



**Politécnico
Castelo Branco**

Escola Superior
de Tecnologia

BordadoBOT

Projeto I & II

João Rafael de Matos Pires

Nº 20220424

Gabriel Horta Charro

Nº 20220414

Orientadores

Arlindo Ferreira da Silva

Ana Paula Neves Ferreira da Silva

Trabalho de Projeto apresentado à Escola Superior de Tecnologia do Instituto Politécnico de Castelo Branco para cumprimento dos requisitos necessários à obtenção do grau de Licenciatura em Engenharia Informática, realizada sob a orientação científica do Professor Doutor Arlindo Ferreira da Silva e coorientação científica da Professor Doutora Ana Paula Neves Ferreira da Silva, do Instituto Politécnico de Castelo Branco.

Junho de 2025

Composição do júri

Presidente do júri

Doutor, Pedro Nuno Moreira da Silva

Professor Adjunto, Escola Superior de Tecnologia

Vogais

Doutor, Arlindo Ferreira da Silva

Professor Adjunto, Escola Superior de Tecnologia

Doutor, Fernando Reinaldo da Silva Garcia Ribeiro

Professor Adjunto, Escola Superior de Tecnologia

Resumo

O setor das artes e do design, especialmente no contexto do bordado, possui uma rica tradição que reflete estilos e técnicas específicas, como o bordado de Castelo Branco. Com o avanço da tecnologia e da inteligência artificial, surgem novas formas de expressão artística, permitindo a reinterpretação inovadora de estilos tradicionais.

Este projeto explora a adaptação e criação de imagens ao estilo do bordado de Castelo Branco utilizando técnicas de inteligência artificial, investigando a interseção entre arte tradicional e tecnologia moderna. Para isso, foram aplicados modelos de Stable Diffusion, capazes de gerar imagens que reflitam as características visuais deste bordado. O trabalho incluiu uma revisão do estado da arte sobre transferência de estilos artísticos com inteligência artificial, a criação de um dataset de imagens do bordado de Castelo Branco e o treino de diversos modelos.

Durante o desenvolvimento, foram realizados testes exaustivos com diferentes parâmetros de geração e otimização dos prompts textuais, com o objetivo de maximizar a qualidade e fidelidade do estilo nos resultados. Essa abordagem permitiu identificar as configurações que mais se adequavam para criar imagens que remetem ao estilo do bordado, tanto no modo txt2img quanto img2img.

Por fim, foi desenvolvido um protótipo multiplataforma, acessível e intuitivo, que permite a utilizadores com diferentes níveis de conhecimento técnico interagir com os modelos treinados e gerar imagens no estilo pretendido. O projeto não apenas preserva e celebra a herança cultural do bordado de Castelo Branco, mas também inova ao integrar inteligência artificial, possibilitando a aplicação desse estilo tradicional a novas imagens e contextos, além de contribuir para a facilitação do acesso a ferramentas criativas baseadas em inteligência artificial

Palavras-chave

Inteligência Artificial

Deep Learning

Transferência de Estilo

Modelos de Difusão

Arte Generativa

Abstract

The arts and design sector, particularly in the context of embroidery, has a rich tradition that reflects specific styles and techniques such as Castelo Branco embroidery. With the advancement of technology and artificial intelligence, new forms of artistic expression have emerged, enabling the innovative reinterpretation of traditional styles.

This project explores the adaptation and creation of images in the style of Castelo Branco embroidery using artificial intelligence techniques, investigating the intersection between traditional art and modern technology. To achieve this, Stable Diffusion models were applied to generate images that reflect the visual characteristics of this embroidery. The work included a review of the state of the art in artistic style transfer with AI, the creation of a dataset of Castelo Branco embroidery images, and the training of several models.

During development, extensive tests were conducted with different generation parameters and prompt optimization, aiming to maximize the quality and stylistic fidelity of the results. This approach allowed the identification of the most suitable configurations for generating images that evoke the embroidery's style, both in txt2img and img2img modes.

Finally, a multiplatform, accessible, and intuitive prototype was developed, enabling users with different technical backgrounds to interact with the trained models and generate stylized images. The project not only preserves and celebrates the cultural heritage of Castelo Branco embroidery, but also innovates by integrating artificial intelligence, making it possible to apply this traditional style to new images and contexts, while contributing to the democratization of access to creative AI-based tools.

Keywords

Artificial Intelligence

Deep Learning

Style Transfer

Diffusion models

Generative Art

Índice geral

1. Introdução	1
1.1 Enquadramento.....	2
1.2 Objetivos	3
1.3 Planeamento do Projeto	4
1.4 Estrutura do Relatório	6
2. Enquadramento Teórico	9
2.1 Inteligência Artificial.....	9
2.1.1 Definição e Conceitos	10
2.1.2 Evolução e Aplicações.....	11
2.2 <i>Machine Learning</i>	11
2.2.1 Tipos de Aprendizagem.....	12
2.2.1.1 Aprendizagem Supervisionada	12
2.2.1.2 Aprendizagem Não Supervisionada.....	12
2.2.1.3 Aprendizagem Por Reforço	13
2.2.2 Redes Neurais.....	13
2.2.3 Deep Learning.....	14
2.3 Modelos de Difusão.....	14
2.3.1 Definição e Conceitos	15
2.3.2 Como Funcionam	15
2.4 <i>Stable Diffusion</i>	18
2.4.1 Arquitetura	18
2.4.2.1 <i>VAE (Variational AutoEncoder)</i>	18
2.4.2.2 <i>LDM (Latent Diffusion Model)</i>	19
2.4.2.3 Rede <i>U-Net</i>	20
2.4.2.4 <i>CLIP (Contrastive Language-Image Pretraining)</i>	20
2.4.3 Processo de Treino	21
3. Estudo do Estado de Arte.....	22
3.1 Metodologia e Processo de Pesquisa	22
3.1.1 Objetivo da Pesquisa	22
3.1.2 Fontes Utilizadas	23
3.1.3 Estratégia de Pesquisa.....	24
3.1.4 Critérios de Exclusão	24
3.2 Análise dos Artigos Encontrados	25
3.3 Discussão dos Resultados	41

3.4 Conclusões	42
4. Transferência de Estilo com <i>Stable Diffusion</i>	43
4.1 O que é Transferência de Estilo	43
4.2 Implicações Éticas e Legais	43
4.3 Técnicas Tradicionais de Transferência de Estilo	44
4.4. Aplicação de Transferência de Estilo com <i>Stable Diffusion</i>	45
5. Caso de Estudo: Bordado de Castelo Branco	46
5.1 História	46
5.2 Criação do <i>Dataset</i>	47
5.2.1 Recolha das Imagens	47
5.2.2 Desafios	48
5.2.3 Preparação dos Dados	48
6. Ferramentas Utilizadas	50
6.1 <i>Python</i>	50
6.2 <i>Google Colab</i>	50
6.3 <i>Hugging Face</i>	50
6.3 <i>Birne</i>	50
6.4 <i>Automatic1111</i>	51
6.5 <i>Dreambooth</i>	51
6.6 <i>GitHub</i>	51
6.7 <i>Dart</i>	51
6.8 <i>Flutter</i>	51
7. Resultados Obtidos pelo primeiro modelo	52
7.1 Treino Realizado	52
7.2 Escolha dos <i>prompts</i>	55
7.3 Exemplos de Imagens geradas através de <i>txt2img</i>	56
7.4 Exemplos de Imagens geradas através de <i>img2img</i>	61
7.5 Avaliação dos Resultados Obtidos pelo Primeiro Modelo	65
8. Melhorias	67
8.1 Primeiro modelo – modelo_melhores	68
8.1.1 Novo <i>Dataset</i>	68
8.1.2. Parâmetros de treino	69
8.2 Segundo modelo – modelo_styles	70
8.2.1. <i>Dataset</i> utilizado	70
8.2.2. Parâmetros de treino	71
8.3 Melhorias nos <i>Prompts</i>	71

8.3.1. Como melhorar os <i>prompts</i> utilizados.....	73
8.3.2. Estudo do impacto dos <i>prompts</i> textuais em <i>txt2img</i>	73
9. Parâmetros dentro do A1111	83
9.1 O que fazem?	83
9.2 Como melhorar?	84
9.3 Como os parâmetros influenciam o resultado em <i>txt2img</i>	84
9.3.1. Estudo do parâmetro <i>sampling_method</i>	85
9.3.2. Estudo do parâmetro <i>schedule type</i>	94
9.3.3. Estudo dos parâmetros <i>cfg_scale</i>	102
9.3.4. Estudo do parâmetro <i>steps</i>	109
9.4 Como os parâmetros influenciam o resultado em <i>img2img</i>	114
9.4.1. Estudo do parâmetro <i>denoising_strength</i>	115
10. Exemplos gerados com os novos modelos	122
10.1 Exemplos <i>txt2img</i>	122
10.2 Exemplos <i>img2img</i>	135
11. Criação de um protótipo para o BordadoBOT.....	144
11.1 Tecnologias Escolhidas	144
11.2 Arquitetura	145
11.3 Explicação das Interfaces e uso do protótipo	145
11.4 Como os <i>prompts</i> são geridos?	151
12. Conclusões	153
12.1 Desafios e limitações	154
12.2 Trabalho futuro.....	155
Referências	156

Índice de figuras

Figura 1 – Comparação entre programação tradicional e <i>Machine Learning</i> (Retirado de: [18])	12
Figura 2 – Exemplo de uma rede neuronal (Retirado de: [22])	14
Figura 3 – Processo de difusão (Retirado de: [27]).....	16
Figura 4 – Processo de difusão reverso (Retirado de: [27]).....	17
Figura 5 – Demonstração visual de ambos processos (Retirado de: [28])	17
Figura 6 – Comparação visual entre diferentes modelos.....	18
Figura 7 – Arquitetura de um VAE (Retirado de: [33]).....	19
Figura 8 – Arquitetura de uma rede U-Net (Retirado de: [35]).....	20
Figura 9 – Ilustração do processo de treino de um modelo de <i>Stable Diffusion</i> (Retirado de: [37])	21
Figura 10 – Uso de inteligência artificial generativa regular nas empresas (Adaptado de:[38]).....	23
Figura 11 – Número de Artigos com o termo ‘Stable Diffusion’ (Fonte: Google Scholar)	23
Figura 12 – Comparação dos resultados obtidos pelos autores e outras abordagens (Retirado de: [41])	26
Figura 13 – Comparação de imagens geradas com outras alternativas (Retirado de: [42]).....	27
Figura 14 – Comparação de imagens geradas com outras abordagens (Retirado de: [20]).....	28
Figura 15 – À esquerda a imagem utilizada como treino para o modelo apresentado, e à direita uma recriação da imagem original feita pelo modelo (Retirado de: [43]) ..	29
Figura 16 – Exemplos de imagens geradas pelos diversos elementos utilizados no modelo apresentado, juntamente das imagens utilizadas para o estilo e prompt textual (Retirado de: [44])	30
Figura 17 – Comparação da aplicação de estilos entre diversos modelos (Retirado de: [46]).....	32
Figura 18 – Exemplo da aplicação do modelo apresentado (Retirado de: [47])	33
Figura 19 – Exemplo de imagens geradas pelo modelo apresentado, a primeira coluna representa a imagem utilizada como treino, enquanto o resto são diferentes outputs do modelo (Retirado de: [48]).....	34
Figura 20 — Comparação entre outros modelos, acompanhada pela imagem de input e descrição textual do estilo (Retirado de [49])	36
Figura 21 — Comparação dos resultados obtidos com outros modelos, as colunas B-D representam os modelos de <i>Stable Diffusion</i> estudados pelos autores, e as colunas E-H representam outras técnicas como o uso de <i>GANs</i> (Retirado de [50])	37
Figura 22 — Exemplos de imagens geradas por outros modelos e pelo modelo apresentado pelos autores, com base num <i>input (Content)</i> e num estilo (<i>Style</i>) submetidos (Retirado de: [51])	39
Figura 23 — Exemplo do uso do <i>NPR</i> para três estilos distintos de pinturas (Retirado de: [54])	44

