PuPill - A Model For Identifying Medications From Images Of Packaging

Lulu Beatson [1] **Sandra Steyaert** [1] **Theo Bourdais** [1] **Chethan Sarabu** [1] **Scott Cressman** [1] **Samia De Brouwer** [1] **Walter De Brouwer** [1] **Gabriel Zaccak** [1] **Meelis Lootus** [1]

Abstract

We introduce an algorithm, PuPill, to recognize drugs from images of drug packaging and return its identity as a National Drug Code and/or RxNorm Clinical Drug. The algorithm marries deep learning based optical character recognition (OCR) to extract textual features from the image with symbolic graph reasoning on the RxNav system, to identify the medication precisely. The algorithm is explainable and interpretable, allowing its predictions to be traced along the RxNav graph and back to regions of the original image. By addressing the task as an information retrieval problem, our algorithm can be fine tuned but does not require retraining when the medications database is updated. We tested the algorithm with 10.8k images, covering 10.6k National Drug Codes belonging to 4.7k RxNorm Clinical Drugs and achieved an accuracy of 91.0%. Moreover, we packaged and deployed PuPill as an API with Android and iOS SDKs capable of collecting user feedback to further improve the algorithm. The ease of data input and precision of the output in standard clinical code systems means that a wide range of downstream tasks (such as switching medication brand or retrieving further adjacent information on the drug) can be made available to a large user base.

1. Introduction

Machine learning systems in the clinic have to connect with structured knowledge bases and also interpret sensory unstructured information from a given patient. Historically, there are two competing themes in Artificial Intelligence that can be mapped to these two kinds of info: symbolic vs. connectionist (Smolensky, 1987). Connectionist approaches have evolved into the "deep learning" theme, accelerated by back-propagation and by their performance on well-labelled datasets (Thompson et al., 2020). At present, trying to achieve the next stage of deep learning is a hot research topic (Goel, 2022).

In a practical setting, clinical systems have to be robust, explainable, updatable, interpretable. Deep learning does not always lend itself well to this: model updates require retraining on an updated dataset; the internal representations of a neural network are difficult to interpret as practical concepts.

The task that we address in this paper is the following: given an image of a package of a prescription or over the counter medication, identify that medication precisely as one of existing FDA-approved medications. This task constitutes a well-fitting instance where a connectionist or deep learning approach would introduce difficulties: (i) every time a new drug is added, the network would need to be retrained which is unstable and costly; (ii) sufficient training data would be required to recognise the package, (iii) it would be difficult to interpret the decisions of the network.

In order to address the task, we combine the best elements of deep learning and symbolic logic (graph reasoning) worlds modularly: we use deep learning models, pre-trained on large sets of images, to perform the task of detecting text from images. We then use a custom, explainable and interpretable graph reasoning approach to find the best match between the detected text and the set medications listed in the NDC Directory and organized by the RxNav Graph.

With updates to the RxNav executable as simple, robust and auditable database updates, and the algorithmic steps fully visible, this algorithm presents an example from medicine which is both practical and opens up many interesting use cases: once the precise medication is identified, adjacent information can be retrieved such as: (-) perhaps cheaper or more effective alternatives to that drug, (-) side effects of the drug, (-) alternative treatments. Our algorithm has the potential to remove the need for specialist knowledge to identify the right codes on the package of the drug and then identify the drug in the context of all available drugs and the information about them. This then opens up a world of new information: the relationships between medications and disease (Yaddanapudi, 2019) and connection to further infor-

[1]Sharecare Inc., Atlanta, GA, USA. Correspondence to: Meelis Lootus <meelis.lootus@sharecare.com>.

Workshop on Interpretable ML in Healthcare at International Conference on Machine Learning (ICML).

mation on the effect and mechanisms of the drugs (Pathak et al., 2011).

1.1. Related Work

Past work in medication recognition from images can be described by the subject of their image, features (extracted by what we refer to as *Pathways*), and framing of the problem (*Frameworks*).

Some subjects of images include: pills (Lee et al., 2012; Larios et al., 2019; W. Chang et al., 2019; Lester et al., 2021; Roy, 2022), blisters (Wang et al., 2018; Ting et al., 2020; Tran et al., 2022), prescription labels (Sarzynski et al., 2017; Givatar Inc, 2020) and packaging of over the counter medications (Roopa et al., 2016; L. Magalhães et al., 2017; Lee et al., 2018; Ciampi et al., 2022).

We refer to ways of getting from input images to outputs collectively as *pathways*. We classify pathways according to which type of features are considered: *image pathways* (Section 1.1.1), *text pathways* (Section 1.1.2), *barcode pathways* (Section 1.1.3), and *multi-pathway* methods (Section 1.1.4).

The task can be framed as: *Classification* where the set of possible predictions is fixed and samples of every class are seen during training; *Similarity Analysis* where prediction is made based on the images similarity to samples in a reference database which may be expanded with images belonging to new classes and where similarity may be measured with any type of pathway; *Information Retrieval* where there is a database of medications from which predictions are drawn and not every medication needs to be represented in the training data. We discuss the database requirement of the former two frameworks in Section 1.1.6.

1.1.1. Image Pathways

(Lee et al., 2018; Naeem & Coronato, 2022; Tran et al., 2022) use an image classification framework, basing their models on MobileNetv2 (Sandler et al., 2018), ResNetv2 (He et al., 2016) and VGG16 (Simonyan & Zisserman, 2014) pre-trained on large image sets. (Ting et al., 2020) uses an object detection framework with YOLOv2 (Redmon & Farhadi, 2017). Other works use an image similarity analysis framework with (Liu et al., 2020) using average perceptual hash algorithm (Zauner, 2010) and (X. C. Benjamim et al., 2012) extracting image features using SURF (Bay et al., 2006). One could also train an image similarity metric (Wang et al., 2014) for medication package images but this requires a large number of labeled training images which are not readily available.

