P.I.L.L.: A Deep Learning Model for Prescription Pill Recognition and Classification

Jett Badalament-Tirrell
School of Data Science
University of Virginia
hwq8mr@virginia.edu

Olivia Byram
School of Data Science
University of Virginia
ocb3wv@virginia.edu

Hamsini Muralikrishnan
School of Data Science
University of Virginia
hm7qgr@virginia.edu

***Abstract*—This study develops a deep learning model for automated pill identification from images. Using the NIH dataset of 4,392 pill images across 2,112 classes at a pill-level split, we trained a baseline model, a ResNet18 trained from scratch, and achieved 1.48 percent top 1 validation accuracy, as well as a 4.58 percent top 5 validation accuracy. This low accuracy prompted the expansion of the baseline model, specifically an initial attempt at optical character recognition (OCR). Using a pretrained ResNet18 improved the top 1 validation accuracy to 3.41 percent and the top 5 validation accuracy to 9.86 percent. It was determined that the image dataset used did not include enough images per class to successfully complete its task and implement Aster or DeepTextSpotter for OCR. Because of this, we looked to implement KNN-retrieval as a second step for our model to actually retrieve the K-nearest neighbors to the query pill embedding from pretrained ResNet18 and also predict the pill class/name. This resulted in a top 1 validation recall of 17 percent and top 5 validation recall of 19 percent.**

As part of this work, we also collected 'real-world' images that were less staged compared to our dataset to supplement our data and test model sensitivity. With real world data, there was a challenge with domain shift, as poor lighting, cluttered backgrounds, and blurs degraded the embedding quality picked up by ResNet18. This poor embedding quality was carried downstream to KNN and affected the performance in real-world data, indicating the sensitivity and inability of the model with this data.

Future work includes collecting more images per class to ensure a more robust dataset, training for more epochs/ longer training time, obtaining better/more real-world data or using domain adaptation techniques which all could significantly improve accuracy. GitHub Repo Link: https://github.com/ocbyram/Group11_DS6050_PILL/tree/main.

I. Introduction and Motivation

It is estimated that approximately 68 percent of all Americans take at least one daily prescription pill, and a large amount of that percentage take more than one. It is common practice to put daily prescriptions and vitamins in pill cases, whether it be for easy storage, travel, or simply to remember if they have taken their daily pills. However, many of those pills look incredibly similar, and it is all too easy to mix them up when they reside in the same section of a pill case. This can lead to confusion, and often results in someone taking the wrong medication at the wrong time. While often benign, this could be life threatening for people on necessary medications. The FDA requires that all pills have a distinct shape, color, size, score, and imprint code. However, the scoring and imprints tend to rub off and there are only so many colors, sizes, and shapes that a pill can be. We propose a solution to this being a deep learning model that is able to take an input of an image of a pill, then output the name of the pill. The name of this model is P.I.L.L. (Pill Identification and Labeling Locator). Our research question is 'Can we develop a deep-learning model that accurately identifies and classifies prescription pills from images using visual features?' We hypothesize that pretrained ResNet18 with KNN retrieval will significantly outperform both from-scratch and pretrained classification approaches.

Pill identification, specifically with deep learning models, is a difficult task. There are thousands of pills, and only a limited amount of distinct shapes, colors, and sizes that they could be. Due to this limitation, many pills wind up looking similar to each other, despite having drastically different functions. Deep learning models can struggle to identify these pills correctly because of factors such as image quality, unclear imprints/text, and the rarity of three dimensional data. Many pill recognition models, such as the one created in [1] are trained on professional quality images. While this works excellently on training and test data, the average user will not have access to images that are free from background clutter or perfectly angled, so their inputs tend to be mislabeled. Additionally, many pills have faded or smudged text, making it impossible for the model to accurately read and classify based on the FDA regulated imprint code. This is an unreliable manner of identifying pills due to how rare it is to have perfect imprints on a pill. Finally, many pills are identifiable due to their size and shape. However, with two dimensional data, it is incredibly difficult for a model to identify the precise shape of a pill. Three dimensional data is rare and difficult to train, meaning that researchers with limited resources must do their best with two dimensional data instead.

Throughout our research, we attempted to address these challenges within pill identification. We created multiple models that do not focus solely on imprints/text, and leveraged multiple 2D views of each pill (front and back images) in order to fill the gap left by no three dimensional images. We implement and compare three approaches: a from-scratch ResNet18 classifier, a pretrained ResNet18 classifier, and a pretrained ResNet18 with K-nearest neighbor retrieval. Our experiments demonstrate that retrieval-based methods are particularly well-suited to our limited data.

