Comparação entre técnicas de *Liveness Detection* para prevenção de *Spoofing* em reconhecimento facial

João Paulo Paixão Rocha, Pedro Augusto Rodrigues Pinheiro

2025

Resumo

O reconhecimento facial é utilizado em diversos lugares, sendo um elo importante para a segurança de estabelecimentos, impedindo pessoas não autorizadas de acessarem recursos sem liberação. Nesse contexto, formas de ataque que burlem o sistema de detecção são uma ameaça real e medidas que impeçam esse tipo de ameaça devem ser adotadas. Uma técnica importante para prevenir ataques de *Spoofing* em deteção facial é o uso de *Liveness*, onde busca detectar se a imagem que está sendo analisada é de uma pessoa viva ou fotos e vídeos pré-gravados. Nesse trabalho, o *Liveness* será detectado por meio de piscadas dos olhos durante a detecção, sendo comparadas implementações dessa técnica para avaliar as performances.

Palavras-chave: *Spoofing*, Reconhecimento Facial, *Liveness Detection*.

1 Introdução

Com o grande desenvolvimento tecnológico das últimas décadas, se tornou cada vez mais comum utilizar a tecnologia de forma a reduzir custos e aumentar a eficiência, tornando todo o processo de segurança mais fácil de ser implementado. Uma das tarefas mais importantes no quesito de segurança, seja a segurança física de um local ou a segurança da informação é o controle de acesso a recursos, permitindo somente que pessoas autorizadas obtenham acesso.

Uma das formas em que a tecnologia auxilia nesse aspecto por meio do uso de reconhecimento facial para controle de acesso, realizando a liberação automática, desde que a pessoa possua autorização para tal. Nesse processo, é preciso obter imagens de quem solicitou o acesso, seja por meio de fotos ou vídeos, e então processar a imagem para identificá-la. Porém, como é uma imagem que será utilizada para o processamento, uma vulnerabilidade que sistemas mais simples possuem é o *Spoofing*, que no contexto de reconhecimento facial, consiste em usar fotos e vídeos falsos para burlar o sistema de autenticação e obter acesso indevido à recursos.

Neste trabalho, serão abordadas técnicas de detecção de *Spoofing* voltadas ao reconhecimento facial e autenticação biométrica, buscando aumentar a robustez e confiabilidade de sistemas simples de controle de acesso por meio de reconhecimento facial. Dessa forma, será feita a comparação de métodos ingênuos de reconehcimento, sem nenhum tipo de prevenção de *Spoofing*, bem como técnicas que utilizam a detecção de olhos piscando (NISHANTH; RAO, 2019) para definir se é realmente uma pessoa, ou se é uma foto sendo mostrada à câmera de controle. Logo, a perfomance das técnicas será comparada e discutida a eficácia desses modelos em ambientes reais

Dessa forma, o objetivo deste trabalho é avaliar o impacto das técnicas de *Liveness Detection* baseadas no movimentos dos olhos para detectar se a face sendo reconhecida é uma pessoa viva ou uma foto ou retrato.

2 Revisão da literatura

2.1 *Liveness Detection e Spoofing*

Spoofing são técnicas que atacantes maliciosos utilizam para obter acesso à locais, informações ou dados que não deveriam, figindo ser outra pessoa e disfarçando sua identidade. Muitas vezes isso ocorre quando o atacante se disfarça com um site, um email ou número de telefone falsos, se passando por outra pessoa(FBI, 2021). Isso ocorre pois o atacante muitas vezes deseja obter acesso à credenciais secretas como logins e senhas de banco. No caso da autenticação biométrica, o *Spoofing* ocorre ao utlizar os dados biométricos de outra pessoa. Casos mais comuns envolvem coletar a digital de alguém ou então utilizar fotos em sistemas de reconhecimento facial.

O *Liveness Detection* é muito importante para o reconhecimento biométrico, uma vez que os algoritmos comuns não conseguem diferenciar entre pessoas vivas e uma falsificação(CHAKRABORTY, 2014). Dessa forma, o sistema de autenticação deve ser capaz de identificar se a imagem captada é real ou não, para garantir proteção contra *Spoofing*. Diversas técnicas podem ser utilizadas para tal, desde verificação de texturas, detecção de movimentos invlountários e até mesmo análises profundas das imagens, em busca de definir a veracidade delas. Assim, o *Liveness Detection* é uma camada importante em sistemas de autenticação biométrica, permitindo que o controle de acesso não seja burlado com truques simples, como mostrar para a camera um retrato de alguém com autorização.

