



**UNIVERSIDADE FEDERAL DO AMAZONAS  
INSTITUTO DE CIÊNCIAS EXATAS  
ENGENHARIA ELÉTRICA**

**GIANLUCCA DE CRISCI**

**DESENVOLVIMENTO DE UM SISTEMA DE CONTAGEM DE  
IMPRESSÃO DE PCBs EM SMT-PRINTING MACHINES, USANDO  
RECONHECIMENTO DE IMAGENS.**

**MANAUS**

**2023**



**UNIVERSIDADE FEDERAL DO AMAZONAS  
INSTITUTO DE CIÊNCIAS EXATAS  
ENGENHARIA ELÉTRICA**

**GIANLUCCA DE CRISCI**

**DESENVOLVIMENTO DE UM SISTEMA DE CONTAGEM DE  
IMPRESSÃO DE PCBs EM SMT-PRINTING MACHINES, USANDO  
RECONHECIMENTO DE IMAGENS.**

Monografia referente ao projeto final de curso no ano de 2023.

Orientador: Prof. Dr. André Luiz Duarte Cavalcante

**MANAUS  
2023**

GIANLUCCA DE CRISCI

**Desenvolvimento de um Sistema de Contagem de Impressão de PCBs em SMT-Printing Machines, Usando Reconhecimento de Imagens**

Trabalho de conclusão de curso, apresentado à Faculdade de Engenharia Elétrica, da Universidade Federal do Amazonas, como parte das exigências para a obtenção do título de Bacharelado em Engenharia Elétrica Eletrônica.

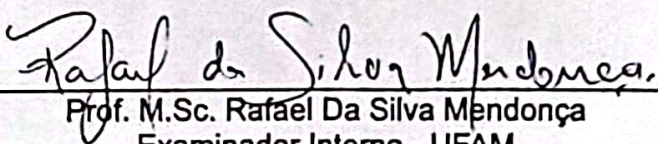
Manaus, 10 de Novembro de 2023.

**BANCA EXAMINADORA**



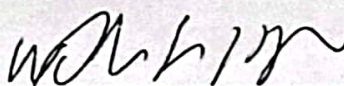
---

Prof. Dr. André Luiz Duarte Cavalcante  
Orientador - UFAM



---

Prof. M.Sc. Rafael Da Silva Mendonça  
Examinador Interno - UFAM



---

Prof. Dr. Waldir Sabino da Silva Júnior  
Examinador Interno - UFAM

## RESUMO

Este trabalho descreve a implementação de um algoritmo para monitoramento e controle de impressões e desgaste de estêncil para placas eletrônicas em um processo fabril. O objetivo do trabalho é demonstrar a possibilidade de uma aplicação já adequada ao ambiente da Indústria 4.0, que atende às exigências cada vez maiores dos clientes no mercado, como o controle de gastos, a melhora da qualidade e o atendimento às normas especificadas dos seus produtos. Descreve-se um caso fabril onde há a utilização do estêncil para a produção das placas. Desenvolveu-se a solução através da detecção do objeto em movimento e assim, o controle da quantidade de vezes que este estêncil é utilizado. Para facilitar o gerenciamento também foi criado um banco de dados na nuvem para armazenar os dados das quantidades e estêncil supervisionados, bem como enviar essas informações para e-mail's específicos selecionados. O desenvolvimento foi realizado em Python. Ao final, constatou-se que, com a automação industrial desenvolvida com este tipo de algoritmo possibilita um controle mais preciso do processo produtivo, o que assegura uma melhor qualidade ao produto final.

**Palavras-chaves:** Indústria 4.0. Algoritmo. Visão Computacional. Detecção do Objeto. Python. Banco de Dados.

## **ABSTRACT**

This work describes the implementation of an algorithm for monitoring and controlling the printing and wear of molds for electronic boards in a manufacturing process. The goal of the work is to demonstrate the possibility of an application already suitable for the Industry 4.0 environment, meeting the increasing demands of customers in the market, such as cost control, quality improvement, and compliance with specified product standards. A manufacturing case is described where molds are used in the production of boards. The solution was developed by detecting the moving object and, thus, controlling the number of times this mold is used. To facilitate management, a cloud database was also created to store data on the quantities and monitored molds, as well as to send this information to selected specific email addresses. The development was carried out in Python. In conclusion, it was found that industrial automation developed with this type of algorithm enables more precise control of the production process, ensuring better quality of the final product.

**Keywords:** Industry 4.0, Algorithm, Computer Vision, Object Detection, Python, Database.

# NOMENCLATURA

*CNN* Convolutional Neural Network

*CPU* Central Processing Unit

*DBMS* Data Base Management System

*GUI* Graphical User Interface

*HSV* Hue, Saturation, Value

*IoT* Internet of Things

*PCB* Printed Circuit Board

*R – CNN* Region Based Convolutional Neural Networks

*SMT* Surface Mount Tecnology

*SMTP* Simple Mail Transfer Protocol

*SQL* Structured Query Language

# SUMÁRIO

<b>Nomenclatura</b>	<b>5</b>
<b>1 Introdução</b>	<b>8</b>
<b>1.1 Considerações iniciais</b>	8
<b>1.2 Objetivos gerais</b>	9
<b>1.3 Objetivos específicos</b>	9
<b>2 Fundamentação Teórica</b>	<b>10</b>
<b>2.1 Indústria 4.0</b>	10
2.1.1 Automação Industrial	10
2.1.2 Elementos da Automação Industrial	11
2.1.3 Qualidade no processo Produtivo	12
<b>2.2 Processamento de Imagens</b>	13
2.2.1 Processamento em Lote de Vídeos	14
2.2.2 Detecção e Rastreamento de Objeto em Vídeo	15
2.2.3 Remoção de Imagem Espelhada em Vídeo	15
<b>2.3 Sistemas para armazenamento de dados</b>	16
2.3.1 Modelos de Banco de Dados	17
2.3.2 Integração com Banco de Dados MySQL	17
2.3.3 Exportação dos Dados para Excel	18
2.3.4 Envio dos dados através do e-mail	19
<b>2.4 Interface Gráfica dentro da indústria</b>	19
2.4.1 Criação de Interface Gráfica para Visualização de Dados	19
<b>3 Descrição do Problema</b>	<b>21</b>
<b>3.1 Aquisição do vídeo</b>	21
<b>3.2 Detecção do Objeto</b>	22
<b>3.3 Remoção do espelhamento e falsas falhas</b>	22
<b>3.4 Adicionando informações visuais no vídeo</b>	22
<b>3.5 Gerenciamento de dados via Banco de Dados MySQL</b>	23
3.5.1 Enviando informação ao Banco de Dados MySQL	23
3.5.2 Visualização de Dados no Banco de Dados MySQL	23
<b>3.6 Atualizações e envio das informações por e-mail na Interface</b>	23
<b>4 Proposta de Solução</b>	<b>24</b>

4.1	Aquisição do vídeo . . . . .	24
4.2	Detecção do Objeto . . . . .	26
4.3	Remoção do espelhamento e falsas falhas . . . . .	28
4.4	Adicionando informações visuais no vídeo . . . . .	30
4.5	Enviando informação ao Banco de Dados MySQL . . . . .	31
4.6	Visualização de Dados no Banco de Dados MySQL . . . . .	32
4.7	Atualizações e envio das informações por e-mail na Interface . . . . .	34
<b>5</b>	<b>Resultados . . . . .</b>	<b>36</b>
5.1	Detecção do Objeto . . . . .	36
5.2	Remoção do espelhamento e falsas falhas . . . . .	37
5.3	Adicionando informações visuais no vídeo . . . . .	39
5.4	Envio e Visualização de Dados no Banco de Dados MySQL . . . . .	42
5.5	Atualizações e envio das informações por e-mail na Interface . . . . .	45
<b>6</b>	<b>Conclusão . . . . .</b>	<b>47</b>
	<b>Referências . . . . .</b>	<b>48</b>



# 1 INTRODUÇÃO

## 1.1 Considerações iniciais

Os avanços na tecnologia da informação permitiram o aparecimento da Indústria 4.0, a qual está se popularizando agora no Brasil. Em indústrias produtoras de placas eletrônicas, como as existentes no Pólo Industrial de Manaus (PIM), a exigência por um melhor controle de produção é cada vez maior. Dessa forma, o controle de quantos componentes são utilizados, quantas vezes é utilizado um certo equipamento e até mesmo qual o lote dos componentes fornecidos para se montar uma placa eletrônica é essencial.

Em particular, todo processo produtivo destas placas começam com a impressão delas, através da pasta de solda ou de uma cola especial, o que é feito através de um estêncil com as aberturas corretas casadas com cada local que se deve haver a montagem de um componente para o despejo desse material. A contagem da quantidade de uso de um estêncil é fundamental para se ter o controle de quando há a necessidade de troca devido ao gasto com o decorrer do uso, tendo em vista que a pasta de solda é espalhada pelas fissuras no estêncil através de uma lamina. Tendo em vista esta necessidade, uma solução não invasiva a máquina para que não haja perda de garantia seria o uso de técnicas de visão computacional como [2].

Técnicas de visão computacional podem ajudar a extrair mais e melhores informações de ambientes e processos, permitindo uma economia significativa de recursos. Além disso, sua aplicação pode aumentar a eficiência humana garantindo um controle mais preciso do processo, particularmente quando se trata de tarefas repetitivas ou tediosas. Dessa forma, a detecção automática de objetos se torna de grande ajuda durante os procedimentos fabris como [3] demonstra com a detecção da qualidade de placas eletrônicas.

A detecção da aplicação de solda em uma placa, criando-se o controle de quantas vezes o estêncil é utilizado, torna possível o controle de quando será necessário solicitar outro estêncil, devido ao desgaste deste material em contato com a lâmina afiada que espalha a solda, sem que haja perda na produtividade. Tendo em vista este problema, este trabalho tem como objetivo abordar a implementação de um algoritmo para o

monitoramento da impressão da placa no processo, com o objetivo de propor um controle automatizado sobre a quantidade de vezes que um estêncil é utilizado na fábrica. Além disso, com comunicação a um banco de dados na nuvem é possível alertar a um, ou mais, operadores cadastrados, quando houver a necessidade de compra de um novo estêncil, através de um e-mail, contendo a devida quantidade de uso atual do produto monitorado pela aplicação.

