

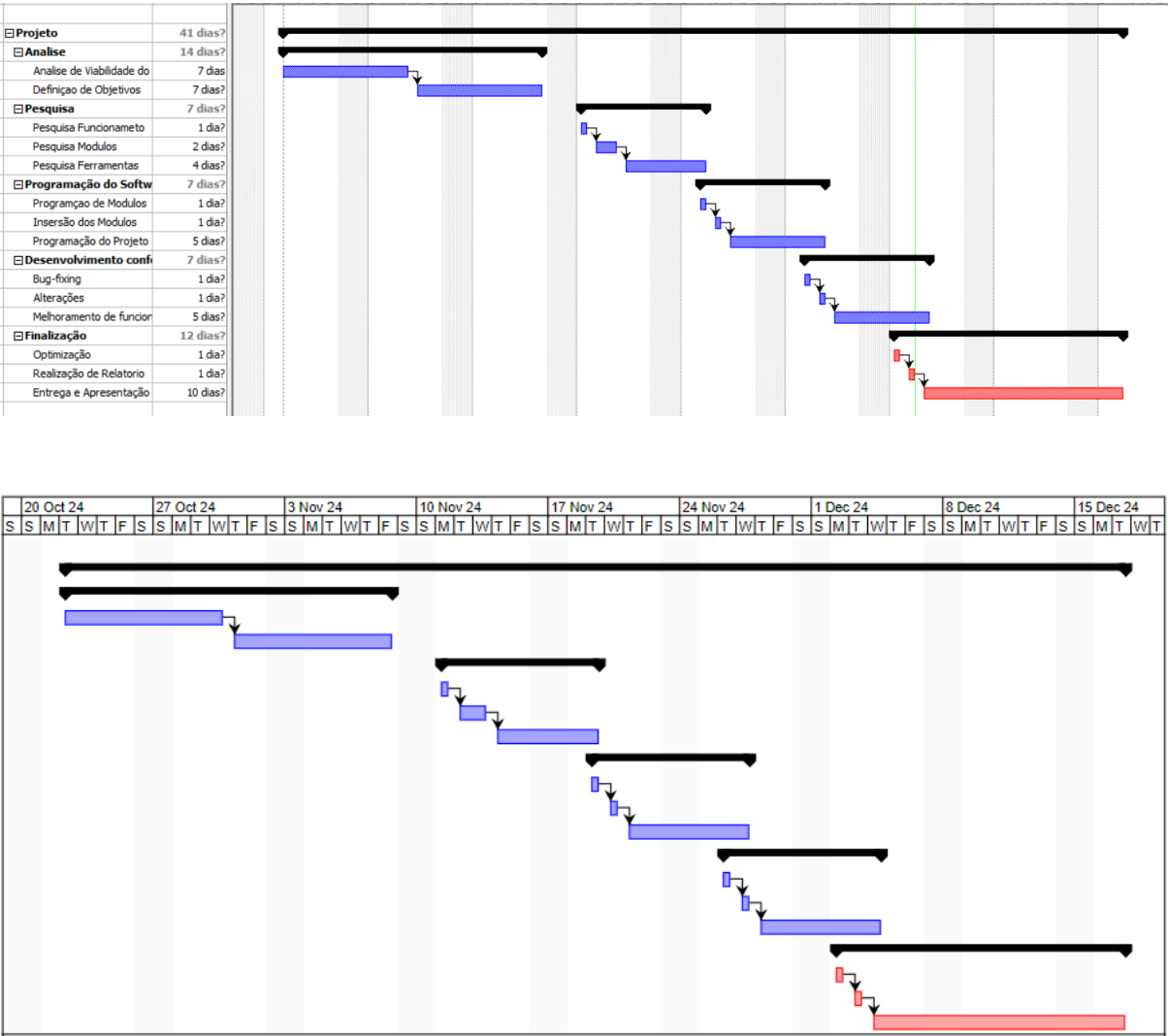
Implementação de um Sistema de Reconhecimento Facial com DeepFace e VGG-Face