2.2 Detecção de Movimentos

Uma técnica comum utilizada para o *Liveness Detection* é a análise dos movimentos dos olhos. Nos humanos, os movimentos dos olhos são muito erráticos e invountários, o que os tornam um bom canditado para analisar vídeos de reconhecimento facial. Como os movientos são involúntarios, é dificil que alguém consiga os controlar e com isso, nas imagens deve haver uma diferença nas formas dos olhos durante a captura dos vídeos(CHAKRABORTY, 2014). Assim, caso não seja detectado nenhum movimento, pequeno ou grande, na área dos olhos, é extremamente provável que as imagens capturadas sejam uma falsificação.

2.3 Detecção de Piscadas

Outra forma de detectar a falsificação de imagens biométricas é utilizando a verificação de piscadas em um vídeo. Assim, caso seja apresentada uma foto, ao invés de

uma pessoa real, o sistema buscará pelo fechamento dos olhos como indicativo de *Liveness*.

O método utilizado para tal verificação é o cálculo do EAR(Eye Aspect Ratio), que compreende a razão entre a distância de pontos nos olhos. Então, o EAR é cálculado por de pontos determinados nos olhos, como visto na Figura 1. Com a distância euclidiana entre os pontos, quando o usuário fechar os olhos, o valor do EAR cai de forma considerável, sendo possível identificar quando houve uma piscada(NISHANTH; RAO, 2019).

O cálculo realizado para o EAR é o seguinte:

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\,\|p_1 - p_4\|}$$

.

Com esse cálculo, é possível identificar quando o olho foi fechado e se a frequência com que o usuário pisca for natural, ou seja, dentro do esperado, a imagem é classificada como real. Agora, se o EAR ficar abaixo do *threshold* definido para o sistema ou o usuário estiver piscando de forma estranha, a imagem será classificada como falsa, indicando possível *Spoofing*.

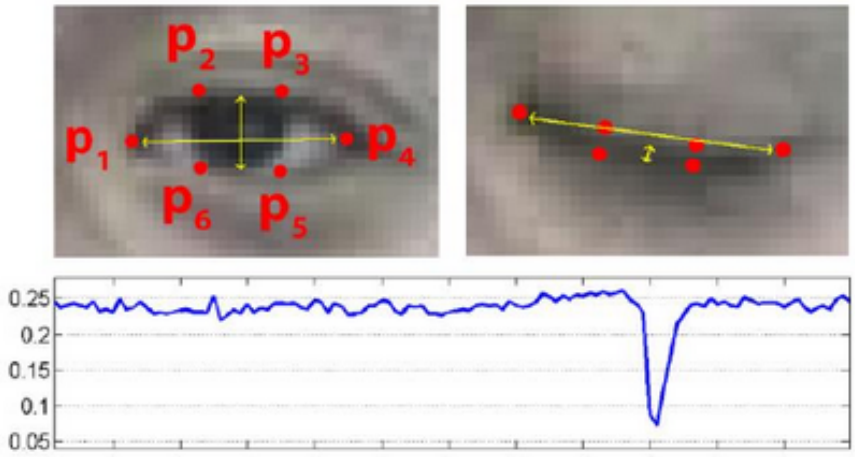

Figura 1 – Cálculo e Pontos utilizados no EAR. Fonte: (NISHANTH; RAO, 2019)

2.4 Detecção de Movimentos da Boca

Uma técnina similar ao cálculo do EAR é o MAR(Mouth Aspect Ratio). Esse método funciona de forma parecida com o EAR, porém agora nos contornos da boca. Como o mesmo conceito do EAR é utilizado, é possível detectar quando o usuário abre a boca ou está sorrindo, o que torna possível criar alguns desafios para testar se a imagem é real, como pedir para que o usuário sorria. Nesse caso, se for um vídeo pré gravado ou uma imagem, ela não cumprirá o desafio, falhando no teste de *Liveness*.

O cálculo do MAR segue a mesma fórmula que o cálculo do EAR, agora seguindo somente os pontos da boca:

$$MAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\,\|p_1 - p_4\|}$$

.

