Universidade da Beira Interior
Departamento de Informática

Human Recognition in Surveillance Settings

Elaborado por:

Ruben Abadesso, 43664

Orientador:

Professor Doutor Hugo Pedro Martins Carriço Proença

24 de junho de 2024

Agradecimentos

A conclusão deste trabalho, bem como da grande maior parte da minha vida académica não seria possível sem a ajuda de todas as pessoas, que tenho o prazer de poder chamar amigos. Um agradecimento especial para todos aqueles que se disponibilizaram para dar a sua própria imagem, para a coleção de dados, do qual este projeto se fundamentou.

Agradeço também ao professor orientador, que se disponibilizou para ajudar e acompanhar toda a evolução do projeto.

Conteúdo

Conteúdo **iii**

Lista de Figuras **v**

1 Introdução **1**
1.1 Enquadramento 1
1.2 Motivação 1
1.3 Objetivos 2
1.4 Organização do Documento 2

2 Estado da Arte **5**
2.1 Introdução 5
2.2 Reconhecimento em vídeo-vigilância 5
2.2.1 Desafios 6
2.2.2 Métodos 6
2.3 U-Net 6
2.4 *Conditional Generative Adversarial Network* (cGAN) 7
2.5 Pix2Pix 8
2.6 Trabalhos Relacionados 10
2.7 Conclusões 11

3 Métodos **13**
3.1 Introdução 13
3.2 Dados 13
3.2.1 Classificações de dados 15
3.2.2 Estatísticas de dados 16
3.3 Treino 17
3.3.1 1º Abordagem - U-Net 17
3.3.2 2º Abordagem - Pix2pix 17
3.3.3 3º Abordagem - Pix2pix com headpose 18
3.4 Tecnologias e Ferramentas Utilizadas 18
3.4.1 PyCharm & DataSpell 18
3.4.2 TensorFlow 19

3.4.3 Keras 19
3.5 Hardware 19
3.6 Conclusões 19

4 Testes e Resultados Obtidos **21**
4.1 Introdução 21
4.2 1º Abordagem - U-Net 21
4.3 2º Abordagem - Pix2pix 22
4.4 3º Abordagem - Pix2pix com *headpose* 24
4.5 Comparação da 2º e 3º Abordagem 26
4.6 Conclusões 26

5 Conclusões e Trabalho Futuro **27**
5.1 Conclusões Principais 27
5.2 Trabalho Futuro 27

Bibliografia **29**

Lista de Figuras

1.1 Condições típicas em ambientes de vigilância, onde o reconhecimento enfrenta graves problemas devido à má qualidade dos dados 2

2.1 Diagrama da Arquitetura U-Net 7
2.2 Diagrama da uma cGAN 8
2.3 Treino da cGAN Pix2pix 9
2.4 Processo de treino do gerador Pix2pix 9
2.5 Processo de treino do discriminador Pix2pix 10

3.1 Imagens sincronizadas de indivíduos em ambientes não controlados (esquerda) e ambientes controlados (direita) 14
3.2 Exemplo das variáveis de pose 15
3.3 Representação das imagens com a respetiva pose mapeada numa imagem RGB 15
3.4 Distribuição dos dados por género e idade 16
3.5 Distribuição da pose da cabeça - Yaw 16
3.6 Distribuição da pose da cabeça - Pitch 16
3.7 Distribuição da pose da cabeça - Roll 17

4.1 Resultados da 1º Abordagem (1) 21
4.2 Resultados da 1º Abordagem (2) 22
4.3 Resultados da 1º Abordagem (3) 22
4.4 Resultados da 2º Abordagem (1) 22
4.5 Resultados da 2º Abordagem (2) 23
4.6 Resultados da 2º Abordagem (3) 23
4.7 Resultados da 2º Abordagem (4) 23
4.8 Resultados da 2º Abordagem (5) 23
4.9 Loss do Gerador da 2º Abordagem 24
4.10 Resultados da 3º Abordagem (1) 24
4.11 Resultados da 3º Abordagem (2) 24
4.12 Resultados da 3º Abordagem (3) 25
4.13 Resultados da 3º Abordagem (4) 25
4.14 Resultados da 3º Abordagem (5) 25
4.15 Loss do Gerador da 3º Abordagem 25

4.16 Comparação da Loss do Gerador da 2º e 3º Abordagem 26

Acrónimos

API *Application Programming Interface*

cGAN *Conditional Generative Adversarial Network*

DL *Deep Learning*

GAN *Generative Adversarial Network*

IDE *Integrated Development Environment*

ML *Machine Learning*

UBI Universidade da Beira Interior

Capítulo

1

Introdução

1.1 Enquadramento

Este relatório tem como tema, *"Human Recognition in Surveillance Settings"*. O reconhecimento de humanos em ambientes de vigilância está entre os mais importantes desafios da inteligência artificial, devido à ampla gama de aplicações (segurança, proteção, criminalidade, etc.). Neste projeto irá ser desenvolvido *software* capaz de reconhecer humanos em ambientes onde a qualidade de imagem capturadas não seja boa.

Este relatório foi feito no contexto da unidade curricular de Projeto de licenciatura em Engenharia Informática na Universidade da Beira Interior (UBI).

1.2 Motivação

Em contexto de ambientes de vigilância, assumindo que os sujeitos, não têm conhecimento do processo de aquisição de dados, espera-se que os dados recolhidos tenham uma qualidade muito fraca, não só em termos de resolução e iluminação, mas também em termos de pose e presença de oclusões.

Assim, existem numerosos esforços a ser desenvolvidos, para desenvolver modelos capazes de reconhecer seres humanos neste tipo de condições, ou seja, utilizando dados de qualidade extremamente baixa. Entre as muitas dificuldades que surgem neste cenário, um dos problemas é a inexistência de informações sólidas sobre as variações reais nos dados coletados, por exemplo, distância, pose, resolução e iluminação.

De forma a contornar estas advertências, pretende-se criar um conjunto de dados, e desenvolver um modelo capaz de reconhecer humanos em ambientes com baixa qualidade.

Figura 1.1: Condições típicas em ambientes de vigilância, onde o reconhecimento enfrenta graves problemas devido à má qualidade dos dados

1.3 Objetivos

Este projeto tem como objetivo coletar um conjunto de dados sincronizados, coletados simultaneamente em ambientes controlados e não controlados.

A ideia é obter um modelo de inteligência artificial, que seja capaz de equiparar dados entre os dois domínios, o que será de interesse óbvio para futuros processos de reconhecimento.

Ao utilizar este modelo, será possível dar-lhe uma imagem de baixa qualidade de um sujeito (em condições de vigilância) e transformá-la para o domínio de "alta qualidade", onde o reconhecimento será facilmente realizado.

