

Detecção de Vídeos Gerados por GAN (Generative Adversarial Networks)

Mikhael M. F. Maia¹, João Gabriel C. França², Joseppe Pedro C. Fellini³, Wendel Márcio O. Filho⁴

¹ Instituto de Informática – Universidade Federal de Goiás (UFG)

Goiânia– GO – Brasil

jacson@inf.ufg.br - professor orientador

Abstract. *This meta-paper presents a consolidated of studies on the detection of videos generated by "Generative Adversarial Networks" (GAN). The main idea of the article is to show that, over time, the number of fake videos, which are a synthesis of human images or sounds, based on artificial intelligence techniques, called "deep fakes", has grown a lot. Thus, it is shown in this article how to detect these videos generated by GAN.*

Resumo. *Este artigo apresenta um consolidado de estudos sobre a detecção de vídeos gerados por "Generative Adversarial Networks" (GAN). A ideia central do artigo é mostrar que, com o passar do tempo, o número de vídeos falsos, que são uma síntese de imagens ou sons humanos, baseadas em técnicas de inteligência artificial, chamados de "deep fakes", cresceu muito. Assim, mostra-se neste artigo como detectar estes vídeos gerados por GAN.*

1. Informações Gerais

1. Informações Gerais

Nos últimos anos, vídeos falsos criados através do uso de tecnologias, como GAN, foram crescendo. Assim, mesmo que as técnicas de criação e edição de vídeos tenham evoluído, ajudando muitos profissionais e empresas sérias, elas também ajudaram no crescimento da desinformação, alterações em resultados políticos, falsidade ideológica e até crimes sexuais pela internet. Tudo isso gerou grandes questionamentos globais sobre como a internet deveria ser usada..

Permitir mudar a identidade de um orador em um conteúdo nas redes sociais não é mais uma tarefa difícil. Antigamente, para editar um conteúdo com um computador, necessitava-se de softwares robustos e de um grande domínio deles previamente. Contudo, estes sistemas e ferramentas de manipulação facial são avançados ao ponto de usuários, sem grandes habilidades ou experiência na área, consigam, facilmente, alterar a imagem o áudio e imagem de um vídeo de uma autoridade, como um político, comprometendo-o com falas inadequadas e interferindo em resultados de eleições.

O foco principal desses vídeos falsos gerados por GAN é a manipulação facial, sendo o tipo de conteúdo mais visto nos famosos "deep fakes", que vem da junção das palavras "deep learning" (aprendizado profundo) e "fake" (falso). Esta técnica usa inteligência artificial para manipular informações ao criar fotos, vídeos e áudios. A

manipulação facial, muitas vezes, passa despercebida. Por isso, o Tribunal Superior Eleitoral (TSE) aprovou doze propostas de resolução em fevereiro de 2024, não permitindo que campanhas utilizassem “deep fakes”.

Por um lado, este avanço tecnológico coopera para o avanço da nossa sociedade, através do uso dessas tecnologias, ajudando em produções de filmes, artes visuais, produção de músicas, etc. Contudo, por outro lado, também facilita a geração de vídeos falsos por usuários maliciosos. A ideia principal deste estudo é mostrar como identificar vídeos falsos gerados por GAN.

Existem padrões que ajudam a entender que um vídeo foi feito por GAN, como falhas em expressões faciais não-naturais, piscar dos olhos anormais, descompasso no movimento dos lábios, iluminação inconsistente, vídeos com bordas irregulares, mudanças abruptas em pixels, frequências de áudio estranhas e tom robótico na voz. Entretanto, vídeos com compressões excessivas, múltiplas operações de edição ao mesmo tempo, redução de amostragem, entre outros pontos, tornam a identificação de traços de um vídeo falso mais difícil. Por esta razão, as técnicas modernas de manipulação facial são muito difíceis de se detectar. Na verdade, muitas manipulações diferentes de faces existem técnicas, ou seja, não há um modelo único explicando essas falsificações). Além disso, estas edições são feitas, normalmente, em algumas partes do corpo e não no corpo todo, ou seja, a face ou parte dela, e não o quadro completo. Consequentemente, esses tipos de vídeos manipulados são normalmente compartilhados através de plataformas sociais que aplicam redimensionamento, dificultando ainda mais o desempenho dos detectores clássicos.

