UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

Stefany Vitória da Conceição Izidio

Desenvolvimento de um Sistema Auxiliar para Controle de Acesso de Veículos para a Universidade Federal Rural de Pernambuco

Recife

Março de 2024

Stefany Vitória da Conceição Izidio

Desenvolvimento de um Sistema Auxiliar para Controle de Acesso de Veículos para a Universidade Federal Rural de Pernambuco

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado em: 08 de março de 2024.

BANCA EXAMINADORA

Cícero Garrozi (Orientador)
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Victor Wanderley Costa de Medeiros
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Recife
Março de 2024

Agradecimentos

Primeiramente, agradeço a Deus pela capacidade de chegar até aqui. Agradeço também à minha família, que sempre acreditou em mim. Em especial, dona Jaqueline, minha mãe, que sempre lutou para que isso fosse possível e nunca mediu esforços para me ajudar. Agradeço também ao meu namorado, Agamenon, que esteve ao meu lado nesses 5 anos e nunca deixou que eu me desmotivasse. Agradeço também a todos os meus colegas e amigos do curso, em especial a Rodrigo, Linaldo e João Victor. Obrigada pelo companheirismo, paciência, risadas e trabalhos em grupo. Por fim, agradeço a todo o corpo docente do BSI pelos conhecimentos compartilhados, em especial ao meu orientador, Cícero. Todos foram essenciais para a minha formação.

Desenvolvimento de um Sistema Auxiliar para Controle de Acesso de Veículos para a Universidade Federal Rural de Pernambuco

Stefany Izidio[1], Cícero Garrozi[2].

[1] Bacharelado em Sistemas de Informação - Universidade Federal Rural de Pernambuco
[2] Departamento de Estatística e Informática - Universidade Federal Rural de Pernambuco
Rua Dom Manuel de Medeiros, s/n, - CEP: 52171-900 – Recife – PE – Brasil

izidiostefany@gmail.com, cicero.garrozi@ufrpe.br.

Resumo. Atualmente, o controle de acesso de veículos à Universidade Federal Rural de Pernambuco é feito manualmente em papéis por funcionários da universidade. Também há a liberação direta de veículos que se cadastram na universidade e recebem um adesivo específico para usar no para-brisa. Este tipo de controle não é muito seguro, por poder ser facilmente clonado e usado por veículos sem autorização real. Além disso, há um curto desvio de atenção do funcionário quando ele realiza o trabalho manual de anotar a placa no papel. Este trabalho tem o objetivo de tornar o processo de controle de veículos mais confiável e seguro através do desenvolvimento de um protótipo de um sistema que auxilia no controle de acesso. Este trabalho propõe uma solução mediante a captura de uma imagem da placa, identificação da placa do veículo e da verificação em uma base de dados se a placa é previamente cadastrada ou não. E, o sistema produz um sinal luminoso para indicar ao funcionário se a placa é ou não cadastrada. Para isso, foi montado um produto de hardware e desenvolvido um software embutido. O hardware é composto por um conjunto de dispositivos eletrônicos como LEDs, câmera, dispositivo de processamento, etc. O software é um conjunto de bibliotecas que foi, em sua maior parte, desenvolvido em Python. Para o software embutido, foi usado um conjunto de imagens com fotos de placas de carros brasileiros para treinar um modelo de detecção de objetos para detectar as placas. Por fim, foi utilizado um serviço de reconhecimento ótico de caracteres para extrair o conteúdo da placa, possibilitando assim registrar e emitir o sinal luminoso ao usuário.

Palavras-Chave: aprendizagem de máquina; detecção de objetos; detecção de placas; reconhecimento óptico de caracteres; visão computacional; inteligência artificial.

Abstract. Currently, vehicle access control to the Federal Rural University of Pernambuco is done manually on paper by university employees. There is also direct release for vehicles that register with the university and receive a specific sticker to use on the windshield. This type of control is not very safe, as it can be easily cloned and used by vehicles without real authorization. Furthermore, there is a short diversion of the employee's attention when he performs the manual work of writing down the sign on paper. This work aims to make the vehicle control process more reliable and safe through the development of a prototype of a system that assists in access control. This work proposes a solution by capturing an image of the license plate, identifying the vehicle plate and checking in a database whether the plate is previously registered or not. And, the system produces a light signal to indicate to the employee whether the license plate is registered or not. To achieve this, a hardware product was assembled and embedded software was developed. The hardware consists of a set of electronic devices such as LEDs, camera, processing device, etc. The software is a set of libraries that were, for the most part, developed in Python. For the embedded software, a set

of images with photos of Brazilian car license plates was used to train an object detection model to detect the license plates. Finally, an optical character recognition service was used to extract the content of the plate, thus making it possible to register and emit the light signal to the user.

Key-Words: machine learning; object detection; license plate recognition; optical character recognition; computer vision; artificial intelligence.

1. Introdução

É de conhecimento popular que, nas últimas décadas, as tecnologias de informação, em geral, como as tecnologias de comunicação, segurança, saúde, entretenimento, transporte, entre outras, tiveram uma evolução exponencial e, com isso, trouxeram mais valor, eficiência, praticidade e qualidade aos produtos e serviços, os quais são ofertados à população. É um fato que o avanço da tecnologia facilita o dia a dia, proporcionando soluções eficientes que visam melhorar os processos cotidianos. Pode-se afirmar que, com o avanço das tecnologias, elas proporcionam mais conforto e produtividade aos seus usuários. Com isso, o número de pessoas que adotam as tecnologias veem crescendo. Por consequência, isso faz com que ela esteja mais presente nas atividades realizadas rotineiramente na vida das pessoas, seja pessoal ou profissional.

No mundo moderno, é de se esperar que tarefas repetitivas e/ou manuais, que antes eram executadas por pessoas, possam ser substituídas por alguma tecnologia que automatize aquela tarefa, isso porque a substituição pode trazer ganho em desempenho, em custo ou em eficiência na execução da tarefa. Quando consideramos o ambiente corporativo, as tecnologias atuam como um meio de ganhar produtividade, de tornar processos mais eficientes e menos custosos e podem significar uma vantagem competitiva para a corporação. Se olharmos de um ponto de vista crítico para as organizações que não adotam tecnologia nos seus processos ou não se mantêm atualizadas, pode-se dizer que elas têm uma desvantagem em relação a outras que o fazem e podem ser consideradas ultrapassadas ou que estão perdendo a chance de serem mais eficientes. Dito isso, podemos concluir que a tecnologia pode ser uma grande aliada e trazer diversos benefícios quando bem aproveitada.

1.1. Contexto

Até o momento no qual este trabalho foi desenvolvido, ao transitar pelo campus sede da Universidade Federal Rural de Pernambuco (UFRPE), localizado na cidade de Recife, no estado de Pernambuco, pode-se observar que existe um registro e controle de acesso de veículos ao campus, feito por funcionários da universidade nas guaritas. Nesses lugares, os funcionários registram manualmente em uma planilha de papel com uma prancheta e canetas [Anexo 1] as placas veiculares e os horários de entrada e saída dos veículos que transitam pelo campus. Quando julgam necessário, também registram outras informações como modelo, cor do veículo e identificação do condutor. É importante comentar que esse registro manual requer um certo desvio de atenção do funcionário em relação ao ambiente, uma vez que ele precisa voltar a sua atenção à planilha para registrar as informações. Atualmente, também é feita a liberação direta da entrada do veículo através do uso de adesivo no para-brisa. Este adesivo é obtido ao realizar o cadastramento do veículo na universidade. Porém, esse tipo de controle mediante adesivos é inseguro, porque eles podem ser facilmente clonados.

Registrar as informações sobre os veículos e seus horários se faz necessário porque podem auxiliar na segurança do campus, pois assim a instituição pode identificar os veículos em que estão autorizados a circular e reconhecer possíveis suspeitos ou não autorizados, além de saber quais veículos estiveram no campus e em qual faixa de horário.

A problemática deste projeto consiste em desenvolver um sistema de controle de acesso que realiza a detecção da placa do veículo, faz uma consulta em alguma base de dados de placas autorizadas e emite um sinal luminoso para informar ao funcionário se a placa está previamente autorizada a ingressar no campus ou não. De modo geral, o sistema visa auxiliar tanto no processo de registro das placas e seus horários de acesso ao campus quanto na verificação de placas previamente registradas através do desenvolvimento de uma solução tecnológica simples que supre essa demanda e diminui ou até faz com que não seja mais necessário o registro manual em planilhas e papéis. Com isso, o processo se torna mais eficiente

e automático. Por se tratar de um protótipo, nesse projeto estamos desconsiderando veículos com placas apenas na parte traseira, como motocicletas ou ciclomotores, devido à dificuldade em obter a imagem da parte traseira do veículo.

1.2. Motivação

A motivação para o desenvolvimento deste projeto veio através da convivência na universidade e do desejo de ajudar a UFRPE enquanto estudante. Ao transitar no campus e observar a realização manual da tarefa, surgiu daí o desejo de ajudar e o questionamento de como seria possível otimizar, modernizar e automatizar esse processo usando computação e inovação. Além disso, a ideia é uma medida mais segura do que os métodos utilizados atualmente, uma vez que a placa pode ser consultada em uma base confiável da UFRPE e tendo em vista que os adesivos podem ser trocados ou falsificados e usados por veículos não devidamente cadastrados. Por fim, essa solução também minimiza o desvio de atenção que ocorre quando o funcionário anota as informações no papel.

1.3. Objetivos

Este projeto tem como principal objetivo desenvolver um protótipo de uma solução simples para auxiliar no processo de registro e reconhecimento de placas para o controle de acesso de veículos na UFRPE. O sistema deve auxiliar o funcionário da guarita em parte do seu trabalho, assim tornando mais eficiente a realização da atividade e agregando valor à necessidade da universidade em obter e armazenar essa informação sobre os veículos que circulam pelo campus. Além disso, aumentar a segurança mediante uma autorização mais confiável de veículos que tenham a placa pré-cadastrada na universidade.

1.3.1. Objetivos específicos

- Desenvolver um produto de hardware para o reconhecimento de placas veiculares;
- Definir e treinar o modelo para a detecção de placas veiculares;
- Programar o software embutido para verificar se a placa de um veículo está presente na lista de placas autorizadas;

- Controlar um emissor de sinalização visual para informar ao vigilante se a placa está autorizada a ingressar na universidade (sinal verde) ou não (sinal vermelho);
- Realizar o registro das placas que ingressaram, a data e o horário de ingresso, reduzindo a necessidade do uso de papel existente no registro manual;
- Automatizar o procedimento de controle de acesso, evitando desvio de atenção do vigilante para outras atividades e fortalecendo a segurança devido ao mínimo de interferência humana no processo.

1.4. Contribuição

O trabalho se trata de um protótipo para uma solução que automatiza, facilita e supre uma demanda da UFRPE, auxiliando no trabalho realizado pelos funcionários e diminuindo a necessidade de anotar e armazenar essas informações manualmente, uma vez que a ideia fosse implementada. Também aprimora a segurança no campus, tornando o processo de acesso mais confiável e difícil de falsificar, além de conservar a atenção do funcionário ao ambiente sem o desvio necessário ao anotar a placa no papel. Por fim, trata-se de um possível produto que pode ser aprimorado, adaptado e comercializado em outros contextos e organizações.

1.5. Organização do trabalho

Este trabalho está organizado da seguinte forma: a Seção 1 contém a introdução, assim como o contexto, motivação, objetivos e contribuição. A Seção 2 apresenta os trabalhos relacionados existentes na literatura, e a Seção 3 apresenta o referencial teórico com os principais conceitos abordados neste projeto. Na Seção 4, são discutidas as ferramentas utilizadas no projeto, e a Seção 5 contém o detalhamento sobre o desenvolvimento do projeto. Por sua vez, a Seção 6 possui a apresentação do modelo final do protótipo e, na Seção 7, está a conclusão e os trabalhos futuros. Por fim, as Seções 8 e 9 apresentam as referências bibliográficas utilizadas e os anexos, respectivamente.

2. Trabalhos relacionados

A área de Sistemas de Reconhecimento de Placas Veiculares é uma área com uma vasta literatura. No estudo de David Wilkerson [9], o autor realizou uma revisão na literatura visando implementar uma abordagem de identificação de placas veiculares brasileiras, por meio de processamento de imagens estáticas. Os objetivos do trabalho foram identificar a posição da placa na imagem, realizar a identificação individual dos caracteres e extraí-los da imagem da placa, resultando em um texto. A metodologia do autor para identificar a abordagem utilizada foi a realização de revisão literária de técnicas já implementadas e discutidas na literatura e realização de testes e a partir daí a seleção para implementação de uma técnica de localização, uma de segmentação e uma de reconhecimento. A execução da implementação se deu em quatro etapas, foram elas: Aquisição, Localização, Segmentação e Reconhecimento. Na aquisição, foi onde houve o levantamento e escolha dos bancos de dados usados, que foi um banco de autoria própria e outro, banco já existente com placas de carros dos Estados Unidos. Para a implementação da parte de localização das placas e segmentação, foram utilizadas técnicas de detecção de bordas, binarização, morfologia matemática e propriedades geométricas. Por fim, o reconhecimento individual foi feito com técnicas de comparação e uso da ferramenta de software "Tesseract OCR". No geral, o autor obteve bons resultados, porém, também foram encontradas algumas dificuldades que impactam diretamente na sua taxa de acertos, como a falta ou o excesso de iluminação na imagem, presença de reflexos que atrapalharam a detecção, e a detecção incorreta de objetos e textos que estavam próximos à placa. Ao realizar a comparação de seus resultados com os resultados existentes na literatura da época, o autor obteve taxas de acerto menores, porém satisfatórias, tendo em vista os obstáculos enfrentados e as características específicas de cada trabalho. Por fim, com a conclusão do trabalho, o autor alcançou todos os objetivos planejados.

No trabalho de Luz Alves et al. [11], os autores apresentam um método para localização e reconhecimento de placas veiculares. Basicamente, o método consiste na extração da região da placa, seguida do reconhecimento do texto da placa. A extração da placa na imagem consiste na identificação da região da imagem que pertence ao objeto placa, e para fazer isso, eles usam uma combinação de

operadores morfológicos matemáticos. Já o reconhecimento do texto da placa é a identificação dos caracteres contidos na região identificada e, para isso, eles usam busca por *template* para identificar possíveis placas candidatas e, em seguida, aplicam uma série de avaliações heurísticas, como analisar a quantidade e o posicionamento dos caracteres e números contidos no texto e eliminar os que estejam sobrepostos, para assim chegarem ao resultado dos caracteres contidos na placa.

Para realizar a avaliação dos resultados, os autores dividiram o método em 3 aspectos. No aspecto da extração da região da placa na imagem, o método deles extraiu corretamente 89% dos dados de validação. Quanto ao reconhecimento dos caracteres, o método obteve 88% de identificações corretas e, por fim, em relação ao reconhecimento geral da placa, obtiveram 78% de placas reconhecidas corretamente. No geral, os autores obtiveram bons resultados e conseguiram concluir o objetivo proposto, que era apresentar um método para localização e reconhecimento de placas veiculares.

O trabalho de Cordeiro [12] teve o objetivo de construir um módulo de reconhecimento para a base de um sistema de reconhecimento automático de placas veiculares para a Universidade Candido Mendes (UCAM). No método proposto no trabalho, o autor realiza um pré-processamento nas imagens. Esse pré-processamento consiste na conversão da imagem para a escala de nível cinza, seguido pela aplicação de um filtro de remoção de ruídos que promove a suavização da imagem. Depois disso, executa seu algoritmo de detecção de placas, que consiste principalmente na análise das projeções de intensidade na imagem. Esta análise se resume a duas fases com os mesmos princípios, os quais são: aplicar filtros nas imagens como filtro de detecção de bordas, filtro gaussiano e filtro de mediana, além de aplicar equações de projeção de intensidade. E, apesar de princípios semelhantes, as duas fases se divergem em alguns coeficientes utilizados.