1.1.2. Text Pathways

(Liu et al., 2020) use a connectionist text proposal network (Tian et al., 2016) for scene text detection then applies Tesseract (Smith, 2007) for text recognition. They then perform text similarity analysis against a historical reference set using a combination of (i) partial and generalized Levenshtein distance and; (ii) universal sentence encoding and cosine distance. (Gomez et al., 2015) also used Tesseract to extract text but use Levenshtein distance to a dictionary in order to extract medication brand names and ingredients.

After text detection and recognition, (Negi et al., 2021) take the largest text to be the medication name and uses pattern matching to extract other fields such as manufacturing and expiry dates.

1.1.3. Barcode Pathways

The Global Trade Item Number (GTIN) is the global standard for identifying products including medications. GTIN may be printed as text and or AIDC technology (Automatic Identification and Data Capture e.g barcodes). Medications sold within the US embed the NDC (Section 2.2.1) within a 12 to 14 digit GTIN (GS1 US, 2022). Several related works use barcode reading (Gomez et al., 2015; L. Magalhães et al., 2017; Sarzynski et al., 2017; Ciampi et al., 2022; Naeem & Coronato, 2022). (Gomez et al., 2015; Naeem & Coronato, 2022) use the open-source barcode reading library ZBar (ZBar, 2022).

1.1.4. Multi-Pathway Methods

Assembling multiple pathways can improve performance over individual pathways (Liu et al., 2020). (L. Magalhães et al., 2017) try up to three pathways, starting with the barcode pathway, stopping when one returns a result. (Gomez et al., 2015; Naeem & Coronato, 2022) automatically select one of their pathways for the image.

1.1.5. Pathway Type Comparison

Image pathways require the most training data, because their predictions are sensitive to the environment of the packages, and are limited in the number of classes. (Tran et al., 2022) validated their approach on realistic mobile images but required a large number of training images and could only predict 5 classes of active ingredient with unspecified strength. (Ting et al., 2020) could predict 250 classes specifying the ingredients and strengths of blister packages with 8 times fewer training images. However all of their images were taken with controlled lighting and background.

Pathways using a similarity framework (Liu et al., 2020; Tian et al., 2016) may, in addition to the reference dataset, require a large number of text or image pairs to train the similarity metric or embedding. Similarity frameworks and text extraction pathways are the easiest to update with new medications. Image classifiers on the other hand require

images for all the classes they aim to predict.

1.1.6. Medications Database Integration

Image pathways (Ting et al., 2020; Naeem & Coronato, 2022; Tran et al., 2022) arriving at the final medication(s) using a classifier do not require the existence nor maintenance of a medications database. Methods which extract the medication name from text (Gomez et al., 2015; Negi et al., 2021) also do not require a medications database unless verifying against possible medications like (Sarzynski et al., 2017). (Sarzynski et al., 2017; Givatar Inc, 2020) extract values that cannot be verified using a medications database: prescription instructions, pharmacy and physician details.

Text and/or image similarity analysis methods (X. C. Benjamim et al., 2012; Liu et al., 2020) require a reference database and can be continuously improved with new images (Liu et al., 2020).

Finally, there are methods which retrieve entries of a medications database. Such methods do not need to have seen an image of every medication in its database and the database can be readily updated. All of these methods involve a text pathway. The database may only provide a list of possible medication names which the method tries to identify within the text (Gomez et al., 2015; Naeem & Coronato, 2022) or may contain other columns so that the method can retrieve the closest rows whilst considering multiple medication attributes extracted from the text (Sarzynski et al., 2017).

Our method uses this last framework with Google Vision OCR (Google, 2022b) and Google Healthcare NLP (Google, 2022a) performing image and text processing and our own NDC Recognition, Multi-word Matching algorithms for extracting medication attributes and retrieving the best matches from a medications database. The use of an information retrieval framework allows PuPill to stay current with the latest set of medications in the RxNav database through database updates; without requiring retraining.

2. Method

2.1. PuPill Pipeline Overview

PuPill takes as input an image of a drug package and gives as output the identity of the drug in that image as an NDC code and/or RxCui code, based on text detected in the image. The pipeline, shown in Figure 1, begins with (1) detecting and spell-correcting all text found in the input image; and is then divided into two Paths: (A) NDC-Path (Section 2.4); (B) NLP-Path (Section 2.5). The NDC-Path is run first and tries to extract the NDC of the drug from the text. The NLP-Path is run only if the NDC-Path fails to identify the drug. This secondary path consists of three steps 2B-4B: (2B) recognising drug-relevant words in the text, (3B) retrieving candidate medications from a database which match the drug-relevant words, (4B) scoring and ranking the candidates.

2.2. Drug Hierarchies

This section introduces the drug ontologies used in the PuPill algorithm: the National Drug Code, RxNorm, RxCui, RxNav and medications database.

2.2.1. National Drug Code

National Drug Codes (NDC) are numeric codes that identify a drug's manufacturer, drug composition, package size and package type. NDC-s are assigned by the FDA to all current human drugs for sale in the US. NDC10 or NDC11 (NDC-s of length 10 or 11) can be printed on packaging in a three part – delimited format where each part has length [4]-[4]-[2], [5]-[3]-[2], [5]-[4]-[1] or [5]-[4]-[2] (Appendix C Figure 3 for illustration). Often, the code is also preceded by the string `NDC`; and any part of an NDC can start with zeroes.

2.2.2. RxNorm and RxCui

RxNorm is a standardized nomenclature for US clinical drugs and their components. RxCui-s are numeric codes accompanying the normalized names in RxNorm (National Library of Medicine, 2022b). RxCui codes are divided into 20 *Term Types* (TTY) depending on the entity they identify; at different levels of abstraction. The drug component types we will use are *Ingredient* (IN), *Precise Ingredient* (PIN), *Brand Name* (BN) and *Semantic Clinical Drug Component* (SCDC). The drug types we will use are *Semantic Clinical Drug* (SCD), *Semantic Branded Drug* (SBD), *Generic Pack* (GPCK) and *Branded Pack*(BPCK). Unlike NDC, the RxNorm system does not include manufacturer or packaging information so neither aspect impact the value of any RxCui associated with a medication.

