PLABIANY RODRIGO ACOSTA

plabiany@gmail.com

May 16, 2022

1 Introdução

A aprendizagem profunda (*Deep Learning*) tem se destacado no cenário de Visão Computacional e as Redes Neurais Convolucionais (CNN) [SZ14] tem dominado o estado da arte [JJM96]. Estudos podem ser encontrados para detecção de objetos [LRPM19, PMS+18, MdSA18, LDFY18, TGC18], segmentação de imagens [ZSQ+17, CSWS16], contagem de objetos [TGC18, CSWS16, LAM19], entre outros. Diversas aplicações da Visão Computacional apontam que os métodos de contagem e localização de objetos consistem em detectar e obter a posição de cada objeto individualmente na imagem. Esses métodos tem auxiliado no controle e contagem de pessoas [GLS+20], carros [AVKR18], animais [SGP+18], e até mesmo bactérias [FLS15].

Embora os métodos recentes tenham obtido resultados promissores, três desafios ainda se mantêm. O primeiro desafio é a alta densidade com objetos sobrepostos, o segundo desafio é a necessidade da rotulação detalhada das imagens, o que é difícil de obter em grandes quantidades, e o terceiro desafio consiste em contar e localizar objetos em uma sequência de imagens.

Este experimento apresenta um método de Redes Neurais Convolucionais para classificação de imagens. O modelo foi treinado e testado em um conjunto de dados que contém 5 marcas de cerveja. Ao receber uma imagem com uma bebida presente o método apresenta qual classe esta imagem pertence. O modelo alcançou 80.5% de precisão e 74.2% de acurácia para imagens de tamanho 256×256, e 80.1% de precisão e 77.2% de acurácia para imagens de tamanho 512×512 no conjunto de teste. Durante os experimentos foram analisados também o tempo de execução do método proposto.

1.1 Trabalhos Correlatos

As Redes Neurais Convolucionais tem dominado estudos de Aprendizado Profundo, e demonstram resultados promissores quando testadas em problemas reais. Tais redes usam a convolução em pelo menos uma de suas camadas, sendo a convolução uma operação linear [GBC16]. Além disso, são capazes de operar com uma ou mais dimensões, portanto é possível utilizar as imagens RGB (3 dimensões) e vídeos (n dimensões) como entradas para treinamento [ARK10].

Frente as dificuldades das CNN's, conforme descrito na Seção 1. As Redes Neurais Convolucionais ocuparam espaços nos artigos ciêntifícos e são considerados por muitos o estado da arte do Aprendizado Profundo (*Deep Learning*) [LRPM19, PMS+18, MdSA18, LDFY18, TGC18, ZSQ+17, CSWS16, TGC18, CSWS16, LAM19].

1.2 Classificação de Imagens

Jinzhu Lu em 2021 [LTJ21] apontou que a influência do fundo da imagem, na classificação final, não é clara. O autor descreve que deve-se levar em consideração o fundo ser homogêneo, e caso contrario a rede aprenderá informações do fundo, o que pode levar a resultados errôneos.

Vittorio Mazzia e outros autores em 2020 [MKC20] apresentaram um trabalho para classificação de imagens do satelite do google Sentinel-2. Mazzia utilizou uma combinação de Rede Neural Recorrente e camadas convolucionais. O método classifica cada pixel de uma imagem e obteve um precisão de 96.5%. Em 2019 Vittorio Mazzia e outros autores, apresentaram uma adaptação da Rede YOLOv3-tiny[MSG, RF18] para detecção de maçãs em imagens. São inseridos camadas convolucionais e *upsamplings* na entrada da rede YOLO. O texto afirma que a sequência de camadas foi inspirada na VGGNet[KS15]. O resultado é a detecção da região onde a maçã esta presente com uma média de precisão mAP 83.64, revocação 83.0% e precisão 69.0%. O texto aponta uma comparação do uso de hardware a fim de

acelerar o processamento das redes em tempo real. Uma análise acontece para vários equipamentos, dentre as comparações é apontado o melhor preço/custo do hardware e o quanto de energia podemos economizar para executar redes complexas.

2 Metodologia

Nesta seção são apresentados os detalhes do método proposto para classificação de imagens contendo cervejas. Na Figura 1 temos passo a passo explicativo das etapas que compõe a arquitetura do método proposto.

