Métodos para a Análise Semântica de Fluxos de Vídeo no contexto de Big Data e Cidades Inteligentes

Methods for Semantic Analysis of Video Streams in the Context of Big Data and Smart City

Gustavo Henrique dos Santos Schaefer, Heitor Silvério Lopes.

RESUMO

O projeto tem como objetivo propor uma infraestrutura capaz de ser modularizada para diferentes aplicações de processamento de vídeo no contexto de *Smart Cities*, e também ser escalável conforme as necessidades do seu usuário. Além disso, foi proposto um sistema modular capaz de detectar pessoas e faces, servindo como alicerce para o sistema de classificação de pessoas pelo uso de máscara facial ou não. Tal aplicação é denominada de *Scalable Face Mask Detection Pipeline* (SFMDP) e é implementada com base na infraestrutura de processamento de vídeo apresentada. Para desenvolver a infraestrutura e o SFMDP, foram utilizadas tecnologias dedicadas ao Big Data, além do uso redes neurais e técnicas de visão computacional. Com a conclusão do projeto, foi possível estabelecer uma estrutura capaz de acomodar aplicações de que exigem processamento próximo ao tempo real, um módulo segmentável com finalidade de classificação de pessoas pelo uso ou não de máscara facial, além de ter sido criado um *dataset* para realizar o treinamento da rede neural de classificação de uso de máscara facial.

Palavras-chave: Big Data, redes neurais, infraestrutura computacional, processamento de imagens.

ABSTRACT

The project aims to propose an infrastructure capable of being modularized for different video processing applications in the context of Smart Cities and also being scalable according to the user's needs. In addition, a modular system capable of detecting people and faces was proposed, serving as a foundation for the classification system for people using a face mask or not. Such application is called Scalable Face Mask Detection Pipeline (SFMDP) and is implemented based on the video processing infrastructure presented. To develop the infrastructure and SFMDP, technologies dedicated to Big Data were used, in addition to the use of neural networks and computer vision techniques. With the completion of the project, it was possible to establish a structure capable of accommodating applications that require near-real-time processing, a segmentable module to classify people by the use or not of a face mask, in addition to having created a dataset to perform the training of the face mask classification neural network.

Keywords: Big Data, Neural Networks, Computational Infrastructure, Image Processing

1 INTRODUÇÃO

Cidades inteligentes são aquelas que otimizam a utilização de recursos para servir melhor os cidadãos. A tarefa de integrar diferentes módulos, fontes de dados e processamento é desafiadora para arquiteturas convencionais de visão computacional (NAJAFABADI; VILLANUSTRE; KHOSHGOFTAAR, 2015). Além disso, tendo em vista a pandemia do COVID-19, em que novos métodos de análise de vídeo vêm sendo desenvolvidos (NAGRATH; JAIN; MADAN; ARORA; KATARIA; HEMANTH, 2021), é válido considerar aplicações que possam se integrar a sistemas de monitoramento, e que consigam auxiliar na classficação de

* Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil; gschaefer@alunos.utfpr.edu.br
† Universidade Tecnológica Federal do Paraná, Campus Curitiba; hslopes@utfpr.edu.br

pessoas pelo usou ou não de máscara facial. Dado o cenário apresentado, como podem ser criadas estruturas computacionais resilientes capazes de lidar com múltiplas fontes de dados, como análise de imagens e vídeos?

Para relacionar a análise de vídeo com uma base de processamento confiável, e estabelecer uma arquitetura de processamento computacional, é apresentada, neste trabalho, uma infraestrutura capaz de ser utilizada por diferentes aplicações. Também foi proposta uma aplicação modular de detecção e classificação de pessoas pelo uso ou não de máscara facial, juntamente com um novo *dataset* denominado UTFPR-FMD1, que consiste em imagens com duas classes (faces com e sem máscara).

2 MÉTODO

Para criar o projeto foram utilizadas tecnologias de processamento de vídeo em tempo real, como o Apache Kafka (KREPS; JAY, 2011), Docker para criação de aplicações em isoladas em container, e Redes Neurais (NN) com Keras (HOLLET, 2015) e Tensorflow (MARTIN ABADI, 2015), além do auxílio de bibliotecas Python como OpenCV (BRADSKI, 2000).