João Paulo Vieira de Oliveira
2023108542

Cronograma do Planejamento do Projeto.....	3
Introdução.....	4
Metodologia de Implementação.....	5
Interface do Utilizador (Tkinter).....	5
Captura de Imagens em Tempo Real (OpenCV).....	5
Registo de Imagens na Base de Dados.....	5
Reconhecimento Facial com DeepFace.....	5
Introdução ao DeepFace.....	5
O Modelo VGG-Face.....	6
Implementação do Reconhecimento.....	6
Robustez e Tratamento de Erro.....	6
Arquitetura e Design do Sistema.....	6
Uso de Threads.....	6
Estrutura da Base de Dados Local.....	6
Resultados e Observações.....	7
Passo 1: Inicialização.....	7
Passo 2: Captura de Imagens.....	8
Passo 3: Reconhecimento Facial.....	9
Passo 4: Tratamento de Erros.....	10
Conclusão.....	11

Cronograma do Planejamento do Projeto

O planejamento do projeto é uma etapa essencial no início do desenvolvimento de sistemas de informação. A Figura 1 apresenta o cronograma projetado do projeto, detalhando as fases e suas respectivas atividades. O planejamento foi estruturado em cinco grandes etapas: Análise, Pesquisa, Programação do Software, Desenvolvimento com Correções e Finalização. A fase de Análise inclui a viabilidade do projeto e a definição dos objetivos, fornecendo uma base sólida para as etapas subsequentes. Em seguida, na etapa de Pesquisa, foram realizados estudos para compreender o funcionamento, os módulos necessários e as ferramentas a serem utilizadas. A etapa de Programação do Software concentra-se na criação e inserção de módulos. Após essa fase, o projeto entra em Desenvolvimento com Correções, onde são feitos ajustes, melhorias e validação de funcionalidades. Finalmente, a etapa de Finalização inclui otimizações, elaboração de relatórios e entrega/apresentação do produto final. Esse planejamento garante organização, cumprimento de prazos e alinhamento com os objetivos propostos.

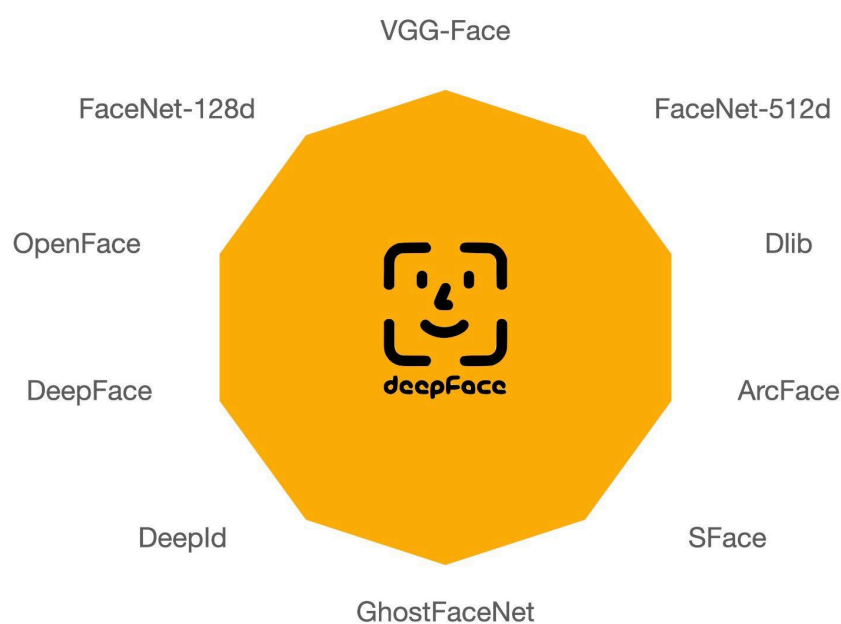


Introdução

O reconhecimento facial é uma das tecnologias mais avançadas da visão computacional, apresentando um vasto potencial em diversas áreas, como segurança, autenticação, monitoramento e interação humano-computador. Essa tecnologia, ao transformar características faciais em dados digitais, permite não apenas a identificação de indivíduos, mas também a análise emocional, a estimativa de idade e outros fatores contextuais. Este trabalho aborda o desenvolvimento de uma aplicação de reconhecimento facial utilizando a biblioteca DeepFace, em conjunto com o modelo VGG-Face, amplamente reconhecido por sua precisão em tarefas de identificação facial. A aplicação foi projetada para ser prática, eficiente e acessível, incorporando funcionalidades como captura de imagens em tempo real, registro local de dados faciais e identificação dinâmica, utilizando modelos pré-treinados.

