Republic Of Algeria Democratic And People's
Ministry Of Higher Education And Scientific Research

University of KASDI Merbah - Ouargla
Faculty of New Technologies of Information and Communication
Department of Computer Science and Information Technologies

Master Academic Thesis

Domain: Computer Science
Specialty: Fundamental computer science

Presented by :
BEDDA Hanane

Theme :

Deep Visual-Semantic embedding for Multi-label image Classification

---

Presented on 22/10/2020 in front of the jury composed of :

Dr. Kherfi Mohammed Lamine	President	University UKM Ouargla
Dr. Debbagh Farah	Supervisor	University UKM Ouargla
Dr. Bouanane Khadra	Examiner	University UKM Ouargla

Academic year : 2019/2020

Acknowledgments

We thank ALLAH first for being able to complete this thesis. My big thanks to my supervisor Ms. Debbagh Farah for her great guidance, advice and support to do this work in all the year. My thanks for professor Kherfi Mohammed Lamine and professor Bouchachia Abd el Hamid for their supervision and trusting us. My thanks to Ms. Bouanane Khadra for accepting the evaluation of the work. I would like to thank my parents for all the support and love they have given me during my studies. I also thank my sister, my friends and my family, for their support during this year.

Abstract

The classification is one of Machine Learning techniques, that aims to categorize data into one or more predefined classes (labels). When the data is a set of images, we talk about image classification, which done basing on their visual content. The image classification can be categorized into two categories which are single-label classification and multi-label classification.

The multi-label image classification (MLC) aims to firstly learn from training set of images, where each one can belong to multiple classes and so after be able to predict more than one class label simultaneously for a new tested image.

In this thesis, we present a multi-label image classification method that contains three modules: word embedding module, visual embedding module and transformation module. The first module consist of a word embedding model that maps words (labels) into a semantic embedding space of d-dimension, where each semantic related labels are close to each other. The second one, is a CNN framework that learn a transformation matrix A with dimensions $k \times d$ from an input image $\mathcal{I}$ and the embedding vectors of its corresponding labels. The last module receive results of the two previous modules, to transform labels from d-dimensional space to a k-dimensional visual-semantic embedding space, which separated the relevant and irrelevant labels to the image $\mathcal{I}$.

Keywords: classification, multi-label images, deep learning, word embedding, visual-semantic embedding.

Résumé

La classification est l'une des techniques d'apprentissage automatique, qui vise à catégoriser les données dans une ou plusieurs classes prédéfinies. Lorsque les données sont un ensemble d'images, on parle de classification d'images, qui était basé sur leurs contenu visuel. La classification d'images peut être organisé en deux catégories qui sont la classification à unique-label et la classification à multi-label.

La classification d'images multi-label (MLC) vise d'abord à apprendre à partir d'un ensemble d'images d'apprentissages, où chacune peut appartenir à plusieurs classes, et ainsi être capable de prédire plus d'une classe simultanément pour une nouvelle image de test.

Dans cette thèse, nous présentons une méthode de classification d'images multi-label qui contient trois modules: module de plongement de mots, module de plongement visuelle et module de transformation. Le premier module consiste en un modèle de plongement de mots qui mappe les mots (labels) dans un espace de plongement sémantique de d-dimensions, où les labels qui possèdent une correlation sémantique sont proches l'une de l'autre. Le deuxième est une architecture CNN qui apprend une matrice de transformation A avec de dimension $k \times d$ à partir d'une image d'entrée $\mathcal{I}$ et les vecteurs de plongement des labels correspondantes. Le dernier module reçoit les résultats des deux modules précédents, pour transformer les labels de l'espace d-dimensions en un espace de plongement visuel-sémantique k-dimensions, ce nouvel espace permet de séparer les labels pertinentes de celles non pertinentes de l'image $\mathcal{I}$.

Mot-clé: classification, images multi-labels, l'apprentissage profond, plongement visuel-sémantique.

ملخص

التصنيف هو تقنية من تقنيات التعلم الآلي، والذي يهدف إلى تصنيف البيانات إلى فئة أو أكثر (محددة مسبقا). عندما تكون البيانات عبارة عن مجموعة من الصور، فإننا نتحدث عن تصنيف الصور الذي يتم على أساس تحديد المحتوى المرئي. يمكن لتصنيف الصور ان يصنف ضمن فئتيين وهما التصنيف أحادي الفئة والتصنيف متعدد الفئات للصور.

التصنيف متعدد الفئات للصور يهدف أولا إلى التعلم من مجموعة بيانات الصور، أين يمكن لكل صورة الإنتماء لعدة فئات، ليتمكن بعد ذلك من التنبؤ بأكثر من فئة واحدة في الوقت ذاته لصورة اختبار جديدة.

قدمنا في هذة الأطروحة طريقة لتصنيف الصور متعددة الفئات، والذي يحتوي على ثلاث وحدات: وحدة تضمين الكلمات، وحدة التضمين المرئي ووحدة التحويل. الوحدة الأولى تتكون من نموذج تضمين الكلمات الذي يعين الكلمات (الفئات) في مساحة تضمين دلالية من d أبعاد، أين كل مجوعة من الفئات ذات صلة دلالية تكون قريبة من بعضها البعض. الوحدة الثانية، عبارة عن هيكل CNN التي تتعلم مصفوفة تحويل A ذات الأبعاد $k \times d$ من الصورة المدخلة $\mathcal{I}$ والأشعة المضمنة للفئات المقابلة. الوحدة الأخيرة تستقبل نتائج الوحدتين السابقتين، لتحويل الفئات من مساحة ذات d أبعاد إلى مساحة تضمين مرئي دلالي ذات k أبعاد، والتي تفصل بين الفئات ذات الصلة والفئات التي ليس لها صلة بالصورة $\mathcal{I}$.

الكلمات المفتاحية:

التصنيف، الصور متعددة الفئات، التعلم العميق، تضمين الكلمات، تضمين مرئي دلالي.

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

Chapter 1

General Introduction

1 Introduction

The artificial intelligence (AI) is a study about the human brain performance, that produce intelligent systems. Among the objectives of this study: knowledge representation, planing, learning and reasoning, ...etc. Intelligent learning systems have the ability to learn and adapt to each changes. Thus, it makes possible to dispense of expecting solutions for all possible situations, this is what called 'machine learning'.

Machine learning (ML) is a part of AI that programming computer algorithms to improve their performance through example-data or experience. These algorithms build models from data to learn from and make predictions on it. This data comes from multiple datasets, which are used in two phases in machine learning. First one is training, where the model learn on training dataset to fit the parameters and create the right output. The second one is testing (prediction), where the model predicts output for new data (test data) which allow to evaluate the model performance(i.e. generalization). In fact, machine learning is used to give solutions to several important tasks such as classification.

Classification is a predictive task , where specific class labels are assigned for a given example of input data (text, audio or image). Classification methods divided into two categories according to the output. First one is the single-label classification, where one class label is assigned for the input data, it includes binary classification and multi-class classification. The second one is the multi-label classification, it is more complex than the previous category, because it should assign more than one label per one example of input data.

This thesis focuses on multi-label classification applied on images as a data.

There are many applications where assigning multiple classes to an image is necessary according to its complexity. So, classify such image into one class (category) is not effective. This is what the multi-label classification came to solve it.

2 Problematic

Comparing to the single-label image classification, multi-label image classification (MLC) is considering as a challenging task, because of the complexity of images and labels information. In fact, to multi-label images a big panoply of literature works were presented. They considered the relation image-label, and based on the transformations or adaptations of the single-label image classification methods to deal with multi-label. Hence, in order to simplify the challenge of MLC task, the most of them consider a total independence among labels as an hypothesis within the given solutions. However, this contradict the reality, where a single image can contain two or more objects (labels) that share a specific semantic context. In other words, it exists a certain semantic correlation between subsets of labels. Therefore, the label-correlation is a very important[23],[24], [22],[36],[35], [59] information that must be considered in the task of multi-label image classification, and it presents a fundamental aspect if we want to let the machine predict labels for images as human do.

The correlation among labels let the MLC become more challenging task: How to model the label correlation? How to integrate and explore this information within the MLC ?

3 Motivation

As we presented in the previous section, despite the huge amount of works on multi-label image classification, there is still a required effort that must be done to consider the very important aspect of the semantic correlation among labels within the classification task. This motivate us to explore this aspect to look for a MLC solution considering label correlation.

4 Contributions

Since the label correlation among labels is an important aspect that must be considered to improve the multi-label image classification results, in our thesis we aims to present

solution to MLC by considering the label correlation. To this end, our main contributions are:

- To consider the semantic correlation among labels, we profit from the advantages and services of the word embedding domain. So, we explored three deep word embedding models. As a result, the semantic correlation between labels is presented and captured in a numerical space. This permit its exploitation by the machine.
- To consider the relations image-labels and label-label, we learn a joint visual-semantic representation deep model between the two modalities: images and labels.
- Instead of using multiple classifier for the multiple labels, we use one classifier. That reduce considerably the time and the complexity of the solution comparing to the other ones.

5 Thesis structure

In addition to the currant chapter about a general introduction, this thesis is organized as followed:

- In chapter 2, we introduce the classification task, image classification and its categories. Then, we focus on the multi-label classification. We give its informal and formal definition, domain applications and its traditional methods. After that, we present the main challenges within multi-label classification (MLC). Finally, we review some exploited domains to deal with multi-label image classification challenges.
- In chapter 3, we present a state of the art of related works on multi -label image classification that consider label correlations. They are from two big categories. The first category use traditional methods, and the second one use deep learning methods.
- In chapter 4, we will present a deep learning solution for multi-label image classification considering the label correlation. Firstly, we will give a general description of this solution. After that, we will present its three modules in detail.
- In chapter 5, we start by presenting the experimental settings: the used text corpora for word embedding, the used image dataset and the performance measurements.

Then, we give the experiments of the solution steps and we present a comparison between word embedding models. After that, we present the obtained results. Finally, we analysis and discussion them.

Chapter 2

Multi-label image classification

Chapter 2

Multi-label image classification

1 Introduction

The classification is one of Machine Learning techniques (tasks), used to classify data (text, image, voice or music) into one or more categories, under the goal of facilitate the study of a large number of data.

Image classification represent a big challenge in computer vision applications. Therefore, it is required in several applications of computer science under different fields in the world. As a consequence, different methods was applied to classify images, which categorized into tow big sets. First one is the single label classification, and this category have two sub set: binary classification and multi-class classification. And second one is the multi-label classification.

In our thesis we focus on (MLC). However, before presenting the (MLC), we give the principle of each image classification category, in order to distinguish between them..

2 Image Classification

2.1 Definition

The classification is a predictive task in machine learning. Usually, it is done by supervised learning techniques. The goal behind classification is to train a model from labeled data. After that, it can predict the label(s) (or class) for a new unlabeled data [31]. The classification deals with multiple forms of data, such as: text, audio, video and images.

Image classification is an action of categorizing an image into one or more predefined classes (labels), according to its visual content. [49]

2.2 Image classification categories

The image classification is a predictive topic, aims to learn a model in generally from labeled images to predict the label or class of a new image. The attributes of the classification divided into two subsets. The first one contains the input features and the variables of prediction. The second subset holds the output attributes and the class of each instance. So, depending on the nature of outputs in the second subset, several types of classification can be identified. [31]

We can define two important categories of image classification that cover all the varieties of image classification problems[27]. However, those categories are not limited for images, but they can have any form of input data.

1. **Single-Label classification**

   In this category, the input data have one label in the output. This category is divided into two sub categories:

   (a) **Binary-label classification**

   Binary-label classification aims to classify the input data (instances) of a given set into two groups on the basis of classification rules. An instance has only one output label, and it can take two different values 'yes or no', 'positive or negative' or '1 or 0', ... [27]. For example (Figure2.1), the output value of the cat image is 1 (positive value), while the output value of the dog image is 0 (negative value), for the same classifier.

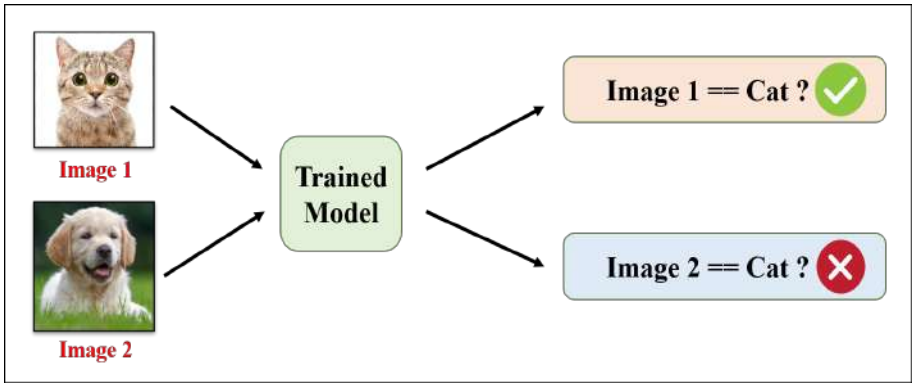

Figure 2.1: Simple example of binary image classification [6]

(b) **Multi-class classification**

Multi-class classification can be seen as a generalization of binary classification. Just as binary classification involves predicting if image is from one of a two classes (positive or negative) (Figure2.2(a)), the input image in multi-class classification can be categorized exactly into one label from a certain predefined label set (Figure2.2(b)) [27]. Therefore, the output of the classification is a vector with one positive value (corresponding to the image label), and zeros otherwise.

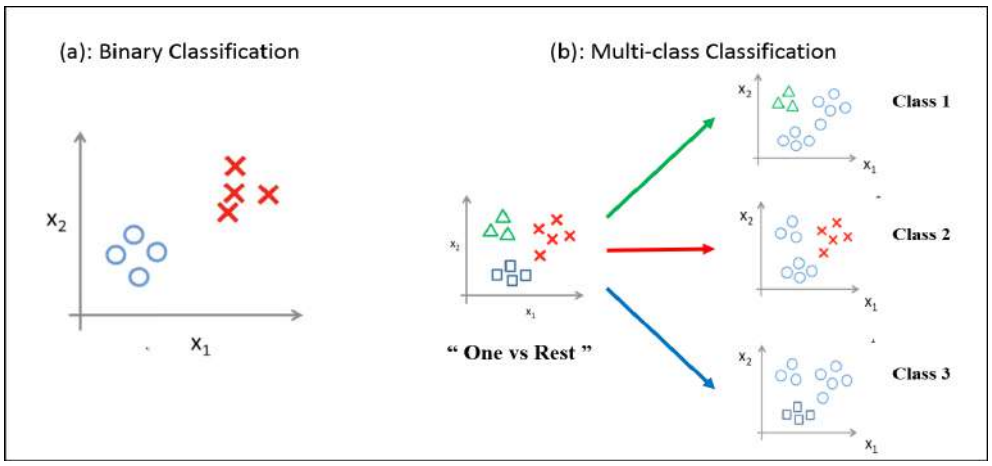

Figure 2.2: Difference between binary classification and multi class classification [7]

2. **Multi-Label classification (MLC)**

Multi-Label classification is more complex. The input data (and for us the input image) can simultaneously associated with more than one class [27].

