Documentação Técnica — Detection Lito Errors

## 1. 📋 Sumário

Este documento descreve o funcionamento da aplicação **Detection Lito Errors**, um sistema desenvolvido em Python com customtkinter para auxiliar na inspeção de folhas de alumínio litografadas, com foco na deteção de defeitos, alinhamento e validação visual. A aplicação é composta por uma interface gráfica interativa que oferece diversos modos de operação para diferentes perfis de utilizador.

## 2. 🎯 Objetivo do Projeto

Criar uma ferramenta de apoio à inspeção automatizada de folhas de alumínio utilizadas na produção de latas, com as seguintes funcionalidades principais:

* Alinhamento entre imagem atual e imagem de referência (template).
* Criação e edição de máscaras da área útil da folha.
* Detecção de defeitos com base em diferenças estruturais e variações de cor.
* Ajuste da posição física da câmera em tempo real.
* Gestão de utilizadores com diferentes permissões.
* Interface simples e intuitiva para operação em ambiente fabril.

## 3. ⚙️ Arquitetura do Software

### Linguagem:

* Python 3.10+

### Bibliotecas Principais:

* customtkinter
* OpenCV
* Pillow
* NumPy

**Estrutura Geral:**

Detection\_Lito\_Errors/

│

├── main.py # Janela principal do software

├── windows/ # Módulo de interfaces gráficas (janelas)

│ ├── adjust\_positions.py

│ ├── alignment\_adjust.py

│ ├── camera\_adjust\_positions.py

│ ├── create\_leaf\_mask.py

│ ├── create\_users.py

│ ├── detect\_cans\_auto.py

│ ├── gallery.py

│ ├── inspection\_window.py

│ └── login\_window.py

├── data/

│ ├── raw/ # Imagens de referência e atuais

│ │ ├── fba\_template.jpg

│ │ ├── fba\_actual.jpg

│ │ └── fba\_template\_persp.jpg

│ └── mask/ # Máscara da área útil

│ └── leaf\_mask.png

├── config/

│ └── users.json # Lista de utilizadores

└── ...

## 4. 🖥️ Interface Principal (main.py)

A aplicação é lançada através do ficheiro main.py, que apresenta uma janela principal (App) com três colunas:

### Coluna Esquerda – Utilizador

* Login
* Novo User (restrito a Admins)

### Coluna Central – Configurações

* Adjust Positions: Define regiões das latas na folha.
* Máscara: Define a área útil da folha.
* Alignment Adjust: Ajusta o alinhamento da folha atual com o template.
* Check Camera Positions: Alinha a folha fisicamente na câmera.
* Auto Detect Cans: Identifica automaticamente os contornos das latas.

### Coluna Direita – Inspeção

* Ver Galeria: Acesso a imagens anteriores.
* Inspecção: Executa a inspeção atual com visualização de defeitos.

## 5. 👥 Perfis de Utilizador

| **Tipo de Utilizador** | **Acesso** |
| --- | --- |
| User | Apenas visualização e inspeção |
| Admin | Pode ajustar posições e máscara |
| SuperAdmin | Acesso total, incluindo ajustes de câmera e auto-detecção |

## 6. 🔄 Fluxo de Navegação

1. **Login** → Verifica permissões
2. **Acesso liberado aos botões conforme permissões**
3. **Execução de funções (cada uma abre uma nova janela com funcionalidades específicas)**
4. **Ao fechar qualquer janela, a janela principal é restaurada**

## 7. 📂 Janelas/Componentes do Sistema

Cada botão do main.py abre uma janela independente, descrita separadamente (cada uma será documentada posteriormente):

| **Janela** | **Módulo** | **Descrição** |
| --- | --- | --- |
| AdjustPositionsWindow | adjust\_positions.py | Define posições das latas |
| LeafMaskCreator | create\_leaf\_mask.py | Criação da máscara da folha |
| AlignmentWindow | alignment\_adjust.py | Alinha folha atual ao template |
| CameraAdjustPosition | camera\_adjust\_positions.py | Ajuste da perspectiva física |
| AutoDetectCans | detect\_cans\_auto.py | Detecção automática de latas |
| InspectionWindow | inspection\_window.py | Executa inspeção e mostra defeitos |
| GalleryWindow | gallery.py | Acesso a inspeções anteriores |
| LoginWindow | login\_window.py | Login de utilizadores |
| NewUserWindow | create\_users.py | Criação de novos utilizadores |

## 8. 🧪 Testes e Validações

* Testes manuais executados com imagens reais e com falhas simuladas.
* Validação da segmentação de defeitos por meio visual e comparação com inspeção manual.

## 9. 📌 Considerações Finais

Este sistema foi desenvolvido com foco na modularidade e extensibilidade. A interface foi desenhada para operar com operadores com diferentes níveis de acesso, e a arquitetura permite futuras expansões, como integração com câmeras em tempo real, armazenamento de históricos em base de dados, ou exportação de relatórios.

## 10. 📎 Anexos

* Prints das janelas
* Exemplos de defeitos detetados
* Máscaras de referência
* JSON de utilizadores de exemplo