Em geral, na indústria de montagem de placas eletrônicas, usa-se os termos *printagem*, *printar*, *printer*, do inglês, como neologismos para seus equivalentes em português *impressão*, *imprimir*, *impressora*, como referência para o maquinário ou o processo dentro da produção de placas eletrônicas, sendo este equipamento responsável por executar a “impressão” da pasta de solda de forma adequada na placa. Este trabalho usará ambos os termos, em português e os seus neologismos em inglês, como sinônimos.

## 1.2 Objetivos gerais

Elaborar um sistema de gerenciamento e monitoramento de impressão e desgaste do estêncil para placas eletrônicas.

## 1.3 Objetivos específicos

O sistema a ser desenvolvido irá:

- Realizar a detecção do movimento durante a impressão;
- Automatizar o processo contagem de impressão em placas;
- Enviar notificações da quantidade de impressões realizadas;
- Mostrar o histórico da quantidade de placas impressas;

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Indústria 4.0

O termo Indústria 4.0 foi utilizado pela primeira vez em 2011, na Feira de Hanover, Alemanha, para definir o que seria a Quarta Revolução Industrial. Foi definido como prioridade uma visão de negócios voltada para a transformação digital. O termo veio de um projeto da indústria alemã denominado *Plattform Industrie 4.0*, que tinha como intuito remodelar os conceitos de produtividade das indústrias alemãs e mantê-las entre as mais competitivas do mundo [4].

A Indústria 4.0 é definida pela combinação de tecnologias físicas e digitais como a robótica, inteligência artificial, detecção em tempo real e Internet das coisas (IoT), criando-se uma empresa interconectada e autônoma. Com essas tecnologias, uma empresa é capaz de se comunicar em tempo real, fornecendo dados suficientes para a geração de decisões que serão aplicadas novamente ao sistema físico. Tal concepção também permite a integração entre fábrica, fornecedores, trabalhadores e consumidores [5].

O termo Indústria 4.0 abrange um conjunto de tecnologias de ponta ligadas à internet com objetivo de tornar os sistemas de produção mais flexíveis e colaborativos. Nessa abordagem, as máquinas usam auto-otimização, autoconfiguração e até mesmo inteligência artificial para completar tarefas complexas, a fim de proporcionar eficiências de custo muito superiores e bens ou serviços de melhor qualidade [6].

Utilizou-se, neste trabalho, esta ideia do uso de várias tecnologias para se realizar um estudo de uma análise em tempo real de um processo físico, para fornecer *feedback* da necessidade da troca deste sistema físico. O sistema físico em questão é um estêncil de impressão de placas de circuito impresso.

#### 2.1.1 Automação Industrial

Automação Industrial refere-se à aplicação de sistemas de controle para operar equipamentos em processos industriais, reduzindo ou eliminando a intervenção humana com o objetivo de melhorar a eficiência dos processos, reduzir custos e melhorar a

qualidade, assegurar mínimos riscos ocupacionais, especialmente em ambientes perigosos através da sua flexibilidade que permite a adaptação rápida a mudanças nas demandas de produção. Isso tudo com a integração do monitoramento em tempo real fornecendo informações para análise e tomada de decisões em tempo hábil para prevenir o problema e não solucionar o problema.

### 2.1.2 Elementos da Automação Industrial

Há uma variedade de componentes inter-relacionados que fazem parte do processo de automação industrial que visam otimizar e controlar processos produtivos. Os elementos-chave dessa abordagem incluem sensores, controladores e atuadores.

Os sensores desempenham o papel de medir variáveis físicas, como temperatura e pressão, convertendo dados ambientais em sinais elétricos ou digitais, os quais são então levados aos sistemas de controle.

Os atuadores agem sobre e/ou influenciam o ambiente com base nas decisões dos sistemas de controle. São os motores, válvulas e outros dispositivos que agem sobre o sistema.

Controladores Lógicos Programáveis (PLCs) representam o cérebro da automação industrial. Esses dispositivos executam algoritmos programados para controlar processos. A flexibilidade destes dispositivos, por serem programáveis, permite a adaptação a diferentes contextos e requisitos de produção.

Além destes, pode-se citar também os Sistemas Supervisórios (SCADA), que são os responsáveis por monitorar e coletar dados em tempo real de diversos pontos do processo industrial e, utilizando interfaces homem-máquina (HMI), fornecem uma visão abrangente e facilitada do controle efetivo do sistema produtivo.

A comunicação entre esses dispositivos é essencial, e as redes industriais, como Profibus e Modbus, desempenham um papel crucial. Elas facilitam a troca eficiente de dados entre sensores, atuadores, PLCs e outros elementos da automação. Esses elementos formam uma hierarquia de controle, desde a captação de dados até a supervisão e integração de sistemas. A automação industrial busca otimizar a produção, aumentar a eficiência e garantir a segurança, destacando-se como uma vantagem competitiva na indústria moderna.

### 2.1.3 Qualidade no processo Produtivo

A garantia da qualidade em um processo produtivo visa que os produtos sendo produzidos atendam ou superem as expectativas dos clientes. A implementação efetiva do controle de qualidade ao longo do processo produtivo é crucial para alcançar padrões elevados e sustentáveis que atendam esses requisitos. Garantia da qualidade é inerente aos padrões que formam a Indústria 4.0.

O controle de qualidade abrange também atividades contínuas que monitoram e regulam os processos de produção, muitas vezes utilizando métodos estatísticos como o Controle Estatístico de Processo (CEP). Um dos métodos para controle de qualidade é a inspeção.

A inspeção de produtos é uma prática comum para verificar a conformidade dos produtos com padrões específicos. A inspeção envolve a análise visual e, em alguns casos, testes físicos ou químicos para identificar defeitos.

As indústrias aderentes à Indústria 4.0 adotam normas internacionais, que vão muito além da conhecida ISO 9001, que estabelece requisitos para sistemas de gestão da qualidade. Obter certificações reconhecidas pela indústria não apenas valida a qualidade do processo produtivo, mas também pode abrir oportunidades de mercado e fortalecer a confiança do cliente.

A qualidade do processo produtivo também se estende aos fornecedores. Avaliações regulares garantem que os insumos atendam aos padrões especificados. Colaborar com fornecedores na busca por melhores práticas e inovações contribui para aprimorar a qualidade dos produtos finais. Ter os fornecedores igual preocupação com a qualidade, gera uma cadeia de qualidade em todo o *supply chain* industrial.

Um dos métodos para melhoria contínua da qualidade é o uso da metodologia PDCA (Plan-Do-Check-Act). Ela é frequentemente aplicada para promover a melhoria através do planejamento de ações, implementação, verificação dos resultados e ajustes para aprimorar constantemente os processos e a qualidade. A busca contínua por inovação e novas tecnologias é um componente chave da melhoria contínua, permitindo que as organizações se mantenham competitivas e atendam às crescentes expectativas do mercado.

As práticas de qualidade no processo produtivo não são apenas métodos, mas uma cultura organizacional. Adotando abordagens proativas, implementando práticas de controle de qualidade eficazes e cultivando uma mentalidade de melhoria contínua, as empresas podem alcançar padrões elevados de qualidade e satisfação do cliente.

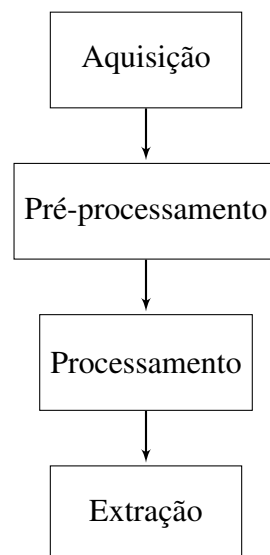
## 2.2 Processamento de Imagens

O Processamento de Imagens é o conjunto de técnicas de engenharia que permitem a manipulação de imagens de entrada e a geração de imagens de saída, alteradas de forma que conhecimento novo possa ser agregado a todo o processo.

Ela desempenha um papel crucial em diversas indústrias, permitindo a extração de informações valiosas a partir de imagens digitais de entrada. À medida que tecnologias avançam, o campo continua a evoluir, proporcionando soluções inovadoras para uma variedade de aplicações como [7].

O processo começa com a aquisição de imagens (ver diagrama na Figura 1), o que é realizado por meio de câmeras ou outros dispositivos. Essas imagens podem ser capturadas em diferentes modalidades, como RGB, infravermelho ou tomografia.

Em seguida é realizado a etapa de pré-processamento que visa melhorar a qua-



**Figura 1 – Etapas de um processamento de imagem**

lidade das imagens. Inclui correções de iluminação, remoção de ruídos e ajustes de contraste para preparar as imagens para análises mais avançadas.

Após o pré-processamento das imagens é realizado o processamento da imagem propriamente dito, como, por exemplo, a realização de uma segmentação da imagem em regiões distintas com base em características comuns. O processo de segmentação, por exemplo, é fundamental para isolar objetos de interesse e facilitar análises posteriores. A segmentação também possibilita a extração de características essenciais para identificar e quantificar objetos na imagem, onde é possível incluir formas, texturas ou padrões relevantes para a análise.

Tendo uma imagem processada, é realizado uma análise dos resultados deste processamento para tomar decisões ou extrair conhecimento. Isso pode envolver a classificação de objetos, detecção de anomalias ou reconhecimento de padrões específicos.

Entretanto, há desafios durante todo essas etapas de processamento de uma imagem (que pode ser visto em 1), como lidar com grandes volumes de dados ou garantir precisão nas análises. Tendências atuais envolvem a integração de inteligência artificial e aprendizado profundo para melhorar a eficiência e precisão das análises.

### 2.2.1 Processamento em Lote de Vídeos

O processamento em lote de vídeos refere-se à capacidade de processar uma coleção de vídeos de uma só vez, em vez de tratar cada vídeo individualmente. Essa técnica é amplamente aplicada em várias áreas, incluindo processamento de mídia, análise de vídeo, visão computacional e muito mais.