The result is a set of labels as shown in (Figure2.3,(b)) unlike single label classification (Figure2.3,(a)).

We will present with more details the Multi-label image classification in the coming section.

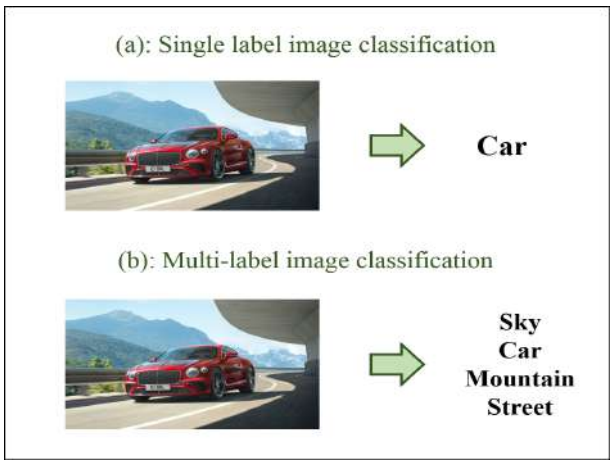

Figure 2.3: The difference between single label and multi-label image classification

We also mention other types of classification :

- **Multidimensional Classification:** works in the same way of multi-label classification, the input date can be associated to more than one label simultaneously. Except that the output vector is not limited to binary values, it contains any values form a predefined set [31].
- **Multiple Instance Learning:** is learning paradigm represent each example by groups of input data (instances) that called bags. The output label belong to the bag instead of one instance [31].

The table below presents the different kinds of classification :

Classification kind	Output type	Nember of outputs
Binary	Binary	1 per instance
Multiclass	Multivalued	1 per instance
Multilabel	Binary	n per instance
Multidimensional	Multivalued	n per instance
Multiinstance	Binary / Multivalued	1 per n instances

Table 2.1: Classification types attending to the output to be predicted [31]

3 Multi-label Image Classification

3.1 Definition

Multi-label classification (MLC) of images is a task that aims to recognize the different objects or labels in images. It is more complicated than the single label classification, because it focuses on discovering more than one label per image. [56]
The multi-label classifier return a vector of output values, unlike the single label classifier which return one value [31]. As shown in (Figure2.4)

(a) Single label Classification

Binary Classification

Instance	Class A
1	1
2	0
3	0
4	1
5	0

Multi-class Classification

Instance	Classes [A,B,C,D,E,F]
1	[1,0,0,0,0,0]
2	[0,0,1,0,0,0]
3	[0,0,0,0,0,1]
4	[0,0,0,0,1,0]
5	[0,1,0,0,0,0]

(b) Multi-label Classification

Instance	Classes [A,B,C,D,E,F]
1	[1,0,0,0,1,0]
2	[0,0,1,0,0,0]
3	[1,1,0,0,0,1]
4	[0,0,0,1,1,0]
5	[0,1,1,1,0,0]

Figure 2.4: Difference between single label and multi-label classification output

Formal definition [31][61] **:**

The MLC task consists to learn a function $\mathcal{H} : \mathcal{X} \longrightarrow 2^q$, where:

- $\mathcal{X} = \mathbb{R}^d$ is the d-dimensional visual feature space of images.
- $\mathcal{Y} = \{y_1, y_2, \ldots, y_q\}$ is the label space with q possible class labels.
- An image $\mathcal{I}$ has two feature vectors:
  1. d-dimensional visual vector $x_i \in \mathcal{X}$ / $x_i = \{x_{i1}, x_{i2}, \ldots, x_{id}\}$.
  2. The output label vector $Y_i \subseteq \mathcal{Y}$.
- To learn the function $\mathcal{H}$ we use a training set $\mathcal{D} = (x_i, Y_i)_{i \in [1,m]}$ of m labeled images.
- For each instance x of $\mathcal{X}$, the multi-label classifier predicts $\mathcal{H}(\mathrm{x}) \subseteq Y$ as the set of proper labels for x.

3.2 Applications domains

- "*Movie Genre Detection from a Movie Poster*" developed by (K.Kundalia et al.,2020). They used the movies posters as input images to predict the movie genres. Also, they created a large dataset on this subject. [37]
- Genetics/Biology : example, analyzing protein properties and gene expression.
- Medical image analysis to diagnose multiple diseases in the same organ of the human body. For example:
  - (Chen.H et al.,2019), proposed a deep Hierarchical Multi-Label Classification (HMLC) to facilitate the Computer-Aided Diagnosis (CAD) for the Chest X-rays (CXRs). [19]
- Social media domain example:
  - Recently, (Lui.L et al., 2020) developed a multi-label convolutional neural network model (BrandImageNet), to predict perceptual brands in the consumers images on social media. [39]

3.3 Multi-label Classification Methods

To solve MLC problems in general and image MLC problems especially, an explosive number of methods is presented in the literature, early existing and modern ones. In the following section we take a look to the traditional methods. Also, we will present the other methods from the state of art in the next chapter.

Traditional methods

They are considered as the first methods used to solve multi-label image classification problems. In the following paragraph we will explain briefly the three categories of these methods, which are shown in the (Figure2.5) :

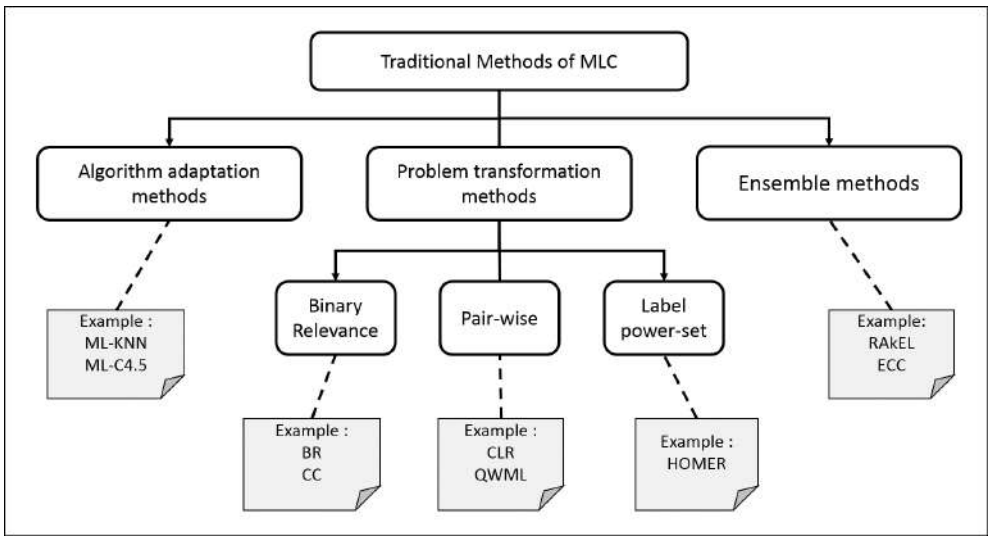

Figure 2.5: Traditional methods of Multi-label Classification [41]

1. **Algorithm adaptation Methods :**

   This category consists to adapt the existing algorithms of binary or multi class classification to directly perform the multi-label classification, like : KNN (k-nearest neighbors) algorithm, the C4.5 algorithm,... , that adapted and became : ML-KNN, ML-C4.5 ...[41]

2. **Problem transformation Methods :**

   Problem transformation methods assign the MLC task or the multi-label problem into one or more single-label classification tasks or single-label problems. One way of doing this is by training a separate classifiers, one for each label. Then the results are transformed into multi-label predictions.

   Under this category it exists a huge number of methods that can be divided into tree sub categories[41][57]:

   - **Binary relevance methods:** Binary Relevance is a simple and efficient approach commonly used in real-world multi-label learning applications. It depends on dividing the multi-label classification problems into a binary classification problems, where it deal with each label as a separate binary classification.
   - **Pair-wise methods:** it consists to learn one classifier for each pair of classes, thus increasing the overall number of classifiers to train to k(k - 1)/2 (from k labels).[31]
   - **Label power-set methods:** considers each unique set of labels in a multi-

label training set, as one of the classes of a new single-label classifier, which outputs the most probable class (which represents a set of labels).[53]

3. **Ensemble Methods :**

   Ensemble methods is an incorporate of multiple approaches (Algorithm adaptation or problem transformation methods). We mention here among the most problem transformation ensembles:

   - **RAkEL system:** (Random k-Labelsets) construct an ensemble of label powerest classifiers learned a small random subset of size k of labels, proposed by G.Tsoumakas, I.Katakis and I.Vlahavas in 2011, it based on dividing the initial set of labels into a number of small random subsets (labelsets).[53]
   - **ECC:** (Ensembles of classifier chain) methods have classifier chains as base classifiers (Classifier chain CC: method that related to the Binary Relevance method where classifiers are linked along a chain, proposed by J.Read in 2009).[41][57]

3.4 Multi-label image classification challenges

Multi-label image classification (MLC) is considered as a challenging task, because of the complex nature of images (image can contain multiple visual objects)[25][14], and the shared semantic among labels[23],[24], [22],[36],[35], [59]. In fact, the majority of traditional methods assumed a complete independence between the labels. In other words, they ignored the semantic correlation between labels. This contradict the reality,where the most of images contain a set of related concepts. That means there is a semantic correlation between subsets of labels,which helps to improve the classification performance, but makes the image MLC a more challenging task.

How to model the label correlation?
How to integrate and explore this information within the MLC?

3.5 Some exploited domains for multi-label image classification challenges

Since the complexity of the multi-label image classification task comparing to the single-label one, and the shared semantic (correlation) among subsets of labels, it is interesting

to exploit the services of existing domains. Especially, Deep learning and embeddings. In one hand, the deep learning permit to deeply learn from complex input data, as complex images. On the other hand, the embeddings permit to obtain numerical representation of input images and permit to present the semantic information behind labels in a numerical format as well. Hence, these numerical results can be exploited by the machine to solve the problem in a best way.
In the fallowing sections we present these two domains:

Deep learning and Convolutional neural network :

Deep learning (DL) is one of the main machine learning techniques. The term *deep* refers to neural networks that have multiple hidden layers (layers between the input and output layers). The depth of the networks allows them to learn more and more complex representations, therefore give predictions that are more accurate.

Deep learning has achieved great success in several applications such as image classification and natural language processing *"even sometimes outperforming humans in certain aspects"*[42]. Deep learning architectures considering as a big challenge in analyzing big data[47]. In fact, DNN deals specifically with different architectures of neural networks as Convolutional Neural Network (CNNs), Auto-encoders(AE), Variational auto-encoder(VAE), generative adversial networks GANs,...etc.

In our thesis we are interested on the CNN architecture, that shown its performance in large applications of image processing, even in natural language processing. Especially, it is one of the mostly used architecture in image classification.
The term convolutional in CNN comes from applying mathematical convolution operations. CNNs have the ability to learn and extract the important features that describes the input data. Thus, the representation of the input data given by the CNN makes the neural network able to discriminate and classify the data more precisely. For more detail on CNNs architecture, the reader can refer to [47].

Embedding and Neural network embeddings :

An embedding task aims to map a discrete categorical input data to a numerical representation [60], a vector of real numbers, in order to be used as an input to processing by machine learning algorithms. In embeddings, the high-dimensional vectors can be translated to a low-dimensional space (embedding space), where inputs which are similar are

placed close to each other [5].

In the context of neural networks, embeddings are low-dimensional, learned continuous vector representations of discrete data. Neural network embeddings are useful because they can reduce the dimensionality of categorical data and meaningfully represent categories in the transformed space.
Neural network embeddings have 3 primary purposes:

- Finding nearest neighbors in the embedding space. These can be used to make recommendations based on user interests or cluster categories.
- As input to a machine learning model for a supervised task.
- For visualization of concepts and relations between categories. This requires a further dimensionality reduction technique to get 2 or 3 dimensions. The most popular techniques for reduction are their-self an embedding methods: t-Distributed Stochastic Neighbor Embedding (TSNE) [40] and Uniform Manifold Approximation and Projection (UMAP) [43][16].

In our thesis we are interested by the two last purposes. Therefore, we will use two differences modalities of data to be embedded: images and words. For that, we will present in the following two sections the word and image embedding:

1. **Word embeddings (language representations):** word embeddings are natural language processing (NLP) techniques, where words from the vocabulary are represented (or mapped to) by vectors of real numbers in a predefined vectors space as shown in (Figure2.6).

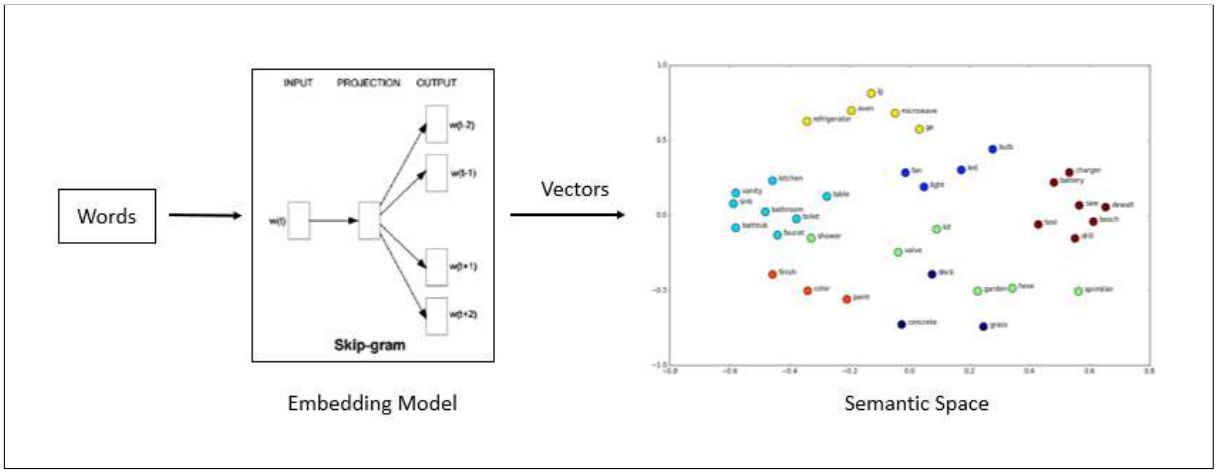

Figure 2.6: Word Embedding Task

There are several word embedding methods like :**Word2Vec**, **GloVe** and **FastText**:

(a) **Word2Vec:** is an algorithm for learning word embedding from a text corpus and produces a vector space, it is unsupervised model depend on the words context (distributional hypothesis) in natural language to provide labeled training data. In Word2Vec there is two kind of architectures, Skip-Gram or continuous bag-of-words (CBOW). Skip-gram takes a word as input and tries to predict its context and it represent rare words well. CBOW takes the context of a word as input, and tries to predict the word in question, it is the inverse model of Skip-Gram[29]. The main parameters of Word2Vec are: the *dimensionality* of vector space which is the dimension of the vectors that describe words (it is between 100 and 1000 in general). The second parameter is the *window size*, which the size and the number of terms in word context (the authors suggests size 10 with Skip-Gram and 5 with CBOW).[13]

(b) **GloVe:** the Global Vectors, is unsupervised algorithm aims to represent words and it is a count-based model [48]. *"Training is performed on aggregated global word-word co-occurrence statistics from a corpus"*. It begins by construct a large matrix of (word, context) pairs in the training corpus, rows represent words, columns represent contexts of one or more words, and the elements correspond to the number of times word co-occurs in the context. After that, GloVe factorizes the matrix into a pair of (word, feature) and (feature, context) matrices (Matrix Factorization).[29]

(c) **FastText:** is a word representation technique allows to users to learn word embeddings and text classifiers, created by Facebook's AI Research (FAIR) lab. FastText is fast and efficient technique, its training method ends up learning morphological details as well. For that, it can give word vectors for unknown or out of vocabulary words. FastText succeed to create word vectors representation for rare or unknown word, against Word2Vec and Glove.[12]

2. **Image embedding (visual representation):** Image embedding aims to represent an input image by a vector of real values, named visual vector [32], [33],[15], [34] or feature vector. the images are then represented in a space of visual characteristics.

