# RECONHECIMENTO DE IMAGENS COM A FINALIDADE DE GERAR CÓDIGOS DE LINGUAGEM DE PROGRAMAÇÃO A PARTIR DE FLUXOGRAMAS FÍSICOS ARQUITETADOS POR ALUNOS COM DEFICIÊNCIA VISUAL

JOÃO PEDRO C. DO NASCIMENTO[1], LUIZ HENRIQUE KIEHN[2]

[1] Estudante do Curso Técnico em Informática Integrado ao Ensino Médio, Bolsista PIBIFSP, IFSP, Câmpus Cubatão, jpcivita2011@hotmail.com.
[2] Mestrado em Ciência da Computação, Professor do IFSP, Câmpus Cubatão, luiz.kiehn@ifsp.edu.br.

Área de conhecimento (Tabela CNPq): 1.03.03.05-7 Processamento gráfico

**RESUMO:** As questões que envolvem o aprendizado de pessoas com deficiência visual, sobretudo no que concerne a conferir-lhes acessibilidade e inclusão, encontram grandes obstáculos que dificultam às práticas docentes nas instituições de ensino. Para superar essas barreiras, é imprescindível incitar o uso de recursos que, por um lado, tornem mais eficiente a aquisição de conhecimento e, por outro, concedam cada vez mais autonomia ao aluno portador dessa condição. Tendo esse objetivo em mente, a finalidade deste projeto é elaborar uma aplicação que permita ao aluno com deficiência visual compreender a lógica computacional através da construção de fluxogramas utilizando peças tridimensionais feitas com material de papelaria e da geração do código fonte correspondente em determinada linguagem de programação com base nas imagens capturadas do fluxograma arquitetado. A presente proposta pretende propiciar o incremento a metodologias didáticas de ensino básico de lógica de programação de computadores, permitindo a inclusão de alunos com a ausência total ou parcial da visão, garantindo-lhes maior grau de independência.

**PALAVRAS-CHAVE**: algoritmo; cego; ensino; fluxograma; inclusão; programação

# COMPUTER PROGRAMMING LANGUAGE CODE GENERATION FROM IMAGE RECOGNITION OF PHYSICAL FLOWCHARTS DESIGNED BY STUDENTS WITH VISUAL IMPAIRMENT

**ABSTRACT:** Issues involving visual impairments people learning, especially regarding on giving them accessibility and inclusion, find major obstacles in the educational institutions teaching practices. To overcome these barriers, it is essential to encourage the use of resources that, on the one hand, make the acquisition of knowledge more efficient and, on the other hand, grant more and more autonomy to the student with this condition. With this goal in mind, the purpose of this project is to develop an application that allows the visually impaired student to build flowcharts of computational algorithms from three-dimensional pieces made with stationery store materials, moving on to the generation of a given programming language source code based on the designed flowchart captured images. This work targets providing improvements to the basic computational logic teaching methodologies, allowing the inclusion of students with partial or total visual impairment, guaranteeing them greater autonomy.

**KEYWORDS**: algorithm; blind; teaching; flowchart; inclusion; programming

## INTRODUÇÃO

Considerando-se que o avanço tecnológico proporcionado pela constante globalização e a ênfase no uso de redes de computadores tornam o acesso às informações algo cada vez mais primordial, é fundamental incitar a busca por estratégias pedagógicas que contribuam para a aquisição de conhecimento a todos os alunos, especialmente a aqueles com deficiência visual. As características

sensoriais que alunos cegos ou com baixa visão apresentam acentuam, sobretudo, as barreiras e dificuldades que o ensino tradicional brasileiro demonstra atualmente em sua constituição. Isso pode ser evidenciado como uma consequência da existente metodologia de preparação tendencialmente visual a que os alunos se encontram a mercê nas instituições de ensino (ALMEIDA, 2003). Além da assimilação de conceitos teóricos, o interesse por assuntos práticos, como a programação de computadores, sempre foi um desafio a ser enfrentado e, por mais que existam instituições que adotem esse conteúdo como uma disciplina escolar, o aprendizado tende a ser pouco efetivo para o aluno com deficiência visual (GOGONI, 2019).

É importante proporcionar ferramentas e técnicas que permitam às pessoas com deficiência visual inserirem-se no mundo da programação de computadores e oferecer sua perspectiva de mundo. Sendo assim, o presente trabalho propõe-se a oferecer um protótipo de uma ferramenta tátil de programação juntamente com recursos de reconhecimento de imagens e caracteres utilizando deep learning, que executa a conversão das imagens e caracteres reconhecidos para código fonte em uma linguagem de programação, que poderá ser compilado em alguma IDE, gerando-se o respectivo código executável.