Para realizar todos os procedimentos necessários para teste e funcionamento do projeto, foi preciso uma infraestrutura computacional potente. Nesse sentido, o Laboratório de Bioinformática e Inteligência Computacional (LABIC) possui um cluster com máquinas robustas (com e sem GPUs) capazes de atender todas as necessidades requeridas pelo projeto, o qual foi realizado entre o ano de 2020 e 2021.

Um dataset corresponde ao conjunto de dados, sejam imagens, texto ou até números, usados para treinar, validar e/ou testar modelos de aprendizado de máquina. Neste projeto, foi criado um novo *dataset*, denominado UTFPR-FMD1, contendo duas classes, sendo elas face com máscara facial e face sem máscara facial.

O dataset UTFPR-FMD1 foi construído através da técnica de raspagem de dados em sites de imagens acessíveis ao público. Um grande número de imagens foi descartado por vários motivos: não é uma pessoa, má qualidade de imagem, tamanho de imagem muito pequeno e não ser uma visão frontal de uma pessoa. Na etapa seguinte, as imagens foram avaliadas de acordo com suas características principais (gênero, idade e raça) para que a o melhor equilíbrio possível de classes pudesse ser alcançado. Uma amostra do *dataset* UTFPR-FMD é apresentada na Fig. 1.

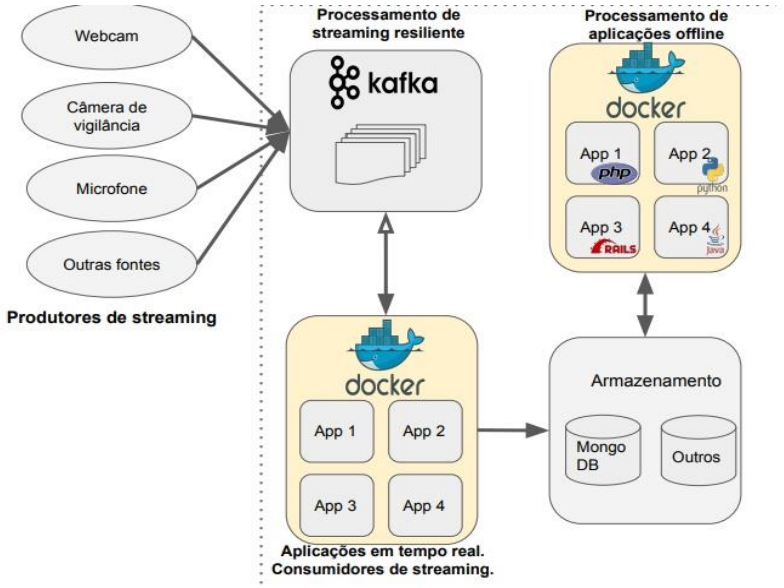
Figura 1- Amostras (Face e Máscara) do dataset UTFPR-FMD1

Fonte: Autoria própria (2021)

Após a criação do *dataset* UTFPR-FMD1, foi dado início na criação da infraestrutura que iria receber o(s) fluxo(s) de vídeo a serem analisados. Como a fonte de entrada pode ser um ou vários fluxos de vídeo, vindos de câmeras e/ou de vídeos gravados, o Kafka se tornou o ponto central entre os dados externos e a análise realizadas por aplicações como: detecção de pessoas e classificação face ou máscara. Para isto, foram utilizados produtores e consumidores Kafka. O produtor é responsável por receber e codificar os dados de entrada, e, em seguida, encaminhar cada quadro de vídeo para um tópico Kafka. Em seguida, as tarefas de processamento e inferência são executadas pelo consumidor, dentro de um container Docker.

