PhD-FSTM-2024-058
The Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 12/07/2024 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Inder Pal SINGH
Born on 17 January 1988 in Palwal Haryana (INDIA)

TOWARDS COMPACT YET EFFECTIVE MULTI-LABEL IMAGE CLASSIFICATION: FROM A SINGLE DOMAIN TO MULTIPLE DOMAINS

Dissertation defence committee

Dr Djamila AOUADA, dissertation supervisor
Assistant Professor, Université du Luxembourg

Dr Andreas HEIN, Chairman
Associate Professor, Université du Luxembourg

Dr Enjie GHORBEL, Vice Chairman
Professor, University of Manouba, TUNISIA

Dr Samia AINOUZ, Member
Professor, INSA Rouen Normandie, FRANCE

Dr Thomas HARTMANN, Member
Head of software engineering, DataThings, Luxembourg

In the loving memory of

my parents

Acknowledgements

Many people have contributed directly or indirectly towards the completion of this thesis. First and foremost, I would like to express my sincere gratitude to my supervisors Prof. Dr. Djamila AOUADA and Prof. Dr. Enjie GHORBEL. This research work would have not been possible without their continuous support. Their insightful feedback and rigorous mentorship have been a constant source of inspiration and my all-around growth. I would also like to express my gratitude to Dr. Anis KACEM for his suggestions, remarks, and discussions throughout this journey. Additionally, I would explicitly like to appreciate the exceptional emotional support from Djamila, Enjie, and Anis during the unfortunate and unpredictable circumstances that I have been through during this journey. Their understanding and constant presence have been a source of strength and comfort during challenging times.

I extend my sincere appreciation to the defense committee members of my thesis, Prof. Dr. Andreas HEIN, Prof. Dr. Samia AINOUZ, and Dr. Thomas HARTMANN, for generously dedicating their time and expertise to review the works submitted as a part of this thesis. I would also like to sincerely appreciate Dr. Eva LAGUNAS TARGARONA, member of the comité d'encadrement de thése (CET), for her valuable feedback throughout this journey. A special thanks to Dr. Kassem AL ISMAEIL for his encouragement and guidance in moving and settling me to Luxembourg to pursue my doctoral studies. Special thanks to my friends and colleagues, with whom I shared my office, Nesryne MEJRI and Ahmet Serdar KARADENIZ for showing the highest level

of patience to bear my presence and listening to my annoying stories. I believe you enjoyed it as much as I did.

In addition, I am indebted to all the current and former members of the Computer Vision, Imaging, and Machine Intelligence (CVI^2) research group including Renato, Oyebade, Miguel, Vincent, Mohamed Adel, Abd El Rahman, Elona, Romain, Carl, and Matthieu. The time that I have spent with them, both within and outside the University, and the discussions have made my journey more enjoyable.

I would also like to sincerely acknowledge the funding sources including Fonds National de la Recherche (FNR) and POST Luxembourg for their generous financial support. The works presented in this thesis are funded under the projects: BRIDGES2020/IS/14755859/MEET-A/Aouada, BRIDGES2021/IS/16353350/FaKeDeTeR and UNFAKE ref.16763798.

I would like to dedicate this work to the memory of my parents, whose love and blessings continue to inspire me every day. Though they are no longer with us, their belief in me has been one of the main reasons for the success of my work. Additionally, I take this opportunity to thank my little girl Mishka whose smiling face has been one of my greatest strengths. Last but not least, my better half Mamta, thank you for your unconditional love and support during all times. I love you.

To everyone mentioned above and to those whose names I may have unintentionally missed, thank you from the bottom of my heart. This thesis would not have been possible without your support, encouragement, and belief in me.

Index

List of Abbreviations

MLIC	Multi-label Image Classification
DNN	Deep Neural Networks
DL	Deep Learning
CNN	Convolutional Neural Networks
GCN	Graph Convolutional Networks
HOG	Histogram of Oriented Gradients
SIFT	Scale-Invariant Feature Transform
SVM	Support Vector Machines
kNN	k-Nearest Neighbors
UDA	Unsupervised Domain Adaptation
mAP	Mean Average Precision
NLP	Natural Language Processing
CP	Average per-Class Precision
CR	Average per-Class Recall

CF1	Average per-Class F1-score
OP	Average Overall Precision
OR	Average Overall Recall
OF1	Average Overall F1-score
ASL	Asymmetric Loss
GRL	Gradient Reversal Layer

List of Figures

List of Tables

Abstract

Multi-label Image Classification (MLIC) is an active research topic within the Computer Vision community. Its objective is to simultaneously identify the presence or absence of multiple objects within an image. Thanks to its practical usefulness, MLIC finds its way into numerous fields of applications such as human action recognition, multi-attribute predictions, and semantic segmentation. With the widespread availability of large-scale labeled datasets and the recent advancements in deep learning, numerous methods have achieved remarkable performance for the task of MLIC. Despite their proven success, these methods usually employ very deep neural networks leading to cumbersome architectures. This strategy practically limits their applicability in a memory-constrained scenario. Additionally, existing MLIC methods usually assume that the test data comes from the same domain as the training data. However, this might not always be true which leads to poor generalization of these methods on the images from unseen domains. To overcome this challenge, commonly known as domain-shift, very few methods have been proposed that extend the concept of Unsupervised Domain Adaptation (UDA) from single-label classification to multi-label classification. However, due to the inherent differences between the two problems, this extension might yield sub-optimal results. Additionally, these methods also suffer from the use of cumbersome network architectures.

In this thesis, we propose to tackle the aforementioned challenges in MLIC, going from the simple scenario where a single-label domain is considered to a more challenging setup involving a cross-domain setting. In the first part of this thesis, we aim to efficiently and effectively model the relationship between multiple objects while keeping a moderate-size architecture for the general task of MLIC. Specifically, we make use of Graph Convolutional Networks (GCN) to model the label relationships using an adaptive graph learning strategy. In the second part of this work, we focus on tackling the domain shift that degrades the performance of MLIC methods. For that purpose, we propose novel UDA approaches specifically tailored to the task of MLIC. The proposed solutions are evaluated on several benchmarks to demonstrate their effectiveness with respect to the state-of-the-art methods.

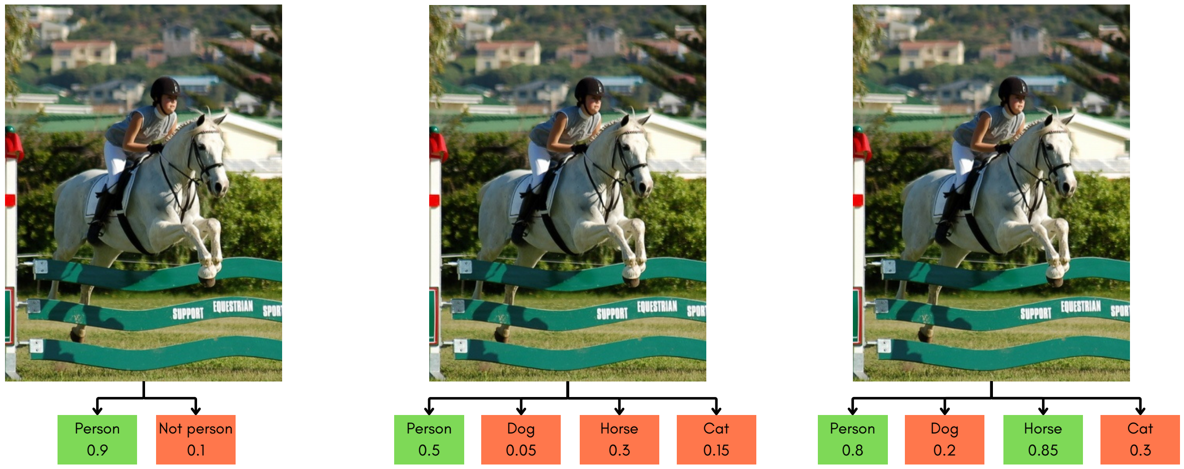
Chapter 1

Introduction

Humans rely on their visual senses to perceive and interpret visual information. They identify the content of a scene by recognizing various visual elements such as patterns, shapes, colors, textures, and other distinctive features inherent in the image. Furthermore, humans also use their prior knowledge and experience to categorize images into relevant categories also termed as labels or classes. Overall, human visual interpretation is a complex cognitive task that depends on sensory perception, pattern recognition, and cognitive reasoning. Hence, replicating such a subtle process using machines poses considerable challenges due to the complexity of the human cognition system.

In recent decades, the computer vision research community has shown a huge interest in developing systems that are capable of identifying the contents of images. This research field, broadly termed as *image classification*, aims to categorize images into predefined classes or labels based on their visual content. Given its practical usefulness, image classification finds its application in numerous fields of interest such as object detection [1], action recognition [2], [3], segmentation [4], human pose estimation [5], [6], video classification [7], and object tracking [8].

In the literature, the problem of image classification is mostly formulated either as



(a) Binary classification (b) Multi-class classification (c) Multi-label classification

Figure 1.1: Types of image classification.

a binary or multi-class classification [9], [10]. Binary classification aims at predicting whether a specific object is present or not. For example, as shown in Figure 1.1a, the goal is to determine if a *person* is present in an image or not. In multi-class classification, the task is to determine which object from a set of predefined classes is present. For instance, in Figure 1.1b, the aim is to identify whether the image contains a *person*, a *dog*, a *horse*, or a *cat*. In this case, the highest probability indicates the classifier prediction.

Nevertheless, some practical scenarios may require the simultaneous recognition of multiple objects/entities in a single image. For instance, identifying multiple human attributes such as gender and clothing style etc. can be useful for person re-identification [11]. Similarly, several other applications such as scene classification [12], [13], multi-object recognition [1], [14], and deepfake detection [15] necessitate simultaneous prediction of multiple objects or attributes in an image. Thus, Multi-label Image Classification (MLIC) emerges as a third alternative for such real-world applications. Instead of predicting a single class, MLIC predicts the presence or absence of a set of objects within an image simultaneously. For example, as demonstrated in Figure 1.1c, it might be more suitable for several use-cases to predict all the objects that are present in the given image, namely a *person* and a *horse*. Herein, in this thesis, we focus mainly on the topic of multi-label image classification. Despite the widespread availability of MLIC approaches in the literature, there exist still several open issues to be solved, hindering their use in several real-world applications. In this chapter, we start by describing the motivation and scope of the present thesis. Then, we detail the challenges in MLIC that are addressed in this research work. Hereafter, we present our main objectives and contributions to the field of MLIC. We finally conclude this introductory chapter by providing the list of publications that have resulted from these PhD investigations.

1.1 Motivation and Scope

Multi-label Image Classification (MLIC) is an active research topic in computer vision. Earlier approaches for MLIC often rely on handcrafted features such as Histogram of Oriented Gradients (HOG) [16] or Scale-Invariant Feature Transform (SIFT) [17], coupled with machine learning algorithms such as Support Vector Machines (SVM) [18] or k-Nearest Neighbors (k-NN). While these methods have shown promising results, heuristically designing optimal features remains extremely challenging. As an alternative, Deep Learning (DL)-based approaches [9], [19]–[21] have been recently investigated. These approaches leverage neural networks including Convolutional Neural Networks (CNN), to automatically learn image representations from raw data by minimizing a loss function, usually encoding the information of misclassification rate. As a result, DL-based methods have achieved improved performance on publicly available large-scale datasets.

Nevertheless, their impressive performance comes at the cost of very large architectures that are unsuitable for memory-constrained environments. Hence, in the first part of this thesis, we focus on investigating adequate mechanisms in order to achieve

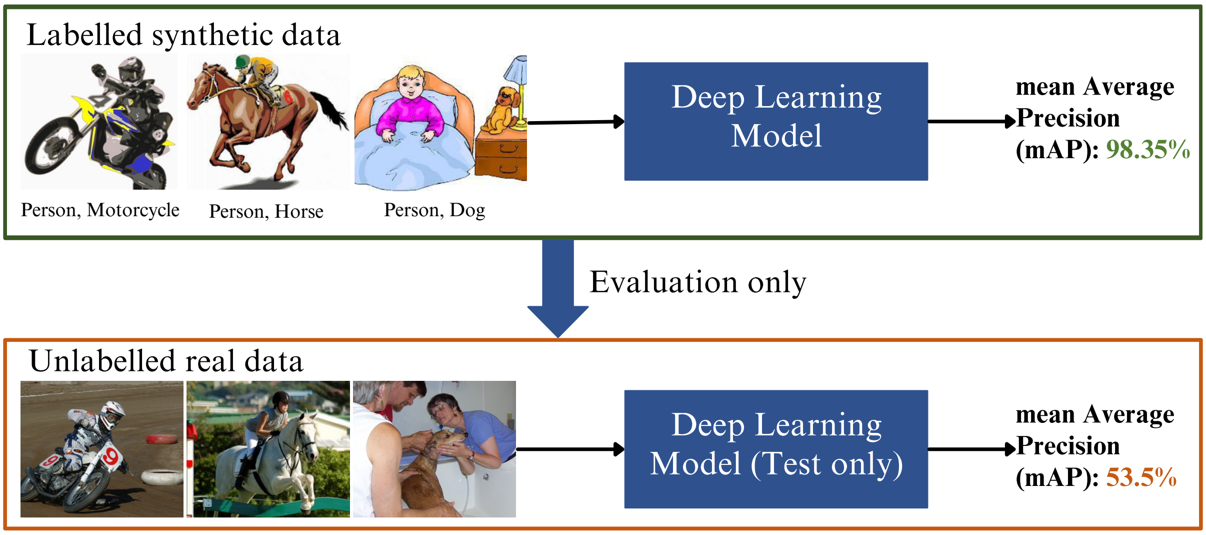


Figure 1.2: Problem of domain-shift in image classification.

competitive performance for the problem of MLIC, while keeping a relatively lightweight architecture. More specifically, we aim to effectively inject some prior information regarding the relationship between multiple objects, into the learning process. This is intuitive, as in real-world images some objects tend to appear together more than others. For instance, it is more common to observe a "person" holding a "badminton racket" rather than a "dog" holding the same within an image.

In the second part of this thesis, we focus on the generalization capabilities of existing MLIC methods, especially for unseen images. Indeed, existing DL-based MLIC methods naturally inherit the limitations of deep learning and hence, are also negatively impacted by the problem of generalization. This drop in performance, commonly known as domain-shfit [22], usually occurs when a trained MLIC model is evaluated on images containing different variations compared to the training images. For instance, as shown in Fig. 1.2, a model trained on synthetic images shows a significant performance drop when evaluated on the real images containing the same object categories. Labeling these new images and retraining the model with the same can help achieve better performance, however, this is both resource-intensive and time-consuming. Hence, in order to avoid costly annotation efforts, researchers have exploited the field of Unsupervised Domain Adaptation (UDA) [23]. Despite their success, UDA methods have been primarily focusing on multi-class classification with very few efforts towards UDA for MLIC. Hence, in the second part of this thesis, we aim to propose methods that are tailored specifically for the task of MLIC in the presence of domain-shift.

In summary, in this thesis, we aim to explore and investigate the problem of multi-label image classification under both single and cross-domain settings with the objective of obtaining more compact yet effective architectures.

1.2 Challenges

In this section, we discuss the challenges related to MLIC that we aim to address in this thesis. These challenges are twofold, namely, the problem of the trade-off between effectiveness and efficiency and the generalization to unseen domains.

1.2.1 Trade-off Between Effectiveness and Efficiency

Improved performance can be attained by employing very deep neural network architectures for MLIC. With the advent of ResNet [9], it is feasible to construct networks of considerable depth without encountering the issue of vanishing gradients. This enables the extraction of more abstract and meaningful image features, thereby facilitating significantly enhanced and accurate classification performance. Several methods [24]–[26] have been subsequently introduced, offering distinct versions of ResNet aimed at even more enhanced image feature extraction while maintaining a good trade-off between training speed and classification performance.

Despite their demonstrated performance in MLIC, existing methods often rely on very large model architectures. For instance, most of the existing methods [19]–[21] use

ResNet101 as their CNN backbone for feature extraction, which necessitates approximately 42.8 million trainable parameters. Very recently, transformer-based architectures have been largely investigated [27]–[29] for better performance. However, the number of trainable parameters in such architectures may range anywhere between 65 to 200 million. Consequently, these methods are impractical for deployment in memory-constrained scenarios. Hence, in this thesis, we aim at addressing the challenge of cumbersome network architectures in existing MLIC methods.

1.2.2 Generalization to Unseen Domains

The widespread availability of large-scale labeled datasets and the latest progress in deep learning are the two main factors behind the success of existing MLIC methods. Hence, inheriting from the limitations of deep learning, existing MLIC methods are also negatively impacted by the problem of generalization, especially when applied to unseen images. Indeed, the real-world applications of MLIC might necessitate to classification of multiple objects in the images with varying visual appearances, lighting conditions, backgrounds, etc. Since it is not possible to include all these variations in the training, a significant performance drop can be observed when tested on images with different variations. As introduced earlier and showcased in Fig. 1.2, this phenomenon is commonly known as domain-shift [22]. In other words, an MLIC method trained using data from a given domain, usually called *source* domain, will suffer from degraded performance when tested on samples belonging to an unseen domain, referred to as *target* domain. A direct solution is to simply label these target data and use them as additional training data. Nevertheless, such a process is both resource-intensive and time-consuming. Hence, in order to overcome this challenge, Unsupervised Domain Adaptation (UDA) [23], [30] can be used as a potential solution that utilizes both labeled source and unlabeled target data to minimize the shift between the source and the target domains.

Despite the popularity of UDA, only a few methods have been developed for MLIC in a cross-domain setting, drawing inspiration from the numerous UDA methods available in multi-class classification tasks. In general, these handfuls of UDA for MLIC methods are the direct extensions of their multi-class classification counterparts. However, this might be suboptimal due to the inherent differences between the nature of the two tasks (See Fig. 1.1). In addition, UDA methods in MLIC also rely on large and complex network architectures to achieve optimal performance which limits their applicability in scenarios where memory resources are limited. Hence, in this thesis, we propose compact yet effective methods specifically designed to address the challenge of MLIC in the presence of domain shift.

1.3 Objectives and Contributions

In this thesis, we address the problem of multi-label image classification both within a single domain and across different domains. More specifically, we overcome the two main limitations of existing MLIC methods; namely 1) the trade-off between effectiveness and efficiency, and, 2) generalization to unseen domains. The following sections introduce the different works that were proposed during the doctoral study to address these challenges.

1.3.1 Image-based Node Embeddings for Graph Convolutional Network-based Multi-label Image Classification

Our first contribution, namely IML-GCN, utilizes image-based node embeddings for effectively modeling the correlations between multiple labels. In general, existing indirect methods for MLIC based on Graph Convolutional Networks (GCN) construct a direct graph over object labels, considering each label a node and the co-occurrence probability of each label pair as the edge. Most of these methods utilize Glove-based word embeddings trained on the Wikipedia dataset as node representations for each label. However, since Glove-based embeddings were originally proposed for Natural Language Processing (NLP), their application in the field of image classification may result in sub-optimal performance. Additionally, these methods often employ very large CNN backbones, limiting their application to a memory-constrained scenario.

To address these issues, we propose IML-GCN, where node embeddings are generated using a trained CNN. This allows the features extracted from the image to more accurately represent the labels. IML-GCN can be trained using the proposed image-based embeddings on a much lighter CNN backbone, reducing the number of training parameters by at least half while maintaining competitive classification performance compared to state-of-the-art methods. This work has been published in [31].

1.3.2 An Adaptive Graph Convolution-based Strategy for Modeling Label Correlations for MLIC in both Single and Cross-domains.

Existing graph-based indirect approaches for MLIC often suffer from cumbersome architectures and a fixed-label graph topology, which may not adequately adapt to the dynamic nature of MLIC tasks. To address these limitations, we propose ML-AGCN, which learns the label topology adaptively in an end-to-end manner. Specifically, we incorporate an attention-based strategy to quantify pair-wise importance between multiple labels. This enables ML-AGCN to flexibly model label relationships and achieve state-of-the-art results on various MLIC benchmarks. Importantly, ML-AGCN maintains a smaller overall model size compared to existing methods, making it more efficient while retaining competitive performance. This work has been published in [32].

Furthermore, we extend this adaptive graph learning strategy to the cross-domain setting, where the objective is to efficiently classify multiple labels while minimizing domain shift between source and target domains. This extended work has been submitted at CVIU [33].

1.3.3 A Discriminator-free Approach for Unsupervised Domain Adaptation in MLIC

In this work, termed as DDA-MLIC, we propose one of the first discriminator-free adversarial approaches for Unsupervised Domain Adaption (UDA) in MLIC. Existing adversarial-based UDA methods for MLIC typically incorporate an additional discriminator that is trained adversarially to ensure that the generated features are indiscriminate to the domains, and hence, implicitly minimizing the domain gap between source and target samples. However, this approach often reduces the task discriminative power of these methods because of the decoupling of classification and domain alignment.

DDA-MLIC solves this limitation by reusing the task classifier as a discriminator, hence eliminating the need for an additional discriminator. Specifically, an adversarial critic, based on the predicted probabilities of source and target samples, is derived from the classifier. The proposed approach achieves state-of-the-art results on several UDA benchmarks for MLIC and has been published in [34]. Additionally, an extended version [35] addressing the non-differentiability limitation of DDA-MLIC has been recently submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.

1.3.4 Application to Deepfakes

To assess the suitability of the developed methods, we consider the practical use case of deepfake detection. Deepfakes are synthesized images or videos, where some or all facial attributes and/or components are altered using deep learning techniques. Detecting deepfakes is crucial to prevent issues like identity theft, fraudulent transactions, and the spread of pornography, among others.

Recent methods treat deepfake detection as a binary classification problem, aiming to determine whether a given image is “real” or “fake”, resulting in poor explanability capabilities. In fact, an image predicted as fake can be produced by one or multiple manipulations. Therefore, we propose to formulate the problem of deepfake detection as a multi-label classification, where each label corresponds to a specific manipulation. We evaluate and compare several direct and indirect MLIC approaches on a recently introduced multi-label deepfake dataset [36].

This work explores the potential of multi-label classification in deepfake detection and highlights the importance of interpretable predictions in real-world applications. The paper was published in [37].

1.4 Publications

JOURNALS

1. **Singh, I.P.**, Ghorbel, E., Oyedotun, O. and Aouada, D., 2023. ”Multi-label Image Classification using Adaptive Graph Convolutional Networks: from a Single Domain to Multiple Domains”. Computer Vision and Image Understanding 2024.

2. **Singh, I.P.**, Ghorbel, E., Kacem, A., Rathinam, A. and Aouada, D, 2024. ”Domain Adaptation for Multi-label Image Classification: a Discriminator-free Approach”. IEEE Transactions on Pattern Analysis and Machine Intelligence. **(under review)**.

CONFERENCES

1. **Singh, I.P.**, Oyedotun, O., Ghorbel, E. and Aouada, D., 2022. ”Iml-gcn: Improved multi-label graph convolutional network for efficient yet precise image classifica-

tion”. In AAAI-22 Workshop Program-Deep Learning on Graphs: Methods and Applications.

2. **Singh, I.P.**, Ghorbel, E., Oyedotun, O. and Aouada, D., 2022, October. ”Multi Label Image Classification using Adaptive Graph Convolutional Networks (ML-AGCN)”. In 2022 IEEE International Conference on Image Processing (ICIP) (pp. 1806-1810). IEEE.

3. **Singh, I.P.**, Mejri, N., Nguyen, V.D., Ghorbel, E. and Aouada, D., 2023, September. ”Multi-label deepfake classification”. In 2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP) (pp. 1-5). IEEE.

4. **Singh, I.P.**, Ghorbel, E., Kacem, A., Rathinam, A. and Aouada, D., 2024. ”Discriminator-free Unsupervised Domain Adaptation for Multi-label Image Classification”. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 3936-3945).

PUBLICATIONS NOT INCLUDED IN THIS THESIS

1. Gómez R.R., Lorentz J., Hartmann T., Goknil A., **Singh I.P.**, Halaç T.G. and Ekinci G.B., 2024. “An AI pipeline for garment price projection using computer vision”. In Neural computing and applications, 2024. https://doi.org/10.1007/s00521-024-09901-w

2. Dupont, E., **Singh, I.P.**, Fuentes, L., Ali, S.A., Kacem, A., Ghorbel, E. and Aouada, D., 2023. “You Can Dance! Generating Music-Conditioned Dances on Real 3D Scans.” In Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2023, Volume 4: VISAPP.

3. Pauly, L., Jamrozik, M. L., del Castillo, M. O., Borgue, O., **Singh, I. P.**, Makhdoomi, M. R., Christidi-Loumpasefski, O.-O., Gaudillière, V., Martinez, C., Rathinam, A., Hein, A., Olivares-Mendez, M., and Aouada, D. (2023). "Lessons from a Space Lab: An Image Acquisition Perspective". International Journal of Aerospace Engineering, 2023, 1-16. https://doi.org/10.1155/2023/9944614

4. Nguyen, D., Mejri, N., **Singh, I.P.**, Kuleshova, P., Astrid, M., Kacem, A., Ghorbel, E. and Aouada, D., 2024. "LAA-Net: Localized Artifact Attention Network for High-Quality Deepfakes Detection". In 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

1.5 Thesis Outline

This dissertation is organized as follows:

- **Chapter 2:** In this chapter, we provide an overview of the background for a better understanding of this thesis. This background is related to Multi-Label Image Classification (MLIC) and Unsupervised Domain adaptation (UDA).

- **Chapter 3:** In this chapter, we explore the limitations of typical word-based embeddings used as node representations in graph-based multi-label classification. We introduce IML-GCN, which leverages image-based node embeddings to model more effectively the label relationships.

- **Chapter 4:** In this chapter, we present ML-AGCN, which addresses the problem of the heuristically fixed graph topology in standard graph-based methods. An adaptive graph module is proposed to adaptively model label correlations under both single-domain and cross-domain settings for MLIC.

- **Chapter 5:** In this chapter, we introduce a fully differentiable discriminator-free approach called DDA-MLIC for addressing the domain shift in MLIC. We demonstrate that reusing the classifier as a discriminator can enhance the task-specific discriminative ability of the overall framework.
- **Chapter 6:** In this chapter, we investigate the applicability of existing MLIC methods for the practical use case of deepfake detection. We explore the potential of multi-label classification in deepfake detection and emphasize the importance of interpretable predictions in real-world applications.
- **Chapter 7:** In conclusion, this final chapter summarizes this thesis work and discusses future perspectives.

Chapter 2

Background

This chapter provides the background necessary for understanding related multi-label classification methods. Specifically, we start by recalling the Graph Convolutional Networks (GCN) and how they are used in MLIC for modeling label correlations. Later, we review the concept of Unsupervised Domain Adaptation (UDA) including the two different paradigms of UDA namely, adversarial-based and moment matching-based methods. Finally, we discuss adversarial-based UDA strategies for MLIC.

2.1 Multi-label Image Classification (MLIC) using Graph Convolutional Networks (GCN)

In this section, we start by reviewing the concept of Graph Convolutional Networks (GCN). Subsequently, we introduce the pioneering graph-based method which might be considered as a baseline to some parts of our work.

2.1.1 Graph Convolutional Networks (GCN)

Graph convolution networks (GCN), initially introduced in [38], are the natural extension of Convolution Neural Networks (CNNs) to graphs. In fact, classical CNNs are designed for Euclidean structures and consequently applying them to graphs that are non-linear is not straightforward. Let us consider a graph $\mathcal{G} = (V, E, \mathbf{F}^0)$ with $V = \{v_1, v_2, ..., v_N\}$ the set of nodes, N the number of nodes, $E = \{e_1, e_2, ..., e_M\}$ the set of edges connecting the nodes, M the number of edges and $\mathbf{F}^0 \in \mathbb{R}^{N \times d}$ the input d-dimensional node features. Let $\mathbf{A}$ be the adjacency matrix defining the weighted connectivity of nodes.

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

$$\mathbf{F}^{l+1} = f(\mathbf{F}^l, A). \tag{2.1}$$

Using the same approach for convolution as [38], we can re-write Eq. (2.1) as:

$$\mathbf{F}^{l+1} = h(\hat{\mathbf{A}}\mathbf{F}^l\mathbf{W}^l), \tag{2.2}$$

where $\mathbf{W}^l \in \mathbb{R}^{d \times d'}$ is the weight matrix to be learned, $\hat{\mathbf{A}} \in \mathbb{R}^{N \times N}$ is the normalized version of $\mathbf{A}$ and h is a non-linear activation function.

2.1.2 GCN-based Multi-label Image Classification

The goal of multi-label image classification is to predict the presence or absence of a set of objects $\mathcal{O} = \{1, 2, ..., N\}$ in a given image $\mathbf{I}$. This can be done by learning a function f such that,

$$\begin{aligned} f\colon \mathbb{R}^{w\times h} &\to [\![0,1]\!]^{N} \\ \mathbf{I} &\mapsto \mathbf{y} = [y_i]_{i\in\mathcal{O}}, \end{aligned} \tag{2.3}$$

where w and h define the pixel-wise width and height of the image, respectively, and $y_i = 1$ indicates the presence of the label i in $\mathbf{I}$, in contrast to $y_i = 0$. Graph-based multi-label image classification methods such as ML-GCN [39], A-GCN [40] and F-GCN [41] usually involve two subnetworks: (1) A feature generator denoted by f_g, and (2) an estimator of N inter-dependent binary classifiers denoted by f_c. The generator mostly corresponds to an out-off-the-shelf CNN network, such as ResNet [9], which produces a d_f-dimensional image feature representation as described below,

$$\begin{aligned} f_g\colon \mathbb{R}^{w\times h} &\to \mathbb{R}^{d_f} \\ \mathbf{I} &\mapsto \mathbf{X} \end{aligned} . \tag{2.4}$$

On the other hand, the second subnetwork f_c is a GCN formed by L layers which takes a fixed graph $\mathcal{G} = (V, E, \mathbf{F})$ as input where card$(V) = N$. In the context of modeling label relationships, each node of the graph $v_i \in V$ represents a label $i \in \{1, 2, ..., N\}$ and and each $\mathbf{f}_i \in \mathbf{F}$ corresponds to its associated label embedding. Then, the probability that two labels appear together in an image defines an edge and is directly encoded in the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ of $\mathcal{G}$. The adjacency matrix is usually pre-computed based on the co-occurrence probabilities that are estimated over the training set [39]. Most of the existing GCN-based MLIC methods [39]–[41] use word embedding [42] representations as input node features denoted by $\mathbf{F}_W$. These node features are generated by Glove [42]. Thus, in this case, we can say that $\mathbf{F}^0 = \mathbf{F}_W$. Furthermore, a re-weighted scheme is typically used [39] where firstly a threshold τ has been used to filter the noisy edges resulting in:

$$A_{ij} = \begin{cases} 0, \text{ if } p_{ij} < \tau, \\ 1, \text{ if } p_{ij} \geq \tau \end{cases}, \tag{2.5}$$

where $p_{ij} = p(I_j|I_i)$ is the the probability of co-occurrence of label j and label i in the same image. Secondly, in order to avoid over-smoothing, the following re-weighted scheme is used [39]:

$$A'_{ij} = \begin{cases} p/\sum_{j=1}^{N} A_{ij}, \text{ if } i \neq j, \\ 1 - p, \text{ if } i = j \end{cases}, \tag{2.6}$$

where p determines the weights assigned to a node itself and other correlated nodes.

Given $\mathbf{F}^l \in \mathbb{R}^{d_l \times L}$ as the input node features of the l^{th} layer, the input features $\mathbf{F}^{l+1}$ of the $(l+1)^{\text{th}}$ are therefore computed by following Eq. (2.1).

Finally, the vertex features produced by the last GCN layer, i.e., $\mathbf{F}^L \in \mathbb{R}^{N \times d_f}$, form the N inter-dependent classifiers.

