**Projects In Machine Learning**

**Assignment 5**

**PROJECT: AI-Powered Skin Disease Detection**

**Group 5 Members**

Abdul Quader Mohammed

Deepti More

Nagaraju Medaraboina

Praiselin Lydia Gladston

Shireesha Thyaranahalli Narayana

Sudharsan Tirumal

**Project Overview**

Develop an AI-powered system that uses machine learning to detect skin diseases from images provided by users. The system will analyze the images, provide a description of the potential disease, offer recommendations for nearby dermatologists, and suggest if the disease requires isolation or special care measures.

**ML Use Cases:**

1. Detection of Skin Disease Kind
2. Disease Progress Tracking
3. Isolation/Preventive Measures

**Use Case 1: Detection of Skin Disease Kind**

**1. Introduction:**

From 1990 to 2017 for all skin diseases, DALY rates increased by 8% to 971 per 100 000 (674-1319), YLD rates increased by 8% to 897 per 100 000 (616-1235), YLL rates increased by 4% to 74 per 100 000 (53-89), and death rates increased by 18% to 5 per 100 000 (3-6). By developing this use case, any user will be able to get an idea of what kind of disease/infection is affecting their skin. This aids in creating an awareness of potentially malignant diseases and can motivate the user to consult a doctor and to take care.

**2. Problem Statement:**

To develop a Machine Learning model that can detect skin disease kind by just a picture of any part of the body assumed to be affected. Train the model with maximum number of diseases that are possible to be identified through images and provide result on top 5 diseases that could be matched with the image provided. The result screen should display the probability of the image being matched with a specific disease type. The result will also include ‘normal-skin’ probability, which will indicate that the provided image might not really indicate a disease.

**3. Dataset:**

The dataset used for this project is a combination of two different datasets - HAM10000 from ISIC (International Skin Imaging Collaboration) challenge and free images available in DermNet. There were about 130 different disease images found in DermNet. Abiding to the terms and conditions of DermNet, we have scraped images from their image gallery, and not putting it to commercial use or exposing the data after any transformation made on them. Merging common diseases found in both the datasets and additional ones found in ISIC, we have collected 134 different disease classes, where each class has images taken from different parts of the body.

**About ISIC:**

The International Skin Imaging Collaboration (ISIC) is an academia and industry partnership designed to use digital skin imaging to help reduce skin cancer mortality. ISIC works to achieve its goals through the development and promotion of standards for digital skin imaging, and through engaging the dermatology and computer vision communities toward improved diagnostics.

**About DermNet:**

### The world’s leading free dermatology resource. We help thousands of people make informed, evidence-based decisions on how to care for skin conditions, by providing reliable information at the click of a button.

### **Dataset Links:**

ISIC - [ISIC Archive (isic-archive.com)](https://gallery.isic-archive.com/#!/topWithHeader/onlyHeaderTop/gallery?filter=%5B%5D)

HAM10000 - [Skin Disease Classification | DenseNet (kaggle.com)](https://www.kaggle.com/code/angelarentsi/skin-disease-classification-densenet/input)

DermNet - [DermNet Image Gallery](https://dermnetnz.org/image-library)

