11º FEPEG FÓRUM ENSINO • PESQUISA EXTENSÃO • GESTÃO
UNIVERSIDADE, SOCIEDADE E POLÍTICAS PÚBLICAS
ISSN: 1806-549X
Realização: Unimontes Universidade Estadual de Montes Claros
Apoio: FAPEMIG FADENOR

CÁLCULO DE ÁREAS DE FIGURAS GEOMÉTRICAS UTILIZANDO OPENCV

Autores: EDUARDO DIAS DA ROCHA, JÉSSICA FLAVIANE FERREIRA, LUIS PAULO TOLENTINO FERNANDES, ADRIANO LAGES DOS SANTOS

Introdução

A Inteligência Artificial (IA) é um dos campos mais recentes da ciência e engenharia que tem como objetivo executar funções humanas consideradas inteligentes (RUSSEL; NORVIG, 2013). Os sistemas de IA têm características como aprendizagem, capacidade de raciocínio, reconhecimento de padrões e inferência. Um importante campo de estudo dessa área são os sistemas especialistas (SE). Esses sistemas procuram simular o raciocínio de um especialista (*expert*) em uma área do conhecimento humano (VASCONCELOS; MARTINS JUNIOR, 2004). Por exemplo, pode-se desenvolver um sistema especialista para ajudar médicos, enfermeiros ou outros profissionais da saúde no diagnóstico de certas doenças. Com base nas respostas do paciente a perguntas formuladas previamente e na base de conhecimento desenvolvida na construção do SE, um diagnóstico e uma margem de certeza seriam fornecidos pelo sistema.

O SE desenvolvido neste trabalho utiliza visão computacional para o cálculo da área de figuras geométricas a partir de uma imagem dessa figura com os valores dos lados (ou raio para círculos). A característica principal da visão computacional é simular a visão humana. Pode-se entender a "entrada" desse sistema como a imagem a ser "enxergada" pela IA presente no sistema, e a "saída" seria a interpretação da imagem – ou parte dela (MARENGONI; STRINGHINI, 2009).

Dadas as possibilidades da visão computacional para interpretação de características de imagens, a motivação deste trabalho é desenvolver o protótipo de um SE que possa auxiliar alunos no aprendizado de geometria por meio do cálculo das áreas de figuras geométricas em imagens, mais especificamente triângulos, quadrados, retângulos e círculos. Além disso, o sistema pode auxiliar Arquitetos e Engenheiros no cálculo de áreas a partir dos desenhos de seus projetos.

Material e Métodos

A. Python e OpenCV.

Para implementação do sistema, escolheu-se a linguagem de programação Python por ser uma linguagem poderosa e de fácil aprendizado, além de possuir estruturas de dados de alto nível eficientes. Também adota uma abordagem simples e efetiva para programação orientada a objetos (PYTHON 2.7).

Já o OpenCV é uma biblioteca de programação multiplataforma, de código aberto, que implementa diversas ferramentas de interpretação de imagens. Podem ser executadas operações de filtros de ruídos, detecção de movimentos, reconhecimentos de padrões e reconstrução em 3D. Outra vantagem é que o OpenCV está disponível gratuitamente na internet com vasta documentação (MARENGONI; STRINGHINI, 2009).

B. Identificação de figuras.

O primeiro passo para o cálculo da área é identificar qual é a figura. Neste trabalho propomos a identificação de quadrados, retângulos, triângulos e círculos. Escolhemos inicialmente essas figuras por serem mais simples e serem a base que compõe outras figuras, como trapézios e pentágonos, por exemplo.

Foi realizado inicialmente um pré-processamento da imagem que contém a figura geométrica para eliminar ruídos, como a textura do papel em que a figura foi desenhada. Segundo Pires e Barcelos (2004), muitas vezes as imagens a serem utilizadas em alguma aplicação possuem distorções ou ruídos que precisam ser corrigidos antes de processar a imagem. Neste caso, transformamos a imagem em uma imagem binária por meio do método *threshold* do OpenCV. A Fig. 1A mostra a imagem utilizada para exemplificar o processo de cálculo da área e a Fig. 1B mostra a binarização dessa figura.

