

Desafio PixForce

Abdou Halique Atanda Adechinan Bouari

Alisson SOBRENOME

Filipe Jose de Souza

Eliezer Soares Flores (Mentor)

Universidade Federal do Pampa, Campus Alegrete

1 Tarefa 1: Contagem de peça perfil

1.1 Método Proposto

Para esta tarefa, realizou-se um teste aplicando-se uma quantização que consiste em discretizar a cor nas imagens. Baseando-se nos *labels* fornecido pelos organizadores, conseguiu-se desenhar os contornos das imagens do *datasets* Anotados. Em seguida buscou-se a treinar uma rede neural com as informações alcançadas. Assim optou-se pela rede YOLO¹. Neste caso se implementou uma rede *fine tuning YOLOv7*². Também empregou-se a segmentação semântica U-net³. No caso da U-net, através das imagens dos *datasets* fornecidos, foram criadas máscaras através de um *script* em python com a biblioteca OpenCV que serviram como máscaras para o treinamento da rede convolutiva, utilizando o modelo proposto no trabalho de Olaf Ronneberger, gerando segmentações de imagem. Por fim, encontraram-se dificuldades por falta de acesso a recursos computacionais e tempo.

¹<https://pjreddie.com/darknet/yolo/>

²[https://learnopencv.com/fine-tuning-yolov7-on-custom-dataset/
#Fine-Tuning-YOLOv7-on-the-Pothole-Detection-Dataset](https://learnopencv.com/fine-tuning-yolov7-on-custom-dataset/#Fine-Tuning-YOLOv7-on-the-Pothole-Detection-Dataset)

³<https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/>

2 Tarefa 2: Contagem de MDPs

2.1 Método Proposto

Para esta tarefa, inicialmente, optou-se por segmentar cada imagem de maneira automática utilizando um classificador convencional, a saber, a máquina de vetores de suporte (do inglês, *Support Vector Machine* – SVM), com um *kernel* linear e com os parâmetros *default* do scikit-learn,⁴ para predizer o rótulo de cada pixel como objeto ou fundo. Essa máquina foi treinada com *patches* 5×5 de imagens rotuladas para as quais foram produzidos manualmente os *ground-truths* de segmentação usando a ferramenta *open-source* GIMP.⁵ Especificamente, o conjunto de treinamento foi composto por 200 *patches* (100 *patches* de objeto e 100 *patches* de fundo) de cada uma das 89 imagens para as quais os *ground-truths* de segmentação foram elaborados (ou seja, de cada das imagens reservadas para o treinamento, as quais correspondem a aproximadamente 2/3 do total de imagens rotuladas – visão de canto – disponíveis para esta tarefa).

Em seguida, a máscara de segmentação inicial, resultante do processo descrito no parágrafo anterior, foi empregada para inicializar o método GrabCut⁶ que resulta em uma máscara de segmentação mais exata (adicionalmente, empregou-se também um preenchimento morfológico de regiões para o pós-processamento da máscara), a qual foi utilizada para selecionar a região de interesse a ser processada na imagem, conforme detalhado no parágrafo a seguir.

Para efetivamente estimar o número de MDPs, primeiramente, aplicou-se o filtro de detecção de bordas de Canny⁷ sobre a imagem na região de interesse. Em seguida, determinou-se a maior distância dividida pela distância média, considerando-se as distâncias entre os cruzamentos por zero, para várias colunas da imagem. A maior razão foi considerada como sendo o total de MDPs detectados.

2.2 Resultados

Nas 61 imagens de teste (visão de canto) e Nas imagens de teste (visão frontal), o método proposto apresentou um erro de contagem médio de $\pm 60\%$.

⁴<https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html>

⁵<https://www.gimp.org/>

⁶https://docs.opencv.org/3.4/d8/d83/tutorial_py_grabcut.html

⁷https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html