Detecção de Vídeos Gerados por GAN (Generative Adversarial Networks)

Mikhael M. F. Maia[1], João Gabriel C. França[2], Joseppe Pedro C. Fellini[3], Wendel Márcio O. Filho[4]

[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 e treinamento.

Os pontos principais deste artigo são detecção de rosto, pré-processamento de imagens, estratégia de predição, aumentação 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 aumentação 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 DINOs 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.

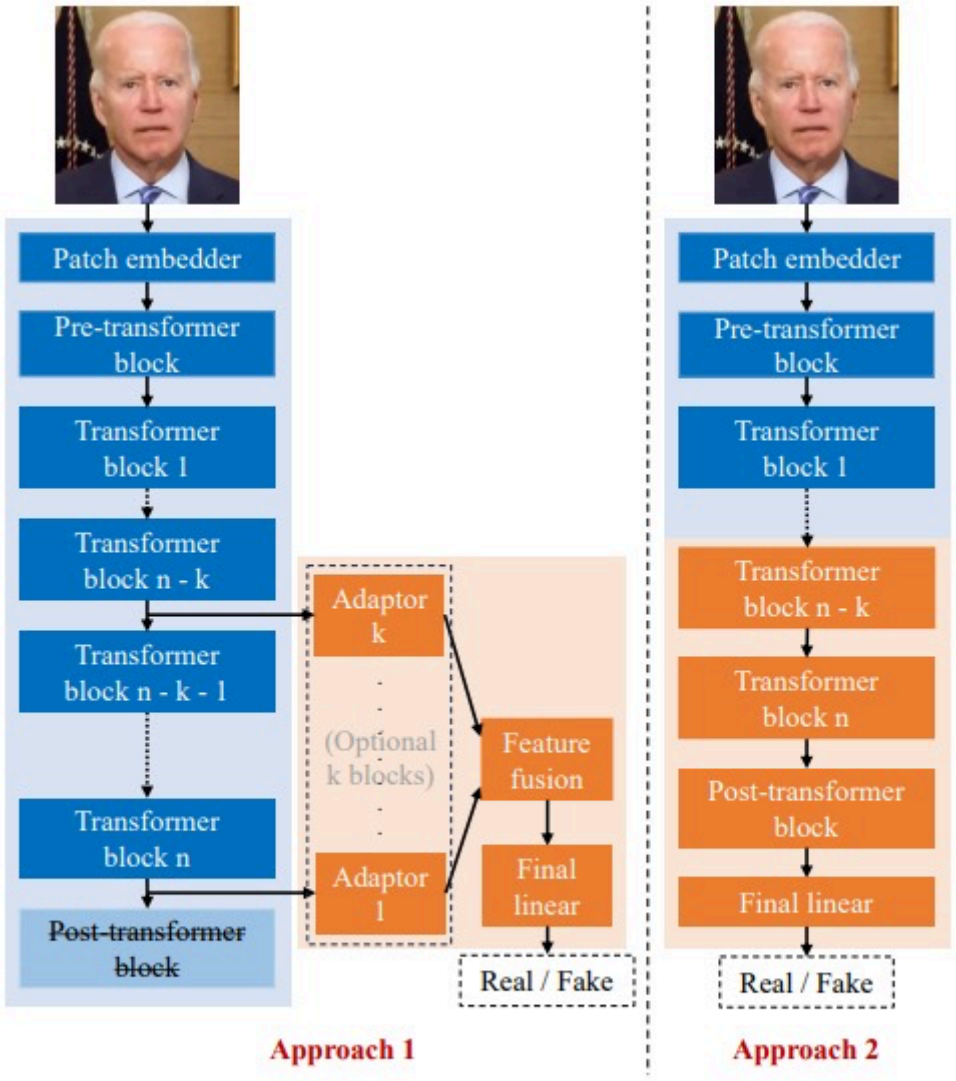


Figure 1. Overview of the two investigated approaches. Blue blocks mean frozen blocks, while orange blocks mean fine-tuned or trained blocks.