In summary, f_c can be defined as follows,

$$\begin{aligned} f_c \colon \mathcal{G}(N) &\to \mathbb{R}^{d_f \times N} \\ \mathcal{G} &\mapsto \mathbf{F}^L = [F_1^L, ..., F_N^L], \end{aligned} \tag{2.7}$$

where $\mathcal{G}(N)$ represents the set of graphs with N nodes, and $\mathbf{F}_i^L$ for $i \in \{1, ..., N\}$ is the generated inter-dependent binary classifier associated to the label i.

Hence, f, which returns the final prediction, can be defined as follows,

$$\begin{aligned} f(\mathbf{I}) &= \text{sig}(f_c(\mathcal{G}) f_g(\mathbf{I})^T) \\ &= \text{sig}(\mathbf{F}^L \mathbf{X}^T), \end{aligned} \tag{2.8}$$

where sig$(x) = \frac{1}{1+e^{-x}}$ is the sigmoid activation function. A typical architecture of graph-based multi-label image classification [39] is showcased in Figure 2.1 where

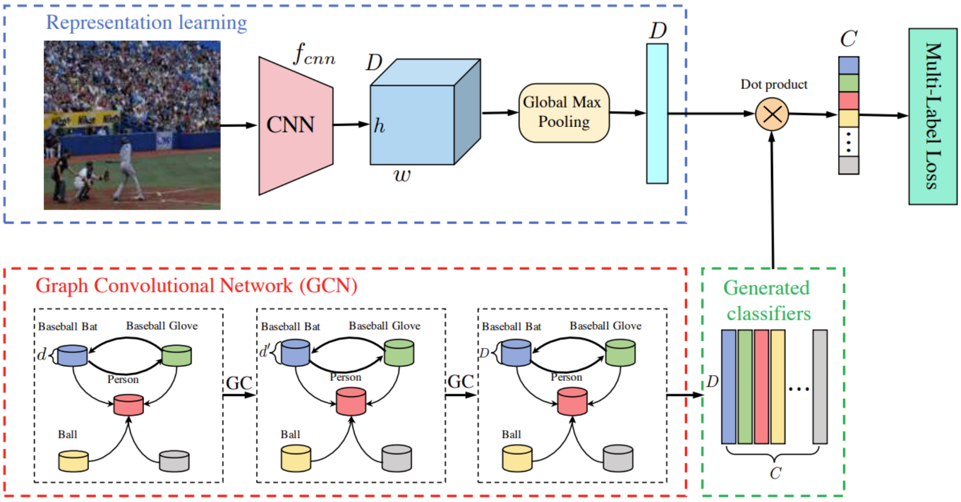


Figure 2.1: Overall framework of graph convolutional network-based multi-label image classification [39].

f_{cnn} represents the feature generator extracting D-dimensional discriminative image features and the GCN subnetwork learn C inter-dependent binary classifiers, with C being the total number of class labels. These learned classifiers utilize the image features extracted by f_{cnn} to provide probability scores indicating the presence of object labels. The overall network is optimized using a supervised multi-label classification loss, typically a multi-label soft margin loss. By minimizing this loss, the network is trained to accurately classify images into multiple labels simultaneously, improving its performance in MLIC tasks.

2.2 Unsupervised Domain Adaptation (UDA)

As introduced earlier, existing MLIC methods usually assume that unseen images and training data are drawn from the same distribution, i.e. the same domain, hence ignoring a possible domain-shift problem [22]. This leads, therefore, to poor generalization capabilities under cross-domain settings (see Figure 1.2). Unsupervised Domain Adaptation (UDA) is a plausible solution to overcome this challenge without relying on costly annotation efforts [23]. UDA aims at learning domain invariant features to bridge the gap between a *source* domain and a *target* domain without access to the associated target labels. In what follows, we discuss the two different paradigms of UDA for any general image classification.

2.2.1 Moment matching-based Unsupervised Domain Adaptation (UDA)

One of the most known paradigm for domain adaptation involves the explicit alignment of the source and target distributions by minimizing. Maximum Mean Discrepancy (MMD) [43] is one of the earliest and widely used strategies, computed from the mean values of the source and target samples after applying a smooth function. If the means differ importantly, this suggests that the two distributions are not well-aligned. The chosen smooth functions are unit balls in characteristic reproducing kernel Hilbert spaces (RKHS).

Several methods have investigated different variants of MMD. For instance, multiple-kernel (MK-MMD) [44] embeds the hidden representations from all task-specific layers in reproducing kernel Hilbert spaces (RKHS). It further reduces the discrepancy using an optimal multi-kernel selection method for mean embedding matching. Given $\mathcal{H}_k$ the reproducing kernal Hilbert space (RKHS) with its characteristics kernel k, the MK-MMD $d_k(p, q)$ between two probability distributions p and q is defined as the RKHS distance between the mean embeddings of p and q. The squared formulation of the same is given as:

$$d_k^2(p, q) = ||\mathbb{E}_p[\phi(\mathbf{I}_s)] - \mathbb{E}_q[\phi(\mathbf{I}_t)]||^2_{\mathcal{H}_k}, \tag{2.9}$$

where ϕ represents the feature map of the respective source or target sample.

A similar approach involves learning a transfer network by aligning the joint distributions of multiple domain-specific layers across domains based on a joint maximum mean discrepancy (JMMD) criterion [45]. Correlation alignment (CORAL) [46] operates similarly to MMD but utilizes a polynomial kernel. It is computed based on the distance between second-order statistics (covariances) of the source and target features.

2.2.2 Adversarial-based Unsupervised Domain Adaptation (UDA)

Over the last few years, several adversarial-based UDA approaches have been proposed in the context of multi-class classification. Generally, these methods incorporate a domain discriminator for identifying the origin of image features, distinguishing between the source and target domains. Given $\mathbf{I}$ the input image and $\mathbf{y} = \{1, 2, ..., N\}$ the set of possible labels, a multi-class classification task generally aims to learn a function f such that $f : \mathbf{I} \mapsto \mathbf{y}$. In a typical UDA setting, a labeled source sample S drawn $i.i.d.$ from $\mathcal{D}_s$, and an unlabeled target sample T drawn $i.i.d.$ from $\mathcal{D}_t$, such that:

$$S = \{(\mathbf{I}_i, \mathbf{y}_i)\}_{i=1}^{n_s} \sim (\mathcal{D}_s)^{n_s};\ T = \{\mathbf{I}_i\}_{i=1}^{n_t} \sim (\mathcal{D}_t)^{n_t}, \tag{2.10}$$

with n_s and n_t, respectively, being the total number of source and target samples.

In order to achieve successful domain adaptation, one of the pioneering works in adversarial learning of neural networks, namely DANN [30], has proposed to utilize a domain discriminator for implicitly minimizing the domain shift. In principle, DANN comprises three key components:

1. **Feature Generator (**G_f**)**: This is typically a Convolutional Neural Network (CNN) denoted as $G_f(\cdot; \theta_f)$ with parameters θ_f. Given any input image $\mathbf{I}$, the feature generator extracts d_f-dimensional discriminative features, represented as $G_f : \mathbf{I} \mapsto \mathbb{R}^{d_f}$.

2. **Task Classifier (**G_y**)**: This is the prediction layer denoted as $G_y(\cdot; \theta_y)$ with parameters θ_y. Given the d_f-dimensional image features, the task classifier aims

to predict the probability of belonging to one of the given classes, defined as $G_y : \mathbb{R}^{d_f} \mapsto [0,1]^N$, where N is the total number of class labels.

3. **Domain Discriminator (G_d)**: This is the domain classification layer denoted as $G_d(\cdot;\theta_d)$ with parameters θ_d. Given both source and target features, the domain discriminator classifies their original domain, such that $G_d : \mathbb{R}^{d_f} \mapsto [0,1]$. Here, a domain label of 0 indicates that the features belong to the source domain, while 1 indicates they belong to the target domain.

The two losses, i.e., the prediction loss and the domain loss are defined by:

$$\begin{aligned} \mathcal{L}_y^i(\theta_f,\theta_y) &= \mathcal{L}_y(G_y(G_f(\mathbf{I}_i;\theta_f);\theta_y),y_i), \\ \mathcal{L}_d^i(\theta_f,\theta_d) &= \mathcal{L}_d(G_d(G_f(\mathbf{I}_i;\theta_f);\theta_d),d_i) \end{aligned}, \tag{2.11}$$

where y_i and d_i are, respectively, the class and the domain labels of the input image $\mathbf{I}_i$. Note that y_i is only available for source images. Hence, the overall pseudo-function of $(\theta_f,\theta_y,\theta_d)$ that is being optimized is given as:

$$E(\theta_f,\theta_y,\theta_d) = \frac{1}{n_s}\sum_{i=1}^{n_s}\mathcal{L}_y^i(\theta_f,\theta_y) - \lambda\left(\frac{1}{n_s}\sum_{i=1}^{n_s}\mathcal{L}_d^i(\theta_f,\theta_d) + \frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{L}_d^i(\theta_f,\theta_d)\right), \tag{2.12}$$

where λ is a hyper-parameter used to tune the trade-off between the two quantities during the learning process.

In their work, as shown in Figure 2.2, the domain discriminator and the feature generator engage in a min-max game facilitated by the Gradient Reversal Layer. The generator aims to fool the discriminator by maximizing its loss and finding a saddle point $\hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_d$ such that:

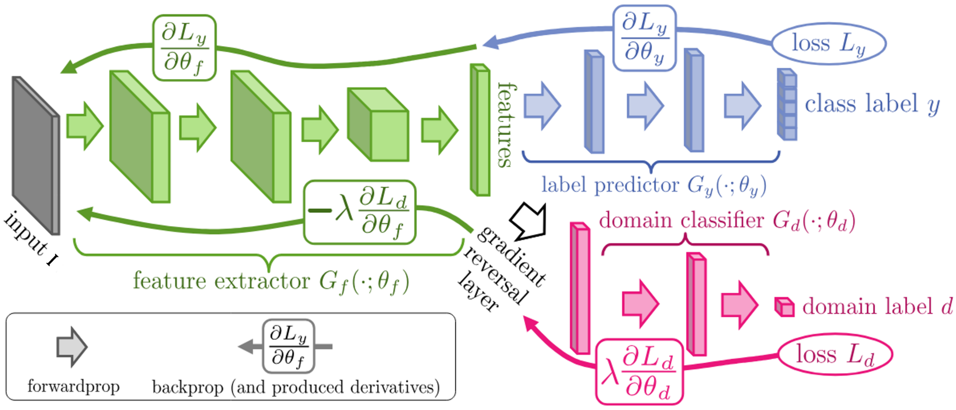
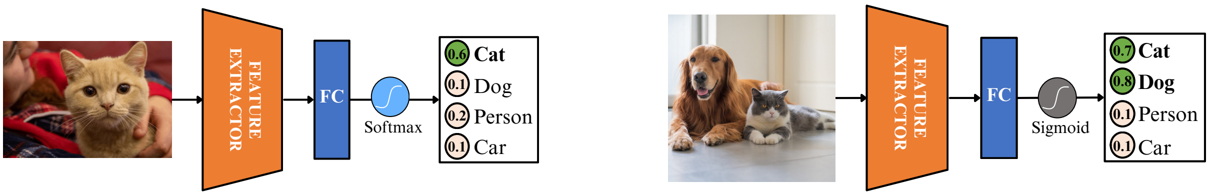

$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
class label y
features
label predictor $G_y(\cdot;\theta_y)$
$-\lambda\frac{\partial L_d}{\partial \theta_f}$
input I
feature extractor $G_f(\cdot;\theta_f)$
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
domain label d
loss L_d
$\lambda\frac{\partial L_d}{\partial \theta_d}$
forwardprop
backprop (and produced derivatives)

Figure 2.2: Domain-Adversarial Training of Neural Networks [30].



(a) Multi-class classification

(b) Multi-label classification

Figure 2.3: Comparison between multi-class and multi-label classification.

$$(\hat{\theta}_f, \hat{\theta}_y) = \arg\min_{\theta_f, \theta_y} E(\theta_f, \theta_y, \theta_d), \tag{2.13}$$

$$\hat{\theta}_d = \arg\max_{\theta_d,} E(\hat{\theta}_f, \hat{\theta}_y, \theta_d). \tag{2.14}$$

The underlying concept is that if the discriminator cannot accurately predict the domain, the generator is effectively extracting domain-invariant features. This process leads to a reduction in the domain gap, as the classifier performs well on the target samples. This framework is crucial and widely used in several downstream unsupervised domain adaptation tasks.

Despite the widespread availability of UDA methods based on both paradigms, most of them have been proposed for the context of binary or multi-class classification. Directly extending them to the problem of MLIC remains challenging due to the inherent differences between the two tasks, as highlighted in Figure 2.3. In the following section, we will discuss the rare UDA works proposed specifically for the context of MLIC.

2.3 Adversarial-based Unsupervised Domain Adaptation for Multi-label Image Classification (MLIC)

First, we start by formulating the MLIC problem in the presence of a domain shift. Let us consider a source dataset for multi-label image classification $\{(\mathbf{I}_s^k, \mathbf{y}_s^k)\}_{k=1}^{n_s}$ drawn from a distribution $\mathcal{D}_s$ and a similar unlabelled target dataset $\{(\mathbf{I}_t^k)\}_{k=1}^{n_t}$ drawn from a different distribution $\mathcal{D}_t$. The matrix $\mathbf{I}_s^k \in \mathbb{R}^{w \times h}$ refers to the k^{th} image sample of $\mathcal{D}_s$ and $\mathbf{y}_s^k \in [\![0, 1]\!]^N$ is its associated label vector, while $\mathbf{I}_s^t \in \mathbb{R}^{w \times h}$ denotes the k^{th} image sample of $\mathcal{D}_t$. The variables n_s and n_t refer respectively to the total number of samples in $\mathcal{D}_s$ and $\mathcal{D}_s$. The goal of unsupervised domain adaptation is to train a model using labeled source domain data sampled from $\mathcal{D}_s$ and unlabelled target domain samples from $\mathcal{D}_t$ for making accurate predictions on the target domain.

Inspired by the predominance of the adversarial-based approaches from multi-class classification, very few works have been proposed for UDA in multi-label classification. Among the most recent and accurate approaches, one can mention DA-MAIC [47], which takes advantage of the graph representation for modeling label correlations. As in ML-GCN [39], they have proposed to build a graph for modeling label correlations based on label co-occurrences. Additionally, to reduce the domain shift between the source and target domains, they have used a domain classifier that is trained in an adversarial manner, similar to [30]. The overall flowchart of their proposed architecture is showcased in Figure 2.4.

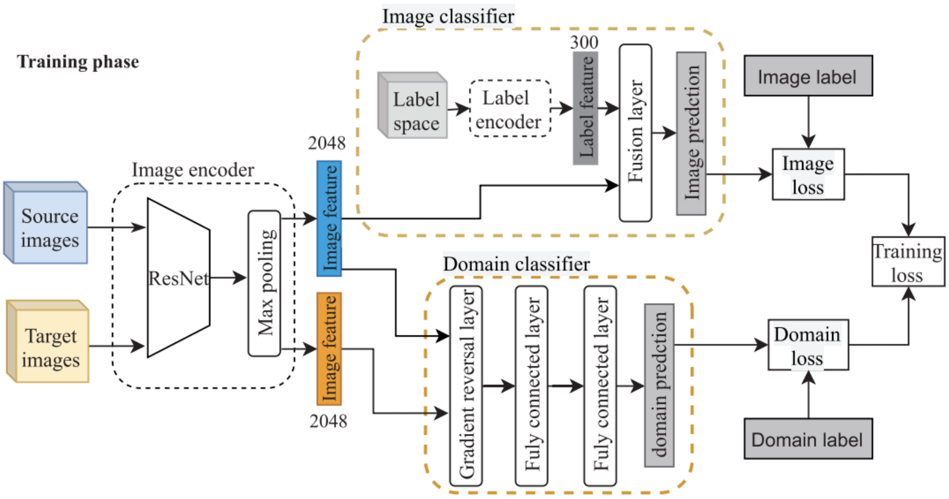


Figure 2.4: Flowchart of the DA-MAIC framework [47].

Chapter 3

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

In this chapter, we introduce our contribution called *Improved Multi-Label Graph Convolutional Network (IML-GCN)*. This MLIC method has the advantage of being precise yet efficient. As introduced in Chapter 1.2.1, despite achieving great performance, previous MLIC approaches usually make use of very large architectures. To handle this, we propose to combine the small version of a newly introduced network called TResNet with an extended version of Multi-label Graph Convolution Networks (ML-GCN); therefore ensuring the learning of label correlations while reducing the size of the overall network. The proposed approach considers a novel image feature embedding instead of using word embeddings. In fact, the latter are learned from words and not images, making them inadequate for the task of multi-label image classification. Experimental results show that our framework competes with the state-of-the-art on two multi-label image benchmarks in terms of both precision and memory requirements.

3.1 Introduction

Multi-label image classification is an active research topic in computer vision. In the literature, multi-label prediction approaches can be classified into two main categories. The first class of methods generally learns a one-stream Deep Neural Network (DNN) for multiple binary classification tasks, without integrating any prior knowledge in the architecture design as in [9], [24], [25]. We refer to these approaches as *direct methods*. Although direct methods have been shown to achieve high performance as in [24], they generally necessitate the use of multiple layers to work effectively. This leads to a high memory consumption; therefore restricting their applicability in a memory-constrained context. In contrast to the latter, the second category of approaches, that we call *indirect methods*, takes advantage of the prior knowledge related to the correlations existing among different objects present in an image [27], [39], [48], [49]. This is intuitive when one considers that in real-life some combinations of objects are more likely to appear together than others. For instance, it is extremely likely for a racket and person to appear together, than a racket and a dog. Indirect methods usually extend direct approaches by adding a subnetwork that models the different label relationships. Intuitively, one would think that using these data-driven approaches would allow obtaining a model with a reduced number of parameters. Nonetheless, it has been noted that most of these methods present a high number of parameters or reduce the memory requirements at the cost of a decrease in terms of precision. In this paper, our assumption is that finding the good combination between direct and indirect architectures will enable us achieving competitive performance, while reducing the size of the model.

For this reason, we design a new framework termed the *Improved Multi-label Graph Convolutional Network (IML-GCN)* that simultaneously considers the newly introduced direct approach called *TResNet* [24] and an indirect model termed the *Image Feature Embeddings-based Graph Convolutional Network (IFE-GCN)*, which extends the graph

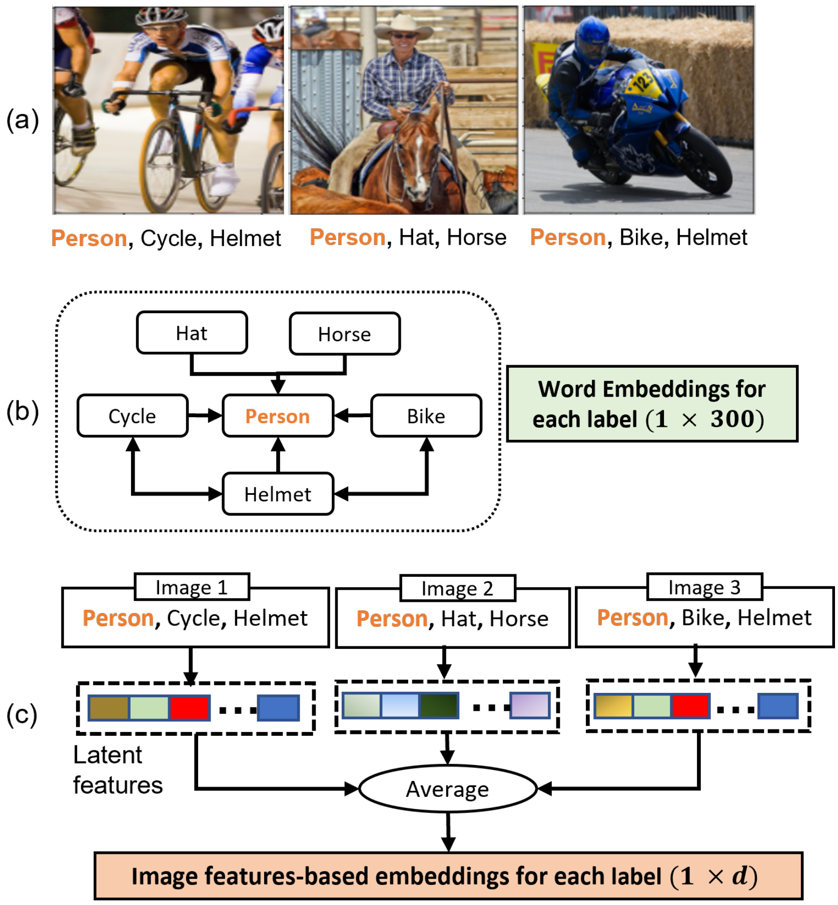


Figure 3.1: a) Image samples with multiple labels, b) representation of graph nodes using word embeddings and, c) our proposed embeddings using learned image-based latent representation.

subnetwork of the *Multi-label Graph Convolutional Network (ML-GCN)* introduced in [39]. TResNet is chosen given its impressive performance in terms of precision even when reducing the number of layers, while an improved version of ML-GCN is considered given its relatively low memory consumption. ML-GCN is one of the most popular works using graphs for modeling the label dependencies. Each label is represented by a node while the relationship between labels is modeled using weighted edges. Then, GLOVE word embeddings [42] are used as node features. Unfortunately, this might lead to an inconstancy since the GCN is used to create binary classifiers that takes image features

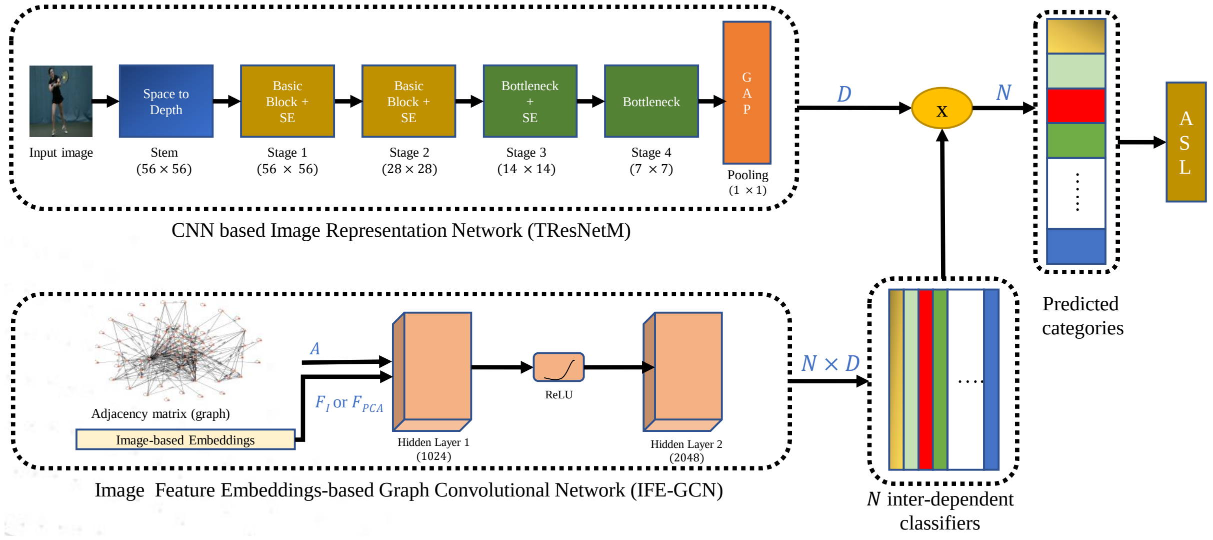


Figure 3.2: Architecture of our IML-GCN approach for multi-label image classification.

as input that are extracted from a second network. In fact, we recall that GLOVE has been initially designed to represent words with vectors in the field of Natural Language Processing (NLP), while visual object features are by nature different.

To overcome that, we propose to replace the word embeddings by novel image embeddings which are more meaningful in this problem of multi-label image classification, as illustrated in Figure 3.1. More specifically, our image embeddings are computed using label-wise image representations that are extracted by a state-of-the-art image feature extractor. Figure 3.2 shows an overview of the proposed framework. Its relevance in terms of precision and number of parameters with respect to the state-of-the-art is shown by performing experiments on two well-known datasets.

The organization of the remaining sections of this chapter is as follows. Section 3.2 details the problem statement and our motivation. Section 3.3 depicts the proposed framework of IML-GCN and details the methodology for generating the image-based embeddings. Section 3.4 details the different experiments, and results discussion followed by an extensive study on model performance. The paper is finally concluded in

Section 3.5, which summarizes the major findings in this work.

3.2 Problem statement

As introduced in Chapter 2.1, Graph Convolutional Networks (GCN) are the natural extension of Convolution Neural Networks (CNNs) to graphs. ML-GCN [39] were among the first to use GCN in the context of multi-label image classification for modeling the label correlations. As detailed in Section 2.1.2, they aim to learn N interdependent label classifiers, one for each of the N object labels.

Despite the good performance of ML-GCN, two main limitations can be noted. First, the backbone network for image representation is very deep (ResNet-101) and therefore naturally induces a heavy architecture leading to high memory computation and consumption. Second, as mentioned in Section 2.1.2, it uses word embeddings generated by Glove [42] to represent each node (label). Unfortunately, they might not be optimal in the context of image classification. In fact, these embeddings have been generated for representing words using unique vectors in the field of NLP, while the latter are used in ML-GCN for generating binary classifiers that take image features as input. Given the difference in nature of the two modalities (words and images), it seems not suitable to consider word embeddings as node features. Based on these two observations, two interesting ideas are driving this work. (1) It would be interesting to investigate if finding an appropriate combination between direct and indirect networks could be a way to achieve state-of-the-art results, while reducing the size of the network. (2) Replacing word embeddings with meaningful image embeddings could help improving the results without an increase of the parameter number.

3.3 Proposed Approach

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

3.3.1 Direct subnetwork: TResNet

Our first subnetwork, as shown in Figure 3.2, is a CNN-based image-representation network whose aim is to extract image features by using a set of convolutional blocks. More specifically, we use TResNetM, a smaller version of TResNet [24], which was designed to boost neural networks accuracy while retaining their GPU training and inference efficiency. This is in line with our objective of reducing the size of the final network. It has already demonstrated state-of-the-art results on single and multi-label datasets [21] while maintaining a balanced trade-off between speed and accuracy. However, it has been noted that the precision drops considerably when employing the small version TResNetM, compared to TResNetL. In general, the refinements on top of plain ResNet architecture include: SpaceToDepth Stem [50], Anti-Alias Downsampling [51], In-Place Activated BatchNorm [52], Novel Block-Type Selection [24] and Optimized SE Layers [53].

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

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

The second subnetwork consists of an improved version of the graph subnetwork introduced in the indirect method ML-GCN [39]. The overall architecture remains the same as the GCN branch of the original ML-GCN, except that we replace the input word embedding-based node representation by our proposed image-feature-based embeddings.

As we can see in Figure 3.2, the output of the proposed GCN network creates N interdependent binary classifiers incorporating the information of label correlations. However, as stated earlier, word embeddings are not adapted for multi-label image classification. Thus, the idea would be to replace these word embeddings with relevant image embeddings that could be sufficiently discriminative to design effective classifiers. Intuitively, the idea would be to generate a vector per object label including relevant image features related to the corresponding object. Below, we depict in details how these novel image embeddings are computed.

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

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

$$(\mathbf{f}_i)_I = mean(\mathbf{S}_i) \tag{3.1}$$

with $\mathbf{F}_I = [(\mathbf{f}_1)_I, (\mathbf{f}_2)_I, ..., (\mathbf{f}_N)_I] \in \mathbb{R}^{N \times d_f}$.

Furthermore, since we employ the image-feature embeddings as inputs to the GCN, improving the signal-to-noise of the input can facilitate the learning of robust

Table 3.1: Comparisons with state-of-the-art methods on the MS-COCO dataset with n_components=80 the number of the components fixed for computing $\mathbf{F}_{PCA}$.

Method	**#Parameters**	**mAP**	**CP**	**CR**	**CF1**	**OP**	**OR**	**OF1**
CNN-RNN [49]	66.2 M	61.2	-	-	-	-	-	-
SRN [19]	∼48M	77.1	81.6	65.4	71.2	82.7	69.9	75.8
ResNet101 [9]	44.5M	77.3	80.2	66.7	72.8	83.9	70.8	76.8
Multi-Evidence [20]	∼47M	-	80.4	70.2	74.9	85.2	72.5	78.4
ML-GCN [39]	44.9M	83	85.1	72	78	85.8	75.4	80.3
SSGRL [55]	92.2M	83.8	**89.9**	68.5	76.8	**91.3**	70.8	79.7
KGGR [56]	∼45M	84.3	85.6	72.7	78.6	87.1	75.6	80.9
C-Tran [27]	120M	85.1	86.3	74.3	79.9	87.7	76.5	81.7
ASL (TResNetM) [21]	**29.5M**	81.8	82.1	72.6	76.4	83.1	76.1	79.4
ASL (TResNetL) [21]	53.8M	86.6	87.4	76.4	**81.4**	88.1	79.2	81.8
Ours (IML-GCN with $\mathbf{F}_I$)	33.5M	85.9	82.7	78.9	80.5	84.6	82.1	**83.3**
Ours (IML-GCN with $\mathbf{F}_{PCA}$)	31.5M	**86.6**	78.8	**82.6**	80.2	79.0	**85.1**	81.9

representations by the GCN. Therefore, we use Principal Component Analysis (PCA), which simultaneously reduces the dimension of the image-feature embeddings from d_f to N such that the new input feature matrix $\mathbf{F}_{PCA} \in \mathbb{R}^{N \times N}$. Thus, these features are used as input to the first layer such as,

$$\mathbf{F}_I = \mathbf{F}_{PCA} \tag{3.2}$$

3.4 Experiments

In this section, we start by presenting the implementation details. Subsequently, we present the results and discussion on two benchmarking multi-label image recognition datasets, which include the MS-COCO [1] and VG-500 [54].

3.4.1 Implementation details:

The Asymmetric Loss (ASL) [21] is used as our loss function. The adjacency matrix for the GCN is computed using the same approach depicted in ML-GCN.

Table 3.2: Comparisons with state-of-the-art methods on the VG-500 dataset with n_components=500 for $\mathbf{F}_{PCA}$.

Method	**# Parameters**	**mAP (%)**
ResNet-101 [9]	44.5M	30.9
ML-GCN [39]	44.9M	32.6
ASL (TResNetM) [21]	**29.5M**	33.6
C-Tran [27]*	120M	38.4*
Ours (IML-GCN with $\mathbf{F}_I$)	33.5M	**34.0**
Ours (IML-GCN with $\mathbf{F}_{PCA}$)	32.1M	**34.5**

*The model is roughly 273% larger than our proposal

The hyper-parameters are empirically fixed. More specifically, we set the threshold to $\tau = 0.1$ in Eq. (2.5). We train the model for 40 epochs using a multi-step learning rate scheduler initialized with a learning rate of 10^{-3} and decayed by a factor of 0.1 at the 10, 20, and 30th epochs. For data augmentation, we use the same Randaugment technique as the baseline [21] during the training. Adam is used as optimizer [57] with a weight decay of $5e^{-4}$.

3.4.2 Experimental results:

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

Performance on MS-COCO: The MS-COCO [1] dataset is a well-known large-scale multi-label image dataset. It contains 122,218 images and covers 80 common objects. Following the conventional training and evaluation protocols for the MS-COCO dataset [20], [58], we report the following statistics: mean Average Precision (mAP), average per-Class Precision (CP), average per-Class Recall (CR), average per-Class F1-score (CF1), the average Overall Precision (OP), average overall recall (OR) and average Overall F1-score (OF1). We report the results obtained for our approach using two types of settings; that is, **(IML-GCN with $\mathbf{F}_I$)** and **(IML-GCN with $\mathbf{F}_{PCA}$)** using image embeddings without and with PCA respectively.