O sistema foi implementado em Python, utilizando bibliotecas como Tkinter, para a interface gráfica do utilizador; OpenCV, para o processamento de vídeo e threading, para a execução paralela de tarefas. A escolha do DeepFace como motor de reconhecimento facial foi motivada por sua robustez e flexibilidade, além da compatibilidade com diversos modelos avançados, como o VGG-Face, o Facenet e outros. Este projeto exemplifica como é possível unir simplicidade de design com eficiência técnica, criando uma solução aplicável tanto em pequenos ambientes locais quanto em sistemas mais complexos e expansíveis.

Além disso, este trabalho não se limita à funcionalidade básica de reconhecimento facial, mas também explora métodos para garantir robustez, como tratamento de erros, organização estruturada de bases de dados e utilização de técnicas de multithreading para manter a responsividade do sistema.



Metodologia de Implementação

Interface do Utilizador (Tkinter)

A aplicação apresenta uma interface gráfica amigável, implementada com a biblioteca Tkinter. Esta interface serve como ponto de interação entre o utilizador e as funcionalidades principais do sistema. A janela principal inclui botões para iniciar a captura de imagens e iniciar o reconhecimento facial, além de instruções claras para orientar o utilizador.

Captura de Imagens em Tempo Real (OpenCV)

A captura de imagens é um componente essencial do sistema. Utilizou-se a biblioteca OpenCV para acessar a webcam e obter frames em tempo real. O sistema inicia a captura ao pressionar o botão “Iniciar Captura” na interface, abrindo uma nova janela onde o vídeo é exibido.

Durante a captura, o FPS (frames por segundo) é calculado dinamicamente para fornecer ao utilizador feedback sobre o desempenho do sistema. Este cálculo utiliza a diferença de tempo entre frames consecutivos para determinar a taxa de atualização da câmara. Uma sobreposição de texto no feed de vídeo exibe o valor do FPS, proporcionando informações úteis sobre a fluidez do processo.

Registo de Imagens na Base de Dados

Ao pressionar a tecla de espaço durante a captura, o sistema solicita ao utilizador que insira o seu nome através de uma janela de diálogo. O nome fornecido é validado e usado para criar um nome de ficheiro único, associado à imagem capturada. As imagens são armazenadas num diretório chamado "db", que funciona como a base de dados local do sistema.

Para garantir a qualidade das imagens capturadas, o sistema salva apenas frames válidos, minimizando a possibilidade de erros ou imagens corrompidas.

Reconhecimento Facial com DeepFace

Introdução ao DeepFace

O DeepFace é uma biblioteca de código aberto projetada para simplificar tarefas de análise facial. Ela oferece suporte a diversos modelos pré-treinados, como VGG-Face, Facenet, OpenFace, e outros, permitindo que desenvolvedores escolham o modelo mais adequado para suas necessidades. Além disso, o DeepFace suporta análise emocional, estimativa de idade e gênero, ampliando suas capacidades além do reconhecimento facial.

O Modelo VGG-Face

O VGG-Face é um modelo de rede neural convolucional amplamente utilizado no reconhecimento facial. Foi desenvolvido pela Universidade de Oxford e é baseado na arquitetura VGG, conhecida pela sua profundidade e simplicidade estrutural. O modelo foi treinado em milhões de imagens faciais, permitindo uma precisão elevada na identificação de indivíduos.

Na nossa aplicação, o VGG-Face foi escolhido devido à sua eficácia e compatibilidade com a biblioteca DeepFace. Este modelo realiza a extração de características faciais, mapeando-as para um espaço de características que permite comparar imagens com alta precisão.

Implementação do Reconhecimento

O reconhecimento facial é iniciado ao pressionar o botão "Iniciar Reconhecimento (DeepFace)" na interface. O sistema utiliza a função `DeepFace.stream`, que lê as imagens armazenadas no diretório "db" e inicia um fluxo em tempo real para identificar rostos capturados pela câmara.

Durante este processo, o DeepFace compara as características faciais extraídas dos frames ao vivo com aquelas registradas na base de dados. Quando uma correspondência é encontrada, o sistema identifica o indivíduo com base nas imagens previamente armazenadas.

Robustez e Tratamento de Erro

Para lidar com potenciais falhas, como imagens mal formadas ou erros de processamento, foi implementado um sistema de captura de exceções.