Explicação do Funcionamento dos Modelos Representados na Imagem

A figura apresenta duas abordagens distintas para a detecção de vídeos falsificados (deepfakes) usando Transformers Visuais (ViTs), que são modelos baseados em arquiteturas de "transformers". Cada abordagem explora diferentes maneiras de utilizar blocos congelados (azuis) e blocos treináveis (laranjas). Vamos detalhar cada uma delas:

Componentes Comuns (Antes da Divergência das Abordagens)

Patch Embedder

Converte cada frame ou imagem do vídeo em patches (pequenos pedaços da imagem divididos em grades) e os transforma em representações vetoriais (tokens) que

podem ser processados por um Transformer.

Esse passo inicial é um dos principais atributos dos ViTs, substituindo as convoluções tradicionais usadas em redes neurais.

Bloco Pré-Transformer (Pre-Transformer Block)

Realiza uma normalização inicial e aplica projeções lineares nos patches.

Prepara os patches para serem inseridos nos blocos transformer subsequentes.

Transformers Blocks

Compostas por várias camadas de atenção auto-supervisionadas e perceptrons multilayer (MLPs).

Cada bloco aprende a capturar representações específicas da imagem, como padrões espaciais, contextos globais e características que ajudam a diferenciar imagens reais de manipuladas.

Comparação entre as Abordagens

Características	Abordagem 1	Abordagem 2
Blocos Congelados	Primeiros blocos (azuis)	Primeiros blocos, mas com menos congelamento.
Blocos Fine-Tuned	Somente os últimos blocos (laranjas).	Mais blocos ajustáveis no final.
Adaptador	Sim, utilizados entre os conjuntos de blocos.	Não utiliza adaptadores.
Fusão de Recursos	Realiza fusão após os adaptadores.	Não realiza fusão; saída direta para classificação.
Computação	Mais eficiente devido ao uso de adaptadores.	Pode ser mais custoso por ajustar blocos finais.

Precisão	Melhor para usar em ambientes híbridos com menos tempo de treinamento adicional.	Pode alcançar maior precisão (se houver muitos recursos de ajuste).

Uso no Cenário de Detecção de *Deepfake*

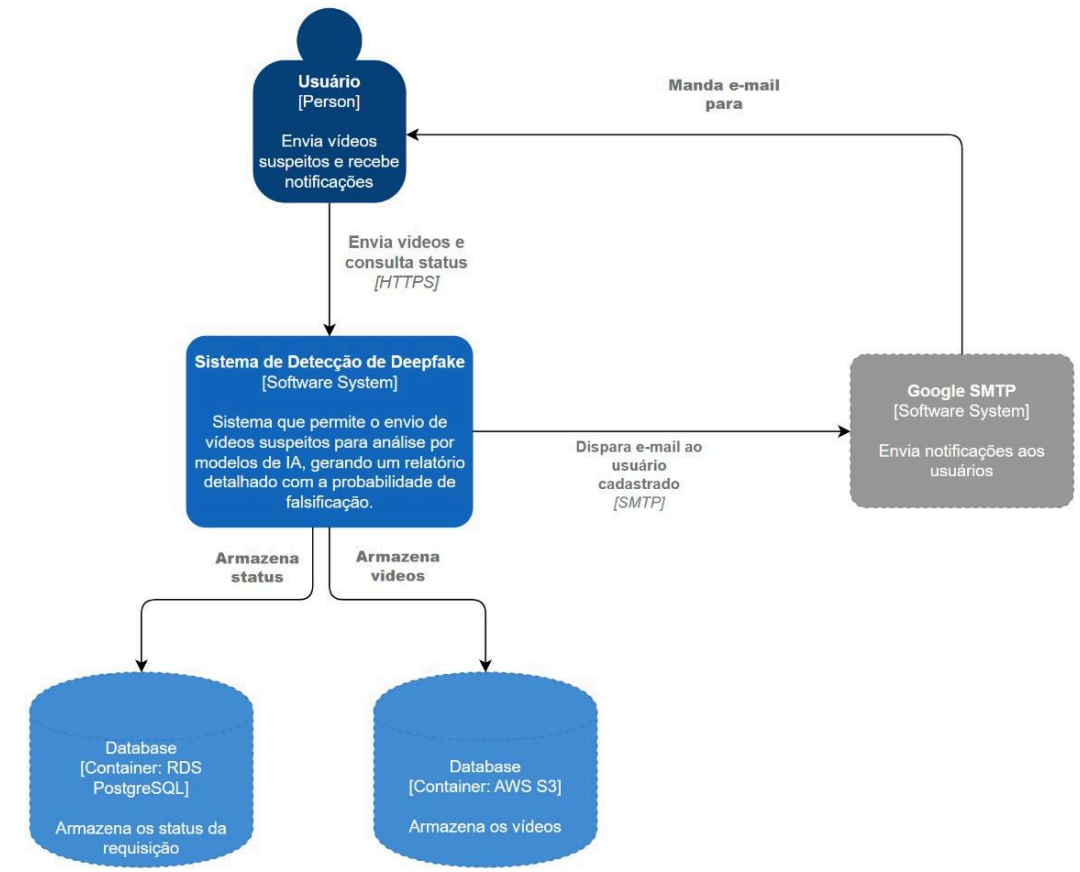
A Abordagem 1 é ideal para treinamento rápido, quando os recursos computacionais são limitados, já que reutiliza mais partes do modelo pré-treinado.

