🗂️ Smart Waste Classifier

A professional Streamlit application for waste classification using deep learning. This application uses a trained ResNet34 model to classify waste images into 5 categories: Cardboard, Glass, Metal, Paper, and Plastic.

✨ Features

- **Real-time Image Classification**: Upload images and get instant predictions
- **Professional Dashboard**: Clean, modern UI with comprehensive analytics
- **Performance Metrics**: Detailed model performance analysis with interactive charts
- **Disposal Recommendations**: Helpful recycling and disposal guidelines
- **Responsive Design**: Works on desktop and mobile devices

🚀 Quick Start

Prerequisites

- Python 3.8 or higher
- `my_model.pkl` file (your trained FastAI model)

Installation

1. **Clone or download the application files:**

   ```bash
   # Create a new directory for the project
   mkdir waste-classifier-app
   cd waste-classifier-app

   # Copy all the application files to this directory
   ```

2. **Install dependencies:**

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

3. **Place your model file:**
   - Ensure `my_model.pkl` is in the root directory
   - This file should contain your trained FastAI ResNet34 model

4. **Run the application:**

```bash
streamlit run app.py
```

5. **Open your browser:**
   - The app will automatically open at `http://localhost:8501`
   - If not, navigate to the URL shown in the terminal

📁 Project Structure

```
waste-classifier-app/
├── app.py                  # Main Streamlit application
├── model_handler.py        # Model loading and prediction logic
├── utils.py               # Utility functions and styling
├── config.py              # Configuration settings
├── requirements.txt       # Python dependencies
├── my_model.pkl           # Your trained model (required)
└── README.md             # This file
```

🎯 Usage Guide

1. Home Page

- Overview of the application and its features
- Model statistics and quick start guide
- Information about waste categories

2. Image Classifier

- Upload waste images (PNG, JPG, JPEG)
- Get instant classification results
- View confidence scores for all categories
- Receive disposal recommendations

3. Model Analytics

- Training data distribution
- Model performance visualizations
- Confusion matrix analysis
- Training loss curves

4. Performance Metrics

- Detailed classification reports

- Per-class performance metrics
- ROC curves and additional insights

🔧 Configuration

The application can be customized through `config.py`:

- **Model settings**: Path, architecture, input size
- **UI colors**: Primary, secondary, and accent colors
- **File limits**: Maximum upload size and dimensions
- **Categories**: Waste types and display names

📊 Model Requirements

Your `my_model.pkl` file should be a FastAI trained model with:

- **Architecture**: ResNet34 (or compatible)
- **Input size**: 224x224 pixels
- **Classes**: 5 waste categories (cardboard, glass, metal, paper, plastic)
- **Format**: FastAI learner export (.pkl)

🛠️ Troubleshooting

Common Issues

1. **Model not loading:**
   - Ensure `my_model.pkl` exists in the root directory
   - Check that FastAI is properly installed
   - Verify the model file is not corrupted
2. **Import errors:**
   - Install all dependencies: `pip install -r requirements.txt`
   - Ensure Python version is 3.8+
3. **Image upload issues:**
   - Check file format (PNG, JPG, JPEG only)
   - Ensure file size is under 10MB
   - Verify image is not corrupted
4. **Slow performance:**
   - Consider using GPU if available
   - Reduce batch size in config

    - Optimize image preprocessing

Error Messages

The application provides detailed error messages for common issues:

- Model loading failures
- Invalid image formats
- File size limitations
- Prediction errors

🎨 Customization

Styling

- Modify CSS in `utils.py` for custom themes
- Update colors in `config.py`
- Add custom fonts and layouts

Features

- Add new waste categories in `config.py`
- Implement batch processing
- Add data export functionality
- Include more visualization options

📈 Performance

The application is optimized for:

- **Fast loading**: Cached model and data
- **Responsive UI**: Modern CSS and efficient rendering
- **Memory efficiency**: Optimized image processing
- **Scalability**: Modular architecture

🔒 Security

Security considerations:

- File upload validation
- Image format verification
- Size limitations
- Input sanitization

🌍 Environmental Impact

This application promotes:

- **Proper waste sorting**: Accurate classification
- **Recycling awareness**: Disposal recommendations
- **Environmental education**: Impact information
- **Sustainable practices**: Waste reduction tips

📝 Development

Adding New Features

1. **New pages**: Add to main navigation in `app.py`
2. **New metrics**: Update analytics in visualization functions
3. **Custom models**: Modify `model_handler.py`
4. **UI components**: Add to `utils.py`

Code Structure

- **app.py**: Main application and page routing
- **model_handler.py**: ML model integration
- **utils.py**: Helper functions and styling
- **config.py**: Configuration management

🤝 Contributing

To contribute to this project:

1. Fork the repository
2. Create a feature branch
3. Make your changes
4. Test thoroughly
5. Submit a pull request

📄 License

This project is open source and available under the MIT License.

🆘 Support

For support:

1. Check the troubleshooting section
2. Review error messages carefully
3. Ensure all dependencies are installed
4. Verify model file integrity

⚡ Performance Tips

1. **Optimize images**: Resize large images before upload
2. **Clear cache**: Restart app if experiencing issues
3. **Update dependencies**: Keep packages current
4. **Monitor resources**: Check memory usage with large files

---

Happy Classifying! ♻️