We will also make use of the RxNorm names of *Dose Forms* (DF) and their organization into *Dose Form Groups* without using their corresponding RxCui codes.

2.2.3. RxNav and Relationships

RxNav is the RxNorm browser. Each RxCui can be related to none or many other RxCui belonging to different TTY. The relationships between different TTY can be visualized as the *RxNav Graph*. Figure 1 (c) shows a portion of the RxNav Graph used by the NLP-Path (Section 2.5). If there is no direct relationship between two TTY then multiple can be composed together according to *Default Paths* defined by RxNav.

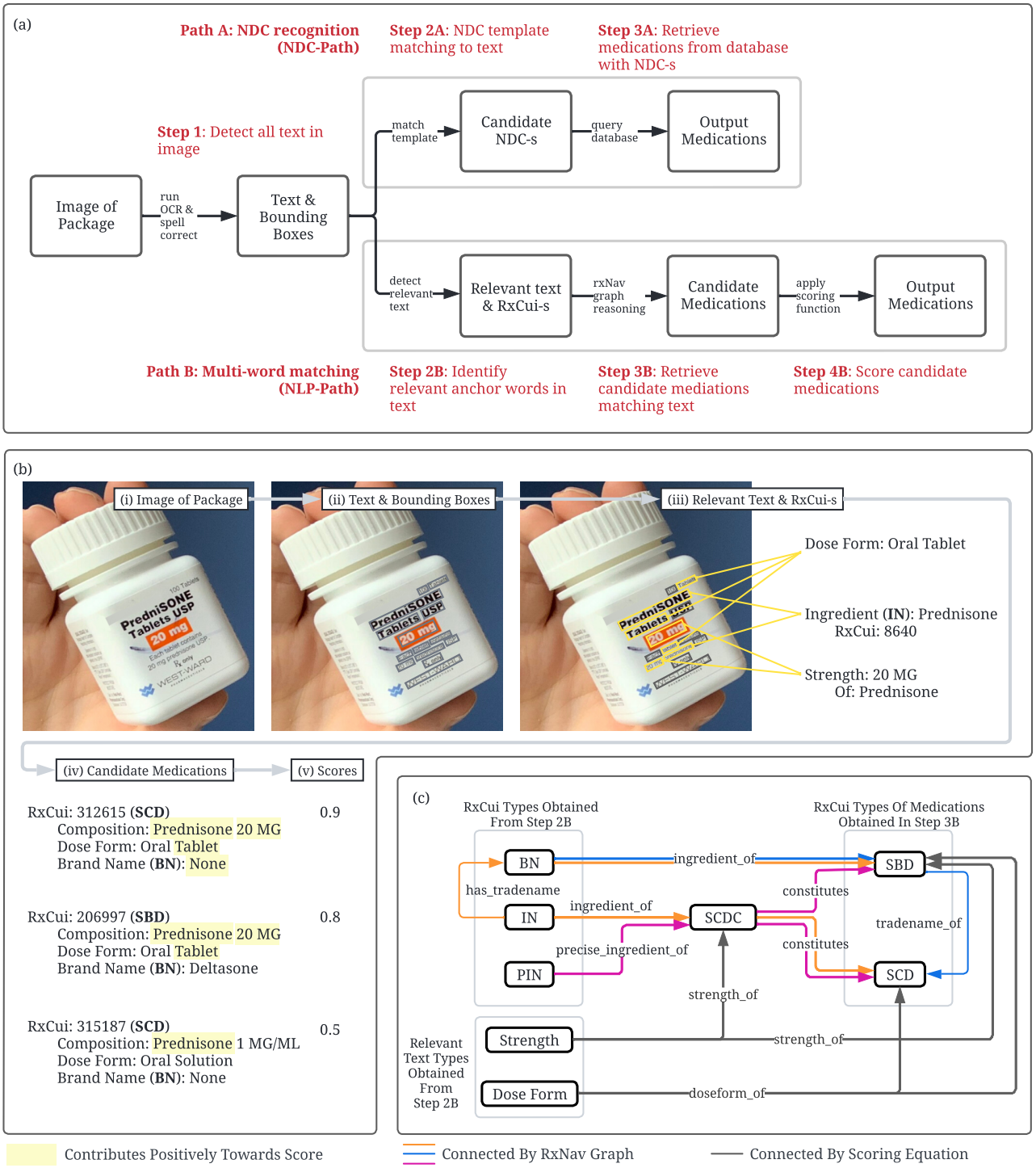

Figure 1. Overview of the algorithm. **Panel (a)** shows the PuPill pipeline with two Paths: (A) NDC-Path 2.4 and (B) NLP-Path 2.5. The NDC-Path solves the drug recognition problem by detecting NDC codes in the package. The NLP-Path solves the problem by considering all the text detected in the package and matching relevant words to a database of drug concepts. **Panel (b)** illustrates the NLP-Path with an example, showing the in/output from each step on the path: (i) input image, (ii) text detected in the image, (iii) text filtered for drug relevant keywords, (iv) retrieved candidate medications matching drug relevant key words (v) candidate medications scored and ranked in decreasing order. **Panel (c)** shows graphical paths for reasoning using RxNav and Scoring Equation (1).

2.2.4. The Medications Database

PuPill uses a medications database, which as of October 2021, contains 220,205 unique NDC11-s belonging to 12,197 drug concepts identified by RxCui codes. These RxCui codes belong to four TTY classes: SCD (48.8%), SBD (47.8%), GPCK (0.006%), BPCK (0.03%). For each medication, the database stores its name, brand name and dose form.