Assim, quando o valor do MAR diminui, é possível saber que a boca do usuário está fechada. De forma semelhante, esse tipo de análise é utilizada para detectar se motoristas estão cansados ou com sono, por meio dos bocejos do motorista. A Figura 2 mostra os pontos utilizados para encontrar a abertura ou não da boca. A quantidade de pontos que será utilizada na análise depende da implementação e da precisão necessária

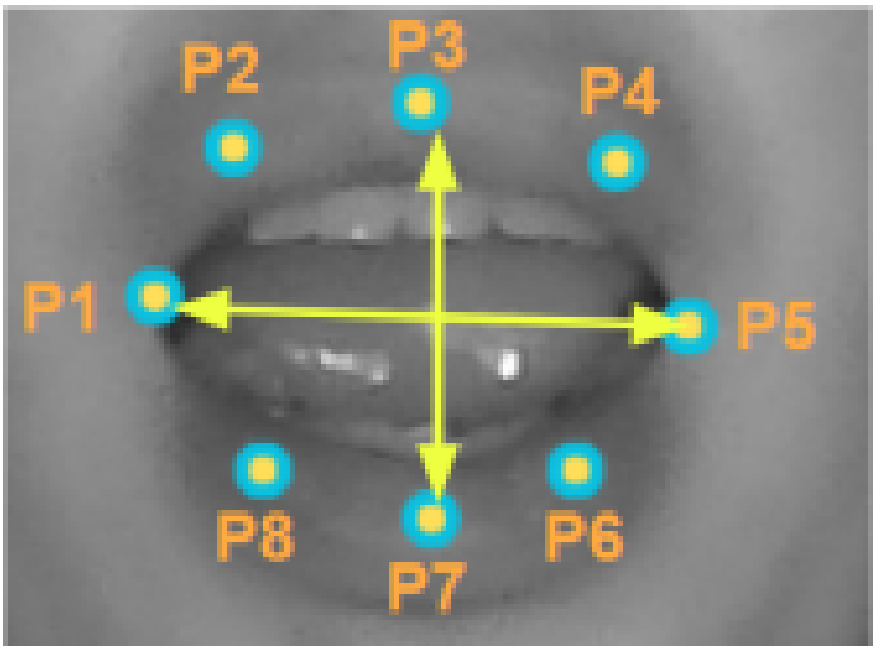

Figura 2 – Cálculo e Pontos utilizados no MAR. Fonte: (FLOREZ et al., 2024)

3 Metodologia

3.1 Técnicas implementadas

Para as análises e comparações, foram implementadas 2 técnicas diferentes para deteção de *Spoofing*, além do reconhecimento facial *naive*, ou seja, sem nenhuma prevenção contra ataques. Assim, é possível comparar a robustez das técnicas quando comparadas a um método mais simples e evidenciar a necessidade de ter prevenções contra tais tipos de ameaças.

A primeira técnica implementada trata-se do uso somente do EAR, buscando nas imagens capturadas pela piscada do usuário. Assim, o sistema detecta os olhos da pessoa e faz a contagem de quantas vezes os olhos foram fechados, e então decide se a imagem é real ou uma falsificação.

Já a próxima técnica usa, aliado ao EAR, o MAR e um sistema de desafios aleatórios que devem ser cumpridos. Logo, além de detectar se a pessoa está fechando os olhos, o sistema é capaz de solicitar que o usuário realize algumas ações, para garantir que é uma pessoa. O sistema pode exigir que o usuário pisque somente um dos olhos, como o esquerdo ou direito, que ele sorria ou então vire a cabeça. Como os desafios são aleatórios, torna

difícil que esse tipo de imagem seja falsificada, ainda mais se métodos rudimentares de *Spoofing* forem utilizados, como mostrar uma foto ou uma máscara.

Nos experimentos, algumas situações simulando ataques de *Spoofing* serão testadas, utilizandos vídeos reais, videos mostrando uma foto estática, e vídeos mostrando outras gravações de rostos já autenticados no sistema. Assim, é possível analisar como cada uma das implementações se comporta quando submetidas a ataques e tentativas de burlar a seguraça.

3.2 Implementação e captura de dados

Para a implementação dos algoritimos, foram utilizadas as bibliotecas MediaPipe e Opencv, para realizar capturar e processar as imagens e a biblioteca *face_recognition* para realizar a detecção facial. Além disso, os modelos foram treinados com fotos estáticas dos autores para poder reconhecê-los.

O ambiente utilizado para a implementação foi o linux e macos, utilizando o software *OBS Studio* para capturar os vídeos utilizados no testes de ataque. Com os vídeos pré-gravados por meio do *OBS*, cada um dos vídeos foi utilizado como entrada para os modelos. Foram gravados 10 vídeos com cenários possíveis, sendo 5 de cada um dos integrantes.

