

MAC5768 - Visão e Processamento de Imagens

EP3 - Relatório

Erico Tiago Meireles
Nº USP: 9312428
Prof.: Ronaldo Fumio Hashimoto

O Exercício Programa 3 (EP3) da disciplina envolveu segmentação de imagens, extração de características e classificação. A primeira parte do projeto foca na segmentação dos objetos de interesse, utilizando técnicas de suavização de imagens e detecção de arestas para criar uma representação binária que destaca as bordas dos objetos.

Após a aplicação de diferentes métodos de suavização e detecção de arestas, para uma amostragem de todas as 1309 imagens existentes (contando todas as imagens originárias do EP2), foi escolhido o método de Gaussian Blurring em conjunto com o de Canny Edge Detection, a partir disso, todos contornos dos objetos são traçados e a imagem salva com esse contorno.

Então, dentro do código é feita a avaliação dessas imagens, como boas ou não para extrair as características, as imagens classificadas como boas ou ruins são salvas no csv 'image_evaluations'. Então a partir desse dataframe, as imagens selecionadas como boas (200) são salvas no diretório selectedData, tanto as originais em cinza (com todas as transformações) quanto a imagem já com o contorno.

Na segunda parte, o objetivo é extrair características significativas dos contornos dos objetos, como área, diâmetro e a maior perpendicular ao diâmetro (minor). Utilizando a biblioteca skimage.measure, as propriedades dos objetos são calculadas automaticamente, o que economiza tempo e aumenta a precisão dos dados. Uma tabela é montada com essas características. Para associar cada contorno ao seu respectivo objeto, foi feito a avaliação manualmente, para cada contorno foi salvo uma nova imagem dentro do selectedData/contours, e a partir disso pode-se classificar corretamente o objeto. Essa etapa é crucial para a construção de um conjunto de dados robusto que será utilizado na fase de classificação.

Importante ressaltar que manualmente foi classificado alguns contornos como "errados", geralmente backgrounds que tinham sido selecionados como contornos. Também importante comentar que nenhuma imagem segmentada retornou todos os objetos, os contornos sempre apareciam 2 ou até 3 vezes no mesmo objeto, dessa forma, uma mesma imagem gerou diferentes linhas para o mesmo objeto.

Finalmente, a terceira parte do EP3 aborda o treinamento de um modelo de classificação e a classificação dos objetos, o modelo é treinado e avaliado em termos de acurácia. Vemos que os 10 objetos são desbalanceados na nossa base, o que mostra que a detecção de bordas funcionou melhor para alguns objetos do que outros, exemplo protetor solar com mais de 150 ocorrências, enquanto a pilha teve menos de 40.

Foi treinado um modelo inicial de Random Forest, esse estava claramente em overfitting, com performance de quase 100% no treino, mas abaixo dos 40% no teste.

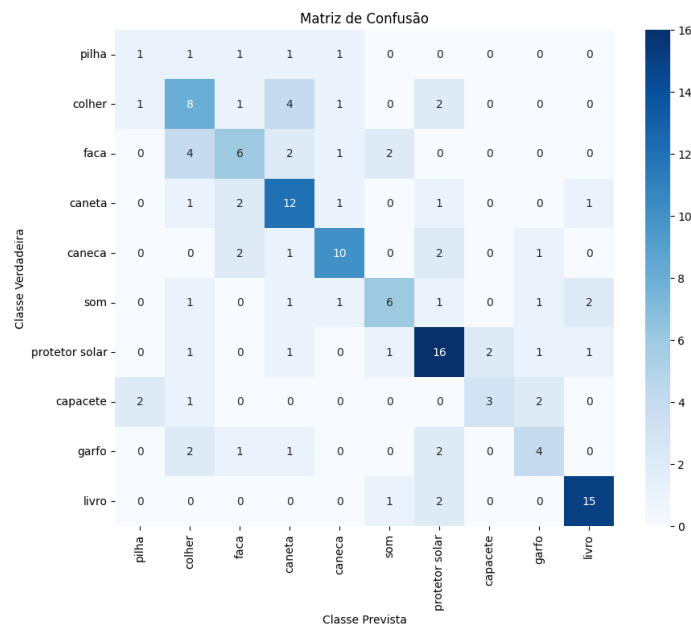
Para melhorar o modelo, foi definido os melhores hiper parâmetros para o Random Forest e, posteriormente, conforme solicitado no enunciado, será feita uma iteração variando o estado aleatório. O melhor modelo teve acurácia de 57% nos testes e os seguintes hiperparâmetros:

```
1 best_clf
[100] ✓ 0.0s
...
* RandomForestClassifier
RandomForestClassifier(max_depth=30, min_samples_leaf=2, min_samples_split=5,
n_estimators=200)
```

Então foi variado o estado aleatório, obtendo os melhores resultados em duas iterações:

```
[102] ✓ 4m 57.3s
...
Iteração 146: Melhor acurácia (56.64%)
Iteração 264: Melhor acurácia (56.64%)
```

A partir daí foi feito a avaliação do modelo no treino e teste (acurácia de 86% e 56% respectivamente). Na figura abaixo, é possível ver a matriz confusão dos objetos no teste.



Acurácia de 56% para o teste ainda é abaixo das expectativas, para melhorias futuras seria importante reavaliar todo o dataset para validar se a classificação foi feita de maneira correta, além disso, seria importante refazer todos os passos do aumento de dataset de imagens para criar situações em que a detecção de borda seja feita de maneira mais eficiente. Outra solução seria treinar outros modelos como SVMs e rede neurais.

Outra possibilidade seria refazer as fotos, pois elas poderiam ter tido melhor iluminação e menos variação para que a efetividade do modelo fosse melhor.

Por fim, como solicitado, foi gerado para algumas amostras a imagem com os objetos reconhecidos dentro de um retângulo:

