RESEARCH ARTICLE

High accurate and explainable multi-pill detection framework with graph neural network-assisted multimodal data fusion

Anh Duy Nguyen[1,3], Huy Hieu Pham[2,3], Huynh Thanh Trung[4], Quoc Viet Hung Nguyen[5], Thao Nguyen Truong[6], Phi Le Nguyen[1] *

1 School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi, Vietnam, 2 College of Engineering & Computer Science, VinUniversity, Hanoi, Vietnam, 3 VinUni-Illinois Smart Health Center, VinUniversity, Hanoi, Vietnam, 4 École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 5 Griffith University, Southport, Queensland, Australia, 6 National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

* lenp@soict.hust.edu.vn

Abstract

Due to the significant resemblance in visual appearance, pill misuse is prevalent and has become a critical issue, responsible for one-third of all deaths worldwide. Pill identification, thus, is a crucial concern that needs to be investigated thoroughly. Recently, several attempts have been made to exploit deep learning to tackle the pill identification problem. However, most published works consider only single-pill identification and fail to distinguish hard samples with identical appearances. Also, most existing pill image datasets only feature single pill images captured in carefully controlled environments under ideal lighting conditions and clean backgrounds. In this work, we are the first to tackle the multi-pill detection problem in real-world settings, aiming at localizing and identifying pills captured by users during pill intake. Moreover, we also introduce a multi-pill image dataset taken in unconstrained conditions. To handle hard samples, we propose a novel method for constructing heterogeneous a priori graphs incorporating three forms of inter-pill relationships, including co-occurrence likelihood, relative size, and visual semantic correlation. We then offer a framework for integrating a priori with pills' visual features to enhance detection accuracy. Our experimental results have proved the robustness, reliability, and explainability of the proposed framework. Experimentally, it outperforms all detection benchmarks in terms of all evaluation metrics. Specifically, our proposed framework improves COCO mAP metrics by **9.4%** over Faster R-CNN and **12.0%** compared to vanilla YOLOv5. Our study opens up new opportunities for protecting patients from medication errors using an AI-based pill identification solution.

Introduction

Background

Oral pills are among the most popular and commonly used methods in healthcare due to their efficacy and simplicity. Pills usually exhibit various visual features in terms of shape, color, and imprinted text. Despite this, erroneously taking pills is exceptionally prevalent due to the significant similarity in pill appearances. According to a WHO report [1], drug misuse rather than illness is responsible for one-third of all deaths. Moreover, according to Yaniv et al. [2], around six to eight thousand people are killed annually by prescription errors. Recently, the US National Centers for Biomedical Computing (NCBCs) stated that taking this country alone, each year, 7,000 to 9,000 people die due to a medication error. This circumstance necessitates the invention of solutions to protect users/patients from taking incorrect pills. This need is more stringent than ever, given the aging of the global population and the rising prevalence of chronic diseases requiring continuous medication.

Image-based pill recognition

In the early stages, pill recognition was handled through a variety of online systems that allowed users to identify pills by manually entering multiple attributes, such as shape, color, and imprinted text [3]. However, these methods are time-consuming and may not be reliable, as the predefined features may not encompass all real-world cases. Recently, Artificial Intelligence (AI) has made tremendous achievements and emerged as a powerful tool for resolving various problems. Although still in its infancy, AI has been used to recognize pills from images, helping prevent incorrect medication. An early effort to classify pills using a Deep Convolution Network (DCN) was introduced in [4]. In [5], the authors provided ePillID, a large pill image dataset comprising 13*K* images representing 8,184 appearance classes. Additionally, they conducted experiments to evaluate various baseline models on the proposed dataset. Even with the best baseline, the experimental findings demonstrated that it fails to discriminate between confusing classes. The problem of few-shot pill recognition was addressed in [6]. The authors also provided new pill image data named CURE. Recently, there have been a few works considering the multi-pill detection problem [7]. The authors adopted two-step deep neural networks consisting of an object localization model and a classifier.

Problem statement

Despite several efforts that have been made, existing solutions for pill identification reveal the following critical shortcomings.

- Most existing works have been restricted to the classification of single-pill images. This constraint limits the solutions' application capacity, as in practice, users usually take multiple pills simultaneously, resulting in multi-pill images in most cases. In such scenarios, the utilization of existing frameworks necessitates the additional use of localization models and depends on the behavior of these models. Extra efforts are needed to harmonize the workings of the two models.

- Most of the current pill image datasets (e.g., ePillID, CURE) are limited to single-pill images. Moreover, all of them were collected in tightly-controlled settings under ideal illumination and background conditions, leading to a lack of diversity. Those previous works trained on these datasets are vulnerable to Out-Of-Distribution (OOD) data (with arbitrary light or angle configurations) when being evaluated in real-world settings.

- No prior work has studied the explainability of the model. This insufficiency diminishes the trustworthiness of the solutions, hence restricting their practical applications in healthcare, which directly related to patiences' safety.

We are, to the best of our knowledge, the first to tackle the multi-pill detection problem in real-world settings. Specifically, we focus on a practical application that recognizes pills in patients' pill intake pictures. Our targeted problem can be formulated as follows. *Given an image capturing multiple pills in patients' pill intake, we aim to determine each pill's location and identity*. In addition to developing a novel pill detection framework with high reliability and explainable capacity, we build a dataset of multi-pill images captured under unconstrained real-world conditions.

Our motivation and key ideas

One of the most significant obstacles in the pill detection problem is the existence of numerous pills with similar shapes, colors, and sizes (Fig 1). We call these hard samples, whose occurrence renders the pill identification problem complicated and challenging to solve by generic object detection.

We argue that relying merely on pills' appearance is insufficient to improve pill detection accuracy, if not impossible. We discovered that besides the challenge (e.g., localizing pills in hard cases such as overlapping pills), the multi-pill detection problem, on the other hand, provides us with an opportunity to improve pill recognition accuracy. Motivated by the human tendency and ability to integrate different data sources while making decisions, our proposed solution seeks to utilize external knowledge to improve detection accuracy. Specifically, we rely on three different relationships between the pills. For each of them, we provide a corresponding *a priori graph* modelling that relation. These connections include: co-occurrence, relative size, and visual semantic correlation.

The first a priori, or co-occurrence graph, demonstrates the frequency with which medications are prescribed for the same diseases; thus, it reflects the likelihood that pills appear in the same image. This a priori understandings originates from the understanding that the consumption of pills is not random; rather, pharmacists prescribe them to treat or alleviate specific symptoms or illnesses. This premise underscores the existence of a robust correlation among concurrently consumed pills. In essence, when provided with a set of contextual pills and an unfamiliar one, the co-occurrence graph enables the restriction of potential choices for the remaining pill. By utilizing this knowledge, we can enhance the accuracy of dealing with hard samples by leveraging the high accuracy of detecting easy samples in the same image. The second one, i.e., the relative size graph, gives us the relative size information of the pills, thus, improving our model's capacity to distinguish pills of identical shape and color but differing in size. Furthermore, this relationship also proves advantageous in tackling the inherent

| | | | | | Enalapril-5mg | Alpha-chymotrysin | Cardiloping-5mg | Mezafen-60mg |

(a)   (b)
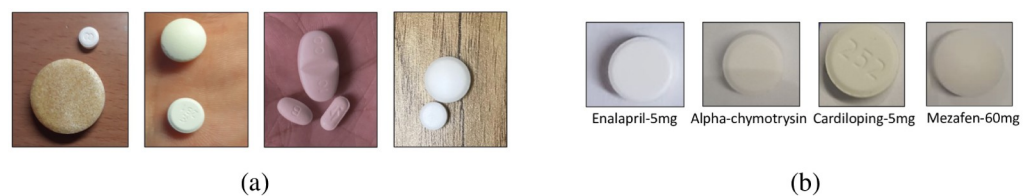
Fig 1. Hard samples with high similarity in terms of shape, color and size (examples taken from our handcrafted dataset). (a) Pills with similar shapes, colors, and different sizes. (b) Pills with similar shapes, sizes and colors.
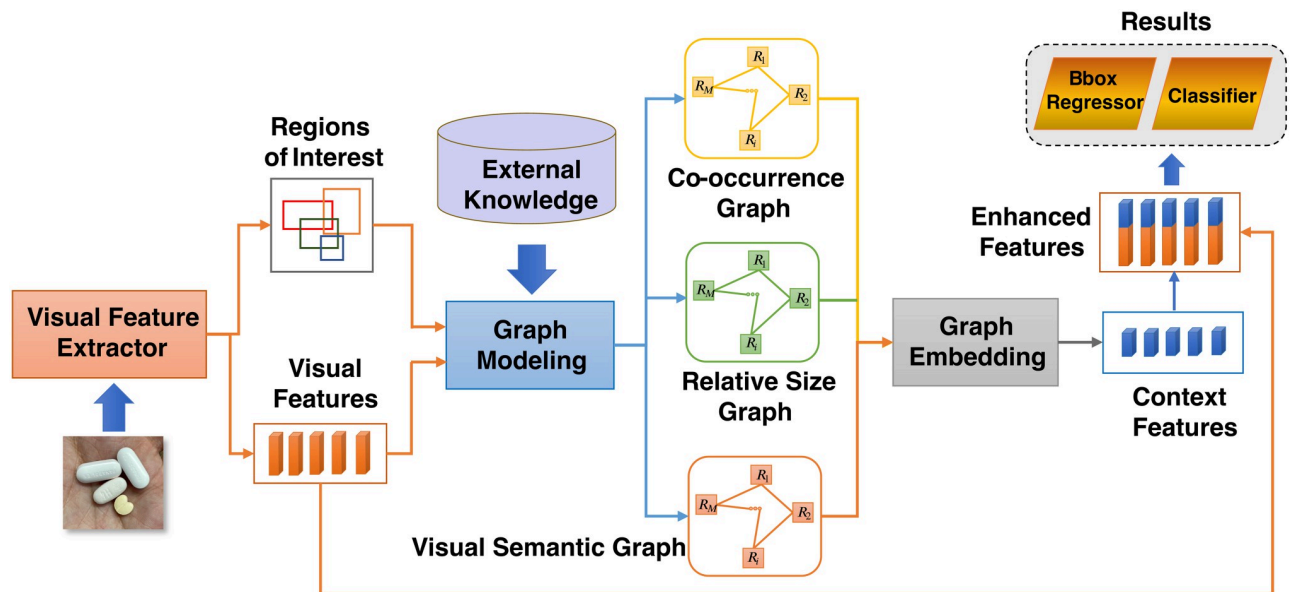
Fig 2. Overview of our proposed solution. The pipeline consists three steps: modeling a priori graphs, transforming them into a vector space, and fusing them with visual features.

challenge of multi-scale handling for conventional Object Detection tasks [8, 9]. Like the original problem, there are many kinds of pills with different scales. The proposed relative size graph can support the detector framework by providing additional sizing information between those pills, thereby enhancing the representations. Lastly, the visual semantic graph learns pills' latent semantic connections embedded in their visual appearance. Unlike previous a priori graphs, this graph directly models the visual alignments of pills in input images, and this information is also beneficial in enriching the visual features produced by the detector.

Besides, to leverage the aforementioned a-priori, our proposed framework offers a multimodal fusion method for incorporating graph-based inter-pill relational information with intra-pill visual features to enhance the detection result. The overview of our proposed model is illustrated in Fig 2. We leverage external knowledge from prescriptions and training datasets to build the co-occurrence and relative size graph. The visual features of the pills are exploited to construct the visual semantic graph. Using the graph embedding module, the three graphs are transformed into the vector space, then fused with the visual features to provide enhanced feature vectors, which are then utilized to create the final results.