Existem muitas situações em que o processamento em lote de vídeos é vantajoso. Por exemplo, ao lidar com grandes volumes de conteúdo de vídeo, como transmissões ao vivo ou vigilância por vídeo, é essencial automatizar tarefas de processamento, como detecção de objetos, rastreamento de movimento ou análise de quadros-chave, como pode ser visto em [8].

O processamento em lote de vídeos pode ser alcançado por meio da criação de pipelines de processamento de vídeo que aceitam uma coleção de vídeos como entrada e aplicam algoritmos de processamento em cada vídeo sequencialmente. Isso economiza tempo e recursos computacionais significativos em comparação com o processamento

manual de cada vídeo.

### 2.2.2 Detecção e Rastreamento de Objeto em Vídeo

As técnicas de detecção de objetos e o rastreamento de seus movimentos em um vídeo são cruciais na área de visão computacional e processamento de vídeo. Essas técnicas têm uma ampla gama de aplicações, incluindo vigilância, veículos autônomos, monitoramento de tráfego, realidade aumentada e muito mais.

A detecção de objetos em vídeo refere-se à identificação e localização de objetos em quadros individuais em um fluxo de vídeo. É uma tarefa fundamental que forma a base para muitas outras aplicações de visão computacional. A detecção bem-sucedida envolve a aplicação de algoritmos que podem reconhecer objetos de interesse com precisão em várias condições de iluminação e fundo como [9] demonstra ser possível a correção.

Uma das abordagens amplamente adotadas para a detecção de objetos é a segmentação de cores, que envolve a conversão do espaço de cores do vídeo para o espaço de cores HSV. Isso permite que objetos de cores específicas sejam destacados, facilitando sua detecção. Técnicas mais avançadas, como redes neurais convolucionais (CNNs), também têm sido aplicadas com sucesso para a detecção de objetos em vídeo, tornando possível a detecção de objetos com formas e tamanhos variados [10].

O rastreamento de movimento de objetos em vídeo é uma etapa subsequente importante. Após a detecção inicial de um objeto em um quadro, o sistema deve ser capaz de rastreá-lo em quadros subsequentes à medida que ele se move. O rastreamento preciso é crucial para tarefas como rastreamento de veículos em sistemas de vigilância ou seguimento de objetos em aplicações de realidade aumentada.

### 2.2.3 Remoção de Imagem Espelhada em Vídeo

A remoção de imagem espelhada em vídeos é uma tarefa importante em várias aplicações, como videoconferências e processamento de vídeos para análises. Ela envolve a correção da orientação espelhada de um vídeo para exibir uma representação correta da realidade.

O processo de remoção de imagem espelhada geralmente requer a manipulação



de cada quadro de vídeo individualmente. É comum utilizar bibliotecas como o OpenCV em conjunto com algoritmos de processamento de imagem para realizar essa tarefa.

Em artigos como [11], é abordado como identificar automaticamente a presença de espelhamento em vídeos e aplicar transformações adequadas para corrigir essa distorção. Tais métodos podem ser essenciais em situações em que a orientação correta da imagem é crucial, como na análise de dados de câmeras de segurança.

A correção de imagem espelhada é uma etapa fundamental em fluxos de trabalho de processamento de vídeo e pode melhorar significativamente a qualidade das análises e apresentações visuais.

## 2.3 Sistemas para armazenamento de dados

Os Bancos de dados desempenham um papel central no armazenamento, organização e recuperação de dados em ambientes digitais. Eles abrangem uma variedade de características e conceitos fundamentais. Estes podem seguir diferentes modelos, sendo o modelo relacional o mais comum. Outros modelos incluem hierárquico, de rede e orientado a objetos. Cada modelo define como os dados são organizados e relacionados como [12] abrange.

Os Sistemas de Gerenciamento de Banco de Dados (SGBD) são softwares que facilitam a criação, manipulação e administração de bancos de dados. Exemplos incluem MySQL, PostgreSQL e Oracle. Eles fornecem uma interface para interação com os dados e garantem a integridade e segurança. A linguagem SQL (Structured Query Language) é amplamente utilizada para interagir com bancos de dados. Permite a execução de consultas para recuperar, inserir, atualizar e excluir dados, proporcionando uma abordagem padronizada e poderosa.

A segurança é crítica em bancos de dados, principalmente em um processo produtivo. SGBDs oferecem recursos como autenticação, autorização e criptografia para proteger dados sensíveis e garantir que apenas usuários autorizados tenham acesso, garantindo assim a segurança dos dados de cada cliente.

### 2.3.1 Modelos de Banco de Dados

Os modelos de banco de dados definem a estrutura e organização dos dados em um sistema de gerenciamento de banco de dados (SGBD). Diferentes modelos oferecem abordagens distintas para armazenar e relacionar informações. A seleção do modelo de banco de dados depende das características específicas de um projeto, abaixo é falado brevemente o funcionamento de alguns modelos.

O modelo relacional é o mais amplamente adotado. Há uma organização dos dados em tabelas bidimensionais, onde as relações entre as entidades são representadas por chaves primárias e estrangeiras. Isso proporciona flexibilidade e facilita consultas complexas usando a linguagem SQL. Já no modelo hierárquico, os dados são organizados em uma estrutura de árvore, onde cada nó pode ter vários filhos. Cada nó representa um registro e a estrutura reflete as relações hierárquicas. Embora eficiente para certos casos, pode ser complexo de manter.

O modelo de rede estende o hierárquico, permitindo que um nó tenha múltiplos pais. Isso possibilita representar relações mais complexas. Apesar de oferecer mais flexibilidade que o hierárquico, pode ser difícil de entender e gerenciar gerando problemas no processo produtivo. Outro modelo que pode ser utilizado é o Modelo Orientado a Objetos onde os dados são representados como objetos, contendo atributos e métodos. Essa abordagem é eficaz para sistemas com estruturas complexas e relacionamentos dinâmicos, alinhando-se frequentemente com linguagens de programação orientadas a objetos. Por fim, o modelo NoSQL (Not Only SQL) abrange diversos subtipos, como documentos, chave-valor, coluna e grafos. Ele oferece flexibilidade para lidar com tipos variados de dados e é especialmente útil em ambientes de grande escala e necessidades de distribuição.

### 2.3.2 Integração com Banco de Dados MySQL

A integração com bancos de dados MySQL desempenha um papel fundamental em muitas aplicações de software, permitindo o armazenamento, recuperação e gerenciamento eficiente de dados. MySQL é um sistema de gerenciamento de banco de dados (DBMS) amplamente utilizado devido à sua confiabilidade, escalabilidade e recursos de código aberto.

A integração com o MySQL em aplicativos Python é geralmente realizada usando bibliotecas específicas, como o conector MySQL para Python. Essas bibliotecas fornecem uma interface de programação que permite que os desenvolvedores Python se conectem a bancos de dados MySQL, executem consultas SQL e recuperem dados de maneira programática como [13] demonstra.

A integração com bancos de dados MySQL oferece várias vantagens, incluindo:

1. **Armazenamento de Dados:** Os dados podem ser armazenados de forma estruturada e organizada em tabelas MySQL, facilitando a recuperação e análise futuras.
2. **Recuperação Eficiente:** Consultas SQL podem ser usadas para recuperar dados específicos com base em critérios personalizados, tornando a recuperação de informações eficiente.
3. **Segurança:** Os sistemas de gerenciamento de banco de dados MySQL oferecem recursos de segurança robustos, incluindo autenticação de usuários e controle de acesso.
4. **Transações:** As transações podem ser usadas para garantir a consistência dos dados, permitindo que as operações sejam agrupadas em unidades lógicas.
5. **Indexação:** Os índices podem ser criados em tabelas para acelerar a recuperação de dados, especialmente em grandes conjuntos de dados.

### 2.3.3 Exportação dos Dados para Excel

A exportação de dados para planilhas Excel é uma tarefa comum em muitos cenários, desde relatórios de negócios até análises científicas. As planilhas Excel oferecem uma maneira organizada e amplamente reconhecida de apresentar dados tabulares. Para realizar essa tarefa em um programa Python, as bibliotecas Python, como pandas e openpyxl, são frequentemente empregadas.

A biblioteca pandas oferece recursos abrangentes para manipulação de dados, incluindo a capacidade de ler, escrever e modificar planilhas Excel. Através do método `to_excel`, os dados podem ser exportados diretamente para um formato de planilha Excel.

### 2.3.4 Envio dos dados através do e-mail

O envio de e-mails por meio de programas Python é uma funcionalidade essencial em muitos aplicativos, como notificações automatizadas, comunicação com usuários e distribuição de relatórios. A biblioteca `smtplib` em Python é amplamente utilizada para estabelecer conexões com servidores de e-mail e enviar mensagens programaticamente.

A autenticação no servidor de e-mail é um passo crítico ao enviar e-mails por meio de programas Python. Isso envolve o uso de credenciais de e-mail, como nome de usuário e senha. Além disso, a configuração adequada do servidor SMTP é fundamental para o sucesso do envio de e-mails. Os livros [14] e [15] exploram as práticas recomendadas para envio seguro de e-mails usando Python.

## 2.4 Interface Gráfica dentro da indústria

Como citado anteriormente, interfaces homem-máquina (HMI) são um elemento chave no processo produtivo, que traz consigo uma gama de vantagens para o usuário, sem a necessidade de interferir manualmente no processo ou buscar algum dado fisicamente. Sendo assim, a criação dessas interfaces é de extrema importância durante a automação de um processo.

### 2.4.1 Criação de Interface Gráfica para Visualização de Dados

A criação de uma interface gráfica para visualização de dados é uma parte fundamental do processo de análise de dados e comunicação de resultados. Essas interfaces permitem que os usuários interajam com conjuntos de dados complexos de forma intuitiva, transformando dados brutos em informações significativas por meio de gráficos, tabelas e elementos visuais interativos.

Interfaces gráficas de usuário (GUIs) [16] para visualização de dados desempenham um papel crucial em diversas áreas, desde análise de negócios até pesquisa científica. Elas facilitam a exploração de padrões, tendências e insights nos dados, tornando a tomada de decisões mais informada.

