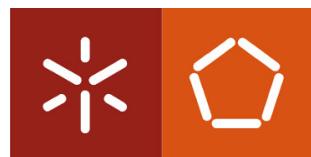


UNIVERSIDADE DO MINHO

ESCOLA DE ENGENHARIA



Aprendizagem Profunda

Sistemas Inteligentes

Mestrado em Engenharia Informática

Geração de retratos de pessoas através de
Deep Generative Models

Grupo 10

| PG50345 | PG50499 | PG50779 |
|--------------|-------------|---------------|
| | | |
| Duarte Lucas | João Torres | Tiago Ribeiro |

Junho, 2023

Conteúdo

| | | |
|----------|--------------------------------|-----------|
| 1 | Introdução | 2 |
| 1.1 | Contextualização | 2 |
| 1.2 | Principais objetivos | 2 |
| 2 | Metodologia | 3 |
| 3 | Dataset | 3 |
| 4 | Descrição do modelo | 4 |
| 4.1 | Escolha do modelo | 4 |
| 4.2 | Arquitetura | 5 |
| 4.2.1 | Gerador | 5 |
| 4.2.2 | Discriminador | 6 |
| 5 | Treino do modelo | 8 |
| 6 | Resultados obtidos | 9 |
| 7 | Conclusão | 12 |

1. Introdução

Este trabalho prático surge no âmbito da unidade curricular de Aprendizagem Profunda, pertencente ao perfil de Sistemas Inteligentes do Mestrado em Engenharia Informática da Universidade do Minho. O projeto desenvolvido teve como motivação a conceção e desenvolvimento de um projeto de *deep learning* utilizando as técnicas abordadas ao longo do semestre. Dito isto, o nosso trabalho incide sobre geração de imagens de retratos.

Neste trabalho, exploramos diferentes técnicas de deep generative models para a geração de imagens. Para isso, utilizamos o dataset *Art portrait*, retirado do *Kaggle*, que contém retratos de pessoas e comparamos o desempenho de diferentes modelos, baseados em redes adversárias generativas (GANs).

Ao longo deste relatório, descrevemos as etapas de pré-processamento dos dados, escolha e ajuste do modelo criado, bem como a avaliação do desempenho e qualidade das imagens geradas.

1.1 Contextualização

A geração de imagens é uma área de pesquisa em constante evolução que tem recebido bastante atenção nos últimos anos. A possibilidade de gerar imagens sintéticas a partir de modelos de *deep learning* tem inúmeras aplicações práticas.

Uma dessas áreas é a produção de conteúdo artístico, onde a geração automática de imagens pode acelerar e tornar mais eficiente o processo criativo.

Nesse contexto, este trabalho insere-se na área de *deep generative models* e tem como objetivo explorar diferentes técnicas para a geração de retratos de pessoas.

1.2 Principais objetivos

O principal objetivo deste projeto consiste no desenvolvimento de um modelo de *deep learning* capaz de gerar retratos de pessoas.

Além disso, procuramos compreender as etapas de pré-processamento dos dados, escolha e ajuste do modelo e avaliação do desempenho e qualidade das imagens geradas, para que seja possível identificar as melhores práticas na utilização de *deep learning* para a geração de imagens sintéticas de retratos.

Por fim, também pretendemos explorar possíveis aplicações práticas dos modelos gerados nesta área e, assim, perceber se a geração automática de imagens pode acelerar e tornar mais eficiente o processo criativo ou prejudicá-lo. Desse modo, procuramos avaliar a capacidade dos modelos de *Deep Generative Models* de gerar imagens de retratos com alta qualidade e diversidade e identificar possíveis limitações e desafios na aplicação dessas técnicas no mundo real.

2. Metodologia

Neste capítulo, é descrita a metodologia utilizada para o desenvolvimento de um modelo gerativo profundo com o objetivo de gerar retratos artísticos. São exploradas as etapas envolvidas no processo, desde a escolha do modelo e do dataset até o treino e avaliação do modelo.

Primeiramente, foi necessário selecionar o modelo a desenvolver e o dataset a utilizar, sendo que o grupo se encontra restringido ao tema de modelos gerativos profundos na pintura/arte. O grupo optou por desenvolver uma rede GAN, mas não um modelo comum, optando pela variante WGAN-GP que irá ser tratada mais adiante neste relatório.