The significance of the proposed framework stems from the fact that this is the first solution to the challenge of identifying multiple pills in a single image captured under real-world conditions. Notably, the proposed solution gains high accuracy even in hard cases, i.e., the existence of pills with substantially similar visual appearances.

Our contributions

Our main contributions can be summarized as follows.

- We introduce the first real-world multi-pill image dataset consisting of 9,426 images representing 96 pill classes. The images were taken with ordinary smartphones in various settings. The dataset will assist in the advancement of research in the field.

- We propose a novel pill detection framework named PGPNet (which stands for a Priori Graph-assisted Pill Detection Network), which leverages three-fold graph-based a priori, including co-occurrence likelihood, relative pill size, and visual semantic correlation to tackle hard pill samples. In addition, we provide a method for constructing these heterogeneous a priori graphs from given prescriptions and the training pill image dataset. Furthermore, we offer a multi-modal fusion method for incorporating graph-based inter-pill relational information with intra-pill visual features to enhance the detection result.

- We conduct thorough experiments to evaluate the efficacy of the proposed solution and compare it to existing state-of-the-art (SOTA). The experimental findings demonstrated that our approach enhances object detection accuracy by at least 9.4% for the COCO mAP metric compared to generic SOTA in object detection.

The remainder of the paper is divided into four sections. We briefly summarize the literature on pill detection and pill image datasets in Section Related works. In Section Methodology, we describe our methodology in detail. Section Dataset and experiment settings evaluates the performance of our proposed PGPNet and compares it with the other methods. Finally, we conclude the paper in Section Conclusion.

Related works

Pill classification

Many studies have employed machine learning to tackle the pill recognition challenge [4, 10]. The authors in [4] first utilized the Manifold ranking-based method to filter out the foreground mask from the input pill image and then used an AlexNet-based network to identify the label. In [11], Ting et al. combined the Enhanced Feature Pyramid Networks and Global Convolution Networks to improve pill localization accuracy. Ling et al. [6] tackled the few-shot pill detection problem with a Multi-Stream (MS) deep learning model. In [12], the authors integrated three handcrafted features, namely shape, color, and imprinted text, to identify pills.

Recently, a few efforts have leveraged the two-stage object detection approach to solve the multi-pill detection challenge [7, 13]. In the first stage, object localization techniques are applied to determine the pills' bounding boxes. These bounding boxes are then fed into a classifier in the second stage to identify the pills. Specifically, in [13], an enhanced feature pyramid network based on the ResNet-50 backbone has been built for pill localization. After that, the pill bounding boxes are fed into an Inception-ResNet v2 for classification. Authors in [7] exploited the Mask-RCNN framework to solve the problem.

Multi-pill detection solutions are still in their infancy. All current works only investigate images acquired in laboratories under optimal lighting and background conditions, with each pill arranged separately. In fact, existing techniques only use specific object localization models to crop the pills and then treat the issue as a typical single-pill classification problem.

Pill image datasets

One of the most widely used pill image datasets is the NIH Pill Image Dataset [14], released by the U.S. National Library of Medicine (NLM). This dataset consists of 4,000 high-quality reference pills and 133,000 pictures captured by digital cameras on mobile phones. In [6], the authors provided the CURE pill dataset consisting of 8,973 single-pill images representing 196 classes. Although taken under various backgrounds and lighting conditions, all of these images are carefully captured from a top-down view and focus on the pills. Authors in [4] contributed

a pill dataset capturing about 400 commonly used tablets and capsules. Ten to twenty-five pictures were taken for each pill, resulting in 5,284 images.

Unfortunately, all of these datasets provide only single-pill images. Most images were captured under quite ideal conditions, e.g., pills were put on a clean background, and the images were taken from the top-down view with the camera focused on the pills.

Existing works' limitations

The literature suffers from the following three main drawbacks. First, most of the existing solutions only investigated images acquired in laboratories under optimal conditions, treating the problem as a conventional single-pill classification problem. Moreover, they did not adequately address the challenge of hard samples, where numerous pills have similar shapes, colors, and sizes. Second, existing pill datasets only provide single-pill images captured under ideal settings and do not account for real-world scenarios in which pills may be disorganized or partially concealed. Finally, no existing work studied the explainability of the models, thus lowering their trustworthiness and restricting their practical application.

Our solution

To fill in these gaps, we are the first to thoroughly investigate the problem of detecting multiple pills taken in the same image under real-world conditions. We propose a detection framework named PGPNet that employs graph-based a priori information, including co-occurrence likelihood, relative pill size, and visual semantic correlation, to tackle the challenge of hard pill samples. Notably, beside evaluating PGPNet's accuracy, we utilize explanatory methodologies to demonstrate its reliability. Finally, we provide the first real-world multi-pill image dataset taken with ordinary smartphones in various unrestricted environments.

Methodology

In this section, we propose a novel pill detection framework named PGPNet (i.e., a **P**riori **G**raph-assisted **P**ill Detection **Net**work).

PGPNet overview

We focus on a practical application that recognizes pills in patient intake pictures. Our model receives a multiple-pill picture as input and generates both the bounding box and the identification of each pill. Here, we incur a critical challenge: *how to distinguish pills with identical appearances (i.e., shape, color, and size)*. We believe that relying solely on the visual features of pills is insufficient to address this issue. Moreover, employing the correlation between pills rather than counting on each pill individually may enhance recognition accuracy. In light of this, we propose introducing two types of a priori, the first indicating the co-occurrence likelihood and the second modeling the relative size of pills. The a priori is extracted from a given prescription and pill image training dataset and represented as heterogeneous graphs. In particular, our strategy for handling difficult situations (i.e., distinguishing pills from distinct classes with similar appearances) is as follows. Our PGPNet first differentiates *easy* pills (those that do not have a remarkable resemblance to other pills) based on their visual appearance in the provided image. Obviously, these easy pills can be determined with high precision. The visual features of those easy samples will then serve as context vectors to assist with making decisions regarding the hard ones. In addition, we materialize this idea with a layer called *Pseudo Classifier* and a mechanism to filter out necessary information from the a priori graphs. In summary, the proposed model comprises four components: **A priori graph modeling**, **visual feature**
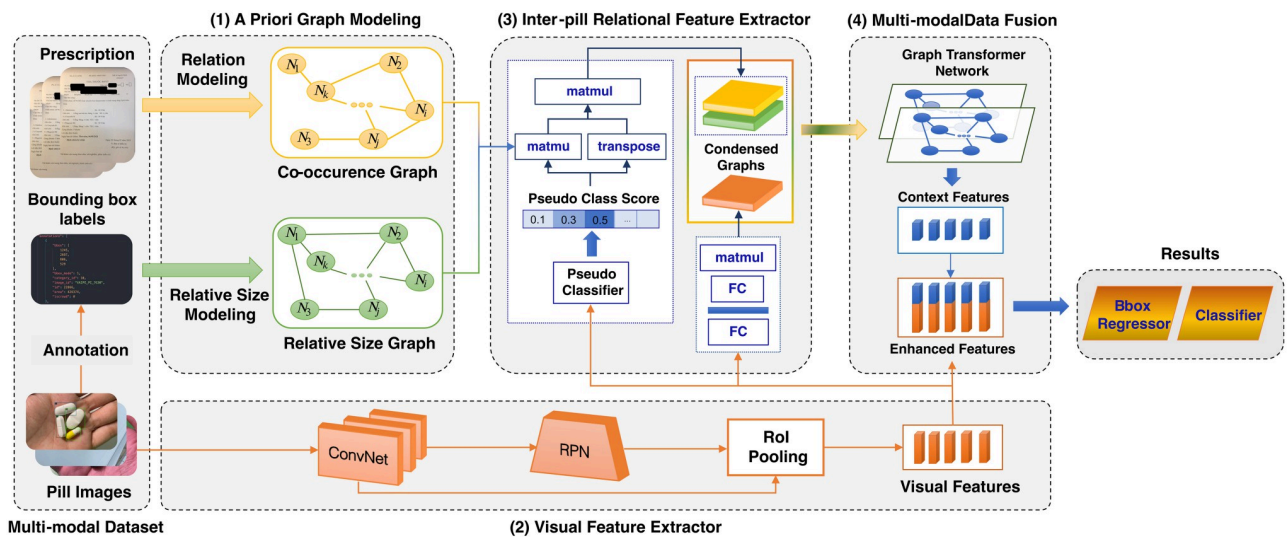
Fig 3. Workflow of PGPNet.

https://doi.org/10.1371/journal.pone.0291865.g003

extractor, **inter-pill relational feature extractor**, and **multi-modal data fusion**, as illustrated in Fig 3.

The overall flow is as follows.

- **Step 1—A priori graph modeling**. We construct two generic graphs, namely *Prescription-based Medical Co-occurrence Graph* (or Co-graph for short) and *Relative Size Graph* (Size-graph for short), that represent the relationship between all the pills in terms of co-occurrence and relative size, respectively. Concerning the former, we leverage a given set of prescriptions from which we can model the interaction between pills (i.e., which pills are likely to be used to treat the same diseases). Based on this information, we developed the Co-graph, whose nodes represent the pill classes and whose edge weights reflect the co-occurrence likelihood between the two vertices. In the meantime, using the coordinates of the bounding boxes from our training dataset for the pill detection task, we determine the area of each box and model the relative size ratios of all the pill classes in the given images. This information is then aggregated to formulate the Size-graph. Section A priori graph modeling covers the details of this algorithm.

- **Step 2—Visual feature extraction**. The original image containing multiple pills is passed through a Convolutional Network (ConvNet) for extracting visual features and a Region Proposal Network (RPN) for detecting potential Regions of Interest (RoI). The outputs of the two modules are fed into an RoI pooling layer to filter out all visual presentations of pills (i.e., RoIs). It is worth noting that the *Visual Feature Extractor* described here follows the architecture of the two-step object detection architecture (e.g., Faster RCNN [15]). However, PGPNet can also be implemented with one-step detection architecture.

- **Step 3—Inter-pill relational feature extraction**. The two a priori graphs are aggregated with the pills' visual features to yield condensed versions of the Co-graph and Size-graph that highlight the relationship between only those pills that are likely to appear in the image. Besides, the pills' visual features are leveraged to construct a so-called *Visual semantic graph* that captures the pills' relationships encapsulated under their appearances.

- **Step 4—Multi-modal data fusion**. Now, the inter-pill relational and intra-pill visual features are fused to obtain enhanced feature vectors, each of which encapsulates the characteristics of a pill as a standalone and its relationship with other pills. These enhanced feature vectors are used to offer the final results.

A priori graph modeling

In this section, we describe our method to construct the two generic graphs, namely the co-occurrence Graph (i.e., Co-graph) and relative size graph (i.e., Size-graph) in Sections Prescription-based co-occurrence graph modeling and Relative size graph modeling, relatively.

Prescription-based co-occurrence graph modeling. We propose to leverage an external source, namely prescriptions, to build the co-occurrence graph. The rationale behind our idea is that, as most pills are intended to cure or alleviate certain diseases or symptoms, there is a significant likelihood that pills meant to treat the same diseases will appear concurrently. Thus, the implicit relationship between the pills can be modeled by assessing the direct interaction between medications and diseases derived through prescriptions. Our Co-graph, $\mathcal{G}_c = \langle V, E, W_c \rangle$, is a weighted graph whose vertices V represent pill classes, and whose edges' weights W_c reflect the co-occurrence likelihood of the pills. As the association between pills is not explicitly present in the prescriptions, we model this relationship utilizing the interaction between medications and diseases using 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 disease.