II. Dataset

We originally began this experiment with a Kaggle dataset (https://www.kaggle.com/datasets/trumedicines/pharmaceutical-tablets-dataset). This was an open-source dataset and contains 252 unique pill images, where each pill image had been convoluted to create 20,000 images by rotating the pills, adding a gray scale, etc. The unique pill images varied in pill size, color, shape, and imprint, allowing a dataset that is diverse. Unfortunately, during the data loading process in Milestone II, we realized that the data is unusable in its current format. Despite being open-source, the images are in a dataset that needs to be parsed with specific code. The owners of the data did not provide the code, meaning we would have to reverse engineer the code, to decode the images. This method is close to impossible since we did not know the general structure of the images to write a decoding algorithm. We discussed this with the teaching assistants, and they suggested performing a necessary dataset switch. We chose another open-source dataset, from the NIH [4]. There are approximately 4,300 reference pill images in this dataset, with pictures of the pills from multiple angles. The dataset content is similar to the original dataset we had, but it is smaller, indicating that we had to perform more data augmentation. Unfortunately, the pills images within this dataset did not always have clear imprints. There were less than 50 percent of the images with clear imprints/text, indicating that we would not be able to perform OCR or use DeepTextSpot/Aster. Figure 1, below, is an example of one of the pills in the dataset. It has both the front and back of the pill pictured.

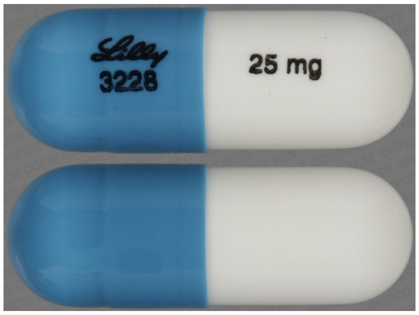
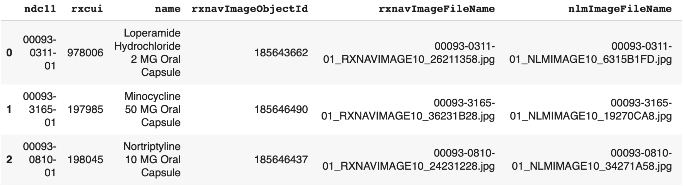


Fig. 1: Example pill image from the NIH dataset showing front and back views.

In addition to the pill images, the dataset included a csv table that included the name of each pill and its associated image file name. This allowed us to easily match each image with its name when training and analyzing the accuracy of our models. An example of the pill name .csv file is included as Figure 2.

After switching datasets, we found that there were 2,112 classes within the 4,393 images. This meant that each pill had multiple views. Due to the few-shot nature of this data, we chose to perform data augmentations of rotations, flips, and jitters. This increased the amount of data that we had to work with, and allowed us to create our splits. The NIH dataset assigns a unique identifier per pill, so we were able to use that identifier to group images and split at the pill level. Our dataset was split at the pill level, meaning that all images of a single pill were assigned to the same split (train, validation, or test).



	ndc11	rxcui	name	rxnavImageObjectId	rxnavImageFileName	nlmImageFileName
0	00093-0311-01	978006	Loperamide Hydrochloride 2 MG Oral Capsule	185643662	00093-0311-01_RXNAVIMAGE10_26211358.jpg	00093-0311-01_NLMIMAGE10_6315B1FD.jpg
1	00093-3165-01	197985	Minocycline 50 MG Oral Capsule	185646490	00093-3165-01_RXNAVIMAGE10_36231B28.jpg	00093-3165-01_NLMIMAGE10_19270CA8.jpg
2	00093-0810-01	198045	Nortriptyline 10 MG Oral Capsule	185646437	00093-0810-01_RXNAVIMAGE10_24231228.jpg	00093-0810-01_NLMIMAGE10_34271A58.jpg

Fig. 2: Pill name CSV file overview.

Some classes contain multiple distinct pill images in the dataset, so the distribution of classes across splits does not follow the same 70/15/15 split as the pill-level split. When a class contains multiple pills, those pills may be assigned to different splits. In this specific project, the train split contains 1,709 classes, the validation split contains 544 classes, and the test split contains 536 classes.

III. Methodology

To develop our methodology, we identified and studied three research experiments/papers. These papers are cited in our references section as [1], [2], and [3]. They all perform some variation of pill recognition through deep learning models, and overall inspired us to begin with a ResNet18 model.

As all three studies did, we focused our models on two primary steps: pill recognition and pill retrieval. For pill recognition, we focused on pill color, shape, size, and embossing to increase the accuracy of the model. Pill retrieval was performed using the identified features to retrieve the pill name from a set database.