O próximo passo na metodologia é o pré-processamento dos dados. Inicialmente, foi realizada uma análise exploratória dos dados para entender a estrutura e as características dos retratos disponíveis no conjunto de dados. Em seguida, as imagens contidas no conjunto de dados sofreram um processo de tratamento.

Após isto, é tratada a conceção do modelo. A arquitetura da rede WGAN-GP consiste num gerador e um discriminador. O gerador recebe como entrada um vetor de ruído aleatório e gera uma imagem de retrato. O discriminador, por sua vez, recebe uma imagem de um retrato e tenta distinguir se a imagem é falsa ou real. Neste processo, é necessário escolher o número de camadas, neurónios em cada camada, entre outras características, destas redes neuronais.

Com o modelo definido, é feito o treino do mesmo, sendo este um processo iterativo em que é treinado o modelo até um certo ponto, sendo nesse momento feita uma avaliação do mesmo e se está a agir de um modo satisfatório. A partir desta análise é realizada a decisão de prolongar o treino do modelo ou se é necessário alterar algo no mesmo para proceder à continuação do treino.

3. Dataset

A escolha do dataset foi baseada no tema do trabalho de investigação desenvolvido pelo grupo anteriormente, sendo que este se debruça sobre o uso de modelos gerativos profundos no campo da pintura/arte. Tendo em conta esta questão o grupo explorou vários conjuntos de dados em plataformas que disponibilizam estes datasets, tendo encontrado o "*Art Portraits*", um conjunto de dados que contém uma coleção de retratos artísticos de diversas épocas e estilos, incluindo pinturas famosas de artistas reconhecidos. Este coleção de imagens foi retirada da plataforma *Kaggle*.

O conjunto de dados abrange uma ampla gama de estilos artísticos e movimentos. Este dataset é apenas constituído pelas imagens dos retratos, não contendo qualquer tipo de labels adicionais, que poderiam ser úteis para a criação de outro tipo de modelos. A resolução das imagens não é constante, sendo que as imagens têm vários tamanhos diferentes, o que nos obriga a efetuar uma transformação nas mesmas para as deixar todas no mesmo formato.



Figura 3.1: Exemplos de imagens contidas no dataset depois de efetuado o *resize*

Com a escolha do dataset efetuada, o objetivo deste projeto passa por gerar novos retratos a partir das imagens disponibilizadas pelo conjunto de imagens selecionado.

4. Descrição do modelo

Neste capítulo, é explorada a implementação de uma *Wasserstein Generative Adversarial Network* com *gradient penalty* (WGAN-GP) para a geração imagens de retratos artísticos. É abordado o motivo da escolha de uma WGAN-GP em relação a uma GAN convencional e é feita uma análise às arquiteturas do gerador e do discriminador.

4.1 Escolha do modelo

As redes generativas adversariais (GANs) são uma ferramenta poderosa para gerar amostras sintéticas de dados. No entanto, as GANs convencionais têm algumas limitações, tais como:

- **Treino instável:** Acontece quando o discriminador fica muito bom na sua tarefa muito rapidamente e o gerador não o consegue acompanhar. Com isto, o gerador não recebe informação útil do discriminador e não consegue melhorar.
- **Modo colapso:** Ocorre quando o gerador produz apenas algumas variações de amostras, em vez de uma ampla gama de resultados desejáveis. Isto pode levar à falta de diversidade nas amostras geradas e diminuir a qualidade do resultado final.
- **Oscilação:** Durante o treino das GANs, o gerador e o discriminador podem entrar num ciclo, em que um ajuste no gerador leva a um ajuste correspondente do discriminador, e vice-versa. Este ciclo pode resultar em oscilações frequentes entre diferentes estados, tornando o treino instável e dificultando a obtenção de resultados satisfatórios.

Para contornar estes problemas, o grupo decidiu desenvolver uma variação das GANs tradicionais, as WGAN-GP. A principal diferença entre estes tipos de GANs está na função de *loss* e na regularização do treino.

Numa WGAN-GP é utilizada a distância de Wasserstein, também conhecida como *Earth Mover's distance* (EMD), como medida de divergência entre a distribuição real e a distribuição

gerada pelo gerador. Esta forma de determinar a distância é uma métrica mais suave e diferenciável, o que facilita o treino do modelo.