A Abordagem 2 é mais adequada para situações onde maior precisão é exigida, mas exige maior poder computacional e mais dados para ajuste.

Ambas as abordagens trabalham na identificação de padrões sutis que indicam manipulação nos vídeos, como movimentos inconsistentes, iluminação incoerente e expressões faciais desalinhadas. Dessa forma, oferecem soluções eficientes para enfrentar os desafios relacionados aos deepfakes gerados por GANs.

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 RDS PostgreSQL (Container)", "Database AWS S3 (Container)" e "Google SMTP (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, utilizando o sistema *Google SMTP*, sobre o andamento e os resultados da análise.

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. A utilização do *Amazon RDS* oferece escalabilidade e alto desempenho para gerenciar grandes volumes de transações.

O Database O Database AWS S3 (Container) é um serviço de armazenamento, hospedado em AWS S3, armazena os vídeos enviados pelos usuários. Sua arquitetura em container torna o gerenciamento e a escalabilidade mais simples e eficiente, especialmente para lidar com grandes arquivos de vídeo.

O sistema de e-mail é modulado pelo sistema Google SMTP (software system), que gerencia o envio de notificações automáticas por e-mail, informando os usuários sobre o andamento e os resultados das análises.

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

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

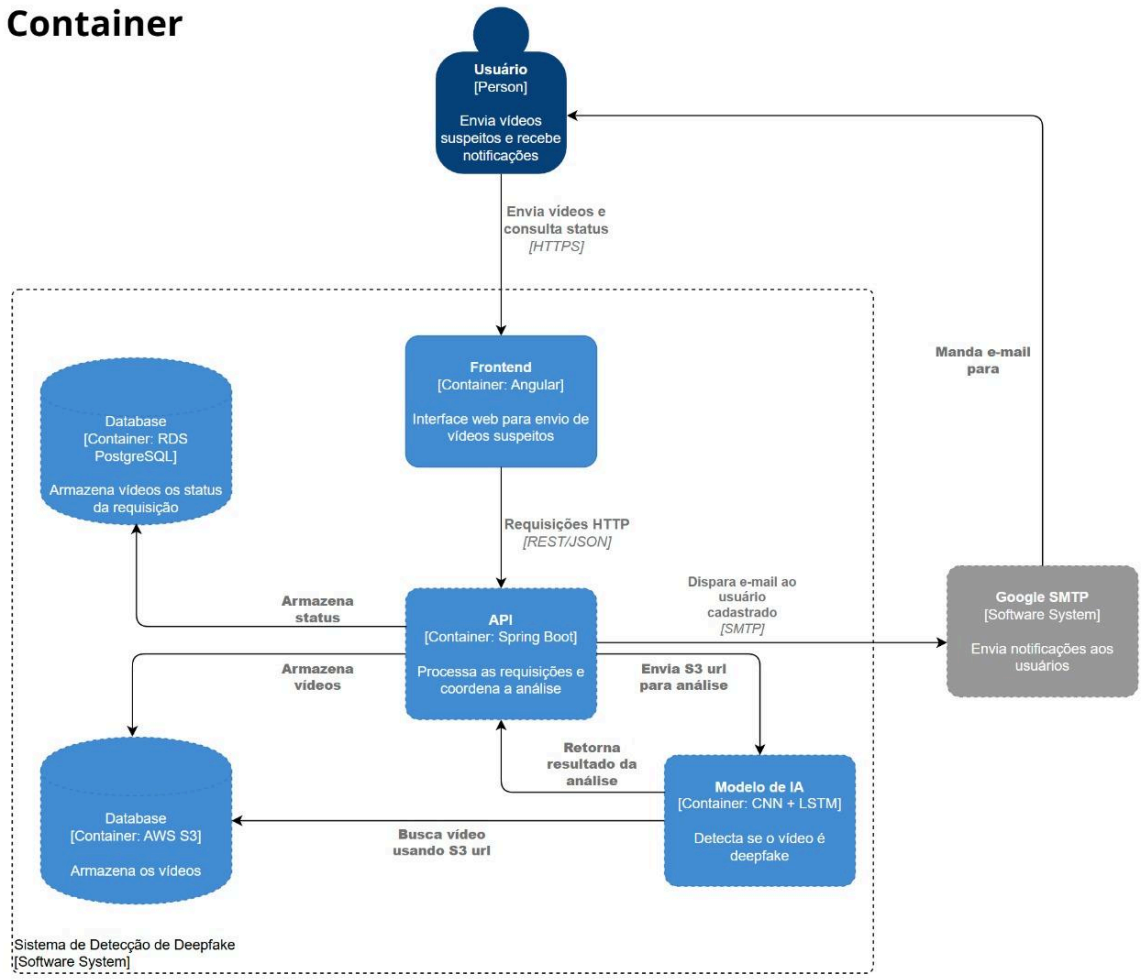
O tráfego seguro entre usuário e sistema é garantido pelo HTTPS, enquanto o Google SMTP assegura o envio confiável de notificações.

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

O armazenamento é eficiente, tendo em vista que a implementação de dois bancos de dados distintos (RDS PostgreSQL e AWS S3) organiza melhor as informações, armazenando separadamente metadados e arquivos grandes.

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

Diagrama de containers