Figura 24 — Exemplos de transferências de estilo com Stable Diffusion (Retirado de: [55])	45
Figura 25 – Exemplo do Bordado de Castelo Branco.....	47
Figura 26 – Seleção das imagens para o <i>dataset</i>	47
Figura 27 – Exemplo de uma foto obtida.....	48
Figura 28 – Seleção das imagens para o dataset após recortes e rotações	49
Figura 29 – Parâmetros do <i>notebook</i> de treino da <i>Dreambooth</i>	52
Figura 30 — Parâmetros de treino	53
Figura 31 — Escolha das imagens de treino	53
Figura 32 — Upload do <i>dataset</i> para treino.....	53
Figura 33 — Interface gráfica do <i>A1111</i>	54
Figura 34 - Primeira imagem gerada	56
Figura 35 - Segunda imagem gerada	57
Figura 36 - Terceira Imagem gerada.....	57
Figura 37 - Quarta Imagem gerada	58
Figura 38 - Quinta Imagem gerada	58
Figura 39 - Sexta Imagem gerada	59
Figura 40 - Sétima Imagem gerada	59
Figura 41 - Oitava Imagem gerada	60
Figura 42 - Nona Imagem gerada	60
Figura 43 - Décima Imagem gerada.....	61
Figura 44 - Parâmetros para as imagens geradas anteriormente	61
Figura 45 — Transferência de estilo com uma flor	62
Figura 46 — Transferência de estilo com uma árvore.....	62
Figura 47 — Transferência de estilo com uma paisagem.....	62
Figura 48 — Transferência de estilo com um rosto.....	63
Figura 49 — Transferência de estilo com uma casa	63
Figura 50 — Transferência de estilo com um cão.....	63
Figura 51 — Transferência de estilo com um camelo	64
Figura 52 — Transferência de estilo com uma águia	64
Figura 53 — Transferência de estilo com um ganso.....	65
Figura 54 — Transferência de estilo com uma chita	65
Figura 55 — Excerto do <i>dataset</i> utilizado para o modelo_melhores.....	69
Figura 56 — Parâmetros para a criação da imagem.....	74
Figura 57 — <i>Prompts</i> utilizados na criação da imagem.....	74
Figura 58 — Imagem gerada pelo modelo original.....	75
Figura 59 — Antigos <i>prompts</i> ajustados aos novos modelos	75
Figura 60 — Imagem gerada pelo modelo_styles com os <i>prompts</i> adaptados	76
Figura 61 — Imagem gerada pelo modelo_melhores com os <i>prompts</i> adaptados	76
Figura 62 — <i>Prompts</i> das próximas imagens a serem geradas	77
Figura 63 — Imagem gerada pelo modelo_melhores.....	77
Figura 64 — Imagem gerada pelo modelo_styles	78
Figura 65 — Adaptação dos <i>prompts</i> novos ao modelo original.....	79

Figura 66 — Imagem gerada pelo modelo original com os <i>prompts</i> adaptados	79
Figura 67 — Prompts sem a referência à qualidade da imagem	80
Figura 68 — Output obtido sem referência à qualidade nos <i>prompts</i>	80
Figura 69 — <i>Prompts</i> sem descrição do estilo do bordado de Castelo Branco	81
Figura 70 — <i>Output</i> sem descrição do estilo do bordado de Castelo Branco	81
Figura 71 — <i>Prompts</i> para o estudo dos parâmetros	85
Figura 72 — Teste para <i>DPM++2M</i>	85
Figura 73 — Teste para <i>DPM++ 2M SDE</i>	86
Figura 74 — Teste para <i>DPM fast</i>	86
Figura 75 — Teste para <i>DPM ++2M SDE Heun</i>	86
Figura 76 — Teste para <i>DPM ++ 25 a</i>	87
Figura 77 — Teste para <i>DPM ++ 3M SDE</i>	87
Figura 78 — Teste para <i>Euler a</i>	87
Figura 79 — Teste para <i>Euler</i>	88
Figura 80 — Teste para <i>LMS</i>	88
Figura 81 — Teste para <i>Heun</i>	88
Figura 82 — Teste para <i>DPM2</i>	89
Figura 83 — Teste para <i>DPM2 a</i>	89
Figura 84 — Teste para <i>DPM adaptive</i>	89
Figura 85 — Teste para <i>restart</i>	90
Figura 86 — Teste para <i>DDIM</i>	90
Figura 87 — Teste para <i>DDIM CFG ++</i>	90
Figura 88 — Teste para <i>PLMS</i>	91
Figura 89 — Teste para <i>UniPC</i>	91
Figura 90 — Teste para <i>LCM</i>	91
Figura 91 — 4 Imagens geradas com <i>LMS</i> com <i>seeds</i> aleatórias	93
Figura 92 — 4 Imagens geradas com <i>DPM ++2M</i> com <i>seeds</i> aleatórias	94
Figura 93 — Teste com <i>Exponential</i>	95
Figura 94 — Teste com <i>Karras</i>	95
Figura 95 — Teste com <i>Polyexponential</i>	95
Figura 96 — Teste com <i>SGM Uniform</i>	96
Figura 97 — Teste com <i>KL Optimal</i>	96
Figura 98 — Teste com <i>Align Your Steps</i>	96
Figura 99 — Teste com <i>Simple</i>	97
Figura 100 — Teste com <i>Normal</i>	97
Figura 101 — Teste com <i>DDIM</i>	97
Figura 102 — Teste com <i>Beta</i>	98
Figura 103 — 4 Imagens geradas com <i>kl optimal</i> com <i>seeds</i> aleatórias	99
Figura 104 — 4 Imagens geradas com <i>simple</i> com <i>seeds</i> aleatórias	99
Figura 105 — 4 Imagens geradas com <i>normal</i> com <i>seeds</i> aleatórias.....	100
Figura 106 — 4 Imagens geradas com <i>automatic</i> com <i>seeds</i> aleatórias.....	100
Figura 107 — Teste com <i>cfg_scale = 1</i>	103
Figura 108 — Teste com <i>cfg_scale = 5</i>	103

Figura 109 — Teste com <i>cfg_scale</i> = 10	104
Figura 110 — Teste com <i>cfg_scale</i> = 15	104
Figura 111 — Teste com <i>cfg_scale</i> = 20	105
Figura 112 — Teste com <i>cfg_scale</i> = 25	105
Figura 113 — Teste com <i>cfg_scale</i> = 30	106
Figura 114 — Teste com <i>cfg_scale</i> = 7 e <i>seed</i> aleatória.....	107
Figura 115 — Teste com <i>cfg_scale</i> = 9 e <i>seed</i> aleatória.....	107
Figura 116 — Teste com <i>cfg_scale</i> = 11 e <i>seed</i> aleatória.....	108
Figura 117 — Teste com <i>cfg_scale</i> = 13 e <i>seed</i> aleatória.....	108
Figura 117 — Teste com <i>steps</i> = 1 - Tempo de geração = 3.3 sec	110
Figura 118 — Teste com <i>steps</i> = 25 - Tempo de geração = 34.7 sec	110
Figura 119 — Teste com <i>steps</i> = 50 - Tempo de geração = 1 min 7.7 sec	111
Figura 120 — Teste com <i>steps</i> = 75 - Tempo de geração = 1 min 40.5 sec	111
Figura 121 — Teste com <i>steps</i> = 100 - Tempo de geração = 2 min 20.1 sec	112
Figura 122 — Teste com <i>steps</i> = 125 - Tempo de geração = 2 min 47.2 sec	112
Figura 123 — Teste com <i>steps</i> = 150 - Tempo de geração = 3 min 25.7 sec	113
Figura 124 — Imagem base para estudo dos parâmetros <i>img2img</i>	115
Figura 125 — <i>Prompts</i> para as gerações <i>img2img</i>	115
Figura 126 — Teste para <i>denoising_strength</i> = 0,25	116
Figura 127 — Teste para <i>denoising_strength</i> = 0,50	116
Figura 128 — Teste para <i>denoising_strength</i> = 0,75	117
Figura 129 — Teste para <i>denoising_strength</i> = 1	117
Figura 130 — Teste para <i>denoising_strength</i> = 0,55	118
Figura 131 — Teste para <i>denoising_strength</i> = 0,60	119
Figura 132 — Teste para <i>denoising_strength</i> = 0,65	119
Figura 133 — Teste para <i>denoising_strength</i> = 0,70	120
Figura 134 — Teste para <i>denoising_strength</i> = 0,625.....	121
Figura 135 — <i>Prompts</i> para a geração de imagens com um cavalo.....	123
Figura 136 — Resultados da criação de imagens com um cavalo	123
Figura 137 — <i>Prompts</i> para a geração de imagens com um peixe.....	124
Figura 138 — Resultados da criação de imagens com um peixe	124
Figura 139 — <i>Prompts</i> para a geração de imagens com uma borboleta.....	124
Figura 140 — Resultados da criação de imagens com uma borboleta.....	125
Figura 141 — <i>Prompts</i> para a geração de imagens com uma águia	125
Figura 142 — Resultados da criação de imagens com uma águia.....	126
Figura 143 — <i>Prompts</i> para a geração de imagens com uma cadeira	126
Figura 144 — Resultados da criação de imagens com uma cadeira.....	127
Figura 145 — <i>Prompts</i> para a geração de imagens com um lápis	127
Figura 146 — Resultados da criação de imagens com um lápis	128
Figura 147 — <i>Prompts</i> para a geração de imagens com uma tesoura.....	128
Figura 148 — Resultados da criação de imagens com uma tesoura	129
Figura 149 — <i>Prompts</i> para a geração de imagens com um teclado.....	129
Figura 150 — Resultados da criação de imagens com um teclado	130