Table 3.1 reports the quantitative results obtained on the MS-COCO dataset. It can be clearly seen that although our models are noticeably smaller than others, they outperform state-of-the-art methods in terms of the mAP. Specifically, our approach achieves an mAP of 86.62% using only 31.5M parameters. Thus, it outperforms ML-GCN by 3.62% and requires around 30% less parameters. Similarly, our approach slightly registers higher mAP than ASL with 0.2% increase, while requesting 42% less parameters. Also, it is observed that the model using the IML-GCN with PCA performs better than the model without PCA in terms of mAP. We can see an improvement of 0.8%. Moreover, it can be noted that 2M fewer parameters are needed when using PCA. Therefore, the obtained results show the interest in applying PCA to the image feature embeddings.

Performance on VG-500: The Visual Genome dataset [54] is another large-scale multi-label image dataset that contains a total of 108,077 images, which covers over thousands of categories. Given that the distribution of the labels is quite sparse, the VG-500 subset [56] that consists of the 500 most frequent objects as categories is used. It is divided into a training set of 98,249 training images and 10,000 test images.

In Table 3.2, we compare our model with recent approaches. It can be seen that we achieve an mAP of 34.5% which is higher than the score reported for ResNet-101 [9], ML-GCN [39] and ASL (TResNetM) [21]. We also note that ResNet101 and ML-GCN employed a larger backbone CNN network, ResNet-101 leading to a higher number of parameters. Only ASL uses fewer number of parameters, which is fair since this network represents the direct backbone of our model. It can be noted that **C-Tran** [27] is the only approach that outperforms our method in terms of mAP. However, they rely on extremely large models. Indeed, the mAP result of 38.4% obtained in [27] used a model that is roughly 273% larger than the model that we propose in this paper, as it can deduced from Table 3.2. The extremely large size of the model in [27] places a limitation on its practical usefulness when considering the high computational resource and latency it incurs. Importantly, our proposed model requires modest computational resources and gives interesting results.

3.4.3 Impact of GCN input features:

This section reports the results of experiments, which were performed to study the performance improvements obtained using the proposed image-feature embeddings as input features for the GCN in comparison to word embeddings. For these experiments, the proposed framework (CNN-GCN architecture) remains the same except that the GCN of IML-GCN is replaced with the graph subnetwork of ML-GCN. We report the performance improvements for three different settings in Table 3.3.

Word embeddings: We use the same GCN subnetwork proposed for ML-GCN [39]. Table 3.3 shows that using the original GCN which incorporates word embeddings as node features lead to a visible decrease of 5% and 1.9% in mAP on MS-COCO and VG-500, respectively. This is expected as the used word embeddings are not relevant to the task of image classification and confirm our assumption.

Image embeddings: When the word embeddings are replaced by the d-dimensional embeddings generated using latent image representations, there is a significant improvement in the accuracy for the two benchmarks as reported in Table 3.3. This shows that the proposed image-feature embeddings can provide more robust representations in comparison to word embeddings.

Table 3.3: Impact of GCN input features.

Dataset	Refinements	mAP (%)
COCO	Word embeddings ($\mathbf{F}_W$)	81.6
	(+) Image based-feature embeddings ($\mathbf{F}_I$)	85.9 (**+4.3**)
	(+) Image based-feature embeddings PCA ($\mathbf{F}_{PCA}$)	86.62 (**+0.7**)
VG-500	Word embeddings ($\mathbf{F}_W$) ML-GCN [39]	32.6
	(+) Image based-feature embeddings ($\mathbf{F}_I$)	33.39 (**+0.8**)
	(+) Image based-feature embeddings PCA ($\mathbf{F}_{PCA}$)	34.47 (**+1.1**)

Image based-feature embeddings PCA: As discussed earlier, PCA is applied to the generated d-dimensional embeddings to obtain N-dimensional feature embeddings with improved signal-to-noise ratio. The results given in Table 3.3 show that applying PCA to the image embeddings improves the performance of the proposed model by 0.7% and 1.1% on MS-COCO and VG-500, respectively.

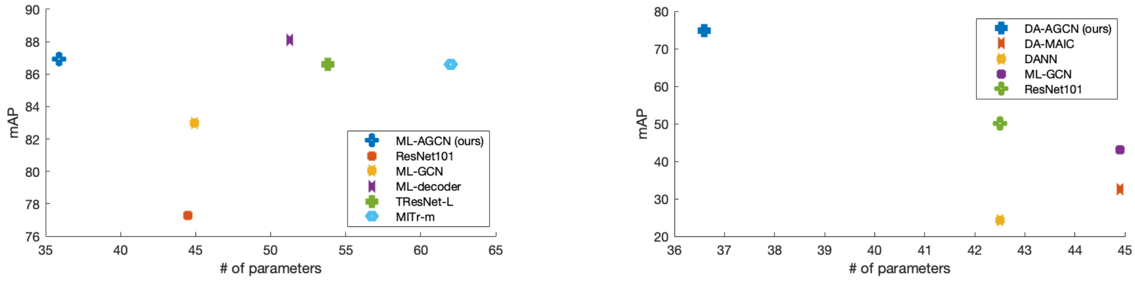
3.5 Conclusion

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

Chapter 4

Multi-label Image Classification using Adaptive Graph Convolutional Networks: from a Single Domain to Multiple Domains

In the previous chapter, we highlight the relevance of image-based node embeddings for representing labels in a CNN-GCN-based framework for the task of MLIC. In this chapter, we further investigate the limitations of existing graph-based approaches and their applicability in a cross-domain scenario. As mentioned in Chapter 2.1, graph-based methods have been largely exploited in the field of multi-label classification, given their ability to model label correlations. Specifically, their effectiveness has been proven not only when considering a single domain but also when taking into account multiple domains, as highlighted in Chapter 2.3. However, the topology of the used graph is not optimal as it is pre-defined heuristically. In addition, consecutive Graph Convolutional Network (GCN) aggregations tend to destroy the feature similarity. To overcome these issues, an architecture for learning the graph connectivity in an end-to-end fashion is



(a) Without UDA

(b) With UDA

Figure 4.1: Comparison of our approach (ML-AGCN) (a) without and (b) with UDA to recent state-of-the-art methods in terms of number of parameters (millions) and mean Average Precision (mAP) on MS-COCO and Clipart → VOC. The considered state-of-the art methods are: MlTr-m [29], TResNet-L [24], ML-Decoder [59], ML-GCN [39], ResNet101 [9], DA-MAIC [47], and DANN [30].

introduced. This is done by integrating an attention-based mechanism and a similarity-preserving strategy. The proposed framework is then extended to multiple domains using an adversarial training scheme. Numerous experiments are reported on well-known single-domain and multi-domain benchmarks. The results demonstrate that our approach achieves competitive results in terms of mean Average Precision (mAP) and model size as compared to the state-of-the-art. The code will be made publicly available.

4.1 Introduction

Thanks to the latest progress in deep learning, most of recent methods rely on a single-stream Deep Neural Networks (DNN), including Convolution Neural Networks (CNN) [9], [19], [20], [24], [60], [61], Residual Neural Networks (RNN) [49] and Transformers [27], [29], [54], [62]. Nevertheless, as highlighted in Chapter 1.1, their impressive performance comes at the cost of very large architectures that are unsuitable for memory-constrained environments. As an alternative, another line of research has tried to integrate priors related to label correlations [31], [39]–[41], [55], [63], [64]. As demonstrated in [31], such an approach contributes to improving scalability. In other words, fewer parameters are required to achieve comparable performance with traditional DNNs.

Graph-based approaches [31], [39], are among the most popular multi-label classification methods that aim at modeling label correlations, both in single as well as cross-domains. Standard multi-label image classification methods such as [31], [39] usually assume that unseen images and training data are drawn from the same distribution, i.e. the same domain, hence ignoring a possible domain shift problem [47]. This leads, therefore, to poor generalization capabilities under cross-domain settings. Unsupervised Domain Adaptation (UDA) is a plausible solution to overcome this challenge without relying on costly annotation efforts [23]. UDA aims at learning domain invariant features to bridge the gap between a *source* domain and a *target* domain without access to the associated labels. In particular, one of the most successful UDA approaches for multi-label classification, namely DA-MAIC [47], leverages graph representations to model label inter-dependencies and couple it with an adversarial training approach [30]. While most existing multi-label classification methods can be extended to the cross-domain setting by just adding a simple discriminator, they mostly require a high number of parameters to work effectively. Hence, the use of a graph-based method for modeling label correlation is an interesting way for obtaining compact yet effective models, as highlighted in [47]. This allows achieving a good compromise between performance and compactness under the cross-domain setting.

Despite their usefulness in both single and cross-domain settings, graph-based methods [31], [39]–[41], [47], [55], [63], [64] are unfortunately subject to three major limitations, namely: (1) The graph structure is heuristically defined. In particular, it is computed based on the co-occurrence of labels in the training data. Hence, this topology might not be ideal for the specific task of multi-label image classification; (2) A threshold is empirically fixed for discarding edges with a low co-occurrence probability.

This means that infrequent co-occurrences are assumed to be noisy. Although this might be true in many cases, assuming that any rare event corresponds to noise does not always hold; and (3) it has been proven in [65] that successive aggregation operations in the GCN usually dissipate the node similarity in the original feature space, hence potentially leading to a decrease in terms of performance.

Herein, we posit that by integrating adequate mechanisms in graph-based approaches for addressing the aforementioned issues, it should be possible to reduce the network size even more while achieving competitive performance in both single-domain and cross-domain settings.

In this chapter, we present the proposed adaptive graph-based multi-label classification method called Multi-Label Adaptive Graph Convolutional Network (ML-AGCN) for both contexts, single domain and across domains. Our idea consists in: (1) learning two additional adjacency matrices in an end-to-end manner instead of solely relying on a heuristically defined graph topology. Note that no threshold is applied, avoiding the loss of weak yet relevant connections. In particular, the first learned graph topology computes the importance of each node pair. This is carried out by employing an attention mechanism similar to Graph Attention Networks (GAT) [66]. Nevertheless, even though learned, the latter does not ensure the conservation of feature similarity. Hence, the second graph structure is built based on the similarity between node features and overcomes the information loss happening through successive convolutions; (2) integrating the proposed adaptive graph-based architecture in an adversarial domain adaptation framework for aligning a labeled source domain to an unlabeled target domain. As shown in Figure 4.1, the results suggest that our method is competitive with respect to the state-of-the-art in terms of both mean Average Precision (mAP) and network size under the single domain and cross-domain settings.

The detail in this chapter is an extended version of [32]. In summary, the main contributions of our proposed work are given below:

1. We propose an adaptive label graph learning strategy that does not make use of any threshold, hence preventing information loss about weak connections.
2. Two parameterized label graphs have been proposed: 1) to quantify each label pair's importance and, 2) to preserve the node feature's dissimilarity.
3. An adversarial Domain Adaptation approach integrating the Adaptive Graph Convolutional Network (DA-AGCN) for multi-label image classification.
4. A deep qualitative and quantitative experimental analysis of the proposed method, both in a single domain as well as cross-domains, with respect to the state-of-the-art.

The remainder of this chapter is organized as follows. Section 4.2 reviews the state-of-the-art on multi-label image classification and domain adaptation for multi-label classification. In Section 4.3 and Section 4.4, the proposed method is detailed. Section 4.5 presents the experimental results and analysis. Finally, Section 4.6 concludes this work and highlights interesting future directions.

4.2 Related works

In this section, we start by presenting the state-of-the-art of multi-label image classification. Then, we focus on reviewing existing domain adaptation approaches for multi-label image classification.

4.2.1 Multi-label Image Classification

This subsection presents various works proposed for Multi-Label Image Classification (MLIC) within a single-domain context. We categorize these methodologies into two main groups based on their intuition and network architectures.

Single-stream deep neural networks

As discussed in Chapter 4.1, most multi-label classification methods [21], [24], [27], [29], [62], [67], [68] employ a single-stream DNN. More specifically, they mainly take inspiration from successful architectures proposed in the context of single-label image classification such as CNN, RNN and transformers. For example, [67] have used a pre-trained OVERFeat [69] model and have adapted it to multi-label image classification. [68] have leveraged the prediction of multiple CNN architectures pretrained on ImageNet [70] such as AlexNet [71] and VGG-16 [60]. Ridnik et al. [21] have employed TResNet [24] using a novel loss called Asymmetric Loss (ASL) that focuses more on positive labels than negative ones. TResNet introduced in [24] is based on a ResNet architecture with a series of modifications for optimizing the GPU network capabilities while maintaining the performance.

Recently, [27] attempted to leverage transformers for modeling complex dependencies among visual features and labels. Similarly, MlTr [29] combines the pixel attention and the attention among image patches to better excavate the transformer's activity in multi-label image classification. More recently, ML-Decoder [59] proposed a transformer-based classification head instead of the standard Global Average Pooling (GAP), for improving the generalization capability. On the other hand, another recent work introduced a novel attention module called Interventional Dual Attention (IDA) [28] that aims to mitigate contextual bias in visual recognition through multiple sampling interventions.

Multi-stream deep neural networks

Going deeper into the network enables the model to learn more abstract features from the image. However, as mentioned in Chapter 1.2.1, this comes at the cost of high memory requirements. Moreover, the aforementioned single-stream methods do not explicitly model the relationship between labels which can be an important semantic element to consider.

In order to incorporate the information of label correlations, a second class of methods have used a second subnetwork in addition to the main backbone. For instance, [19] have introduced a Spatial Regularization Net (SRN) to learn the underlying relationships between labels by generating label-wise attention maps. Similarly, [72] have employed image representations from intermediate convolutional layers to model both local and global label semantics.

Graphs can also be an interesting way for modeling label correlations [31], [39], [58], [62], while keeping the network size reasonable. ML-GCN [39] was among the pioneering works that utilized a Graph Convolutional Network (GCN) to learn interdependent label-wise classifiers and combine it with a standard DNN that learns discriminative image features. Similarly, IML-GCN [31] makes use of a similar GCN-CNN architecture. However, they suggest employing image-based embeddings as node features of the input graph as a replacement to the glove-based word embeddings [42] used in [39]. Indeed, GLOVE embeddings might be inappropriate for MLIC since they have has been initially proposed for representing words in the field of Natural Language Processing (NLP), while images are intrinsically different.

As discussed in Chapter 4.1, despite their proven performance in terms of both precision and network size, these graph-based approaches including ML-GCN and IML-GCN suffer from some weaknesses. First, the input graph is predefined based on label co-occurrences in the training set. As a result, the heuristically defined topology labels might be sub-optimal for MLIC. Second, a threshold is set empirically in order to ignore weak edges. This induces that rare co-occurrences are presumed to be noisy. Even though this assumption often holds, some infrequent events might occur in a real-life scenario. Lastly, as discussed in [65], the similarity of node features through multiple GCN layers might fade, potentially resulting in a performance decrease. The proposed

ML-AGCN aims at solving these issues by adaptively learning the graph topology in an end-to-end manner, while leveraging an adequate mechanism for preserving the feature node similarity.

4.2.2 Unsupervised Domain Adaptation for Multi-label Image Classification

Over the last years, DL methods have achieved a remarkable progress in the field of computer vision and pattern recognition. However, the effectiveness of DL approaches heavily depends on the availability of a large amount of annotated data. For mitigating the huge cost caused by data annotation, the field of unsupervised domain adaptation [30], [43], [73]–[75] has been widely investigated over the last decade. It aims at making use of an existing labeled dataset from a related domain called *source domain* to enhance the model performance on a domain of interest termed *target domain*, for which only unlabeled data are provided.

Unsupervised domain adaptation methods can be separated into two main categories. The first one [43], [73], [74] aims at explicitly reducing the domain gap by minimizing statistical discrepancy measures between the two domains. Alternatively, the second class of methods implicitly minimizes this domain gap by adopting an adversarial training approach [30], [75]. The main idea consists in using a domain classifier to play a min-max two-player game with the feature generator. This strategy is designed to enforce the generation of domain-invariant features that are sufficiently discriminative.

Nevertheless, most existing techniques focus on the task of single-label image classification. In fact, very few papers have considered domain adaptation for multi-label image classification [39], [76], [77],[34].

Among these rare references, we can mention ML-ANet [76], which explicitly minimizes the domain gap by optimizing multi-kernels maximum mean discrepancies (MK-

MMD) in a Reproducing Kernel Hilbert Space (RKHS). More recently, an adversarial approach has been adopted in [77] where a condition-based domain discriminator similar to conditional-GANs [78] has been employed. However, similar to the first category of methods for traditional multi-label image classification, these two approaches [76], [77] neglect the important information of label dependencies.

A graph-based approach called DA-MAIC has been then proposed as an alternative [47]. As in ML-GCN [39], they have proposed to build a graph for modeling label correlations based on label co-occurrences. Additionally, to reduce the domain shift between the source and target domains, they have used a domain classifier that is trained in an adversarial manner. Unfortunately, DA-MAIC is impacted by the same drawbacks affecting ML-GCN [39] as detailed in Chapter 4.1 and 4.2.1. More details regarding these issues are given in the next section.

Inspired by [79], a very recent work [34] called DDA-MLIC redefining the adversarial loss based on the multi-label classifier has been proposed, hence eliminating the need for an additional discriminator. Although this approach has shown promising results, it addresses the problem of domain adaptation from a different perspective. Indeed, the presented paper in this chapter aims to propose a suitable multi-label classification mechanism that could be beneficial for both single-domain and cross-domain settings while DDA-MLIC investigates a more adequate adversarial strategy. Therefore, the two methods (DDA-MLIC and ours) tackles two different yet complementary aspects of UDA for multi-label classification.

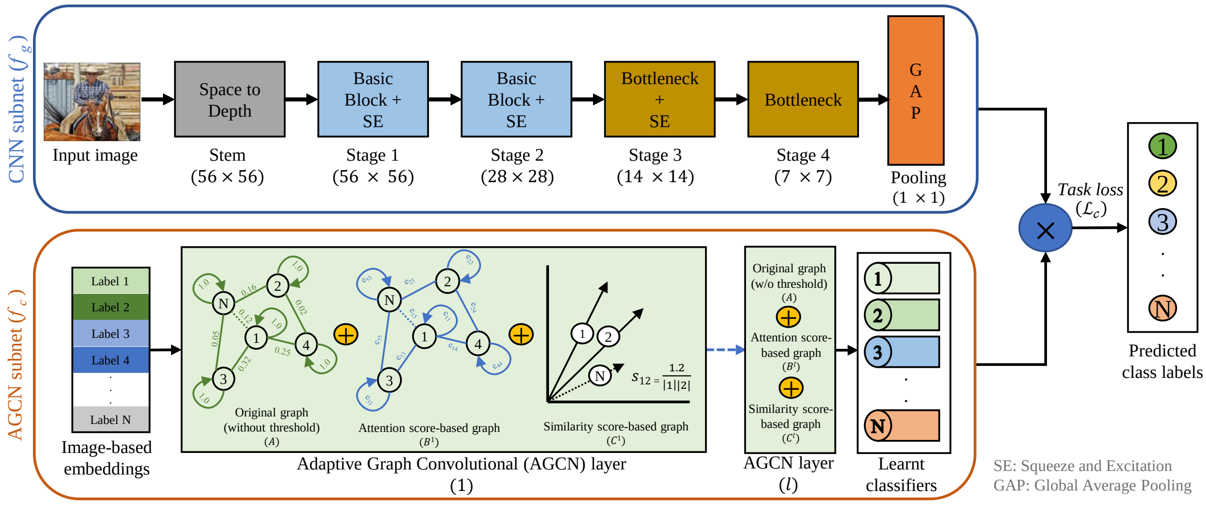


Figure 4.2: Architecture of ML-AGCN [32]: On the one hand, the CNN subnet learns relevant image features from an input image. On the other hand, the GCN subnet estimates interdependent label classifiers by taking into account one fixed adjacency matrix $\mathbf{A}$ and two adaptive adjacency matrices $\mathbf{B}^{(l)}$ and $\mathbf{C}^{(l)}$. Finally, the classifiers are applied to the CNN features for predicting the labels.

4.3 Multi-Label Adaptive Graph Convolutional Network (ML-AGCN)

As explained in Section 4.1, three main limitations can be noted in graph-based methods: (1) the computation of the adjacency matrix $\mathbf{A}$ is made heuristically and is decoupled from the training process; (2) a threshold τ (Eq. (2.5)) is empirically fixed for completely ignoring rare co-occurrences; and (3) as shown in [65], aggregating successively node features in a graph may induce the loss of the similarity/dissimilarity information present in the initial feature space. To handle these challenges, a novel graph-based approach called Multi-Label Adaptive Graph Convolutional Network (ML-AGCN) is introduced.

4.3.1 Overview of the Proposed Architecture

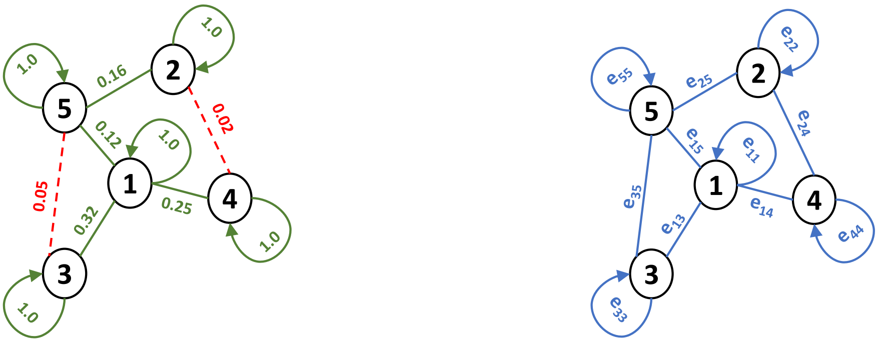
Similar to [39] and [31], a network formed by two subnets is adopted as illustrated in Figure 4.2. The first is a CNN that extracts a discriminative representation from a given input image, while the second is a GCN-based network that learns N interdependent classifiers. As in [31], TResNet-M which represents a small version of TResNet [24] is employed as a CNN subnetwork. TResNet is a direct extension of ResNet, which fully exploits the GPU capabilities to boost the model efficiency. The graph-based subnet, Adaptive Graph Convolutional Network (AGCN), uses the same image embeddings proposed in [31] as feature nodes. Nevertheless, in contrast to [31] and [39], it relies on an end-to-end learned graph topology. More details regarding this subnetwork are provided in Chapter 4.3.2. Similar to [31], the Asymmetric Loss (ASL) [21] denoted by $\mathcal{L}_c$ is used for optimizing ML-AGCN such that,

$$\begin{aligned}\mathcal{L}_c = \mathbb{E}_{(\mathbf{I}_s,\mathbf{y}_s)\sim\mathcal{D}_s} \sum_{i=1}^{N} y_s^{(i)}(1-p^{(i)})^{\gamma_+}\log(p^{(i)}) \\ + (1-y_s^{(i)})(p_m^{(i)})^{\gamma_-}\log(1-p_m^{(i)}),\end{aligned} \tag{4.1}$$

where γ_+ and γ_- are focusing parameters for positive and negative samples, respectively, and $y_s^{(i)}$ and $p^{(i)}$ are the respective ground truth and predicted probability with respect to the label i and $p_m^{(i)}$ is the shifted probability given by $\max(p^{(i)} - m, 0)$, where m is a threshold used for reducing the effect of easy negative samples [21].

4.3.2 Graph-based Subnet: Adaptive Graph Convolutional Network (AGCN)

Our intuition is that by integrating a suitable mechanism, it should be possible to boost the classification performance and reduce at the same time the size of graph-based



(a) Fixed graph $\tau = 0.1$ (b) Parameterized graph $\tau = 0$

Figure 4.3: (a) An example of a fixed label graph with a threshold set to $\tau = 0.1$ [32]. Dashed (red) edges indicate the ignored edges; (b) The proposed parameterized graph topology considering all the edges.

methods. Hence, as illustrated in Figure 4.3, we propose to adaptively learn the graph topology by reformulating Eq. (2.2) as below,

$$\mathbf{F}^{l+1} = \sigma((\mathbf{A} + \mathbf{B}^{(l)} + \mathbf{C}^{(l)})\mathbf{F}^{l}\mathbf{W}^{l}), \tag{4.2}$$

where σ is a LeakyRELU activation function.

Instead of relying solely on the adjacency matrix $\mathbf{A}$ defined in [39], two additional parameterized graphs called attention-based and similarity adjacency graphs, respectively denoted by $\mathbf{B}^{(l)}$ and $\mathbf{C}^{(l)}$, are defined. In this case, no threshold is applied to $\mathbf{A}$ for ignoring rare co-occurrences. It is also important to note that $\mathbf{A}$ is fixed, while $\mathbf{B}^{(l)}$ and $\mathbf{C}^{(l)}$ vary from one layer to another. In the following, we detail how $\mathbf{B}^{(l)}$ and $\mathbf{C}^{(l)}$ are computed.

Attention-based adjacency matrix

Instead of ignoring rare co-occurrences in $\mathbf{A}$, the matrix $\mathbf{B}^{(l)} = (b_{ij}^{(l)})_{i,j \in \mathcal{O}}$ is defined based on an attention mechanism where the importance of each edge is quantified. To that aim, inspired by [66], an attention score denoted by e_{ij} is calculated for each pair of vertices (v_i, v_j) as follows,

$$e_{ij} = \sigma(\mathbf{a}^{(l)^T}(\mathbf{W}\mathbf{F_i}^{(l)}||\mathbf{W}\mathbf{F_j}^{(l)}), \tag{4.3}$$

where $\mathbf{W} \in \mathbb{R}^{d^{(l+1)} \times d^{(l)}}$ represents a learnable weight matrix, $\mathbf{a}^{(l)^T} \in \mathbb{R}^{2d^{(l+1)}}$ are the learnable attention coefficients and || refers to the concatenation operation.

A softmax function is then applied to the computed normalized attention scores such that,

$$\alpha_{ij}^{(l)} = \frac{\text{exp}(e_{ij}^{(l)})}{\sum_{k \in \mathcal{N}(i)} \text{exp}(e_{ik}^{(l)})}, \tag{4.4}$$

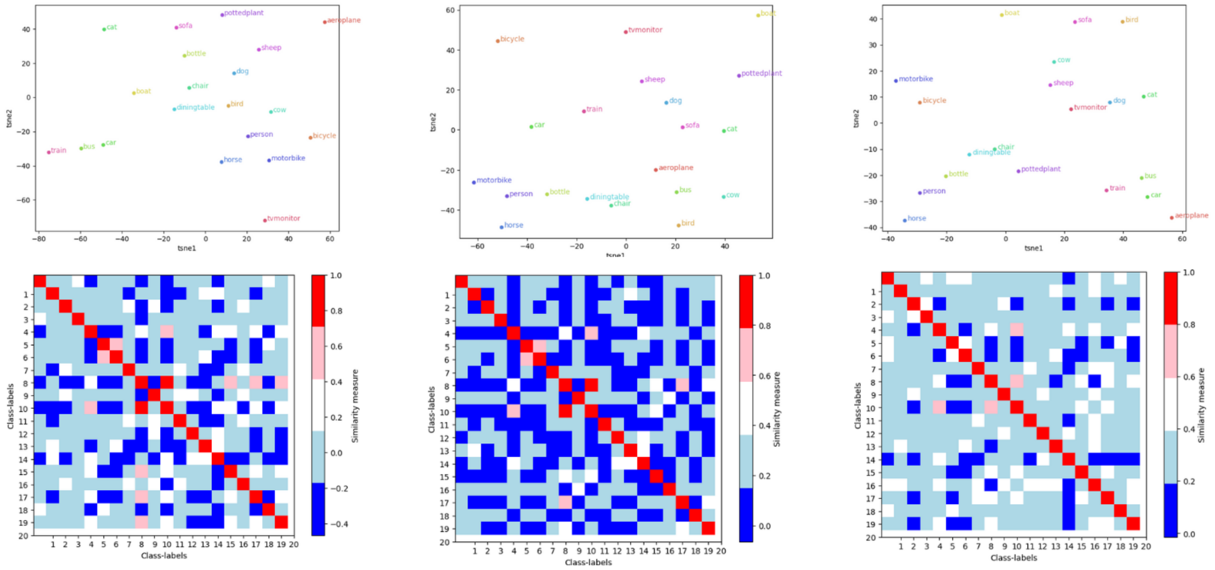
with $\mathcal{N}(i)$ defining the neighborhood of the node i and α_{ij} being the obtained normalized attention score.

We recall that the goal of the GCN subnet is to generate interdependent label classifiers (Chapter 2.1.2). This means that each classifier must be predominated by the information related to the label it belongs to. Hence, the attention score of the node in question should be maximal. For this purpose, an additional step called self-importance mechanism is proposed for computing the attention-based adjacency matrix $\mathbf{B}^{(l)} = (b_{ij}^{(l)})_{i,j \in \mathcal{O}}$ as follows,

$$\begin{cases} b_{ij}^{(l)} = \alpha_{ij}^{(l)} + \max\limits_{k \in \mathcal{O}}(\alpha_{ik}^{(l)}) \text{ if } i = j \\ b_{ij}^{(l)} = \alpha_{ij}^{(l)} \text{ if } i \neq j \end{cases} . \tag{4.5}$$

Similarity-based adjacency matrix $(\mathbf{C})$

As illustrated in the top row of Figure 4.4, the GCN aggregation process tends to modify the node similarity in the original feature space [65]. Subsequently, the second row illustrates a significant change in the similarity measure between the node features before and after the GCN operation (see Fig 4 (a) and (b), respectively). Although



(a) Original

(b) ML-GCN [39]

(c) Ours

Figure 4.4: Comparison of node feature similarity: The top row presents a tSNE visualization, while the bottom row illustrates cosine-similarity map between the graph nodes for VOC dataset: a) using the original image-based embeddings (before GCN), b) after applying two layers of standard GCN using the proposed architecture in ML-GCN [39], and c) after applying two layers of AGCN using our approach (i.e., ML-AGCN).

learned in an end-to-end manner, the attention-based graph $\mathbf{B}$ only quantifies the connectivity importance of each node pair and does not guarantee that the feature similarity is not lost through graph convolutions. Hence, we propose to incorporate an additional matrix $\mathbf{C}^{(l)} = (c_{ij}^{(l)})_{i,j\in\mathcal{O}}$ to preserve the node feature similarity. It is obtained by calculating the cosine similarity $c_{ij}^{(l)}$ between each pair of vertices (v_i, v_j) as follows,

$$c_{ij}^{(l)} = \frac{\mathbf{F}_i^{(l)} . \mathbf{F}_j^{(l)}}{\|\mathbf{F}_i^{(l)}\| \|\mathbf{F}_j^{(l)}\|}, \tag{4.6}$$

where $\|.\|$ denotes the L_2 Euclidean norm.

Finally, the output of the final layer L denoted by $\mathbf{F}^L$ is used in Eq. (2.1.2) for predicting the labels. As shown in Figure 4.4 (c), by using this strategy, the node similarity is relatively preserved, as compared to a standard GCN.