Agora pra este diagrama O diagrama de containers representa um sistema de detecção de *deepfake*, projetado para permitir que o usuário envie vídeos suspeitos para análise por meio de modelos de IA. O sistema processa os vídeos, executa uma análise com base em IA e retorna ao usuário um relatório indicando se o vídeo é autêntico ou falso.

O diagrama está estruturado em várias partes: Usuário (Person), Frontend (Container: Angular), API (Container: *Spring Boot*), Modelo de IA (Container: CNN + LSTM), Database *RDS PostgreSQL* (Container), Database *AWS S3* (Container) e *Google SMTP* (Software System).

O usuário (Person) interage diretamente com o sistema para enviar vídeos suspeitos e consultar o status das análises realizadas. A comunicação entre o usuário e o sistema ocorre por meio de uma interface web (Frontend) via HTTPS, garantindo a segurança e a integridade dos dados.

O front-end (Angular) fornece a interface web para que os usuários possam interagir

com o sistema. Ele é responsável por facilitar o envio de vídeos e a consulta do status dos mesmos, utilizando requisições REST/JSON para comunicação com a API. Ele serve como a camada de apresentação do sistema, permitindo uma experiência amigável e eficiente para o usuário.

API (Container: *Spring Boot*) desempenha o papel de intermediária entre os diferentes componentes do sistema.

Coordenação: Processa as requisições vindas do front-end, armazena informações no banco de dados, interage com o modelo de IA e dispara notificações por e-mail para os usuários.

Armazenamento: Registra o status das requisições no banco de dados *RDS PostgreSQL* e salva os vídeos enviados no banco de dados *AWS S3*.

Integração: Envia a URL do vídeo armazenado no S3 para o Modelo de IA, obtém os resultados da análise e retorna ao Frontend.

Notificações: Dispara e-mails para os usuários cadastrados utilizando o sistema *Google SMTP*.

Modelo de IA (Container: CNN + LSTM) é responsável por detectar se os vídeos enviados são *deepfakes* ou autênticos.