   In fact, one of the popular deep learning image embedding models is the CNN. this deep neural network shows its power to learn visual representations for images.

Hence, the visual vector of an image can be obtained from the last layers of a CNN image classification model [60]. The (Figure2.7) shows a CNN architecture with its several layers.

Different from the previous purpose of using CNN, in our thesis we will use it for other objective. we will use it to learn a linear transformation matrix. More detail will be given in the chapter 4.

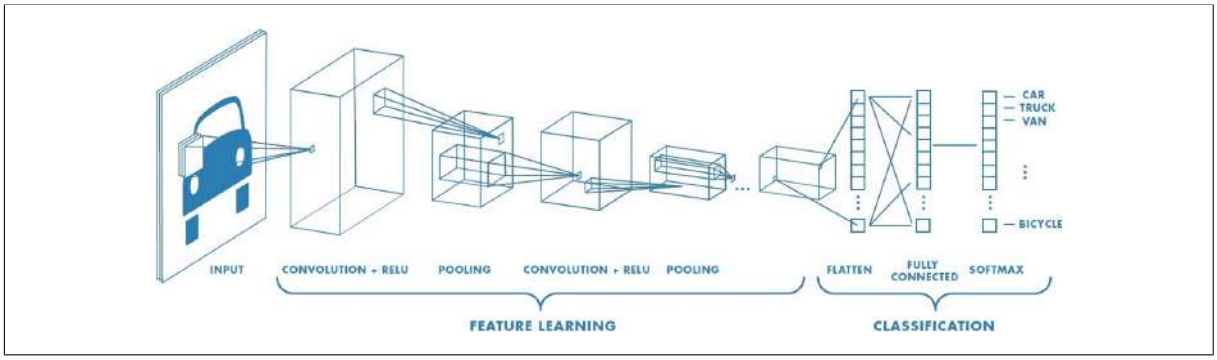

Figure 2.7: Convolutional Neural Network architecture [9]

3. **Image-Word embeddings:** or visual-semantic embeddings [60], that aim to combine the two modalities representations, and represent them in the same embedding space. (Figure2.8) shows an example of model embedded images and labels to the same space.

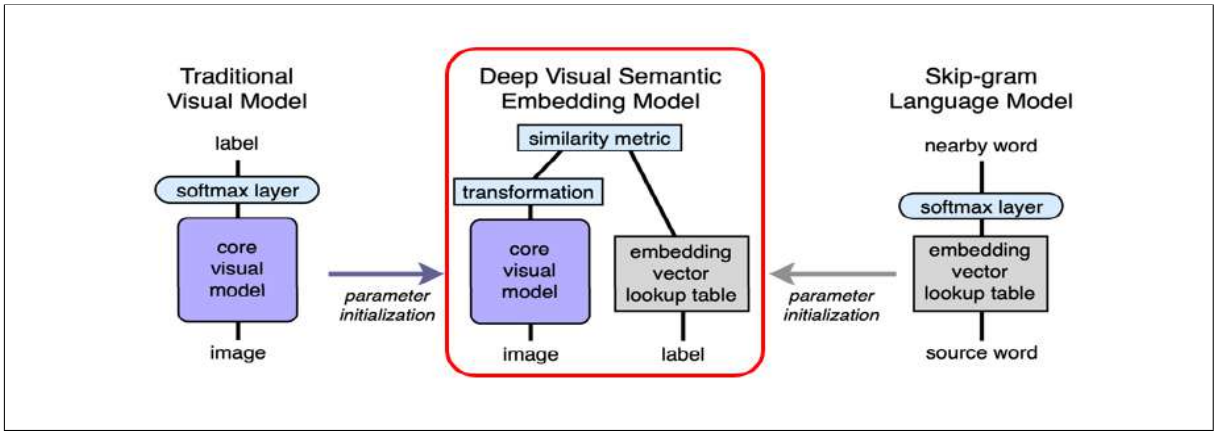

Figure 2.8: Visual-Semantic embedding model[28]

4 Conclusion

In this chapter we have introduced the classification, which is a machine learning technique. We have focused on image classification task, and explain in detail its categories: the single label classification with its subsets, and the multi-label classification. So, we gave more detail on the multi-label image classification (MLC) which is the subject of our study, and some MLC domain applications. An analysis of MLC traditional methods and their categories has been presented as well. After that, we concentrated on the challenges in the MLC task, and especially the label-correlation. Finally we presented specific domains that serve in resolving theses challenges.

At the end of this chapter we can say that to get more performance in MLC, we must take into account the labels correlation. For exploring this issue, in the coming chapter, we will present related works on multi-label image classification that consider label correlations.

Chapter 3

Multi-label image classification considering label-correlation: State of the art

Chapter 3

Multi-label image classification considering label-correlation: State of the art

1 Introduction

Unlike the single-label image classification, multi-label image classification (MLC) is considering as a challenging task, according to the complexity of images and labels information. Learning from multi-label data passed by several attempts, such as using a transformation or adaptation of the single-label image classification methods to deal with multi-label, as shown in the previous chapter. Therefore,there is a big interest on using deep learning methods because of its great performing in the classification tasks in general.

In fact, in order to simplify the challenge of this task, a big panoply of literature works consider a total independence among labels as an hypothesis within the given solutions. However, this contradict the reality, where it exists a certain semantic shared (correlation) between subsets of labels. Hence, the label-correlation is an important[23],[24], [22],[36],[35], [59] information that must be considered for improving the multi-label classification tasks. For that, other literature solutions take attention to the label-correlation, they can be classified into two big categories as shown in (Figure3.1). The first category gave solutions that learn the dependencies among labels using traditional methods. The second category learn the label correlations using deep learning methods.

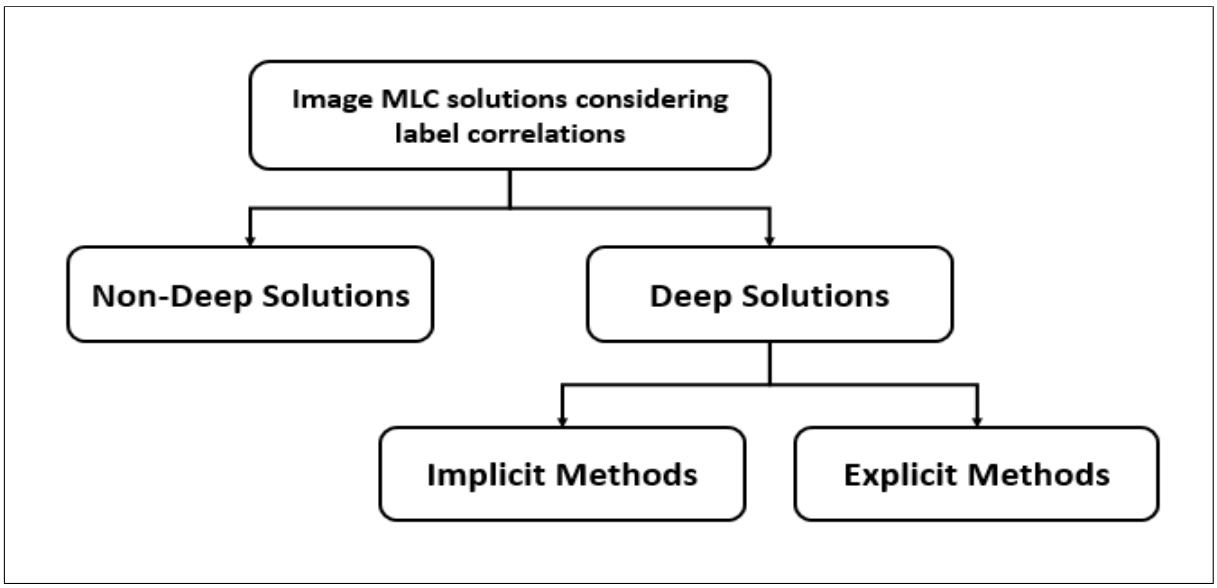

Figure 3.1: Classification of multi-label image classification methods considering label-correlation

We can categorize the second one into two sub-categories according to the label correlations learning. The first sub-category, learns the dependencies among labels implicitly, and the second one learns label correlations explicitly.
In the following sections we present these two categories.

2 Non-Deep solutions

This category of solutions takes into account the dependencies among labels to solve multi-label image classification problems, and this by expansion of traditional methods, without using deep learning tasks.
As an instance, (Read, Jesse, et al. 2011)[50] presented a novel binary relevance method, by using a chain of binary classifiers. To model the label correlations, the chained method can pass label information between classifiers [50]. The limitation of this method is that the complexity increase with large number of labels.

3 Deep learning solutions

We can classifier the recent solutions based on deep learning into two categories. Firstly, solutions that learn the dependencies among labels implicitly via attention mechanisms. Secondly, solutions that explicitly learn the label correlations by compound models with complex architectures.

3.1 Deep learning solutions considering label correlations implicitly

These solutions used deep learning methods and learn the local correlations among labels or a set of features per images implicitly

As an instance, (Li, Changsheng, et al. 2019) proposed a deep neural network (DNN) for multi-label image classification, based on REconstruction regularized Two-way Deep distance Metric (RETDM) learning. Original images and labels are embedded via a CNN and a DNN, respectively to a latent space. They present a two-way distance metric learning strategy, to capture the dependence of image features, the dependence of labels, and the correlations between images and labels on the embedded space.[38] The framework of their model is shown in (Figure3.2).

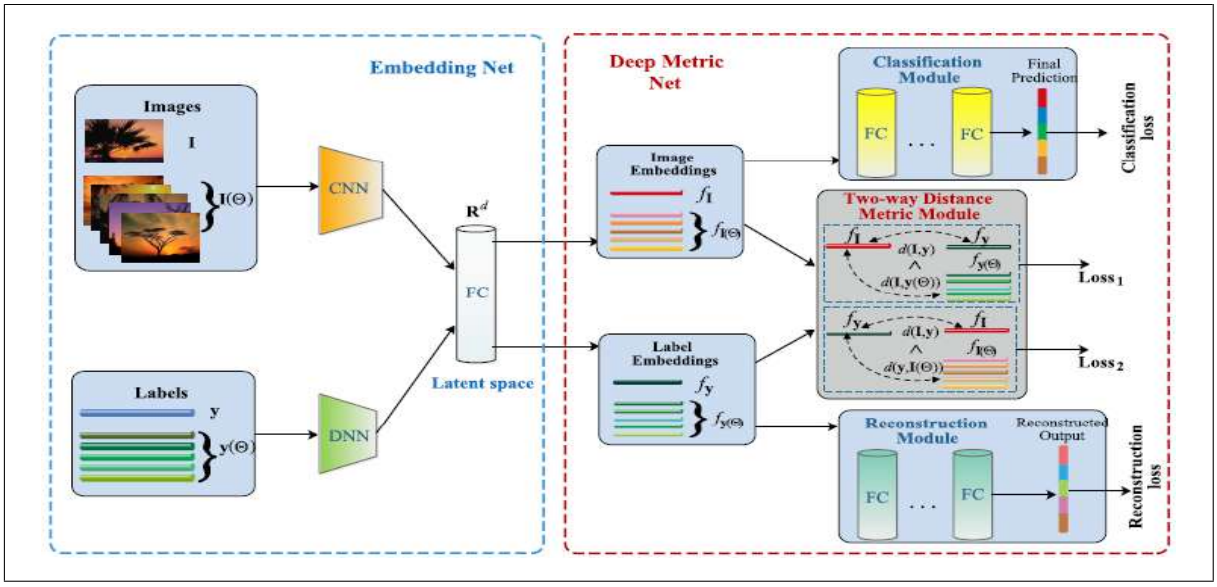

Figure 3.2: Overall framework of (Li, Changsheng, et al. 2019) model [38]

3.2 Deep learning solutions considering label correlations explicitly

Numerous literature solutions of multi-label image classification modeled the label dependency explicitly via deep models. After the great success of deep learning algorithms in several domains and with other methods too, those solutions take into account this success and use deep learning algorithms to achieve the goal of multi-label image classification. As instances:

– (Chen, Zhao-Min, et al. 2019) proposed a multi-label classification model based on

Graph Convolutional Network (GCN). To explicitly model the inter dependencies between labels by GCN, they designed a label correlation matrix (graph structure). They represent each node (label) of the graph as a word embedding of the corresponding label. [20]

– (Song, Lingyun, et al. 2018) [52], proposed a deep Multi-Modal CNN for Multi-Instance Multi-Label image classification (MMCNN-MIML). This model incorporates both images and textual context information for generating multi-modal instances. Also it groups labels in its later layers, to benefit from the label correlations. (Figure3.3) shows the architecture of (MMCNN-MIML), that consists of 4 modules. Module I, visual instance generation (VIG). Module II, group context generation (GCG). Module III, multi-modal instance generation (MMIG). Module IV, MIML image classification.