Além disso, na WGAN-GP é introduzido o *gradient penalty* para assegurar que o discriminador se aproxime de ser uma função Lipschitz contínua, o que significa que a função tem uma taxa de variação limitada em relação à distância entre os pontos do seu domínio. Esta propriedade é importante para garantir a estabilidade do treino, evitando gradientes extremos ou oscilações na função de custo. Em termos práticos, impor a restrição Lipschitz ao discriminador ajuda a controlar a taxa de mudança das saídas do discriminador e a evitar problemas como o colapso do gradiente.

Portanto, a escolha da WGAN-GP em vez de uma GAN convencional deve-se às vantagens teóricas e práticas oferecidas por esta abordagem, como uma convergência mais estável e um treino mais suave.

4.2 Arquitetura

Nesta secção, vamos explorar em detalhe as duas principais redes que compõem uma *Wasserstein Generative Adversarial Network* (WGAN): a rede geradora e a rede discriminadora. Essas redes desempenham papéis opostos, mas complementares, no processo de treino de uma WGAN. A rede geradora é responsável por criar amostras sintéticas que se assemelham ao conjunto de dados de treino, enquanto a rede discriminadora procura distinguir entre as amostras geradas pela rede geradora e as amostras reais do conjunto de dados. Através destes processos trabalham ambas em conjunto para melhorar as suas capacidades, o que resulta na geração de amostras cada vez mais realistas ao longo do tempo. Vamos examinar a arquitetura e o funcionamento de ambas as redes.

4.2.1 Gerador

A rede geradora é crucial para o bom funcionamento de uma *Wasserstein Generative Adversarial Network* (WGAN). Tem como objetivo gerar imagens realistas, aprendendo os padrões subjacentes e a distribuição de um determinado conjunto de dados.

A arquitetura desta rede consiste em várias camadas, cada uma responsável por transformar o vetor de ruído de entrada numa representação de dimensão superior que se assemelha às imagens alvo. Vamos analisar a arquitetura do gerador passo a passo:

Camada de entrada: O gerador recebe um vetor de ruído aleatório como entrada. Este funciona como uma representação do espaço latente e é normalmente amostrado a partir de uma distribuição gaussiana. No código, o vetor de ruído de entrada tem uma dimensão de `z_dim`, que é definida como 100.

Camadas lineares com ativação LeakyReLU: A rede geradora começa com uma camada linear que recebe o vetor de ruído como entrada e produz uma representação de dimensão superior. Esta representação é então passada através de uma função de ativação LeakyReLU, que introduz a não linearidade na rede. A LeakyReLU ajuda o modelo a aprender mapeamentos complexos ao permitir pequenos valores negativos, de forma a evitar o problema de

desaparecimento do gradiente.

Camadas ocultas: O gerador consiste em várias camadas ocultas, cada uma composta por uma transformação linear seguida de uma ativação LeakyReLU. Existem três camadas ocultas, aumentando progressivamente a dimensionalidade da representação. Isto ajuda o gerador a captar padrões e detalhes mais complexos à medida que se aprofunda.

Camada de saída: A camada linear final na rede do gerador transforma a representação de alta dimensão num tensor de saída que corresponde à forma desejada das imagens geradas. Neste caso, o objetivo do gerador é produzir imagens RGB com uma resolução de 64x64 pixels. A camada de saída tem $3 \times 64 \times 64$ unidades para representar os três canais de cor (RGB) e as dimensões espaciais.

Activação Tanh: O tensor de saída do gerador passa por uma função de ativação tangente hiperbólica (\tanh). A função \tanh escala os valores dos pixels para o intervalo $[-1, 1]$, correspondendo à normalização desejada para os dados da imagem. Garante que as imagens geradas tenham valores de pixel comparáveis aos das imagens reais.

O método *forward* da rede geradora pega no vetor de ruído de entrada e passa-o através das camadas sequenciais definidas acima. O resultado é o tensor da imagem gerada, que representa uma imagem sintética que imita a distribuição das imagens reais no conjunto de dados de treino.

Durante o treino, o principal objetivo do gerador é produzir imagens realistas que possam enganar com êxito a rede discriminadora. Ao minimizar a perda calculada a partir das previsões do discriminador nas amostras geradas, o gerador aprende a gerar imagens que se assemelham às reais. Este é atualizado através de *backpropagation* e de técnicas de otimização, como o otimizador *Adam* utilizado.