O modelo de IA é baseado em técnicas de redes neurais convolucionais (CNN) e redes de memória de longa duração (LSTM), que são tecnologias avançadas para análise de vídeos.

Recebe a URL dos vídeos, realiza a análise baseada em IA e retorna o resultado (verdadeiro ou falso) à API.

O Database (Container: RDS PostgreSQL) opera como um repositório para armazenar os status das requisições, incluindo informações sobre os vídeos processados e resultados gerados. Implementado em um container Amazon RDS, que garante alta escalabilidade e desempenho no gerenciamento de um grande volume de transações.

O Database (Container: AWS S3) é utilizado para armazenar os vídeos enviados pelo usuário.

Armazenamento escalável: É otimizado para lidar com arquivos de grande tamanho, como vídeos, mantendo alta eficiência e flexibilidade.

O sistema utiliza URLs geradas pelo S3 para localizar os vídeos e enviá-los ao Modelo de IA.

Google SMTP (Software System) é responsável por enviar notificações automáticas aos usuários referentes ao andamento e aos resultados das análises.

O envio ocorre de forma assíncrona, garantindo confiabilidade na entrega das mensagens.

Utiliza o protocolo SMTP fornecido pelo Google para gerenciar a comunicação por e-mail.

Justificativa arquitetural

Cada container tem uma função bem definida, o que facilita a manutenção e a escalabilidade do sistema.

Segurança: A comunicação entre o Frontend e a API ocorre via HTTPS,

protegendo os dados do usuário.

Uso eficiente de containers: O RDS PostgreSQL organiza e armazena os status das requisições para análise de vídeos. O AWS S3 separa e armazena os vídeos enviados de forma escalável e eficiente.

Confiabilidade no envio de notificações: A integração com o Google SMTP assegura que os usuários serão notificados com precisão e eficiência.

Benefícios da arquitetura

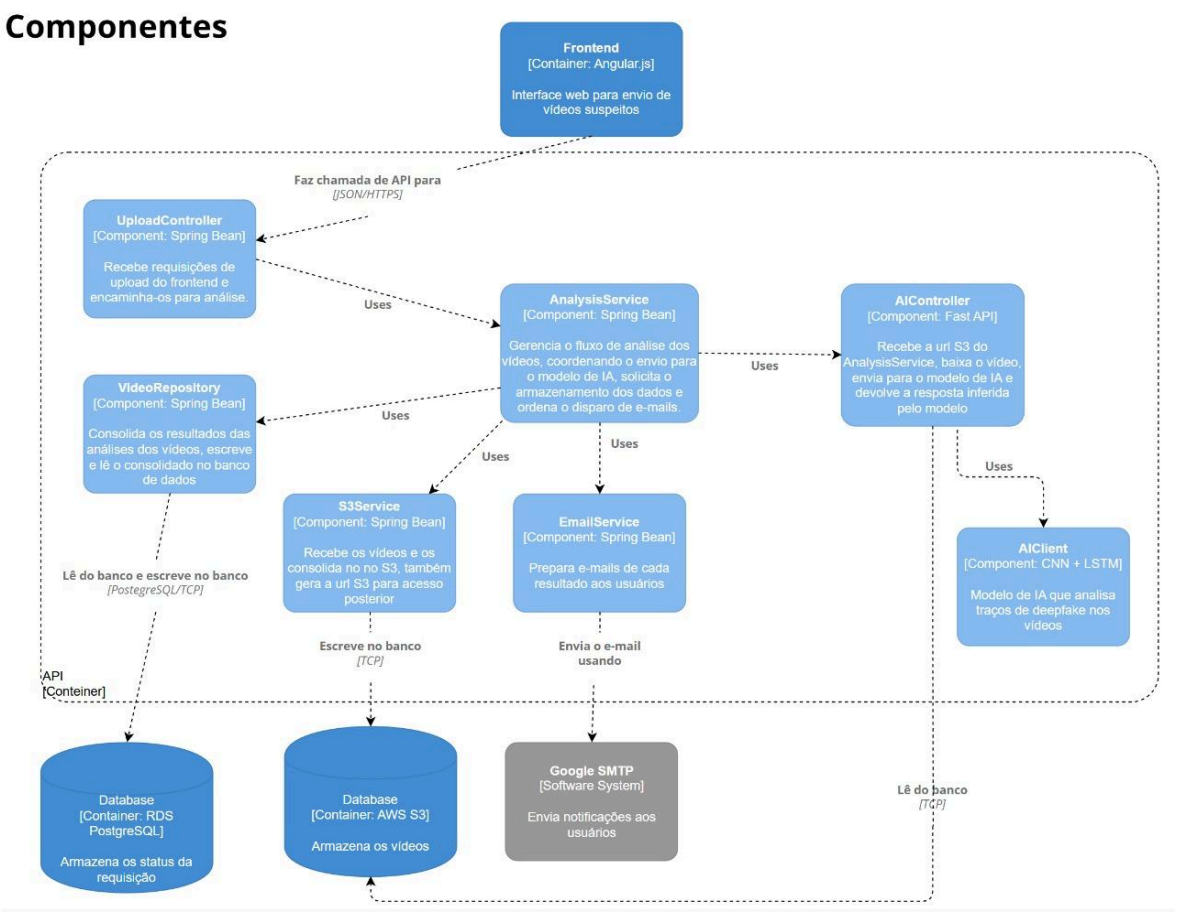
A divisão clara dos componentes promove maior organização e facilidade de manutenção.

