Ferramenta para Validação de Imagens em Estações de Rádio Base usando Reconhecimento de Texto em Cenas Naturais

A Tool for Images Validation in Cell Sites using Text Recognition in Natural Scenes

José Antônio dos Santos [1] orcid.org/0000-0002-1917-3003
Carmelo Bastos-Filho [1] orcid.org/0000-0002-0924-5341
Victor Mendonca de Azevedo [2] orcid.org/0000-0003-2943-4622

[1] Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil
[2] Fundação para Inovações Tecnológicas – FITEC, Recife, Brasil

E-mail do autor principal: José Antônio dos Santos josedossanos@outlook.com

Resumo

Atualmente, empresas do segmento de construção, instalação e manutenção de estações de rádio necessitam criar relatórios com informações e fotos reais para comprovar cada serviço prestado. A criação desse relatório pode ser lento, custoso e imprevisível devido ao processo manual envolvido na captura incorreta das imagens. Por outro lado, técnicas de visão computacional podem diminuir significantemente o tempo e o custo dessa atividade, evitando capturas ilegíveis ou com informações incorretas nas imagens placa da estação. Dessa forma, esse trabalho tem como objetivo propor uma ferramenta móvel para realizar uma validação dessas imagens da placa da estação, utilizando técnicas de visão computacional. Com isso, foi desenvolvida uma ferramenta utilizando a linguagem Python, a rede pré-treinada EAST e as bibliotecas Tesseract e Kivy. Como resultado para as imagens da placa esse método conseguiu extrair corretamente o texto chave predeterminado. Entretanto, para as imagens diferentes da placa ainda necessita de alguns ajustes para extrair o texto chave. O objetivo desse trabalho foi atingido, pois, a ferramenta foi capaz de validar a imagem da placa. Contudo, novas estratégias serão analisadas para melhorar o reconhecimento do texto.

Palavras-Chave: Texto em cenas naturais; detecção de texto; reconhecimento de texto; OCR.

Abstract

In the construction, installation, and maintenance of Radio station, employees need to create reports with information and real photos to prove that each provided service was accomplished. The creation of this report is generally slow, costly, and unpredictable. This occurs mainly due to the manual process involved in the incorrect acquiring of the images. On the other hand, computer vision techniques can significantly decrease the time and cost of this activity, avoiding illegible or incorrectly captures of the station board images. Thus, this work aims to propose a mobile tool to perform a validation of these images of the station board, using computer vision and artificial intelligence techniques. Thus, a tool was developed using the Python language, the pre-trained network EAST, and the Tesseract and Kivy libraries. We validated the approach in real-world cases, and the method was able to extract the default key text correctly. However, in non-board images, the proposal still needs some tuning to extract the key-text correctly. The primary goals of the research were accomplished since the tool was able to perform a validation of the board image. We intend to include new strategies to improve text recognition capabilities.

Key-words: *Scene text; text detection; text recognition, OCR.*

DOI: 10.25286/repa.v5i2.1358

1 Introdução

Com o aumento na utilização das novas tecnologias da informação e da comunicação pela sociedade, surge a demanda pela produção de equipamento eletrônicos, e a instalação da infraestrutura necessária para suportar a grande quantidade de envios e transmissões pelos usuários.

As Estações Rádio Base (ERBs) fornecem sinais de rádio usados para transmitir e receber informações entre uma ou mais partes envolvidas em uma comunicação. Com isso, as prestadoras de serviço, denominadas operadoras, montam suas próprias ERBs para fornecerem serviços aos seus usuários. Para manter o foco no serviço prestado, tais operadoras optam por terceirizar a construção e manutenção dessas estações, dessa forma, essa empresa terceirizada fica encarregada de construir a estação, instalar equipamentos que foram enviados e requisitados pela operadora nas configurações definidas pela mesma. Por fim, a empresa fica encarregada de coletar e enviar relatórios com informações e fotos reais de cada parte da instalação que esteja finalizada comprovando a construção e a instalação correta de cada um dos equipamentos.

