



SALÃO DE  
INICIAÇÃO CIENTÍFICA  
XXX SIC

15 A 19  
OUTUBRO  
CAMPUS DO VALE



# Um Estudo sobre Abordagens Modernas para Processamento de Imagens



Bolsista: Arthur Endres Balbão

Orientador: Manuel Menezes de Oliveira Neto

## 1. Introdução

Este trabalho consistiu da realização de um estudo sobre abordagens modernas para processamento de imagens. Foram contemplados a linguagem de programação *Halide* e os fundamentos de *deep learning*. *Halide* visa facilitar o desenvolvimento de aplicações de alto desempenho em processamento de imagens, introduzindo a separação explícita de um programa em dois componentes: algoritmo e *schedule*. Já *deep learning* é um campo de pesquisa na área de aprendizado de máquina que se utiliza de redes neurais convolucionais (CNNs). Essas redes estão no centro de diversos avanços recentes na área de inteligência artificial, constituindo o estado-da-arte em diversas subáreas de processamento de imagens e visão computacional, notadamente em processamento de imagens médicas e controle de veículos autônomos.

## 2. Halide

### 2.1. Algoritmo vs schedule

Um conceito fundamental introduzido pela linguagem *Halide* é a separação entre as noções de **algoritmo** e **schedule**. Um **algoritmo** é a descrição das operações que devem ser executadas a fim de se realizar uma tarefa. Já o **schedule** estabelece a forma na qual essas operações serão executadas pelo processador. Assim, esta separação permite testar diversas opções de otimização (diversos *schedules*) para um mesmo algoritmo e identificar a mais eficiente. Essa separação explícita pode ser vista no procedimento mostrado na Figura 1.

```
Func halide_blur(Func in) {  
  Func tmp, blurred;  
  Var x, y, xi, yi;  
  
  // The algorithm  
  tmp(x, y) = (in(x-1, y) + in(x, y) + in(x+1, y))/3;  
  blurred(x, y) = (tmp(x, y-1) + tmp(x, y) + tmp(x, y+1))/3;  
  
  // The schedule  
  blurred.tile(x, y, xi, yi, 256, 32)  
    .vectorize(xi, 8).parallel(y);  
  tmp.chunk(x).vectorize(x, 8);  
  
  return blurred;  
}
```

Figura 1. Procedimento em Halide para borrar imagens, ilustrando a separação entre algoritmo e schedule [1].

### 2.2. Solução multiplataforma

A linguagem oferece suporte para geração de código para diversas arquiteturas e sistemas operacionais como *Windows*, *Linux*, *macOS*, *Android*, *iOS*, entre outros.

### 2.3. Metodologia de estudo

Como parte do estudo da linguagem *Halide*, foi realizado um tutorial constituído por uma série de programas que exemplificam diversos aspectos da mesma. Tais programas foram modificados para a obtenção de novos resultados como forma de realização de experimentos. Além disso, também foram lidos artigos técnicos publicados em periódicos internacionais, assim como palestras disponíveis online que tratam sobre o uso da ferramenta.

## 3. Deep Learning

### 3.1. Arquitetura em camadas

Redes neurais convolucionais se utilizam de uma arquitetura em camadas que realizam operações de convolução para identificar padrões em imagens. Além disso, também estão presentes camadas de *pooling*, cujo principal objetivo é diminuir o tamanho da representação a fim de reduzir o número de parâmetros, assim minimizando o custo computacional. A imagem abaixo representa a arquitetura da *VGG-net* (Figura 2), uma CNN utilizada para reconhecimento visual de objetos. Um exemplo de classificação de imagens é ilustrado na Figura 3.

