

**Fundação Getulio Vargas
Escola de Matemática Aplicada
Curso de Graduação em Matemática
Aplicada**

Título da dissertação

Emanuel Bissiatti de Almeida

Rio de Janeiro - Brasil
2023

**Fundação Getulio Vargas
Escola de Matemática Aplicada
Curso de Graduação em Matemática
Aplicada**

Título da dissertação

“Declaro ser o único autor do presente projeto de monografia que refere-se ao plano de trabalho a ser executado para continuidade da monografia e ressalto que não recorri a qualquer forma de colaboração ou auxílio de terceiros para realizá-lo a não ser nos casos e para os fins autorizados pelo professor orientador.”

Nome

Rio de Janeiro - Brasil
2023

**Fundação Getulio Vargas
Escola de Matemática Aplicada
Curso de Graduação em Matemática
Aplicada**

Título da dissertação

**“Projeto de Monografia apresentado à Escola de Matemática
Aplicada como requisito parcial para continuidade ao trabalho de
monografia.”**

Aprovado em ____ de _____ de _____
Grau atribuído ao Projeto de Monografia: _____

**Professor Orientador:
Escola de Matemática Aplicada
Fundação Getúlio Vargas**

1 Metodologia

O problema de dado uma imagem, classificar os seguimentos de reta de acordo com o respectivo ponto de fuga que passa por essa aresta. Para resolver isso, o problema foi subdividido em três etapas:

1. Detecção de arestas
2. Detecção de retas
3. Classificação de seguimentos

Cada etapa posterior depende da etapa anterior, sendo assim, a saída da etapa i é a entrada da etapa $i + 1$. A seguir, cada etapa será detalhada.

2 Detecção de arestas

A detecção de arestas é um problema clássico de processamento de imagens, onde o objetivo é identificar os limites de objetos contidos em uma imagem. Em especial, esse trabalho busca identificar as arestas que delimitam as faces de edifícios, que em geral são linhas retas.

2.1 Algoritmos clássicos

Os algoritmos clássicos de detecção de arestas são baseados na aplicação de filtros em uma imagem que identificam mudanças bruscas de intensidade. Observe que a imagem de uma aresta é uma região onde a intensidade da imagem muda bruscamente, sendo assim, os algoritmos de detecção de arestas são baseados em filtros que identificam essas mudanças de intensidade por meio de derivadas da intensidade da imagem ([SHRIVAKSHAN; CHANDRA-SEKAR, 2012](#)). A figura 2.1, mostra a fotografia do edifício da biblioteca nacional e o resultado da aplicação do algoritmo de Sobel e do algoritmo de Canny.

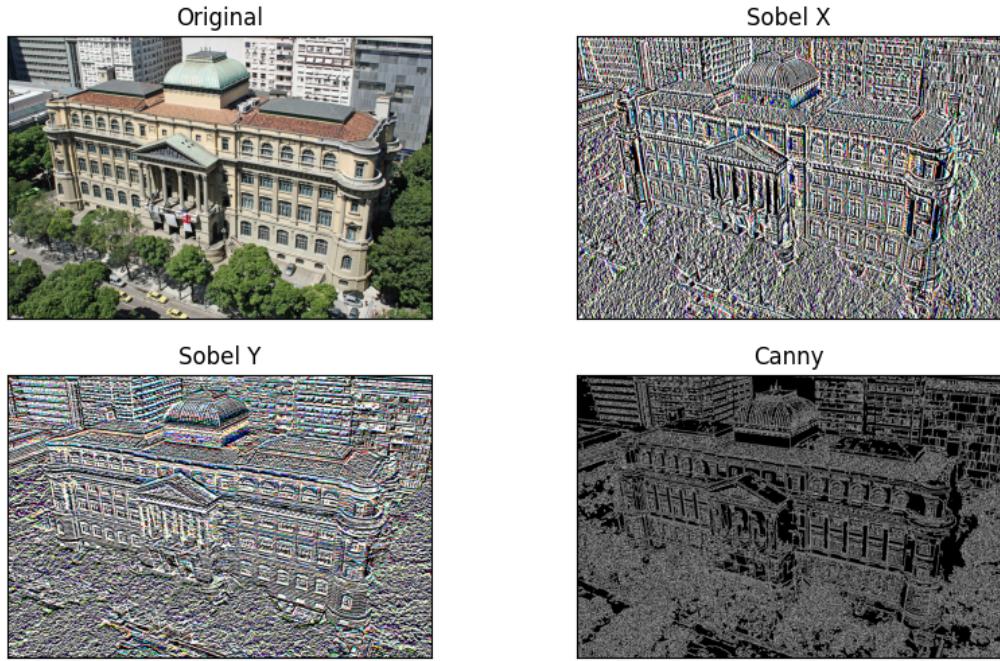


Figura: Imagem original e resultado do algoritmo de Sobel.