**Snippet of the Dataset:**

**A close-up of a person's skin

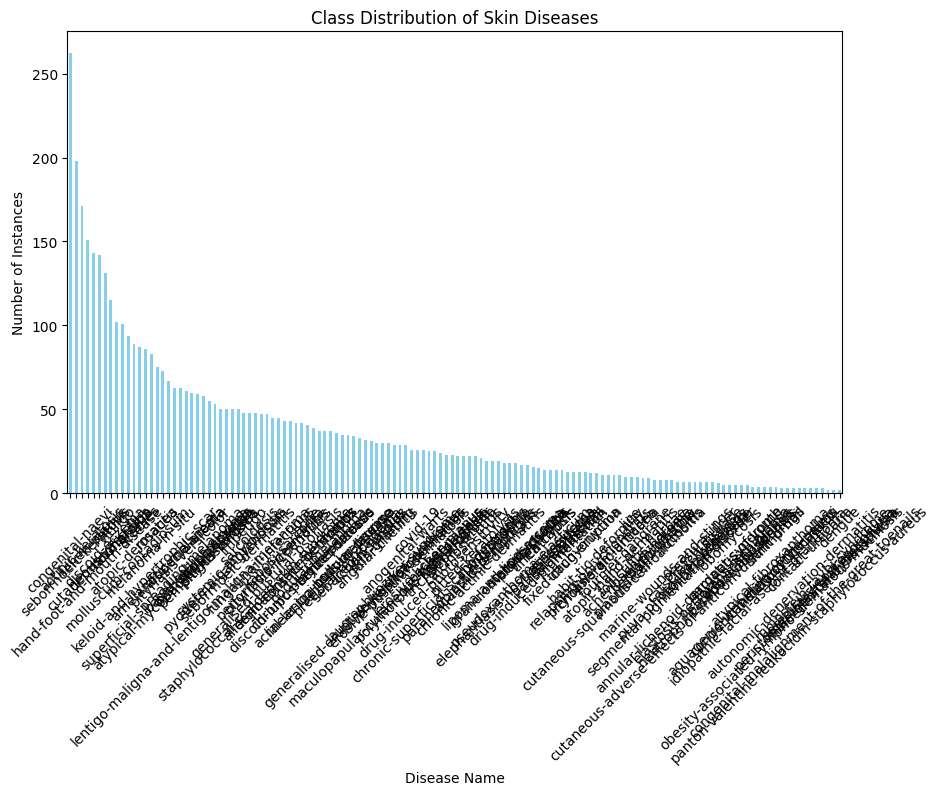
Description automatically generated**

**A screenshot of a computer

Description automatically generated**

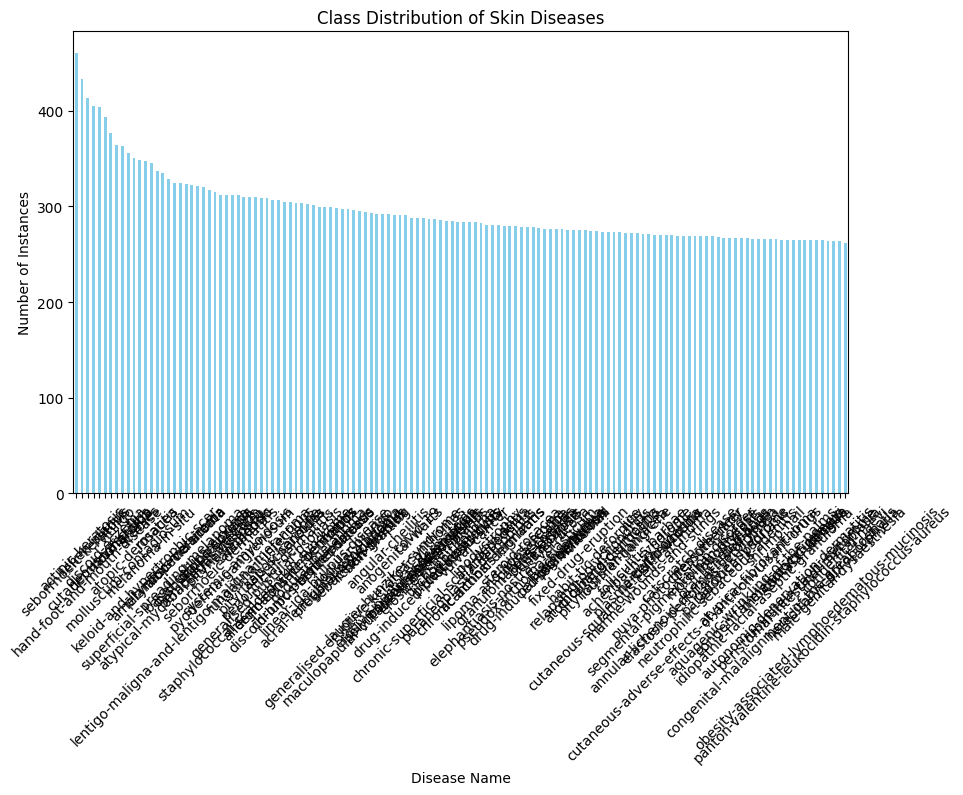
**Load the Dataset:**

The dataset is combined and formed a metadata with columns ‘disease\_name’, ‘body\_part’, ‘image\_path’ and ‘condition’. The column ‘body\_part’ refers to the part of the body of which the image is from. The ‘condition’ column mentions if the disease is Benign or Malignant.

**Dataset Class Imbalance:**

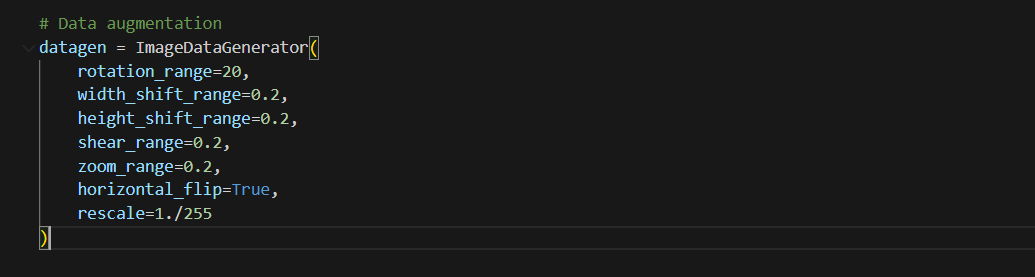
The distribution of dataset classes shows that it is skewed. There is a drastic imbalance of the number of image instances in disease classes. ‘Congenital-nevi’ has over 250 instances, but ‘panton-valentine-leukocidin-staphylococcus-aureus’ has only around 10 instances. When a dataset has an imbalanced distribution of classes, machine learning models tend to become biased towards the majority class. This happens because the learning algorithm tries to minimize overall error, leading to a situation where it favors the majority class simply because it's more prevalent.

To balance the dataset, we re-sample it. After re-sampling, the distribution is as follows:



**Data Augmentation:**

Since a Machine Learning model depends on the training dataset to classify the prediction dataset, it is crucial to maintain close similarity between the two. This fact emphasizes the importance of a proper dataset. Images should be of taken in a variety of environment to comprise that diverse dataset. Since there is a limited amount of realistic data available, we could leverage data augmentation technique to synthesize such data.

Data augmentation is a powerful technique used in image classification projects to enhance the performance and robustness of machine learning models. It involves creating new training data by applying various transformations to the original dataset. 

We have used **ImageDataGenerator** method to generate synthetic images. The parameters set to perform the image transformation operations is as given in the code snippet above.

**4. Model Development Methodology:**

In the context of skin disease analysis, the utilization of deep learning techniques has been notably advanced. In his study, Milton (2018)[1] at the Autonomous University of Barcelona presented a method for automated skin lesion classification using an ensemble of deep neural networks. This method was part of the ISIC 2018: Skin Lesion Analysis Towards Melanoma Detection Challenge, which aimed to enhance the accuracy and reliability of melanoma detection through advanced computational techniques. The study demonstrated significant advancements in the field, providing insights into how deep learning models can be effectively integrated to improve diagnostic outcomes in dermatology.

We have utilized the same concept and created an ensemble model that comprised of the pre-trained models InceptionResNetV2, DenseNet201 and InceptionV3.

***How are these models relevant?***

1. **InceptionResNetV2:**

Combines the inception module with residual connections. The inception module is designed to capture features at multiple scales, while residual connections help in mitigating the vanishing gradient problem and allow for deeper networks. Excellent at handling diverse feature extraction due to its multiscale processing and can capture complex features in images.

1. **DenseNet201:**

Features dense connectivity where each layer receives input from all previous layers and passes its own feature maps to all subsequent layers, enhancing gradient flow and reusing features. Highly efficient in parameter usage and mitigates vanishing gradient issues even in very deep networks. Known for extracting fine-grained features and capturing intricate details.

1. **InceptionV3:**

Utilizes the inception module but without the residual connections. It applies factorized convolutions to efficiently capture spatial information and reduce computational cost. Strong in capturing a wide variety of features due to its factorized convolutions and efficiency in computational resource usage.

We have used **Adam optimizer and categorical\_crossentropy loss function** for our ensemble model. The Adam optimizer dynamically adjusts learning rates for each parameter, enabling faster convergence and reduced need for manual tuning, which is crucial for efficiently training deep, complex networks. On the other hand, the categorical cross-entropy loss function is tailored for multi-class classification, providing a probabilistic framework that aligns with the **‘softmax’** outputs and penalizes incorrect predictions heavily. This combination ensures robust performance and efficient training, making it well-suited for the complex task of accurately classifying diverse skin disease images.