We implemented three unique approaches to complete this research project. We created our first baseline mode, which was a 'from scratch' ResNet18 model. This model establishes a lower bound on performance and demonstrates the difficulty of learning effective visual representations from limited data. With the ResNet18 structure, we chose to use 300 x 300 resolution images from our dataset. In general, ResNet18 works well with 224-250 pixel images. However, reducing the number of pixels degraded the quality of our images, so we chose to stick with 300x300 for the sake of accuracy. The model uses the standard ResNet18 architecture with convolutional layers, batch normalization, ReLU activations, and residual skip connections.

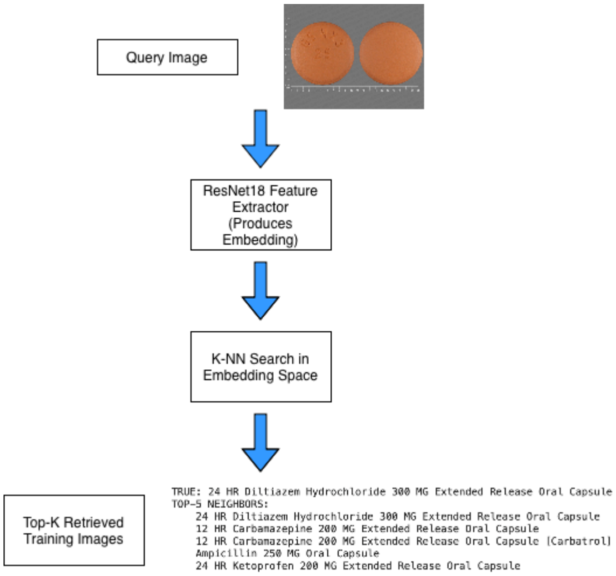
The architecture of our 'from scratch' ResNet18 consists of an initial convolutional block (7×7 convolution with 64 filters, batch normalization, ReLU activation, and max pooling) followed by four residual layers with channels 64, 128, 256, and 512. The network outputs a 512-dimensional feature vector, which is passed through a fully connected layer to produce 2,112 logits (one per pill class).

We trained the ‘from scratch’ ResNet18 model using cross-entropy loss with the Adam optimizer (which has a learning rate of 1e-3 and batch size of 32) for 10 epochs. This learning rate is optimal as it allows for smooth updates to the model with small steps, ensuring adequate convergence. We applied the data augmentation pipeline described in the data section of this paper during training.

The second approach also served as our second baseline, a pretrained ResNet18 model. The difference between a ResNet18 model built from scratch and one that is pretrained is that the weights in the scratch model are randomly initialized, while the weights in the pretrained model are initialized with the ImageNet pretrained weights. The advantage of using a ‘from scratch’ ResNet18 model is that there is much less bias, as it begins with random parameters and trains from a complete baseline. This allows us to completely control what the model is learning through training. However, it does not perform well on small datasets, which we have mentioned as a concern. Conversely, a pretrained ResNet18 model does much better with smaller datasets. This is because the model is pretrained on 224 x 224 ImageNet with a randomly initialized classification head. It tends to learn much quicker than a scratch model because it already has a baseline that is able to identify edges and other aspects of images. However, there is a concern of bias and several uncontrollable factors since you are not training from the ground up. We were interested in comparing these two approaches to determine which one worked best with our limited dataset.

To implement the pretrained model, we replaced the last layer with 2,112, the total number of classes that we had. The architecture of the pretrained model was exactly the same as the ‘from scratch model’, giving us a clean slate to perform an ablation study as part of our experiments. Additionally, we trained this pretrained ResNet18 model exactly the same as the scratch model, using cross entropy and the Adam optimizer, which works by updating the network weights using both momentum and the gradient. Adam was used to finetune the parameters in the model automatically with a learning rate of 0.0001 and batch size of 32. Again, we trained for ten epochs and used the same data augmentation pipeline here during training as noted in the data section.

The third approach became our final model. This approach used the Pretrained ResNet18 model as a feature extractor, with KNN search for retrieval instead of a classification head. The methodology of our final model follows the pipeline shown in Figure 3. Given a query pill image, we extract visual features using a ResNet18 encoder pre-trained on ImageNet. These features form an embedding vector. We then used K-nearest neighbor (KNN) search with cosine similarity to retrieve the top-k most similar pill images from our training set, along with their class labels. The final prediction is determined by majority voting among the retrieved labels.


