**Tema**

Classificação de placas automotivas utilizando inteligência artificial para a entrada em estacionamentos.

**Problema**

Em condomínios, estacionamentos ou outros ambientes com trânsito de veículos, nota-se que o monitoramento, quando é realizado, é feito de forma manual e passível de erros. Porém já existem soluções capazes de monitorar automaticamente o trânsito, e que não são implementadas nesses espaços.

**Hipótese**

A implementação de sistemas de identificação automática de veículos não é realizada em espaços comerciais, institucionais e entre outros, devido ao custo elevado das soluções existentes. Logo, criando a necessidade de facilitar o acesso desse tipo de tecnologia para melhorar a segurança nessas zonas.

**Objetivos**

Este trabalho tem como objetivo descrever a criação de um sistema de reconhecimento de placas veiculares de baixo custo, assim como sua implantação e comparar com soluções disponíveis no mercado .

**Justificativa**

Na atualidade existem diversas tecnologias e equipamentos capazes de reconhecer placas de carros, e são amplamente utilizados em estradas e rodovias. Porém essas soluções têm um custo elevado, o que causa um acesso e uso limitado dessas soluções. Entretanto, a identificação automática de veículos têm vantagens do ponto de vista dos seguranças de ambientes como estacionamentos e condomínios, o que poderia facilitar e registrar o trânsito nesses ambientes. Logo, vê-se a necessidade de haver um sistema de identificação automática que seja de baixo custo e alta escalabilidade.

**Referencial Teórico**

1. **Automação**

Segundo Rosário (2009) a automação é todo processo que realiza tarefas e atividades de forma autônoma ou que auxilia o homem em suas tarefas do dia-a-dia. Atualmente, a automação está de uma forma mais explícita na vida de todos, desde portas automáticas e escadas rolantes num shopping a um sistema de inteligência artificial em um estacionamento em grandes cidades, com essas tecnologias de automação, começa a surgir a ideia de uma cidade inteligente.

1. **Cidades Inteligentes**

Segundo Rizzon Janine Bertelli (2017) atualmente Smart City é muitas vezes descrita como a ampla utilização de tecnologias da informação em infraestruturas tradicionais, outrossim vem para aprimorar a atuação conjunto dos capitais humanos e sociais. No uso de tecnologias e técnicas para inovar o setor urbano pode-se resolver diversos problemas ao mesmo tempo, garantindo assim qualidade e sustentabilidade no desenvolvimento.

Fato é que com o avanço da Inteligência Artificial (IA), aprendizado de máquina (AM) e Internet of Things (IoT), o conceito de Smart City está ficando cada vez mais ligado à realidade. Assim permitindo não só automatizar rotinas como também poupar recursos humanos em atividades banais e repetitivas.

1. **Internet das Coisas**

Para Madakam R. Ramaswamy (2015) só existe uma única definição aceitável para o conceito de Internet of Things (IoT) que é: Uma rede aberta e abrangente de objetos inteligentes que têm a capacidade de se auto-organizar, compartilhar informações, dados e recursos, reagindo e agindo diante de situações e mudanças no ambiente. Posto que é dependente de inovação técnica em vários campos, desde sensores sem fio até nanotecnologia.

Em primeira análise, se for analisado o cenário de desenvolvimento atual, percebe-se uma gama de equipamentos e aplicativos que já fazem uso desse conceito. A partir disso, Qian (2012) afirma que, os objetos que inicialmente serão os mais privilegiados e desenvolvidos de forma inteligente serão aqueles que integram ainda mais o ser humano com o ambiente, como por exemplo equipamentos domésticos, logísticos, elétricos e agrícolas.

1. **Sistemas Embarcados**

Segundo Cunha (2007) um sistema embarcado (SE) é conseguir colocar a capacidade computacional dentro de um circuito integrado, equipamento ou sistema. Um sistema independente, feito para conseguir fazer uma tarefa em específico.

No entanto, como bem descreve Kaelbling (2010), os SEs são difíceis de encaixar em um ambiente dinâmico, não é possível apenas com programação direta, seguindo linhas de comando simples e sem uma tomada de decisão feita pelo próprio sistema.