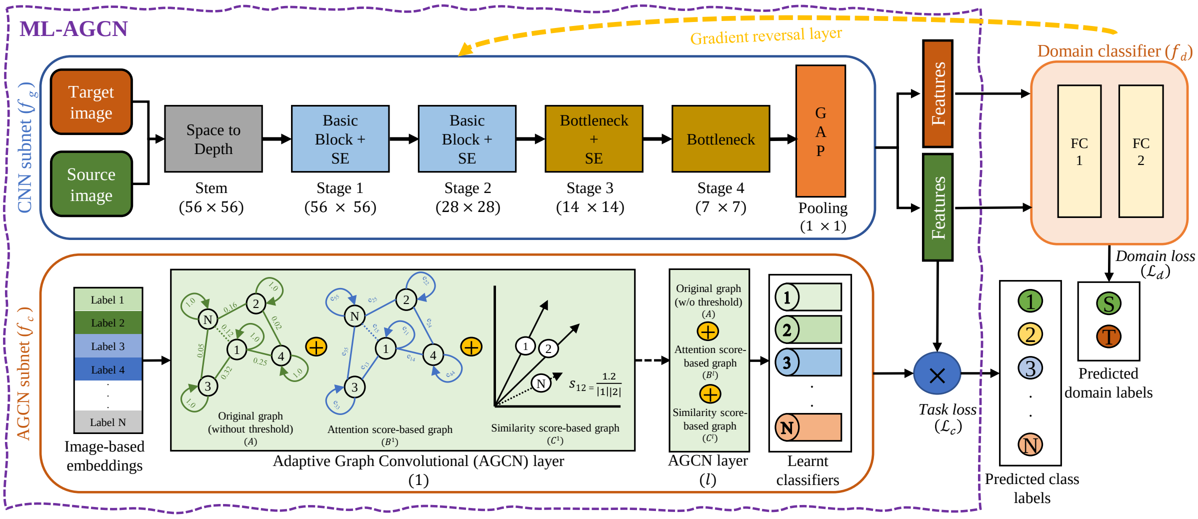


Figure 4.5: Architecture of the proposed DA-AGCN for multi-label image classification (best viewed in color). Images from both source and target datasets are given as input to the CNN subnet that generates image features. The AGCN-subnet, similar to ML-AGCN [32], learns in an end-to-end manner the attention and similarity-based adjacency matrices $\mathbf{B}^{(l)}$ and $\mathbf{C}^{(l)}$, respectively, and generates accordingly inter-dependent label classifiers using only labeled source images. In addition, a domain classifier is considered.

4.4 Unsupervised Domain Adaptation for Multi-label Image Classification using ML-AGCN

The proposed architecture for multi-label image classification called DA-AGCN is illustrated in Figure 4.5. Similar to ML-AGCN, we make use of TResNet-based backbone f_g to extract discriminative image features and an Adaptive Graph Convolutional Network f_c to learn N interdependent classifiers. Given source and target input images from $\mathcal{D}_s$ and $\mathcal{D}_t$, respectively, the goal of f_g is to generate D-dim domain-invariant image features. The latter is used to predict the image labels, as depicted in Eq. (2.1.2). The label classification loss $\mathcal{L}_c$ is therefore defined as in Eq. (4.1).

For obtaining domain-invariant representations, a domain classifier $f_d\colon \mathbb{R}^{d_f} \rightarrow [\![0, 1]\!]$ is employed and trained in an adversarial manner. Given the features $\mathbf{X} = f_g(\mathbf{I})$

extracted from a given image $\mathbf{I}$, f_d predicts the domain of the input image as follows,

$$\begin{aligned} f_d \colon \mathbb{R}^{d_f} &\to [0,1] \\ \mathbf{X} &\mapsto \hat{d}. \end{aligned} \tag{4.7}$$

where $\hat{d}$ is the predicted domain label of $\mathbf{I}$. Note that the ground-truth domain label $d = 0$ if $\mathbf{I}$ is from the source domain and $d = 1$ if sampled from the target domain.

As in [30], a domain loss is defined as below,

$$\mathcal{L}_d = \mathbb{E}_{f_g(\mathbf{I}_s)\sim\mathcal{D}_s} \log \frac{1}{f_d(f_g(\mathbf{I}_s))} + \mathbb{E}_{f_g(\mathbf{I}_t)\sim\mathcal{D}_t} \log \frac{1}{(1 - f_d(f_g(\mathbf{I}_t))}. \tag{4.8}$$

Given $\mathcal{L}_c$ defined in Eq. (4.1), the final objective function used to optimize the network is $E(\theta_g, \theta_d, \theta_c)$ defined by,

$$E(\theta_g, \theta_d, \theta_c) = \mathcal{L}_c(\theta_c, \theta_g) + \lambda \mathcal{L}_d(\theta_d, \theta_g), \tag{4.9}$$

where θ_g, θ_d and θ_c refer, respectively, to the parameters of f_g, f_d and f_c and λ is a hyper-parameter defining the weight of $\mathcal{L}_d$. The network is trained in an adversarial manner using the GRL for obtaining the optimal parameters $(\hat{\theta}_g, \hat{\theta}_c, \hat{\theta}_d)$ such that,

$$\begin{aligned} (\hat{\theta}_g, \hat{\theta}_c) &= \min_{\theta_g, \theta_c} E(\theta_g, \theta_c, \hat{\theta}_d), \\ \hat{\theta}_d &= \max_{\theta_d} E(\hat{\theta}_g, \hat{\theta}_c, \theta_d). \end{aligned} \tag{4.10}$$

4.5 Experiments

In this section, the experimental settings and results are presented and discussed for: (1) single-domain multi-label image classification (Chapter 4.5.1); and (2) domain-

Table 4.1: Comparison with the state-of-the-art methods on the MS-COCO dataset. (Best and second-best performances are indicated in bold and are underlined, respectively).

Category	Method	Resolution	# params	mAP	CP	CR	CF1	OP	OR	OF1
CNN	Multi-Evidence [20]	448x448	49.0	-	80.4	70.2	74.9	85.2	72.5	78.4
	SRN [19]	224x224	48.0	77.1	81.6	65.4	71.2	82.7	69.9	75.8
	ResNet101 [9]	448x448	44.5	77.3	80.2	66.7	72.8	83.9	70.8	76.8
	MCAR [61]	448×448	44.9	83.8	85	72.1	78	88	73.9	80.3
	MCAR [61]	576x576	44.9	84.5	84.3	73.9	78.7	86.9	76.1	81.1
	TResNet-L [21]	448x448	53.8	86.6	87.4	76.4	81.4	88.1	79.2	81.8
RNN	CNN-RNN [49]	224x224	66.2	61.2	-	-	-	-	-	-
Graph-based	ML-GCN (1-layer) [39]	448x448	43.1	80.9	82.9	69.7	75.8	84.8	73.6	78.8
	IML-GCN (1-layer) [31]	448x448	**29.5**	81.3	81.3	72.2	76.0	86.7	77.9	82.1
	ML-GCN [39]	448x448	44.9	83.0	85.1	72.0	78.0	85.8	75.4	80.3
	ML-GCN (TResNetM) [39]	448×448	31.9	82.4	87.6	66.6	75.2	91.0	70.3	79.3
	A-GCN [40]	512x512	44.1	83.1	84.7	72.3	78.0	85.6	75.5	80.3
	F-GCN [41]	-	44.3	83.2	85.4	72.4	78.3	86.0	75.7	80.5
	CFMIC [63]	448×448	45.1	83.8	85.8	72.7	78.7	86.3	76.3	81.0
	SSGRL [55]	576x576	92.2	83.8	**89.9**	68.5	76.8	**91.3**	70.8	79.7
	FLNet [64]	-	46.0	84.1	84.9	73.9	79.0	85.5	77.4	81.1
	IML-GCN [31]	448x448	31.5	86.6	78.8	**82.6**	80.2	79.0	**85.1**	81.9
Transformer-based	IDA [28]	576x576	55.1	86.3	-	-	80.4	-	-	82.5
	MlTr-s [29]	384x384	33.0	83.9	82.8	75.5	77.3	83	78.5	79.9
	STMG [62]	384x384	197.0	84.3	85.8	72.7	78.7	86.7	76.8	81.5
	C-Tran [27]	576x576	120.0	85.1	86.3	74.3	79.9	87.7	76.5	81.7
	MlTr-m [29]	384x384	62.0	86.8	84	80.1	**81.7**	84.6	82.5	83.5
	Ml-Decoder [59]	448x448	51.3	**88.1**	-	-	-	-	-	-
Ours: ML-AGCN (1-layer)		448×448	29.9	86.7	79.6	82.4	80.7	79.8	84.5	82.1
Ours: ML-AGCN (2-layers)		448×448	35.9	86.9	86.2	78.3	**81.7**	87.2	80.7	**83.8**

adaptation for multi-label image classification (Chapter 4.5.2).

4.5.1 Multi-label Image Classification

Implementation details

As discussed in Chapter 4.3, TResNet-M [24] is used as the CNN backbone. In particular, the fully connected layer after the Global Average Pooling (GAP) layer is removed. The output of the GAP layer produces a 2048-dimensional latent image representation. The dimension of the AGCN output has also been set to 2048. For the node features, we use the image-based embeddings proposed in [31] instead of the usual word-based embeddings used in [39]. The model is trained for 40 epochs using *Adam*, with a maximum learning rate of $1e-4$ using a cosine decay.

Datasets

In the following, the datasets that have been used for the experiments are presented.

MS-COCO MS-COCO [1] is a widely used large-scale multi-label image dataset. It contains 80K training images and 40k testing images. Each image is annotated with multiple object labels from a total of 80 categories.

VG-500 The VG-500 dataset [54] is a well-known dataset for multi-label image classification. It includes 500 different objects as categories. The dataset comes with a training set of 98,249 images and a testing set of 10,000 images.

PASCAL-VOC 2007 The PASCAL Visual Object Classes Challenge [80] introduced in 2007 is one of the most commonly used multi-label image classification datasets. It contains about 10K image samples with 5011 and 4952 images as training and testing sets, respectively. The images show 20 different object categories with an average of 2.5 categories per image.

Quantitative analysis

Comparison with state-of-the-art methods in terms of mAP and model size We compare the performance of ML-AGCN with current state-of-the-art methods by reporting the mean Average Precision (mAP) as well as the number of model parameters. Additionally, similar to [31], [39], we report the following evaluation metrics on the MS-COCO dataset, namely, average per-Class Precision (CP), average per-Class Recall (CR), average per-Class F1-score (CF1), average Overall Precision (OP), average overall recall (OR) and average Overall F1-score (OF1).

Table 4.1, 4.2 and 4.3 report the quantitative comparison of the proposed approach with respect to state-of-the-art methods on the MS-COCO, the VG-500, and the VOC datasets, respectively. It can be clearly seen that our method achieves competitive results as compared to existing methods in terms of mAP while considerably reducing the model size. More specifically, similar to SSGRL, we achieve the best mAP performance on VOC-2007 while reducing the number of parameters from 92.2 to 35.8 million. Moreover, ML-AGCN reaches the second-best mAP performance on MS-COCO and VG-500. However, as indicated in Table 4.1 and Table 4.2, our method achieves the second-best performance on MS-COCO and VG-500. Specifically, ML-decoder [59] and C-Tran [27] outperform our approach by 1.2% and 0.5% in mAP on MS-COCO and VG-500. This slight difference in performance might be explained by the fact that ML-decoder and C-Tran incorporate a noticeably higher number of parameters, i.e., 51.3 and 120 million, respectively, against only 35.9 and 32.1 million for ML-AGCN. Consequently, our approach achieves comparable performance while necessitating around 30% and 70% less parameters than ML-decoder and C-Tran, respectively. Such results show that our approach can maintain competitive performance while considerably reducing the network as compared to the state-of-the-art.

It is also worth mentioning that ML-AGCN outperforms the ML-GCN [39] baseline by 3.9%, 6.3%, and 0.5% in terms of mAP on the MS-COCO, VG and VOC datasets, respectively, while keeping a comparable number of parameters.

Moreover, for a fair comparison with this baseline, we re-train ML-GCN by replacing the original CNN backbone (ResNet101) with the same architecture used in our experiments, namely, TResNetM. The results that are reported in Table 4.1, Table 4.2, and Table 4.3 show that ML-AGCN outperforms ML-GCN with a higher margin when using the same backbone, thereby confirming the relevance of the proposed adaptive graph convolution module.

In addition, we also report that the obtained performance when including only one layer in the graph-subnet of ML-GCN [39], IML-GCN [31] and ML-AGCN. The obtained results confirm the relevance of the proposed adaptive learning. In fact, it can be seen

Table 4.2: Comparison with the state-of-the-art methods on the VG-500 dataset.(Best and second to the best performance is indicated in bold and underline respectively).

Method	# params	mAP	CP	CR	CF1	OP	OR	OF1
ML-GCN (TResNetM) [39]	33.2	26.6	30.7	7.7	10.3	**75.4**	10.0	17.7
ResNet101 [9]	44.5	30.9	39.1	25.6	31.0	61.4	35.9	45.4
ML-GCN [39]	44.9	32.6	42.8	20.2	27.5	66.9	31.5	42.8
TResNet-M [21]	**29.5**	33.6	-	-	-	-	-	-
IML-GCN [31]	32.1	34.5	-	-	-	-	-	-
SSGRL [55]	92.2	36.6	-	-	-	-	-	-
KGGR [54]	45.0	37.4	47.4	24.7	32.5	66.9	36.5	47.2
C-Tran [27]	120.0	**38.4**	**49.8**	27.2	**35.2**	66.9	39.2	49.5
Ours: ML-AGCN (1-layer)	32.1	37.9	47.2	**31.8**	34.7	64.1	**42.1**	**50.8**
Ours: ML-AGCN (2-layers)	37.4	37.1	47.3	24.7	29.0	67.6	38.1	48.7

Table 4.3: Comparison with the state-of-the-art methods on the VOC-2007 dataset. (Best and second to the best performance is indicated in bold and underline respectively).

Methods	# params	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
CNN-RNN [49]	66.2	84.0	96.7	83.1	94.2	92.8	61.2	82.1	89.1	94.2	64.2	83.6	70	92.4	91.7	84.2	93.7	59.8	93.2	75.3	**99.7**	78.6
VeryDeep [60]	138	89.7	98.9	95.0	96.8	95.4	69.7	90.4	93.5	96.0	74.2	86.6	87.8	96.0	96.3	93.1	97.2	70.0	92.1	80.3	98.1	87.0
ResNet101 [9]	44.5	89.9	99.5	97.7	97.8	96.4	65.7	91.8	96.1	97.6	74.2	80.9	85	98.4	96.5	95.9	98.4	70.1	88.3	80.2	98.9	89.2
HCP [68]	138.0	90.9	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
ML-GCN [39]	44.9	94.0	99.5	98.5	98.6	98.1	80.8	94.6	97.2	98.2	82.3	95.7	86.4	98.2	98.4	96.7	99.0	84.7	96.7	84.3	98.9	93.7
ML-GCN (TResNetM) [39]	31.8	94.1	99.8	98.0	**98.9**	98.0	80.4	95.9	96.0	97.8	83.4	98.6	86.4	98.6	**99.0**	95.1	98.8	82.7	98.8	84.5	**99.7**	92.0
F-GCN [41]	-	94.1	99.5	98.5	98.7	98.2	80.9	94.8	97.3	98.3	82.5	95.7	86.6	98.2	98.4	96.7	99.0	84.8	96.7	84.4	99.0	93.7
FLNet [64]	46.0	94.4	99.6	98.1	**98.9**	97.9	84.6	95.3	96.2	96.5	**85.6**	96.1	87.2	97.7	98.6	97.0	98.1	86.5	97.4	86.5	98.8	90.8
TResNet-L [21]	53.8	94.6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CFMIC [63]	45.1	94.7	99.7	98.5	98.8	**98.3**	83.9	96.5	97.5	98.8	83.1	96.1	87.4	98.6	98.9	**97.2**	99.0	85.4	97.1	84.9	99.2	94.2
MCAR [61]	44.9	94.8	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**
SSGRL [55]	92.2	**95.0**	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
Ours ML-AGCN (1-layer)	29.5	94.5																				
Ours ML-AGCN (2-layers)	35.8	**95.0**	**99.9**	98.0	98.5	98.0	81.6	96.8	96.6	98.2	**85.6**	**99.4**	88.2	**99.2**	**99.0**	96.5	98.8	84.8	**99.5**	**88.1**	98.9	94.5

that our method outperforms existing graph-based methods under this setting. More specifically, ML-AGCN achieves an improvement of 5.7% and 5.3% in terms of mAP when compared to ML-GCN and IML-GCN, respectively, on the MS-COCO dataset. Similarly, on the VG-500 benchmark, a significant increase of 20.7% is recorded in terms of mAP in comparison to IML-GCN.

Ablation study In order to analyze the impact of each adjacency matrix used in our approach, namely, the attention-based matrix $\mathbf{B}$ and similarity-based matrix $\mathbf{C}$, an ablation study is carried out as shown in Table 4.4. It can be noted that by considering $\mathbf{B}$ in addition to $\mathbf{A}$ (without threshold), an improvement of 5.1%, 20.5%, and 0.04% in terms of mAP is made, respectively, on MS-COCO, VG-500, and VOC-2007. The magnitude of improvement seems dependent on the number of classes contained in the considered dataset. In fact, while VG-500 is formed by 500 classes, MS-COCO and VOC-2007 are respectively composed of 80 and 20 categories. This highlights the

Table 4.4: The ablation of adaptively learning $\mathbf{B}$ and $\mathbf{C}$ for multi-label image classification in single domain.

Method	**mean Average Precision (mAP %)**		
	MS-COCO	**VG-500**	**VOC-2007**
ML-AGCN (A)	81.1	17	94.27
ML-AGCN (A+B)	86.6 **(+5.1%)**	37.5 **(+20.5%)**	94.31 **(+0.04%)**
ML-AGCN (A+B+C)	86.7 **(0.1%)**	37.9 **(+0.4%)**	95.0 **(+0.69%)**

Table 4.5: Hyper-parameter analysis: comparison of performance of ML-AGCN when varying the number of AGCN layers. Best performance is indicated in **bold**.

# layers	**# params (millions)**	**mean Average Precision (%)**		
		MS-COCO	VG-500	VOC-2007
1-layer	29.9	86.7	37.9	94.5
2-layers	35.9	**86.9**	**37.1**	**95.0**
3-layers	37.3	85.7	24.6	94.7
4-layers	40.7	85.4	-	94.6

importance of adaptively modeling label correlations, especially when dealing with a large number of classes, which is likely to occur in a practical scenario. An additional mAP improvement of 0.1%, 0.4%, and 0.17% can be seen when including $\mathbf{C}$ in the AGCN subnet. In conclusion, it can be noted that $\mathbf{B}$ contributes more importantly to the resulting enhancement. This could be explained by the fact that $\mathbf{C}$ is less needed, as only two layers are considered in the graph subnet. As shown in Fig 4., while the use of C allows to better preserve the feature similarity as, the latter is not completely lost through layers when using 2 standard GCN convolutions.

Hyper-parameter analysis Table 4.5 reports the mean Average Precision (mAP) for MS-COCO, VG-500, and VOC-2007 datasets, when varying the total number of layers in ML-AGCN. The best results are generally obtained when only two layers are considered. However, it should be noted that the results are very stable for MS-COCO and VOC-2007, with a variation lower than 1.5% for all the configurations.

Qualitative analysis

In Figure 4.6, the Gradient-weighted Class Activation Mapping (Grad-CAM) [81] is visualized for some examples from the VOC dataset. While the first column of images

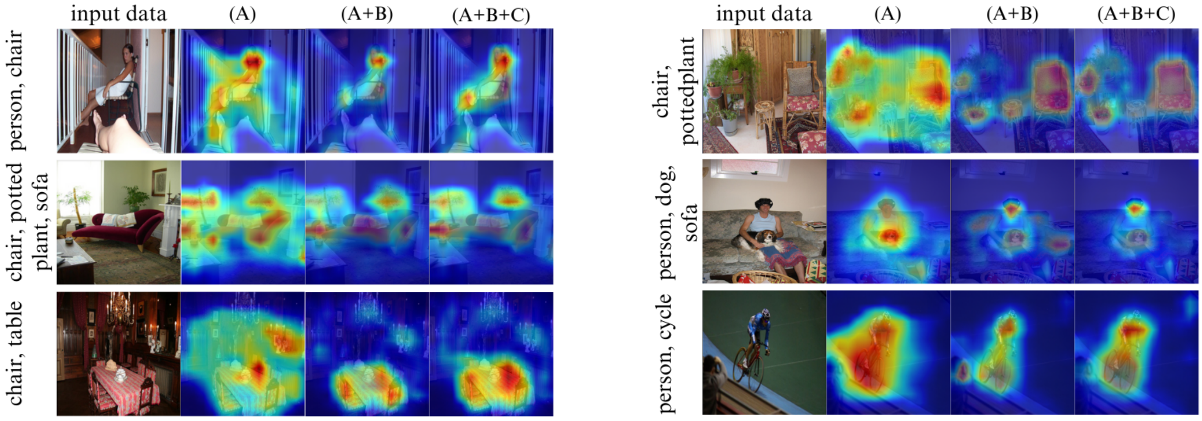


Figure 4.6: Grad-CAM visualization of the predictions with ML-AGCN [32] using samples from the VOC dataset using only $\mathbf{A}$, then $\mathbf{A}$ and $\mathbf{B}$, and finally $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$.

represents the input images, the second, third, and fourth columns show, respectively, the Grad-cam visualization from a model trained using only $\mathbf{A}$, $\mathbf{A}$ and $\mathbf{B}$, and finally $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. These qualitative results are in line with the quantitative ones. As shown in Figure 4.6, the use of $\mathbf{B}$ allows activating more precisely the regions of interest in the image, while the contribution of $\mathbf{C}$ in this refinement is less impressive but remains visible.

4.5.2 Unsupervised Domain Adaptation for Multi-label Image Classification

Implementation details

We reproduce the results of current state-of-the-art methods due to the limited availability of DA approaches for multi-label image classification. In particular, we first consider standard multi-label image classification methods (without DA) and refer to them as MLIC. Since no target images were employed during the learning process, this is equivalent to source-only training. Second, we evaluate the zero-shot performance of existing Vision Language Models (VLM) on the target dataset referred to as VLM in our experiments. For this purpose, we use a pre-trained Contrastive Language-Image Pre-Training (CLIP) [82] model with a ViT-B/32 transformer-based backbone. As commonly done in MLIC [83], we compute a similarity score, given an input image to VLM, between the predicted text features and ground truth features of the considered labels. The application of a sigmoid activation function to these scores results in the presence probabilities of the considered objects. Lastly, we reproduce the outcomes of two state-of-the-art domain adaptation methods, namely DANN [30] and DA-MAIC [47], that we refer as DA in our experiments. Moreover, we replace the multi-label softmargin-loss by the traditional cross-entropy loss when training DANN. We generate Glove-based word embeddings [42] as node features for training DA-MAIC. Additionally, in order to showcase the effectiveness of the proposed AGCN subnet, we present the results of DANN and DA-MAIC by considering the same backbone as ours: an additional experiment based on TResNet-M instead of the traditional ResNet101 is carried out. The images have been resized to 224 × 224, unless stated differently. Our domain classifier includes one hidden layer of dimension 1024. A maximum learning rate of $1e^{-4}$ using a cosine decay is considered. The model is trained for a total of 40 epochs or until convergence.

Datasets

Similar to DA-MAIC [47], we use three multi-label aerial image datasets in our experiments, namely, AID [84], UCM [85] and DFC15 [86]. Additionally, due to the limited number of suitable datasets for the task of DA in MLIC, we convert two well-known object detection datasets, initially used for DA in the context of object detection, to multi-label annotations, namely PASCAL VOC 2007 [80] and Clipart1k [87].

AID multi-label aerial dataset The original AID dataset [88] contains 10000 high-resolution aerial images. The images cover a total of 30 categories. A multi-label version of this dataset was produced in [84], where 3000 aerial images from the original AID dataset have been selected and assigned with multiple object labels. In total, they include 17 labels: airplane, sand, pavement, buildings, cars, chaparral, court, trees, dock, tank, water, grass, mobile-home, ship, bare-soil, sea, and field. 80% and 20% of the images have been used respectively for training and testing [84].

UCM multi-label aerial dataset UCM multi-label dataset [85] is derived from the UCM dataset [89]. It consists of images showing 21 land-use classes. The images have a resolution of 256 x 256 pixels. Later in [85], 2100 of these aerial images were annotated with multiple tags in order to generate a multi-label aerial image dataset. This dataset shares the same number of labels as AID multi-label dataset [84] i.e., 17 labels. In our experiments, 80% and 20% of data are respectively used for training and testing.

DFC15 multi-label aerial dataset The DFC15 multi-label dataset [86] was initially introduced in 2015. It has a total of 3342 high-resolution image samples and includes 8 object labels. In our experiments, the 6 categories in common with UCM and AID datasets, including water, grass, building, tree, ship, and car, are considered. 80% and 20% are respectively used for training and testing.

VOC and Clipart1k datasets The Clipart1k [87] dataset contains 20 object categories, similar to PASCAL-VOC 2007 [80]. We create a multi-label annotation for each image by considering the category of each object bounding box. Clipart1k provides a total of 1000 image samples. 50% and 50% are respectively used for training and testing.

Quantitative analysis

Comparison with state-of-the-art methods in terms of mAP and model size The proposed domain adaptation approach for multi-label image classification is compared

Table 4.6: Comparison with the state-of-the-art in terms of mAP and number of model parameters using two settings, i.e., AID → UCM and UCM → AID. (Best performance is indicated in bold and second-best performance is underlined).

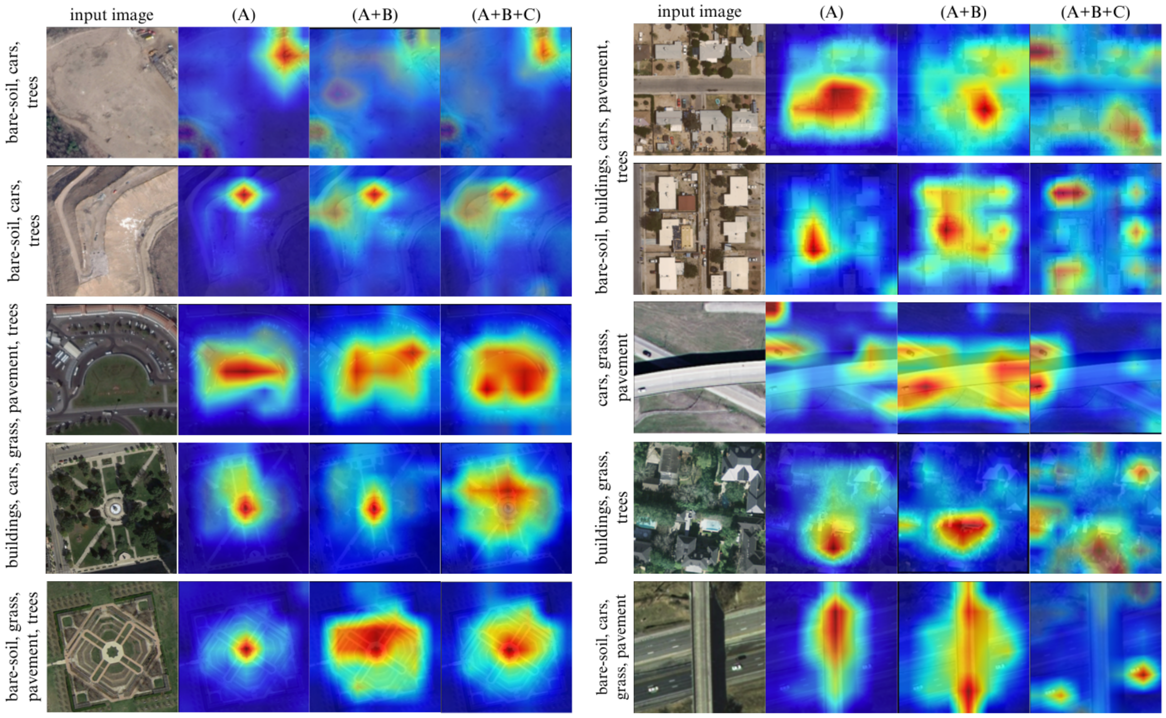
Category	**Backbone**	**Method**	**# params**	**AID → UCM**							**UCM → AID**						
				mAP	**CP**	**CR**	**CF1**	**OP**	**OR**	**OF1**	**mAP**	**CP**	**CR**	**CF1**	**OP**	**OR**	**OF1**
MLIC	ResNet50	RESNET [9]	23.5	54.5	57.9	42.3	43.2	70.2	67.1	68.6	50.3	57.2	26.8	32.2	90.0	44.2	59.3
	ResNet101	RESNET [9]	42.5	57.5	**60.0**	47.5	47.0	69.1	71.5	**70.3**	51.7	50.6	29.6	33.9	88.0	48.5	62.5
		ML-GCN [39]	44.9	53.7	55.3	44.3	45.9	70.2	68.7	69.4	51.3	50.1	29.9	34.0	88.0	49.7	63.6
	TResNet-M	ML-GCN [39]	31.8	53.5	57.8	40.5	41.1	64.2	70.4	67.2	52.5	51.1	25.8	31.9	**91.6**	40.8	56.5
		ASL [21]	29.4	55.4	48.7	52.8	47.1	58.7	79.1	67.4	54.1	54.5	40.2	41.9	85.4	65.1	73.9
		ML-AGCN [32]	36.6	55.2	36.6	**64.9**	45.1	45.0	**88.1**	59.6	52.1	48.2	**47.4**	**42.9**	77.1	**79.8**	**78.4**
VLM	ViT-B/32	CLIP [82]	151.3	42.2	45.8	25.2	28.2	51.2	23.6	32.3	42.4	44.8	21.8	22.4	54.1	15.9	24.5
DA	ResNet101	DANN [30]	42.5	55.3	54.7	57.9	52.8	59.0	81.6	68.5	50.5	60.4	24.9	32.3	89.2	42.2	57.3
		DA-MAIC [47]	44.9	49.7	52.4	48.7	45.4	56.9	72.5	63.8	48.7	51.6	40.9	41.5	78.9	65.5	71.6
	TResNet-M	DANN [30]	29.4	52.5	59.1	31.6	36.3	**70.9**	53.7	61.1	51.6	52.1	23.2	27.9	83.2	27.8	41.7
		DA-MAIC [47]	31.8	54.4	55.3	37.5	38.6	68.0	67.9	67.9	50.5	51.8	22.9	29.0	**91.6**	35.2	50.8
		DDA-MLIC [34]	29.4	**63.2**	52.5	63.7	**55.1**	59.4	82.8	69.2	54.9	53.9	30.4	35.5	84.6	41.0	55.3
		DA-AGCN (Ours)	36.6	59.0	54.0	59.4	52.3	57.1	82.3	67.4	**57.2**	**61.2**	40.0	41.8	86.1	66.9	75.3

Table 4.7: Comparison with the state-of-the-art in terms of mAP and number of model parameters using two settings, i.e., UCM → DFC and AID → DFC. (Best performance is indicated in bold and second-best performance is underlined).

Category	**Backbone**	**Method**	**# params**	**AID → DFC**							**UCM → DFC**						
				mAP	**CP**	**CR**	**CF1**	**OP**	**OR**	**OF1**	**mAP**	**CP**	**CR**	**CF1**	**OP**	**OR**	**OF1**
MLIC	ResNet50	RESNET [9]	23.5	52.9	52.3	47.1	44.2	52.3	50.5	51.3	67.7	58.5	31.9	34.9	65.9	39.1	49.1
	ResNet101	RESNET [9]	42.5	56.9	52.9	61.5	48.7	46.1	63.7	53.5	66.4	74.4	31.2	36.9	67.2	37.2	47.9
		ML-GCN [39]	44.9	58.9	56.7	57.9	45.8	45.7	65.0	53.7	64.6	72.4	32.0	35.6	64.4	38.9	48.5
	TResNet-M	ML-GCN [39]	31.8	53.5	52.0	47.9	41.6	52.7	51.1	51.9	66.6	60.2	35.0	38.1	64.9	39.4	49.0
		ASL [21]	29.4	56.1	49.6	68.4	49.9	43.5	74.1	54.8	68.9	66.3	53.1	44.0	52.6	57.0	54.7
		ML-AGCN [32]	36.6	51.6	41.5	**83.8**	52.3	40.2	**88.7**	55.3	70.3	68.4	56.1	47.8	53.8	58.5	56.0
VLM	ViT-B/32	CLIP [82]	151.3	64.3	**88.1**	22.3	31.4	**78.7**	22.0	34.4	63.8	**88.4**	22.7	31.3	**79.5**	22.5	35.1
DA	ResNet101	DANN [30]	42.5	64.3	57.0	65.9	51.4	48.0	65.9	55.6	67.2	72.8	44.7	48.7	56.3	49.1	52.4
		DA-MAIC [47]	44.9	50.1	54.5	45.8	38.9	44.2	48.8	46.4	65.6	69.7	54.2	52.8	57.1	58.4	57.7
	TResNet-M	DANN [30]	29.4	43.0	40.7	13.6	19.3	46.0	15.6	23.3	64.1	77.3	22.6	30.1	68.6	26.5	38.2
		DA-MAIC [47]	31.8	55.4	49.8	60.4	44.7	47.3	64.1	54.4	65.8	71.4	39.3	39.7	59.9	44.6	51.1
		DDA-MLIC [34]	29.4	62.1	47.6	75.5	55.3	48.9	76.2	**59.6**	70.6	67.2	55.7	49.3	55.0	58.4	56.6
		DA-AGCN (Ours)	36.6	**65.7**	51.8	78.1	**55.7**	45.2	80.8	58.0	**76.5**	68.5	**61.7**	**59.0**	60.0	**60.2**	**60.1**

Table 4.8: Comparison with the state-of-the-art in terms of mAP and number of model parameters using the two settings, i.e., VOC → Clipart and Clipart → VOC. (Best performance is indicated in bold and second-best performance is underlined).