A criação eficaz de uma GUI para visualização de dados envolve a seleção adequada de ferramentas e tecnologias que podem traduzir eficientemente dados em

elementos visuais compreensíveis. A literatura científica aborda amplamente esse tópico, com destaque para artigos como e , que exploram técnicas de design de interface e estratégias de interação.

### **3 DESCRIÇÃO DO PROBLEMA**

O uso de um estêncil na produção de placas eletrônicas é constante e diário, muitas fábricas chegam a produzir mais de um tipo de placa por dia 24h por dia. Entretanto, esse estêncil possui uma vida útil que dependendo da sua espessura pode variar, mas em média eles tendem a durar 100 mil impressões onde, com o decorrer do uso começam a apresentar menos consistência na aplicação da pasta de solda na placa, se tornando um risco para a produção que pode parar por problemas como insuficiência de solda na placa, ou até mesmo curtos encontrados após a refusão da solda.

Ademais, a requisição deste estêncil muitas vezes precisa ser feita com antecedência devido ao longo tempo para chegada do material no polo industrial de Manaus. Muitas vezes este material vem de São Paulo, e pode levar 1 mês para chegar após o pedido ter sido feito, isso se deve ao fato de que há uma fila para a produção destes estêncil, além do trânsito do material até o Amazonas.

O controle do uso do estêncil muitas vezes é feito manualmente, onde o operador após início e término da produção de cada produto, anota em um caderno ou sistema, e muitas vezes esse documento é esquecido ou não preenchido, fazendo com que o controle seja perdido em um processo tão crítico para o fornecimento de um produto de qualidade que atenda as especificações necessárias para cada cliente. Tendo isso em mente, para sanar este problema foi pensando na utilização de um programa para aquisição de vídeo e controle automático do processo através da análise deste vídeo, onde foi necessário primeiro entender quais problemas seriam encontrados durante este processo:

#### **3.1 Aquisição do vídeo**

Para a aquisição de vídeo, era necessário primeiro identificar qual objeto poderia ser colocado dentro da máquina de impressão em funcionamento, sem que afetasse a mesma durante seu processo e ao mesmo tempo não utilizasse nenhum dado da máquina de impressão para não ferir a garantia da mesma. Foi pensado em um celular simples, que seria de fácil acesso, não ocuparia muito espaço, seria de baixo custo para o projeto e para a bateria seria possível somente manter o mesmo conectado na tomada.

Com isso o problema da câmera seria solucionado, entretanto, seria necessário avaliar que programação seria usada e como seria conectada ao celular, como esses dados seriam recebidos e tratados.

## **3.2 Detecção do Objeto**

Para detectar o uso do estêncil para cada impressão, seria necessário achar um objeto que é constantemente utilizado e ao mesmo tempo possível de detecção através da câmera. Como o estêncil em si está a todo momento estático na máquina, foi necessário pensar em outra forma para realizar este controle.

Durante a impressão da placa eletrônica, há uma lâmina que passa por cima do estêncil espalhando a pasta de solda nas aberturas dele, sendo assim, um bom objeto para ser detectado durante a impressão das placas.

## **3.3 Remoção do espelhamento e falsas falhas**

O objeto a ser detectado foi definido, entretanto, a impressão das placas ocorres em ambos os sentidos, podemos adotar nesse caso de cima para baixo nas imagens e de baixo para cima. Seria necessário diferenciar um do outro, realizar a finalização da impressão e iniciar novamente para o próximo ciclo.

Ademais, devido o estêncil ser feito na maioria dos casos de alumínio, havia a chance de ser gerado reflexos durante a filmagem que dificultariam a detecção do objeto e seu movimento, gerando possíveis falhas.

## **3.4 Adicionando informações visuais no vídeo**

Para facilitar caso necessário a visualização das imagens por um usuário, seria preciso deixar de forma mais dinâmica o vídeo, para que fosse entendido qual processo está sendo realizado e detectado a cada quadro.

## **3.5 Gerenciamento de dados via Banco de Dados MySQL**

Optou-se por realizar o tratamento de dados via banco de dados MySQL. Essa escolha se deveu pela fácil acesso a este gerenciador, sua ampla gama de documentação e facilidade de uso.

### **3.5.1 Enviando informação ao Banco de Dados MySQL**

Para não haver necessidade de se armazenar os dados próximo a máquina em funcionamento, seria necessário a implementação de um banco de dados na rede. Com isso, seria preciso realizar a comunicação entre o programa e o banco de dados escolhido, neste caso MySQL.

### **3.5.2 Visualização de Dados no Banco de Dados MySQL**

Para facilitar a pesquisa no banco de dados, seria necessário a criação de um programa que poderia rodar em qualquer computador e fizesse a busca dos dados e apresentasse ao usuário de forma simples e eficaz, facilitando assim também a validação se a detecção de cada impressão estava realmente acontecendo.

## **3.6 Atualizações e envio das informações por e-mail na Interface**

Após a criação da interface, para sanar a necessidade de baixar os dados do banco de dados e envia-los de forma manual, há a opção de escolher o envio automático dos dados por e-mail. Mas para isso seria necessário escolher um protocolo de envio e uma conta para realizar esta função.



## 4 PROPOSTA DE SOLUÇÃO

Para sanar o problema do controle e uso do estêncil para cada produto, a proposta deste trabalho é criar um programa para automatizar estes processos e armazenar na nuvem as informações do seu uso para não gerar a necessidade de manter um processo manual constante na fábrica. Há aplicações atuais que utilizam a própria máquina que realiza a impressão desta placa para realizar esta contagem, entretanto, ela utiliza os sinais digitais do Circuito lógico Programado que possui, com os direitos autorais de cada fornecedor, para realizar este processo.

Ademais, para não haver a necessidade de se checar os dados armazenados constantemente, foi pensado na proposta de se automatizar também o envio destas informações por e-mail quando necessário para pessoas específicas. Abaixo é demonstrado a solução para cada problema comentado anteriormente.

### 4.1 Aquisição do vídeo

Para a aquisição do vídeo estudado foi utilizada uma câmera ultra-wide com 8 MP, do celular Android A32. Outras câmeras podem ser utilizadas, uma vez que optou-se por utilizar os vídeos não compactados no formato raw, em um empacotamento mp4. É possível realizar a conexão com a câmera do celular Android através de código no python tendo o endereço de IP do mesmo, onde programas como DroidCam conseguem fazer essa hospedagem, entretanto, para fins de estudo foram tirados vídeos do processo para serem analisados.

Diferentes opções de compressão e transporte de vídeo permitem balancear a carga de CPU e o armazenamento, isto é, o espaço ocupado pelo vídeo, e a velocidade de leitura/gravação. Isto impacta diretamente no desempenho do sistema. A Tabela 1 mostra uma comparação entre os formatos mais comuns de compactação de vídeo com o formato raw.

Em nosso caso, optamos por não utilizar nenhum método de compressão para que não houvesse perda da qualidade da imagem a ser analisada. Sendo assim, obteve-se os dados não comprimidos ocasionando na menor carga de CPU, e a maior carga de

<b>Critério</b>	<b>Vídeo RAW</b>	<b>H.264 (AVC)</b>	<b>H.265 (HEVC)</b>
Qualidade	Excelente	Boa	Excelente
Tamanho do arquivo	Enorme	Pequeno	Pequeno
Compatibilidade	Baixa	Alta	Moderada
Uso de CPU/GPU	Alto	Moderado	Moderado
Taxa de bits (bitrate)	Alta	Média	Média
Perda de qualidade	Nenhuma	Compressão perceptível	Compressão perceptível
Aplicações comuns	Edição profissional, masterização	Streaming, Blu-ray	Streaming 4K/8K, Blu-ray

<b>Critério</b>	<b>VP9</b>	<b>AV1</b>	<b>MP4</b>
Qualidade	Boa	Excelente	Boa
Tamanho do arquivo	Pequeno	Pequeno	Moderado
Compatibilidade	Moderada	Moderada	Alta
Uso de CPU/GPU	Moderado	Moderado	Moderado
Taxa de bits (bitrate)	Média	Média	Média
Perda de qualidade	Compressão perceptível	Compressão perceptível	Compressão perceptível
Aplicações comuns	Streaming, vídeo online	Streaming, vídeo online	Streaming, vídeo online

**Tabela 1 – Comparação entre formatos de compactação de vídeo e vídeo raw.**

armazenamento o que poderia levar a uma sobrecarga no disco rígido, entretanto, a implementação cuida de descartar vídeos já tratados e analisados, fazendo assim com que o disco rígido do sistema não fique cheio.

Uma vez captado o vídeo e extraído no celular, é feito o upload do mesmo para o computador que irá realizar o tratamento de dados. Dentro desse computador, há a implementação que irá realizar a detecção do movimento da lâmina em cima da máscara de solda, somente quando uma PCB estiver pronta para ser printada. Após a printagem, a contagem é realizada e armazenada no sistema, e quando esse valor está próximo de atingir o limite de printagens indicado pelo fornecedor da máscara de solda, o sistema alerta o Engenheiro do processo. A Figura 2 descreve o fluxograma de operações após a captação do vídeo.

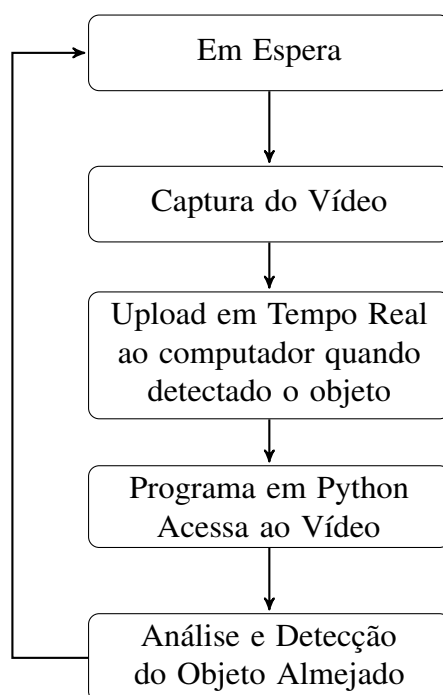


Figura 2 – Fluxograma do Processo de Análise de Vídeo

## 4.2 Detecção do Objeto

O código começa importando as bibliotecas necessárias, como o OpenCV para processamento de vídeo e o NumPy para manipulação de arrays. A função `draw_red_board` é definida para desenhar caixas delimitadoras vermelhas ao redor dos objetos verdes detectados.

