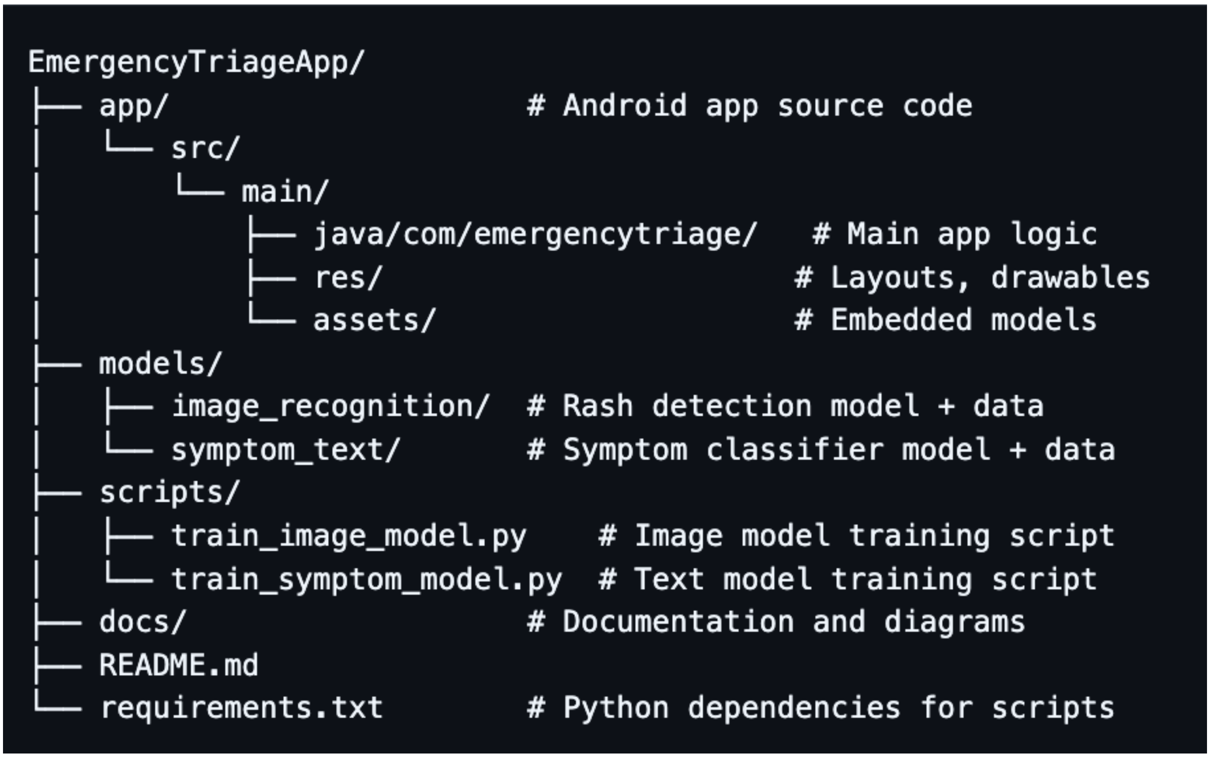
# 1.Introduction

EmergencyTriageApp is an Android application designed to assist in rapid medical triage by leveraging AI models for both image recognition (skin rashes) and text-based symptom analysis. The app aims to support healthcare workers and users in emergency situations by providing fast, reliable assessments and recommendations.

# 2.Project Structure



```
EmergencyTriageApp/
├── app/                    # Android app source code
│   └── src/
│       └── main/
│           ├── java/com/emergencytriage/   # Main app logic
│           ├── res/                       # Layouts, drawables
│           └── assets/                    # Embedded models
├── models/
│   ├── image_recognition/  # Rash detection model + data
│   └── symptom_text/       # Symptom classifier model + data
├── scripts/
│   ├── train_image_model.py    # Image model training script
│   └── train_symptom_model.py  # Text model training script
├── docs/                   # Documentation and diagrams
├── README.md
└── requirements.txt        # Python dependencies for scripts
```

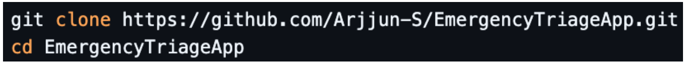
# 3.System Requirements

- **OS:** Windows, MacOS, or Linux
- **Android Studio:** Latest version recommended
- **Python:** 3.8 or newer (for model scripts)
- **Required Libraries:** TensorFlow or PyTorch, scikit-learn, pandas, etc.
- **Android Device:** API Level 21+ or emulator

# 4. Installation & Setup

## 4.1 Cloning the Repository

Open a terminal and run:



```
git clone https://github.com/Arjjun-S/EmergencyTriageApp.git
cd EmergencyTriageApp
```

### 4.2 Setting Up Android Studio

1. Open Android Studio.
2. Select **Open an existing project** and choose the EmergencyTriageApp directory.
3. Allow Gradle to sync and download dependencies.
4. Connect your Android device or start an emulator.

