User Documentation

Getting Started

BoneAger is a bone age assessment tool designed to automatically and accurately estimate a child's skeletal maturity from left -hand X-rays. Built on a trained deep-learning model, BoneAger delivers specialist-level accuracy in seconds, reducing the time and subjectivity of manual assessments.

The tool was developed with multiple types of users in mind:

- **Rural and low-resource clinics** benefit from an offline, lightweight version that runs on basic hardware, making specialist-level assessment accessible without internet or high-end equipment.
- **Paediatric endocrinologists** can quickly obtain consistent, evidence-based bone age estimates to inform diagnosis and treatment of growth disorders.
- **Hospital IT teams** can deploy BoneAger on standard clinical hardware ensuring data security and easy updates.
- **Parents** gain the assurance that their child's skeletal maturity can be assessed promptly and accurately, helping clinicians act without unnecessary delays.

With Bone-Ager, you can:

- Upload a patient's hand X-ray
- Instantly receive a bone age prediction
- View a visual explanation of the model's decision
- Export and save results

System Requirements

- **Computer:** Windows 10/11 (recommended), macOS, or Linux
- **Software:** Python 3.11, pip
- **Internet:** Only required for initial download – runs offline afterwards

---

Installation

1. **Download the software**
   Obtain Bone-Ager from the provided download link or GitHub page.

Install dependencies

Open your terminal (Command Prompt, PowerShell, or Terminal app) and run:

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

Add the model

- Download the `best_bone_age_model.pth` file separately.
- Replace the `best_bone_age_model.pth` in your `bone_age` directory with your separately downloaded `best_bone_age_model.pth.`

Launch Bone-Ager

```
cd bone_age/frontend
streamlit run main.py
```

Using Bone-Ager

1. Opening the App

After launching, Bone-Ager will open in your default web browser with a clean, easy-to-read interface.

---

2. Uploading an X-ray

1. Click the **“Upload X-ray”** button.
2. Select the patient’s **hand X-ray** (JPEG, PNG, or DICOM converted to PNG/JPEG).
3. The image will appear in the preview window.

---

3. Getting Results

- Click **“Run Prediction”**.
- Within 5 seconds, the predicted bone age (in years) will display

---

4. Saving & Exporting Results

- Click **“Save Report”** to download the result as a PDF.
- Reports include:
    - Patient details (if entered)
    - Prediction result
    - Uploaded image

Specific Topics

Understanding the Prediction

- **Bone age**: The estimated skeletal maturity of the patient.
- Predictions may vary slightly depending on your computer’s operating system.

Accessibility Features

- Large, high-contrast fonts for easy reading
- Supports keyboard navigation for clinicians who prefer not to use a mouse
- Works offline for secure, on-site use

FAQs

Q: Does Bone-Ager store patient data?
A: No. All processing is done locally, and no images are uploaded to the internet.

Q: Can I use multiple X-rays at once?
A: Yes, Bone-Ager includes a batch-processing feature where users can either drag-and-drop files or select them. Each X-ray is then processed one at a time for maximum accuracy.

Q: What image formats are supported?
A: JPEG,PNG and DICOM images

Q: Does Bone-Ager store patient data?
A:Our model is built off the PyTorch library, used for many other models and will produce slightly different predictions because of this. You might also encounter slightly different performances based on the cpu of your computer as well as the os used.

Notes

Different Python types will require different downloading methods, some requiring different options as well, e.g., python3, py, pip3, -m, etc. Some systems use `python3`, while others rely on the `py` launcher for version management. Similarly, package installation might require `pip3` instead of just `pip` to ensure packages are installed for Python 3. It's important to know which commands your environment uses to avoid conflicts or errors.