Figura 151 — <i>Prompts</i> para a geração de imagens com uma rosa	130
Figura 152 — Resultados da criação de imagens com uma rosa	131
Figura 153 — <i>Prompts</i> para a geração de imagens com uma pedra	131
Figura 154 — Resultados da criação de imagens com uma pedra	132
Figura 155 — <i>Prompts</i> para a geração de imagens com uma árvore	132
Figura 156 — Resultados da criação de imagens com uma árvore.....	133
Figura 157 — <i>Prompts</i> para a geração de imagens com uma paisagem.....	133
Figura 158 — Resultados da criação de imagens com uma paisagem.....	134
Figura 159 — <i>Prompts</i> para as gerações <i>img2img</i>	136
Figura 160 — Imagem base para o primeiro teste	136
Figura 161 — <i>Output</i> gerado para <i>steps</i> = 20	137
Figura 162 — <i>Output</i> gerado para <i>steps</i> = 35	137
Figura 163 — <i>Output</i> gerado para <i>steps</i> = 50	138
Figura 164 — Imagem base para o segundo teste	138
Figura 165 — <i>Output</i> gerado para <i>steps</i> = 20.....	139
Figura 166 — <i>Output</i> gerado para <i>steps</i> = 35.....	139
Figura 167 — <i>Output</i> gerado para <i>steps</i> = 50.....	140
Figura 168 — Imagem base para o terceiro projeto	140
Figura 169 — <i>Output</i> gerado para <i>steps</i> = 20.....	141
Figura 170 — <i>Output</i> gerado para <i>steps</i> = 35.....	141
Figura 171 — <i>Output</i> gerado para <i>steps</i> = 50.....	142
Figura 172 — <i>AppBar</i> para a seleção do modo	146
Figura 173 — Exemplo da seleção de modelos.....	146
Figura 174 — Exemplo da interface da <i>tab txt2img</i>	147
Figura 175 — Exemplo do uso da aplicação no modo <i>text2img</i>	148
Figura 176 — Exemplo da interface da <i>tab img2img</i>	149
Figura 177 — Exemplo de como carregar uma imagem	149
Figura 178 — Exemplo do uso do modelo <i>img2img</i>	150
Figura 179 — Continuação do exemplo de uso do modelo <i>img2img</i>	150
Figura 180 — Funcionamento do protótipo em <i>txt2img</i> visualmente.....	151
Figura 181 — Imagem gerada pelos <i>prompts</i> em português.....	152
Figura 182 — Imagem gerada pelos mesmos <i>prompts</i> , mas em inglês	152

Lista de tabelas

Tabela 1 - Cronograma de tarefas do projeto I	5
Tabela 2 - Cronograma de tarefas do projeto II	5
Tabela 3 - Descrição dos Atributos	40
Tabela 4 - Resumo das características analisadas em cada artigo.....	40

Lista de abreviaturas, siglas e acrónimos

AIGC (*Artificial Intelligence Generated Content*)

API (*Application Programming Interface*)

BO (*Bayesian Optimization*)

CNN (*Convolutional Neural Network*)

DCGAN (*Deep Convolutional Generative Adversarial Network*)

DL (*Deep Learning*)

FID (*Fréchet Inception Distance*)

GAN (*Generative Adversarial Network*)

IA (*Inteligência Artificial*)

IES (*Interactive Evolutionary Systems*)

KID (*Kernel Inception Distance*)

LoRA (*Low-Rank Adaptation*)

LDM (*Latent Diffusion Model*)

ML (*Machine Learning*)

NPR (*Non-Photorealistic Rendering*)

RNN (*Recurrent Neural Network*)

SEAN (*Semantic Image Synthesis with Controllable Layered Feature Decomposition*)

SD (*Stable Diffusion*)

SGDM (*Style-Guided Diffusion Model*)

StySim (*Style Similarity*)

T2I (*Text-to-Image*)

VAE (*Variational Autoencoder*)

1. Introdução

A transferência de estilos é uma área englobada dentro da visão computacional [1] que consiste em transportar características estilísticas de um domínio para outro mantendo os elementos do segundo [2]. Resulta assim numa fusão entre duas realidades, permitindo resultados onde a estrutura da imagem inicial fica enriquecida com os elementos artísticos do estilo de preferência.

Devido à natureza complexa dos estilos artísticos, esta tarefa é muito desafiadora, uma vez que têm de ser detetados inúmeros detalhes como cor, textura e até possíveis geometrias presentes no estilo. No entanto, o grande número de avanços dentro da área da Inteligência Artificial, incluindo recentemente os modelos de difusão, como o *Stable Diffusion*, têm alavancado um rápido e positivo avanço na forma como a transferência de estilos é realizada, permitindo ter resultados convincentes [3].

O uso destas tecnologias torna possível explorar novas formas de expressão artística, integrando o património cultural tradicional com ferramentas digitais inovadoras, abrindo caminho para a reinterpretação e preservação de estilos únicos em contextos contemporâneos.

Ferramentas baseadas em *Stable Diffusion* demonstram um grande potencial para a criação e personalização de imagens, permitindo ao utilizador adaptar qualquer imagem ao estilo desejado, ou mesmo criar das zero composições alinhadas com referências específicas. Este projeto surge, assim, no cruzamento entre arte tradicional e tecnologia, propondo-se a resolver o desafio de adaptar imagens contemporâneas ao estilo do bordado de Castelo Branco através de técnicas de inteligência artificial, explorando a interseção entre herança cultural e inovação digital.

Inicialmente durante a fase relativa a Projeto I, pretendeu-se analisar o sucesso de técnicas que utilizam modelos de difusão, como o *Stable Diffusion*, para realizar transferência de estilo, conferindo a imagens novas as características visuais do bordado de Castelo Branco. Para tal realizámos uma revisão sistemática do uso de destes modelos para transferência de estilo, criação um *dataset* que recolhesse o máximo de imagens possíveis do caso de estudo, identificar tecnologias que pudessem auxiliar o processo e desenvolver um modelo experimental, treinado com o *dataset* construído.

Com a transição para Projeto II, o objetivo principal foi continuar e melhorar o trabalho desenvolvido anteriormente, com foco em melhorias técnicas, realização de testes adicionais e desenvolvimento de um protótipo funcional que permitisse ao utilizador interagir com o modelo e gerar imagens com o novo estilo.

Este segundo momento incluiu como principais tarefas o refinamento do modelo, ajuste de parâmetros e testes adicionais, bem como o desenvolvimento de um protótipo que permitisse a experimentação prática por parte dos utilizadores. A avaliação contínua dos resultados e a adaptação das estratégias conforme necessário foram igualmente contempladas, garantindo a evolução do projeto e a obtenção de resultados cada vez mais robustos e alinhados com os objetivos iniciais.

1.1 Enquadramento

A utilização de Inteligência Artificial no mundo artístico tem sido um tópico bastante debatido [4]. A geração de conteúdos nunca vistos em segundos, quase indistinguíveis de criações humanas, levanta sérias questões sobre a criatividade, autoria e limites éticos destas tecnologias. O uso de ferramentas que abordam o uso de *Stable Diffusion* têm demonstrado um grande potencial para a criação de imagens com estilos complexos permitindo ao utilizador diversas personalizações numa imagem à sua escolha ou então a sua criação do zero.

O uso deste tipo de tecnologias, apesar do seu enorme potencial criativo e artístico, continua a ser um desafio para o utilizador comum devido vários fatores técnicos e práticos. Estas tecnologias estão, muitas vezes, por de trás de aplicações complexas, e requisitos de *hardware* bastante exigentes, o que acaba por afastar muitos utilizadores comuns deste tipo de ferramentas.

Além destas exigências, os processos de instalação [5] e configurações podem ser um verdadeiro obstáculo. A instalação de ferramentas como o *Automatic1111* geralmente requer a utilização de repositórios *Git*, o que pode causar dificuldades para utilizadores sem experiência técnica. Para além desta etapa, é também necessária a instalação de diversos módulos *Python* e outros pequenos componentes de *software* como o caso do *CUDA*, que facilmente geram problemas de dependências requerendo conhecimentos mais avançados.

Outro aspeto que pode agravar ainda mais as dificuldades já existentes é a necessidade de fazer uma gestão cuidada das aplicações instaladas, *drivers* e

bibliotecas presentes no computador. Pequenos erros, como uma escolha errada da versão do *Python* ou o uso de um *driver* mais antigo podem bloquear o uso da ferramenta, obrigando o utilizador a procurar por ajuda técnica.

1.2 Objetivos

No âmbito da unidade curricular de Projeto I, pretende-se analisar o sucesso de técnicas que dão uso a modelos de difusão, como é o caso de Stable Diffusion, para fazer uma transferência de estilo que permita dar as características estilísticas presentes no Bordado de Castelo Branco a uma outra imagem. Foram então delineados os principais objetivos para confirmar a viabilidade do uso desses modelos para a dada transferência de estilo:

- Fazer uma revisão sistemática do uso de modelos de difusão para transferência de estilo
- Criação de um dataset de referência que recolha o máximo de imagens possíveis do caso de estudo
- Identificar possíveis tecnologias que possam auxiliar o processo
- Criação de um modelo experimental, que, ao ser treinado com o dataset construído, seja capaz de gerar resultados promissores que lembrem aos elementos fundamentais do Bordado de Castelo Branco

No âmbito da unidade curricular de Projeto II, o principal objetivo passou por transformar o BordadoBOT num protótipo funcional e multiplataforma, acessível a uma grande variedade utilizadores. Tivemos também como objetivo a realização de um estudo sobre o efeito de cada um dos parâmetros disponíveis no *A1111* de modo a otimizar os resultados obtidos pelos modelos. Para alcançar este propósito, foram definidos os seguintes objetivos:

- Construir um novo dataset, mais pequeno e focado na qualidade das imagens, de modo a avaliar a capacidade de aprendizagem do modelo com dados de maior precisão.
- Treinar dois novos modelos, com o intuito de explorar oportunidades de melhoria dos resultados, recorrendo à otimização dos parâmetros utilizados durante a fase de treino.

- Aperfeiçoar os *prompts* e parâmetros envolvidos na geração das imagens, com o objetivo de maximizar a qualidade dos *outputs* gerados pelo modelo mais promissor entre os três treinados.
- Desenvolver um protótipo multiplataforma, acessível e intuitivo, capaz de utilizar os modelos treinados, permitindo que qualquer utilizador, independentemente dos seus conhecimentos técnicos, possa beneficiar das potencialidades do *Stable Diffusion* na criação de imagens estilizadas com o bordado de Castelo Branco.

1.3 Planeamento do Projeto

Com o objetivo de chegarmos ao objetivo delineado para Projeto I, este projeto foi dividido em 6 principais tarefas espalhadas pelo 1º semestre do ano letivo de 2024/2025.

- Tarefa A – **Elaboração do relatório:** A redação deste documento tem como objetivo documentar o trabalho e pesquisa desenvolvidos, estendendo-se assim por todo o semestre.
- Tarefa B – **Construção de um dataset:** Um dos principais objetivos deste projeto é a criação de um dataset de referência relacionado ao bordado de Castelo Branco.
- Tarefa C – **Revisão Sistemática:** Investigação da eficácia relacionada com o uso de modelos que utilizam *Stable Diffusion* para efetuar transferências de estilo.
- Tarefa D – **Procura por ferramentas:** Analisar quais as ferramentas mais comuns dentro da aplicação de modelos de difusão.
- Tarefa E – **Estudo das ferramentas:** Com base na análise anterior fazer um estudo sobre quais serão mais adequadas ao trabalho
- Tarefa F – **Resultados iniciais:** Através do uso do dataset construído e ferramentas mais promissoras, obter resultados iniciais de transferência de estilo

Tabela 1 - Cronograma de tarefas para projeto I

Tarefa	2025									
	Setembro		Outubro		Novembro		Dezembro		Janeiro	
	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena
A										
B										
C										
D										
E										
F										

Em projeto II, de modo a cumprir com todos os objetivos propostos, foram delineadas as seguintes tarefas a realizar durante o segundo semestre do ano letivo de 2024/2025:

- Tarefa G – **Continuação da elaboração do relatório** – A continuação da redação deste documento tem como objetivo estender documentação o trabalho feito continuando a partir do documento usado em Projeto I.
- Tarefa H – **Construção de um novo dataset** – A partir da base criada no semestre anterior, refinar o dataset de modo a apenas incluir as melhores imagens disponíveis.
- Tarefa I – **Treino de dois novos modelos** – Testagem de novos parâmetros de treino, resultando na criação de dois modelos, um treinado com base nas melhores imagens e outro utilizando novas *strings* de treino.
- Tarefa J – **Melhorias nos prompts e parâmetros** – Experimentação intensiva de diversas combinações de *prompts* e parâmetros de modo a melhorar os resultados obtidos.
- Tarefa K – **Desenvolvimento do protótipo** – Criação de uma aplicação multiplataforma acessível para a interação com os modelos treinados

Tabela 2 - Cronograma de tarefas para projeto II

Tarefa	2025									
	Fevereiro		Março		Abril		Maio		Junho	
	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena
G										
H										
I										
J										
K										

1.4 Estrutura do Relatório

Este relatório é constituído por 12 capítulos. Neste primeiro capítulo, o objetivo é identificar o problema que nos dispomos a resolver e assim como a sua possível solução. É também apresentada uma introdução ao trabalho, assim como o enquadramento e as principais tarefas delineadas para ambas as fases relativas a projeto I e II.