1.4 Organização do Documento

De modo a refletir o trabalho que foi feito, este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo – **Introdução** – iniciado na página 1, refere-se ao enquadramento e motivação para o projeto, os seus objetivos e a respetiva organização do documento.

2. O segundo capítulo – **Estado da Arte** – iniciado na página 5, descreve os conceitos mais importantes no âmbito deste projeto, bem como os métodos estado da arte.

3. O terceiro capítulo – **Métodos** – iniciado na página 13, retrata as diferentes tecnologias e ferramentas utilizadas no desenvolvimento do projeto, bem como todos os dados e procedimentos aplicados.

4. O quarto capítulo – **Teste e Resultados Obtidos** – iniciado na página 21, detalha os diversos resultados obtidos após a implementação e sua comparação e interpretação em termos visuais.

5. O ultimo capítulo – **Conclusões e Trabalho Futuro** – iniciado na página 27, expõe as principais conclusões obtidas na realização do projeto.

Capítulo

2

Estado da Arte

2.1 Introdução

Para entender as técnicas e métodos abordados ao longo da realização desde projeto, neste capítulo serão discutidos os principais conceitos e metodologias, como *Conditional Generative Adversarial Networks (cGANs)* e as arquiteturas *U-Net* e *Pix2Pix*, que formam a base para o desenvolvimento de soluções inovadoras nesta área. Através da análise dos trabalhos relacionados, será possível comparar abordagens e resultados alcançados, e identificar formas de melhorar o protótipo.

2.2 Reconhecimento em vídeo-vigilância

O reconhecimento facial é um problema de investigação bem estabelecido no domínio da visão computacional, com o objetivo de reconhecer identidades humanas através de imagens faciais. O reconhecimento facial é reconhecido como uma das ferramentas mais importantes para uma grande variedade de aplicações de identidade, desde a cumprimento da lei, segurança de informação, negócios, entretenimento e comércio eletrónico.

Uma das forças por detrás do sucesso do reconhecimento facial são os rápidos avanços nas técnicas de *Machine Learning* (ML) e dispositivos de computação poderosos. Nas principais técnicas, a precisão do reconhecimento facial em imagens de boa qualidade atingiu um nível sem precedentes graças à ML. No entanto, o sucesso destes métodos não se adapta aos dados com imagens faciais de baixa resolução capturados em vídeos de vigilância em ambientes não controlados.

2.2.1 Desafios

A baixa qualidade e resolução de imagens de vídeo faz com que a precisão do reconhecimento diminua drasticamente. É frequente os indivíduos se moverem, afastando-se da vista da câmara, aparecendo em muitos *frames*. Não olham diretamente para a câmara, o que diminui o número de amostras de rostos frontais e muitas destas amostras de rostos podem não ser adequadas para o reconhecimento de rostos. Nos vídeos de vigilância, o controlo limitado das condições de captura, como a variação das poses, das expressões, da iluminação, a cooperação dos indivíduos, a oclusão ou a desfocagem do movimento, são alguns dos fatores que dificultam o reconhecimento de rostos nos vídeos de vigilância.

2.2.2 Métodos

Além de todos os desafios que já foram mencionados, sobre o reconhecimento facial em ambientes de vídeo-vigilância, é de importante perceber, que um computador apenas consegue reconhecer ou identificar, movimentos, pessoas, objetos, etc, quando já foi treinado com esse objetivo.

Reconhecimento facial de um individuo, apenas é possível se o computador souber à priori quem é a pessoa a identificar. Esses métodos são denotados como aprendizagem supervisionada.

Com isso, neste relatório é proposto outro método reconhecimento facial, não supervisionado. Onde ao receber imagens de má qualidade de indivíduos, irá recriar uma imagem de boa qualidade do respetivo indivíduo.

2.3 U-Net

U-Net é uma arquitetura de *Deep Learning (DL)* que foi introduzida pela primeira vez no artigo *"U-Net: Convolutional Networks for Biomedical Image Segmentation"*[1]. O objetivo principal desta arquitetura era enfrentar o desafio da limitação de dados anotados no campo médico. Essa rede foi projetada para aproveitar uma quantidade menor de dados, mantendo a velocidade e a precisão.

A arquitetura *U-Net* é única na medida em que consiste num caminho de contração e num caminho de expansão. O caminho de contração contém camadas de codificação que captam informações contextuais e reduzem a resolução espacial da entrada, enquanto que o caminho de expansão contém camadas de descodificação que descodificam os dados codificados e utilizam as informações do caminho de contração através de ligações de salto para gerar um mapa de segmentação.

O caminho de contração na *U-Net* é responsável pela identificação das características relevantes na imagem de entrada. As camadas de codificação executam operações convolucionais que reduzem a resolução espacial dos mapas de características ao mesmo tempo que aumentam a sua profundidade, captando assim representações cada vez mais abstratas da entrada.

Por outro lado, o caminho de expansão trabalha na descodificação dos dados codificados e na localização das características, mantendo a resolução espacial da entrada. As camadas do descodificador no caminho de expansão aumentam a amostragem dos mapas de características, ao mesmo tempo que efetuam operações convolucionais. As ligações de salto do caminho de contração ajudam a preservar a informação espacial perdida, o que ajuda as camadas de descodificação a localizar as características com maior precisão.

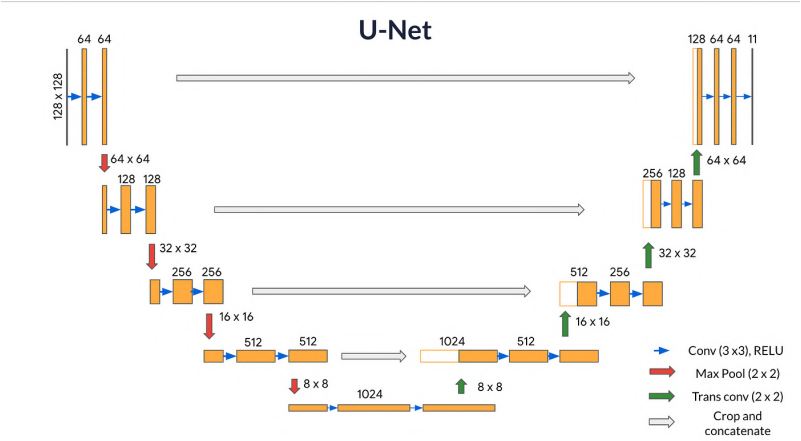

Figura 2.1: Diagrama da Arquitetura U-Net

2.4 cGAN

cGAN é um tipo de *Generative Adversarial Network (GAN)* em que o processo de geração é condicionado por informações adicionais. Esse condicionamento pode ser qualquer tipo de informação auxiliar, como classes, imagens ou texto.