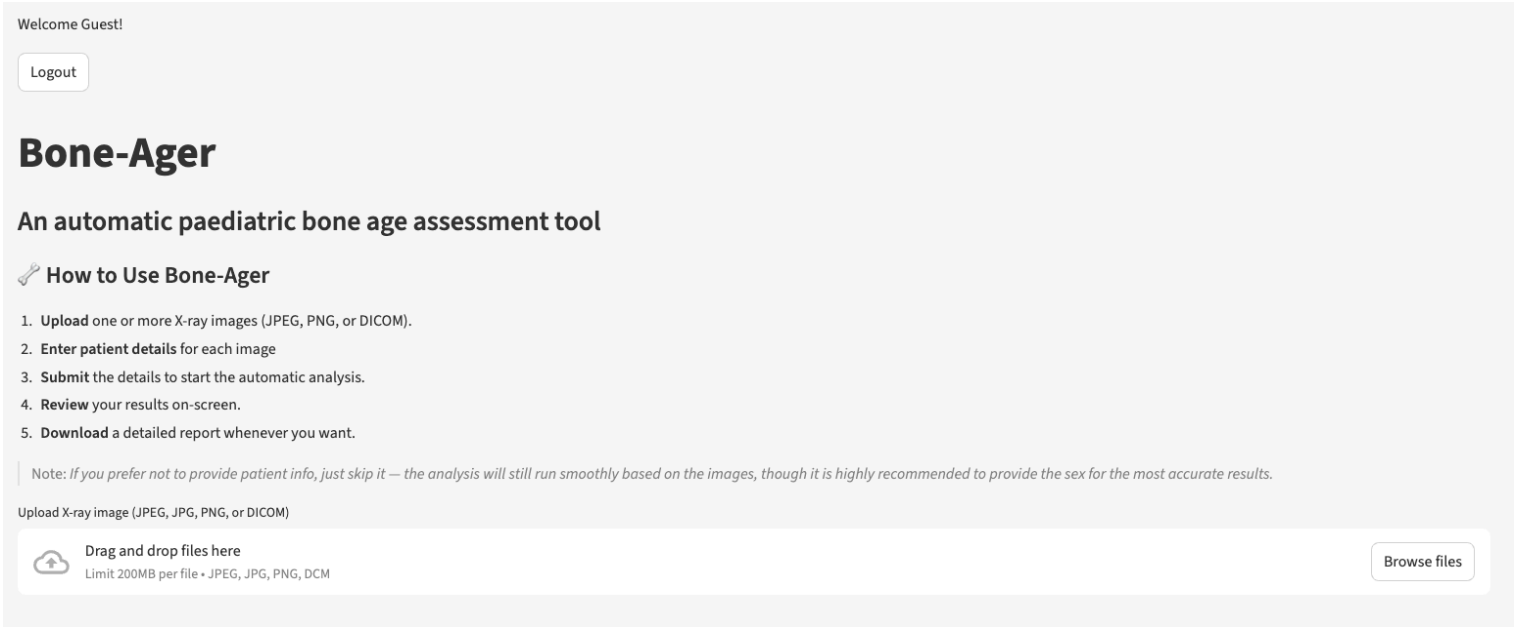
Step-by-Step Example

Goal: Predict bone age for a patient's X-ray.

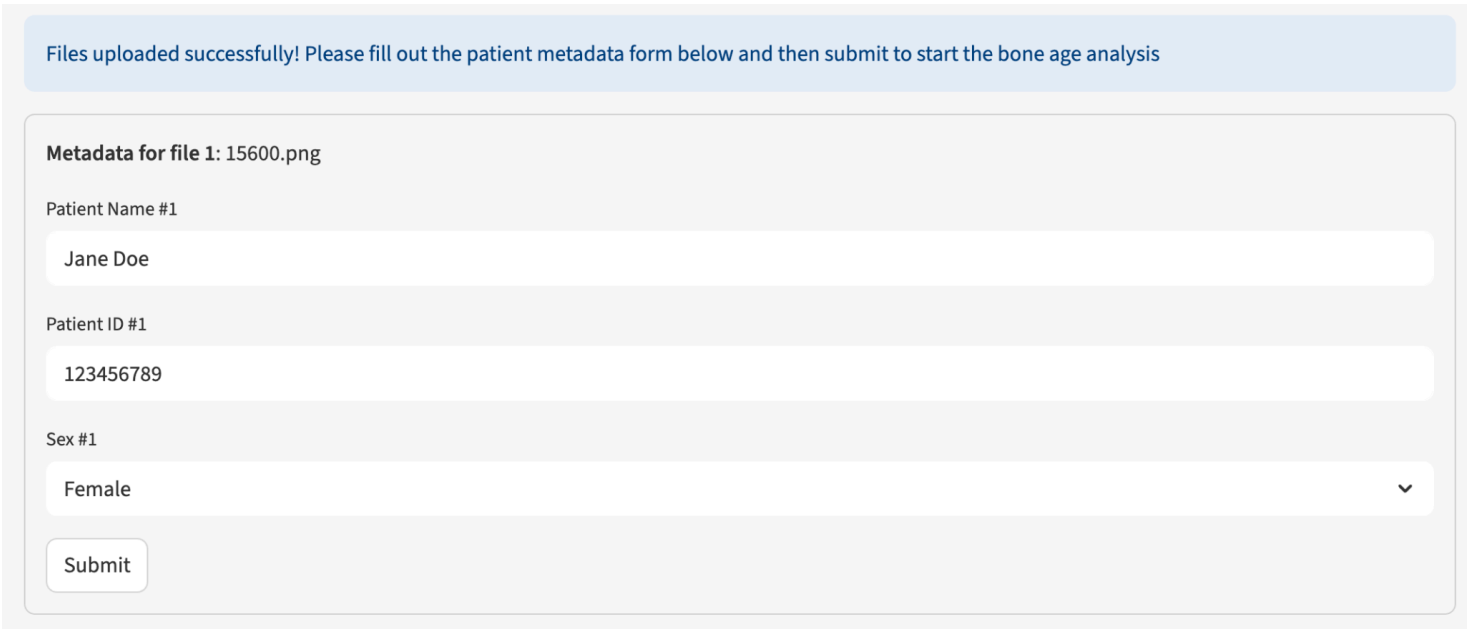
1. **Launch the App**

```
cd bone_age/frontend
streamlit run main.py
```

2. **Upload** the X-ray.

