香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Image-Text Based Dermatological Diagnosis Helper: Vector and Relational Database

CSC3170 Database System

School of Data Science

Yu Chen 120090789
Yusen Chen 120090891
Zhizhen Chen 120090823

Table of Contents

Abstract

Skin diseases have always been persistent challenges to human physical health. In the current healthcare environment, the imbalance between the number of doctors and patients has driven many individuals turn to self-detection online. However, internet-based information tends to be disorganized and unreliable. With the development of deep learning, its combination with dermatological diagnosis has demonstrated great effectiveness. In response to this evolving scenario, therefore, we introduce **Image-Text Based Dermatological Diagnosis Helper** - an innovative system designed to offer guidance for individuals who are seeking dermatological diagnosis and treatment. The system combines two types of databases: the relational database is for storing data including patients' basic information, symptom descriptions, information about associated diseases, recommended medical treatments, and drug information. The vector database is designed for storing pictures of different skin diseases' symptoms. Additionally, it facilitates a direct channel for individuals to purchase necessary medicines from nearby drug stores. To enhance user experience, we have developed a user-friendly interface (UI) that streamlines data queries and processing. The implementation of this database system is poised to significantly benefit patients by providing them with tools for self-diagnosis and self-treatment. By bridging the gap between online information and accurate medical guidance, this system has the potential to enhance the overall healthcare experience in a more accessible and efficient manner.

1 Background

The skin is the body's largest organ and an important barrier. The main function of the skin is to protect the human body from harmful substances and prevent the outflow of various nutrients in the human body. In human's daily life, skin health is affected by many factors, such as solar radiation, smoking and alcohol consumption. These factors not only affect the integrity of skin function, but also cause certain damage to the skin, which would lead to adverse effects on human health, and even threaten human life in serious cases. Therefore, skin diseases have become one of the common diseases of human beings. Skin diseases span all cultural regions and occur at all ages. About 30 to 70 percent of people are at high risk. According to the statistics of the National Health Commission, the total number of dermatological hospital visits in China in 2020 is about 8.839 million, compared with 2013, the number has increased by 2.25 million. Skin diseases not only have a significant impact on human beings, such as impaired daily activities, loss of interpersonal relationships, internal organ damage, and even lead to death. The condition can also constitute a mental illness, leading to isolation, depression, and even suicide. Therefore, dermatosis has become one of the major topics in the medical field.

However, the correct diagnosis of skin diseases is challenging because a variety of visual cues such as individual skin shape, body part distribution, skin color, scales, and arrangement should be used to facilitate diagnosis. The diagnostic process can be complex when the individual components are analyzed separately. Several examples of clinical images of typical skin diseases are shown in fig.1. As we can see, the similarities between the symptoms of different diseases are extremely high, so it is usually only experienced doctors who can use these methods to achieve good diagnostic accuracy. Therefore, it would be beneficial to develop an effective method that can automatically distinguish skin cancer from non-cancer and distinguish skin cancer types as an initial screening tool.

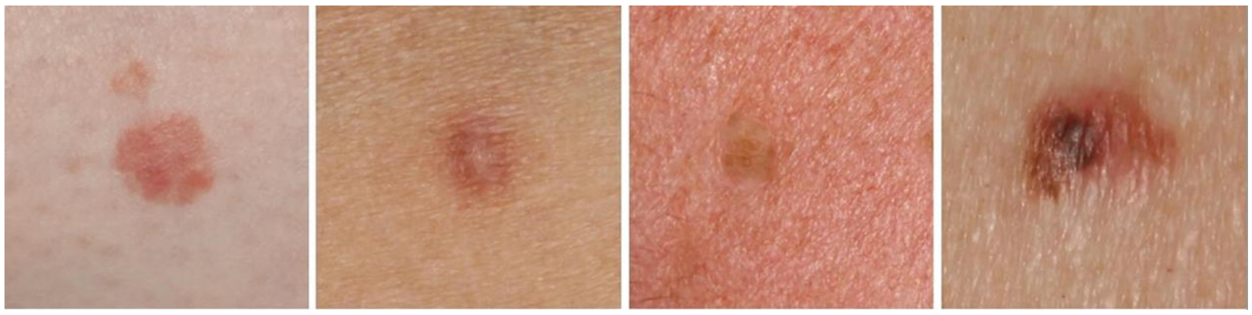


Figure 1 Examples of Skin Disease. From left to right, the illustrated diseases are Intraepithelial Carcinoma, Dermatofibroma, Seborrhoeic Keratosis and Melanoma, respectively

In 2017, China witnessed 240 million dermatology outpatient visits, yet only 22,000 dermatologists were recorded in the 2015 Chinese Medical Association census. This stark disparity highlights a significant shortage of dermatological medical resources. Faced with limited access to medical treatments and their associated high costs, many individuals turn to online platforms for information on unusual skin patterns. However, the internet rarely gives the answers people wants. Firstly, the abundance of inaccurate information online, due to its expansive and disorganized nature, poses a risk of misinformation and potential misdiagnoses. Secondly, current search engines lack digital search capabilities, complicating symptom descriptions and increasing the likelihood of misdiagnoses. Lastly, online information is often chaotic, with diverse responses to similar queries and pervasive medication advertisements, creating confusion and time-consuming searches. This complex web of information not only consumes time but also overwhelms users seeking reliable guidance for dermatological concerns.

With the rapid development of artificial intelligence technology, deep learning has rapidly developed computer vision. Dermatological medical image processing has become an important part of image processing, machine science, intelligent medicine and other intersecting fields, and has been highly concerned. Various studies have shown that deep learning methods are able to outperform humans in many computer vision tasks. One of the things behind the success of deep learning is its ability to automatically learn semantic features from massive data sets. In particular, there has been a lot of work on applying deep learning methods to dermatological diagnostic.