Finalmente, depois de treinar o gerador, a função *test_generator* é fornecida para gerar um conjunto de imagens de amostra. Recebe vetores de ruído aleatórios como entrada, passa-os pelo gerador e visualiza as imagens geradas utilizando a biblioteca *Matplotlib*.

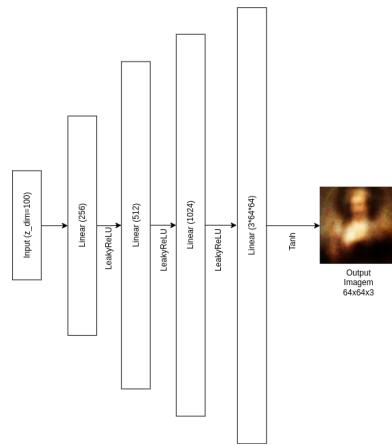


Figura 4.1: Arquitetura da rede geradora

4.2.2 Discriminador

A rede discriminadora desempenha um papel fundamental na *Wasserstein Generative Adversarial Network* (WGAN). A sua principal função é avaliar a autenticidade das amostras geradas e

diferenciá-las das amostras reais. Ao fornecer *feedback* à rede geradora, o discriminador orienta o processo de aprendizagem e ajuda a melhorar a qualidade das imagens geradas. A arquitetura da rede discriminadora foi concebida para processar imagens de entrada e classificá-las como reais ou falsas.

Camada de entrada: O discriminador recebe uma imagem como entrada, que normalmente tem a forma de um tensor que representa os pixéis da imagem. Neste caso, as imagens de entrada têm um tamanho de 64x64 pixéis com três canais de cor (RGB).

Camadas lineares com activação LeakyReLU: A rede discriminadora começa com uma camada linear que transforma a imagem de entrada numa representação de dimensão superior. A saída desta camada é então passada através de uma função de activação *LeakyReLU*, introduzindo a não-linearidade na rede, que permite a aprendizagem de mapeamentos complexos, permitindo pequenos valores negativos na rede.

Camadas ocultas: Tal como o gerador, o discriminador consiste em várias camadas ocultas, cada uma composta por uma transformação linear seguida de uma activação *LeakyReLU*. Há três camadas ocultas, reduzindo gradualmente a dimensionalidade da representação. Isto ajuda o discriminador a aprender características e padrões discriminativos à medida que se aprofunda.

Camada de saída: A camada linear final na rede do discriminador transforma a representação de baixa dimensão numa única unidade de saída. Esta unidade de saída representa a confiança do discriminador no facto de a imagem de entrada ser verdadeira ou falsa. Um valor de saída mais alto indica uma maior probabilidade de a imagem ser real, enquanto um valor mais baixo sugere que a imagem tem maior probabilidade de ser falsa.

Durante o processo de treino, o objetivo do discriminador é classificar com precisão as amostras reais e geradas. É treinado utilizando uma combinação de amostras reais do conjunto de dados e amostras geradas a partir da rede geradora. A perda do discriminador é calculada com base na diferença entre as suas previsões e os rótulos verdadeiros das amostras.

O algoritmo de otimização utiliza o método de penalização do gradiente para impor a restrição *Lipschitz*, um componente crucial das WGAN-GP. A função *compute_gradient_penalty* calcula a perda de penalidade de gradiente, que incentiva a suavidade no limite de decisão do discriminador e ajuda a estabilizar o processo de treino.

Ao atualizar iterativamente os parâmetros da rede discriminadora, o modelo aprende a distinguir, de forma mais eficaz, as amostras reais das geradas. À medida que a rede geradora melhora a sua capacidade de enganar o discriminador, este torna-se mais hábil na distinção entre amostras reais e falsas, conduzindo a uma dinâmica de aprendizagem competitiva e adversária.

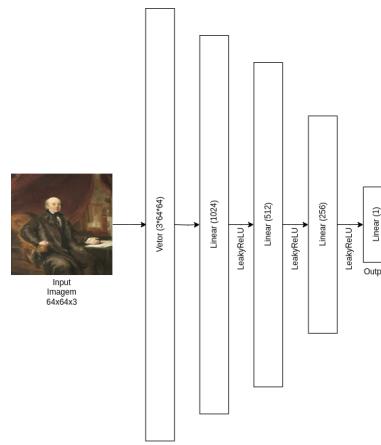


Figura 4.2: Arquitetura da rede discriminadora

5. Treino do modelo

O treino do modelo segue um procedimento específico que envolve o treino de várias *epochs* e a atualização da rede discriminadora várias vezes para cada iteração. Nesta secção vamos detalhar todo este processo.