1. **Aprendizado Profundo *(Deep Learning)***

Para Ponti (2018) o deep learning é capaz de gerar uma função em que na sua entrada teria um conjunto de dados brutos e fornecer uma representação adequada na saída para o problema em questão. Como é dito em: “O Deep Learning permite que modelos computacionais que são compostos de múltiplas camadas de processamento aprendam representações de dados com vários níveis de reconhecimento.” (LECUN YOSHUA BENGIO, 2015). Ainda de acordo com Ponti e Da Costa (2018) a maior diferença entre aprendizado de máquina (AM) e Deep Learning (DL) seria que em AM o processo de treinamento de pesos da rede neural se daria com um único parâmetro para transformar o dado de entrada em dados de saída, porém em DL esse processo se daria com um vetor de parâmetros.

1. ***Transfer Learning***

A transferência de aprendizagem (Transfer Learning) já é costumeiramente utilizada no dia a dia humano, onde aplicamos conhecimentos prévios adquiridos de vivências ou estudos passados em uma nova tarefa que nos foi designada. Quão mais parecida for a tarefa dos conhecimentos prévios, mais fácil será para realizá-la. Para Torrey (2010) o transfer learning na área de AM, seria a capacidade de desenvolver métodos para que ocorra a transferência de conhecimento prévio daquela rede, ou seja, conseguir o “conhecimento” dos neurônios da rede de uma tarefa “padrão” e aplicá-los em uma nova rede para que melhore a aprendizagem dessa rede, aproximando ainda mais de como funciona no ser humano.

1. **Conjunto De Dados**

O conjunto de dados é uma parte fundamental no treinamento de redes neurais, pois é dele que vêm as entradas e saídas que compõem a rede. Segundo Géron (2019) uma rede só será capaz de aprender se os dados de treinamento contiverem recursos relevantes suficientes e poucos recursos irrelevantes. Uma forma encontrada para solucionar os problemas das bases de dados, como oversampling e undersampling, que são gerados pela falta de recursos relevantes, foi complementar a falta de recursos criando dados de forma artificial. Segundo Machado (2007), os dados sintéticos são amplamente usados para equilibrar conjuntos de dados que possuem classes raras, classes de dados que aparecem em menor volume do que as demais.

1. ***You Only Look Once (YOLO)***

Segundo Redmon (2018) na Yolo foram fundidos dois componentes separados da detecção de objetos utilizando caixas delimitadoras, primeiro a utilização de toda a imagem e a previsão em todas as classes. O design YOLO permite treinamento de ponta a ponta, desde o modelo da rede até a edição de Layers, e resultados em tempo real, enquanto mantém a precisão média alta. O sistema divide a imagem de entrada em uma grade S × S. Se o centro de um objeto cai em uma célula da grade (uma determinada região quadrada da imagem), essa é responsável por detectar esse objeto. Cada uma delas prevê B caixas delimitadoras e pontuações de 23 confiança para essas caixas. Essas pontuações de confiança refletem a confiança do modelo de que a caixa contém um objeto e também o quão precisa a caixa é prevista (REDMON, 2018).

**Metodologia**

O desenvolvimento de um sistema de leitura automática de placas de carro requer uma metodologia abrangente e bem definida. O processo pode ser dividido em várias etapas interconectadas que garantem a eficácia e precisão do sistema. Dessa forma iniciou-se pela definição dos objetivos e requisitos. Em seguida se deu a aquisição do *hardware,* controladores, sensores e câmera.

A criação do *dataset* é essencial para o treinamento das redes neurais e, uma vez obtidas, pode-se passar para a etapa de pré-processamento. Isso inclui redimensionamento, correção de iluminação e eliminação de ruídos para melhorar a qualidade das imagens. Dessa forma, treinou-se um IA capaz de detectar placas em imagens e vídeos, e retirar a área de interesse da foto a partir de uma base de dados fornecida pelo *Google Images.*

Então, entra em cena uma segunda IA, especializada em reconhecimento óptico de caracteres (OCR), que lê os caracteres da placa com alta precisão. Novamente, o treinamento adequado é fundamental para alcançar a exatidão desejada. Considerando que não existem uma base dados com as fontes utilizadas no padrão da resolução CONTRAN n° 231 ou padrão da resolução GMC n° 33/14, teve-se que criar um *dataset* sintético com dados que não vem da realidade.