O segundo capítulo, tem como função apresentar as bases teóricas que foram necessárias para o desenvolvimento da nossa solução. Vão ser principalmente abordados temas relacionados com inteligência artificial, sendo este o principal tópico do projeto. Dentro da área, vai ser dada ênfase a dois principais tópicos fundamentais para a compreensão dos modelos utilizados, nomeadamente *Machine Learning* e o uso de modelos de difusão. Estes irão desempenhar um papel chave na nossa abordagem ao problema.

No terceiro capítulo vai ser feita a análise do estado da arte, onde serão exploradas e analisadas as diversas soluções já existentes que dão uso a modelos de *Stable Diffusion* para a transferência de estilos. Neste capítulo vão ser descritas as fontes utilizadas para a pesquisa dos artigos relevantes, assim como os critérios para a inclusão dos mesmos no nosso estudo. Após a análise dos mesmos, vão ser tiradas as principais conclusões e feito um balanço dos resultados obtidos.

No quarto capítulo, é feita uma análise mais detalhada acerca da transferência de estilo. O seu principal objetivo é esclarecer o que realmente é a transferência de estilos, assim como analisar outras técnicas usadas para alcançar o objetivo de transportar os elementos estilísticos de umas imagens para outras. É também feita uma reflexão sobre a ética deste tipo de métodos, assim como dado um exemplo concreto de transferência de estilo com o uso de modelos de difusão.

No quinto capítulo, vamos fazer uma análise ao nosso caso de estudo, o bordado de Castelo Branco, para dar a conhecer um pouco da sua história e influência na região. Ainda neste capítulo vai ser abordado em mais detalhe o procedimento utilizado para a construção do nosso dataset. Iremos descrever a recolha das imagens e preparação das mesmas para poderem ser utilizadas com treino para o nosso modelo, bem como algumas dificuldades encontradas neste processo.

No sexto capítulo vamos listar todas as tecnologias/ferramentas que utilizámos para alcançar os nossos objetivos, apresentando a sua função na nossa solução.

No sétimo capítulo, é onde pretendemos apresentar os nossos resultados através da demonstração de algumas imagens geradas pelo nosso modelo após o treino. Esta demonstração vai incluir imagens geradas por diversas formas: através de *img2img* onde o modelo terá de efetuar a transferência de estilo à imagem fornecida, e também através de *txt2img* para ser possível avaliar a capacidade do modelo em criar imagens que tenham os elementos do bordado de Castelo Branco. Após a apresentação das imagens vai ser feito um balanço dos resultados obtidos.

O capítulo oitavo é dedicado aos novos modelos e *prompts* melhorados, detalhando as principais estratégias adotadas para melhorar o desempenho comparativamente com o modelo original. Nesta secção, são exploradas as principais decisões tomadas no processo de treino que levaram a um aumento de qualidade nos *outputs* gerados. Assim como a explicação para a inclusão de novos *tokens* nos *prompts*.

No capítulo nove, é feita uma análise aprofundada parâmetros disponíveis no *Automatic1111*, explorando diversas configurações através de testes experimentais. Este capítulo apresenta exemplos práticos de como cada parâmetro influencia a geração de imagens, tanto no modo *txt2img* como no *img2img*. Incluindo ainda uma análise crítica dos resultados obtidos em cada configuração, evidenciando os pontos fortes e limitações das diferentes opções disponíveis.

O décimo capítulo reúne os resultados gerados a partir da aplicação dos melhores parâmetros identificados no capítulo anterior, realizando testes mais variados e focados em situações práticas e reais. O objetivo principal é avaliar se a otimização dos parâmetros confere maior flexibilidade ao modelo, permitindo-lhe adaptar-se a diferentes tipos de *prompts* e cenários mais difíceis.

No capítulo onze, o foco está sobre o desenvolvimento do protótipo, explicando as decisões técnicas e de *design* tomadas em cada etapa. Através de capturas de ecrã da aplicação em funcionamento num simulador *Android* demonstrando o fluxo de utilização e a interação do utilizador com a aplicação. É ainda também explicada a arquitetura da aplicação, incluindo a forma como os *prompts* são geridos e enviados para o modelo.

O capítulo doze reúne as principais conclusões do projeto. Neste, são apresentadas as dificuldades que surgiram durante o desenvolvimento, os pontos positivos alcançados e as limitações encontradas.

Este documento resulta da junção das unidades curriculares de Projeto I e Projeto II, apresentando os capítulos 1 a 7 com pequenas atualizações relativamente à versão final apresentada em Projeto I. Os capítulos 8 a 12 são novos e correspondem ao trabalho desenvolvido durante o Projeto II, refletindo o aprofundamento e novas melhorias realizadas nesta fase.

2. Enquadramento Teórico

O objetivo deste capítulo é aprofundar os conhecimentos de forma teórica, apresentando as tecnologias e metodologias relevantes para a transferência de estilo do Bordado de Castelo Branco para outra imagem. Além disso procura-se fundamentar as bases relacionadas com Inteligência Artificial, *Machine Learning*, Modelos de Difusão e o uso de *Stable Diffusion* que vão ser fundamentais para o entendimento do processo aplicado a cada imagem.

2.1 Inteligência Artificial

A inteligência artificial é uma área que engloba vários campos incluindo *Machine Learning*, *Deep Learning*, processamento de linguagem natural, robôs inteligentes entre outros. O seu principal objetivo é através de diversas técnicas e algoritmos, conseguir resolver problemas simples da vida humana como tomadas de decisão e resolução de problemas, baseando-se principalmente em conceitos como lógica, probabilidade e matemática. O problema reside no facto de não podemos fornecer a capacidade de raciocínio às máquinas. Assim, será necessário criar uma forma de simular ou copiar o funcionamento do cérebro humano [6]. Permitindo assim a estas tecnologias otimizar e automatizar tarefas.

Com o avanço das tecnologias, é possível, atualmente, ter acesso a múltiplas ferramentas capazes de fornecer suporte a diversas atividades humanas. O que nos interessa principalmente é capacidade de observação e aprendizagem de um estilo específico. Ou seja, neste contexto, não será apenas utilizada para a deteção dos elementos visuais extraídos de uma imagem, mas também para transportar esses elementos para outras imagens possibilitando a criação de novas realidades artísticas. Com isto, a inteligência artificial sai do mundo analítico do qual foi inicialmente desenhada para uma nova realidade onde pode ser parte do mundo artístico.

Machine Learning surge quando permitimos à máquina aprender por si [7] através de dados fornecidos, de forma a conseguir realizar previsões e análises. A técnica mais explorada neste trabalho vai ser o uso de *Latent Diffusion Models* que através de estudos recentes [8] demonstrou obter melhores resultados quando comparada com abordagens anteriores, este modelo vai utilizar técnicas tais como *Deep Learning* através de redes neurais profundas para transformar as imagens para num formato que as redes possam entender e interpretar. Estas imagens passarão para o Espaço Latente, onde ficarão extremamente comprimidas, mas mantendo os detalhes necessários para uma transferência de estilo eficaz, o que diminui a quantidade de recursos computacionais necessários [9], temas estes serão explorados mais a fundo nos próximos capítulos.

2.1.1 Definição e Conceitos

A inteligência artificial é atualmente uma das áreas mais estudadas no mundo da ciência da computação [10]. Este ramo permite que as máquinas se comportem de forma inteligente, permitindo-lhes lidar autonomamente com problemas cada vez mais diversos. Resumidamente o termo inteligência artificial refere-se a sistemas ou máquinas que simulam a inteligência humana para realizar tarefas e podem melhorar iterativamente com base na informação que recebem.

Esta área não é apenas um tópico da ciência da computação, mas também filosófico, uma vez que, desde o início, se abordam questões sobre a possibilidade de dotar as máquinas da capacidade de pensar como seres humanos. Desde cedo foram definidos dois principais tipos de inteligência artificial. O primeiro, mais simples e frequentemente referido como *Weak AI* (ou IA Fraca) tem como principal objetivo realizar uma função específica. Este tipo de soluções é capaz de resolver problemas específicos dentro de um escopo limitado. Normalmente é utilizado para tarefas simples que, por vezes, podem causar dificuldades aos humanos [11]. É possível encontrar a IA Fraca em tarefas como reconhecimento de padrões, análises médicas, e jogos de estratégia, mas não tem entendimento do que está para além daquele cenário, nem compreende perfeitamente a tarefa que executa, limita-se responder com base na informação com o qual foi treinada [12].

Por outro lado, a *Strong AI* (ou IA Forte) consiste na possibilidade das máquinas terem a capacidade de raciocínio e compreensão equivalente à humana, sendo capazes de analisar uma série de parâmetros diferentes e tomar decisões em diversos contextos, sem depender de um conjunto de regras predefinidas ou de um domínio controlado. Esta não é uma área existente, mas sim uma possibilidade que tem sido debatida desde o início do estudo da inteligência artificial [12]. Com o passar dos anos e devido aos enormes avanços desta área, a IA Fraca tornou-se altamente avançada, sendo possível encontrar ferramentas amplamente utilizadas nas mais diversas áreas do conhecimento, já estando presente em sistemas de reconhecimento de voz, traduções automáticas, carros autónomos e ferramentas de recomendação.

2.1.2 Evolução e Aplicações

A inteligência artificial é uma das áreas mais promissoras na ciência da computação, tendo como principal foco o desenvolvimento de sistemas que realizam tarefas que normalmente exigem inteligência humana. Este sistema tem como objetivo aprender e melhorar iterativamente utilizando métodos que atualmente denominados de *Machine Learning*. Estes algoritmos têm como principal objetivo analisar grandes quantidades de dados e aprender com esses dados, através da detecção de padrões e extração de informações.