Treino de várias *epochs*: O código inclui um loop que itera sobre um número especificado de *epochs* (controlado pela variável *num_epochs*). Em cada *epoch*, as redes do gerador e do discriminador são treinadas através do uso de diferentes conjuntos de amostras reais do conjunto de dados.

Treino do discriminador várias vezes: Em cada *epoch*, a rede do discriminador é treinada várias vezes antes de atualizar a rede do gerador. No caso deste modelo, o discriminador é treinado cinco vezes para cada conjunto de amostras reais.

a) Amostras reais: Para cada iteração de treino do discriminador, um conjunto de amostras reais é passado pela rede. O discriminador avalia essas amostras e calcula as pontuações de validade para as mesmas.

b) Amostras geradas: De seguida, a rede geradora gera um conjunto de amostras falsas por amostragem a partir do espaço latente. Estas amostras geradas são então passadas pelo discriminador e as suas pontuações de validade são calculadas.

c) Gradient Penalty e Adversarial Loss: Depois de avaliar as amostras reais e geradas, o discriminador calcula a *gradient penalty loss* através do uso da função *compute_gradient_penalty*. Este ajuda a impor a restrição de *Lipschitz* na WGAN-GP. A *adversarial loss*, que consiste na média negativa dos índices de validade reais e na média dos índices de validade gerados.

d) Backpropagation e otimização do discriminador: Os gradientes são então calculados e retropropagados através da rede do discriminador. Os parâmetros deste são atualizados através do otimizador (*disc_opt*) tendo em conta estes gradientes, melhorando assim a capacidade do discriminador para distinguir entre amostras reais e geradas.

Treino do gerador: Depois de treinar o discriminador, a rede do gerador é treinada uma vez. O gerador gera um novo conjunto de amostras falsas, que são passadas pelo discriminador. A *adversarial loss* é calculada através das pontuações de validade atribuídas pelo discrimina-

dor. Os gradientes são retropropagados através da rede do gerador e os seus parâmetros são atualizados utilizando o otimizador (*gen_opt*).

Guardar as informações das redes ao longo do treino: O nosso código inclui pontos de verificação para guardar os modelos do gerador e do discriminador em intervalos regulares. A cada 5 *epochs*, o estado atual de ambas as redes é guardado em ficheiros separados.

Guardar informações das *losses* das redes ao longo do treino: O *loop* de treino controla as perdas do discriminador e do gerador (*d_losses* e *g_losses*, respetivamente) para cada *epoch*. Essas perdas são armazenadas em listas separadas para monitorizar o progresso do treino. Estas perdas também são guardadas em ficheiros a cada 5 *epochs*.

De forma geral, o processo de treino envolve o treino das redes geradora e discriminadora de forma alternada, com o discriminador a ser treinado várias vezes em cada *epoch*. Este treino contraditório permite que o gerador melhore a sua capacidade de gerar amostras realistas, enquanto o discriminador aprende a distinguir melhor entre amostras reais e falsas.

6. Resultados obtidos

Neste capítulo, são apresentados os resultados obtidos durante o treino da rede WGAN-GP. Serão fornecidas imagens dos resultados visuais do modelo em diferentes *epochs*, a fim de ilustrar a evolução da geração de retratos ao longo do treino.

Primeiramente, o modelo foi comparado nas *epochs* iniciais com modelos de GANs disponíveis na plataforma *Kaggle*. É possível observar esta comparação na figura 6.1, onde são exibidas as imagens geradas pelo modelo definido pelo grupo e imagens geradas pelos modelos disponibilizados. Tal como nos modelos escolhidos para comparação, a WGAN-GP também é capaz de captar alguns elementos das imagens reais, como por exemplo, as silhuetas das pessoas representadas nos retratos, porém não são capazes de detalhar outros pormenores mais complexos, como a expressão facial. Logo podemos observar que o modelo desenvolvido se comporta de maneira similar com os outros modelos.