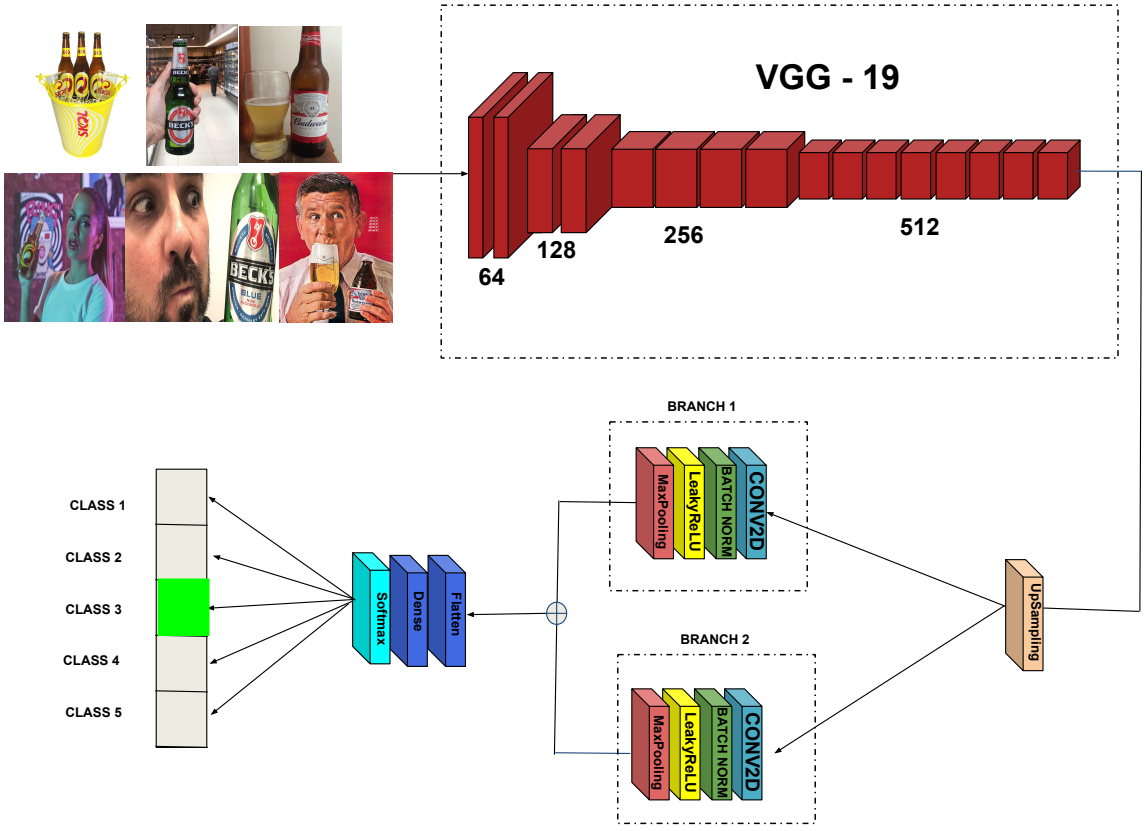

Figure 1: Ilustração do modelo utilizado para classificação de imagens de cerveja.

2.1 Banco de Dados

O banco de dados contém 329 imagens contendo 5 marcas de cerveja, com tamanhos $n \times n$ e n se altera para cada imagem. Na Figura 2 temos alguns exemplos e podemos verificar que as cervejas, assim como suas logomarcas, aparecem em multiplas posições e em várias situações.

2.2 Pré-processamento das Imagens

As imagens foram redimensionadas para um tamanho fixo de n, assim cada imagem passam a ter o mesmo tamanho e um formato quadratico pois $n \times n$. Se $n = 512$ então a imagem passa ter $512 \times 512 \times 3$ pixels. As RGB sofreram apenas alterações de tamanho, as imagens CMYK que possuem 4 canais sao convertidas para 3 canais utilizando os 3 primeiros canais. Cada imagem recebe como rótulo um valor inteiro informando a classe na qual a imagem pertence.