A primeira fase da análise é a detecção de uma região baseada na projeção de intensidade vertical da imagem. A segunda fase é a análise da projeção da intensidade horizontal que tem o resultado da primeira análise como entrada. Ao fim das duas fases, é possível criar um histograma que é utilizado para determinar uma região candidata na imagem para a placa veicular. Com o intuito de realizar testes

de reconhecimento de caracteres nas imagens detectadas, o autor usou o Tesseract para realizar o OCR. Segundo o autor, o seu método obtém boas taxas de acerto em imagens com placas bem detalhadas e realçadas, e o método teve dificuldades com imagens distorcidas ou que continham outros objetos como letreiros e adesivos próximos à placa, pois esses fatores causam significativa interferência no reconhecimento da região. No geral, o método obteve bons resultados com os dados testados, atingiu uma taxa de acertos de 92% na projeção vertical, 70% na projeção horizontal e uma taxa de acerto global de 81%.

3. Referencial Teórico

3.1. Sistemas de reconhecimento de placas de veículos

Os sistemas de reconhecimento de placas veiculares, ou no inglês "*License Plate Recognition*" (LPR), são recursos que buscam reconhecer placas de identificação veicular, geralmente em algum tipo de mídia como imagens ou vídeos. Segundo Alves et al. (2011), o reconhecimento de placas veiculares em imagens digitais é um importante problema da área de visão computacional.

Esses sistemas são desenvolvidos com o objetivo específico de identificar as placas veiculares, sendo formados por softwares e um conjunto diverso de hardware, podendo conter diversos tipos de componentes como sensores, câmeras, dispositivos de processamento e dispositivos adicionais voltados à necessidade específica do sistema, como luzes e monitores.

Geralmente, esse tipo de sistema fica capturando mídias em tempo real e processando por meio de algoritmos treinados para identificar o conteúdo das placas. Outra alternativa é o sistema ser acionado por algum gatilho programado, como um sensor ou o acionamento manual pressionando um botão/controle remoto, ou algo semelhante, e então capturar a mídia e realizar o processamento da imagem. Essa segunda alternativa é normalmente utilizada quando se deseja poupar algum recurso, evitando que o sistema fique funcionando com toda sua capacidade e componentes por mais tempo que o necessário.

Esses sistemas são geralmente usados em estacionamentos, pedágios e na gestão de tráfego de veículos. São importantes devido aos benefícios que oferecem em termos de eficiência, segurança e automação.

3.2. Reconhecimento Óptico de Caracteres

O Reconhecimento óptico de caracteres, ou, no inglês, "*Optical Character Recognition*" (OCR), é uma tecnologia que visa identificar e extrair caracteres ou textos de um arquivo digital. Dessa forma, essa tecnologia possibilita extrair informações de um arquivo não editável em termos de texto e transformá-las em conteúdo manipulável. O OCR desempenha um papel essencial em várias aplicações, automatizando a extração de informações de texto em documentos digitais. Essa tecnologia utiliza algoritmos para analisar padrões visuais em imagens ou documentos e reconhecer os caracteres por meio de técnicas de processamento de imagem, aprendizado de máquina e redes neurais. É comumente aplicado em diversas áreas, como na digitalização de documentos, onde ocorre a conversão de documentos físicos para eletrônicos, e na automatização de extração de informações em documentos como formulários, faturas, recibos e outros.

Existem algumas opções de serviços de OCR na nuvem, como o Google *Cloud Vision API*, *Microsoft Azure Computer Vision* e *Amazon Textract*. Cada um desses serviços possui suporte a diversos idiomas, capacidade de reconhecer tanto texto impresso quanto texto manuscrito e integração com suas respectivas plataformas. O serviço de OCR usado neste projeto foi o *Google Cloud Vision*. Ele foi escolhido por ser fácil de configurar, por ter uma quantidade inicial de até 1000 imagens gratuita e permitir o uso através de uma biblioteca no python, a biblioteca "google.cloud".

3.3. YOLO

O YOLO, que significa "*You Only Look Once*", é um framework de detecção de objetos em imagens e vídeos desenvolvido por Joseph Redmon e Ali Farhadi na Universidade de Washington em 2015. Essa ferramenta é amplamente conhecida e utilizada no ramo de visão computacional, por possibilitar a criação e utilização de modelos de algoritmos de reconhecimento de objetos em tempo real. O YOLO

simplifica o processo de desenvolvimento de aprendizado de máquina, possibilitando o treinamento e configuração de um modelo detector de objetos preciso, de maneira simplificada e eficiente. Além disso, possui modelos e classes pré-treinadas e configuradas que o usuário pode utilizar sem precisar realizar mais treinamentos.

Desde o seu lançamento até agora, existem algumas versões do YOLO. A versão mais recente é a 8, que foi a versão utilizada neste projeto. O YOLO possui 5 tipos: “YOLOv8-cls”, que faz a classificação de imagens; “YOLOv8”, que serve para detecção de objetos; “YOLOv8-seg”, que faz segmentação de instâncias; “YOLOv8-pose”, que serve para fazer estimativa de pose em imagens; e, por fim, “YOLOv8-obb”, que serve para detenção orientada. Cada tipo tem 5 tamanhos diferentes, que vão de nano, pequeno, médio, grande até enorme. Quanto maior o tamanho do modelo, melhor será a qualidade de previsão, porém também será mais demorado o seu funcionamento.

O modelo utilizado neste projeto foi o YOLOv8, usado para detecção de objetos em mídias, e o tamanho foi o “nano”. Este modelo foi escolhido por ser o mais adequado ao objetivo. Quanto à escolha do tamanho, a qual foi o menor disponível e, portanto, mais leve e rápido, ela foi feita considerando os limites de recursos computacionais do dispositivo que realizaria o processamento.

4. Ferramentas

O protótipo desenvolvido é constituído por um conjunto de hardware e software embutido que são detalhados a seguir.

4.1. Hardware

Quanto às ferramentas de hardware, por se tratar de um protótipo que não deve ser a versão final, o objetivo foi usar ferramentas que fossem acessíveis quanto ao custo e também fossem fáceis de manipular e substituir. Dessa forma, em versões futuras do projeto, seria possível a atualização e evolução dos componentes.

Na construção do protótipo, foi usado um semáforo de duas cores, uma bateria de 12 volts, um Raspberry Pi, um módulo de câmera para Raspberry Pi, um carregador portátil, um mini display, um relé, um LED, um botão e alguns fios *jumpers* para conexão dos dispositivos. Todos os componentes serão detalhados a seguir.

4.1.1. Raspberry Pi

O Raspberry Pi é o principal dispositivo de hardware do sistema, por ser nele que o software é executado. No projeto, foi utilizado o Raspberry Pi 4 modelo B com 4 GB de memória RAM. Ele foi escolhido por sua capacidade de processamento e, principalmente, pela sua interface GPIO (*General-Purpose Input/Output*), onde você pode conectar facilmente componentes e testar ideias sem a necessidade de placas de circuito adicionais. A interface permite que o Raspberry Pi controle dispositivos externos, como LEDs e outros componentes eletrônicos. No protótipo, a interface foi usada para conectar e/ou controlar os componentes listados a seguir:

- LED: Um LED foi conectado ao Raspberry Pi através da interface GPIO para indicar quando o sistema está pronto para uso.

- Botão: O botão foi conectado na interface e serve como gatilho para que o sistema inicie e dê continuidade ao processamento.

- Display OLED (*Organic Light-Emitting Diode*): Um pequeno display foi usado, e nele é exibido um texto, que consiste nos caracteres da placa quando a detecção e extração ocorrem corretamente ou em uma mensagem indicando o contrário.

- Relé: O relé foi conectado no Raspberry Pi, no semáforo e na bateria de alimentação e possibilita controlar o semáforo ligando e desligando conforme o resultado do processamento no Raspberry Pi.