### ‘4.3 Running the App

- Click **Run** or **Debug** in Android Studio.
- The app will build and install on your selected device or emulator.
- Register or login as a user.
- Upload an image of a hand/rash or enter symptoms text for analysis.

## 5. Model Descriptions

### 5.1 Image Recognition Model

- **Purpose:** Recognize and classify skin rashes from images.
- **Architecture:** Deep CNN (e.g., MobileNetV2 or ResNet)
- **Input:** JPEG/PNG image of skin (usually hand)
- **Output:** Classification (normal, rash type X, etc.)

### 5.2 Symptoms Text Model

- **Purpose:** Analyze symptom descriptions and predict urgency or condition.
- **Architecture:** LSTM or BERT-based sequence classifier
- **Input:** Free text (e.g., "fever, headache, rash")
- **Output:** Urgency level, probable diagnosis

## 6. Custom Model Training

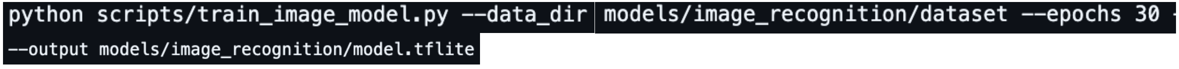
You can train your own models to improve accuracy or adapt to new data.

### 6.1 How to Train Image Model

1. **Prepare Data:**
    - Collect images labeled (e.g., "normal", "measles", "eczema").
    - Organize them in directories by class.
2. **Install Requirements:**

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

3. **Run Training Script:**



```
python scripts/train_image_model.py --data_dir models/image_recognition/dataset --epochs 30
--output models/image_recognition/model.tflite
```

- **--data_dir**: Path to your image dataset.
- **--epochs**: Training epochs (adjust as needed).
- **--output**: Path to save trained model (.tflite for Android).

### 6.2 How to Train Symptoms Text Model

1. **Prepare Data:**

    - Gather symptom descriptions and labels (CSV or TXT format).
    - Clean data: remove stopwords, correct spelling, tokenize.

2. **Install Requirements:**

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

3. **Run Training Script:**

```
python scripts/train_symptom_model.py --data_dir models/symptom_text/dataset --epochs 20
--output models/symptom_text/model.tflite
```

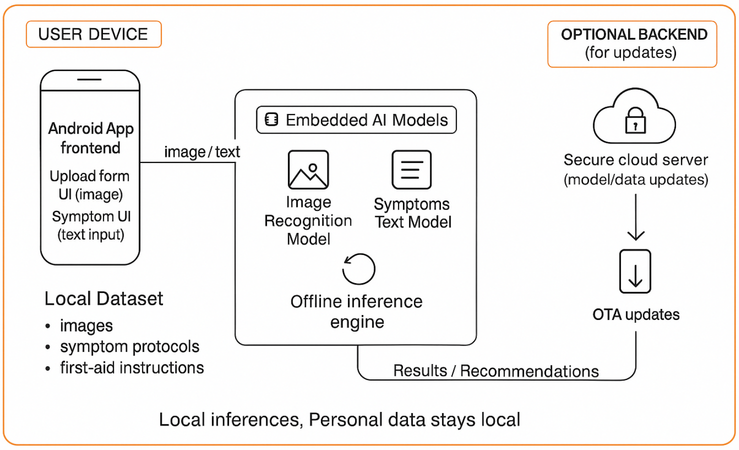
    - **--data_dir:** Path to your symptom dataset.
    - **--epochs:** Training epochs.
    - **--output:** Path to save trained model.