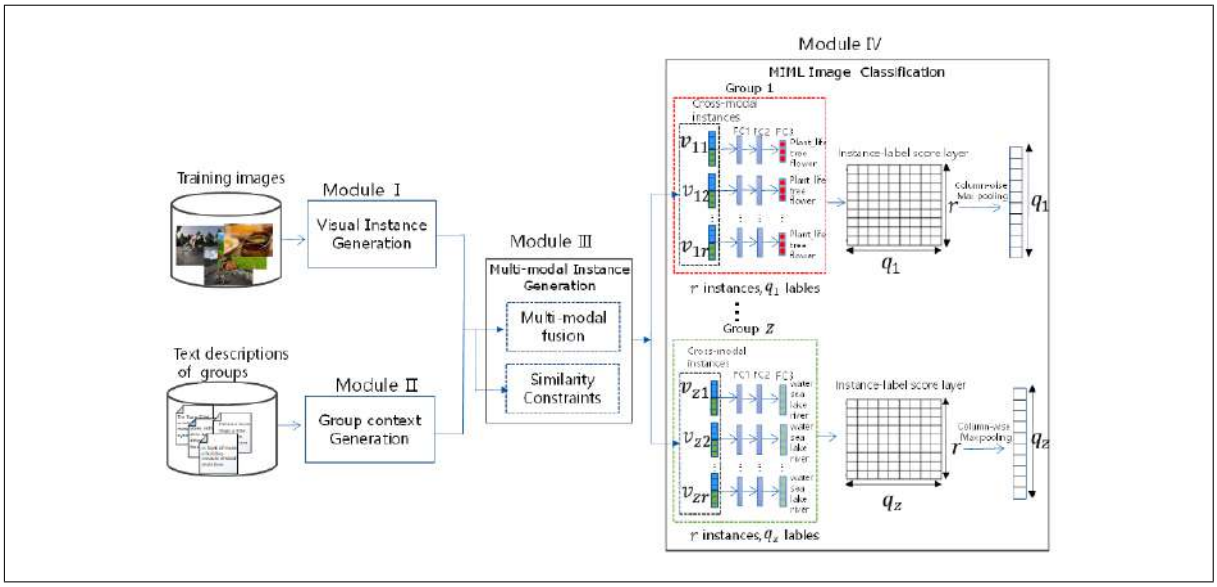

Figure 3.3: The framework of MMCNN-MIML [52]

In Module I, they described the generation of visual instances from images by exploiting the architecture of CNNs. They generated the instance representations by feature maps of the adaptation layer II as shown in (Figure3.4). In this layer, they split the generated visual instances into different groups. So, in each group, relevant labels share the same visual instances, which is help to learn features specific to these labels.

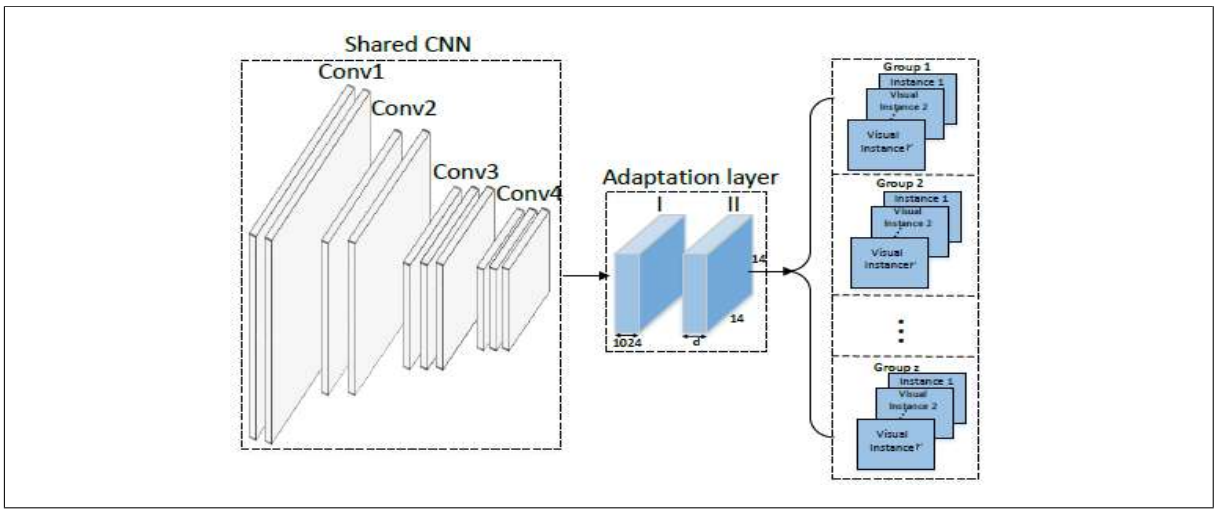

Figure 3.4: The architecture of Module I [52]

– (Durand, T, et al. 2019), proposed a method to learn a ConvNet with partial labels, in addition to a loss function, which automatically adapts to the proportion of known labels for each image. To illustrate the possibility of learning using partial labels, they compared labeling strategies for multi-label datasets. They use in the learned model a proposed method to predict some missing labels and adds them to the training set. To explicitly model the correlation between labels and improve the predictions of each one, they develop an approach based on Graph Neural Networks (GNNs). In GNN for MLC, each node represents single label and the edges are the connections between the labels. They use fully-connected graph to model all labels correlation. With the ConvNet output, it was initialized the node hidden states.[26]

– (Cevikalp, H., et al. 2020), presented a semi-supervised multi-label image classification method, and used the robust ramp loss in their method, to be able to learn from images with noisy and incomplete labels. To label the unlabeled data, they used label propagation based on the nearest labeled neighbors in the feature space. The proposed classifier was integrated within a deep CNN, to allow the classifier to use with hand-crafted features, or jointly trained with the feature extractor. Using the underlying features, the CNN can model correlations between labels. [18]

– (Wang, Jiang, et al. 2016), proposed a CNN-RNN framework to explicitly model the label dependencies as well as image-label relevance. It learned a joint embedding space as shown in (Figure3.5), the red dots and the blue ones are the label and the image embeddings respectively, and the black ones represent the sum of the image

and recurrent neuron output embeddings. [54]

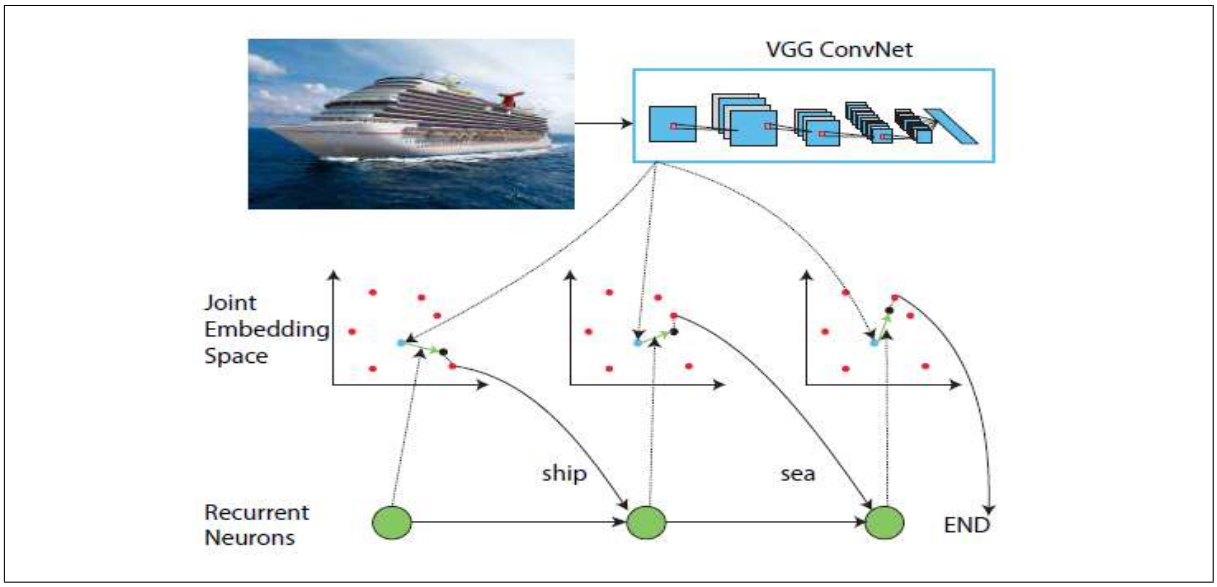

Figure 3.5: CNN-RNN framework for image MLC [54]

– (Yeh, Mei-Chen, and Yi-Nan Li. 2019), extended the visual-semantic embedding model presented in[28] to solve multi-label image classification. they proposed a new visual recognition model. The model consisted of a CNN framework and word embedding model as shown in (Figure3.6). The model learns a mapping (a transformation matrix) from an image instead of a latent visual vector, and use the image transformation matrix $\boldsymbol{A}$ to map words from an embedding word space **W** into a new space **W'**, where the relevant labels to an image are near to each other. [58]

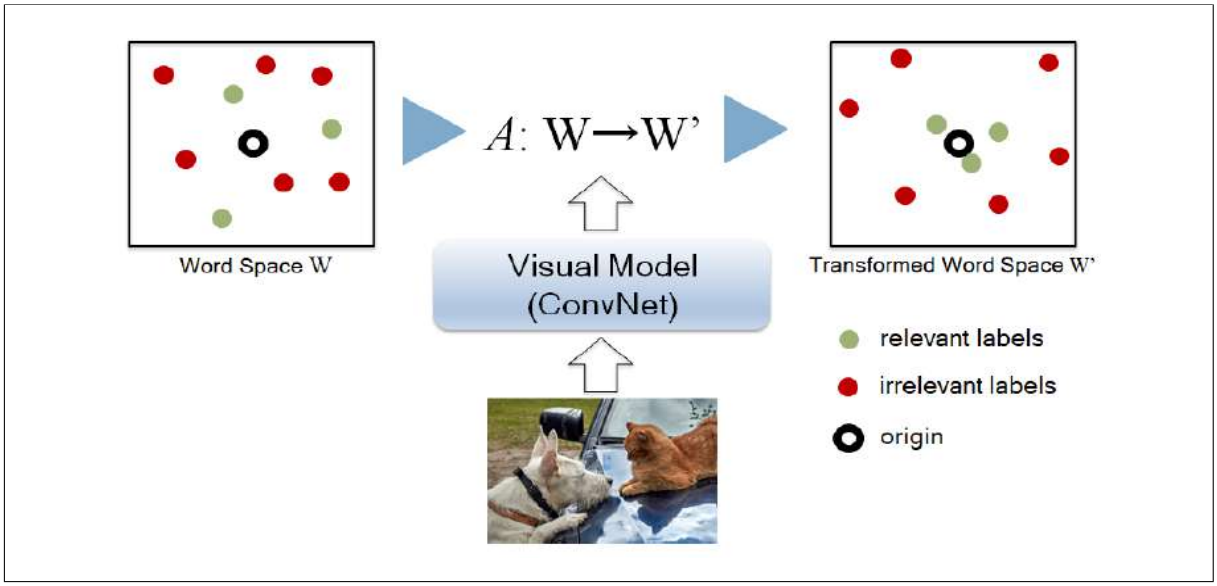

Figure 3.6: Visual-Semantic model for image MLC [58]

4 Conclusion

In this chapter we presented some recent literature works for multi-label image classification that make attention to the label-correlation. To well distinguish between them, we categorized them into two categories. This classification depend on the manner to learn the label correlations: implicitly or explicitly. Therefore, we mentioned a various recent solutions based on deep learning.

In our work we are interesting by exploring the category of methods that deeply and explicitly model the label-correlation. Especially, we will explore the last presented work[58], and give our proper contributions, as well. We justify our choice by the quality and clarity of presenting this work, that let us understand the most steps of the presented solution.

Chapter 4

A deep solution for multi-label image classification considering label correlation

Chapter 4

A deep learning solution for multi-label image classification considering label correlation

1 Introduction

Inspired by the success of deep learning (DL) in multi label image classification (MLC), and according to the great performance of Visual-Semantic model [58], We interested by exploring theses two axis to deal with multi-label image classification and considering label correlation as well. For that, we selected to explore an interesting recent work that contains all theses axis: label-correlation, image-label correlation and multi-label image classification (MLC)[58]. Hence, we consecrate this chapter to describe the solution given by the authors and present where we gave additional contributions.

2 General description of the solution

Yeh, Mei-Chen, and Yi-Nan Li [58] give a solution to multi-label image classification that consider the label correlations. The principal idea behind the presented solution is that considering this task as a binary classification of labels: the correct labels of an image are considered as positive ones and the other labels from the label set are considered as negative ones. Therefore, given an image $\mathcal{I}$ to be labeled, the task of the classifier is to partitioning the label set into two disjoint sets (positive and negative). For more understanding, we give an example. Supposing that we have a set of labels

$L = (Person, Chair, House, Tree, boat, Fish, Cat, Sea)$,

(Figure4.1) shows a separation of label set L to positive and negative labels according to its relevant to a given image.

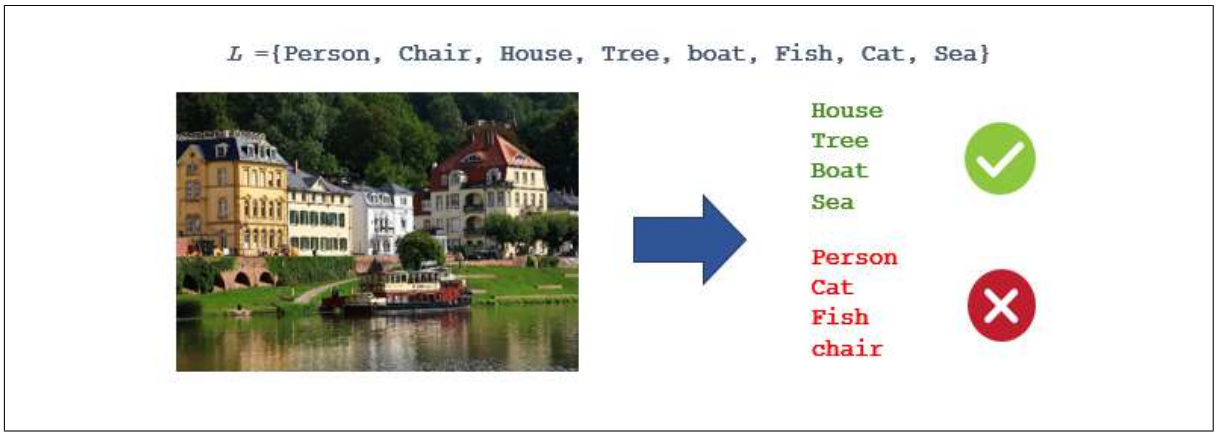

Figure 4.1: Example of the principle idea behind the presented classification as a binary classification

To do so, the authors model the visual-semantic multi-label image classifier by a linear transformation matrix, which is learned via a visual model (CNN) and by considering the label-correlation, the result from a word embedding model. For illustration, we can present the global architecture of this solution in the (Figure4.2).