2.3. Step 1: Detect All Text In Image

Step 1 takes as input an image of a drug package and gives as output spell-corrected text found in that image. The algorithm uses Google Vision OCR to extract text from the image as a list of words with bounding boxes. We remove all symbols that are not alphanumeric or `/ , . ; ( ) -`. Then the words are passed, one-by-one, into a spell correction model (Search.io, 2021). If an input word is not in our dictionary, the spell correction algorithm replaces it with the best match from a dictionary. A dictionary word is considered a better match if it has lower Levenshtein distance to the input word and is more common. The dictionary containing 25,782 words and their frequencies is created by counting all the words in all the medication names in consideration.

2.4. Path A: NDC Recognition (NDC-Path)

2.4.1. Step 2A: NDC Template Matching

Step 2A takes as input the text returned by Step 1 and applies template matching to find all possible NDC-s in the words. The allowed formats for NDC-s are given in Section 2.2.1 but to allow for small errors in Step 1, the templates used here allow the delimiter character to be -, a space or a combination.

2.4.2. Step 3A: Retrieve Medications From Database With NDC-s

Step 3A takes as input the candidate NDC-s found in Step 2A, converts them into NDC11 format and retrieves the medications identified by these NDC11s. Figure 3 shows an example of each NDC10 format being transcribed and converted to NDC11 by insertion of `0` in from of the shortened segment. If any medications are identified by candidate NDC-s, the algorithm terminates and returns those medications. Otherwise, the NLP-Path is triggered (Section 2.5).

2.5. Path B: Multi-word Matching (NLP-Path)

2.5.1. Step 2B: Identify Relevant Anchor Words In Text

Step 2B takes the text returned by Step 1 and uses Google Healthcare NLP to extract a set of ingredients, brand names and the IN/PIN, BN type RxCui-s which identify them. For the ingredients, it may also detect and associate strengths to them.

2.5.2. Step 3B: Retrieve Candidate Medications Matching Text

Step 3B takes as input the RxCui-s identifying ingredients and brand names from Step 2B and outputs RxCui-s of types SCD and SBD identifying medications which either (i) contain the input ingredients or (ii) are sold as the input brand names or (iii) are generic with the same composition as one with an input brand name. For each input RxCui, we obtain the output RxCui-s satisfying (i), (ii), (iii) using the RxNav Graph. Figure 1 (c) illustrates the paths of reasoning to obtain the output RxCui-s. We refer to the medications identified by output RxCui-s of this step as *candidates*.

2.5.3. Step 4B: Score Candidate Medications

Step 4B assigns a numeric score in $[0, 1]$ to each candidate medication, based on textual features. The Scoring Equation (1) was designed to indicate how likely a drug concept identified by a given SCD/SBD RxCui is to be the drug in a drug package image (given the image text and IN/PIN, BN RxCui extracted from it). The score S_i of the i^{th} candidate is given by:

$$\begin{aligned} S_i = k_I I_i + k_C C_i + k_{St} St_i + k_1[G_i(1 - \text{Any}_j(B_j))] \\ + k_2[(1-P)B_i + PG_i] + k_F F_i + k_H H_i \end{aligned} \quad (1)$$

where the coefficients $k_I, k_C, k_{St}, k_1, k_2, k_F, k_H$ sum to one, are initialized uniformly. Each term takes values in $[0, 1]$, represents a medication attribute and is defined as:

Ingredient I_i = the Jaccard index between the set, $\widehat{\text{Ing}}$, of IN/PIN RxCui-s from Step 2B and the set, Ing_i, of IN/PIN RxCui-s identifying ingredients contained the i^{th} candidate.

$$I_i = \text{Jaccard}(\widehat{\text{Ing}}, \text{Ing}_i) = \frac{|\widehat{\text{Ing}} \cap \text{Ing}_i|}{|\widehat{\text{Ing}} \cup \text{Ing}_i|} \quad (2)$$

Composition C_i = proportion of the ingredients identified in the intersection of sets of IN/PIN RxCui-s from Step 2B and set of IN/PIN RxCui-s contained in the i^{th} candidate which have the same strength in the i^{th} candidate as the strength linked to ingredients in Step 2B[1].

$$C_i = \frac{|\text{Ingredients in } \widehat{\text{Ing}} \cap \text{Ing}_i \text{ with correct strength}|}{|\widehat{\text{Ing}} \cap \text{Ing}_i|} \quad (3)$$

Strength St_i = proportion of candidate strengths present in the text[1].

[1] See appendix A for how varying representations of strength are compared.

Dose Form F_i = mean of the following 4 components: f_i = proportion of words in the i^{th} candidate's dose form which are in the text; f_i' = 1 if any of the i^{th} candidate's dose form synonyms or keywords are in the text, 0 otherwise; g_i = 1 if the name of any dose form group that the i^{th} candidate's dose form belongs to can be found in the text; g_i' = 1 if any synonym or keyword of any dose form group that the i^{th} candidate's dose form belongs to can be found in the text.

Brand B_i = 1 if the i^{th} candidate was nominated in Step 3B by a BN RxCui from Step 2B or the i^{th} candidate is branded and its brand name can be found in the text, 0 otherwise.

Generic G_i = 1 if the i^{th} candidate is generic, 0 otherwise.

Generic Phrase Parameter P = 1 if the text contains a phrase which suggests the package is a generic product, 0 otherwise.

Text Height H_i = maximum height of drug-relevant text found to match the attributes of the i^{th} candidate. The height of each word is computed from its bounding box obtained in Step 1 and min-max normalized across all the package words.

2.6. Deployment

The PuPill API was deployed on Google Cloud where inference is performed by sending the image within the payload of a request and receiving the predictions in the response body. We also created mobile SDK-s and Apps for both Android and iOS to demonstrate PuPill.

3. Experimental

3.1. Datasets

We used a large dataset of medication labels from Daily-med and constructed three other small datasets from images available on the internet. All images in the Daily-med dataset are identified by both NDC and RxCui codes. For images in the small datasets, if they depict an NDC as text then we manually transcribe the NDC and lookup the RxCui, otherwise, we find the RxCui by searching for the medication by name in RxNav using its online tool. For each image in the three small datasets which has a visible NDC code, we also include a version of the image with the NDC masked. Table 1 shows the number of images and Figure 2 (a) shows samples from each dataset.