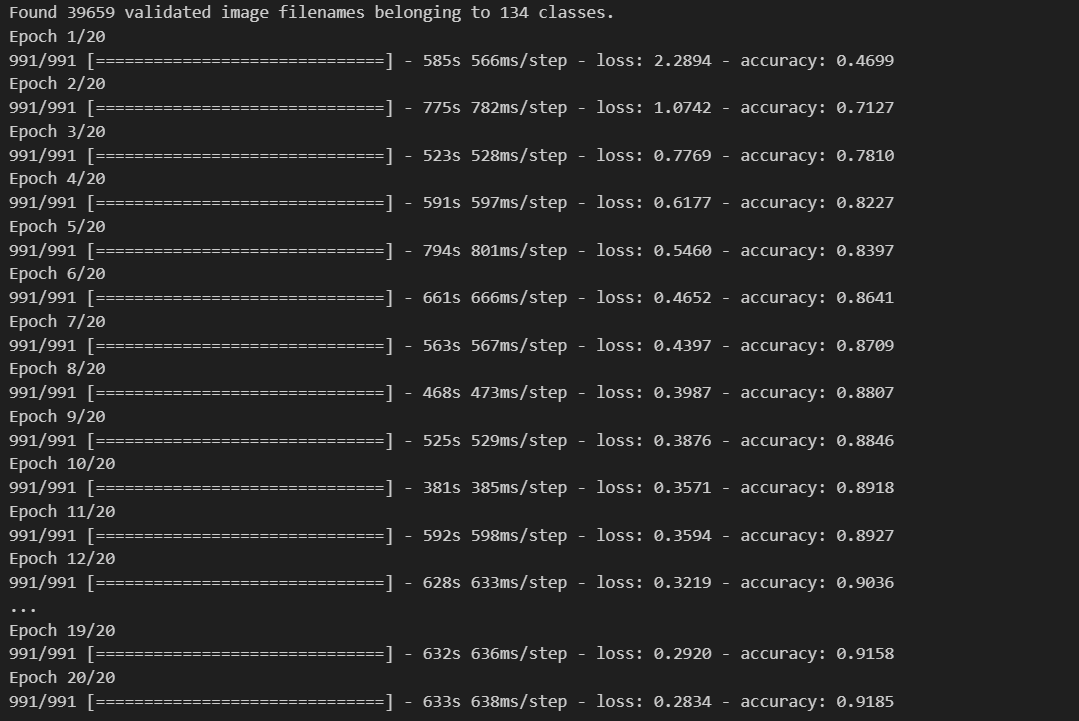
**Model Utilization:**

The trained model is being saved into a h5 file [2]. For Keras models, saving in a .h5 format can include the entire model (architecture, weights, and optimizer states), making it easy to resume training or deploy the model for inference. The format is designed to handle large datasets efficiently. It supports features like data compression and chunking, which helps in managing memory and storage resources effectively.

In our user interface, when an image is uploaded for classification, the model that was saved into a h5 file, is loaded using **load\_model()** function and use it for prediction. The image is passed into the loaded model and the prediction result provides us with the probability of matching with a disease class.

**5. Results:**

Model performance was verified using the metric “accuracy”. The training dataset showed an accuracy of 91.85% but during testing, the images show a maximum of 80% accuracy to the diseases it matches with.



**6. Discussion:**

Since model performance with the testing dataset was not up to expectations, there are some related suggested works to be made on the model to verify improvements:

1. Utilization of any other CNN [3] included between the model layers to test accuracy.
2. Try out other dataset sampling technique like SMOTE for better balancing of minority classes.

**8. References:**

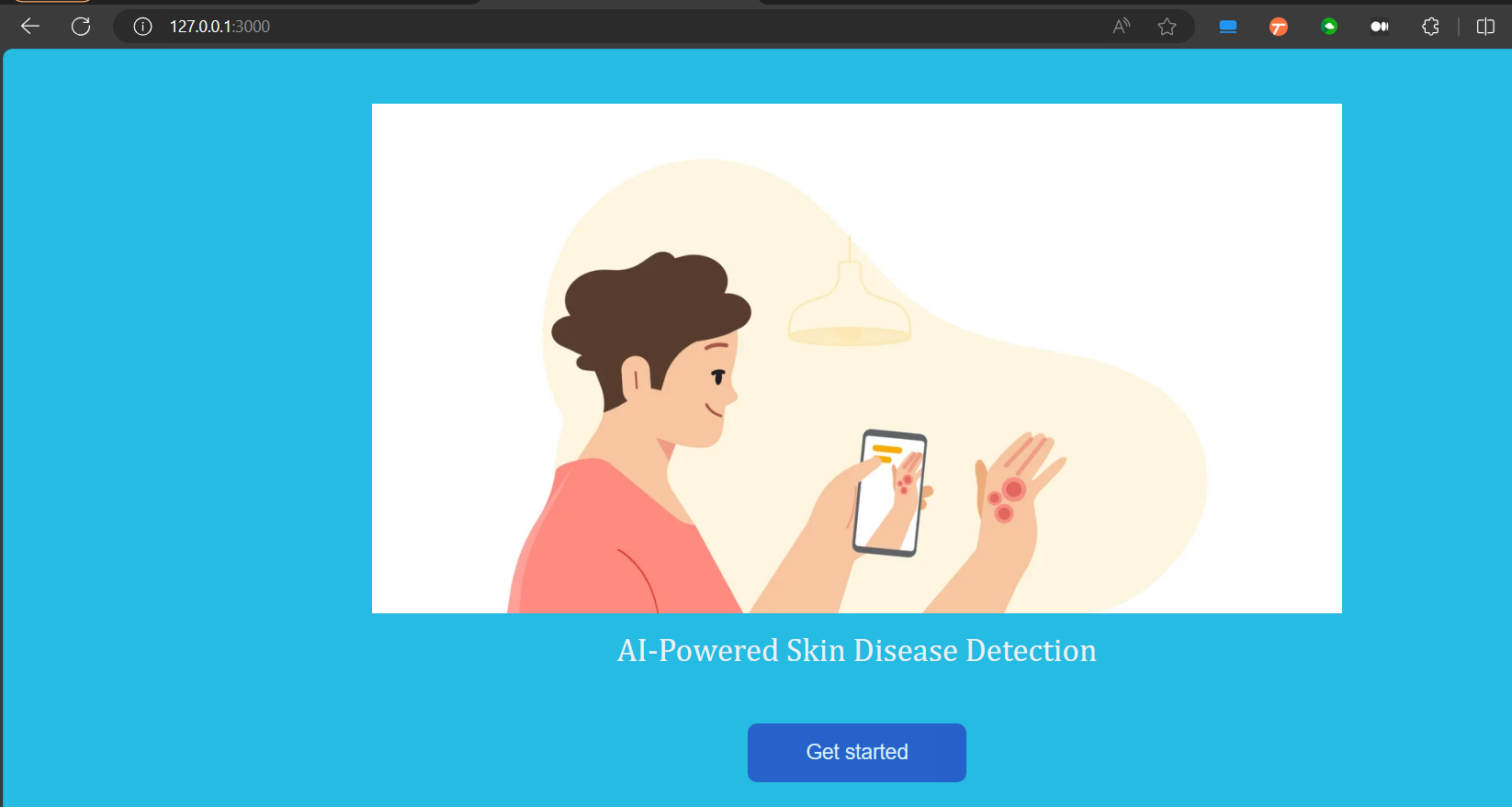
[1] Milton, M. A. A. (2018). *Automated skin lesion classification using ensemble of deep neural networks in ISIC 2018: Skin lesion analysis towards melanoma detection challenge*. Autonomous University of Barcelona.