Finalizado e testado o processo de leitura e identificação da placa cabe integrar o sistema com outros sistemas existentes, como sistemas de segurança ou de gerenciamento de estacionamento, é um passo crucial. A validação e os testes em condições reais são essenciais para garantir que o sistema atenda aos requisitos estabelecidos.

**Recursos**

1. **Treinamento**

O treinamento da YOLO-*Tiny* foi realizado com o conjunto de dados criado. O treinamento durou aproximadamente 2 horas e resultou em uma precisão de 95% no grupo de teste.

1. **Pré Processamento**

Para o pré-processamento, foram utilizados filtros de *thresholding*, escala cinza e morfologia. Esses filtros foram utilizados para melhorar a qualidade das imagens e remover ruídos.

1. **Reconhecimento de caracteres**

Para o reconhecimento de caracteres, foi desenvolvida uma rede neural convolucional (RNC) utilizando as bibliotecas Tensorflow e Keras. A rede foi treinada com um conjunto de dados de 1000 imagens de caracteres de placas de carros, cada uma com dimensões 85x55 pixels. O conjunto de dados foi criado de forma sintética, utilizando a biblioteca TextRecognitionDataGenerator (TRDG). Para armazenar as imagens geradas foi utilizada a tecnologia HDF5.

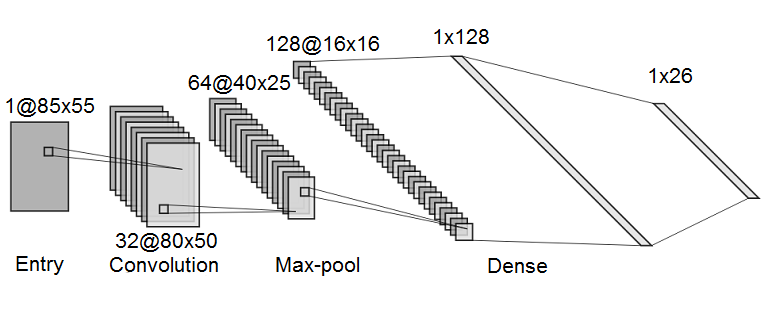


Figura 1 - Arquitetura proposta para o reconhecimento de caracteres utilizando RNC

A arquitetura da rede neural é mostrada na Figura 1. A rede possui três camadas convolucionais, com 32, 64 e 128 filtros, respectivamente. A última camada é uma camada totalmente conectada, com 26 saídas, uma para cada caractere do conjunto de dados.

A rede foi treinada com 10 épocas, utilizando um conjunto de dados de 80% para treino e 20% para teste. Os resultados do treinamento mostraram que a rede obteve uma precisão de 98% no conjunto de teste

1. **Circuito**

O protótipo de hardware do projeto consiste em um sistema de detecção de distância utilizando um Raspberry Pi, um sensor ultrassônico HC-SR04 e uma câmera de 1080p.

O Raspberry Pi foi escolhido por ser um minicomputador de baixo custo e fácil de conectar com outros componentes. A câmera de 1080p foi escolhida para capturar imagens de alta qualidade. O sensor ultrassônico HC-SR04 foi escolhido para medir a distância entre o objeto e o sistema.