Daily-med dataset. The National Library of Medicine makes drug labels, which are submitted to the FDA and currently in use, available as the Daily-med database. The labels are flattened, similar to how they would be printed. This does not provide realistic lighting or backgrounds that PuPill will encounter in mobile images but does allow us to test on a wide variety of medications with accurate NDC or RxNorm identities.

Table 1. Number of raw images with and without NDC and the total number of image obtained after masking NDC.

DATASET	RAW IMAGES VISIBLE NDC (Y)	RAW IMAGES WITHOUT NDC (N)	TOTAL IMAGES (2Y + N)
POPULAR BRANDED	50	0	100
POPULAR GENERIC	27	10	64
REALISTIC	18	56	92
TOTAL	95	66	256

We downloaded 10.6k images in the Daily-med SPL format with annotations identifying 10,511 unique NDC-s, belonging to 4,641 unique drug concepts identified by RxCui-s

Popular Branded Dataset. This dataset consists of images of the 50 most prescribed drugs in the United States (DrugReport, 2020).

Popular Generic Dataset. This dataset contains images of generic medications which have the same ingredients as those in the popular branded dataset but may vary slightly in strength or dose form where an equivalent cannot be found.

Realistic Dataset. The purpose of this dataset was to provide images with more realistic environments styles to what we expect PuPill to encounter in mobile images.

3.2. Evaluation Measures

3.2.1. TOP-k RXCUI ACCURACY

For each image, PuPill may return multiple candidates ranked in descending score. A candidate is correct if its RxCui agrees with the ground truth of the image. Then define for $k = 1, 2, 3, \ldots$

$$\text{Top-}k \text{ RxCui Accuracy} = \frac{|\{\text{Images with correct RxCui at rank} \leq k\}|}{|\text{Images}|} \quad (4)$$

We report Top-5 accuracy because five is likely to be the maximum number of candidates that could be displayed in a mobile app and be friendly to the user. We also report "All" accuracy which includes candidates of any rank.

3.2.2. MEASURES FOR MEDICATION ATTRIBUTES

An incorrect output from the NLP-Path is likely to share some similarities with the ground truth. We break down the medication concept identified by an RxCui into its attributes in order to measure this similarity. This also gives us an

indication of which fields a user may need to correct during feedback.

The base attributes of a medication concept we will consider are: *BrandName*[2], *DoseForm*, a set of *Ingredients* and a *Strength* for each ingredient. We will also combine *Ingredients* and *Strength* into the *Composition* attribute and *Composition* and *DoseForm* into the *Name* attribute. For these attributes, we then measure similarity between the candidate and ground truth in a similar way to scoring the candidate against the textual information extracted from the image in Section 2.5.3.

For *BrandName*, *DoseForm*, *Name* we check if the candidate's value is the same as ground truth and define Top-k attribute measures of each, similar to Top-k RxCui accuracy. For *Composition* and *Ingredients*, we compute the Jaccard index between the candidate's set and the ground truth's set and define Top-k measure to be the max Jaccard index within the first k candidates, averaged across the dataset.

3.3. Testing Hardware

During testing, inference requests were sent to the PuPill API by a MacBook Pro (2019, 2.3 GHz 8-Core Intel i9, 64GB). We recorded the total latency from the test device using a Python script and the latency of processes in the algorithm using Datadog[3].

4. Results

4.1. Accuracy

4.1.1. Overall performance

Across the images in all the datasets, the overall Top-1, Top-5 and "All" accuracies were 91.0%, 94.5% and 98.4%. Of all the images, only 0.46% still did not return any medications after both pathways. Performance figures on each dataset subsets are shown in Table 2.

4.1.2. Path A: NDC Recognition Performance

Of the images with visible NDC (clear enough to be transcribed by human annotators), 94.0%, 85.2%, 55.6% returned predictions using the NDC-Path from the popular branded, generic and realistic datasets respectively. The Top-1 success rate of these small dataset images using the NDC-Path was 98.8% (97.9%, 100%, 100%).

We did not have annotations beforehand of which Daily-med images depicted an NDC as text; however, we found that 88.5% of them returned predictions using the NDC-Path with a high Top-1 RxCui accuracy of 99.6%.

[2]We think of generic medications as having an empty string for *BrandName*.

[3]Datadog is an observability platform for cloud applications.

Table 2. All, Top-5, Top-1 RxCui accuracy of PuPill on datasets with breakdown by subsets of images with visible/without NDC in small datasets and breakdown by path used in the Daily-med dataset.

Subset	ALL	TOP-5	TOP-1
Popular Branded	94.0%	93.0%	91.0%
Visible NDC	96.0%	96.0%	96.0%
Without NDC	92.0%	90.0%	86.0%
Popular Generic	96.8%	79.4%	63.5%
Visible NDC	100.0%	96.3%	92.6%
Without NDC	94.4%	66.7%	41.7%
Realistic	96.7%	82.6%	70.7%
Visible NDC	100.0%	100.0%	94.4%
Without NDC	95.9%	78.4%	64.5%
Daily-med	98.4%	94.6%	91.1%
NDC-Path	99.6%	99.6%	99.6%
NLP-Path	89.0%	55.4%	25.7%

4.1.3. Path B: Multi-word Matching Performance

Daily-med had the lowest NLP-Path Top-1 RxCui accuracy: 25.7% compared to 86.0%, 41.7%, 64.5% for the Popular Branded, Generic and Realistic datasets.

The NLP-Path's Top-1 composition, dose form and brand name accuracy are 54.6%, 62.9% and 57.8% respectively (Table 3). We can expect dose form to be an easier attribute to get right because there are only 112 possible values (National Library of Medicine, 2022a).

