



# RECONHECIMENTO DE PORTAS LÓGICAS UTILIZANDO UMA REDE NEURAL CONVOLUCIONAL COM YOLOv3 A PARTIR DE UM AMBIENTE VIRTUAL 3D

FELIPE DIAS LOPES<sup>1</sup>; EDUARDO VARGAS ZUMMACH<sup>2</sup>; FELIPE MACHADO<sup>3</sup>;  
JOAB TAVARES FAGUNDES<sup>4</sup>; MARCELO LEMOS ROSSI<sup>5</sup>

<sup>1</sup>Universidade Federal de Pelotas – fdlopes@inf.ufpel.edu.br

<sup>2</sup>Universidade Federal de Pelotas – eduardo.zummach@hotmail.com

<sup>3</sup>Universidade Federal de Pelotas – felypmachado@gmail.com

<sup>4</sup>Universidade Federal de Pelotas – joabtavaresf02@gmail.com

<sup>5</sup>Universidade Federal de Pelotas – marcelo.rossi@ufpel.edu.br

## 1. INTRODUÇÃO

Os constantes avanços pelos quais a computação vem passando na última década possibilitaram o crescimento de áreas diretamente conectadas a forma como o ser humano visualiza e interage com o mundo, como o campo da visão computacional que tem como base o reconhecimento de padrões visuais do mundo, como cor e forma de objetos que existem na natureza (TERASHIMA, 2001). O avanço das técnicas de visão computacional combinado com aumentos significativos em poder de processamento e ao grande volume de dados gerados todos os dias na internet permitiram que uma grande variedade de tarefas, ligadas ao reconhecimento de padrões, passassem a ser realizadas por máquinas com alta precisão.

Esse trabalho é resultado de uma pesquisa em andamento, cujo objetivo é realizar a detecção e classificação de peças, portas lógicas, em tempo real para auxiliar pessoas com deficiência visual a seguirem seus estudos de forma mais autônoma. Como inspiração para a pesquisa, foi tomada como base a vivência de uma estudante paranaense que perdeu completamente a visão ao decorrer da faculdade e ainda assim concluiu sua graduação (GARCIA, 2016).

Devido a pandemia de COVID-19 e a necessidade de manter distanciamento social, o acesso ao laboratório ficou impossibilitado e, conseqüentemente, o uso das peças físicas. Como forma de contornar este problema foi desenvolvido um ambiente virtual utilizando a plataforma Unity 3D (UNITY, 2020) para dar sequência ao projeto de maneira remota.

De acordo com Correia (2016) a ferramenta Unity 3D possibilita a criação de objetos com alto nível de detalhamento gráfico, o que a torna ideal para o desenvolvimento de ambientes 3D realistas. Dessa forma, buscou-se, no desenvolvimento do ambiente virtual, simular o cenário real de aplicação, através da simulação de iluminação, sombras e texturas, além de utilizar peças virtuais com geometrias idênticas às peças físicas.

A criação do ambiente virtual possibilitou a obtenção de um grande conjunto de imagens das peças virtuais, proporcionando, assim, a construção de um dataset para realizar o treinamento de uma Rede Neural Convolutiva. De acordo com Kriehzhevsky (2012), as Redes Neurais Convolucionais são um importante ramo da Inteligência Artificial, uma vez que elas possibilitam que computadores busquem solucionar problemas que envolvem visão computacional.

Para o processo de reconhecimento utilizou-se o framework YOLOv3, o qual foi desenvolvido para detecção e classificação de imagens. Segundo Redmon e Farhadi (2018), o YOLOv3 possui um bom equilíbrio entre velocidade e precisão,

sendo capaz de realizar inferências em vídeos a uma taxa de 30 quadros por segundo, em média, o que o torna ideal para reconhecimento de padrões em tempo real.

## 2. METODOLOGIA

Inicialmente, foi realizado o desenvolvimento do ambiente virtual através da plataforma Unity 3D, utilizando-se cubos 3D em todo cenário visível com o objetivo de capturar o clique do mouse, o que permitiu a inserção e seleção de peças no ambiente virtual em tempo de execução. As peças foram modeladas previamente na ferramenta Blender (BLENDER, 2020) e após isso, importadas para a Unity 3D, as quais são inseridas no ambiente virtual através de um menu de seleção de peças.