Além dos componentes citados acima, foi utilizado também um módulo de câmera para Raspberry Pi Ov5647 de 5 MP para realização da captura da imagem. Quanto à alimentação do Raspberry Pi, ele precisa de uma fonte de alimentação de 5 volts e

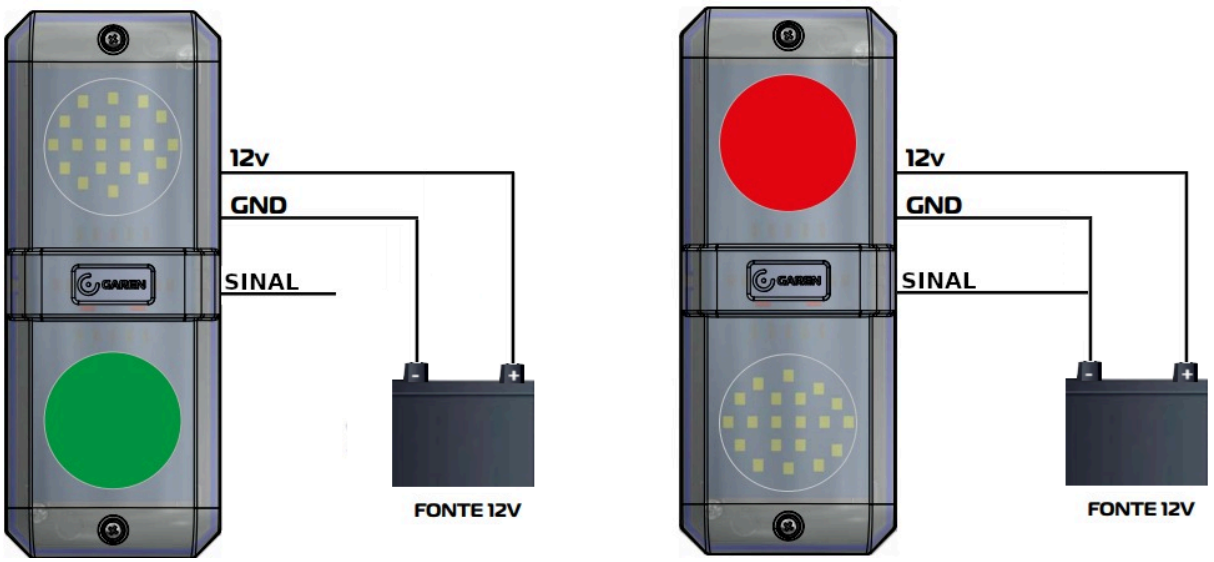
pelo menos 3 amperes. No projeto deste protótipo, foi usado um carregador portátil que fornece essa alimentação.

4.1.2. Semáforo

O protótipo utiliza um semáforo de LED simples de 12 volts que possui duas cores (verde e vermelho) para sinalizar se a placa é uma placa conhecida ou não. Entende-se placa conhecida como uma placa previamente cadastrada junto à UFRPE, onde o verde indica que a placa é conhecida, e o vermelho indica o contrário. Caso a detecção não seja realizada com sucesso – o que pode acontecer devido à má qualidade da imagem, entre outros fatores–, o semáforo ficará alternando entre as duas cores até que o gatilho seja disparado, e então uma nova detecção poderá ser realizada.

Quando ligado, o semáforo possui apenas 2 estágios, ou o verde, ou o vermelho está ligado. Para ativar o verde, basta conectar o fio positivo e o fio GND na fonte de alimentação em seus pólos correspondentes. Já para ligar o vermelho, basta conectar o fio de sinal no polo negativo da alimentação, conforme figura 1. Para realizar a alimentação do semáforo no protótipo, foi utilizada uma bateria de 12 volts que geralmente é utilizada em motocicletas.

Figura 1 - Esquema de estágio do semáforo.



Fonte: https://suporte.segbr.com.br/hc/pt-br/articles/18302632639123-SINALEIRA-AUDIOVISUAL-12V-SEG-Manual-de-instru%C3%A7%C3%B5es com adaptações do autor.

4.2. Software

Todo software ou algoritmo usado, ou desenvolvido no projeto desse protótipo foi pensado e escolhido de modo a manter a baixa complexidade do sistema para o usuário final, que era um dos objetivos do projeto. A maior parte do sistema é desenvolvido em Python, utilizando suas bibliotecas, e o sistema operacional utilizado no Raspberry Pi foi o Raspberry Pi OS, o qual é o sistema operacional oficial da Raspberry.

4.2.1. Python

Python é uma linguagem de programação de alto nível muito utilizada no ramo de aprendizagem de máquina devido à sua diversidade de pacotes voltadas a essa área. A versão 3 da linguagem foi usada para treinar o modelo de detecção de placas e também como a linguagem na qual o sistema foi feito. Foi escolhido por sua sintaxe simples, o que contribui na manutenção do código, pelo poder que seus pacotes oferecem, como manipular a interface GPIO do Raspberry Pi e pelo suporte ao YOLO, e também por ser uma linguagem nativa do Raspberry OS. Algumas das principais bibliotecas utilizadas foram a RPi.GPIO, que é uma biblioteca que permite configurar e manipular a interface GPIO do Raspberry Pi, a biblioteca da Ultralytics que dá suporte para treinar e carregar o modelo, e a biblioteca google.cloud usada para configurar e usar o serviço que faz a extração do texto.

4.2.2. Google colab

Para realizar o treinamento do modelo de detecção das placas no YOLO, foi utilizada a plataforma *Colab* do *Google*. O *Colab* foi escolhido, por permitir que seja executado código python pelo navegador por meio de um ambiente cloud, o que é adequado para este caso. Para o treinamento, foi preciso obter uma assinatura do *Colab PRO* e o treinamento foi realizado em um ambiente com 12 GB de memória RAM no sistema, 78 GB de armazenamento e GPU Tesla V100-SXM2 com 16 GB de RAM.

4.2.3. Dataset

O dataset é um recurso fundamental quando se trata de aprendizagem de máquina e treinamento de modelos. O conjunto utilizado para treinamento do modelo de detecção de placas foi o dataset disponibilizado gratuitamente para fins não lucrativos RodoSol-ALPR[1] [1]. Segundo os autores, o dataset contém 20 mil imagens capturadas por câmeras fixas localizadas em uma Rodovia no estado brasileiro do Espírito Santo, e as imagens são de diferentes tipos de veículos, como carros, motos, ônibus e caminhões, captadas durante todo o dia e em diversos dias. Todas as imagens têm resolução de 1.280 × 720 píxeis. Como comentado anteriormente, para o projeto deste protótipo, as imagens de veículos que possuem apenas placas traseiras foram desconsideradas, restando 10 mil imagens que foram utilizadas no treinamento do modelo. Essas imagens restantes eram compostas por 5 mil placas no modelo Mercosul e 5 mil no antigo modelo que era adotado pelo Brasil. Ambos os modelos podem ser observados nas figuras 2 e 3, respectivamente.

Figura 2 - Modelo de placa Mercosul.

Fonte: https://utschbrasil.com/placa-mercosul/

Figura 3 - Antigo modelo de placa brasileiro.

Fonte: https://pt.m.wikipedia.org/wiki/Ficheiro:Placa_passeio_de_ve%C3%ADculos_no_Brasil_2008.png.

[1] https://github.com/raysonlaroca/rodosol-alpr-dataset/

5. Desenvolvimento do projeto

A solução implementada no protótipo foi desenvolvida em Python e é executada em sua maior parte localmente no Raspberry Pi. O código está disponível em um repositório[2], e nele estão todos os requisitos para instalação. O principal módulo desenvolvido neste projeto foi um modelo de detecção de objetos, que foi treinado e que será explanado a seguir, além do software embutido criado para a orquestração do hardware e integração dos serviços.