4 Experimentos

Para testar a capacidade de cada técnica implementada, foi preparado um conjunto de dados contendo os seguintes cenários:

- Entrada legitima
- Tentativa de spoofing com imagem estática
- Tentativa de spoofing com video piscando
- Tentativa de spoofing com video sorrindo
- Tentativa de spoofing completa, piscando, sorrindo e mexendo a cabeça

Além disso, foram gravados vídeos diferentes para cada um desses cenários, conseiderando condições diferentes, locais diferentes e iluminação diferentes. Com isso, os vídeos simulam vídeos em condições diversas, como por exemplo em um ambiente externo, ambiente interno e diferentes iluminações do ambiente externo, como luzes mais frias e luzes quentes, como uma luz amarela.

Para os testes, foram gravados vídeos de 3 pessoas diferentes, nos cenários descritos anteriormente, de forma a aumentar a quantidade de dados e varieadade de rostos e pessoas que são detectadas pelo sistema.

As Figuras 3 e 4 mostram como o reconhecimento *naive*, ou seja, sem nenhuma proteção, e o reconhecimento em conjunto com EAR se comportam quando submetidos a tentativas de *Spoofing*. Na Figura 3, é possível identificar que mesmo se tratando de uma foto, o sistema ainda reconheceu o rosto mostrado, indicando pouca proteção contra os ataques. Na Figura 4, é possível observar o oposto, por se tratar de uma foto, o *Liveness* não foi detectado, e portanto a imagem foi classificada como falsa. Na imagem é possível

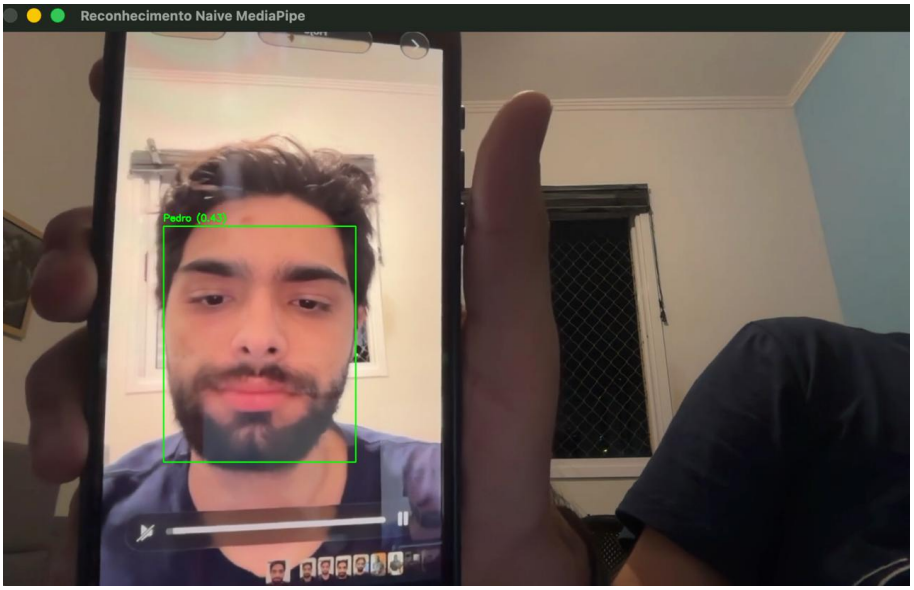

Figura 3 – Tentativa de *Spoofing* no método *naive*.