Alan Turing, um dos pioneiros deste campo, é amplamente conhecido pela criação do conceito do Teste de *Turing*. Neste teste, originado no seu famoso artigo seminal “*Computing Machinery and Intelligence*” (1950), é levantada uma questão bastante pertinente: “Serão as máquinas capazes de pensar?”. Nesse artigo, ele apresenta uma série de argumentos em que, em vez de se focar no conceito de “pensar”, seria mais eficaz criar um teste prático que respondesse à sua questão original. Toda esta linha de pensamento foi o que deu origem ao que hoje conhecemos como o Teste de *Turing*, no qual a premissa original é testar a inteligência da máquina através da análise da sua capacidade de enganar um interlocutor humano. Ou seja, se uma máquina conseguisse, de forma convincente, passar-se por humana, aos olhos de Alan Turing, faria com que se tornasse inteligente [13] [14]. Apesar de inovador para altura, este teste demonstrou-se ser ineficiente e até injusto devido à natureza humana.

De seguida, já nos anos 80, foram feitos grandes avanços dentro da área do *Machine Learning*, devido à introdução de novos e mais complexos algoritmos, como o caso da retropropagação. Este novo método permitiu à inteligência artificial aproximar-se do que é o cérebro humano, uma vez que usa uma arquitetura semelhante à dos nossos neurónios. Através da receção de dados de entrada, estes neurónios artificiais aplicam funções de ativação gerando assim valores de saída. Estas inovações dos anos 80 fazem parte hoje do núcleo do que entendemos por IA [15].

2.2 Machine Learning

A conseqüente evolução dentro desta área abriu espaço para a área de *Machine Learning* tornar-se uma das mais impactantes subáreas dentro da inteligência artificial, estando atualmente presente em diversas áreas de operação, como saúde e finanças, sendo principalmente utilizada pela sua capacidade de análise de dados [16]. O que capacitou este crescimento, foi a sua especialização na detecção de padrões em vez da aplicação de regras pré-estabelecidas, como ocorre em outras áreas das ciências da computação, onde um agente aplica as regras num conjunto para obter respostas. Já em *Machine Learning*, esse processo varia um pouco, pois

ao fornecer o nosso conjunto de dados e as respostas, o modelo irá nos devolver as regras, promovendo assim a análise dos dados, permitindo tirar conclusões muitas vezes escondidas entre os dados e de difícil análise para nós, seres humanos [17].

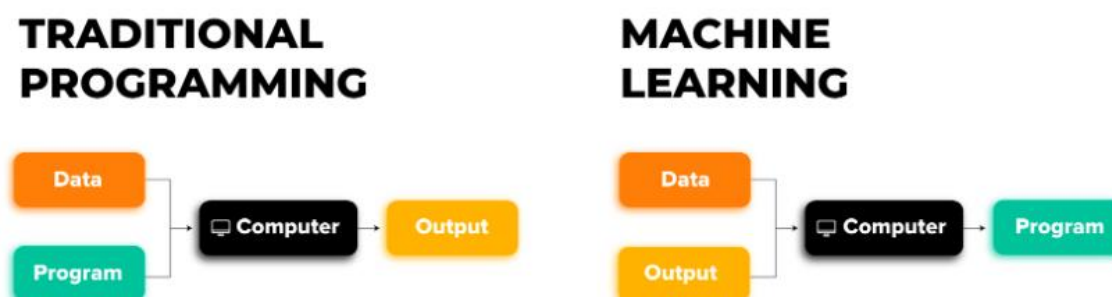


Figura 1 – Comparação entre programação tradicional e *Machine Learning* (Retirado de: [18])

2.2.1 Tipos de Aprendizagem

Para os modelos de *Machine Learning* conseguirem fazer esta análise dos dados, é fundamental que primeiro sejam treinados. O processo de treino pode ser feito utilizando diferentes estratégias, cada uma com as suas aplicações e pontos fortes específicos. Esses tipos diferentes podem ser classificados como:

2.2.1.1 Aprendizagem Supervisionada

Ao utilizar esta estratégia de treino, estamos a empregar um conjunto de dados que possuem uma característica alvo, onde todos os dados estão rotulados com um valor dessa dada característica. O que faz com que todas as entradas possuem uma saída, permitindo, com base nas entradas recebidas, que o modelo faça uma previsão do atributo alvo[19].

2.2.1.2 Aprendizagem Não Supervisionada

Nesta abordagem, os dados já não possuem rótulos nem um atributo que possamos considerar como objetivo. Aqui, o trabalho do modelo é interpretar os dados na totalidade e tentar encontrar padrões ou regras de agrupamento, que permitam separar a totalidade dos dados em diferentes grupos com características semelhantes entre eles [19].

2.2.1.3 Aprendizagem Por Reforço

Na aprendizagem por reforço, o modelo vai aprender com base no feedback recebido das interações com ambiente. Diferentemente de outras abordagens, aqui não existem rótulos; em vez disso, o modelo vai explorar os dados e, conforme os resultados obtidos, vai modificar a sua estratégia e alterar ações de modo a maximizar as recompensas positivas[19].

2.2.2 Redes Neurais

As redes neurais (ou ANNs) são uma classe de algoritmos associados ao *Machine Learning* que têm como principal tarefa a detecção de padrões e aprendizagem com base numa grande quantidade de dados. A arquitetura destes modelos, tal como nos primeiros tempos de inteligência artificial, vem da biologia humana, que, de forma muito simplificada, tenta simular o comportamento de um neurónio humano [20]. É possível verificar esta inspiração na forma como as camadas estão interligadas.

O funcionamento deste mecanismo envolve principalmente a organização de três componentes essenciais: a camada de entrada, uma (ou várias) camadas ocultas, e por fim a camada de saída. A camada de entrada tem como principal função receber os dados e, conseqüentemente, transmiti-los para a(s) camadas seguintes. Na(s) camada(s) oculta(s) serão realizadas uma série de operações matemáticas.

Estas operações variam entre somas ponderadas e a aplicação de várias funções de ativação, como *ReLU (Rectified Linear Unit)*, *Sigmoid* ou *Tahn*. As funções de ativação são de grande importância, uma vez que o seu principal papel é capturar e interpretar as relações não lineares entre os diversos atributos dos dados, garantindo assim que consiga lidar facilmente com dados onde a variação é maior [21]. Após o processamento nas camadas ocultas, o resultado das operações é passado para a última camada, a camada de saída, que, com base nos valores recebidos, vai gerar um resultado.

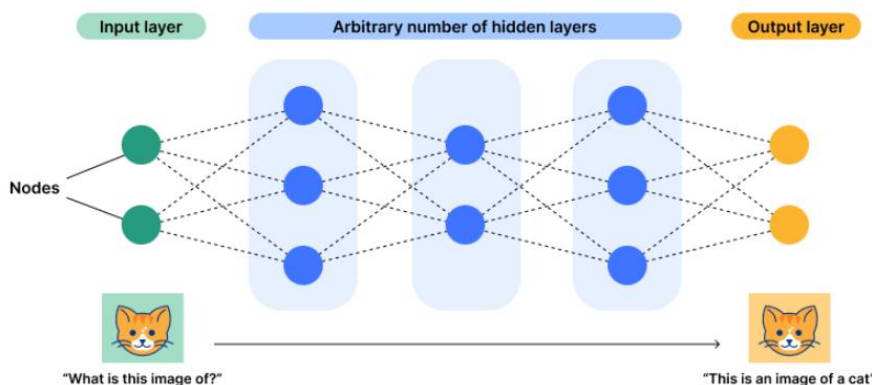


Figura 2 – Exemplo de uma rede neuronal (Retirado de: [22])

2.2.3 Deep Learning

Deep Learning é uma subárea do *Machine Learning* que tem como principal foco o uso e melhoramento de redes neuronais artificiais em diversas tarefas como classificação, regressão e reconhecimento de padrões. Nesta subárea o principal foco está centrado em redes neuronais profundas (com 3 ou mais camadas de neurónios) e as suas capacidades de realizar essas tarefas.

Avanços recentes dentro desta área são os principais fatores para existência de modelos generativos tão eficazes como os que temos atualmente, inovações como *Transformers* e *Autoencoders* Variacionais que têm um papel central nos modelos de processamento de linguagem natural mais usados [23]. Além disso, as evoluções das arquiteturas das redes neuronais, através da criação de redes convulsionais (*CNNs*) e redes recorrentes (*RNNs*), permitiram a obtenção de resultados muito positivos. Estas redes tornaram-se hoje os alicerces de modelos baseados em visão computacional, permitindo o seu uso nas mais diversas áreas onde a deteção de objetos é fundamental. Consolidaram-se assim, como ferramentas robustas dentro da medicina, transportes e segurança.

Os progressos feitos também na área do *hardware* não só proporcionaram uma nova capacidade computacional, mas também trouxeram diversas técnicas de otimização que nos permitiram superar limitações no treino [24].

2.3 Modelos de Difusão

Os modelos de difusão são sistemas probabilísticos que foram propostos por Jascha Sohl-Dickstein em 2015, inspirados pela física termodinâmica fora do equilíbrio. A sua conceção baseia-se principalmente na reversão de uma distribuição complexa de dados e na transformação desta numa nova distribuição

mais acessível, onde seria possível o seu tratamento e análise. O problema inicial que levou à ideia deste modelo vem do processo de difusão da física termodinâmica. Nele, ao colocar uma gota de um líquido colorido dentro de outro líquido, o primeiro, após o primeiro contacto com o segundo, acaba por ter uma distribuição completamente aleatória e impossível de prever [25]. Ou seja, por outras palavras, estes modelos foram criados com o objetivo inicial de prever a posição inicial de moléculas após entrarem em contacto com outro líquido pré-existente.

2.3.1 Definição e Conceitos

O núcleo do conceito destes modelos baseia-se na adição consecutiva de ruído nos dados através de um número finito de passos até a amostra inicial já não ser reconhecível, ficando apenas uma imagem de ruído puro. Em seguida, o modelo deve aprender a reverter este processo, ou seja, gerar uma nova imagem a partir de ruído puro e criar assim uma imagem clara. Este processo também é realizado progressivamente através de uma cadeia de *Markov*. Para alcançar esta capacidade de reversão do processo de difusão, o modelo, durante a sua fase de treino, será exposto a diferentes versões dos dados de treino com ruídos cada vez maiores. O que torna o processo completamente diferente de outros modelos semelhantes, como os *GANs*, é o fato de o modelo não aprender diretamente com os dados inseridos, mas sim aprender a estimar a melhor forma de remover o ruído e restaurar gradualmente a imagem, em vez de depender de uma competição adversária [25].

Durante o processo de treino, de forma a otimizar a aprendizagem, serão utilizadas métricas, como a divergência de *Kullback-Leibler* (métrica usada para medir a diferença entre duas distribuições de probabilidade) ou o erro quadrático médio, uma vez que a diminuição dos valores obtidos dessas métricas indicará que o modelo está a conseguir produzir novas imagens realistas com base no ruído fornecido [25].

2.3.2 Como Funcionam

Para descrever o funcionamento destes modelos, é necessário, primeiramente, descrever os seus principais componentes e etapas fundamentais para a criação de novas imagens, incluindo a formulação matemática e a estrutura das redes neuronais envolvidas em todo o processo. Estes modelos fundamentam-se principalmente em duas etapas distintas: o processo de difusão e o processo de difusão reversa [25].

Na primeira etapa, o processo de difusão será feito progressivamente, adicionando ruído Gaussiano ao longo de um número finito de passos T , seguindo a seguinte equação [26]:

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = N(\mathbf{x}_t; \sqrt{\alpha_t} \mathbf{x}_{t-1}, (1 - \alpha_t)I)$$

O que esta equação faz é, a cada passo no espaço de tempo T , a amostra x_t vai ser obtida através da versão anterior x_{t-1} aplicando uma transformação linear multiplicativa ($\sqrt{\alpha_t} x_{t-1}$), e somando ruído gaussiano, $((1 - \alpha_t) I)$. Este processo é fundamental para garantir que, após um número suficientemente grande de passos, a amostra original vai ser substituída por uma constituída de apenas ruído [26].