5.1. Modelo de detecção de placas

Para ser possível identificar as placas veiculares nas imagens capturadas pelo sistema, um modelo de detecção de objetos foi previamente treinado para detectar as placas veiculares. O treinamento foi feito com parte do conjunto de imagens do dataset RodoSol-ALPR [1] usando a tecnologia YOLO na versão 8. E, o treinamento foi previamente realizado para que quando o sistema estiver em execução, o modelo fosse apenas carregado e utilizado. Como mencionado nos parágrafos anteriores, o algoritmo de treinamento foi realizado em um notebook[3] no *Colab*. Para que tudo isso fosse possível, foi realizado um tratamento no conjunto de dados para que ficassem conforme o algoritmo do YOLO esperava. Além das imagens, o dataset possui um arquivo de texto para cada imagem com algumas informações, são elas: "*type*", "*plate*", "*layout*" e "*corners*" que indicam, respectivamente, o tipo de veículo que consta na imagem, o conteúdo de texto da placa, o layout da placa e as coordenadas delimitadoras de onde a placa está localizada na imagem.

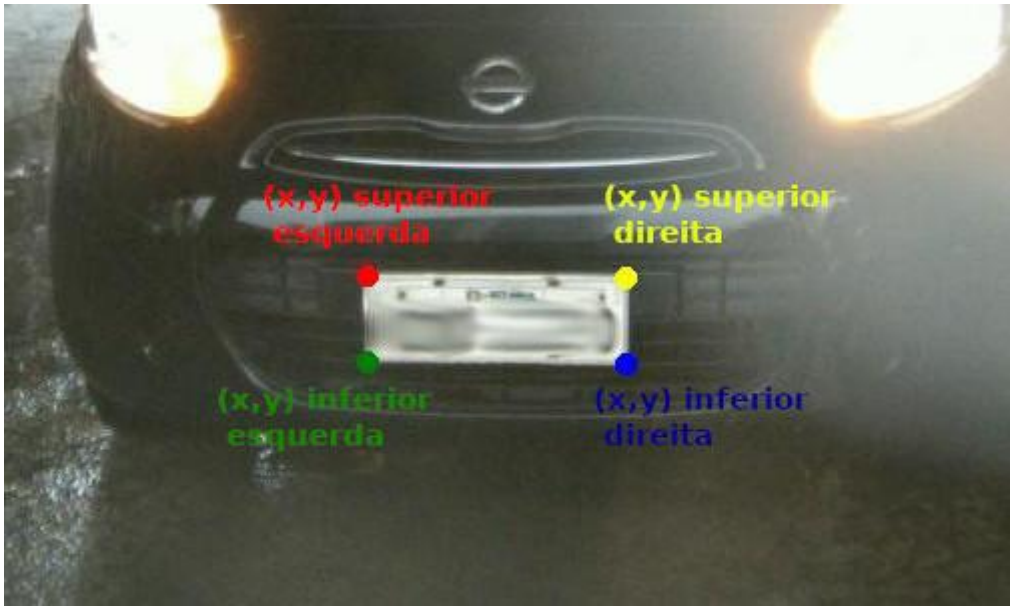
5.1.1. Normalização das Coordenadas para o treinamento do YOLO

Além das imagens, para realizar o treinamento do modelo, era necessário obter informações sobre o layout, o conteúdo da placa e as coordenadas normalizadas da caixa delimitadora. O tratamento mencionado anteriormente envolveu o cálculo e a normalização das coordenadas do dataset para estarem em um intervalo entre 0 e 1. Esse tratamento foi feito por meio de um *script*[4] em Python que consistia

[2] https://github.com/Stefanyvitoria/TCC
[3] https://github.com/Stefanyvitoria/TCC/blob/master/src/utils/Reconhecimento_de_placas.ipynb
[4] https://github.com/Stefanyvitoria/TCC/blob/master/src/utils/formatar_dataset.py

principalmente em calcular quatro pontos: os centros "x" e "y" e a altura e largura da placa. A figura 4 está exemplificando as coordenadas fornecidas no conjunto de dados e o cálculo de cada um desses pontos:

Figura 4 - Exemplo das coordenadas fornecidas no dataset.



Fonte: Autor.

- **X central** - O x central é a soma entre o x da coordenada superior esquerda e metade da largura da placa, dividido pelo tamanho total da largura da imagem. Isso é representado pela seguinte fórmula:

  X central = (x superior esquerda + ((x superior direita - x superior esquerda) / 2)) / largura total da imagem.

- **Y central** - O y central é a soma entre o y da coordenada superior esquerda e metade da altura da placa, dividido pelo tamanho total da altura da imagem. Isso é representado pela fórmula a seguir:

  Y central = (y superior esquerdo + ((y inferior direito - y superior direito) / 2)) / altura total da imagem.

- **Largura** - A largura normalizada se dá pela diferença entre o valor x da coordenada superior direita e o valor x da coordenada superior esquerda,

tudo isso dividido pela largura total da imagem. Isso é representado pela fórmula abaixo:

Largura normalizada = (x superior direito - x superior esquerdo) / largura total da imagem.

- **Altura** - A altura normalizada se dá pela diferença entre o valor y da coordenada inferior direita e o valor y da coordenada superior direita, tudo isso dividido pela altura total da imagem. Isso é representado pela fórmula a seguir:

  Altura normalizada = (y inferior direito - y superior direito) / altura total da imagem.

5.1.2. Separação do conjunto de dados

Além do tratamento das coordenadas, também foi preciso separar os dados em duas partes: uma de treinamento e outra de validação. O conjunto de dados já é originalmente dividido aleatoriamente em 40% dos dados para treinamento, 40% para teste e 20% para validação, e essa divisão conserva o percentual de amostras para cada tipo de veículo e layout. Então, com o objetivo de conservar esse percentual, foi usada no projeto a divisão original com uma adaptação.

Foram utilizadas 10 mil imagens, separadas em 20% para validação e 80% para treinamento do modelo. Os 80% correspondem à soma dos dados originalmente separados para teste e treinamento e os 20% são os originalmente separados para validação, preservando assim o percentual de amostras de cada layout.

5.2. Treinamento do modelo de detecção de placas

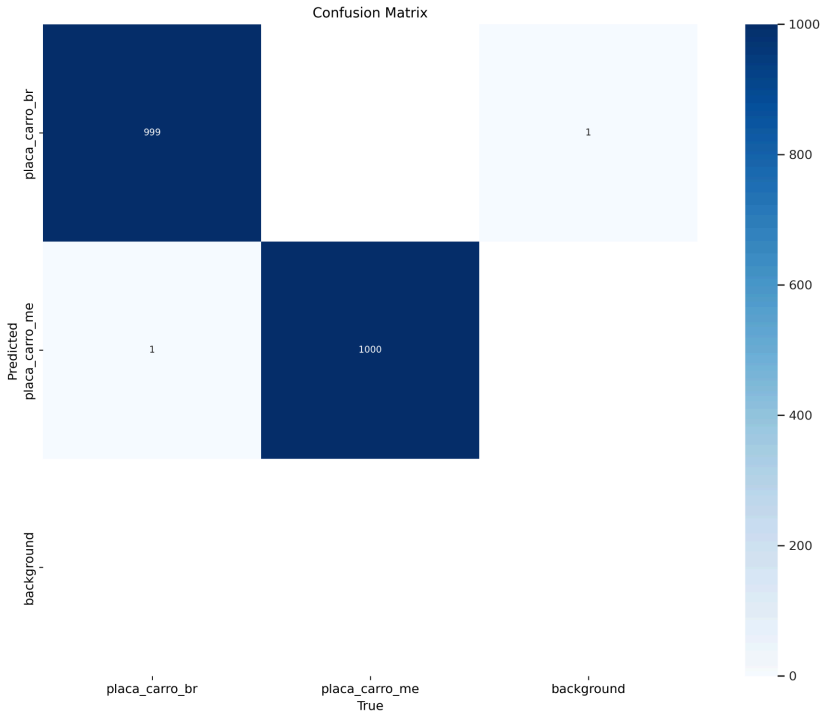
Uma vez que os cálculos para normalizar as coordenadas foram concluídos e os dados foram separados, já se tinham todas as informações necessárias para realizar o treinamento. Os dados foram então organizados na estrutura necessária que o YOLO exige e armazenados na nuvem, onde o treinamento foi feito.

