

Análise de Desempenho da Execução Remota de Método Aplicado ao Monitoramento de Animais com VANT

**Bruno Campos de Vasconcellos¹, José Pedro Pereira Trindade²,
Leandro Bochi da Silva Volk², Leonardo Bidese de Pinho¹**

¹ Programa de Pós-Graduação em Computação Aplicada, UNIPAMPA, Bagé - RS, Brasil

²Embrapa Pecuária Sul, Bagé - RS, Brasil

brunoc.vasconcellos@gmail.com, jose.pereira-trindade@embrapa.br,

leandro.volk@embrapa.br, leonardopinho@unipampa.edu.br

Resumo. *Durante o desenvolvimento do “Método Aplicado ao Monitoramento de Animais Através de Imagens de VANT” onde imagens foram coletadas por meio de aerolevanteamento e após processadas com Redes Neurais Convolucionais, percebeu-se a necessidade potencial de emprego de unidades de processamento multi e many-core para aumentar a eficácia do método, em especial para o monitoramento de áreas maiores que 20 hectares. Neste contexto o presente trabalho avalia a eficácia de ambientes computacionais remotos com máquinas virtuais compostas por CPU e GPU, disponibilizadas por meio do Google Colab, tendo como referência as métricas tempo e acurácia. Os resultados confirmaram a eficácia e uma significativa melhora no desempenho obtido com a arquitetura many-core da GPU.*

1. Introdução

A computação aplicada ganha destaque nos mais diversos contextos, tendo papel fundamental para o aumento da eficiência na agropecuária, em particular na pecuária. Entretanto, em ambientes de manejo extensivo, a observação humana das diferentes variáveis que compõem este complexo sistema demanda tempo significativo, dependendo das características da propriedade (principalmente a extensão) e dos animais (em especial a quantidade/densidade). Entre as tecnologias disponíveis com potencial para auxiliar os pecuaristas no manejo extensivo se destacam os Veículos Aéreos Não Tripulados (VANT), capazes de sobrevoar áreas de interesse, transportando diversos tipos de sensores, de forma semiautônoma. Diante da necessidade de controlar o estoque de animais, representa uma das atividades mais importantes, implicando na necessidade de realizar um inventário de animais com frequência diária ou semanal. As variações no estoque de animais podem ocorrer por morte, nascimento, abigeato (furto ou roubo de animais) e transferência entre produtores por meio de venda [Barbalho et al. 2005].

Como solução para este problema foi desenvolvido o “Método Aplicado ao Monitoramento de Animais Através de Imagens de VANT”, o qual tem sido testado com base em aerolevanteamentos em uma área experimental de pastagens de 20 hectares da Embrapa Pecuária Sul, em Bagé-RS. As imagens, após o aerolevanteamento, foram submetidas a uma técnica de aprendizado de máquina profundo, afim de encontrar os animais nas imagens e contá-los na área, sendo os resultados contrastados com os registros de presença de animais na área de interesse para avaliação da acurácia do método. Mais

precisamente, o método se baseia na técnica *Fast Region-based Convolutional Network* (Fast R-CNN), a qual usa como entrada uma imagem inteira e um conjunto de objetos propostos. A rede primeiro processa a imagem inteira com várias camadas convolucionais e de *pool* máximo para produzir um mapa de atributos. Então, para cada objeto, uma camada de agrupamento da *Region of Interest* (ROI) extrai um vetor de características de comprimento fixo do mapa de atributos.

Notou-se que, para dar suporte às necessidades computacionais de operações paralelas com imagens, processadores com técnicas *multithreading* e múltiplos núcleos de processamento vem sendo consolidado como uma alternativa para melhorar o desempenho computacional. Para a nova geração de processadores com arquiteturas *many-core*, a quantidade elevada de núcleos demanda um novo tipo de projeto baseado em um sistema heterogêneo com diversos núcleos diferentes, memórias cache, redes de interconexões, protocolos de coerência adequados, a fim de obter um alto desempenho no sistema final [Alves et al. 2012].

Desta forma, o uso de arquiteturas computacionais, como sistemas compostos por CPU de múltiplos núcleos e *Graphics Processing Unit* (GPU), capazes de processar paralelamente operações em um conjunto de imagens, deve ser levado em consideração ao propor soluções envolvendo reconhecimento de padrões. No método desenvolvido, utilizou-se a biblioteca de computação numérica chamada TensorFlow, disponibilizada pela Google, a qual possui suporte a processamento paralelo em CPU com múltiplos cores e também GPU. Neste contexto, este trabalho visa avaliar o potencial de ambientes computacionais remotos - com unidade de processamento CPU e GPU - na execução das etapas de treinamento e classificação do método desenvolvido, dando ênfase em duas métricas: tempo de execução (tempo definido no início do treinamento e classificação de imagens em cada ambiente, até o fim deste) e acurácia (mede a qualidade de predição nos resultados da classificação de imagens).

