Contagem de moedas e notas - Tópicos em processamento de imagens T01

Caio Henrique Suzuki Polidoro[1]: RGA 2015.1905.002-0

[1]Faculdade de Computação – Universidade Federal de Mato Grosso Sul (UFMS)
Caixa Postal 549 – 79.070-900 – Campo Grande – MS – Brazil

caio.polidoro@aluno.ufms.br

1. Definição do problema

Este projeto tem como objetivo identificar a quantidade de notas e moedas em uma imagem de entrada, sendo como pré-requisitos: o fundo da imagem deve ser em branco, pode haver variação de iluminação, podem haver objetos na imagem que não sejam moedas ou notas.

2. Metodologia

2.1. Descrição da imagem de entrada (restrições)

A imagem deve ter tamanho (em torno de) 3024 x 4032 pixels, é independente de orientação e lado das cédulas. As cédulas e moedas devem ter um tamanho fixo (filtragem final usa os tamanhos e proporções para assumir qual objeto é qual).

2.2. Como executar o programa para todas as imagens de entrada

Para executar o programa para uma pasta de imagens de entrada, é necessário executar a linha de comando python3 evaluate_results.py -i caminho/para/pasta/de/imagens/de/entrada (no caso, a pasta utilizada para os testes foi a notas-e-moedas-exemplo). As imagens devem estar nomeadas da forma 5c7n.jpg onde 5 é o número de moedas e 7 é o número de cédulas da imagem.

Desta forma, a saída será dada informando para toda imagem se ocorreu um acerto ou erro na resposta, e ao final há a quantidade total de erros e acertos.

2.3. Passo-a-passo de como funciona

Como é possível observar na Figura 1, o programa recebe uma imagem de entrada por vez, e então é feito a conversão de BGR para tons de cinza, com o objetivo de obter apenas um canal de intensidade, em seguida é feito o ajuste do tamanho da imagem, este passo foi focado mais na questão visual para que seja possível visualizar os resultados de cada operação realizados. O último passo comum tanto à detecção de moedas quanto de notas é um Gaussian Blur, feito com o intuito de borrar detalhes dos objetos que serão detectados, evitando que ”bordas”falsas atrapalhem as operações que virão pela frente.

Para a detecção de moedas, é realizada uma transformada de Hough através da função HoughCircles do opencv, os parâmetros foram ajustados para que as moedas de um determinado tamanho (raio) sejam encontradas, por esta razão é importante o tamanho da imagem de entrada.

Para a detecção das cédulas, em primeiro lugar é feita a binarização das intensidades dos pixels através do método adaptativo gaussiano de limiarização, a ideia é formar ”blobs”brancos para cada cédula. Para isso as seguintes operações são realizadas: fechamento e então dilatação, para juntar alguns pontos que ao ser binarizados, apesar de fazerem parte da mesma nota ficaram separados. Com os ”blobs”formados, usa-se a função findContours para extrair os contornos deles, em seguida, aplica-se minAreaRect para encontrar os retângulos referentes aos ”blobs”encontrados. Como passo final, os retângulos são filtrados por área e proporção, dado que as cédulas em geral tem uma proporção semelhante entre largura e altura. Com isso, é possível obter as cédulas da imagem.

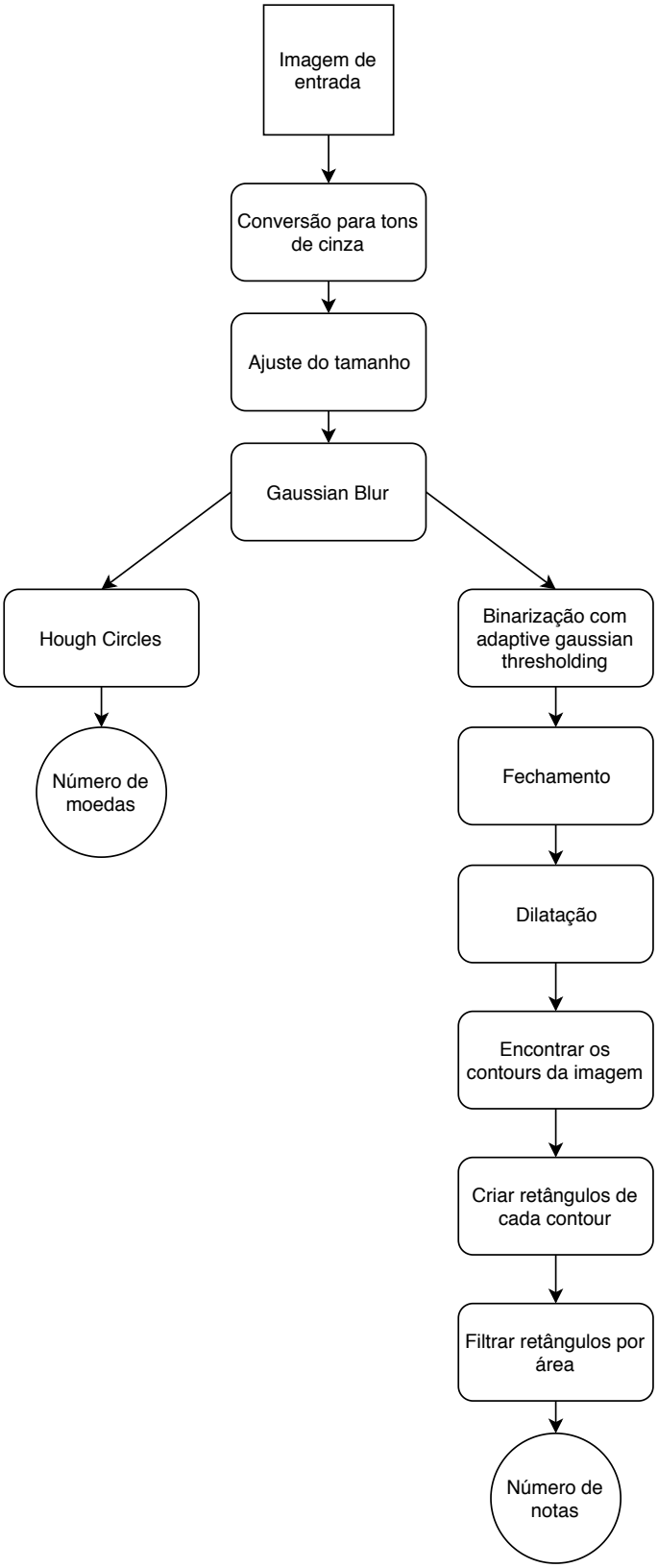

Figura 1. Flowchart da estratégia utilizada para identificar a quantidade de moedas e notas na imagem.

2.4. Detalhes complementares do código

É possível rodar o código para apenas uma imagem, através do programa count_coins_and_bills.py, rodando este programa, é possível visualizar o passo-a-passo de cada operação realizada, pelo parâmetro show_steps.

3. Conclusão

Foi possível obter 14 acertos em 16 imagens, a estratégia utilizada teve como premissa realizar um pré-processamento para que as funções decisivas (HoughCircles e minAreaRect) pudessem detectar corretamente os objetos. Uma desvantagem dessa abordagem é sua dependência na proporção dos objetos em relação ao tamanho da imagem.