Para essa etapa de validação, um ou mais técnicos de campo devem fotografar as imagens comprovando os itens da instalação para uma outra equipe, que deve analisar os itens enviados e verificar se elas estão nos padrões que permitam reconhecer de qual estação é, e os equipamentos instalados. Esse processo pode se repetir até que a captura esteja no padrão definido.

Contudo, essas atividades podem se tornarem lentas e custosas para ambas as equipes. Dessa forma, a finalização só pode ser concluída quando todas as imagens estiverem dentro desse padrão. A possibilidade de repetição dessa atividade pode torná-la lenta e imprevisível. Além disso, todas essas atividades desde a captura das imagens até a validação são realizadas de forma manual, tornando um processo custoso.

Entretanto, através de técnicas de visão computacional é possível diminuir a quantidades de vezes em que um item é reprovado, tendo em vista que, a maioria das reprovações são imagens ilegíveis ou com informações incorretas. Um dos itens com grande quantidade de rejeições, é o que contém a imagem da placa, que possui informações sobre localização de cada ERBs, operadora, um código de identificação único, entre outras.

Com isso, o objetivo desse trabalho é propor uma ferramenta móvel para realizar uma pré-validação da imagem da placa da estação, utilizando técnicas de visão computacional disponíveis com base nas informações de cada ERBs.

O trabalho este organizado da seguinte forma, Na Seção 2 é apresentado o Referencial Teórico que embasam a abordagem da solução do problema mencionado anteriormente. Na Seção 3 é explanado o método que foi utilizado para atingir o objetivo aqui proposto. Na Seção 4 são apresentados e discutidos os Resultados atingidos. Por fim a Conclusão e Trabalhos Futuros, com possíveis melhorias para ferramenta, podem ser encontrados na Seção 5.

2 Referencial Teórico

Nesta Seção é apresentado uma breve explanação acerca dos conteúdos utilizados neste trabalho e um trabalho relacionado.

2.1 Visão Computacional

A Visão Computacional busca descrever o mundo que vemos em uma ou mais imagens e reconstruir suas propriedades, como forma, iluminação e distribuição de cores, de forma a ser compreendida computacionalmente [1]. Atualmente a Visão Computacional vem sendo usada em uma ampla variedade de aplicações no mundo real, algumas delas como: Reconhecimento Óptico de Caracteres (OCR), Detecção de falhas em equipamentos, Reconhecimento de objetos, Construção de modelo 3D, Imagem médica ente outras subáreas [1].

2.2 Reconhecimento Óptico de Caracteres

Na Visão Computacional, o ramo que é usado para detectar e extrair caracteres de documentos, imagens digitalizadas ou texto escrito a mão e transformar em formato de texto editável, é chamado de OCR [2]. Essa técnica permite que a máquina reconheça o texto e seja capaz de realizar operações usando esse resultado.

Atualmente é possível encontrar muitos tipos de software de OCR disponíveis no mercado como: Desktop OCR, Server OCR, Web OCR etc. A taxa de precisão de qualquer ferramenta OCR varia de 71% a 98%. Apesar das inúmeras ferramentas de OCR que estão disponíveis, apenas algumas delas são de código aberto e gratuitas [3].

2.3 Trabalho Relacionado

No trabalho de [4], teve como objetivo a identificação e a leitura dos caracteres de placas veiculares em uma imagem. Para isso, criaram uma detector de objetos com o método conhecido com *Haar-cascade*, para detectar placa de veículos, logo após, utilizaram o efeito "*Threshold*" para destacar as características textuais e tão assim aplicar o Tesseract para reconhecer os caracteres contido na placa [4]. Na abordagem desse trabalho, foram utilizadas técnicas que dispensam o uso de detector de objetos e várias camadas de filtros ou efeitos na imagem, entretanto, é feito o uso de um robusto detector de palavras em cenas com condições naturais.