GAN é uma estrutura de *DL* que é utilizada para gerar exemplos aleatórios e plausíveis com base nas nossas necessidades. Contém duas partes essenciais que estão sempre a competir entre si num processo repetitivo (como adversários):

- Rede Geradora: É a rede neuronal responsável pela criação de novos dados. Podem ser sob a forma de imagem, texto, vídeo, som, etc., consoante os dados com que são treinados.

- Rede Discriminadora: O seu trabalho consiste em distinguir entre dados reais e falsos (do conjunto de dados e os dados gerados pelo gerador).

O objetivo do gerador é criar novos dados suficientemente reais para "enganar"o discriminador de modo a que este não consiga distinguir entre dados reais e falsos (gerados), enquanto que o papel do discriminador é ser capaz de identificar se os dados são gerados ou reais.

Se quisermos que a *GAN* gere dados de um tipo especifico ou aprenda a distinguir classes diferentes, podemos fornecer ao modelo uma condição especifica.

As *cGANs* funcionam da mesma forma que os *GANs*. A geração de dados numa *GAN* é condicionada por informações de entrada específicas, que podem ser etiquetas, informações de classe ou quaisquer outras características relevantes. Este condicionamento permite uma geração de dados mais precisa e direcionada.

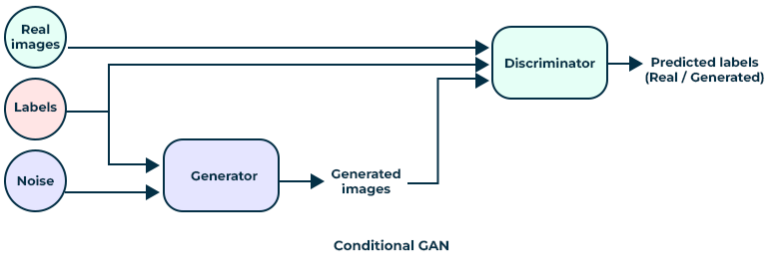

Figura 2.2: Diagrama da uma cGAN

2.5 Pix2Pix

Pix2Pix é uma *cGAN* que aprende a mapear imagens de entrada para imagens de saída, tal como descrito no artigo *"Image-to-Image Translation with Conditional Adversarial Networks"*[2]. O *Pix2pix* foi criado para uma tarefa especifica, pode ser aplicado a uma vasta gama de tarefas, incluindo a síntese

de fotografias a partir de mapas de etiquetas, a geração de fotografias coloridas a partir de imagens a preto e branco, a transformação de fotografias do *Google Maps* em imagens aéreas e até a transformação de esboços em fotografias.

A arquitetura desta rede contém:

- Um gerador baseado na arquitetura *U-Net*.
- Um discriminador representado por um classificador convolucional *PatchGAN*.

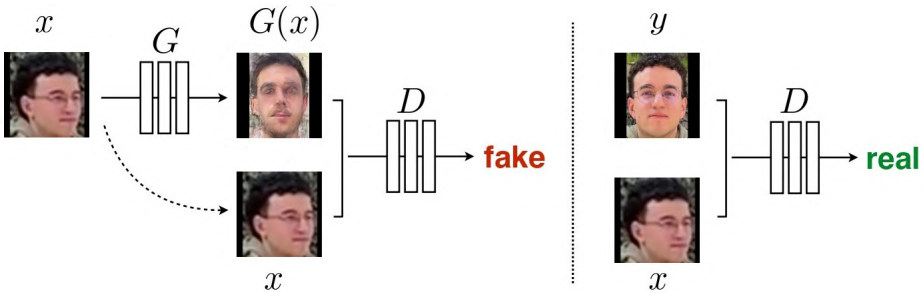

Figura 2.3: Treino da cGAN Pix2pix

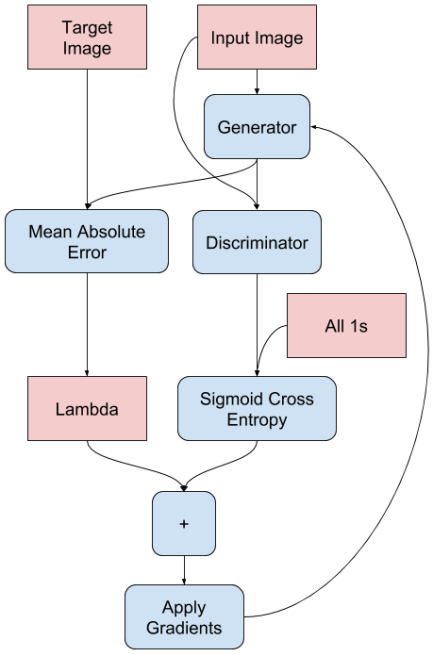

Figura 2.4: Processo de treino do gerador Pix2pix

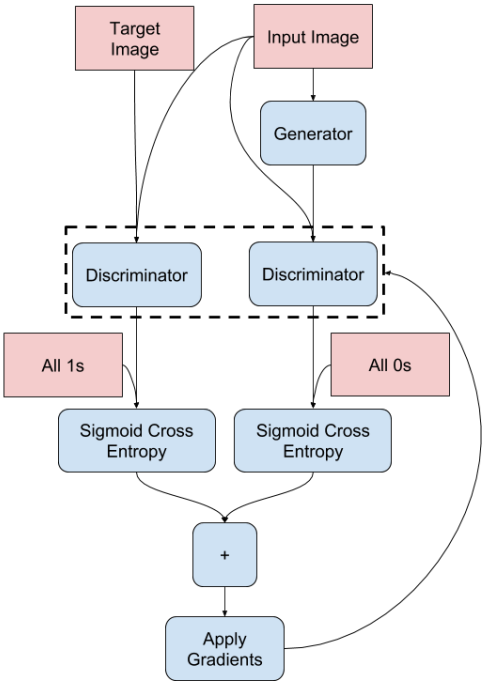

Figura 2.5: Processo de treino do discriminador Pix2pix

O processo de treino do gerador e discriminador estão representados nas figuras 2.4 e 2.5 respetivamente:

Este será a base do modelo final, que foi criado com o propósito de criar imagens de boa qualidade de um individuo, depois de receber um imagem de má qualidade.

2.6 Trabalhos Relacionados

No decorrer do segundo semestre do ano letivo de 2023/24 este projeto foi também desenvolvido por alunos de mestrado em Engenharia Informática na unidade curricular de Computação Visual.