A função principal `process_video` realiza todo o processamento do vídeo. Ela recebe o caminho para o arquivo de vídeo a ser processado e um tamanho de saída desejado para os quadros do vídeo. O vídeo é aberto usando o OpenCV e um loop é iniciado para processar cada quadro.

Cada quadro é lido e redimensionado para o tamanho de saída desejado. Em seguida, o quadro é convertido para o espaço de cores HSV, que facilita a detecção de cores específicas. Uma máscara é criada para identificar os pixels verdes no quadro usando limites de cor pré-definidos.

Os contornos são encontrados na máscara usando a função `cv2.findContours`.

Contornos com áreas superiores a 1000 pixels são considerados e uma caixa delimitadora vermelha é desenhada ao redor dessas áreas usando a função `draw_red_board`.

O quadro resultante, com as caixas delimitadoras desenhadas, é exibido em tempo real em uma janela. O loop continua até que o usuário pressione a tecla 'q', momento em que o vídeo é encerrado e a janela é fechada como pode se observar no código abaixo:

```
1 import cv2
2 import numpy as np
3
4 def draw_red_board(image, x, y, w, h):
5     cv2.rectangle(image, (x, y), (x + w, y + h), (0, 0,
6         255), 2)
7
8 def process_video(video_path, output_size=(640, 480)):
9     cap = cv2.VideoCapture(video_path)
10
11     while cap.isOpened():
12         ret, frame = cap.read()
13         if not ret:
14             break
15
16         frame = cv2.resize(frame, output_size)
17
18         hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
19
20         lower_green = np.array([40, 40, 40])
21         upper_green = np.array([80, 255, 255])
22
23         mask = cv2.inRange(hsv, lower_green, upper_green)
24
25         contours, _ = cv2.findContours(mask,
26             cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
27
28         for contour in contours:
29             if cv2.contourArea(contour) > 1000:
```

```
28         x, y, w, h = cv2.boundingRect(contour)
29         draw_red_board(frame, x, y, w, h)
30     cv2.imshow("Video com Caixa Delimitadora Vermelha",
31               frame)
32
33     if cv2.waitKey(30) & 0xFF == ord('q'):
34         break
35
36     cap.release()
37     cv2.destroyAllWindows()
38
39 if __name__ == "__main__":
40     video_path = "path/to/your/video.mp4"
41     output_size = (640, 480)
42     process_video(video_path, output_size)
```

**Código 4.1 – Código inicial Detecção de Objeto**

### 4.3 Remoção do espelhamento e falsas falhas

A função `draw_red_board` foi alterada para desenhar uma caixa delimitadora vermelha ao redor do objeto desejado, onde é possível também ver as coordenadas do canto superior esquerdo da caixa com os seguintes ações de cada parâmetro:

1. Desenha uma caixa delimitadora vermelha em torno do objeto detectado usando a função `cv2.rectangle`.
2. Adiciona um texto formatado que exibe as coordenadas (x, y) do canto superior esquerdo da caixa delimitadora ao quadro usando a função `cv2.putText`.
3. Em seguida é adicionado um nome a cada objeto detectado para facilitar o controle.

```
1 def draw_red_board(image, x, y, w, h):
2     cv2.rectangle(image, (x, y), (x + w, y + h), (0, 0,
3         255), 2)
```

```
3     text = f"({x}, {y})"  
4     cv2.putText(image, text, (x, y - 10),  
                 cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1)
```

#### Código 4.2 – Código inicial Função Draw Red Board

Facilitando assim a análise, devido ao espelhamento e falsas falhas, para ajustar a detecção do objeto almejado no vídeo. Com isso, foi possível adicionar as seguintes condições:

```
1 highest_y = output_size[1]  
2 top_box_coords = None  
3  
4 for contour in contours:  
5     if cv2.contourArea(contour) > 1000:  
6         x, y, w, h = cv2.boundingRect(contour)  
7         if y < highest_y:  
8             highest_y = y  
9             top_box_coords = (x, y, w, h)  
10  
11 if top_box_coords is not None:  
12     x, y, w, h = top_box_coords  
13     draw_red_board(frame, x, y, w, h)
```

#### Código 4.3 – Código de condições para detecção do objeto

`highest_y = output_size[1]`: Inicializa uma variável `highest_y` com a altura do quadro de saída. Isso é feito para garantir que, no início, o valor de `highest_y` seja maior do que qualquer valor `y` possível para as coordenadas das caixas delimitadoras. Isso será usado para rastrear a caixa delimitadora mais alta no quadro.

`top_box_coords = None`: Inicializa uma variável `top_box_coords` como `None`. Essa variável será usada para armazenar as coordenadas da caixa delimitadora mais alta encontrada no quadro.

Loop de Contornos: O código continua a percorrer todos os contornos encontrados na máscara (regiões verdes).

if `y < highest_y`: Esta condição verifica se a coordenada `y` do canto superior esquerdo da caixa delimitadora atual é menor do que `highest_y`, ou seja, se a caixa delimitadora atual está mais alta no quadro do que qualquer caixa delimitadora anteriormente rastreada.

Atualização de `highest_y` e `top_box_coords`: Se a condição for verdadeira, isso significa que a caixa delimitadora atual é a mais alta até agora. Portanto, `highest_y` é atualizado com o valor `y` da caixa atual, e `top_box_coords` é atualizado com as coordenadas da caixa delimitadora atual.

if `top_box_coords is not None`: Após o loop de contornos, verifica-se se `top_box_coords` foi atualizado (ou seja, se uma caixa delimitadora mais alta foi encontrada no quadro).

Desenho da Caixa Delimitadora Mais Alta: Se `top_box_coords` não for vazio, isso significa que uma caixa delimitadora mais alta foi encontrada. O código obtém as coordenadas dessa caixa e chama a função `draw_red_board` para desenhar uma caixa vermelha em torno dela. Ainda sim, algumas cores verdes pequenas no vídeo estavam sendo detectadas, sendo assim, a função do delimitador vermelho foi adicionado a condição de que a largura do objeto detectado precisaria ser maior que 100 pixels.

## 4.4 Adicionando informações visuais no vídeo

Para facilitar a visualização do usuário caso necessário a análise por uma pessoa, o código foi aprimorado e modificado com várias adições e ajustes significativos:

1. **Variáveis ‘printing’ e ‘printed’**: Duas novas variáveis, ‘printing’ e ‘printed’, foram adicionadas para rastrear o estado de impressão do objeto. ‘printing’ indica se o objeto está atualmente em processo de impressão, enquanto ‘printed’ indica se a impressão foi concluída.
2. **Comentários no Texto**: Os comentários relacionados à exibição de texto foram modificados. Em vez de exibir as coordenadas e o tamanho da caixa delimitadora, o código agora exibe mensagens mais informativas, como "Status: Printing" e "Status: Printed", com base no estado das variáveis ‘printing’ e ‘printed’.

3. **Condições para Atualizar ‘printing’ e ‘printed’:** Duas novas condições foram adicionadas após a detecção da caixa delimitadora:
  - Se a coordenada ‘y’ da caixa delimitadora for maior que 100 e ‘printing’ for ‘False’, a variável ‘printing’ é definida como ‘True’. Isso indica que o processo de impressão está em andamento.
  - Se a coordenada ‘y’ da caixa delimitadora for menor que 40 e ‘printing’ for ‘True’, a variável ‘printed’ é definida como ‘True’. Isso indica que a impressão foi concluída.
4. **Mensagens de Status:** As mensagens exibidas no canto inferior direito do vídeo agora variam com base nos estados de ‘printing’ e ‘printed’. Se ‘printing’ for ‘True’ e ‘printed’ for ‘False’, a mensagem exibida será "Status: Printing". Se ‘printed’ for ‘True’, a mensagem será "Status: Printed". Caso contrário, a mensagem será "Waiting".

Essas modificações permitem um rastreamento mais preciso do processo de impressão em tempo real e a exibição de mensagens informativas no vídeo.

## 4.5 Enviando informação ao Banco de Dados MySQL

Foi realizada algumas mudanças no código para ser possível se comunicar com um banco de dados em MySQL utilizando o próprio Python, as mudanças mais significativas foram:

1. **Integração com o MySQL:** - O novo código inclui integração com o MySQL para armazenar informações sobre o status da impressão em uma tabela de banco de dados.
2. **Conexão com o MySQL:** - Uma conexão com um banco de dados MySQL é estabelecida usando os seguintes parâmetros: host, usuário, senha, nome do banco de dados e porta.



3. **Criação da Tabela:** - Uma nova tabela chamada 'printing\_info' é criada no banco de dados MySQL se ela ainda não existir. A tabela possui três colunas: 'id', 'status' e 'timestamp'.
4. **Função 'insert\_data':** - Uma nova função chamada 'insert\_data' é definida para inserir dados na tabela 'printing\_info'. Ela recebe um cursor e um status como parâmetros e insere o status juntamente com um carimbo de data e hora.
5. **Inserção de Dados:** - No interior do loop de processamento de vídeo, quando a caixa vermelha passa abaixo de um certo limite ( $y < 40$ ) e a impressão está em andamento ('printing' é True) e ainda não foi marcada como impressa ('printed' é False), o código insere o status "Impresso" no banco de dados usando a função 'insert\_data'. A flag 'data\_inserted' evita inserções duplicadas.
6. **Novas Flags:** - Duas novas flags, 'data\_inserted' e 'printed', são introduzidas para rastrear a inserção de dados no banco de dados e o status de impressão.
7. **Limpeza do MySQL:** - A conexão com o MySQL é devidamente encerrada usando 'connection.close()' no final do script para garantir uma saída limpa.