Figura 1 - Interface do ambiente virtual 3D com algumas peças inseridas e botões de seleção abaixo



Após a etapa de implementação do ambiente, deu-se início a construção do dataset para realizar o treinamento do modelo de Inteligência Artificial. Para isso, foram gravados vídeos da tela do computador enquanto o ambiente virtual estava em execução. Isso possibilitou uma grande variedade de combinações para obtenção de imagens das peças, já que dentro do cenário também foi aplicada a movimentação de câmera com Cinemachine. Em seguida foram extraídos, de forma computacional, todos os quadros que compunham o vídeo, obtendo assim uma ampla quantidade de imagens para a construção do dataset.

Através do procedimento descrito foram obtidas 6.452 imagens das peças para a criação do dataset, das quais 4.663 foram destinadas para o treinamento e 1.789 utilizadas para testar o modelo. As imagens foram divididas em 7 classes, sendo elas: AND, Connection, Input, Not, OR, Output e XOR. Para elaboração deste conjunto, foram utilizadas tanto imagens individuais das peças quanto imagens com inúmeras peças inseridas no ambiente ao mesmo tempo, evitando, assim, gerar um modelo super ajustado a classificação de uma classe por imagem ao invés de realizar detecção e reconhecimento de várias classes por imagem.



Ainda na elaboração do dataset, foram utilizadas imagens de resoluções variadas, 600x600, 800x800 e 1440x810 pixels, pois quanto maior a resolução maior a dificuldade em detectar um objeto, já que ele fica menor quanto maior for a resolução. Esta diferença incentiva a percepção de profundidade durante o treinamento já que imagens de peças menores passam a impressão de estarem mais distantes que peças maiores.

Para o treinamento da rede foi utilizada a biblioteca OpenCV e o framework YOLOv3 na plataforma Google Collaboratory, que disponibiliza, de forma gratuita por tempo limitado, uma máquina virtual de alto desempenho, ideal para realizar treinamentos de modelos de Inteligência Artificial, os quais em uma máquina comum levaria muito tempo.

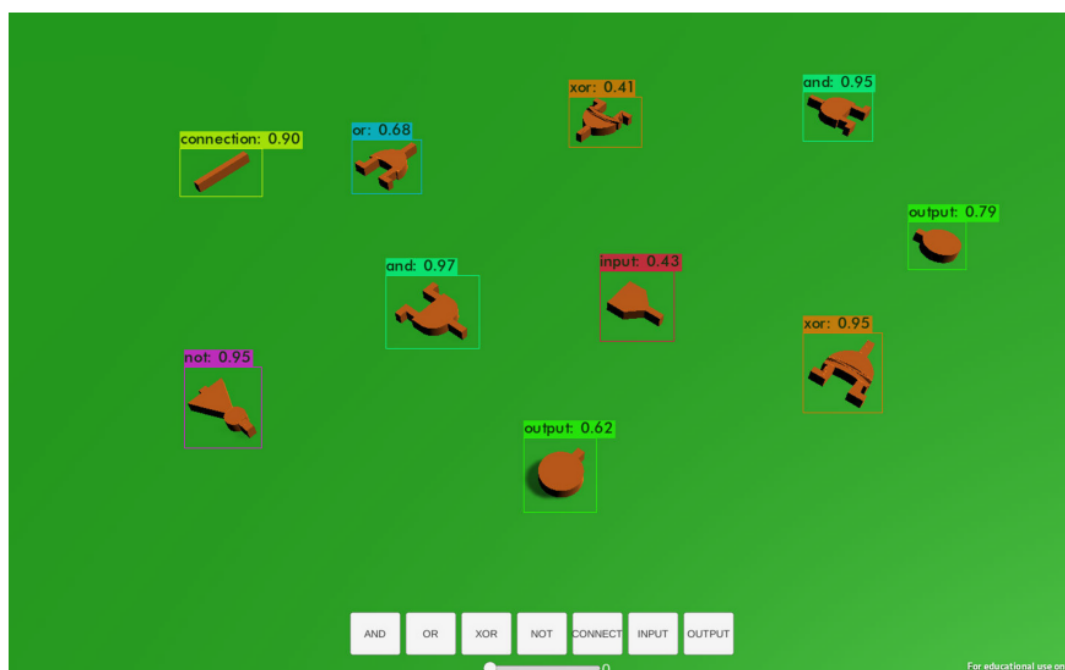
### 3. RESULTADOS E DISCUSSÃO

Inicialmente tentou-se realizar o treinamento do modelo de rede neural com a YOLOv3, sem utilizar os pesos apresentados para o dataset COCO, o qual é utilizado por muitos modelos de redes como comparativo, porém o desempenho não foi satisfatório, levando demasiado tempo para a rede conseguir convergir e extrair as características necessárias na realização de boas inferências sobre o dataset construído durante a pesquisa.