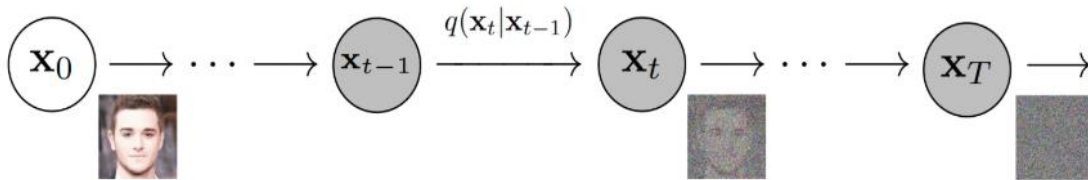


Figura 3 – Processo de difusão (Retirado de: [27])

Na segunda etapa, no processo de difusão reversa, o modelo aprende a reverter os passos feitos no processo de difusão, começando com um dado de ruído puro que será progressivamente refinado até chegar a uma imagem sem ruído. Esta reversão é feita através da seguinte distribuição condicional [26]:

$$p_\theta(x_{t-1} | x_t) = N(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

A partir desta distribuição, o modelo neuronal vai prever a média $\mu_\theta(x_t, t)$ e a covariância $\Sigma_\theta(x_t, t)$, permitindo que a amostra seja refinada gradualmente até atingir a distribuição dos dados reais, ou seja, a imagem sem ruído. A intuição por de trás desta equação é que o modelo aprende a prever como era a versão anterior, com menos ruído, utilizando um processo de aproximação estatística. O objetivo do modelo é, após a previsão, minimizar a função que quantifica a diferença entre o ruído previsto e ruído real através da seguinte função [26]:

$$\mathbb{E}_{x_0, t, \epsilon} [\|\epsilon - \epsilon_\theta(x_t, t)\|^2]$$

Esta função vai calcular a diferença entre o ruído gerado e a estimativa proveniente do processo de difusão reversa gerado pela rede neuronal profunda. Através da diminuição dos valores gerados, isso permite que o modelo consiga desfazer a difusão e assim gerar novos dados de qualidade através de ruído gaussiano puro.

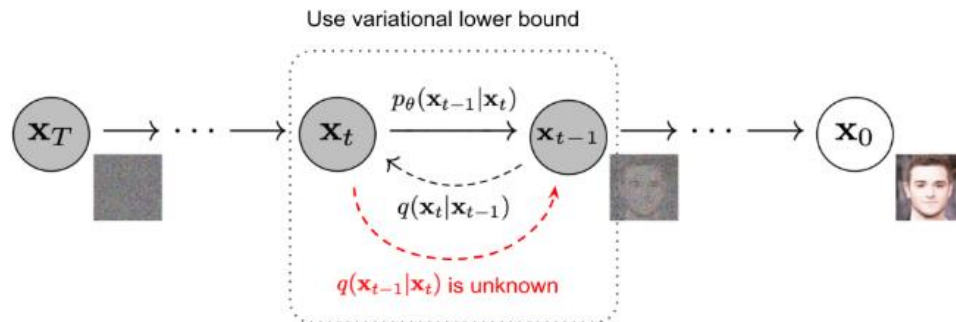


Figura 4 – Processo de difusão reversa (Retirado de: [27])

Na figura abaixo é possível visualizar de forma ilustrativa o funcionamento destes modelos, com o processo de difusão a amarelo, e o processo reverso em azul.

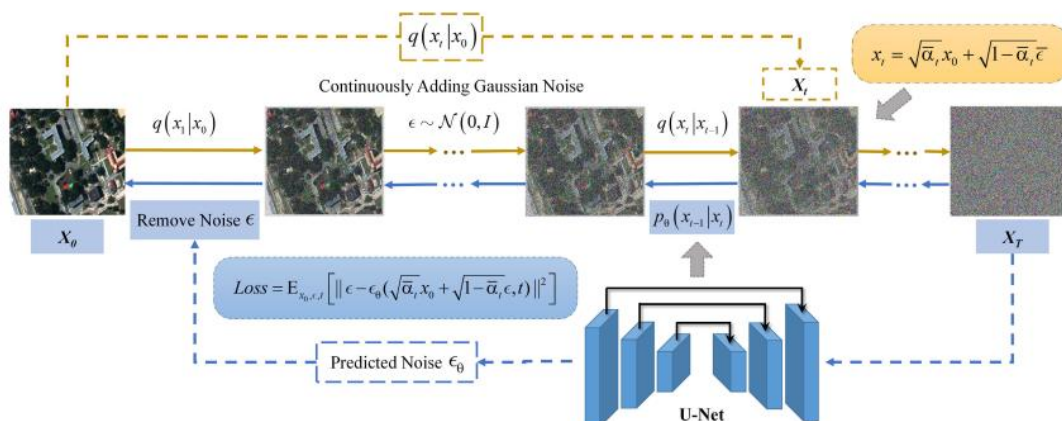


Figura 5 – Demonstração visual de ambos processos (Retirado de: [28])

Em suma, estes tipos de modelos focam-se na aplicação e remoção de ruído Gaussiano, característica essa que os difere de outros. Nestes modelos, o ruído é progressivamente adicionado e removido através de um processo probabilístico, permitindo a reconstrução de dados a partir de ruído. Enquanto isso, as GANs geram dados por meio da interação de um discriminador e um gerador, sendo a principal função do gerador criar dados enquanto o discriminador avalia-os [29]. O principal objetivo das VAEs é converter os dados para o espaço latente, onde são armazenadas as principais características, e reutilizando-as para a gerar novas imagens [30]. Os modelos baseados em fluxo seguem uma abordagem semelhante à dos modelos de difusão, mas, em vez de corromper os dados com ruído,

simplificam as imagens durante a fase de treino, com o objetivo de transformar as distribuições complexas das imagens originais, em versões mais simples [31].

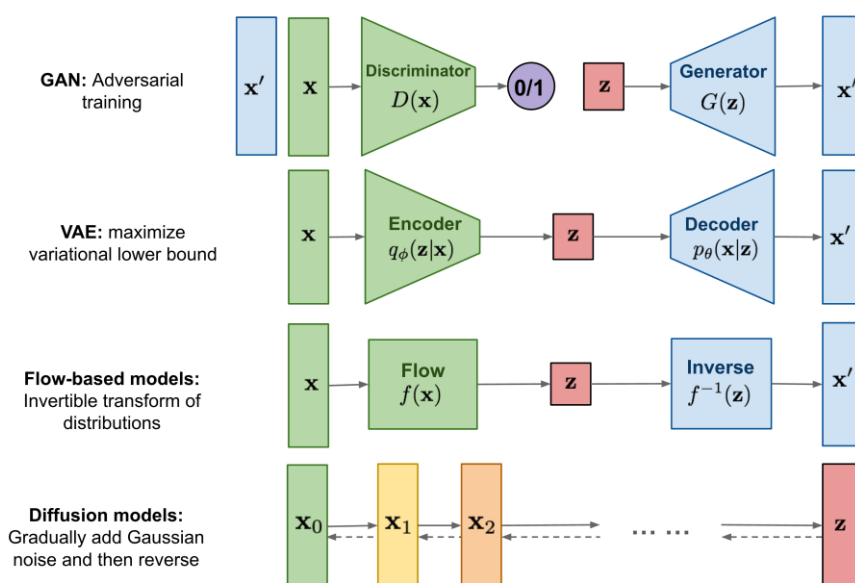


Figura 6 – Comparação visual entre diferentes modelos

2.4 Stable Diffusion

Stable Diffusion é uma abordagem mais recente e otimizada em relação aos tradicionais modelos de difusão. As principais diferenças estão na forma com os dados, ou neste caso as imagens, são transformadas em dados computacionais. Estes avanços introduziram o uso do espaço latente que simplifica o processamento das imagens. Estas deixam de ser transformadas com base nos valores de R, G e B dos pixels e são transformadas em vetores latentes que capturam de maneira mais generalizada as características principais da imagem, como textura, cor e padrões visuais [32].

2.4.1 Arquitetura

A arquitetura dos modelos de *Stable Diffusion* é composta por vários componentes que, em conjunto, permitem a criação de imagens de alta qualidade com grande eficiência computacional. Isto possibilita que os processos de difusão operem no espaço latente, ao invés de tratar cada pixel individualmente. Os seus principais componentes são:

2.4.2.1 VAE (Variational AutoEncoder)

O VAE, um tipo de rede neuronal, é adotado em modelos de *Stable Diffusion* pela sua capacidade de simplificação dos dados. Este componente é responsável por transformar os dados do mundo dos pixels para o espaço latente, sendo um dos principais fatores para a otimização computacional destes modelos [30].

Para serem representadas computacionalmente, as imagens precisam de ser transformadas em um formato interpretável. Normalmente, esta é feita e mantida no universo dos pixels. Ou seja, por exemplo, numa imagem 50x50, são precisas três matrizes de 50x50 onde cada uma representa o valor de R (*Red*), G (*Green*) e B (*Blue*) de cada pixel. Embora seja forma comum e viável de representar imagens, não é computacionalmente eficiente para o processo de difusão. Portanto, o *VAE* é o componente responsável por transformar uma imagem para o espaço latente. Neste novo espaço, as imagens são representadas por vetores numéricos que codificam as informações semânticas como imagem, textura, cor e padrões [30]. Tornando assim o tamanho da estrutura dados muito mais pequeno.

Para entender melhor o efeito deste componente, podemos analisar o seu funcionamento com o seguinte exemplo: uma imagem de 1024x1024, com um fator de redução de 4, vai ser transformada numa matriz de 256 por 256 valores, resultando num total de 262.144 valores presentes. Enquanto no universo dos pixels, precisaríamos de representar com três matrizes, cada uma de 1024 por 1024, dando um total de 3.145.728, oferecendo assim uma diminuição de cerca de 12 vezes o número total de dados.

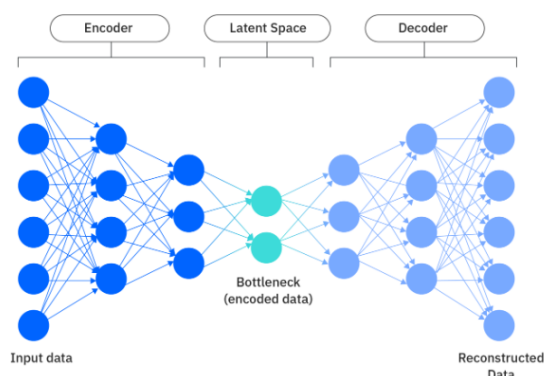


Figura 7 – Arquitetura de um VAE (Retirado de: [33])

2.4.2.2 LDM (*Latent Diffusion Model*)

O *LDM* será o nosso modelo de difusão, uma versão aprimorada dos modelos tradicionais de difusão. Nele, os dados são tratados no espaço latente, originados pelo processo do *VAE*. Este componente segue o mesmo princípio dos modelos de difusão, no qual o ruído é progressivamente adicionado aos dados até que se transformem em uma distribuição de ruído puro. Após esse processo, o modelo aprende a gerar, com base numa distribuição aleatória de ruído, novos dados limpos e de alta qualidade [8].