Query Image
ResNet18 Feature Extractor (Produces Embedding)
K-NN Search in Embedding Space
Top-K Retrieved Training Images

```
TRUE: 24 HR Diltiazem Hydrochloride 300 MG Extended Release Oral Capsule
TOP-5 NEIGHBORS:
    24 HR Diltiazem Hydrochloride 300 MG Extended Release Oral Capsule
    12 HR Carbamazepine 200 MG Extended Release Oral Capsule
    12 HR Carbamazepine 200 MG Extended Release Oral Capsule [Carbatrol]
    Ampicillin 250 MG Oral Capsule
    24 HR Ketoprofen 200 MG Extended Release Oral Capsule
```

Fig. 3: Pipeline of the pill identification methodology.

We chose the KNN search for retrieval due to its strong ability to use embedding directly, whereas classifiers have to finetune in order to use embeddings. KNN is particularly well designed for small datasets that do not include many images per class. This is because it is able to generalize for small datasets and prevents overfitting. By adding the KNN search for retrieval to our pretrained model, we are eliminating the need for classifiers. Classifiers typically need a large number of examples and overfit when there are not enough images in each class, making it a poor choice for our research and dataset.

To implement KNN, we extracted the features with the pretrained ResNet18 model and used the cosine metric during our implementation of sklearn.neighbors’ KNeighborsClassifier function. Additionally, we set the neighbors equal to five because we wanted to return the top five matches for each pill. By returning the top five matches, we are providing multiple options to a user and increasing the chances that one of them will be correct. It is essential to provide a user multiple choices when it is a potentially life-threatening scenario because they are able to weigh the options and make the best possible choice for themselves based on what they know about their own medication history/usage.

Overall, our methodology was to create and compare two visual-only models, a scratch ResNet18 and a pretrained ResNet18. We then chose to use the pretrained ResNet18 embeddings to create a retrieval pipeline with KNN. Originally, we hoped to use OCR, but were unable to due to a lack of clear textual embeddings on the images. We compensated for this with our KNN search for retrieval instead. In the next section, we will describe how we evaluated each aspect of this method.

IV. Experiments

As we mentioned in the data section, we chose to split our data at pill-level. To reiterate, this means that all images of a single pill were assigned to the same split (train, validation, or test). We used a general 70/15/15 ratio split for training/testing/validation (respectively). This was not precisely followed because of the pill-level split. When a class contains multiple pills, those pills may be assigned to different splits. As a result, the train split contained 1,709 classes, the validation split contained 544 classes, and the test split contained 536 classes.

To evaluate our method, we chose to complete two ablation studies as well as conduct an error analysis. We chose to complete ablation studies because we wanted to compare the changes that we implemented in each step of our approach. This allowed us to easily compare the three models against each other and choose the most comprehensive and accurate. We chose to do an error analysis in addition so that we could understand the limitations of our final model and address general patterns of issues that we discover. Originally, our methodology planned to use OCR with the systems DeepTextSpotter and Aster. Unfortunately, this was unable to happen due to data limitations and the lack of images with clear text. Had this been possible, we would have completed a third ablation study comparing the performance of visual only models versus visual+text models.

For the first ablation study, we focused on comparing the 'from scratch' ResNet18 model to the pretrained ResNet18 model. The purpose of this study was to ultimately justify which model was suited better for the pipeline for feature extraction. To implement this study, we wrote the code for our models as defined in the methodology, then evaluated both models after training for ten epochs each.

We did a head to head comparison of the 2 models, using top 1 and top 5 accuracy as our comparison metrics. Essentially, top 1 accuracy quantifies the percentage of samples where the most probable class is the true class and top 5 accuracy is the percentage of samples where the 5 most probable classes contain the true class/ pill. We chose to use accuracy over other metrics because we were focused on finding the model that produces the most correct labels. Having low accuracy indicates that the model will not generate correct labels, even if the other metrics are high. We used the results of this evaluation to decide which model to move forward with in the model pipeline.

For the second ablation study, we shifted our focus to comparing the pretrained ResNet18 (also known as the softmax classifier) model to our hybrid approach of utilizing KNN in addition to the pretrained ResNet18 model. Similarly to ablation study 1, we trained the models again and ran the pretrained ResNet18 model with KNN, then evaluated both approaches.

Within this second study, we evaluated by calculating the following metrics: recall@1, recall@5, recall@10 and MRR, with the goal of isolating if the KNN addition would truly prove useful. recall@k is a metric designed to identify if the correct pill is within the top k retrieved results by KNN. MRR on the other hand, instead of using a binary approach like recall@k, gives a rating that adjusts based on how high the target class appeared. If the correct pill was the closest within the embedding space as returned by KNN, it would get the highest MRR score, if it was only the third closest, it would get a lower MRR score, and so on.

