

Computação de Borda versus Computação em Nuvem: Impacto do Pré-processamento de Imagens de Retinas

Cristiano A. Künas, Dayla R. Pinto, Philippe O. A. Navaux, Lisandro Z. Granville

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre, RS – Brasil

{cakunas, dayla.rpinto, navaux, granville}@inf.ufrgs.br

Resumo. *Na área da saúde, a prevenção é uma forma eficaz de evitar a progressão de doenças, muitas das quais podem ser tratadas quando diagnosticadas precocemente. A procura por exames preventivos tem aumentado e não se consegue atender essa procura com eficiência. Logo, existe a necessidade de automatizar e aumentar a eficiência de exames de triagem. Entretanto, a captura de dados para estes sistemas geralmente utiliza vários dispositivos de hardware sob condições ambientais diversas, induzindo ruído nos dados. Portanto, antes da fase de triagem, a seleção de uma estrutura de pré-processamento eficaz é fundamental. Neste artigo, é discutido o desenvolvimento de uma aplicação para pré-processamento de imagens de retinas para uso eficiente em sistemas de triagem e o impacto que o pré-processamento causa na interconexão de rede. Foi reduzido em até $\approx 73\%$ o tempo de execução com a versão paralela. Também foi reduzido em $\approx 11,5\times$ a largura de banda utilizada, alcançando taxa de transferência acima de 5 imagens/segundo com pré-processamento na Borda, $2,57\times$ maior do que na Nuvem.*

1. Introdução

A Retinopatia Diabética (RD) é uma doença progressiva e complicação vascular retiniana primária do diabetes mellitus. Em 2019, foi relatado que 463 milhões de pessoas tinham diabetes em todo o mundo, e esse número deve chegar a 693 milhões até 2045. A prevalência global de diabetes em 2019 é estimada em 9,3% (463 milhões de pessoas), aumentando para 10,2% (578 milhões) em 2030 e 10,9% (700 milhões) em 2045 [Wang et al. 2020]. Em 2020, uma meta-análise global de um total de 12.620 pacientes com diabetes mostrou que a prevalência de RD atingiu 35,36%, e a visão de 11,72% dos pacientes foi seriamente afetada [Saeedi et al. 2019].

As complicações da RD são muitas vezes evitáveis se detectadas e tratadas precocemente. Na prática, as características clínicas dessas complicações são evidentes no fundo do olho em exames oftalmológicos [Lechner et al. 2017]. Os avanços tecnológicos têm possibilitado o desenvolvimento de algoritmos para detecção automatizada de RD e edema macular diabético em imagens de fundo de retina. A captura dessas imagens geralmente utiliza uma variedade de dispositivos de hardware, sob uma variedade de condições ambientais, induzindo ruído na imagem final. Para reduzir essa heterogeneidade, que acaba afetando o desempenho do diagnóstico, bem como destacar alguns detalhes das imagens, o pré-processamento das imagens passa a ser uma etapa necessária. Do ponto de vista clínico, cerca de 20% das imagens da retina não são utilizadas devido à baixa

qualidade das imagens. Portanto, antes da triagem de RD, a seleção de um esquema de pré-processamento eficaz é obrigatória [Chatterjee et al. 2021].

Com o surgimento de novos aplicativos inteligentes, ocorreu uma revolução na gestão da informação, sobretudo incluindo rotinas de processamento, armazenamento e capacidade de computação. Na última década, assistimos a uma evolução significativa dos paradigmas de computação. A computação em nuvem é um dos paradigmas mais conhecidos e popular, e a sua adoção continua a acelerar à medida que as empresas buscam agilidade, flexibilidade e novas fontes de vantagem competitiva. Segundo a Gartner¹, até 2023 40% de todas as cargas de trabalho corporativas serão implantadas em infraestrutura de nuvem e serviços de plataforma, acima de 20% comparado com 2020. Evitá-la é quase inconcebível, mas ainda existem algumas barreiras à adoção da nuvem e a disseminação esmagadora de dispositivos e aparelhos inteligentes, como a Internet das Coisas (*Internet of Things* - IoT) apontou todas as limitações de um paradigma tão centralizado.

A computação em borda serve como extensão descentralizada e provê soluções que facilitam o processamento de dados na fonte de geração ou mais próximos dos criadores e usuários de dados, resolvendo problemas induzidos pela centralização, tais como otimização dos requisitos de latência, economia de banda e autonomia. Para sistemas de inferência, por exemplo, dispositivos de borda capturam dados e os enviam para a nuvem para processamento, mas a transferência de dados do dispositivo de borda para a nuvem pode levar muito tempo se o volume de dados for muito grande. Portanto, faz sentido processar os dados capturados localmente no nó de borda para minimizar o tempo de transferência na rede.

Este artigo tem como objetivo mostrar que o pré-processamento de dados pode ser realizado no nó de borda, e que também pode ser acelerado através de técnicas de paralelismo. Foi avaliado um método simples de pré-processamento de imagens de retina (localizando o centro e redimensionando) e verificado seu desempenho em dois ambientes: Nuvem e Borda.