Visando tornar a detecção mais precisa e também permitir a distinção da placa detectada, o modelo foi treinado com 2 classes. A primeira classe foi representada pelo identificador “0” e rótulo “placa_carro_br”, representando placas com o antigo layout brasileiro. A segunda classe foi representada pelo identificador “1” e rótulo “placa_carro_me”, representando placas com o layout Mercosul.

Inicialmente, o modelo foi treinado com 20 épocas. Em seguida, esse número foi aumentado até se obter um desempenho satisfatório do modelo, considerando as limitações de recursos do projeto. Portanto, o modelo final foi treinado por 100 épocas, e sua matriz de confusão pode ser visualizada na Figura 5.

Como podemos ver na Figura 5, a matriz de confusão do modelo apresenta apenas 2 ocorrências de um falso positivo, onde o resultado era uma placa no layout antigo e o modelo identificou como o layout Mercosul. Além disso, houve um teste de background que o modelo identificou como uma placa no layout antigo.

Figura 5 - Matriz de confusão do modelo de detecção.

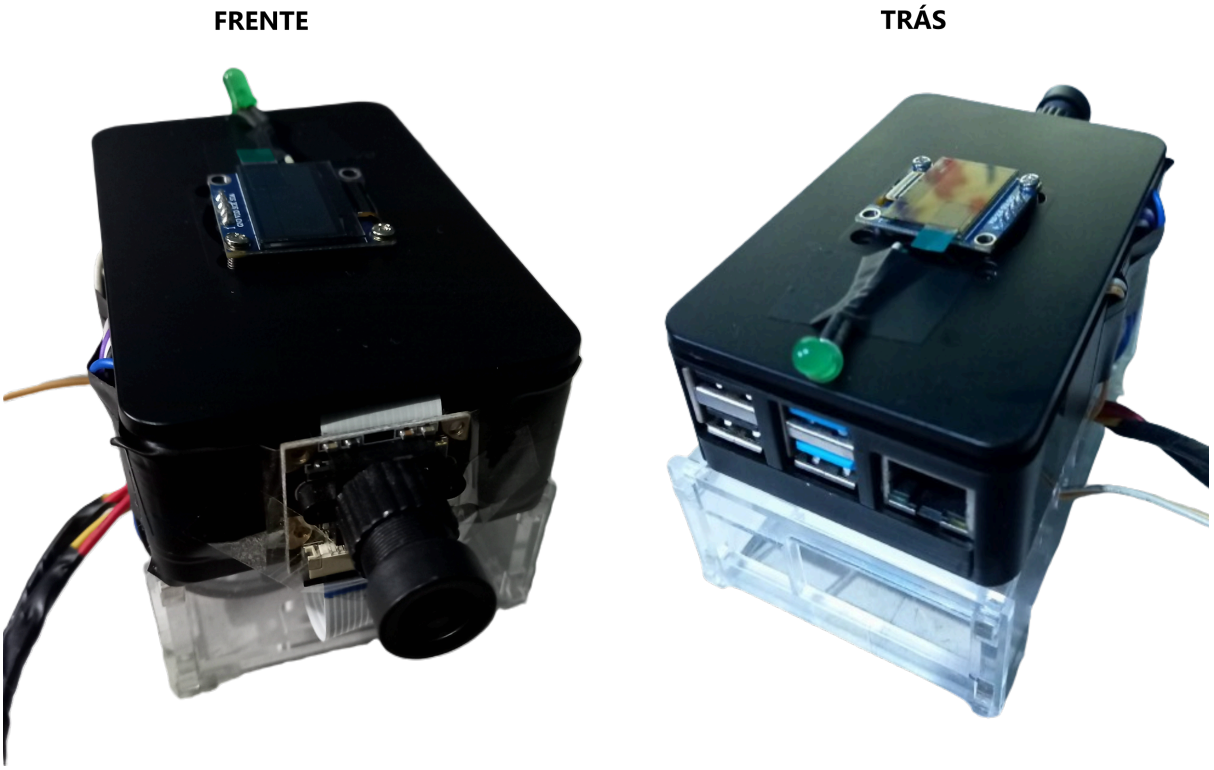


Fonte: Autor.

6. Apresentação do modelo final

A versão final do hardware que compõe o protótipo pode ser observada na figura 6. Sua disposição está da seguinte forma: o Raspberry Pi está em uma caixa protetora preta. Conectados a ele estão o display, o botão, o LED e a câmera. Abaixo dele, em uma caixa de acrílico, está o relé, que se conecta à interface GPIO. Por fim, podemos observar que existem alguns fios conectados no relé, que são para conectá-lo ao semáforo e à bateria.

Figura 6 - Protótipo final.



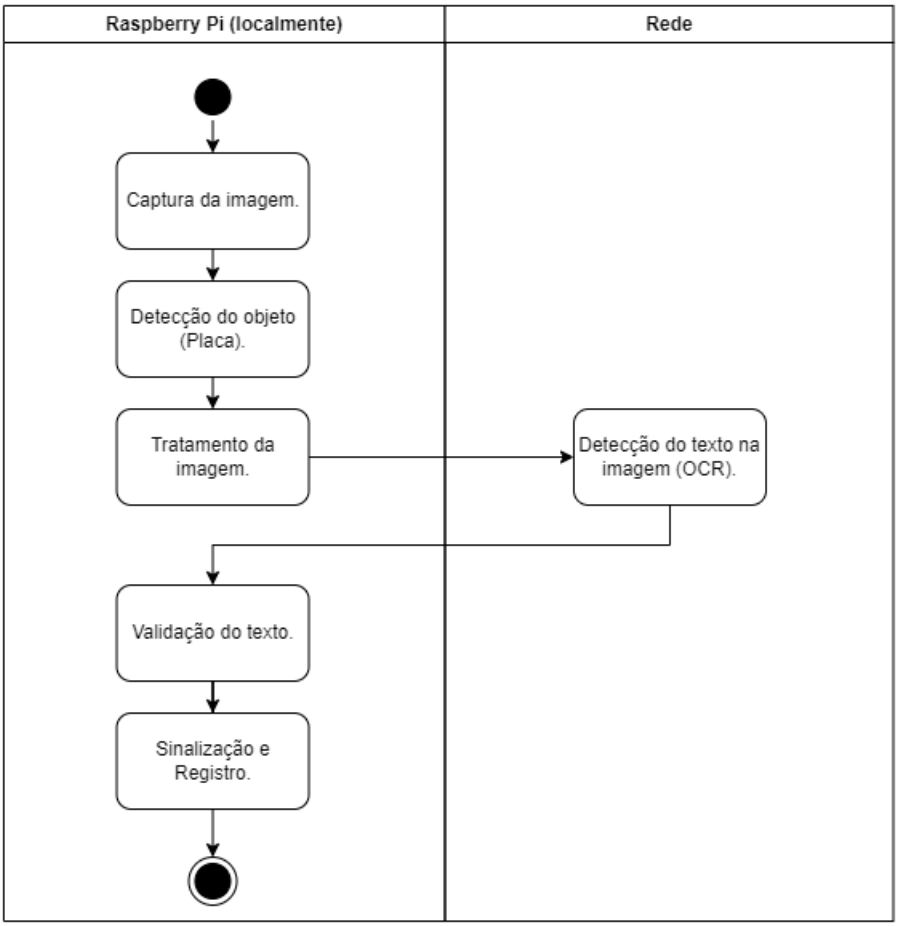
Fonte: Autor.

6.1. Arquitetura

Como podemos ver na figura 7 que contém a representação da arquitetura da aplicação, a maior parte das tarefas realizadas na aplicação é executada localmente, no raspberry pi, apenas o módulo que realiza a extração do texto na imagem (OCR) é executado na nuvem. Por ser preciso realizar esse processamento na nuvem é necessário que o dispositivo tenha acesso à internet.

O sistema tem início com a ação do usuário de disparar o gatilho. Com isso, as seguintes tarefas são realizadas localmente: captura da foto, aplicação da imagem capturada no modelo YOLO de detecção de objetos (que já foi treinado previamente) e tratamento da imagem, gerando uma nova imagem apenas com a região da placa que foi detectada. Em seguida, a etapa de extração do texto na nova imagem é realizada na nuvem, com o serviço de OCR do Google Cloud Vision. E, por fim, o fluxo da aplicação volta a ser executado localmente, onde o resultado do módulo de extração de texto é validado na base de dados e gravado, sendo sinalizado posteriormente. É importante ressaltar que por o projeto se tratar de uma prototipação, a base de dados com as placas pré-cadastradas está localizada localmente no raspberry pi, porém há a possibilidade de realizar essa validação em uma base real da universidade. Durante o desenvolvimento do trabalho, houve a tentativa de contato com a área responsável da UFRPE para discutir essa possibilidade, mas o contato não foi retornado.

Figura 7 - Arquitetura.



Fonte: Autor.

6.2. Fluxo lógico

A figura 8 contém o fluxo lógico do sistema através do diagrama de atividades. Como podemos observar, o sistema pode ser dividido em três grandes partes, sendo elas: captura de imagem e identificação da placa, extração dos caracteres e, por fim, sinalização e registro da placa. Cada uma dessas partes é detalhada a seguir.

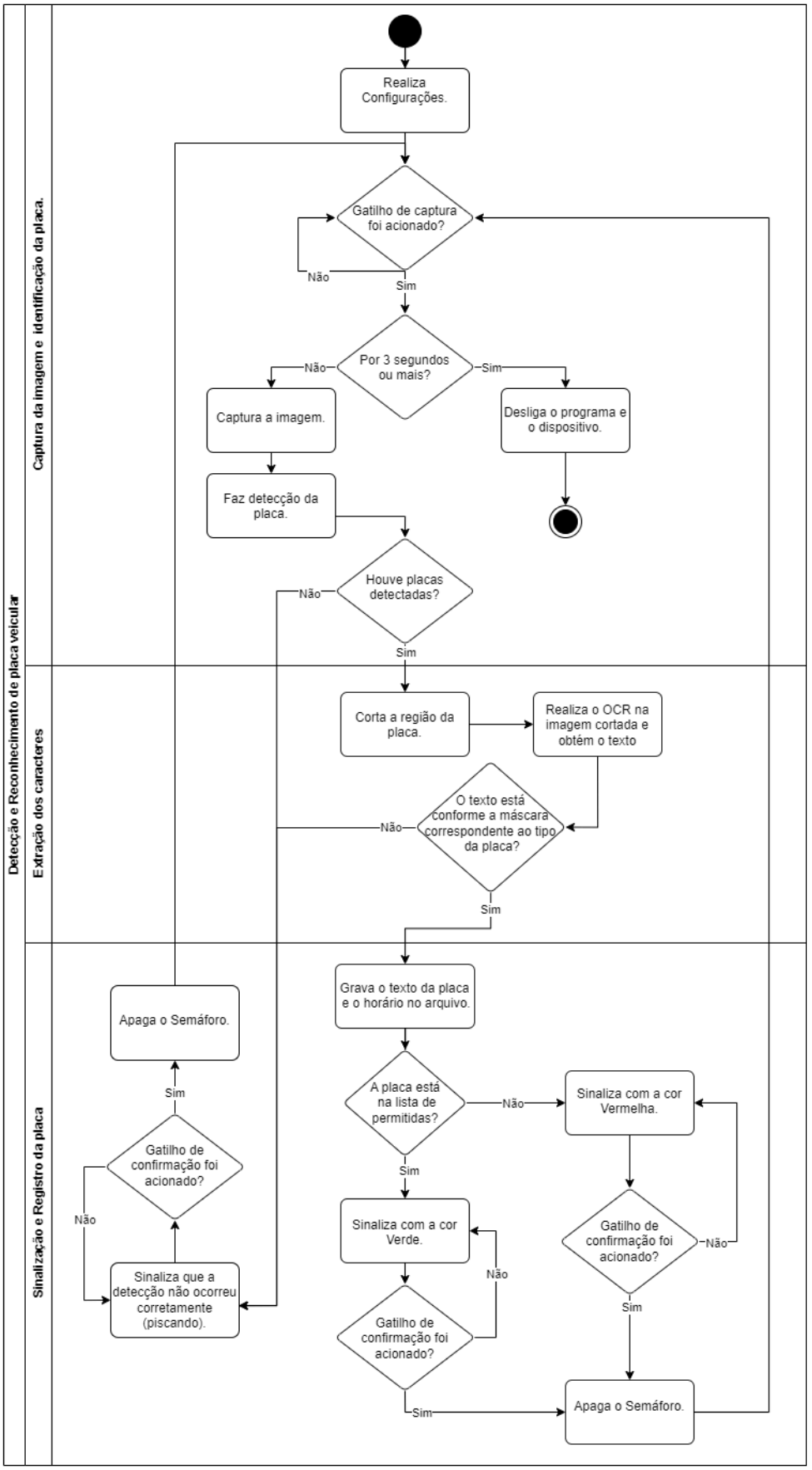
6.3. Captura de imagem e identificação da placa

Quando a aplicação é inicializada, é primeiramente realizada uma configuração inicial. Isso consiste em carregar as placas registradas e configurar os pinos da interface GPIO. Em seguida, o aplicativo entra em um loop infinito para se manter em execução e fica à espera do botão de gatilho ser acionado. Quando o gatilho é disparado por 3 segundos ou mais, o sistema finaliza e desliga o dispositivo. Já quando é disparado por menos de 3 segundos, uma imagem é capturada e aplicada no modelo de detecção, que realiza a identificação da região da placa. Quando não há identificação correta, o sistema mostra no display a mensagem “Sem deteccoes” e sinaliza para o usuário no semáforo, intercalando rapidamente entre as cores verde e vermelho, até que o usuário pressione novamente o botão de gatilho, indicando que visualizou o resultado. Quando há identificação da região da placa, o sistema segue para o tratamento da imagem e a realização do OCR.

6.4. Extração dos caracteres

A extração dos caracteres tem início com o recorte da imagem capturada para obter uma nova imagem que contenha somente a região da placa detectada. Em seguida, a imagem cortada é aplicada no método que realiza a identificação dos caracteres, que por sua vez retorna uma *string* com conteúdo identificado. Após isso, há a remoção de caracteres que não sejam números ou letras, com o intuito de remover o “-” presente no layout de placas antigo. Há também a verificação se o texto retornado segue uma máscara determinada. Para placas identificadas com o layout antigo, a máscara se dá por 3 letras seguidas de 4 números; já para placas identificadas com o layout Mercosul, a máscara se dá por 3 letras, seguidas de um número, uma letra e dois números.

Figura 8 - Diagrama de atividades do sistema.



Fonte: Autor.

6.5. Sinalização e registro da placa

Quando o texto extraído não corresponde à máscara, o sistema mostra no display a mensagem “Sem deteccoes” e sinaliza para o usuário através do semáforo, intercalando rapidamente entre as cores verde e vermelho, até que o usuário pressione novamente o botão de gatilho. Quando o texto corresponde à máscara, o sistema segue para o registro.

O registro do conteúdo das placas detectadas e o horário em que foram detectadas é feito em um arquivo local no Raspberry Pi. O arquivo é diário, e há o expurgo com tempo programável. Após realizar a gravação, o sistema verifica se a placa detectada está na lista de conhecidas e sinaliza através do semáforo com o verde ou o vermelho. Se a placa for uma placa previamente registrada, então o sistema sinaliza com o verde; caso contrário, sinaliza com o vermelho. Após o sinal, o sistema aguarda que o usuário pressione o botão de gatilho novamente para indicar que visualizou o resultado e reiniciar o fluxo, aguardando uma nova detecção.