Concomitantemente, a criação dessas ferramentas municiará educadores da área da Informática com recursos didáticos específicos ao ensino tecnológico de alunos com deficiência visual.

## MATERIAL E MÉTODOS

Para elaborar a ferramenta que permita a deficientes visuais entender e elaborar programas de computador, diversas frentes estão sendo abordadas e integradas neste trabalho, quais sejam: a) a elaboração das peças físicas de fluxograma, almejando que sua estrutura seja apta a pessoas cegas, compreendendo o uso de materiais facilmente encontrados em papelarias, o que reafirma a proposta de ser uma ferramenta de produção acessível; b) o registro da imagem do fluxograma para ser utilizada na etapa de seu reconhecimento; c) a utilização de um software de reconhecimento de imagens, que se utiliza de Redes Neurais Convolucionais (CNN), e que gera um arquivo contendo as identificações e descrições das imagens e textos reconhecidos; d) a utilização de um software de conversão das identificações e textos, obtidos na etapa anterior, em código fonte em uma linguagem de programação pré-determinada (C# e/ou Portugol); e) o uso de uma IDE para importar, compilar e executar o código fonte gerado na etapa anterior.

Para a confecção das peças que teriam contato direto com o aluno, foram usados materiais como isopor, papel EVA, papel contact e imãs. Na superfície de cada peça é inserido um cartão impresso em Braille que contém seus caracteres em relevo, necessário para a leitura e compreensão pelo aluno cego. O suporte de cada cartão foi feito de pedaços pequenos de cartolina enrolados e fixados com cola específica para este material. As peças foram produzidas de acordo com a necessidade dos algoritmos que seriam ministrados na experimentação com o público alvo da ferramenta.

Sendo assim, as peças seguem os formatos de terminal, decisão lógica, processamento, entrada e saída de dados. Para o formato de decisão lógica foram atribuídos dois modelos, que se diferem para consagrar suas duas possibilidades de uso, como mostra a FIGURA 1.


FIGURA 1. Estruturas lógicas presentes no fluxograma físico.

Os fluxos são indicados com setas feitas do material emborrachado EVA. Cada componente possui um imã atrás, para que assim o aluno possa arquitetar o algoritmo posicionando facilmente as peças em uma superfície metálica, a qual consistiu, na experimentação, de um painel metálico tatilmente seccionado e forrado com um fundo preto.

Para início da programação da aplicação que realiza a leitura do fluxograma, utilizou-se o Anaconda, uma ferramenta gerenciadora que possibilita a instalação de linguagens de programação na máquina, entre outras funcionalidades. Dentro desta pesquisa, essa ferramenta auxiliou na instalação da linguagem de programação Python – escolhida devido à sua menor complexidade em sintaxe, e pelo maior número de materiais disponíveis abrangendo seu aprendizado. A IDE escolhida para o desenvolvimento foi a PyCharm, pois os seus recursos de complemento de código inteligente e de inspeções tornaram essa ferramenta a opção mais produtiva para o desenvolvimento deste projeto.

Visando consagrar o reconhecimento das imagens obtidas, assim como a classificação dos seus componentes, houve a necessidade da criação de uma Rede Neural Convolucional. Consequentemente, houve a necessidade também do encontro de um dataset adequado – um conjunto que compreendesse diversas imagens dos formatos do fluxograma e das setas para que o uso dessa ferramenta realizasse o esperado. Quanto ao desenvolvimento da rotina de treinamento, usou-se uma arquitetura menos complexa, inspirada no modelo da Visual Geometric Group, o VGGNet (HASSAN, 2018). A ferramenta que possibilitou o treinamento da CNN foi o Tensorflow que, criado pela Google, consiste em uma biblioteca open-source usada para a geração de modelos de machine learning, tornando sua aplicabilidade mais fácil graças a sua interface escrita em Python: o Keras, uma API front-end.

Para que pudesse ser feita a leitura dos fluxogramas arquitetados e consequentemente a geração dos códigos, programou-se um script que possui métodos que atuam na seguinte ordem de execução: a) leitura da imagem enviada seguindo a dimensão de três colunas por oito linhas, como mostra a FIGURA 2; b) o recorte individual de seus quadrantes; c) a classificação dos componentes de cada quadrante a partir das redes neurais treinadas; d) a extração do texto escrito em cada estrutura lógica; e) o mapeamento do fluxo do fluxograma; f) a geração do código fonte em um arquivo de texto.

