	93.37	90.5	96.5
Croplevel 2	93.84	90.9	98.3
Croplevel 3	94.04	91.0	98.5
Croplevel 4	91.5	88.6	96.4

Table 7. Ablation study with comparisons of different Crop-Forwarding levels

5. Conclusion

This paper introduces the Learnable Label Correlation Network (LCNet), a multi-label classification model that combines a correlation graph with spatial features and label

queries. Our stackable Learnable Label Correlation Decoder (LLCD) enables deeper relation learning through label semantics, while Crop-Forwarding allows the model to make use of higher resolution images, reducing loss of spatial resolution data, without increasing the resolution of the CNN or ViT. LCNet significantly outperformed all previous state-of-the-art results on MS-COCO and Pascal-VOC. It also resulted in better results on the synthetic fiber dataset, where the semantic relations are even closer. The improvements highlight the model's robustness and potential applicability in complex, multi-label environment, where leveraging label semantics can provide an additional edge in detection.

References

[1] Emanuel Ben-Baruch, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification, 2021. 2, 6, 7

[2] Zhao-Min Chen, Xiu-Shen Wei, Xin Jin, and Yanwen Guo. Multi-label image recognition with joint class-aware map disentangling and label correlation embedding. *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 622–627, 2019. 6

[3] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5177–5186, 2019. 2, 3, 6

[4] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs, 2021. 3

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 2, 6, 8

[6] Dhatri Ganda and Rachana Buch. A survey on multi label classification. *Recent Trends in Programming Languages*, 5 (1):19–23, 2018. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2, 4, 6

[8] Francisco Herrera, Francisco Charte, Antonio J Rivera, María J Del Jesus, Francisco Herrera, Francisco Charte, Antonio J Rivera, and María J del Jesus. *Multilabel classification*. Springer, 2016. 1

[9] Bencheng Liao, Xinggang Wang, Lianghui Zhu, Qian Zhang, and Chang Huang. Vig: Linear-complexity visual sequence learning with gated linear attention, 2024. 6

[10] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 2, 6, 7

[11] Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Query2label: A simple transformer way to multi-label classification, 2021. 2, 6, 7

[12] Weiwei Liu, Haobo Wang, Xiaobo Shen, and Ivor W Tsang. The emerging trends of multi-label learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(11): 7955–7974, 2021. 1

[13] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021. 2, 6

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. 4

[15] Venkatesh N Murthy, Vivek Singh, Terrence Chen, R Manmatha, and Dorin Comaniciu. Deep decision network for multi-class image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2240–2248, 2016. 1

[16] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 4

[17] Kirill Prokofiev and Vladislav Sovrasov. Combining metric learning and attention heads for accurate and efficient multi-label image classification, 2022. 2, 3, 4, 6, 7

[18] Kirill Prokofiev and Vladislav Sovrasov. Combining metric learning and attention heads for accurate and efficient multi-label image classification. In *VISIGRAPP*, 2022. 3

[19] Xiwen Qu, Hao Che, Jun Huang, Linchuan Xu, and Xiao Zheng. Multi-layered semantic representation network for multi-label image classification. *International Journal of Machine Learning and Cybernetics*, 14(10):3427–3435, 2023. 3

[20] Anju Rani, Daniel O Arroyo, and Petar Durdevic. Imagery dataset for condition monitoring of synthetic fibre ropes. *arXiv preprint arXiv:2309.17058*, 2023. 2, 6, 7, 8

[21] Tal Ridnik, Hussam Lawen, Asaf Noy, and Itamar Friedman. Tresnet: High performance gpu-dedicated architecture, 2020. 2, 4, 6, 7, 8

[22] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses, 2021. 6, 7, 8

[23] T-YLPG Ross and GKHP Dollár. Focal loss for dense object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2980–2988, 2017. 2

[24] Vishal Shah and Neha Sajnani. Multi-class image classification using cnn and tflite. *International Journal of Research in Engineering, Science and Management*, 3(11):65–68, 2020. 1

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 2

[26] Haiping Wu, Bin Xiao, Noel C. F. Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22–31, 2021. 2, 4, 6

[27] Shichao Xu, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Zhu Qi. Open vocabulary multi-label classification with dual-modal decoder on aligned visual-textual features, 2023. 2, 3, 4

[28] I. Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *CoRR*, abs/1905.00546, 2019. 2, 6

[29] Ruijie Yao, Sheng Jin, Lumin Xu, Wang Zeng, Wentao Liu, Chen Qian, Ping Luo, and Ji Wu. Gkgnet: Group k-nearest neighbor based graph convolutional network for multi-label image recognition. *ArXiv*, abs/2308.14378, 2023. 3, 6, 7

[30] Jin Ye, Junjun He, Xiaojiang Peng, Wenhao Wu, and Yu Qiao. Attention-driven dynamic graph convolutional network for multi-label image recognition. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI*, page 649–665, Berlin, Heidelberg, 2020. Springer-Verlag. 6, 7

[31] Renchun You, Zhiyao Guo, Lei Cui, Xiang Long, Yingze Bao, and Shilei Wen. Cross-modality attention with semantic graph embedding for multi-label classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34 (07):12709–12716, 2020. 3, 4, 6

[32] Jin Yuan, Shikai Chen, Yao Zhang, Zhongchao Shi, Xin Geng, Jianping Fan, and Yong Rui. Graph attention transformer network for multi-label image classification. *arXiv preprint arXiv:2203.04049*, 2022. 2, 3, 4, 6, 7

[33] Jiawei Zhao, Yifan Zhao, and Jia Li. M3tr: Multi-modal multi-label recognition with transformer. In *Proceedings of the 29th ACM International Conference on Multimedia*, page 469–477, New York, NY, USA, 2021. Association for Computing Machinery. 2, 3

[34] Feng Zhu, Hongsheng Li, Wanli Ouyang, Nenghai Yu, and Xiaogang Wang. Learning spatial regularization with image-level supervisions for multi-label image classification, 2017. 6

Supplementary pages

A. Performance of sub-parts

The performance of the proposed model's individual components was evaluated on the MS-COCO2014 dataset. We analyzed the impact of label queries, Crop-Forwarding (CF), and the Graph Attention Transformer (GAT) graph on key metrics, including mean Average Precision (mAP) and F1 score. The results are summarized in Tab. 8.

mAP	F1	Label query	CF	GAT graph	Mem
86.13	81.1	**1**	0	0	5401
89.4	85.3	**1**	0	**1**	8183
87.42	81.3	**1**	**1**	0	12461
90.2	86.0	**1**	**1**	**1**	20784

Table 8. Test results of the performance of each individual part on the MS-COCO2014 dataset

- **Base Model (Label Query + Image Only):** Using only Labels as queries and the spatial features from the image, the model achieves an mAP of 86.13 and an F1 score of 81.1. This configuration forms the foundation of the model and demonstrates the effectiveness of aligning label queries with spatial features.
- **Impact of Adding GAT Graph:** Introducing the GAT Graph significantly enhances both metrics: mAP and F1-score. the mAP increases to 89.4 ($+3.80\%$ from baseline). F1: Improves to 85.3 ($+5.17\%$ from baseline).
- **Impact of Adding Crop-Forwarding (CF):** Incorporating CF alone results in an mAP of 87.42 ($+1.50\%$ from baseline) and an F1 score of 81.3 ($+0.25\%$ from baseline).
- **Best Configuration (LCNet):** Combining Label Query, CF, and GAT Graph achieves the best performance. The mAP increases to 90.2 ($+4.73\%$ from baseline), while the F1-score increases to 86.0 ($+6.04\%$ from baseline).