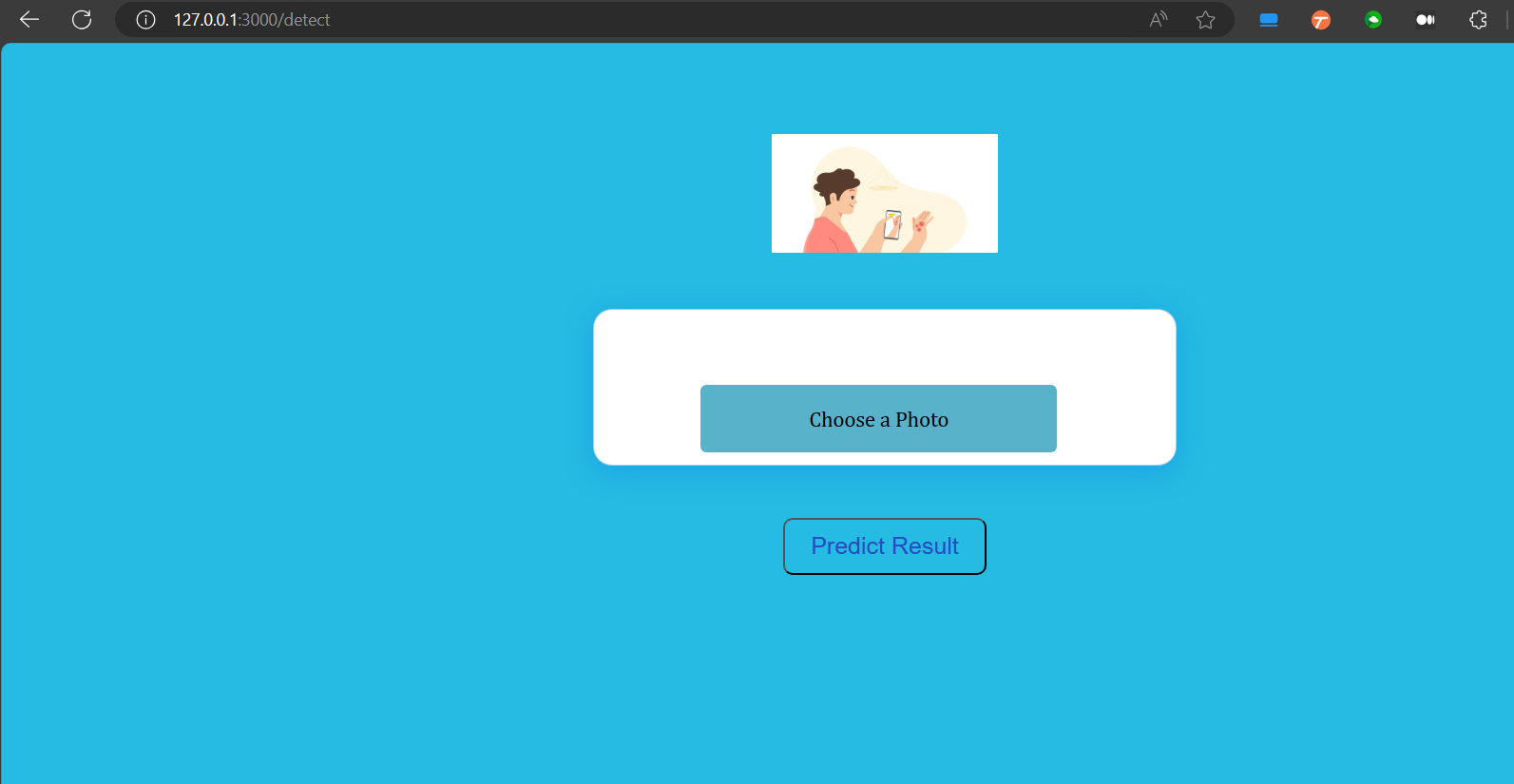
[2] Keras Team. (n.d.). *Model saving and loading*. Keras Documentation. Retrieved from<https://keras.io/2.16/api/models/model_saving_apis/model_saving_and_loading/>

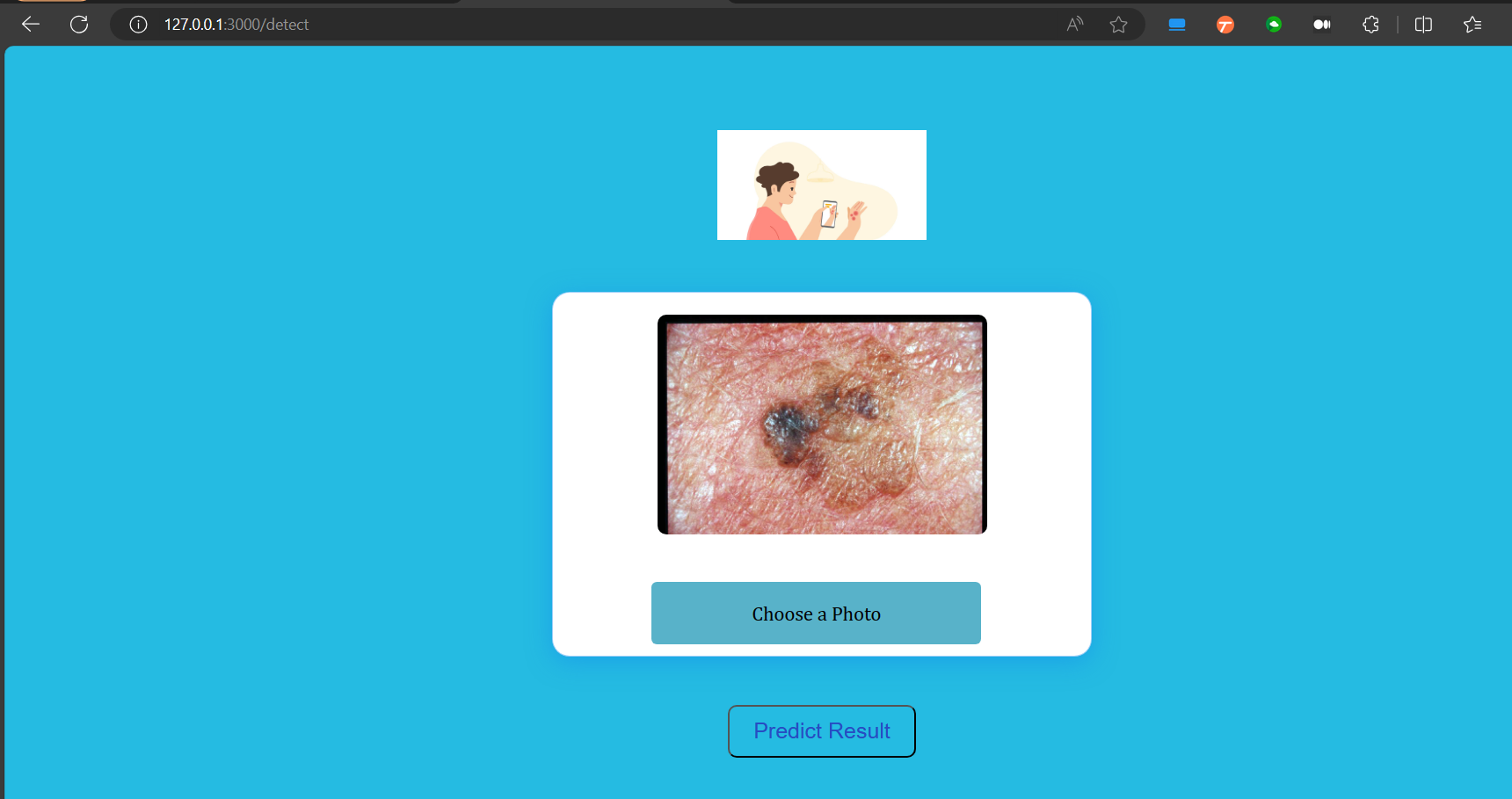
[3] Allugunti, V. R. (2022). A machine learning model for skin disease classification using convolution neural network. *International Journal of Computing, Programming and Database Management*, *3*(1), 141-147.

