

ALGORITMO PARA DETECÇÃO DE PORTAS LÓGICAS EM AMBIENTE VIRTUAL

FELIPE MACHADO¹; JOAB TAVARES FAGUNDES²; FELIPE DIAS LOPES³;
MARCELO LEMOS ROSSI⁴

¹Universidade Federal de Pelotas – fmachado@ufpel.edu.br

²Universidade Federal de Pelotas – joabtavaresf02@gmail.com

³Universidade Federal de Pelotas – felopes_7@hotmail.com

⁴Universidade Federal de Pelotas – marcelo.rossi@ufpel.edu.br

1. INTRODUÇÃO

A visão computacional é uma área da inteligência artificial dentro da computação que se preocupa em dar às máquinas capacidade de processar e interpretar imagens do mundo real para facilitar a automatização de tarefas. Uma maneira de robôs interpretarem imagens é através de redes neurais com treinamentos de modelos capazes de criar e aplicar filtros que ajudam no reconhecimento de padrões (KRIZHEVSKY; SUTSKEVER; HILTON, 2012). Outra maneira é manualmente criar estes filtros e transmitir os resultados a outro sistema, e é nisso que este trabalho se baseia.

A eletrônica digital é uma área fundamental no entendimento das novas tecnologias e no ensino de eletrônica, pois no interior de qualquer chip eletrônico há milhões de “chaves” que ligam ou desligam, um arranjo destas chaves é capaz de executar determinado processamento lógico, a este arranjo damos o nome de porta lógica, a exemplo temos as portas AND, OR, XOR e NOT.

Com base na necessidade de acessibilidade para os mais de 35 milhões de brasileiros com alguma deficiência visual, segundo dados do IBGE de 2010 (IBGE, 2010), o presente trabalho explora uma maneira de conectar o ensino de eletrônica digital com visão computacional.

Assim, este trabalho tem como objetivo apresentar uma metodologia para que um sistema de visão computacional possa identificar portas lógicas, utilizadas em eletrônica digital, de tal forma que essa informação possa auxiliar no ensino de eletrônica digital para deficientes visuais.

Para isso foi desenvolvido um código computacional utilizando funções da biblioteca OpenCV na linguagem Python capaz de identificar, através de fotografias, portas lógicas tridimensionais dispostas em um ambiente de aprendizado. Devido a necessidade de afastamento social por causa do surto de COVID-19, foi desenvolvido um ambiente virtual na plataforma Unity para o desenvolvimento do trabalho.

2. METODOLOGIA

Primeiramente, realizou-se a criação do ambiente virtual através da plataforma Unity utilizando peças com o formato das portas lógicas que foram modeladas tridimensionalmente. Este ambiente virtual permite a montagem de circuitos digitais e fazer simulações da posição da câmera, de forma a simular um ambiente que teríamos em uma bancada.

Após isso, deu-se início a revisão teórica das funções presentes na biblioteca OpenCV aptas a identificar objetos em imagens. A função ORB

(Oriented FAST and Rotated BRIEF) foi escolhida pois permite identificar pontos semelhantes em duas imagens e, dessa forma, identificar a possibilidade da existência do objeto de uma imagem em outra imagem (RUBLEE; et al., 2011).

Neste trabalho foi utilizado uma imagem *template* quadrada de 300 por 300 pixels da porta lógica OR, com o objetivo de encontrar onde na imagem há um objeto. Verificou-se que a porta lógica utilizada como template não exerce tanta influência, já que elas são muito parecidas.

Como o ORB encontra *Keypoints* baseado em brilho. Verificou-se que obteve-se mais pontos homólogos (*matches*) quando as imagens estão em tons de cinza e com bordas realçadas por um filtro gaussiano.

Com todos os *matches* das duas imagens, criamos uma máscara com os pontos semelhantes de uma região em específico através da homografia, que é uma matriz 3x3 relacionando as duas imagens. Com esta máscara podemos através do ponto central destes *matches* recortar um retângulo de 200 pixels de lado e analisar somente um pedaço da imagem.

Este recorte da imagem original juntamente com os templates de cada porta lógica são então aplicados a outro filtro de bordas, o *Canny*, que binariza e reduz o ruído deixando somente os contornos do objeto. Aplicamos uma função, independente de rotação e tamanho, para demarcar os contornos e utilizando o maior contorno interno, comparamos o objeto do recorte com cada template de porta lógica. Obtemos um número relacionado ao grau de semelhança, onde quanto menor este número maior a semelhança.