3 Método

Para atingir o objetivo dessa pesquisa, inicialmente ocorreu análise das imagens capturadas por técnicos de campo para o item referente a imagem de placa, essas capturas foram obtidas de uma empresa que realiza a montagem, instalação e manutenção de equipamentos e ERBs. Com base nessa análise, foram identificadas duas formas diferentes para o item ser aprovado pela outra equipe, são elas: a captura da imagem da placa ou a capa do documento do projeto da estação, ambas correspondentes a cada estações em questão.

Neste sentido, a validação da placa deve ocorrer com base nas informações contidas nas mesmas ou na capa do documento do projeto, caso a estação não possua placa ainda. Dessa forma, houve a necessidade de determinar uma informação que é única para cada estação, além disso, essa mesma informação deve estar presente na placa e na capa do projeto preliminar, uma vez que na ausência de um o outro é usado como substituto. Diante disso, torna-se possível verificar a existência dessa informação, que nesse caso é o código de identificação da estação nas imagens.

Tendo em vista que, o código de identificação da estação é uma chave alfanumérica contendo na maioria dos casos encontrados 7 caracteres que juntos referem-se a uma determinada estação, a validação pode ser feita através dessa chave. Ou seja, se a mesma estiver presente e visível na imagem, independentemente de estar contida em uma placa ou uma folha de papel, já é aceito como uma pré-validação.

Com isso, a fermenta abordada neste trabalho para atender esses requisitos está dividida em três módulos diferentes apresentados na **Figura 1**. Essa abordagem foi definida devido as condições encontradas na base de imagens, como: a falta de padronização das placas quando havia placa nas fotos, má iluminação, variação da rotação e diferentes perspectivas.

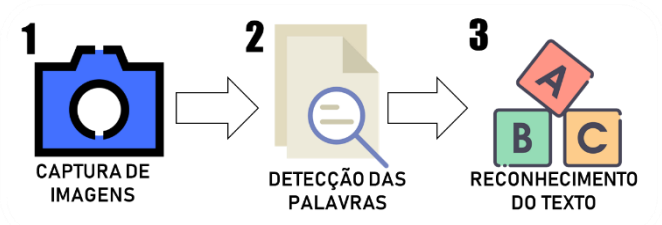

Figura 1: Módulos do sistema.

No primeiro módulo, é realizado a captura das imagens em tempo real, que além disso, é realizado um pré-processamento na mesma. Dessa forma, cada imagem capturada passou por um redimensionamento de tamanho para atender um pré-requisito do módulo posterior, que determina o valor da resolução de cada imagem deve ser múltiplo de 32 e conter 3 canais de cores RGB. Dessa forma, foram testadas as seguintes variações de resoluções: 32x32, 160x160, 256x256, 320x256, 480x480, 640x480 e 3200x3200. Entretanto, os valores definidos para serem utilizados nessa base de imagens foram 320x320, contendo os 3 canais de cores RGB que foram capturados originalmente, ademais na seção de Resultados essa melhora será explanada.

Em seguida, cada imagem é submetida ao segundo módulo, Detecção das palavras, que verifica a existência de texto na imagem, neste módulo é utilizado uma rede neural convolucional pré-treinada chamada de *Efficient and Accurate Scene Text Detector* (EAST). O EAST funciona em conjunto com a biblioteca multiplataforma de processamento de imagens OpenCV, e é utilizado para produzir diretamente previsão de palavra ou linha de texto, sua

DOI: 10.25286/repa.v5i2.1358

escolha justifica-se pelo motivo de apresentar capacidade de lidar com vários cenários desafiadores, como iluminação não uniforme, baixa resolução, orientação variável e distorção de perspectiva [5]. Tais características estão presentes nas imagens obtidas pela empresa.