O consumidor Kafka coleta os quadros de um tópico e inicia o pré-processamento. No entanto, como o Kafka oferece um relacionamento de muitos para muitos entre o consumidor e o produtor, a arquitetura do sistema pode ser modificada dependendo das necessidades da aplicação. Sendo assim, a arquitetura proposta possui capacidade de escalar na medida em que novos produtores e consumidores (containers Docker) são adicionados. Uma visão geral da infraestrutura proposta pode ser vista na Fig. 2.

Figura 2- Arquitetura final de processamento e suas etapas modularizadas entre diferentes aplicações



Fonte: Autoria própria (2021)

Após estabelecer a estrutura incumbida de receber e processar imagens, foi iniciada a criação do SFMDP, um pipeline composto por um modulo de detecção de pessoas, um módulo de detecção facial e um módulo de classificação de faces, intitulado FaceMask-v1, o qual foi treinado no *dataset* UTFPR-FMD1.

Os três módulos foram criados na intenção de executar um fluxo lógico, afim de aperfeiçoar o objetivo final, que é a classificação de faces por uso de máscara. Entretanto, os módulos de detecção de corpo e face podem atuar como blocos independentes.

O módulo de detecção de pessoas, como já dito, pode ser utilizado de maneira isolada ou integrado com o módulo de classificação de máscara. Por este motivo, além da detecção de corpo humano, outros processamentos são realizados para garantir que a região de detecção ***(****bounding-****box***) do corpo seja recortada e enviada corretamente para as fases posteriores.

Um experimento de observação empírica com imagens reais identificou que os erros de inferência mais comuns na detecção de face estão relacionados a diferentes objetos, em sua maioria redondos, que são confundidos com cabeças. Portanto, esta foi a motivação para realizar um pré-processamento e detecção do corpo humano antes de detectar a face na imagem completa. Deste modo, é possível melhorar a qualidade dos dados enviados para a próxima etapa do pipeline, pois ao detectar primeiro um corpo, o sistema consegue delimitar a área de trabalho da rede de detecção de faces, assim evitando muitos falsos positivos.

Na etapa de detecção de pessoas, a inferência foi realizada pelo Yolo-v4 (BOCHKOVSKIY; WANG; LIAO, 2020), treinado no dataset COCO (LIN; MAIRE; BELONGIE; BOURDEV; GIRSHICK; HAYS; PERONA; RAMANAN; ZITNICK; DOLLÁR, 2014. No entanto, apenas a classe "*person*" foi utilizada. Após

realizar a detecção, o algoritmo recebe o *bouding-box* da pessoa e realiza um recorte na imagem, encaminhando-a para a próxima fase (detecção de face).

Para a detecção de faces foi utilizado a rede neural ResNet (HE; ZHANG; REN; SUN, 2016), pré-treinada com o *dataset* Wider Face (YANG; LUO; LOY; TANG, 2016). Para evitar o achatamento da imagem, um redimensionamento é feito mantendo o seu *aspect-ratio* (taxa de proporção), melhorando assim sua qualidade quando redimensionada pela rede ResNet.

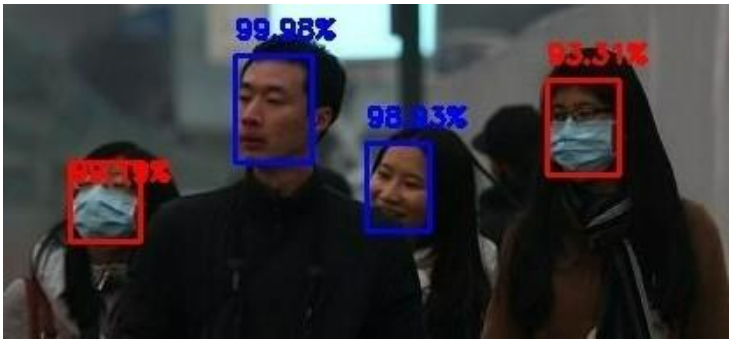
Após esse pré-processamento, a rede ResNet pode detectar a cabeça e retornar seu *bounding-box*. Para evitar falsos positivos em imagens com baixa resolução, foram estabelecidos limiares para o tamanho mínimo de uma face. Para a resolução 480p (854 × 480), o tamanho mínimo foi definido como 15 × 15 pixels. Para as outras resoluções, as proporções mínimas são definidas pela Equação (1) e Equação (2):