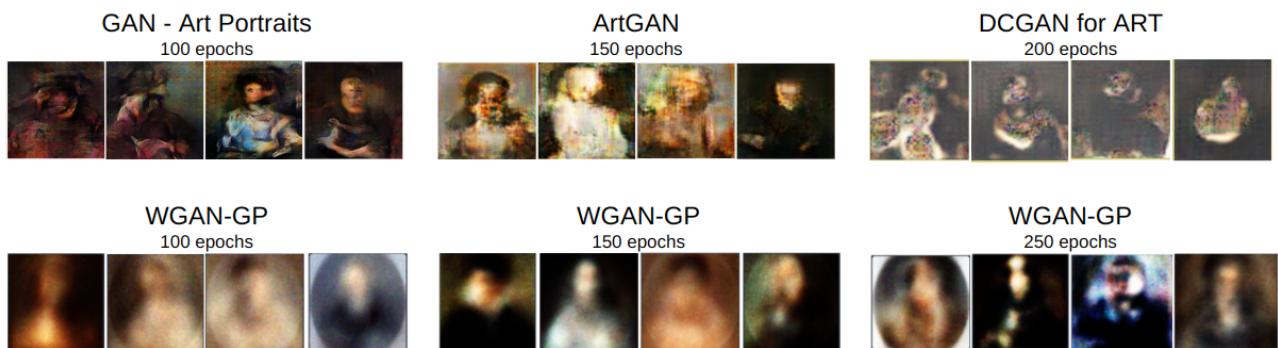


Figura 6.1: Comparação entre os vários modelos [2, 3, 4]

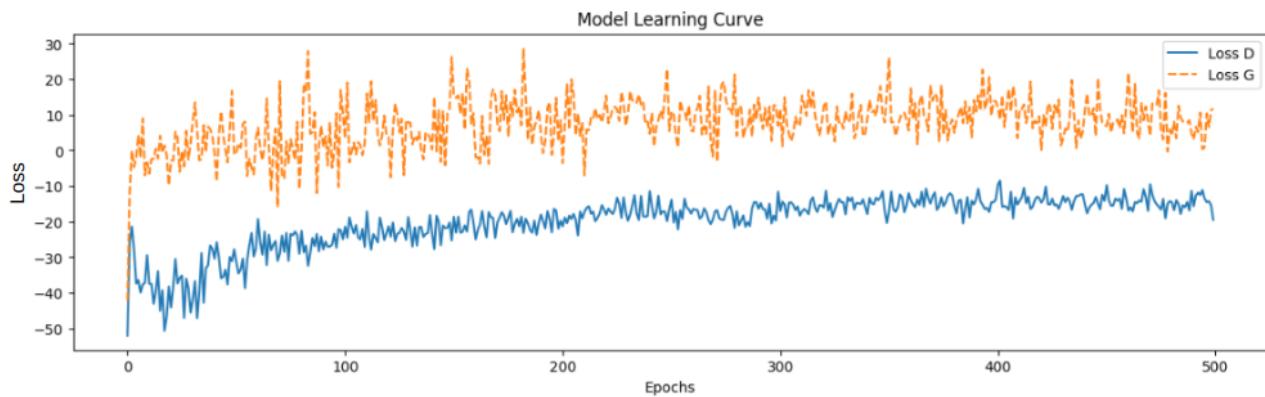


Figura 6.2: Gráfico da *loss* desde o início do treino até às 500 *epochs*

Na figura acima é possível ver as *losses* obtidas pela rede do gerador e discriminador, sendo que não foi identificado nenhum problema que tenha posto em causa a competência do modelo para gerar imagens.

Na busca de melhorar as imagens geradas pela WGAN-GP, o grupo prosseguiu o treino do modelo. O modelo foi treinado de 500 em 500 epochs, sendo que cada uma destas etapas levava cerca de 11 horas. Após este número de epochs, o grupo observava as imagens e métricas geradas e continuava a treinar o modelo.