Com as várias abordagens e soluções que foram apresentadas pelos diversos alunos, podemos distinguir o protótipo a seguir apresentado, pelas seguintes qualidades:

- Escolha detalhada dos dados a usar - Com múltiplos alunos a recolher dados para a realização do trabalho, alguns dos dados capturados, não ficaram aceitáveis, de forma a serem retirados do treino do protótipo final.

- Garantia de boa divisão de dados - Podemos garantir que os resultados apresentados pelos dados de teste, nunca foram apresentados ao gerador, de forma a evitar qualquer tipo de viés.
- Estudo e implementação de diferentes abordagens - Ao longo da realização do mesmo, foram exploradas diferentes formas possíveis na criação do prototipo, que serão apresentadas a seguir.
- Meticuloso ajuste de parametrizações - os modelos de geração e discriminação formam estudados e ajustados de forma a se adaptar aos dados.

Um aspeto negativo no protótipo apresentado, será a falta de tratamento dos dados, que permitiria remover o plano de fundo dos indivíduos, para que o modelo se pudesse focar mais nas características faciais. Além disso também não foram usados informações de classe dos indivíduos, tal como género, idade, cor de pele e acessórios.

2.7 Conclusões

Neste capitulo formam descritos os principais tópicos de *DL* que permitiram a realização deste projeto, tal como a estrutura *cGAN* e as arquiteturas *U-Net* e *Pix2pix*. Foram também comparados vantagens e desvantagens com trabalhos relacionados com o tema desenvolvido.

Capítulo

3

Métodos

3.1 Introdução

As primeiras etapas deste projeto envolvem a coleção de dados, bem como o seu tratamento. Visto que a foi recolhido uma vasta quantidade de dados, 80% do tempo utilizado para a realização deste projeto, foi usado para essa tarefa.

Com os dados normalizados, resta então treinar um modelo, para que seja capaz de gerar imagens de faces de indivíduos com boa qualidade, depois de lhe ser atribuído uma imagem de baixa qualidade.

Na secção 3.2 encontra-se a descrição dos dados usados, como foram capturados, a sua organização e classificação. A secção 3.3 descreve as abordagens e o processo de treino dos modelos. A secção 3.4 contém as tecnologias e ferramentas utilizadas no contexto do projeto. A secção 3.5 contém a descrição dos principais componentes da máquina onde foram treinados os modelos.

3.2 Dados

É importante voltar a referir que este projeto tem como objetivo coletar um conjunto de dados sincronizados, coletados simultaneamente em ambientes controlados e não controlados.

Se os nossos dados vão ser faces de indivíduos, como pode uma pessoa estar simultaneamente num ambiente controlado e não controlado?

Como se trata da captura de fotografias da face de um individuo, em ambiente de boa e má qualidade, optou-se por usar duas câmaras. Uma câmara é colocada em frente ao individuo, para capturar a face com a maior qualidade

de imagem possível. A segunda câmara, de forma a representar um ambiente de má qualidade, capturou o individuo entre uma distancia de 2 a 15 metros.

Os fatores que influenciaram para que estas imagens se classifiquem como sendo de má qualidade são, por exemplo, variação de luz, distancia, foco, vibração, sombra, presença de ocultações, etc.

Para responder à questão de sincronização, os dados foram capturados com as câmaras fotográficas de *smartphones*, que permitiram o uso de uma aplicação que coloca nas fotografias, o *timestamp* do momento da captura.

No total, foram capturados 159 indivíduos, cada um em 2 sessões distintas, realizadas em dias e condição diferentes. No total foram somadas mais de 190 mil imagens, com um peso de mais de 95 GBytes. No entanto, visto que apenas iremos utilizar imagens da face dos indivíduos, conseguimos reduzir para 1.5 GBytes o espaço ocupado pelos dados.

Depois de feita a normalização dos dados, o que obtemos é as imagens representadas na figura 3.1.

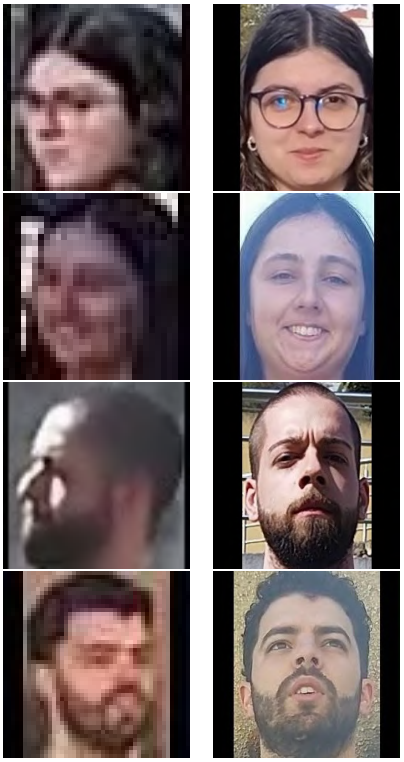

Figura 3.1: Imagens sincronizadas de indivíduos em ambientes não controlados (esquerda) e ambientes controlados (direita)

3.2.1 Classificações de dados

Depois da obtenção dos dados, estes foram classificados em múltiplas classes para permitir uma melhor gestão e melhor distinção, no momento da geração e discriminação de imagens. Algumas dessas classes são: género, idade, cor de pele, uso de óculos, presença de outros acessórios.

Além disso, através de um modelo, disponível online, de estimação de pose da cabeça de um individuo, em relação à câmara, foi possível classificar o individuo com *pitch*, *yaw* e *roll* como mostra a figura 3.2. Estes valores foram posteriormente mapeados para uma imagem tridimensional, em que cada dimensão corresponde a um valor da orientação, para a respetiva imagem de referencia. Estes novos dados foram guardados em disco para poderem ser usados posteriormente no treino do modelo.

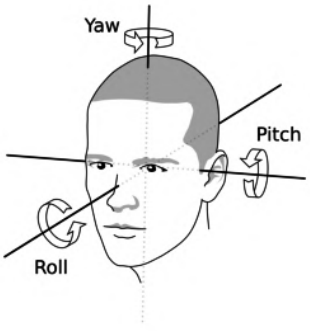

Figura 3.2: Exemplo das variáveis de pose

Figura 3.3: Representação das imagens com a respetiva pose mapeada numa imagem RGB

Aquando do processo de leitura de dados, imagens com o formato $128 \times 256 \times 3$, irão ser divididas em duas e sobrepostas formando uma imagem com formato $128 \times 128 \times 6$, que será usada para treinar o modelo.

3.2.2 Estatísticas de dados

De forma a entender como os dados estão organizados, vamos ver algumas das principais estatísticas da distribuição de dados.