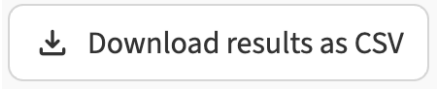


3. Input patient data

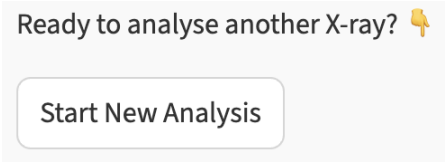


4. Run Prediction

5. **Save Report** if needed



Analyse more images



Technical Documentation

Overview

Bone-Ager is a machine learning-based desktop application for paediatric bone age assessment using hand X-rays. It allows clinicians to obtain consistent, accurate bone age predictions in under 1 minute, with offline capabilities and a streamlined user interface.

This technical document serves as a guide for system administrators and technical users tasked with installing, maintaining, and troubleshooting Bone-Ager across multiple environments.

System Requirements

Operating System Compatibility		
System	**Status**	**Notes**
Windows 10/11	Supported	Best performance
macOS	Limited	Model execution may be slower
Linux	Supported	Requires additional setup for GUI
Cloud	Experimental	GUI must be forwarded or adapted

Software Versions

- **Python**: 3.11
- **Pip**: 22+
- **Streamlit**: 1.29+

Installation Guide

Step 1: Clone the Repository

```
git clone https://github.com/jjjaden-hash/DESN2000-BINF-M13B_GAMMA.git
cd bone_age
```

Step 2: Set Up Python Environment

(Optional but recommended: Create a virtual environment)

```
python3.11 -m venv venv
source venv/bin/activate  # On Windows: venv\Scripts\activate
```

Step 3: Install Dependencies

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

Step 4: Add the Model

1. Download the trained Bone-Ager model from the provided link.
2. Move the model file (`boneager_model.pth` or similar) into the `bone_age/` directory.

Step 5: Launch the Application

```
cd frontend
streamlit run main.py
```

Software Architecture

High-Level Architecture

```
User Interface (Streamlit)
        │
        ▼
Frontend Handlers (main.py)
        │
        ▼
ML Backend (predictor.py)
        │
        ├── Preprocessing (OpenCV / PIL)
        └── Inference (PyTorch)
```

Key Components

- **Frontend (Streamlit)**: Displays upload tool, results, and patient demographics
- **Model Inference**: PyTorch-based deep learning model trained on RSNA data
- **Data Preprocessing**: Image conversion, resizing, and normalisation
- **Grad-CAM**: Optional explainability tool for visualising model focus regions
- **Export**: PDF report generation (in future versions)

Testing and Maintenance

Regular Maintenance Tasks	
Task	**Frequency**
Dependency Check	Monthly
Model Version Update	As released
Performance Benchmarking	Per release

Future Maintenance Suggestions

- Add auto-update feature for model
- Integrate logging for crash/error diagnostics
- Build in test suite using `pytest` and dummy images

Known Issues & Troubleshooting

Common Issues

Issue	Fix
App runs slowly on macOS	Use Windows or Linux if possible
Streamlit fails to launch	Ensure Python 3.11 is activated
Model not found error	Confirm model file is correctly placed
PIL/OpenCV import errors	Reinstall dependencies
GUI not rendering in cloud	Requires streamlet to expose ports

Permissions (macOS-specific)

- Ensure Streamlit has permission to access files (System Preferences → Privacy & Security).
- Gatekeeper may block model file downloads, manually allowed.

Future Enhancements

- Support for PACS/DICOM integration
- Contrast-aware UI for accessibility
- Admin login and patient tracking
- Grad-CAM explainability toggle
- PDF report download

Contact & Credits

Developed by **Team M13B Gamma**
GitHub: DESN2000-BINF-M13B_GAMMA

References

Halabi SS, et al. The RSNA Pediatric Bone Age Machine Learning Challenge. *Radiology*. 2019.

Thodberg HH, et al. The BoneXpert method for automated determination of skeletal maturity. *IEEE Trans Med Imaging*. 2009.

Martin DD, et al. Validation of a new method for bone age in Japanese children. *Horm Res Paediatr*. 2010.

van Rijn RR, et al. Bone age assessment: Automated techniques coming of age? *Acta Radiol*. 2013.