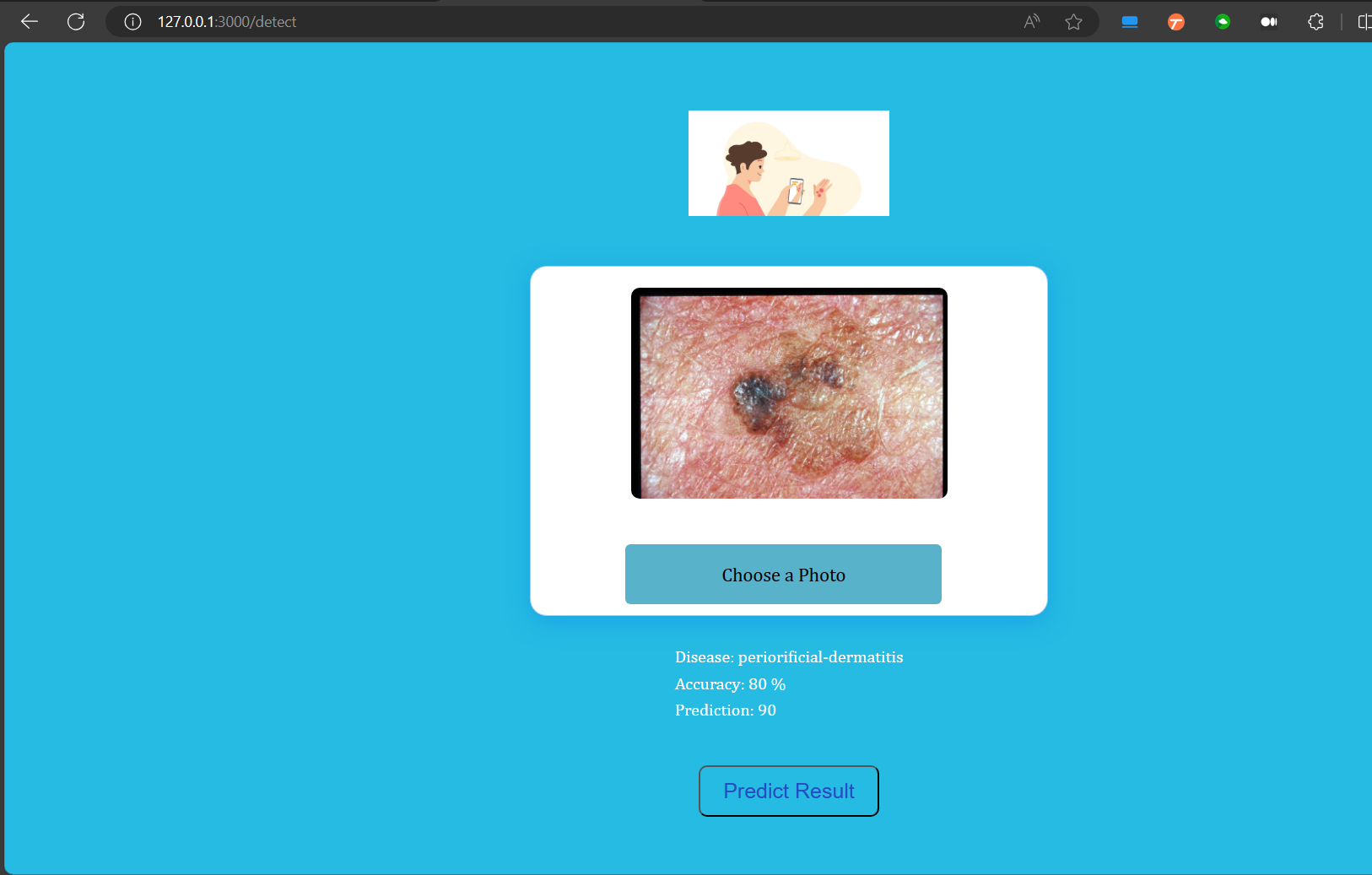
**UI Development:**

We are using a Flask application for skin disease detection. Users can upload an image, and the model will predict the disease name, accuracy, and prediction probability.









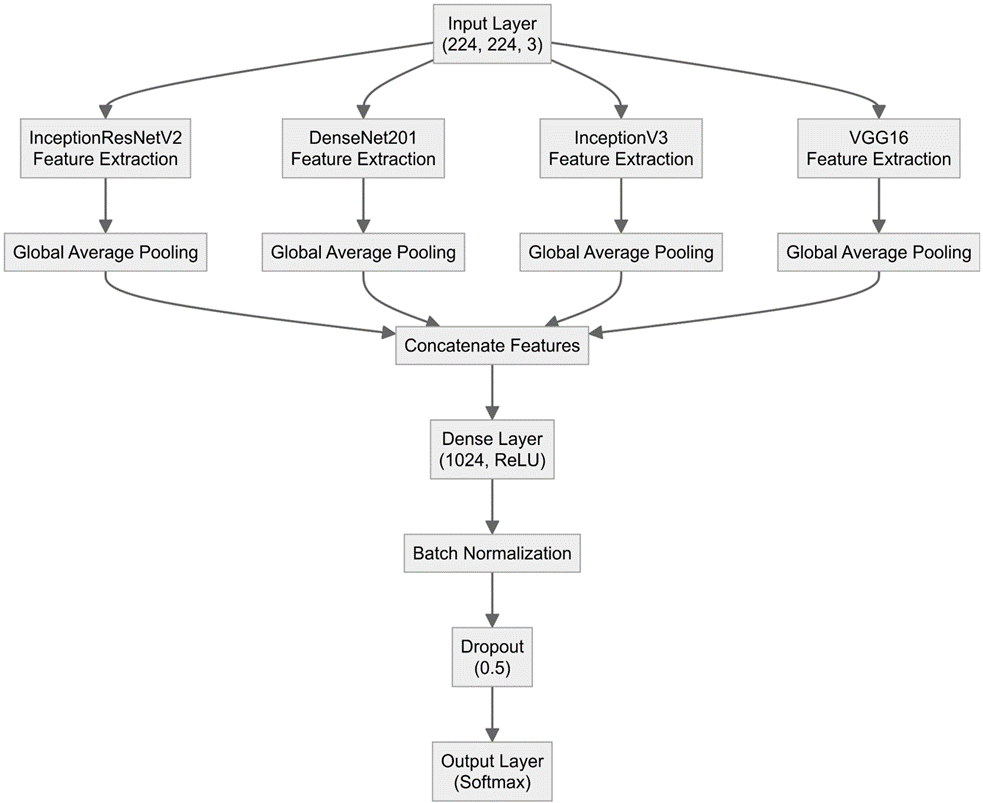
**Suggested Work for Use Case 1:**

Since validation results and test data had low accuracy, an additional neural network layer was suggested.

VGG16, known for its deep and straightforward structure, excels at capturing detailed hierarchical patterns in images, making it effective for feature extraction in skin disease detection.

Usually when used in an ensemble model, VGG16 enhances prediction accuracy and robustness, enabling reliable identification of subtle features in skin lesions.

Adding VGG16 to our final model layer with ‘Imagenet’ initial weights.

****

# 

# 

# **Performance Comparison:**

|  | **Without VGG16** | **With VGG16** |
| --- | --- | --- |
| **Training accuracy (15 epochs)** | **95.83%** | **95.13%** |
| **Validation Set** | | |
| **Loss** | **16.499** | **16.682** |
| **Accuracy** | **35.24%** | **35.29%** |
| **Precision** | **37.66%** | **37.34%** |
| **Recall** | **34.90%** | **34.96%** |

# **Further Improvement:**

Use SMOTE to balance minority classes to have a balanced dataset and re-verify performance.