Images of branded drugs from all datasets using the NLP-Path get a Top-1 RxCui accuracy of 57.7% compared to 23.1% for generic drugs. Looking at the attributes, 74.2% of branded drugs get the correct brand name and on average 85.0% similarity in ingredients with the first candidate, compared to 52.8% and 68.6% among images of generic drugs. In other words, for 47.2% of images of generic drugs, the first candidate is branded.

Table 3. All, Top-5, Top-1 attribute measures (defined in Section 3.2) of the NLP-Path

Criteria	ALL	TOP-5	TOP-1
RxCui	89.6%	58.2%	30.4%
Name	91.1%	70.1%	45.6%
Brand Name	93.8%	79.2%	57.8%
Dose Form	94.0%	83.4%	62.9%
Ingredients	92.9%	83.8%	71.8%
Composition	92.4%	76.3%	54.6%

4.2. Inference time

Across the images in all the datasets, mean total time for inference and persisting of the inference and image was 732 ms. For the 87.3% of images returning a medication from the NDC-Path, inference time was only 335 ms. For the remaining 12.7% which also ran the NLP-Path, their inference time was 1,710 ms. So the NDC-Path is both faster and more accurate than the secondary NLP-Path.

Outside of candidate scoring in Step 4B, the most costly processes are Google OCR in Step 1 taking 334 ms and Google NLP in Step 2B taking 549 ms. A detailed breakdown of latency can be seen in Appendix D.

The mean total latency experienced by the test device (Section 3.3) was 1.4 seconds. This time is measured from sending the request to the API to receiving a response.

5. Discussion & Conclusion

5.1. Results

The majority of images in the Daily-med dataset used the NDC-Path which we saw was fast, precise and accurate. We can take advantage of this by advising users to include the NDC in their images if they can find it.

Images from the Daily-med dataset using the NLP-Path returned 3 to 5 times more candidates than images from the small datasets. This is because the flattened labels in Daily-med include ingredient listings which are not visible from the front of an assembled package typically depicted in the small datasets (Figure 2). For each ingredient detected, Step 3B will generate candidates containing that ingredient as one of its main active ingredients. However, these ingredients are not necessarily the main active ingredients of the true medication. This may explain why NLP-Path performance on the Daily-med dataset was lower compared to the small datasets. This could be fixed by expanding the medications database to include the non-active ingredients of each medication but we expect PuPill to encounter images more similar to the small datasets where this is not a problem.

PuPill NLP-Path performs better on images of branded medications than generic. Brand name is usually prominent on the packaging and a brand name identifies a set of ingredients. This means, that in images of branded medications, the brand name is likely to be detected and produce candidates with the correct ingredients.

5.2. Performance Compared to Related Works

(Liu et al., 2020) is similar to PuPill in that it is able to predict the Daily-med SetId[4] of its images which can be mapped to RxCui of the same level of the drug hierarchy as our medications database. They also have a relatively large number of possible classes: 669 SetId-s (196 of which are opioid) and are able to learn new classes by building on their reference dataset using feedback. Testing on 300 images, they achieve a Top-1 accuracy of 80.0% (88.0% on opioid SetId-s). We are able to improve upon this with an expandable database of 12,197 possible RxCui and a Top-1 accuracy of 91.1% on a larger set of Daily-med images.

(Tran et al., 2022) used more realistic mobile photos of ophthalmic bottles with varied background and lighting. They achieve a test accuracy of 86% compared to the Top-1 of 71% achieved by PuPill on our Realistic dataset. However, their experiment only used 5 classes of ingredients which is fewer and less precise than SCD, SBD type RxCui-s returned by PuPill. (Ting et al., 2020) used the same lighting and background in all their photos of blister packages. Their model had 250 classes on a similar level of the drug hierarchy as PuPill's output (but again far less comprehensive) and had an accuracy over 90% (F1 up to 95.99%).

Table 5 compares PuPill to several related works.

5.3. Interpretability & Explainability

We presented a method for drug recognition from images, that is explainable, interpretable, maintainable, and created based on clinical reasoning. It modularly combines deep learning and symbolic graph reasoning approaches. The image to text part of the pipeline is solved using deep learning approaches and the text analysis involves graph reasoning algorithms and fitting to the RxNav knowledge base. A connectionist deep learning approach could have been used end to end that would take as input a drug package and output the result - however such an approach would require the training set of package images to exhaustively or close to it represent all possible packages expected in production. Such a model would need to be retrained every time a new drug ended on the market - which would be resource intensive, require deep learning expertise in practice, and be not explainable or interpretable. In contrast, our model has a simple update process (with new drugs added through database updates) and is fully explainable and interpretable.

Our method constitutes a way to modularly add and edit domain knowledge that works in a practical setting where accuracy and rapid updates and fixes are important, and opens the question: what are some other ways to combine the symbolic standard interoperable (and also useful) knowledge graphs such as ICD-10, CPT with deep learning approaches, to create solutions that are updatable and explainable; take as input unstructured data from a given patient and connect it to the context knowledge embedded in the graphs.

[4]SetId is used by dailymed to associate multiple revisions of a drug label.

5.4. Mobile Deployment

With most smartphones capable of OCR, we could send only the OCR results to the PuPill API for inference. Not only could this reduce the latency by reducing the payload size but it could also give the user an understanding of the algorithm and an immediate indication of the image quality if OCR data is overlaid. This would build trust and improve accuracy.

5.5. Conclusion

We have introduced an algorithm for recognition of drugs in drug package images, and validated it extensively. PuPill outperforms other methods identifying drugs at the same level of the drug hierarchy in terms of the number of medications it can recognise and the accuracy with which it does so. Also, by demonstrating the explainability and interpretabiliy of our approach, we hope this work encourages more solutions in healthcare to combine symbolic graph reasoning with deep learning.

Acknowledgements

The authors would like to thank the following team members for their expertise and development effort through the project: Ian Timmis, Geert Trooskens, Alberto Villacorta, Yuri Subach, Avneesh Mehta (engineering), Emmanuel Coloma, Vijay Sivaji (design).

