Uma imagem com texto, Tipo de letra, Gráficos, logótipo

Descrição gerada automaticamente

Licenciatura Engenharia Informática e Multimédia

Processamento de Imagem e Visão

Semestre de Inverno 2023 / 2024

Trabalho Prático 1

Docente Nuno XXXXXXUXUXUXUXUXUXUXUXUXU

14 de Novembro de 2023

Trabalho realizado por:

Fábio Dias, nº 42921

Diogo , nº YXYXY

# Índice

ASAS

# Índice de Figuras

asasas

# Introdução

Para este trabalho prático, foi-nos pedido o desenvolvimento de um algoritmo que identifique moedas e obtenha a quantia total apresentada em imagens.

Para tal, foi disponibilizado um conjunto de imagens de treino, todas estas obtidas nas mesmas condições. As moedas foram colocadas em cima de uma mesa de superfície homogénea e clara, captada por uma câmara montada num tripé, ajustada de modo que o plano do sensor fosse paralelo ao plano da mesa.

Este algoritmo a desenvolver deve ser robusto para conseguir lidar com presença de objetos que não sejam moedas, existência de pequenas sombras e eventual contacto entre objetos, inclusive moedas.

A intenção é obter características dos objetos nas imagens, neste caso são as moedas, e conseguir classificá-las corretamente para coincidirem com o seu valor real. Para tal, é preciso pré-processar as imagens de forma genérica, ou seja, um conjunto de operações específicas para aplicar a todas, e depois extrair estas características.

Para este trabalho prático, foi usada a linguagem de programação Python, com o auxílio a diversas bibliotecas como o Numpy e OpenCV.

# Abordagem Inicial

No enunciado deste trabalho prático, foi-nos disponibilizada uma sequência típica de tarefas e operações relacionadas com o desenvolvimento deste algoritmo.

Inicialmente, temos de carregar as imagens e guardá-las em memória para depois serem manipuladas conforme o nosso objetivo.

Convertemos as imagens para tons de cinzento. Assim, a imagem passa de três canais, vermelho, verde e azul para apenas um. Desta forma, é requerido menos processamento e reduz a influência da iluminação, o que destaca melhor o contraste entre as moedas e a mesa.

O passo seguinte é a binarização da imagem. Isto significa que as imagens vão possuir apenas dois valores. Neste caso, ou 0 ou 255. 0 sendo preto e 255 sendo branco. Mas é necessário especificar o que vai ser branco e o que vai ser preto. Desta forma, podemos usar o algoritmo de Otsu. Este encontra automaticamente um valor para o limiar de decisão entre estes dois grupos. Ou seja, este vai encontrar o valor para um limiar ótimo para cada imagem, de forma adaptativa. Finalizando a binarização, quando visualizamos a imagem, conseguimos observar padrões circulares no fundo preto. Estes serão as moedas.

Este processo não é perfeito. As moedas podem ter circunferências perfeitas, mas buracos no círculo ou baías. Algumas moedas ficam irreconhecíveis ao ponto de não terem o aspeto de círculo nem parecerem pertencer ao mesmo objeto. Para corrigir isto e tentar obter algo semelhante a uma moeda, precisamos de efetuar operações morfológicas. Estão são um conjunto de técnicas de processamento de imagem utilizadas para modificar a forma de objetos em imagens. São utilizadas para pré-processamento e análise de imagem para melhorar características, como as bordas dos objetos.

Algumas destas operações usadas são a erosão, esta é utilizada para encolher o objeto, separar objetos assim como remover pequenos detalhes e ruído nas imagens; a dilatação, que serve para expandir objetos, preencher alguns buracos e engrossar bordas; outra operação usada foi o fechamento, que é uma operação de dilatação seguida de uma erosão. Esta também é útil para preencher buracos e suavizar contornos.

Para estas operações, é necessário ter elementos estruturantes. Estas são máscaras com diferentes tamanhos que percorrem cada pixel na imagem, substituindo o seu valor por um diferente de acordo com a operação desejada.

Após diversas tentativas e com diversos elementos estruturantes, concluiu-se que embora fosse um caminho possível, era esgotante e provou-se ineficaz na medida de investimento de tempo. Assim, partimos para outra abordagem.

# Abordagem Final

Para esta abordagem, foi necessário o carregamento das imagens assim como a conversão para os tons de cinzento, previamente efetuados para a abordagem anterior.

O próximo passo é desfocar ligeiramente a imagem. Isto porque o desfoque pode suavizar a imagem e reduzir o ruído, conseguindo assim uma melhor detecção das bordas das moedas.