Outside of the ablation studies, we wanted to determine how and why errors were occuring once we determined our final model. We wanted to get more clarity on if the embedding space had meaningful structure, regardless of how the addition of KNN affected performance. We generated full top-5 prediction sets for every single image in the test/validation split, and examined in which cases the top-1 prediction was wrong. In these cases, we logged the true label, the top-1 predicted label, and the full set of 5 returned by the KNN.

With this we were able to see which pills were most frequently misclassified, and also which pill pairs get confused with one another most often. We then summarized these patterns, counting how often each pair occurred and created a visual to keep track of the most common confusions.

The goal of this experiment was to determine if the confusions made by the model were reasonable, as in if the pills often confused were similar in size, shape, and color, ideally to a level where even a human might confuse the two.

Finally, we evaluated the generalization of our hybrid retrieval system under the domain shift of moving to "real-world" pill photographs, by this we mean to describe images taken at non-straight angles, perhaps from an individual's iphone in varying lighting conditions. These images different heavily from our NLM dataset, which had straight-on angles under direct lighting, with little in the images aside from the pills themselves. It is also worth noting that we ensured the real-world images contained pills within our training set. For each of the real-world images we collected, we passed the image through our pipeline extracting the top-5 nearest pills to assess performance.

In the next section, we discuss our comprehensive results and what they mean.

V. Results and Discussion

In Milestone II, we reported 89.6 percent validation accuracy for the pretrained ResNet18 using image-level splits. However, this metric was misleadingly high because front and back views of the same pill could appear in both training and validation sets, allowing the model to essentially recognize pills it had already seen from different angles rather than generalizing to truly unseen pills. Following feedback, we restructured our evaluation to use pill-level splits, where all images of a given pill remain together in a single split. Under this more rigorous evaluation, the pretrained ResNet18 achieves 3.41 percent top-1 accuracy and 9.86 percent top-5 accuracy, which is a dramatic decrease that reflects the true difficulty of generalizing to completely unseen pills in this extreme few-shot regime (average of 2 images per pill class). All results reported in this paper use pill-level splits to ensure realistic evaluation.

It is also important to note that our data provided a severe limitation in the overall accuracy of each approach. With such extreme few-shot examples, the accuracy and generalization was essentially capped at a relatively low level.

Table 1, below, shows the results from our first ablation study, comparing the ‘scratch’ ResNet18 to the pretrained ResNet18. The results show us that the scratch model’s top 1 accuracy was 0.0148, and the top 5 accuracy was 0.0458. This is a small increase, but not significant. Comparatively, the pretrained model’s top 1 accuracy was 0.0341 (which is approximately .02 higher than the scratch model) and the top 5 accuracy was 0.0986 (approximately 0.05 higher than the scratch model). The pretrained ResNet18 model performed better despite having the exact same architecture as the scratch ResNet18 and being trained in the same manner. This indicates that the reason the pretrained model performs better is due to the weights being initialized with the ImageNet pretrained weights, rather than randomly initialized as the scratch model was. The pretrained model was able to understand edges, shapes, colors, prior to even seeing the pill data as it was pretrained, giving it an advantage. This confirmed our original hypotheses that the pretrained model would perform better due to our data limitations, and reassured us as we moved onto the second step of our methodology, comparing the pretrained model with the pretrained ResNet18 + KNN model.

Model	Top-1 Acc	Top-5 Acc
From scratch ResNet18	0.0148	0.0458
Pretrained ResNet18	0.0341	0.0986

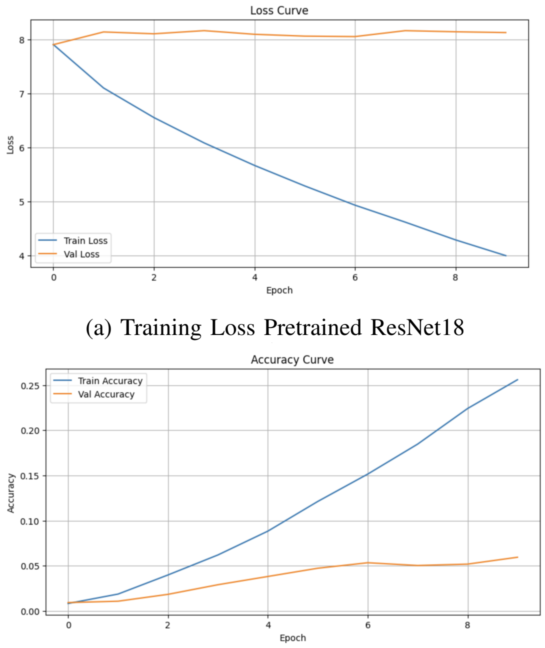
TABLE I: Ablation Study 1 Results

Below, in figure 4, we show the training/validation loss and accuracy curves that we created from training the pretrained model. As you can see from these, our training loss got significantly worse over time while validation loss stayed somewhat stable. Additionally, training accuracy got fairly high while validation accuracy stayed steady and increased only slightly. This indicates that there could be slight overfitting while training, which we hope to eliminate by adding in KNN.