$$MinWidth = \frac{W}{854} \times WidthMinimun480p \quad (1)$$

$$MinHeight = \frac{H}{480} \times HeigthMinimun480p \quad (2)$$

Onde o *MinWidth* e o *MinHeight* são, respectivamente, a largura e altura mínima para uma cabeça ser filtrada em uma determinada resolução. Os valores 854 e 480 são, respectivamente, a largura e a altura da resolução 480p. W e H são a largura e a altura do quadro atual. *WidthMinimum480p* e *HeightMinimum480p* são o tamanho mínimo da cabeça testado para a resolução 480p (15 × 15).

Figura 3 - Resultado final do SFMDP, com classificação de Face (azul) e Máscara (vermelho)



Fonte: Autoria própria (2021)

Ao mesmo tempo, apenas *bounding-boxes* de cabeças quadrados ou proporções de próximas são filtradas. Se a diferença de altura e largura for maior que 72, a imagem é descartada. Após esta etapa, a região onde há uma cabeça é encaminhada para a etapa de classificação da máscara facial.

O modelo binário de classificação face/máscara é baseado em transferência de aprendizado do modelo *MobileNet-v2* (SANDLER; HOWARD; ZHU; ZHMOGINOV; CHEN, 2018), inicializado com os pesos treinados no *dataset ImageNet*. Para a classificação da face, foi aplicado uma camada de *AveragePooling2D* (3 × 3), seguido de três camadas de convolução, com 128, 256 e 256 canais, utilizando a função de ativação *Rectified Linear Unit* (ReLU). A camada de convolução final consiste em dois canais com a função de ativação *softmax*. Finalmente, foi adicionada à rede a função de *loss binary-cross-entropy*.

Ao final, é retornado um valor corresponde à probabilidade do rosto pertencer à classe “face” e o outro à classe “máscara”. Portanto, a classe da imagem é o maior valor retornado dentre os dois (ex.: [0.6, 0.4]. Após

ser definida a classe, o resultado é desenhado na imagem, utilizando como base o *bounding-box* do rosto. Um exemplo de conclusivo do SFMDP pode ser visto na Fig. 3, que demonstra a classificação final de pessoas.

3 RESULTADOS

Neste trabalho, foi criada uma arquitetura de processamento baseada em princípios de big data, que fornecem escalabilidade e resiliência. A arquitetura proposta foi feita tendo em vista a flexibilidade e performance. Os resultados do desempenho do SFMDP sendo executado na infraestrutura proposta são mostrados na Tab. 1, provando que a arquitetura é capaz de lidar com uma aplicação modular.

Tabela 1 - Desempenho geral do SFMDP na infraestrutura apresentada.

Resolução	FPS sem GPU	FPS com GPU	Aumento de FPS
432 x 240	3,58	9,41	163%
640 x 360	2,82	5,69	100%
854 x 480	2,80	5,51	98%
1280 x 720	2,77	5,40	95%
1920 x 1080	2,70	5,15	90%

Fonte: Autoria própria (2021)

Da mesma forma em que o tempo de inferência é relevante para o a arquitetura, métricas de precisão também são fundamentais. A Tab. 2 mostra o desempenho do SFMDP em diversos *datasets*, comprovando a eficácia do pipeline proposto. As métricas em questão são relevantes para avaliar arquitetura, pois são frequentemente usadas nas mais diversas publicações e servem como base de comparação com outros modelos.

Tabela 2 - Medidas de desempenho do SFMDP para diferentes conjuntos de dados.