Após o pré-processamento, são identificados os componentes conectados presentes na imagem, também buscando reduzir ruídos. Por meio do pacote *skimage*, encontramos os componentes conectados presentes na imagem. Caso haja, por exemplo, algum segmento de reta desenhado aleatoriamente, ele não será considerado como uma figura por não ser conectado a nenhum outro componente. Além disso, componentes muito pequenos geralmente não representam as figuras a serem detectadas, sendo muitas vezes são os números ou ruídos, como verificamos em diversos testes. Assim, reduzimos a busca por figuras na imagem para componentes conectados que possuem tamanho superior a 200. Em seguida, identificamos o segundo maior contorno da imagem por meio do método *findCountours*, pois conforme verificados nos testes o maior contorno identificado é da imagem como um todo. Com os valores que representam o contorno da figura, utilizamos o método *approxPolyDP* do OpenCV para obter a quantidade de segmentos que forma a figura. Assim, para três segmentos temos um triângulo, para quatro segmentos temos um quadrado ou retângulo, que são diferenciados pelas diferentes entre largura e altura, e para um caso geral temos círculos. A Fig. 1C mostra a identificação da figura de exemplo.

C. Identificação dos números.

A segunda etapa deste trabalho consiste em identificar os números referentes aos lados das figuras, para os polígonos, e referente ao raio, para círculos. Primeiramente, foi treinada manualmente uma base de figuras com o *K-Nearest Neighbors* (KNN) para rotulação dessas figuras. Construímos uma imagem para o treinamento com 80 números de zero a nove em dez fontes textuais, inclusive algumas que se aproximam da escrita humana.

ISSN: 1806-549X

Em seguida identificamos os componentes da imagem que representam os números, normalmente com tamanho entre 10 e 28 pixels nas imagens de teste. Caso essa faixa de pixels não seja suficiente para a detecção dos números, são buscados os componentes entre 10 e 100 pixels. Os valores dos números desenhados na imagem são comparados com as imagens rotuladas por meio do KNN para identificar qual são os valores dos lados ou raio das figuras geométricas. A Fig. 1D mostra a identificação dos números na figura geométrica de exemplo.

D. Cálculo das áreas.

Com a identificação da figura e dos números que representam os comprimentos de seus lados ou raio, podemos calcular a área da figura.

Para quadrados, como possuem todos os lados iguais, foi feita apenas uma multiplicação do primeiro número identificado, já que todos os números são iguais. Para retângulos, o vetor de números é ordenado de forma decrescente. Caso tenham sido encontrados apenas dois valores - caso em que foram desenhados apenas os valores dos dois lados necessários para o cálculo - é feita a multiplicação dos dois valores identificados. Caso haja mais números, são multiplicados o primeiro e o terceiro valor, já que foram ordenados de forma decrescente. Já para o caso dos triângulos utilizamos o Teorema de Heron para o cálculo da área do triângulo conhecendo seus lados. Por fim, para o círculo a área é obtida pela multiplicação do número ? pelo raio elevado ao quadrado.

E. Base de testes.

Para realização de testes e verificação do desempenho do algoritmo no cálculo das áreas foi construída uma base de teste. Essa base contém recortes de fotos de figuras desenhadas à mão e em editores de imagens. Ao todo foram utilizadas 50 imagens, todas no formato *.png*: 20 quadrados, 15 retângulos, 10 triângulos e cinco círculos. A Fig. 1E mostra exemplos das imagens da base de teste. Cada tipo de figura já foi previamente catalogado em pastas separadas (uma pasta para triângulos, outra para quadrados, outra para retângulos e outra para círculos). Além disso, calculamos previamente a área de cada figura e armazenamos em um arquivo *.csv* referente à cada tipo de figura, no mesmo diretório de arquivos das figuras correspondentes.

Para testar o desempenho do algoritmo, foram realizados dois testes: um teste para avaliar a identificação das figuras e outro teste para avaliar o cálculo das áreas. Para todas as figuras, armazenamos o resultado da identificação e conferimos com a base dados, onde as figuras já estão separadas. E, também para todas as figuras, armazenamos o resultado do cálculo da área pelo algoritmo e conferimos com o arquivo *.csv* correspondente que contém os valores corretos das áreas. Ao final, calculamos uma porcentagem de quantas figuras e quantas áreas foram acertadas pelo algoritmo.

Alguns pontos importantes devem ser ressaltados. O primeiro deles é que o cálculo da área é dependente de que a figura e todos os números daquela imagem tenham sido identificados de forma correta. Como a base de números treinada foi pequena, alguns números desenhados não foram identificados corretamente, especialmente nas fotos de figuras desenhadas à mão. Além disso, há muitos ruídos nas imagens desenhadas à mão que interferem na identificação tanto dos dígitos quanto das figuras.