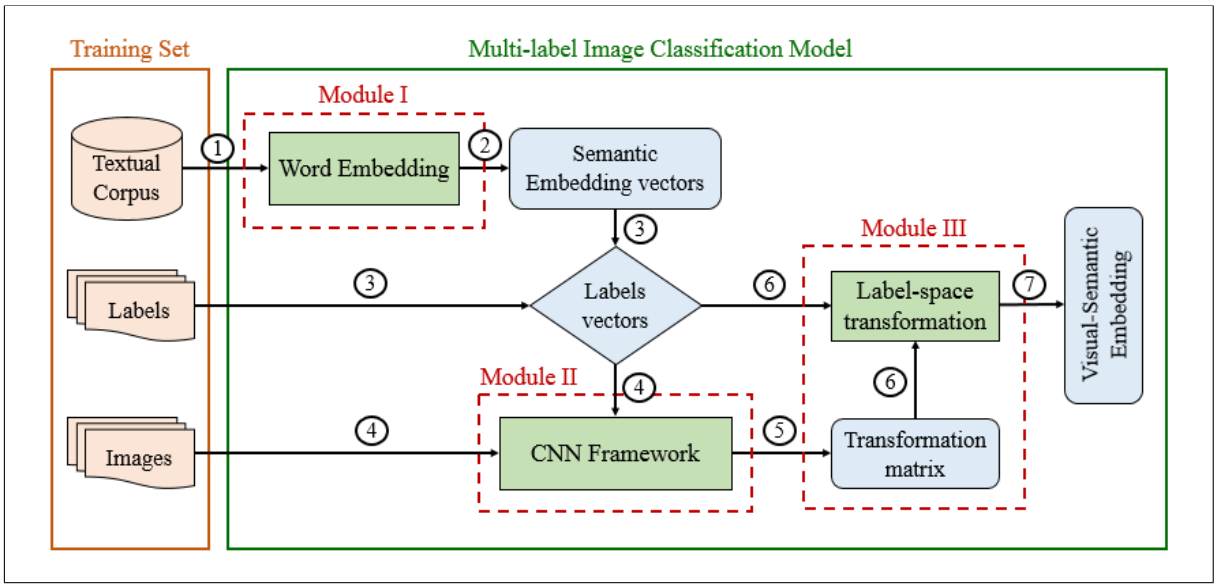

Figure 4.2: The general architecture of the presented model

As shown in the figure, the solution contains three modules: word embedding module, visual embedding module and transformation module. The first module consist of a word embedding model that maps words (labels) into a semantic embedding space of

d-dimension, where each semantic related labels are close to each other. The second one, is a CNN framework that learn a transformation matrix A with dimensions $k \times d$ from an input image $\mathcal{I}$ and the embedding vectors of its corresponding labels. The last module receive results of the two previous modules, to transform labels from d-dimensional space to a k-dimensional visual-semantic embedding space, which separated the relevant and irrelevant labels to the image $\mathcal{I}$. In the following section, we will describe these modules in detail.

3 Detailed description of the solution

The visual-semantic multi-label classifier of images exploits three big and challenging domains in machine learning: word embedding, multi-modal (text and image) representation and classification, in order to improve the performance of multi-label classifier of images. The modules that make up the model are defined in detail as follows:

3.1 Module I (Word embedding)

Word embedding module learns to represent labels (categories) in a semantic space, by representing each label in a fixed-length d-dimension embedding vector of real values, as shown in (Figure4.3). This module predicts the adjacent labels in order to give similar semantic embedding vectors for the labels which are semantically related.

In the proposed solution the authors used Word2Vec as a model for word embedding. For more exploration we will use several models as, Word2Vec, Glove and FastText. These models are deep neural networks usually used for classification or word/sentence representation in NLP domain. Therefore, to train these models a large textual corpora must be used as an input, like Wikipedia, Google news,...etc.

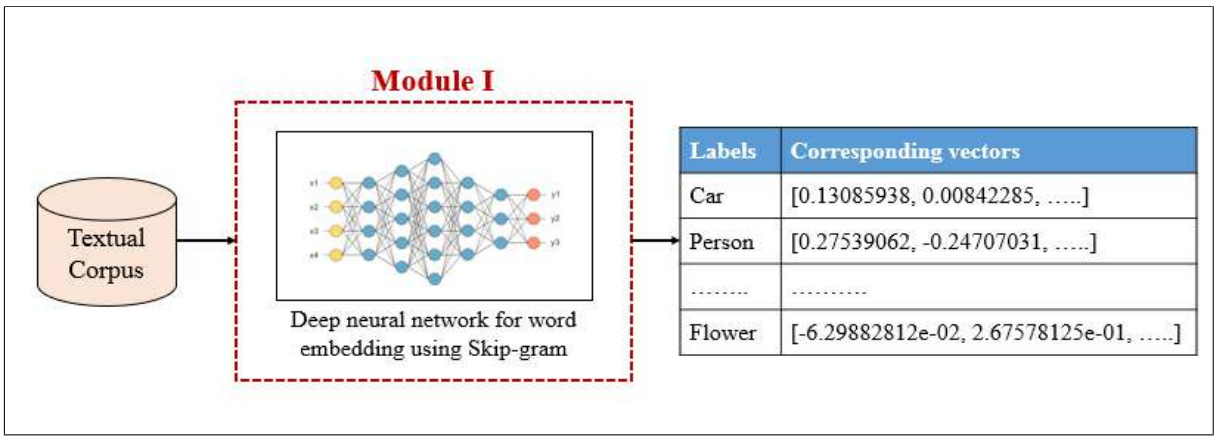

Figure 4.3: The architecture of word embedding module

3.2 Module II (Visual embedding)

It is a CNN model that takes as input an image $\mathcal{I}$ and embedding vectors of its labels. The objective of this CNN is not to give a visual representation of this image $\mathcal{I}$ (visual vector) but is to learn from it a linear transformation matrix A considering its label embeddings (positive labels), so that the distance between transformed positive label vectors pi and the origin is smaller than that of negative ones nj (the other labels from label set):

$$\|Ap_i\|_2 < \|An_j\|_2 \tag{4.1}$$

Where A is the transformation matrix, pi are the corresponding d-dim vectors of the relevant (positive) labels to an image and nj are the d-dim vectors of the irrelevant (negative) labels to the image.

To train this CNN the authors used two loss functions. In our work we choose the second one: the log-sum-exp pairwise loss function as bellow:

$$L_{lsep} = \log(1 + \sum_i \sum_j \exp(\|Ap_i\|_2 - \|An_j\|_2)) \tag{4.2}$$

This loss function is smooth and differentiable , and this make it easier to optimize[58].

When the train is achieved and the loss function in optimized, this module is used to predict the matrix A for each new image to be labeled.

To achieve the described objective, in general, any deep visual neural network can be used to train the image dependent word classifier[58]. The main thing is that the dimension of the output layer is set to the size of the transformation matrix A. The

authors used the visual model VGG-16. However, in our work we will experiment other deep neural network. This deep neural network uses a dataset of images and the word embedding d-dimension vectors (results from the first module) for training, and get in the output the transformation matrix A with dimensions $k \times d$ (Figure4.4) shows the general architecture of the module II.

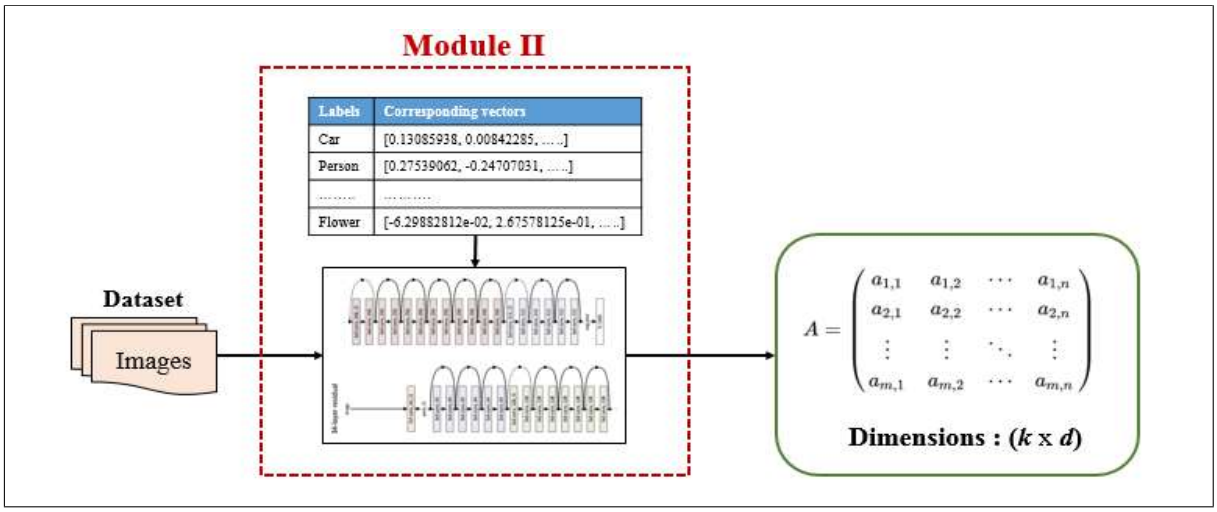

Figure 4.4: The architecture of the visual embedding module

3.3 Module III (Prediction of relevant labels by Transformation)

At this stage, the two previous modules are trained. Hence, In the presence of a new image $\mathcal{I}$ to be multi-label classified, the current module aims to predict the relevant labels to this input image. For doing, the previous module predict the corresponding transformation matrix A, and the current module transform the original d dimensional embedding vectors of all labels into new k dimensional visual-semantic space, and thus by multiplying each embedding vector by the predicted matrix A of the image $\mathcal{I}$. The result is that the relevant labels to the input image $\mathcal{I}$ will be near to the origin in the new space as shown as in (Figure4.5).

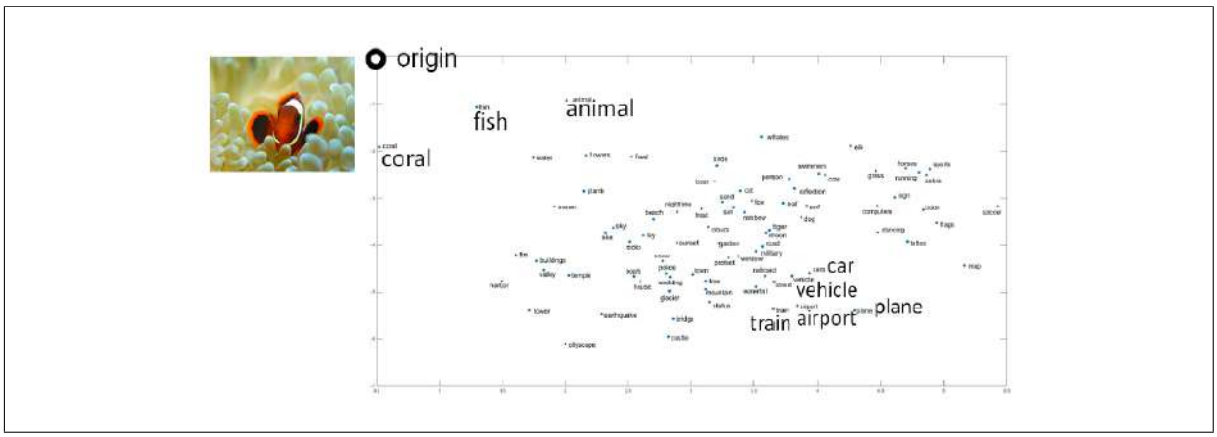

Figure 4.5: Example of visualization labels space (reduced to 2D space) of an input image[58]

4 Conclusion

In this chapter we deeply described the solution that we explored from the literature and we presented where we gave additional contributions. In the following chapter we will present our experiments and evaluate the results with analyze.

Chapter 5

Experiments, Results and Discussion

Chapter 5

Experiments, Results and Discussion

1 introduction

In this chapter we will present our experiments on the presented solution. Hence, we start by presenting the experimental settings. After that, we give the experiments of the solution steps and analyzing the obtained results. We also present a comparison between word embedding models. Noting that, we implement our proper code for the different modules (the authors code is not given in the web).

2 Experimental Settings

In this section, we present the experimental settings (development tools, the image dataset, the textual corpora and the evaluation protocol) that we used in our experiments.

2.1 Development Tools

We used in separation of the dataset the personal computer with the following characteristics:

- Toshiba Windows 7 Professional with 8,00 Go memory capacity, processor Intel(R) Core(TM) i7-4510U CPU @ 2.60 GHz and system 64 bits.

For code implementation we use Kaggle (online editor) with the following specifications:

- Intel(R) Xeon(R) CPU @ 2.30GHz, 16GB RAM.

2.2 Textual corpora for word embedding

In order to train a word-embedding model , we need to use a textual corpus as an input. For that, we do the experiments of corresponding module using different corpus as follows:

- Google News corpus, contain about 100 billion running words.[3]
- Common Crawl corpus, contains petabytes of data collected over 8 years of web crawling. [11]
- UMBC WebBase corpus, is a dataset of high quality English paragraphs containing over three billion words.[8]
- statmt.org news dataset
- Wikipedia 2017 dataset, contain about 5 million articles, with more than 23 million individual sections.[1]

2.3 Multi-label dataset " NUS-WIDE "

NUS-WIDE [21] *"A Real-World Web Image Dataset from National University of Singapore"*, is a multi-label image dataset, created by NUS's Lab for Media Search, contains:

- 269,648 Images associated 5,018 tags from Flickr.
- Images list divided into 161,789 training images, and 107,859 testing images [55].
- Ground-truth for 81 concepts.

(Figure5.1) shows different examples of images from NUS-WIDE dataset with the corresponding labels.

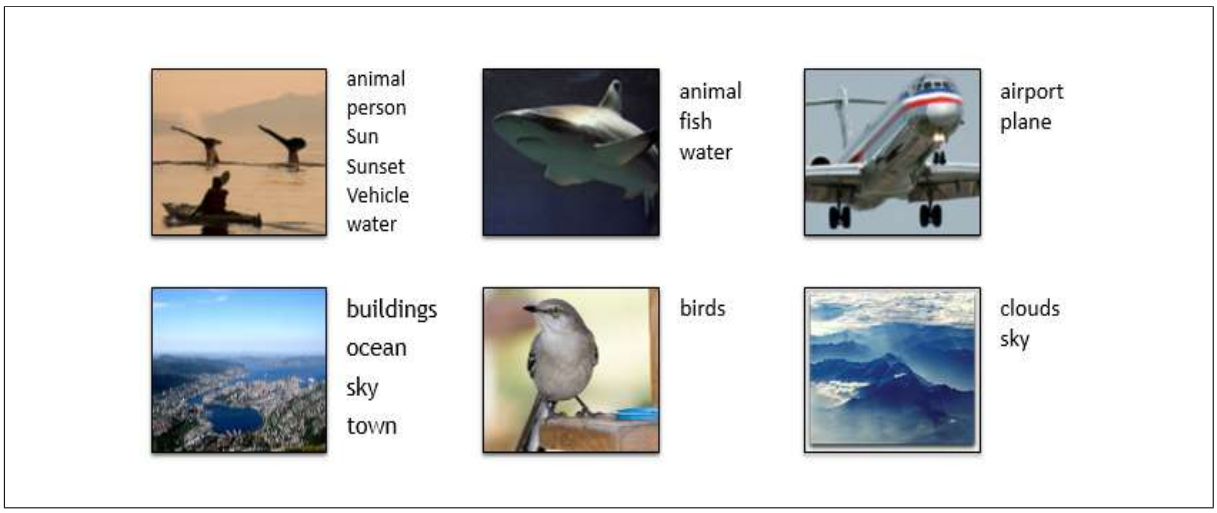

Figure 5.1: Example from NUS-WIDE images with the corresponding labels of each one

2.4 Evaluation Protocol

We will evaluate the word-embedding module and the prediction module as well, , as following:

Word embedding evaluation protocol:

In order to evaluate the obtained vectors from word embedding models, and comparing between their results, we use two evaluation protocols. The first one consists to visualize the embedding results, and the second one aims to compute the similarity between labels. So, to visualize the embedding results we need to apply a dimensionality reduction to two or three dimensions. For that we using in experiments the following two methods:

- **t-SNE** (t-distributed stochastic neighbor embedding), is a machine learning algorithm used for embedding and dimensionality reduction as well. t-SNE aims to map a high-dimensional data into a low-dimensional space of two or three dimensions. So that, they can be visualized by giving each data point a location in the reduced space.[40]

- **UMAP** (Uniform Manifold Approximation and Projection for Dimension Reduction), is a machine learning technique for dimensionality reduction, aims to accurately represent local structure and better incorporate global structure. It is well scalable with large datasets [16]. It can applied on real world data and try to find a low dimensional embedding to data. It has superior run time performance and

preserves more the global structure. The UMAP algorithm competes with t-SNE for visualization quality [43].