AIZOO Tech Dataset (CHIANG, 2020)				
Classe	**Precisão**	**Recall**	**F1-Score**	**Acurácia**
Face	0,950	0,966	0,958	0,966
Mascara	0,939	0,958	0,948	0,940
FMD Dataset (ANDREW, 2020)				
Classe	**Precisão**	**Recall**	**F1-Score**	**Acurácia**
Face	0,813	0,800	0,806	0,777
Mascara	0,962	0,968	0,965	0,940

Fonte: Autoria própria (2021)

4 CONCLUSÃO

Levando em consideração os estudos feitos durante a criação do projeto, é possível notar que infraestruturas computacionais são essenciais para o desenvolvimento de aplicações em tempo real, como análise de vídeos. 1, é possível notar que a arquitetura criada é capaz de lidar com grandes fluxos de processamento em uma velocidade aceitável para as exigências de um sistema de monitoramento, neste caso o SFMDP – média de 6,2 FPS para as resoluções apresentadas.

De maneira análoga, a precisão do pipeline SFMDP foi considerada boa para diferentes *datasets*, o que comprova a eficácia final do sistema de detecção e classificação de pessoas pelo uso de máscaras. Vale ressaltar

que a precisão do SFMPD pode variar conforme o uso, visto que o pipeline depende de três redes neurais. Entretanto, é valido dizer que o SFMDP é confiável e serviu como validação para a infraestrutura criada, a qual é especifica para aplicações próximas ao tempo real.

Contudo, o SFMDP apresentou uma taxa de resposta menor do que a de outros sistemas que necessitam de retorno em tempo real, como 30 FPS. Da mesma forma, o SFMDP pode ser ainda mais otimizado, com criação de um sistema de tracking de pessoas, evitando que o mesmo individuo seja detectado repetidamente num espaço curto de tempo, e, consequentemente, aumentando sua performance.

AGRADECIMENTOS

Agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico pela bolsa de Iniciação Científica concedida a mim. Também agradeço o meu orientador Heitor Silvério Lopes e aos colegas de laboratório que me auxiliaram no projeto. Além disso, agradeço a Universidade Tecnológica Federal do Paraná pelo ótimo ensino superior.

REFERÊNCIAS

NAJAFABADI, M.M; VILLANUSTRE, F; KHOSHGOFTAAR, T.M. et al. **Deep learning applications and challenges in big data analytics**. Journal of Big Data 2, 1. 2015.

NAGRATH, P., JAIN, R., MADAN, A., ARORA, R., KATARIA, P., HEMANTH, J., 2021. SSDMNV2: **A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2**. Sustainable Cities and Society 66, 102692.

KREPS, JAY. "**Kafka: A Distributed Messaging System for Log Processing**." (2011).

MARTIN ABADI et al. **TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems**. [S.l.: s.n.], arXiv preprint arXiv:1603.04467, 2015.

HOLLET, F. et al. **Keras**. [S.l.: s.n.], 2015. Disponível em: https://keras.io.

BRADSKI, G. **The OpenCV Library**. Dr. Dobb's Journal of Software Tools, 120; 122- 125, 2000.

BOCHKOVSKIY, A., WANG, C.Y., LIAO, H.Y.M., 2020. **YOLOv4: Optimal Speed and Accuracy of Object Detection**. arXiv 2004.10934.

SANDLER, M., HOWARD, A., ZHU, M., ZHMOGINOV, A., CHEN, L.C., 2018. **MobileNetV2: inverted residuals and linear bottlenecks**, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520.

HE, K., ZHANG, X., REN, S., SUN, J., 2016. **Deep Residual Learning for Image Recognition**, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

LIN, T.Y., MAIRE, M., BELONGIE, S., BOURDEV, L., GIRSHICK, R., HAYS, J., PERONA, P., RAMANAN, D., ZITNICK, C.L., DOLLÁR, P., 2014. **Microsoft COCO: Common Objects in Context**, in: Proceedings of the European Conference on Computer Vision, pp. 740– 755.

S. Yang, P. Luo, C. C. Loy, and X. Tang, "**WIDER FACE: A Face Detection Benchmark**," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5525–5533.

CHIANG, D., 2020. **Detect faces and determine whether people are wearing mask**. URL: https://github.com/**AIZOOTech**/FaceMaskDetection.

ANDREW, M., 2020. **Face mask detection (FMD)**. https://www.kaggle.com/andrewmvd/face-mask-detection.