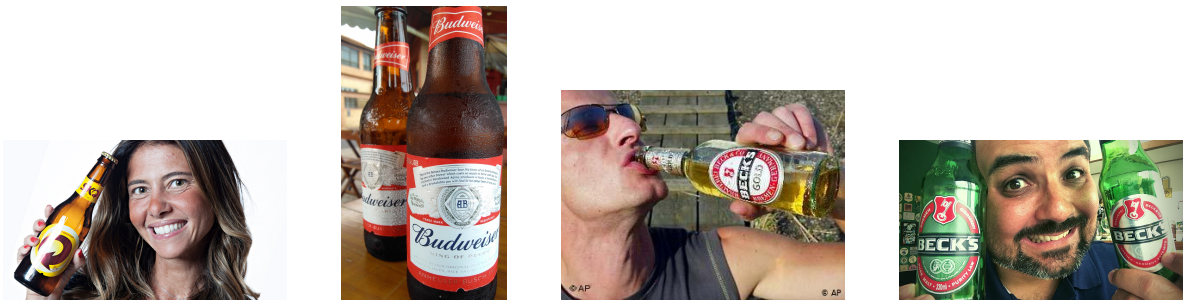

Figure 2: Exemplo de Imagens do Banco de Dados.

2.3 Metodologia para Classificação das Imagens

O método recebe uma imagem I de tamanho $n \times n$ e gera como saída as probabilidades de I pertencer a uma das 5 classes impostas. A saída gerada é um vetor com 5 posições e cada posição contém uma probabilidade de I ser daquela classe, desta forma onde conter a maior probabilidade é a posição/classe que I pertence.

A ordenação metodológica pode ser descrita em quatro passos principais e podemos acompanhar cada passo através da Figura 1. Inicialmente as imagens passam por um pré-processamento (passo 1) para que todas as imagens tenham um tamanho fixo $n \times n$. Após isso um mapa de características é obtido usando uma CNN (e.g., VGG19[KS15]) (passo 2). O mapa de característica é dado como entrada para o *branch 1* e *branch 2* (passo3). A saída dos *branchs* são concatenados, e os filtros são normalizados para a quantidade de classes(passo 4).

2.4 Extração do Mapa de Características

Nessa etapa, a imagem I de entrada é submetida a uma CNN VGGNet [KS15] para obtenção de um mapa de características. Esse mapa tem como objetivo extrair informações relevantes da imagem de entrada. A entrada da VGGNet consiste em uma imagen RGB com dimensão $n \times n \times 3$ e n tem um valor fixo.

2.5 Estimativa do Mapa de Densidade

Dado o mapa de característica F, o primeiro ramo do método proposto estima o mapa de densidade com o objetivo de classificar as imagens. Essa predição é realizada por dois conjuntos de camadas convolucionais denominados *branch*. Nos dois *branch*, uma serie de camadas Ω geram o mapa de densidade $\hat{C}^1 = \Omega(F)$. Sendo $\hat{C}^1$ a concatenação dos dois *branchs* e Ω a sequência de quatro camadas: a primeira é uma convolução com 64 filtros de tamanho 2×2, a segunda é a normalização para redimensionamento, na terceira camada aplica-se um $LeakyReLU$ e por fim na quarta temos um *MaxPooling2d*. As sequências de camadas dos *branchs* foram inspirados no trabalho de Vittorio Mazzia e outros autores, em 2019 [MSKC20].

2.6 Configuração dos Experimentos

Para avaliar a abordagem proposta, as amostras foram divididas aleatoriamente, 60% das imagens foram destinados ao treinamento, 20% para os teste e 20% para a validação. Para o treinamento, o otimizador Gradiente Descendente Estocástico foi usado com um momento de 0,9. O ajuste de hiper parâmetros na taxa de aprendizado e no número de épocas foi realizado usando o conjunto de validação para reduzir o risco de *overfitting*. Após experimentos preliminares para ajuste de hiper parâmetro, a taxa de aprendizado foi de 0,001 e o número de épocas foi de 100.

Todos os experimentos foram executados na GPU (Unidade de Processamento Gráfico) NVIDIA Tesla (CUDA version 11.2 e memória gráfica de 17 GB), sistema operacional do Google Colab, semelhante ao linux. A implementação proposta faz o uso da API (Interface de Programação de Aplicativos) de alto nível denominada Keras [C+15], escrita na linguagem de programação Python. As rotinas desta interface são executadas através da biblioteca de código aberto TensorFlow [AAB+15].

