

Relatório da Tarefa 1

Ana Rúbia

03/09/2019

1 Parte 1 e 2

A parte 1 e a parte 2 da atividade foram interessantes para se familiarizar com o ambiente, com o código, as bibliotecas; então, foi bem tranquilo seguir o passo a passo dado na explicação da tarefa. O desafio estava em, basicamente, escolher a melhor estrutura de dados para armazenar as informações das imagens.

Eu escolhi armazenar tudo em um grande dicionário de dicionários. O dicionário mais externo me indicava se a base que eu estava usando era a modelo ou as bases de testes, então o dicionário mais externo tem 3 chaves: MODELO, Base1, Base2. Eu não mudei o nome das pastas, e nem troquei de lugar as imagens, apenas com essa divisão já era possível acessar os itens que eu queria. Já no dicionário mais interno, cada chave era o nome de uma imagem e o valor era o nome da imagem em si, por exemplo: {"Hulk": "Hulk.png"}.

Ao desenvolver o código e obter os resultados da comparação de histogramas, usando a função de interseção fornecida na descrição da atividade, eu achei muito estranho ele está errando tanto, até o Hulk ele dizia que era o homem de ferro rs. Analisando o meu código, o loop que criei me respondia a seguinte pergunta: dada uma imagem modelo, qual desses cinco (da base de teste) são mais parecidos? e a maioria das vezes caía no homem de ferro. Depois, eu mudei o código para verificar entre todas as ocorrências daquela imagem teste, qual modelo era mais parecido. Dessa maneira, o programa errou menos.

Para finalizar, era necessário fazer um gráfico para visualizar melhor as similaridades. Todos os dados computados ficaram armazenados nos dicionários, então eu criei uma função "mostra_gráfico" que mostra o par QUEMx,Probabilidade para cada imagem na base de teste. Um exemplo pode ser visto abaixo na imagem em anexo [grafico_hulk.png].

2 Parte 3

Aqui as coisas mudaram um pouco. Nessa parte eu descobri uma maneira mais eficiente de carregar as imagens rs. No entanto, resolvi continuar utilizando a estratégia dos dicionários do jeito que estava fazendo antes.

A primeira parte dessa tarefa era testar tudo novamente agora com todas as bases QUEM, então eu criei uma pasta nova chamada "BaseQUEM" onde eu

juntei todas as bases de teste em um só lugar, para facilitar. Os resultados para as outras imagens de teste foram bem piores, ele acertou bem pouco no geral.

A segunda parte era pra utilizar outras formas de comparação de histogramas. Primeiro, eu li o tutorial do link fornecido na descrição da atividade e segui o passo a passo. Porém, lá ele utiliza apenas uma imagem que compara com outras 4, a minha intenção era só generalizar o código, mas estava ficando confuso porque não tinha sido eu quem tinha feito, por isso decidi ignorar o código dado no tutorial e fazer o meu. Fiz quatro funções: uma para carregar a base (leitura das imagens e armazenamento de histogramas), uma para comparar os histogramas (utilizando os métodos do OPENCV), uma para classificar as imagens (eu filtro apenas os maiores valores) e por fim, uma função de teste, no qual a pessoa pode escolher uma imagem de teste e o programa retorna, para cada método de calcular a similaridade, qual o é a imagem modelo correspondente.

Os resultados utilizando essas funções foram melhores, para os cinco primeiros QUEM, o programa só errou o QUEM3, que era o homem de ferro e ele disse que era o Batman. Porém, do QUEM6 até o QUEM16, ele falou que todo mundo era o homem de ferro. Algumas funções foram um pouco melhores, como a Hellinger, mas mesmo assim, os resultados não foram tão bons.

3 Parte 4

Nessa parte, ficou um pouco mais complicado. Até então, eu só tinha uma imagem modelo para utilizar, aí nesse tinha 4 imagens rs. Eu fiquei pensando se eu deveria escolher uma e utilizar normalmente ou todas. Resolvi utilizar todas.

Essencialmente, a ideia foi: para cada ONDE comparar cada MODELO e guardar os maiores resultados em uma lista e por fim, ordenar essa lista pegando apenas os maiores valores para cada método de similaridade. Vou explicar melhor o passo a passo abaixo.

A ideia dessa parte, era reutilizar o código feito na parte 3, eu reutilizei (continuo tendo quatro funções: `carrega_base`, `compara_histogramas`, `classifica` e `input_teste`), porém tive que fazer algumas modificações por conta da mudança de quantidade de imagens modelo.

Primeiro, fiz o carregamento das imagens da mesma maneira que foi feito nas atividades anteriores, com a diferença que agora, a base modelo tem uma lista de imagens, logo, uma lista de histogramas. Em seguida, criei a função `compara_histogramas`, onde eu comparei cada histograma utilizando as funções do OPENCV e do SCIPY. A estrutura do dicionário ficou resultante ficou assim: `{"METODO": {"ONDEx": {"CLASSE": } } }`. Nesse dicionário tem todas as informações, de todas as imagens de teste e a comparação com os histogramas; ele serve de entrada para a função `classifica`, que me retorna um novo dicionário contendo as imagens teste mapeadas em alguma classe. A partir daí, eu utilizo o dicionário resultante da classificação na função `input_teste`, onde a pessoa escolhe um ONDE que deseja saber, e o programa retorna, para cada método de cálculo de similaridade, qual classe seria.

O resultado da classificação até que ficou boa. O melhor método foi o de Hellinger, ele não errou nenhum para essa base de dados. O ONDE3 é a mais difícil de classificar, quase todos os métodos erram, exceto a Hellinger e Chebyshev. E a mais fácil de classificar é a Shire, todo mundo acerta.

4 Conclusão

Esse trabalho foi bem divertido de se fazer. Deu para ver como uma técnica simples (comparação de histogramas) pode ser utilizada para fazer uma coisa tão legal e importante, embora não seja a melhor técnica para se fazer isso, é um começo para quem quer se aventurar no mundo de processamento de imagens. Outra coisa importante é saber que técnica utilizar para calcular a similaridade entre os histogramas e que, na função dada no pdf para calcular a interseção, a ordem dos histogramas importa rs. Deu para notar o quão poderoso e divertido é trabalhar com imagens. Em relação a melhor forma de fazer um código que processa imagens, eu ainda preciso aprender muito sobre como otimizar as coisas rs. No geral, o trabalho ajudou a solidificar aquilo que foi visto em aula e forma prática e dinâmica.