2. Materiais e Métodos

O Google disponibiliza uma máquina virtual (VM) com Linux e ambiente de desenvolvimento Python já preparados para pesquisas voltadas para inteligência artificial com CPU e GPU, gratuita e em *cloud*, chamado Google Colab. Uma VM com ou sem GPU é acionada por demanda, permitindo avaliar a execução de um programa completo ou em blocos de código. A plataforma pode ser útil na aceleração de processamentos computacionais de projetos de aprendizado de máquina nos quais o fator tempo é determinante. Tem como base o ambiente web Jupyter Notebook, o qual permite acesso remoto sem qualquer tipo de instalação necessária e o compartilhamento de scripts de execução de qualquer linguagem interpretada, principalmente projetos Python, podendo ser mais rápidos que 20 cores físicos em aplicação de aprendizado profundo, eliminando a etapa de configuração de ambientes de execução para treinamentos naquela aplicação [Carneiro et al. 2018].

Foram definidos dois ambientes, tendo como configuração base um máquina virtual composta por um Intel Xeon CPU @ 2.30GHz 1 core, 2 CPUs, 2 Thread(s) por core, Ubuntu 18.04, Python 2, biblioteca TensorFlow 1.14, com Linux Kernel 4.14, 12 GB de memória RAM. O **Ambiente 1 (GPU)** continha uma GPU de 11 GB modelo NVIDIA Tesla K80 (chipset GK210) NVIDIA Cuda compilation tools, release 9.2, V9.2.148, enquanto que o **Ambiente 2 (CPU)** não dispunha de GPU.

Para a realização dos experimentos, foram adotados os seguintes métodos:

1. A execução da primeira etapa envolveu a coleta das imagens *Red Green Blue* (RGB) na área com animais a campo, com animais na posição em pé, por volta das 16 h, com plano de voo definido para 100 m de altitude, com sobreposição horizontal e vertical de 60%.
2. Neste passo, foi criado um ortomosaico (conjunto de imagens unidas através da sobreposição de imagens do voo tornando apenas uma imagem) com o intuito de diminuir a contagem duplicada de animais.
3. Dividiu-se o ortomosaico em imagens de 400 por 400 pixels (px) e aproveitou-se apenas imagens que contém animais (14), e a partir destas foram selecionados 48 animais e anotados com sua respectiva posição na imagem (*bounding box*). Estas anotações de animais foram definidas como conjunto de treinamento, representando 60% dos 80 animais encontrados no aerolevanteamento. O conjunto de imagens de teste foi dos 40% restantes.
4. O ortomosaico foram dividido em tamanhos de 200 px até 800 px quadrados, com intervalos de 100 px, com a intenção de diminuir o tamanho total da imagem e identificar as variações no desempenho na classificação e acurácia.
5. Dentro do Colab foi criado um script para importação de imagens de treinamento e classificação.
6. As imagens da etapa dois foram submetidas ao algoritmo *Fast Region Convolutional Neural Network* realizando um treinamento por meio de hiperparâmetros de variação de escala e proporção de 0.25 a 3.0, stride de tamanho 16, batch_size 1 dividido em épocas de 600 e 2000, e *threshold* mínimo de 80%, na configuração de GPU e CPU.

3. Resultados Experimentais

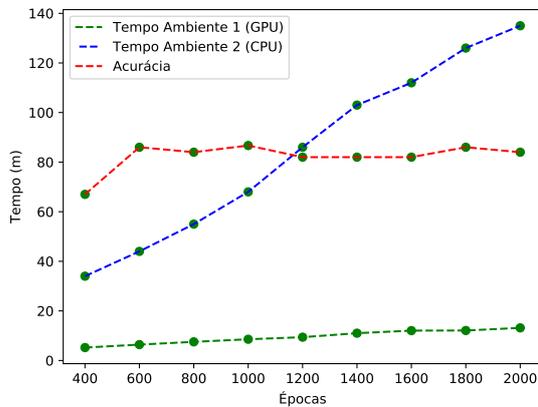
No Ambiente 1 (GPU), conforme mostra a Figura 1a com imagens tamanho 400px, que obteve melhor acurácia, 400 épocas o tempo de treinamento foi de 321 s. Com 2000 épocas o tempo de treinamento foi de 822 s. Aplicando média e variância populacional, variando entre as épocas (400 a 2000) e todas as classificações neste ambiente é possível perceber que a média fica em 572 s com uma variância de 0,31, demonstrando que não houve variação significativa na métrica tempo em relação a variação de tamanho das imagens. Como é possível verificar na Figura 1a, no Ambiente 2 (CPU) com imagens tamanho 400px, com 400 épocas o tempo de treinamento foi de 2048 s. Já com 2000 épocas o tempo de treinamento foi de 7232 s. Aplicando em todos os tempos de classificação e variação de épocas de 400 a 2000, chegou-se a média de 30,01 s e uma variação populacional de 252,17, demonstrando estatisticamente que existe uma relação entre tamanho e quantidade das imagens, e a métrica tempo neste ambiente. A acurácia variou de 67% para 86%. Notou-se que, com o aumento das épocas para valores maiores de 600, a acurácia obteve média de 84% com desvio padrão de 5,92.

A figura 1b demonstra que o tempo (m) de cada treinamento variou significativamente com o uso da GPU em relação ao da CPU. Principalmente nos tamanhos de imagem de 200 px, onde a quantidade de imagens foi maior em relação a outros tamanhos.

4. Considerações Finais

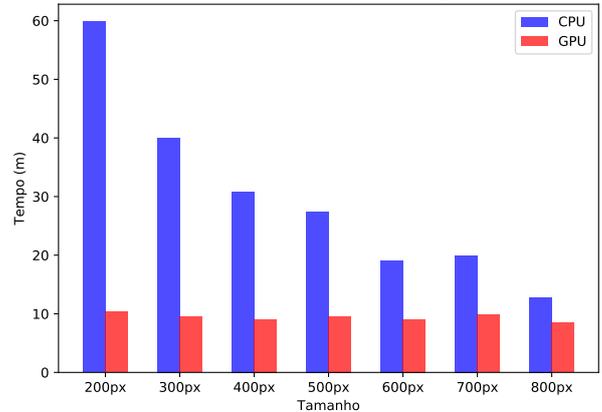
O Google Colab, como exemplo de ambiente para processamento remoto de imagens, foi capaz de suportar, em tempos suficientes, as tarefas de treinamento e classificação do

Relação entre tempo, acurácia e épocas ambiente 1 (GPU) e 2 (CPU)



(a) Comparação de desempenho com diferentes métricas.

CPU e GPU Classificação de Imagem



(b) Efeito do tamanho das imagens.

Figura 1. Gráficos comparativos.

método desenvolvido. No entanto, notou-se que há limitações no ambiente, com destaque para a destruição dos arquivos a cada 12 horas. Como forma de amenizar o problema, a ferramenta permite salvar scripts e comunicar-se com outras formas de armazenamento e transferir arquivos e refazer uma configuração anteriormente criada.

Contudo, salienta-se que o tempo obtido como o Ambiente 1 (GPU) demonstra o potencial do seu uso em uma versão evoluída do método de monitoramento, na qual parte das etapas possa ocorrer durante o voo de um VANT, já que o tempo mencionado seria capaz de suportar a demanda. Outra vantagem observada no Ambiente 1 é a oportunidade de usar a ferramenta na repetição de experimentos apenas mudando o conjunto imagens no treinamento, sem haver preocupação com modificações nas configurações de hardware.

Por fim, no que se refere à acurácia do método, mesmo que tenha alcançado menos de 90% neste trabalho (no qual dedica-se ao a avaliação do desempenho computacional no método proposto), cabe ressaltar que houve o aumento de 10% obtido com a ampliação de épocas, sem modificações do conjunto de imagens, foi possível apenas aumentando um parâmetro da rede. Objetivo que só foi alcançado com os recursos computacionais capazes de aplicar arquiteturas *many-core*, como o Ambiente 1. Isto sugere que o método aerolevante têm potencial de aplicação em áreas significativamente maiores que a adotada nos experimentos iniciais.

Referências

- Alves, M. A. Z., Freitas, H., and Navaux, P. (2012). *Ensino de arquiteturas de processadores many-core e memórias cache utilizando o simulador Simics*, pages 74–110.
- Barbalho, V. F., Nova, S. P. D. C. C., Pereira, A. C., and Oliveira, A. B. S. (2005). O controle de estoque de animais na pecuária bovina de corte: uma questão de continuidade. In *Anais do Congresso Brasileiro de Custos-ABC*.
- Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G.-B., De Albuquerque, V. H. C., and Rebouças Filho, P. P. (2018). Performance analysis of google colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, 6:61677–61685.