Category	**Backbone**	**Method**	**# params**	**VOC → Clipart**							**Clipart → VOC**						
				mAP	**CP**	**CR**	**CF1**	**OP**	**OR**	**OF1**	**mAP**	**CP**	**CR**	**CF1**	**OP**	**OR**	**OF1**
MLIC	ResNet50	RESNET [9]	23.5	42.0	57.6	15.3	22.5	82.3	25.8	39.3	45.4	40.3	9.8	13.0	84.7	25.5	39.2
	ResNet101	RESNET [9]	42.5	38.0	64.8	14.3	22.5	82.3	18.3	29.9	50.1	66.2	17.5	25.5	83.9	29.6	43.7
		ML-GCN [39]	44.9	43.5	62.5	20.3	28.4	86.6	27.8	42.1	43.1	57.9	21.0	26.8	73.5	30.6	43.2
	TResNet-M	ML-GCN [39]	31.8	53.0	75.3	26.9	37.3	90.3	31.3	46.4	67.6	80.8	39.9	50.8	88.8	46.0	60.6
		ASL [21]	29.4	56.8	72.0	38.5	47.6	82.8	45.7	58.9	64.2	69.0	30.7	37.3	80.0	45.7	58.2
		ML-AGCN [32]	36.6	53.7	75.5	35.5	44.4	79.1	39.9	53.1	38.0	45.5	25.1	28.2	61.8	36.6	45.9
VLM	ViT-B/32	CLIP [82]	151.3	58.0	43.6	**57.8**	43.6	45.0	42.3	43.6	73.1	53.3	**76.6**	60.1	53.5	58.8	56.0
DA	ResNet101	DANN [30]	42.5	33.9	47.7	17.0	20.6	57.3	24.1	33.9	24.3	28.3	16.4	14.2	39.8	24.7	30.5
		DA-MAIC [47]	44.9	25.8	28.0	3.1	5.1	92.6	2.9	5.6	32.6	48.2	12.2	14.4	50.2	31.9	39.0
	TResNet-M	DANN [30]	29.4	40.0	82.4	17.2	27.4	**93.8**	17.5	29.5	67.0	76.8	23.3	32.6	**93.1**	20.4	33.4
		DA-MAIC [47]	31.8	62.3	77.4	42.6	51.6	83.1	**51.0**	**63.2**	74.3	84.5	53.9	63.0	83.7	57.7	68.3
		DDA-MLIC [34]	29.4	61.4	**84.7**	28.1	39.4	90.9	33.3	48.8	**77.0**	**86.9**	29.3	38.2	88.4	35.3	50.4
		DA-AGCN (Ours)	36.6	**62.9**	75.8	46.8	**54.0**	80.0	50.4	61.8	75.8	70.9	71.8	**69.8**	66.1	**75.6**	**70.5**


input image
(A)
(A+B)
(A+B+C)
bare-soil, cars, trees
bare-soil, cars, trees
buildings, cars, grass, pavement, trees
bare-soil, grass, pavement, trees
input image
(A)
(A+B)
(A+B+C)
bare-soil, buildings, cars, pavement, trees
cars, grass, pavement
buildings, grass, trees
bare-soil, cars, grass, pavement

(a) UCM → AID

(b) AID → UCM

Figure 4.7: Grad-CAM visualization of the multi-label predictions using the proposed DA-AGCN a) UCM → AID, and b) AID → UCM setting using only $\mathbf{A}$, then $\mathbf{A}$ and $\mathbf{B}$, and finally $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$.

with state-of-the-art methods. The same metrics described in Chapter 4.5.1 are used, including mAP, CP, CR, CF1, OP, OR and OF1. In addition to the four protocols followed in DA-MAIC [47], i.e., AID → UCM, UCM → AID, AID → DFC, and UCM → DFC, two more combinations VOC → Clipart and Clipart → VOC are provided. Two categories of methods are considered in our evaluation: (1) the conventional Multi-Label Image Classification (MLIC); and the Domain Adaptation-based (DA) methods that aim at explicitly reducing the gap between source and target datasets.

Table 4.6 reports the results obtained considering AID and UCM datasets. Two different settings are followed: AID → UCM, where AID is the source dataset and UCM is the target dataset, and UCM → AID, where UCM is the source dataset and AID is the target dataset. It can be clearly seen, in both types of DA settings, that the proposed DA-AGCN outperforms existing state-of-the-art for UCM → AID. However, we are the second best in terms of mAP for AID → UCM, where the recent discriminator-free DDA-MLIC [34] achieves a higher mAP. On the other hand, an improvement of mAP by 10.65% and 4.59% has been respectively recorded for AID → UCM and UCM → AID as compared with the DA-MAIC [47] baseline. This suggests that learning the graph topology in an end-to-end manner helps improve the performance in the presence of a domain shift. Moreover, the proposed method outperforms existing state-of-the-art VLM method by more than 17% in terms of mAP despite having approximately five times the lesser number of model parameters.

In Table 4.7, DA-AGCN is also compared with existing methods by considering the settings UCM → DFC and AID → DFC. It can be clearly noticed that DA-AGCN outperforms the existing works, including the the VLM and very recent DDA-MLIC, in terms of mAP and model size for both UCM → DFC and AID → DFC. More precisely, an improvement of 1.4% and 6.2% is achieved in terms of mAP compared to the second-best performing methods.

Table 4.8 also confirms the superiority of the proposed approach as compared to other state-of-the-art techniques. However, it is to note that the improvement given by the adaptive graph remains limited for Clipart → VOC, with a slightly lower performance than DDA-MLIC. This might be explained by the fact that DA for MLIC datasets include a relatively low number of classes (8 to 20 categories). Hence, the interest of adaptively modeling label correlations might be not entirely visible. This demonstrates the necessity of creating benchmarks for DA under an MLIC context, including a wider number of classes.

In summary, our approach achieves the best performance for 4 settings over 2 and the second-best performance for the two remaining ones, ranking just after DDA-MLIC. This highlights the relevance of the proposed adaptive method under the unsupervised domain adaptation settings. In future works, it will be interesting to study the complementarity of the two best performing approaches, namely, ours and DDA-MLIC.

Ablation study Table 4.9 reports the obtained results when considering each module comprised in DA-AGCN. More specifically, the first row reports the mAP score on the target dataset using a model trained on the source dataset without incorporating the learning of label correlations and without any domain adaptation. The results in the second row are obtained by incorporating the proposed adaptive graph learning strategy for modeling the label correlations (ML-AGCN). It can be clearly seen that adaptively learning the label graph topology based on the proposed graphs $\mathbf{B}$ and $\mathbf{C}$ leads to a significant performance improvement. More specifically, an mAP improvement of at least 16 to 20% for UCM $\rightarrow$ AID, AID $\rightarrow$ UCM and UCM $\rightarrow$ DFC can be observed. Note that the results in the second row are obtained without any domain adaptation. Finally, the last row reports the results when adding an additional domain classifier to implicitly minimize the domain gap. This further improves the mAP score by an additional 2 to 6%, compared to the ML-AGCN, across all benchmarks, by this remains marginal as compared to the enhancement induced by the graph structure adaptive learning. This might return to the fact that the domain gap in aerial datasets remains relatively small, hence benefiting more from the adaptive graph strategy than the alignment.

Additionally, in Table 4.10, we report the performance obtained on five datasets by incorporating different combinations of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. It can be noted in the first row that the original adjacency matrix $\mathbf{A}$ alone cannot effectively model the label relationship, hence leading to the lowest mAP score across almost all benchmarks. The next two rows of the table further showcase that using either $\mathbf{B}$ or $\mathbf{C}$ provides a significant performance improvement when compared to only $\mathbf{A}$, across all datasets except VOC and Clipart where $\mathbf{C}$ alone does not yield superior performance as compared to $\mathbf{A}$. This not surprising since $\mathbf{C}$ aims to preserve the node feature similarity, and is not expected

Table 4.9: Ablation study: impact of using an adversarial strategy and learning an adaptive graph topology.

Methods/Dataset (mAP)	**UCM → AID**	**AID → UCM**	**AID → DFC**	**UCM → DFC**
TResNet only	34.98	37.03	62.36	54.77
TResNet + AGCN	53.62 **(+18.64)**	53.08 **(+16.05)**	58.44 (-3.92)	74.4 **(+19.68)**
TResNet + AGCN + DC	55.79 **(+2.17)**	55.37 **(+2.29)**	65.11 **(+6.67)**	76.49 **(+2.04)**

Table 4.10: Ablation study: effect of adaptively learning $\mathbf{B}$ and $\mathbf{C}$ in the presence of a domain shift.

Graphs			**mean Average Precision (mAP %)**					
A	**B**	**C**	**AID → UCM**	**UCM → AID**	**AID → DFC**	**UCM → DFC**	**VOC → Clipart**	**Clipart → VOC**
✓			33.6	42.8	42.6	58.3	53.1	68.3
	✓		55.3	50.6	53.7	69.7	61.3	**80.0**
		✓	47.0	51.9	56.7	70.0	50.9	43.4
✓	✓		45.1	55.4	51.5	67.9	52.5	75.3
	✓	✓	54.7	52.7	60.0	71.8	58.6	76.1
✓		✓	48.1	54.5	53.7	71.6	62.0	75.6
✓	✓	✓	**59.0**	**57.2**	**65.7**	**76.5**	**62.9**	75.8

to model the label correlations. Furthermore, the next three rows show the variation in performance when coupling two to three matrices. For instance, combining either $\mathbf{B}$ or $\mathbf{C}$ with the original adjacency matrix $\mathbf{A}$ always provides a higher mAP when compared to only $\mathbf{A}$. Finally, the last row confirms the superiority of the proposed combination $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ as the highest mAP score across all benchmarks is reached except for Clipart → VOC. A possible explanation to this exception is the synthetic nature of the Clipart dataset, where the label co-occurrence is well-thought and might be representative enough.

Furthermore, the importance of preserving the node feature similarity is demonstrated in the last row, where a notable improvement in mAP is observed as compared to the $\mathbf{A} + \mathbf{B}$ setting. As compared to the single-domain setup, the improvement resulting from $\mathbf{C}$ is more significant (see Table 4.4, where the contribution of $\mathbf{C}$ is relatively marginal). This might be explained by the fact that in a cross-domain setting, an adversarial training is adopted for enforcing the generation of domain-invariant features. This min-max game known for its instability, might amplify the node feature dissimilarity phenomenon.

Table 4.11: Sensitivity analysis: performance of the proposed DA-AGCN using different values of the hyper-parameter (λ), defined in Eq. (4.9).

lambda (λ)	mean Average Precision (mAP %)					
	AID→UCM	UCM→AID	AID→DFC	UCM→DFC	VOC→Clipart	Clipart→VOC
0.1	52.2	52.5	56.5	68.4	51.2	73.9
0.2	53.3	54.9	53.1	70.6	52.8	**75.8**
0.3	53.5	52.0	58.2	**72.8**	52.6	71.6
0.4	51.7	**57.2**	57.0	70.5	50.0	74.2
0.5	51.6	56.7	55.1	71.1	52.1	73.9
0.6	**57.0**	56.3	**58.9**	69.8	52.6	72.7
0.7	54.7	52.2	56.9	71.1	52.7	72.5
0.8	50.4	53.1	55.4	69.3	52.5	73.8
0.9	53.5	54.2	55.7	72.1	**53.0**	75.3
1.0	53.4	53.9	56.4	70.7	52.5	74.0

Hyper-parameter analysis In Table 4.11, we report the mean Average Precision (mAP) on the six considered benchmarks when varying the hyper-parameter (λ), introduced in Eq (4.9). Specifically, we vary its value from 0.1 to 1.0. It can be noted that the obtained results are relatively stable with a variation in maP of more or less 5%.

Qualitative analysis

Grad-CAM [81] visualization in the presence of a domain shift can be seen in Figure 4.7. While Figure 4.7 (a) demonstrate these results for UCM $\rightarrow$ AID, Figure 4.7 (b) showcases the visualizations for AID $\rightarrow$ UCM. The left-most column is the input image, and the next three consecutive columns represent the Grad-CAM visualization of the image using a model trained with; $\mathbf{A}$, $\mathbf{A}$ and $\mathbf{B}$, and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ respectively. It can be clearly seen that adaptively learning the graph topology $\mathbf{B}$ and $\mathbf{C}$ helps activate the most relevant areas of interest, leading to better classification performance.

Failure cases

In Fig. 4.8, some failure cases are presented using the Grad-CAM visualization. The first column shows the input image with ground truth labels while the subsequent three columns are the activated Grad-CAMs with $\mathbf{A}$, $\mathbf{A}$ and $\mathbf{B}$ and $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$, respectively. It can be noticed that when the targeted object occupies most of the image, the use of $\mathbf{B}$ and $\mathbf{C}$ makes the model confuse the object with the background. A good illustration

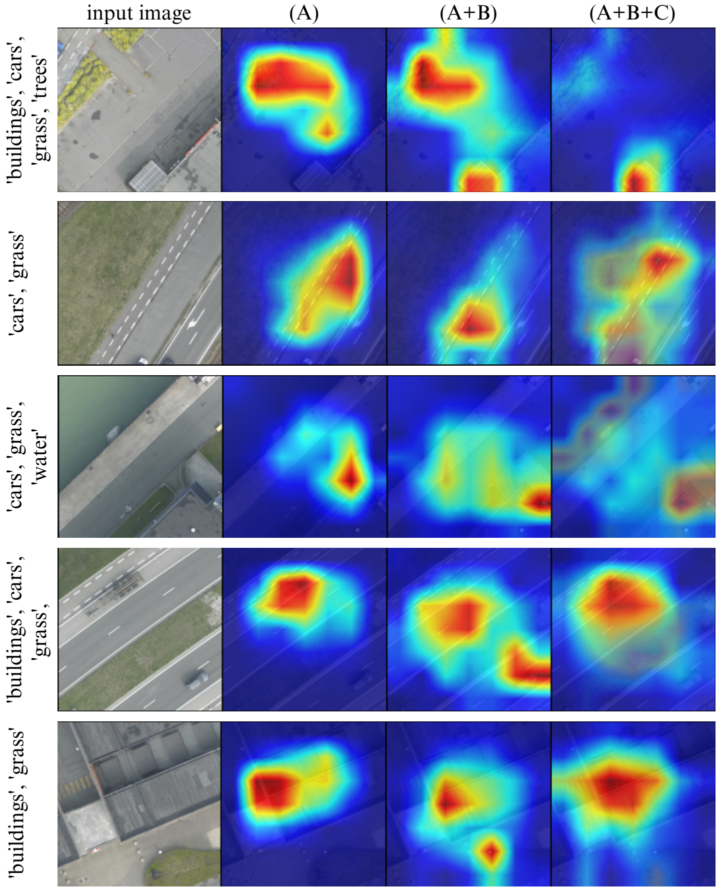


Figure 4.8: Failure cases: Grad-CAM visualization for AID → DFC with ground truth multi-labels showcasing performance degradation with adaptive graphs.

of this can be seen in the second row of Fig. 4.8. The *car* is confused with the *grass*. In future work, modeling the object occupancy will be investigated for handling these failure cases.

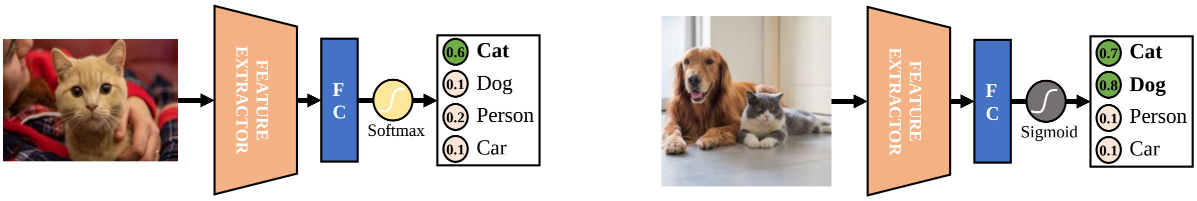
4.6 Conclusion

Existing graph-based methods have shown great performances for multi-label image classification in the context of both single-domain and cross-domain. However, these methods mostly fix the graph topology heuristically while discarding edges with rare co-occurrences. Furthermore, it has been demonstrated in [65] that successive GCN aggregations tend to destroy the initial feature similarity. Hence, as a solution, an adaptive strategy for learning the graph in an end-to-end manner is proposed in this chapter. In particular, attention-based and similarity-preserving mechanisms are adopted. The proposed framework for multi-label classification in a single domain is then extended to multiple domains. For that purpose, an adversarial domain adaptation strategy is employed. The results performed for both single and cross-domain support the effectiveness of the proposed method in terms of model performance and size as compared to recent state-of-the-art methods. The current work is restricted to scenarios where the source and target data have shared object categories. Nevertheless, in future works, we intend to investigate a more challenging problem, namely, open set domain adaptation where only a few categories of labels are shared between source and target data. Additionally, since the current method makes use of an additional discriminator for successful domain adaptation, this might lead to the problem of mode collapse. Hence, to overcome this limitation, in the next chapter, we present an alternative paradigm of discriminator-free UDA for MLIC.

Chapter 5

Discriminator-free Unsupervised Domain Adaptation for Multi-label Image Classification

In the previous chapter, we have discussed some of the recent adversarial-based Unsupervised Domain Adaptation (UDA) methods for Multi-Label Image Classification (MLIC), including the proposed DA-AGCN. In general, these methods incorporate an additional discriminator subnet which poses a significant drawback. Decoupling classification and discrimination tasks in these methods may harm their task-specific discriminative power, hindering the learning of domain-invariant features. In this chapter, we introduce a new paradigm of discriminator-free adversarial-based UDA in MLIC. In this work, termed DDA-MLIC, we overcome the aforementioned limitation by introducing a novel adversarial critic directly derived from the task-specific classifier. Specifically, we employ a two-component Gaussian Mixture Model (GMM) to model both source and target predictions, distinguishing between two distinct clusters. Our approach utilizes a Deep Neural Network (DNN) to estimate the parameters of each GMM component. Subsequently, the source and target GMM parameters are leveraged to formulate an



(a) Single-label image classification

(b) Multi-label image classification

Figure 5.1: The work of [79] cannot be directly applied to MLIC due to the differences between the two tasks [34]: (a) Single-label image classification uses a softmax activation function to convert the predicted logits into probabilities such that the sum of all class probabilities is equal to one; and (b) on the other hand, multi-label image classification uses sigmoid activation where each logit is scaled between 0 and 1, giving higher probability values for the objects present in an image.

adversarial loss using the Fréchet distance. This framework enables effective domain adaptation with end-to-end differentiability for MLIC tasks. The proposed method is evaluated on several multi-label image datasets covering three different types of domain shift. The obtained results demonstrate that DDA-MLIC outperforms existing state-of-the-art methods in terms of precision while requiring a lower number of parameters. The code is publicly available at `github.com/cvi2snt/DDA-MLIC`.

5.1 Introduction

Multi-label Image Classification (MLIC) is an active research topic within the computer vision community, given its relevance in numerous applications such as object recognition [1], scene classification [12], and attribute recognition [37], [90]. Its primary objective is to predict the presence or absence of a predefined set of objects within a given image. Thanks to the recent advancements in deep learning, several MLIC methods [21], [31], [32], [39] have achieved remarkable performance on well-known benchmarks [1], [80]. However, inheriting from the limitations of deep learning, existing MLIC methods are also negatively impacted by the *domain shift* phenomenon. In other words, an MLIC method trained using data from a given domain, usually called *source domain*, will suffer from degraded performance when tested on samples belonging to an unseen domain, referred to as *target domain*. A direct solution is to simply label these target data and use them as additional training samples. Nevertheless, such a process is both resource-intensive and time-consuming.

To handle this issue, *Unsupervised Domain Adaptation (UDA)* methods have been proposed [23], [30], [44], [91] as an alternative. Instead of relying on annotated source data solely, UDA techniques take advantage of unlabelled target samples to minimize the shift between the source and the target domains.

Existing UDA approaches have been primarily focusing on the problem of single-label image classification [30], [44], [74], [75], [91], [92] and semantic segmentation [93]–[96], giving less attention to other computer vision tasks including multi-label classification. Indeed, a limited number of UDA methods [33], [47], [77] has been proposed for the specific case of multi-label image classification. These methods mainly take inspiration from adversarial UDA techniques for single-label image classification to implicitly reduce the domain shift. Similar to [30], these adversarial approaches integrate an additional domain discriminator coupled with a min-max two-player game. This strategy guides the generator to extract domain-invariant features that can fool the discriminator. However, as highlighted in [79], adopting such an adversarial training may cause mode collapse, resulting in a lower task-specific discriminative power.

To handle this issue in the context of single-label classification, Chen et al. [79] proposed to reuse the classifier as a discriminator. More precisely, they introduced an adversarial critic based on the difference between inter-class and intra-class correlations of the classifier probability predictions. However, unlike single-label classification, the per-class prediction probabilities in MLIC are not linearly dependent, thereby are not constrained to sum up to one, as depicted in Fig. 5.1. Thus, a direct extension of the approach proposed in [79] to MLIC is only possible by employing multiple binary classifiers, e.g., one for each class. In this way, the critic proposed in [79] can be used by computing the correlations between the predicted probabilities of each binary classifier. Nevertheless, such an approach remains sub-optimal since the domain alignment is realized for each label classifier independently, disregarding the inter-class correlations. Our experiments in Section 5.6 support this hypothesis.

In this work, we propose a novel discriminator-free adversarial-based UDA method called DDA-MLIC, specifically tailored to MLIC. Motivated by [79], we reuse the task-specific classifier as a discriminator to avoid mode collapse. For that purpose, a novel critic suitable to the task of MLIC has been proposed. In particular, this critic is computed by clustering probability predictions into two sets (one in the neighborhood of 0 and another one in the neighborhood of 1), estimating their respective distributions and quantifying the distance between the estimated distributions from the source and target data. The proposed idea is mainly inspired by the following observation: source samples tend to be classified (as positive or negative) more confidently than target ones, as illustrated in Fig. 5.2. The same figure also shows that the distribution of predictions is formed by two peaks; suggesting the suitability of a bimodal distribution model. Therefore, we argue that the distribution shape of probability predictions can implicitly enable the discrimination between source and target data. Practically, we propose to fit a Gaussian Mixture Model (GMM) with two components on both the source and target predictions. Finally, a Fréchet distance [97] between the estimated pair of components is employed for defining the introduced discrepancy measure. However, the use of the standard Expectation-Maximisation (EM) algorithm for estimating GMM introduces two main limitations, namely:

(1) **The non-differentiability**: the EM step is not differentiable as it breaks the chain rule. Hence, the EM does not contribute to the gradient-based optimization.

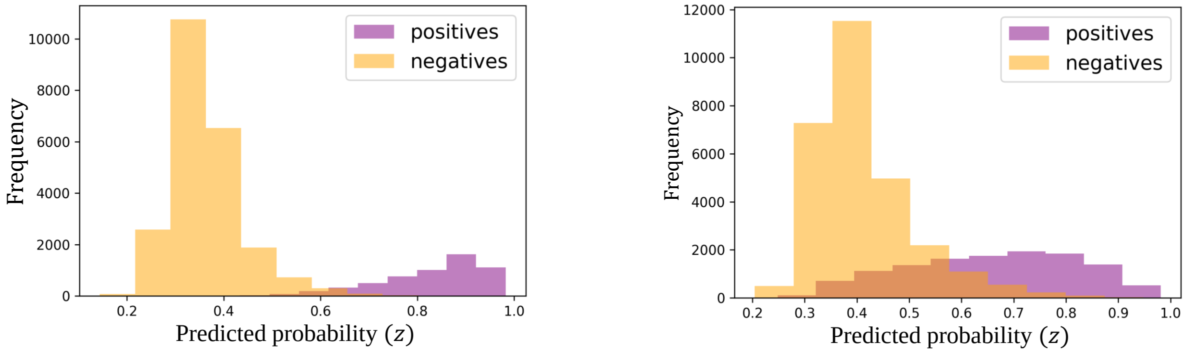
(2) **The demanding computational cost**: the EM algorithm is resource-intensive since it is based on iterative optimization.

Hence, in [35], we further extend the proposed DDA-MLIC method to handle the two aforementioned limitations. Instead of relying on a non-differentiable and iterative traditional EM algorithm, the proposed method utilizes a neural block that mimics the EM optimization process. This block called *DeepEM* is used for computing the GMM parameters based on a closed-form solution while ensuring the backpropagation of the related gradients through the entire network. As a result, only a single iteration is needed. The experimental results show that the proposed approach outperforms state-of-the-art methods in terms of mean Average Precision (mAP) while significantly reducing the average training time per batch and the number of network parameters.

In summary, our main contributions are the following:

- A novel domain discrepancy for multi-label image classification based on the distribution of the task-specific classifier predictions;
- An effective and efficient adversarial unsupervised domain adaptation method for multi-label image classification. The proposed adversarial strategy does not require an additional discriminator, hence reducing the network size during training;
- A differentiable and non-iterative GMM parameter estimation strategy leads to better precision and faster training times.
- A comprehensive experimental analysis, demonstrating the superiority of the proposed method over state-of-the-art techniques in terms of mean Average Precision (mAP) and training time.

The rest of the paper is organized as follows. Section 5.2 discusses existing works on standard MLIC and UDA for multi-class and multi-label classification. Section 5.3 introduces some mathematical preliminaries related to the EM algorithm for GMM estimation. Section 5.4 formulates the problem of UDA in MLIC and details our motivation for reusing the classifier as a discriminator. Section 5.5 introduces the discriminator-free



(a) Source → Source

(b) Source → Target

Figure 5.2: Histogram of classifier predictions[2]. Predicted probabilities using source-only trained classifier[2] on: (a) source dataset[3] $(\mathcal{I}_s)$, and (b) target dataset[3] $(\mathcal{I}_t)$.

approach DDA-MLIC and later details the proposed DeepEM block proposed as a replacement of the standard EM. The experimental analysis and discussion are detailed in Section 5.6. Section 5.7 discusses the limitations of the proposed approach. Finally, Section 5.8 concludes this work and draws some interesting perspectives.

5.2 Related works

In this section, we start by reviewing the state-of-the-art on standard Multi-label Image Classification (MLIC). Then, we discuss related works in the general field of UDA. Lastly, we present the limited literature devoted to the topic of UDA for MLIC.

5.2.1 Multi-label Image Classification (MLIC)

In the literature, recent works in MLIC have benefited from the widespread availability of large-scale multi-label image datasets [1], [80], [87] and the proven success of deep Convolutional Neural Networks (CNN) [9], [60]. For example, Hypotheses-CNN

[2] TResNet-M [24] trained on UCM [85] dataset.
[3] Source: UCM [85] validation set (420 images), Target: AID [84] validation set (600 images).

Pooling (HCP) [68] has leveraged the predictions of multiple CNN architectures, such as AlexNet [71] and VGG-16 [60], pre-trained on ImageNet [70]. Recently, in [21] authors have proposed Asymmetric Loss (ASL) that focuses more on the items present in the image (positive labels) than the ones that are absent (negative labels). ASL coupled with a recently introduced CNN architecture, named TResNet [24], has shown impressive performance for MLIC. Alternatively, ML-GCN [39] employs a Graph Convolutional Network (GCN) to model the label correlations. Based on a similar strategy, ML-AGCN [32] proposed to adaptively learn the label graph topology instead of heuristically defining it.

Nevertheless, the performance of these methods is conditioned by the availability of large-scale annotated datasets. Notably, a significant drop in performance can be observed when applied to unseen domains [47].

5.2.2 Unsupervised Domain Adaptation (UDA) for Single-label Image Classification

UDA techniques have been proposed for enhancing the robustness of deep learning frameworks while avoiding costly labeling interventions. Most of UDA efforts have been dedicated to the task of multi-class classification. In particular, they have mainly followed two paradigms to reduce the disparity between source and target domains. The first paradigm explicitly aims to minimize the distance between the statistical moments of source and target distributions [44], [73], [74], [91]. The second one makes use of an adversarial learning strategy [30] in order to implicitly reduce the domain shift [75], [92]. In general, these methods employ an additional domain discriminator that is meant to distinguish between source and target data. To generate domain-invariant features, an adversarial training strategy is adopted, where the goal is to fool the discriminator while maintaining an acceptable discriminative power.

Despite their effectiveness, existing discriminator-based adversarial approaches may suffer from the problem of mode collapse, which typically occurs under adversarial training. In order to handle this challenge, a discriminator-free adversarial approach has been recently proposed in [34]. Specifically, an adversarial critic based on the difference between inter-class and intra-class correlations of the probability predictions of the classifier is proposed. However, the predicted probabilities in Multi-Label Image Classification (MLIC) do not necessarily sum up to one, in contrast to single-label classification, as illustrated in Fig. 5.1. Hence, as mentioned earlier, this approach can only be extended to MLIC by reformulating the problem as multiple binary classifications that might lead to sub-optimal results.

5.2.3 Unsupervised Domain Adaptation (UDA) for Multi-label Image Classification (MLIC)

As highlighted earlier, only few UDA methods have been proposed for the specific case of MLIC [33], [47], [76]. ML-ANet [76] follows a moment-matching strategy as they propose the use of Multi-Kernel Maximum Mean Discrepancies (MK-MMD) in a Reproducing Kernel Hilbert Space (RKHS). More recently, motivated by the progress made in adversarial-based UDA, attempts to generalize discriminator-based UDA methods to MLIC have emerged. Specifically, DA-MAIC [47] adopts a graph-based MLIC framework and couples it with a domain discriminator trained adversarially. Similarly, DA-AGCN [33] also follows a standard discriminator-based strategy, but injects an additional attention mechanism in the graph-based MLIC subnetwork.

However, these discriminator-based methods are equally threatened by the mode collapse issue discussed in [79]. Therefore, in this work, we propose a novel adversarial critic extracted from the task-specific classifier itself, thereby eliminating the use of an additional discriminator.

5.3 Mathematical Preliminaries

For a deeper understanding of the paper, this section briefly recalls the Expectation Maximization (EM) algorithm for the estimation of Gaussian Mixture Model (GMM) parameters.

The GMM is a mixture model that is formed by K Gaussian components, as depicted in the following equation,

$$P(\mathbf{x}|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k), \tag{5.1}$$

where $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ is an n-dimensional continuous-valued data vector (i.e. observations or features). The tuple $\Theta_k = \{\pi_k, \mu_k, \Sigma_k\}$, is formed by π_k, μ_k and Σ_k which denote the mixture weights, the mean vector and the covariance matrix of the k^{th} GMM component, respectively, with $\sum_{k=1}^{K} \pi_k = 1$.