Liao[1] built a universal dermatological diagnostic system using pre-trained VGG-16, VGG-19, and GoogLeNet networks. In the recent work of Liao et al.[2], the authors utilized the pre-trained AlexNet to perform the task of classifying diseases and lesions. They noted that lesion type labeling should also be considered a target for automated diagnostic systems so that the system can achieve high precision in describing skin lesions. A multi-task deep neural network is proposed by Kawahara et al.[3] to be trained on a multimodal dataset, including clinical and dermoscopic images and patient metadata, to classify the 7-point melanoma checklist criteria and perform skin lesion diagnosis. A network trained with multiple multitasking loss functions is capable of handling combinations of input patterns. The model classifies the 7-point checklist and diagnoses skin conditions, and generated multimodal feature vectors suitable for image retrieval and clinical discriminant region localization.

Based on the current situation of dermatological diagonsis and literature review, we proposed the Image-Text Based Dermatological Diagnosis Helper. The platform adopts the "CLIP" model to realize the function of dermatological diagnostic by disease pictures or text descriptions. At the same time, we use vector databases FAISS and Relational Database. Stores information such as disease image embedding, patient information(id, age, sex, phone number etc.) Information such as drug store can help people get the corresponding diagnosis and medication guidance at home by uploading their skin disease photos or providing corresponding symptom descriptions. At the same time people can order the needed medicine from the provided web page.

Given the high prevalence of skin diseases, the platform will benefit everyone. Whether in areas with inadequate medical resources, or because of busy work, people can obtain timely treatment plans and self-treatment, saving time and money, while effectively reducing the anxiety caused by not being diagnosed. The platform will offer its services free of charge to better serve users. At the same time, we designed a user-friendly interactive platform that is not only beautiful but also easy to operate.

2 System Architecture

In essence, our database system is a comprehensive integration of diagnostic and drug purchasing functionalities. It comprises two primary components: a **diagnosis system** and a **drug ordering system**. The diagnosis system is fundamentally a vector database, and in contrast, the drug ordering system operates as a relational database. This dual-component structure allows for a cohesive and efficient platform that seamlessly combines diagnostic capabilities with the convenience of purchasing prescribed medications.

As previously highlighted, our database was meticulously crafted to tackle the challenges associated with human reliance on visual inspection for diagnosing skin diseases and the inefficiencies of online searches. To utilize the system, users simply upload a photo of the affected skin area. Subsequently, the vector database of the drug ordering system encodes the image, facilitating the retrieval of the most closely matched image along with its corresponding diagnosis and treatment plan (inclusive of the recommended medication) stored in the database. Users can promptly proceed to order the prescribed medication through diagnosis system, thereby achieving a streamlined process for both diagnosis and drug treatment.

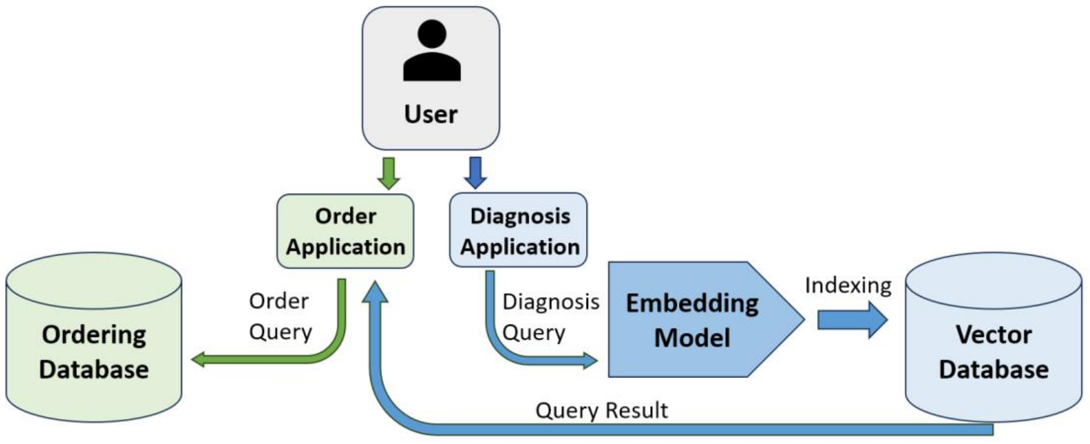


Figure 2 System Overview

2.1 Diagnosis System

Traditional databases are structured using tables that contain structural information. AI tools, such as text embedding or image encoding, produce high-dimensional vectors, offering greater power and flexibility compared to fixed symbolic representations. However, conventional databases designed for SQL queries are not well-suited to handle these new representations. The massive influx of multimedia items results in billions of vectors. Identifying similar entries is inefficient with standard query languages. So, we designed a vector database.

Firstly, we employ the embedding model to generate vector embeddings for the images and texts earmarked for indexing. Secondly, these vector embeddings are incorporated into the vector database, with a reference back to the original content from which the embedding originated. Finally, when receiving a query from the application, we utilize the same embedding model to create embeddings for the query itself. These query embeddings are then employed to search the database for vector embeddings that closely align with the query. This method enables efficient retrieval of relevant content based on the similarity of vector embeddings (as shown in the fig.3).

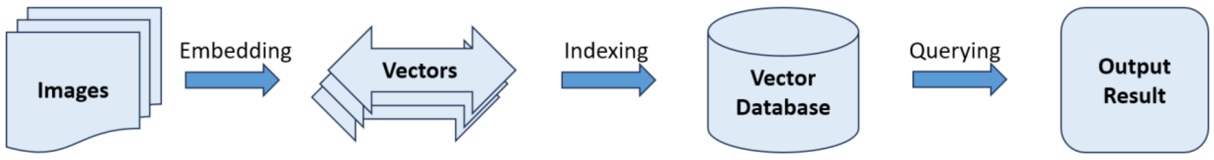


Figure 3 Vector Database Overview

2.1.1 Embedding

Our system utilizes the CLIP (Contrastive Language–Image Pre-training) to encode the images and texts into vectors. We encoded a dataset containing both images and texts as the training data in our database, which serves for the following storing and retrieving task.