Conclusion The analysis above demonstrates the individual effect of each solution and the effect of combining Label Query, CF, and GAT Graph. While Label Query serves as the backbone, GAT Graph offers substantial gains by modeling label dependencies, and CF enhances spatial feature utilization. When combined, these components yield a robust and high-performing multi-label classification model, achieving a 4.73% increase in mAP and a 6.04% increase in F1 score compared to the baseline.

B. Computational requirements comparison

We evaluate the computational requirements of our proposed model by analyzing memory consumption, training speed (T-Speed), and inference speed (I-Speed) on the MS-COCO and Pascal-VOC datasets. The T-Speed is defined as the training time per batch, while the I-speed is defined as the time the model takes to generate an output from a single image as the input. This analysis highlights the impact of varying the number of Learnable Label Correlation Decoder (LLCD) layers and Crop-Forwarding (CF) levels on computational demands.

Dataset: MS-COCO2014 The results in Tab. 9 demonstrate how increasing LLCD layers and CF levels affects memory usage, training, and inference speeds on the MS-COCO2014 dataset.

LLCD	CF	Memory (MB	T-Speed (s)	I-speed (s)
1	1	11309	0.341	0.033
2	1	14440	0.384	0.043
4	1	16248	0.362	0.035
6	1	18457	0.367	0.035
6	2	19991	0.454	0.055
6	3	20784	0.748	0.063
6	4	23262	3.425	0.087

Table 9. Computational speeds and memory usage on MS-COCO2014

Observations:

- **Memory Usage:** As expected, memory consumption grows with the number of LLCD layers and CF levels. Increasing LLCD layers from 1 to 6 results in a 63.2% increase in memory, while adding CF levels from 1 to 4 results in an additional 25.9% increase.
- **Training Speed (T-Speed):** The T-Speed remains efficient with increased LLCD layers, rising from 0.341 seconds at 1 layer to 0.367 seconds at 6 layers (a 7.6% increase). However, higher CF levels significantly impact training time. For instance, CF level 4 leads to a 833% increase in T-Speed compared to CF level 1 with 6 LLCD layers.
- **Inference Speed (I-Speed):** The I-Speed increases modestly with deeper LLCD layers but grows significantly with CF levels. For 6 LLCD layers, increasing CF levels from 1 to 4 causes a 148.6% increase in inference time (from 0.035s to 0.087s).

Dataset: Pascal-VOC2007 The analysis for the Pascal-VOC2007 dataset, as shown in Tab. 10, follows a similar pattern as the MS-COCO computational requirements, but with a reduced computational demand, due to the smaller scale of the dataset.

LLCD	CF	Memory	T-Speed	I-speed
1	1	4376	0.264	0.046
2	1	6142	0.249	0.065
4	1	7454	0.335	0.084
6	1	8216	0.465	0.17
6	2	12073	0.586	0.318
6	3	14355	0.743	0.458
6	4	19428	1.535	0.758

Table 10. Computational speeds and memory usage on Pascal-VOC2007

Observations:

- **Memory Usage:** Memory requirements grow similarly to the MS-COCO dataset. Increasing LLCD layers from 1 to 6 leads to an 87.7% increase in memory, and increasing CF levels from 1 to 4 results in a 136.3%increase for the same number of layers.
- **Training Speed (T-Speed):** Adding LLCD layers slightly impacts training speed, while CF levels have a greater influence. Moving from CF level 1 to 4 for 6 LLCD layers increases T-Speed from 0.465 seconds to 1.535 seconds, representing a 230.1% increase.
- **Inference Speed (I-Speed):** The impact on inference speed mirrors that seen in MS-COCO with only a small increase in speed with more decoder layers. For 6 LLCD layers, moving from CF level 1 to 4 increases inference time from 0.170 seconds to 0.758 seconds, a 345.9% increase..

Conclusion: The results illustrate the trade-offs between performance (Accuracy, speed, and Precision) and computational demands. While deeper LLCD layers and higher CF levels enhance the model's ability to capture complex label relations and spatial features, they also require greater memory and computational resources. Especially, Crop-Forwarding has a large impact on computation-time, which is likely due to the serialization of the backbone function to limit memory usage.