The second ablation study, with results shown in Table 2 below, compared the pretrained ResNet18 model as the “Softmax Classifier” to the ResNet18 model + KNN retrieval pipeline. As mentioned in the experiments section, we used recall@1 and recall@5 as the evaluation metrics for this study. The pretrained ResNet18 model achieved a recall@1 of 0.06 and recall@5 of 0.15. On the other hand, the pretrained ResNet18 + KNN retrieval achieved a recall@1 of 0.17 and recall@5 of 0.19. This indicated that the method with ResNet18 + KNN retrieval performed much stronger than the softmax classifier. Due to the results of this study, we decided to move forward with the ResNet18 + KNN model pipeline to create the pill embeddings, retrieve the nearest neighbors to the query pill, and ultimately, predict the pill. Overall, the results that we gathered from these two ablation studies allowed us to scientifically choose and defend our final method.

Method	**Recall@1**	**Recall@5**
Softmax Classifier	0.06	0.15
K-NN Retrieval	**0.17**	**0.19**

TABLE II: Ablation Study 2 Results

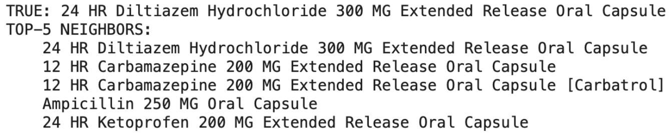


(a) Training Loss Pretrained ResNet18

(b) Training Accuracy Pretrained ResNet18

Fig. 4: Training metrics for Pretrained ResNet18

Since we decided the pretrained ResNet18 model with KNN was the best method, we have included an example of the output that a user receives after inputting an image of their pill. The user is returned a list of the top 5 neighbors (top five most likely pill labels) in order of most likely to least likely. Figure 5 below shows the actual pill label next to ‘TRUE:’ followed by the five nearest neighbors. This output shows a clear success where the first label returned is correct.



```
TRUE: 24 HR Diltiazem Hydrochloride 300 MG Extended Release Oral Capsule
TOP-5 NEIGHBORS:
    24 HR Diltiazem Hydrochloride 300 MG Extended Release Oral Capsule
    12 HR Carbamazepine 200 MG Extended Release Oral Capsule
    12 HR Carbamazepine 200 MG Extended Release Oral Capsule [Carbatrol]
    Ampicillin 250 MG Oral Capsule
    24 HR Ketoprofen 200 MG Extended Release Oral Capsule
```

Fig. 5: Final Method Output.

After choosing and practicing with our final pipeline, pretrained ResNet18 + KNN, we completed our error analysis study. This was incredibly important due to the low recall scores that we were achieving with our final pipeline. The results of the error study were able to show us the top twenty-five most confused pills, with the number one most confused being Loperamide Hydrochloride 2 MG Oral Capsule confused as Doxycycline Monohydrate 100 MG Oral Capsule [Monodox]. We have included figure 6 below to show the most confused class pairs. As we had hoped, the model was making mistakes on pills that were particularly visually similar indicating a meaningful feature embedding space.

With this, we were able to look at the differences between the most commonly mismatched pills/labels and identify general patterns that seem to cause errors. The largest indicator is

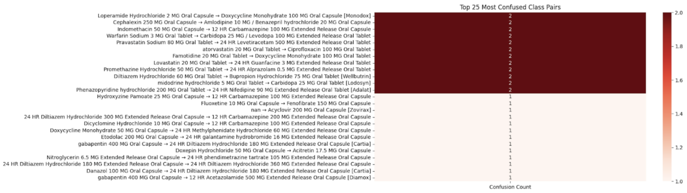


Fig. 6: Top 25 Most Confused Class Pairs.

when pills have very similar shapes and sizes. For example, the top most confused pill is shown below in figure 7. Both pills are capsules that have the exact same shape and seemingly the same size due to the image quality. Additionally, they are similar colors as one is a light orange and the other is dark yellow. The trend of similarly colored and shaped pills being confused most often continued for each of the top twenty five.