Por fim, utilizando a informação do local da região do recorte, demarcamos um retângulo com o nome da peça com a melhor semelhança. Este mesmo local, na imagem em tons de cinza, é substituído por um retângulo sólido da cor branca, que tem o objetivo de “apagar” aquele local analisado, assim a imagem pode novamente ser utilizada para encontrar novos objetos. Quando não há *matches* suficientes provindos dos ORB, pontos suficientes na máscara ou ainda quando o recorte não possui contornos internos o algoritmo chega ao seu fim de execução.

A Figura 1 apresenta um fluxograma simplificado representando todas as operações descritas acima. Cabe ressaltar que há duas imagens iguais, uma colorida para a demarcação dos resultados, apresentada em caixa alta no fluxograma, e a mesma imagem em escalas de cinza que é analisada para encontrar os objetos.

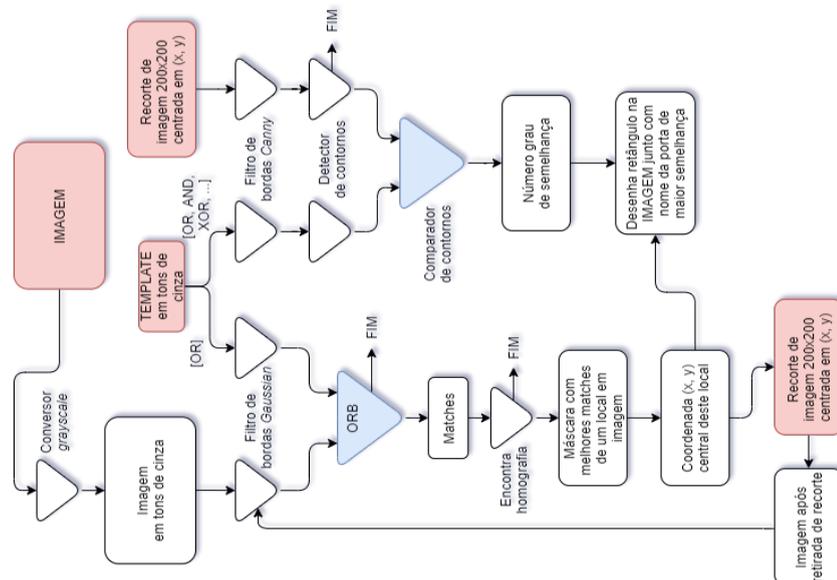


Figura 1: Fluxograma do algoritmo desenvolvido

3. RESULTADOS E DISCUSSÃO

Inicialmente realizou-se testes utilizando o próprio ORB para a identificação final, de maneira semelhante a realizada como os contornos, porém não houve bons resultados pois comparando o recorte com cada template e analisando o número de *matches* notou-se que as peças AND, OR e XOR apresentavam quantidades similares de *matches* e em contrapartida na peça CONNECT havia cerca de um terço desta quantidade devido a menor área, logo o algoritmo nunca detectava a peça CONNECT.

Devido a alta taxa de erro pela utilização de apenas o ORB para a identificação das portas lógicas foi incorporado ao algoritmo a capacidade medir o contorno interno das portas lógicas dentro de cada recorte de imagem. Comparando o tamanho dos contornos internos de cada porta lógica foi possível elevar o grau de acerto durante o processo de identificação a níveis satisfatórios.

A Figura 2 apresenta o ambiente virtual utilizado juntamente com os resultados de identificação quando já foi adicionado a capacidade de comparar os contornos internos de cada porta lógica.