2.1.2 Compression and Indexing

After we finished the embedding works, we constructed the vector database based on FAISS. The main algorithm is Product Quantization (PQ). PQ splits a high-dimensional vector into multiple sub-vectors, each compressed into a single number, thereby representing the original high-dimensional vector using a reduced set of numbers. The process involves decomposing the $N \times D$ dimensional vector $\boldsymbol{X}$ into M groups of D/M dimensional sub-vectors. Then, we cluster each group using k-means, resulting in K mapping results (codebooks) for each group. These mapping results can be either clustering center vectors or category IDs. The original vectors are then represented by the Cartesian product of these codebooks.

To optimize storage space, each group of sub-vectors typically has a codebook of size 256. The conventional approach involves using the clustering center vector as the mapping result for the codebook, maintaining the same storage space. In this scenario, each vector in every group of sub-vectors is represented by the center of the corresponding class, and each class ID requires only 8 bits ($log_2 256$) for storage. Consequently, the original 1024-dimensional floating-point (32-bit) vector can be compressed into eight 8-bit integers.

To efficiently process queries involving millions of vectors, our system employed an inverted index for accelerated query performance. The algorithm initially clusters N vectors into K' classes using k-means, with each class represented by a cluster center vector. Subsequently, the algorithm computes the residual vector of the original vector in relation to its corresponding cluster center vector. This residual vector is then encoded using the PQ mentioned earlier, and the encoded result is linked to the inverted chain of class IDs associated with the location of the original vector.

2.1.2 Similarity Retrieve

Given a set of vectors $\{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\}$ in d-dimension, we constructed a data structure in RAM based on FAISS. Once this structure is established, when provided with a new vector $\boldsymbol{x}$ in dimension d, it efficiently performs the operation:

$$i = argmin_i \left|\left|x - x_i\right|\right|_2$$

We used symmetric distance computation (SDC) to compute the Euclidean distance between the query vector and the candidate vector. The query vectors and candidate vectors are represented using cluster centroids, respectively, and the distances between the vectors can be replaced by an approximation of the distances between the cluster center vectors.

$$\hat{d}(x, y) = d\big(q(x), q(y)\big) = \sqrt{\sum\nolimits_j d\left(q_j(x), q_j(y)\right)^2}$$

where $q_j(x)$ denotes the quantization of the j^{th} sub-vector of the vector x, i.e., the clustering center vector where the j^{th} sub-vector is located.

2.2 Drug Ordering System

This system is designed to help users buy medication conveniently after obtaining a diagnosis. The user can order for drug after obtaining diagnosis, which containing symptom and treatment. Our system integrates information of patients, diseases, drug orders, drug stores and inventory of drugs, which is a relational database.

2.2.1 Entities and Relationships

Patient Patient is the user of our database system. They have unique username and password for logging in. Their personal information contains phone number, age, sex and area. We assigned each user with a unique identity: *Patient_ID*.

Disease Disease (exactly dermatological) contains disease's name, symptom and relevant treatment. The treatments usually involve in therapeutic drug, sometimes therapeutic measures. Also, we assigned each disease with a unique identity: *Disease_ID*.

Drug_Store Drug_Store contains information of store's name and area. We assigned each drug store with a unique identity: *Store_ID*.

Drug_Order Drug_Order is placed by patient and sent to drug store. It contains the information of patient's ID, store's ID for references. And it saves its own information of drug's ID and according amount. Patients can only purchase one kind of drugs in a single order. We also assigned each disease with a unique identity: *Order_ID*.

Drug_Inventory This entity serves for maintaining the drug store's inventory amount of each kind of drugs. We assigned each drug store with a unique identity: *Drug_ID*.

Other relations and assumptions:

- A patient can be diagnosed multiple diseases, and a kind of disease can be diagnosed on multiple patients.
- One patient can place multiple orders, but one order can only be placed by one patient.
- One order can only be sent to one drug store, but each store can receive multiple orders.
- Each drug store can hold multiple drug inventory, and each drug inventory can be saved by multiple drug store.

2.2.2 Relational Schema

- **Patient** (Patient_ID^{PK}, Username, Password, Phone, Age, Sex, Area)
- **Disease** (Disease_ID^{PK}, Disease_Name, Symptom, Treatment)
- **Patient_With_Disease** (Patient_ID^{PK,FK1}, Disease_ID^{PK,FK2})
- **Drug_Order** (Order_ID^{PK}, Patient_ID^{FK1}, Store_ID^{FK2}, Drug_ID, Amount)
- **Drug_Store** (Store_ID^{PK}, Store_Name, Store_Area)
- **Store_With_Drug** (Store_ID^{PK,FK1}, Drug_ID^{PK,FK2})
- **Drug_Inventory** (Drug_ID^{PK}, Inventory)