Arquitetura e Design do Sistema

Uso de Threads

Para garantir que a interface do utilizador permaneça responsiva durante tarefas intensivas, como captura de vídeo ou reconhecimento facial, o sistema utiliza threads. A execução em paralelo permite que as tarefas de processamento sejam realizadas sem interferir na interação do utilizador com a interface gráfica.

Estrutura da Base de Dados Local

A base de dados local foi projetada para ser simples e eficiente. As imagens são organizadas por nomes de utilizadores, garantindo um acesso fácil durante o reconhecimento. Esta estrutura pode ser facilmente expandida ou integrada a sistemas mais complexos, como bases de dados centralizadas.

Resultados e Observações

Passo 1: Inicialização

```
class FacialRecognitionApp:
    def __init__(self):
        self.root = Tk()
        self.root.title("Reconhecimento Facial com DeepFace")
        self.root.geometry("300x200")

        self.captura_status = False
        self.create_ui()

    def create_ui(self):
        Button(self.root, text="Iniciar Captura", command=self.start_capture).pack(pady=10)
        Label(self.root, text="Pressione 'Iniciar Captura' para começar\nPressione 'Esc' para sair").pack(pady=10)
        Button(self.root, text="Iniciar Reconhecimento (DeepFace)", command=self.start_recognition).pack(pady=10)

    def start_capture(self):
        self.captura_status = True
        capture_thread = Thread(target=self.capture_frames)
        capture_thread.start()

    def ask_user_name(self, frame, img_counter):
        user_name = simpledialog.askstring("Input", "Nome:", parent=self.root)
        if user_name:
            sanitized_name = "".join(c for c in user_name if c.isalnum())
            os.makedirs("db", exist_ok=True)
            img_name = f"db/{sanitized_name}_{img_counter}.png"
            if frame is not None:
                cv2.imwrite(img_name, frame)
                print(f"{img_name} salvo!")
            else:
                print("Frame inválido. Não foi possível salvar.")
```

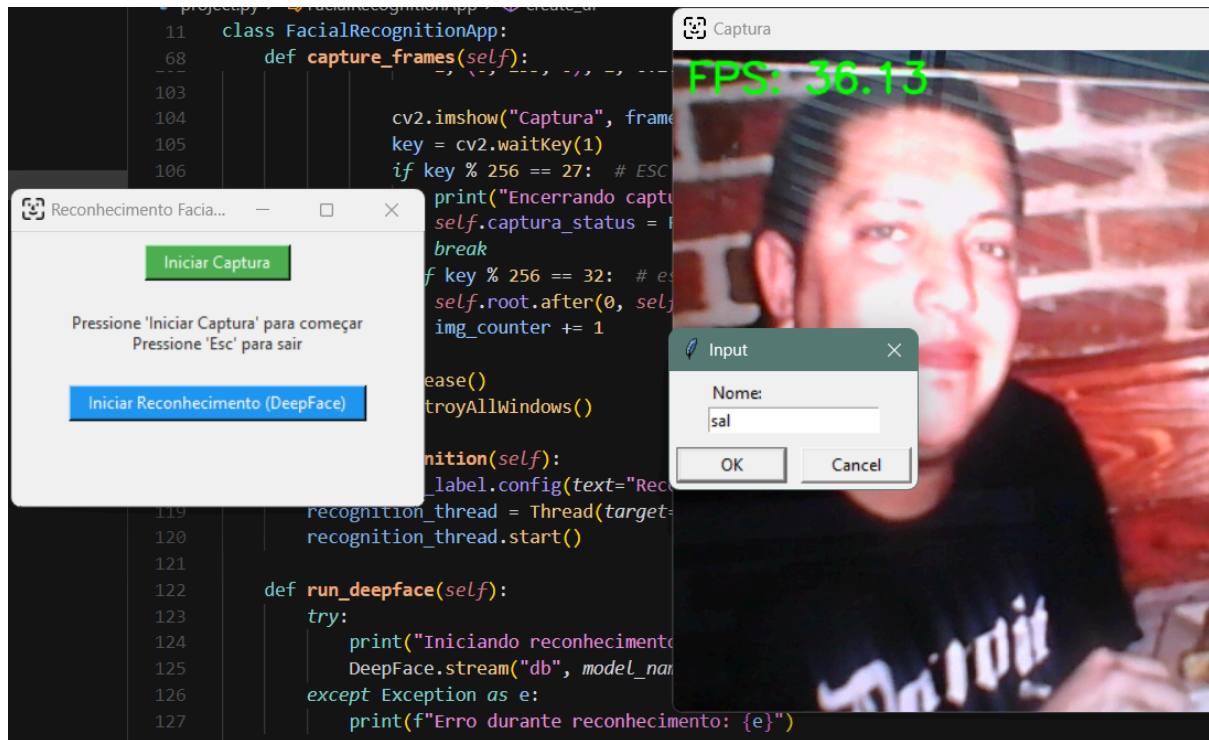
1. Ao executar o sistema, a interface gráfica desenvolvida em Tkinter é exibida na tela.
2. A interface foi projetada para ser simples e intuitiva, oferecendo opções claras para o utilizador interagir com o sistema.
3. As duas principais funcionalidades – Captura de Imagens e Reconhecimento Facial – são apresentadas por meio de botões:
4. “Iniciar Captura”: Ativa a webcam para salvar imagens.
5. “Iniciar Reconhecimento (DeepFace)” Inicia o módulo de reconhecimento facial.

