HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

GRADUATION THESIS

KGPNet - A Knowledge Graph-assisted Framework for Pill Detection

NGUYỄN ANH DUY
duy.na184249@sis.hust.edu.vn

Major: Information Technology
Specialization: Global ICT Program

Supervisor: Dr. Nguyễn Phi Lê ______________

Signature

School: School of Information and Communications Technology

HANOI, 08/2022

ACKNOWLEDGMENT

To summarize my journey being a HUST student with one word, I would definitely choose *fortunate*. Entering college, it is the time for the transformation from a child into an adult. It is an inevitable process that everyone has to undergo, for me, it is really a blessing as I could go through this at HUST, since it plays a significant part in determining the course of my development.

The very first thanks is for my beloved supervisor, Dr. Nguyen Phi Le. For me, this woman takes so many roles, and all of them are extremely important. At work, Dr.Le came and led me in the right direction, up to now. I have always admired her so much for her strong sense of goal pursuit, aside from her brilliant talent and knowledge. I remembered Dr.Le told me not to do anything in haft, but to do it at your best, then if it fails, at least I had tried. Up to now, it always reminds me what is the right thing to do for my career. Second, in life, Dr. Le takes the role of a mother to me. She actually cares about my feelings, and wipes out all the boundaries between a supervisor and a student. I truly regret all the times I let her down, made her sad, and do not want to blame for any other factor. I appreciate this great relationship and hope it will remain even after my journey at HUST come to an end.

My second thanks is to all my friends I met at HUST, at AIoTLab. For me, you guys are not the longest friends, but are the most supporting ones that I could ever get up to now. Being a freshman in HUST, at the time, I was actually an introvert by all means. I felt it hard to open my feelings, as well as my opinions. Luckily, I met and talked with you guys, which made a big difference in my life at times. You might not even recognize, but all the conversations we talked ever since then gradually encouraged me a lot, which helped me walk out of my shell and became a better version of myself. To be honest, you guys all contain the characteristics of my role model person type who I want to become some day; and I learn from you guys alot!

Last but not least, I want to send my thanks to my family. To the present, as I have grown, you do not teach me anything directly related to my work. What you teach me is love, the way you care for others. No matter how grown I become, you still show me unconditional love, even at times I make you sad, let you wait, or even say something that hurts you much. After all, I know deep in my heart there is still a place that will never treat me wrong, that loves me in the way I actually am.

ABSTRACT

In many healthcare applications, identifying pills given their captured images under various conditions and backgrounds has been becoming more and more essential. However, this task is challenging, and a few works have addressed this issue satisfactorily. The lack of high-quality pill images also raises a significant concern when dealing with this problem. Due to the shortage of data, deep learning-based frameworks fail to converge and are unable to identify the most discriminated characteristics between classes. Additionally, some pills have extremely few visual variants, which is insufficient for models to distinguish in some cases. To alleviate these issues, this study presents a novel framework for pill detection, named KGPNet, that uses external links between labels in the form of a knowledge graph. Specifically, we propose a novel method for modeling the implicit association between pills in the presence of an external data source, in this case, prescription information. Second, a deep Graph Neural Network-based approach is used to perform node embedding using this graph presentation. Third, a final framework is provided that uses this information to accomplish final localization and categorization. In this output module, the whole graph representation of all labels is soft-mapped based on the current input images in order to provide an adaptive graph presentation. This output is then utilized to fetch into a Graph Transformer module, resulting in a semantically-rich context vector that assists with the final detection. To our knowledge, this is the first study to use external prescription data to establish associations between medicines and leveraging that information to enhance the performance of pill detection problem. The architecture of KGPNet is lightweight and has the flexibility to incorporate into any recognition backbones. It can integrate any external data source in the form of graph representation. Experimentally, the proposed KGPNet demonstrates its huge potential by outperforming all comparison benchmarks concerning the targeted detection task. In addition, different testing scenarios have been undertaken to determine the influence of the external graph, as well as different proposed modules on the model's performance. Specifically, KGPNet shows its superior by an enhancement of 9.4% for COCO mAP metrics, over Faster R-CNN. In addition, when putting together with another method which also leverages external knowledge, KGPNet enhances mAP score by 4.0.

Keyword: *Pill Detection, Knowledge Graph, Graph Embedding, CNN.*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	**Definition**
CNN	Convolutional Neural Network
GNN	Graph Neural Network
GTN	Graph Transformer Network
IoU	Intersect over Union
MCG	Medical Co-occurence Graph
RoI	Region of Interest
RSG	Relative Size Graph

CHAPTER 1. INTRODUCTION

In this chapter, I will present an overview of the problem to be solved, as well as a summary of the existing works and their key shortcomings. The study objectives and directions are next described, followed by the contributions of my thesis.

1.1 Problem Statement

Medicines are used to cure diseases and improve patients' health. Medication mistakes, however, may have serious consequences, including diminishing the efficacy of the treatment, causing adverse effects, or even leading to death. Some common mistakes in medication use are listed below.

- **Using pills in wrong times**. This circumstance cause little serious effect, but potential risks still exist and should be avoid.
- **Using pills with wrong amounts, medications**. This is the most common mistake made by patients during the pill usage. This mistake may lead to degradation in drugs' efficiency, even cause unwanted side effects.
- **Using wrong types of pills**. This is the most unwanted mistake considering drug intake. The seriousness of this action is highest, owing to it could bring about new pathologies or even be lethal in some cases.

As stated in a WHO report, drug abuse rather than sickness accounts for one-third of all deaths [1]. Moreover, according to Yaniv *et al.* [2], medication errors claim the lives of about six to eight thousand people every year. Recently, US National Centers for Biomedical Computing (NCBCs) states that taking this country alone, each year, 7000 to 9000 people die due to a medication error. Additionally, hundreds of thousands of individuals encounter adverse reactions or other issues associated to a medicine, but seldom report them. Each year, the overall cost of caring for patients impacted by medication-related mistakes surpasses 40 billion, with over 7 million individuals affected. In addition to the financial expense, drug mistakes cause patients psychological and physical pain and suffering. In Vietnam, according to National Center of DI&ADR [3], adverse drug reactions and medication mistakes were reported in 17,276 cases - or 177 per million persons. Figure 1.1 provides a breakdown of the report's ratios categorized by economic areas. As seen by the graph, the majority of received complaints related to patients from established provinces/cities. In addition, several inaccuracies in the actual usage of pharmaceuticals have not been reported to the national center for analysis of data. Numerous challenges associated with the care and treatment of illnesses contribute to the prevalence of medication mistakes in Vietnam. First, the digitalization of electronic medical records for patients in Vietnam is still in its beginnings, and the majority of medical records are still maintained on paper. This circumstance complicates the management of those records and recommended prescription in particular. The second issue stems from Vietnam's approach to regulating the drug trade.

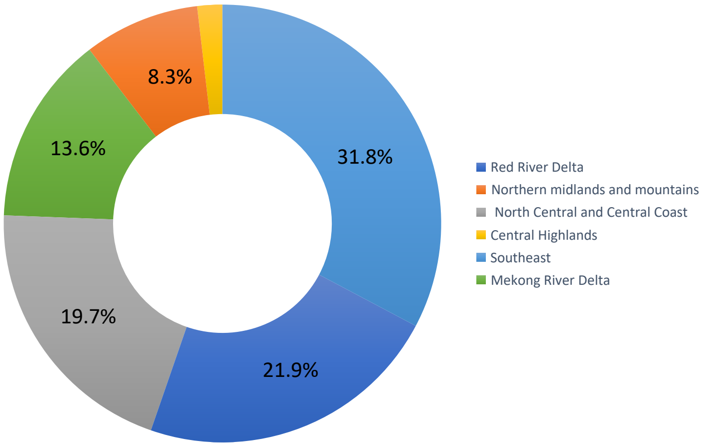


Figure 1.1: Anually ADR Report in VietNam, 2021.

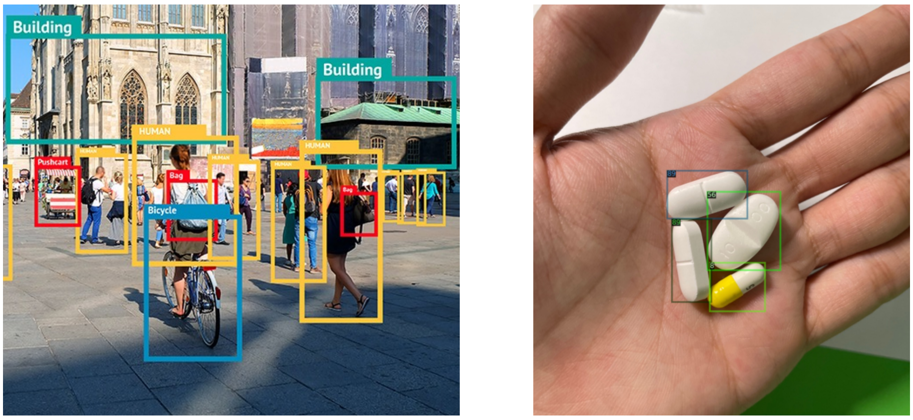
In Vietnam, it is simple to purchase medicine at any drugstore without a prescription. This results in major difficulties such as purchasing the incorrect drug, which leads to improper pharmaceutical use. To emphasize the significance of taking medication correctly, WHO has chosen the subject Medication Without Harm for World Patient Safety Day 2022 [4].

Medication errors is mainly owing to the difficulty in manually distinguishing pills owing to the wide variety of drugs and similarities in pill colors and shapes. To this end, I devote my thesis to dealing with the **Pill Detection** problem for assisting people to recognise their medications automatically with high accuracy.

1.2 Background and Research Problems

The pill recognition problem is a branch of well known Object Detection tasks, named **Intra-class Object Detection**. However, compared with the original task, there are some similarities as well as divergences, which are summarized in Table 1.1.

All differences mentioned in Tab. 1.1 can be better illustrated by Figure 1.2. In recent years, Machine Learning (ML) and Deep Learning (DL) have emerged as viable techniques for tackling general object classification problems. Some of works dealing with this problem would be mentioned in Chapter 2. All of the aforementioned discrepancies between the two tasks, however, make it difficult to properly adapt these frameworks to the pill detection problem, or, if adapt successfully, to



Object Detection

Pill Detection

Figure 1.2: Image instances for two tasks Object Detection and Pill Detection.

Table 1.1: Comparison between two tasks Object Detection and Pill Detection

	Object Detection	**Pill Detection**
Similarity	Two main objectives: ▷ **Localize** the position of objects - pills (Determine the bounding boxes contain objects) ▷ **Classify** the object lying in each bounding box	
Difference	Objects need to be classified belong to **different categories**	Pills need to be classified is in fact just **a single label** in normal Object Detection
	There is **spatial information** in images	There is no 'depth' dimension in pill images, hence **no spatial information**
	The relations between objects in images is **explicit** (The man rides bike, dog chases balloon,...)	The relations between pills in images is **implicit**, and should be formulated

maintain a reliable accuracy. This served as the first motivation for me to create an algorithm to solve the pill detection challenge.

There are some previous publications, which would be briefly discussed in Chapter 2, that did propose solution to pill detection problem. However, despite numerous efforts, this task remains problematic for current existing works, owing to following reasons.

- Firstly, **all the previous studies** dealing with this problem limit their frameworks in **recognizing only a single pill per each image**. This makes their frameworks not be highly applicable in most cases, in which patients have to take more than one pill at a time. In addition, to the best of my knowledge, there is **no publicly available dataset** of pill images that contains various pills in a single images, which can be a great hindrance for existing works.

- Secondly, **pill misidentification** often occurs with tablets that look substantially similar. Figure 2.6 shows some of the misclassification results made by a deep learning model. It can be seen that the **existing frameworks**, which just rely on visual appearances of pills, **can not distinguish pills that share almost identical shapes, colors.** The situation is even worse when taking into account other conditions of environment such as variance of light, shadow, angle of pill captures, . . .

1.3 Research Objectives and Conceptual Framework

The first objective of this work is to **construct a collection of pill pictures** according to what patients really consume. By building up this dataset, it can **serve as the foundation for solving many existing practical problems** and also open up many more research directions. For doing so, the medication captures have to satisfy some criteria. These photos may depict different types of medicines, but they must all be related. This is due to the fact that medicines taken together should not interfere with one another while treating or easing certain symptoms or conditions. In reality, there are several contraindications that prevent patients from taking certain medications together.

This research also aims to overcome the second deficiency of all previous efforts. Intuitively, there should be **no feasible solution** to the challenge of distinguishing two identical-looking tablets based just on their visual characteristics. This is true even for human experts such as pharmacists, doctors, . . . Motivated by this understanding, the purpose of this research is to **provide an additional source of information that can assist pill detection in such difficult circumstances.** In addition, I propose **a novel architecture that exploits this external knowledge** to improve accuracy and, in particular, to prevent the misclassification of tablets with identical appearances. I utilize the information retrieved from a specific collection of prescriptions as external knowledge. By using such external information, we can discover the link between the medications, such as the probability of their co-occurrence. This information will be used to improve the accuracy of the pill detection model.

1.4 Contributions

To summarize, my main contributions in this work are as follows:

- I am the first to address a so-called **contextual pill detection** problem, which recognizes pills in a picture of a patient's pill intake.

- I build the first pill captures dataset that contain the actual medications taken from real prescriptions.
- I propose a novel deep learning-based approach to solve the contextual pill detection problem. Specifically, we design a method to construct a prescription-based knowledge graph representing the relationship between pills. This knowledge graph is then exploited to improve pill localization and classification accuracy.
- I design an auxiliary loss function with a dedicated strategy to enhance the classification accuracy.
- I conduct thorough experiments on my custom dataset of drugs taken in real-world settings and compare the performance of the proposed solution to existing methods. The experimental findings indicate that my proposed model outperforms significantly the others.

1.5 Organization of Thesis

The organization of the thesis is as follows.

Chapter 1: Introduction. Covers an overview of the problem to be solved, current works and limitations, goals and directions of the solution, and finally the contributions of my project.

Chapter 2: Literature Review. Discusses about the context of the problem, as well as related studies in the field of object and pill detection.

Chapter 3: Preliminary. Presents Some fundamental concepts of convolutional neural networks, graph neural networks, which directly relate to my proposal.

Chapter 4: Methodology. Describe my proposed method for Pill Detection problem in detail.

Chapter 5: Numerical Result. Demonstrates of the dataset used, real-world experimental scenarios, baseline assessments on the built data set. Then, compares the proposed KGPNet method with other baseline models and finally discusses how effective KGPNet is.

Chapter 6: Conclusions. Presents general conclusions for the project and some future development directions.

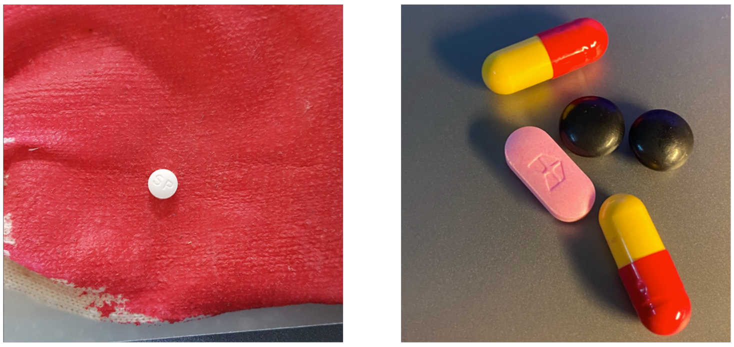
CHAPTER 2. LITERATURE REVIEW

This chapter discusses about the context - the scope of my investigating problem, as well as some core related studies in the field of object and pill detection.

2.1 Scope of research

As previously stated, all the previous studies dealing with the problem of Pill Detection limit their frameworks in recognizing only a single pill per each image [5] [6] [7]. This makes their frameworks not be highly applicable in most cases, in which patients have to take more than one pill at a time. To be more specific, these are some major shortcomings for choosing that research scope.