Seguidamente, procuramos os círculos presentes nas imagens e anotamos a que correspondem os mesmos, assim como os seus raios. Esta será a principal característica a ter em conta nas avaliações e classificações das moedas: o seu raio.

Por fim, soma-se o valor encontrado nos objetos classificados como moedas em cada imagem.

# Implementação do Algoritmo Inicial

Como previamente abordado, o primeiro é carregar todas as imagens para o nosso programa. Para isto, usaremos um método da biblioteca OpenCV denominado “imread”.

IMAGEM DO IMREAD

Para observar as imagens, podemos usar o método “imshow” do OpenCV. Também será necessário usarmos os métodos “waitKey” e “destroyAllWindows” para uma melhor resultado. Para simplificar estas três chamadas de funções, criámos uma função denominada “showImage” que recebe o título da imagem e a imagem em si e cujo corpo são estes três métodos do OpenCV. Dado que as imagens são guardadas em arrays, foi criado outro método auxiliar que recebe um array de imagens e mostra-as uma a uma.

IMAGEM DOS MÉTODOS

Seguidamente, convertemos todas estas para tons de cinzento com o uso do método “cvtColor” também do OpenCV.

IMAGEM DO CVTCOLOR

Próximo passo é a binarização das imagens. Para isto, usamos utilizamos o método “threshold” e usamos o algoritmo de Otsu.

IMAGEM DO THRESHOLD

Partimos para as operações morfológicas. Aqui foi onde as imensas dificuldades começaram a aparecer e, consequentemente, quando a ideia de uma outra abordagem começou a surgir.

Foquemo-nos na primeira imagem. Esta possui todas as moedas que vão ser encontradas no conjunto de treino devidamente espalhadas e distribuídas, logo é possível retirar as máscaras de cada uma das moedas.

Para tal, após o carregamento das imagens e da conversão para tons de cinzento, passámos para a binarização dando uso ao algoritmo de Otsu.

[IMAGEM DO RESULTADO DA BINARIZAÇAO DA PRIMEIRA IMAGEM]

Como é possível observar, conseguimos diferenciar todas as moedas, mas algumas possuem buracos. Assim, é crucial o uso das operações morfológicas, em particular do fecho. Para isto, após tentativa-erro, chegamos à conclusão de que um disco de 9 pixeis por 9 pixeis era o elemento estruturante ótimo para o fecho de todos os buracos, assim como a suavização da máscara.

[IMAGEM DO RESULTADO DAS OPERACOES BINARIAS]

Estas operações conseguiram optimizar as máscaras das moedas para a primeira imagem. Com o uso do método “connectedComponentWithStats” conseguimos extrair a localização inicial de cada objecto, assim como a sua largura e altura, a sua área e as coordenadas do seu centroide.

[IMAGEM DO CONNECTEDCOMPONENTWITHSTATS]

[IMAGEM DA ALGUMAS MÁSCARAS (OU TODAS COM O SUBPLOT)]

Após tentarmos para outras imagens, concluímos que este conjunto de operação não apresentava tão bons resultados.

[IMAGEM APOS OPERACOES MORFOLOGICAS]

Após enumeras tentativas, foi decidido tentar uma outra implementação.

# Implementação do Algoritmo Final

O processo inicial é o mesmo da implementação anterior: carregar as imagens e passá-las para tons de cinzento. A seguir, aplicar um desfoco gaussiano com uma matriz de 15 pixeis por 15 pixeis, “kernel”, e passamos o valor 0 ao parâmetro “sigmaX”. Ocultando o valor do parâmeto “sigmaY” faz com que o valor seja igual ao passado no “sigmaX”. Se é 0, o desvio padrão será calculado automaticamente a partir do kernel passado.

[IMAGEM DA PRIMEIRA IMAGEM COM BLUR]

De seguida, usamos o método “HoughCircles” do OpenCv que

# Conclusões

Qsqsqsqsq2e eifq h83 y38yr r3 3 ufefhewfhuewf ewfhewfeiwf iewfiewfiewfiewhfwefiwefiwefiwe ifweif weif ewifweif ewf ie fiew fewewiew iew eeewiewiefw

# Bibliografia

Fkefjefjqfqe oeq ehewowhwhwewhwg h guiwiweu eu ef uef efewu fewuifwuifwfuifhwewuifewhewhuewhuewufewhwufwwifwfwugw gwgu wgwi gw g.