- The greater the weight of an edge E_{ij} connecting pill classes C_i and C_j, the more likely that these two medications will be prescribed simultaneously.

We first define a so-called `Diagnose-Pill` impact factor, which reflects how important a pill is to a diagnosis. Inspired by the Term Frequency (`tf`)—Inverse Dense Frequency (`idf`) often used in the Natural Language Processing domain, we define the impact factor of a pill P_j to a diagnosis D_i, denoted as $\mathcal{I}(P_j, D_i)$, as follows

$$\mathcal{I}(P_j, D_i) = \mathtt{tf}(D_j, P_i) \times \mathtt{idf}(P_i) = \frac{|\mathbb{S}(D_j, P_i)|}{|\mathbb{S}(D_j)|} \times \log\frac{|\mathbb{S}|}{|\mathbb{S}(P_i)|}, \tag{1}$$

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, $\mathtt{tf}(D_j, P_i)$ measures how often pill P_i is prescribed for diagnosis 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 `idf`(P_i).

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

$$p(P_j, D_i) = \frac{\mathcal{I}(P_j, D_i)}{\sum_{D \in \mathbb{D}} \mathcal{I}(P_j, D)}, \tag{2}$$

where $\mathbb{D}$ denotes the set of all diagnoses. Given $p(P_j, D_i)$, we define the weight $W_c(P_i, P_j)$ of the edge E_{ij} connecting vertices P_i and P_j as the probability $p(P_i, P_j)$ that P_i and P_j are prescribed for the same diseases. $W_c(P_i, P_j)$ can be formulated as follows.

$$W_c(P_i, P_j) := p(P_i, P_j) \approx \sum_{D \in \mathbb{D}} p(P_i, D) \times p(P_j, D). \tag{3}$$

Relative size graph modeling. The Size-graph is represented by a directed graph $\mathcal{G}_s = \langle V, E, W_s \rangle$. The edge weight W_s is modeled so that the weight of an edge $\vec{E}_{ij}$ connecting from P_i to P_j is proportional to the size ratio of P_i to P_j. The primary source for constructing the Size-graph is the annotations of the training dataset's bounding boxes. As the camera locations for multiple pictures are different, the exact size of each bounding box cannot be utilized directly. Therefore, we instead define a so-called *size indicator*, a normalized representation of pill size, which is determined as follows.

- **Step 1**: We begin with an arbitrary pill class by initializing its size indicator to 1, while those of other pill classes are initialized to 0.

- **Step 2**: From the current node P_i, we traverse through all its 1-hop neighbors P_j, and calculate P_j's size indicator s_j as $s_j := s_i \times \frac{|B_j|}{|B_i|}$, where B_i, B_j are the two bounding boxes of P_i and P_j in a particular image in the training set. Step 2 is repeated until all the vertices of $\mathcal{G}_s$ are traversed.

Given the size indicators of all vertices, we now define the weight of edge $\vec{E}_{ij}$ as the ratio of s_i to s_j.

Visual feature extractor

This block is responsible for localizing and extracting the features of Regions of Interest (RoIs). For this purpose, we adopt components from Faster RCNN [15], a conventional two-step object detector architecture. Nevertheless, our proposed framework is compatible with any alternative object detection architecture. The Visual Feature Extractor consists of three components: a Convolutional Network, a Region Proposal Network, and an RoI Pooling Layer, as depicted in Fig 3. RPN is a fully convolutional network that takes the visual feature vector from the previous module and generates proposals with various scales and aspect ratios. The RoI Pooling layer works simply by splitting each region proposal into a grid of cells and then applying the max pooling operation to each cell in the grid. The combination of the grids' values forms the visual feature vectors of the RoIs.

Inter-pill relational feature extractor

To enhance the efficacy of this a priori, we observed that rather than the whole graphs representing the interaction between all pills, we should utilize sub-graphs concentrating on the ones most likely to appear in the image. Motivated by this observation, we employ the Inter-Pill Relational Feature Extractor, responsible for extracting condensed sub-graphs from generic Co-graph and Size-graph. Moreover, previous studies have pointed out that the appearance of pills conveys implications about their efficacy or ingredients [16]. In light of this, utilizing pills' visual feature vectors, we develop a visual-based graph that models the implicit relationship between medications indicated by their visual appearance.

Condensed co-graph and size-graph. Our main idea is to employ a so-called *Pseudo Classifier*, which provides approximate classification results using solely visual features of RoIs.

Effectively, the pills could be divided into two categories: simple samples and difficult samples. Pseudo Classifier can readily recognize the former group since they possess distinguishable visual characteristics. However, the latter require extra information about nearby tablets to help in their recognition. These temporary identification results produced by our Pseudo Classifier are then utilized as a filter layer to eliminate redundant information from the original Co-graph and Size-graph, leaving only relevant contextual information about pill classes probable to appear in the input image.

In our current implementation, the pseudo classifier is straightforwardly implemented as a fully connected layer. Let N be the number of pill classes and M be the number of pill bounding boxes (i.e., RoIs) in the input image. Suppose $P = [p_{ij}]_{M \times N}$ is the matrix whose row vectors represent the logits produced by the pseudo classifier, and $\mathcal{A}_c = [a_{kl}^c]_{N \times N}$, $\mathcal{A}_s = [a_{kl}^s]_{N \times N}$ denote the weighted adjacency matrices of the Co-graph $\mathcal{G}_c$ and Size-graph $\mathcal{G}_s$, respectively. The condensed adjacency matrices, denoted as $\tilde{\mathcal{A}}_c$ and $\tilde{\mathcal{A}}_s$ are matrices of size $M \times M$; each row depicts the condensed relational information of a pill, i.e., a specific RoI, with others in the input image. $\tilde{\mathcal{A}}_c$ and $\tilde{\mathcal{A}}_s$ are obtained by performing a composition of matrix multiplications as follows.

$$\tilde{\mathcal{A}}_c = \sigma(P) \cdot \mathcal{A}_c \cdot \sigma(P)^T \tag{4}$$

$$\tilde{\mathcal{A}}_s = \sigma(P) \cdot \mathcal{A}_s \cdot \sigma(P)^T, \tag{5}$$

where σ denotes the `Softmax` activation function. Intuitively, the item in the i-th row and j-th column of $\tilde{\mathcal{A}}_c$ and $\tilde{\mathcal{A}}_s$ highlights the relationship between the i- and j-th RoIs.

Visual semantic graph. As mentioned above, the visual semantic graph $\tilde{\mathcal{G}}_v = \langle \tilde{V}, \tilde{E}_v, \tilde{W}_v \rangle$ is in charge of capturing the visually semantic correlation among pills in the input image. The detailed algorithm to construct this graph is as follows. All visual feature vectors are first passed through a non-linear function $\mathcal{F} : \mathbb{R}^h \to \mathbb{R}^{h'}$ to transform from the original h-dimensional space into a h'-dimensional latent one, where their relationship can be best presented. The latent output vectors are then directly used for calculating the correlations between RoIs. Let R_i, R_j be two RoIs in the input image, and z_i, z_j are their feature vectors created by the Visual Feature Extractor block, the weight of the edge connecting R_i and R_j is defined as $\tilde{W}_v(R_i, R_j) = z_i \cdot z_j$.

Multi-modal data fusion

After going through the second and third blocks, we get the visual features of the RoIs and three relational graphs representing the relationships between the RoIs. This information is now fed into the Multi-modal Data Fusion to generate the final feature vectors, each of which encapsulates both the intra-Pill visual characteristic of an RoI and the inter-Pill interaction of that RoI with the others. The Multi-modal Data Fusion comprises two steps: graph embedding and data concatenation. The former obtains the heterogeneous relational graph $\mathcal{G}$ and transforms it into context features in the vector space, while the latter concatenates the context feature vectors with visual features to generate the final enhanced features. We utilize the Graph Transformer Network (GTN) [17] for graph embedding. The reason for choosing the GTN is due to its ability to handle heterogeneous input and adaptive graph structures. Before going into the details of the GTN, it is crucial to define the node attribute of graph $\tilde{\mathcal{G}}$. As each node of $\tilde{\mathcal{G}}$ represents an RoI, the node attribute should be the most representative characteristic of the ROIs. Using the retrieved RoI visual features to depict the relevant ROIs is the most

natural solution but is not advantageous due to several factors, including the unreliability in dealing with ambiguous samples or the *intra−variance* in visual features of one class [18]. To this end, classifier weights have been introduced as a simple yet effective alternative. According to [19], the classifier weights connected to the i-th neuron in the last layer (which is denoted as $\omega_i = [\omega_{1i}, \ldots, \omega_{Hi}]^T$ in Fig 4(a)) correspond to the i-th pill class, encapsulating the representative characteristics of this class. Let $p_k = [p_{k1}, \ldots, p_{kN}]$ be the logit vector of the k-th RoI, where p_{ki} depicts the likelihood for the k-th RoI to be classified into the i-th class; we define $\sum_{i=1}^{H} p_{ku} \times \omega_i$ the attribute of the k-th RoI. Intuitively, this attribute can be considered a decomposition of the RoI's characteristic in the space of the classes' features.

Fig 4(b) depicts the GTN's architecture, which consists of two phases. The former can be seen as a meta-path generator that fuses information from multiple input adjacency matrices to generate a composite graph structure. This newly generated graph serves as the second stage's input, which comprises a Graph Convolutional Network (GCN) and is responsible for producing a representation for each node.

Specifically, the GTN consists of l Graph Transformer (GT) layer; the l-th layer applies the C-channel 1D convolution operation on the input graph $\tilde{\mathcal{G}}$ to obtain a stack of new graph structure $\mathbb{Q}_1^{(l)} \in \mathbb{R}^{M \times M \times C}$ as follows.

$$\mathbb{Q}_1^{(l)} = F(\tilde{\mathcal{G}}; W_\phi^{(l)}) = \phi(\tilde{\mathcal{G}}; \sigma(W_\phi^{(l)})), \tag{6}$$

where ϕ indicates the convolution layer, $W_\phi^{(l)} \in \mathbb{R}^{C \times 1 \times 1 \times K}$ represents the parameter of ϕ, and K implies the number of relations contained in the original graph $\tilde{\mathcal{G}}$. The stacked graph $\mathbb{Q}_1^{(l)}$ serves as the first component in creating length l meta-paths, while $\mathbb{Q}_2^{(l)}$ is taken as $\tilde{\mathcal{G}}^{(l-1)}$, i.e., $\tilde{\mathcal{G}}^{(l)} = \mathbb{Q}_2^{(l)} \odot \mathbb{Q}_2^{(l)}$. To balance computational overheads and model performances, with PGPNet, we fix $l = 2$.

The resulting graph $\tilde{\mathcal{G}}^{(2)}$, together with RoIs' representative features X_{RoI}, are then utilized as the input for the Graph Convolution Network (GCN) to generate the final node presentations. These vectors are directly concatenated with their corresponding RoIs' visual features before getting fed into the Bounding Box Regressor and Classifier to produce the final detection results.