- The normal frameworks would require patients to take n images corresponding to n pills they takes at a single time, which is not time-efficiency.
- The pills that are visually identical can make these frameworks confuse in differentiate them, which is not accuracy-reliable.
- The work of these frameworks would only base on the input images, and in turn only one pill at a time, hence can not leverage the inter-connections between the pills.



A dataset used by some previous works

Desired dataset that mach with current scope

Figure 2.1: Illustration for pills captures that match with different research scopes.

I focus on a practical application that recognizes pills in photos of a patient's pill intake, in contrast to previous efforts. Figure 2.1 makes an illustration for medication images that match with this direction. With this new study scope, there is a great deal more information that can be used, and if it is properly resolved, the outcome framework will undoubtedly be appropriate to the reality, as it will address the deficiencies of the old scope.

2.2 Related Works

In recent years, Machine Learning (ML) and Deep Learning (DL) have emerged as viable techniques for tackling general object classification problems, as well as our pill detection task. Subsection 2.2.1 would encapsulate some details of popular Object Detection frameworks. Some works with higher correlation to Pill Detection is presented in 2.2.2.

2.2.1 Object Detection

Object Detection is one of the most fundamental and challenging computer vision tasks. Deep learning approaches have emerged as a potential method for learning feature representations directly from data and have led to notable advances in generic object detection. This section is limited to only the works targeting generic Object Detection task and following this approach.

Generic object detection, also called generic object category detection (Zhang et al. 2013 [8]), is defined as follows. Determine, given an image, if there are instances of objects from specified categories (often several categories; for example, 200 categories in the ILSVRC object detection challenge [9]) and, if present, report the spatial location and extent of each occurrence. Unlike detection of specific instances, a larger focus is placed on recognizing a broad variety of natural categories. The major challenges in targeting this task are **high quality/accuracy** and **high efficiency**. High quality detection must precisely localize and recognize objects regardless the wide variety of object categories (high distinctiveness) and the intra-class variance of object instances from the same category (high robustness). High efficiency necessitates that the full detection work be executed in real-time with enough memory and storage requirements.

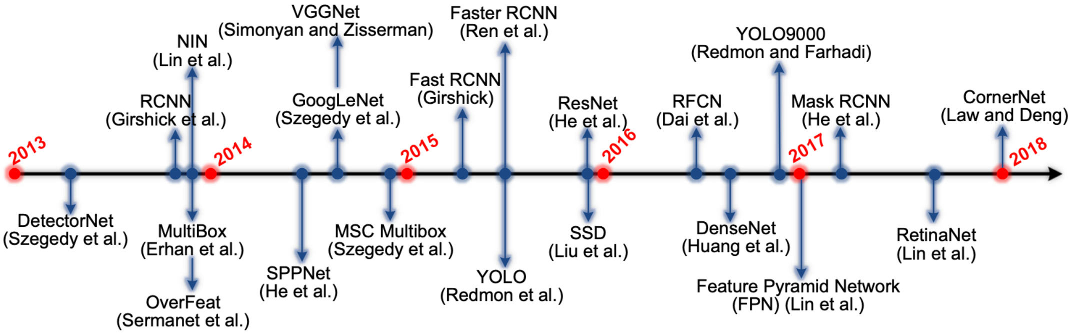


Figure 2.2: Milestones in Generic Object Detection

Figure 2.2 illustrates the evolution of Object Detection framework ever since Deep Learning entered the field, organized into two main categories:

- **Two stage detection frameworks**: An image is used to produce category-independent region proposals, CNN features are extracted from these areas, and then category-specific classifiers are used to identify the category labels of the proposals.

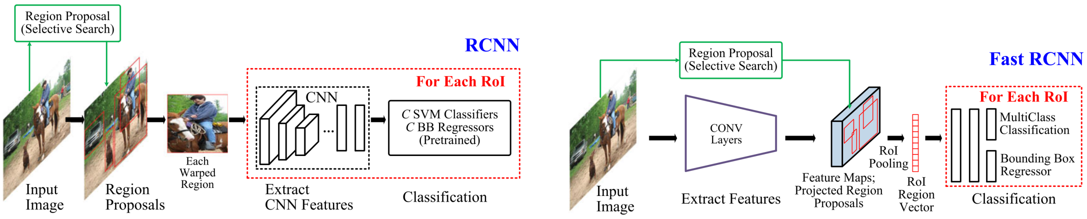
- **One stage detection frameworks**, or region proposal free frameworks: Predict class probabilities and bounding box offsets directly from entire images using a single feed-forward CNN in a homogeneous setup, without region proposal creation or post-classification.

a, Region Based (Two Stage) Detection Frameworks

This section briefly introduces some famous works following this approach, containing in Figure 2.2.

R-CNN [10]. Girshick *et al.* [10] [11] were among the first to investigate CNNs for generic object detection and developed R-CNN, which combines AlexNet [12] with a region proposal selective search [13] (Figure 2.3a). The procedures carried out by R-CNN is as follow.

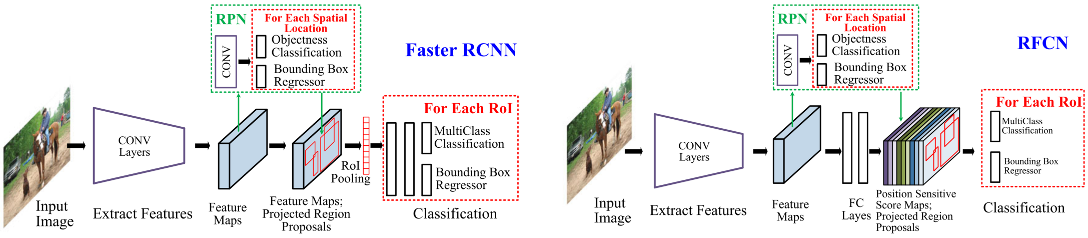
- Region Proposals Generation: Selective search algorithm is used to generate 2000 class-independent proposals in each image.
- Feature extraction for proposals: AlexNet-based CNN is fine-tuned with the warped images of size 227×227 cut from the original images corresponding to the output proposals.
- Object classifier with SVM: The features from previous stage are fed into SVM for detecting the presence or absence of an object belonging to a particular class.
- Bounding box regression: The authors include a bounding-box regression step to learn corrections in the predicted bounding box location and size.



(a) (b)

Figure 2.3: High level's diagrams of region-based (two stage) frameworks.
a - R-CNN; **b** - Fast R-CNN

Fast R-CNN [14]. Fast R-CNN was proposed to address some major shortcomings of R-CNN, while enhancing its overall performance in both speed as well as accuracy. Some majors advancements can be seen in Figure 2.3b. With a new streamlined process of training both BBox Regressor and Classifier simu-



(a) **(b)**

Figure 2.4: High level's diagrams of region-based (two stage) frameworks. **a** - Faster R-CNN; **b** - RFCN

taneously, Fast R-CNN makes allowance for an end-to-end style of training. In addition, Fast R-CNN also leverages the concept of spreading the convolution computation among region proposals. With a newly-introduced RoI Pooling operation, it is the features that get warped to fixed length, instead of the images.

Faster R-CNN [15]. Fast R-CNN, though has gone a long way to boosted up the speed over R-CNN, still has a bottleneck of using external Region Proposal module. To this end, Ren *et al.* [15] offered an efficient and accurate Region Proposal Network (RPN) for generating region proposals. Efficiently, they re-utilize the same backbone network used by Fast R-CNN module, using features from the last shared convolutional layer to achieve the task of RPN for region proposal. Architecture of Faster R-CNN is described in 2.4a.

RFCN [16]. Pointed out that Faster R-CNN still relies on a region-wise subnetwork that work with individual RoI, RFCN is proposed by Dai *et al.*, which is fully convolutional (no hidden FC layers) with almost all computations shared over the entire image. Constrast with Faster R-CNN in which the computation after RoI Pooling layer cannot be shared, RFCN introduces a shared RoI sub-network and from that RoI crops are taken from the last layer of CONV features prior to prediction. In this module, a bank of specialized CONV layers (Fig.2.4b) is used for producing position-sensitive score maps, together with modified RoI Pooling with position information being awared.

b, Unified (One Stage) Frameworks

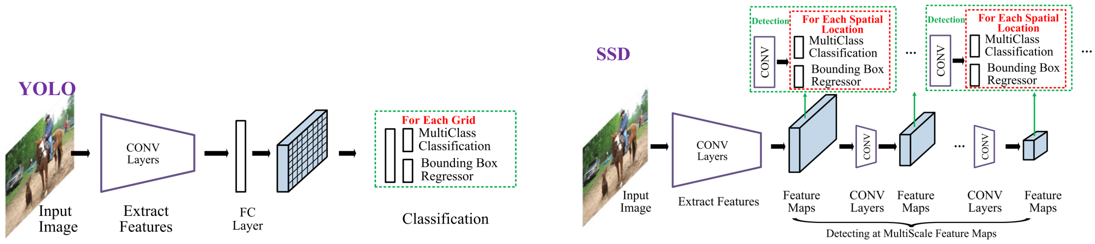
Since R-CNN [10], region-based pipeline techniques have prevailed, with frameworks utilizing Faster R-CNN [15] producing the best performance on major benchmark datasets. However, these works show their major drawback of great computational overheads, hence can not be applied to embedded systems or mobile devices. That's the main motivation for the approach of **unified** detection strategy.

OverFeat [17]. This framework can be considered as one of the first single-stage object detectors based on fully convolutional networks. The key procedures of OverFeat is as follow.

- Identification of potential object possibilities by categorizing multiscale images using a sliding window technique. OverFeat uses AlexNet backbone [12], with replacement of FC layers by 1×1 Convolutional Layer. This makes it can take an arbitrary sized images as input, unlike previous works. In addition, OverFeat leverages multiscale features to improve the overall performance.
- Major voting predictions by offset max pooling. OverFeat applies offset max pooling after the last CONV layer, results in many predictions for voting.
- Bounding box regression. Once being identified, an object is fed into a Bounding box regressor for predicting its box.
- Predictions combination. OverFeat uses a greedy merge strategy to combine the individual bounding box predictions across all locations and scales.

YOLO [18]. Redmon *et al.* introduced YOLO (You Only Look Once) framework which casts object detection as a regression problem from image pixels to spatially separated bounding boxes and associated class probabilities, illustrated in 2.5a. With the ablation of region proposal module, YOLO directly predicts detections using a small set of candidate regions. Features of entire pictures are used directly. By deviding an image into $S \times S$ grid, each predicting C class probabilities, B bounding box locations, together with confidence scores. With entire image's features as input, YOLO can greatly reduce false positive cases with background confusions, but in turn perform worse than Faster R-CNN for localization task. Aforementioned techniques are major advancements of YOLOv1, and below lists the technical improvements of other alternatives toward times.

- **YOLOv2**. This version introduce the concept of anchor box. With it, the IoU of pre-defined anchor box and the predicted bounding box can be calculated, which acts as a threshold to decide whether the probability of the detected object is sufficient to make a prediction or not.
- **YOLOv3**. YOLOv3 comprised of 75 convolutional layers without fully connected or pooling layers, resulting in a significant reduction in model size and weight. It utilized residual models (from the ResNet model) for multiple feature learning using the feature pyramid network (FPN) while retaining low inference times.
- **YOLOv4**. YOLOv4 introduced the bag of freebies (techniques that improve model performance without raising the cost of inference) and the bag of specials (techniques that increase accuracy while increasing the computation cost).



(a) **(b)**

Figure 2.5: High level's diagrams of region-free (single stage) frameworks. **a** - YOLO; **b** - SSD

- **YOLOv5**. With this most-recent advancement, the automated learning of anchor boxes is installed, and there are also changes in data augmentation and loss calculations.

SSD [19]. With the aim of preserving real-time speed without sacrificing too much detection accuracy, Single Shot Detector (SSD) (Figure 2.5b) was proposed, faster than YOLO [18] and with detection accuracy approximated that of Faster R-CNN [15]. Similar to YOLO, SSD predicts a predetermined number of bounding boxes and scores, followed by an NMS step to generate the final detection. The CNN network in SSD is fully convolutional, with early layers based on a conventional design, followed by multiple decreasingly sized auxiliary Conv layers. SSD conducts detection at several scales by acting on numerous Conv feature maps, each of which predicts category scores and box offsets for bounding boxes of appropriate sizes, as the spatial resolution of the information in the final layer should be too coarse for exact localization.

2.2.2 Pill Detection

Many studies have employed machine learning in the pill recognition challenge [5], [7], [20]. Some common techniques such as convolutional neural networks (CNN) and Graph Neural Networks (GNN) are often used.

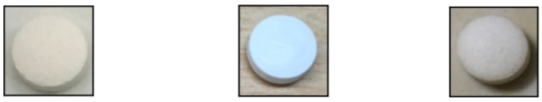
Specifically, in [5], the authors first segment the input pill image by a Manifold ranking-based method [21]. Initially, from the input image, an affinity graph is built for proximity pixels based on their color. Following, manifold ranking is performed in two stages to filter out the foreground mask. The extracted foreground image is fed into an AlexNet based network for identifying its label.

In [20], Enhanced Feature Pyramid Networks (EFPNs) and Global Convolution Network (GCN) are combined to enhance the pill localization accuracy. Besides, the authors leveraged the Xception network [22] to solve the pill recognition problem.

The authors in [7] studied how to help visually impaired chronic patients in taking their medications correctly. To this end, they proposed a so-called MedGlasses system, which combines AI and IoT. MedGlasses comprises smart glasses capable of recognizing pills, a smartphone app capable of reading medication information from a QR code and reminding users to take the medication, and a server system to store user information.

Furthermore, numerous efforts have strived to improve pill recognition accuracy by incorporating handcrafted features such as color, shape, and imprint. Ling *et al.* [23] investigated the problem of few-shot pill detection. The authors proposed a Multi-Stream (MS) deep learning model that combines information from four streams: RGB, Texture, Contour, and Imprinted Text. In addition, they offered a two-stage training technique to solve the data scarcity constraint; the first stage is to train with all samples, while the second concentrates only on the hard examples.

In [24], the authors integrated three handcrafted features, namely shape, color, and imprinted text, to identify pills. Specifically, the authors first used statistical measurements from the pill's histogram to estimate the number of colors in the pill. The imprinted text on the pill was then extracted using text recognition tools. The author also used the decision tree technique to determine the pill shape. The color, shape, and imprinted text information are then used as input features to train the classification model.



Groundtruth:	Myonal_50mg	Ayale	Betaserc_16mg
Prediction:	Betaserc_16mg	Myonal_50mg	Ayale

Figure 2.6: Ill-predicted medicines

Despite numerous efforts, pill detection remains problematic for current existing works. Firstly, all the previous studies dealing with this problem limit their frameworks in recognizing only a single pill per each image. This makes their frameworks not be highly applicable in most cases, in which patients have to take more than one pill at a time. In addition, to the best of my knowledge, there is no publicly available dataset of pill images that contains various pills in a single images, which can be a great hindrance for existing works.

Secondly, pill misidentification often occurs with tablets that look substantially similar. Figure 2.6 shows some of the misclassification results made by a deep learning model. It can be seen that the existing frameworks, which just rely on visual appearances of pills, can not distinguish pills that share almost identical shapes, colors. The situation is even worse when taking into account other conditions of environment such as variance of light, shadow, angle of pill captures, …

CHAPTER 3. PRELIMINARIES

The purpose of this chapter is to provide an introduction to some of the core ideas of convolutional neural networks, as well as graph neural networks. In addition to that, the rudimentary histories of certain networks are also discussed below. From this, the reader will have a much easier time getting caught up with the most important elements and my contributions, which will be discussed in the subsequent chapters.

3.1 Extracting Visual Features Given Input Images

This section will present some fundamental concepts as well as the inner workings of well-known modules extensively used to extract visual information from photos. In 3.1.1, Convolutional Neural Networks, which are often utilized for classification and various computer vision tasks, get first analysed. Following, 3.1.2 discusses some well-known Convolutional Network backbones that may be used to a variety of downstream visional applications.

3.1.1 Convolutional Neural Network (CNN)

Convolutional Neural Network is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data - here images. It is used for extracting the visual features given an image input, followed by some layers specifically designed for each visional problem.