In Addition of evaluating the word embedding results by visualization of the reduced obtained embedding vectors, we aim to do an empirical evaluation as well. For that, we compute the similarity between label embedding vectors to show if the near labels in the semantic space (embedding space) have a big similarity. In other words, to show if the semantically related labels have a big similarity measure comparing to the other labels in the embedding space. To do so, we use the known measure: **cosine similarity** measurement. It computes the cosine between two vectors A and B. It is calculated by the equation [46], and gives values between 0 and 1. Hence, a value of 1 signify identical labels (synonyms), value of 0 signify independent labels, and values in between express how much two labels are semantically correlated.

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} \tag{5.1}$$

where $\|A\|$ is the norm of the vector A.
So, if two labels are semantically similar, or have a big semantic correlation, the cosine between their vectors will be high and vice versa.

Multi-label prediction evaluation protocol:

In order to measure the classifier performance we use the accuracy. Accuracy is the most fundamental metric for evaluation of the classification, it shows us how good performing the classifier. The value of accuracy range from 0 to 1 [51]. Its general formula [4] is:

$$Accuracy = \frac{Correct\ Predictions}{Total\ Number\ of\ Examples} \tag{5.2}$$

3 Experimental Results

In this section, we will present in detail the obtained results from the different experiments we done:

3.1 Word embedding results

We have experienced three pre-trained models of word embedding, Word2Vec, GloVe and FastText.

The obtained results from each one are 300 dimensions word embedding vectors of the given vocabulary. We selected the word embedding vectors of 81 labels of NUS-WIDE.

In fact, evaluating the word embedding quality from vectors of 300 dimensions directly is not possible. For that, we will evaluate the obtained results by :

- The visualization of resulted vectors after applying dimensionality reduction,
- Computing the semantic similarity between pairs of labels

Visualization of the word embedding resulting vectors:

In order to visualize the obtained embedding vectors, we selected 81 concepts of NUS-WIDE , and apply the two dimensionality reduction techniques t-SNE and UMAP to map vectors from a 300 dimensions to 2 dimensions.

1. **Word2Vec word embedding results:**

   Word2Vec is a pre-trained word embedding neural network on a part of Google News corpus containing 3 million words and phrases [45]. It is available in : [3] . it gives in output vectors of 300 dimensions.

   The figure5.2 and the figure5.3 show the visualization of the **Word2Vec** obtained word-embedding vectors of the 81 labels of NUS-WIDE using t-SNE and UMAP respectively .

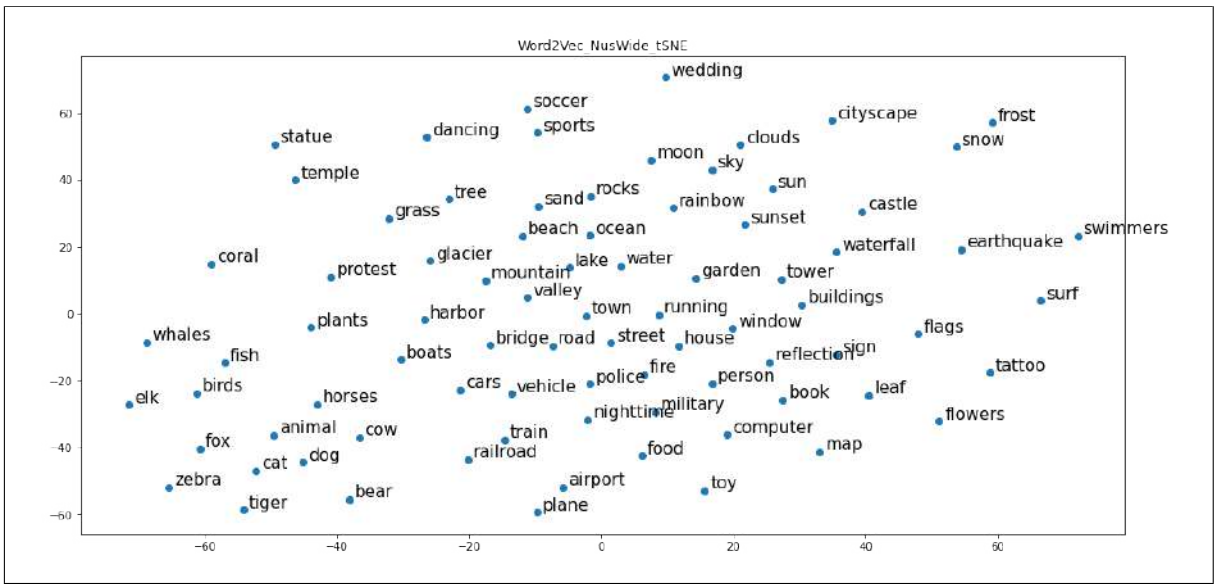

Figure 5.2: Visualization of the **Word2Vec** obtained word-embedding vectors of the 81 labels of NUS-WIDE using t-SNE

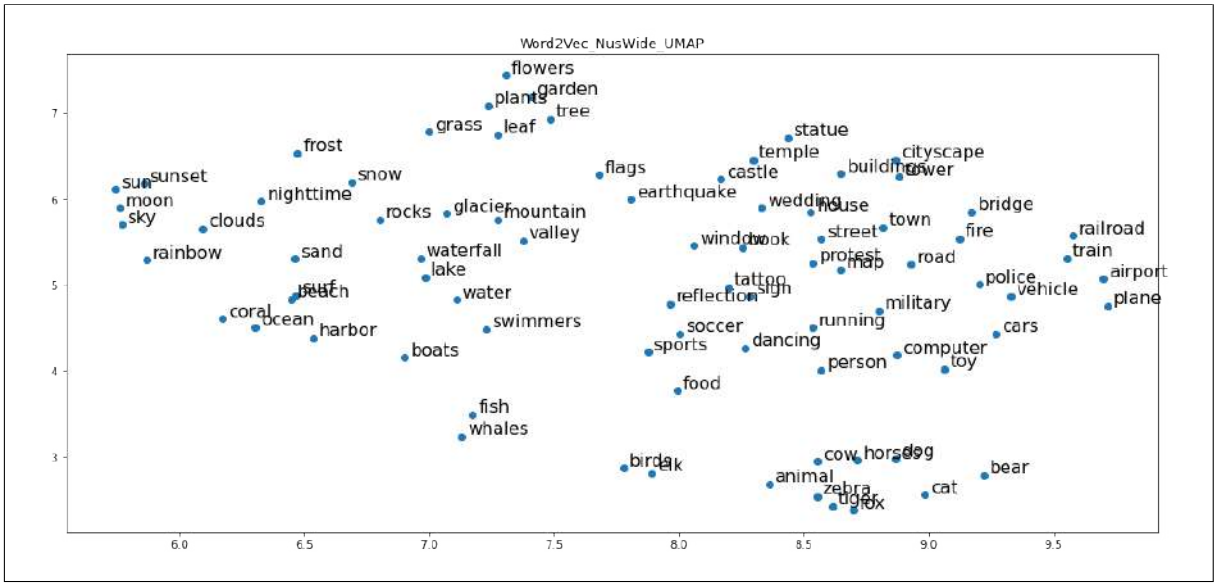

Figure 5.3: Visualization of the **Word2Vec** obtained word-embedding vectors of the 81 labels of NUS-WIDE using UMAP

2. **GloVe word embedding results:**

   GloVe is pre-trained word embedding neural network. for its train 1.9 Million words used from Common Crawl dataset [48]. It gives in output vectors of 300 dimensions.

   The figure5.4 and the figure5.5 show the visualization of the **GloVe** obtained word-embedding vectors of the 81 labels of NUS-WIDE using t-SNE and UMAP respectively .

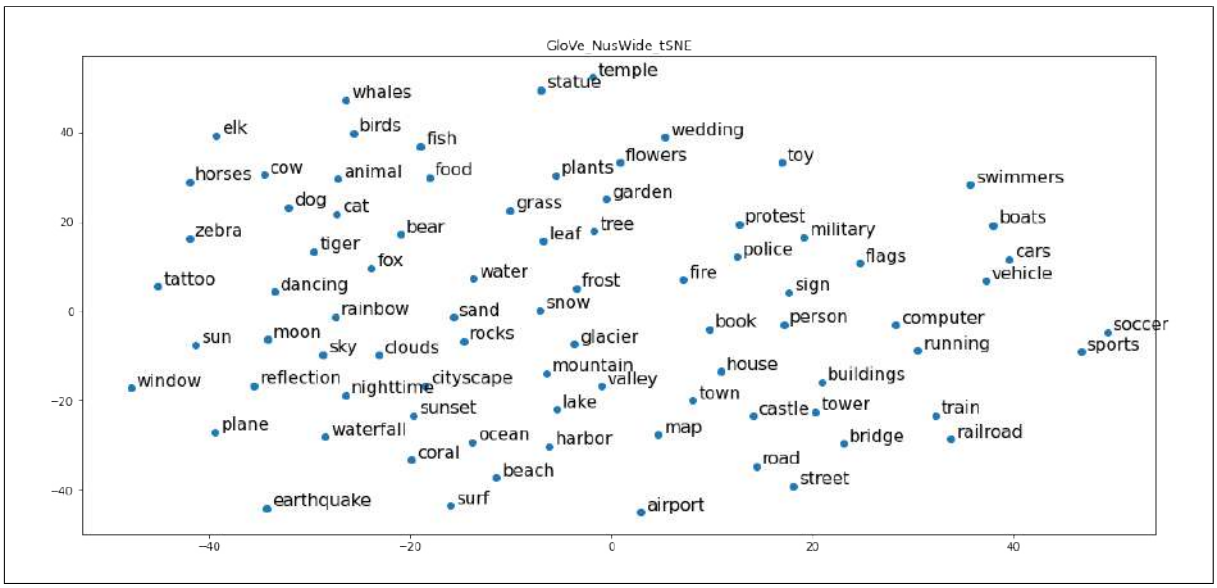

Figure 5.4: Visualization of the **GloVe** obtained word-embedding vectors of the 81 labels of NUS-WIDE using t-SNE

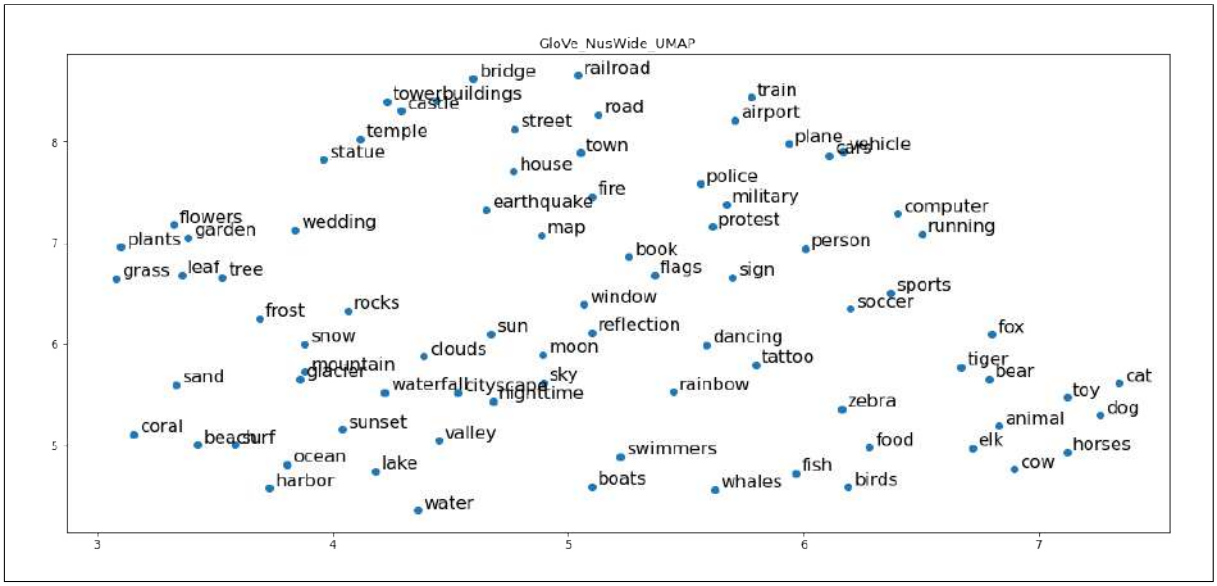

Figure 5.5: Visualization of the **GloVe** obtained word-embedding vectors of the 81 labels of NUS-WIDE using UMAP

3. **FastText word embedding results:**

   FastText is a pre-trained word embedding neural network on 1 Million of words and phrases collected from Wikipedia 2017 corpus, UMBC WebBase and statmt.org news dataset[44].it gives in output vectors of 300 dimensions. It is available in [2]. FastText is the newest and the fastest model comparing by Word2Vec and GloVe[17].

The figure5.6 and the figure5.7 show the visualization of the **FastText** obtained word-embedding vectors of the 81 labels of NUS-WIDE using t-SNE and UMAP respectively .