Essas mudanças permitem que o código registre e gerencie o status de impressão em um banco de dados MySQL, proporcionando uma solução mais abrangente de rastreamento e armazenamento em tempo real.

## 4.6 Visualização de Dados no Banco de Dados MySQL

Após o envio das informações ao banco de dados, foi criado um código para a visualização dos dados armazenados utilizando a biblioteca 'tkinter' para criar uma interface gráfica simples para exibir os registros da tabela, onde foram criadas as funções get\_data, display\_data usando a interface do tkinter.

A função 'get\_data()' é responsável por recuperar os dados da tabela 'printing\_info' no banco de dados MySQL. Ela realiza as seguintes etapas:

1. Estabelece uma conexão com o banco de dados MySQL usando os parâmetros de host, usuário, senha, nome do banco de dados e porta.

2. Cria um cursor para executar consultas SQL.
3. Executa uma consulta SQL que seleciona todos os registros da tabela 'printing\_info'.
4. Recupera os dados da consulta usando 'cursor.fetchall()'.
5. Fecha a conexão com o banco de dados.
6. Retorna os dados recuperados.

A função 'display\_data()' é responsável por exibir os dados recuperados da tabela 'printing\_info' na interface gráfica. Ela realiza as seguintes etapas:

1. Chama a função 'get\_data()' para obter os dados da tabela.
2. Limpa todos os registros existentes na visualização ('tk.Treeview') para evitar duplicatas.
3. Itera pelos registros recuperados e insere cada registro na visualização. Cada registro é exibido em uma linha com três colunas: ID, Status e Timestamp.

O código utiliza a biblioteca 'tkinter' para criar uma janela de interface gráfica. Aqui estão os principais elementos da interface gráfica:

- 'tk.Treeview': Um componente que exibe os dados da tabela em forma de grade. Possui três colunas: ID, Status e Timestamp.
- 'tk.Scrollbar': Uma barra de rolagem vertical que permite navegar pelos registros da tabela.
- 'tk.Button': Um botão "Refresh Data" que, quando clicado, chama a função 'display\_data()' para atualizar os dados exibidos.

No final do código, a função 'display\_data()' é chamada para exibir os dados iniciais da tabela quando o programa é iniciado. Já o loop principal ('root.mainloop()') é necessário para manter a janela da interface gráfica em execução e responder a eventos

do usuário, como cliques de botão. Este código cria uma interface simples e interativa para visualizar os dados da tabela 'printing\_info' em um banco de dados MySQL.

## 4.7 Atualizações e envio das informações por e-mail na Interface

Após a criação da interface base, foram realizadas adições que permitem que o usuário exporte os dados para um arquivo Excel ou envie-os por e-mail diretamente da interface utilizando as funções `export_to_excel` e `send_email` criadas no código.

A função `'export_to_excel()'` foi adicionada para permitir a exportação dos dados da tabela para um arquivo Excel. Ela realiza o seguinte:

1. Chama a função `'get_data()'` para obter os dados da tabela.
2. Cria um DataFrame do Pandas com os dados.
3. Abre uma caixa de diálogo para escolher o local e o nome do arquivo Excel a ser salvo.
4. Se um local e nome de arquivo forem selecionados, os dados serão salvos no arquivo Excel nesse local.

A função `'send_email()'` foi adicionada para permitir o envio dos dados da tabela por e-mail como um anexo em formato Excel. Ela realiza o seguinte:

1. Solicita o endereço de e-mail do destinatário por meio de uma caixa de diálogo.
2. Chama a função `'get_data()'` para obter os dados da tabela.
3. Cria um DataFrame do Pandas com os dados.
4. Salva os dados em um arquivo Excel chamado "printing\_info.xlsx".
5. Configura detalhes do e-mail, como assunto, corpo, remetente e senha do remetente.

6. Anexa o arquivo Excel ao e-mail.
7. Usa o servidor SMTP (smtp.gmail.com) para enviar o e-mail.

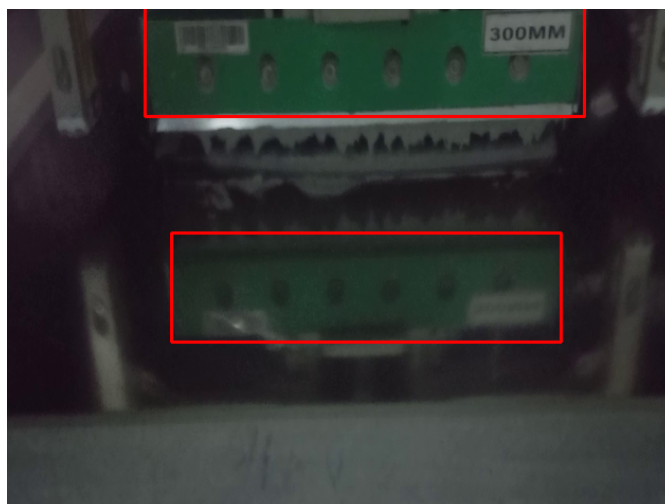
Foram adicionados dois botões à interface para utilização dessas funções:

- ‘btn\_export’: Um botão "Export to Excel" que, quando clicado, chama a função ‘export\_to\_excel()’ para exportar os dados para um arquivo Excel.
- ‘btn\_send\_email’: Um botão "Send Email" que, quando clicado, chama a função ‘send\_email()’ para enviar os dados por e-mail como anexo.

## 5 RESULTADOS

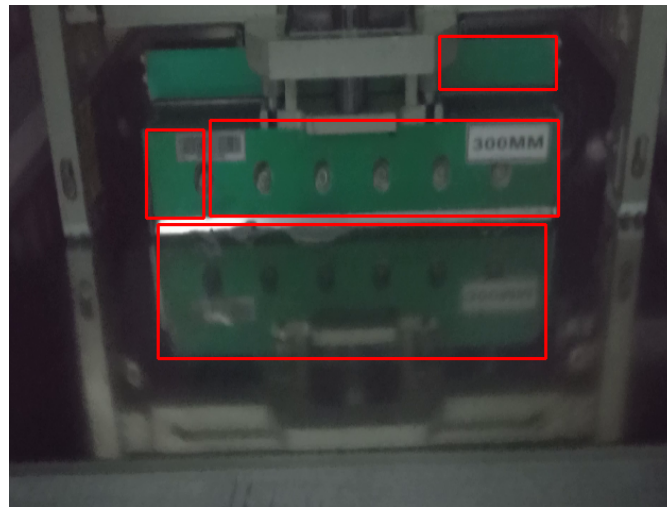
### 5.1 Detecção do Objeto

Inicialmente, ao testar o código, nos primeiros segundos do vídeo já foi possível identificar o problema de espelhamento que era gerado devido ao material do estêncil (Stencil). Mas a princípio, o objeto alvejado estava sendo reconhecido. Para facilitar a análise de todo o projeto, foi desenhado o quadrado vermelho em volta do objeto a ser detectado, além de outras informações futuras que serão demonstradas com mais detalhes em cada sessão.



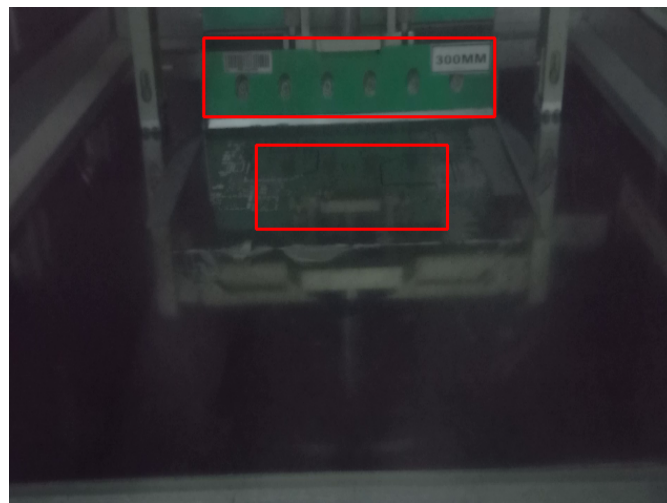
**Figura 3 – Detecção do Objeto Inicial - Pré impressão da placa.**

Em seguida foi possível identificar a detecção de vários objetos durante o vídeo, e como as imagens abaixo demonstram, não seria possível identificar mais no vídeo e nem ao nível de código o objeto almejado, devido à perda do monitoramento.



**Figura 4 – Detecção do Objeto Inicial - Impressão da placa.**

Ao final do vídeo, é possível observar no final da impressão, novamente, que o espelhamento é um problema para realizar o monitoramento das impressões que estão sendo realizadas.



**Figura 5 – Detecção do Objeto Inicial - Pós impressão da placa.**

## **5.2 Remoção do espelhamento e falsas falhas**

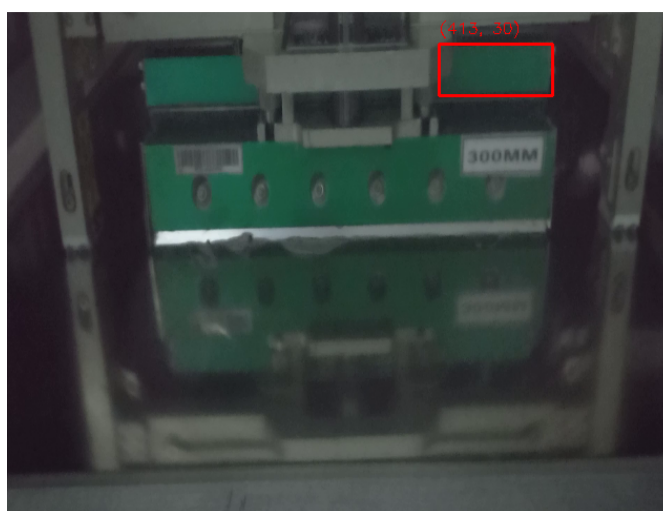
Os problemas encontrados ao rodar o código pela primeira vez deixaram alguns objetivos principais a serem corrigidos, então depois de algumas correções foi possível

alcançar o seguinte resultado:



**Figura 6 – Remoção do Espelhamento - Pré impressão da placa.**

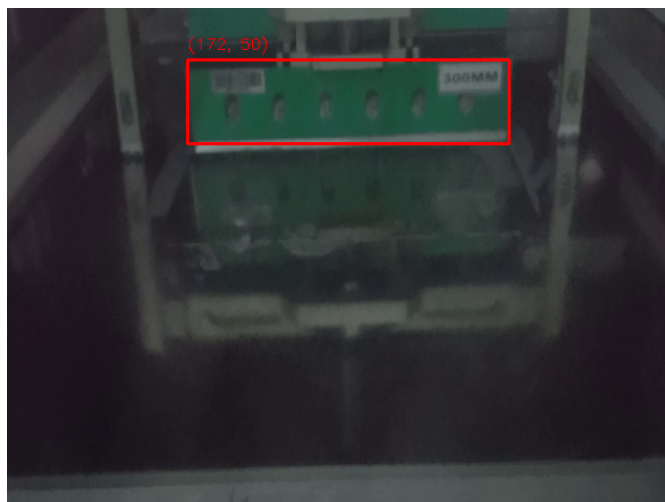
Os primeiros segundos do vídeo demonstraram um sucesso na remoção do espelhamento, entretanto, logo em seguida durante a impressão da placa, foi possível observar que o objeto almejado deixava de ser identificado e o código identificava uma parte ver acima do mesmo.