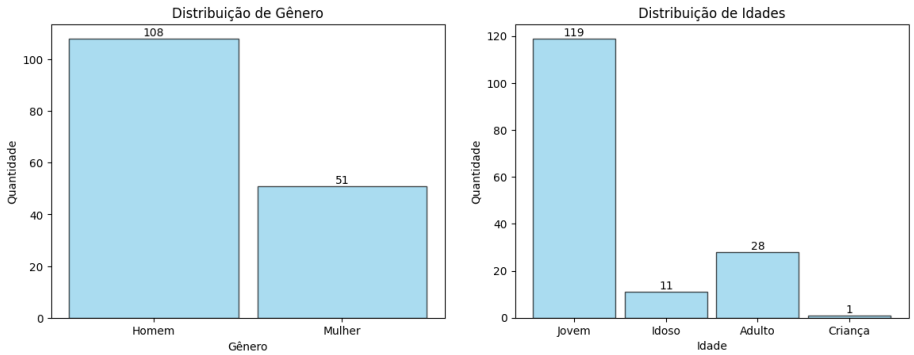

Figura 3.4: Distribuição dos dados por género e idade

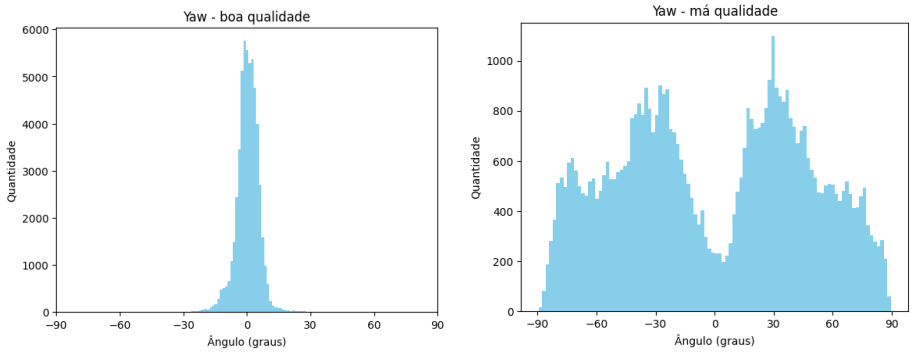

Figura 3.5: Distribuição da pose da cabeça - Yaw

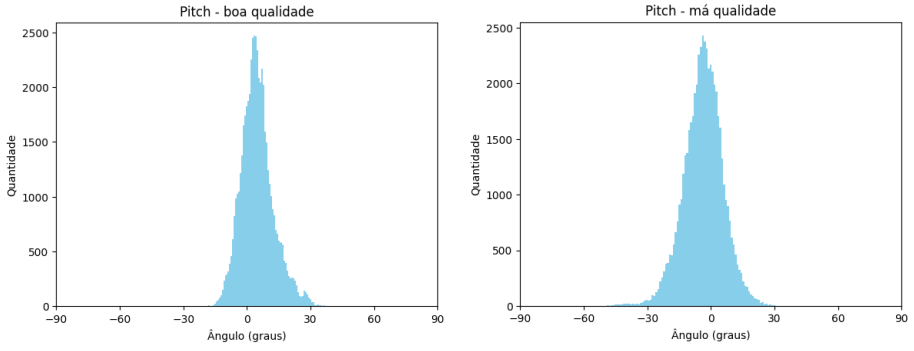

Figura 3.6: Distribuição da pose da cabeça - Pitch

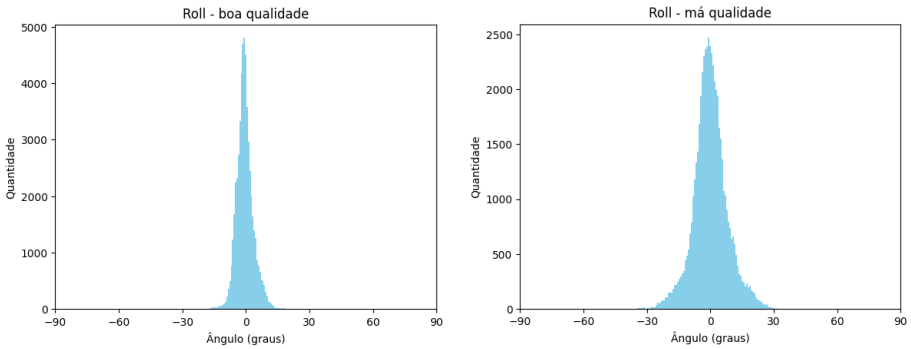

Figura 3.7: Distribuição da pose da cabeça - Roll

3.3 Treino

O processo de treino está dividido em 3 secções, equivalentes a cada uma das 3 abordagens tomadas. Todos os ficheiros de código estão disponíveis *online*. [3]

3.3.1 1º Abordagem - U-Net

Numa primeira tentativa de responder ao nosso problema de geração de imagem, foi abordada a solução mais simples, que se baseia num *Encoder-Decoder*, baseada na arquitetura U-Net, referida na secção 2.3 deste relatório.

Este modelo recebe como dados de entrada, imagens como mostrado na figura 3.1. Uma imagem de entrada (esquerda) e a respetiva imagem *ground truth* (direita).

Nesta tentativa foi usado a base de dados completa (com todos os indivíduos capturados):

- Treino - 105 indivíduos, contendo 62526 pares de imagens
- Validação - 46 indivíduos, contendo 27508 pares de imagens
- Teste - 8 indivíduos, contendo 4769 pares de imagens

O processo de treino foi realizado em 1 *epoch*, que demorou pouco mais de 50 minutos.

3.3.2 2º Abordagem - Pix2pix

Nesta fase, foi usado como base a arquitetura *Pix2Pix*, referida na secção 2.5 deste relatório.

Este modelo recebe como dados de entrada, imagens como mostrado na figura 3.1. Uma imagem de entrada (esquerda) e a imagem *ground truth* (direita).

Nesta tentativa foi usado a base de dados completa (com todos os indivíduos capturados):

- Treino - 105 indivíduos, contendo 61672 pares de imagens
- Validação - 46 indivíduos, contendo 27731 pares de imagens
- Teste - 8 indivíduos, contendo 4800 pares de imagens

O processo de treino foi realizado em 3 *epochs*, que demorou pouco mais de 4h horas e 30 minutos.

3.3.3 3º Abordagem - Pix2pix com headpose

Por ultimo, foi usado outra vez como base a arquitetura *Pix2Pix*, referida na secção 2.5 deste relatório, mas com dados diferentes. Neste foram usados os dados referidos na imagem 3.7.