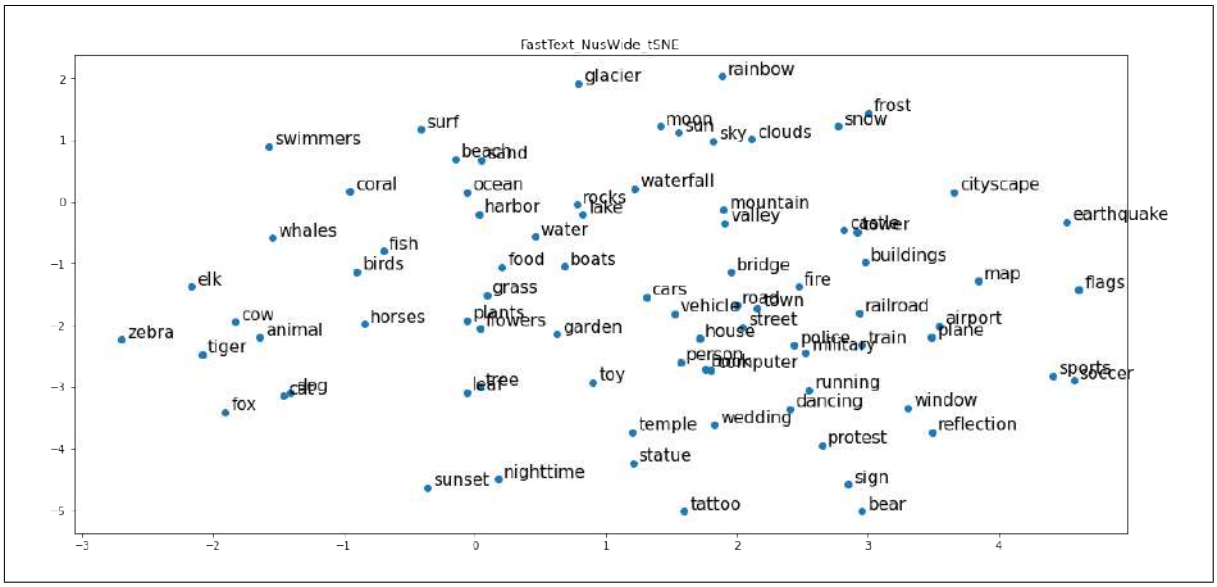

Figure 5.6: Visualization of the **FastText** obtained word-embedding vectors of the 81 labels of NUS-WIDE using t-SNE

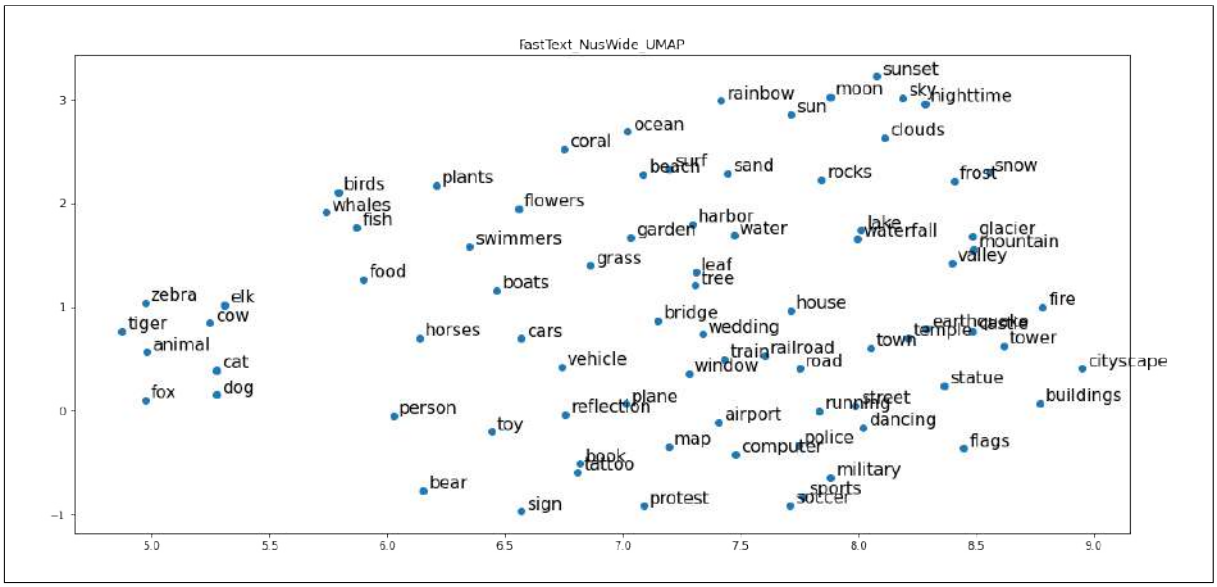

Figure 5.7: Visualization of the **FastText** obtained word-embedding vectors of the 81 labels of NUS-WIDE using UMAP

From these different figures it is clear that the three models Word2Vec, GloVe or FastText are globally a good word embedding models and this either by using t-SNE or UMAP. Seeing that the labels which are semantically similar, they are mapped close to each other. For example the labels: cat, dog, tiger, cow, horses are mapped close to

each other. This is very real because they are semantically correlated and shared a same semantic class which is 'animals'. A profound analyze will be given in the corresponding section.

Semantic similarity between word embedding vectors:

To measure the capability of the tree models in capturing the semantic in the embedding vectors, we compute the semantic similarity between vectors using **Cosine Similarity**. As an example we choose to measure the similarity between the two labels 'flower' and 'garden' in one hand and between 'whales' and 'bed' in the other hand. The obtained cosine similarity between corresponding vectors from the different word embedding models, are presented in the table 5.1. Remembering that the cosine similarity gives values between 0 and 1.

Example / Model	Word2Vec	GloVe	FastText
'flower' - 'garden'	0.59	0.62	**0.63**
'whales' - 'bed'	**0.03**	0.09	0.24

Table 5.1: Semantic similarity between pairs of word embedding label vectors of Word2Vec, GloVe and FastText respectively

After computing the semantic similarity between several pairs of labels, we noted that FastText was the most permanent in capturing the semantic correlation among labels. Hence, we select its word embedding results of the 81 concepts of NUS-WIDE as an input for the coming experiments on learning a CNN model and prediction module as well.
The figure 5.8) shows a selection of the word embedding vectors for some NUS-WIDE labels from the resulting vectors of the word embedding module

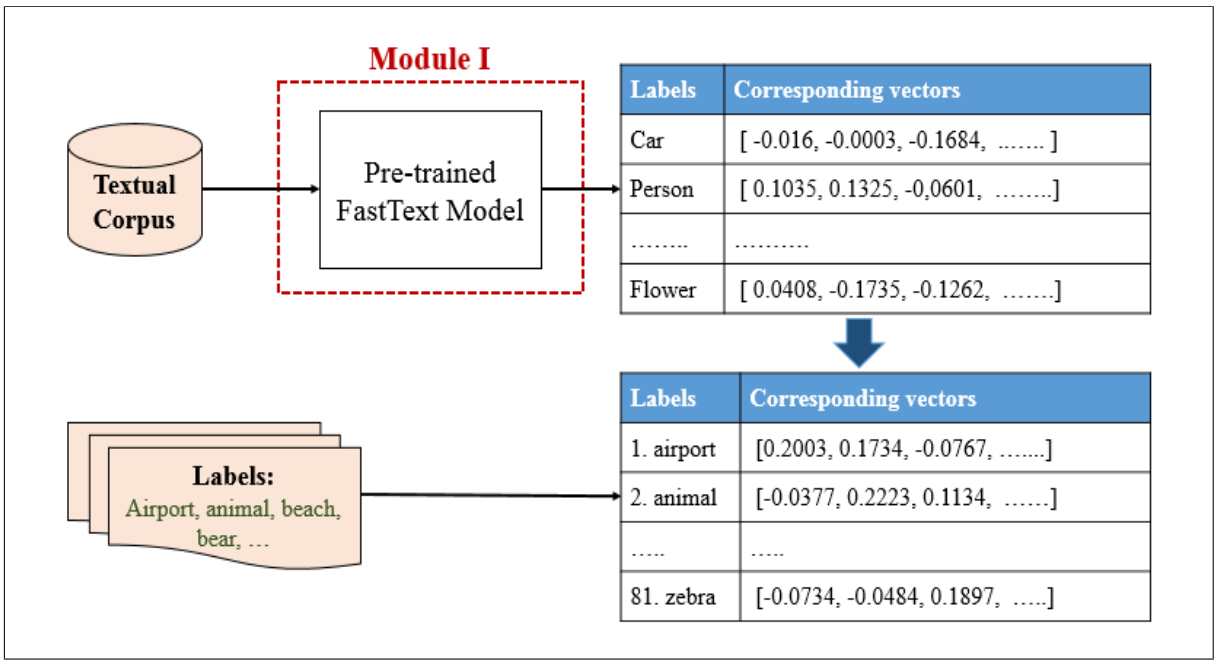

Figure 5.8: Word embedding vectors for some NUS-WIDE labels from the resulting vectors of the word embedding module

3.2 Visual embedding results using ResNet

As we presented in the previous chapter, the goal is to use a CNN model that takes as input an image $\mathcal{I}$ and embedding vectors of its labels (positive labels) in order ,not to learn a visual representation but, to learn a linear transformation matrix A. So that the distance between transformed positive label vectors pi and the origin is smaller than that of negative ones ni (the other labels from label set). For doing, any CNN can be fine-tuned, such as AlexNet, ResNet, GoogLeNet and VGGNet.

In fact, **ResNet** (Residual Network) won in ImageNet Large Scale Visual Recognition Competition for image classification 2015.[60], and it shows its performance in image classification against other CNNs. For that, we selected it to learn this matrix.

ResNet was proposed by (He, Kaiming, et al. in 2015) at Microsoft Research. This new architecture is a residual learning framework to make easy to train networks that are more deeper. Its architecture inspired by VGG-19, but this network uses 34-layer with adding the shortcut connections, and this is what makes it a residual network. In addition, the residual networks are easier to optimize.[30]. The way this network works is by use residual learning to every few stacked layers (called blocks). in this block it use the "shortcut connections" (Figure5.9) to skipping a number of layers, in order to skipped any layer hurts the performance.[30]

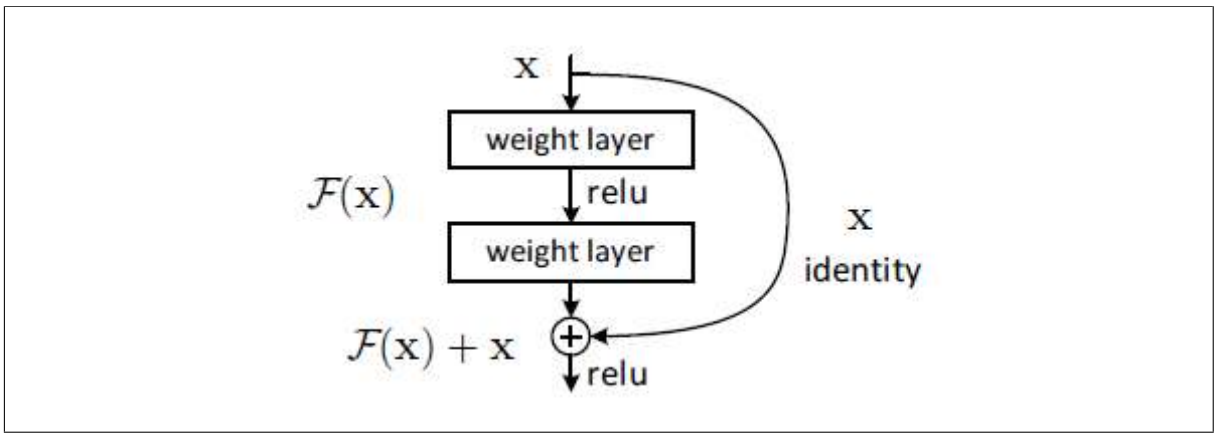

Figure 5.9: The architecture of a building block in ResNet[30]

Therefore, to make the ResNet50 learn a matrix A for a given multi labeled image, we proceed as follows :

1. We imported a pre-trained ResNet50 model as shown in (Figure5.10).

2. Because that the model is trained on a single label dataset (ImageNet), we fine-tuned it and add our output layer suit the multi-label model as shown as in (Figure5.11). This by using the per-trained model weights and delete the trainable parameters without the added of the last layer.

```
Model: "resnet50"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 200, 200, 3) 0
__________________________________________________________________________________________________
conv1_pad (ZeroPadding2D)       (None, 206, 206, 3)  0           input_1[0][0]
__________________________________________________________________________________________________
conv1_conv (Conv2D)             (None, 100, 100, 64) 9472        conv1_pad[0][0]
__________________________________________________________________________________________________
conv1_bn (BatchNormalization)   (None, 100, 100, 64) 256         conv1_conv[0][0]
__________________________________________________________________________________________________
conv1_relu (Activation)         (None, 100, 100, 64) 0           conv1_bn[0][0]
__________________________________________________________________________________________________
pool1_pad (ZeroPadding2D)       (None, 102, 102, 64) 0           conv1_relu[0][0]
__________________________________________________________________________________________________
pool1_pool (MaxPooling2D)       (None, 50, 50, 64)   0           pool1_pad[0][0]
• • • • • •
conv5_block3_3_conv (Conv2D)    (None, 7, 7, 2048)   1050624     conv5_block3_2_relu[0][0]
__________________________________________________________________________________________________
conv5_block3_3_bn (BatchNormali (None, 7, 7, 2048)   8192        conv5_block3_3_conv[0][0]
__________________________________________________________________________________________________
conv5_block3_add (Add)          (None, 7, 7, 2048)   0           conv5_block2_out[0][0]
                                                                 conv5_block3_3_bn[0][0]
__________________________________________________________________________________________________
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0           conv5_block3_add[0][0]
__________________________________________________________________________________________________
avg_pool (GlobalAveragePooling2 (None, 2048)         0           conv5_block3_out[0][0]
__________________________________________________________________________________________________
predictions (Dense)             (None, 1000)         2049000     avg_pool[0][0]
==================================================================================================
Total params: 25,636,712
Trainable params: 25,583,592
Non-trainable params: 53,120
```

Figure 5.10: Visualization of the pre-trained ResNet50 summary

```
Model: "functional_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 200, 200, 3) 0
__________________________________________________________________________________________________
conv1_pad (ZeroPadding2D)       (None, 206, 206, 3)  0           input_1[0][0]
__________________________________________________________________________________________________
conv1_conv (Conv2D)             (None, 100, 100, 64) 9472        conv1_pad[0][0]
__________________________________________________________________________________________________
conv1_bn (BatchNormalization)   (None, 100, 100, 64) 256         conv1_conv[0][0]
__________________________________________________________________________________________________
conv1_relu (Activation)         (None, 100, 100, 64) 0           conv1_bn[0][0]
__________________________________________________________________________________________________
pool1_pad (ZeroPadding2D)       (None, 102, 102, 64) 0           conv1_relu[0][0]
• • • • • •
conv5_block3_add (Add)          (None, 7, 7, 2048)   0           conv5_block2_out[0][0]
                                                                 conv5_block3_3_bn[0][0]
__________________________________________________________________________________________________
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0           conv5_block3_add[0][0]
__________________________________________________________________________________________________
avg_pool (GlobalAveragePooling2 (None, 2048)         0           conv5_block3_out[0][0]
__________________________________________________________________________________________________
flatten (Flatten)               (None, 2048)         0           avg_pool[0][0]
__________________________________________________________________________________________________
output_layer (Dense)            (None, 3000)         6147000     flatten[0][0]
==================================================================================================
Total params: 29,734,712
Trainable params: 6,147,000
Non-trainable params: 23,587,712
```

Figure 5.11: The summery of ResNet50 after fine-tuned

3. Because of NUS-WIDE dataset (Train and Test images folders) wasn't separated and because it contains a big number of images, it takes time in separation and also in training. So, we create a sub folders of training and testing images with a small numbers. Training set contains 2967 images while the testing set contains 1992 images, taking into consideration removal of the images that doesn't contain any labels (positive labels) from the 81 labels. At the same time we separate the corresponding tags (annotations) and the images names of our new data from the original file.

4. To make the ResNet train on the image dataset, we complete by the following steps:

   - We was mentioned that we will use the word embedding model FastText. So, the final multi-label model combine between the **FastText** and **ResNet**.
   - We pass the training set of images to the ResNet from dataset, in addition to the corresponding labels (annotations) and the embedding labels vectors of those images from FastText model. Where, the last layer length of ResNet is 3000 (300×10), 300 is the length d of obtained vectors from FastText, and 10 is a chosen number k.