2. Metodologia

Vídeos gerados por GAN são detectados através de quatro etapas, sendo elas “coleta de dados”, “pré-processamento”, “treinamento do modelo” e “avaliação e validação”.

A coleta de dados foca na utilização de datasets públicos, como o primeiro colocado no "*Kaggle Deepfake Detection Challenge*" e o "*Video Face Manipulation Detection Through Ensemble of CNNs*". Já o balanceamento de classes está vinculado à implementação de “oversampling” e “undersampling”, garantindo uma representação equilibrada de vídeos reais e sintéticos. Por fim, tem-se uma categorização dos tipos de manipulação.

O pré-processamento é dividido em três partes, “extração de frames”, “detecção e alinhamento facial”, “análise de qualidade” e “extração de características específicas”. A extração de frames foca nos segmentos-chave do vídeo, realizando-a em 30 frames por segundo. A detecção e alinhamento facial utiliza MTCNN, *Multi-task Cascaded Convolutional Networks*, para a localização precisa das faces.

O treinamento do modelo ocorre a partir de redes neurais convolucionais (CNNs), explorando arquiteturas pré-treinadas, como *EfficientNet*. Além disso, necessita-se do uso de “transfer learning”, “data augmentation”, “treinamento com otimização adaptativa”, diminuição do risco de “overfitting” e uma função de perda para medir a diferença entre as previsões do modelo e rótulos reais dos vídeos analisados.

3. Trabalhos relacionados

Três trabalhos foram utilizados como referência principal para a modelagem deste sistema, "*Kaggle Deepfake Detection Challenge*", o "*Video Face Manipulation Detection Through Ensemble of CNNs*" e "Exploring Self-Supervised Vision Transformers for Deepfake Detection: A Comparative Analysis".

Kaggle Deepfake Detection Challenge

Este artigo descreve uma solução baseada na detecção de "*deepfakes*", através de uma análise quadro a quadro, utilizando CNNs e técnicas avançadas de pré-processamento e treinamento.

Os pontos principais deste artigo são detecção de rosto, pré-processamento de imagens, estratégia de predição, aumento de dados, preparação e processamento de dados, treinamento e inferência.

A detecção de rosto é muito importante. Uma parte que as inteligências artificiais necessitam de certo treinamento é para a detecção de objetos. No caso deste projeto, o foco estava no rosto das pessoas. O detector escolhido foi o MTCNN, tendo em vista o tempo de processamento do kernel com certas restrições.

O pré-processamento de imagens utilizou a rede EfficientNet B7. Esta rede teve um treinamento prévio utilizando ImageNet e Self-Training com Noisy Student. O tamanho da entrada foi fixado em 380 x 380 pixels para otimização de memória. Além disso, as imagens foram recortadas com margem extra de 30% ao redor do rosto no processo de treinamento.

A estratégia de predição focou em analisar cada um dos vídeos em 32 quadros. Além disso, uma heurística adaptativa foi utilizada para medir os valores de saída de rede, melhorando, assim, a precisão da predição.

A aumento de dados pesada foi aplicada utilizando a biblioteca *Albumentations*, incluindo a compressão de imagens, ruído gaussiano, desfoque, flip horizontal, transformações isotrópicas de redimensionamento e remoção de partes do rosto para melhor generalização, através do *GridMask*.

A preparação e processamento de dados focaram na detecção de faces com o MTCNN, extração de recortes faciais dos vídeos, geração de landmarks faciais, criação de máscaras de diferença SSIM entre imagens reais e falsas e divisão dos dados em 16 grupos para treinamento e validação.

O treinamento ocorreu em 5 modelos *EfficientNet B7* com diferentes seeds. Ele exigiu 4 GPUs de 12 GB+ (Titan V, 1080Ti, 2080Ti ou V100). Os checkpoints foram salvos a cada época.

A inferência foi separada em três partes, o script *predict_folder.py* reproduz o kernel de predição, modelos pré-treinados podem ser baixados via *download_weights.sh* e o *predict_submission.sh* permite inferência em múltiplos vídeos e gera um CSV com os resultados. A solução utiliza um ambiente *Docker* para a configuração e execução do modelo.

