Reconhecimento de placa por câmera

**Thiago Henrique do Couto Pereira ¹, Uriel Gonçalves Paiva da Conceição ².**

1Universidade do Vale do Paraíba/Instituto de Pesquisa e Desenvolvimento, Avenida Shishima Hifumi, 2911, Urbanova - 12244-000 - São José dos Campos-SP, Brasil, autor1@xxx.exemplo.br, autor2@xxx.exemplo.br.

2Universidade de São Paulo/Departamento de Geografia, Avenida Prof. Lineu Prestes, 338, Cidade Universitária - 05508-000 - São Paulo-SP, Brasil, autor3@xxx.exemplo.br.

**Resumo**

O sistema de reconhecimento de placa dos veículos será voltado para a área de controle de acesso para portarias. O mesmo irá facilitar o acesso de usuários cadastrados no sistema pela placa do seu veículo, identificando-a e liberando ou barrando seu acesso.

O reconhecimento da placa será feito através de uma câmera Ip ou também pode ser feito por um webcam conectado ao computador. O sistema registra uma foto a cada períodos de tempo e converte o código da placa para texto , assim sendo possível a comparação com as placas cadastradas dentro do banco de dados, trazendo segurança e rapidez para o controle de acesso.

**Palavras-chave**: Controle de acesso, reconhecimento de placa, segurança.

**Área do Conhecimento:** Engenharias – Engenharia da computação

**Introdução**

Na atual situação do nosso país, o controle de acesso de usuários é indispensável, isto é, o gerenciamento de acesso em diferentes ambientes, garantindo a segurança e integridade das pessoas, com principal objetivo, bloquear os indivíduos não autorizados (Segware performace em segurança - 2023).

Comumente o software que realiza o controle de acesso é mais utilizado em ambientes com uma grande quantidade de movimento como exemplo condomínios, hospitais, instituições de ensino e eteceteras. Porem também pode se ter sua utilização em menor escala como exemplo o estacionamento de um estabelecimento ou a garagem de algum domicilio.

Seja eleo motivo da utilização do controle de acesso, a segurança pessoal e patrimonial é um risco comum em todos os lugares do país, entretanto a necessidade do investimento em um sistema de controle de acesso adequado é cada vez mais solicitada.

Uma das principais vantagens do controle de acesso além de trazer segurança ele traz a possibilidade de visualizar eventos de entrada e saída, assim tendo o total domínio do fluxo de movimentação do estabelecimento.

Com o sistema de reconhecimento por placa do veículo, uma câmera posicionada estrategicamente possibilita a captura de imagens geradas por períodos de tempo da parte frontal de algum automóvel que deseje intencionalmente adentrar em um estabelecimento.

O software tem como principal função identificar a placa e trabalhar a imagem com rapidez para comparar a leitura no presente momento com o código cadastrado no sistema pelo operador, para que assim o sistema identifique e mande uma resposta de liberação ou bloqueio de acordo com o banco de dados.

O registro apresentado é importado para o banco de dados para geração de relatórios e identificação do acesso para os operadores, sendo possível filtrar pela data, placa ou nome de usuário, obtendo informações detalhadas do acesso.

**Metodologia**

O presente projeto apresenta um programa em Python capaz de fazer o Reconhecimento de placa por câmera. Para tal é utilizado a linguagem de programação Python, com essa linguagem o trabalho com processamento de imagem se torna mais intuitivo, tendo em vista as bibliotecas disponíveis.

Foi adotado o SGBD MySQL para a construção do banco de dados, contudo sendo possível o armazenamento de informações dos usuários, veículos e operadores do sistema.

Para o processo de desenvolvimento foram utilizadas as seguintes bibliotecas: *OpenCV, Tkinter, PySimpleGUI, OS, time, mysql.connector, Tesseract, Pillow, RE*.

A biblioteca *OpenCV* (*Open Source Computer Vision*) é amplamente utilizada em projetos de visão computacional. Neste projeto, o *OpenCV* foi utilizado para a captura de vídeo da câmera e processamento das imagens. Ela fornece funções para detectar, segmentar e reconhecer objetos em tempo real (REFERENCIA, 2003). No contexto do controle de acesso por placa de veículo, o *OpenCV* foi utilizado para extrair a placa de veículo das imagens capturadas pela câmera e ajustar a imagem salva em um arquivo na estação local para realizar a leitura pela biblioteca *Tesseract*.

O programa desenvolvido sempre realiza o processamento da última imagem capturada. Após a captura da imagem foram realizadas as seguintes etapas de processamento da imagem contendo a placa de veículo (*Corporation; Intel* - 2023):

1. Ajuste da imagem em escala de cinza:

Nessa etapa, a imagem colorida é convertida para escala de cinza. Isso é feito utilizando a função *cv2.cvtColor()* do *OpenCV*, que converte a imagem para o espaço de cores em escala de cinza.

1. Transformação da imagem em matriz binária:

Após o ajuste em escala de cinza, a imagem é binarizada, convertendo os pixels em apenas duas cores: preto e branco. Isso é realizado por meio de técnicas de limiarização (*thresholding*) utilizando a função *cv2.threshold()* do *OpenCV*. O objetivo é tornar a placa de veículo mais destacada e facilitar a detecção dos caracteres.

1. Ajuste da saturação em preto e branco:

Nessa etapa, a imagem binarizada é ajustada para aumentar o contraste entre as áreas escuras (placa de veículo) e as áreas claras (fundo). Isso é feito utilizando técnicas de equalização de histograma, por meio da função *cv2.Thresh\_otsu()* mais o *cv2.Thresh\_binary()* do *OpenCV*. O objetivo é melhorar a legibilidade dos caracteres da placa.

Essas etapas de pré-processamento da imagem no *OpenCV* ajudam a melhorar a qualidade e a legibilidade da placa de veículo, permitindo um reconhecimento mais preciso dos caracteres por parte do *Tesseract OCR.*

O Tkinter é uma biblioteca gráfica padrão do Python que permite a criação de interfaces gráficas de usuário (GUI). Neste projeto, *Tkinter* foi utilizado para criar a interface do programa, incluindo janelas, botões e caixas de texto. Através do *Tkinter*, foi possível fornecer uma interface amigável para interação com o usuário.

A biblioteca *PySimpleGUI* é uma biblioteca Python que facilita a criação de interfaces gráficas de usuário. Ela foi utilizada em conjunto com o *Tkinter* para simplificar a construção da interface do programa. *PySimpleGUI* fornece recursos adicionais, como layouts flexíveis e elementos de interface pré-configurados, contribuindo para o desenvolvimento rápido e eficiente do programa.

Foi utilizado a biblioteca *OS* pois a mesma fornece uma interface para interagir com o sistema operacional. Neste projeto, a biblioteca *OS* foi utilizada para acessar e manipular os arquivos do sistema, permitindo a leitura e gravação das imagens capturadas pela câmera.

A biblioteca *Time* fornece funções relacionadas à medição e manipulação de tempo. Neste projeto, a biblioteca *Time* foi utilizada para implementar temporizações, como pausas e atrasos, para controlar o fluxo do programa.

A biblioteca *MySQL.connector* é essencial para se conectar e interagir com bancos de dados MySQL em Python. Neste projeto, a biblioteca foi utilizada para armazenar e recuperar dados relacionados aos veículos e seus respectivos acessos. Ela permite a execução de consultas SQL para inserção, atualização e recuperação de informações do banco de dados.

A função da biblioteca *Tesseract* no programa é realizar o reconhecimento óptico de caracteres (OCR) na placa de veículo pré-processada pelo *OpenCV*. O *Tesseract* é uma biblioteca de OCR de código aberto amplamente utilizada, capaz de extrair texto de imagens. No contexto do controle de acesso por placa de veículo, o *Tesseract* é empregado para identificar os caracteres presentes na imagem da placa binarizada e convertê-los em texto legível.

Ao receber a imagem pré-processada da placa de veículo, o *Tesseract* utiliza algoritmos de processamento de imagem e aprendizado de máquina para analisar e reconhecer os caracteres presentes na imagem. Ele examina cada região da imagem, identifica os padrões dos caracteres e os interpreta para gerar o texto correspondente.

O uso do *Tesseract* no programa permite automatizar o processo de leitura das placas de veículo, proporcionando uma maneira eficiente e precisa de obter as informações contidas nas placas e utilizá-las para controle de acesso.

Com a *Pillow*, é possível redimensionar imagens, ajustar a resolução e realizar operações de escalamento. Essas funcionalidades são úteis para adaptar as imagens capturadas pela câmera para o tamanho adequado e resolução desejada, assim exibindo a imagem da câmera na tela de controle de acesso.

E por fim a biblioteca "*RE*" em Python é utilizada para trabalhar com expressões regulares, permitindo a manipulação e validação de padrões de texto. No contexto de manipulação de CPF (Cadastro de Pessoa Física), a biblioteca "*RE*" pode ser utilizada para verificar se um CPF fornecido está em um formato válido e para realizar manipulações específicas nesse tipo de dado.