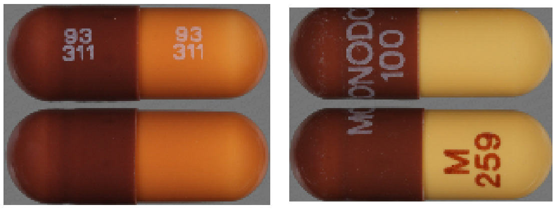


Fig. 7: Fail Pill Match, Top 25 Cases.

Shown in figure 8, below, there is an example of a success case for our pill recognition model. The pretrained ResNet18 + KNN model was able to accurately label the pill on the right of the figure as Diltiazem Hydrochloride 300 MG, after using the pill image on the left for training. As we discussed in the error analysis, the model does well with general patterns. It tends to accurately identify study pills that have distinct colors and shapes, as well as clear imprints. The shown in our success case is an abnormal color that is not common among pills. This allowed the model to significantly decrease the number of options for the correct label. Additionally, the imprint on the pill is distinct and clear on the pill, which aids in matching the label as well. We were surprised that this case performed as well as it did, due to the shape of the pill being a general capsule that is common amongst pills. This indicates that color and imprints are among some of the most important features of pill identification.

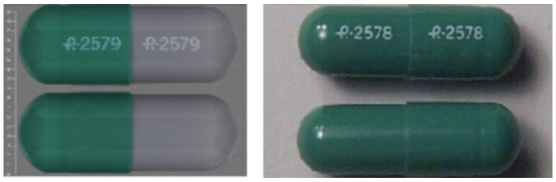


Fig. 8: Success Pill Match.

Conversely, figure 9, below, shows an example of a failure case with our final model. This was not in the top twenty-five most confused cases, which indicates the general patterns continue through the entire dataset. This case was less surprising to us due to the common shape and color of the pill. This image shows the pill Dicyclomine Hydrochloride 10 MG Oral Capsule on the left was confused with the pill Clindamycin 150 MG Oral Capsule on the right. Both pills have similar coloring as RGB images, and their shading is similar as well (darker on the left side of the pill, and lighter towards the ends). Additionally, they both appear to be the exact same shape and size, which is a fairly prevalent choice that pharmaceutical companies use when creating pills. This example reiterates the fact that our model performs poorly on standard pill colors and shapes.

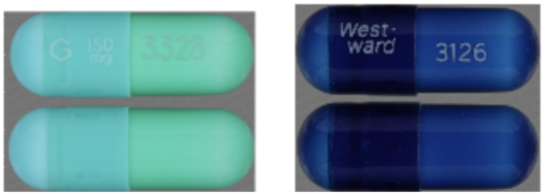


Fig. 9: Fail Pill Match, Not Top 25

Overall, our model performs best at pattern recognition when the pills have distinct colors, sizes, shapes, and imprints. Specifically, pills with contrasting coating colors and imprint colors performed well as it was easier for the model to recognize and decipher, as expected. We tended to see many incorrect pill labels on small white pills, as those are the most common shape and color of pill and their imprints are normally unclear and hard to decipher. We saw a larger amount of success on pills with unique colors or shapes. The model seems to be unable to generalize well for the pills that would be most commonly confused for each other by the user as well. This indicates the model needs to be adjusted in the future, as the ideal use case is to distinguish between very similar pills.

Finally, we used our model with five real-world images that we collected. These images had poor lighting, were blurry, and had cluttered backgrounds. The recall of the model plummeted to 0 for both recall@1 and recall@5, indicating that the model was unable to identify a single real-world image. This showed us that domain shift is a major challenge for real-world pill recognition. We noticed that the model was predicting similarly colored and shaped pills, which is promising overall since that was already an issue that we had with the NIH pill data. This could mean that our model would perform better on real-world images if we chose more distinctly colored, shaped, and sized pill images.

Overall, what worked for this model was KNN retrieval with the pretrained ResNet18 features. This method produced the highest metrics during our ablation studies, with a recall@1 of 0.17 and recall@5 at 0.19. In terms of how the model works, this indicates that it produces the correct pill label with the first result approximately 17 percent of the time, and the correct pill label within the first five results approximately 19 percent of the time. This is not significantly high, indicating that our data limitations had a higher impact than expected, even when implementing the pretrained ResNet18 and KNN to accommodate the dataset. It is also important to note that the model achieved 0 recall@1 and recall@5 on real world images, since this has a large impact on the intended use. In the next section, we discuss limitations and future works.