Video Face Manipulation Detection Through Ensemble of CNNs

Este artigo mostra como detectar manipulações faciais, utilizando um método com um conjunto de redes neurais convolucionais (CNNs). Ele utiliza a arquitetura EfficientNetB4 como base, levando à duas modificações:

Adição de um mecanismo de atenção para focar nas partes mais relevantes da face e treinamento siamês para melhorar a extração de características discriminativas.

Os 4 modelos treinados neste artigo foram EfficientNetB4 padrão, EfficientNetB4 com atenção, EfficientNetB4 com treinamento siamês, EfficientNetB4 com atenção e treinamento siamês. O ensemble desses modelos é usado para a classificação final.

Experimentos realizados em dois datasets, com FF++ com 4000 vídeos manipulados e DFDC com 119.000 vídeos reais e falsos.

Os resultados do artigo mostraram que o mecanismo de atenção ajuda a focar em regiões importantes da face; O treinamento siamês melhora a separação de características reais e falsas; O ensemble supera o baseline (XceptionNet) em ambos os datasets; O método proposto ficou entre os 3% melhores na competição DFDC do Kaggle.

Exploring Self-Supervised Vision Transformers for Deepfake Detection: A Comparative Analysis

O estudo realizou uma comparação do uso de Transformers Visuais (ViTs) pré-treinados na detecção de *deepfakes* faciais sob duas abordagens, *backbones* com os pesos congelados como extratores de características em múltiplos níveis e o *fine-tuning* de seus blocos finais de transformers.

Assim, os autores empregaram classificadores simples na POC do *paper*, o que proporciona duas vantagens, a redução de fatores externos que poderiam influenciar a comparação e a garantia de que os resultados possam ser generalizados para qualquer classificador posterior, seja ele simples ou complexo.

Portanto, os resultados indicaram que o uso de aprendizado auto-supervisionado (SSL) em ViTs, especialmente os modelos DINO pré-treinados em grandes conjuntos de dados não relacionados à detecção de *deepfakes*, proporciona um desempenho superior na identificação de diferentes tipos de *deepfakes* em comparação com o pré-treinamento supervisionado.