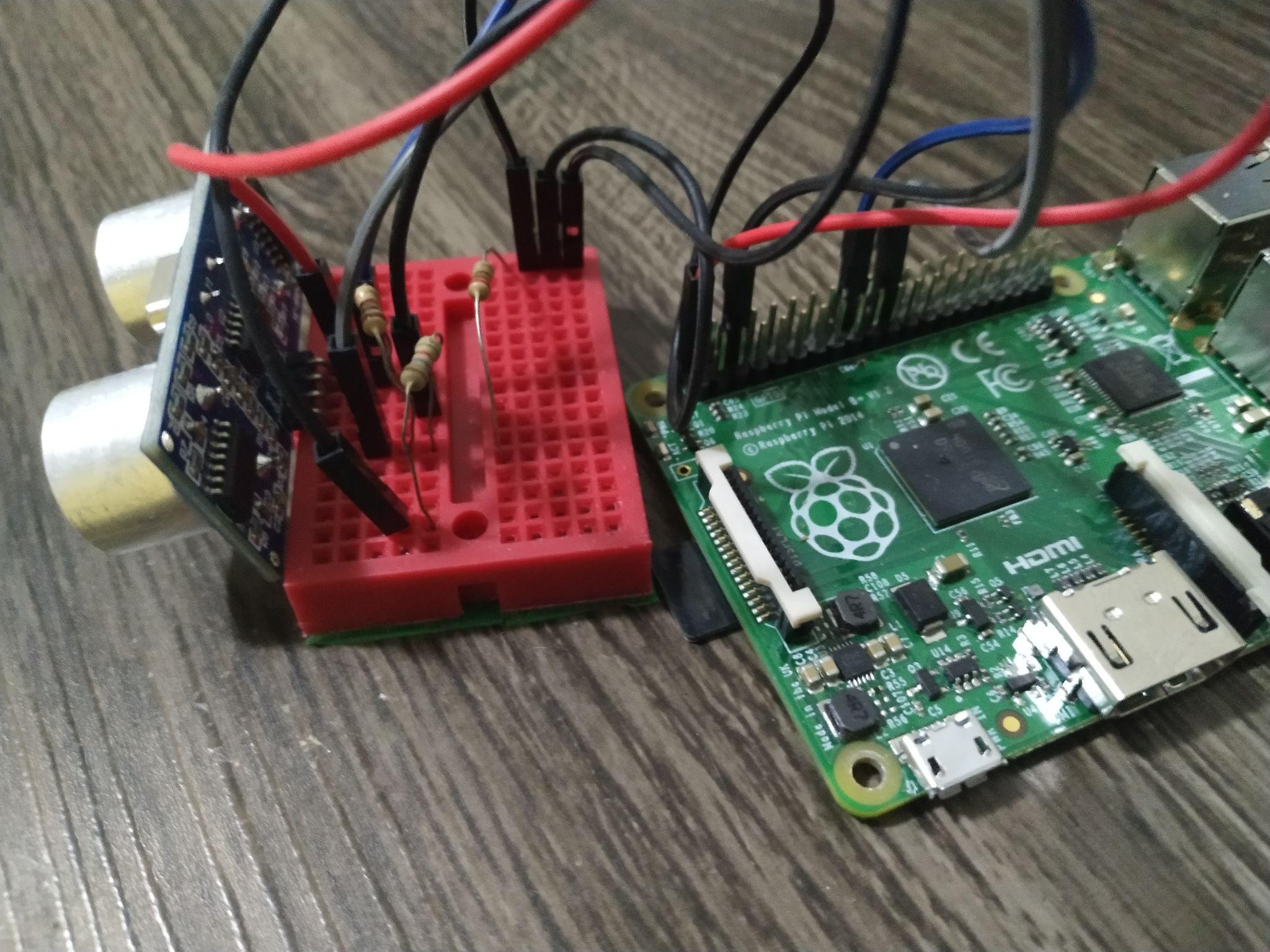


Figura 2 - Circuito montado

Na figura 2, o circuito foi montado em uma protoboard de 24x24 pinos. Para a comunicação entre o Raspberry Pi e o sensor ultrassônico foi utilizado um divisor de tensão para adequar a tensão de saída do sensor à tensão de pinagem do Raspberry Pi.

Para capturar imagens, foi utilizado um script em Python. Para compartilhar as imagens com o servidor, foi utilizado o software Samba.

O Raspberry Pi foi escolhido como o principal componente do hardware do projeto por ser um minicomputador de baixo custo e fácil de conectar com outros componentes. No entanto, durante o desenvolvimento do projeto, foi identificado que o Raspberry Pi não tinha recursos suficientes para atender às necessidades do sistema.

Em particular, o Raspberry Pi não tinha poder de processamento suficiente para executar a rede neural convolucional (RNC) responsável pelo reconhecimento de caracteres.

Para resolver esse e outros problemas, foi necessário substituir o Raspberry Pi pela Jetson Nano. A Jetson Nano é um minicomputador de alto desempenho desenvolvido pela NVIDIA. A Jetson Nano tem um processador mais poderoso do que o Raspberry Pi e mais memória RAM, o que permite que ela execute a RNC e dê mais liberdade de memória.



Figura 3 - Jetson Nano com câmera

Na figura 3, mostra como ficou o circuito usando o jetson nano. A substituição do Raspberry Pi pela Jetson Nano foi um passo importante para o desenvolvimento do projeto. Com a Jetson Nano, o sistema foi capaz de atingir o desempenho desejado.