2.2.3 ER Diagram

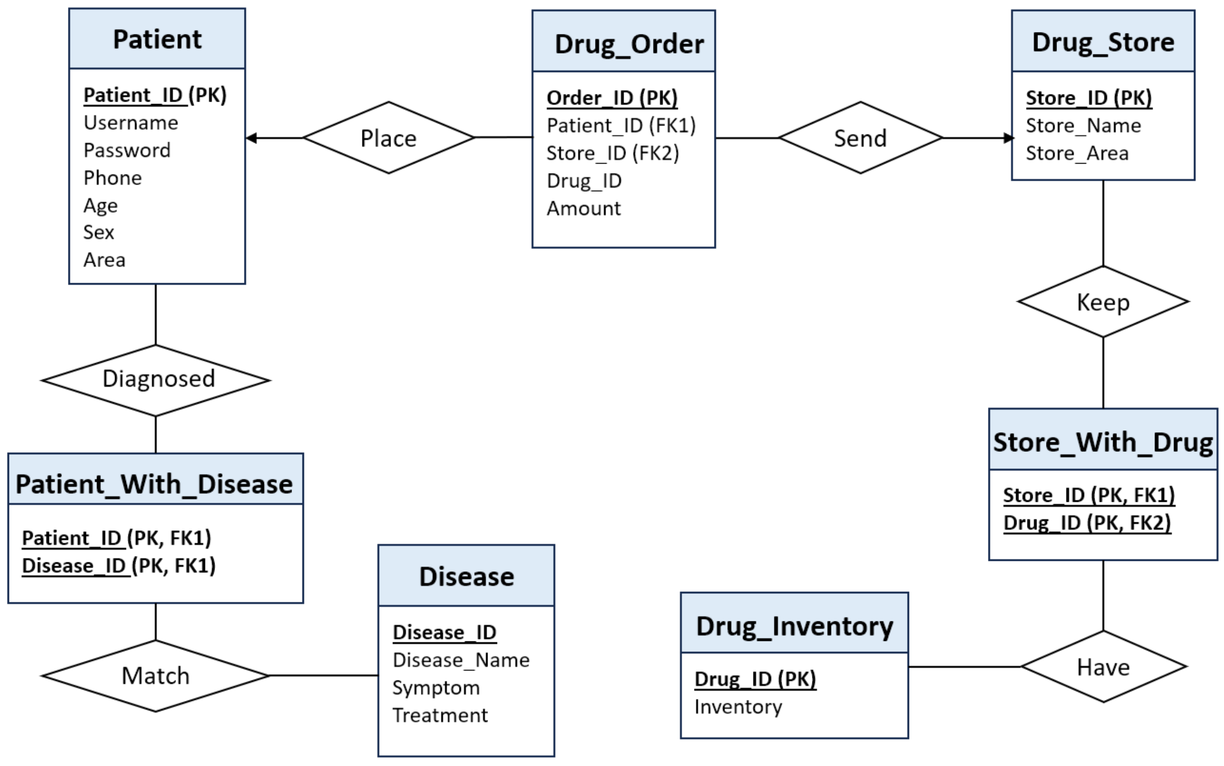


Figure 4 ER Diagram

3 Implementation and Testing

3.1 Data Collection and Generation

We utilized the Dermnet dataset as our primary resource for skin disease information, which encompasses a wide array of common skin conditions. This comprehensive collection has been integral to the training process of our embedding model. For detailed descriptions of each disease and their corresponding treatments, we source information from the American Academy of Dermatology (AAD) website. The AAD site provides extensive insights into various skin conditions, including their types, treatments, skincare tips, and preventive measures.

To simulate patient data, we employ the Python's *random* package. Furthermore, our drugstore information is gathered from 11 drugstores located in proximity to our university. For each drugstore, we collect key details such as their names, phone numbers, and the range of skin disease medications they stock, primarily sourced from their Meituan pages. Since obtaining precise inventory numbers for these medications is challenging, we again utilize the Python's *random* package to approximate the stock levels of each drugstore. Similarly, the medicine orders made by different patients are also synthesized using Python to create a diverse and realistic dataset

3.2 Embedding and Index Building

For our system, we primarily utilize the pre-trained CLIP model 'ViT-B/32' as the foundation to process skin disease images. This model encodes each image into a multi-dimensional vector, which is then stored in our vector database.

```python
def compute_embeddings(img_dir, save_path, batch_size, num_workers):
    if not os.path.exists(os.path.dirname(save_path)):
        os.makedirs(os.path.dirname(save_path))
    device = "cuda" if torch.cuda.is_available() else "cpu"
    model, preprocess = clip.load('ViT-B/32', device)
    # Load the data
    dataset = ClipSearchDataset(img_dir = img_dir, preprocess = preprocess)
    dataloader = DataLoader(dataset, batch_size=batch_size, shuffle=False,
                            num_workers=num_workers)
    disease_name, img_path_list, embedding_list = [], [], []
    for img, img_path, dis in tqdm(dataloader):
        with torch.no_grad():
            features = model.encode_image(img.to(device))
            features /= features.norm(dim=-1, keepdim=True)
            embedding_list.extend(features.detach().cpu().numpy())
            img_path_list.extend(img_path)
            disease_name.extend(dis)
    result = {'img_path': img_path_list, 'embedding': embedding_list,
              'disease_name': disease_name}
    with open(save_path, 'wb') as f:
        pickle.dump(result, f, protocol=4)
```

Once all image embeddings are extracted, we proceed to store them in our FAISS database. In our implementation, an inverted flat index structure is used to organize these embeddings. To calculate the similarity between stored embeddings and new image vectors, we employ the inner product method, which effectively measures the cosine similarity.

```python
def create_faiss_index(embeddings_path, save_path):
    with open(embeddings_path, 'rb') as f:
        results = pickle.load(f)

    embeddings = np.array(results['embedding'], dtype=np.float32)

    index = faiss.index_factory(embeddings.shape[1], "IVF256,Flat",
                                faiss.METRIC_INNER_PRODUCT)
    index.add(embeddings)
    # save index
    faiss.write_index(index, save_path)
```

3.3 Functions and Testing

Our platform, named ***Image-Text Based Dermatological Diagnosis Helper***, is an interactive web platform developed using the Python package *Streamlit*. It comprises five key components: a **Log-in** Page for secure access, an **Image-Based Diagnosis** Block for analyzing skin conditions through images, a **Text-Based Diagnosis Block** for inputting symptoms in text form, a **Medicine Purchase** Block for ordering medications, and a **History Order** Block to review past orders