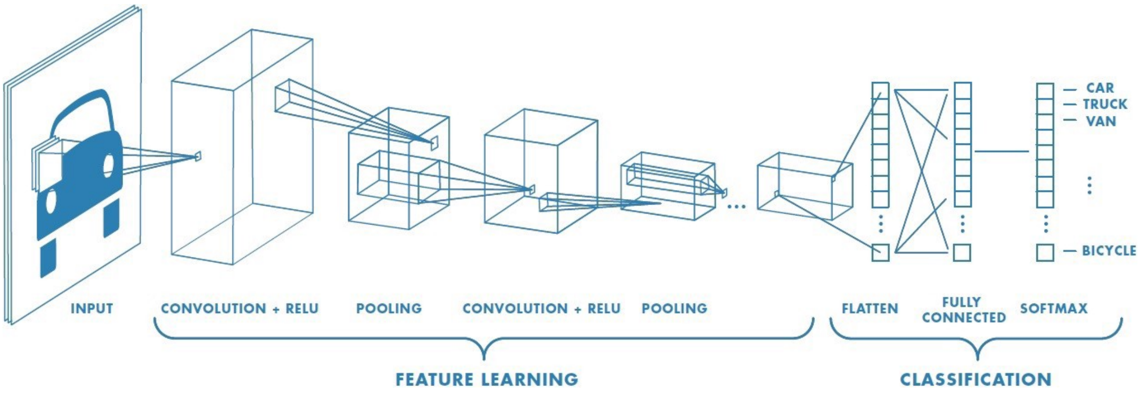
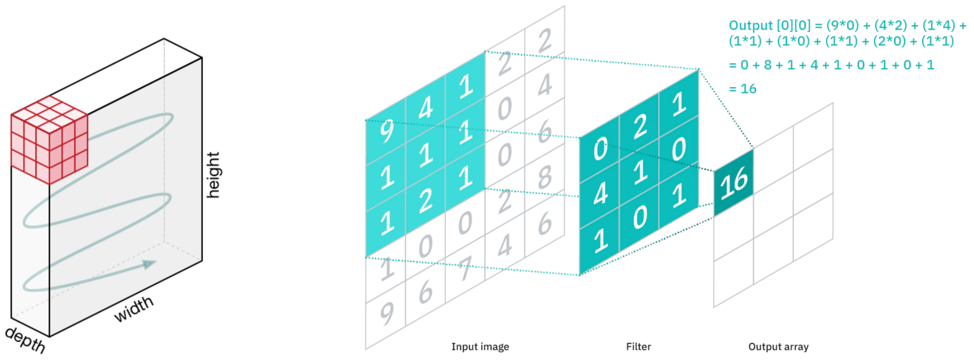


Figure 3.1: Overall flow of Convolutional Neural Networks

Figure 3.1 describes the general flow of most CNNs, together with their basic building blocks. Convolutional Networks are the combination of three main types of layers, namely **Convolutional Layer**, **Pooling Layer** and **Fully-connected (FC) layer**. With each layer added, the CNN’s complexity increases, allowing it to recognize a larger portion of the image. Earlier layers emphasize detailed characteristics containing in regions, such as colors, shapes and borders. As the visual input goes through the CNN's layers, it begins to identify bigger aspects or forms of the overall object before being used for dedicated visional tasks.

Among the three types of components, **Convolutional Layer** is the core building block of a CNN, in which the majority of computations are carried out. This layer's working is based on two major factors: mathematical `Conv` operator and a learnable set of parameters which takes the role of **Convolutional Filter** for capturing desirable visual features.



(a) **(b)**

Figure 3.2: Convolutional layer operation. **a** - Movement of the Filter Kernel over image receptive fields; **b** - `Conv` operation applied at a specific location.

Given a three-dimensional (height, width, and depth) input, the Convolutional layer's learnable filters will traverse through the image's receptive fields to extract distinctive characteristics. Specifically, the filter is initially applied to a section of the image, followed by the calculation of a dot product between the input pixels and the filter. The output array is then given this dot product. After then, the filter shifts by a stride and the procedure is repeated until the kernel has traversed the whole image. A feature map, activation map, or convolved feature is the ultimate result of a sequence of dot products from the input and the filter.

Once the underlying workings of a Convolutional layer are understood, its advantages over a fully-connected layer in image processing may be outlined as follows.

- **Local Connectivity**. Each neuron is only connected to a local region of the input volume, helping reduce the number of parameters to be learnt, while still be harmonized with the fact that only features in local regions have strong relation to each other.
- **Parameter sharing**. Intuitively, if one feature is useful to compute at some spatial position (x, y), then it should also be useful to compute at a different position (x_2, y_2). Parameter sharing scheme is used to control the number of

parameters while still achieve a good performance.

3.1.2 Modern Convolutional Networks (ConvNets)

Two contributions in 1980 [25] and 1989 [26] by Kunihiko Fukushima and Yann LeCun set the groundwork for research on convolutional neural networks. More famously, Yann LeCun employed back-propagation technique to effectively train neural networks to locate and recognize patterns within a collection of handwritten zip codes. Throughout the 1990s, he and his colleagues would continue their study, culminating in LeNet-5 [27], which used the same research ideas to document recognition. Since then, other CNN architectural variants have arisen (Fig. 3.3). Aware of the thesis's length restriction, only some well-known and outstanding studies on Convoltional Neural Networks are introduced in this section.

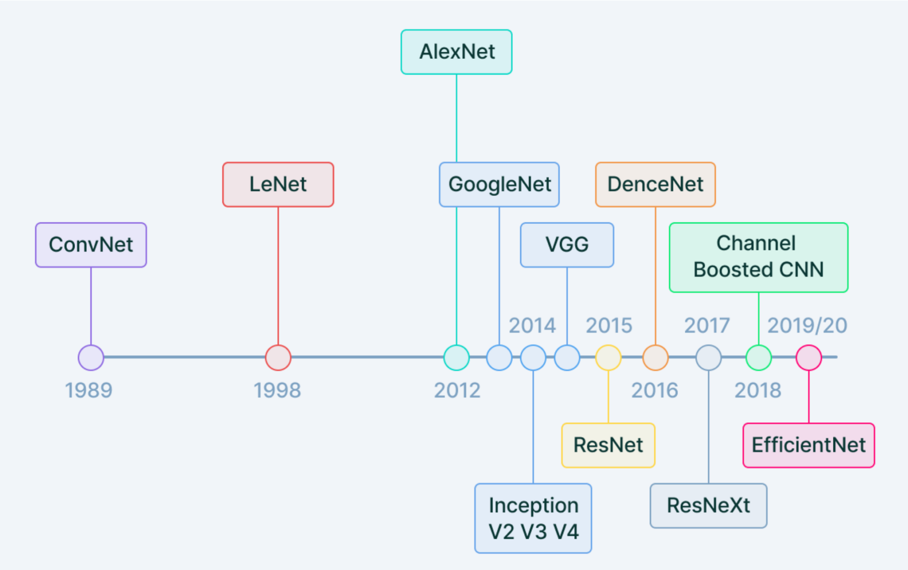


Figure 3.3: Evolution of Convolutional Networks up to present.
Image source: ***https://www.v7labs.com/blog/convolutional-neural-networks-guide***

LeNet [27]: This was the first convolutional neural network to be introduced. It was the first ConvNet that made use of back-propagation technique to optimize a deep visonal framework. LeNet was trained on 2D grayscale pictures of $32 \times 32 \times 1$ pixels. The objective was to recognize handwritten numbers on bank checks. Following two convolutional-pooling layer blocks were two fully linked classification layers.

AlexNet [12]: AlexNet was trained on the Imagenet dataset [28] with 15 million high-resolution images with shape $256 \times 256 \times 3$. ReLU activation function was utilized for the first time between convolution layers and pooling layers, together with overlapping pooling with stride window size. Five convolutional-pooling layer blocks were followed by three dense layers that were fully linked for classification.

VGGNet [29]: Instead of continually adding more dense layers to the model, VGGNet has a different strategy to increase performance. More layers of narrow convolutions were regarded more effective than fewer layers of larger convolutions, therefore the essential innovation consisted of combining layers into blocks that were repeatedly utilized in the design.

GoogleNet [30]: The idea around GoogleNet design is to widen the network horizontally instead of vertically. It consists of Inception blocks with 1×1, 3×3, and 5×5 convolution layers, followed by 3×3 max pooling with padding (so that the output has the same shape as the input) on the preceding layer, and concatenates their output.

ResNet [31]: Observations indicate that as network depth increases, accuracy becomes saturated and finally declines. Therefore, data scientists offered skip connections as a solution. These connections give an additional channel for input and gradients to flow, accelerate training, and permit the omission of one or more layers. This architecture introduced the unit of Residual block based on this insight.

3.2 Extracting Node Features Given Input Graph

In this section, some well-known graph-related tasks, as well as the concepts of graph neural networks and its inner functionality would be discussed. Subsection 3.2.1 would discuss the some fundamental graph problems. Some well-known GNN units for node embedding tasks are further discussed in 3.2.2.

3.2.1 Graph-related Problems

While deep learning efficiently uncovers hidden patterns in Euclidean data, there are a growing number of applications in which data is represented as graphs. Existing machine learning techniques face substantial difficulties due to the complexity of graph data. As graphs may be irregular, a graph may contain a variable number of unordered nodes of varying size, and nodes from a graph may have a variable number of neighbors, making it difficult to perform some key operations (such as convolutions) to the graph domain. Some well-known tasks dealing with graph data are listed in Table 3.1.

Table 3.1: Some well-known graph-related tasks.

Problem	Description
Link Prediction	The problem of predicting whether there exist an edge between two nodes
Node classification	The problem of classifying the true category for each node of a graph
Clustering & Community detection	The problem of clustering groups of nodes or graphs into clusters

While table 3.1 summarizes the major challenges in graph domain, there is a procedure in common before solving those actual downstream tasks, which is **Graph-based Embedding**. For tackling this problem, earlier works made appliance of matrix factorization [32] [33] or random walks [34]. However, they came with a major drawback of lacking generalization for unseen data, and are grouped as **transductive learning methods**. Graph Neural Network emerged as an independent research branch and was able to resolve the problem of the previous approach. Some introduction about it would be discussed in 3.2.2.

3.2.2 Some well-known Graph Neural Networks

Though earlier works was established quite early [35], [36], it was not until 2014 did the graph neural network techniques actually flourish.

First, before discussing some works of popular graph neural units, there are some basic denotations designed for graph. A graph is represented as $\mathcal{G} = (V, E)$ where V is the set of vertices or nodes, and E is set of edges. Let $v_i \in V$ to denote a node and $eij = (v_i, v_j) \in E$ to denote an edge pointing from v_j to v_i. The neighborhood of a node v is defined as $N(v) = u \in V|(v, u) \in E$. The adjacency matrix A is a $n \times n$ matrix with $A_{ij} = 1$ if $e_{ij} \in E$ and $A_{ij} = 0$ in opposite cases. A graph may have node attributes X, where $X \in R^{n \times d}$ is a node feature matrix with $x_v \in R_d$ representing the feature vector of a node v.

I choose the module GCN proposed in [37] for briefly discuss the work of ConvGNN, which is a representative and widely-applied technique branch in GNNs. First introduced by Thomas et. al. [37], one hidden layer of GCN is presented as folllow.

$$H^i = f(H^{i-1}, A), \tag{3.1}$$

in which:

- H^i is the output of $i + 1$-th layer, each H^i has the shape $n \times f^i$, where f^i is the hidden feature dimension.
- H^0 is initialized as node features of nodes X.

Function f in GCN has the formula:

$$f(H^i, A) = \sigma(D^{-1/2}\tilde{A}D^{-1/2}H^iW^i). \tag{3.2}$$

In equation 3.2, $\tilde{A}$ is the modified version of adjacency matrix A by adding self-loop connection, D is the degree matrix of A that help normalize A to reduce the effect of nodes that connect with many neighbors (nodes with high degrees).

Overall, GCN [37] is still an architecture with simple design, and hence has some major shortcomings:

- **Memory requirement**. The weight of model is updated with full-batch gradient descent. It is in harmony with the update formula mentioned above, in which the full adjacency matrix A has to be kept in memory. This cause a memory burden with a large graph and a dense-adjacency matrix.
- **Directed edges and edge features**. The proposed version of GCN only targets graphs without edge features (having binary adjacency matrix) and be undirected.
- **Limiting assumption**. The adding of $\tilde{A} = A + I$ assume that the contribution of node v_i to itself compared with its neighbor nodes are the same.
- **Transductive setting**. Also, with new nodes added to the graph, GCN have a low adaption with those and re-train process is needed to pertain performance.

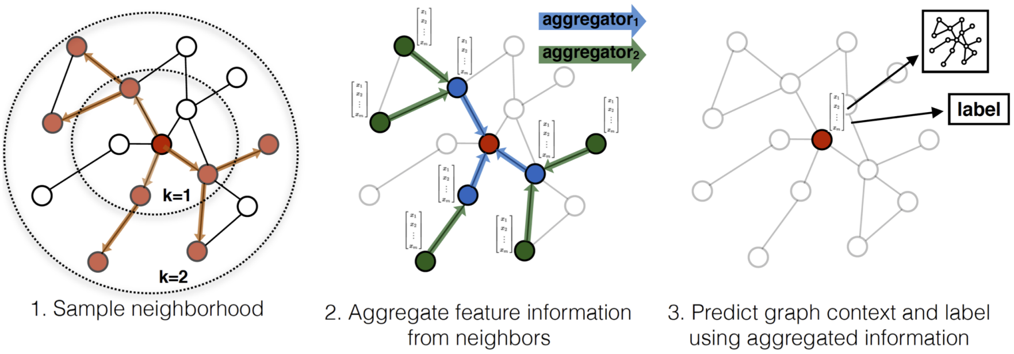

k=1
k=2
1. Sample neighborhood
aggregator1
aggregator2
2. Aggregate feature information from neighbors
label
3. Predict graph context and label using aggregated information

Figure 3.4: Visual illustration of the GraphSAGE sample and aggregate approach. *Image source: [38]*

To address those drawbacks of GCN, in 2017, a new module of GraphSAGE [38] was first introduced by William and coauthors. Figure 3.4 presents the overall idea of GraphSAGE: information of one node is aggregated based on information of its neighbors. Algorithm 1 presents more detailed inner working of GraphSAGE, which is self-explained and not covered here. This work made a foundation for the idea of Aggregation function, which is used for accumulating information in the *neighborhood* to formulate the context information for a graph node. The chosen Aggregation functions can simply be operations of *mean*, *pooling*, or a delicate networks (LSTM etc.).

Compared to previous work of GCN, GraphSAGE has some major advances:

- Be an inductive method. It well adapt with new node data.
- Be based on the intuitive and natural idea of constructing node presentation

Algorithm 1 GraphSAGE embedding generation [38]

```
Input:
▷ $\mathcal{G}(\mathcal{V}, \mathcal{E})$ - Input graph
▷ $K$ and $\mathbf{W}^k, \forall k \in \{1, \ldots, K\}$ - Number of aggregation functions used and $k$-th weight matrix
▷ $AGGREGATE_k$ - differentiable aggregator functions $\mathcal{N} : v \rightarrow 2^v$
Ouput:
▷ $\mathbf{z}_v$ - Vector representation of node $v \in \mathcal{V}$
Initialize $h_v^0 \leftarrow x_v \; \forall v \in \mathcal{V}$;
for $k = 1 \ldots K$ do
    for $v \in \mathcal{V}$ do
        $\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow AGGREGATE_k\left(\left\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\right\}\right)$;
        $\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W}^k \cdot \mathrm{CONCAT}\left(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k\right)\right)$
    end for
    $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \left\|\mathbf{h}_v^k\right\|_2, \forall v \in \mathcal{V}$
end for
$\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$
```

by neighborhood context.

- Be updated with mini-batch gradient descent and be a spatial gnn method. It resolve the major memory problem of GCN.