C. Resulting: GAT-graphs

To illustrate the impact of the GAT and semantic graph, we visualized the graph structures generated from the MS-COCO and Pascal-VOC datasets. The GAT enables deeper semantic learning, allowing the graph to capture diverse types of information, including correlation and co-occurrence patterns. Nodes with stronger semantic relationships are positioned closer to each other, while the thickness of edges between nodes represents the strength of these relationships, with thicker edges indicating more robust connections.

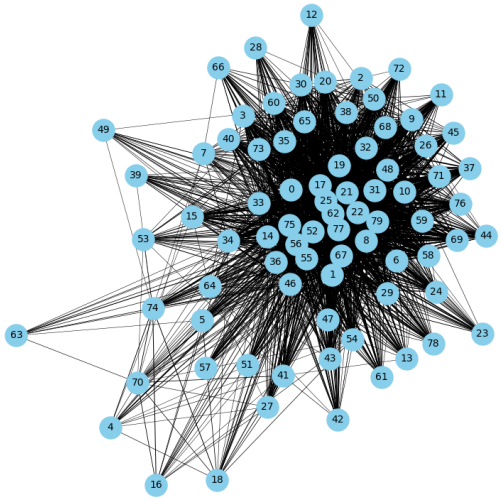


Figure 5. Graph visualization result of the deeper label semantic relations from the MS-COCO2014 dataset

For the MS-COCO dataset, the graph comprises 80 nodes, as shown in Fig. 5, resulting in a dense and complex structure. In contrast, the Pascal-VOC dataset, with only 20 classes, produces a more streamlined and less cluttered graph, as depicted in Fig. 6. Notably, in both cases, some nodes lack edges, reflecting the absence of any significant relationships between certain node pairs. This highlights the ability of the graph to adaptively represent the underlying semantic structure of the datasets.

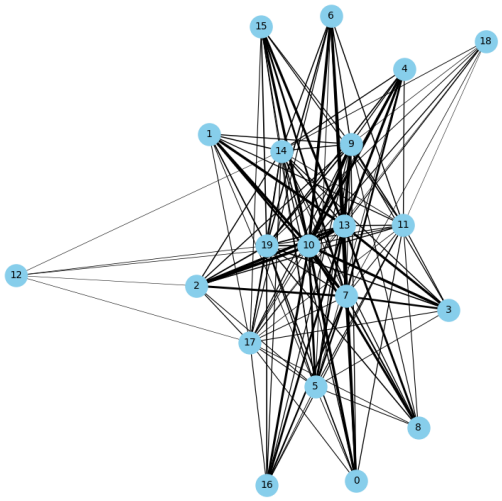


Figure 6. Graph visualization result of the deeper label semantic relations from the Pascal-VOC2007 dataset

D. Attention maps

Attention maps in image transformers provide insights into how the model processes and focuses on different regions of the image when making predictions. These maps visualize the attention weights assigned to various spatial locations in the image, showing which areas of the input the model attends to more strongly during the encoding and decoding processes.

In Figure Fig. 7, we present the result of an attention map from the 6th encoder layer of the LLCD with an input image of a bear Fig. 8. The model is trained on the MS-COCO dataset and, the attention map highlights how the model identifies and attends to specific regions, such as the bear shape located in the center of the image. The concentration of attention around the bear indicates that the model recognizes the shape as a critical part of the image to classify the object as a bear.

Additional examples can be found below in Fig. 9 and Fig. 10. These examples show more of the decoder layer attention map results on the two datasets (MS-COCO, VOC2007).

Figure 7. Attention map output of the 6th encoder layer, detecting the bear shape in the center

Figure 8. Input image of model with the class: Bear

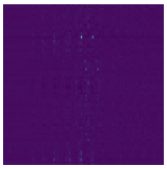
E. Model limitations

While LCNet demonstrates notable improvements in multi-label classification, there are several areas where its performance and applicability could be further refined, offering exciting opportunities for continued research and development.

