Visionrd - Label Editor Documentation

# Overview

Visionrd - Label Editor is an open-source graphical tool designed to facilitate image labeling tasks efficiently. It allows users to load a set of images, manage and edit labels, and save the results in an organized manner. The tool leverages the capabilities of PySide6, OpenCV, and PIL for image processing, making it robust and user-friendly.

# Prerequisites

Ensure you have the following installed:  
- Python: Version 3.7 or later  
- Required Python packages:  
 - PySide6  
 - OpenCV (cv2)  
 - Pillow (PIL)  
 - NumPy

# Features

- Easy-to-use interface for loading and displaying images.  
- Edit, update, and manage image labels seamlessly.  
- Zoom in/out functionality for better image inspection.  
- Grayscale image viewing for enhanced contrast.  
- Search and replace functionality for quick label editing.  
- Background color customization using an integrated color picker.  
- Handy keyboard shortcuts for faster navigation and label editing.  
- Auto-save functionality to prevent data loss.  
- Undo recent changes for mistake-proof editing.

# Installation

1. Clone the repository:  
 ```bash  
 git clone https://github.com/visionrd-ai/Visionrd---Label---Editor.git  
 ```  
2. Navigate to the project directory:  
 ```bash  
 cd Visionrd---Label---Editor  
 ```  
3. Install the required dependencies:  
 ```bash  
 pip install -r requirements.txt  
 ```

# Getting Started

To start the application, execute:

```bash  
python label\_editor.py  
```

# Usage

**1. Loading Labels**  
- Click on \*\*"Load Labels"\*\* and select your label file (e.g., `labels.txt`).  
- Ensure the file is formatted as `image\_path label` per line.

**2. Selecting Output Folder**  
- Click \*\*"Select Output Folder"\*\* to choose where you want to save edited labels.

**3. Label Editing**  
- Modify labels in the input field as needed.  
- Click \*\*"Save"\*\* or press `Enter` to save and proceed to the next image.

**4. Image Controls**  
- Use \*\*Next\*\* and \*\*Back\*\* buttons or arrow keys to navigate images.  
- Zoom in/out using the `+`/`-` buttons or `Up`/`Down` arrow keys.  
- Toggle grayscale mode with the \*\*Grayscale\*\* button.  
- Change the background color using the \*\*Pick Color\*\* button.

**5. Search & Replace**  
- Click \*\*"Search & Replace"\*\* to open the dialog.  
- Enter the search and replacement terms to update all matching labels.

**6. Undo**  
- Click \*\*"Undo"\*\* or press `Backspace` to revert recent changes.

# Keyboard Shortcuts

| Shortcut | Action |  
|-----------------|--------------------------------------|  
| `Right Arrow` | Save and move to next image |  
| `Left Arrow` | Move to the previous image |  
| `Up Arrow` | Zoom in |  
| `Down Arrow` | Zoom out |  
| `Delete` | Delete the current label |  
| `Backspace` | Undo the last change |  
| `Enter/Return` | Save label and move to the next image|

# Application Structure

The main files and folders in the project:

visionrd-label-editor/  
│  
├── label\_editor.py # Main application script  
├── visionrd\_logo.png # Application icon (optional)  
├── README.md # Project documentation  
└── requirements.txt # Python package requirements

# Error Handling

The application displays error messages if an image fails to load. Warning messages appear for invalid actions, such as entering an out-of-range index.

# Customization

You can customize the application to suit your needs:  
- Adjust `auto\_save\_interval` (default is 5000 ms) for the auto-save feature.  
- Modify `zoom\_factor` for different zooming behavior.  
- Use the color picker to personalize the background color.

# Future Scope

- Incorporate advanced image labeling techniques, including bounding boxes and polygons.  
- Add support for different file formats (CSV, JSON, etc.) for exporting labels.  
- Implement a multi-user collaborative environment.  
- Introduce AI-assisted labeling for faster label generation.

# Technologies Used

- PySide6: For building the graphical interface.  
- OpenCV: For image processing and manipulation.  
- Pillow (PIL): For handling image file formats.  
- NumPy: For efficient data management.

# Acknowledgements

- PySide6: For the Python bindings to the Qt toolkit.  
- OpenCV: For providing comprehensive image processing tools.  
- Pillow (PIL): For image manipulation capabilities.

# Contributing

Contributions are highly appreciated! To contribute:  
1. Fork the repository  
2. Create a new branch: `git checkout -b feature-name`  
3. Commit your changes: `git commit -m 'Add new feature'`  
4. Push to the branch: `git push origin feature-name`  
5. Open a pull request