There are many other variances and works ([39] [40] [41]) active in the field for solving different problems or to amend shortcomings of previous works. Reasons and details of the technique being leveraged in this work would be discussed in Chapter 4.

CHAPTER 4. METHODOLOGY

In this chapter, I propose a novel pill detection framework named KGPNet (which stands for **K**nowledge **G**raph-assisted **P**ill Detection **Net**work). I first present the main components of the KGPNet in 4.1. I then describe how different graphs are designed and built based on the prescription information and visual appearances (Section 4.2). Section 4.3 explains how pill visual features are get extracted. Next, I combine the built knowledge graphs and extracted visual features to enhance pill detection performance by utilizing two modules - Adaptive Graph Generator (4.4) and Graph Transformer Module (4.4.3). Finally, I introduce an auxiliary loss to improve the effectiveness of the proposed learning model (Section 4.5).

4.1 KGPNet Overview

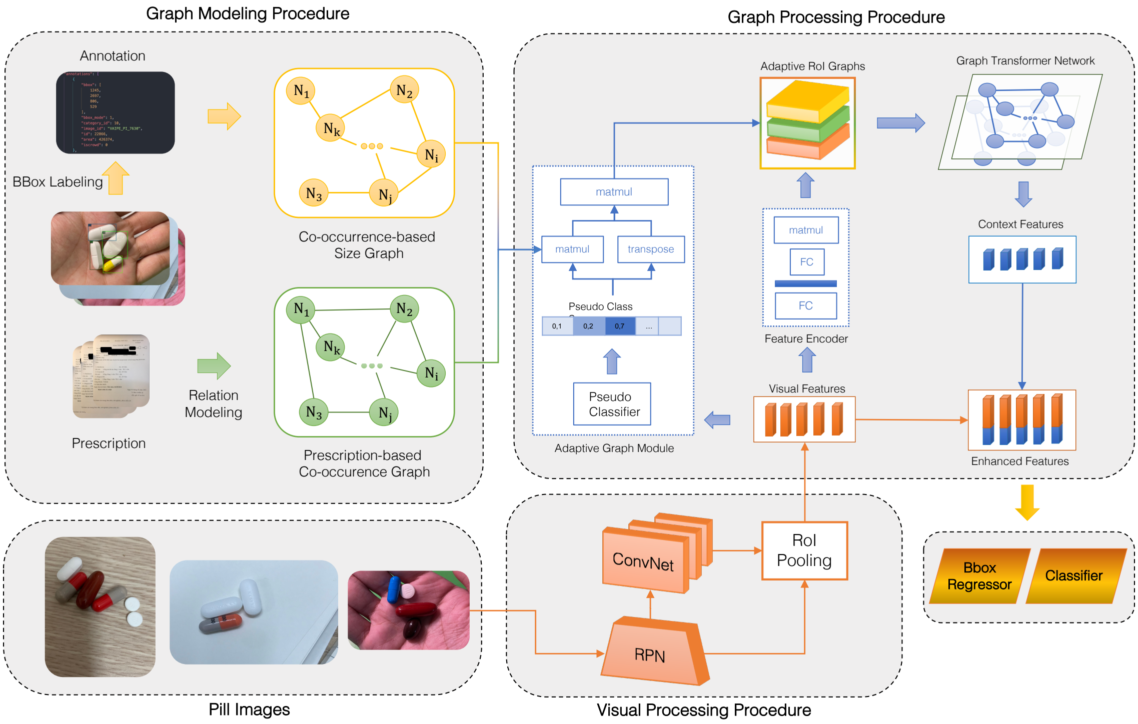


Figure 4.1: Overview of the proposed framework. First, the Graph Modeling Procedure is used to generate a non-directed Medical Co-occurence Graph (MCG), denoted as $\mathcal{G}_1 = \langle V, E_1, W_1 \rangle$, from given prescriptions, and using $\mathcal{G}_1$ together with bounding box information from the training dataset, Relative Size Graph (RSG) $\mathcal{G}_2 = \langle V, E_2, W_2 \rangle$ is then built. Second, the pill images are taken through the Visual Processing Procedure in which these images are passed via a backbone network, a Region Proposal Network (RPN) and a RoI Extractor to retrieve the visual representations for their Region of Interests. The outputs of two previous steps are the inputs for Graph Processing Procedure. At this stage, the information from MCG and PSG are first filtered to keep only what are relevant to input images. This work is carried out by Adaptive Graph Module. At the same time, RoI features are leveraged to create another graph of viusually-semantic relation. Next, all filtered graph information are transformed by Graph Transfomer Network to formulate the context presentations for each RoIs, before getting concatenated to the visual embedding to enrich the visual features. Finally, enhanced vectors are the input for the final classifier as well as box predictor.

As described in the preceding sections, I examine the pill detection problem in this study. Specifically, I focus on a practical application that recognizes pills in the patient's pill intake picture. My proposed model is shown in Figure 4.1. As input, the model gets an image of pills and provides both the bounding box and the name of each medication. To increase identification accuracy, I utilize external knowledge acquired from a specific set of prescriptions in addition to relative size information extracted from annotations of the training dataset. The intuition behind my proposal is that by utilizing a large number of prescriptions, we may learn the co-occurrence likelihood of the pills, as well as their relative size information, thereby, improve pill detection accuracy.

As illustrated in Figure 4.1, the proposed model comprises three major components: **graph modeling**, **visual processing** and **graph processing.** The first block, i.e., graph modeling, is in charge of creating two graphs: one modeling drug interactions and another representing relative size of drugs. The visual processing block is used to extract visual features of the pills - here the Region of Interests (RoIs) in each image, while the graph processing module attempts to depict the relationship between the pills and then combines the visual characteristics of the pills with their graph-based features to generate the final localization and classification decision. The overall flow is as follows.

- **Step 1 - Graphs modelling for supporting KGPNet**. I construct a graph from a given set of prescriptions, with nodes representing pills and edges reflecting drug linkages. I name this graph the *Prescription-based Medical Co-occurence Graph* or PMCG for short. Following, with the bounding boxes' coordinates information from training dataset, I can calculate the area of each box, and model the relative size ratios of all the pills in the given images (the detailed algorithm is presented in Section 4.2). This information is aggregated to formulate *Relative Size Graph* (RSG in short).
- **Step 2 - Feature extraction with Visual Processing Module**. The original image containing multiple pills is passed through a Convolutional Network for extracting visual features, and a Region Proposal Network for detecting potential region of interests. The output of two modules are fed into RoI extractor to filter out all visual presentations of pills - RoIs. It should be noted this *Visual Processing Module* follow the architecture of some well-known two-step Object Detection framework, here Faster RCNN for object detection. I let the fusion of KGPNet with one-step detector frameworks for future work.
- **Step 3 - Feature enhancement by Graph Processing Module**. The pills'

visual features, along with two general graphs MCG and RSG will then be put into the *graph processing module* to generate the beneficial context vectors. On the one hand, the visual features will be fed into *adaptive graph module* to make the pseudo classification decision. On the other hand, these features are put in a *feature encoder module*. The objective of the adaptive graph module is to make a soft decision for the labels of each RoIs, from which the mapping of two general graphs into adaptive ones dedicated for the input images can be produced. For the feature encoder, its output is then leverage to build up another visual-based graph - the third graph along with two mapped graphs. These three graphs, representing three different relations are got transformed by *Graph Transformer Network* to generate best context vectors for each RoIs.

- **Step 4 - Final predictions with enhanced features**. The RoI visual features acquired in Step 2 and the context embedding vector retrieved in Step 3 will be concatenated. The context vectors take the roles of being the presentations for neighbor pills. By observing both visual features as well as the neighborhood pills, the final Classifier module are then able to produce final prediction results.

4.2 Graphs Modelling For Supporting KGPNet

To start off, I will discuss about the first step in the working flow of KGPNet, which is graph modeling procedure. The central idea of my proposed methodology is to use external information to improve the precision of the focused task, namely, pill detection. The first realization is based on the link between medications as shown by their respective prescriptions. A prescription-based medical co-occurrence graph (PMCG) is developed for this purpose. In genuine pill captures, I suspect that all medications are given to treat or mitigate certain ailments or symptoms. Therefore, I may establish this implicit relationship by examining the direct connections between medications and diseases. This information is included on prescriptions given to patients by pharmacists. Subsection 4.2.1 describes the formulation of PMCG in depth. For the second source of information, the minimal variations in medication's forms, colors, and patterns is its main motivation. Since there are numerous varieties of medicines, which outnumber their limited visual differences and qualities, every trait is vital in recognizing them. With the current existing frameworks of two-step object detector, however, all the information about size of RoIs is diminished after going through **RoI Pooling** layer. By utilizing the information about bounding box annotation, we can formulate the relative size information, and re-merge it into the visual features for best classification accuracy. Detailed algorithm for formulate Relative Size Graph (RSG) is presented in Subsection 4.2.2.

4.2.1 Prescription-based Medical Co-occurrence Graph Modelling

This section discusses my methodology for constructing a Medical Co-occurrence Graph from a collection of prescriptions. This knowledge graph is a weighted graph, denoted as $\mathcal{G}_1 = \langle V, E, W_1 \rangle$, whose vertices V represent pill classes, and the weights W_1 indicate the relationship between the pills. With prescriptions as the initial data, two factors can be used to formulate graph edges E_1, which are diagnoses and medications. As the relationship between pills is not explicitly presented in prescriptions, I model the relation representing the edge between two nodes (i.e., pill classes) C_i and C_j based on the following criteria.

- There is an edge between two pill classes C_i and C_j if and only if they have been prescribed for at least one shared diagnosis.
- The weight of an edge E_{ij} connecting pill classes C_i and C_j reflects the likelihood that these two medications will be given at the same time.

Instead of directly weighting the `Pill-Pill` edges, I determine the weights via `Diagnose-Pill` relation. In particular, I first define a so-called `Diagnose-Pill` impact factor, which reflects how important a pill is to a diagnosis or, in other words, how often a pill is prescribed to cure a diagnosis. Inspired by the Term Frequency (`tf`) — Inverse Dense Frequency (`idf`) often used in NLP domain, I define the impact factor of a pill P_j to a diagnose D_i (denoted as $I(P_j, D_i)$) as follows.

$$\mathcal{I}(P_j, D_i) = \texttt{tf}(D_j, P_i) \times \texttt{idf}(P_i) = \frac{|\mathbb{S}(D_j, P_i)|}{|\mathbb{S}(D_j)|} \times \log \frac{|\mathbb{S}|}{|\mathbb{S}(P_i)|}, \qquad (4.1)$$

where $\mathbb{S}$ represents the set of all prescriptions, $\mathbb{S}(D_j, P_i)$ depicts the collection of prescriptions containing both D_j and P_i, and $\mathbb{S}(D_j)$ illustrates the set of prescriptions containing D_j. Intuitively, $\texttt{tf}(D_j, P_i)$ measures how often the pill P_i is prescribed for diagnose D_j, thus it reflects the significance of P_i regarding treating D_j. However, in practice, some pills are more popular among prescriptions (e.g., Sustenance, Dorogyne, Betaserc, etc.), which may cause negative bias when applying only the `tf` term. That effect can be mitigated by the term $\texttt{idf}(P_i)$.

Once finished formulating the impact factors of the pills and diagnoses, I transform each term $\mathcal{I}(P_j, D_i)$ into a probabilistic view by a simple normalization over all diagnoses as follow.

$$p(P_j, D_i) = \frac{\mathcal{I}(P_i, D_i)}{\sum_{D \in \mathbb{D}} \mathcal{I}(P_i, D)}, \qquad (4.2)$$

D is the set of all diagnoses. With $p(P_j, D_i)$, the weights between two pills $p(P_i, P_j)$ can be formulated as

$$\mathcal{W}(P_i, P_j) = p(P_i, P_j) = \sum_{D \in \mathbb{D}} p(P_i, D) * p(P_j, D), \tag{4.3}$$

where $\mathcal{W}(P_i, P_j)$ depicts the weight between pills P_i, P_j, and $\mathbb{D}$ denotes the set of all diagnoses. It should be noted that the Equation 4.3 make an assumption on the independence of two events (P_i, D) and (P_j, D).

4.2.2 Relative Size Graph Modeling

Algorithm 2 Relative Size Graph Formulation

```
Input:
▷ $\mathcal{A}$ - set of annotations for all images of training dataset
▷ $\mathcal{G}_1 =< V, E, W_1 >$ - Medical Co-occurrence Graph
Ouput:
▷ $\mathcal{G}_2 =< V, E, W_2 >$ - Relative Size Graph
Initialize $s_0 \leftarrow 1$; $s_i \leftarrow 0 \ \forall i \neq 0 \in V$;
procedure CalSize($i$ - investigating class)    ▷ Find indicators of $i$'s neighbors
    for $v_j$ in $\mathcal{N}_i$ do    ▷ Set of $i$'s 1-hop neighbors from $\mathcal{G}_1$
        if $s_j \neq 0$ then continue;
        end if
        $img_{ij} \leftarrow \mathcal{A}_{ij}$;
        $x^i_{max}, x^i_{min}, y^i_{max}, y^i_{min} \leftarrow \mathcal{A}_{ij}[i]$;    ▷ BBox annotation for class $i$ in $img_{ij}$
        $x^j_{max}, x^j_{min}, y^j_{max}, y^j_{min} \leftarrow \mathcal{A}_{ij}[j]$;    ▷ BBox annotation for class $j$ in $img_{ij}$
        $area_i \leftarrow (x^i_{max} - x^i_{min}) \times (y^i_{max} - y^i_{min})$;
        $area_j \leftarrow (x^j_{max} - x^j_{min}) \times (y^j_{max} - y^j_{min})$;
        $s_j \leftarrow s_i \times \frac{area_j}{area_i}$;
        CalSize($j$)    ▷ Recursively this calculation for node $j$
    end for
end procedure
CalSize(0)    ▷ Spread the graph from initial node 0
for $v_i$ in $V$ do
    for $v_i$ in $V$ do
        if $s_i \neq 0$ and $s_j \neq 0$ then
            $E \leftarrow E \cup \{e_{ij}\}$    ▷ Create an edge between $i$ and $j$
            $w_{ij} \leftarrow \frac{s_i}{2s_j} + \frac{s_j}{2s_i}$    ▷ Average 2 ratios for symmetricity
            $W_2 \leftarrow W_2 \cup \{w_{ij}\}$
        end if
    end for
end for    ▷ Finish formulating $\mathcal{G}_2$
```

This subsection describes my methodology for producing the Relative Size Graph. Let RSG be denoted by $\mathcal{G}_2 = \langle V, E, W_2 \rangle$, in which set of vertices V and edge E are the same as PMCG graph $\mathcal{G}_1$. The reason for this similarity is owing to the fact that both graphs present the relationship between pills - the label, hence share V. Moreover, the relative size between two pills can only be formulated if they are in the same images, which, in turn, be in the same prescription, hence the edges set E is also shared. W_2, however, is different as compared to W_1, since it represents the relationship about relative sizes between pills.

As mentioned before, the main information source for building Relative Size Graph is the bounding boxes annotations of the training dataset. Since the positions of cameras for different images are not identical, the actual size of each bounding box can not be directly used. I instead create a new value for size normalized representation, called size indicator, denoted by s_i for each class i. By setting a initial value $s_0 = 1$ for class 0, I can traverse through all 1-hop neighbors v_i of Node v_0 in the graph $\mathcal{G}_1$, and recursively calculate all indicators s_i. The edge between two node v_i and v_j is then can be calculated by the ratio between two indicator s_i and s_j. Detail formulation is presented by Algorithm 2.

4.3 Visual Processing Module

In the following, the procedure of feature extraction for locations of interest is presented. It is the responsibility of the Visual Processing block. In order to accomplish this, this block must be able to distinguish what should be the major focus and extract the characteristics associated to that region. For this purpose, I utilize components from a conventional 2-step object detector architecture. My Visual Processing module uses Faster RCNN, to be precise [15]. From Figure 4.1, it can be seen that this blocks consists of three submodules: a Convolutional Network, a Region Proposal Network and a RoI Pooling Layer.