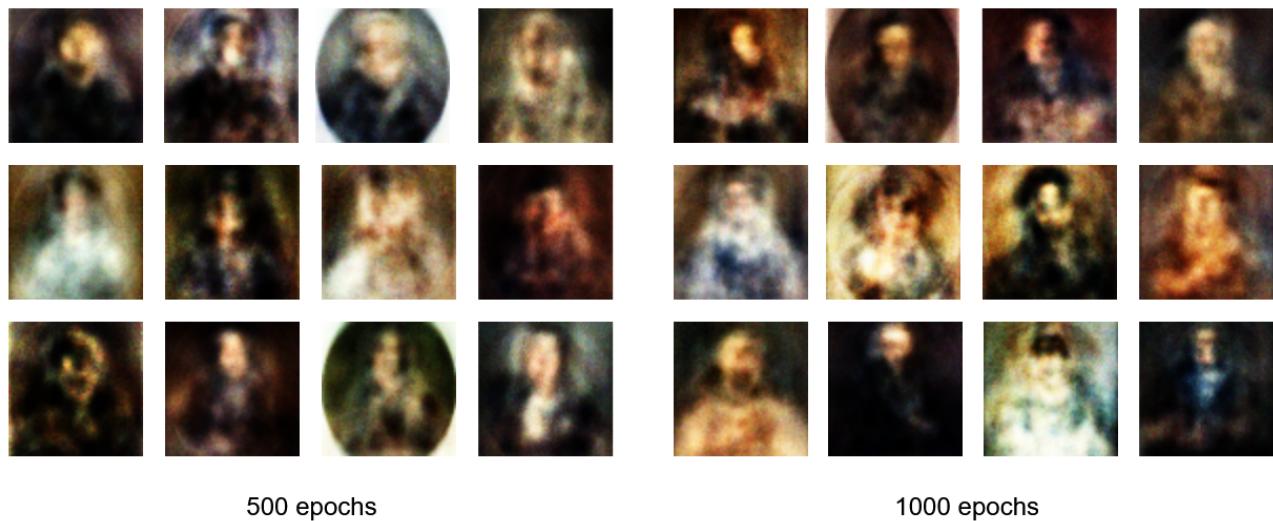


Figura 6.3: Resultados após 500 e 1000 *epochs*

Ao longo do treino, já foi possível observar o modelo a conseguir captar outros aspectos das imagens reais com uma maior definição, como por exemplo, os rostos das pessoas a serem desenhadas, sendo possível identificar elementos como olhos, bocas, entre outros.



Figura 6.4: Resultados após 1500 e 2000 *epochs*

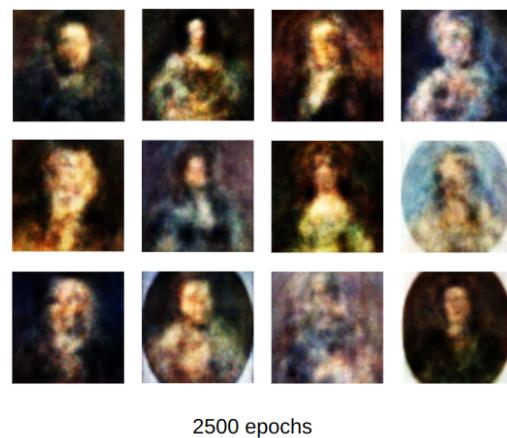


Figura 6.5: Resultados após 2500 *epochs*

Até chegar às 2500 *epochs* o modelo demorou cerca de 2 dias e meio a treinar. Ao longo do treino foi possível visualizar um incremento da qualidade das imagens geradas, porém mesmo com esta quantidade de treino ainda é possível distinguir facilmente uma imagem gerada de uma imagem real e ainda existe a presença de imagens com baixa qualidade, devido à presença de ruído.