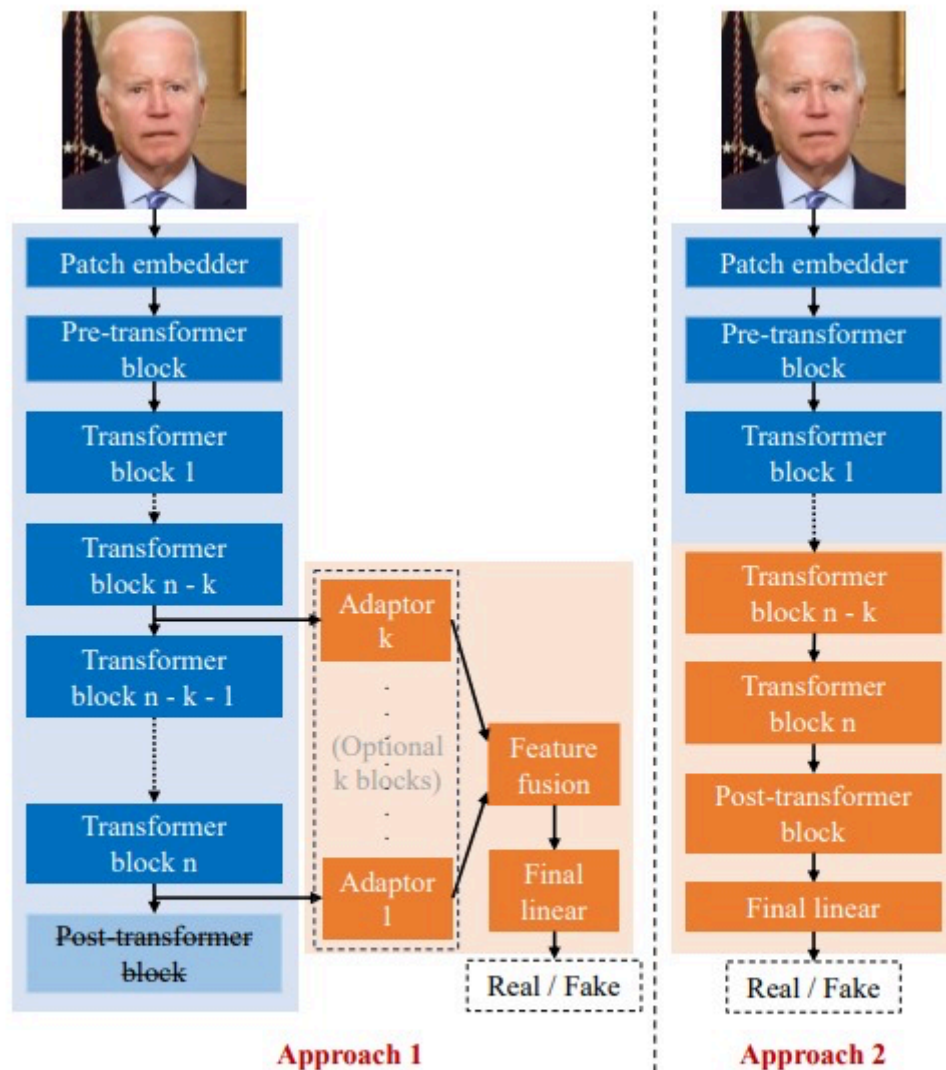
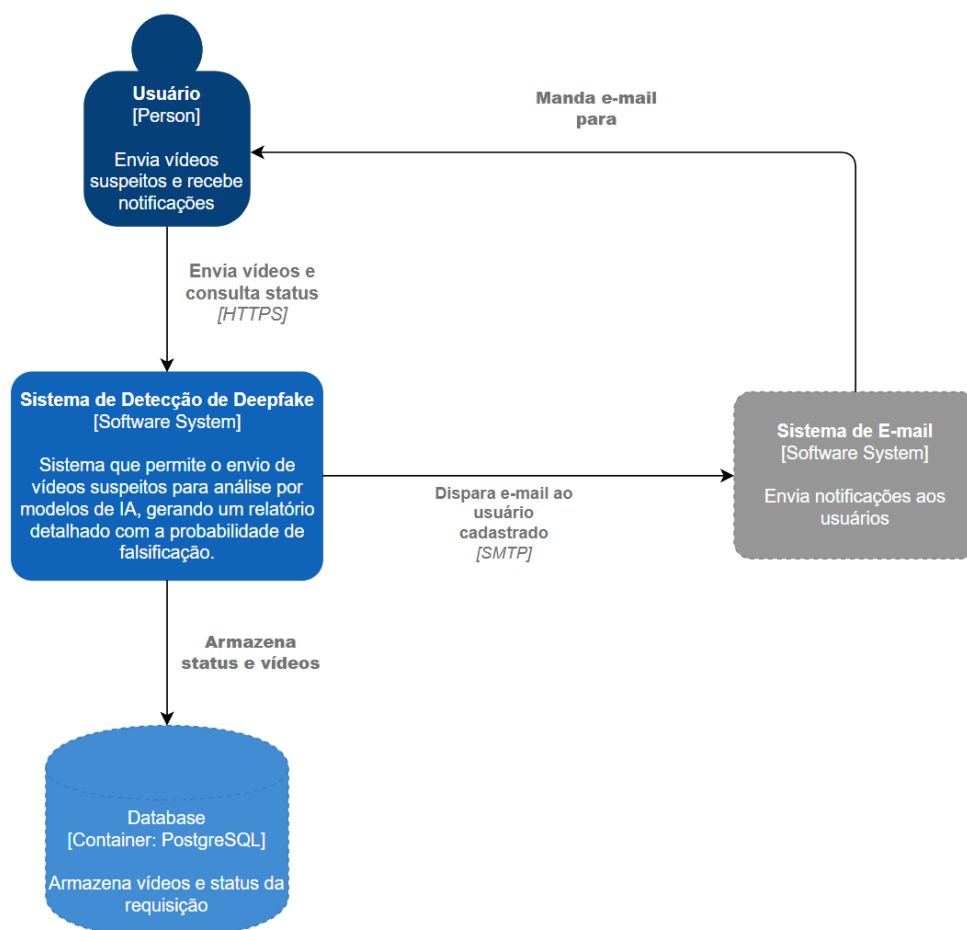


Figure 1. Overview of the two investigated approaches. Blue blocks mean frozen blocks, while orange blocks mean fine-tuned or trained blocks.

