



# Logo Applier - Batch Logo Watermarking Tool

---

Python 3.8+

License MIT

Platform Windows | macOS | Linux

[IT](#) [Versione italiana](#)

Professional desktop application to automatically apply logos and watermarks to image batches, featuring an intuitive GUI and customizable background support.

version 1.0.0

## ✦ Key Features

- 🎯 **Interactive Positioning** - Real-time logo preview while hovering with mouse
- 📐 **Automatic Mode** - Apply logo to predefined fixed positions (4 corners)
- 🎨 **Custom Backgrounds** - Add colored backgrounds to logos (6 colors, 3 shapes)
- 📏 **Smart Resizing** - Logo adapts to image orientation
- 💾 **Settings Persistence** - Your preferences are remembered
- 📊 **Progress Bar** - Real-time processing monitoring
- 🔄 **Batch Processing** - Process hundreds of images in minutes
- 🏗️ **Modular Architecture** - Code organized following software engineering principles

## 🚀 Quick Installation

### Requirements

- Python 3.8 or higher
- Pillow (automatically installed)

### Setup

```
# Clone the repository
git clone https://github.com/yourusername/logo-applier.git
cd logo-applier

# Install dependencies
pip install -r requirements.txt

# Run the application
python main.py
```

## 📖 How to Use

## 1. Basic Setup

1. Select the **folder with images** to process
2. Select the **logo file** (preferably PNG with transparency)
3. Select the **destination folder** for processed images

## 2. Logo Settings

- **Size:** Choose between 5%, 10%, 15%, 20% or 25% (automatically adapts to orientation)
- **Margin:** Set distance from edges (1%-15%)

## 3. Logo Background (Optional)

Customize your logo with a background:

- **Colors:** None, White, Yellow, Orange, Red, Light gray, Pale pink, Light yellow, Light orange, Pastel blue, Sky blue, Mint green
- **Shapes:** Circular, Rectangular, Oval

## 4. Positioning Mode

### **Manual Mode**

Maximum precision for each image:

- Preview window opens for each photo
- Logo follows mouse cursor in real-time
- Click to confirm position
- Perfect for artistic photography or specific compositions

### **Automatic Mode**

Fast and efficient for uniform batches:

- Choose one of 4 fixed positions:
  - Top left
  - Top right
  - Bottom left
  - Bottom right
- Ideal for large quantities of similar images

## Use Cases

### Professional Photographer

```
Mode: Manual
Logo size: 10%
Background: White Rectangular
Position: Custom for each photo
```

Perfect for portfolios where each image requires specific positioning.

## E-commerce

```
Mode: Automatic
Logo size: 15%
Background: None
Position: Bottom right
```

Ideal for quickly applying branding to hundreds of product photos.

## Social Media

```
Mode: Automatic
Logo size: 20%
Background: Sky Blue Circular
Position: Top right
```

Highly visible and appealing logo to maximize branding.

## Project Architecture

```
logo-applier/
├── main.py           # Entry point
├── config.py         # Configuration
├── gui/              # Graphical interface
│   ├── main_window.py # Main window
│   └── preview_window.py # Preview window
└── utils/            # Utilities
    ├── image_processor.py # Image processing
    └── settings_manager.py # Settings management
```

The project follows **SOLID principles** and uses modular architecture to facilitate maintenance and future extensions.

## Advanced Features

### Smart Resizing

Logo is resized adaptively:

- **Horizontal Images:** size calculated on width
- **Vertical Images:** size calculated on height

## Border Protection

In manual mode, the logo cannot exceed image borders - it's automatically limited to the visible area.

## Partial Save

If you interrupt processing, all already processed images are automatically saved.

## Supported Formats

- **Input:** JPG, JPEG, PNG, BMP, GIF
- **Output:** High-quality JPEG (95%)

## Customization

### Adding New Colors

Edit `config.py`:

```
BACKGROUND_COLORS = {  
    # ... existing colors ...  
    'Green': '#00FF00', # New color  
}
```

### Changing Output Quality

In `config.py`:

```
OUTPUT_QUALITY = 100 # Maximum quality
```

## Performance

- **10 Full HD images:** ~20-30 seconds
- **100 4K images:** ~10-15 minutes
- **RAM Memory:** 100-500MB (depends on resolution)

## Troubleshooting

"Module not found"

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

"No module named 'gui'"

Verify that `__init__.py` files exist in `gui/` and `utils/`:

```
touch gui/__init__.py utils/__init__.py
```

## Logo not visible

- Use PNG with transparent background
- Increase size percentage (15-20%)
- Try adding a colored background

## Contributing

Contributions are welcome!

1. Fork the project
2. Create a branch (`git checkout -b feature/NewFeature`)
3. Commit changes (`git commit -m 'Add: NewFeature'`)
4. Push to branch (`git push origin feature/NewFeature`)
5. Open a Pull Request

## Contribution Ideas




- ☐ Video support (watermark on frames)
- ☐ Logo rotation
- ☐ Effects (shadow, transparency)
- ☐ Text watermark support
- ☐ Batch thumbnail preview

## License

This project is released under the MIT License. See [LICENSE](#) for details.

## Author

### Michele Barbella

-  Email: [m.barbella5@gmail.com](mailto:m.barbella5@gmail.com)
-  LinkedIn: [michele-barbella](#)
-  GitHub: [@michelebarbella](#)