O algoritmo de Sobel é um filtro convolucional que calcula a magnitude do gradiente da intensidade da imagem. O algoritmo de Canny é um algoritmo mais sofisticado que utiliza o algoritmo de Sobel para calcular a magnitude do gradiente da intensidade da imagem e aplica uma série de filtros para remover ruídos e destacar as arestas da imagem. Em geral o algoritmo de Canny apresenta melhores resultados que o algoritmo de Sobel, porém, o algoritmo de Canny é mais custoso computacionalmente, apesar disso, o custo computacional não é está sendo considerado nesse trabalho.

Em análise, ambos os algoritmos não apresentam resultados satisfatórios para essa aplicação. O algoritmo de Sobel apresenta muitos ruídos e o algoritmo de Canny apresenta muitas arestas que não são relevantes para a aplicação. Essa aplicação busca encontrar as arestas que delimitam as faces de edifícios, entretanto, esses algoritmos detectam todas as arestas da imagem, incluindo as que compõem a textura da parede dos edifícios e as janelas.

Uma abordagem para resolver esse problema seria utilizar um desses algoritmos e depois aplicar um modelo que seleciona o subconjunto de arestas

ótimas para a aplicação, ou seja, do conjunto de arestas dado pelos algoritmos, selecionar apenas aquelas que delimitam os prédios por meio de aprendizado de máquina. Entretanto, essa abordagem não é eficiente, pois, as técnicas de aprendizado de máquina precisa de um conjunto de treinamento para aprender a selecionar as arestas ótimas, e para obter esse conjunto de treinamento é necessário selecionar manualmente as arestas ótimas. Existem alguns conjuntos de que já foram selecionados manualmente, como o banco de dados ([COUPRIE et al., 2013](#), Nyu Dataset) que realiza uma ótima segmentação de objetos em uma imagem. Porém, para trabalhar com modelos de aprendizado de máquina em imagens é necessário um grande conjunto de dados para treinamento e um grande poder computacional para processar esses dados a qual não está disponível para a realização desse trabalho. Sendo assim, foi decidido utilizar um modelo de aprendizado profundo pré-treinado para a seleção de arestas. Essa abordagem escolhida será melhor comentada na próxima seção.

2.2 Redes neurais convolucionais

As redes neurais ([BISHOP, 2016](#)) são uma combinação de funções matemáticas lineares e não lineares que, em uma sequência de camadas, transformam a informação produzindo uma saída única para cada entrada a partir dos pesos da rede. Em geral, esses pesos são iniciados aleatoriamente e são ajustados por meio de um algoritmo de otimização que minimiza uma função de perda. Entretanto, quanto maior a rede, maior é o número de parâmetros que precisam ser ajustados e maior é o custo computacional para realizar o ajuste desses parâmetros, e alguns dados precisam de um grande número de parâmetros para serem representados. Por exemplo, uma imagem de 1000×1000 pixels precisa de $1000 \times 1000 = 1000000$ parâmetros para ser representada, considerando apenas uma camada de entrada. Por conta disso, uma das técnicas utilizadas para trabalhar com imagens é por meio das redes convolucionais.

Uma rede neural convolucional ([GOODFELLOW; BENGIO; COURVILLE, 2016](#)) é um conjunto de técnicas de aprendizado profundo que foram desenvolvidas para trabalhar com dados tabulares. Essa técnica é uma extensão das redes neurais tradicionais, onde as camadas de neurônios são organizadas em camadas convolucional e camadas de pooling. As camadas convolucional são responsáveis por extrair as características da imagem e as camadas de pooling são responsáveis por reduzir a dimensionalidade da imagem. A figura 2.2 mostra um exemplo de uma rede neural convolucional,

nesse exemplo, após aplicar uma rede neural convolucional foi Aplicada uma rede neural tradicional para classificar a imagem.