Com isso, para cada texto detectado em uma imagem será guardado temporariamente suas posições de início e fim em cada um dos eixos *x* e *y*. Com essa abordagem é possível obter recortes sobre a imagem com textos em diferentes tamanhos, rotações ou perspectivas que estejam presentes na mesma imagem. Com isso, cada palavra detectada é submetida ao terceiro módulo, que é responsável identificar cada caracter desse recorte da imagem e associar uma palavra válida da língua portuguesa.

No terceiro módulo, Reconhecimento do texto, é utilizado a biblioteca de Tesseract, que é uma biblioteca de código aberto desenvolvida inicialmente pela Hewlett-Packard e atualmente é mantida pela Google. A biblioteca tem o objetivo de ler textos e caracteres de uma imagem, ou seja, tem a capacidade de transformar a imagem de um texto em arquivos editáveis de texto [2,4,6].

Seu uso neste trabalho justifica-se pelo motivo de ser multiplataforma, permitindo utilizar desde computadores com sistema operacional Microsoft Windows, sistemas baseados em Unix ou até mesmo dispositivos móveis como Android. Além disso, também é adequado para reconhecer um maior número de dados de texto e reduz significativamente os erros criados no processo de reconhecimento de caracteres [2].

O módulo Reconhecimento do texto só é executado se houve uma ou mais palavra recebida do módulo anterior. Ao receber um recorte de imagem com a palavra é aplicado um filtro para escala de cinza na mesma para poder ser utilizado no Tesseract, que exige apenas um recorte da imagem por vez com apenas um canal de cor. Para cada recorte é retornado um texto que foi reconhecido pela biblioteca de OCR reconhecido para cada recorte, que é concatenado com todos os outros textos retornados dos recortes da mesma imagem. Por fim, é removido desse texto final os espaços, tabulações e quebra de linhas existentes no mesmo.

Além de reconhecer e realizar os pré-processamentos supracitados, o módulo 3 também está responsável de buscar no texto final a existência do código de identificação da estação e salvar essa imagem com as configurações originais e com os textos detectados demarcados apenas no caso dessa chave ser encontrada.

Para desenvolvimento da ferramenta com esses módulos mencionados nesta seção, foi utilizada a linguagem de programação Python na sua versão 3.7, juntamente com a biblioteca de interface gráfica Kivy, para construção janelas e seus componentes. Essa biblioteca foi utilizada tendo em vista sua portabilidade para uma grande variedade de dispositivos incluindo celulares com o sistema operacional do Google Android.

Na **Figura 2** é possível visualizar dois protótipos de tela da ferramenta em sua versão para dispositivos móveis. Na **Figura 2** (a) apresenta a ferramenta em ação, buscando texto na imagem que está sendo capturada pela câmera. Na **Figura 2** (b) é demonstra ao usuário uma imagem capturada, que ocorre apenas quando o texto que foi dado como entrada for localizado. Nesta etapa a ferramenta fica em espera por duas possíveis ações do usuário: "Repetir" ou "Aceitar".

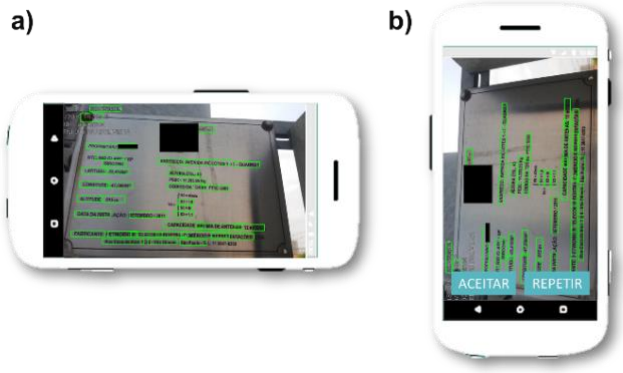

Figura 2: Protótipo da ferramenta para dispositivos móveis.