Nesta tentativa foi usado metade da base de dados, para garantir que apenas o melhores dados são dados ao modelo:

- Treino - 68 indivíduos, contendo 35521 pares de imagens
- Validação - 18 indivíduos, contendo 15814 pares de imagens
- Teste - 5 indivíduos, contendo 3000 pares de imagens

O processo de treino foi realizado em 3 *epochs*, que demorou quase de 4h horas.

3.4 Tecnologias e Ferramentas Utilizadas

3.4.1 PyCharm & DataSpell

Integrated Development Environment (IDE) para a linguagem *Python*, e *Jupyter Notebooks*, que permitem uma grande variedade de ferramentas dentro de um ambiente virtual convenientemente usado para o desenvolvimento na área de ciência de dados. [4] [5]

3.4.2 TensorFlow

O *tensorflow* é uma biblioteca *open-source* usada na área de ML que providencia uma vasta quantidade de ferramentas, bibliotecas e ajuda comunitária. [6]

3.4.3 Keras

O *keras* é uma *Application Programming Interface* (API) *open-source* usada em DL simultaneamente com o *tensorflow*, permitindo um desenvolvimento mais facilitado ao programador.[7]

3.5 Hardware

O *hardware* utilizado no processo de treino do modelo foi o seguinte:

- *CPU: AMD Ryzen™ 9 5900HX, 8-core*
- *RAM: 16GB DDR4-3200 SO-DIMM*
- *GPU: NVIDIA® GeForce RTX™ 3060, 6GB GDDR6*
- *Storage: 1 TB M.2 NVMe™ PCIe® 3.0 SSD*

3.6 Conclusões

Neste capitulo percebemos a importância que a captura de dados teve para a resolução deste projeto, tendo sido a tarefa mais exaustiva e mais demorada. Isto porque em DL o processo de obtenção e uniformização de dados é o mais importante.

Com o desenvolvimento de múltiplas abordagens, podemos perceber que o uso de métodos de treino diferentes, irão levar a uma evolução nos dados de saída.

Um lado mau no treino deste modelos, é a maquina em que estes dados estão a ser processados, tornando lento o desenvolvimento do projeto.

Capítulo

4

Testes e Resultados Obtidos

4.1 Introdução

Neste capitulo serão mostrados os resultados de cada abordagem tomada. É de notar que apenas os resultados da 3º abordagem importam para uma conclusão relativa ao projeto desenvolvido. As 2 primeiras abordagens servem apenas como método de comparação quanto à qualidade final.

4.2 1º Abordagem - U-Net

Apesar do desenvolvimento da arquitetura *U-Net* ter sido um grande avanço na área de ML, esta não é suficiente para a geração de imagens faciais de boa qualidade.

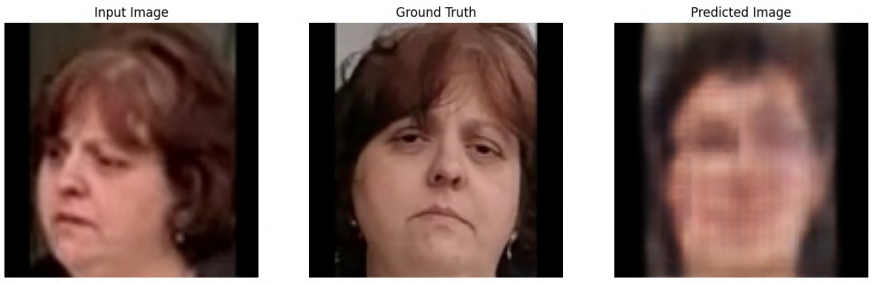

Figura 4.1: Resultados da 1º Abordagem (1)

No entanto, podemos olhar para estes resultados, não como um fracasso, mas sim como um progresso. Pois esta arquitetura pode ser usada em conjunto com outros métodos para criar melhores resultados.

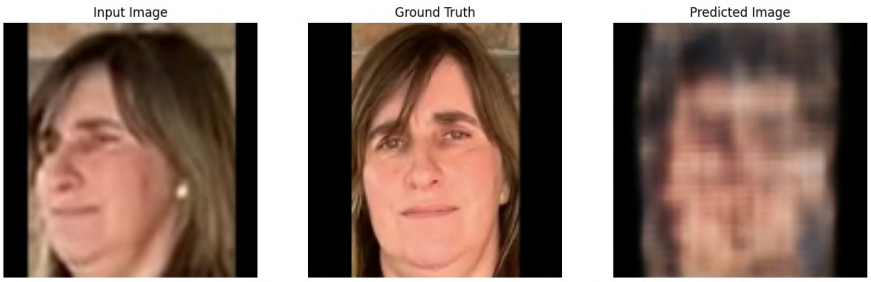

Figura 4.2: Resultados da 1º Abordagem (2)

Figura 4.3: Resultados da 1º Abordagem (3)

4.3 2º Abordagem - Pix2pix

Com o uso de uma *cGAN* podemos observar resultados muito melhores, graças a método competitivos, que fazem o gerador (com a mesma arquitetura *U-Net*) produzir melhores imagens.

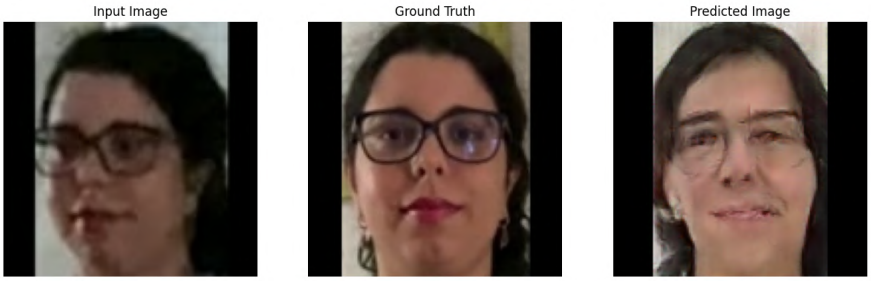

Figura 4.4: Resultados da 2º Abordagem (1)

Ainda assim podemos ver que as previsões não são perfeitas, dando ainda mais espaço para novas melhorias.