(a)   (b)

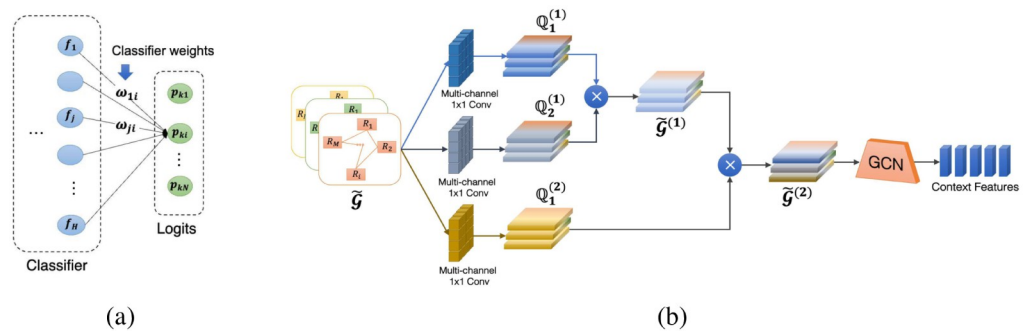**Fig 4. Illustration of Graph Transformer Network.** (a) **Node attribute modeling**. The classifier weights corresponding to each pill class capture the representative features of this class. (b) **Graph Transformer Network (GTN) architecture** [17]. GTN selects adjacency matrices from a set of matrices for a heterogeneous graph and learns new meta-path graphs by multiplying two selected adjacency tensors.

PGPNet's losses

This section presents details about our model's objectives and the corresponding losses to achieve those goals. We employ two types of losses. The former is the conventional two-step object detector's loss, which determines pills' bounding boxes and produces the final pill classification result. While the latter is our proposed auxiliary loss that utilizes the co-occurrence graph to enhance the accuracy of the Pseudo Classifier.

Two-step object detectors' losses. *The region proposal network's losses.* The loss for Region Proposal Network consists of two components: classification loss combined and bounding box regression loss. Let p_i, p_i^* be the predicted probability of anchor i being an object and the ground truth label whether anchor i is the object, respectively; t_i and t_i^* depict the differences of four predicted coordinates, and the ground truth coordinates with the coordinates of the anchor boxes, respectively. The classification loss $\mathcal{L}_{cls}$ and bounding box regression loss $\mathcal{L}_{box}$ are defined as follows.

$$
\begin{aligned}
\mathcal{L}_{RPN}(\{p_i\}, \{t_i\}) \quad &= \frac{1}{N_{cls}} \sum_i \mathcal{L}_{cls}\left(p_i, p_i^*\right) \\
&+ \frac{\lambda}{N_{box}} \sum_i p_i^* \cdot L_1^{smooth}\left(t_i - t_i^*\right),
\end{aligned}
\tag{7}
$$

where

$$
L_1^{smooth}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}.
\tag{8}
$$

Here $\mathcal{L}_{cls}$ is a binary classification *log loss*, N_{cls} and N_{box} are two normalization terms, where N_{cls} is set to the mini-batch size, while N_{box} is the number of anchor boxes. λ is a hyper-parameter, which is responsible for balancing between $\mathcal{L}_{cls}$ and $\mathcal{L}_{box}$.

Output's losses. The PGPNet's final results consist of the coordinates of the RoIs' bounding boxes and predicted labels for the RoIs. We employ two distinct losses to accomplish this objective. While the loss for a bounding box regressor is equal to that of the RPN network, the classification loss $\mathcal{L}_{cls}^{out}$ is instead the cross entropy loss for the multilabel classification task, which is represented as follows $\mathcal{L}_{cls}^{out} = -\sum_{i=0}^{N} p_i^* \log(p_i)$.

Triplet co-occurrence enhancement loss. In this section, we propose an auxiliary loss named Triplet Co-occurrence Enhancement Loss which leverages the co-occurrence graph to boost the accuracy of the Pseudo Classifier. The idea behind the auxiliary loss is that it encourages the co-occurrence likelihood of pills that are close together on the co-occurrence graph. To this end, we construct our auxiliary loss as a contrastive loss that maximizes the co-occurrence probability of positive pairings (i.e., pills joined by edges with the most significant weights in the co-occurrence graphs) while minimizing the co-occurrence probability of negative pairs (i.e., pills that are not connected or connected by edges with the smallest weights). In action, for each training mini-batch, PGPNet would treat all the ground truth pills in given images as the set of anchors and build up their corresponding positive as well as negative sets. After that, Triplet Co-occurrence Enhancement Loss would do its job of enhancing the robustness of the Pseudo Classifier. The detail of the auxiliary loss is as follows.

Let's denote the i-th Region of Interest as R_i with its corresponding label of l_i. Moreover, let N_{pos}^i and N_{neg}^i be the positive and negative samples of R_i, where N_{pos}^i comprises $k + 1$ nearest neighbors and N_{neg}^i consists of $k + 1$ furthest neighbors of R_i. We suppose that the groundtruth labels of N_{pos}^i and N_{neg}^i are $L_{pos} = \{l_{pos}^0, l_{pos}^1, \ldots, l_{pos}^k\}$, and $L_{neg} = \{l_{neg}^0, l_{neg}^1, \ldots, l_{neg}^k\}$, respectively.

The auxiliary loss concerning the i-th RoI is defined by

$$
\begin{aligned}
\mathcal{L}_{aux}^i \quad &= p_i(l_i)p(N_{pos}^i) - (1 - p_i(l_i))p(N_{neg}^i) \\[2mm]
&= p_i(l_i)\sum_{j=0}^{k}[1 - \prod_{m=0}^{M}(1 - p_m(l_{pos}^{ij}))] \\[2mm]
&\quad -(1 - p_i(l_i))\sum_{q=0}^{k}[1 - \prod_{n=0}^{M}(1 - p_n(l_{neg}^{iq}))],
\end{aligned}
\tag{9}
$$

and those for RoI is $\mathcal{L}_{aux} = \sum_{i=0}^{M} \mathcal{L}_{aux}^i$. In Formula (9), M is the total number of RoIs in the image, p is the output after going through `softmax` activation of logits produced by the Pseudo Classifier. The objective during the training process is to maximize $\mathcal{L}_{aux}$, which in turn maximizes each positive term $p_i(l_i)p(N_{pos}^i)$ while minimizing the negative opposition $(1 - p_i(l_i))p(N_{neg}^i)$.

Dataset and experiment settings

We conduct extensive experiments to validate the effectiveness of the proposed approach. In the following, we first introduce our in-house pill identification dataset, called VAIPE, which will be used to evaluate the proposed approach, and then explain our evaluation metrics and experimental settings. To assess the effectiveness of the proposed method, we conducted comparative assessments against a number of established models, including the detection backbones we selected, such as Faster R-CNN [15] and YOLOv5 [20], as well as other related frameworks such as SGRN [21] and the Mask RCNN-based approach described in [7]. We also perform ablation studies to investigate the efficiency of key components in our framework.

Dataset and pre-processing

Motivation. To the best of our knowledge, previous studies on the pill identification problem [6, 22, 23] only focus on datasets collected in constrained environments. For instance, existing datasets such as NIH Dataset [14] are constructed under ideal conditions in lighting, backgrounds, and equipment or devices. The CURE dataset [6] provides only one pill per image. Hence, these datasets do not reflect the real-world scenarios in which patients take an arbitrary number of drugs, and their environmental conditions (e.g., backgrounds, lighting conditions, mobile devices, etc.) are greatly varied. Additionally, many pills have nearly identical visual appearances. The fact that they appear alone in the images of these datasets will inevitably confuse the detection frameworks. Consequently, none of the existing datasets can be directly applied to the real-world pill detection problem or can only be applied with low reliability. There is no publicly available dataset of these pills images in which the pills follow the intakes of actual patients. This limits the development of machine learning algorithms for the detection of pills from images as well as for building real-world medicine inspection applications. To address this challenge, we build and introduce a new, large-scale open dataset of pill images, which we called VAIPE.

Data descriptor. The VAIPE is a large-scale and open pill image dataset for visual-based medicine inspection. The dataset contains approximately 10,000 pill images that were manually collected in unconstrained environments. In this study, no hypotheses or new interventional procedures were generated. Also, no investigational products or clinical trials were used on patients. In addition, there were no changes in treatment plans for any patients involved.

Pill images were retrospectively collected, and all identifiable information about patients was de-identified. Therefore, there was no requirement for ethics approval [24].

Pill images are collected in many different contexts (e.g., various backgrounds, lighting conditions, in-hand or out-of-hand, etc.) using smartphones. These images are then manually labeled using the information from the relevant prescriptions. In summary, the number of pills per image is about $5 - 10$, and the total number of pill images collected was 9, 426 pill images with 96 independent pill labels. To train the proposed deep learning system, the pill images from the VAIPE dataset are resized so that the shortest edges have a size of 800, with a limit of 1, 333 on the longer edge. The ratios are kept the same as the original images if the max size is reached, then downscaled so that the longer edge does not exceed 1, 333.

Data validation. Patient privacy was controlled and protected. In particular, all images were manually reviewed to ensure that all individually identifiable health information of the patients has been removed to meet the General Data Protection Regulation (GDPR) [25]. Annotations of pill images were also carefully examined. Specifically, all images were manually reviewed case-by-case by a team of 20 human readers to improve the quality of the annotations.

Comparison with existing datasets. Table 1 provides a summary of the aforementioned datasets (including NIH, CURE, and VAIPE) together with other ones of moderate sizes, meta-data, and other properties. Compared to the two previous datasets, the VAIPE dataset is constructed under a much more flexible procedure that reflects the characteristic real-world data distributions. Hence, the introduced dataset can serve as a reliable data source for training *generic pill detectors*.

Evaluation metrics

We evaluate the proposed method and other related works by the COCO APs metrics [26]. This set of metrics is widely accepted and used for evaluating state-of-the-art object detectors. Mean Average Precision (mAP), as its name suggests, is the mean of Average Precision (AP) overall C classes and all the targeted IoU thresholds in the threshold set T calculated by $mAP = \frac{1}{C|T|} \sum_i^C \sum_{t \in T} AP_{i,t}$, where Average Precision ($AP_{i,t}$) is the area under the Precision-Recall curve, calculated for the class i at a given IoU threshold t.

Comparison with state-of-the-art methods

Comparison benchmarks. To show the effectiveness of the proposed method, we conducted a comparison with the state-of-the-art object detectors, including our detection backbones: Faster R-CNN [15], YOLOv5 [20], and related works: SGRN [21], Mask RCNN-based approach [7]. Throughout the literature, the baseline with which PGPNet presently integrates

Table 1. An overview of existing public datasets for the task of image-based pill detection.

	NIH	CURE	VAIPE
Number of pill images	7,000	8,973	9,426
Number of pill categories	1,000	196	96
Number of capture devices	1	1	> 20
Instance per category	7	40-50	> 30
Illumination conditions	1	3	> 50
Backgrounds	1	6	> 50
Number of prescriptions	0	0	1,527

https://doi.org/10.1371/journal.pone.0291865.t001

Table 2. Details of training and testing datasets.

Training dataset		Testing dataset	
Prescriptions	Images	Prescriptions	Images
1,527 (100%)	7,514 (78%)	0 (0%)	1,912 (22%)

https://doi.org/10.1371/journal.pone.0291865.t002

is Faster R-CNN [15]; hence, the original framework is utilized for our comparison. We adopt two different CNNs and one Transformer-based module for visual feature extractor, namely ResNet-50-C4, ResNet-50-FPN and Swin Transformer V2—SwinV2 [27]. Specifically, for two ConvNets, we use a single feature map produced by convolution block C4 of the ResNet-50 model in ResNet-50-C4. In ResNet-50-FPN, we replace C4's feature map with multi-scale feature maps produced by Feature Pyramid Network (FPN) [28]. As for the Swin Transformer module, we are currently utilizing the SwinV2-T configuration [27] to ensure that the number of model parameters is comparable to that of ResNet-50. In addition, we also make adaption for PGPNet with YOLOv5 [20] detection backbone. Two configurations of YOLOv5s and YOLOv5n are currently adopted. Also, the most relevant frameworks compared with our PGPNet are also put into comparison: a representative approach that utilizes an external knowledge graph for Object Detection task [21]; a Mask RCNN-based baseline that also proposed to solve the same task of multi-pill detection [7]. For a fair comparison, a fixed set of hyper-parameters is used for PGPNet throughout all experiments.

