

# Paper title - English

## Título do artigo - somente para documento escrito em Português

BRAGA, S. J.\*; GOMI, E. S.\*

\*Escola Politécnica da Universidade de São Paulo

**Abstract**—Abstract here.

**Keywords**— word 1; word 2.

**Resumo**— É necessária a inserção do resumo para artigo escrito em Português.

**Palavras-chave**— palavra 1; palavra 2.

**Classificação**— Mestrado

**Categoria**— Iniciante

### I. INTRODUÇÃO

#### II. DIAGNÓSTICO DE GLAUCOMA

#### III. REDES NEURAI PROFUNDAS

##### A. Transfer Learning

##### B. Redes pré-treinadas

#### IV. EXPERIMENTOS E RESULTADOS

Os experimentos foram conduzidos em um servidor com Ubuntu 16.04 com duas GPUs NVidia Quadro K5200 com 8GB de memória cada, 24 CPUs Intel Xeon e 128GB de memória RAM. O framework de deep learning utilizado em todos os experimentos foi o Caffe, fornecido pela universidade de Berkeley [1].

O objetivo dos experimentos, com e sem transfer learning, é a classificação binária de olhos normais ou com glaucoma em imagens de OCT. A performance das redes foram avaliadas utilizando um conjunto menor de imagens não vistas para classificação e cálculo das métricas de avaliação. O mesmo conjunto de imagens é utilizado em todos os experimentos. O dataset, pré-processamento e resultados são descritos em detalhes nas próximas seções.

##### A. Dataset

O dataset original foi obtido com o departamento de oftalmologia da Unicamp. O dataset consiste de imagens de OCT com tamanho 136x136 de 56 olhos com glaucoma e 66 olhos normais, totalizando 122 pacientes. Os gráficos de espessura de fibras nervosas foram obtidos através da extração das imagens do PDF do exame. Foram selecionados para o experimento somente os olhos de pacientes que foram manualmente classificados por especialistas.

Para a separação do dataset em treino e validação, foram selecionados 20% de olhos normais e 20% de olhos com glaucoma para validação, e o restante para treino, totalizando

98 imagens de treino e 24 para validação. As imagens selecionadas para teste não estão presentes no dataset de treino, para que o algoritmo possa classificar imagens ainda não vistas.

Para evitar overfitting, foi empregada uma técnica para aumentar o número de exemplos a partir das imagens no dataset de treino. Cada imagem foi rotacionada 100 vezes em ângulos aleatórios entre 0 e 360 graus, gerando assim um dataset de treino com 9800 imagens. As imagens de validação não foram rotacionadas.

##### B. Pré-processamento

Para utilização do transfer learning, foi necessário fazer a subtração do pixel médio em todas as imagens do dataset de treino. O valor médio de cada pixel da imagem é calculado sobre todas as imagens do dataset de treino. Essa imagem média é então subtraída de cada imagem do dataset. Dessa forma, todos os pixels de entrada estão na mesma ordem de grandeza, evitando que os gradientes desapareçam ou explodam.

Onde houveram falhas na aquisição da imagem, gerando áreas escuras, pixels com valores RGB próximos ao preto foram substituídos pelo valor de preto absoluto RGB (0, 0, 0) para que não tenham influência sobre a decisão do classificador.

##### C. Resultados com transfer learning

Neste experimento, utilizamos a mesma arquitetura da rede VGG16, alterando a saída da última camada totalmente conectada para duas saídas, correspondente às duas classes a serem classificadas: normal e glaucoma. A camada de entrada da rede também foi alterada para a resolução das imagens do nosso dataset. As imagens do dataset Imagenet com que a rede foi treinada tinham a resolução de 224x224 pixels, enquanto que as imagens de OCT têm resolução de 136x136 pixels.

Os pesos pré-treinados foram carregados para inicialização apenas das camadas convolucionais. As três últimas camadas totalmente conectadas foram iniciadas com valores aleatórios devido à diferença de resolução entre as imagens do dataset Imagenet e as imagens de OCT classificadas nesse experimento. Essa diferença gera quantidades de parâmetros diferentes na saída da última camada convolucional. Sendo assim, as camadas totalmente conectadas foram inicializadas com valores aleatórios de uma distribuição normal com desvio padrão 0.01.

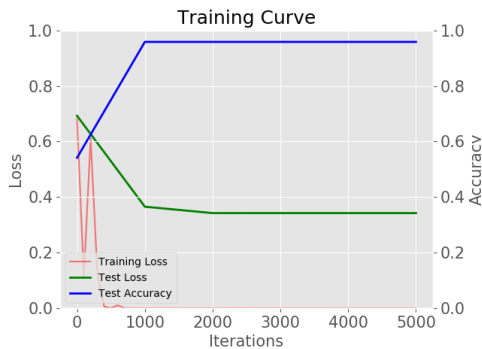


Figura 1. Acurácia e perda de treino e validação da rede VGG16 com transfer learning.

O treinamento foi realizado em todas as camadas da rede, utilizando o gradiente descendente estocástico por 5000 iterações, com mini batches de 15 imagens. Os parâmetros de momentum e weight decay foram definidos como 0.9 e 0.0005, respectivamente. A taxa de aprendizagem inicial foi de 0.001. A cada 1000 iterações a taxa de aprendizagem foi diminuída utilizando a equação 1.

$$base\_lr * \gamma^{\lfloor \frac{iter}{step} \rfloor} \quad (1)$$

Onde  $base\_lr$  é a taxa de aprendizagem inicial,  $\gamma$  é um parâmetro do Caffe definido com o valor 0.1,  $iter$  é o número da iteração atual e  $step$  é um parâmetro definido como 1000.

A validação do modelo foi feita utilizando um dataset de 24 imagens ainda não vistas pelo algoritmo. Foi obtida acurácia final de 95.8%, com sensibilidade de 100% e especificidade de 92.3%. A figura 1 mostra a evolução dos valores de perda e acurácia durante o processo de treinamento da rede. A validação foi feita a cada 1000 iterações. É possível identificar a estabilização da acurácia após 1000 iterações, quando o valor de perda do treinamento chega próximo à zero.

#### D. Resultados sem transfer learning

#### V. DISCUSSÃO

#### VI. CONCLUSÃO

#### REFERÊNCIAS

- [1] JIA, Y. et al. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.