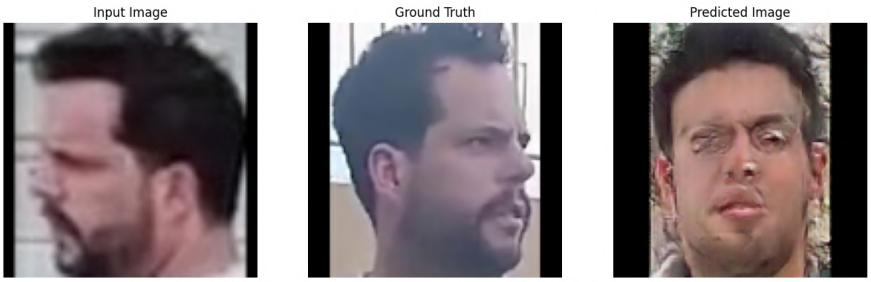

Figura 4.5: Resultados da 2º Abordagem (2)

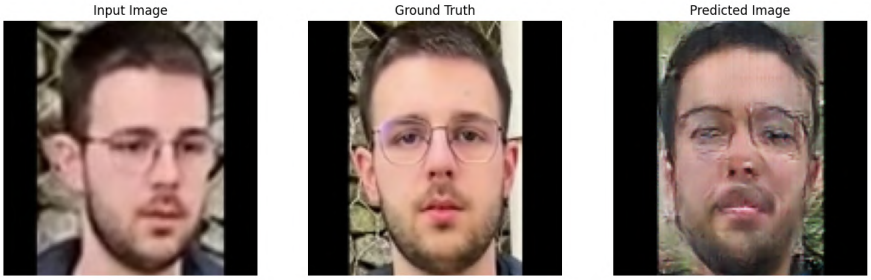

Figura 4.6: Resultados da 2º Abordagem (3)

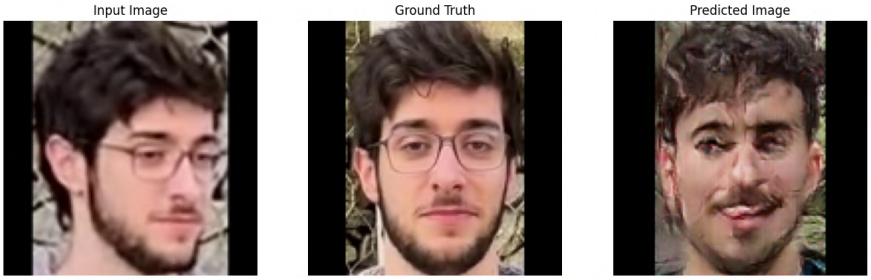

Figura 4.7: Resultados da 2º Abordagem (4)

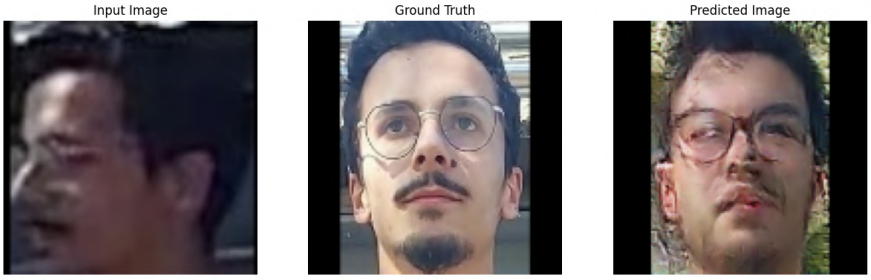

Figura 4.8: Resultados da 2º Abordagem (5)

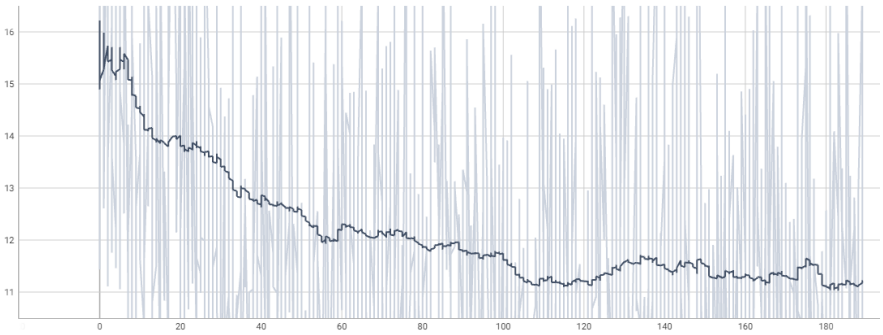

Figura 4.9: Loss do Gerador da 2º Abordagem

4.4 3º Abordagem - Pix2pix com *headpose*

Por ultimo, usando a arquitetura usada na abordagem 2, mas com uma melhor gestão de dados. Neste ponto teve-se em conta o uso de dados consistentes e capturados com o melhor cuidado.

Alem disso foi usado o conjunto de dados com as respetivas posições da cabeça em relação à câmara *(pitch, yaw, e roll)*.

Figura 4.10: Resultados da 3º Abordagem (1)

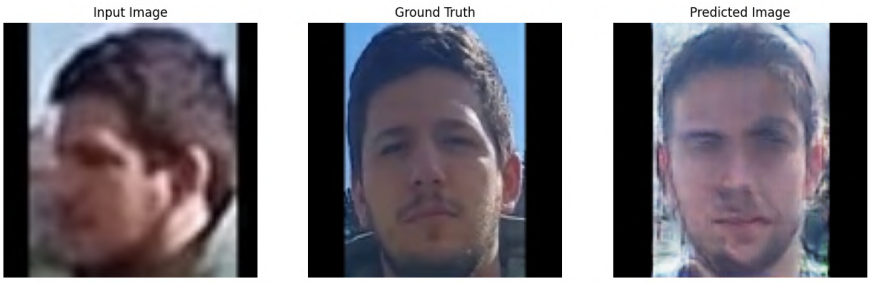

Figura 4.11: Resultados da 3º Abordagem (2)

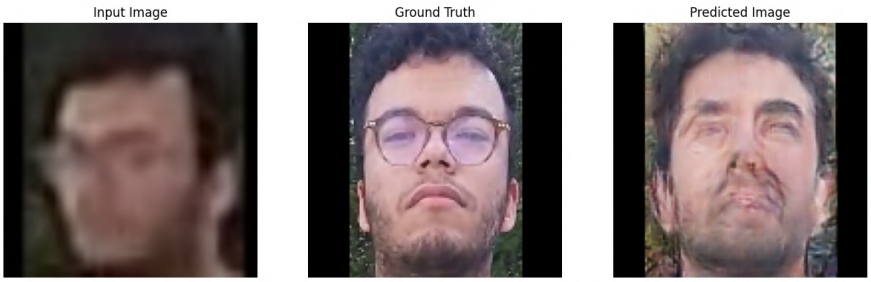

Figura 4.12: Resultados da 3º Abordagem (3)