Na opção "Repetir", a ferramenta ignora a primeira captura e tenta buscar uma imagem contendo o texto que lhe foi dado. Na opção "Aceitar", a imagem capturada e armazenada no disco e uma nova busca é iniciada. Para exemplificar melhor é apresentado na **Figura 3** o fluxograma com todas funcionalidades da ferramenta.

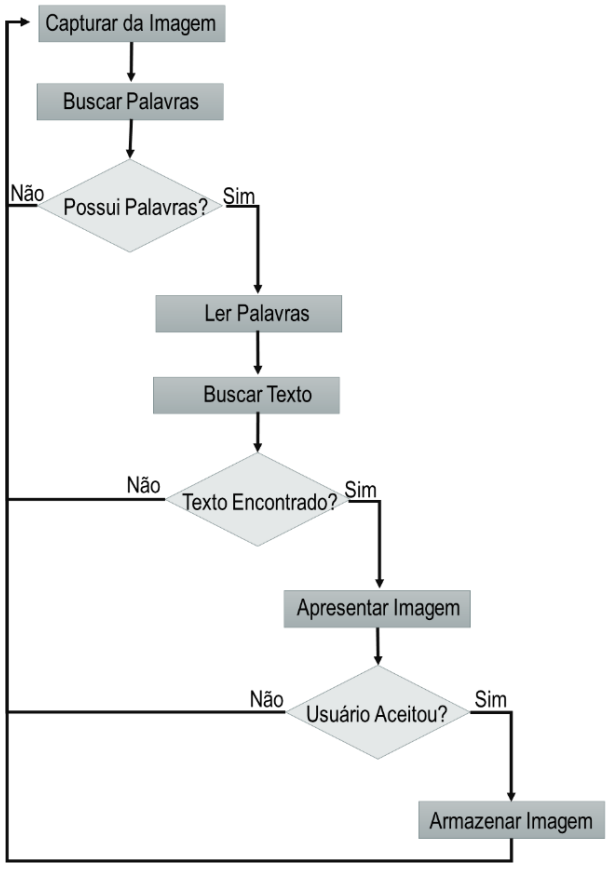

Figura 3: Fluxograma geral da ferramenta.

4 Resultados e Discussão

Como resultado obtido no segundo módulo da ferramenta, tornou possível identificar que os valores da resolução que apresentou melhor resultado nessa base de imagens foram 320x320, contendo os 3 canais de cores RGB. As resoluções: 32x32, 160x160, 256x256, 320x256, 480x480, 640x480 e 3200x3200, apresentaram uma quantidade menor de detecção de palavras detectadas.

Além disso, alguns filtros foram aplicados nas imagens como: suavização e desfoque, porém, a utilização de um ou mais filtros não apresentaram uma boa aderência nessa base de imagens e também ocasionavam na diminuição de palavras detectadas ou no reconhecimento do texto, em alguns casos ocorriam de nenhum texto ser retornado pela ferramenta.

Já para as imagens da placa, o método adotado apresenta melhores resultados, como pode ser visto um exemplo na **Figura 4** (a) que contém a imagem da placa que foi dada como entrada na ferramenta, e a **Figura 4** (b) que mostra imagem após aplicação do módulo 2, o qual detecta as palavras no texto e apresenta na tela da ferramenta. Por fim a **Figura 4** (c), demonstra o resultado obtido da ferramenta, que contém todos os textos que foram possíveis de serem extraídos pelo módulo 3.