References

Bay, H., Tuytelaars, T., and Gool, L. V. Surf: Speeded up robust features. In *European conference on computer vision*, pp. 404–417. Springer, 2006.

Ciampi, M., Coronato, A., Naeem, M., and Silvestri, S. An intelligent environment for preventing medication errors in home treatment. *Expert Systems with Applications*, 193:116434, May 2022. ISSN 0957-4174. doi: 10.1016/j.eswa.2021.116434. URL https://www.sciencedirect.com/science/article/pii/S0957417421017218.

DrugReport. 50 commonly prescribed drugs in america, 2020. URL https://www.drugreport.com/50-commonly-prescribed-drugs-in-america/.

Givatar Inc. Script-scan for prescription labels, 2020. URL https://play.google.com/store/apps/details?id=com.scriptscan&gl=GB.

Goel, A. Looking back, looking ahead: Symbolic versus connectionist ai. *AI Magazine*, 42(4):83–85, 2022.

Gomez, A., Diodati, G., Martínez von Scheidt, M., Luna, D., and Quirós, F. *Augmented Reality: Real-Time Information Concerning Medication Consumed by a Patient*, volume 216. August 2015. Journal Abbreviation: Studies in health technology and informatics Publication Title: Studies in health technology and informatics.

Google. Google healthcare nlp, 2022a. URL https://cloud.google.com/healthcare-api/docs/concepts/nlp.

Google. Google vision ocr, 2022b. URL https://cloud.google.com/vision/docs/ocr.

GS1 US. Gs1 standards resources for dscsa implementation support, 2022. URL https://www.gs1us.org/industries/healthcare/gs1-standards-in-use/pharmaceutical/dscsa-resources.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

L. Magalhães, B. Ribeiro, N. Alves, and M. Guevara. A three-staged approach to medicine box recognition. In *2017 24º Encontro Português de Computação Gráfica e Interação (EPCGI)*, pp. 1–7, October 2017. doi: 10.1109/EPCGI.2017.8124317. Journal Abbreviation: 2017 24º Encontro Português de Computação Gráfica e Interação (EPCGI).

Larios, N., Usuyama, N., Hall, A., Hazen, R., Ma, M., Sahu, S., and Lundin, J. Fast and accurate medication identification. *npj Digital Medicine*, 2:10, February 2019. doi: 10.1038/s41746-019-0086-0.

Lee, S., Jung, S., and Song, H. Cnn-based drug recognition and braille embosser system for the blind. *Journal of Computing Science and Engineering*, 12(4):149–156, 2018.

Lee, Y.-B., Park, U., Jain, A. K., and Lee, S.-W. PillID: Matching and retrieval of drug pill images. *Pattern Recognition Letters*, 33(7):904–910, 2012. Publisher: Elsevier.

Lester, C. A., Li, J., Ding, Y., Rowell, B., Yang, J. X., and Kontar, R. A. Performance evaluation of a prescription medication image classification model: an observational cohort. *npj Digital Medicine*, 4(1): 118, July 2021. ISSN 2398-6352. doi: 10.1038/s41746-021-00483-8. URL https://doi.org/10.1038/s41746-021-00483-8.

Liu, X., Meehan, J., Tong, W., Wu, L., Xu, X., and Xu, J. DLI-IT: a deep learning approach to drug label identification through image and text embedding. *BMC Medical Informatics and Decision Making*, 20 (1):68, April 2020. ISSN 1472-6947. doi: 10.1186/s12911-020-1078-3. URL https://doi.org/10.1186/s12911-020-1078-3.

Mündler, N. Quantulum3. https://github.com/nielstron/quantulum3, 2022.

Naeem, M. and Coronato, A. An AI-Empowered Home-Infrastructure to Minimize Medication Errors. *Journal of Sensor and Actuator Networks*, 11(1), 2022. ISSN 2224-2708. doi: 10.3390/jsan11010013.

National Library of Medicine. Rxnorm appendix 2 dose forms, 2022a. URL https://www.nlm.nih.gov/research/umls/rxnorm/docs/appendix2.html.

National Library of Medicine. Rxnorm, 2022b. URL https://www.nlm.nih.gov/research/umls/rxnorm/overview.html.

Negi, A., Bhure, A., Patil, D., Maskara, A., and Bhalekar, M. Medicine Identification Application for Visually Impaired People. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(14):748–755, 2021.

Pathak, J., Murphy, S. P., Willaert, B. N., Kremers, H. M., Yawn, B. P., Rocca, W. A., and Chute, C. G. Using rxnorm and ndf-rt to classify medication data extracted from electronic health records: experiences from the rochester epidemiology project. In *AMIA Annual Symposium Proceedings*, volume 2011, pp. 1089. American Medical Informatics Association, 2011.

Redmon, J. and Farhadi, A. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

Roopa, K., Kumar, P. S., and Kumar, G. R. Portable Camera Based Assistance Label Reading for Blind Person. 2016.

Roy, A. Identification in Drug Prescription Using Artificial Intelligence. *Available at SSRN 4012788*, 2022.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Sarzynski, E., Decker, B., Thul, A., Weismantel, D., Melaragni, R., Cholakis, E., Tewari, M., Beckholt, K., Zaroukian, M., Kennedy, A. C., and Given, C. Beta Testing a Novel Smartphone Application to Improve Medication Adherence. *Telemedicine and e-Health*, 23(4):339–348, April 2017. ISSN 1530-5627. doi: 10.1089/tmj.2016.0100. URL https://doi.org/10.1089/tmj.2016.0100. Publisher: Mary Ann Liebert, Inc., publishers.

Search.io. Fuzzy. https://github.com/sajari/fuzzy, 2021.

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

Smith, R. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pp. 629–633. IEEE, 2007.

Smolensky, P. Connectionist ai, symbolic ai, and the brain. *Artificial Intelligence Review*, 1(2):95–109, 1987.