As demais seções deste artigo estão organizadas da seguinte forma. A Seção 2, apresenta uma visão geral sobre ferramentas de análise de rede, da RD e de pré-processamento de imagens de retina. Na Seção 3, é discutido alguns trabalhos relacionados. A metodologia de implementação e as arquiteturas são descritas na Seção 4. A Seção 5, apresenta os resultados da avaliação abrangendo desempenho de execução e taxa de transferência de imagens em versões sequenciais e paralelas. Por fim, na Seção 6, são apresentadas as conclusões e os trabalhos futuros.

2. Background

Nesta seção, são apresentados conceitos sobre ferramentas para análise de tráfego de rede, da Retinopatia Diabética (RD) e técnicas de pré-processamento de imagens de retina.

2.1. Análise de tráfego de rede

A análise de tráfego de rede tem como objetivo capturar informações úteis e material relevante para o gerenciamento da rede. Em geral, a análise de tráfego de rede pode ser definida como a inferência de informações a partir da observação do fluxo de dados no

¹<https://www.gartner.com/>

tráfego da rede. A análise de rede é categorizada por tempo (ou frequência), por critérios e pelos objetivos da análise, assim podendo ser classificada em: análise em tempo real, análise em lote ou análise forense [Asrodia and Patel 2012].

Uma ferramenta popular para a captura do tráfego de redes são os *sniffers*, isto é, programas de computador que observam o tráfego de entrada e saída nos dispositivos de rede. O *sniffer* captura os pacotes endereçados a outras máquinas, salvando-os para análise posterior (ou, mais raramente, inline). *Sniffers* são legitimamente utilizados por administradores de rede ou sistema para monitorar e apoiar na solução de problemas de rede [Ying-hua et al. 2011]. Embora existam centenas de *sniffers* no mercado, o TCPdump e o Wireshark são as ferramentas mais usadas e apreciadas [Goyal and Goyal 2017].

O TCPdump é uma ferramenta de linha de comando usada para monitoramento de rede, captura de pacotes, e depuração de protocolos. O TPCdump é gratuito e de código aberto, funcionando apenas em sistemas baseados em Linux [Solomon et al. 2016]. Normalmente, o TCPdump é utilizado para capturar pacotes de dados transferidos pela rede nos seguintes casos: projetar redes/protocolos, verificar se serviços de rede estão funcionando corretamente, monitorar e fazer estatísticas com base no tráfego [Solomon et al. 2016]. Para esta finalidade, o TCPdump fornece muitas opções, onde os detalhes dos pacotes capturados podem ser explorados e visualizados em vários formatos.

O Wireshark, por sua vez, é um analisador de protocolos de rede de código aberto, utilizado para coletar, solucionar problemas e ajudar administradores de redes, permitindo que os dados de uma rede sejam rastreados em tempo real e armazenados para posterior análise [Luo et al. 2010]. O Wireshark é credenciado pela sua interface gráfica simples, suas poderosas opções de captura e filtragem, além de suportar as plataformas Linux e Windows. Ele é também capaz de escanear redes *Ethernet*, Wi-Fi ou até mesmo *Bluetooth* [Goyal and Goyal 2017].

2.2. Retinopatia Diabética

O *Diabetes Mellitus* é uma doença metabólica caracterizada por um aumento anormal de açúcar ou glicose no sangue. Quando não tratada adequadamente, o paciente estará sujeito a complicações tais como ataque cardíaco, derrame cerebral, insuficiência renal, lesões de difícil cicatrização e problemas na visão [Pires and Rocha 2011]. Os problemas na visão ocorrem porque o diabetes afeta o sistema circulatório, incluindo rupturas vasculares progressivas causadas por hiperglicemia crônica, podendo ser desenvolvida independente da gravidade do paciente, provocando o que se chama de retinopatia diabética (RD) [Janghorbani et al. 2000].

A RD é uma complicação decorrente da condição diabética prolongada, surgindo depois de dez a quinze anos [Lee et al. 2001]. A hiperglicemia causa danos aos minúsculos vasos sanguíneos dentro da retina [Vandarkuzhali et al. 2013]. Esses vasos sanguíneos vazam sangue e fluido na retina, ocasionando problemas como microaneurismas, hemorragias, anormalidades microvasculares intrarretinianas e exsudatos [Alzami et al. 2019]. A Figura 1 ilustra retinas com e sem RD.