The Maximum Likelihood Estimation (MLE) is a common approach for estimating the mixture parameters $\Theta = \{\Theta_k\}_{k \in \{\ 1,2,...,K\}}$ by maximizing the log-likelihood of the observations, given by,

$$l(\Theta|\mathbf{x}) = \log\ P(\mathbf{x}|\Theta). \tag{5.2}$$

Since directly maximizing $l(\Theta|\mathbf{x})$ is intractable, the EM algorithm maximizes instead a lower bound of $l(\Theta|\mathbf{x})$ defined as,

$$Q = \log \left\{ \sum_{\mathcal{C}} P(\mathcal{C}|\mathbf{x}, \Theta) \right\}. \tag{5.3}$$

where $\mathcal{C}$ refers to a set of discrete latent variables.

An iterative optimization of this bound alternates between two steps: in the E-step, a new estimate of the posterior probability distribution over $\mathcal{C}$ is computed, given the estimation of the parameters Θ denoted as Θ^m from the previous iteration m. In the context of GMM, the set $\mathcal{C}$ is defined as a set of binary latent variables $\mathcal{C} =$

$(c_{ik})_{i\in\{1,\ldots,n\},k\in\{1,\ldots,K\}}$ for simplifying the optimization process. This helps in calculating the responsibility of each component k in the mixture as follows,

$$\gamma_{ik}(\Theta^m) = Q(c_{ik} = 1|\Theta_k^m, x_i) \tag{5.4}$$

In the M-step, the parameters Θ are updated in order to maximize the expected log-likelihood using the posteriors computed in the E-step such that,

$$\Theta^{m+1} = \underset{\Theta^m}{\operatorname{argmax}}\, Q(\Theta^{m+1}|\Theta^m). \tag{5.5}$$

Thanks to the introduction of the binary latent variables $\mathcal{C}$, the computation of the updated parameters in Eq. (5.5) can be done using a closed-form solution as detailed below,

$$\begin{aligned} \pi_k^{m+1} &= \frac{1}{n}\sum_{i=1}^{n}\gamma_{ik}, \quad \mu_k^{m+1} = \frac{\sum_{i=1}^{n}\gamma_{ik}x_i}{\sum_{i=1}^{n}\gamma_{ik}}, \\ \Sigma_k^{m+1} &= \frac{\sum_{i=1}^{n}\gamma_{ik}(x_i-\mu_k^m)(x_i-\mu_k^m)^T}{\sum_{i=1}^{n}\gamma_{ik}}. \end{aligned} \tag{5.6}$$

The algorithm refines the values of the estimated parameters iteratively until convergence. Hence, this optimization process remains computationally costly.

5.4 Problem Formulation and Motivation

In this section, we first formulate the problem of Unsupervised Domain Adaptation (UDA) for Multi-Label Image Classification (MLIC). Later, we detail our motivation behind reusing the multi-label classifier as a discriminator.

5.4.1 Problem Formulation

Let $\mathcal{D}_s = (\mathcal{I}_s, \mathcal{Y}_s)$ and $\mathcal{D}_t = (\mathcal{I}_t, \mathcal{Y}_t)$ be the source and target datasets, respectively, with P_s and P_t being their respective probability distributions such that $P_s \neq P_t$. Let us assume that they are both composed of N object category labels. Note that $\mathcal{I}_s = \{\mathbf{I}_s^j\}_{j=1}^{n_s}$ and $\mathcal{I}_t = \{\mathbf{I}_t^j\}_{j=1}^{n_t}$ refer to the sets of n_s source and n_t target image samples, respectively, while $\mathcal{Y}_s = \{\mathbf{y}_s^j\}_{j=1}^{n_s}$ and $\mathcal{Y}_t = \{\mathbf{y}_t^j\}_{j=1}^{n_t}$ are their associated sets of labels.

Let us denote by $\mathcal{I}$ the set of all images such that $\mathcal{I} = \mathcal{I}_s \cup \mathcal{I}_t$. Given an input image $\mathbf{I} \in \mathcal{I}$ with $\mathbf{y} \in \{0,1\}^N$ being its label, the goal of *unsupervised domain adaptation* for *multi-label image classification* is to estimate a function $f : \mathcal{I} \mapsto \{0,1\}^N$ such that,

$$f(\mathbf{I}) = \mathbb{1}_{f_c \circ f_g(\mathbf{I}) > \tau} = \mathbb{1}_{\mathbf{Z} > \tau} = \mathbf{y} \ , \tag{5.7}$$

where $f_g : \mathcal{I} \mapsto \mathbb{R}^d$ extracts d-dimensional features, $f_c : \mathbb{R}^d \mapsto [0,1]^N$ predicts the probability of object presence, $\mathbf{Z} = f_c \circ f_g(\mathbf{I}) \in [0,1]^N$ corresponds to the predicted probabilities, $\mathbb{1}$ is an indicator function, $>$ is a comparative element-wise operator with respect to a chosen threshold τ. Note that only $\mathcal{D}_s$ and $\mathcal{I}_t$ are used for training. In other words, the target dataset is assumed to be unlabeled.

To achieve this goal, some existing methods [47] have adopted an adversarial strategy by considering an additional discriminator f_d that differentiates between source and target data. Hence, the model is optimized using a classifier loss $\mathcal{L}_{cls}$ such as the asymmetric loss (ASL) [21] and an adversarial loss $\mathcal{L}_{adv}$ defined as,

$$\begin{aligned} \mathcal{L}_{adv} = \mathbb{E}_{f_g(\mathbf{I}_s) \sim \bar{P}_s} \log \frac{1}{f_d(f_g(\mathbf{I}_s))} + \\ \mathbb{E}_{f_g(\mathbf{I}_t) \sim \bar{P}_t} \log \frac{1}{(1 - f_d(f_g(\mathbf{I}_t))} \ , \end{aligned} \tag{5.8}$$

where $\bar{P}_s$ and $\bar{P}_t$ are the distributions of the learned features from source and target samples $\mathcal{I}_s$ and $\mathcal{I}_t$, respectively.

While the adversarial paradigm has shown great potential [47], the use of an additional discriminator f_d which is decoupled from f_c may lead to mode collapse as discussed in [79]. Inspired by the same work, we aim at addressing the following question – *Could we leverage the outputs of the task-specific classifier $f_c \circ f_g$ in the context of multi-label classification for implicitly discriminating the source and the target domains?*

5.4.2 Motivation: Domain Discrimination using the Distribution of the Classifier Output

The goal of MLIC is to identify the classes that are present in an image (*i.e.*, *positive labels*) and reject the ones that are absent (*i.e.*, *negative labels*). Hence, the classifier f_c is expected to output high probability values for the positive labels and low probability values for the negative ones. Formally, let $z = \theta(f_c(f_g(\mathbf{I}))) = \theta(\mathbf{Z}) \sim \hat{P}$ be the random variable modelling the predicted probability of any class and $\hat{P}$ its probability distribution, with θ being a uniform sampling function that returns the predicted probability of a randomly selected class. In general, a well-performing classifier is expected to classify confidently both negative and positive samples. Ideally, this would mean that the probability distribution $\hat{P}$ should be formed by two clusters with low variance in the neighborhood of 0 and 1, respectively denoted by $\mathcal{C}_0$ and $\mathcal{C}_1$. Hence, our hypothesis is that a drop in the classifier performance due to a domain shift can be reflected in $\hat{P}$.

Let $z_s = \theta(f_c(f_g(\mathbf{I}_s))) \sim \hat{P}_s$ and $z_t = \theta(f_c(f_g(\mathbf{I}_t))) \sim \hat{P}_t$ be the random variables modelling the predicted probability obtained from the source and target data and $\hat{P}_s$ and $\hat{P}_t$ be their distributions, respectively. Concretely, we propose to investigate whether the shift between the source and target domains is translated in $\hat{P}_s$ and $\hat{P}_t$. If a clear difference is observed between $\hat{P}_s$ and $\hat{P}_t$, this would mean that the classifier f_c should be able to discriminate between source and target samples. Thus, this would allow the definition of a suitable critic directly from the classifier predictions.

To support our claim, we trained a model[4] f using the labelled source data $\mathcal{D}_s$ without involving the target images[5] $\mathcal{I}_t$. In Fig.5.2 (a), we visualize the histogram of the classifier probability outputs when the model is tested on the source domain. It can be clearly observed that the predicted probabilities on the source domain, denoted by $\mathbf{z}_s$, can be grouped into two separate clusters. Fig.5.2 (b) shows the same histogram when the model is tested on target samples. In contrast to the source domain, the classifier probability outputs, denoted by $\mathbf{z}_t$, are more spread out in the target domain. In particular, the two clusters are less separable than in the source domain. This is due to the fact that the classifier f_c benefited from the supervised training on the source domain, and as a result it gained an implicit discriminative ability between the source and target domains.

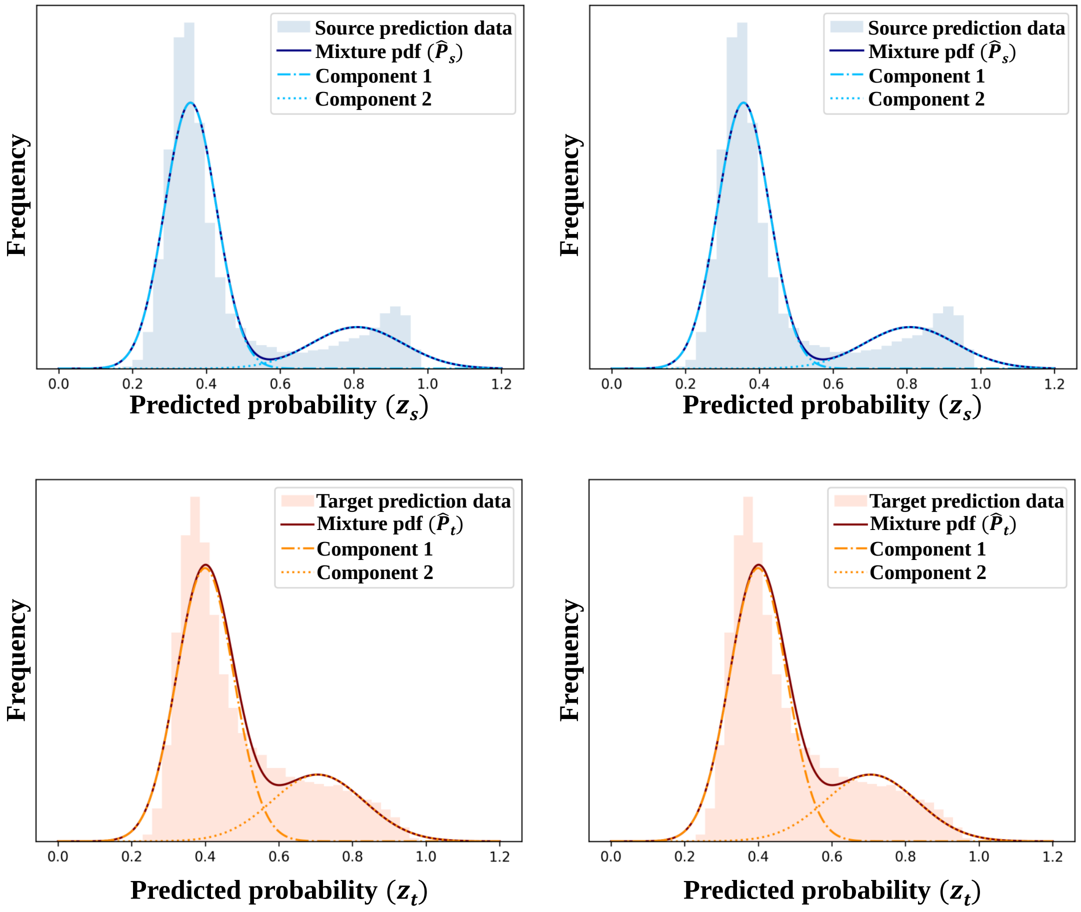
Motivated by the observations discussed above, we propose to reuse the classifier to define a critic function based on $\hat{P}_s$ and $\hat{P}_t$. In what follows, we describe our approach including the probability distribution modelling ($\hat{P}_s$ and $\hat{P}_t$) and the adversarial strategy for domain adaptation.

5.5 Proposed Approach

In this section, we first detail the DDA-MLIC approach, including the novel adversarial critic derived from the task classifier using the standard EM algorithm. Later, we introduce the proposed DeepEM block that overcomes the limitations of the traditional EM, namely, non-differentiability with respect to the overall architecture and a high computational cost. Finally, we provide an overview of the proposed architecture incorporating an end-to-end learning process for multi-label prediction and unsupervised

[4] TResNet-M [24] trained on UCM [85] dataset.

[5] Source: UCM [85] validation set (420 images), Target: AID [84] validation set (600 images).



(a) GMM fitting.

(b) Gaussians of components.

Figure 5.3: (a) The classifier[6] predictions $\mathbf{z}_s$ and $\mathbf{z}_t$ for both source and target datasets[7], respectively, can be grouped into two clusters. Hence, a two-component GMM can be fitted for both source ($\hat{P}_s$) and target ($\hat{P}_t$). While the first component is close to 0, the second is close to 1, (b) A component-wise comparison between source ($\hat{P}^1_s$, $\hat{P}^2_s$) and target ($\hat{P}^1_t$, $\hat{P}^2_t$) Gaussians of distributions extracted from the fitted GMM confirms that target predictions are likely to be farther from 0 and 1 with a higher standard deviation than the source.

discriminator-free domain adaptation.

[6] TResNet-M [24] trained on UCM [85] dataset.

[7] Source: UCM [85] validation set (420 images), Target: AID [84] validation set (600 images).

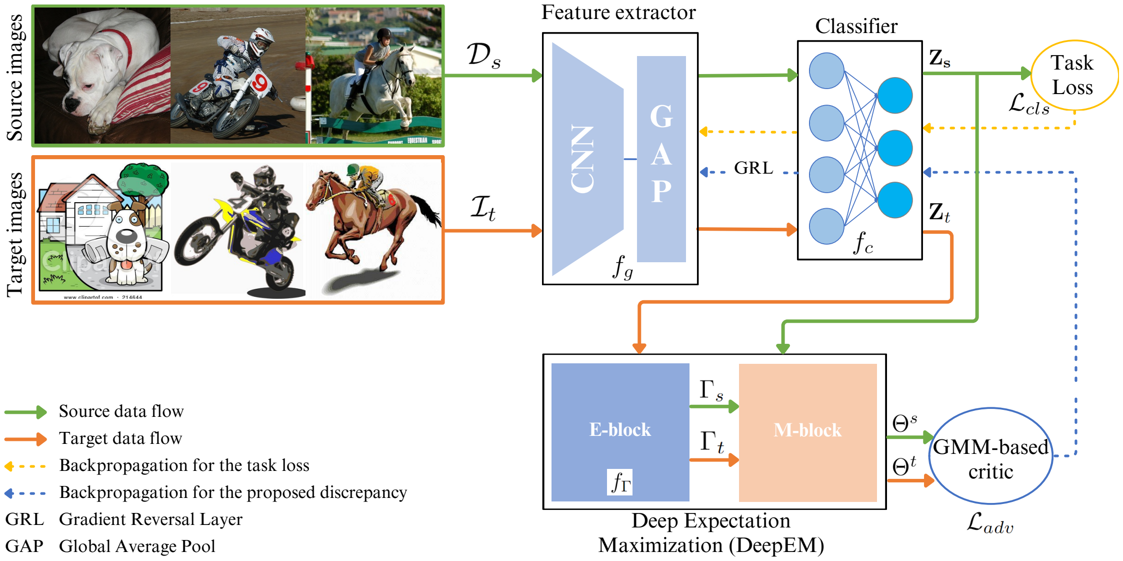


Figure 5.4: The overall architecture of DDA-MLIC with the proposed DeepEM block consists of the following components: The feature extractor (f_g) learns discriminative features from both source and target images. The task classifier (f_c) performs two actions simultaneously: 1) it learns to accurately classify source samples using a supervised task loss $\mathcal{L}_{cls}(\mathcal{D}_s)$, and 2) when acting as a discriminator, it aims to minimize the proposed GMM-based discrepancy $\mathcal{L}_{adv}(\mathcal{D}_s, \mathcal{I}_t)$ between source $(\mathbf{z}_s)$ and target $(\mathbf{z}_t)$ predictions using the proposed DeepEM block, while f_g simultaneously works to maximize it.

5.5.1 DDA-MLIC: an Implicit Adversarial Critic from the Task Classifier

As discussed in Section 5.4.2, the classifier probability predictions are usually formed by two clusters with nearly Gaussian distributions. Consequently, as shown in Fig.5.3 (a), we suggest approximating the two distributions $\hat{P}_s$ and $\hat{P}_t$ by a two-component Gaussian Mixture Model (GMM) defined in Eq. (5.1) as follows,

$$\hat{P}_s(\mathbf{z}_s) \approx \sum_{k=1}^{2} \pi_k^s \mathcal{N}(\mathbf{z}_s|\mu_k^s, \sigma_k^s) \ , \tag{5.9}$$

and,

$$\hat{P}_t(\mathbf{z}_t) \approx \sum_{k=1}^{2} \pi_k^t \mathcal{N}(\mathbf{z}_t|\mu_k^t, \sigma_k^t) \ , \tag{5.10}$$

where $\mathcal{N}(\mathbf{z}_t|\mu_k^t, \sigma_k^t)$ denotes the k-th Gaussian distribution, with the mean μ_k^t and the standard deviation σ_k^t, fitted on the target predicted probabilities $\mathbf{z}_t$ and π_k^t its mixture weight such that $\pi_1^t + \pi_2^t = 1$. Similarly, $\mathcal{N}(\mathbf{z}_s|\mu_k^s, \sigma_k^s)$ denotes the k-th Gaussian distribution, with the mean μ_k^s and the standard deviation σ_k^s, fitted on the source predicted probabilities $\mathbf{z}_s$ and π_k^s its mixture weight such that $\pi_1^s + \pi_2^s = 1$. For estimating the two GMM models, the EM algorithm presented in Section 5.3 used.

In both the source and target domains, we posit that the initial component of the Gaussian Mixture Model (GMM) aligns with the cluster $\mathcal{C}_0$ (featuring a mean in proximity to 0), while the second component corresponds to $\mathcal{C}_1$ (with a mean in proximity to 1). However, due to a large number of negative predictions as compared to positive ones, the component $\mathcal{C}_0$ tends to be more dominant. In fact, in a given image, only few objects are usually present from the total number of classes. To alleviate this phenomenon, in DDA-MLIC, we proposed to extract two Gaussian components from the source and target GMM, ignoring the estimated weights illustrated in Fig.5.3 (b).

In order to reuse the classifier as a discriminator in the context of UDA for MLIC, in DDA-MLIC [34], we proposed to redefine the adversarial loss $(\mathcal{L}_{adv})$ by computing a Fréchet distance (d_{F}) [97] between each pair of the estimated source and target components, using Eq. (5.5) and (5.6), from a given cluster as follows,

$$\mathcal{L}_{adv} = \sum_{k=1}^{2} \alpha_k d_{\mathrm{F}}(\mathcal{N}(\mathbf{z}_t|\mu_k^t, \sigma_k^t), \mathcal{N}(\mathbf{z}_s|\mu_k^s, \sigma_k^s)) \ , \tag{5.11}$$

with α_k weights that are empirically fixed. Since the computed distributions are univariate Gaussians, the Fréchet distance between two distributions, also called the 2-Wasserstein (2W) distance, is chosen as it can be explicitly computed as follows,

$$d^2_{\mathrm{F}}(\mathcal{N}(\mathbf{z}_1|\mu_1,\sigma_1),\mathcal{N}(\mathbf{z}_2|\mu_2,\sigma_2)) = (\mu_1-\mu_2)^2+(\sigma_1-\sigma_2)^2, \tag{5.12}$$

where $\mathcal{N}(\mathbf{z}_1|\mu_1,\sigma_1)$ and $\mathcal{N}(\mathbf{z}_2|\mu_2,\sigma_2)$ are two Gaussians with a mean of μ_1 and μ_2 and a standard deviation of σ_1 and σ_2, respectively. In addition, compared to the commonly used 1-Wasserstein (1W) distance, it considers second-order moments. Finally, in [98], the 2W distance has been demonstrated to have nicer properties e.g., continuity and differentiability, for optimizing neural networks as compared to other divergences and distances between two distributions such as the Kullback-Leibler (KL) divergence and the Jensen-Shannon (JS) divergence. The relevance of the 2W distance is further discussed in Section 5.6.4.

5.5.2 Deep Expectation Maximization (DeepEM)

As introduced in Section 5.3, due to its iterative nature, the Expectation-Maximization (EM) algorithm is computationally demanding. Furthermore, the GMM fitting step based on EM is non-differentiable with respect to the DDA-MLIC architecture. As a result, it does not impact the backward propagation, posing a challenge to the overall learning process. Alternatively, we propose to use additional block that termed Deep Expectation Maximization (DeepEM), inspired by [99]. The proposed DeepEM consists of two blocks: 1) a Multi-layer Perceptron (MLP) network called *E-block*, denoted as f_Γ, and 2) a parameter-free computational block, called *M-block*.

E-block

Let $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_B\}$ represents the predicted probabilities for any given batch formed by B images, where $\mathbf{z}_i \in \mathbb{R}^N$ is the predicted probabilities vector for sample i. Given these predicted probabilities as input, the E-block outputs the responsibility matrix denoted as $\Gamma \in \mathbb{R}^{BN\times 2}$ such that,

$$\begin{aligned}\Gamma &= f_\Gamma(\mathbf{Z}) \\ &= [\hat{\gamma}_{ik}]_{i\in\{1,BN\},k\in\{1,2\}},\end{aligned} \tag{5.13}$$

where $\hat{\gamma}_{ik}$ is the responsibility estimated by the network, thereby replacing the standard responsibility in Eq. (5.4). In this case, n and K defined in Section 5.3 corresponds respectively to BN and 2.

In summary, the E-block replaces the E-step of the standard EM algorithm. By employing this DNN-based strategy, we overcome two challenges: 1) the chain rule is not broken allowing the differentiability of the E-step, 2) the iterative and resource-intensive computation of the responsibilities is not required. Instead, this can be achieved in a single iteration.

M-block

Let us denote the source and target responsibilities by Γ_s and Γ_t, respectively resulting from the E-block (f_Γ). Given the predicted probabilities $\mathbf{z}_s$ and $\mathbf{z}_t$ for the source and target samples, respectively, the M-block computes the GMM parameters $\Theta_k^s = (\pi_k^s, \mu_k^s, \sigma_k^s)$ and $\Theta_k^t = (\pi_k^t, \mu_k^t, \sigma_k^t)$ for $k \in \{1, 2\}$ using the closed-form solution depicted in Eq. (5.6). The proposed adversarial critic defined in Eq. (5.11) can be directly applied as follows,

$$\mathcal{L}_{adv} = \sum_{k=1}^{2} \alpha_k \left((\mu_k^s - \mu_k^t)^2 + (\sigma_k^s - \sigma_k^t)^2\right), \tag{5.14}$$

where α_k is a hyperparameter that regulates the contributions from each GMM component. The variation of this hyperparameter α is discussed in Section 5.6.4.

5.5.3 Overall Architecture

The overall architecture of the proposed approach is illustrated in Fig. 5.4. Similar to [79], the proposed DDA-MLIC consists of: (1) a feature extractor (f_g) that aims to extract discriminative image features from both source $(\mathcal{I}_s)$ and target $(\mathcal{I}_t)$ images, and (2) a classifier (f_c) that performs the multi-label classification and at the same time implicitly discriminates between source and target data.

When acting as a classifier, f_c aims to minimize the supervised classification loss [21] $(\mathcal{L}_{cls})$ using the annotated source dataset $\mathcal{D}_s$. However, when operating as a discriminator, the output of f_c is is fed to the proposed DeepEM block. It returns the estimated source and target GMMs, i.e., Θ^s and Θ^t, respectively, thereby enabling the computation of the proposed adversarial loss $\mathcal{L}_{adv}$ as reformulated in Eq. (5.14). A Gradient Reversal Layer (GRL) between f_g and f_c enforces the feature extractor to fool the classifier when acting as a discriminator, thereby implicitly learning domain-invariant features. Consequently, the network is trained by engaging in a min-max game as depicted below,

$$\min_{f_c} \max_{f_g} \mathcal{L}_{adv}. \tag{5.15}$$

In summary, the overall loss function used to train DDA-MLIC is described below,

$$\min_{f_g, f_c} \left\{ \mathcal{L}_{cls}(\mathcal{D}_s) + \lambda \max_{f_g} \mathcal{L}_{adv}(\mathcal{D}_s, \mathcal{I}_t) \right\}, \tag{5.16}$$

where λ is another hyper-parameter that weights $\mathcal{L}_{cls}$ and $\mathcal{L}_{adv}$.

5.6 Experiments

In this section, we report the performed experiments and discuss the obtained results. First, we present the datasets used for our experimental study. Later, we detail the experimental settings as well as the implementation details. Finally, we report and analyze the obtained results.

5.6.1 Experimental Settings

Datasets

In our experiments, different types of domain gaps are considered, namely, (1) the domain shift due to the use of different sensors, (2) the domain gap existing between simulated and real data (3) the discrepancy resulting from different weather conditions. Due to the limited availability of cross-domain multi-label datasets, we convert several object detection and semantic segmentation datasets to the task of MLIC.

Cross-sensor domain shift Similar to [47], we use three multi-label aerial image datasets that have been captured using different sensors resulting in different resolutions, pixel densities and altitudes, namely: 1) **AID** [84] multi-label dataset was created from the original multi-class AID dataset [88] by labeling 3000 aerial images, including 2400 for training and 600 for testing, with a total of 17 categories. 2) **UCM** [85] multi-label dataset was recreated from the original multi-class classification dataset [89] with a total of 2100 image samples containing the same 17 object labels as AID. We randomly split the dataset into training and testing sets with 2674 and 668 image samples, respectively. 3) **DFC** [86] multi-label dataset provides 3342 high-resolution images with training and testing splits of, respectively, 2674 and 668 samples labeled from a total of 8 categories. In our experiments, the 6 common categories between DFC and the other two benchmarks are used.

Sim2real domain shift We use the following two datasets to investigate the domain gap between real and synthetic scene understanding images. 1) **PASCAL-VOC**[80] is one of the most widely used real image datasets for MLIC with more than $10K$ image samples. It covers 20 object categories. The training and testing sets contain 5011 and 4952 image samples, respectively. 2) **Clipart1k** [87] provides 1000 synthetic clipart image samples, annotated with 20 object labels, similar to VOC. Since it is proposed for the task of object detection, we make use of the category labels for bounding boxes to create a multi-label version. Half of the data are used for training and the rest is used for testing.

Cross-weather domain shift In order to study the domain shift caused by different weather conditions, two widely used urban street datasets have been used, namely: 1) **Cityscapes** [100] which is introduced for the task of semantic image segmentation and consists of 5000 real images captured in the daytime. 2) **Foggy-cityscapes** [101] which is a synthesized version of Cityscapes where an artificial fog is introduced. We generate a multi-label version of these datasets for the task of MLIC considering only 11 categories out of the original 19 to avoid including the objects that appear in all the images.

Implementation Details

The proposed work makes use of TResNet-M [24] as a backbone and the Asymmetric Loss (ASL) [21] as the task loss. All the methods are trained using the Adam optimizer with a cosine decayed maximum learning rate of 10^{-3}. For all the experiments, we make use of NVIDIA TITAN V with a batch size of 64 for a total of 25 epochs or until convergence. The input image resolution has been fixed to 224×224.

Baselines

To evaluate our approach, we categorized methods into three groups. *MLIC* methods are trained solely on source datasets and evaluated directly on target datasets without any domain adaptation strategy, including both direct and indirect approaches. *Disc.-based* and *Disc.-free* methods utilize both labeled source and unlabeled target datasets, with and without the domain discriminator, respectively, and are evaluated on the target dataset. We adapt our baseline discriminator-free approach DALN [79], originally designed for UDA in single-label classification, to MLIC by computing the adversarial critic for multiple binary predictions.

Evaluation Metrics

Similar to [34], we report several metrics including the number of model parameters (# params), mean Average Precision (mAP), average per-Class Precision (CP), average per-Class Recall (CR), average per-Class F1-score (CF1), average Overall Precision (OP), average overall recall (OR) and average Overall F1-score (OF1). We consider seven datasets: AID, UCM, DFC, VOC, Clipart, Cityscapes, and Foggycityscapes, resulting in seven experimental settings: AID → UCM, UCM → AID, AID → DFC, UCM → DFC, VOC → Clipart, Clipart → VOC, and Cityscapes → Foggy. For example, AID → UCM indicates that AID is fixed as the source dataset during training while UCM is the target dataset. The reported results are based on the testing set of the target dataset.

5.6.2 Quantitative Analysis

Comparison with the state-of-the-art methods

Table 5.1, Table 5.2, Table 5.3 and Table 5.4 quantitatively compare the proposed approach to state-of-the-art methods. It can be seen that our model requires an equal or fewer number of parameters than other state-of-the-art works, with a total number of 29.4 million parameters. We achieve the best performance in terms of mAP for AID → UCM, UCM → AID, AID → DFC, UCM → DFC, Clipart → VOC and Cityscapes → Foggy.

Table 5.1: Cross-sensor domain shift: comparison with the state-of-the-art in terms of number of model parameters (in millions), and % scores of mAP, per-class averages (CP, CR, CF1) and overall averages (OP, OR, OF1) for aerial image datasets. Two settings are considered, *i.e.*, AID → UCM and UCM → AID. Best results are highlighted in **bold**.

Type	**Method**	**# params**	**AID → UCM**							**UCM → AID**						
			mAP	**P_C**	**R_C**	**F_C**	**P_O**	**R_O**	**F_O**	**mAP**	**P_C**	**R_C**	**F_C**	**P_O**	**R_O**	**F_O**
MLIC	ResNet101 [9]	42.5	57.5	**60.0**	47.5	47.0	69.1	71.5	**70.3**	51.7	50.6	29.6	33.9	88.0	48.5	62.5
	ML-GCN [39]	44.9	53.7	55.3	44.3	45.9	70.2	68.7	69.4	51.3	50.1	29.9	34.0	88.0	49.7	63.6
	ML-AGCN [32]	36.6	55.2	36.6	**64.9**	45.1	45.0	**88.1**	59.6	52.1	48.2	**47.4**	**42.9**	77.1	**79.8**	**78.4**
	ASL (TResNetM) [21]	29.4	55.4	48.7	52.8	47.1	58.7	79.1	67.4	54.1	54.5	40.2	41.9	85.4	65.1	73.9
Disc-based	DANN (TResNetM + ASL) [30]	29.4	52.5	59.1	31.6	36.3	**70.9**	53.7	61.1	51.6	52.1	23.2	27.9	83.2	27.8	41.7
	DA-MAIC (TResNetM+ASL) [47]	36.6	54.4	55.3	37.5	38.6	68.0	67.9	67.9	50.5	51.8	22.9	29.0	91.6	35.2	50.8
Disc-free	DALN (TResNetm + ASL) [79]	29.4	53.1	53.3	32.4	36.7	69.2	53.9	60.6	53.2	52.2	29.3	32.7	82.0	41.2	54.8
	DDA-MLIC (OURS)	29.4	**63.2**	52.5	63.7	**55.1**	59.4	82.8	69.2	54.9	53.9	30.4	35.5	84.6	41.0	55.3
	DDA-MLIC with DeepEM (OURS)	29.4	60.5	53.5	51.8	48.8	61.4	76.8	68.2	**56.2**	**58.0**	19.0	26.5	**97.4**	31.5	47.6

Table 5.2: Cross-sensor domain shift: comparison with the state-of-the-art in terms of number of model parameters (in millions), and % scores of mAP, per-class averages (CP, CR, CF1) and overall averages (OP, OR, OF1) for aerial image datasets. Two settings are considered, *i.e.*, AID → DFC and UCM → DFC. Best results are highlighted in **bold**.