Implementation details

We conduct all the experiments using Pytorch (version 1.10.1) on an Intel Xeon Silver 4210 2.20GHz system with $2 \times$ NVIDIA GeForce RTX 3090 GPUs. We train and test all targeted models on the training and testing sub-datasets of the VAIPE dataset provided in Table 2. Specifically, we initialize all the networks with the weights achieved by pre-training them on COCO 2017 dataset [29]. We then train the models in 20, 000 iterations with a batch size of 16 without the use of Early Stopping strategy. AdamW [30] optimizer is used with an initial learning rate of 0.001. We also preprocess the training data by applying simple augmentation techniques: random horizontal and vertical flips; random rotation to prevent overfitting. During the evaluation process, no augmentation is used. For our PGPNet implementation, we set the dimensions of node embeddings at 64. We also design the Graph Transformer Module with only one layer and 10 channel set.

Experimental results

This section reports our experimental results. We evaluate the effectiveness of PGPNet in three aspects: robustness, reliability, and explainability. The details are described below.

Robustness and reliability of PGPNet

Comparison with Faster R-CNN and YOLOv5. *Detection performance.* Table 3 shows the experimental results of PGPNet and the state-of-the-art object detector frameworks (Vanilla), e.g., Faster R-CNN (two-step detector), and YOLOv5 [20] (one-step detector) on the VAIPE dataset. As shown, PGPNet obtained better results than Faster R-CNN by large performance gaps for all evaluation metrics. Specifically, when using the ResNet-50-C4 model as the visual feature extractor model, the average precision mAP of Faster R-CNN was 62.6, while that of PGPNet was 68.3. The proposed method improves the performance over the baseline Faster R-CNN by 9.2%. Under strict metrics, e.g., AP75, PGPNet also outperforms Faster

Table 3. Comparing PGPNet's detection performance with state-of-the-art vanilla object detectors on VAIPE dataset. Best results are highlighted in **bold** text.

Method			mAP	AP50	AP75	APs	APm	APl
Two-step	Faster R-CNN (ResNet-50-C4)	Vanilla	62.6	87.0	74.4	75.0	58.3	62.9
		PGPNet	**68.3** (+9.2%)	**92.5**	**81.7**	**80.0**	**64.3**	**68.7**
	Faster R-CNN (ResNet-50-FPN)	Vanilla	63.7	86.6	76.9	71.2	58.1	64.6
		PGPNet	**69.7** (+9.4%)	**94.4**	**83.4**	**90.0**	**66.4**	**70.1**
	Faster R-CNN (SwinV2-T)	Vanilla	59.7	84.5	72.3	66.9	54.0	60.1
		PGPNet	**62.6** (+4.8%)	**87.2**	**75.5**	**68.6**	**56.6**	**62.9**
One-step	YOLOv5n	Vanilla	37.9	50.8	45.4	87.5	49.1	38.3
		PGPNet	**43.0** (+12.0%)	**58.4**	**51.3**	**82.5**	**52.4**	**43.7**
	YOLOv5s	Vanilla	57.5	75.8	68.3	85.0	58.3	57.0
		PGPNet	**63.4** (+10.2%)	**85.9**	**76.4**	**89.9**	**58.3**	**64.1**

R-CNN 8–9%. In addition, we observed similar behavior when using the ResNet-50-FPN model. The proposed PGPNet makes an improvement of 9.4% for the mAP metrics. With a Transformer-based backbone, here a Swin Transformer V2 configuration—SwinV2-T [27], the results are slightly worse compared to those produced by ResNet-based counterparts, for both the vanilla and PGPNet alternatives. However, PGPNet still shows its superior when being installed with this backbone, as the empirical result for AP metrics is improved by 4.8% compared to the vanilla SwinV2-T Faster R-CNN model.

For YOLOv5, PGPNet outperformed Vanilla by a significant margin across all performance metrics in both YOLO instances. Specifically, the average precision AP of the vanilla model with YOLOv5n was 37.9, while that of PGPNet was 43.0 (12% improvement). In the case of a larger alternative, YOLOv5s, a similar conclusion can be drawn, namely that PGPNet improves overall mAP metrics by 5.9, e.g., 10.2%.

Fig 5 visualizes the AP for all classes in the dataset when using Faster R-CNN as the backbone. The first three bins denote Faster R-CNN alternatives, and the latter three are the corresponding PGPNet configurations. The dots in the figure represent AP values for classes; the vertical line is the indicator for the mean value, while the rectangle bar is the 95% High-Density Interval (HDI) band. Apart from the fact that the mean AP over all classes of PGPNet
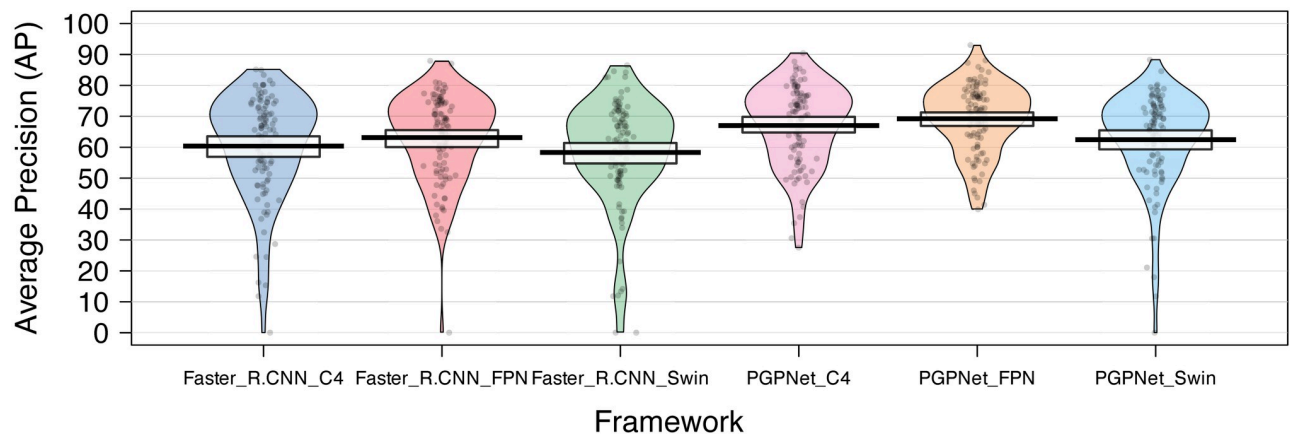

Fig 5. Comparison of the PGPNet performance with the Faster R-CNN baseline over each individual class.

variances is better than those produced by Faster R-CNN, we found that PGPNet also has more reliable and stable results over all classes. Specifically, PGPNet helps to improve the AP of classes that Faster R-CNN frequently confuses (the points with low APs in the blue and pink beans). As a result, the three beans of Faster R-CNN exhibit a large variance, i.e., the AP ranges from 0 to around 90. In contrast, the beans of PGPNet performance are more condensed and have shorter tails, i.e., the AP ranged from 40 (or 50) to around 90. By integrating both the visual data presented in Fig 5 and the numerical results outlined in Table 3, we can observe that our PGPNet model has effectively enhanced the detection accuracy for hard labels (through the introduction of reduced variances in the AP scores of different classes as illustrated in Fig 5), leading to improved overall AP scores as shown in Table 3.

Pill classification accuracy. To further investigate the robustness of the proposed PGPNet, we adopt the visualization techniques presented in [31] to understand the prediction accuracy (of the pill classification task) better. In this technique, all models' predictions are categorized by their confidence scores into different bins, in which the average accuracy can be calculated. By observing the confidence-accuracy correlation, we can tell whether the models are under or overconfident with their predictions [31]. Fig 6(a) visualizes those reliability plots of Faster R-CNN and PGPNet. It implies that both models have a propensity toward overconfidence, as the average accuracy of each confidence band is lower than the mean confidence score of that bin. However, that tendency is greatly alleviated in the circumstance of PGPNet, which means that the bins' heights are much closer to the perfect Confidence-Accuracy balance line (the red dashed diagonal line). Fig 6(b) compares PGPNet's confidence-accuracy correlation with that of YOLOv5. With this backbone, we observed that the proposed PGPNet could produce predictions with a high level of reliability. All the heights of bins are much closer to values suggested by the perfectly-balanced line compared to Vanilla's result.

Comparison with existing relevant frameworks. Our work is the first to leverage an external graph in dealing with the Pill Detection challenge; thus, none of the preceding works are genuinely tight-correlated. Indeed, earlier research only shared some common ground with our approach: (1) About methodology or (2) about the research problem.

For the first group, there are works that utilized external information to solve the Object Detection problem. We adopt one of the most current studies in this direction—[21] to solve our targeted problem and serve as a baseline for PGPNet. Spatial-aware Graph Relation Network (SGRN) [21] is a framework that adaptively discovers and incorporates key semantic and spatial relationships for reasoning over each RoI.

With respect to the research problem, as stated earlier, while there are many works that target the single-pill detection problem [4–6], only a few directly solve the task of detecting
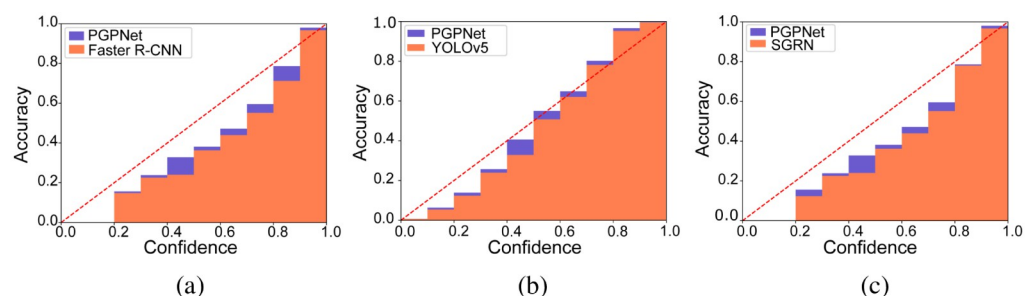

Fig 6. Reliability investigation for PGPNet and different baseline performances. (a) Faster R-CNN. (b) YOLOv5. (c) SGRN.

https://doi.org/10.1371/journal.pone.0291865.g006

Table 4. Performance comparison of PGPNet with SGRN and Kwon's Pipeline.

Model	mAP	AP50	AP75	APs	APm	APl
Faster RCNN [15]	63.7	86.6	76.9	71.2	58.1	64.6
SGRN [21]	65.9	88.8	79.6	76.3	61.6	66.3
Kwon's Pipeline [7]	36.2	38.5	37.2	30.3	33.1	36.0
PGPNet	**69.7**	**94.4**	**83.4**	**90.0**	**66.4**	**70.1**

multiple pills per image [7, 13]. We attempt to adopt the most recent technique proposed in [7] as another baseline to compare with PGPNet. In the original work, the authors' purpose is somewhat different from ours, since they attempt to develop a framework that is solely trained on single-pill images since they argue that the multi-pill dataset would scale up exponentially if the number of pills increased. This argument is not held in our intuition, and we believe, in reality, that the pills taken together have to be prescribed by pharmacists. We keep the pipeline as the original work, with some adoptions for working with our VAIPE dataset: (1) Change Mask R-CNN to Faster R-CNN; (2) The training single-pill dataset is cropped from our VAIPE dataset with bounding box annotations; (3) The automated data labeling process is skipped. Since the original work did not name the proposed pipeline, we called it *Kwon's Pipeline* for short.