Além disso, mensagens orientativas na interface guiam o utilizador sobre como proceder, tornando o sistema acessível mesmo para aqueles sem conhecimento técnico avançado. Esse passo é crucial, pois serve como o ponto de entrada para todas as funcionalidades do sistema.

Passo 2: Captura de Imagens

Quando o utilizador clica em “Iniciar Captura”, o sistema utiliza a biblioteca OpenCV para ativar a webcam. A partir desse momento, um stream de vídeo em tempo real é exibido na interface gráfica.

O FPS (Frames per Second) é calculado dinamicamente e mostrado no canto da janela. Isso permite que o utilizador tenha noção do desempenho e da fluidez da captura.



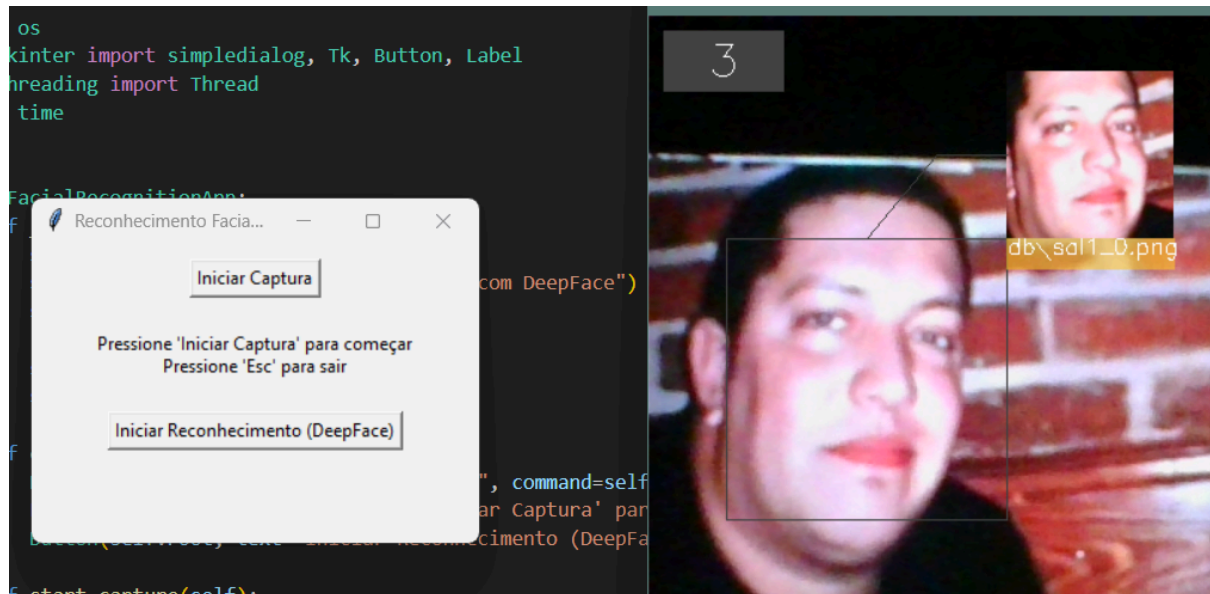
Para salvar uma imagem:

1. O utilizador pressiona a tecla “Espaço”.
2. O sistema exibe um campo para que o utilizador forneça um nome para a imagem capturada.
3. O nome inserido é validado (para evitar entradas inválidas ou vazias).
4. A imagem é salva no diretório "db" com um nome único correspondente ao fornecido pelo utilizador.