Type	**Method**	**# params**	**AID → DFC**							**UCM → DFC**						
			mAP	**P_C**	**R_C**	**F_C**	**P_O**	**R_O**	**F_O**	**mAP**	**P_C**	**R_C**	**F_C**	**P_O**	**R_O**	**F_O**
MLIC	ResNet101 [9]	42.5	56.9	52.9	61.5	48.7	46.1	63.7	53.5	66.4	74.4	31.2	36.9	67.2	37.2	47.9
	ML-GCN [39]	44.9	58.9	**56.7**	57.9	45.8	45.7	65.0	53.7	64.6	72.4	32.0	35.6	64.4	38.9	48.5
	ML-AGCN [32]	36.6	51.6	41.5	**83.8**	52.3	40.2	**88.7**	55.3	70.3	68.4	**56.1**	47.8	53.8	**58.5**	56.0
	ASL (TResNetM) [21]	29.4	56.1	49.6	68.4	49.9	43.5	74.1	54.8	68.9	66.3	53.1	44.0	52.6	57.0	54.7
Disc-based	DANN (TResNetM + ASL) [30]	29.4	43.0	40.7	13.6	19.3	46.0	15.6	23.3	64.1	77.3	22.6	30.1	68.6	26.5	38.2
	DA-MAIC (TResNetM+ASL) [47]	36.6	55.4	49.8	60.4	44.7	47.3	64.1	54.4	65.8	71.4	39.3	39.7	59.9	44.6	51.1
Disc-free	DALN (TResNetm + ASL) [79]	29.4	44.7	43.7	23.8	27.6	48.9	27.4	35.1	65.6	**82.6**	21.3	32.0	**75.2**	22.1	34.1
	DDA-MLIC (OURS)	29.4	62.1	47.6	75.5	**55.3**	48.9	76.2	**59.6**	70.6	67.2	55.7	**49.3**	55.0	58.4	**56.6**
	DDA-MLIC with DeepEM (OURS)	29.4	**63.2**	50.7	50.9	42.7	**50.7**	56.3	53.4	**73.1**	74.9	49.5	47.7	63.1	51.0	56.4

Table 5.3: Sim2Real domain shift: comparison with the state-of-the-art in terms of number of model parameters (in millions), and % scores for mAP, per-class averages (CP, CR, CF1) and overall averages (OP, OR, OF1) for scene understanding datasets. Two settings are considered, *i.e.*, VOC → Clipart and Clipart → VOC. Best results are highlighted in **bold**.

Type	**Method**	**# params**	**VOC → Clipart**							**Clipart → VOC**						
			mAP	**P_C**	**R_C**	**F_C**	**P_O**	**R_O**	**F_O**	**mAP**	**P_C**	**R_C**	**F_C**	**P_O**	**R_O**	**F_O**
MLIC	ResNet101 [9]	42.5	38.0	64.8	14.3	22.5	82.3	18.3	29.9	50.1	66.2	17.5	25.5	83.9	29.6	43.7
	ML-GCN [39]	44.9	43.5	62.5	20.3	28.4	86.6	27.8	42.1	43.1	57.9	21.0	26.8	73.5	30.6	43.2
	ML-AGCN [32]	36.6	53.7	75.5	35.5	44.4	79.1	39.9	53.1	38.0	45.5	25.1	28.2	61.8	36.6	45.9
	ASL (TResNetM) [21]	29.4	56.8	72.0	38.5	47.6	82.8	45.7	58.9	64.2	69.0	30.7	37.3	80.0	45.7	58.2
Disc-based	DANN (TResNetM + ASL) [30]	29.4	47.0	77.0	22.0	32.5	86.8	23.6	37.1	67.0	76.8	23.3	32.6	**93.1**	20.4	33.4
	DA-MAIC (TResNetM+ASL) [47]	36.6	**62.3**	77.4	**42.6**	**51.6**	83.1	**51.0**	**63.2**	74.3	84.5	53.9	63.0	83.7	57.7	68.3
Disc-free	DALN (TResNetm + ASL) [79]	29.4	45.0	82.2	21.4	32.6	92.0	22.7	36.4	66.7	78.3	22.2	31.7	90.8	18.0	30.0
	DDA-MLIC (OURS)	29.4	61.4	**84.7**	28.1	39.4	90.9	33.3	48.8	77.0	86.9	29.3	38.2	88.4	35.3	50.4
	DDA-MLIC with DeepEM (OURS)	29.4	62.0	80.8	23.4	34.6	**94.8**	25.4	40.0	**82.8**	**88.6**	**57.0**	**65.8**	86.4	**58.8**	**70.0**

Table 5.4: Cross-weather domain shift: comparison with the state-of-the-art in terms of number of model parameters (in millions), and % scores of mAP, per-class averages (CP, CR, CF1) and overall averages (OP, OR, OF1) for urban street datasets. Cityscapes → Foggy is the setting that is considered. Best results are highlighted in **bold**.

Type	Method	# params	Cityscapes → Foggy						
			mAP	P_C	R_C	F_C	P_O	R_O	F_O
MLIC	ResNet101 [9]	42.5	58.2	53.6	27.8	32.2	**93.2**	48.3	63.7
	ML-GCN [39]	44.9	56.6	56.1	34.6	38.8	89.4	56.9	69.6
	ML-AGCN [32]	36.6	60.7	60.1	48.3	50.9	81.7	**71.2**	**76.1**
	ASL (TResNetM) [21]	29.4	61.3	66.7	**50.8**	**53.8**	79.2	70.5	74.6
Disc-based	DANN (TResNetM + ASL) [30]	29.4	53.5	50.6	12.5	18.6	89.5	21.8	35.1
	DA-MAIC (TResNetM+ASL) [47]	36.6	61.9	70.7	37.2	42.7	90.2	59.6	71.7
Disc-free	DALN (TResNetm + ASL) [79]	29.4	54.8	56.8	9.5	25.4	90.2	33.8	49.2
	DDA-MLIC (OURS)	29.4	62.3	**73.7**	45.7	48.9	84.1	69.3	76.0
	DDA-MLIC with DeepEM (OURS)	29.4	**63.2**	71.9	43.2	45.4	85.8	67.3	75.5

The first four rows of Table 5.1, Table 5.2, Table 5.3 and Table 5.4 report the obtained results using different methods of MLIC without DA [9], [21], [32], [39]. It can be observed that our method consistently outperforms all these methods under all settings (cross-sensor, sim2Real, and cross-weather) in terms of mAP showing the effectiveness of the proposed DA method for MLIC.

Furthermore, the results reported in the 5^{th} and 6^{th} rows of Table 5.1, Table 5.2, Table 5.3, Table 5.4 show that the proposed discriminator-free DA method clearly outperforms discriminator-based DA approaches for MLIC [30], [47] on cross-sensor and cross-weather domain shift settings in terms of mAP. This observation does not hold for the sim2Ream domain shift, where our approach records an mAP improvement of 8.5% over other discriminator-based approaches on Clipart → VOC setting, but wslightly surpasses with 0.3% in terms of mAP DA-MAIC [47] on VOC → Clipart setting.

We also compare our method to the discriminator-free method proposed in [79] for single-label DA and adapted to MLIC as stated in Section 5.2. Unsurprisingly, our method outperforms the adapted version of DALN for MLIC under all settings, reaching an improvement of more than 16% in terms of mAP on the Clipart → VOC and VOC → Clipart scheme.

Table 5.5: Ablation study (w/o: without, w/: with). The reported % scores are mAP.

Methods	**AID→UCM**	**UCM→AID**	**AID→DFC**	**UCM→DFC**	**VOC→Clipart**	**Clipart→VOC**
Ours w DeepEM	60.52	56.23	63.23	73.06	61.97	82.80
Ours w/o DeepEM	63.24 (+2.7)	54.90 (-1.4)	62.13 (-1.1)	70.64 (-2.4)	61.44 (-0.5)	76.96 (-5.8)
Ours w/o DA	55.45 **(-5.1)**	54.12 **(-2.1)**	56.09 **(-7.1)**	68.91 **(-4.1)**	56.78 **(-5.2)**	64.15 **(-18.7)**
Ours w/ Discr.	52.54 **(-8.0)**	51.60 **(-4.6)**	51.60 **(-11.6)**	64.06 **(-9.0)**	46.97 **(-15.0)**	67.03 **(-15.8)**

Table 5.6: mAP comparison of the proposed EM-based GMM clustering with k-means clustering.

Methods	**AID→UCM**	**UCM→AID**	**AID→DFC**	**UCM→DFC**	**VOC→Clipart**	**Clipart→VOC**
Ours w DeepEM	60.52	56.23	63.23	73.06	61.97	82.80
Ours (with k-means)	53.58 **(-6.9)**	52.20 **(-4.0)**	58.46 **(-4.8)**	68.06 **(-5.0)**	49.24 **(-12.7)**	68.27 **(-14.5)**

Table 5.7: mAP comparison of using KL-divergence and 1-Wasserstein (1W) distance as discrepancy for domain alignment.

Methods	**AID→UCM**	**UCM→AID**	**AID→DFC**	**UCM→DFC**	**VOC→Clipart**	**Clipart→VOC**
Ours w DeepEM	60.52	56.23	63.23	73.06	61.97	82.80
Ours (with KL)	56.44 **(-4.1)**	53.51 **(-2.7)**	53.17 **(-10.1)**	64.55 **(-8.5)**	52.62 **(-9.3)**	77.86 **(-4.9)**
Ours (with 1W)	53.60 **(-6.9)**	53.20 **(-3.0)**	57.80 **(-5.4)**	69.70 **(-3.4)**	60.50 **(-1.5)**	75.50 **(-7.3)**

Deep EM versus traditional EM

The last two rows of Table 5.1, Table 5.2, Table 5.3 and Table 5.4 report the results using the proposed DDA-MLIC without and with DeepEM. The adoption of a differentiable EM strategy showcases a substantial performance improvement under the three settings. It is worth highlighting that the mAP score is improved by approximately 6% in Sim2Real domain shift for Clipart → VOC with the proposed DeepEM block. This further supports the relevance of the proposed differentiable approach.

Training time

In order to showcase the efficiency of the proposed DeepEM, Fig. 5.5 compares the average training time needed to process one batch of source and target images using DDA-MLIC, with and without DeepEM. The figure shows that by replacing the traditional iterative EM process with an appropriate deep neural network significantly reduces the training time.

Table 5.8: Sensitivity analysis: A comparison of mAP (%) by varying the values of regularizers for each GMM component.

α **values** (α_1, α_2)	**Cross-sensor**				**Sim2real**		**Cross-weather**
	AID→UCM	**UCM→AID**	**AID→DFC**	**UCM→DFC**	**VOC→Clipart**	**Clipart→VOC**	**City→Foggy**
α_1=0.1, α_2=0.9	55.86	54.05	60.29	71.43	82.55	56.80	61.66
α_1=0.2, α_2=0.8	55.67	**56.23**	60.16	70.99	80.46	56.61	61.34
α_1=0.3, α_2=0.7	56.00	54.20	**63.23**	**73.06**	81.92	58.48	**63.23**
α_1=0.4, α_2=0.6	55.57	55.89	60.65	72.84	81.29	57.71	61.57
α_1=0.5, α_2=0.5	57.92	55.00	60.91	71.33	**82.80**	**61.67**	61.80
α_1=0.6, α_2=0.4	54.85	55.72	61.22	70.19	81.86	58.28	62.95
α_1=0.7, α_2=0.3	58.35	54.44	61.51	71.85	82.26	57.52	60.59
α_1=0.8, α_2=0.2	**60.52**	55.46	58.96	70.78	81.39	58.03	61.86
α_1=0.9, α_2=0.1	58.21	54.14	58.42	71.40	81.54	57.83	60.91
α_1=1.0, α_2=0.0	58.46	54.83	58.47	72.15	81.87	59.94	61.85

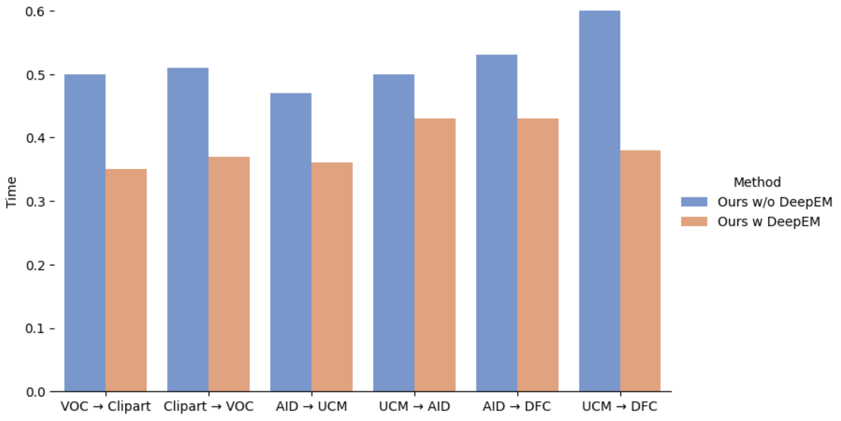


Figure 5.5: Comparison of average training time per batch.

Ablation Study

The results of the ablation study are presented in Table 5.5. We report the obtained mAP for the following settings, *i.e.*, AID → UCM, UCM → AID, UCM → AID, UCM → DFC, VOC → Clipart and Clipart → VOC. The initial two rows display the mAP results obtained using the proposed approach with *(w)* and without *(w/o)* the inclusion of the DeepEM. The third row illustrates the mAP score in the absence of any domain adaptation strategy. The final row shows the results achieved when employing an adversarial domain adaptation approach, utilizing a conventional domain discriminator. Clearly, leveraging the classifier as a discriminator leads to a noticeable improvement in classification performance when dealing with a domain shift.

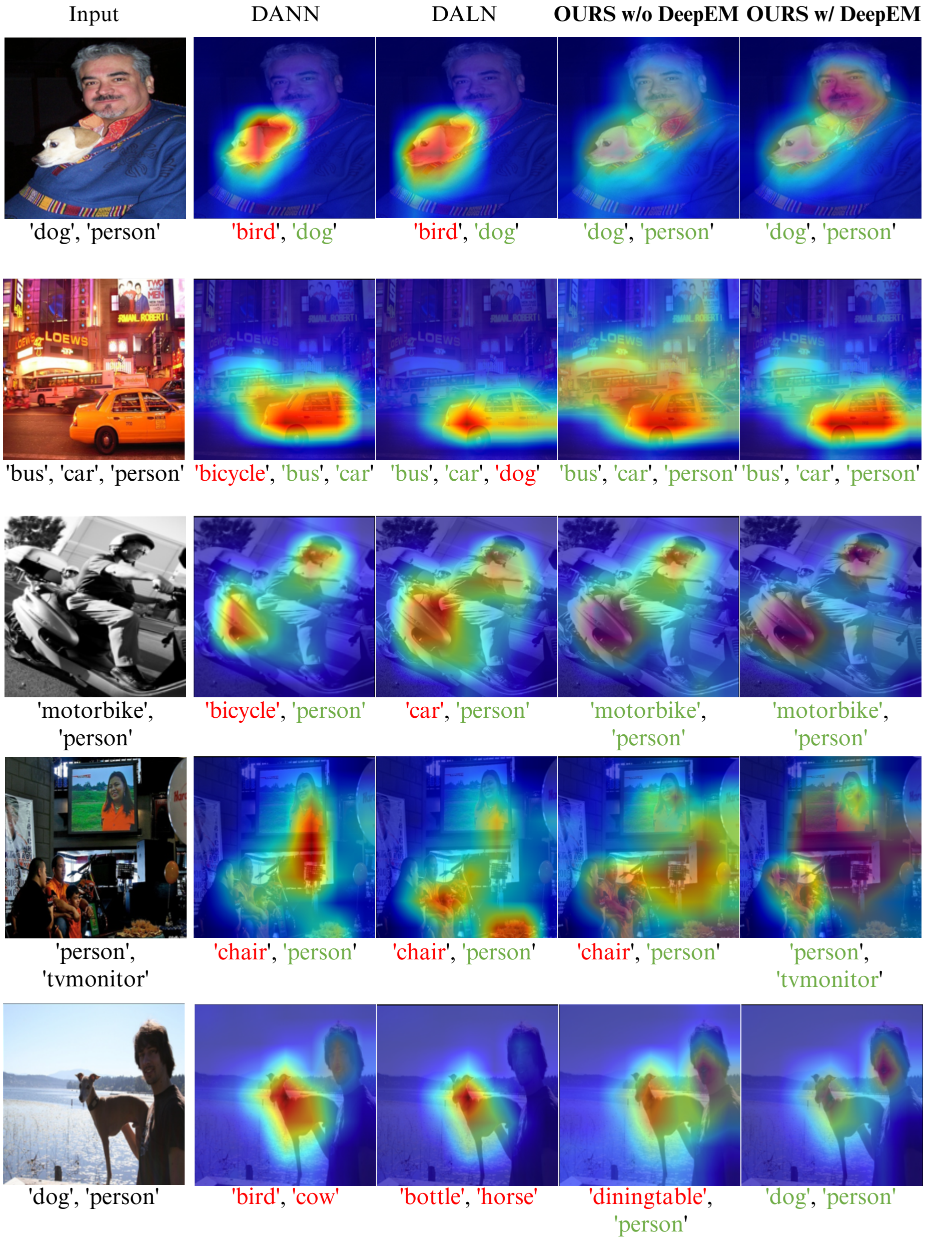


Figure 5.6: Qualitative analysis: Heatmap visualization of the proposed approach, existing discriminator-based (DANN [30]), and discriminator-free (DALN [79]) methods. The first column exhibits some input images with their ground truth labels, while the next columns display the heatmaps generated by the considered methods, with predicted labels highlighted in green (if correct) and red (if incorrect).

5.6.3 Qualitative Analysis

Fig. 5.6 presents a qualitative comparison between the proposed method and existing approaches. Specifically, we compare the proposed discriminator-free approach (with and without DeepEM) to both existing discriminator-based (DANN [30]) and discriminator-free (DALN [79]) methods. For a set of image samples, we visualize the Gradient-weighted Class Activation Mapping (Grad-CAM) [81] for all the aforementioned methods, along with the corresponding predicted labels. It can be noted that compared to our method both DANN and DALN fail to precisely activate the regions incorporating the present objects, hence leading to an incorrect prediction of the labels. Furthermore, the relevance of the differentiable EM-based strategy is illustrated in the last two columns of Fig. 5.6. In these examples, the DeepEM-based network allows focusing more precisely on the corresponding objects, as compared to the standard EM, thereby enabling the correct prediction of the different categories present in the displayed images.

5.6.4 Additional Analysis

Sensitivity analysis

In Table 5.8, we report the variation in performance when varying the values α_1 and α_2, defined in Eq. (5.11). More specifically, we report the mAP score achieved by DDA-MLIC with DeepEM for ten different combinations of α_1 and α_2 for the three types of domain shift. We start by giving smaller weights to the first GMM component (negative labels) and higher importance to the second one(positive labels). In most cases, especially under the cross-sensor and cross-weather domain shifts, we observe that assigning more weights to the positive component yields a better mAP score. This is in line with the reasoning behind the widely used Asymmetric Loss (ASL) [21], which focuses more on the positive labels than negative ones. For the Sim2real domain shift, however, assigning equal weights to both components provides the best mAP score.

GMM versus k-means

To justify the choice of GMM as a clustering technique, we compare it with another popular non-probabilistic clustering technique, known as k-means. In contrast to k-means, which uses hard thresholding to assign data points to specific clusters, GMM employs soft thresholding by maximizing the likelihood. Table 5.6 compares the mAP scores obtained using the two methods. It is evident that using k-means results in a significant performance drop for all benchmarks.

Distance and divergence measure analysis

As mentioned in Section 5.5.1, the proposed adversarial critic, directly derived from the task-specific classifier, is based on the 2-Wasserstein distance (denoted as 2W) between the estimated source and target GMM components. To demonstrate the effectiveness of the 2W distance over other popular distances, we report, in Table 5.7 the mAP scores when integrating the KL-divergence and the 1-Wasserstein distance in our framework. The results clearly demonstrate that utilizing the 2W distance as a discrepancy measure outperforms other distances, thanks to its continuity and differentiability properties. Specifically, using the KL divergence or the 1-W distance as a discrepancy measure results in a slight to significant reduction in mAP across all benchmarks, ranging from 1.5% to 10%.

5.7 Limitations

While the proposed method has demonstrated superior performance across nearly all benchmarks, it is essential to acknowledge certain limitations. Specifically, in the case of the AID→UCM benchmark (see Table 5.1), we observe that, under the cross-sensor domain shift, the proposed DDA-MLIC with DeepEM did not yield a noticeable performance improvement when compared to its counterpart based on the traditional EM. Furthermore, in the case of the sim2real domain shift, particularly in the VOC→Clipart benchmark (see Table 5.3), the performance of the discriminator-free method did not surpass the discriminator-based approach DA-MAIC [47]. We assume that this drop in performance might be due to the absence of an additional graph-based subnet, which is a crucial component for explicitly modeling label correlations. This limitation highlights the need for investigating additional strategies for modeling the label correlations. Moreover, our approach assumes that the categories that are present in source and target images are identical. Hence, it would be interesting to investigate in future works more challenging scenarios such as open-set unsupervised domain adaptation, where the source and target datasets might include non-common labels.

5.8 Conclusion

In this chapter, a discriminator-free UDA approach for MLIC has been introduced. Unlike existing methods that employ an additional discriminator trained adversarially, our method utilizes the task-specific classifier to implicitly discriminate between source and target domains. This strategy aims to enforce learning domain-invariant features, while avoiding mode collapse. To achieve this, we redefine the adversarial loss using a Fréchet distance between the corresponding Gaussian Mixture Model (GMM) components estimated from the classifier probability predictions. A DNN-based Deep Expectation Maximization (DeepEM) is proposed to estimate the parameters of the GMM for ensuring differentiability and avoiding a costly iterative optimization. Experiments conducted on several benchmarks encoding different domain shifts demonstrated that the proposed approach achieves state-of-the-art performance, while reducing the need for cumbersome architectures.

Chapter 6

Application of MLIC to Multi-label Deepfake Detection

In this chapter, we detail our investigation regarding the suitability of current multi-label classification approaches for deepfake detection. With the recent advances in generative modeling, new deepfake detection methods have been proposed. Nevertheless, as highlighted in Chapter 1.3.4, they mostly formulate this topic as a binary classification problem, resulting in poor explainability capabilities. Indeed, a forged image might be induced by multi-step manipulations with different properties. For a better interpretability of the results, recognizing the nature of these stacked manipulations is highly relevant. For that reason, we propose to model deepfake detection as a multi-label classification task, where each label corresponds to a specific kind of manipulation. In this context, state-of-the-art multi-label image classification methods are considered. Extensive experiments are performed to assess the practical use case of deepfake detection.

6.1 Introduction

The recent advances in Deep Learning (DL) techniques have led to the emergence of highly realistic facial manipulations, known as deepfakes. The subtlety of these forgeries makes their distinction from authentic images increasingly challenging. Given this threat, many efforts have been dedicated to developing deepfake detection techniques [102]–[104]. Typically, these approaches formalize the problem of deepfake detection as a binary classification [102], [103], [105]–[109]. Given an input image or video, they predict whether it has been forged or not; therefore classifying it as 'real' or 'fake'. However, binary predictions are opaque and are difficult to interpret, while in real-world applications, explainable predictions in deepfake detectors are of utmost importance. In fact, an image predicted as fake can be produced by one or multiple manipulations. In existing face editing software, such as FaceTune[1], it is common for the same image to undergo several edits, which we refer to as *stacked manipulations* or *multi-step operations*, as illustrated in Figure 6.1 (a).

As an alternative, in this chapter, we introduce our paper that aims to reformulate the task of deepfake detection as a multi-label classification problem, where each label corresponds to a specific manipulation. Such a formulation is supported by the fact that multiple forgeries can be present in the same image.

Recently, Shao et al. [36] highlighted the necessity of detecting multi-step manipulations. For that purpose, they have introduced a novel deepfake dataset incorporating sequences of facial forgeries, along with their annotations. However, instead of considering multi-label classification, they framed the problem of deepfake detection as an image-to-sequence task. This means that their goal was not only to recognize the different manipulations applied to a given image, but also to retrieve their chronological order. Nevertheless, predicting the temporal structure of a forgery sequence adds

[1] https://www.facetuneapp.com/

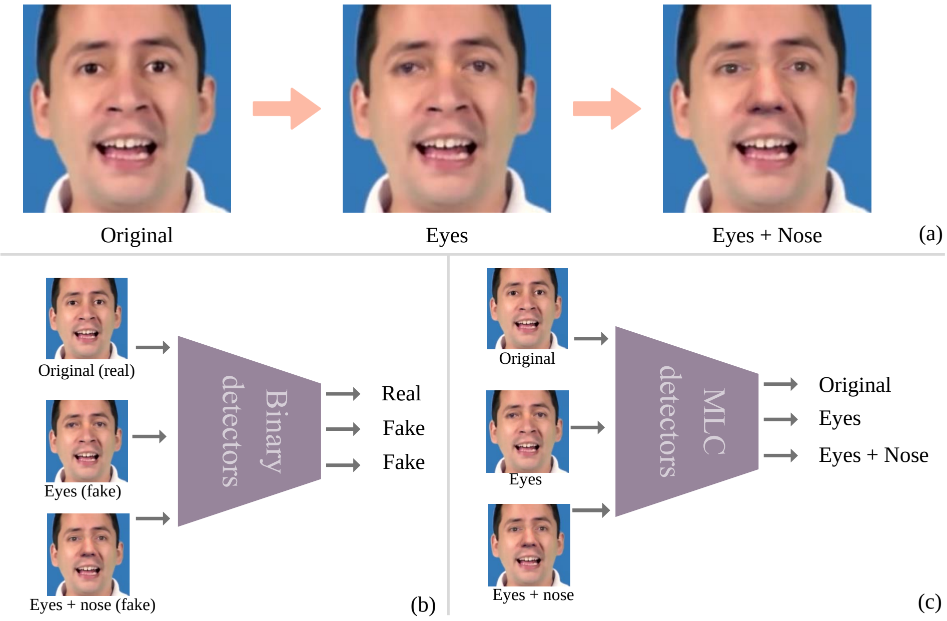


Figure 6.1: **(a)** Examples of single-step manipulation affecting only the eyes and multi-step manipulations affecting both the eyes and the nose. **(b)** Binary deepfake detectors treat single-step and multi-step forged images equally, which implicitly assumes that only one manipulation took place in the image. **(c)** Whereas Multi-Label deepfake Classifiers (MLC) predict more informative outputs by indicating the labels of the applied manipulations.

complexity to the problem without having a clear benefit in a practical scenario.

In this paper, we argue that for detecting stacked manipulations, it is sufficient to formulate deepfake detection as a multi-label image classification task. As we are the first to explicitly rethink deepfake detection as such, we propose to show the suitability of existing multi-label image classification methods for the practical scenario of detecting multi-step manipulations. Our main finding is that current deepfake multi-label image datasets might be too simplistic since they were created under controlled conditions. This emphasizes the need for more realistic deepfake datasets, as the existing ones may not accurately reflect the performance of state-of-the-art multi-label classification methods.

In summary, our contributions are twofold: (1) we reformulate deepfake detection as a multi-label classification problem and show that more explainable predictions can be achieved regardless of the forgery order; (2) we compare multiple state-of-the-art multi-label classification techniques in the context of deepfake detection and present an extensive analysis of the obtained results.

In the remainder of this chapter, Section 6.2 formulates the problem of multi-label deepfake classification. Section 6.3 presents an overview of the considered multi-label image classification techniques. In Section 6.4, we detail the experimental setup and present our results. Finally, Section 6.5 concludes this work and offers interesting perspectives.

6.2 Formulating Deepfake Detection as a Multi-label image Classification problem

Let $(\mathcal{I})$ be a dataset formed by a set of real and fake images. Given an image $\mathbf{I} \in \mathcal{I}$, traditional deepfake detection methods consider that the label of $\mathbf{I}$ belongs to $[\![0, 1]\!]$. In other words, they classify an image as real or fake, formulating the problem as a simple binary classification. Nevertheless, a deepfake image might result from multi-step manipulations that enclose different properties. As detecting the nature of these manipulations is highly relevant for obtaining a more explainable output, we propose to define the problem of deepfake detection as a multi-label classification. Let $\mathbf{I} \in \mathcal{I}$ be a given image, we aim at estimating a function f that predicts the presence or not of N different manipulations. This can be written as follows,

$$\begin{aligned} f \colon \mathbb{R}^{w \times h} &\to [\![0, 1]\!]^N \\ \mathbf{I} &\mapsto \mathbf{y} = (y_i)_{i \in [\![1, N]\!]}, \end{aligned} \tag{6.1}$$

where w and h are respectively the pixel-wise width and height of the image. It is to note that $y_i = 1$ if the manipulation i is present in $\mathbf{I}$, otherwise, $y_i = 0$.

6.3 Comparison of multi-label image classification for deepfake detection

The multi-label image classification problem has received a lot of attention from the computer vision research community in recent years. Many methods have demonstrated outstanding performances in light of current developments in deep learning techniques. In this chapter, we evaluate the performance and assess the current state of existing multi-label image classification methods in the context of deepfake detection, as formulated in Chapter 6.2. For that purpose, as mentioned in Chapter 1.1, two main categories of methods are considered, namely, direct and indirect methods. We describe these methods in the subsections that follow.

6.3.1 Direct methods

In order to determine if multiple objects are present in an image or not, direct methods employ a single stream deep neural network f that directly maps a given image to a binary vector. In other words, f is usually learned in an end-to-end manner in this case. Generally, these single-stream architectures are constituted of two main components, namely: (1) a block of Convolutional Neural Networks (CNN) which seeks to extract discriminative image features; and (2) a classification head that employs a Multi-Layer Perceptron (MLP) to directly translate these features into the probability of occurrence of each considered label.

Among direct methods, the ResNet architecture is probably one of the most successful [9], [21], [24], [25]. Recently, TResNet [24]-an improved version of ResNet [9] that takes advantage of GPU capabilities, has also been proposed for multi-label classification. Moreover, by combining the recently introduced Asymmetric Loss (ASL) [21] with TResNet, improved results have been achieved. Note that the ASL loss acts differently on positive and negative labels.

Herein, we compare the effectiveness of some popular direct techniques in the context of deepfake detection, namely: (1) ResNet50 [9]; (2) ResNet101 [9]; and (3) TResNetM [24]. Additionally, we couple these methods with ASL [21]. Chapter 6.4 provides more details on the quantitative performance of these methods.

6.3.2 Indirect methods

While direct approaches have shown great performance, they tend to require a large number of layers to work effectively, as mentioned in Chapter 1.2.1. To avoid using very deep networks, a second research line has attempted to model label dependencies. In fact, label correlations are important cues since some labels are more likely to appear together in the same image. For example, we have a higher chance to observe a "sheep" and some "grass" in one image than a "sheep" and a "bicycle". We refer to these approaches as indirect methods.

Graphs have been particularly useful for modeling label correlations. Graph-based approaches are typically formed by two streams. They usually combine a CNN denoted by f_1 that learns discriminative image features with a Graph Convolutional Network (GCN) for generating interdependent label-wise classifier denoted by f_2 [31], [32], [39]. These generated classifiers are directly applied to the features resulting from f_1. In other words, images are mapped to a binary vector using the function $f = f_2 \circ f_1$. The pioneering work on graph-based multi-label classification [39] made use of word embeddings [42] to represent graph nodes. More recent techniques [31], [32] have generated image-based embeddings to improve the performance. Additionally, earlier graph methods are mainly based on a pre-computed fixed adjacency matrix where weak edges are ignored using an empirically fixed threshold. This may lead to a significant loss of information. To overcome this issue, ML-AGCN [32] attempts to adaptively learn the adjacency matrix by computing an attention weight for each node pair.