4. Arquitetura C4

Diagrama de contexto



O diagrama de contexto representa um sistema de detecção de *deepfake*. O foco deste sistema é permitir o envio de supostos vídeos suspeitos pelo usuário à análise por modelos de IA. Assim, o modelo de IA recebe o vídeo, analisa ele e retorna com um relatório, mostrando se ele é falso ou não.

O diagrama é dividido em algumas partes, “Usuário (Person)”, “Sistema de Detecção de Deepfake (Software System)”, “Database (PostgreSQL Container)” e “Sistema de E-mail (Software System)”.

O usuário envia vídeos suspeitos para análise e recebe notificações sobre o status da verificação. A comunicação com o sistema ocorre via HTTPS, garantindo segurança e integridade dos dados.

O Sistema de Detecção de Deepfake (Software System) é o principal componente que processa os vídeos usando IA para uma posterior análise do modelo de IA. Ele armazena vídeos e status das requisições em um banco de dados PostgreSQL. Posteriormente, ele envia notificações por e-mail para os usuários via SMTP.

O banco de dados atua como repositório para armazenar os vídeos enviados e seus respectivos status. A utilização de containers garante flexibilidade na escalabilidade e manutenção.

O Sistema de E-mail gerencia as notificações enviadas aos usuários. A comunicação com o sistema de detecção ocorre via SMTP, garantindo um fluxo assíncrono de mensagens.

A justificativa pela escolha desta decisão arquitetural vem de três pontos principais, separação de responsabilidades, uso de HTTPS e SMTP e banco de dados em container.

O sistema está modularizado, separando análise de vídeos, armazenamento e notificações para facilitar escalabilidade e manutenção.

O tráfego seguro entre usuário e sistema é garantido pelo HTTPS, enquanto o SMTP permite um envio assíncrono e confiável de notificações.

O PostgreSQL em container facilita a implantação, escalabilidade e portabilidade da solução.

Essa arquitetura promove segurança, escalabilidade e modularidade, tornando o sistema robusto para aplicações em larga escala.