VI. Limitations and Future Work

Our work has several limitations that suggest directions for future research.

The largest limitation of this research was the extremely limited dataset. We struggled to find a reasonable, open source dataset that had enough pictures of each class in order to have accurate splits. There were only a couple of pill images for each class, which did not allow us enough flexibility within our training. We used simple data augmentations such as flips, rotations, and jitter because we had so few images within each class. These provided a small improvement, but with so few examples, the model still struggled to generalize and accuracy remained capped at a relatively low level. The second limitation that we identified was that this was a visual-only approach, with no textural evidence. As noted in our dataset description, fewer than 50 percent of pill images in the NIH dataset have legible imprints, making OCR-based methods unreliable for our work. If we had been able to incorporate textual features, pill identification accuracy may have been higher. The third limitation was the domain shift to real world-images with our final pretrained ResNet18 + KNN model. Our model was unable to accurately predict the label of any real-world pill image, indicating that it does not perform well and that domain shift is still a large challenge for pill-recognition. An example of one of the real-world pill images that we used for this analysis is shown in figure 10.

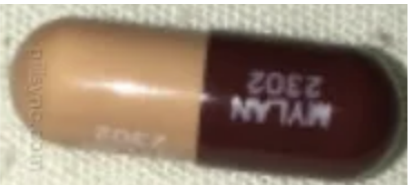


Fig. 10: Real World Pill Image.

A fourth limitation identified was that our model assumes there are no multi-pill images. This means that a user is not able to input an image of their pill container or multiple pictures next to each other. This limits the practicality for anyone looking to use this model.

The final limitation that we have identified is that we only trained each model for ten epochs due to computational resource constraints. Often, at least fifteen to thirty epochs are needed for the model to fully learn and be able to generalize patterns to test data. This likely contributed to our low accuracy and recall.

There are several solutions to the limitations and challenges that could be addressed in future work.

In the future, a larger and more diverse dataset should be collected for training and testing. Having at least ten images of pills per class would be ideal. Specifically, ensuring that some of those images are real-world images with background clutter, blurriness, and odd angles would help the model to compensate for non-professional images input by users.

Additionally, in the future, the images collected should have clear text or imprints. This would allow the completion of OCR, treated as a late-fusion feature, so one could compute a simple text embedding (bag-of-characters, or a pretrained text encoder). Additionally, the tools Aster or DeepTextSpotter would be useful in an ablation study to identify the most accurate character recognition system.

In the future, someone with a larger amount of resources and computational memory would likely benefit from training the model for longer than ten epochs. This would allow the model to have more training time to recognize and memorize patterns that it could generalize to the test data. In turn, this would also allow for higher performance and therefore, higher accuracy with the model.

Another aspect of future work would be adapting this model to handle multi-pill images. Many users may have a need to import an image with three different pills that they cannot distinguish between. By adapting the final model to identify each pill accurately, the user would be able to separate the pills and correctly label them.

Finally, integrating an external pill database would be a fantastic approach in the future. Many pharmaceutical companies have extremely comprehensive datasets, which would provide a significant amount of data for training and testing.

With these changes made with future work, Americans would certainly benefit from having access to this model that quickly identifies their pills when mixed up. It's common practice to use pill organizers and this model would definitely eliminate the guessing game that people often play when trying to figure out which pill to take, avoiding confusion and any adverse events. Although LLMs do have capability to process images, they are not specifically trained on pill databases, making them not as specialized and therefore, not as reliable. Again, with a specialized model pipeline like P.I.L.L, users can focus on their health and wellness without frustration.

VII. References

References

[1] H. Heo, H. Kang, S. Lee, J. Jeong, and J. Kim, "An Accurate Deep Learning Based System For Automatic Pill Identification: Model Development and Validation," *JMIR Med Inform*, vol. 11, no. 6, e43948, 2023.

[2] S. Kavitha, R. Ramya, S. Singh, R. Kumar, and P. Bhattacharya, "Real-time pill identification and classification using deep learning framework for medicine inspection systems," *Journal of Healthcare Engineering*, vol. 2025, Article ID 9876543, 12 pages, 2025.

[3] T. Nguyen, L. Wang, H. Lee, et al., "High Accurate and Explainable Multi-Pill Detection Framework with Graph Neural Network-Assisted Multimodal Data Fusion," *IEEE Access*, vol. 11, pp. 123456–123470, 2023.

[4] National Institutes of Health, "Computational Photography Project for Pill Identification," *NIH Data Discovery Portal*, 2025. [Online]. Available: https://datadiscovery.nlm.nih.gov/id/dataset/5jdf-gdqh