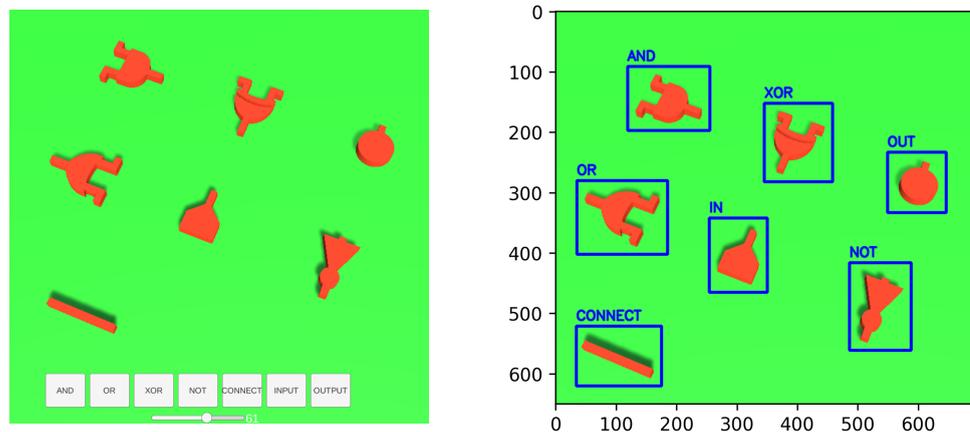


Figura 2: Ambiente virtual (esquerda) e resultados de identificação (direita)

Como podemos verificar na Figura 2, o algoritmo proposto foi capaz de identificar, corretamente, todas as portas lógicas, além dos elementos IN, OUT e CONECT que serão utilizados em trabalhos futuros. Ainda na Figura 2, podemos verificar que as portas lógicas AND, OR e XOR possuem o formato muito semelhante e ainda sim a metodologia proposta foi capaz de diferenciar estes elementos similares.

Já Figura 3 contém a etapa do processamento realizado pelo algoritmo para identificação das portas lógicas, na qual é apresentado como isso ocorre para a porta lógica XOR.

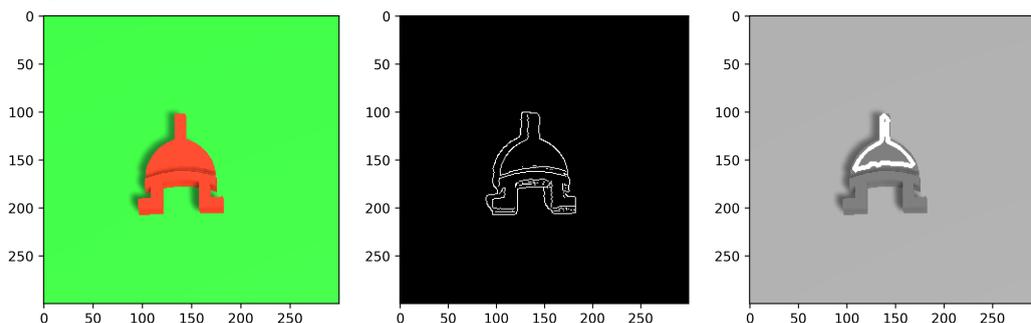


Figura 3: Etapas do processamento para identificação das portas lógicas

Uma característica importante da porta lógica XOR é que ela se assemelha muito à porta lógica OR sendo a diferença um rasgo presente na porta XOR. O que podemos ver na Figura 3 é que a metodologia proposta consegue realçar o rasgo presente na imagem da porta lógica XOR e extrair o contorno apenas da parte necessária para a identificação dela e a diferenciação em relação à porta lógica OR.

4. CONCLUSÕES

A partir dos resultados obtidos podemos indicar que o algoritmo proposto consegue satisfatoriamente identificar e diferenciar a geometria de cada porta lógica existente no ambiente virtual. Como o reconhecimento das portas lógicas é feito ao avaliar a geometria delas espera-se, para trabalhos futuros, que resultado semelhante possa ser obtido quando for possível retomar o experimento em bancadas de laboratório.

5. AGRADECIMENTOS

Os autores deste trabalho agradecem à Universidade Federal de Pelotas, ao CNPq, à Fapergs pelas bolsas de estudo concedidas e à Unity Technologies pela licença do software.

6. REFERÊNCIAS BIBLIOGRÁFICAS

IBGE 2010. **Pesquisa sobre pessoas com deficiência visual**. Acessado em 20 jul. 2021. Online. Disponível em: <https://www.ibge.gov.br/apps/snig/v1/index.html?loc=0,0R,0U&cat=-1,-2,-3,128&ind=4642>

KRIZHEVSKY, A. SUTSKEVER, I. HILTON, G. E. **Imagenet classification with deep convolutional neural networks**. Advances in neural information processing systems. p.1097-1105, 2012.

RUBLEE, E. RABAUD, V. KONOLIGE, K. BRADSKI, G. **ORB: An efficient alternative to SIFT or SURF**. **2011 International Conference on Computer Vision**, Barcelona, p.2564-2571, 2011.