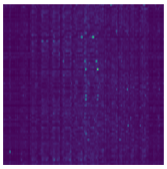
Scalability to Large Datasets: LCNet performs well on the MS-COCO and Pascal-VOC datasets, but as the model scales to larger datasets, the computational demands, particularly related to memory usage in the classification head and the LLCD decoder, could become a limiting factor.

Adaptation to Embedded Systems: LCNet's memory usage and computational complexity, driven by the deep LLCD architecture and the need for fast detection, may pose challenges for deployment on embedded or resource-constrained, real-time applications and systems.

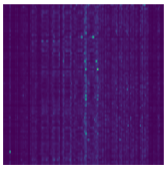
However, these limitations offer a promising avenue for further research into lightweight, optimized versions of the model, enabling it to run effectively on embedded systems and large datasets while maintaining the quality of predictions.



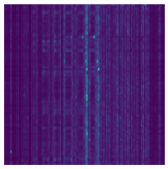
(a) Layer 1



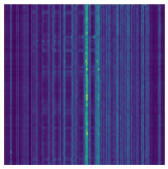
(b) Layer 2



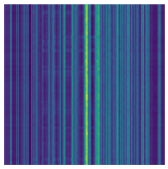
(c) Layer 3



(d) Layer 4



(e) Layer 5



(f) Layer 6

(g) Layer 1

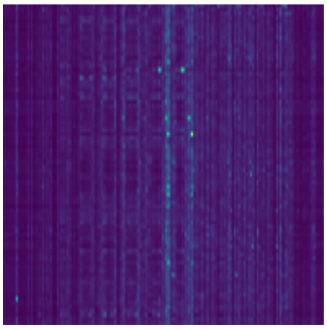
(h) Layer 2

(i) Layer 3

(j) Layer 4

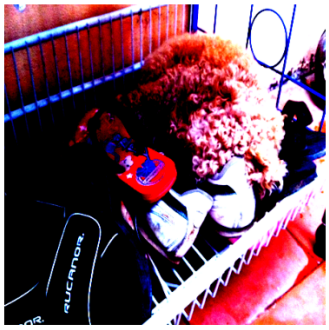
(k) Layer 5

(l) Layer 6



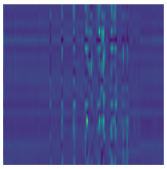
(m) Mean attention map

(n) Attention map overlay

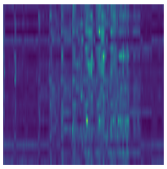


(o) Original image

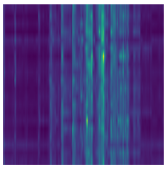
Figure 9. COCO attention map for visualizing a dog



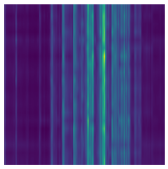
(a) Layer 1



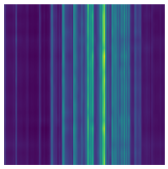
(b) Layer 2



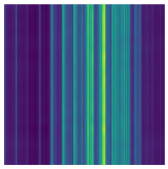
(c) Layer 3



(d) Layer 4



(e) Layer 5



(f) Layer 6

(g) Layer 1

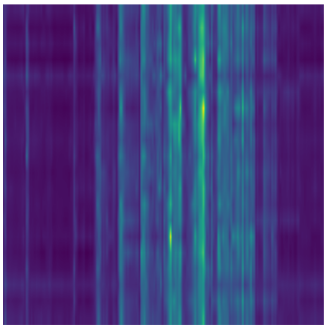
(h) Layer 2

(i) Layer 3

(j) Layer 4

(k) Layer 5

(l) Layer 6



(m) Mean attention map

(n) Attention map overlay

(o) Original image

Figure 10. VOC attention map for classification of a plane