2.4.2.3 Rede U-Net

A rede *U-Net* é uma arquitetura baseada em redes neuronais profundas originalmente criada para a segmentação de imagens médicas, apesar de se demonstrado extremamente eficaz para o processamento das imagens, incluindo modelos generativos como é o caso de *Stable Diffusion* [34]. A sua estrutura é composta por dois principais elementos, um codificador e um decodificador.

O codificador vai progressivamente diminuir a dimensão da imagem enquanto ao mesmo tempo captura as principais características. Já o decodificador vai reconstruir a imagem preservando as características obtidas pelo codificador até ela ter o mesmo tamanho da original. Devido à natureza dos modelos de *Stable Diffusion*, esta rede vai operar por cima do espaço latente, ou seja, o redimensionamento que ocorre dentro de uma *U-Net* vai ser feito simultaneamente que o processo de difusão reversa, quando estes terminarem, o output vai ser colocado numa VAE e então é retornada uma imagem final visível [35].

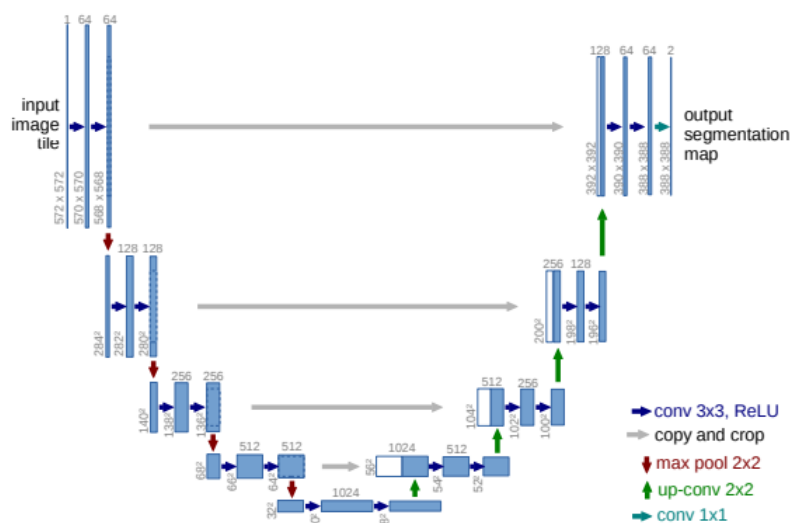


Figura 8 – Arquitetura de uma rede U-Net (Retirado de: [35])

2.4.2.4 CLIP (Contrastive Language-Image Pretraining)

O *CLIP* é um modelo capaz de processar vários tipos de dados como texto, imagens, vídeos e áudio permitindo as entender relações entre os diferentes tipos de inputs. Num contexto dos modelos de *Stable Diffusion*, o *CLIP* tem como principal função compreender o conteúdo dos *prompts* textuais. Após a interpretação do texto, o modelo irá mapear os prompts para vetores latentes, para depois serem processados pelo VAE [36].

Este componente utiliza duas redes neuronais: uma para processar o conteúdo visual e outra para as descrições textuais. Dessa forma, o modelo consegue criar

um alinhamento eficaz entre ambos, permitindo estabelecer correspondências entre o texto fornecido e as imagens geradas [36].

2.4.3 Processo de Treino

O processo de treino de modelos *Stable Diffusion* ocorre em vários passos. Primeiramente, é fornecida uma imagem de treino com uma quantidade aleatória de ruído ao modelo. A rede neuronal tenta então separar o conteúdo original da imagem do ruído. Com base na previsão do ruído, é possível subtraí-lo da imagem para obter uma estimativa de como seria a imagem sem ele. Em seguida, parte do ruído calculado na etapa anterior, por exemplo, 75% do ruído, é adicionado novamente à imagem prevista como limpa no passo anterior. Isso resulta em uma versão ligeiramente menos ruidosa da imagem original, que será utilizada como input na próxima etapa. O processo é repetido diversas vezes até que o número de passos de treino seja completado. Por exemplo, se for usado como parâmetro de treino 100 passos, o ciclo será repetido 100 vezes por cada imagem. A figura abaixo representa de forma ilustrativa como uma etapa do processo de treino funciona [37] [26].

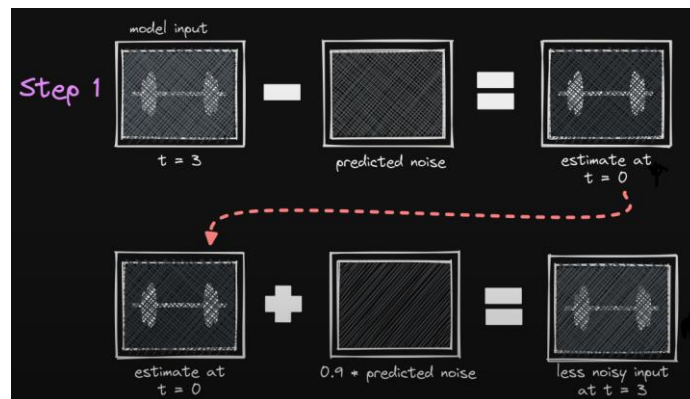


Figura 9 – Ilustração do processo de treino de um modelo de *Stable Diffusion* (Retirado de: [37])

3. Estudo do Estado de Arte

Nesta secção será feita uma revisão sistemática da literatura sobre o uso de modelos de *Stable Diffusion* na transferência de estilo. Esta revisão permite aos autores conhecer e compreender os trabalhos e investigações já feitas tornando assim possível explorar novas soluções, adaptar soluções já existentes, e/ou confirmar se a investigação já feita é replicável.

3.1 Metodologia e Processo de Pesquisa

Para a elaboração de uma revisão sistemática da literatura sobre o uso de modelos de *Stable Diffusion* vamos nos focar nos seguintes aspetos:

- Propósito e questões da investigação;
- Fonte utilizadas;
- Estratégia de pesquisa;
- Critérios de exclusão;
- Análise dos artigos encontrados;
- Conclusão e discussão dos resultados.

3.1.1 Objetivo da Pesquisa

Nesta revisão sistemática vamos investigar o uso de para treino de modelos, metodologias e arquiteturas de redes neurais empregadas em tecnologias *Stable Diffusion*, assim como fazer uma comparação direta com outras metodologias e tecnologias. Mantendo a ênfase na área da geração de imagens para compreender como diferentes abordagens podem impactar a coerência visual e os resultados obtidos.

Uma das motivações para a realização desta revisão é o rápido avanço das tecnologias de geração de conteúdos através de inteligência artificial, como é possível ver na Figura 10, e também entender como esta emergente abordagem diverge das restantes. Na figura 11 é possível visualizar uma comparação entre o uso de inteligência artificial regular com inteligência artificial generativa nas empresas, mais concretamente o número de organizações que adotaram estas técnicas para pelo menos uma função dentro da empresa.

Uso de IA generativa e IA regular nas empresas

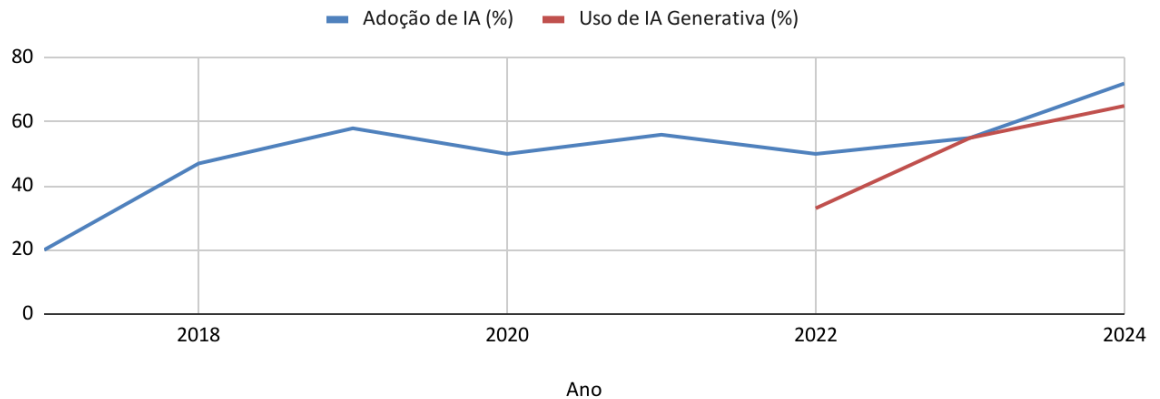


Figura 10 – Uso de inteligência artificial generativa regular nas empresas (Adaptado de:[38])

Numero de Artigos com o termo 'Stable Diffusion' desde 2000

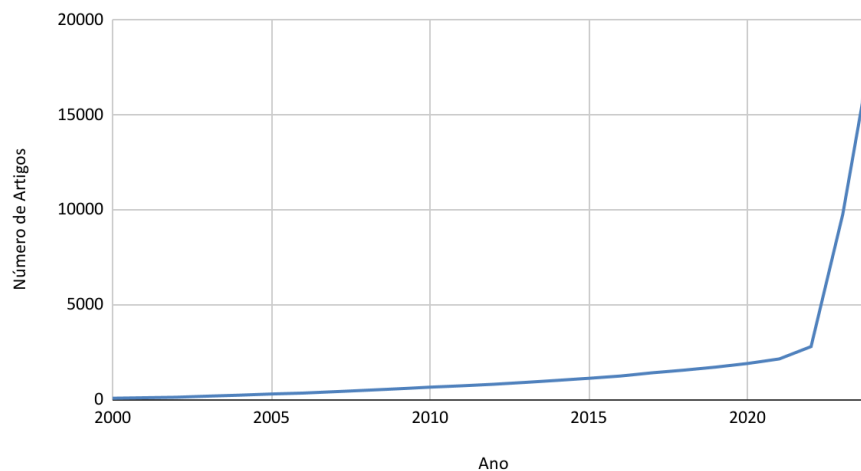


Figura 11 – Número de Artigos com o termo 'Stable Diffusion' (Fonte: Google Scholar)

3.1.2 Fontes Utilizadas

Para a pesquisa de trabalhos e estudos relevantes ao nosso tema, foram utilizadas duas principais plataformas *web* que disponibilizam acesso a documentos científicos, revistas, artigos, *e-books* e relatórios dos mais diversos temas dentro da área das ciências da computação.

A *IEEE Xplore* [39] é uma biblioteca digital disponibilizada pelo *Institute of Electrical and Electronics Engineers*, que fornece acesso a literatura técnica de alta qualidade relacionada sobre engenharia elétrica, ciência da computação, eletrônica e outras disciplinas relacionadas. Nesta plataforma é possível ter acesso a mais de 6 milhões de documentos técnicos.

A *B-On* [40] ou também conhecida como a Biblioteca do Conhecimento Online, foi criada pela FCCN (Fundação para a Computação Científica Nacional) em 2004, com a função de fornecer acesso permanente a diversos textos completos de milhares de publicações científicas. O seu principal objetivo é centralizar e ampliar o acesso à informação científica internacional.