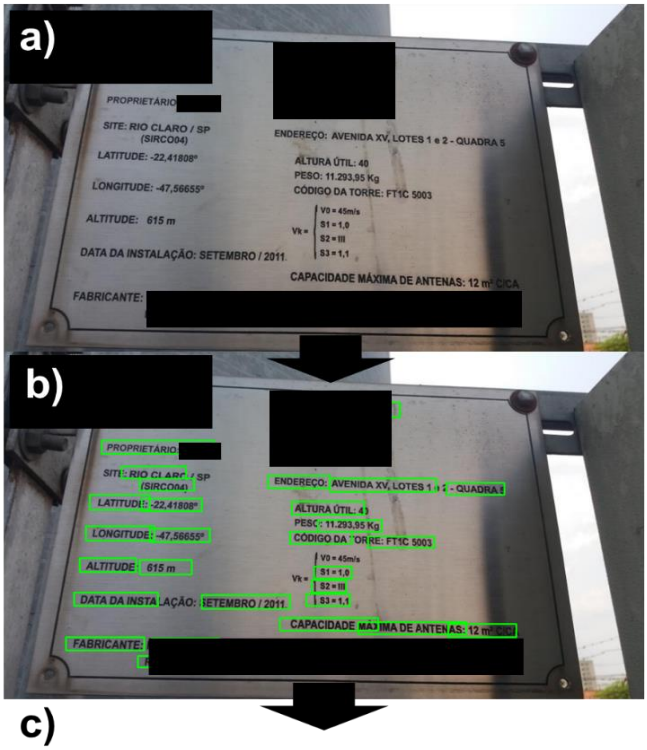

Figura 4: (a) Exemplo de imagem original da placa. (b) Imagem após aplicação do módulo 2. (c) Imagem do resultado do texto extraído pelo módulo 3.

Vale descartar que, o texto apresentado sofreu algumas pequenas mudanças, como: alguns espaços e quebras de linhas foram removidos para melhorar sua apresentação, assim como a formatação em coluna dupla para melhor ajuste. Outra alteração ocorreu quando o nome do proprietário da estação, o fabricante e seu endereço estavam expostos, nessas ocorrências a informação foi alterada para

DOI: 10.25286/repa.v5i2.1358

"(OCULTADO)", essa medida foi tomada para garantir a confidencialidade as empresas envolvidas. Dessa mesma forma ocorreu nas imagens, nas quais foram introduzidas tarjas pretas nos campos que continham as informações mencionadas nesse parágrafo.

Para as imagens com a capa do projeto, o método adotado ainda precisa de algumas melhorias, apesar da imagem que foi dado como entrada na ferramenta aparentar-se estar nítida aos olhos humanos, como ser visto na **Figura 5** (a), e os resultados da detecção das palavras também abrangerem quase por com completo o documento, como está apresentado na **Figura 5** (b), o texto que foi retornado da ferramenta não possui o a chave única utilizada para identificar a estação. Na **Figura 5** (c) é possível visualizar os textos extraídos pelo módulo 3. As capas utilizadas na **Figura *5*** foram criadas com informações de exemplo.

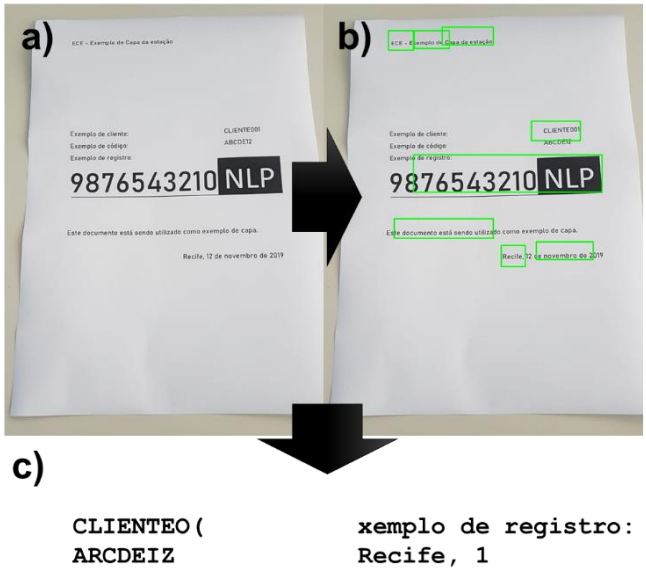

Figura 5: (a) Exemplo de capa do documento do projeto da estação. (b) Imagem após aplicação do módulo 2. (c) Imagem do resultado do texto extraído pelo módulo 3.