For selecting ConvNet architecture, there are various candidate modules to choose from such as VGG [29] or ResNet [31] to extract the visual features. ResNet-50 [31] is currently being utilized for all experiments. I personally choose this version among various alternatives because it well balances between the complexity and accuracy for this pill detection task. After passing an image through this feature extractor, I receive a 4096-dimensional feature vector. By forward propagating a typical picture through five convolutional blocks and one fully connected layer, features are generated (Figure 4.2). I encourage readers referring to [31] for more network architecture details. Besides, the remaining two modules are taken from the original framework Faster R-CNN [15]. Region Proposal Network (RPN) (Fig-

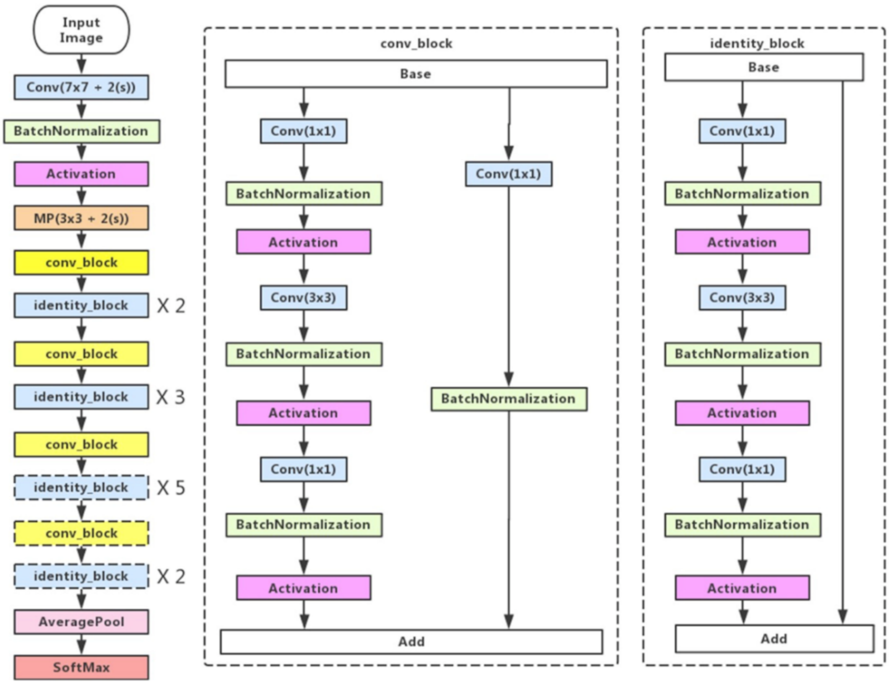


Figure 4.2: (Left) ResNet50 architecture. The last Softmax is discarded in my feature extractor. **(Middle)** Convolution block which changes the dimension of the input. **(Right)** Identity block which will not change the dimension of the input.
Image source: Optimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images [42].

ure 4.3) is a fully convolutional network that takes the visual feature vector from previous module and generates proposals with various scales and aspect ratios. Rather than using **pyramids of images** or **pyramids of filters**, RPN make use of k anchor boxes. An anchor box is a reference box of a specific scale and aspect ratio. With multiple reference anchor boxes, then multiple scales and aspect ratios exist for the single region. This can be thought of as a pyramid of reference anchor boxes. Each region is then mapped to each reference anchor box, and thus detecting objects at different scales and aspect ratios.

The last layer in Visual Processing module is a RoI Pooling layer. It simply works by splitting each region proposal into a grid of cells. The max pooling operation is applied to each cell in the grid to return a single value. All values from all cells represent the feature vector. If the grid size is 2×2, then the feature vector length is 4. The reason for this operation is to extract a fixed-length feature vector from each region proposal. With this process, the latter module of KGPNet can parallelize the computation and leverage the connections amongs RoIs for enhancing classification accuracy. However, as previously mentioned, this is the main reason for diminishing size information of pills. Relative Size Graph proposed in 4.2.2 help alleviate this issue.

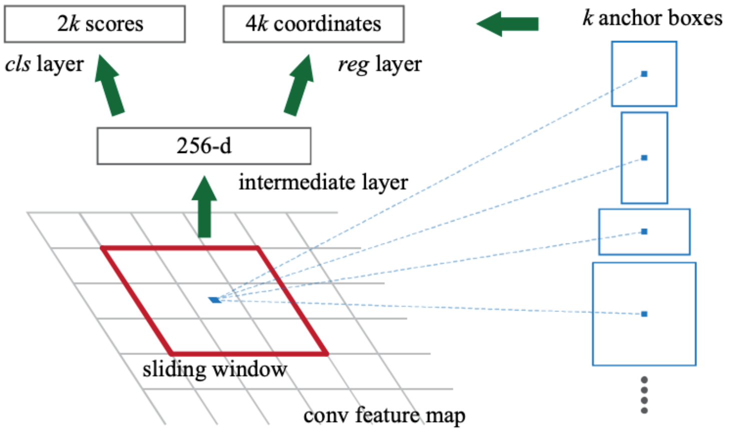


Figure 4.3: Region Proposal Network (RPN).
Image source: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [15]

4.4 Graph Processing Module

This section would cover my major contribution in the architecture of KGPNet - Graph Processing Module. It comprises of three sub-components that works for different purposes. The first one is Adaptive Graph Module, which is responsible for extracting adaptive graphs of RoIs from the original MCG and RSG graphs. Details about it would be discussed in 4.4.1. Following, a visual-based graph is generated with the aid of a simple feature encoder, presented in 4.4.2. Lastly, section **??** discusses about Graph Transformer module, which learn the best context presentations for enhancing information of each RoI.

4.4.1 Adaptive Graph Module

After defining the generic knowledge graphs of MCG and RSG, this module is derived intuitively from a realization. Specifically, while addressing a specific picture - a query, KGPNet - a student should be able to choose which portion of the two graphs - two reference books - should be consulted. This concept is natural and closely related to the motivation of **Attention mechanism** [43], which has its origin from the field of Natural Language Processing (NLP).

This module's primary component is a Pseudo Classifier, which provides approximate classification results using solely visual features of RoIs. These temporary identification results are then utilized as a filter layer to pull from the MCG and RSG graphs just information pertaining to the pills in the image (and omitting information from the nodes that are not associated with the pills in the picture). Effectively, the pills in my dataset may be divided into two categories: simple samples and difficult samples, illustrated in Fig.4.4. Pseudo Classifier can readily recognize

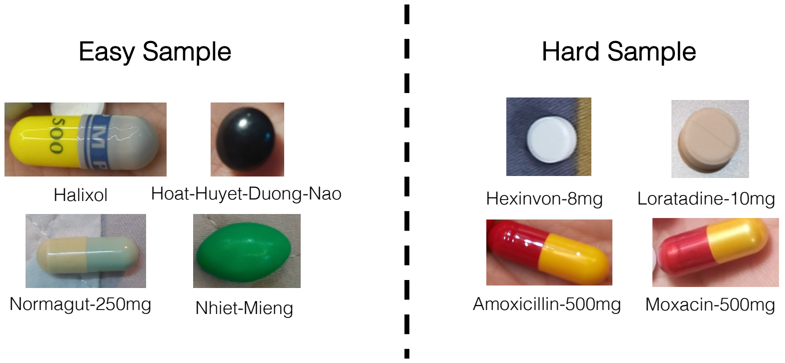


Figure 4.4: Examples of easy and hard samples.

the former since they possess distinguishable visual characteristics. However, the latter require extra information about nearby tablets to help in their recognition. Using only the visual-based Pseudo Classifier, I am able to filter out the majority of the simple ones and used them as context pills for recognizing the remaining hard ones.

In current implementation, pseudo classifier is straightforwardly implemented as a fully connected layer. Having attained the results of this module, a composites of simple matrix multiplications can be applied to extract the sections of original knowledge graphs that need to be focused. Let N be the number of pill classes and M be the number of pills in the input image. Suppose $P = [p_{ij}]_{i=\overline{1,M};j=\overline{1,N}}$ is the matrix whose row vectors represent the logits produced by the pseudo classifier, and $\mathcal{A}_1 = [a^1_{kl}]_{k=\overline{1,N};l=\overline{1,H}}$, $\mathcal{A}_2 = [a^2_{kl}]_{k=\overline{1,N};l=\overline{1,H}}$ denote the weighted adjacency matrices for MCG $\mathcal{G}_1$ and RSG $\mathcal{G}_2$, respectively. The adaptive adjacency matrices, denoted as $\tilde{\mathcal{A}}_1$ and $\tilde{\mathcal{A}}_2$ are matrices of size $M \times M$, each row depicts the condensed relational information of a pill - a specific RoI with others in the input image. $\tilde{\mathcal{A}}_i$ is calculated by performing a composition of matrix multiplications as follows.

$$\tilde{\mathcal{A}}_i = \sigma(P) \cdot \mathcal{A} \cdot \sigma(P)^T. \tag{4.4}$$

Here the symbol σ denotes the `Softmax` activation function. Intuitively, the i-th row of $\tilde{\mathcal{A}}_i$ is a weighted sum of all the $\mathcal{G}_i$'s adjacency matrix, whose weights are the classification probabilities corresponding to the i-th pill in the input image.

4.4.2 Visual-based Graph Formulation

After utilizing external knowledge graphs to help in pill recognition, another realization can be thought of is to exploit the relation contained in the visual appearances themselves. Despite the fact that medications appear to be arbitrarily shaped or colored, several medical studies instead demonstrate an actual correlation between their aesthetics and their efficacy or active constituents [44][45][46].

With this motivation, I also make use of RoIs' visual features for creating the third graph which models the visually semantical relationship among pills in the input image. All visual features are first employed through a simple non-linear function $\mathcal{F}$: $\mathbb{R}^{H} \rightarrow \mathbb{R}^{H^{'}}$ for bringing them from original H-dimensional space into a $H^{'}$-dimensional latent one in which their relations can be best presented. The output latent vectors are then directly used for calculating the correlations between RoIs as follow.

$$w_{ij}^{3} = z_i \cdot z_j, \tag{4.5}$$

in which w_{ij}^{3} denotes the **visual-based** weight between two RoIs, z_i and z_j are latent presentations of i-th and j-th RoI, respectively. When performing a matrix multiplication instead of above pair-wise product, I can achieve the weighted adjacency matrix W_3 for the third graph $\mathcal{G}_3$, which is densely populated and its weight values indicate the relavances of RoIs visually.

After this module, KGPNet have extracted three adaptive graphs' relations $\tilde{\mathcal{A}}_i - i = \overline{1,3}$. Since all three possess the same set of nodes $\tilde{V}$, which are the regions of interest (RoIs) of the input pictures, it can be interpreted as a single heterogeneous graph $\tilde{\mathcal{G}}$ with one node type and three edge types corresponding to three distinct senses of relationship.

4.4.3 Graph Transformer Module

The weighted adjacency matrices extracted from $\mathcal{G}_1$, $\mathcal{G}_2$ or $\mathcal{G}_3$ can not be easily **digested** and combined with RoIs' visual features without an explicit form of aggregation. In addition, the vectors should be well designed for each RoI, since the effects of different neighbors to a particular pill are not identical. Intuitively, there should be a module capable of accumulating these previously constructed relations together with information representative for each RoI in order to construct the context vector that best describes the surrounding environment for the pill under consideration. Graph Transformer Network (GTN) is the final module in charge of creating such vectors that are best suited for RoIs.

Before analyzing GTN, it is crucial to figure representative attributes for each RoI in order to generate the most relevant context vectors. Using the retrieved RoI visual features to depict the relevant RoIs is the most natural solution. However, earlier researches[6][47] and experimental findings indicate that this option is not particularly advantageous. There are several factors account for this behavior, listed below.

- In the circumstances of heavy occlusions and ambiguities, the visual features are not reliable.

- In the dataset, different images have different light conditions, camera angles, zoom levels, which, in turn make an $intra-variance$ in visual features of one class.
- Two pills with identical appearances would in turn having similar visual features, hence not representative.

In these previous works [6][47], the authors also provide a simple yet effective alternatives to the visual features. The weights of the classifier for each category actually include high-level semantic information since they represent the feature activation learned from all pictures. The weight is tuned with the aim of producing correct classification results regardless the variances of input features, so can be a more stable option. Formally, let $W \in \mathrm{R}^{H\times C}$ denote the weights of the previous classifiers (parameters) for all the C categories. The representative vectors of RoIs batch can be obtained by copying the parameters W from the previous classification layer in the bbox head of the detection networks, then multiplied with the output $P \in \mathrm{R}^{M\times C}$ of Pseudo Classifier.

$$W_{RoI} = P \cdot W^T. \tag{4.6}$$

Note that the classifiers are updated in each iteration during training so that the depictive features W_{RoI} becomes more accurate from time to time. Furthermore, this approach enable my model to be trained in an end-to-end style. Graph Trans-

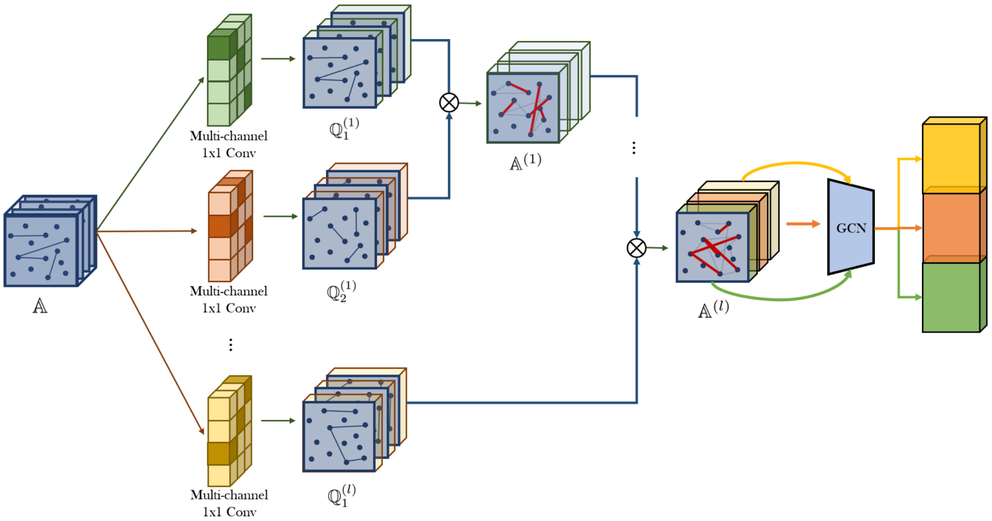


Figure 4.5: Graph Transformer Network (GTN) architecture. GTN softly selects adjacency matrices (edge types) from the set of adjacency matrices $\mathbb{A}$ of adaptive heterogeneous graph $\tilde{\mathcal{G}}$ and learns new meta-path graphs represented by $\tilde{\mathbb{A}}$ via the matrix multiplication of two selected adjacency tensors $\mathbb{Q}_1$ and $\mathbb{Q}_2$. The soft adjacency matrix selection are weighted sums of candidate adjacency matrices obtained by C channels of 1×1 convolution with non-negative weights with `softmax` activation. Image source: *Graph Transformer Networks* [48]

former Network (GTN) is the aggregator for generating context vectors correlative to all RoIs. It should be able to learn the node embeddings of graphs with following characteristics: being heterogeneous in nature and having adaptive graph structures.The heterogeneity of the input graph coincides with that of my adaptive graph $\tilde{\mathcal{G}}$. The latter criterion must be met since the input graphs for images are varied and the graph formed for a particular image during one iteration is not identical to the graph generated for the same image during subsequent iterations because the learning process is ongoing. For this module, I leverage the architecture proposed in [48]. It is particularly suited for scenarios involving these aforementioned criteria.

GTN's fundamental concept is to produce new graph structures and simultaneously learn node representations on the learnt graphs. GTNs seek alternative graph structures utilizing various candidate adjacency matrices to execute more effective graph convolutions and learn more potent node representations, as opposed to the majority of CNNs on graphs, which assume the graph is supplied.