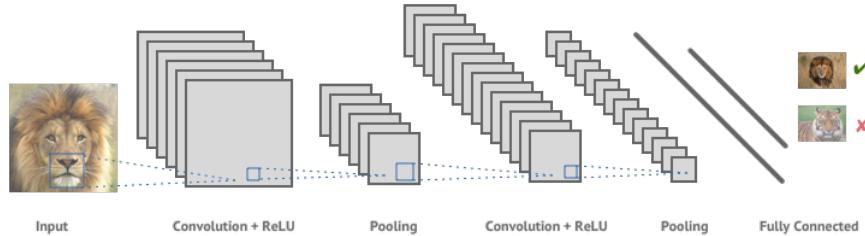


Figura: Rede neural convolucional.

Durante o treinamento de uma rede neural convolucional, os pesos são ajustados por meio de um algoritmo de otimização que minimiza uma função de perda. Entretanto, o ajuste dos pesos de uma rede neural convolucional é mais eficiente que o ajuste dos pesos de uma rede neural tradicional, pois, os pesos são compartilhados entre as camadas convolucionais, ou seja, os pesos são ajustados para todas as regiões da imagem. Por exemplo, considere uma imagem de 1000×1000 pixels e uma camada convolucional de 3×3 pixels, nesse caso, os pesos são ajustados para todas as regiões de 3×3 pixels da imagem. Sendo assim, o número de parâmetros que precisam ser ajustados é muito menor que o número de parâmetros de uma rede neural tradicional. Além disso, o cientista não precisa se preocupar em selecionar as características da imagem, pois, as camadas convolucionais são responsáveis por extrair as características da imagem automaticamente, os filtros possuem seus parâmetros ajustados durante o treinamento da rede neural, sendo que a única preocupação do cientista é definir a arquitetura da rede neural o que incluiu o número de camadas convolucionais e o número de filtros de cada camada convolucional.

Treinar uma rede neural convolucional do zero é um processo custoso, pois, é necessário um grande conjunto de dados para treinamento e um grande poder computacional para processar esses dados. Felizmente, após um modelo ser treinado, isto é, após encontrar os pesos ótimos de uma rede neural dado a função de erro e o conjunto de dados, é possível reutilizar esse modelo em outras aplicações, e, caso seja necessário, aplicar um pós-processamento para direcionar o modelo para uma aplicação específica. Sendo assim, foi

encontrado na literatura um modelo de aprendizado profundo que já foi treinando para a tarefa de detecção de arestas, um dos modelos já estabelecidos para essa tarefa é o Holistically-Nested Edge Detection (HED) (XIE; TU, 2015). Esse modelo foi um aprimoramento da rede VGG-16 (SIMONYAN; ZISSERMAN, 2014), uma rede convolucional genérica que possuiu seus pesos treinados com milhões de imagens para Classificação, especializando-o para a detecção de arestas a partir do banco de dados BSDS500 (ARBELAEZ et al., 2011) que contém 500 imagens de treinamento e 200 imagens de teste que tiveram suas arestas selecionadas manualmente. Após o treinamento, o HED foi capaz de selecionar as arestas das imagens do banco de dados BSDS500 com uma acurácia de 0.790.

A vantagem do modelo HED em relação os modelos clássicos de detecção de arestas é que, por meio do aprendizado de máquina, ele consegue delimitar as bordas que compõem os objetos da imagem. Isso ocorre porque esse modelo conseguiu aprender as representações hierárquicas das imagens, ou seja, ele resolve o problema de ambiguidade entre a detecção de bordas e a segmentação de objetos. A figura 2.2 mostra a fotografia do edifício da biblioteca nacional e o resultado da aplicação do modelo HED.



Figura: Imagem original e resultado do modelo HED.

2.3 Resultados preliminares

A seguir são apresentados os resultados comparativos da detecção de arestas utilizando o algoritmo de Canny e o modelo de aprendizado profundo pré-treinado HED. A figura 2.3 mostra a imagem original e o resultado da aplicação do algoritmo de Canny e do modelo HED. Observe que o modelo HED consegue selecionar as arestas que descreve melhor a forma dos

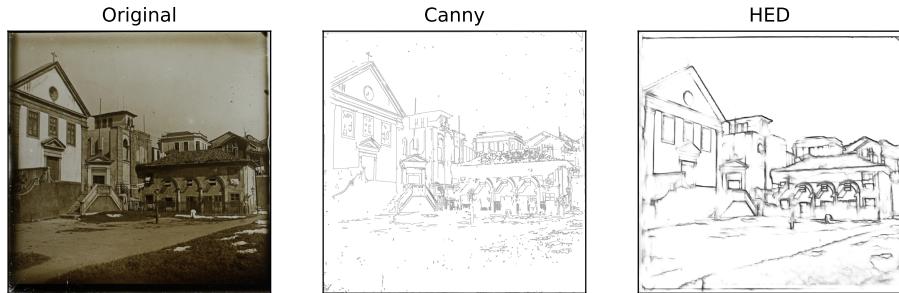


Figura: Imagem original e resultado da detecção de arestas na base de dados do ImageRio.

edifícios, mas, distorce um pouco a forma das arestas. Por outro lado, o algoritmo de Canny não consegue selecionar as arestas que o descrevem melhor, mas, não distorce a forma das arestas. Para realizar o processamento foi utilizados a implementação dos modelos pela biblioteca ([CORPORATION, 2023](#), Opencv) na interface Python.