3.3.1 Log-in Page

The login page serves as the entry point to our web platform, where users can enter their Username and Password. Upon clicking the "Login" button, the system initiates a search using the provided Username to find the matching password. If the entered password is correct, users are granted access to the main pages of the platform. In case of an incorrect password, a warning is displayed, prompting users to re-enter their password. Here we give an example of the password retrieving and verification process:

```sql
SET @InputUsername = 'fgx';
SET @EnteredPassword = 'edr5gg';
SELECT Userpassword INTO @StoredPasswordHash
FROM Patient
WHERE Username COLLATE utf8mb4_0900_ai_ci = @InputUsername COLLATE utf8mb4_0900_ai_ci;
SELECT
    CASE WHEN @EnteredPassword COLLATE utf8mb4_unicode_ci = @StoredPasswordHash COLLAT
              utf8mb4_unicode_ci
        THEN 'Login successful! Redirecting to main pages...'
        ELSE 'Incorrect password. Please re-enter your password.'
    END AS Result;
```

3.3.2 Image Based Diagnosis

In the Image-Based Diagnosis block, users can upload an image of their skin condition. The system then processes this image to provide a diagnosis, along with a detailed description of the symptoms and potential treatments. This process involves encoding the uploaded image into a normalized vector using a pre-trained model. Subsequently, the FAISS database computes the cosine similarity between this vector and all stored image embeddings, identifying the most similar image in our database. Using the index of this image, the system retrieves the associated disease name and its stored path.

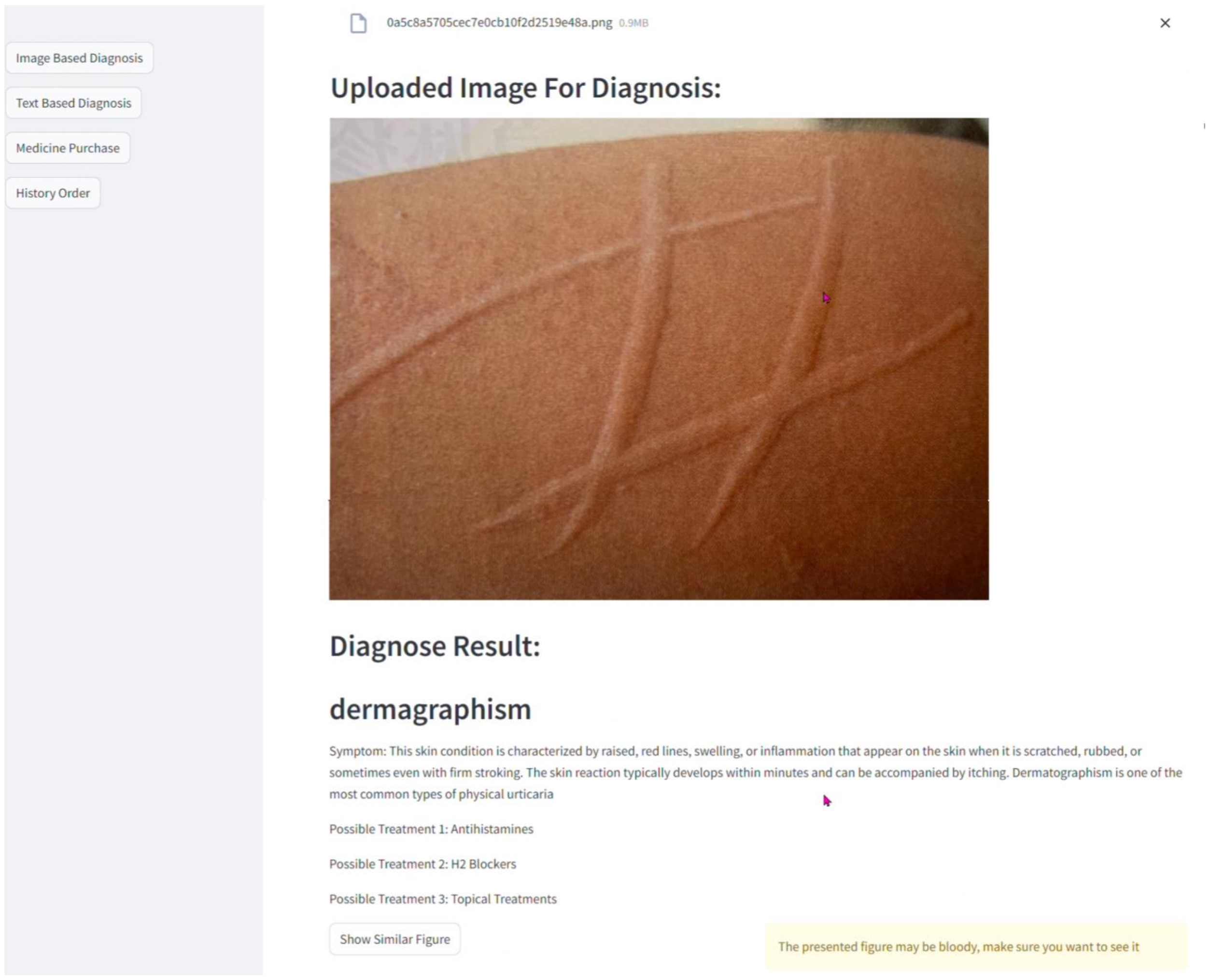
```python
img_tensor = preprocess(img).unsqueeze(0).to(device)
with torch.no_grad():
    features = model.encode_image(img_tensor.to(device))
features /= features.norm(dim=-1, keepdim=True)
embedding_query = features.detach().cpu().numpy().astype(np.float32)
D,I = index.search(embedding_query, num_search)
match_path_list = [embedding_path_list[i] for i in I[0]]

# get the name of the disease
show_disease = match_path_list[0].split('/')[-1].split('.')[0]
show_disease = '-'.join(show_disease.split('-')[:-1])
```

Furthermore, the system utilizes the identified disease name to fetch corresponding symptom descriptions and treatment options from our database through specific SQL operations:

```sql
SET @InputDiseaseName = 'piebaldism';
SELECT Symptom, Treatment
FROM Disease
WHERE Disease_Name COLLATE utf8mb4_0900_ai_ci = @InputDiseaseName COLLATE utf8mb4_0900_ai_ci;
```

Finally, the page displays the user's uploaded image alongside the diagnosed symptoms and suggested treatments. Given that some images in our database may be graphic, we provide an option for users to view these images at their discretion, accessible via a dedicated button.



Uploaded Image For Diagnosis:

Diagnose Result:

dermagraphism

Symptom: This skin condition is characterized by raised, red lines, swelling, or inflammation that appear on the skin when it is scratched, rubbed, or sometimes even with firm stroking. The skin reaction typically develops within minutes and can be accompanied by itching. Dermatographism is one of the most common types of physical urticaria

Possible Treatment 1: Antihistamines

Possible Treatment 2: H2 Blockers

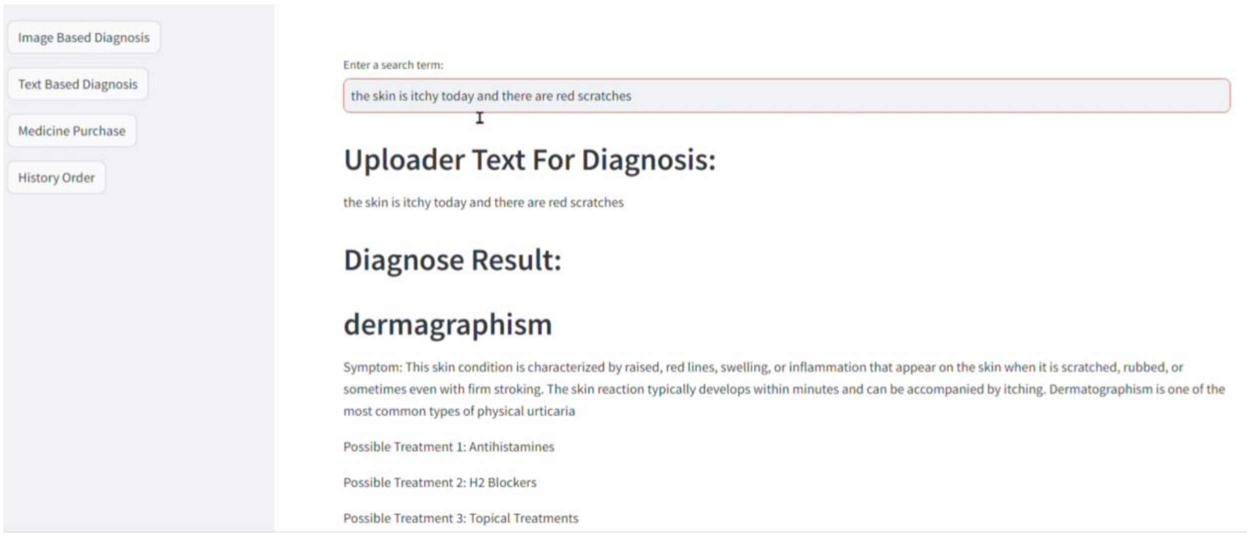
Possible Treatment 3: Topical Treatments

The presented figure may be bloody, make sure you want to see it

Figure 5 Image Diagnosis Interface

3.3.3 Text Based Diagnosis

In the Text-Based Diagnosis block, users have the option to describe their symptoms and observations regarding their current skin condition through text. Initially, the input text is tokenized by our pre-trained model, and subsequently encoded into embeddings similar to those used in the Image-Based Diagnosis block. After obtaining the text embedding, the system follows the same process as in the Image-Based Diagnosis block to provide a diagnosis and relevant information



Enter a search term:

the skin is itchy today and there are red scratches

Uploader Text For Diagnosis:

the skin is itchy today and there are red scratches

Diagnose Result:

dermagraphism

Symptom: This skin condition is characterized by raised, red lines, swelling, or inflammation that appear on the skin when it is scratched, rubbed, or sometimes even with firm stroking. The skin reaction typically develops within minutes and can be accompanied by itching. Dermatographism is one of the most common types of physical urticaria