Herein, we compare the effectiveness of some recent indirect graph-based techniques in the context of deepfake detection, namely: (1) ML-GCN [39]; (2) IML-GCN [31]; and (3) ML-AGCN [32]. We use both word [42] and image-based [31] node embeddings to assess the performance of the aforementioned indirect methods. We generate the label graph using the co-occurrences of each manipulation pair in an image over the entire dataset, as in [39]. Chapter 6.4 gives more details on the quantitative performance of these methods.

6.4 Experiments

6.4.1 Datasets

For our experiments, we use the dataset referred to as *Deep-Seq* proposed in [36]. Initially, this dataset was proposed for image-to-sequence tasks. Nevertheless, its annotations are compatible with multi-label classification. As compared to [36], the order constraint is not considered. More specifically, the dataset consists of two subsets depicting various manipulations. The first subset, called *Sequential facial components manipulations (Seq-Com-Deepfake)*, shows forgeries that alter the appearance of facial attributes such as hair bangs or the beard. In the second subset, the manipulations are applied by swapping facial regions, such as the eyes, the mouth, etc., between an original and a reference image, respectively. This subset is termed *Sequential facial attributes manipulations (Seq-Att-Deepfake)*. For both sub-collections, one to five manipulations are applied to the same image. Hence, the label vector is formed by five elements ($N = 5$).

6.4.2 Evaluation metrics

We provide the mean Average Precision (mAP) as well as the number of model parameters (# Params) in order to assess the effectiveness of current state-of-the-art multi-label classification approaches in the context of deepfake detection. In addition, as in [21], we report the following evaluation metrics on both subsets of the Deep-Seq dataset: average per-Class Precision (CP), average per-Class Recall (CR), average per-Class F1-score (CF1), the average Overall Precision (OP), average overall recall (OR) and average Overall F1-score (OF1).

6.4.3 Implementation details

In the context of deepfake detection, the effectiveness of both direct and indirect approaches is assessed. For that purpose, we employ ResNet [9] and TResNet [24] as direct approaches, in addition to ML-GCN [39], IML-GCN [31] and ML-AGCN [32] as indirect methods. More specifically, we utilize both Resnet50 and Resnet101 variations. For TResNet, we adapt a smaller version known as TResNet-M.

We use the original train and test split that was initially provided in the dataset [36] to train our models. For the subset of facial attribute manipulations (Seq-Attr-Deepfake), we use 41600 samples for training and 4160 samples for testing, and for the subset of facial component manipulations (Seq-Comp-Deepfake), we use 29408 and 2860 samples for training and testing, respectively. Using conventional image augmentation techniques, the image samples are reshaped to 224x224 as suggested in the original methods [9], [24], [32], [39]. We train the models on an NVIDIA TITAN V GPU with a total memory of 12GB using PyTorch in Python with a batch size of 128 for a total of 40 epochs or until convergence.

Table 6.1: Comparison of existing multi-label image classification methods on deepfake attribute manipulations subset (**Seq-Attr-Deepfake**). Best results are highlighted in bold.

Category	**Method**	**# Params** (↓)	**mAP** (↑)	**CP** (↑)	**CR** (↑)	**CF1** (↑)	**OP** (↑)	**OR** (↑)	**OF1** (↑)
Direct methods	ResNet50 [9]	**23.8**	96.0	**93.5**	80.7	86.5	93.7	80.9	86.9
	ResNet 50 (with ASL) [9]	**23.8**	95.9	89.2	91.6	**90.4**	89.3	91.7	**90.5**
	ResNet101 [9]	42.8	**96.1**	92.9	83.6	87.8	93.1	83.7	88.2
	ResNet101 (with ASL) [9]	42.8	96.0	88.1	**92.3**	90.1	88.1	**92.4**	90.2
	TResNetM [24]	29.4	93.9	93.4	69.2	79.1	**93.8**	69.4	79.8
	TResNetM (with ASL) [21]	29.4	94.0	86.1	89.5	87.7	86.2	89.5	87.8
Indirect methods	ML-GCN† [39]	44.9	**95.1**	93.3	74.9	82.9	93.6	75.0	83.3
	IML-GCN [31]	**31.6**	82.5	76.7	72.4	74.3	76.9	72.6	74.7
	IML-GCN† [31]	31.7	94.0	84.8	90.6	87.6	84.9	90.6	87.7
	ML-AGCN [32]	36.3	82.7	74.8	77.1	75.9	74.9	77.1	76.0
	ML-AGCN† [32]	36.6	94.3	85.3	**90.9**	**88.0**	85.3	**90.9**	**88.0**
	ML-AGCN† w/o ASL [32]	36.6	94.5	**93.8**	70.0	79.9	**94.1**	70.2	80.4

†Graph-based indirect approaches with word embeddings [42]

Table 6.2: Comparison of existing multi-label image classification methods on deepfake component manipulations subset (**Seq-Comp-Deepfake**). Best results are highlighted in bold.

Category	**Method**	**# Params** (↓)	**mAP** (↑)	**CP** (↑)	**CR** (↑)	**CF1** (↑)	**OP** (↑)	**OR** (↑)	**OF1** (↑)
Direct methods	ResNet50 [9]	**23.8**	89.8	89.0	68.5	77.0	89.6	68.5	77.6
	ResNet 50 (with ASL) [9]	**23.8**	90.5	80.3	87.7	83.8	80.3	87.9	84.0
	ResNet101 [9]	42.8	91.7	**89.3**	74.4	80.7	**89.7**	74.5	81.4
	ResNet101 (with ASL) [9]	42.8	**92.7**	82.7	**90.0**	**86.2**	82.7	**90.0**	**86.2**
	TResNetM [24]	29.4	87.1	88.3	58.2	68.6	89.4	58.7	70.8
	TResNetM (with ASL) [21]	29.4	87.2	79.9	82.1	80.9	80.1	82.2	81.1
Indirect methods	ML-GCN† [39]	44.9	**89.6**	87.5	69.4	76.9	87.8	69.8	77.8
	IML-GCN [31]	**31.6**	81.7	**94.9**	18.8	27.9	92.3	20.0	32.9
	IML-GCN† [31]	31.7	87.7	79.8	82.1	80.9	80.0	82.2	81.1
	ML-AGCN [32]	36.3	81.7	80.1	65.5	71.8	80.2	65.3	72.0
	ML-AGCN† [32]	36.6	87.1	78.1	**84.9**	**81.3**	78.2	**85.1**	**81.5**
	ML-AGCN† w/o ASL [32]	36.6	88.0	90.7	54.4	65.5	**91.7**	54.7	68.5

†Graph-based indirect approaches with word embeddings [42]

6.4.4 Experimental Results

We report in Table 6.1 and Table 6.2 the results obtained for both Seq-Att-Deepfake and Seq-Com-Deepfake subsets, respectively.

Comparison of direct methods

In general, all the results obtained for direct methods are comparable. However, it is interesting to note that, in our experiments, ResNet50 outperforms TResNetM in terms of mAP regardless of ResNet's depth, with an improvement of approximately 2%. It should be noted, though, that TResNetM allows for a larger batch size than ResNet50 while still utilizing the same GPU memory.

In addition, surprisingly, the use of the ASL loss does not seem to influence the results importantly on the Seq-Attr-Deepfake subset, only inducing a variation of 0.1% in mAP. On the other hand, a marginal performance improvement on the other subset i.e., Seq-Comp-Deepfake can be noticed when comparing direct methods to their counterpart ASL-based ones. However, the recall (CR, OR) and F-1 score (CF, OF), both per-class and overall, increase significantly when these methods are combined with ASL. This might be explained by two points: 1) since ASL aims to focus more on positive labels than negative ones, the model tends to predict more false positives; and 2) the models may overfit the distribution of manipulations in Seq-Attr-Deepfake.

Finally, it can be noted that the best performance is achieved when using a deeper architecture, i.e ResNet101, with an enhancement between 2% and 4%, in terms of mAP, on Seq-Attr-Deepfake and Seq-Com-Deepfake, respectively. Nevertheless, this slight improvement comes at the cost of an important increase in terms of number of parameters (almost multiplied by a factor of 2).

Comparison of indirect methods

The largest architecture corresponding to ML-GCN outperforms other graph-based methods. More specifically, an improvement of 0.5-12% and 1.6-8% can be observed in terms of mAP for Seq-Attr-Deepfake and Seq-Comp-Deepfake subsets, respectively. This is consistent with the results obtained for direct methods, as the feature extraction branch of ML-GCN is based on ResNet101. Moreover, while image embeddings have significantly improved the performance on standard multi-label image classification datasets, word embeddings give a higher mAP when tested on both deepfake subsets. In fact, as reported in Table 6.1 and Table 6.2, for both IML-GCN and ML-AGCN, the mAP decreases by more than 12% when paired with image-based embeddings. This is

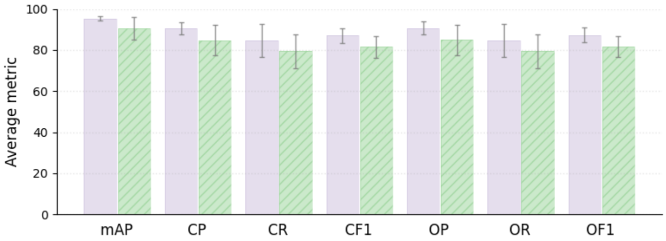


Figure 6.2: Comparison of the average performance of numerous direct (non-hatched) and indirect (hatched) approaches on the Seq-Attr-Deepfake subset.

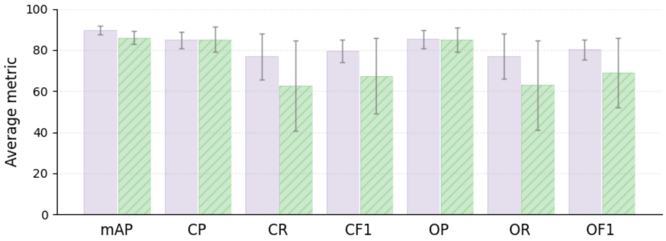


Figure 6.3: Comparison of the average performance of numerous direct (non-hatched) and indirect (hatched) approaches on the Seq-Comp-Deepfake subset.

counter-intuitive since, unlike the word embeddings that were initially proposed for the task of Natural Language Processing (NLP), image-based embeddings are semantically more meaningful for image classification as discussed in [31]. This might be caused by the fact that the discrepancy between the image embeddings produced by two different manipulations is not significant. In contrast to generic objects, image embeddings may fail to describe the manipulation semantics. Last but not least, the attention mechanism proposed in [32] does not improve the performance of the standard ML-GCN.

Direct methods versus indirect methods

In Fig. 6.2 and Fig. 6.3, we visualize the average performance of both direct and indirect methods on Seq-Attr-Deepfake subset and Seq-Comp-Deepfake subset, respectively. Given Fig. 6.2 and Fig. 6.3, two observations can be made. First, direct methods seem to be more suitable for multi-label deepfake classification. This comes again in contradiction with the results obtained in the generic field of multi-label image classification, showing that modelling the label correlations is highly beneficial. This can be explained by the fact that the present deepfake dataset has not been spontaneously generated, but has been produced in a controlled environment. The distribution of the generated manipulations is assumed to be uniform, which does not necessarily reflect a realistic scenario. Second, we can observe that the facial components manipulations subset is relatively more challenging than the facial attribute subset, especially for indirect methods that enclose a high standard deviation in terms of performance metrics.

6.5 Conclusion

Existing deepfake detection techniques model the problem as a simple binary classification task, with the aim to determine whether or not a particular image is fake. However, this makes the classification task hardly interpretable. For obtaining more explainable outputs, the work presented in this chapter proposes to tackle deepfake detection problem as a multi-label classification problem, with the objective of simultaneously identifying several categories of image manipulations. To this end, state-of-the-art multi-label classification methods are benchmarked on a recently proposed deepfake dataset incorporating multi-label annotations. This allows assessing the effectiveness of current multi-label classification methods, including both direct and indirect, in the practical use case of deepfake detection. Multiple results are against intuition, showing the need to investigate further multi-label deefake classification. These future investigations might be supported by the introduction of more complex and realistically generated multi-label deepfake datasets.

Chapter 7

Conclusion

In this chapter, we briefly summarize the main findings of our research works in the field of MLIC. We further discuss the potential extension and future directions of this thesis work.

7.1 Summary

In this thesis, we address the problem of Multi-label Image Classification (MLIC), both within a single-domain and under a cross-domain setup. Our first contribution, namely IML-GCN [31], addresses the problem of cumbersome network architectures in existing works by utilizing more meaningful label representations in a CNN-GCN-based setting for the general task of MLIC. Thanks to the proposed image-based node embeddings, IML-GCN efficiently learns to model the relationships between multiple objects using a much smaller CNN backbone compared to the existing methods. The reported results are competitive with the state-of-the-art methods while keeping a smaller overall model size.

Our second contribution, termed ML-AGCN [32] aims at addressing the three main limitations of existing GCN-based MLIC approaches, as mentioned in Chapter 4.1, namely; 1) the fixed label graph topology, 2) the heuristically defined threshold to ignore noisy edges, and 3) the dissimilarity in graph node features. The proposed ML-AGCN overcomes these limitations by following an adaptive label graph learning strategy leading to more robust and efficient multi-label classifiers. Furthermore, we extend the adaptive learning strategy to tackle the problem of domain shifts in MLIC. The proposed work, namely DA-AGCN [33], makes use of an additional discriminator to implicitly minimize the domain gap between two different domains of the input images.

However, the usage of an additional domain discriminator in adversarial-based UDA approaches leads to the problem of mode collapse. To tackle this issue, as a third contribution, we propose to reuse the multi-label classifier as a discriminator in an MLIC framework termed DDA-MLIC [34], [35]. Specifically, a Gaussian Mixture Model (GMM)-based adversarial critic is directly derived from the output of the classifier thereby eliminating the need for an additional discriminator. A deep neural network-based GMM parameter estimation strategy is adopted, making the entire learning process differentiable and efficient in an end-to-end manner. We showcase that the proposed methodology achieves state-of-the-art results on several MLIC for UDA benchmarks.

Finally, in [37], we investigate the applicability of MLIC in solving an important real-world problem, namely, deepfake detection. More specifically, we propose to evaluate the effectiveness of existing MLIC methods, including the ones proposed in this thesis, for deepfake detection. Existing methods formulate the task of identifying forgeries in an image by considering it as a binary classification, i.e. whether the given image is fake or not leading to less interpretable outcomes. Hence, in [37], we propose to reformulate deepfake detection as a multi-label classification problem where each label corresponds to the type of manipulation in images. Our analysis showcases the need for more complex and realistic multi-label deepfake datasets.

7.2 Future Directions

As future works, we intend to investigate research directions motivated by two main applications, namely, deepfake detection and space situation awareness.

7.2.1 Deepfake Detection

In Chapter 6, we highlighted the need to generate realistic and more complex multi-label deepfake datasets that can leverage the state-of-the-art performance of existing MLIC methods. In fact, deepfakes as any visual data might be exposed to the problem of the domain shift e.g., different illuminations, unconstrained environment, etc that can impact the performance of multi-label deepfake detectors. Therefore, exploring Unsupervised Domain Adaptation to reduce the impact of the domain shift seems highly relevant. For instance, it would be interesting to utilize a synthetically generated pseudo-fake dataset to improve the performance of an unlabeled actual deepfake dataset. The research direction of UDA for multi-label deepfake detection is still under-explored and hence, opens new opportunities that can save costly annotation efforts by utilizing synthetically generated datasets that are practically free to annotate.

7.2.2 Space Situation Awareness

An emerging applicative field where research can be foreseen is Space Situational Awareness (SSA). With the ever-increasing number of satellites orbiting around Earth, it is of utmost importance to protect them from colliding with space debris to allow smooth and uninterrupted services provided by these satellites. Hence, identifying the multiple components of the item present in a space image can be useful to differentiate the spacecraft from the debris. However, obtaining space imagery itself is a very expensive task, not to mention the additional challenges it may pose for annotating them.

Numerous efforts are being made by researchers to generate space-like lab-simulated synthetic datasets. An interesting and under-explored research direction can be towards exploring Unsupervised Domain Adaptation (UDA) for MLIC by exploiting these datasets in order to generalize well on the real-space images.

Additionally, the idea of discriminator-free UDA for classification (DDA-MLIC) can be extended and explored for regression problems, especially keypoint detection for identifying the pose of a spacecraft and object detection for trajectory estimation. The research problem for regression is more challenging than the classification due to the continuous nature of the output. Hence, the identification of an adversarial critic from the regressor opens up an interesting direction to study. This practical use case can directly tackle real-world problems like spacecraft rendez-vous for automatic docking.

References

[1] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, “Microsoft coco: Common objects in context,” in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, Springer, 2014, pp. 740–755.

[2] K. Papadopoulos, E. Ghorbel, O. Oyedotun, D. Aouada, and B. Ottersten, “Deepvi: A novel framework for learning deep view-invariant human action representations using a single rgb camera,” in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, IEEE, 2020, pp. 138–145.

[3] K. Papadopoulos, E. Ghorbel, D. Aouada, and B. Ottersten, “Vertex feature encoding and hierarchical temporal modeling in a spatio-temporal graph convolutional network for action recognition,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 452–458.

[4] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[5] Y. Cai, L. Ge, J. Liu, *et al.*, in *In Proceedings of the IEEE/CVF international conference on computer vision*, Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks, 2019, pp. 2272–2281.

[6] E. Ghorbel, K. Papadopoulos, R. Baptista, *et al.*, “A view-invariant framework for fast skeleton-based action recognition using a single rgb camera,” in *14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Prague, 25-27 February 2018*, 2019.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[8] N. Wang and D.-Y. Yeung, “Learning a deep compact image representation for visual tracking,” *Advances in neural information processing systems*, vol. 26, 2013.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[11] Y. Li, C. Huang, C. C. Loy, and X. Tang, “Human attribute recognition by deep hierarchical contexts,” vol. 9910, Oct. 2016, pp. 684–700, ISBN: 978-3-319-46465-7. DOI: 10.1007/978-3-319-46466-4_41.

[12] J. Shao, K. Kang, C. C. Loy, and X. Wang, “Deeply learned attributes for crowded scene understanding,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4657–4666. DOI: 10.1109/CVPR.2015.7299097.

[13] J. Shao, C. C. Loy, K. Kang, and X. Wang, “Slicing convolutional neural network for crowd video understanding,” in *2016 IEEE Conference on Computer Vision*

and Pattern Recognition (CVPR), 2016, pp. 5620–5628. DOI: 10.1109/CVPR.2016.606.

[14] K. Kang, W. Ouyang, H. Li, and X. Wang, “Object detection from video tubelets with convolutional neural networks,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. DOI: 10.1109/cvpr.2016.95. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2016.95.

[15] I. Singh, N. Mejri, V. Nygyen, and D. Ghorbel E. anf Aouada, “Multi-type deepfake detection,” in *MMSP*, 2023.

[16] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, Ieee, vol. 1, 2005, pp. 886–893.

[17] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[18] C. J. Burges, “A tutorial on support vector machines for pattern recognition,” *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[19] F. Zhu, H. Li, W. Ouyang, N. Yu, and X. Wang, “Learning spatial regularization with image-level supervisions for multi-label image classification,” Jul. 2017, pp. 2027–2036. DOI: 10.1109/CVPR.2017.219.

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

[21] T. Ridnik, E. Ben-Baruch, N. Zamir, *et al.*, “Asymmetric loss for multi-label classification,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 82–91.

[22] A. Torralba and A. A. Efros, “Unbiased look at dataset bias,” in *CVPR 2011*, IEEE, 2011, pp. 1521–1528.

[23] Y. Ganin and V. Lempitsky, “June. unsupervised domain adaptation by backpropagation,” in *International conference on machine learning*, PMLR, 2015, pp. 1180–1189.

[24] T. Ridnik, H. Lawen, A. Noy, E. Ben Baruch, G. Sharir, and I. Friedman, “Tresnet: High performance gpu-dedicated architecture,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1400–1409.

[25] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[26] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[27] J. Lanchantin, T. Wang, V. Ordonez, and Y. Qi, “General multi-label image classification with transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 478–16 488.

[28] T. H. L. Ruyang Liu Jingjia Huang and G. Li, “Causality compensated attention for contextual biased visual recognition.,” in *ICLR*, 2023.

[29] X. Cheng, H. Lin, X. Wu, *et al.*, “Mltr: Multi-label classification with transformer,” in *2022 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2022, pp. 1–6.

[30] Y. Ganin, E. Ustinova, H. Ajakan, *et al.*, “Domain-adversarial training of neural networks,” *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[31] I. P. Singh, O. Oyedotun, E. Ghorbel, and D. Aouada, “Iml-gcn: Improved multi-label graph convolutional network for efficient yet precise image classification,” *In AAAI-22 Workshop Program-Deep Learning on Graphs: Methods and Applications*, 2022.

[32] I. P. Singh, E. Ghorbel, O. Oyedotun, and D. Aouada, “Multi label image classification using adaptive graph convolutional networks (ml-agcn),” in *In IEEE International Conference on Image Processing*, 2022.

[33] I. P. Singh, E. Ghorbel, O. Oyedotun, and D. Aouada, “Multi-label image classification using adaptive graph convolutional networks: From a single domain to multiple domains,” *arXiv preprint arXiv:2301.04494*, 2023.

[34] I. P. Singh, E. Ghorbel, A. Kacem, A. Rathinam, and D. Aouada, “Discriminator-free unsupervised domain adaptation for multi-label image classification,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 3936–3945.

[35] I. P. Singh, E. Ghorbel, A. Kacem, and D. Aouada, “Domain adaptation for multi-label image classification: A discriminator-free approach,” *arXiv preprint arXiv*, 2024.

[36] S. Rui, W. Tianxing, and L. Ziwei, “Detecting and recovering sequential deepfake manipulation,” in *In Proceedings of European Conference on Computer Vision*, 2022, pp. 712–728.

[37] I. P. Singh, N. Mejri, V. D. Nguyen, E. Ghorbel, and D. Aouada, “Multi-label deepfake classification,” in *2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, 2023, pp. 1–5.

[38] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.

[39] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, “Multi-label image recognition with graph convolutional networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5177–5186.

[40] Q. Li, X. Peng, Y. Qiao, and Q. Peng, “Learning category correlations for multi-label image recognition with graph networks,” *arXiv preprint arXiv:1909.13005*, 2019.

[41] Y. Wang, Y. Xie, Y. Liu, K. Zhou, and X. Li, “Fast graph convolution network based multi-label image recognition via cross-modal fusion,” in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 1575–1584.

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

[43] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, “A kernel method for the two-sample-problem,” *Advances in neural information processing systems*, vol. 19, 2006.

[44] M. Long, Y. Cao, J. Wang, and M. Jordan, “Learning transferable features with deep adaptation networks,” in *International conference on machine learning*, PMLR, 2015, pp. 97–105.

[45] M. Long, H. Zhu, J. Wang, and M. I. Jordan, “July. deep transfer learning with joint adaptation networks,” in *In International conference on machine learning*, PMLR, 2017, pp. 2208–2217.

[46] B. Sun, J. Feng, and K. Saenko, “Return of frustratingly easy domain adaptation,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.

[47] D. Lin, J. Lin, L. Zhao, Z. J. Wang, and Z. Chen, *Multilabel Aerial Image Classification With Unsupervised Domain Adaptation*. IEEE Transactions on Geoscience and Remote Sensing, 2021.

[48] 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 the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5513–5522.

[49] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2285–2294. DOI: 10.1109/CVPR.2016.251.

[50] M. Sandler, J. Baccash, A. Zhmoginov, and A. Howard, "Non-discriminative data or weak model? on the relative importance of data and model resolution," Oct. 2019, pp. 1036–1044. DOI: 10.1109/ICCVW.2019.00133.

[51] J. Lee, T. Won, T. K. Lee, H. Lee, G. Gu, and K. Hong, *Compounding the performance improvements of assembled techniques in a convolutional neural network*, 2020. arXiv: 2001.06268 [cs.CV].

[52] S. Rota Bulò, L. Porzi, and P. Kontschieder, "In-place activated batchnorm for memory-optimized training of dnns," Jun. 2018, pp. 5639–5647. DOI: 10.1109/CVPR.2018.00591.

[53] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141. DOI: 10.1109/CVPR.2018.00745.

[54] R. Krishna, Y. Zhu, O. Groth, *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, May 2017. DOI: 10.1007/s11263-016-0981-7.

[55] T. Chen, M. Xu, X. Hui, H. Wu, and L. Lin, "Learning semantic-specific graph representation for multi-label image recognition," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 522–531. DOI: 10.1109/ICCV.2019.00061.

[56] T. Chen, L. Lin, X. Hui, R. Chen, and H. Wu, "Knowledge-guided multi-label few-shot learning for general image recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020. DOI: 10.1109/TPAMI.2020.3025814.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015, pp. 1–15.

[58] Y. Wang, D. He, F. Li, *et al.*, "April," in *In Proceedings of the AAAI Conference on Artificial Intelligence*, No. 07: Multi-label classification with label graph superimposing. (Vol. 34, 2020, pp. 12 265–12 272.

[59] T. Ridnik, G. Sharir, A. Ben-Cohen, E. Ben-Baruch, and A. Noy, "Ml-decoder: Scalable and versatile classification head," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 32–41.

[60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[61] 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.

[62] Y. Wang, Y. Xie, L. Fan, and G. Hu, "Stmg: Swin transformer for multi-label image recognition with graph convolution network," *Neural Computing and Applications*, vol. 34, no. 12, pp. 10 051–10 063, 2022.

[63] Y. Wang, Y. Xie, J. Zeng, H. Wang, L. Fan, and Y. Song, "Cross-modal fusion for multi-label image classification with attention mechanism," *Computers and Electrical Engineering*, vol. 101, p. 108 002, 2022.

[64] D. Sun, L. Ma, Z. Ding, and B. Luo, "An attention-driven multi-label image classification with semantic embedding and graph convolutional networks," *Cognitive Computation*, pp. 1–12, 2022.

[65] W. Jin, T. Derr, Y. Wang, Y. Ma, Z. Liu, and J. Tang, "March. node similarity preserving graph convolutional networks," in *In Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, pp. 148–156.

[66] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *stat*, vol. 1050, 2017.

[67] S. Razavian, A. A., and S. H., "J," in *In Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, and Carlsson, S. CNN features off-the-shelf: an astounding baseline for recognition, 2014, pp. 806–813.

[68] Y. Wei, W. Xia, M. Lin, *et al.*, "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.

[69] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, *Overfeat: Integrated recognition, localization and detection using convolutional networks*. In ICLR, 2014.

[70] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[71] A. Krizhevsky, I. Sutskever, and G. Hinton, ""imagenet classification with deep convolutional neural networks,"," *in Proc. Neural Inf. Process. Syst*, vol. 1106, 2012.

[72] X. Qu, H. Che, J. Huang, L. Xu, and X. Zheng, "Multi-layered semantic representation network for multi-label image classification," *International Journal of Machine Learning and Cybernetics*, pp. 1–9, 2023.

[73] M. Li, Y. M. Zhai, Y. W. Luo, P. F. Ge, and C. X. Ren, "Enhanced transport distance for unsupervised domain adaptation," in *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 936–13 944.

[74] Y. Zhang, T. Liu, M. Long, and M. Jordan, "May. bridging theory and algorithm for domain adaptation," in *In International Conference on Machine Learning*, PMLR, 2019, pp. 7404–7413.

[75] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," *Advances in neural information processing systems*, vol. 31, 2018.

[76] G. Li, Z. Ji, Y. Chang, S. Li, X. Qu, and D. Cao, "Ml-anet: A transfer learning approach using adaptation network for multi-label image classification in autonomous driving," *Chinese Journal of Mechanical Engineering*, vol. 34, no. 1, pp. 1–11, 2021.

[77] D. D. Pham, S. M. Koesnadi, G. Dovletov, and J. (Pauli, "April)," in *In IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, U. Adversarial, Ed., Domain Adaptation for Multi-Label Classification of Chest X-Ray, 2021, pp. 1236–1240.

[78] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint*, 2014. arXiv: `1411.1784`.

[79] L. Chen, H. Chen, Z. Wei, *et al.*, “Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7181–7190.

[80] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2009.

[81] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *In Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[82] A. Radford, J. W. Kim, C. Hallacy, *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[83] X. Sun, P. Hu, and K. Saenko, “Dualcoop: Fast adaptation to multi-label recognition with limited annotations,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 30 569–30 582, 2022.

[84] Y. Hua, L. Mou, and X. X. Zhu, “Relation network for multilabel aerial image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4558–4572, 2020.

[85] B. Chaudhuri, B. Demir, S. Chaudhuri, and L. Bruzzone, ““multilabel remote sensing image retrieval using a semisupervised graph-theoretic method,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 56, no. 2,” *pp*, vol. 1144, 2018.

[86] Y. Hua, L. Mou, and X. X. Zhu, ""recurrently exploring class-wise attention in a hybrid convolutional and bidirectional lstm network for multi-label aerial image classification,"," *ISPRS J. Photogramm. Remote Sens*, vol. 149, 2019.

[87] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa, "Cross-domain weakly-supervised object detection through progressive domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5001–5009.

[88] G. S. Xia, J. Hu, F. Hu, *et al.*, "Aid: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, 2017.

[89] Y. Yang and S. Newsam, ""bag-of-visual-words and spatial extensions for land-use classification," in *" in International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, 2010.

[90] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[91] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3723–3732.

[92] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[93] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "Fcns in the wild: Pixel-level adversarial and constraint-based adaptation," *arXiv preprint arXiv:1612.02649*, 2016.

[94] Y. Chen, W. Li, and L. Van Gool, “Road: Reality oriented adaptation for semantic segmentation of urban scenes,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7892–7901.

[95] Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei, “Fully convolutional adaptation networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6810–6818.

[96] Y. Li, L. Yuan, and N. Vasconcelos, “Bidirectional learning for domain adaptation of semantic segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6936–6945.

[97] D. Dowson and B. Landau, “The fréchet distance between multivariate normal distributions,” *Journal of multivariate analysis*, vol. 12, no. 3, pp. 450–455, 1982.

[98] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *International conference on machine learning*, PMLR, 2017, pp. 214–223.

[99] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz, “Deepgmr: Learning latent gaussian mixture models for registration,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, Springer, 2020, pp. 733–750.

[100] M. Cordts, M. Omran, S. Ramos, *et al.*, “The cityscapes dataset for semantic urban scene understanding,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[101] C. Sakaridis, D. Dai, and L. Van Gool, “Semantic foggy scene understanding with synthetic data,” *International Journal of Computer Vision*, vol. 126, pp. 973–992, 2018.

[102] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in *In Proceedings of IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1–11.

[103] L. Li, J. Bao, T. Zhang, *et al.*, "Face x-ray for more general face forgery detection," in *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5001–5010.

[104] N. Mejri, E. Ghorbel, and D. Aouada, "Untag: Learning generic features for unsupervised type-agnostic deepfake detection," in *In IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023, pp. 1–5.

[105] N. Mejri, K. Papadopoulos, and D. Aouada, "Leveraging high-frequency components for deepfake detection," *In 2021 IEEE 23rd International Workshop on Multimedia Signal Processing*, pp. 1–6, 2021.

[106] K. Shiohara and T. Yamasaki, "Detecting deepfakes with self-blended images," in *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 720–18 729.

[107] L. Chen, Y. Zhang, L. Song Y. Liu, and J. Wang, "Self-supervised learning of adversarial examples: Towards good generalizations for deepfake detections," in *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[108] T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong, and W. Xia, "Learning self-consistency for deepfake detection," in *In Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 023–15 033.

[109] Y. Wang, K. Yu, C. Chen, X. Hu, and S. Peng, "Dynamic graph learning with content-guided spatial-frequency relation reasoning for deepfake detection," in *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7278–7287.