Figure 4.5 describe the detail architecture of GTN. GTN softly selects two stacks of graph structures $\mathbb{Q}_1$ and $\mathbb{Q}_2 \in \mathbb{R}^{M \times M \times C}$ from candidate adaptive adjacency matrices $\mathbb{A}$. For doing so, it computes the convex combinations of adjacency matrices by a C-channel 1×1 convolution as in Fig. 4.5 with the weights from `softmax` function as:

$$\mathbb{Q} = F(\mathbb{A}; W_\phi) = \phi(\mathbb{A}; \sigma(W_\phi)), \tag{4.7}$$

where ϕ is the convolution layer and $W_\phi \in \mathbb{R}^{C \times 1 \times 1 \times K}$ is the parameter of ϕ. Second, it learns C new graph structure by the composition of two stacked relations (i.e., matrix multiplication of two adjacency tensors, $\mathbb{Q}_1 \cdot \mathbb{Q}_2$). The output graphs result $\tilde{\mathbb{A}}$, together with RoIs' representative features W_{RoI}, is then used as the input for normal Graph Convolution Network (GCN) to produce final node presentations - our desired context vectors. These vectors are directly concatenated with their corresponding RoIs' visual features, before getting fed into both Bounding Box Regressor and Classifier to enhance the results of both tasks.

4.5 KGPNet's Objectives and Losses

This section presents the details about my model's objectives and the corresponding losses to achieve those goals. Subsection 4.5.1 covers the losses which are widely used in 2-step detectors and also in KGPNet. My auxiliary loss, which bases on co-occurrence information, is described in 4.5.2.

4.5.1 Two-step Object Detectors' Losses

a, Region Proposal Network's Losses

The loss for RPN consists of two components: classification loss combined with bounding box regression loss.

$$\mathcal{L}\left(\{p_i\},\{t_i\}\right) = \frac{1}{N_{\text{cls}}}\sum_i \mathcal{L}_{\text{cls}}\left(p_i, p_i^*\right) + \frac{\lambda}{N_{\text{box}}}\sum_i p_i^* \cdot L_1^{\text{smooth}}\left(t_i - t_i^*\right), \tag{4.8}$$

in which

$$L_1^{\text{smooth}}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{4.9}$$

In this composite loss function, p_i, p_i^* are the predicted probability of anchor i being an object and the ground truth label whether anchor i is the object respectively. The $\mathcal{L}_{cls}$ is again a **log loss** function with 2 classes – sample is the target object versus not. The regression loss uses a smoothing $L1$ function. Here t_i and t_i^* are the differences of four predicted coordinates and the ground truth coordinates with the coordinates of the anchor boxes, respectively. The N_{cls} is a normalization term set to the mini-batch size and the N_{box} is also a normalization term set to the number of anchor boxes. The λ is set to 10, which is a balancing parameter such that $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{box}}$ are weighted equally.

b, Output's Losses

KGPNet's final results consist of predicted labels for RoIs and modifications to the bounding boxes provided by RPN. Due of this, there are two distinct losses associated with these outcomes. While the loss for a bounding box regressor is equal to that of RPN network, the classification loss is instead the cross entropy loss for multilabel.

$$\mathcal{L}_{cls}^{out} = -\sum_{i=0}^{C} p_i^* log(p_i) \tag{4.10}$$

4.5.2 Triplet Co-occurrence Enhancement Loss

Motivationally, the frequency with which distinct pills co-occur should influence KGPNet's behavior. Specifically, if the framework can identify the existence of pill A in the image with a high degree of confidence, it should also make assumptions about the appearance of A's neighbors based on the Medical Co-occurrence Graph $\mathcal{G}_1$.

I suggest the usage of an auxiliary loss called Triplet Co-occurrence Enhancement Loss for this reason. Given that A is a groundtruth pill in the picture, this loss aims to adjust the output of Pseudo Classification by motivating the probability of A and its neighbors produced by this layer.

Let's denote the i-th Region of Interest be r_i with its corresponding actual label be l_i. The set of top $(k+1)$ **closest** and **furthest** neighbors of r_i are N^i_{pos} and N^i_{neg} in that order. Here, **closest** neighbors indicates nodes which have connections - edges of highest weights with l_i, while the term **furthest** denotes the opposite. For N^i_{pos}, the groundtruth labels set is $L_{pos} = \{l^0_{pos}, l^1_{pos}, \dots, l^k_{pos}\}$, and $L_{neg} = \{l^0_{neg}, l^1_{neg}, \dots, l^k_{neg}\}$ is label set of N^i_{neg}.

The objective for the i-th RoI is as

$$\begin{aligned}\mathcal{L}^i_{aux} &= p_i(l_i)p(N^i_{pos}) - (1 - p_i(l_i))p(N^i_{neg}) \\ &= p_i(l_i)\sum_{j=0}^{k}[1 - \prod_{m=0}^{M}(1 - p_m(l^j_{pos}))] - (1 - p_i(l_i))\sum_{q=0}^{k}[1 - \prod_{n=0}^{M}(1 - p_n(l^q_{neg}))],\end{aligned} \tag{4.11}$$

and for all RoI is

$$\mathcal{L}_{aux} = \sum_{i=0}^{M} \mathcal{L}^i_{aux}. \tag{4.12}$$

In Eq.4.11 and 4.12, M is the total number of RoIs in the image, p is the output after going through `softmax` activation of logits produced by Pseudo Classifier.

The objective during the training process is to **maximize** the quantity $\mathcal{L}_{aux}$, which in turn maximize each **positive** term $p_i(l_i)p(N^i_{pos})$ while minimize the **negative** opposition $(1 - p_i(l_i))p(N^i_{neg})$.

CHAPTER 5. NUMERICAL RESULTS

This chapter covers the experimental results of the proposed KGPNet architecture for the Pill Detection challenge. In addition, the performance of KGPNet is compared to existing approaches in the field of object detection as well as to another study that applies an external Knowledge Graph. In addition, various ablation experiments are offered to provide a better understanding of the impact of each module on KGPNet's overall performance.

5.1 Custom Dataset

To the best of my knowledge, previous studies working on Pill Indentification problem only limited their works on the dataset captured in the laboratory, with limited environmental conditions (light, angle, zoom level, . . .) (*NIH Dataset* [49]); and there is only one pill per a single image (*CURE Dataset* [50]). Due to those reasons, these datasets do not greatly correlate with the reality, in which patients actually take arbitrary number of drugs, and those drugs are supporting each others to cure some symptoms. This, in turn, make the existing models less appropriate to actual real-world medications photos captured by patients.

Table 5.1: Comparison of existing pill images datasets (CURE and NIH) with my custom VAIPE PP dataset.

	NIH	CURE	VAIPE PP
Number of pill images	7000	8973	9426
Number of pill categories	1000	196	96
Instance per category	7	40-50	> 30
Illumination conditions	1	3	> 50
Backgrounds	1	6	> 50
Number of prescriptions	0	0	1,527

As a matter of fact, there is also no publicly available dataset of these pills images, in which the pills follows intakes of actual patients. Motivated by this realization, this study, as a part of **VAIPE** - a project aiming at developing a protective healthcare monitoring and supporting system for Vietnamese, dedicates to build a set of large-scale open data containing prescription images and prescription drug images, called VAIPE PP.

In 2021 and 2022, 1527 prescriptions were obtained from anonymous patients at 4 major institutes in Vietnam, treating a variety of diseases. After carefully examining the data and patients' privacy, each prescription is carefully labeled under the support of Vietnamese optical character recognition (OCR) models. Afterwards, the medications are purchased in accordance with the relevant prescriptions, then images are taken following the correct tablets for each usage throughout the day. Specifically, prescriptions will mostly be separated by timeslots to take the medicines (**morning**, **noon afternoon** and **evening**). For each slot, patients should only take a subset of pills contained in the prescriptions, corresponding to the amount assigned. The images are taken following those timeslots in prescriptions, in many different contexts (various backgrounds, lighting conditions, in-hand or out-of-hand, . . .) and by different smartphones. For each prescription, the number of images taken is about $5 - 10$, and the total number of drug images collected was 9426 pill images with 96 independent drug labels.

Table 5.1 summarizes the details meta-data about three datasets of NIH, CURE, VAIPE PP. As it suggests, compared to two previous datasets, VAIPE PP is constructed following a much more flexible procedure, with fewer restrictions. Owing to this reason, VAIPE PP has a great generalization, and can serve as a reliable data source for training **generic pill detectors**.

5.2 Evaluation Methodology

5.2.1 Evaluation Metrics

For assessing KGPNet as well as other backbones and related works's performances, COCO mAPs metrics are currently being adopted. This set of metrics is widely used for evaluating the works of object detection as well as segmentation problem.

Mean Average Precision (mAP), as its name suggest, is the mean of Average Precision (AP) over all classes - calculated over recall values from 0 to 1.

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i. \tag{5.1}$$

Average Precision (AP) is the area under the Precision-Recall curve, calculated for one class, at a given IoU threshold. It should be noted that for COCO Evaluator, it make no distinction between AP and mAP and assume the difference is clear from context. From now on, AP and mAP are used interchangeably. IoU - Intersect over Union is a factor to evaluate the fitness of predicted bounding boxes compared with the groundtruth ones, computed by taking the ratio between intersected area over the union area of the two boxes. By setting an IoU threshold, it can clarify whether a predicted instance is True Positive (TP) or False Positive (FP), hence affect the AP results. Table 5.2 summarizes the set of COCO APs evaluation metrics. The main metric is AP. Since AP is averaged over multiple Intersection over Union

Table 5.2: COCO APs evaluation metrics set.

	Metrics	Description
Average Precision (AP)	AP	Average AP at IoU thres = $\overline{0.5, 1.0}$, step 0.05 (COCO metric)
	AP50	AP given IoU thres = 0.5 (PASCAL VOC metric)
	AP75	AP given IoU thres = 0.75 (strict metric)
AP Across Scales	APs	AP for small object : area $< 32^2$
	APm	AP for medium object: $32^2 <$ area $< 96^2$
	APl	AP for large object: area $> 96^2$

(IoU) values, this metrics rewards detectors with better localization.

5.2.2 Evaluation Scenarios

a, Comparison benchmarks

As indicated before, the baseline with which KGPNet presently integrates is Faster R-CNN [15]; hence, the original framework is also utilized for comparison. There are to Faster R-CNN alternatives being installed: one with single Convolutional features C4 and the other with FPN's characteristics. Also, the most representative framework that also utilize external knowledge graph for Object Detection task [6] is also adopt for this research task to compare with KGPNet.

b, Hyper-parameters settings

Below list describes the settings applied to all benchmarks, as well as KGPNet framework.

- For all baselines as well as the related work used for comparison, the input settings are provided the same as the requirements of those frameworks (Faster R-CNN [15] requires only images, etc.), while other parameters are tuned for their best performances.
- For KGPNet, a fixed set of hyper-parameters is used throughout all experiments. The chosen set is not guaranteed to produce the best results, but can still illustrate KGPNet's robustness over other methodology in dealing with Pill Detection problem.

c, Train and evaluation settings

Table 5.3 provides information on the training and evaluation procedures shared by all frameworks. The selected training approach of 20000 iterations with a batch size of 16 is based on the experimental finding when the majority of object detector frameworks get converged. All frameworks initial starting points are the weights

Table 5.3: Details about training and evaluation processes' hyper-parameters.

Training Procedure		Evaluation Procedure		Train Iterations	Batch Size
Prescriptions	Images	Prescriptions	Images		
1527 (100%)	7514 (78%)	0 (0%)	1912 (22%)	20000	16

achieved by pre-training them with COCO 2017 dataset [51].

5.2.3 Implementation Details

In my KGPNet implementation, the dimensions of node embeddings are set as 64. The Graph Transformer Module has only one layer, with number of channels set as 10. The optimizer used is AdamW [52], and the initial learning rate is 0.001. During the training process, the input images are resized so that the shortest edges have the size of 800, with a limit of 1333 on the longer edge. The ratios are kept the same as the original images. If max size is reached, then downscale so that the longer edge does not exceed 1333. For augmentation, simple random horizontal and vertical flips are installed. All the implementation is performed with the help of **Pytorch** framework, and the training, as well as evaluation processes, are conducted with 2 NVIDIA GeForce RTX 3090 GPUs.

All models installed are trained in an end-to-end style until the maximum number of iterations (mentioned in Table 5.3) reached.

5.3 Experimental Results of KGPNet

This section would discuss about the actual performance of KGPNet, together with some baselines and other related works.

5.3.1 Comparison with Object Detection framework

Since the current installation of KGPNet is associated with Faster R-CNN architecture [15], a detailed comparison between KGPNet and this baseline is carried out. Two alternatives of feature extractor backbone are used: ResNet-50-C4 and ResNet-50-FPN.

Table 5.4 presents the numeric results of KGPNet and Faster R-CNN with two different backbones, and Fig. 5.1 makes a visualization from those results. In both circumstances, KGPNet show its superior over Faster R-CNN by large performance gaps for all metrics being used. Specifically, with the use of single feature map produced by convolution block C4 in ResNet-50 backbone, the average precision AP of Faster R-CNN is 62.6654, while that of KGPNet is 68.3751 (9.2% enhancement). In addition, when replace C4's feature map with multi-scale feature maps produced by Feature Pyramid Network (FPN) [53], similar observation can be drawn, that

KGPNet make an improvement of 9.4% for overall AP metrics.

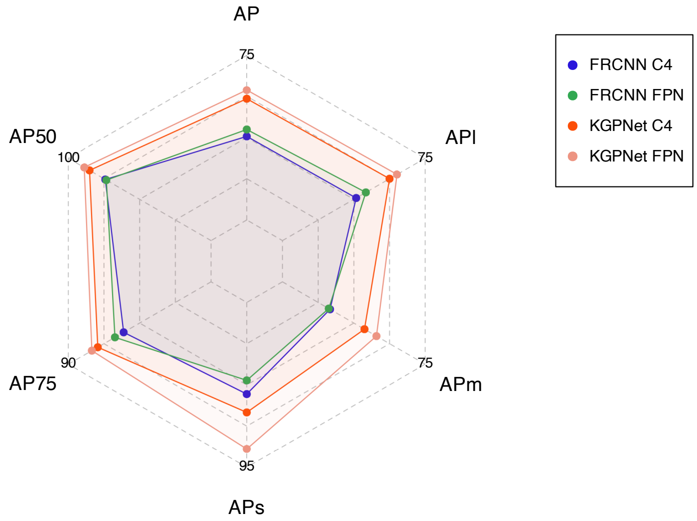


Figure 5.1: Comparison of KGPNet performance with Faster R-CNN baseline.

Table 5.4: Comparison of KGPNet performance with Faster R-CNN baseline.

Backbone		AP	AP50	AP75	APs	APm	APl
ResNet-50-C4	Faster R-CNN	62.6654	87.0327	74.4862	75	58.3266	62.9044
	KGPNet	**68.3751**	**92.55893**	**81.728**	**80**	**64.358**	**68.748**
ResNet-50-FPN	Faster R-CNN	63.7127	86.66296	76.925	71.2623	58.109	64.5976
	KGPNet	**69.7001**	**94.4101**	**83.3843**	**90**	**66.4566**	**70.0557**

For more detailed performance over each class, Figure 5.2 visualizes the AP at $IoU = .50 : .05 : .95$ for all labels in the dataset. The yellow and green beans represent results achieved by two alternatives of the KGPNet, while the blue and the pink ones denotes Faster R-CNN performance. From this pirate plot, apart from the fact that the mean AP over all classes of KGPNet variances are better than those produced by Faster R-CNN, there are also some great insights which could be drawn. By observing the density bean of each group, it can be seen that KGPNet also produce a more reliable and stable results over all classes. While the two beans of Faster R-CNN exhibit a great variance, the beans of KGPNet performance are more condensed. Additionally, the thin rectangles of two KGPNet groups indicate a condensed 95% High Density Interval (HDI) with limited value ranges.

Upon investigating the performance, with the use of FPN, Faster R-CNN and KGPNet can achieve a more robust result while remain an acceptable speed. Hence, from now on without any further mention, all the experiments are carried out with FPN backbone.