A RD pode ser clinicamente dividida em dois estágios: retinopatia diabética proliferativa (RDP) e retinopatia diabética não proliferativa (RDNP) [Stitt et al. 2016]. A RDP é uma forma mais avançada de RD marcada pela proliferação de tecido fibrovascular levando a hemorragia vítrea e descolamento de retina [Wong et al. 2006]. Também

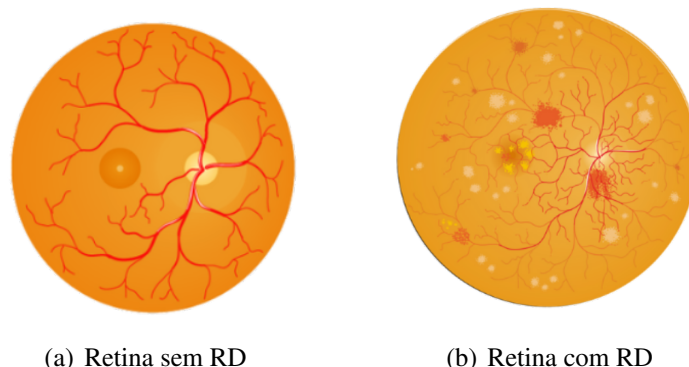


Figura 1. Ilustração de retinas com e sem Retinopatia Diabética: Figura 1(a) apresenta a retina sem RD; Figura 1(b) apresenta a retina com RD, com extravasamentos de sangue e plasma [Servier 2022].

é caracterizada pela proliferação de vasos retinianos, cujo crescimento é variável. Elas são comumente identificadas de acordo com sua localização na retina, no disco óptico ou próximo, sendo mais propensas a proliferar na superfície posterior do vítreo e hemorragia no vítreo. Ainda, com o tempo, muitas vezes novos vasos podem se contrair, resultando em descolamento da retina [Waheed 2018]. A RDNP é um estágio inicial da RD caracterizada por hemorragias retinianas e microaneurismas [Wong et al. 2006].

Em relação aos protocolos de classificação clínica da RDP e RDNP, o *Global Diabetic Retinopathy Project Group* [Wilkinson et al. 2003] propôs uma escala de gravidade da doença em cinco níveis de gravidade: I) **Sem retinopatia aparente**: sem anormalidades; II) **Retinopatia diabética não proliferativa leve**: presença de microaneurismas na retina; III) **Retinopatia diabética não proliferativa moderada**: mais do que apenas microaneurismas, mas menos do que retinopatia diabética não proliferativa grave; IV) **Retinopatia diabética não proliferativa grave**: mais de 20 hemorragias intra-retinianas em cada um dos quatro quadrantes, peroladas venosas em pelo menos dois quadrantes e anormalidades microvasculares intra-retinianas em pelo menos um quadrante na ausência de RDP; e V) **Retinopatia diabética proliferativa**: neovascularização, hemorragia vítrea/pré-retiniana.

Para realizar a triagem do nível de gravidade da RD recomenda-se exames oftalmológicos regulares para as pessoas com *diabetes mellitus*, pois o diagnóstico oportuno e o manejo subsequente da condição são essenciais para estabelecer o tratamento precoce [Network 2010], visto que a RD pode se desenvolver e progredir para estágios avançados sem produzir nenhum sintoma imediato ao paciente, estabelecendo riscos, como a perda de visão [Stitt et al. 2016].

2.3. Pré-processamento de imagens de retinas

A etapa de pré-processamento de imagens é feita para produzir uma imagem com melhor qualidade. A maioria das imagens retinianas não são uniformemente iluminadas e, às vezes, apresentam baixo contraste visual e ruído, dificultando a detecção de lesões [Jaafar et al. 2011]. Isso ocorre por conta de erro do ângulo do dispositivo de captura, movimento dos olhos, entre outros, e tornam a imagem do fundo inadequada para o diagnóstico da doença. Assim, a seleção do método de pré-processamento apropriado para

a imagem do fundo da retina é uma etapa obrigatória [Chatterjee et al. 2021].

Uma das principais etapas do pré-processamento é a remoção dos ruídos, que são categorizados como: ruído de sal e pimenta (ocorre aleatoriedade do pixel preto e branco); ruído gaussiano (ocorre variação do valor da intensidade com a distribuição normal de *gaussian*); e ruído de cintilação (contém pixels brancos aleatórios) [Swathi et al. 2017]. Outras etapas importantes são o aprimoramento de contraste, correção de tonalidade e redimensionamento [Chatterjee et al. 2021].

Os filtros utilizados no pré-processamento de imagens podem ser feitos por métodos lineares e não lineares. No método linear, o algoritmo aplica o filtro linearmente a todos os pixels sem definir a imagem corrompida ou não corrompida. O algoritmo de método não linear, por sua vez, aplica o filtro definindo qual pixel está corrompido ou não corrompido. O filtro não linear produz melhor resultado em comparação com o filtro linear [Swathi et al. 2017].

3. Trabalhos relacionados

O pré-processamento de imagens costuma ser uma etapa necessária na maioria dos estudos revisados. Os trabalhos apresentados nesta seção se relacionam com este artigo por tratarem de soluções para essa problemática.

O trabalho de Gulshan *et al.* [Gulshan et al. 2016] teve como objetivo desenvolver um algoritmo para detecção automatizada de RD e edema macular diabético em fotografias de fundo de retina. Foram utilizadas imagens obtidas no EyePACS nos Estados Unidos e em três hospitais de olhos na Índia: Aravind Eye Hospital, Sankara Nethralaya e Narayana Nethralaya. Todas as imagens foram classificadas de 3 a 7 vezes por 54 oftalmologistas licenciados nos EUA e residentes seniores de oftalmologia. As classificações foram quanto à presença de RB, edema macular diabético e qualidade de imagem utilizando rubrica. 128.175 imagens da retina com qualidade excelente, boa e adequada foram consideradas graduáveis, assim utilizadas no treinamento da rede neural. O algoritmo resultante teve alta sensibilidade e especificidade para detectar retinopatia diabética.

