

Project Report on Breast Ultrasound Image Classification

Introduction

This project develops a web-based application for classifying breast ultrasound images into benign, malignant, or normal categories, utilizing advanced machine learning techniques. The primary goal is to provide a tool that supports the early detection and preliminary screening of breast cancer, enhancing the diagnostic capabilities of healthcare professionals.

Objectives

- To implement a user-friendly web application for breast ultrasound image classification.
- To integrate a machine learning model that accurately classifies breast ultrasound images.
- To provide educational content on breast cancer and the importance of early detection.

Technologies Used

- **Flask** Flask, a Python web framework, is the backbone of our application, providing a lightweight and efficient way to handle web requests, manage user sessions, and render HTML templates dynamically.
- **HTML and CSS** The user interface is built using HTML5 and CSS3, ensuring that the application is visually appealing and accessible. The CSS is further enhanced with Bootstrap to make the interface responsive and navigable on various devices.
- **Bootstrap** Bootstrap's grid system and pre-built components were utilized to design a robust and responsive user interface, making the application accessible and easy to use across different screen sizes and devices.
- **Python** Python serves as the core programming language for handling backend operations, including image processing, model predictions, and data management.
- **Machine Learning** We trained a CNN model using the TensorFlow framework and integrated it into the website to analyze and classify ultrasound images. This model was chosen based on its high accuracy and efficiency on the validation set.
- **JavaScript** JavaScript and jQuery were used to enrich the interactive elements of the application, such as form validations, handling user interactions, and dynamic content updates.
- **Docker** Docker is used to containerize the Flask application, ensuring consistency across different environments and simplifying deployment. This allows the application to run in isolated environments with all dependencies included, avoiding conflicts and simplifying the deployment process.

Project Structure and Implementation

The application's architecture is structured around Flask's capabilities to serve dynamic content using a model-view-controller (MVC) pattern:

- **Base Template:** All other HTML templates extend from a base layout (`base.html`), which includes the common layout and navigation elements.

Project Report on Breast Ultrasound Image Classification

- **Welcome Page:** Provides an overview of the application and general information on breast cancer.
- **Disclaimer Page:** Ensures that users acknowledge the application's purpose as educational and not for diagnostic use.
- **Authentication Pages:** Secure login and signup pages that manage user access and profile creation.
- **Prediction Page:** Allows users to upload breast ultrasound images and submit them for classification.
- **Result Page:** Displays the uploaded image and result after image processing and classification, including the category of breast cancer.

Docker Integration

Docker was used to package the Flask application and its dependencies into a container. The process involves the following steps:

1. **Create a Dockerfile:** The `Dockerfile` specifies the base image, dependencies, and application setup.
2. **Build the Docker Image:** Run the following command in the directory containing your `Dockerfile` to build the Docker image.

```
docker build -t flaskapp .
```

3. **Run the Docker Container:** Use the following command to start a container from the built image.

```
docker run -p 5000:5000 flaskapp
```

This maps port 5000 on your host machine to port 5000 in the Docker container, allowing you to access the application at `http://localhost:5000`.

Running the Application Without Docker

If Docker is not available, you can run the application directly using Python. Follow these steps:

1. **Install Dependencies:** Ensure you have all required Python packages installed. You can use a virtual environment for this purpose. First, create a virtual environment and activate it:

```
python -m venv venv  
source venv/bin/activate # On Windows: venv\Scripts\activate
```

Then, install the dependencies listed in `requirements.txt`:

```
pip install -r requirements.txt
```

2. **Run the Flask Application:** Use the following command to start the Flask application:

Project Report on Breast Ultrasound Image Classification

```
python app.py
```

By default, Flask will run the application on `http://localhost:5000`.

Challenges and Solutions

- **Model Integration:** Integrating the deep learning model within the Flask app required optimizations to manage memory usage and ensure swift response times. Solutions included asynchronous processing and optimizing the model for faster inference.
- **Data Privacy:** Implementing encryption for user data and secure handling of image uploads were critical challenges. We utilized Flask's secure session management and HTTPS to protect data integrity and confidentiality.
- **User Experience:** Ensuring a seamless user experience despite the computational demands of image processing involved asynchronous JavaScript requests and server-side optimizations.
- **Dockerization:** Configuring Docker to properly build and run the application required ensuring all dependencies and files were included in the Docker image. Issues such as file paths and missing dependencies were addressed by refining the Dockerfile and verifying the container's setup.

Results and Discussion

The application was tested with a dataset of ultrasound images, showing an area under the curve (AUC) of 0.82. The tests included diverse cases to evaluate the model's robustness against various image qualities and tumor presentations.

Conclusion

The Breast Ultrasound Image Classification application demonstrates the effective use of deep learning in medical imaging, providing a tool that could potentially augment the diagnostic process. This project highlights the importance of integrating AI in healthcare to improve early detection rates and support clinical decision-making.