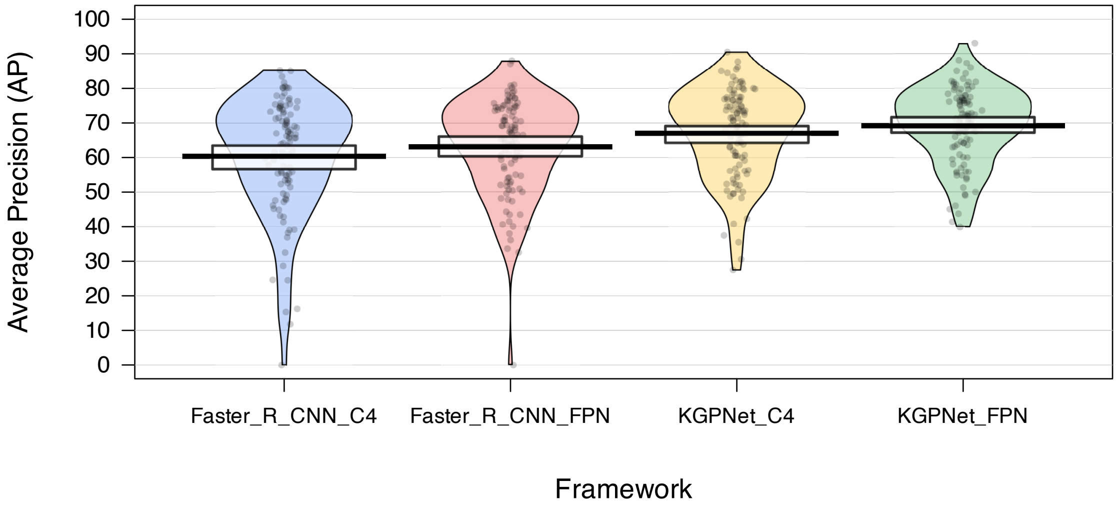
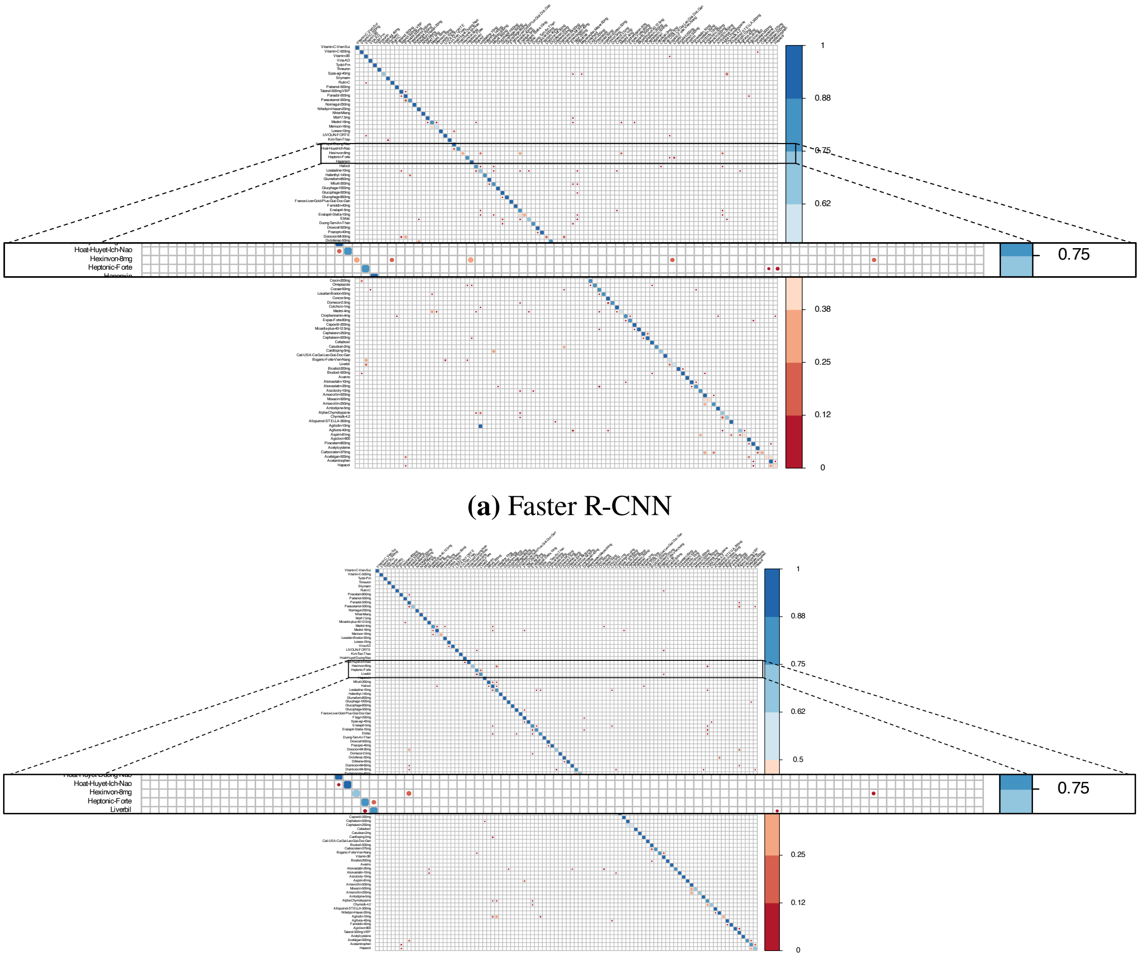


Figure 5.2: Comparison of KGPNet performance with Faster R-CNN baseline over each individual class.

5.3.2 Effect of Medical Co-occurrence Graph on performance of Pill Detection framework

This section is dedicated for describing the effects of MCG on the performance of Pill Detector frameworks. Figure 5.3a and 5.3b respectively display the label confusion matrix produced by Faster R-CNN and KGPNet on testing dataset. The two confusion matrices are *row-wisedly normalized* before being used for plotting. In addition, all the zero values are *disregarded* in this visualisation, hence having white color. By observing the main diagonal of the matrix, it can be seen that there are many labels which are still misclassified (in red color) made by Faster R-CNN. The situation is much positive with KGPNet, most of the labels are correctly classified and the confusion cases are greatly reduced.

For further investigation, I make a visualization of a label - **Hexinvon-8mg** that is usually puzzled by Faster R-CNN in Figure 5.4. As the figure illustrates, all the pills that are usually mixed up with **Hexinvon-8mg** - suggested by Confusion Matrix 5.3a, possess very identical appearances compared with each other (round shape, white color, etc.). For KGPNet, by observing the row of **Hexinvon-8mg**, it can be seen that the problem is successfully alleviated with the aid of MCG graph. Figure 5.5 illustrates an example case, in which **Hexinvon-8mg** is successfully distinguished by KGPNet with a high confidence scores, while in the case of Faster R-CNN, this pill is miscategorized as **Alpha-Chymotrypsine**. By basing on the context pills around the hard pill samples - in this case **Hexinvon-8mg**, KGPNet have well differentiated and determined the true label in most cases.



(a) Faster R-CNN

(b) KGPNet

Figure 5.3: Confusion Matrices of predicted labels made by two framework without and with MCG Leverage

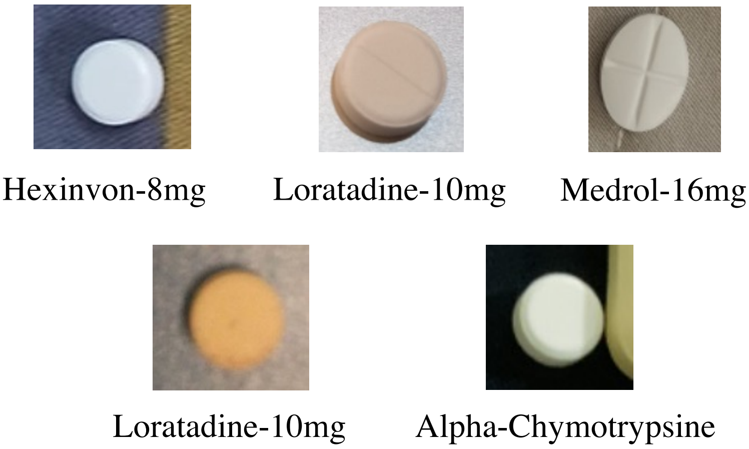
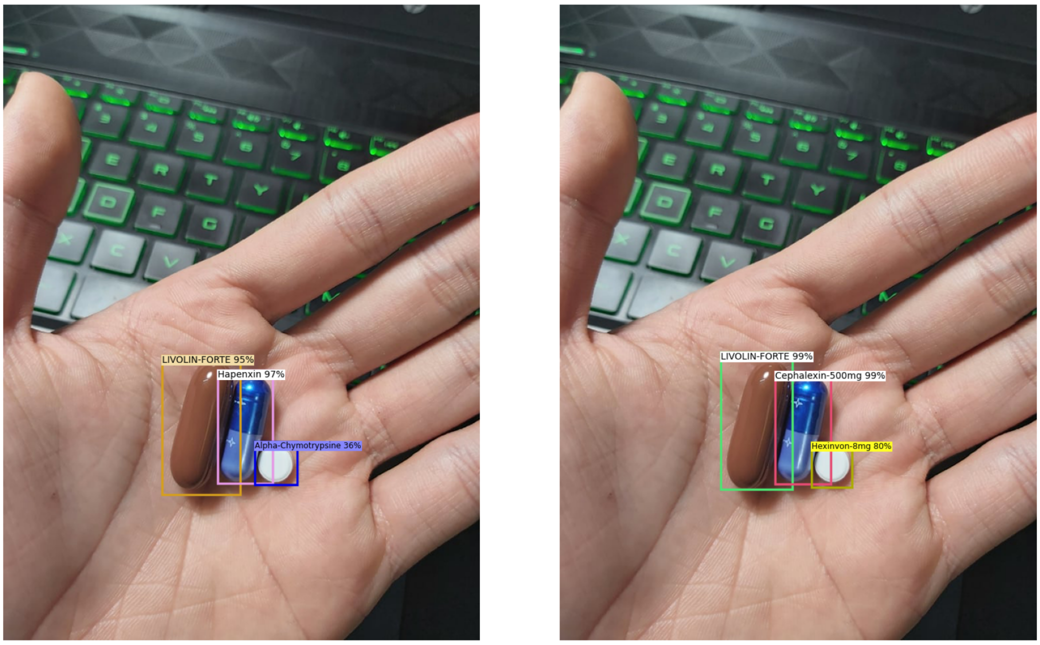


Figure 5.4: Misclassified pills made by Faster R-CNN

5.3.3 Comparison with Object Detection framework which leverage external knowledge

As previously mentioned, this is the first study that focuses on solving Pill Detection challenge utilizing an external knowledge graph. Due to this, none of the preceding works are genuinely tight-correlated. Indeed, earlier research has also



(a) **(b)**

Figure 5.5: Predictions for a hard sample made by Faster R-CNN and KGPNet given the same image. **a** - Faster R-CNN; **b** - `Conv` KGPNet.

utilized external information to improve the Object Detection problem. I utilize one of the most current studies with this direction - [6] to solve my specified problem.

Briefly, aside from the majority of studies that construct graphs using external data (handcrafted linguistic knowledge, etc.) or by implicitly learning a fully-connected graph between regions of interest (RoIs), Spatial-aware Graph Relation Network (SGRN) [6] is a framework that adaptively discovers and incorporates key semantic and spatial relationships for reasoning over each RoI. Due to this, SGRN requires no extra information and hence can be installed with minor changes compared to the original public work, and thus remains its original strength.

Table 5.5: Comparison of KGPNet performance with SGRN.

Framework	AP	AP50	AP75	APs	APm	APl
Faster R-CNN	63.71	86.66	76.92	71.26	58.10	64.59
SGRN [6]	65.88	88.83	79.64	76.31	61.58	66.28
KGPNet	**69.70**	**94.41**	**83.38**	**90.00**	**66.45**	**70.05**

Table 5.5 summarizes the results of SGRN, compared with Faster R-CNN and KGPNet. SGRN is also a module that replaces the output layer of the original Faster R-CNN, much as KGPNet does. This is the reason why Faster R-CNN is put into comparison. Fron the numbers, it can be seen the effectiveness of SGRN over Faster R-CNNN, but the situation is different upon comparing with KGPNet. The

overall AP metrics achieved by SGRN is 65.88, and KGPNet achieves the better score with the gap of nearly 4. Upon other metrics AP50, AP75, AP[s, m, l], KGPNet show its superior by enhancing the performance from 5.1% for AP75 metrics to 17.1% for APs metrics.

Adapting to the challenge of Pill Detection, SGRN reveal a major weakness. The information of spatial relationships between pills in an image is noisy and arbitrary, hence it is an unreliable source of information. Since SGRN works rely on this knowledge, it can not produce a good enough result for dealing with cases of hard samples, compared with this proposal.

5.4 Ablation studies

For studying the effectiveness of each individual component to the overall KGPNet's performance, the detailed ablation studies have been conducted, with the configuration mentioned in Table 5.6. The $+$ sign indicates the presence of a component in a specific version, while $-$ denotes the opposite. Since my vanilla backbone model is Faster R-CNN, without any proposed module, it also listed in the table. The ablation of each module is quite straight-forward to adopt without any replacement needed.

Table 5.6: Different versions of KGPNet with the ablations of components.

	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{G}_3$	GTN	$\mathcal{L}_{aux}$
Faster R-CNN	-	-	-	-	-
KGPNet v1	+	-	-	-	-
KGPNet v2	+	-	+	+	+
KGPNet v3	+	+	-	+	+
KGPNet v4	+	+	+	-	+
KGPNet v5	+	+	+	+	-
KGPNet v6	+	+	+	+	+

Especially, for the adoption of KGPNet without GTN module, a normal Graph Convolutional Network (GCN) is used for learning the node presentations. This module is shared for all three homogeneous graphs $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_3$, and the final used node presentations used is the average results of three outputs.

Figure 5.6 has visually demonstrated the performance of different KGPNet versions. There are some important insights that can be drawn from this illustration. First, the KGPNet version with all components work best by showing its superior over all recorded metrics. Second, all KGPNet versions is better than vannila Faster R-CNN backbone, which shows the effectiveness of each individual component on the overall performance of KGPNet, compared with the well known Object Detec-

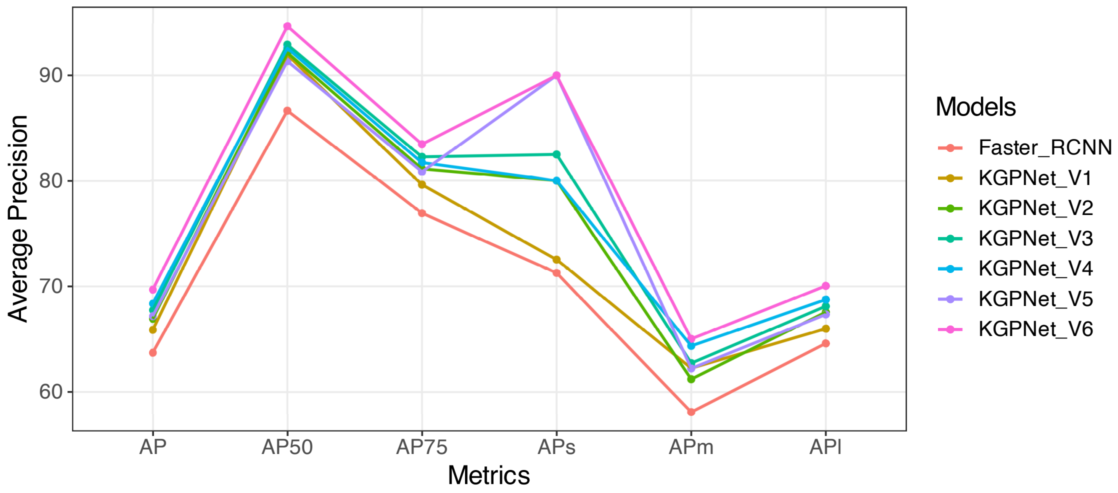


Figure 5.6: Performance comparison of different KGPNet versions with the ablations of components.

tion framework Faster R-CNN.