# **REST API:**

**1.** Name: Model Testing

**2.** Description: Provide test input image to the trained model to get disease name.

**3**. HTTP Command: POST

**4**. Signature: [https://skin-det.com/detect/api=Skin-Disease-Kind&version=1.0.4&imagename=254895324-298644.jpeg&image=###](https://skin-det.com/detect/api=Skin-Disease-Kind&version=1.0.4&imagename=254895324-298644.jpeg&image=)

**5**. Responses:

1. Code 200 = Success. Response received with the test result.

2. Code 201 = Success. Request timed out.

3. Code 404 = Error. Request invalid and missing parameters.

4. Code 503 = Error. Service unavailable.

**6**. Model (JSON):

{

"api": "Skin-Disease-Kind",

"version": "1.0.4",

"imagename": "254895324-298644.jpeg",

"image": "###"

}

**7**. Example HTTPS Response (JSON):

{ "status": "ResultGenerated",

"test\_id": "1345645",

"message": {

"diseaseJson": { "diseaseNames": { "acral-lentiginous-melanoma",

"anogenital-warts",

"morbihan-disease" },

"predictions": { "0.7934",

"0.1276",

"0.0894" }

}

}

}

# 

# 

# 

# 

# **Use Case 2: Disease Progress Tracking**

**1. Introduction:**

Disease progress tracking is crucial for monitoring the progression of diseases over time. It helps in evaluating the effectiveness of treatments, adjusting medical interventions, and predicting future health outcomes. Machine Learning (ML) algorithms can analyze patterns in input images to assess the severity of a disease.

**Early Detection:** Regular monitoring enables early detection of complications, preventing severe outcomes.

**Treatment Adjustment:** If a disease worsens or doesn't improve, progress tracking provides evidence to change medication or escalate treatment plans.

**Patient Motivation:** Understanding their progress can motivate patients to adhere more closely to treatment regimens and lifestyle recommendations.

**Record Maintenance:** Tracking disease progression helps maintain a detailed record for future reference or consultation.

**2. Implementation:**

**Model: EfficientNet-v2**

EfficientNet-v2 uses a compound scaling method to optimize the network's depth, width, and resolution.This results in high accuracy and computational efficiency, making it suitable for medical image analysis.

**3. Dataset and Methodology:**

We used a dataset from Kaggle focused on acne severity classification.

The model was cross-validated in 5 folds, using different subsets of the dataset to ensure robustness and reliability.

The original dataset is split for the purpose of training and testing a machine learning model.

1. **Dataset Splitting:**

The entire dataset is divided into a number of equally sized subsets, known as folds. In 5-fold cross-validation, the dataset is split into 5 folds.

1. **Training and Testing Process:**

The model is trained and tested 5 times, each time using a different fold as the test set and the remaining folds as the training set.

For example:

In the first iteration, Fold 1 is used as the test set, and Folds 2, 3, 4, and 5 are used for training.

In the second iteration, Fold 2 is used as the test set, and Folds 1, 3, 4, and 5 are used for training.

This process continues until each fold has been used once as the test set.

1. **Performance Evaluation:**

The performance metric (e.g., accuracy, precision, recall) is calculated for each iteration.

The final performance score is obtained by averaging the results from all 5 iterations.

### **Dataset Link:**

<https://www.kaggle.com/datasets/lexuanhieu131297/acne-severity-classification>

**4. Data Preprocessing and Augmentation:**

**Library Used:** torchvision

**Transformations:**

* **Image Resize:** Adjusts the image size to a standard dimension.
* **Horizontal Flip:** Flips the image horizontally.
* **Vertical Flip:** Flips the image vertically.
* **Elastic Transform:** Applies elastic deformations to the image.
* **Rotation:** Rotates the image to a specified degree.

**Purpose:** These transformations are applied to perform data augmentation and standard preprocessing. Data augmentation increases the diversity of the training set, helping the model generalize better.

**5. Learning Rate Adjustment:**

**Validation Loss:** Based on the validation loss, the learning rate of the model is adjusted to find a balance between convergence speed and stability.

**Scheduler:** ReduceLROnPlateau

* **Function:** This learning rate scheduler reduces the learning rate when the model's performance has plateaued.
* **Benefit:** Lowering the learning rate prevents the model from overfitting by making too many small adjustments when they are no longer needed.

**6. Test dataset metrics:**

| **Fold** | **Accuracy** | **F1-Score** |
| --- | --- | --- |
| **0** | **63%** | **65%** |
| **1** | **93%** | **94%** |
| **2** | **95** | **96%** |
| **3** | **98%** | **98%** |
| **4** | **98%** | **99%** |

**7. Conclusion:**

Using the EfficientNet-v2 model and a comprehensive cross-validation process, our approach to skin condition progress tracking achieves high accuracy and robustness. By implementing various data augmentation techniques and carefully adjusting the learning rate, we ensure that the model generalizes well to new data and provides reliable assessments of disease severity. This can significantly enhance patient care by providing precise and efficient monitoring, ultimately leading to better treatment outcomes and patient adherence.

**Use Case 2 Implementation - WIP**

1. Implement user login to have a record of the date of the image input and the level of severity of the disease.

2. Have a local database to maintain the track records.

3. Using the model’s result (probability), we will compare the previous status of the disease and inform the user if the condition has progressed positively or negatively

**UI Development:**

The application utilizes a user-friendly interface developed using Flask (Restful API), a lightweight web framework for Python.