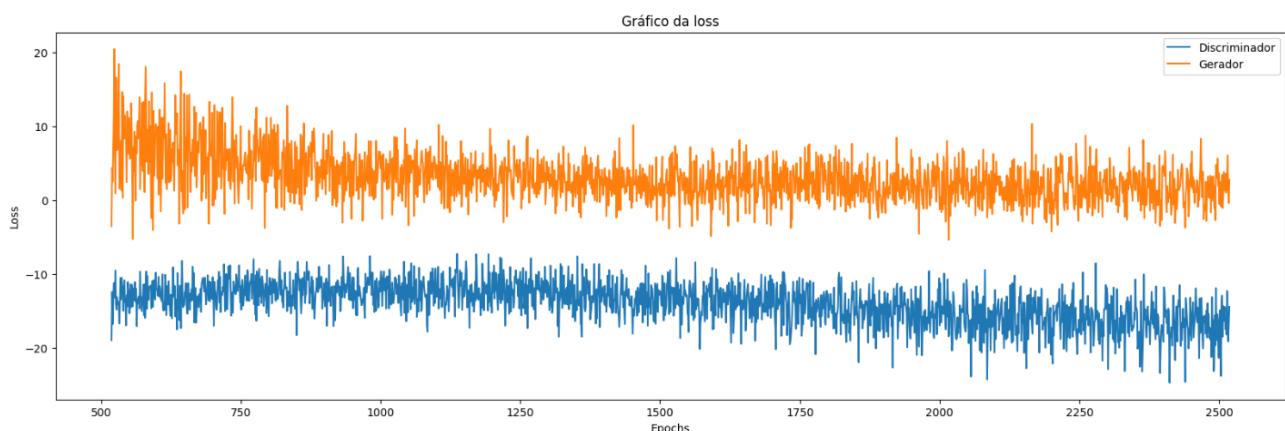


Figura 6.6: Gráfico da *loss* desde as 500 a 2500 *epochs*

Ao analisar o gráfico da *loss* do gerador e do discriminador não é possível identificar nenhum problema grave. O treino ocorre com normalidade, sendo que tira partido das vantagens do modelo WGAN-GP, dando mais estabilidade ao treino e impedindo a entrada no modo colapso. Os valores de *loss* de ambas as redes não se afastam muito do 0, sendo que a *loss* do gerador está a diminuir, enquanto que acontece o inverso com o discriminador.

Embora os resultados obtidos sejam encorajadores, é importante destacar as limitações encontradas durante o desenvolvimento deste projeto. Uma limitação significativa é a dependência do tamanho e da diversidade do dataset "Art Portraits". Além disso, é importante considerar que o modelo WGAN-GP utilizado neste projeto requer recursos computacionais significativos e um longo tempo de treino para alcançar resultados satisfatórios. A melhoria da eficiência e velocidade de treino do modelo é uma área de pesquisa que pode ser explorada em trabalhos futuros.

7. Conclusão

O uso das WGAN-GP representa uma boa escolha para o projeto, pois aborda algumas limitações comuns das GANs convencionais. Através da distância de Wasserstein e da introdução do *gradient penalty*, as WGAN-GP proporcionam um treino mais estável, o que evita problemas como oscilações não desejáveis.

É importante destacar que as WGAN-GP não são uma solução definitiva para todos os problemas relacionados à geração de amostras sintéticas de dados, mas representam uma abordagem avançada e eficaz para contornar algumas limitações comuns das GANs convencionais. A implementação correta e o ajuste adequado dos hiperparâmetros são essenciais para obter os melhores resultados.

Em conclusão, os resultados obtidos com este modelo foram bons. As amostras geradas mostraram uma evolução ao longo das epochs, captando elementos presentes nas pinturas do dataset escolhido. No entanto, é importante reconhecer as limitações do projeto, como a dependência do tamanho e diversidade do dataset, além dos recursos computacionais significativos e tempo de treino prolongado exigidos pelo modelo.

Há também espaço para melhorias neste trabalho. Seria interessante explorar técnicas de pré-processamento de dados para aumentar a diversidade e qualidade do dataset, o que iria proporcionar uma gama mais ampla de retratos gerados. Além disso, a otimização dos hiperparâmetros do modelo e a investigação de arquiteturas de redes neurais mais avançadas podem contribuir para melhorar ainda mais a qualidade e diversidade dos retratos gerados.

Assim, enquanto grupo conseguimos distribuir bem o trabalho entre todos. Os membros do grupo ajudaram-se mutuamente e, de forma geral, consideramos que tivemos um aproveitamento positivo. Concluindo, este trabalho ajudou-nos a desenvolver novas aptidões e a consolidar todos os conceitos lecionados em aula, nomeadamente sobre *Deep Generative Models* e *GANs*.

Bibliografia

- [1] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A. C. (2017). Improved training of wasserstein GANs. Advances in neural information processing systems, 30.
- [2] GAN - Art Portraits, <https://www.kaggle.com/code/emre1k/gan-art-portraits>.
- [3] ArtGAN, <https://www.kaggle.com/code/robertonacu/artgan>.
- [4] DCGAN for ART, <https://www.kaggle.com/code/gitanjali1425/dcgan-for-art>.
- [5] Repositório WGAN-GP, <https://github.com/tiagoribeiro2001/Projeto-AP>