CHAPTER 6. CONCLUSION AND FUTURE WORKS

6.1 Summary

In this research, I present a unique method for resolving problems in the image-based pill detection task. The proposed technique explicitly reveals the relationships among pills in medication captures, and makes good uses of them to improve the identification of pills from photos. These information is specifically represented as a medical knowledge graph that is utilized to help the primary job of pill recognition. In addition, prescriptions are extra data source for modeling one of the knowledge graphs being used.

Extensive experiments on a collection of real-world pill photos - VAIPE PP revealed that the proposed framework outperforms the baseline that employs solely pill images by a significant margin - up to 9.4% for COCO AP metrics. I also study the influence of the prescription-based medical co-occurence graph on pill detection performance and find that the graph plays a key role in enhancing the overall system's performance and solving the major problem of this task. I believe that utilizing the external graphs will improve pill identification outcomes.

6.2 Suggestion for Future Works

I am now developing this study by incorporate this proposed module with many more Object Detection frameworks to better verify its robustness regardless of choosen backbones. In addition, since the desired outcome of this work is an actual application that can aid patients in recognizing their pills, additional effort should be made to balance out the two factor of accuracy and efficiency. Finally, I want to think of establishing its applicability in other clinical contexts.

SCIENTIFIC PUBLICATIONS

A preliminary version of this work, which targets Pill Classification problem, named **Image-based Contextual Pill Recognition with Medical Knowledge Graph Assistance**, has recently been accepted for publication in ACIIDS 2022: 14^{th} Asian Conference on Intelligent Information and Database Systems.

- **ACIID 2022 - Conference**: **Anh Duy Nguyen**, Thuy Dung Nguyen, Huy Hieu Pham, Thanh Hung Nguyen, Phi Le Nguyen, **"Image-based Contextual Pill Recognition with Medical Knowledge Graph Assistance"**, *14th Asian Conference on Intelligent Information and Database Systems (ACIIDS), 2022.*

In addition, while experiencing a short journey through Hanoi University of Science and Technology, under the direction of Dr. Nguyen Phi Le and Dr. Nguyen Thanh Hung, I was also able to achieve some of my earlier works accepted for publication, which is an honorable accomplishment.

- **WCNC 2022 - Conference**: Tuan Anh Nguyen Dinh, **Anh Duy Nguyen**, Truong Thao Nguyen, Thanh Hung Nguyen, Phi Le Nguyen, "Spatial-temporal Coverage Maximization in Vehicle-based Mobile Crowdsensing for Air Quality Monitoring", *IEEE Wireless Communications and Networking Conference (WCNC) 2022.*
- **IAE/IEA 2021 - Conference**: **Anh Duy Nguyen**, Viet Hung Vu, Minh Hieu Nguyen, Duc Viet Hoang, Thanh Hung Nguyen, Kien Nguyen, Phi Le Nguyen, “Efficient Prediction of Discharge and Water Levels Using Ensemble Learning and Singular-Spectrum Analysis-based Denoising”, *The 34th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems IEA/AIE 2021.*
- **MDPI Sensors 2021 - Journal**: Van Quan La, **Anh Duy Nguyen**, Thanh Hung Nguyen, Kien Nguyen, Phi Le Nguyen, "An On-demand Charging for Connected Target Coverage in WRSNs using Fuzzy Logic and Q-learning", *MDPI Sensors, 21(16), 5520.*

REFERENCE

[1] W.-J. Chang, L.-B. Chen, C.-H. Hsu, C.-P. Lin and T.-C. Yang, "A deep learning-based intelligent medicine recognition system for chronic patients," **IEEE Access**, vol. 7, pp. 44 441–44 458, 2019. DOI: `10.1109/ACCESS.2019.2908843`.

[2] Z. Yaniv, J. Faruque, S. Howe **et al.**, "The national library of medicine pill image recognition challenge: An initial report," in **2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)**, 2016, pp. 1–9. DOI: `10.1109/AIPR.2016.8010584`.

[3] T. N. Hà, **Tổng kết công tác báo cáo adr năm 2021**, 2022.

[4] **World patient safety day 2022**, `https://www.who.int/news-room/events/detail/2022/09/17/default-calendar/world-patient-safety-day-2022`, Accessed: 2022-04-14.

[5] Y. F. Wong, H. T. Ng, K. Y. Leung, K. Y. Chan, S. Y. Chan and C. C. Loy, "Development of fine-grained pill identification algorithm using deep convolutional network," **Journal of Biomedical Informatics**, vol. 74, pp. 130–136, 2017, ISSN: 1532-0464. DOI: `https://doi.org/10.1016/j.jbi.2017.09.005`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S1532046417302022`.

[6] H. Xu, C. Jiang, X. Liang and Z. Li, "Spatial-aware graph relation network for large-scale object detection," in **2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, 2019, pp. 9290–9299. DOI: `10.1109/CVPR.2019.00952`.

[7] W.-J. Chang, L.-B. Chen, C.-H. Hsu, J.-H. Chen, T.-C. Yang and C.-P. Lin, "Medglasses: A wearable smart-glasses-based drug pill recognition system using deep learning for visually impaired chronic patients," **IEEE Access**, vol. 8, pp. 17 013–17 024, 2020. DOI: `10.1109/ACCESS.2020.2967400`.

[8] X. Zhang, Y.-H. Yang, Z. Han, H. Wang and C. Gao, "Object class detection: A survey," **ACM Comput. Surv.**, vol. 46, no. 1, 2013, ISSN: 0360-0300. DOI: `10.1145/2522968.2522978`. [Online]. Available: `https://doi.org/10.1145/2522968.2522978`.

[9] O. Russakovsky, J. Deng, H. Su **et al.**, "ImageNet Large Scale Visual Recognition Challenge," **International Journal of Computer Vision (IJCV)**, vol. 115, no. 3, pp. 211–252, 2015. DOI: `10.1007/s11263-015-0816-y`.

[10] R. Girshick, J. Donahue, T. Darrell and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in **2014 IEEE Conference on Computer Vision and Pattern Recognition**, 2014, pp. 580–587. DOI: 10.1109/CVPR.2014.81.

[11] ——, “Region-based convolutional networks for accurate object detection and segmentation,” **IEEE transactions on pattern analysis and machine intelligence**, vol. 38, no. 1, pp. 142–158, 2015.

[12] A. Krizhevsky, I. Sutskever and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in **Advances in Neural Information Processing Systems**, F. Pereira, C. Burges, L. Bottou and K. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436
Paper.pdf.

[13] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers and A. W. M. Smeulders, “Selective search for object recognition.,” **Int. J. Comput. Vis.**, vol. 104, no. 2, pp. 154–171, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/ijcv/ijcv104.html#UijlingsSGS13.

[14] R. Girshick, **Fast r-cnn**, cite arxiv:1504.08083Comment: To appear in ICCV 2015, 2015. [Online]. Available: http://arxiv.org/abs/1504.08083.

[15] S. Ren, K. He, R. Girshick and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in **Advances in Neural Information Processing Systems**, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama and R. Garnett, Eds., vol. 28, Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

[16] J. Dai, Y. Li, K. He and J. Sun, “R-fcn: Object detection via region-based fully convolutional networks.,” in **NIPS**, D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon and R. Garnett, Eds., 2016, pp. 379–387. [Online]. Available: http://dblp.uni-trier.de/db/conf/nips/nips2016.html#DaiLHS16.

[17] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus and Y. LeCun, **Overfeat: Integrated recognition, localization and detection using convolutional networks**, 2013. DOI: 10.48550/ARXIV.1312.6229. [Online]. Available: https://arxiv.org/abs/1312.6229.

[18] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, **You only look once: Unified, real-time object detection**, 2015. DOI: 10.48550/ARXIV.

1506.02640. [Online]. Available: https://arxiv.org/abs/1506.02640.

[19] W. Liu, D. Anguelov, D. Erhan **et al.**, "SSD: Single shot MultiBox detector," in **Computer Vision – ECCV 2016**, Springer International Publishing, 2016, pp. 21–37. DOI: 10.1007/978-3-319-46448-0_2. [Online]. Available: https://doi.org/10.1007%2F978-3-319-46448-0_2.

[20] H.-W. Ting, S.-L. Chung, C.-F. Chen, H.-Y. Chiu and Y.-W. Hsieh, "A drug identification model developed using deep learning technologies: Experience of a medical center in taiwan," **BMC Health Services Research**, vol. 20, 2020, ISSN: 1472-6963. DOI: 10.1186/s12913-020-05166-w. [Online]. Available: https://bmchealthservres.biomedcentral.com/articles/10.1186/s12913-020-05166-w#citeas.

[21] D. Zhou, J. Weston, A. Gretton, O. Bousquet and B. Schölkopf, "Ranking on data manifolds," in **Advances in Neural Information Processing Systems**, S. Thrun, L. Saul and B. Schölkopf, Eds., vol. 16, MIT Press, 2003. [Online]. Available: https://proceedings.neurips.cc/paper/2003/file/2c3ddf4bf13852db711dd1901fb517fa-Paper.pdf.

[22] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in **Proceedings of the IEEE conference on computer vision and pattern recognition**, 2017, pp. 1251–1258.

[23] S. Ling, A. Pastor, J. Li **et al.**, "Few-shot pill recognition," in **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, 2020.

[24] T. P. Proma, M. Z. Hossan and M. A. Amin, "Medicine recognition from colors and text," in **Proceedings of the 2019 3rd International Conference on Graphics and Signal Processing**, ser. ICGSP '19, Hong Kong, Hong Kong: Association for Computing Machinery, 2019, 39–43, ISBN: 9781450371469. DOI: 10.1145/3338472.3338484. [Online]. Available: https://doi.org/10.1145/3338472.3338484.

[25] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," **Biological Cybernetics**, vol. 36, pp. 193–202, 1980.

[26] Y. LeCun, B. Boser, J. S. Denker **et al.**, "Backpropagation applied to handwritten zip code recognition," **Neural Computation**, vol. 1, pp. 541–551, 1989.

[27] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," **Proceedings of the IEEE**, vol. 86, no. 11, pp. 2278–2324, 1998, ISSN: 0018-9219.

[28] ImageNet, **Imagenet object localization challenge**, 2018. [Online]. Available: `https://www.kaggle.com/c/imagenet-object-localization-challenge`.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in **3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: `http://arxiv.org/abs/1409.1556`.

[30] C. Szegedy, W. Liu, Y. Jia **et al.**, **Going deeper with convolutions**, cite arxiv:1409.4842, 2014. [Online]. Available: `http://arxiv.org/abs/1409.4842`.

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

[32] H. Yang, S. Pan, P. Zhang, L. Chen, D. Lian and C. Zhang, "Binarized attributed network embedding.," in **ICDM**, IEEE Computer Society, 2018, pp. 1476–1481, ISBN: 978-1-5386-9159-5. [Online]. Available: `http://dblp.uni-trier.de/db/conf/icdm/icdm2018.html#YangP00LZ18`.

[33] X. Shen, S. Pan, W. Liu, Y.-S. Ong and Q.-S. Sun, "Discrete network embedding.," in **IJCAI**, J. Lang, Ed., ijcai.org, 2018, pp. 3549–3555, ISBN: 978-0-9992411-2-7. [Online]. Available: `http://dblp.uni-trier.de/db/conf/ijcai/ijcai2018.html#0001PLOS18`.

[34] B. Perozzi, R. Al-Rfou and S. Skiena, "Deepwalk: Online learning of social representations," in **Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, ser. KDD '14, New York, New York, USA: Association for Computing Machinery, 2014, 701–710, ISBN: 9781450329569. DOI: `10.1145/2623330.2623732`. [Online]. Available: `https://doi.org/10.1145/2623330.2623732`.

[35] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," University of Pisa, Dipartimento di Informatica, Pisa, Italy, Tech. Rep. TR-16/95, 1995.

[36] M. Gori, G. Monfardini and F. Scarselli, "A new model for learning in graph domains," in **Proceedings. 2005 IEEE International Joint Conference on**

Neural Networks, 2005., vol. 2, 2005, pp. 729–734. DOI: 10.1109/IJCNN.2005.1555942.

[37] T. N. Kipf and M. Welling, **Semi-supervised classification with graph convolutional networks**, 2016. DOI: 10.48550/ARXIV.1609.02907. [Online]. Available: https://arxiv.org/abs/1609.02907.

[38] W. L. Hamilton, R. Ying and J. Leskovec, **Inductive representation learning on large graphs**, 2017. DOI: 10.48550/ARXIV.1706.02216. [Online]. Available: https://arxiv.org/abs/1706.02216.

[39] X. Wang, H. Ji, C. Shi **et al.**, **Heterogeneous graph attention network**, cite arxiv:1903.07293Comment: 10 pages, 2019. [Online]. Available: http://arxiv.org/abs/1903.07293.

[40] K. Xu, W. Hu, J. Leskovec and S. Jegelka, **How powerful are graph neural networks?** 2018. DOI: 10.48550/ARXIV.1810.00826. [Online]. Available: https://arxiv.org/abs/1810.00826.

[41] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang and M. Wang, **Lightgcn: Simplifying and powering graph convolution network for recommendation**, 2020. DOI: 10.48550/ARXIV.2002.02126. [Online]. Available: https://arxiv.org/abs/2002.02126.

[42] Q. Ji, J. Huang, W. He and Y. Sun, “Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images,” **Algorithms**, vol. 12, no. 3, 2019, ISSN: 1999-4893. DOI: 10.3390/a12030051. [Online]. Available: https://www.mdpi.com/1999-4893/12/3/51.

[43] D. Bahdanau, K. Cho and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” English (US), 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015, Jan. 2015.

[44] R. Srivastava and A. More, “Some aesthetic considerations for over the-counter (otc) pharmaceutical products,” **Int. J. of Biotechnology**, vol. 11, pp. 267 –283, Nov. 2010. DOI: 10.1504/IJBT.2010.036600.

[45] O. Droulers and B. Roullet, “Pharmaceutical packaging color and drug expectancy,” **Advances in Consumer Research**, vol. 32, pp. 164–171, Jan. 2005.

[46] d. V. A. de Craen AJ Roos PJ and K. J, “Effect of colour of drugs: Systematic review of perceived effect of drugs and of their effectiveness,” **BMJ**, pp. 1624–1626, 1996. DOI: 10.1136/bmj.313.7072.1624.

[47] H. Xu, C. Jiang, X. Liang, L. Lin and Z. Li, “Reasoning-rcnn: Unifying adaptive global reasoning into large-scale object detection,” in **2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, 2019, pp. 6412–6421. DOI: `10.1109/CVPR.2019.00658`.

[48] S. Yun, M. Jeong, R. Kim, J. Kang and H. J. Kim, **Graph transformer networks**, 2019. DOI: `10.48550/ARXIV.1911.06455`. [Online]. Available: `https://arxiv.org/abs/1911.06455`.

[49] Z. Yaniv, J. Faruque, S. Howe **et al.**, “The national library of medicine pill image recognition challenge: An initial report,” Oct. 2016, pp. 1–9. DOI: `10.1109/AIPR.2016.8010584`.

[50] S. Ling, A. Pastor, J. Li **et al.**, “Few-shot pill recognition,” in **2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, 2020, pp. 9786–9795. DOI: `10.1109/CVPR42600.2020.00981`.

[51] T.-Y. Lin, M. Maire, S. Belongie **et al.**, “Microsoft coco: Common objects in context,” in **Computer Vision – ECCV 2014**, D. Fleet, T. Pajdla, B. Schiele and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 740–755, ISBN: 978-3-319-10602-1.

[52] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in **7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019**, OpenReview.net, 2019. [Online]. Available: `https://openreview.net/forum?id=Bkg6RiCqY7`.

[53] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, **Feature pyramid networks for object detection**, 2016. DOI: `10.48550/ARXIV.1612.03144`. [Online]. Available: `https://arxiv.org/abs/1612.03144`.