**Figura 7 – Remoção do Espelhamento - Impressão da placa.**

Como é possível observar, para facilitar a análise do que deveria ser corrigido, foi

adicionado a informação do posicionamento do quadrado em vermelho sendo desenhado em volta de cada objeto. Ao final da impressão, novamente foi possível observar que o espelhamento foi corrigido, e que somente o objeto almejado estava sendo identificado.

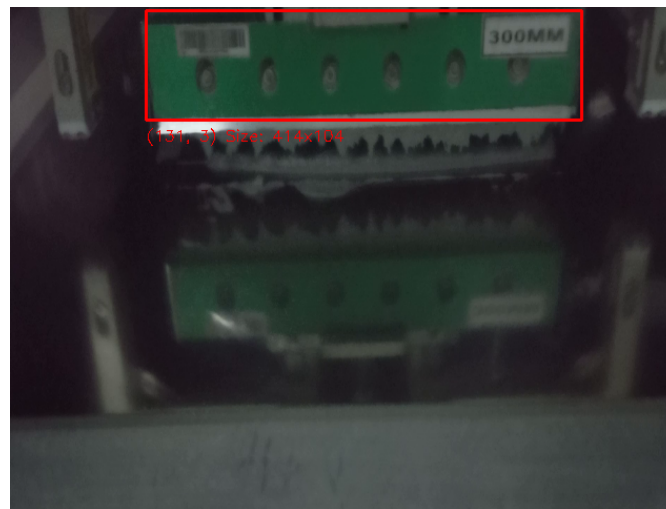


**Figura 8 – Remoção do Espelhamento - Pós impressão da placa.**

### **5.3 Adicionando informações visuais no vídeo**

Seguindo para o programa seguinte, seria necessário corrigir o problema do ponto verde sendo identificado como objeto almejado, para ter certeza de que o objeto não estava sendo identificado, foi adicionado também o tamanho de cada quadrado vermelho em volta de cada objeto, para garantir que a detecção correta estaria acontecendo.

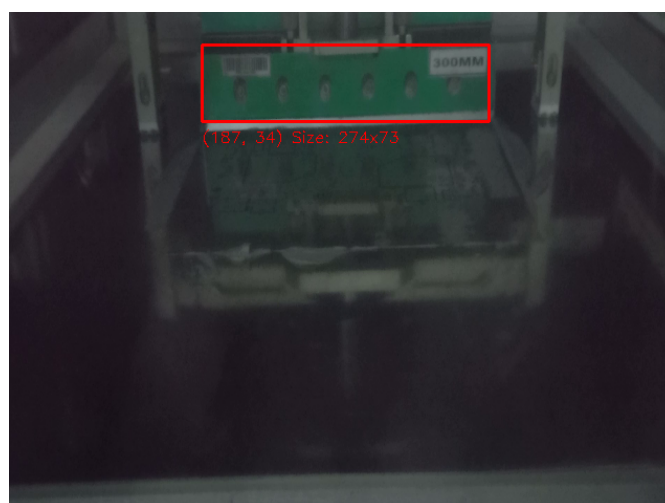




**Figura 9 – Adicionando informações sobre o objeto detectado - Pré impressão da placa.**



**Figura 10 – Adicionando informações sobre o objeto detectado - Impressão da placa.**



**Figura 11 – Adicionando informações sobre o objeto detectado - Pós impressão da placa.**

Como é possível observar nas 3 imagens acima, não houve erro de espelhamento e o objeto almejado foi identificado durante o processo de impressão da placa. Sendo assim, para garantir que este processo estava acontecendo e estava finalizando no tempo correto, foram adicionados as informações de Imprimindo e Impresso no canto inferior direito do vídeo.



**Figura 12 – Adicionando informações de impressão - Impressão da placa.**

A imagem acima demonstra o processo no meio da impressão, e a imagem abaixo demonstra a finalização da impressão, com as variáveis de controle funcionando

corretamente.

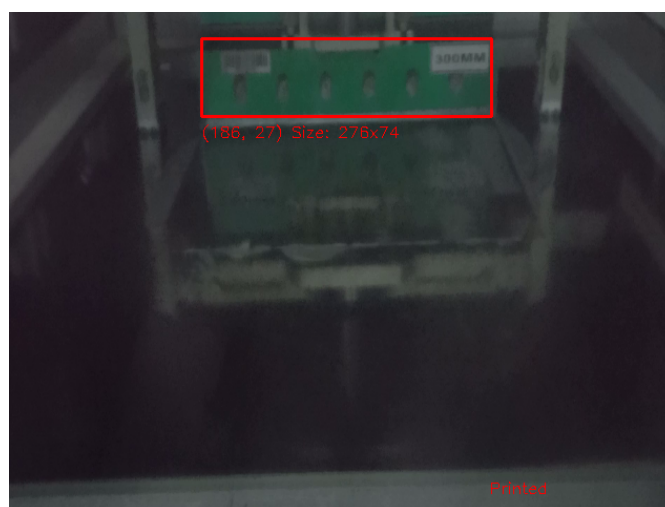
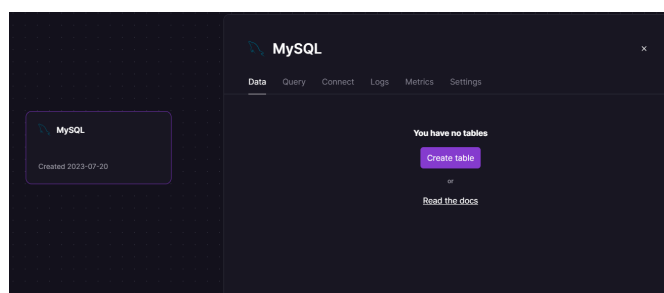


Figura 13 – Adicionando informações de impressão - Pós impressão da placa.

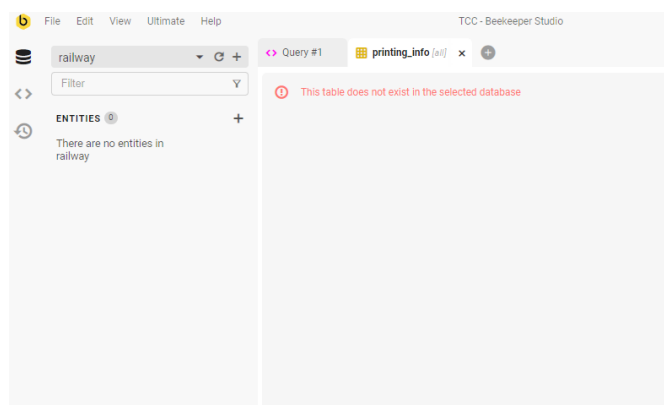
## 5.4 Envio e Visualização de Dados no Banco de Dados MySQL

O programa de detecção de impressão do objeto agora está concluído, com as variáveis de controle da impressão funcionando, foi possível iniciar o envio e controle dos dados a um Banco de Dados na nuvem. Foi escolhido para o projeto utilizar o MySQL, para facilitar a hospedagem na nuvem e realizar o teste de comunicação, foi utilizado o site railway. Tendo o servidor para hospedar o banco de dados, foi feita a inclusão no código para o envio e comunicação da informação de forma automática durante o processamento do vídeo, ou seja, o processamento dos dados acontece em paralelo com o processamento do vídeo. O início do programa é o mesmo, não houve alteração nessa parte.

Durante o processo de impressão, foi adicionado a função para caso não haja uma tabela no banco de dados, seja criada automaticamente uma tabela, onde seria capaz de armazenar mais de uma tabela, para armazenamento dos dados. As imagens abaixo demonstram exatamente o momento que a tabela ainda não havia sido criada.