Escalabilidade: Tanto o PostgreSQL quanto o AWS S3 em containers permitem responder a um grande volume de análise e armazenamento.

Eficiência: A separação de bancos de dados entre status e vídeos otimiza o desempenho geral.

Segurança e confiabilidade: Uso de protocolos seguros (HTTPS, SMTP) protege as interações e entrega consistente de resultados.

Diagrama de componentes



O diagrama de componentes descreve a estrutura interna do Sistema de Detecção de Deepfake, detalhando como seus módulos interagem para processar vídeos enviados pelos usuários, realizar análises de deepfake, armazenar dados e enviar notificações.

O sistema é composto por nove elementos principais divididos em Frontend, Componentes backend Spring Beans, Modelo de IA (AIClient) e Sistemas externos (Database, Google SMTP).

Descrição dos componentes

Frontend (Container: Angular.js)

Exibe a interface gráfica para o usuário e interage com a API para envio de vídeos suspeitos.

Permite o upload de vídeos para análise.

Faz chamadas à API via JSON/HTTPS.

Exibe informações sobre o status das análises e resultados.

UploadController (Component: Spring Bean)

Recebe requisições de upload de vídeos do Frontend e as encaminha aos demais serviços internos para processamento.

Registrar o vídeo enviado pelo usuário no sistema.

Validar as informações recebidas.

Acionar o AnalysisService para dar início ao fluxo de processamento e análise.

AnalysisService (Component: Spring Bean)

Gerencia o fluxo completo de análise dos vídeos, coordenando a comunicação entre os módulos.

Receber a URL gerada pelo S3Service para o armazenamento do vídeo.

Solicitar a análise do vídeo ao AIController.

Coordenar o armazenamento de metadados no VideoRepository.

Acionar o EmailService para enviar notificações sobre os resultados ao usuário.

AIController (Component: FastAPI)

Coordena a interação entre o sistema principal e os modelos de IA responsáveis pela análise do vídeo.

Receber as URLs S3 do AnalysisService e baixar os vídeos.

Encaminhar os vídeos para o AIClient para processamento e análise.

Retornar o resultado inferido (verdadeiro ou falso) ao AnalysisService.

AIClient (Component: CNN + LSTM)

Executa a análise automatizada dos vídeos para determinar a probabilidade de se tratar de um deepfake.

Processar os vídeos utilizando modelos avançados de IA que combinam Redes Neurais Convolucionais (CNN) e Redes de Memória de Longa Duração (LSTM).

Fornecer análises rápidas e precisas.

VideoRepository (Component: Spring Bean)

Componente responsável por consolidar e armazenar os metadados das análises no banco de dados relacional.

Escrever resultados das análises no banco RDS PostgreSQL.

Permitir consultas futuras sobre o status dos vídeos analisados.

S3Service (Component: Spring Bean)

Gerencia o armazenamento de vídeos no banco de dados não relacional AWS S3.