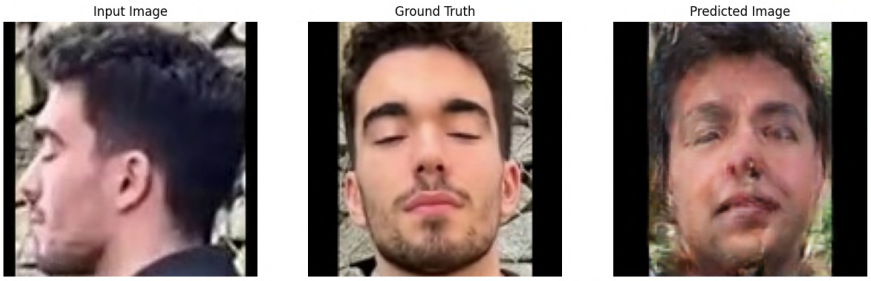

Figura 4.13: Resultados da 3º Abordagem (4)

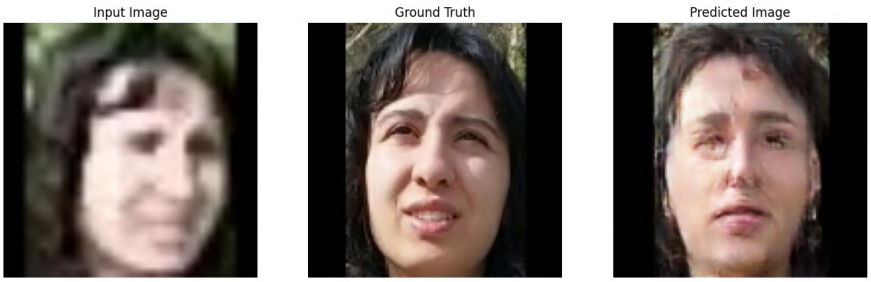

Figura 4.14: Resultados da 3º Abordagem (5)

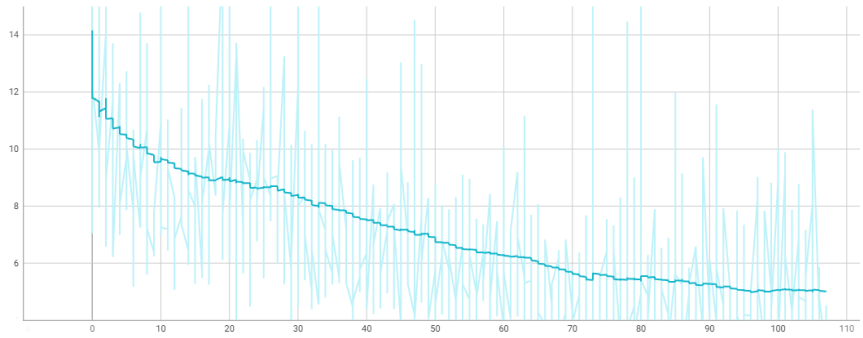

Figura 4.15: Loss do Gerador da 3º Abordagem

Esta abordagem apresenta ainda uma maior evolução na geração de faces humanas, podendo assumir que com maior tratamento de detalhes e melhor classificação, a criação de uma face que apresenta todo o detalhe do respetivo individuo, é possível de ser feito.

4.5 Comparação da 2º e 3º Abordagem

Claramente conseguimos ter a perceção da melhoria que houve da 2º para terceira abordagem, apenas adicionando ao treino, os valores da posição da cabeça e do uso de imagens bem capturadas.

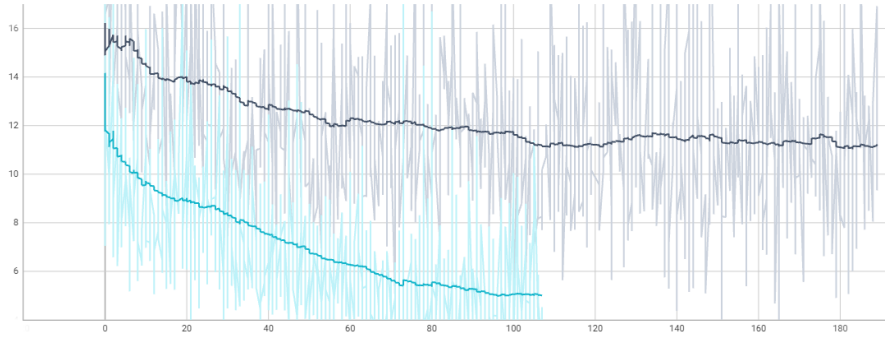

Figura 4.16: Comparação da Loss do Gerador da 2º e 3º Abordagem

4.6 Conclusões

Com a observação dos resultados podemos concluir que a ultima abordagem está mais perto de resultados perfeitos, ainda que visualmente confusos. Num trabalho futuro, com uma nova diferente abordagem ao tema, acredita-se que se irão conseguir resultados melhores.

Capítulo

5

Conclusões e Trabalho Futuro

5.1 Conclusões Principais

Este projeto explorou diversas abordagens para a geração de imagens faciais utilizando técnicas avançadas de DL, incluindo a arquitetura *U-Net* e *cGANs* com a abordagem *Pix2pix*. Os resultados demonstraram que, embora a *U-Net* por si só não seja suficiente para gerar imagens faciais de alta qualidade, a combinação com uma *cGAN* apresentou melhorias significativas.

Os testes mostraram que a adição da pose da cabeça aos dados de entrada na terceira abordagem (*Pix2pix* com *headpose*) resultou numa evolução notável na qualidade das imagens geradas, indicando que um melhor tratamento de detalhes e uma classificação aprimorada são promissores para futuras melhorias.

5.2 Trabalho Futuro

Tendo ficado em aberto a possibilidade de melhoria na qualidade de imagens de saída, este projeto irá ser continuado de forma a implementar técnicas que ficaram por fazer, tal como o uso de segmentação nas imagens de entrada de forma a ocultar o *background*, deixando apenas visível a face dos indivíduos, e o uso das classes (género, idade, cor de pele, uso de óculos, presença de outros acessórios) para diferenciação de indivíduos.

Bibliografia

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. [Online] https://arxiv.org/pdf/1505.04597. Último acesso a 23 de Junho de 2024.

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018. [Online] https://arxiv.org/pdf/1611.07004. Último acesso a 23 de Junho de 2024.

[3] Github - repositório publico - projeto licenciatura. [Online] https://github.com/rAbadesso/Projeto_Licenciatura. Último acesso a 23 de Junho de 2024.

[4] Pycharm: The python ide for data science and web development — jetbrains.com. [Online] https://www.jetbrains.com/pycharm/. Último acesso a 24 de Maio de 2024.

[5] Dataspell: Jetbrains tool for data analysts - jetbrains.com. [Online] https://www.jetbrains.com/dataspell/. Último acesso a 24 de Maio de 2024.

[6] Tensorflow — tensorflow.org. [Online] https://www.tensorflow.org/. Último acesso a 24 de Maio de 2024.

[7] Keras team. keras: the python deep learning api — keras.io. [Online] https://keras.io/. Último acesso a 24 de Maio de 2024.