Essa abordagem permite que as imagens sejam organizadas de maneira estruturada e facilmente recuperadas posteriormente durante o processo de reconhecimento facial.

Passo 3: Reconhecimento Facial

Ao clicar em “Iniciar Reconhecimento (DeepFace)”, o sistema ativa o módulo de reconhecimento facial utilizando a função DeepFace.stream.



Neste processo:

1. A ferramenta DeepFace lê as imagens armazenadas no diretório “db”, que servem como base de dados de rostos registrados.
2. A webcam continua capturando o stream de vídeo em tempo real.
3. Para cada rosto capturado, o modelo VGG-Face realiza a comparação com as imagens da base de dados.
4. O DeepFace calcula a similaridade entre as características faciais do rosto detectado e das imagens registradas.
5. Quando uma correspondência é encontrada:
6. O sistema exibe na tela o nome do utilizador correspondente ao rosto identificado.
7. Esse nome é recuperado diretamente da imagem armazenada, mantendo consistência com o fornecido pelo utilizador no momento da captura.

Essa funcionalidade demonstra a eficácia do modelo VGG-Face na identificação de rostos, mesmo em condições variáveis, como diferenças de iluminação ou ângulos da face.

Passo 4: Tratamento de Erros

Durante a execução do sistema, possíveis erros e exceções são monitorados e tratados para evitar falhas críticas:

1. Caso a webcam não esteja disponível ou falhe ao ser ativada, o sistema emite uma mensagem de erro informando o problema ao utilizador.
2. Durante a captura ou salvamento de imagens, o sistema verifica a validade dos dados inseridos, impedindo que arquivos com nomes inválidos ou duplicados sejam gerados.
3. No processo de reconhecimento facial, caso a base de dados esteja vazia ou nenhuma correspondência seja encontrada, o sistema informa o utilizador de forma clara e amigável.

O tratamento dessas exceções garante que o sistema mantenha sua robustez e não interrompa a execução inesperadamente.

Esse cuidado com o tratamento de erros torna a aplicação mais estável, confiável e adaptada a cenários reais de uso, onde imprevistos podem ocorrer.

Observações

O sistema desenvolvido demonstrou ser uma solução prática e eficaz para tarefas de reconhecimento facial. A interface amigável, combinada com a precisão do modelo VGG-Face, permite identificar indivíduos com alta confiabilidade. O uso de bibliotecas modernas, como DeepFace e OpenCV, destaca a viabilidade de criar sistemas complexos com ferramentas acessíveis.

No futuro, o sistema pode ser expandido para incluir funcionalidades adicionais, como: Reconhecimento emocional; Estimativa de idade e gênero; Integração com bases de dados externas ou serviços na nuvem para maior escalabilidade.

Adicionalmente, melhorias no desempenho, como a otimização do cálculo de FPS ou o uso de hardware especializado (GPU), podem aumentar ainda mais a eficiência do sistema.

Conclusão

Este projeto destacou a implementação de um sistema completo e funcional de reconhecimento facial, abrangendo desde a captura de imagens até a identificação em tempo real de indivíduos. Utilizando o modelo VGG-Face e a biblioteca DeepFace, o sistema mostrou-se altamente eficaz, combinando precisão técnica com uma interface amigável para o utilizador. Essa integração de ferramentas modernas demonstra como é possível criar aplicações avançadas de visão computacional de forma acessível e funcional.

A abordagem modular do sistema permite não apenas uma operação eficiente, mas também a possibilidade de expansões futuras. Entre as melhorias e funcionalidades a serem exploradas, destacam-se a inclusão de análise emocional, estimativa de idade e género, e a integração com bases de dados na nuvem para escalabilidade. Adicionalmente, o uso de hardware especializado, como GPUs, pode melhorar significativamente o desempenho em aplicações mais exigentes.

Por fim, o trabalho ilustra o poder das tecnologias de inteligência artificial quando combinadas com ferramentas modernas de desenvolvimento de software. O reconhecimento facial, além de sua aplicação prática no sistema proposto, abre portas para novos avanços em segurança, automação e interatividade, representando um marco no desenvolvimento de soluções tecnológicas para o futuro. O sistema desenvolvido é um exemplo prático de como as inovações podem ser adaptadas e aplicadas para atender a necessidades diversas, consolidando-se como um recurso essencial no cenário da visão computacional moderna.