Detection performance. Table 4 summarizes the comparison of PGPNet, SGRN and Kwon's Pipeline when adopting the visual feature extractor architecture from Faster R-CNN with the Resnet-50-FPN model.

Clearly, SGRN outperforms the baseline Faster R-CNN in terms of overall performance but could not outperform our proposed method PGPNet. Specifically, the mAP metrics achieved by SGRN is 65.9, and PGPNet achieves the better score with a gap of nearly 4. Upon other metrics, AP50, AP75, APs, APm, and APl, PGPNet shows its superiority by enhancing the performance from 5.1% (e.g., in AP75 metrics) up to 17.1% (e.g., in APs metrics). This is an expected result because SGRN reveals a major weakness when applying to the challenge of Pill Detection. The spatial relationships between pills in an image are arbitrary and frequently changed. Such noisy and unreliable information leads to the performance of SGRN being unstable and sometimes not producing good enough results. In the case of Kwon's Pipeline, the situation is even worse since it cannot even beat the vanilla one-step Faster RCNN trained with the mutple-pill VAIPE training set. The result of this pipeline is 43.1% and 48.2% worse than vanilla Faster R-CNN and PGPNet, respectively. One reason for this deficiency is owing to the quality of its training data. There are many circumstances in which overlap or occlusion occurs, which makes the cropped images also contain parts of other pills.

Pill classification accuracy. Fig 6(c) shows the correlation between the confidence and accuracy of PGPNet in comparison with those of SGRN. Both the frameworks are based on the Faster R-CNN backbone and achieve similar results, e.g., an overconfidence trend in every bin. All the predictions with confidence scores smaller than 0.2 are totally unreliable (with 0% accuracies). In addition, PGPNet also shows its superiority over SGRN in some bins, in which the overconfidence situation is reduced effectively. We do not plot the Confidence-Accuracy of Kwon's Pipeline owing to the apparent performance gap compared to our PGPNet.

Ability in dealing with hard samples. In the following, we investigate the ability of PGPNet to deal with the occlusion phenomenon caused by overlapping pills, which is one of the most critical issues in dealing with multi-pill detection. To this end, we create a so-call *custom occlusion sub-dataset* of VAIPE, which contains images with heavy occlusion phenomena, i.e., having at least two RoIs with the IoU beyond 30% (Fig 7). We also create a *custom non-*

Fig 7. Images with occlusion phenomena in custom occlusion dataset. The rectangles depict examples of tablets with overlapping boundary boxes.

https://doi.org/10.1371/journal.pone.0291865.g007

occlusion sub-dataset which contains samples that are in the same classes that appear in the *custom occlusion sub-dataset* but with no occlusion. The quantitative result is summarized in Table 5. The (-) mark in the table suggests disregarded or unavailable metrics.

As the numbers suggest, even in cases where heavy occlusion occurs, PGPNet still shows its superiority over Faster R-CNN. Specifically, the mAP over all classes in the *custom occlusion sub-dataset* suggests a gap of 8.3% between the two approaches. Interestingly, with the aid of classifier weight as the distinguishing characteristic for each class, PGPNet, even when dealing with occlusion cases, still enhances the performance by 1.9% compared to Faster R-CNN handling the non-occlusion case (e.g, 67.5 vs. 65.6, respectively). Fig 8 provides more information about the AP for each class in the *custom occlusion sub-dataset*. PGPNet still outperforms Faster R-CNN in most cases with a large gap and also produces a more reliable result by introducing a smaller variance over the AP metrics.

PGPNet's explainability

This section is dedicated to analyzing the results produced by PGPNet through a specific sample. This example demonstrates that the operation of PGPNet is very congruent with our initial motivation and that our designed architecture can materialize this motivation.

Experiment settings.   In this experiment, we choose a hard sample, namely *Hexinvon-8mg*, with a relatively common appearance, for investigation. Fig 9 visualizes *Hexinvon-8mg* together with other pills in our dataset with almost identical visual appearance (round shape, white tint, etc.).

As illustrated, these pills are readily confused with *Hexinvon-8mg*. Indeed, Fig 10 depicts an example in which *Hexinvon-8mg* is miscategorized as *Alpha-Chymotrypsine* by Faster R-CNN. Our PGPNet can, however, successfully distinguish *Hexinvon-8mg* with a high confidence score.

Table 5. Impact of heavy occlusion images on testing performance of PGPNet and Faster R-CNN.

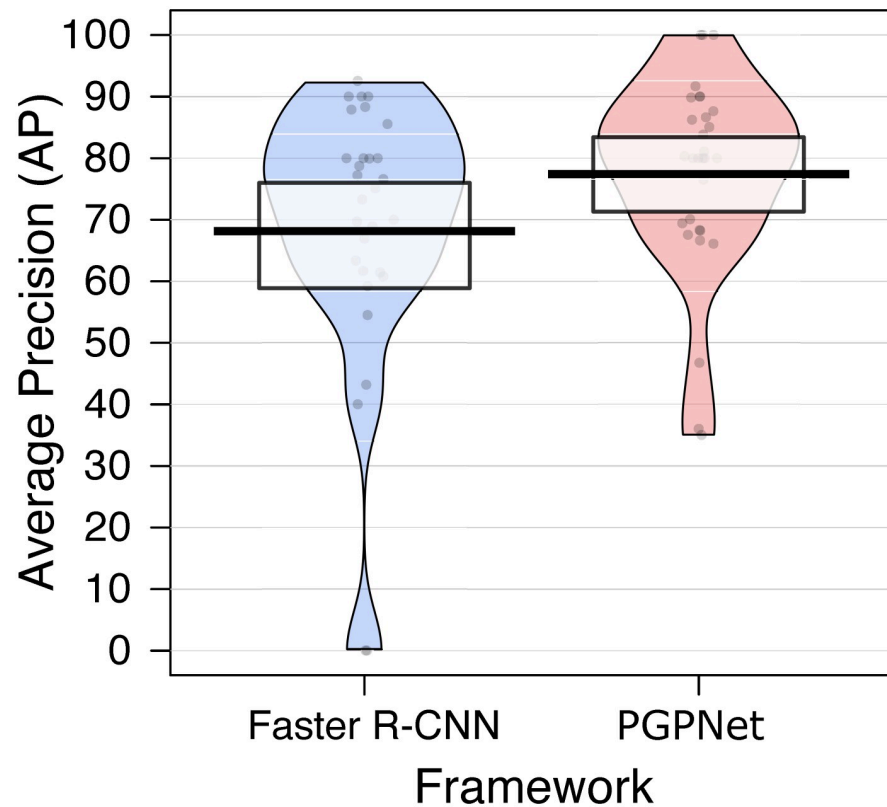| Test dataset	Custom Occlusion		Non-Occlusion	
Method	Faster R-CNN	PGPNet	Faster R-CNN	PGPNet
mAP	59.2	**67.5**	65.6	**71.7**
AP50	76.5	**81.1**	87.4	**92.9**
AP75	68.9	**76.4**	80.8	**87.0**
APs	-	-	80.0	**90.0**
APm	61.6	**68.3**	56.5	**64.7**
APl	60.7	**70.1**	65.1	**70.6**

https://doi.org/10.1371/journal.pone.0291865.t005

Fig 8. Comparison of PGPNet performance with Faster R-CNN over each individual class in occlusion dataset.

In the following, we applied several Explainable AI techniques to explain the results inferred by our PGPNet. The image of interest consists of three pills: LIVOLIN-FORTE, Hapenxin, and Hexivon, as shown in Fig 10.

Explanation of the prediction results. We adopt the Excitation Backpropagation technique proposed by Zhang [32] to construct the saliency maps (Fig 11), which indicate what the classifier has learned to produce the final results.

Firstly, for the easy samples, i.e., LIVOLIN-FORTE and Hapenxin, our model focuses precisely on those pill regions to make the prediction decision. In contrast, in the case of the hard sample, i.e., *Hexinvon-8mg*, however, two regions are highlighted: one at the position of

Hexivon-8mg   Loratadine-10mg   Enalapril-5mg   Alpha-Chymotrypsine   Medrol-16mg

Fig 9. Some sample pills with very identical visual appearance with *Hexinvon-8mg*.
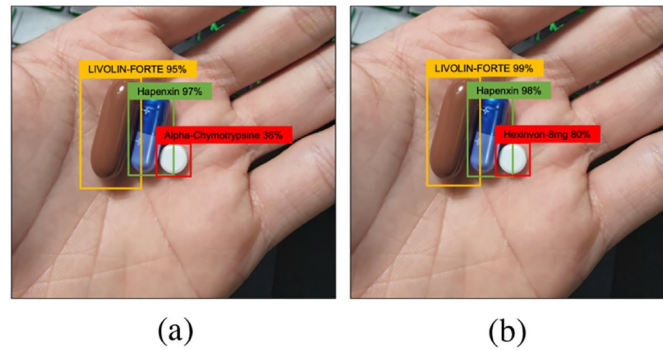
Fig 10. Predictions for a hard sample made by Faster R-CNN and PGPNet given the same image. (a) Faster R-CNN. (b) PGPNet.

Hexinvon-8mg and the other at the location of LIVOLIN-FORTE. It indicates that the classifier solely requires information about LIVOLIN-FORTE and Hapenxin to identify these pills. Nevertheless, for *Hexinvon-8mg*, the classifier must additionally incorporate information about its neighbor, i.e., LIVOLIN-FORTE. This hypothesis is also supported by the Probabilistic score matrix shown in Fig 12. The probabilistic score matrix represents the prediction results generated by our Pseudo Classifier, which relies mainly on the pill's visual characteristics. As demonstrated, Pseudo Classifier can accurately detect the proper labels of two simple samples, with their prediction scores approaching 1, and boost up their neighbors' probabilities (label ID 7, 17, etc.). However, in the case of *Hexinvon-8mg*, the probability scores are relatively low, with all RoIs being investigated achieving scores of only about 0.3.