2.6.1 Transfer Learning

Em vez de treinar a abordagem proposta do zero, os pesos da primeira parte foram inicializados com pesos pré-treinados no ImageNet em um conceito conhecido como *transfer learning*. Esse método proporciona uma certa rapidez no treinamento de CNNs, evitando treinamentos do zero (aleatoriamente). Quanto maior e mais geral o conjunto de treinamento do modelo pré-treinado, maior será a eficiência da transferência de aprendizado ([AAB+15, LJY18]). A ImageNet é um conjunto que contém 1,2 milhões de imagens com 1000 categorias [RDS14]. Por ser um conjunto extenso existem vários modelos treinados nas arquiteturas mais conhecidas da literatura.

3 Resultados

Nas subseções abaixo são demonstrados os resultados com detalhes condizente com os experimentos e as técnicas que geraram resultados relevantes.

3.1 Resultados Quantitativos

Na Tabela 1 temos os resultados dos experimentos com o método proposto após o treinamento de 100 épocas. A tabela aponta os valores com base no tamanho das imagens. Conforme citado na Seção 2.2 as imagens tem tamanho fixo $n \times n$. A alteração no tamanho das imagens impacta nos valores resultantes de acurácia, assim como a métrica Kappa de Cohen [McH12]. As imagens tendo tamanho 32×32 possui 66% de acurácia e conforme o tamanho aumenta, o método passa a atingir valores maiores. Por outro lado quanto maior o tamanho da imagem maior será o tempo de execução, tanto no treinamento, quanto na predição de uma única imagem. O tamanho 512×512 apresentou 77.2% de acurácia, Kappa 70.0% e Precisão 80.1%.

Tamanho	**Acurácia**	**Kappa**	**Precisão**	**Revocação**	**F1 metric**
32×32	0.666	0.570	0.693	0.662	0.672
64×64	0.651	0.545	0.674	0.634	0.640
128×128	0.727	0.649	0.700	0.703	0.698
256×256	0.742	0.660	0.805	0.717	0.734
512×512	0.772	0.700	0.801	0.737	0.758

Table 1: Resultados experimentais para classificação de imagens contendo 5 marcas de cervejas.

.

Na Tabela 2 temos os valores resultantes do processamento no treinamento e na classificação de apenas uma imagem. Os valores estão representados em segundos. É notório que o tamanho da imagem interfere no tempo de exeucção. Não foram executados treinamentos para imagens maior que 512×512, pois o Colab Laboratory apresenta erros de alocação de memória.

Tamanho	**Tempo de Treinamento(seg)**	**Tempo Classificar 1 imagem(seg)**
32×32	33.488	0.349
64×64	42.167	0.344
128×128	77.990	0.362
256×256	263.109	0.504
512×512	923.315	0.983

Table 2: Resultados experimentais para classificação de imagens contendo 5 marcas de cervejas.

.

3.2 Resultados Qualitativos

Existe um tipo de imagem que o método acerta com maestria. Para entermos melhor apresento duas imagens na Figura 3. A primeira contém apenas uma garrafa(long neck) com o fundo branco e os símbolos da marca estao nítidos. Imagens assim facilmente são acertadas pelo método. Na segunda imagem ainda na Figura 3, temos uma mulher tomando uma cerveja com a latinha inclinada, além da coloração da latinha não ser a cor usual da marca. Esta imagem pode ser confundida pela rede.

Figure 3: Exemplos do Banco de Dados.

4 Conclusão

Os resultados atingiram valores significativos visto que as imagens apresentam várias situações e posições. Apresentados também uma ou mais quantidades de cervejas dentro de uma única imagem. O modelo alcançou uma acurácia de 77.2%, e precisão de 80.1% na classificação de cinco marcas de cervejas e um tempo de 0.50 segundos para classificar uma imagem de tamanho 512×512. Notoriamente quanto maior o tamanho das imagens maior são os valores apontados como resultado.

Após este experimento temos que é necessário a rotulação de cada imagem, cada simbolo da marca em específico. Talvez a rede passe a gerar resultados melhores usando no treinamento apenas as regiões onde a marca aparece.