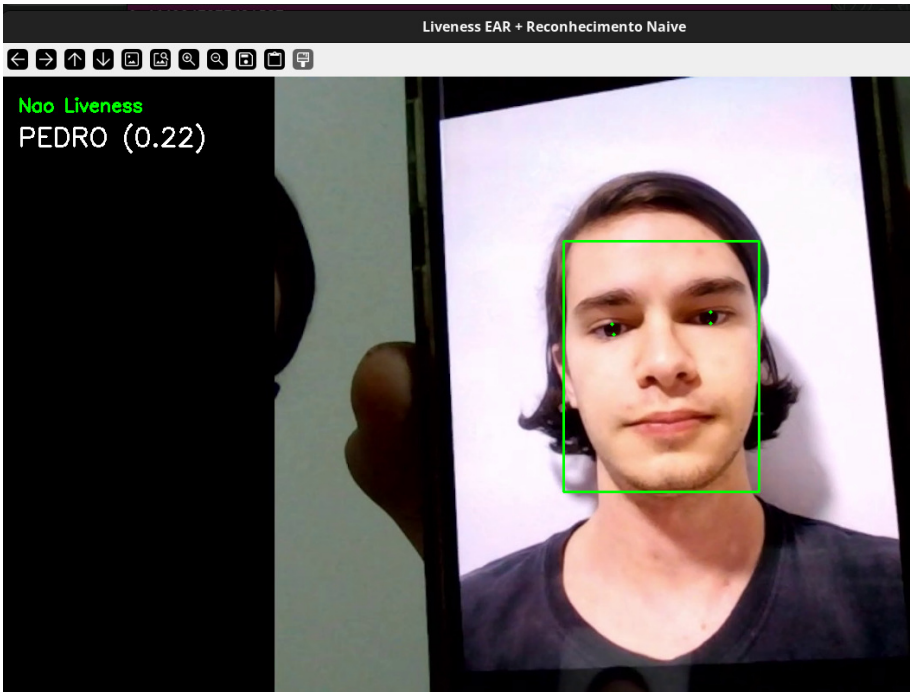

Figura 4 – Tentativa de *Spoofing* no método com EAR.

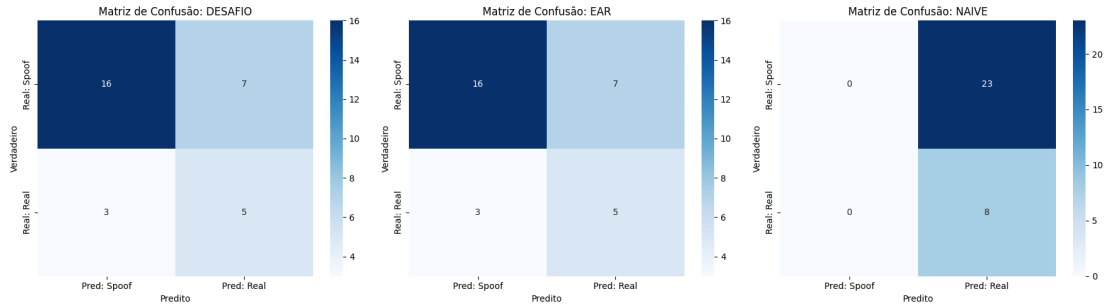

Figura 5 – Matriz de confusão de cada caso de teste.

ver que o *Liveness* não foi detectado. Foram utilizados dois sistemas e duas câmeras diferentes, em ambos o desempenho foi similar, não afetando a precisão dos resultados. Outro ponto foi o uso de dispositivos diferentes para tentar o *spoofing*, exibindo as fotos ou vídeos utilizados no ataque em telas diferentes, para adicionar mais essa camada de variabilidade.

Após os testes obtemos o seguinte resultado:

Conforme podemos observar na Figura 5, é nítido que a implementação exigindo desafio oferece uma proteção maior contra spoofing, porem não é impenetrável, por conta da aleatoriedade de do desafio exigido, se o invasor possuir conhecimento prévio do sistema de segurança ele pode preparar um video de spoofing contendo o movimento requirido, uma desvantagem desse método é que sua detecção é menos fluida, pois exige mais passos e leva mais tempo. Por conta disso, a eficácia do método com os desafios ficou identica ao método utilizando somente o EAR, pois por necessitar de um movimento aleatório, os vídeos pré-gravados, mesmo os reais, possuem uma chance de não serem aceitos como verdadeiros. Por conta disso, há alguns vídeos reais que o sistema identifica como *Spoofing*.

Porém, essa forma foi necessária para implementar e automatizar os testes. Outro ponto, é que em testes reais, o sistema possui uma acurácia maior, uma vez que os erros que abaixam sua acurácia não ocorrem, pois os vídeos reais podem reagir da forma correta. Dessa forma, a acurácia real do teste será maior que o observado.

O método EAR convencional apresenta uma boa proteção, mas por ser simples é mais fácil de ser burlado, isso é observado nos casos de spoofing bem preparado, possuindo a mesma fraqueza que a implementação de desafio, porém com a vantagem de ser um método mais simples e fluido para o usuário. Nesse caso, se o atacante conhecer o sistema, um vídeo exibindo as piscadas será suficiente para burlá-lo.

Esses 2 métodos obtiveram a mesma matriz, porém não por que realemente tenham a mesma eficácia. Isso ocorreu por conta da variabilidade nos testes do modelo com desafios, já que por conta dos desafios, alguns vídeos reais não foram detectados como tal. Além disso, não foram os mesmos vídeos que foram preditos como *spoofing* mas eram reais, em ambos os casos.