Now, we utilize another explainable AI technique named GNNExplainer [33] to investigate further the reason for identifying the hard sample, *Hexinvon-8mg*. GNNExplainer is a model-agnostic architecture that can provide interpretable explanations for predictions of graph-based models. Specifically, GNNExplainer may identify a subgraph and a subset of node features that have a significant role in the prediction outcomes. In our experiment, we treat our Graph Transformer Network as a module that produces regression output, i.e., the context vectors corresponding to all RoIs. For a more comprehensible result, we set the number of RoIs selected from the RPN module to ten, consisting of the five RoIs with the greatest *objectness* scores and the other five with the lowest score. We utilize GNNExplainer to identify the
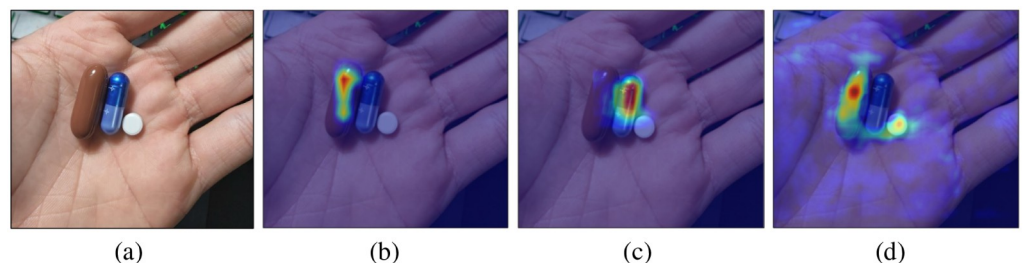

Fig 11. The saliency maps for each of the groundtruth labels included in the image instance. (a) Input. (b) LIVOLIN-FORTE. (c) Hapenxin. (d) Hexinvon-8mg. For simple samples (LIVOLIN-FORTE and Hapenxin), the classifier focuses on the exact location of the tablets to determine their identity. In contrast, for the hard case (Hexivon-8mg), information on both Hexivon and LIVOLIN-FORTE served as evidence.
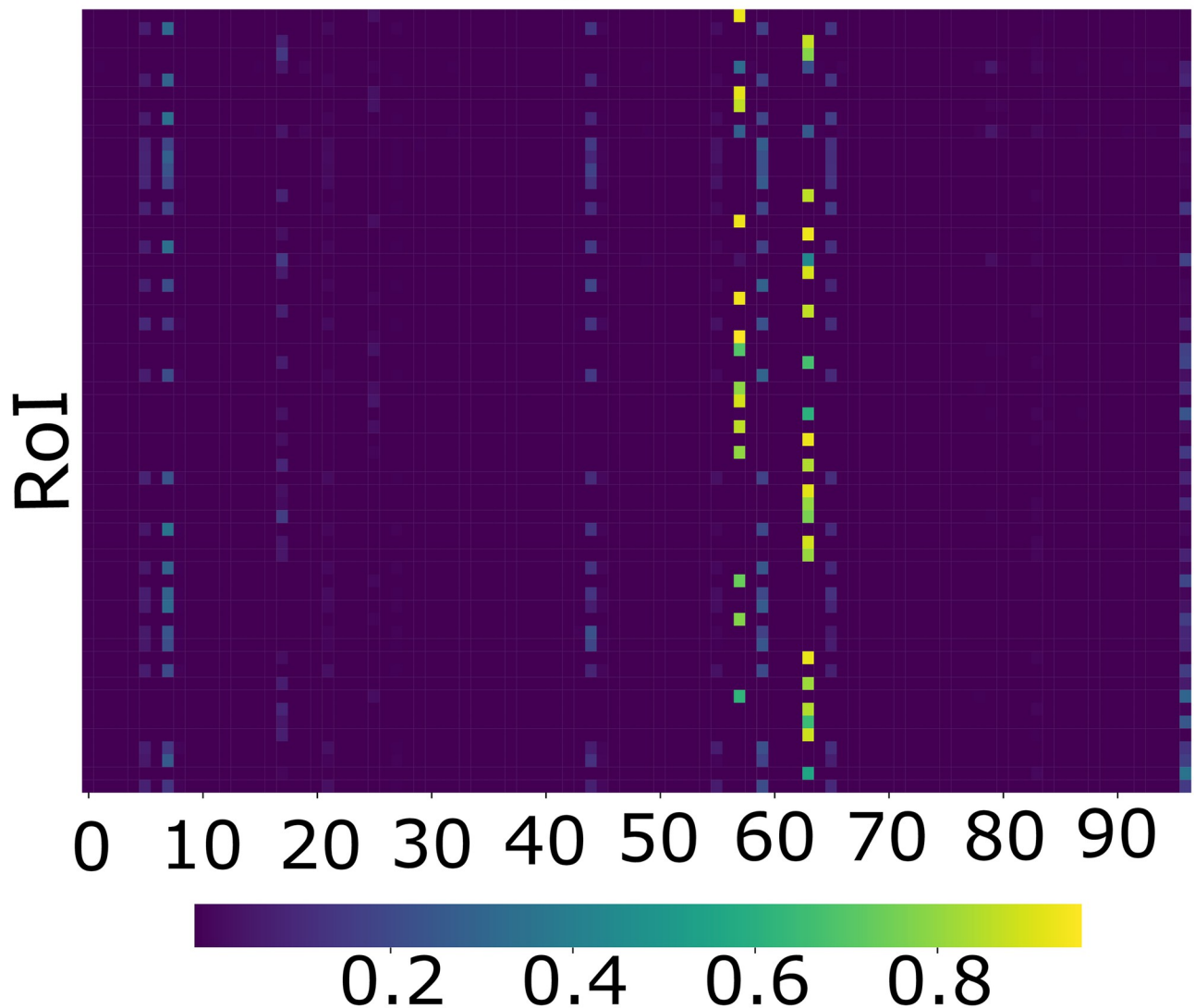
Predicting Probabilities

Fig 12. Probabilistic scores produced by PGPNet's Pseudo Classifier.

sub-graph that contributes the most in recognizing *Hexinvon-8mg*. The results are demonstrated in Fig 13. In this figure, the white box depicts the RoI of *Hexinvon-8mg*, the two orange boxes and blue boxes represent the RoIs of LIVOLIN-FORTE, and Hapenxin, respectively, while the five gray boxes indicate the RoIs of noise. The black edges represent the vital connections, whose weights are proportionate to the width of the edges. First, there are almost no edges between the nodes representing *Hexinvon-8mg* and those of the noise RoIs. It implies that the noise RoIs do not cue the prediction of *Hexinvon-8mg*. In contrast, there are bolded linkages between the RoIs of LIVOLIN-FORTE, Hapenxin, and *Hexinvon-8mg*. These findings, along with the saliency map (Fig 11), interpret that PGPNet has learned both the visual characteristic of the pill itself and the relationship between that pill and the others to make the final decision.
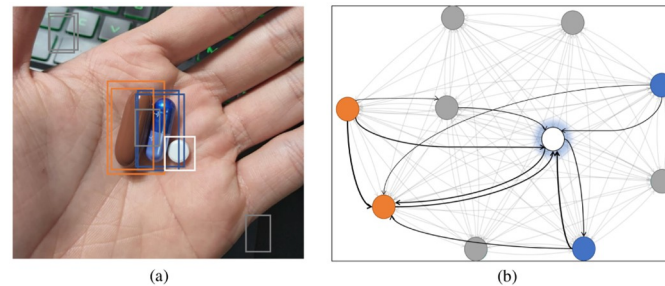
Fig 13. Interpretation of the prediction result for *Hexinvon-8mg* using GNNExplainer. (a) Bounding boxes of the RoIs indicate the RoIs in (b) sub-graph identified by GNNExplainer, which are most influential to the prediction of *Hexinvon-8mg*.

Ablation studies

In this section, we perform extensive ablation studies to investigate the impacts of the main techniques proposed in our PGPNet and to investigate how each component in the proposed method helps to improve learning performance. Specifically, we alter the Co-occurrence Graph and observe how it affects the detection results in Section Effect of co-occurrence graph's quality. We then assess the effects of using the relational graphs, the Graph transformer network, and the proposed auxiliary loss in Sections Effects of the relational graphs, Effects of the multi-modal data fusion block and auxiliary loss, respectively.

Effect of co-occurrence graph's quality. In this section, we perform two experiments to observe how the performance changes when the nodes set and edges set of MCG are modified, respectively. This can determine whether the noisy graph information can hurt the performance of our PGPNet.

Edge set modification. We first observe the behavior of our PGPNet when adding noise edges and removing actual edges. We set up four scenarios which are the combinations of removing 25% and 50% of the edges in the set E_1, and adding a number of synthesized edges corresponding to 25% and 50% of the cardinality of E_1.

Fig 14(a) illustrates the performance of PGPNet with all Medical Co-occurrence Graph variances when put into comparison with the original one. The performance here is denoted by the general metrics AP. As indicated by AP density, PGPNet with original MCG generates a
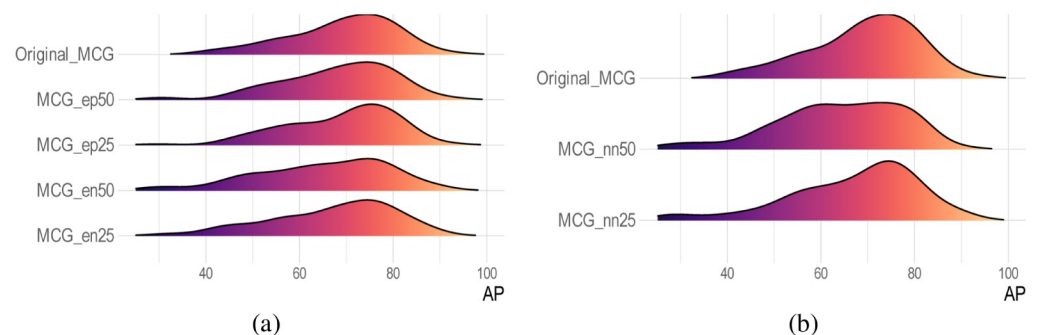

Fig 14. Empirical result of node set and edge set modification. (a) Distributions of Average Precision recorded over all classes produced by PGPNet with different MCG versions. (b) Distributions of Average Precision recorded over the classes in N_A set produced by PGPNet with different MCG versions.

more concentrated density with a smaller variance and a higher mean than other variances. In addition, when 50% of edges are eliminated, the performance is clearly inferior to when 25% of edges are eliminated. The figure concludes with the intriguing observation that eliminating edges at random would result in a greater performance decrease than adding noisy edges. This is because, even with the addition of noisy edges, PGPNet could still filter out unnecessary information through the training process. When excluding edges, the situation is different because the framework cannot learn the external knowledge contained in the eliminated edges.

Node set modification. To observe PGPNet's performance when the Medical Co-occurrence Graph lacks information on some specific nodes—classes, we design two different scenarios. In the first one, 25% of nodes are removed from the original graph; this set is denoted as N_A. For the latter, 50% of nodes are eliminated, and the corresponding set N_B is ensured to be a superset of N_A. The performances of PGPNet in two circumstances are compared with itself when having the full MCG, considering only the classes that appeared in the set N_A.

Fig 14(b) depicts the outcome of this experiment. The AP across all N_A classes is used to evaluate performance here. As indicated by the graph, node removals also result in a significant decrease in model performance. More interestingly, the more nodes eliminated, the greater drop is captured. Specifically, the AP density in case MCG contains only 50% of remaining nodes has a great variance, with the mean value only around 60%.

In the following, we study the effectiveness of the relational graphs, Graph Transformer Network (GTN) block, and auxiliary loss. The detailed configurations are presented in Table 6. The "+" sign indicates the presence of a component in a specific version, while the "−" denotes the opposite.

Effects of the relational graphs. In this section, we study the effectiveness of the Size-graph and visual-based graph. To this end, we implement two simplified versions of PGPNet, namely PGPNet-v2 and PGPNet-v3, in which we remove the Size-graph and visual-based graph, respectively. As shown in Table 6, eliminating the Size-graph causes a decrease in performance from 3.9% to 11.1%, while omitting the visual-based graph reduces the accuracy from 2.8% to 8.3%. An interesting finding is that the deterioration gap when removing the size graph is more significant than that when eliminating the visual-based graph in terms of all evaluation metrics. These findings imply the effectiveness of the Size-graph over the visual-based graph. Moreover, it can be observed that mAP is the most impacted when the relational graphs are removed, followed by AP50 when comparing mAP, AP50, and AP75. This can be explained as follows. In AP75, we measure the precision of RoIs with the IoU beyond 75%, which presumably has a high degree of confidence regarding the objective. In contrast, when

Table 6. Performance of PGPNet with the diferent combination of its components, i.e., when removing (marked as ×) / keeping (marked as ✓) the relational graph, GTN and auxiliary loss. Numbers inside the (.) represent the gap in percentage compared to the full version of PGPNet.