References

[AAB+15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[ARK10] I. Arel, D. C. Rose, and T. P. Karnowski. Deep machine learning - a new frontier in artificial intelligence research [research frontier]. *IEEE Computational Intelligence Magazine*, 5(4):13–18, Nov 2010.

[AVKR18] Seyedmajid Azimi, E. Vig, Franz Kurz, and Peter Reinartz. Segment-and-count: Vehicle counting in aerial imagery using atrous convolutional neural networks. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-1:19–23, 09 2018.

[C+15] François Chollet et al. Keras. https://keras.io, 2015.

[CSWS16] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1611.08050, 2016.

[FLS15] A. Ferrari, S. Lombardi, and A. Signoroni. Bacterial colony counting by convolutional neural networks. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 7458–7461, Aug 2015.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[GLS+20] Paulo Gonçalves, Bernardo Lourenço, Samuel Santos, Rodolphe Barlogis, and Alexandre Misson. Computer vision intelligent approaches to extract human pose and its activity from image sequences. *Electronics*, 9:159, 01 2020.

[JJM96] A. K. Jain, Jianchang Mao, and K. M. Mohiuddin. Artificial neural networks: a tutorial. *Computer*, 29(3):31–44, March 1996.

[KS15] Andrew Zisserman Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2015.

[LAM19] Antonio C. Sobieranski1 Luiz Antonio Macarini. Using convolutional neural network to detect and count individuals on eucalyptus plantation. *X Computer on the Beach*, pages 1–1, 03 2019.

[LDFY18] Weijia Li, Runmin Dong, Haohuan Fu, and Le Yu. Large-scale oil palm tree detection from high-resolution satellite images using two-stage convolutional neural networks. *Remote Sensing*, 11(1), 2018.

[LJY18] FeiFei Li, Justin Johnson, and Serena Yeung. Cs231n: Convolutional neural networks for visual recognition. http://cs231n.github.io/assets/cnn/depthcol, Spring 2018.

[LRPM19] Axel Barroso Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key.net: Keypoint detection by handcrafted and learned CNN filters. *CoRR*, abs/1904.00889, 2019.

[LTJ21] Jinzhu Lu, Lijuan Tan, and Huanyu Jiang. Review on convolutional neural network (cnn) applied to plant leaf disease classification. *Agriculture*, 11(8), 2021.

[McH12] M. L. McHugh. Interrater reliability: the kappa statistic. *Biochemia Medica*, 22:276 – 282, 2012.

[MdSA18] J. F. Rodrigues Wesley N. Gonçalves Bruno B. Machado Marco d. S. Arruda, G. Spadon. Recognition of endangered pantanal animal species using deep learning methods. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2018.

[MKC20] Vittorio Mazzia, Aleem Khaliq, and Marcello Chiaberge. Improvement in land cover and crop classification based on temporal features learning from sentinel-2 data using recurrent-convolutional neural network (r-cnn). *Applied Sciences*, 10(1), 2020.

[MSG] S. Yu M. Sung and Y. Girdhar. Detecção de peixes em tempo real baseada em visão usando rede neural convolucional. In *OCEANS 2017 - Aberdeen*.

[MSKC20] Vittorio Mazzia, Francesco Salvetti, Aleem Khaliq, and Marcello Chiaberge. Real-time apple detection system using embedded systems with hardware accelerators: An edge AI application. *CoRR*, abs/2004.13410, 2020.

[PMS+18] A. P, H. MP, H. Sounder, N. K, V. P V, and R. Hebbar. Cnn based technique for automatic tree counting using very high resolution data. In *2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C)*, pages 127–129, April 2018.

[RDS14] Olga Russakovsky, Jia Deng, and Hao Su. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[RF18] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[SGP+18] F. Sarwar, A. Griffin, P. Periasamy, K. Portas, and J. Law. Detecting and counting sheep with a convolutional neural network. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, Nov 2018.

[SZ14] K. Simonyan and A. Zisserman. Very deep cnn for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[TGC18] H. Tayara, K. Gil Soo, and K. T. Chong. Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network. *IEEE Access*, 6:2220–2230, 2018.

[ZSQ+17] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.