Possible Treatment 1: Antihistamines

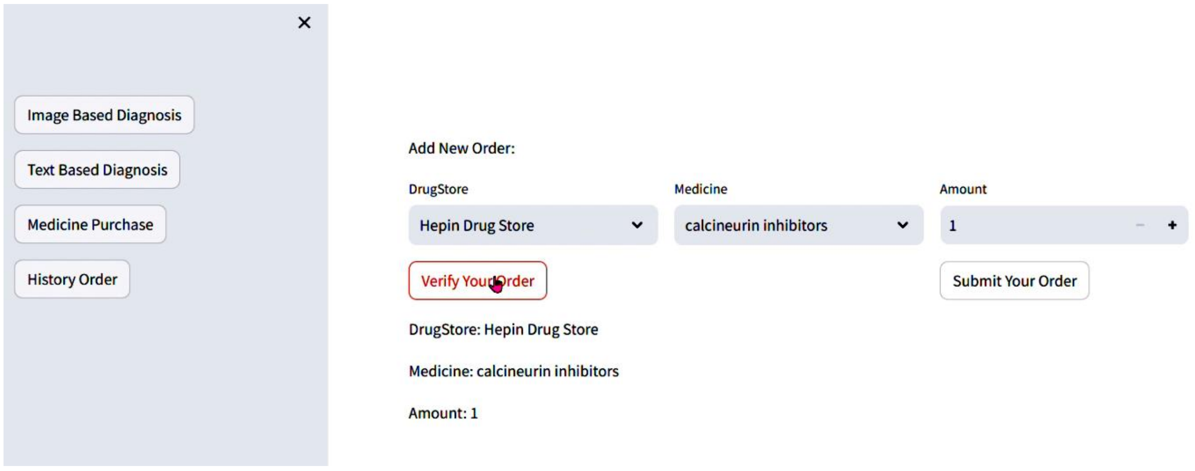
Possible Treatment 2: H2 Blockers

Possible Treatment 3: Topical Treatments

Figure 6 Text Diagnosis Interface

3.3.4 Medicine Purchase

In the Medicine Purchase block, users can select their preferred drugstore and medicine from a drop-down menu. After specifying the quantity of medicine they require, users can click the "Verify Your Order" button. This action prompts our system to use the selected drugstore and drug as keys to retrieve the current inventory amount of the chosen drug from that drugstore. If the desired quantity is available, the page will display the order details. Then, by clicking on the "Submit the Order" button, users can finalize their order, which the system will then record in our database.



Add New Order:

DrugStore: Hepin Drug Store

Medicine: calcineurin inhibitors

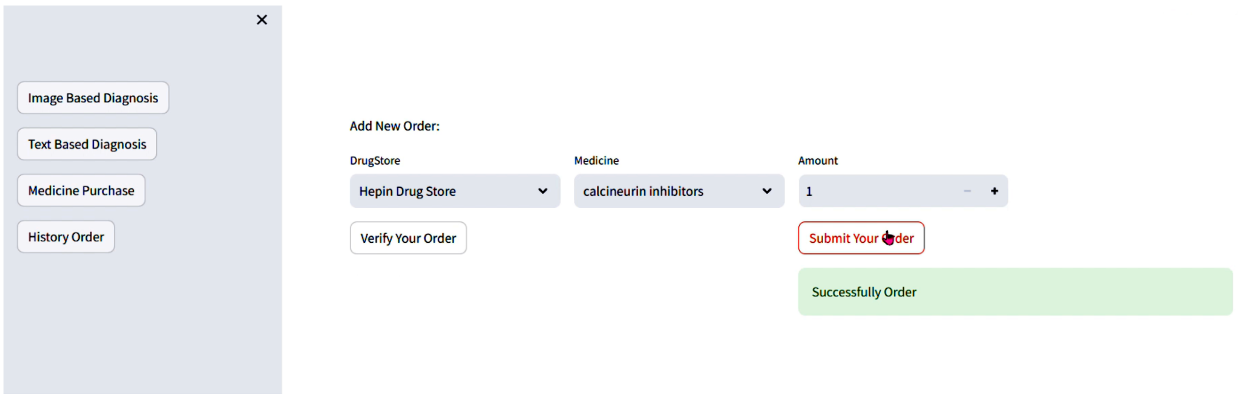
Amount: 1

DrugStore: Hepin Drug Store

Medicine: calcineurin inhibitors

Amount: 1

Figure 7 Verify function exhibition



Add New Order:

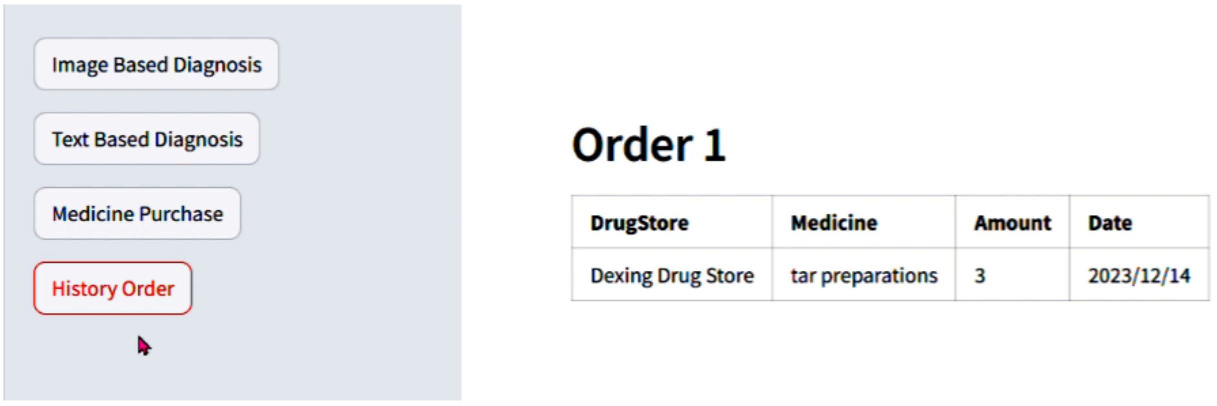
DrugStore	Medicine	Amount
Hepin Drug Store	calcineurin inhibitors	1