Por sua vez, o método *naive*, contando somente com o reconhecimento facial e sem nenhuma camada de proteção obteve de forma esperada a pior performance. Como nesse método nenhuma medida de prevenção é tomada ao capturar a imagem e o sistema só tentará reconhecer a face apresentada, todas as tentativas de *Spoofing* foram bem-sucedidas, indicando que sistemas que contam somente com essa proteção estão extremamente

vulneráveis à ataques, mesmo que os mais simples, como mostrar uma foto para a camera.

Os testes não mostraram ter problemas com rostos ou condições de iluminação diferentes, uma vez que ele conseguiu reconhecer os vídeos corretos, errando quando o desafio não coincidiu com o vídeo registrado, por exemplo. Porém, algo que influenciou a detecção, foi a amplitude do fechamento dos olhos. Um dos vídeos reais que foram identificados como *spoofing*(real3.mp4) continha a pessoa fechando os olhos levemente, fazendo com que o sistema não detectasse o *liveness*. Essa variabilidade nos paramêtros também explica os erros obtidos pelos 2 modelos com EAR e desafios.

5 Considerações finais

Dessa forma, os métodos mais elaboradas de detecção de *Liveness* obtiveram resultados melhores na proteção contra ataques de *Spoofing*, demonstrando a capacidade de detectar quando imagens falsas são mostradas para a camera, na tentativa de burlar a segurança. Em primeiro lugar, a implementação utilizando somente o EAR, ou seja, verificando se os olhos se fecham, já torna o sistema capaz de evitar tentativas de exploração simples, como utilizar uma foto da pessoa que tem acesso liberado. Porém, tentativas mais robustas de ataque podem envolver uso de vídeos pré gravados da pessoa, o que possibilita que o sistema detecte os olhos e quando eles piscam. Nesse caso, a combinação do EAR, uso do MAR e seleção aleatória de desafios aumentam a capacidade de detecção de tentativas de burlar a segurança, uma vez que não é necessário que o usuário pisque somente, mas também obedeça a comandos que o sistema de reconhecimento emite. Assim, se os comandos não forem obedecidos, o reconhecimento de *Liveness* falha. Porém, ainda há algumas vulnerabilidades nesses métodos. Caso o atacante conheça os padrões de desafios, é possível fabricar um vídeos que seja liberado pelo reconhecimento biométrico.

Além disso, os sistemas são sensíveis à amplitude de abertura ou fechamento dos olhos e da boca, ou seja, caso haja pouco movimento, os modelos suspeitarão que se trata de um *Spoofing*. Isso ocorre pois, se a proporção for muito pequena, não ultrapassará o *threshold* para ser considerado vivo, mesmo que o vídeo seja real. Assim, os movimentos utilizados para detecção devem ser visíveis e claros.

Para trabalhos futuros, são buscadas maneiras de melhorar a classificação da imagem entre real ou *Spoofing*, melhorando a capacidade de reconhecimento. As Redes Nerais Convolucionais (CNN) podem ser treinadas para classificar as imagens quanto à veracidade e quando usadas em conjunto com a detecção de *Liveness*, melhoram a segurança do sistema. Tais modelos híbridos já demonstraram possuir maior precisão e performance quando comparados a modelos utilizando somente as CNNs(HASAN; ROHAN; ROY, 2019).

Referências

CHAKRABORTY, D. D. S. An overview of face liveness detection. *International Journal on Information Theory*, v. 3, n. 2, p. 11–25, 2014. Citado na página 2.

FEDERAL BUREAL OF INVESTIGATION. *Spoofing and Phishing Spoofing and phishing are key parts of business email compromise scams.* 2021. <https:

//www.fbi.gov/how-we-can-help-you/scams-and-safety/common-frauds-and-scams/spoofing-and-phishing>. Accessed: 2025-12-15. Citado na página 2.

FLOREZ, R. et al. A real-time embedded system for driver drowsiness detection based on visual analysis of the eyes and mouth using convolutional neural network and mouth aspect ratio. *Sensors*, v. 24, n. 19, 2024. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/24/19/6261>. Citado na página 4.

HASAN, M.; ROHAN, T.; ROY, S. Efficient two stage approach to detect face liveness : Motion based and deep learning based. *4th International Conference on Electrical Information and Communication Technology (EICT)*, p. 1–6, 12 2019. Citado na página 8.

NISHANTH, D.; RAO, G. Liveness detection based on human eye blinking for photo attacks. *International Journal of Engineering and Advanced Technology*, v. 9, p. 4074–4077, 10 2019. Citado 2 vezes nas páginas 2 e 3.