O trabalho de Voets *et al.* [Voets et al. 2019] replica o artigo científico de Gulshan *et al.*, no qual, o código-fonte não foi publicado, apenas descrito. O objetivo, assim como no estudo original, foi criado por meio de aprendizado profundo, que envolve um procedimento de treinamento de uma rede neural para realizar a tarefa de classificação de imagens. Para reimplementar o algoritmo original para detecção de RDR, os autores utilizaram imagens semelhantes de um conjunto de dados EyePACS disponível publicamente no Kaggle, e outra distribuição do conjunto de dados Messidor-2, pois o conjunto de dados original não estava mais disponível. Avaliaram a gradabilidade das imagens por conta própria; as imagens consideradas graduadas foram pré-processadas conforme descrito pelo protocolo do estudo original para pré-processamento e utilizadas no treinamento da rede neural. Na etapa de pré-processamento o centro e o raio de cada imagem foram redimensionados para altura e largura de 299 pixels, com o centro do fundo no meio da imagem. Como resultado, não foi obtido o mesmo desempenho do estudo original. Por ter utilizado imagens disponíveis publicamente, o estudo original teve acesso a dados de qualidade superior aos que estão disponíveis publicamente, e isso provavelmente explica parte do desvio nos resultados.

O trabalho de Chatterjee *et al.* [Chatterjee et al. 2021] teve como objetivo apontar

um esquema de pré-processamento de retina mais adequado na aplicação de triagem de retinopatia diabética. Para a metodologia utilizada no trabalho, os autores inicialmente revisaram os métodos mais utilizados para pré-processamento de imagens médicas, a saber, primordialmente, imagem de fundo de retina. Descobriram que a equalização de histograma (HE) e a equalização de histograma adaptável limitada ao contraste (CLAHE) são os métodos de pré-processamento mais renomados para este fim. Para fazer a comparação dos mais usuais, escolheram também um esquema de pré-processamento de preservação de brilho proposto por Datta *et al.* (2016), com uma revisão nesta área específica, observa-se que o pré-processamento de imagem da retina deve ter capacidade de preservação de brilho. Uma próxima etapa da metodologia foi a coleta de imagens do Departamento de Oftalmologia de Sri Aurobindo Seva Kendra, do Kolkata e do Multi-specialty Hospital. Todas as imagens foram capturadas pela mesma configuração de câmera. Em seguida, foram criados três bancos de dados das imagens de retinas coletadas com diferentes resoluções de imagem. Dos resultados obtidos através da análise e medição do índice de similaridade de estrutura (SSIM) e erro de brilho médio absoluto (AMBE), que significa se o SSIM for 1, a imagem retiniana de saída é original e mais próxima de 1 indica a melhor qualidade das imagens retinianas, e se AMBE mais baixo e SSIM mais alto refletem o melhor.

4. Metodologia

A infraestrutura da arquitetura de teste da aplicação pode ser descrita como uma composição de três camadas, como está representado na Figura 2: (1) Camada de nuvem, que executa o modelo de detecção de RD treinado para inferência; (2) Camada de armazenamento, composta por um bucket no Google Cloud Storage que é utilizado para armazenar os dados enviados para a nuvem; e (3) Camada de borda, composta por máquina local que executa tarefas de pré-processamento e envio para a nuvem.

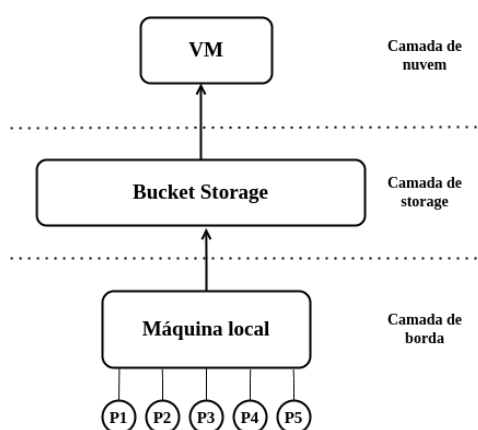


Figura 2. A arquitetura é composta por 3 camadas: Nuvem, Storage e Borda

A implementação da aplicação é feita em Python e executada em Shell Script. A aplicação é dividida em dois módulos. O primeiro módulo executa todo o pré-processamento dos dados na camada de borda e depois envia para a camada de storage na Nuvem. Diferentemente, o segundo módulo faz o oposto, primeiramente enviando os dados originais – sem qualquer pré-processamento – para a camada de storage, e posteriormente a camada de Nuvem se responsabiliza pelo pré-processamento

dos dados. Em ambos os módulos, avaliamos tanto o pré-processamento quanto o envio dos dados, de maneira sequencial e paralela através de processos usando a classe `ProcessPoolExecutor()` do Python. Na aplicação paralela, cada processo recebe uma parcela dos dados e executa funções determinadas. Por exemplo, no primeiro módulo, a função principal carrega uma imagem do armazenamento local, pré-processa (como sugere o estudo original [Gulshan et al. 2016]) localizando o centro e o raio do fundo do olho e redimensionando-a para uma altura e largura de 299 pixels, para então fazer *upload* para o *bucket* no Google Storage. O algoritmo de pré-processamento está disponível no repositório complementar em gitlab.com/cristianokunas/wscad-2022.