**Tip:** Adjust script arguments to match your data and requirements.

# 7. Usage Guide

1. **Launch App:** Register/login as a user.
2. **Image Analysis:** Upload a hand/rash image for AI-based classification.
3. **Symptoms Analysis:** Enter text-based symptoms for rapid triage.
4. **Results:** App provides diagnosis, urgency, and recommendations.
5. **Feedback:** Optionally provide feedback for misclassifications to improve the models.

# 8. Architecture Diagram



*"The architecture diagram illustrates how the EmergencyTriageApp operates entirely on a user's smartphone, with an intuitive Android UI for inputting images and symptoms. Embedded AI models (for image and text) perform real-time inference locally, ensuring offline capability and privacy. A local dataset supports first-aid protocols and recommendations. An optional backend is included for secure updates to models and protocols, but all essential features run independently on the device. Data flows from user input to local inference, displaying immediate results and recommendations."*

# 9. Image Placeholders

**Chicken Pox**

**Acne Skin**

**Tinea(Fungus)**

**Normal skin**

# 10. Roadmap (Next 2 Months)

**October - November 2025**

**1. Data Collection**

- Gather genuine, diverse skin rash and normal images (via clinics, open datasets, or partnerships).
- Collect varied symptom descriptions (medical records, surveys, etc.).

**2. Preprocessing & Cleaning**

- Crop, adjust, and normalize images.
- Annotate and balance datasets.
- Clean and normalize text data.

**3. Model Improvement**

- Retrain image and text models with new data.
- Experiment with advanced architectures (EfficientNet, transformers).
- Validate models using real-world test sets.

**4. App Enhancement**

- Improve UI for easier input.
- Add user feedback mechanism for false positives/negatives.
- Integrate analytics for usage tracking.

**5. Documentation & Release**

- Update README and user manuals.
- Prepare for Play Store release (beta testing phase).

# 11. Vision for Universal Emergency App Deployment (SAMSUNG)

## Future Implementation as a Universal Offline First-Aid App

EmergencyTriageApp has the potential to become a **default emergency first-aid application pre-installed on every smartphone**, especially on all Samsung mobile devices. The goal is to empower users to perform immediate, reliable triage and first-aid assessment in offline and resource-limited environments.

## Key Features for Universal Deployment:

- **Offline Functionality:**
  All AI models (image and text) are embedded within the APK using TensorFlow Lite or ONNX, enabling full offline performance with no server or internet required.
- **Comprehensive Local Dataset:**
  Includes a large, validated dataset of skin conditions, symptoms, and first-aid protocols covering diverse age groups and populations.
- **Optimized for Samsung Devices:**
  The app leverages Samsung's hardware accelerators (e.g., Neural Processing Unit via NNAPI) for fast, on-device inference, and is tested across a wide range of Samsung models for performance and compatibility.
- **User Interface:**
  Simple, multi-language UI designed for rapid emergency use, with clear instructions and voice prompts.

- **Security and Privacy:**
  All data processing is strictly local; no personal data leaves the device. Fully compliant with GDPR and Samsung security requirements.
- **First-Aid Protocol Integration:**
  The app can provide step-by-step first-aid instructions (CPR, wound care, allergic reactions, etc.), triage recommendations, and emergency contacts.

### Technical Requirements and Implementation Steps:

- **Model Format:**
  Use TensorFlow Lite (.tflite) and ONNX for compact neural networks deployable in mobile environments.
- **Hardware Acceleration:**
  Integrate with Samsung NNAPI and GPU/TPU delegates for real-time inference.
- **APK Packaging:**
  Bundle all assets, models, and datasets inside the APK or via AAB (Android App Bundle) to ensure 100% offline operation.
- **Localization:**
  Provide multi-language support for global deployment.
- **Testing:**
  Rigorous device testing using Samsung's device farm and emulators for full compatibility across Galaxy series.
- **OTA Updates:**
  Enable model and protocol updates via secure, signed over-the-air updates when internet is available.
- **Certification:**
  Comply with medical device software standards (ISO 13485, IEC 62304) for use as a first-aid tool.

**With these enhancements, EmergencyTriageApp can become a robust, life-saving emergency app, available on every Samsung smartphone as a built-in offline first-aid solution, leveraging state-of-the-art mobile AI and comprehensive medical data.**

## TEAM DETAIL:

Team Blue Orchids

**SRM Institute of Science and Technology(KTR)**

Theme Number: 1(Multimodal AI)

Team Members:

- Arjjun S
- B B Hemanth
- H A Pranav Jai
- S M Sudharshan
- Bala Tharun R