Diagrama de containers

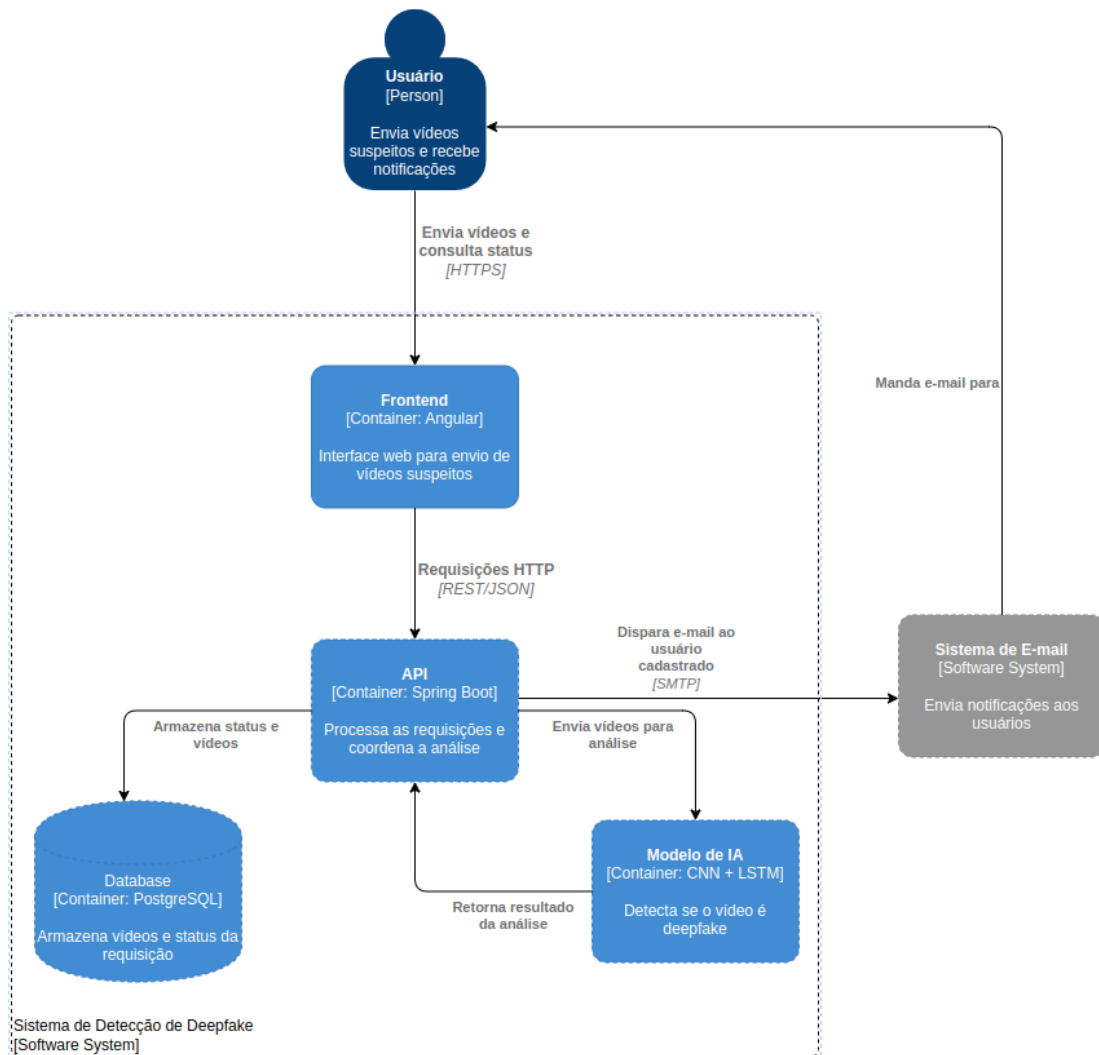
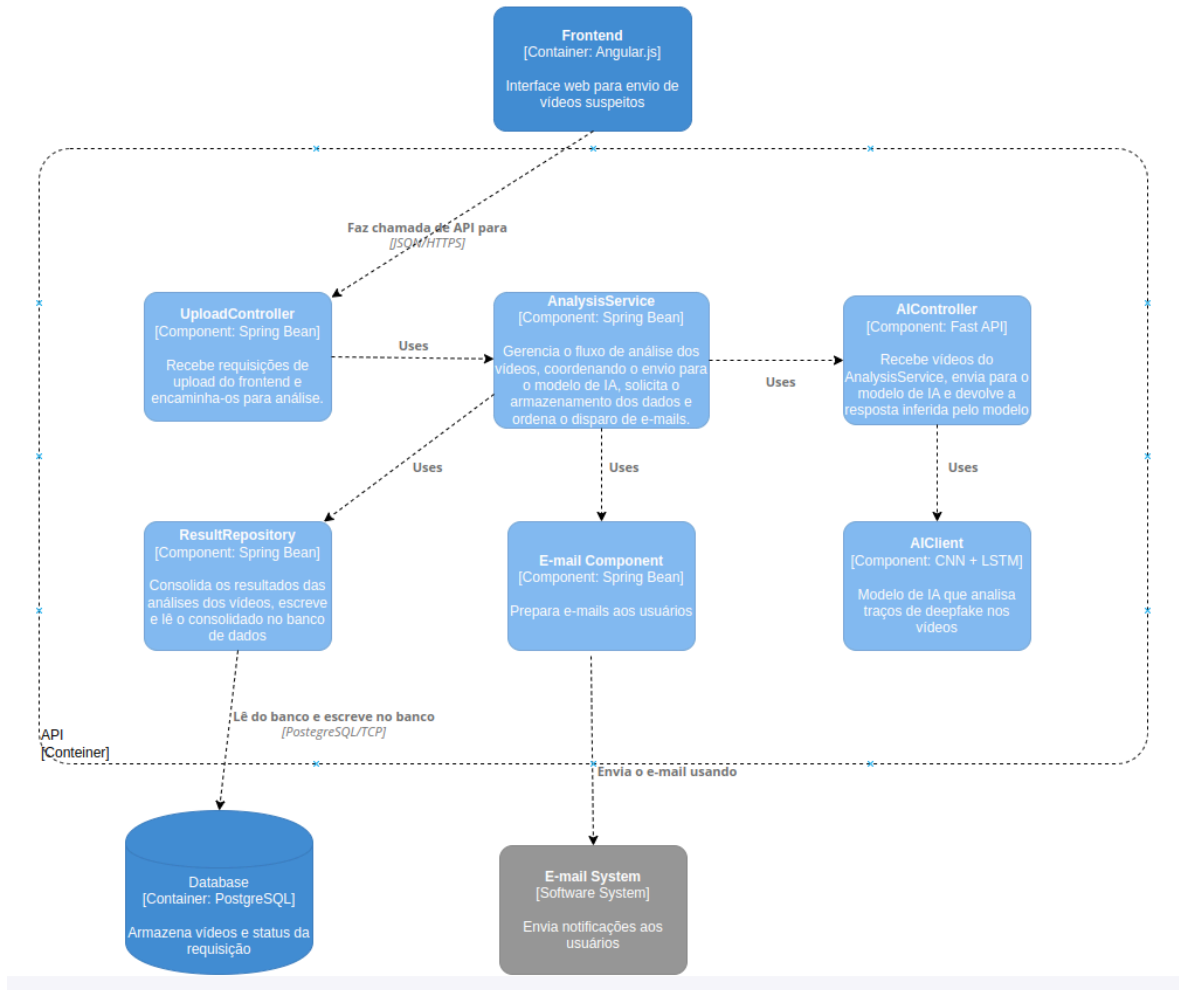