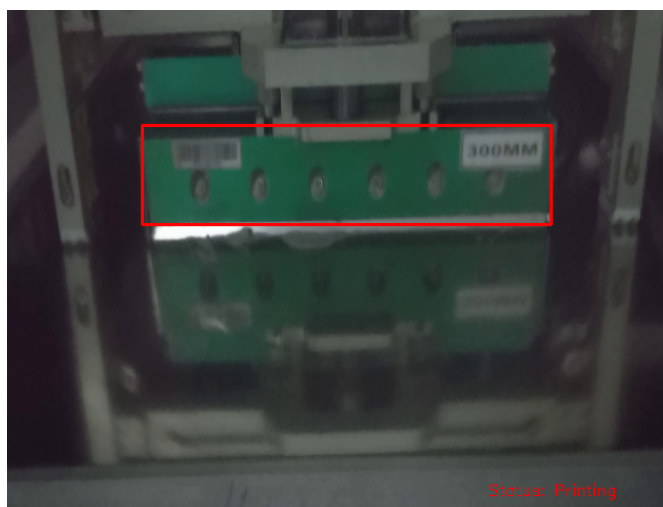


**Figura 14 – Banco de dados sem nenhuma tabela criada.**

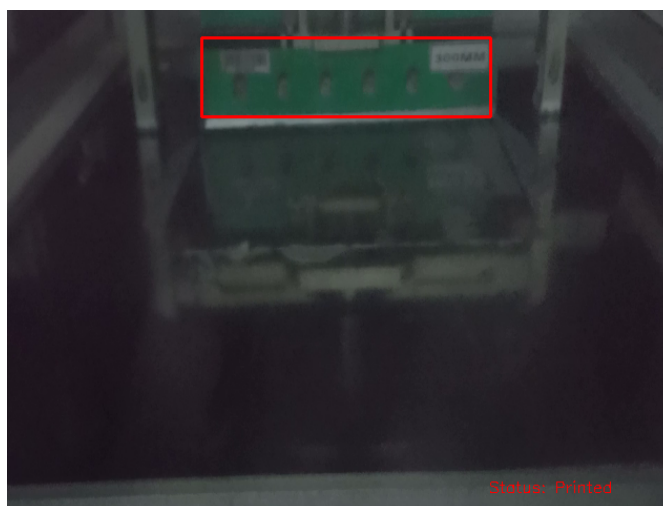


**Figura 15 – Busca no banco de dados sem nenhuma tabela criada.**

Assim que a impressão acontece, seguido da finalização, os dados são enviados para a tabela criada, como pode se observar nas imagens abaixo. Os dados enviados com sucesso para o banco de dados, e utilizando um programa de visualização de banco de dados, é possível confirmar que o dado realmente foi criado e armazenado.



**Figura 16 – Código com o banco de dados aplicado - Impressão da placa.**



**Figura 17 – Código com o banco de dados aplicado - Pós impressão da placa.**

Id	status	timestamp
1	Printed	2023-07-25 15:11:21

**Figura 18 – Banco de dados após finalização do processamento do vídeo.**

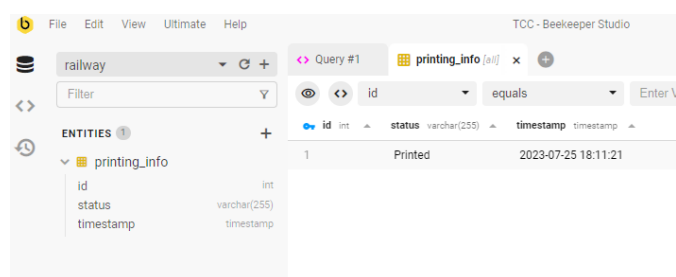


Figura 19 – Busca no banco de dados após finalização do processamento do vídeo.

## 5.5 Atualizações e envio das informações por e-mail na Interface

Com o dado armazenado no banco de dados, e sabendo que é possível ter acesso a essa informação, foi criado um programa secundário, que poderia estar localizado em qualquer computador do usuário, onde é possível visualizar todos os dados no banco de dados sem um programa de terceiro.

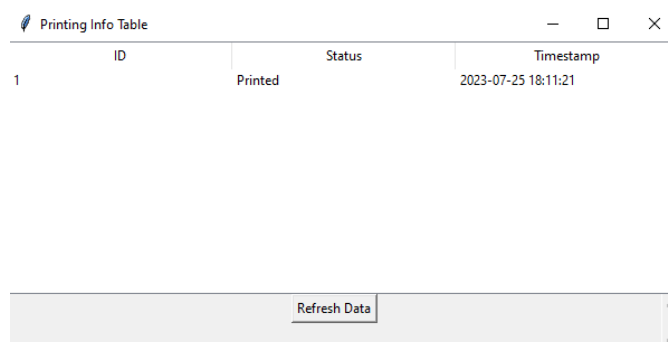
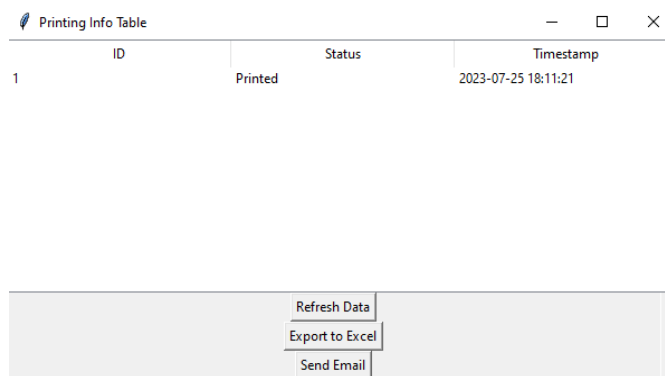
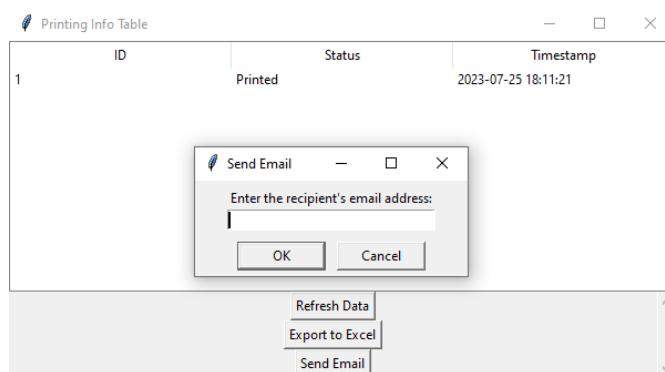


Figura 20 – Interface inicial para visualização do banco de dados.

Para finalizar, foi criado uma forma de transformar esses dados em uma tabela excel, caso seja necessária alguma análise manual. Ademais, com o intuito de automatizar o processo, foi adicionado um botão para enviar e-mail automaticamente com um excel do banco de dados.



**Figura 21 – Interface com a adição de botões para Excel e E-mail.**



**Figura 22 – Interface para escolha do e-mail destinatário.**

Para questão de demonstração, esse botão foi criado e foi adicionado a opção de colocar um e-mail destinatário. Mas o intuito é que o projeto aplicado já possua alguns e-mails específicos que deve mandar os dados, quando uma variável almejada seja alcançada a pedido do cliente.

## 6 CONCLUSÃO

Foi explorado uma variedade de aplicações práticas para visão computacional e integração de dados usando Python. Começando com a detecção de objetos verdes em fluxos de vídeo e rastreando seus movimentos, foi desenvolvido um sistema robusto que pode monitorar o processo de impressão em tempo real e registrar os resultados em um banco de dados MySQL.

A capacidade do projeto foi expandida implementando o processamento em lote de vídeos, permitindo a análise de múltiplos arquivos de vídeo em um diretório específico. Isso demonstra a escalabilidade do sistema para lidar eficientemente com grandes conjuntos de dados. Ademais, criou-se uma interface gráfica amigável para visualizar os dados coletados e exportamos para o Excel para análises adicionais. Adicionalmente, foi integrado a funcionalidade de e-mail para enviar os dados como anexos, tornando mais fácil compartilhar e coletar os dados analisados a distância.

Em última análise, foi abordado o desafio da remoção de imagens espelhadas em vídeos, aprimorando a precisão na detecção e rastreamento de objetos. Essas soluções baseadas em Python oferecem um conjunto versátil de ferramentas para pesquisadores, engenheiros e analistas de dados que trabalham com dados de vídeo, bancos de dados e interfaces gráficas. Com refinamentos e personalizações adicionais, esse sistema pode ser adaptado para uma ampla variedade de aplicações, desde controle de qualidade na fabricação até vigilância e além.



## REFERÊNCIAS

- T. Sadhukhan, S. Chatterjee, R. K. Das, S. Ghosh, D. Das, and J. Sanyal, “Efficient removal of noise from an image using hsv filtering,” in 2019 Global Conference for Advancement in Technology (GCAT), 2019, pp. 1–4.
- E. Şipoş, A. Oneş, and L.-N. Ivanciu, “Pcb quality check: Optical inspection using color mask and thresholding,” in 2022 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), 2022, pp. 1–5.
- R. Drath and A. Horch, “Industrie 4.0: Hit or hype? [industry forum],” Industrial Electronics Magazine, IEEE, vol. 8, pp. 56–58, 06 2014.
- M. Cotteleer and B. Sniderman. (2017) Forces of change: Industry 4.0. [Online]. Available: <https://www2.deloitte.com/us/en/insights/focus/industry-4-0/overview.html>
- M. Bahrin, F. Othman, N. Azli, and M. Talib, “Industry 4.0: A review on industrial automation and robotic,” Jurnal Teknologi, vol. 78, 06 2016.
- D. Perdios, M. Vonlanthen, F. Martinez, M. Arditi, and J.-P. Thiran, “Cnn-based image reconstruction method for ultrafast ultrasound imaging,” IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 69, no. 4, pp. 1154–1168, 2022.
- M. Al-Qawasmi, N. Mekhiel, K. Bannor, and B. C. Devakumar, “Batch image processing in facial detection applications,” in 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), 2021, pp. 475–480.
- W. Liu, Y. Xue, Y. Tang, and A. Sun, “A self-regularized low-light image enhancement based on the brightness information,” in 2022 4th International Conference on Natural Language Processing (ICNLP), 2022, pp. 119–124.
- S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” 2015.
- R. FU, P. KUANG, Y. ZHOU, H.-R. YAN, and T.-Y. ZHENG, “Area-aware reflection detection and removal for single image,” in 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, 2019, pp. 307–310.

G. Ongo and G. P. Kusuma, “Hybrid database system of mysql and mongodb in web application development,” in 2018 International Conference on Information Management and Technology (ICIMTech), 2018, pp. 256–260.

D. L. Yu, “Python database programming: Sqlite (step-by-step),” Medium, 2021. [Online]. Available: <<https://medium.com/swlh/python-database-programming-sqlite-step-by-step-828ebf0bb2e>>

K. Markham, Manipulação de Dados com Python e Pandas: Um Guia Prático. Independently published, 2018.

A. Sweigart, Automate the Boring Stuff with Python. No Starch Press, 2015.

B. Chaudhary, Tkinter: GUI Application Development Blueprints, 2015.