O conjunto de dados utilizado nos experimentos é o *APTOS 2019 Blindness Detection* disponível na competição Kaggle para retinopatia diabética². Este foi escolhido por possuir um tamanho razoável em comparação à outros citados anteriormente, facilitando nossos experimentos. Esta base de dados é comumente utilizada para aplicações de aprendizado de máquina na detecção de RD, e está dividida em dois subconjuntos. Neste trabalho, selecionamos o subconjunto de treinamento, contendo 3.662 imagens de retina. As imagens estão no formato PNG (Portable Network Graphics) e têm tamanhos variados (Figura 6).

O ambiente de nuvem é composto por uma máquina virtual tipo N1 Standard (16 núcleos, 60 GB de Memória RAM), sistema operacional Linux Ubuntu 20.04 LTS, localizado no leste da América do Sul. O ambiente de armazenamento faz uso de um *bucket* do Google Cloud Storage fornecido através de uma instância do Firebase. Este Storage é uma solução de armazenamento de objetos com grande capacidade, além de alta disponibilidade e redundância. O Cloud Storage para Firebase permite a manipulação dos arquivos de maneira segura e com facilidade por meio de seu SDK.

O ambiente de execução na borda é composto por um equipamento com um processador *Intel Core i7-9750* com 6 núcleos físicos (2,60 GHz). Este equipamento possui 16 GB de Memória RAM DDR4, GPU NVIDIA GeForce GTX RTX 2060 com 6GB. Utilizou-se o sistema operacional *Linux Ubuntu 20.04.4 LTS* com versão de *kernel 5.13.0-41*. A versão do NVIDIA CUDA *Compiler* utilizada foi a 11.4.100.

5. Resultados

Nesta seção, apresentamos a avaliação de desempenho alcançado por nossa aplicação na plataforma experimental apresentada na Seção anterior. Apresentamos métricas de tempo de execução e taxa de transferência de imagens em versões sequenciais e paralelas. Os resultados apresentados nesta Seção são a média de pelo menos 10 execuções. O erro relativo foi menor que 5% usando uma confiança estatística de 95% pela distribuição t de Student.

5.1. Avaliação de rede

Para avaliar a implementação, primeiro avaliamos a conexão de rede. Esta avaliação mediu a quantidade máxima de dados que poderiam ser enviados do nó de borda para o provedor de nuvem. As medições iniciaram com a análise da taxa de transferência por

²<https://www.kaggle.com/c/aptos2019-blindness-detection/data>

meio da ferramenta Iperf [Tirumala et al. 2005]. Os resultados obtidos deste primeiro passo estão descritos na Tabela 1.

Tabela 1. Medições de rede com a ferramenta Iperf.

Parâmetro	Borda p/ Cloud
Intervalo	60 segundos
Total transferido	406 MBytes
Largura de banda	56.8 Mbits/segundo

A Figura 3 apresenta a largura de banda e a quantidade de bytes transferidos a cada segundo. Comparando com a medição realizada com a ferramenta *iperf3*, é perceptível que ao enviar os dados originais utilizamos toda a largura de banda disponível, com média de ≈ 66 Mbps (Figura 3(a)), que pode sobrecarregar a rede. O oposto ocorre ao enviar os dados pré-processados, na qual reduzimos em cerca de $\approx 11,5\times$ a largura de banda utilizada, com média de $\approx 5,7$ Mbps (Figura 3(b)). Também observamos uma variação nos picos mínimos e máximos de transferência para os dados originais (Figura 3(a)). Isso ocorre devido às diferenças de tamanho entre os arquivos.

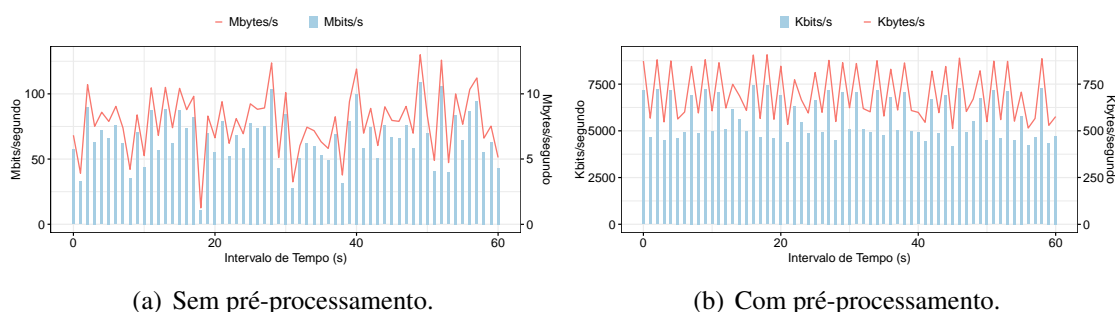


Figura 3. Largura de banda utilizada e total transferido.