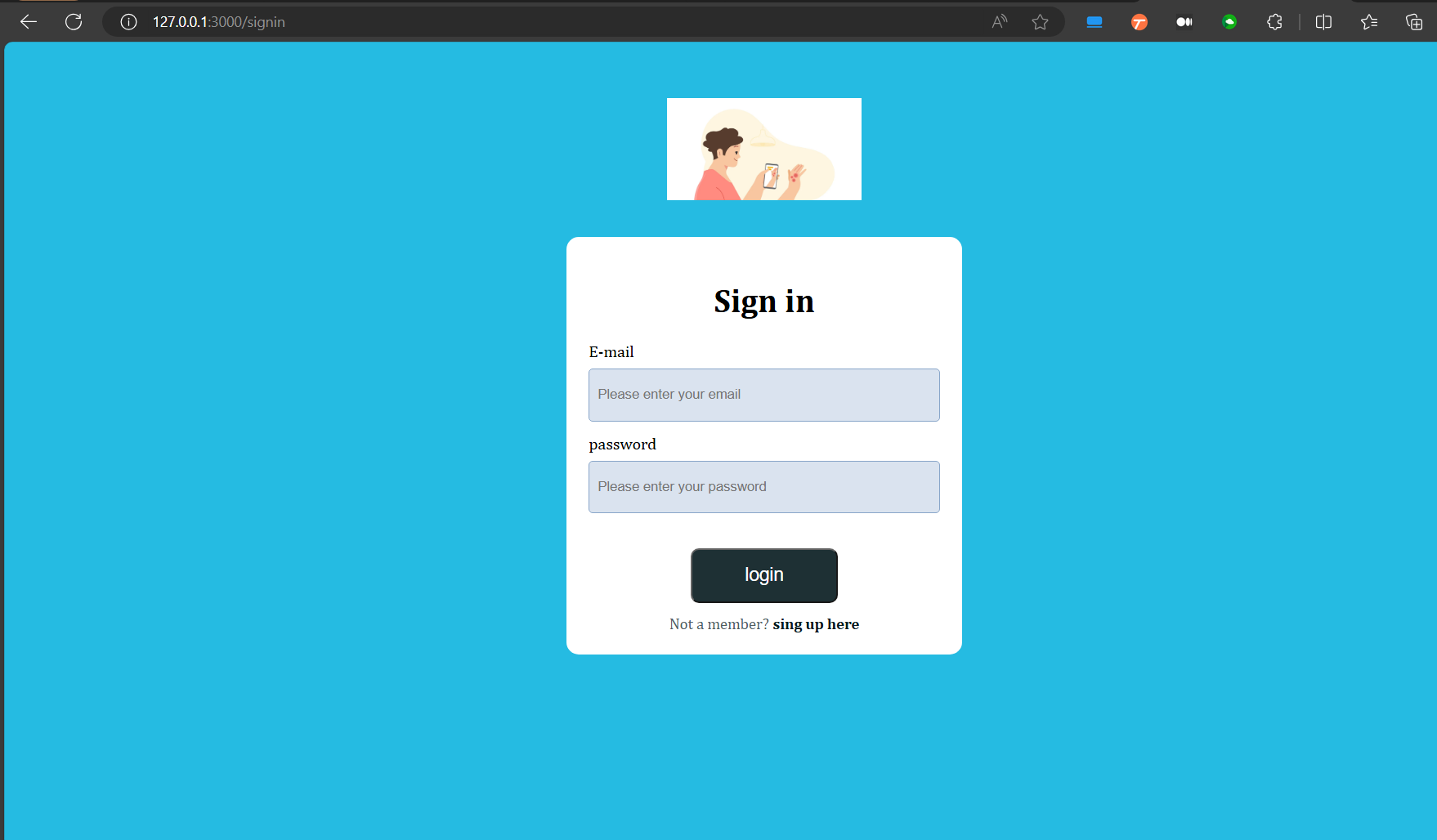
**User Interface Components:**

**Sign In Page (signin.html):**

* Allows existing users to sign in to their accounts.
* Includes input fields for email/username and password.