FIGURA 2. Dimensões computadas de cada um dos quadrantes.

Para os primeiros métodos de manipulação de imagem desse script, a biblioteca Open Source Computer Vision Library, ou comumente conhecida como OpenCV, tornou-se indispensável, enquanto que para os métodos de geração das linhas de código, levantaram-se duas APIs para o reconhecimento dos textos que se localizam no interior das estruturas lógicas: o Tesseract e a OCR Space. Ambas são ferramentas de OCR – Optical Character Recognition. A primeira é uma opção código-aberto desenvolvida pela HP entre os anos de 1984 e 1994, enquanto a segunda foi elaborada pela empresa AT9T Software GmbH.

## RESULTADOS E DISCUSSÃO

Foram treinados dois modelos diferentes de Rede Neural Convolucional: um para atender a classificação das estruturas lógicas e outro para classificar a orientação das setas. Nos primeiros treinamentos para realizar a classificação das estruturas lógicas do fluxograma, mesmo com valores altos de acurácia, a rede demonstrou uma forte presença de sobre-ajuste nos dados ao não conseguir generalizar imagens das mesmas peças com ângulos diferentes, por exemplo. Com a inviabilidade de retirar mais fotografias para melhorar o dataset, foi necessário mexer na estrutura das peças. Primeiramente, essas foram feitas com a mesma cor de fundo em todas. Assim, após o resultado dos primeiros treinamentos, houve a necessidade de diversificar as cores destas seguindo a premissa de que

assim seria a melhor maneira de diferenciá-las; logo atribuiu-se uma cor diferente para cada estrutura, chegando nos resultados da FIGURA 1. No fim, essa rede pontuou 90% de acurácia, e 35% de perda.

Para a identificação das setas, ao optar por utilizar cores e não as suas orientações para diferenciá-las, a rede apresentou ótimos resultados. A fim de evitar erros nas classificações, escolheu-se fazer o uso das cores, assim como realizado na diferenciação dos formatos do fluxograma, visto que as setas podem ser facilmente confundidas por terem a mesma aparência física. Essa rede neural apresentou as seguintes pontuações: perda de somente 29%, e uma acurácia de 93%.

Optou-se por trabalhar, inicialmente, com a sintaxe da linguagem de programação Portugol, para a realização das gerações dos códigos, devido a essa linguagem já ser trabalhada em uma das disciplinas do ensino de programação computacional no campus em que esse projeto foi desenvolvido. Idealmente, o aluno cego arquitetaria o fluxograma com o auxílio de cartões impressos em Braille, onde nestes conteriam os direcionamentos do algoritmo a ser trabalhado. No entanto, fez-se necessário deixar o reconhecimento dos caracteres em Braille para uma próxima oportunidade, visto que o tempo de vigência não foi o bastante para englobar a dificuldade e complexidade de construir uma ferramenta assim. Dessa maneira, para conseguir efetivar a ideia da proposta desse projeto, utilizaram-se cartões com caracteres latinos para serem analisados pela tecnologia OCR, como mostra a FIGURA 3.

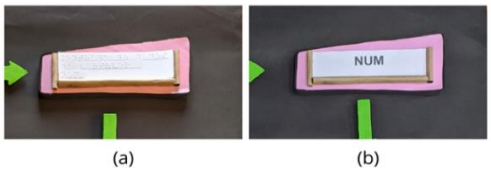


FIGURA 3. Adaptação dos cartões em Braille (a) para caracteres latinos (b)

Para a extração desses textos, foram testadas duas opções de API de reconhecimento OCR: o Tesseract e o OCR Space. Inicializando os testes, a principal limitação que o Tesseract apresentou foi a confusão entre o que deveria ou não ser lido como um caractere. Sendo assim, mesmo com as etapas de pré-processamento de imagem, não se obteve êxito em eliminar totalmente os demais elementos que não fossem texto em todos os testes, comprometendo o resultado final das linhas de código geradas. Ao observar os empecilhos demonstrados, a API OCR Space conseguiu alcançar melhor desempenho. A principal diferença é a facilidade que o OCR Space disponibiliza ao usuário ao não demandar etapas de pré-processamento antes do texto ser enviado para análise. Os principais problemas que essa API apresentou foi no reconhecimento de alguns caracteres específicos, principalmente aos que correspondem a símbolos matemáticos (Nota: este problema está sendo contornado pela substituição desses caracteres por palavras correspondentes, as quais serão convertidas para os respectivos símbolos em etapa posterior), e na estabilidade de sua conexão com os servidores na versão gratuita.