- During the training we separate the corresponding 300-*dim* vectors of the positive and negative labels of each image to pass them after that to the loss function.
- The last step of the model creation is developing the log-sum-exp loss function by using the 300-*dim* embedding vectors, the annotations and the elements of the last layer.

  In fact, we stopped in this step because of the problem: the development of this loss function, and this according to the complex implementation rules of creating a loss function, and because it used other parameters out of the main parameters that passed by the model-fit function.

4 Discussion of Results

In this section we will discuss the obtained results :

4.1 Discussion of word embedding results

According to the visualizations above, we can observe the following:

1. After experiment the three models of word embedding (Word2Vec, Glove and FastText) in (Section 3.1), we note that in the two visualizations with t-SNE and UMAP for the three models the labels that belong to the same semantic category, they are mapped close to each other. However, in the separation between the different categories, UMAP shows a good performance against t-SNE, in all models. This because of UMAP is better in the preservation of the global structure[16] (the global positions of clusters) of data than t-SNE, without forgetting the local structure.[10]

2. By looking to the UMAP visualizations of the three models, we remark that FastText model separated the concepts well in comparison to Word2Vec and GloVe. As an example if we take the word 'bird', is an animal but in the same time is close to nature ('plants', 'flowers', ..), and that is clear in FastText model by UMAP. As a second example, words ('whales' and 'fish') are also animals but in other hand are close to the sea concepts as ('swimmers' and 'boats').

3. After computing the semantic similarity between several pairs of labels, we noted that FastText was the most permanent in capturing the semantic correlation among labels. FastText take into account the internal structure of words which could be very useful for words that occur rarely, and also for morphologically rich languages.

4.2 Discussion of visual embedding results

ResNet50 presents a complex architecture compared to the other ones like VGG16. This makes the customizing of the training function and the called sub-functions a very complicated task.

5 Conclusion

In this chapter we have presented our experiments on the explored solution. We have started by define the experimental settings: textual corpora and image dataset. After that we have mentioned the used evaluation protocol during the experiments. We continued by given the results that we got it in all steps. Finally, we have evaluated and analyzed the obtained results.

General conclusion

Through research on artificial intelligent systems, we can find several operations of human brain that was modeled or simulated by computers, such as knowledge, learning, reasoning, recognition and classification, ...etc. Machine learning is a part of artificial intelligence that cover a considerable number of those operations (tasks). Therefore, the image classification is one of machine learning tasks, that aims to categorize images into one or more predefined classes (labels). Multi-label image classification is considering as an important and challenging task. The main idea behind multi-label image classification is to classifier images, where assign for each image more than one class.

Multi-label image classification is considering as a challenging task, according to the complexity of images and labels information. In order to simplify the challenge, a big panoply of literature works consider a total independence among labels as an hypothesis within the given solutions. However, this contradict the reality, where it exists a certain semantic shared (correlation) between subsets of labels. For that, other literature solutions take attention to the label-correlation, we have classified them into two big categories. The first category gave solutions that learn the dependencies among labels using traditional methods. The second category learns the label correlations using deep learning methods. We have categorized the second one into two sub-categories according to the label correlations learning. The first sub-category, learns the dependencies among labels implicitly, and the second one learns label correlations explicitly. In this thesis we have explored in a deep solution that learns label correlations explicitly.

The presented solution of multi-label image classification considering label correlations is a visual-semantic multi-label image classifier. The principal idea behind this solution is that considering this task as a binary classification of labels. Therefore, given an image to be labeled, the task of the classifier is to partitioning the label set into two disjoint

sets (positive and negative). This classifier consist on three modules. First one, used FastText as a word embedding model to represent labels into a semantic d-dimensional space. Second module, used CNN framework witch is a ResNet that learns a transformation matrix from the input image considering its label embeddings (positive labels). The last module used the results of the two previous module (embedding vectors and transformation matrix) and transformed labels from the semantic d-dimensional space to a new visual-semantic k-dimensional space, to predict the relevant labels to the input image.

As well as, we did a comparison between three word embedding models Word2Vec, GloVe and FastText. In order to optimize the performance of the existing solution, we have used FastText model according to its performing and speed. And we use ResNet due of its grate performance in image classification against other CNNs that have a simple architecture.

As difficulties, the studied field is wide and contain a huge number of different works. However, we tried to cover a number of some important works. The main difficult and problem we found it is in implementation, and exactly in customizing of machine learning functions to emulate it with our problem.

As perspectives of our work:

- Firstly, we hope to complete the implementation of the presented solution to show the given results for multi-label image classification .
- Experiment other visual deep learning models with other word embedding models.
- In addition of applying a linear transformation in order to joint the two modalities images and labels (visual and semantic modalities), look for other methods for multi-modalities representation or modalities fusion. The goal is to improve the performance of the multi-label image classifier that consider the two modalities.
- Improve the presented solution to be robust against noisy labels (false labels in the training images).
- Adapt the presented solution for a semi-supervised configuration (when a part of training data is unlabeled).

- Exploring more deep solutions for modeling the label correlation either implicitly or explicitly, in order to look for better results on capturing this semantic information.
- More exploring the domains of multi-modal representation and multi-modal fusion.
- Combining our work with these of other colleagues, like image segmentation by MLC, and Semi-supervised learning for MLC.
- Explore the active learning for MLC.
- Explore the graph based neural networks (as hyper graph NN, Graph convolutional NN: GCNN,...) for MLC.
- Exploring the implicit methods via intention mechanism.
- Applying the MLC solutions for other domain as the Medicine.

Bibliography

[1] English wikipedia articles 2017-08-20 sqlite — kaggle. https://www.kaggle.com/jkkphys/english-wikipedia-articles-20170820-sqlite. (Accessed on 09/18/2020).

[2] English word vectors · fasttext. https://fasttext.cc/docs/en/english-vectors.html. (Accessed on 09/12/2020).

[3] Google code archive - long-term storage for google code project hosting. https://code.google.com/archive/p/word2vec/. (Accessed on 09/12/2020).

[4] Machine learning glossary — google developers. https://developers.google.com/machine-learning/glossary#a. (Accessed on 09/18/2020).

[5] The magic behind embedding models — by mohamed gharibi — towards data science. https://towardsdatascience.com/the-magic-behind-embedding-models-c3af62f71fb. (Accessed on 06/22/2020).

[6] More performance evaluation metrics for classification problems you should know. https://www.kdnuggets.com/2020/04/performance-evaluation-metrics-classification.html. (Accessed on 07/07/2020).

[7] Tips and tricks for multi-class classification — by mohammed terry-jack — medium. https://medium.com/@b.terryjack/tips-and-tricks-for-multi-class-classification-c184ae1c8ffc. (Accessed on 07/07/2020).

[8] Umbc webbase corpus of 3b english words - umbc ebiquity umbc ebiquity. https://ebiquity.umbc.edu/blogger/2013/05/01/umbc-webbase-corpus-of-3b-english-words/. (Accessed on 09/18/2020).

[9] Understanding of convolutional neural network (cnn) — deep learning — by prabhu — medium. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148. (Accessed on 07/08/2020).

[10] Understanding umap. https://pair-code.github.io/understanding-umap/. (Accessed on 09/18/2020).

[11] Want to use our data? – common crawl. https://commoncrawl.org/the-data/. (Accessed on 09/18/2020).

[12] Word embeddings in nlp — word2vec — glove — fasttext — by aravind cr — analytics vidhya — aug, 2020 — medium. https://medium.com/analytics-vidhya/word-embeddings-in-nlp-word2vec-glove-fasttext-24d4d4286a73. (Accessed on 09/19/2020).

[13] Word2vec - data analytics post. https://dataanalyticspost.com/Lexique/word2vec/. (Accessed on 07/03/2020).

[14] Oussama Aiadi, Belal Khaldi, Mohammed Lamine Kherfi, Yacine Ghorfa, and Rayhana Rezzag Bara. A supervised probabilistic model for visual object recognition.

[15] Oussama Aiadi and Mohammed Lamine Kherfi. Image classification using texture features and support vector machine (svm). 2019.

[16] Mebarka Allaoui, Mohammed Lamine Kherfi, and Abdelhakim Cheriet. Considerably improving clustering algorithms using umap dimensionality reduction technique: A comparative study. In *International Conference on Image and Signal Processing*, pages 317–325. Springer, 2020.

[17] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[18] Hakan Cevikalp, Burak Benligiray, and Omer Nezih Gerek. Semi-supervised robust deep neural networks for multi-label image classification. *Pattern Recognition*, 100:107164, 2020.

[19] Haomin Chen, Shun Miao, Daguang Xu, Gregory D Hager, and Adam P Harrison. Deep hierarchical multi-label classification of chest x-ray images. In *International Conference on Medical Imaging with Deep Learning*, pages 109–120, 2019.

[20] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5177–5186, 2019.

[21] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yan-Tao Zheng. Nus-wide: A real-world web image database from national university of singapore. In *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*, Santorini, Greece., July 8-10, 2009.

[22] Farah Debbagh. *Sélection des concepts et calcul de proximité sémantique pour réduire le silence dans la recherche d'images par le texte: vers des moteurs qui se configurent automatiquement*. PhD thesis, UNIVERSITE DE MOHAMED KHIDER BISKRA, 2017.

[23] Farah DEBBAGH, Mohammed Lamine KHERFI, and Mohamed Chaouki BABAHENINI. Une solution au problème de l'oubli en recherche d'images par les concepts et les relations sémantiques. In *La Conférence Internationale sur l'Intelligence Artificielle et les Technologies de l'Information ICA2IT*, page 7, 2014.

[24] Farah Debbagh, Mohammed Lamine Kherfi, and Mohamed Chaouki Babahenini. A semantic relatedness-based solution for reducing missing problem in tbir. *International Journal of Signal and Imaging Systems Engineering*, 10(3):146–156, 2017.

[25] Khaoula Drid, Mebarka Allaoui, and Mohammed Lamine Kherfi. Object detector combination for increasing accuracy and detecting more overlapping objects. In *International Conference on Image and Signal Processing*, pages 290–296. Springer, 2020.

[26] Thibaut Durand, Nazanin Mehrasa, and Greg Mori. Learning a deep convnet for multi-label classification with partial labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 647–657, 2019.

[27] Meng Joo Er, Rajasekar Venkatesan, and Ning Wang. An online universal classifier for binary, multi-class and multi-label classification. In *2016 IEEE International*

Conference on Systems, Man, and Cybernetics (SMC), pages 003701–003706. IEEE, 2016.

[28] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

[29] Antonio Gulli, Amita Kapoor, and Sujit Pal. *Deep Learning with TensorFlow 2 and Keras: Regression, ConvNets, GANs, RNNs, NLP, and More with TensorFlow 2 and the Keras API.* Packt Publishing, Limited, 2019.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[31] Francisco Herrera, Francisco Charte, Antonio J Rivera, and María J Del Jesus. Multilabel classification. In *Multilabel Classification*, pages 11–28. Springer, 2016.

[32] Belal Khaldi, Oussama Aiadi, and Kherfi Mohammed Lamine. Image representation using complete multi-texton histogram. *Multimedia Tools and Applications*, pages 1–19, 2020.

[33] Belal Khaldi and Mohammed Lamine Kherfi. Modified integrative color intensity co-occurrence matrix for texture image representation. *Journal of Electronic Imaging*, 25(5):053007, 2016.

[34] Aicha KORICHI, Oussama AIADI, and Mohammed Lamine KHERFI. A comparative study on arabic handwritten words recognition using textures descriptors.

[35] Meriem Korichi, Mohamed Lamine Kherfi, Mohamed Batouche, and Khadra Bouanane. Extended bayesian generalization model for understanding user's intention in semantics based images retrieval. *Multimedia Tools and Applications*, 77(23):31115–31138, 2018.

[36] Meriem Korichi, Mohamed Lamine Kherfi, Mohamed Batouche, Zineb Kaoudja, and Hadjer Bencheikh. Understanding user's intention in semantic based image retrieval: combining positive and negative examples. In *IFIP International Conference on Computational Intelligence and Its Applications*, pages 66–77. Springer, 2018.

[37] Kaushil Kundalia, Yash Patel, and Manan Shah. Multi-label movie genre detection from a movie poster using knowledge transfer learning. *Augmented Human Research*, 5(1):11, 2020.

[38] Changsheng Li, Chong Liu, Lixin Duan, Peng Gao, and Kai Zheng. Reconstruction regularized deep metric learning for multi-label image classification. *IEEE transactions on neural networks and learning systems*, 2019.

[39] Liu Liu, Daria Dzyabura, and Natalie Mizik. Visual listening in: Extracting brand image portrayed on social media. *Marketing Science*, 39(4):669–686, 2020.

[40] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[41] Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Sašo Džeroski. An extensive experimental comparison of methods for multi-label learning. *Pattern recognition*, 45(9):3084–3104, 2012.

[42] Andrew Maxwell, Runzhi Li, Bei Yang, Heng Weng, Aihua Ou, Huixiao Hong, Zhaoxian Zhou, Ping Gong, and Chaoyang Zhang. Deep learning architectures for multi-label classification of intelligent health risk prediction. *BMC bioinformatics*, 18(14):523, 2017.

[43] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[44] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[45] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[46] Mark Needham and Amy E Hodler. *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*. O'Reilly Media, 2019.

[47] Witold Pedrycz and Shyi-Ming Chen. *Deep Learning: Concepts and Architectures*. Springer, 2020.

[48] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[49] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[50] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333, 2011.

[51] Sandro Skansi. *Introduction to Deep Learning: from logical calculus to artificial intelligence*. Springer, 2018.

[52] Lingyun Song, Jun Liu, Buyue Qian, Mingxuan Sun, Kuan Yang, Meng Sun, and Samar Abbas. A deep multi-modal cnn for multi-instance multi-label image classification. *IEEE Transactions on Image Processing*, 27(12):6025–6038, 2018.

[53] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23(7):1079–1089, 2010.

[54] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2016.

[55] Liang Xie, Peng Pan, Yansheng Lu, and Shixun Wang. A cross-modal multi-task learning framework for image annotation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 431–440, 2014.

[56] Zheng Yan, Weiwei Liu, Shiping Wen, and Yin Yang. Multi-label image classification by feature attention network. *IEEE Access*, 7:98005–98013, 2019.

[57] Edward KY Yapp, Xiang Li, Wen Feng Lu, and Puay Siew Tan. Comparison of base classifiers for multi-label learning. *Neurocomputing*, 2020.

[58] Mei-Chen Yeh and Yi-Nan Li. Multilabel deep visual-semantic embedding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(6):1530–1536, 2019.

[59] ABDELMADJID YOUCEFA. *Understanding user intention in image retrieval using multiple concept hierarchies*. PhD thesis, Université de Ouargla-Kasdi Merbah.

[60] Chao Zhang, Zichao Yang, Xiaodong He, and Li Deng. Multimodal intelligence: Representation learning, information fusion, and applications. *IEEE Journal of Selected Topics in Signal Processing*, 2020.

[61] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2013.