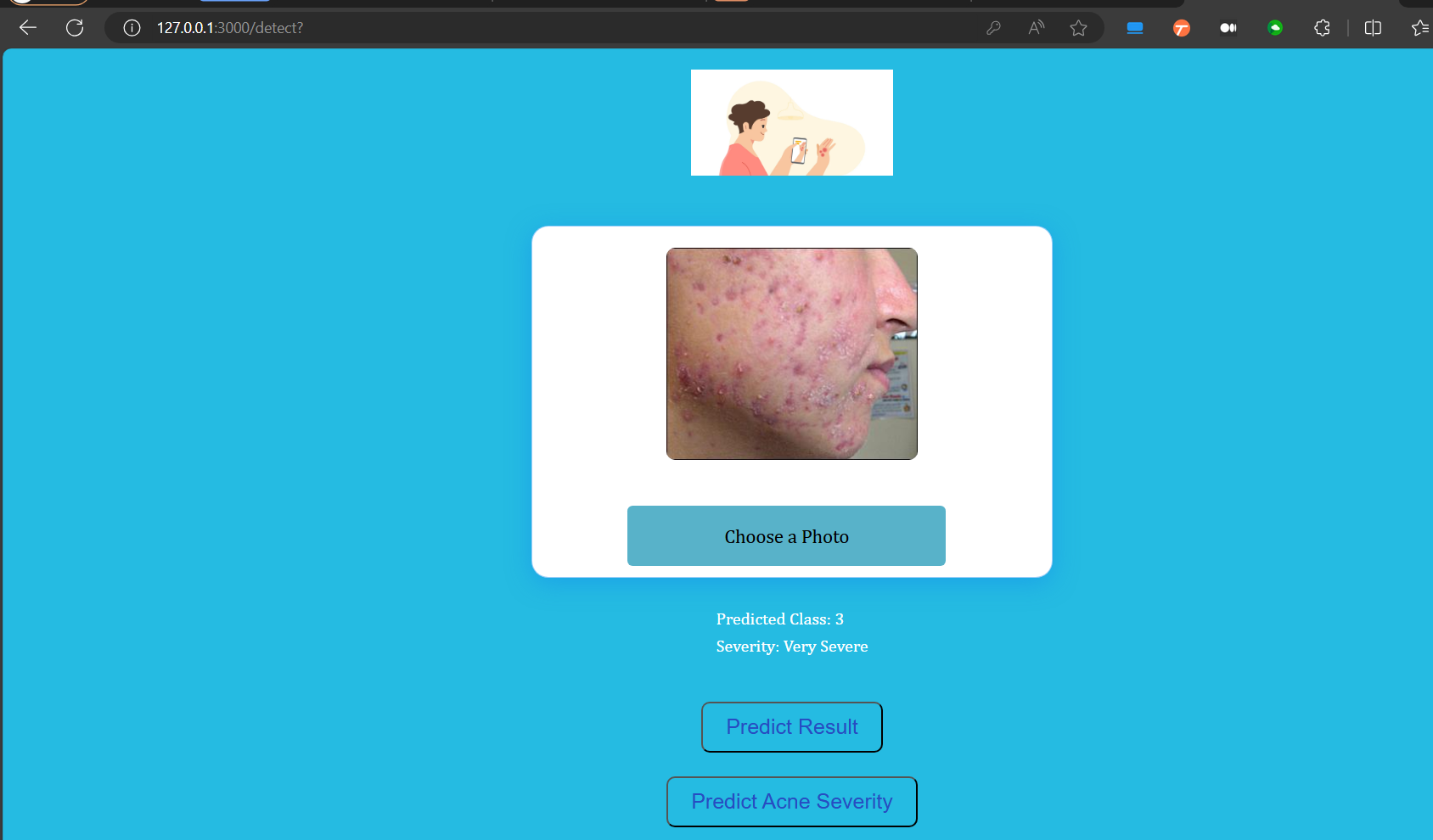
**Sign Up Page (signup.html):**

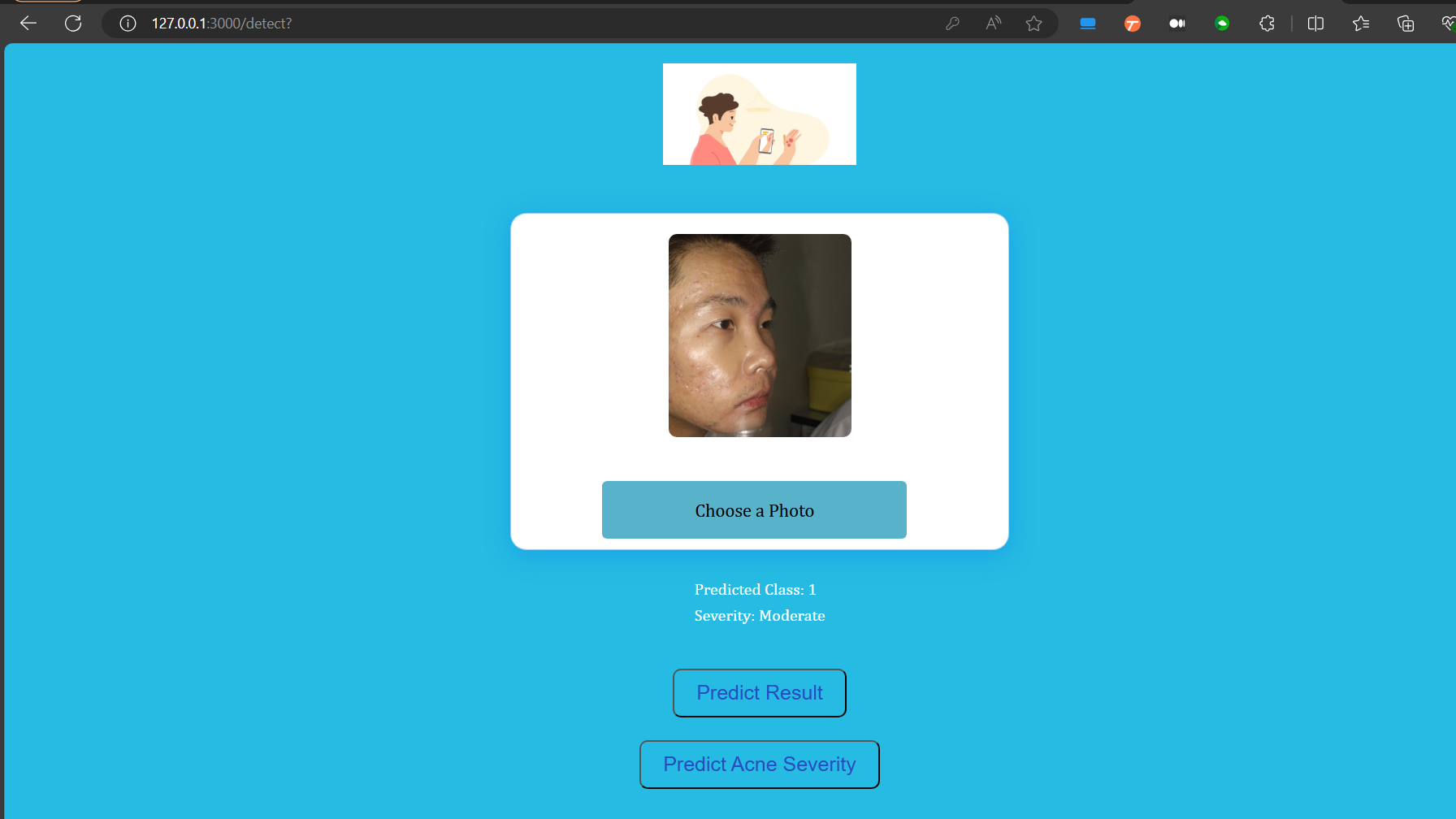
* Enables new users to create an account.
* Includes input fields for user information such as name, email, password, etc.

****

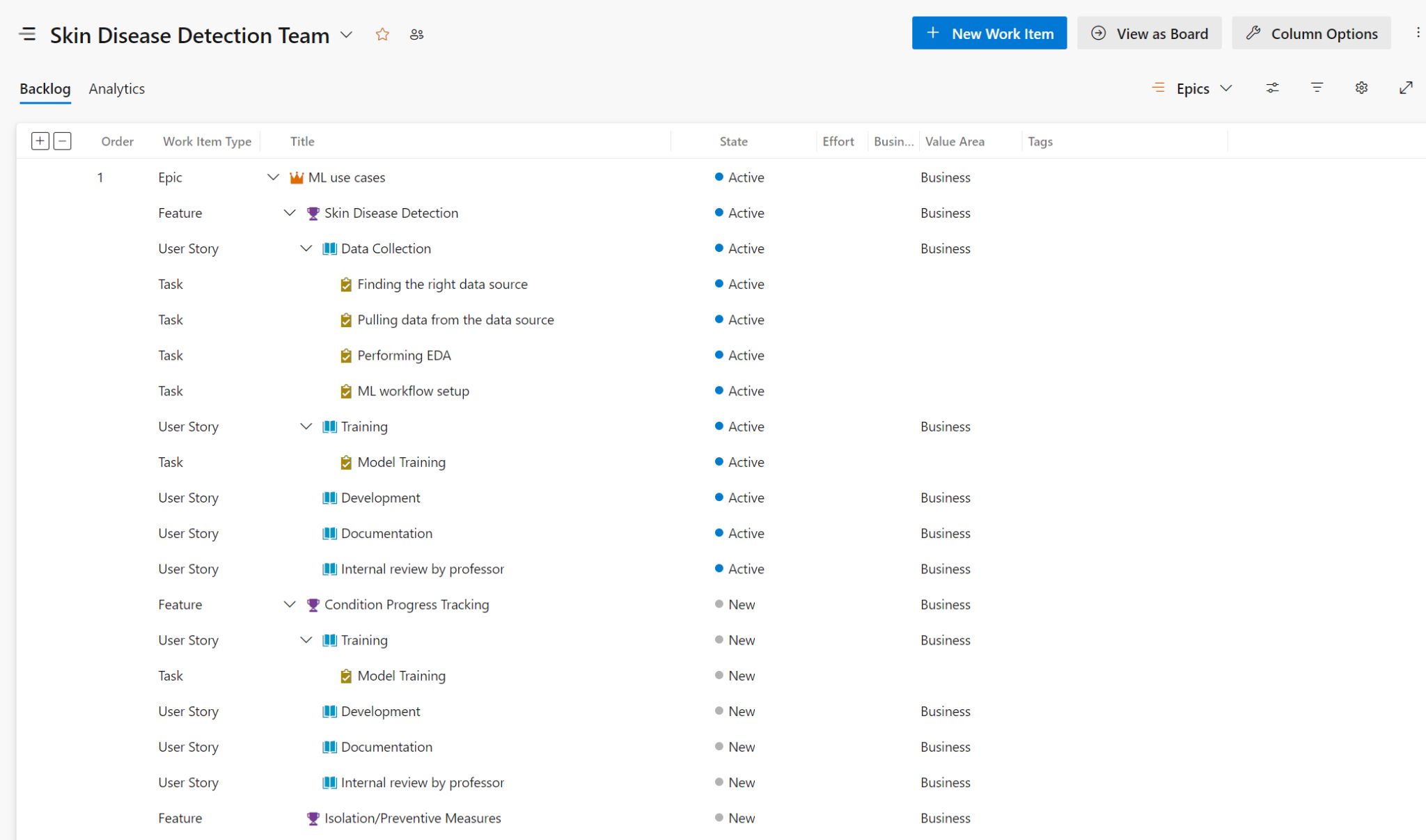
**Progress Tracking Page (detect.html reused):**

* Similar to the Disease Detection Page, users can upload images to track the severity of their acne over time.
* Displays the progress tracking results, providing insights into the changes in acne severity like moderate, severe or very Severe.
* We are maintaining a database to keep the record of patient progress.

****

****

**Azure DevOps:**

****

**Github Link:**

<https://github.com/shireeshatn/Skin-Disease-Detection>