Diagrama de componentes



References

- Ballé, Johannes, Laparra, Valero, and Simoncelli, Eero P. Density modeling of images using a generalized normalization transformation. CoRR, abs/1511.06281, 2015. URL <http://arxiv.org/abs/1511.06281>.
- Bengio, Yoshua, Mesnil, Grégoire, Dauphin, Yann, and Rifai, Salah. Better mixing via deep representations. CoRR, abs/1207.4404, 2012. URL <http://arxiv.org/abs/1207.4404>.
- Blundell, C., Uria, B., Pritzel, A., Li, Y., Ruderman, A., Leibo, J. Z., Rae, J., Wierstra, D., and Hassabis, D. Model-Free Episodic Control. ArXiv e-prints, June 2016.
- Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., and Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. ArXiv e-prints, June 2016.
- Denton, Emily L., Chintala, Soumith, Szlam, Arthur, and Fergus, Robert. Deep generative image models using a laplacian pyramid of adversarial networks. CoRR, abs/1506.05751, 2015. URL <http://arxiv.org/abs/1506.05751>.
- Dinh, Laurent, Sohl-Dickstein, Jascha, and Bengio, Samy. Density estimation using real NVP. CoRR, abs/1605.08803, 2016. URL <http://arxiv.org/abs/1605.08803>.
- Donahue, J., Krähenbühl, P., and Darrell, T. Adversarial Feature Learning. ArXiv e-prints, May 2016.
- Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., and Courville, A. Adversarially Learned Inference. ArXiv e-prints, June 2016.
- Sung, W., Shin, S., Hwang, K.: Resiliency of deep neural networks under quantization. arXiv preprint arXiv:1511.06488 (2015)
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2016)
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2016)
- Yang, J., Shen, X., Xing, J., Tian, X., Li, H., Deng, B., Huang, J., Hua, X.s.: Quantization networks. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2019)
- Yoo, J., Cho, M., Kim, T., Kang, U.: Knowledge extraction with no observable data. In: Proc. Adv. Neural Inf. Process. Syst. pp. 2701–2710 (2019)
- Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., Gan, C.: Graph convolutional networks for temporal action localization. In: Proc. IEEE Int. Conf. Comp. Vis. (2019)
- Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160 (2016)
- Zhuang, B., Liu, L., Tan, M., Shen, C., Reid, I.: Training quantized neural networks with a full-precision auxiliary module. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2020)
- Zhuang, B., Shen, C., Tan, M., Liu, L., Reid, I.: Towards effective low-bitwidth convolutional neural networks. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2018)