Receber os vídeos enviados pelos usuários do UploadController.

Armazenar os vídeos no AWS S3.

Gerar URLs para acessar os vídeos posteriormente e compartilhar com o AnalysisService.

EmailService (Component: Spring Bean)

Gerencia o envio de notificações por e-mail para informar aos usuários sobre o andamento e os resultados das análises.

Preparar mensagens com o status de cada vídeo analisado.

Usar o Google SMTP para envio confiável de notificações por e-mail.

Database (Container: RDS PostgreSQL)

Armazena os status e resultados dos vídeos analisados.

Utiliza o protocolo PostgreSQL/TCP para leitura e escrita de dados.

Database (Container: AWS S3)

Armazena os vídeos enviados pelos usuários.

Oferece escalabilidade e eficiência na manipulação de grandes arquivos.

Google SMTP (Software System)

Envia notificações automáticas sobre o status e os resultados das análises.

Justificativa Arquitetural

Cada componente desempenha uma função claramente definida, promovendo modularidade e facilidade de manutenção.

Escalabilidade e eficiência: O uso combinado de RDS PostgreSQL e AWS S3 permite separar o armazenamento de metadados e vídeos, otimizando a performance do sistema. O modelo de IA com CNN + LSTM melhora a precisão e velocidade da análise.

Confiabilidade e segurança: HTTPS protege a comunicação entre Frontend e UploadController. Google SMTP garante a entrega eficiente de notificações via e-mail.

Benefícios da Arquitetura

Modularidade permite evolução e manutenção fáceis.

Desempenho elevado devido à separação de armazenamento (PostgreSQL e S3).

Interface amigável ao usuário com comunicação eficiente pela API.

Confiabilidade no envio de notificações e processamento de vídeos.

5. Aplicações futuras

Aplicativos Móveis: Desenvolver aplicativos que possam detectar deepfakes diretamente em dispositivos móveis. Isso permitiria aos usuários verificar a autenticidade

de vídeos antes de compartilhá-los nas redes sociais ou por mensagem.

Navegadores e Extensões: Integração de detectores de deepfakes como extensões de navegador para alertar os usuários sobre vídeos suspeitos enquanto navegam na web. Esses plugins poderiam analisar vídeos em tempo real e fornecer feedback imediato.

Sistemas Operacionais: Implementação de APIs de detecção de deepfake em sistemas operacionais de dispositivos, permitindo que qualquer aplicativo que utilize vídeos possa verificar sua autenticidade automaticamente.

Educação e Conscientização

Campanhas de Alfabetização Digital: Programas educacionais voltados para ajudar o público a entender o que são deepfakes, como identificá-los e por que é importante verificar a autenticidade de conteúdos digitais.

Workshops e Treinamentos: Organizar workshops para jornalistas, educadores e profissionais de mídia sobre como usar ferramentas de detecção de deepfake para garantir a integridade das informações difundidas ao público.

Recursos Educativos Interativos: Desenvolvimento de plataformas interativas online que permitam aos usuários aprender mais sobre deepfakes e testar suas habilidades em identificá-los usando exemplos reais e sintéticos.

Segurança e Defesa

Proteção de Identidade: Sistemas de vigilância que monitoram plataformas de imprensa e redes sociais para identificar e tomar medidas contra deepfakes que visem criar desinformação ou prejudicar indivíduos específicos.

Segurança Nacional: Aplicação em sistemas de segurança nacional para monitorar informações de inteligência, garantindo que vídeos ou áudios falsificados não sejam usados para campanhas de desinformação em larga escala.

Cibersegurança Corporativa: Empresas poderiam implementar tecnologias de detecção de deepfake para proteger a integridade dos seus dados e comunicações internas, prevenindo fraudes e vazamentos de informações por meio de vídeos manipulados.

References

Ball´ e, 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., Houthooft, 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., Kr¨ ahenb¨ uhl, 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 preprintarXiv: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)

Ramesh, H., et al. (2023). "Deepfake Detection: An Overview of Current Techniques and

Future Challenges". Journal of AI Research.

Martinez, L., et al. (2024). "Enhanced CNN Architectures for Improved Detection of GAN-generated Media". Advanced Computing Conference Proceedings.