5.2. Avaliação de desempenho

A aplicação possui dois módulos: um que realiza o pré-processamento na Borda e envia para o *storage* na Nuvem e outro que envia os dados originais para o *storage* e os processa na Nuvem. Portanto, comparamos o tempo de **execução sequencial** entre os ambientes de Borda e de Nuvem, e também comparamos o tempo de **execução paralela** entre os ambientes de Borda e de Nuvem. Conforme explicado na metodologia, a versão paralela divide as tarefas entre processos. Para ambas as execuções fixamos em 5 processos.

As execuções paralelas na Borda e na Nuvem apresentam um ganho de $\approx 71,12\%$ e $\approx 73,01\%$ respectivamente, quando comparadas com a versão sequencial. Já o tempo médio de execução da aplicação sequencial na Borda foi de $\approx 2.318,03s$, um ganho de $\approx 63.63\%$ em comparação com a mesma versão executada na Nuvem. Na versão paralela, o tempo médio de execução na Borda foi de $\approx 669,52s$, que representa um ganho de $\approx 61,07\%$ em relação a mesma versão executada na Nuvem. Como podemos observar, as execuções na Borda apresentam melhor desempenho do que na Nuvem porque não sobrecarregam a rede. Com o pré-processamento dos dados na Borda, além de diminuir o

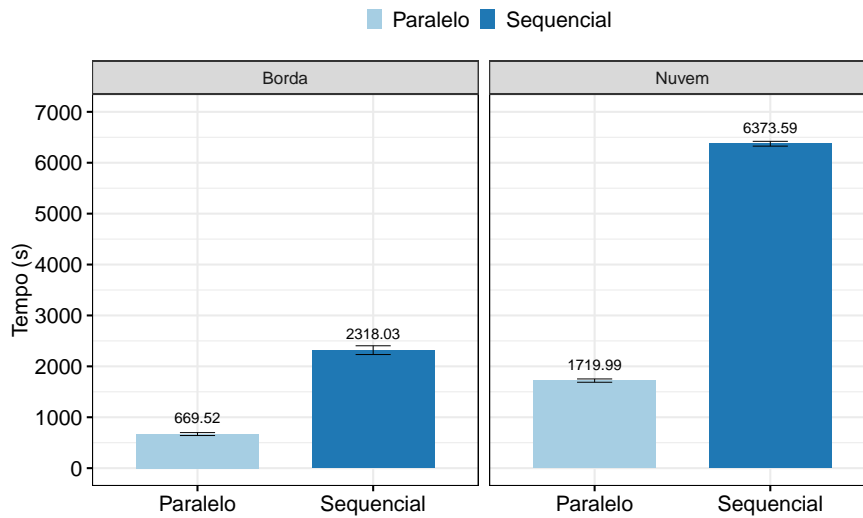


Figura 4. Tempos de execução de versões sequenciais e paralelas: comparativo entre execução na Borda e na Nuvem.

tamanho total dos dados em $\approx 44 \times$ (diminuindo de $\approx 8.204,5$ MB para $\approx 185,7$ MB), também economizamos banda para a transferência desses dados.

Apresentamos a taxa de transferência da aplicação na Figura 5. Essa taxa representa a quantidade de imagens pré-processadas e transferidas a cada segundo. Percebe-se que no modelo de execução sequencial na nuvem, onde a primeira etapa é o envio dos dados originais para a nuvem, a taxa de transferência é demasiadamente baixa. Isso ocorre porque o conjunto de dados apresenta uma grande variação nos tamanhos das imagens originais. Na Figura 6(a) demonstramos essa variação, com tamanhos variando de ≈ 200 KB até ≈ 7.500 KB. Esta disparidade de valores impacta significativamente no tempo de execução e no consumo de banda.

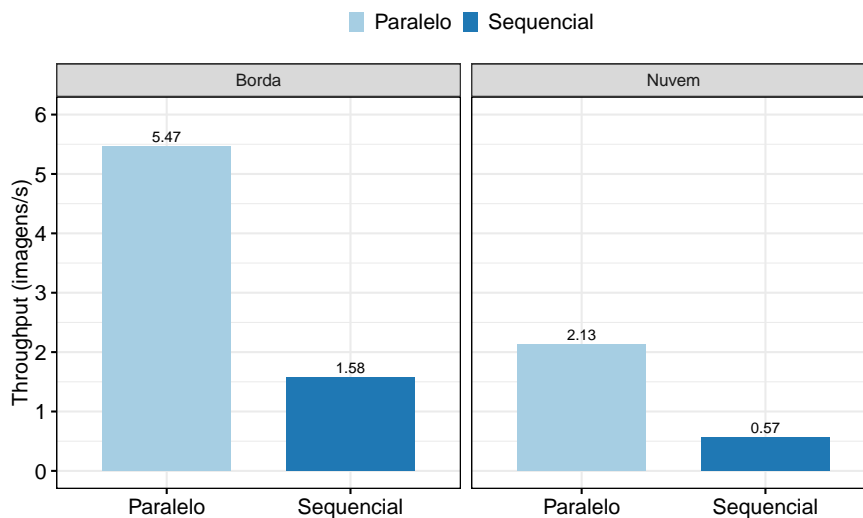


Figura 5. Taxa de transferência de imagens/segundo, em execuções sequenciais e paralelas, em ambientes de Borda e de Nuvem.