Resultados e discussão

Os resultados dos testes são apresentados na Tabela 1. Podemos perceber que para os triângulos e círculos o acerto na identificação das formas foi de 100%. Isso se deve ao fato de que todas as figuras desenhadas à mão - as imagens com muitos ruídos - são quadrados ou retângulos. Também se pode perceber que os valores de acerto das áreas são maiores para os tipos com mais acerto de figuras, pois a identificação incorreta de uma figura pode acarretar no cálculo incorreto de sua área. Por fim, o acerto total de figuras foi de 86% e o acerto total de áreas foi de 62%.

Os resultados mostram que é preciso melhorar a base de dígitos, para melhorar a identificação dos números, além de aumentar a base de teste para ter resultados mais precisos.

Considerações finais

Concluímos que o trabalho atingiu as expectativas por conseguir um resultado razoável dentro do que foi proposto. Ainda assim, é preciso melhorar a identificação dos dígitos, causas da maior quantidade de erros de cálculo das áreas. Também é preciso aumentar a base de imagens de teste, para obter resultados mais precisos.

Quanto a trabalhos futuros, pode-se expandir este trabalho para identificação de valores de lados faltantes de figuras, cálculo de áreas de outras figuras, cálculo de perímetros, entre outras aplicações.

Referências bibliográficas

MARENGONI, M; STRINGHINI, D. Tutorial: Introdução à Visão Computacional usando OpenCV. **Revista Teórica de Matemática Aplicada - RITA**, v. 16, n. 1, 2009. Disponível em: <http://www.seer.ufrgs.br/index.php/rita/article/view/rita_v16_n1_p125/7289>. Acesso em: 28 set. 2017.

PIRES, V. B.; BARCELOS, C. A. Z. Filtros de suavização para eliminação de ruídos e identificação de bordas. Disponível em: <https://projetos.extras.ufg.br/conpeex/2004/pibic/exatas/Vinicius.html>. Acesso em 13 set. 2017.

PYTHON 2.7. General Python FAQ. Disponível em: <https://docs.python.org/2.7/faq/general.html#what-is-python>. Acesso em 28 set. 2017.

VASCONCELOS, V. V.; MARTINS JUNIOR, P. P. Protótipo **de Sistema Especialista em Direito Ambiental para auxílio à decisão em situações de Desmatamento Rural**. NT-CETEC-MG. 2004. Disponível em: <https://pt.scribd.com/do /90446335 /Prototipo-de-Sistema-Especialista-em-Direito-Ambiental-para-Auxilio-a-Decisao-em-Situacoes-de-Desmatamento- RuralNT-CRHA-27-2004>. Acesso em: 28 set. 2017.

RUSSEL, S. J.; NORVIG, P. **Inteligência Artificial**. Tradução de Regina Célia Simille. Rio de Janeiro, Elsevier, 2013.

ISSN: 1806-549X

Tabela 1. Acertos de figuras e áreas.

	Quadrados	Retângulos	Triângulos	Círculos
Acerto das figuras	80%	80%	100%	100%
Acerto de áreas	65%	53%	70%	60%

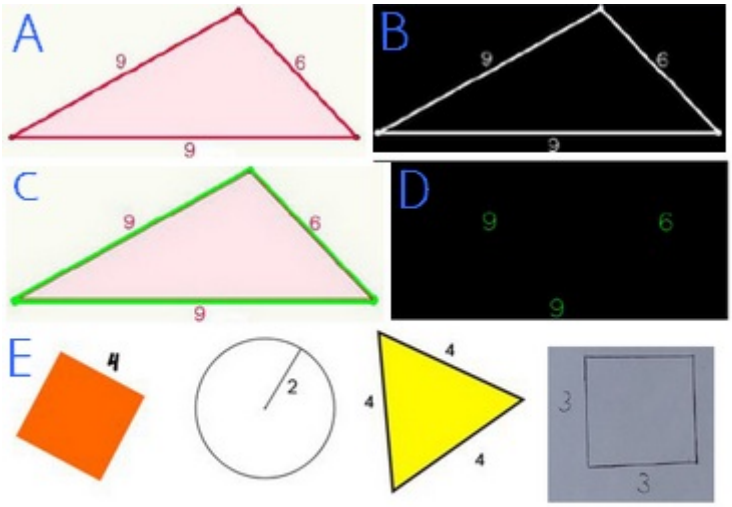

Figura 1. Fig. 1A Figura geométrica de exemplo, Fig. 1B Binarização da figura geométrica, Fig. 1C Identificação da figura geométrica, Fig. 1D Identificação dos números, Fig. 1E Exemplos de figuras geométricas da base de testes.