1. **Resultados gerais**

Com o sistema feito iniciou-se a etapa de testes bem como a implantação do sistema no em algum espaço de entrada e saída de veículos. Inicialmente, fazendo uma comparação do custo da solução criada com sistemas comercializados no mercado nacional, que assim como é mostrado em Hikvision (2020), que possui um valor de R$ 10.260,90, e como pode-se ver na tabela 1 tem-se uma diferença de R$ 8.188,90 entre o preço da solução desenvolvida e da comercializada.

**Tabela 1 - Tabela de preço dos componentes usados**

| Componente | Preço |
| --- | --- |
| Jetson Nano | R$1990,00 |
| Sensor HC-SR04 | R$13,90 |
| Câmera | R$69,00 |
| Total: | R$2.072,00 |

Fonte: autoria própria

Já na parte do software, para a testar e indicar a acurácia da RNC Yolo de reconhecimento de placas, foi utilizado um conjunto de 100 imagens de placas veiculares para teste e generalização, de forma que quando foram processadas teve-se uma taxa de assertividade de 95%. na Figura 4 pode-se ver um exemplo desde a segmentação da placa detectada até a leitura dos caracteres.

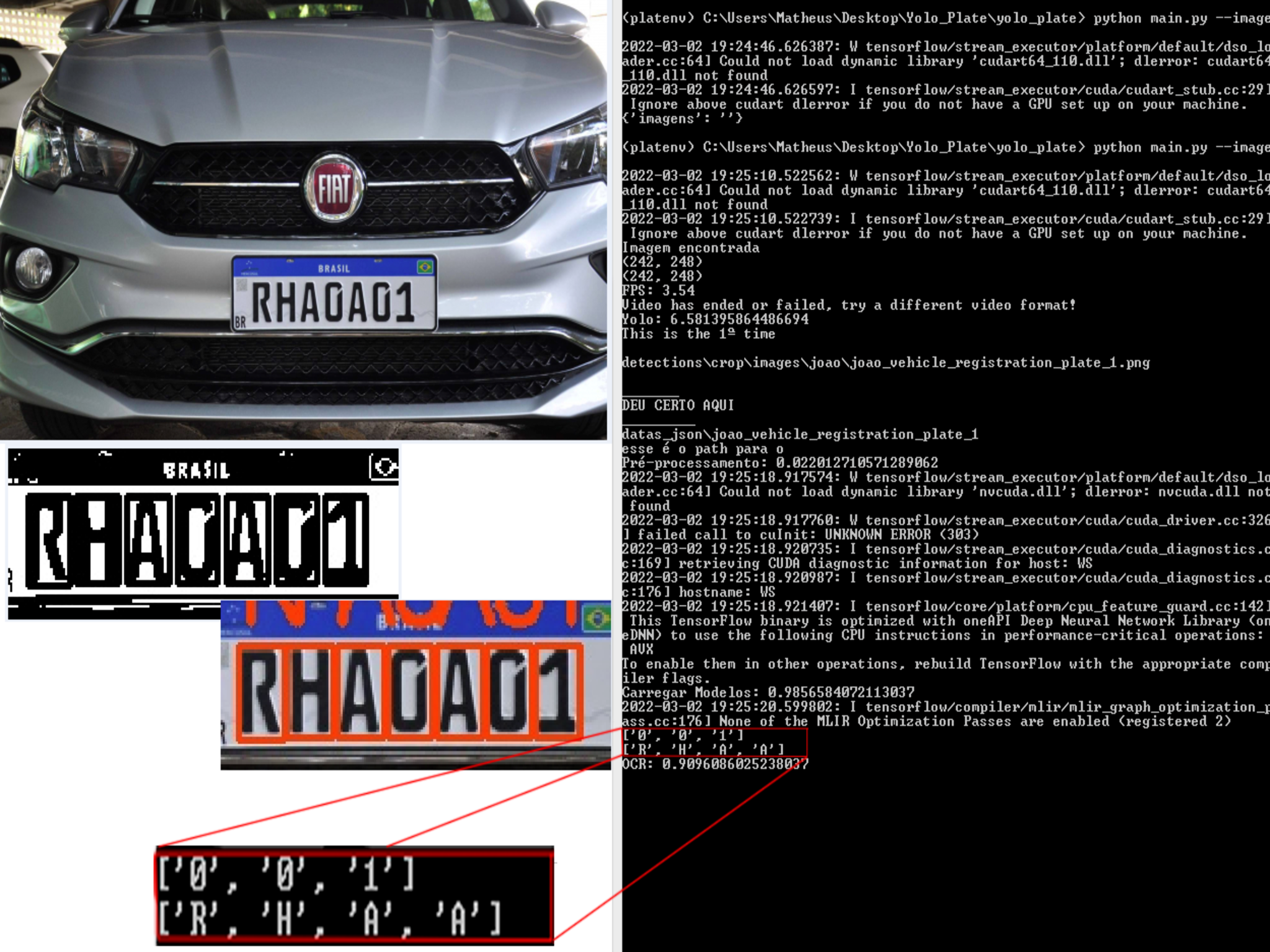


Figura 4 - Exemplo visual das três etapas da classificação

Já para a arquitetura de classificação de caracteres utilizou-se um conjunto com 100 imagens de cada caractere onde teve-se 91.1% de precisão para números e 94.2% para letras. Sendo que, o trabalho de Barreto (2018) possui uma taxa de acerto de 84% para números e 92% para letras. Na Figura 5, a seguir, pode ser observado o exemplo de uma imagem onde os caracteres foram segmentados e classificados corretamente.



Figura 5 - Exemplo de uso do sistema com placas padrão mercosul e anterior

**Cronograma**

**Tabela 2 - Cronograma de ações do projeto**

| **Etapa** | **Duração Estimada** | **Início** | **Término** |
| --- | --- | --- | --- |
| Definição dos Objetivos do Projeto | 2 semanas | [Data Inicial] | [Data Final] |
| Aquisição de Hardware | 1 semana | [Data Inicial] | [Data Final] |