Successfully Order

Figure 8 Submit Order function exhibition

3.3.5 History Order

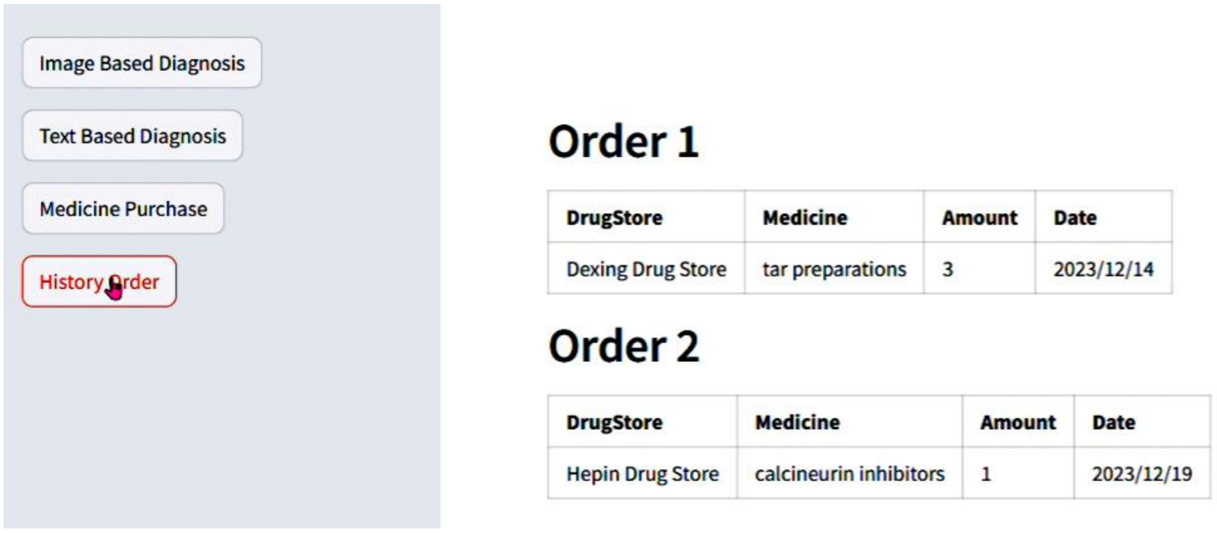
In the History Order block, our system retrieves a user's past order information from the database using their *Username* as a key. This order information includes details such as the drugstore name, the medicine ordered, the quantity, and the expected arrival date. Additionally, when a user places a new order from the Medicine Purchase block, this latest order will be immediately displayed at the top of the screen, allowing for easy tracking and reference.



Order 1

DrugStore	Medicine	Amount	Date
Dexing Drug Store	tar preparations	3	2023/12/14

Figure 9 History Order Interface (previous)



Order 1

DrugStore	Medicine	Amount	Date
Dexing Drug Store	tar preparations	3	2023/12/14

Order 2

DrugStore	Medicine	Amount	Date
Hepin Drug Store	calcineurin inhibitors	1	2023/12/19

Figure 10 History Order Interface (after purchase)

4 Conclusion and Future Work

4.1 Conclusion

In summary, our ***Image-Text Based Dermatological Diagnosis Helper***, seamlessly integrates online diagnostics with medication ordering. On the one hand, we employ CLIP to convert images and texts into vectors. Subsequently, FAISS is utilized to compress, index, and retrieve the data, facilitating efficient diagnosis. On the other hand, we've implemented a relational database, allowing users to conveniently order medications following the diagnosis results. Through rigorous testing, our system has demonstrated a harmonious blend of accuracy and efficiency.

4.2 Limitations

- **Enhanced Diagnostic Capabilities** Explore advanced image and text analysis techniques to further improve diagnostic accuracy. Investigate the integration of additional machine learning models or algorithms to refine the diagnostic process.
- **Expand Data Sources** Consider incorporating a broader range of dermatological data sources to enhance the diversity and comprehensiveness of the diagnostic capabilities.
- **Security and Compliance** Conduct regular security audits to ensure the protection of sensitive medical information. Stay updated on healthcare data regulations and compliance standards to adapt the system accordingly.

4.3 Future Works

Given the limited computational resources, database size constraints, and other factors, our current focus is on skin diseases that can be conveniently represented through pictures. Despite these limitations, the system has demonstrated excellent performance. Looking ahead, we envision integrating our system with hospital medical systems. Hospitals possess more extensive disease data, specialized diagnostic methods, and treatment measures. In this collaborative effort, our system can contribute efficient storage methods and user-friendly access interfaces. This integration is anticipated to significantly enhance people's access to healthcare by combining the strengths of both systems. Users will benefit from the comprehensive medical expertise of hospitals, while our system provides a streamlined and accessible platform for efficient storage and retrieval of relevant data. This collaboration aims to create a synergistic approach, leveraging the strengths of each system to improve overall healthcare accessibility and quality.

By expanding the database size, our system can leverage the temporal and spatial advantages inherent in Cloud Databases. This allows seamless integration with electronic health records, offering a more comprehensive overview of the patient's electronic health record. Furthermore, this enhancement enables real-time collaboration with healthcare professionals, facilitating activities such as live consultations and the provision of additional guidance.

References

[1] H. Liao, *A deep learning approach to universal skin disease classification*, University of Rochester Department of Computer Science, CSC.

[2] H. Liao, Y. Li, J. Luo, *Skin disease classification versus skin lesion characterization: Achieving robust diagnosis using multi-label deep neural networks*, 2016 23rd International Conference on Pattern Recognition (ICPR) IEEE (2016), pp. 355-360.

[3] J. Kawahara, S. Daneshvar, G. Argenziano, G. Hamarneh, *Seven-point checklist and skin lesion classification using multitask multimodal neural nets*, IEEE J. Biomed. Health Inf., 23 (2) (2018), pp. 538-546.

[4] *2020 China Health Statistical Yearbook*. (2021, December 6). Chinese Government Statistic and Information Center. http://www.nhc.gov.cn/mohwsbwstjxxzx/tjzxtjcbw/tjsj_list.shtml

[5] Jeff Johnson, Matthijs Douze, Herve Jegou (Facebook), *Billion-scale similarity search with GPUs*. https://doi.org/10.48550/arXiv.1702.08734

[6] Alexey Dosovitskiy, et al., *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. https://doi.org/10.48550/arXiv.2010.11929