Thompson, N. C., Greenewald, K., Lee, K., and Manso, G. F. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, 2020.

Tian, Z., Huang, W., He, T., He, P., and Qiao, Y. Detecting Text in Natural Image with Connectionist Text Proposal Network, 2016. URL https://arxiv.org/abs/1609.03605.

Ting, H.-W., Chung, S.-L., Chen, C.-F., Chiu, H.-Y., and Hsieh, Y.-W. A drug identification model developed using deep learning technologies: experience of a medical

center in Taiwan. *BMC Health Services Research*, 20 (1):312, April 2020. ISSN 1472-6963. doi: 10.1186/s12913-020-05166-w. URL https://doi.org/10.1186/s12913-020-05166-w.

Tran, T. T., Richardson, A. J. W., Chen, V. M., and Lin, K. Y. Fast and Accurate Ophthalmic Medication Bottle Identification Using Deep Learning on a Smartphone Device. *Ophthalmology Glaucoma*, 5(2):188–194, 2022. ISSN 2589-4196. doi: https://doi.org/10.1016/j.ogla.2021.08.001. URL https://www.sciencedirect.com/science/article/pii/S2589419621001848.

W. Chang, L. Chen, C. Hsu, C. Lin, and T. Yang. A Deep Learning-Based Intelligent Medicine Recognition System for Chronic Patients. *IEEE Access*, 7:44441–44458, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2908843.

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., and Wu, Y. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1386–1393, 2014.

Wang, J.-S., Ambikapathi, A., Han, Y., Chung, S.-L., Ting, H.-W., and Chen, C.-F. Highlighted Deep Learning based Identification of Pharmaceutical Blister Packages. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pp. 638–645, 2018. doi: 10.1109/ETFA.2018.8502488.

X. C. Benjamim, R. B. Gomes, A. F. Burlamaqui, and L. M. G. Gonçalves. Visual identification of medicine boxes using features matching. In *2012 IEEE International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS) Proceedings*, pp. 43–47, July 2012. ISBN 1944-9410. doi: 10.1109/VECIMS.2012.6273190. Journal Abbreviation: 2012 IEEE International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS) Proceedings.

Yaddanapudi, S. *Machine Learning Based Drug-Disease Relationship Prediction and Characterization*. PhD thesis, University of Cincinnati, 2019.

Zauner, C. Implementation and benchmarking of perceptual image hash functions. 2010.

ZBar. Zbar, 2022. URL http://zbar.sourceforge.net/.

A. Strength Comparison

Wherever we compare two strengths, if their strings do not match, then compare their normalized representations. E.g Suppose we detect the substring “20 mg per 5 ml” is a strength in the text and a candidate contains an ingredient with strength “4 mg/ml” then we would like to say a strength match exists. To detect a strength substring in the text we use quantulum3 (Mündler, 2022). To normalize the strength substrings, the strength algorithm detects and parses fractional strengths into (*numerator, denominator*) and each part has a (*value, unit*). It normalizes the units, with the appropriate multiplier applied to the values. Then it divides the numerator value by the denominator value so that the new representation has a denominator value of 1.

B. Dataset Samples

(a) Popular Branded, Popular Generic and Realistic

(b) Daily-med

Figure 2. Sample images from the datasets with visible and without NDC. For the popular branded dataset, an example of before and after masking NDC is shown.

C. NDC Examples

NDC10 Format	[4]-[4]-[2]	[5]-[3]-[2]	[5]-[4]-[1]
NDC10 Value	0074-3054-53	50580- 608-01	53451-0101- 1
Conversion to NDC11	**0**0074-3054-53	50580-**0**608-01	53451-0101-**0**1

Figure 3. Cropped images of three different NDC10 formats and their conversion to NDC11

D. Detailed Breakdown of Latency

Table 4. Breakdown of latency within the PuPill API

Process	Mean Time (ms)
Google OCR & Spell Correction	334.00
NDC-Path	0.96
Total by End of NDC-Path	334.96
Google NLP	549.00
NLP-Path (excl. Google NLP)	827.00
Total by End of NLP-Path	1,710.96
Save Inference	52.30
Persist Image	170.00
Total	732.01

E. Choice of NLP Service

Either Google Healthcare NLP or AWS Comprehend Medical could have been chosen for the Step 2B of the NLP-Path. Both take as input a text and output drug-related entities mentioned in the text along with their RxCui codes and linked strengths where appropriate. For Google, the types of RxCui it can output are BN, IN, PIN but for AWS, they also output medication types SCD, SBD, GPCK and BPCK. However in experiments, we found that we could not rely on the medication types produced by AWS to be a good set of candidates instead of taking the BN, IN, PIN codes and traversing the RxNav Graph as in Step 3B. The AWS service is not able to make the assumption that there is only a single medication being described in the text. As a result, it tends to output SCD, SBD, GPCK, BPCK identifying single ingredient medications for each ingredient it detects. Of the two services, the BN, IN, PIN RxCui from Google were produced candidates containing the correct medication more often.

F. Comparison to Related Work

Table 5. Comparison of PuPill with Related Works

PAPER	PATHWAYS	CLASSES	PACKAGE TYPE	ACCURACY	MEDICATION LEVEL
PUPILL (OURS)	TEXT	12197	ALL	91.1%	RXCUI
(LIU ET AL., 2020)	IMAGE, TEXT	669	LABELS	80%	DAILY-MED SETID
(TRAN ET AL., 2022)	IMAGE	5	BOTTLES	88%	GENERIC NAME
(TING ET AL., 2020)	IMAGE	250	BLISTER	"BETTER THAN 90%"	GENERIC NAME + STRENGTH
(WANG ET AL., 2018)	IMAGE	272	BLISTER	"ALMOST 100%"	-
(NEGI ET AL., 2021)	TEXT	-	ALL	74%	NAME
(X. C. BENJAMIM ET AL., 2012)	IMAGE	7	BOX	98.57%	-