O melhor resultado em termos de taxa de transferência é obtido na versão paralela executada na Borda, alcançando ≈ 5 imagens/segundo. Isso ocorre porque a etapa de pré-processamento nos permite normalizar os dados, que além de reduzir o tamanho de cada imagem, também diminui a variação de tamanhos observada nos dados originais (30 - 70 KB), como mostra a Figura 6(b).

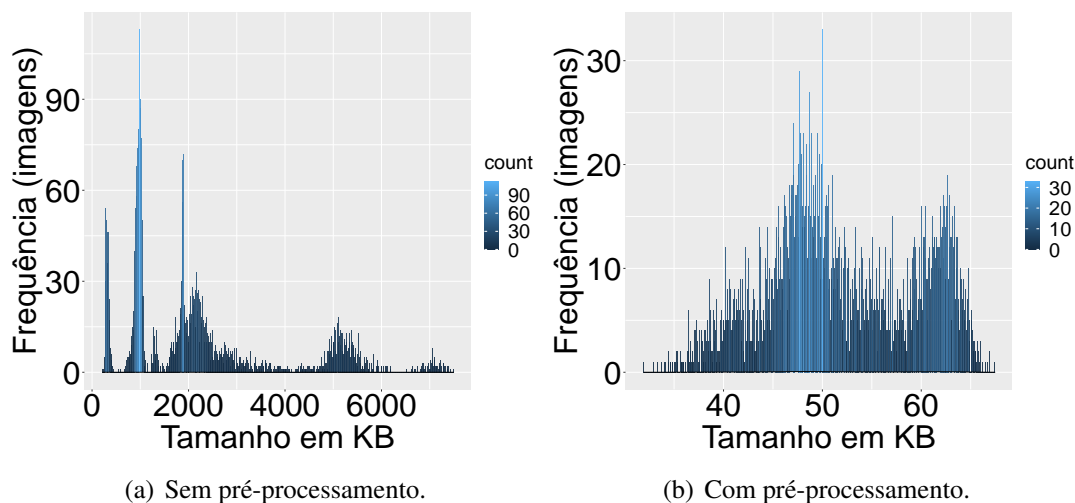


Figura 6. Histograma de frequência de imagens por tamanho em KB sem e com pré-processamento.

6. Conclusões e Trabalhos Futuros

Neste artigo, apresentamos uma aplicação para pré-processamento de imagens de retinas para uso eficiente em sistemas de triagem e o impacto que o pré-processamento causa na rede. Avaliamos um método simples de pré-processamento de imagens de retina e verificamos seu desempenho, em versões sequenciais e paralelas, executando em ambientes de Nuvem e de Borda.

Com pré-processamento na Borda, reduzimos em $\approx 72\%$ o tempo de execução e $\approx 11,7\times$ a largura de banda utilizada, alcançando taxa de transferência acima de 5 imagens/segundo no melhor caso, $2,52\times$ maior do que na Nuvem. Isso se deve à redução no tamanho total dos dados, $\approx 44\times$ menor do que os dados originais.

Nossos experimentos mostraram que o tamanho dos dados pode influenciar negativamente no tempo de execução da aplicação. Neste contexto, estratégias de paralelismo se tornam importantes. A interconexão de rede também pode ser um desafio, uma vez que os dados originais muitas vezes possuem tamanhos elevados, o que congestiona a conexão. Deste modo, o pré-processamento de dados na Borda pode contribuir para a economia de banda.

Trabalhos futuros estenderão a avaliação de desempenho com técnicas de pré-processamento mais sofisticadas. Além disso, planejamos estender o estudo avaliando conjuntos de dados maiores e de maior resolução.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, pelo edi-

tal CNPq/MCTI/FNDCT - Universal 18/2021 sob número 406182/2021-3, pelo projeto GREEN-CLOUD: Computação em Cloud com Computação Sustentável (2016/2551-0000 488-9), da FAPERGS e CNPq, programa PRONEX 12/2014 e pelo projeto CIARS RITEs/FAPERGS.