Vale salientar que tanto para a placa, quanto para a placa, foram utilizadas imagens obtidas de uma base criada anteriormente, as quais suas características, como iluminação e rotação, já tinham sido armazenadas. Em uma abordagem em tempo real é esperado um conjunto maior de imagem por segundos, podendo variar de acordo com a performance do dispositivo que esteja sendo utilizado, oferecendo a ferramenta maior diversidade de luz, rotação e perspectiva para tentar encontrar o texto chave.

Contudo, este caso será analisado para identificar qual módulo apresentou pior resultado. Como é possível visualizar na **Figura 5** (b), a palavra com o código foi identificada e demarcada, mesmo assim o recorte pode não ter sido realizado corretamente. Essa possível falha do módulo Detecção de palavras pode ter ocasionado mau funcionamento do módulo de Reconhecimento do texto. Mas caso tenha sido apenas no módulo 3, acredita-se que a solução pode ser a aplicação de alguns filtros que resolvam esse problema sem prejudicar a ferramenta quando for aplicada nas imagens de placa.

Para esse texto algumas alterações também foram efetuadas, como: espaços e quebras de linhas removidos, formatação em coluna tripla.

5 Conclusão e Trabalhos Futuros

Por meio dos resultados obtidos, tornou-se possível verificar que o objetivo desse trabalho foi atingido, uma vez que a ferramenta foi capaz de realizar uma pré-validação da imagem da placa, diminuindo a quantidade de capturas com baixa luminosidade, má qualidade da imagem ou ilegibilidade das informações que nela estão contidas.

Como trabalho futuros fica a aplicação de novos conjuntos de filtros e novas estratégias de pré-processamento para melhorar o reconhecimento do texto. Em um segundo momento, será estudado o desenvolvimento e implantação da ferramenta mobile, possibilitando a validação mesmo offline no local da instalação. Ainda é possível o uso de uma versão web com processamento direto no servidor.

Além disso, também fica a ser estudado, testado e validado a utilização de outras bibliotecas ou soluções para substituição dos módulos de Detecção de palavras e/ou Reconhecimento de texto. Essa é uma tentativa de buscar uma abordagem com melhor performance, tendo em vista que esse modelo utilizará recursos de um dispositivo móvel, ou com melhor assertividade na detecção/reconhecimento de palavras placas de ERBs.

Agradecimentos

Os autores agradecem a SECTI/CMA-Parqtel/UPE/FACEPE pela oportunidade de apreender e aplicar conhecimentos de forma prática e contribuir na comunidade. Em especial a Fundação para

Inovações Tecnológicas – FITec-Recife, a qual contribuiu diretamente com recursos tecnológicos e financeiros para a realização desse trabalho.

Referências

[1] SZELISKI, R. Computer Vision. London: Springer London, 2011.

[2] PAWAR, N. et al. Image to Text Conversion Using Tesseract. International Research Journal of Engineering and Technology (IRJET), v. 06, p. 516–519, 2019.

[3] PATEL, C.; PATEL, A.; PATEL, D. Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study. International Journal of Computer Applications, v. 55, n. 10, p. 50–56, 2012.

[4] LEITE, L.; ANTONELLO, R. Identificação Automática De Placa De Veículos Através De Processamento De Imagem E Visão Computacional. [s.d.].

[5] ZHOU, X. et al. EAST: An efficient and accurate scene text detector. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, v. 2017-Janua, n. April, p. 2642–2651, 2017.

[6] ADRIAN ROSEBROCK. Using Tesseract OCR with Python - PyImageSearch. Disponível em: <https://www.pyimagesearch.com/2017/07/10/using-tesseract-ocr-python/>. Acesso em: 16 nov. 2019.

DOI: 10.25286/repa.v5i2.1358