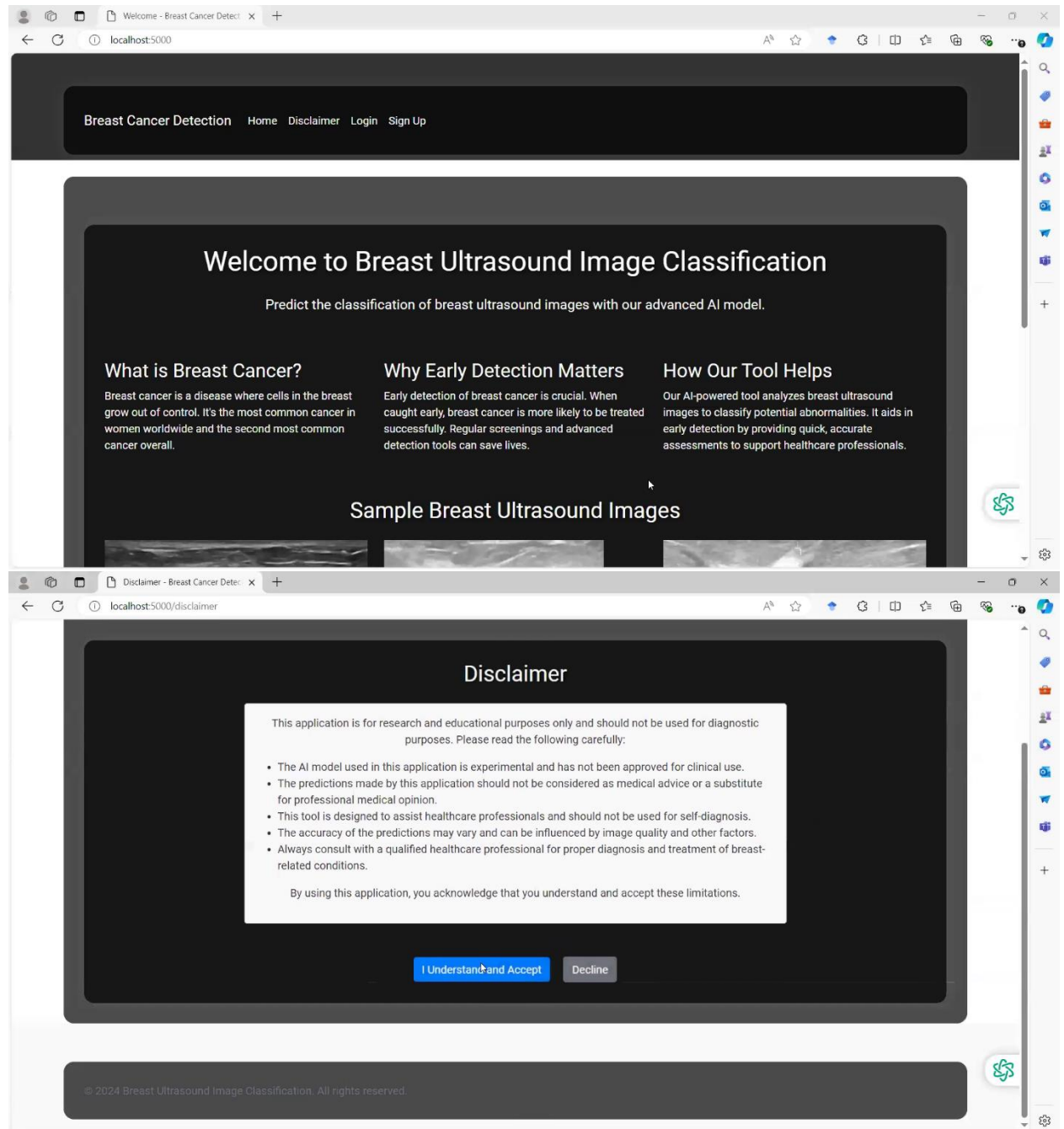
Future Work

- **Expansion to Multi-Class Detection:** Future iterations could expand the model to detect other forms of abnormalities.
- **Real-Time Analysis:** Implementing real-time image processing and classification to provide immediate feedback during ultrasound examinations.
- **Larger Dataset Training:** To improve the model's accuracy and generalizability, training on a more extensive and diverse dataset will be considered.

This report outlines the development and deployment of the Breast Ultrasound Image Classification application, including the use of Docker to streamline and standardize the deployment process.

Project Report on Breast Ultrasound Image Classification

Screenshots:



Project Report on Breast Ultrasound Image Classification

The image displays two screenshots of a web application interface for Breast Cancer Detection.

Top Screenshot: Sign Up Page

The browser address bar shows `localhost:5000/signup`. The page title is "Sign Up - Breast Cancer Detection". The form contains the following fields:

- Username:** `doe.Jane`
- First Name:** `Jane`
- Last Name:** `Doe`
- Email:** `jane@gmail.com`
- Password:** (Empty field)

Below the Password field, a message states: "Password must be at least 8 characters long, include one uppercase, one lowercase letter, and end with a number."

A blue "Sign Up" button is located below the Password field. Below the button, there is a link: "Already have an account? [Login](#)".

Bottom Screenshot: Login Page

The browser address bar shows `localhost:5000/login`. The page title is "Login - Breast Cancer Detection". The navigation bar at the top includes links: "Breast Cancer Detection", "Home", "Disclaimer", "Login", and "Sign Up".

The form contains the following fields:

- Username:** `Shahapratim`
- Password:** (Masked with dots)

A blue "Login" button is located below the Password field. Below the button, there is a link: "Don't have an account? [Sign Up](#)".

Project Report on Breast Ultrasound Image Classification