	Component					Performance					
	$\mathcal{G}_c$	$\mathcal{G}_s$	$\mathcal{G}_v$	GTN	$\mathcal{L}_{aux}$	mAP	AP50	AP75	APs	APm	APl
Faster R-CNN	×	×	×	×	×	63.7 (-8.6)	86.7 (-8.4)	76.9 (-7.9)	71.3 (-20.8)	58.1 (-10.6)	64.6 (-7.8)
PGPNet-v1	✓	×	×	×	×	65.9 (-5.5)	91.9 (-2.9)	79.6 (-4.6)	72.5 (-19.4)	62.3 (-4.3)	66.0 (-5.8)
PGPNet-v2	✓	×	✓	✓	✓	66.9 (-3.9)	92.1 (-2.7)	81.1 (-2.8)	80.0 (-11.1)	61.3 (-5.9)	67.6 (-3.6)
PGPNet-v3	✓	✓	×	✓	✓	67.8 (-2.8)	92.9 (-1.9)	82.3 (-1.5)	82.5 (-8.3)	62.7 (-3.5)	68.1 (-2.8)
PGPNet-v4	✓	✓	✓	×	✓	68.4 (-1.9)	92.6 (-2.2)	81.7 (-2.1)	80.0 (-11.1)	64.4 (-1.0)	68.8 (-1.8)
PGPNet-v5	✓	✓	✓	✓	×	67.2 (-3.6)	91.3 (-3.5)	80.9 (-3.1)	90.0 (+0.0)	62.2 (-4.3)	67.3 (-3.9)
PGPNet	✓	✓	✓	✓	✓	69.7	94.7	83.5	90.0	65.0	70.0

we reduce the IoU threshold, such as AP50 and mAP, the overlap area of the objective drops, resulting in a model with a significant degree of uncertainty. In this case, integrating relational graphs provides additional data that reduces uncertainty, thereby boosting detection accuracy.

Effects of the multi-modal data fusion block and auxiliary loss. To investigate the effectiveness of the GTN, we implement PGPNet-v4, omitting the GTN block and relying solely on the GCN to learn the node representation. Results in Table 6 reveal that GTN enhances the model's accuracy from 1.0% to 11.1%. Comparing mAP, AP50, and AP75, AP50 and AP75 are slightly more influenced by GTN than mAP, but the gaps are trivial. We employ PGPNet-v5, which eliminates the proposed auxiliary loss and compare its performance with the original PGPNet. As illustrated in Table 6, adopting our auxiliary loss may result in a 3 to 4 percent performance gain for most evaluation metrics. In the final ablation study, we implement PGPNet-v1, which retains only the co-occurrence graph and removes all the other components. As depicted in Table 6, the detection accuracy degrades significantly, with a gap ranging from 2.9% to 19.4%. However, even with this version, PGPNet is still superior to Faster RCNN, with a performance margin of up to 7.1%.

In conclusion, the PGPNet version with all components exhibits its superiority in all evaluation metrics. In addition, all versions of PGPNet are superior to the Faster R-CNN backbone, demonstrating the contribution of each component to the overall performance of PGPNet.

Conclusion

Contributions

We proposed PGPNet, a reliable and explainable pill detection framework in real-world settings. To deal with hard samples, PGPNet leveraged external knowledge, including co-occurrence likelihood, relative pill size, and visual semantic correlation during the training process. We implemented PGPNet into two popular object detectors and evaluated the proposed method on a real-world multiple pill detection dataset. The experimental results demonstrated that it could improve these models by considerable margins. Moreover, our comprehensive ablation studies proved the robustness, reliability, and explainability of the proposed framework.

Limitations and future works

While our proposed PGPNet framework demonstrates significant improvement in pill detection accuracy, we are aware of some potential failure cases as follows. Firstly, when pills are partially or completely obscured by other pills, the network may not be able to detect them (Fig 7 illustrates some of those cases in our VAIPE dataset). The co-occurrence graph may not provide sufficient information to detect the pills in these cases; leading to the drop of detection accuracy. Besides, PGPNet framework relies on the co-occurrence graph and other graph-based a priori information to enhance the precision of pill detection. If the graph construction process or a priori knowledge is inaccurate, the detection accuracy may be negatively affected. Notably, we conducted experiments on changing nodes and edges of a priori graphs to verify this argument (Section *Effect of co-occurrence graph's quality*). The results demonstrated that when a priori graphs do not include all pill classes, the detection accuracy decreases proportionally to the ratio of missing nodes. In addition, the edge ablation study revealed that we achieved the highest accuracy by removing 25% of the edges with the lowest weights. This phenomenon occurs because the least essential edges are potentially noisy interactions between the pills. In addition, we are aware that in practice, new drugs are frequently introduced; thus, pill detection solutions should be updated regularly to identify these new ones. In the PGPNet, every time a new pill class appears, we must reconstruct the graphs and retrain the model,

incurring a substantial computational cost. Thus, we will dedicate our future efforts to developing a continual learning mechanism that helps update the graphs and shorten the training time when dealing with the appearance of new pill classes.

Author Contributions

Conceptualization: Anh Duy Nguyen, Huy Hieu Pham, Phi Le Nguyen.

Formal analysis: Anh Duy Nguyen.

Investigation: Anh Duy Nguyen.

Methodology: Anh Duy Nguyen, Phi Le Nguyen.

Supervision: Huy Hieu Pham, Huynh Thanh Trung, Quoc Viet Hung Nguyen, Thao Nguyen Truong, Phi Le Nguyen.

Validation: Anh Duy Nguyen.

Visualization: Anh Duy Nguyen.

Writing – original draft: Anh Duy Nguyen.

Writing – review & editing: Huy Hieu Pham, Thao Nguyen Truong, Phi Le Nguyen.

References

1. World Patient Safety Day 2022; 2022. https://www.who.int/news-room/events/detail/2022/09/17/default-calendar/world-patient-safety-day-2022.

2. Yaniv Z, Faruque J, Howe S, Dunn K, Sharlip D, Bond A, et al. The national library of medicine pill image recognition challenge: An initial report. In: AIPR; 2016. p. 1–9.

3. Pill Identifier Drugs.com; 2022. https://www.drugs.com/imprints.php.

4. Wong YF, Ng HT, Leung KY, Chan KY, Chan SY, Loy CC. Development of fine-grained pill identification algorithm using deep convolutional network. Journal of biomedical informatics. 2017; 74:130–136. https://doi.org/10.1016/j.ijmedinf.2017.06.007 PMID: 28923366

5. Usuyama N, Delgado NL, Hall AK, Lundin J. ePillID dataset: a low-shot fine-grained benchmark for pill identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops; 2020. p. 910–911.

6. Ling S, Pastor A, Li J, Che Z, Wang J, Kim J, et al. Few-shot pill recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 9789–9798.

7. Kwon HJ, Kim HG, Lee SH. Pill detection model for medicine inspection based on deep learning. Chemosensors. 2021; 10(1):4. https://doi.org/10.3390/chemosensors10010004

8. Khan SD, Alarabi L, Basalamah S. A unified deep learning framework of multi-scale detectors for geospatial object detection in high-resolution satellite images. Arabian Journal for Science and Engineering. 2022; p. 1–16.

9. Khan SD, Basalamah S. Multi-scale person localization with multi-stage deep sequential framework. International Journal of Computational Intelligence Systems. 2021; 14(1):1217–1228. https://doi.org/10.2991/ijcis.d.210326.001

10. Chang WJ, Chen LB, Hsu CH, Chen JH, Yang TC, Lin CP. MedGlasses: A wearable smart-glasses-based drug pill recognition system using deep learning for visually impaired chronic patients. IEEE Access. 2020; 8:17013–17024. https://doi.org/10.1109/ACCESS.2020.2967400

11. Ting HW, Chung SL, Chen CF, Chiu HY, Hsieh YW. A drug identification model developed using deep learning technologies: experience of a medical center in Taiwan. BMC health services research. 2020; 20(1):1–9. https://doi.org/10.1186/s12913-020-05166-w PMID: 32293426

12. Proma TP, Hossan MZ, Amin MA. Medicine recognition from colors and text. In: Proceedings of the 3rd International Conference on Graphics and Signal Processing; 2019. p. 39–43.

13. Ou YY, Tsai AC, Wang JF, Lin J. Automatic drug pills detection based on convolution neural network. In: 2018 International Conference on Orange Technologies (ICOT). IEEE; 2018. p. 1–4.

14. C3PI Dataset; 2018. https://www.nlm.nih.gov/databases/download/pill_image.html.

15. Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems. 2015;28.

16. De Craen AJ, Roos PJ, De Vries AL, Kleijnen J. Effect of colour of drugs: systematic review of perceived effect of drugs and of their effectiveness. Bmj. 1996; 313(7072):1624–1626. https://doi.org/10.1136/bmj.313.7072.1624 PMID: 8991013

17. Yun S, Jeong M, Kim R, Kang J, Kim HJ. Graph transformer networks. Advances in neural information processing systems. 2019;32.

18. Xu H, Jiang C, Liang X, Lin L, Li Z. Reasoning-rcnn: Unifying adaptive global reasoning into large-scale object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2019. p. 6419–6428.

19. Gong C, He D, Tan X, Qin T, Wang L, Liu TY. Frage: Frequency-agnostic word representation. Advances in neural information processing systems. 2018;31.

20. Jocher G. YOLOv5 by Ultralytics; 2020. Available from: https://github.com/ultralytics/yolov5.

21. Xu H, Jiang C, Liang X, Li Z. Spatial-aware graph relation network for large-scale object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 9298–9307.

22. Tan L, Huangfu T, Wu L, Chen W. Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification. BMC medical informatics and decision making. 2021; 21:1–11. https://doi.org/10.1186/s12911-021-01691-8 PMID: 34809632

23. Zhang G, Luo Z, Cui K, Lu S. Meta-detr: Few-shot object detection via unified image-level meta-learning. arXiv preprint arXiv:210311731. 2021;2(6).

24. National Research Ethics Service (NRES) NHSDrRDrA Health Research Authority. BWorld Robot Control Software; 2013. http://www.hra.nhs.uk/documents/2013/09/defining-research.pdf.

25. General Data Protection Regulation (GDPR); 2022. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02016R0679-20160504.

26. COCO—Common Objects in Context; 2022. https://cocodataset.org/#detection-eval.

27. Liu Z, Hu H, Lin Y, Yao Z, Xie Z, Wei Y, et al. Swin transformer v2: Scaling up capacity and resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2022. p. 12009–12019.

28. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 2117–2125.

29. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer; 2014. p. 740–755.

30. Loshchilov I, Hutter F. Decoupled weight decay regularization. arXiv preprint arXiv:171105101. 2017;.

31. Valdenegro-Toro M. I find your lack of uncertainty in computer vision disturbing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 1263–1272.

32. Zhang J, Bargal SA, Lin Z, Brandt J, Shen X, Sclaroff S. Top-down neural attention by excitation backprop. International Journal of Computer Vision. 2018; 126(10):1084–1102. https://doi.org/10.1007/s11263-017-1059-x

33. Ying Z, Bourgeois D, You J, Zitnik M, Leskovec J. Gnnexplainer: Generating explanations for graph neural networks. Advances in neural information processing systems. 2019;32.