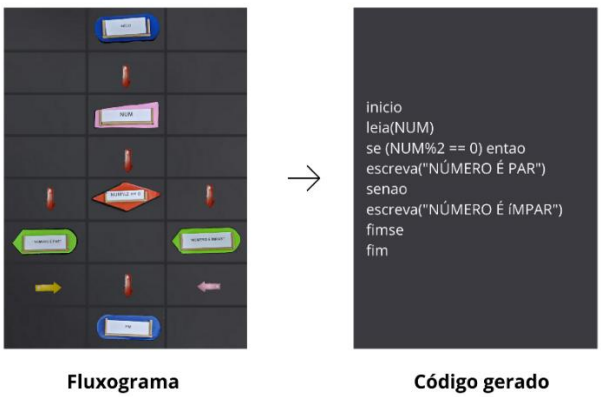


FIGURA 4. Geração do código feita após uma simulação com algoritmo básico.

Infelizmente, devido à pandemia do Novo Coronavírus, a validação desse projeto com o público alvo – alunos cegos do próprio campus – não conseguiu ser atingida. Essa etapa se faz importante, pois, embora já se tenha efetuado uma validação dos objetos anteriormente com um aluno deficiente visual,

toda alteração feita no material tátil para conseguir melhorar o reconhecimento das imagens deveria ter sua efetividade avaliada por esse público. Cabe observar aqui que, visando agilizar os trabalhos, e considerando a impossibilidade momentânea de realizar os testes com deficientes visuais, a partir de um certo ponto, as imagens de testes – os fluxogramas arquitetados – foram confeccionadas em editores de imagem, tendo como base as imagens inicialmente obtidas dos objetos físicos sobrepostos ao painel, o que não invalidou a essência do projeto. Dessa forma, atingiram-se os resultados esperados, dado o estágio de prototipagem da aplicação.

## CONCLUSÕES

Em vista do obtido nos resultados, conseguiu-se atingir a essência da proposta do projeto, ficando pendentes algumas questões que se configuram como melhorias a serem implementadas em trabalhos futuros. Entre estas melhorias, destacam-se: a) incremento de um tradutor de caracteres em Braille, a fim de que não seja mais necessário usar caracteres latinos; b) incremento do reconhecimento de mais estruturas lógicas, como as de laços de repetição; c) adicionar a opção de gerar o código em mais de uma sintaxe de linguagem de programação, podendo atender mais de uma disciplina do curso do aluno; d) aumentar o espaço e a quantidade de quadrantes, permitindo que o aluno possa desenvolver algoritmos mais complexos; e) incrementos em IDEs a serem utilizadas pelos deficientes visuais de forma a proporcionar-lhes maior autonomia.

## AGRADECIMENTOS

Agradeço a todo o trabalho e carinho que meu orientador teve comigo nesse período de vigência do projeto. Reconheço ainda a importância do incentivo de meus familiares e amigos próximos, os quais nunca deixaram de acreditar em mim.

## REFERÊNCIAS

ALMEIDA, Edson Silva. Um modelo de ensino de lógica digital para deficientes visuais. Orientador: Marcos José Negreiros Gomes. 2003. Dissertação (Mestrado Integrado Profissionalizante em Computação) - Computação do Centro de Ciências e Tecnologia, Universidade Estadual do Ceará, Fortaleza, Ceará, 2003. p. 133. Disponível em: http://www.uece.br/mpcomp/index.php/arquivos/doc_download/208dissertacao-49. Acesso em: 2 mar. 2020.

GOGONI, Ronaldo. Project Torino, o sistema da Microsoft para ensinar programação a deficientes visuais. [S. l.]: MEIO BIT, 2019. Disponível em: https://meiobit.com/396807/microsoft-project-torino-programacao-deficientes-visuais/. Acesso em: 6 mar. 2020.

HASSAN, Muneeb. VGG16 – Convolutional Network for Classification and Detection. [S. l.]: Neurohive, 29 nov. 2018. Disponível em: https://neurohive.io/en/popular-networks/vgg16/. Acesso em: 9 abr. 2020.