Referências

- Alzami, F., Megantara, R. A., Fanani, A. Z., et al. (2019). Diabetic retinopathy grade classification based on fractal analysis and random forest. In *2019 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pages 272–276. IEEE.
- Asrodia, P. and Patel, H. (2012). Network traffic analysis using packet sniffer. *International journal of engineering research and applications*, 2(3):854–856.
- Chatterjee, A., Datta, N. S., Dutta, H. S., Majumder, K., and Chatterjee, S. (2021). A study on retinal image preprocessing methods for the automated diabetic retinopathy screening operation. *Applications of Artificial Intelligence and Machine Learning*, pages 375–384.
- Goyal, P. and Goyal, A. (2017). Comparative study of two most popular packet sniffing tools-tcpdump and wireshark. In *2017 9th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 77–81. IEEE.
- Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., et al. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410.
- Jaafar, H. F., Nandi, A. K., and Al-Nuaimy, W. (2011). Detection of exudates from digital fundus images using a region-based segmentation technique. In *2011 19th European signal processing conference*, pages 1020–1024. IEEE.
- Janghorbani, M., Jones, R. B., and Allison, S. P. (2000). Incidence of and risk factors for proliferative retinopathy and its association with blindness among diabetes clinic attenders. *Ophthalmic Epidemiology*, 7(4):225–241.
- Lechner, J., O’Leary, O. E., and Stitt, A. W. (2017). The pathology associated with diabetic retinopathy. *Vision research*, 139:7–14.
- Lee, S. J., McCarty, C. A., Taylor, H. R., and Keeffe, J. E. (2001). Costs of mobile screening for diabetic retinopathy: a practical framework for rural populations. *Australian Journal of Rural Health*, 9(4):186–192.
- Luo, F., Dong, L., and Jia, F. (2010). Method and implementation of building forces protocol dissector based on wireshark. In *2010 2nd IEEE International Conference on Information Management and Engineering*, pages 291–294. IEEE.
- Network, S. I. G. (2010). Management of obesity: a national clinical guideline. *Scottish Intercollegiate Guidelines Network: Edinburgh*, 20.
- Pires, R. and Rocha, A. (2011). Combinação de classificadores para um sistema automático de triagem de retinopatia diabética. In *Proceedings of the SPS 2011*.

- Saeedi, P., Petersohn, I., Salpea, P., Malanda, B., Karuranga, S., Unwin, N., Colagiuri, S., Guariguata, L., Motala, A. A., Ogurtsova, K., et al. (2019). Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the international diabetes federation diabetes atlas. *Diabetes research and clinical practice*, 157:107843.
- Servier, M. A. (2022). Eyes archives - servier medical art. [Accessed Jun. 1, 2022].
- Solomon, T., Zungeru, A. M., and Selvaraj, R. (2016). Network traffic monitoring in an industrial environment. In *2016 Third International Conference on Electrical, Electronics, Computer Engineering and their Applications (EECEA)*, pages 133–139. IEEE.
- Stitt, A. W., Curtis, T. M., Chen, M., Medina, R. J., McKay, G. J., Jenkins, A., Gardiner, T. A., Lyons, T. J., Hammes, H.-P., Simo, R., et al. (2016). The progress in understanding and treatment of diabetic retinopathy. *Progress in retinal and eye research*, 51:156–186.
- Swathi, C., Anoop, B., Dhas, D. A. S., and Sanker, S. P. (2017). Comparison of different image preprocessing methods used for retinal fundus images. In *2017 Conference on Emerging Devices and Smart Systems (ICEDSS)*, pages 175–179. IEEE.
- Tirumala, A., Qin, F., Dugan, J., Ferguson, J., and Gibbs, K. (2005). Iperf: the tcp/udp bandwidth measurement tool (2005). URL: <http://iperf.sourceforge.net>.
- Vandarkuzhali, T., Ravichandran, C., and Preethi, D. (2013). Detection of exudates caused by diabetic retinopathy in fundus retinal image using fuzzy k means and neural network. *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) e-ISSN*, pages 2278–1676.
- Voets, M., Møllersen, K., and Bongo, L. A. (2019). Reproduction study using public data of: Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *PloS one*, 14(6):e0217541.
- Waheed, N. K. (2018). Proliferative diabetic retinopathy. In *Atlas of Retinal OCT: Optical Coherence Tomography*, pages 88–89. Elsevier.
- Wang, X.-N., Dai, L., Li, S.-T., Kong, H.-Y., Sheng, B., and Wu, Q. (2020). Automatic grading system for diabetic retinopathy diagnosis using deep learning artificial intelligence software. *Current Eye Research*, 45(12):1550–1555.
- Wilkinson, C., Ferris III, F. L., Klein, R. E., Lee, P. P., Agardh, C. D., Davis, M., Dills, D., Kampik, A., Pararajasegaram, R., Verdager, J. T., et al. (2003). Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. *Ophthalmology*, 110(9):1677–1682.
- Wong, T. Y., Klein, R., Islam, F. A., Cotch, M. F., Folsom, A. R., Klein, B. E., Sharrett, A. R., Shea, S., of Atherosclerosis (MESA, M.-E. S., et al. (2006). Diabetic retinopathy in a multi-ethnic cohort in the united states. *American journal of ophthalmology*, 141(3):446–455.
- Ying-hua, L., Bing-Ru, Y., Dan-yang, C., and Nan, M. (2011). State-of-the-art in distributed privacy preserving data mining. In *2011 IEEE 3rd International Conference on Communication Software and Networks*, pages 545–549. IEEE.