**Agradecimentos***:*

Caro Professor,

Gostaria de expressar minha profunda gratidão pelo seu apoio e orientação durante a elaboração do meu Trabalho de Conclusão de Curso. Sua dedicação, conhecimento e feedbacks construtivos foram fundamentais para o sucesso deste projeto. Sou extremamente grato por toda a ajuda e suporte que você me proporcionou.

**Resultados**

Os resultados apresentados neste projeto foram satisfatórios.

Foi possível realizar a leitura e o processamento de diversos modelos de placas de veículos diferentes com exido. Nos testes houve uma grande porcentagem de acertos no reconhecimento das letras e números das placas, em alguns casos ocorreram algumas divergências.

É perceptível que nas placas de modelo da MERCOSUL há uma confusão no sistema para diferenciar o número 0 e a letra O, ocorrendo erros na leitura de certas placas.

Outro ponto que é valido ressaltar é o enquadramento da imagem, em fotos onde apresenta o para-choque inteiro do carro, em alguns casos o processamento da imagem reconhece sombras ou alguma característica física do carro como algum caractere, apresentando uma resposta errada.

Para solução deste problema é necessário realizar a tratativa da cor do fundo reconhecido, assim verificando se o quadrado lido é uma placa ou não.

**Discussão**

Para este projeto foi utilizado como referencia a Metodologia de leitura desenvolvida pelo Bruno Uemura. Portanto foi aprimorado o processamento da imagem utilizando o whitelist no tesseract, facilitando na leitura dos caracteres utiizados na placa, já no projeto modelo é utilizado o blacklist do tesseract, aumentando a margem de erro devido ler caracteres especiais e não só letras maiúsculas e números.

Outro ponto a se reparar é a leitra em tempo real, neste projeto a cada 15 segundos como padrão ou um tempo especificado pelo usuário é registrado uma foto, fazendo a verificação de liberação e bloqueio.

Foi adicionado interface gráfica para criação de operadores, login, cadastro de veículos e monitoramento pelo software.

**Conclusão**

Submeta seu artigo respeitando a data limite para tal, que será rigorosamente respeitada.

A próxima seção ilustra o formato a ser seguido para referências de livros, teses e obras completas; capítulos de livros; periódicos; anais de congressos e publicações eletrônicas.

**Referências**

Axmark, David. MySQL Connector/Python Developer Guide. MySQL. Disponível em: https://dev.mysql.com/doc/connector-python/en/. Acesso em: 17 maio 2023.

Catunda, Heitor. JANELA PARA CÓDIGO COM PYSIMPLEGUI – TELA PARA CÓDIGOS EM PYTHON! Hashtag. Disponível em: https://www.hashtagtreinamentos.com/janela-para-codigo-com-pysimplesgui-python. Acesso em: 19 maio 2023.

Clark, Jeffrey. Python Imaging Library (Fork). Pillow.org. Disponível em: https://pypi.org/project/Pillow/. Acesso em: 19 maio 2023.

Corporation, Intel. OpenCV Tutorials. OpenCV. Disponível em: https://docs.opencv.org/4.x/d9/df8/tutorial\_root.html. Acesso em: 14 maio 2023.

McKinney, Wes. Miscellaneous operating system interfaces. Python. Disponível em: https://docs.python.org/3/library/os.html. Acesso em: 16 maio 2023.

Neeleman, Daniel. O que é controle de acesso e para que serve?. Segware performace em segurança. Disponível em: https://www.segware.com/post/o-que-e-controle-de-acesso-e-para-que-serve. Acesso em: 12 maio 2023.

Packard, Hewlett. Tesseract User Manual. Tesseract. Disponível em: https://tesseract-ocr.github.io/tessdoc/. Acesso em: 19 maio 2023.

Van Rossum, Guido. Interface Python para Tcl/Tk. Python. Disponível em: https://docs.python.org/pt-br/3/library/tkinter.html. Acesso em: 15 maio 2023.

Van Rossum. Operações com expressões regulares . Pynthon. Disponível em: https://docs.python.org/pt-br/3/library/re.html. Acesso em: 19 maio 2023.

Van Rossum. Time access and conversions. Python. Disponível em: https://docs.python.org/3/library/time.html. Acesso em: 16 maio 2023.

Belvis, Ricardo. et al. Um Sistema de Reconhecimento Automático de Placas de Automóveis. 1999. Número de folhas ou volumes. (Departamento de Engenharia Elétrica) – Faculdade de Engenharia Industrial, São Bernardo do Campo, SP, Brasil, 1999.