Em uma segunda tentativa foram utilizados os pesos disponibilizados, aplicando a técnica conhecida como *transfer learning* que utiliza o conhecimento adquirido para solucionar determinada tarefa na busca de resolver um problema diferente, mas que contenha semelhança entre eles. Dessa forma, o treinamento atingiu bons resultados, durando em torno de 4 horas para concluir, chegando a 3.000 épocas, obtendo uma perda de 0,18 dentro do conjunto de treinamento e teste.

A Figura 2 apresenta a capacidade da rede neural em reconhecer as peças após o treinamento, onde é possível verificar o nome das peças e a taxa de certeza que a rede neural utilizou para identificar cada peça.

Figura 2 - Aplicação do modelo treinado em uma imagem.



Por se tratar de um trabalho ainda em andamento os resultados são parciais e não chegaram a ser testados em um ambiente real, apenas no virtual. Contudo, os resultados foram promissores tendo uma alta taxa de acerto na detecção e classificação das peças, isso levando em conta que as imagens de validação não fazem parte do conjunto de treinamento e de teste, ou seja, imagens totalmente novas para o modelo treinado com a YOLOv3.

#### 4. CONCLUSÕES

A premissa para esta pesquisa foi verificar a possibilidade de treinar um modelo de inteligência artificial através da construção de um dataset com imagens de um ambiente virtual. Conforme apresentado na Figura 2 há um indicativo de que sim, é possível utilizar um ambiente virtual para criar um dataset e treinar a rede neural para fazer o reconhecimento, já que os resultados obtidos foram promissores e demonstraram que a rede foi capaz de aprender de forma bastante satisfatória a detectar e classificar os diferentes tipos de portas lógicas, mesmo em cenários com um número elevado de peças.

Acreditamos que o modelo de Inteligência Artificial treinado nesse ambiente virtual seja capaz de generalizar seu aprendizado para realizar o reconhecimento de peças em um cenário no mundo real tão bem quanto foi capaz de realizar no virtual, uma vez que foi possível obtermos um elevado grau de abstração que permitirá o reconhecimento das peças físicas no ambiente real com a mesma rede neural utilizada neste trabalho. Para maximizar as chances da generalização ocorrer entre cenários, algumas estratégias podem potencialmente ajudar, como enriquecer o dataset com um maior número de imagens, tanto virtuais quanto reais, em diferentes resoluções e treinar o modelo por um período maior de épocas.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

- BLENDER. **blender.org - Home of the Blender project - Free and Open 3D Creation Software**, 2020. Acessado em 19 de set. 2020. Online. Disponível em: <https://www.blender.org/>
- CORREA, Daniel Malgarin. **Construção de um Ambiente Virtual para Simular a Condução de uma Cadeira de Rodas**. 2016.
- GARCIA, Vera. **Paranaense será primeira deficiente visual do Brasil a defender tese de mestrado em Engenharia Elétrica**. DeficienteCiente, 26 de jan. de 2016. Online. Acessado em 18 de set. de 2020. Disponível em: <https://www.deficienteciente.com.br/paranaense-sera-primeira-deficiente-visual-do-brasil-a-defender-tese-de-mestrado-em-engenharia-eletrica.html>
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. **Imagenet classification with deep convolutional neural networks**. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 2012. Anais. . . [S.l.: s.n.], 2012. p.1097–1105.
- REDMON, J.; FARHADI, A. **YOLOv3: An Incremental Improvement**. arXiv, [S.l.], 2018.
- TERASHIMA, N. **Intelligent Communication Systems: Toward Constructing Human Friendly Communication Environment**. [S.l.]: Elsevier, 2001.
- UNITY TECHNOLOGIES. **Unity: Plataforma de desenvolvimento em tempo real do Unity**, 2020. Acessado em 19 de set. 2020. Online. Disponível em: <https://unity.com/pt>