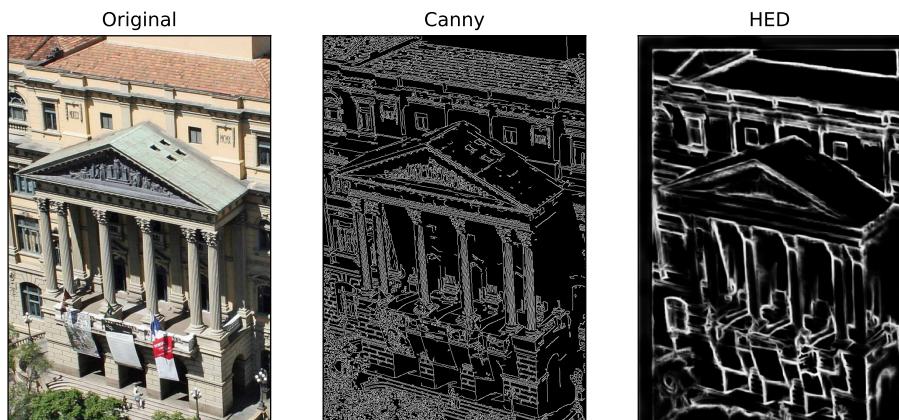


Figura: Imagem original e resultado da detecção de arestas.

A figura 2.3 mostra a imagem original e o resultado da aplicação do algoritmo de Canny e do modelo HED na base de dados do ImageRio, o banco de dados motivador a esse trabalho.

3 Detecção de arestas

O retorno da etapa anterior é uma imagem segmentada, onde os objetos que compõem a imagem original estão destacados enquanto as outras frequências da imagem não apresenta informação relevante. Sendo assim, nessa etapa são utilizados algoritmos tradicionais de computação gráfica para determinar as equações das retas que passam pelos seguimentos da imagem:

3.1 Transformada de Hough

A transformada de Hough é uma técnica que permite a detecção de formas geométricas em uma imagem. A técnica foi desenvolvida por Paul Hough em 1962,

3.2 Detecção de seguimentos

3.3 Aprimoramento

Observa-se que há muito ruído em alguns traços dos objetos selecionados pelo modelo de aprendizado de máquina, para evitar isso,

4 Classificação de seguimentos

A saída da etapa anterior S contém o conjunto de pontos que descrevem os limites de seguimentos de retas contidos na imagem

5 Referências

Referências

- ARBELAEZ, Pablo et al. Contour Detection and Hierarchical Image Segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 33, n. 5, p. 898–916, 2011. DOI: [10.1109/TPAMI.2010.161](https://doi.org/10.1109/TPAMI.2010.161).
- BISHOP, C.M. **Pattern Recognition and Machine Learning**. [S.l.]: Springer New York, 2016. (Information Science and Statistics). ISBN 9781493938438. Disponível em: <https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf>.
- CORPORATION, Intel. **Open Source Computer Vision Library**. [S.l.: s.n.], 2023. <https://opencv.org/>.
- COUPRIE, Camille et al. Indoor semantic segmentation using depth information. **arXiv preprint arXiv:1301.3572**, 2013.
- GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.
- SHRIVAKSHAN, GT; CHANDRASEKAR, Chandramouli. A comparison of various edge detection techniques used in image processing. **International Journal of Computer Science Issues (IJCSI)**, Citeseer, v. 9, n. 5, p. 269, 2012.
- SIMONYAN, Karen; ZISSERMAN, Andrew. Very Deep Convolutional Networks for Large-Scale Image Recognition. **arXiv preprint arXiv:1409.1556**, 2014.
- XIE, Saining; TU, Zhuowen. **Holistically-Nested Edge Detection**. [S.l.: s.n.], 2015. arXiv: [1504.06375 \[cs.CV\]](https://arxiv.org/abs/1504.06375).