

# APRENDIZADO DE TRANSFORMAÇÕES DE IMAGENS VIA CLASSIFICAÇÃO DE MICRORREGIÕES<sup>1</sup>

Pedro H. B. de Almeida\*  
Orientadora: Nina S. T. Hirata\*

\* Instituto de Matemática e Estatística da Universidade de São Paulo



## INTRODUÇÃO

Sistemas de visão computacional dependem de informações extraídas das imagens. A maioria das análises depende, por sua vez, de um processamento adequado das imagens, dentre os quais destaca-se a segmentação de regiões de interesse.

O problema de segmentação de imagens consiste no particionamento do domínio de imagens em segmentos disjuntos, de forma que cada segmento corresponda a um objeto ou parte de objeto que sejam componentes de interesse na análise em questão.

O aprendizado de operadores de imagens refere-se ao problema de, dados pares de imagens entrada-saída, determinar a transformação que mapeia a imagem de entrada para a correspondente imagem de saída.

Devido à quantidade de pixels, o custo computacional do processamento pixel a pixel é alto. A classificação de microrregiões pode ser uma opção interessante para se contornar esse problema. Com isso, métodos que precisariam ser aplicados a cada pixel individualmente podem ser substituídos por métodos que processam microrregiões.

## OBJETIVOS

Comparar a segmentação de imagens realizada via classificação pixel a pixel com a realizada via classificação de microrregiões.

## MÉTODOS

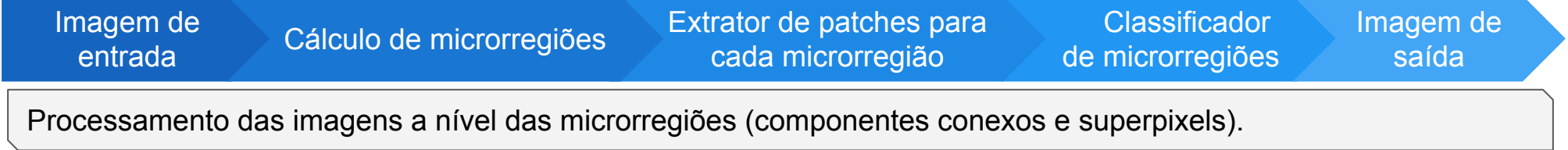
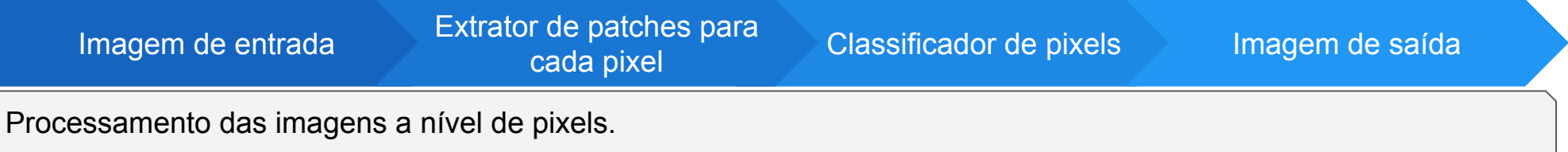
Como conjunto de dados, foram utilizadas imagens de revistas, nas quais, cada pixel da imagem estava rotulado como "texto" (valor 0) ou "não-texto" (valor 1) [1].

Para a extração de componentes conexos (CC) da imagem, utilizou-se a função "label" da biblioteca scikit-imagem que rotula regiões conexas de um array de inteiros. Dois pixels estão conectados quando eles são vizinhos e têm o mesmo valor. Para cada CC, atribuiu-se o rótulo mais frequente sobre o componente.

Já para a extração de superpixels (SP), utilizou-se uma versão própria do SLIC, que gera superpixels apenas em determinadas regiões da imagem. Para gerar esses superpixels locais, usa-se o algoritmo de *clustering* K-means num espaço cujas dimensões são compostas pela cor e pela distância espacial dos pixels (x,y,z) [2]. Para cada SP, atribuiu-se o rótulo mais frequente sobre ele.

Com relação ao treinamento, enquanto que a extração de features é feita de modo manual por quem está treinando no modelo de Redes Neurais, no aprendizado profundo, almeja-se que essa extração ocorra de maneira automatizada, junto com a classificação. Nesse sentido, utilizou-se uma Rede Neural Convolucional (ConvNet), que é bastante parecida com uma Rede Neural, isto é, ela é feita de neurônios que têm pesos e vieses que serão aprendidos. A diferença é que uma ConvNet assume explicitamente que as entradas são imagens, o que permite codificar certas propriedades na arquitetura [3].

Os *pipelines* desse problema são ilustrados a seguir:



## RESULTADOS

Dimensões da imagem de treino: 250x346x3 (RGB)			
	Pixels	Componentes conexos	Superpixels
Número de <i>patches</i>	86500	135	277
Tempo para extração de <i>patches</i> (segundos)	1.7740120887756348	0.21044301986694336	2.9262919425964355
Acurácia	0.9183072463768116	0.9832775919732442	0.9763651181744091
Precisão	0.8150964391691394	0.9857442348008386	0.9816053511705686
<i>Recall</i>	0.7692532088681447	0.9974543911752227	0.9940728196443692
Dimensões da imagem de treino: 2202x3047x3 (RGB)			
	Pixels	Componentes conexos	Superpixels
Número de <i>patches</i>	MemoryError	3322	16395
Tempo para extração de <i>patches</i> (segundos)	MemoryError	5.441878795623779	62.490108489990234
Acurácia	MemoryError	0.9959069366652306	0.9908612811449263
Precisão	MemoryError	0.9973770491803279	0.9969210174029451
<i>Recall</i>	MemoryError	0.9984682713347921	0.9889774236387783

## DISCUSSÃO

O número de *patches* foi reduzido em aproximadamente 99,84% no caso dos componentes conexos (CCs) e em 99,67% no caso dos superpixels (SPs). Isso possibilita que o treinamento seja feito com imagens de maiores dimensões e em maior quantidade.

Nota-se que o desempenho da classificação de microrregiões foi superior ao da classificação pontual, sendo os CCs a granularidade que mais se destaca na acurácia.

Em contrapartida, chega-se a perder muito tempo para calcular as microrregiões, sendo o cálculo dos SPs, através do algoritmo localSLIC, o mais demorado. No entanto, como a quantidade de pixels chega com muita frequência aos milhões, pode acontecer do tempo para o cômputo dos *patches* de CCs ser menor que o tempo para o cálculo dos *patches* de cada um dos pixels.

Em suma, as evidências indicam que a granularidade mais vantajosa para se fazer a classificação binária "texto" e "não-texto" é a de componentes conexos.

## AGRADECIMENTOS

Pedro H. B. de Almeida foi financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) sob o processo nº 2018/11899-8.

## REFERÊNCIAS

- [1] Julca-Aguilar, F. D., Maia, A. L. M., and Hirata, N. S. T. (2017). Text/non-text classification of connected components in document images. In 30th Conference on Graphics, Patterns and Images (SIBGRAPI), pages 450–455.
- [2] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Susstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell., 34(11):2274–2282.
- [3] Karpathy, A. Class notes. CS231n: Convolutional Neural Networks for Visual Recognition.

<sup>1</sup> Projeto de Iniciação científica apresentando no contexto da disciplina MAC0215 - Atividade Curricular em Pesquisa.