7. Conclusão e trabalhos futuros

A principal proposta deste trabalho foi desenvolver um protótipo de um sistema auxiliar de controle e registro de acesso de veículos para a UFRPE. Para isso, foi desenvolvida uma solução composta por um conjunto de componentes de hardware e também, foi desenvolvido um software embutido para controlar os dispositivos de hardware e realizar o processamento da parte lógica.

O software embutido possui 3 principais módulos: captura de imagem e detecção da placa, extração do conteúdo da placa, e sinalização e registro desse conteúdo. No que se refere à captura de imagem e detecção da placa, para a captura foi utilizado um módulo de câmera para Raspberry Pi e recursos de software nativos do sistema operacional. Já para a detecção da placa como um objeto, isso foi feito através do uso de um modelo de detecção de objetos, que foi treinado neste projeto com um conjunto de imagens que possui diversas fotos de placas veiculares de carros brasileiros. Quanto à extração do conteúdo da placa como texto, devido à limitação de tempo para a realização deste projeto, a extração foi realizada através de um

serviço de OCR generalista que analisa uma imagem e retorna os caracteres identificados nela. Por fim, no que se refere a sinalização e registro, a permissão e o bloqueio de acesso dos veículos são realizados através de um componente de hardware que emite um sinal luminoso para o usuário do sistema (semáforo), e o registro da placa do veículo é realizado em arquivos que podem ser facilmente manipulados e armazenados em outros locais.

Em relação ao hardware, todos os componentes utilizados na solução podem ser substituídos por componentes de melhor desempenho e confiabilidade, e assim potencializar a capacidade da solução desenvolvida.

No que concerne aos objetivos do trabalho, todos foram alcançados: o produto de hardware e software foi desenvolvido; o modelo para a detecção de placas veiculares foi treinado; o sistema sinaliza com dispositivo luminoso para o usuário o resultado da análise e realiza o registro automaticamente. Com isso, o processo se torna mais natural e evita desvios de atenção do usuário para outras atividades, o que fortalece a segurança.

Algumas dificuldades foram encontradas durante o desenvolvimento do trabalho, como a limitação de recursos para realizar o treinamento do modelo de detecção, a ausência de uma base de placas contendo imagens com boa resolução e com a existência de dois layouts de placas para realizar o treinamento, e o teste de várias tecnologias de OCR para descobrir qual seria a mais adequada ao projeto. No entanto, todas essas dificuldades foram superadas. Para superar a limitação de recursos, o treinamento foi realizado na nuvem com uma assinatura do *Google Colab PRO*; o conjunto de imagens foi fornecido gentilmente pelo seu autor, e o OCR foi realizado com o *Google Vision* usando os limites gratuitos da ferramenta.

Como sugestão de trabalhos futuros estão:

- Realização do treinamento do módulo de OCR, pois é possível que, com o treinamento específico utilizando as fontes das placas, este módulo tenha um melhor desempenho e possibilite a execução do sistema totalmente local no Raspberry PI.

- Realizar o teste prático na guarita da universidade visando ajustes para uma implementação real do protótipo.
- Exibir informações pré-cadastradas do veículo, como modelo e cor, em um display para o usuário após a validação da placa, permitindo a verificação visual do veículo para coibir o uso de placas falsas.
- Integrar o sistema com uma base de dados real da UFRPE.
- Realizar a integração do sistema para controlar a abertura da cancela e registrar somente as placas de veículos que de fato ingressaram no campus. Pois pode ocorrer de um veículo ser impedido de acessar a universidade e, assim, esse veículo não será registrado.
- Realizar a integração do sistema para controlar o fechamento da cancela, e assim o sistema sairia do sinal verde automaticamente, reduzindo ainda mais a necessidade de interferência humana.

Por fim, vale ressaltar que o sistema proposto neste projeto também pode ser utilizado em outras organizações ou em estacionamentos que necessitam de controle de acesso de veículos.

8. Referências

1. LAROCA, R.; CARDOSO, E.; LUCIO, D.; ESTEVAM, V.; MENOTTI, D. **On the Cross-dataset Generalization in License Plate Recognition**. In: Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 5: VISAPP. SciTePress, 2022. p. 166-178. ISBN 978-989-758-555-5. ISSN 2184-4321. DOI: 10.5220/0010846800003124.

2. YERYOMENKO, OLEKSIY. **Automatic Number Plate Recognition with Raspberry Pi**. Medium, 2022. Disponível em: https://medium.com/@alexey.yeryomenko/automatic-number-plate-recognition-with-raspberry-pi-e1ac8a804c79. Acesso em: dez. 2023.

3. MADEIRA, DANIEL. **Projeto Raspberry pi 3 no acionamento de lâmpadas**. usinainfo, 2018. Disponível em:

https://www.usinainfo.com.br/blog/utilizando-o-raspberry-pi-3-no-acionamento-de-lampadaspronto-post-de-automacao-residencial/. Acesso em: dez. 2023.

4. JOCHER, G.; CHAURASIA, A.; QIU, J. **Ultralytics YOLO**. Versão 8.0.0. 2023. Disponível em: https://github.com/ultralytics/ultralytics. Acesso em: dez. 2023.

5. GERMANOV, ANDREY. **How to Detect Objects in Images Using the YOLOv8 Neural Network**. freecodecamp, 2023. Disponível em: https://www.freecodecamp.org/news/how-to-detect-objects-in-images-using-yolov8/#data. Acesso em: dez. 2023.

6. BERNARDI, ELY. **Os sistemas de identificação veicular, em especial o reconhecimento automático de placas**. 2014. Tese de Doutorado. Universidade de São Paulo. Disponível em: https://www.teses.usp.br/teses/disponiveis/3/3138/tde-11052016-162646/publico/PPGET_Corrigida_2015_ElyBernardi.pdf. Acesso em dez. 2023.

7. WOJAHN, CARLOS CRISTIANO. **Sistema de detecção de veículos e análise de situação cadastral**. 2022. Disponível em: https://arandu.iffarroupilha.edu.br/bitstream/itemid/251/1/TCC%20Versao%20Final%20Carlos%20Cristiano.pdf. Acesso em dez. 2023.

8. SILVA, ALLEF ANDERSON. **Reconhecimento de placas com o Google Vision**. embarcados, 2019. Disponível em: https://embarcados.com.br/reconhecimento-de-placas-com-o-google-vision/. Acesso em: jan. 2024.

9. KÜSTER, DAVID WILKERSON. **Identificação de placa veicular por processamento de imagem e OCR**. 2018. Trabalho de Conclusão de Curso – Universidade Federal Do Espírito Santo, Vitória/ES. Disponível em: https://ele.ufes.br/sites/engenhariaeletrica.ufes.br/files/field/anexo/david_w_kuster.pdf. Acesso em: jan. 2024.

10. BELVISI, RICARDO et al. **Um sistema de reconhecimento automático de**

placas de automóveis. In: Anais do XIX Congresso Nacional da Sociedade Brasileira de Computaçao-Encontro Nacional de Inteligência Artificial. 1999. p. 537-539.

11. ALVES, LUZ; ALEXANDRE, WONDER; ABREU, DANILO DE SOUSA; ARAÚJO, SIDNEI ALVES DE; LIBRANTZ, ANDRÉ FELIPE HENRIQUES. **Uma abordagem para localização e reconhecimento de placas de licenciamento veicular por meio de operadores morfológicos e busca por template**. Exacta. 2011, v. 9, n. 3, p. 355-362. ISSN 1678-5428. Disponível em: https://www.redalyc.org/pdf/810/81021140008.pdf. Acesso em: fev. 2024.

12. CORDEIRO, TIAGO SAMAHA. **Reconhecimento Automático De Placas Utilizando Inteligência Computacional**. 2015. Disponível em: https://mpoic.ucam-campos.br/wp-content/uploads/2015/11/Tiago-Samaha-Cordeiro 1.pdf. Acesso em: fev. 2024.

9. Anexos

1. Foto da planilha na qual os funcionários registram as informações dos veículos que acessaram o campus.