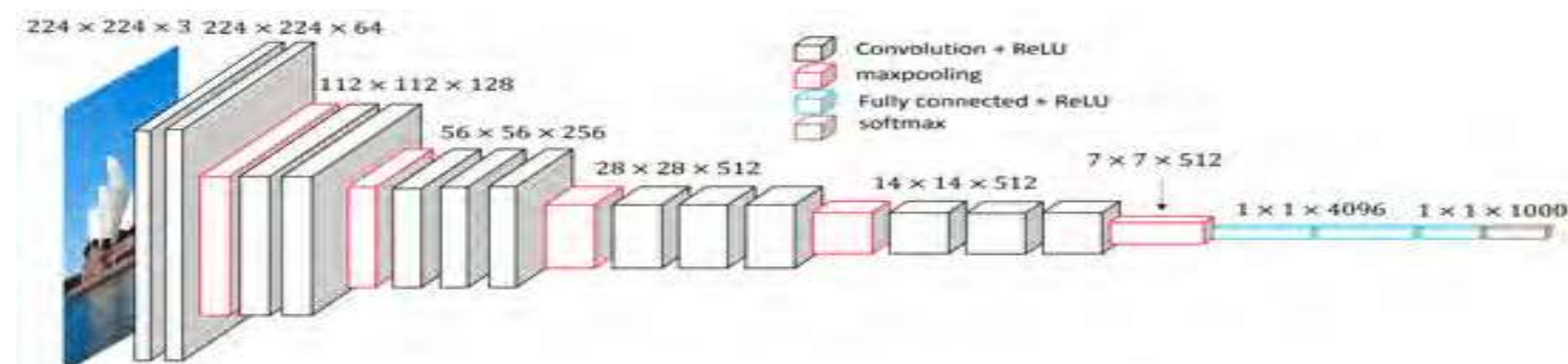
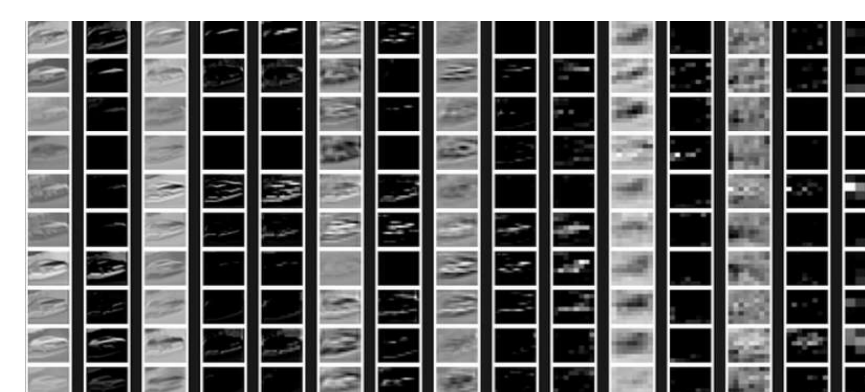


Figura 2. Exemplo de arquitetura em camadas de uma CNN (*VGG-net*) [2].



(a)



(b)



(c)

Figura 3. Exemplo de utilização de uma CNN para classificação de imagens. A rede recebe como entrada uma imagem (Figura 3a) e a processa por meio de várias camadas onde são realizadas operações de convolução e *pooling* (Figura 3b). A última camada atua como um classificador e indica a probabilidade de que a imagem seja de uma classe ou de outra. Para este exemplo, a rede classificou corretamente o objeto na figura como sendo um carro, isto é, essa classe foi interpretada como sendo a mais provável dentre as demais (Figura 3c) [3].

### 3.2. Estudos realizados

Foi realizado um curso on-line sobre *deep learning* [4] lecionado na Universidade de *Stanford* em 2017. Durante a realização deste curso, foram desenvolvidos trabalhos práticos como a implementação de uma rede neural e outros tipos de classificadores (*nearest neighbour*, máquina de suporte vetorial). Também foi realizado um curso on-line sobre aprendizado de máquina [5], assim como foi realizada a leitura de um livro sobre redes neurais [6].

## 4. Conclusão

A realização destas atividades propiciou um crescimento acadêmico e uma experiência muito importante para a formação do aluno, visto que as ferramentas estudadas podem ser utilizadas para a solução de inúmeros problemas em ciência e tecnologia.

## Referências

- [1] J. Ragan-Kelley, A. Adams, S. Paris, M. Levoy, S. Amarasinghe, and F. Durand. (2012). Decoupling algorithms from schedules for easy optimization of image processing pipelines. *ACM Trans. Graph.*, 31(4):32:1.
- [2] Acharya, Debaditya & Yan, Weilin & Khoshelham, Kourosh. (2018). Real-time image-based parking occupancy detection using deep learning, In Proc. of the 5th Annual Conference of Research@Locate, volume 2087, page 35.
- [3] Imagem disponível em <http://cs231n.github.io/convolutional-networks/>
- [4] Curso sobre CNNs da Universidade de *Stanford*, disponível em <https://youtu.be/vT1JzLTH4G4>
- [5] Curso sobre aprendizado de máquina lecionado por Andrew Ng, disponível em <https://pt.coursera.org/learn/machine-learning>
- [6] Nielsen, M. A. (2015) *Neural networks and deep learning*. Disponível em <http://neuralnetworksanddeeplearning.com/>

## Agradecimentos



CNPq Processos 104383/2018-6,  
306196/2014-0 e 423673/2016-5.