| Criação dos *datasets* | 2 semanas | [Data Inicial] | [Data Final] |
| Pré-processamento de Imagens | 5 semanas | [Data Inicial] | [Data Final] |
| IA de Detecção de Placas | 2 semanas | [Data Inicial] | [Data Final] |
| Recorte da Placa | 2 semana | [Data Inicial] | [Data Final] |
| IA de Reconhecimento de Caracteres | 4 semanas | [Data Inicial] | [Data Final] |
| Integração com Sistemas Existentes | 8 semanas | [Data Inicial] | [Data Final] |
| Testes e Validação | 8 semanas | [Data Inicial] | [Data Final] |

Fonte: Autoria própria

**Referências**

CONTRAN. Resolução contran nº 231 de 15/03/2007. Diário Oficial da União, Brasília,

DF, 2007.

CONTRAN. Resolução nº 780, de 26 de junho de 2019. Diário Oficial da União, Brasília,

DF, 2019.

CUNHA, A. O que são sistemas embarcados. Saber Eletrônica, 2007.

GÉRON, A. Hands-on Machine Learning with Scikit-Learn, Keras TensorFlow. [S.l.: s.n.], 2019. 851 p.

HIKVISION. Ds-2cd7a26g0/p-iz(h)s. 2020. Disponível em: <<https://www>. hikvision.com/pt-br/products/IP-Products/Network-Cameras/DeepinView-Series/ DS-2CD7A26G0-P-IZ-H-S/>.

KAELBLING, L. P. Learning in embedded systems. MIT Press, 2010.

LECUN, Y.; BENGIO, G. H. Y. Deep learning. Nature, 2015.

MADAKAM, R.; RAMASWAMY, S. T. S. Internet of things (IoT): A literature review. Journal of Computer and Communications, 2015.

PONTI, M. A. Como funciona o deep learning. Tópicos em Gerenciamento de Dados e Informações, 2018.

QIAN, Y-j.; WU, Zhi-hong. Tecnologia da Internet das Coisas e pesquisa de aplicativos. College of Communication Engineering, Jilin University, 2012.

REDMON, E. A.; YOU, J. You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, p. 779–788, 2016.

RIZZON, Janine Bertelli et al. Smart city: um conceito em construção. Revista Metropolitana de Sustentabilidade, 2017.

ROSÁRIO, J. M. Automação Industrial. [S.l.]: Baraúna, 2009. v. 1.

TORREY, J. S. L. Transfer learning. Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, 2010.