

Retinoblastoma Detector

![Python](https://img.shields.io/badge/Python-3.10-blue)
![PyTorch](https://img.shields.io/badge/Framework-PyTorch-red)
![Streamlit](https://img.shields.io/badge/UI-Streamlit-green) ![License: MIT](https://img.shields.io/badge/License-MIT-yellow)

An AI-powered application designed to detect Retinoblastoma, a rare eye cancer, from images of eyes or faces. The project uses deep learning, computer vision, and a user-friendly web app interface for diagnosis and research purposes.

Features

1. **Eye Detection**: Automatically detects eyes in face photos using Mediapipe.
2. **Disease Classification**: Classifies eyes as Healthy or Retinoblastoma and provides confidence scores.
3. **Interactive Web App**: Built with Streamlit for user-friendly interaction.
4. **Customizable Model**: Trainable Convolutional Neural Network (CNN).

Setup Instructions

1. Clone the Repository

Clone the repository to your local machine:

```
``bash
git clone https://github.com/your-username/retinoblastoma-detector.git
cd retinoblastoma-detector
``
```

2. Install Dependencies

This project uses Poetry for dependency management. Install Poetry if you don't already have it:

```
``bash
pip install poetry
``
```

Activate the environment and install all dependencies:

```
``bash
poetry install
``
```

3. Prepare the Training Data

Organize the `Training/` folder with the following structure:

```
``
```

```

Training/
├── eye/                # Healthy eye images
│   ├── image1.jpg
│   └── image2.jpg
├── retinoblastoma/    # Retinoblastoma-affected eye images
│   ├── image1.jpg
│   └── image2.jpg
...

```

Ensure that there are sufficient images in both categories to train the model effectively.

4. Train the Model

Run the training script:

```

```bash
poetry run python train.py
```

```

This script:

1. Loads the training data from the `Training/` directory.
2. Trains the Convolutional Neural Network (CNN).
3. Saves the trained model to the `models/` directory as `retinoblastoma_detector.pth`.

Usage Instructions

1. **Upload an Image**: Choose an image of an eye or face. Supported formats: `.jpg`, `.jpeg`, `.png`.
2. **View Predictions**: For face images, eyes are detected and cropped automatically. Each eye is classified as Healthy or Retinoblastoma, and confidence scores are displayed.
3. **Adjust Confidence Threshold**: Use the sidebar slider to filter predictions by confidence levels.

How It Works

1. **Eye Detection**: Uses Mediapipe to detect and crop eyes from face images.
2. **Image Preprocessing**: Resizes images to `224x224` and normalizes pixel values using mean `[0.5]` and standard deviation `[0.5]`.
3. **Model Architecture**: A CNN is used for binary classification (Healthy vs. Retinoblastoma).
4. **Inference**: Cropped eye regions are passed through the trained model, and predictions are output with confidence scores.

Dependencies

All dependencies are listed in `requirements.txt`. Notable ones include:

- **torch**: Deep learning framework.
- **torchvision**: For data augmentation and image preprocessing.
- **opencv-python-headless**: Image processing library.
- **mediapipe**: For eye detection in face images.
- **streamlit**: For building the interactive web app.

Future Improvements

1. **Pretrained Models**: Use transfer learning with pretrained architectures like ResNet.
2. **Dataset Expansion**: Include more diverse images for better generalization.
3. **Batch Processing**: Support uploading and processing multiple images simultaneously.
4. **Mobile Optimization**: Enhance the Streamlit interface for better mobile usability.
5. **API Integration**: Add RESTful API endpoints for remote inference.

Contributing

We welcome contributions to enhance the project! To contribute:

1. **Fork the repository**.

2. Create a new branch:

```
``bash
git checkout -b feature/your-feature
``
```

3. Commit your changes:

```
``bash
git commit -m "Add your feature"
``
```

4. Push your changes:

```
``bash
git push origin feature/your-feature
``
```

5. Open a **Pull Request** on GitHub.

License

This project is licensed under the **MIT License**. See the `LICENSE` file for details.

Contact

For questions or feedback, feel free to reach out:

- **Your Name**
- [Your GitHub Profile](<https://github.com/your-username>)
- [Your Email Address](mailto:your-email@example.com)