Ambas as plataformas têm mecanismos para refinar a pesquisa, sendo possível a inclusão de operadores lógicos como *AND*, *OR* e *NOT* que se demonstraram altamente úteis para procura de artigos que não só faziam referência às tecnologias em questão, como também nos permitiu filtrar para os casos de uso mais relevantes. Tornando a análise em específico daqueles que tinham como foco a geração de imagens mais fácil, tópico relevante para o nosso objetivo.

3.1.3 Estratégia de Pesquisa

Com o objetivo de alcançar todos os possíveis artigos que se pudessem demonstrar relevantes ao estudo do estado da arte, é importante fazer pesquisas com base nas palavras-chaves que melhor identificam o nosso tema. No nosso caso, procuramos especificamente por artigos que combinam o uso de técnicas que usam *Stable Diffusion* para a geração de imagens e transferência de estilos. A *string* de pesquisa utilizada foi:

AB "Style Transfer" AND "Stable Diffusion"

Nesta pesquisa, selecionamos todos os artigos que mencionam 'transferência de estilo' e '*Stable Diffusion*' no resumo. Optámos por essa abordagem porque, em muitos casos, essas tecnologias eram citadas, mas não representavam o foco principal do trabalho desenvolvido. Como resultado, identificamos 21 artigos na plataforma *B-On*.

3.1.4 Critérios de Exclusão

O primeiro critério aplicado foi a remoção dos artigos que não foram escritos em inglês, este critério levou à remoção de um artigo que estava escrito em mandarim. O segundo critério aplicado foi o acesso ao texto integral, que eliminou um trabalho ficando com 19 artigos no total. Com o objetivo de alcançar apenas os artigos em que o foco principal era a transferência artística de estilos, removemos todos os artigos onde abordavam o uso destas técnicas para fins diferentes do desejado, isto inclui os artigos onde o foco é a medicina, biologia e a edição/alteração de vídeos através de modelos de difusão, sobrando assim 10 artigos que seguiriam à fase de análise.

3.2 Análise dos Artigos Encontrados

No artigo [41] os autores realizaram um teste baseado na geração de avatares animados utilizando tecnologias *GAN*. Eles propuseram um mecanismo de transferência de estilo capaz de intercalar dois estilos de avatares até alcançar uma fusão entre ambos. Para isso, utilizaram as *frameworks* *StyleRig* (uma técnica que combina *StyleGAN* com edição baseada em *rigging*) e *SEAN* (*Semantic Image Synthesis with Controllable Layered Feature Decomposition*). O *StyleRig* aprende um espaço de estilo controlável, ajustando múltiplas dimensões de parâmetros até atingir um estilo intuitivo e manipulável. Já o *SEAN* explora os estilos dos avatares por meio de aprendizagem não supervisionada.

Os autores destacam a importância da geração de avatares animados não supervisionados, um método que reduz a necessidade de um conjunto de dados para treino. Isso significa que essa tecnologia pode gerar avatares sem depender de amostras reais, dispensando um pré-treino, e realizando a geração de forma aleatória.

Além disso, exploram a criação de avatares animados multimodais, que podem ser gerados a partir de diferentes tipos de entrada, como voz e texto.

Para o treino, utilizaram a abordagem baseada na *DCGAN* (*Deep Convolutional Generative Adversarial Network*), uma arquitetura de redes neurais que conta com dois componentes principais: o gerador, cria as imagens a partir dos vetores de ruído, e o discriminador que avalia os dados criados pelo gerador e tenta diferenciar as amostras reais daquelas que foram criadas artificialmente.

O objetivo do estudo foi explorar o potencial da *DCGAN* na geração automática de avatares animados de alta qualidade. Os autores reconhecem, no entanto, que ainda há espaço para avanços, abrindo caminho para futuras pesquisas que aprimorem a transformação e fusão de estilos nos avatares animados.



Figura 12 – Comparação dos resultados obtidos pelos autores e outras abordagens (Retirado de: [41])

Os autores do artigo [42] procuraram criar imagens incorporadas em códigos QR de forma a manter o seu normal funcionamento. Para isso, desenvolveram o *Text2QR*, um método que equilibra a estética com a funcionalidade deste tipo de tecnologias. Como parte da solução, introduziram o módulo *QAB (QR Aesthetic Blueprint)*, que gera uma “*blueprint*” que faz a união da imagem desejada com o código QR. Após ter a união de ambos os elementos usam o *SELR (Enhancing Latent Refinement)* irá melhorar a combinação entre a imagem e código QR, mas mantendo o foco na funcionalidade dos códigos.

A *framework* utilizada pelos autores baseia-se em 3 etapas, primeiro os utilizadores geram as suas imagens utilizando o modelo de *Stable Diffusion*, enquanto simultaneamente escolhem uma mensagem para inserir via código QR, de seguida vai ser usado o módulo *QR Aesthetic Blueprint* que tem o conteúdo da imagem gerada refletindo a mensagem, e por fim é construída uma equação para quantificar o conteúdo e a mensagem que for gerada, resultando assim num código QR legível e esteticamente apelativo.

Em suma o *Text2QR* usa *Stable Diffusion* para transformar imagens em códigos QR absorvendo o estilo de uma imagem fornecida, uma maneira inovadora de estilizar uma tecnologia muito utilizada mantendo a capacidade de leitura intacta. Apesar de não fazerem referência ao treino, é um grande indicador da capacidade de tecnologias de *Stable Diffusion* em estilizar imagens.

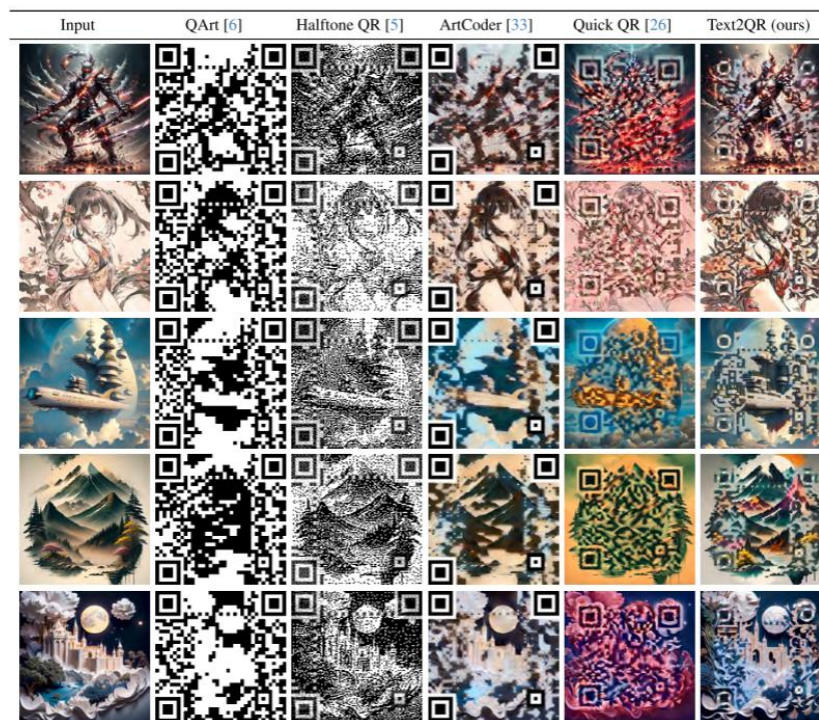


Figura 13 – Comparação de imagens geradas com outras alternativas (Retirado de: [42])

No artigo [43] os autores propõem-se a apresentar um modelo de difusão capaz de aprender e aplicar diversos estilos de forma eficiente e sem a necessidade de muitos dados para treino. Esta abordagem consiste principalmente em duas fases, na primeira é adaptar distribuição do ruído de modo a refletir melhor as características do estilo alvo. E de seguida um refinamento de um componente *U-Net* do *Stable Diffusion* vai utilizar esta nova distribuição mais eficaz, para o efeito pretendido.

Os autores demonstraram resultados significativos e muito acima das técnicas anteriormente utilizadas. Este tipo de tecnologia juntamente com um eficaz *Prompt Engineering* demonstram ter uma precisão muito alta na transferência de um estilo, dados os exemplos usados de esboços de *anime*, personagens do universo *Pokémon* e pinturas de Salvador Dalí.

Por fim, fazem referência a alguns pontos fracos como a herança de defeitos com base no modelo em que foi efetuado o treino, juntamente com a necessidade de ajustes manuais para uma otimização da qualidade dos resultados. Apesar de alguns impasses, continua a representar um avanço significativo devido à sua surpreendente velocidade de treino, e capacidade de diversas aplicações artísticas demonstrando assim, ter um grande potencial para a expandir o mundo a geração e personalização de imagens através do uso de inteligência artificial.



Figura 14 – Comparação de imagens geradas com outras abordagens (Retirado de: [20])

Os autores em [44] propuseram a junção de *IES* (*Interactive Evolutionary Systems*) e *BO* (*Bayesian Optimization*) para avaliar interactivamente um lote de imagens fornecidas, e com base nos parâmetros dados, o modelo de difusão iria gerar novas imagens adaptando as fornecidas com as características estilísticas pretendidas.

Dessa forma, é possível criar múltiplos designs alternativos através do *IES*, baseados em um conjunto reduzido de parâmetros. Os autores destacaram um exemplo de uma aplicação do modelo onde era gerado uma série de *designs* que

evoluem a partir de um número mínimo de parâmetros, seguindo um processo de mutação e seleção, similar à evolução biológica.

Essa abordagem expandiu-se para diversas aplicações que utilizam técnicas de interação com o utilizador, permitindo que participe ativamente da seleção ou avaliação das soluções geradas. Em essência, o sistema incorpora as preferências do utilizador, refinando os resultados com base em suas escolhas ao longo do processo evolutivo.

Neste estudo, o sistema gera um conjunto de animações a partir do input do utilizador, onde os parâmetros são selecionados a partir de pequenas opções e tratados de forma independente. A *BO* é um método de otimização que utiliza uma função de aquisição para equilibrar média e variância, sendo especialmente útil em situações onde a função principal é complexa e difícil de avaliar.



Figura 15 – À esquerda a imagem utilizada como treino para o modelo apresentado, e à direita uma recriação da imagem original feita pelo modelo (Retirado de: [43])

Em [45] os autores abordam técnicas de personalização de estilos com o uso de uma abordagem de nome *SGDM* (*Style-Guided Diffusion Model*) que, através da rede neuronal *VGG19*, em conjunto com matrizes de *Gram*, conseguem extrair e aplicar estilos artísticos na geração de imagens. Apesar deste trabalho não referir em particular a transferência de estilos demonstra avanços na capacidade de personalizar imagens fornecidas sem haver a necessidade de treino para cada novo estilo.

O principal objetivo deste artigo foi superar limitações observadas em modelos de geração de imagens. Os autores propõem um método inovador que elimina a necessidade de treino específico para cada novo estilo, além de aprimorar o alinhamento entre os prompts textuais e o estilo desejado. Para isso, apresentam o *SGDM*, um modelo que extrai automaticamente o estilo desejado a partir de um conjunto de imagens de referência. Com base nessas imagens, o sistema gera uma máscara de ruído, que orienta a difusão do modelo para criar imagens alinhadas ao estilo fornecido.

Esta abordagem mostrou ter melhores resultados do que as técnicas usadas anteriormente, como *ControlNet* e *T2I-Adapter*, nos seguintes aspectos: Personalização de estilo uma vez que através de métricas como o *StySim*, foi possível chegar à conclusão de que o modelo apresenta um melhor desempenho na personalização adaptativa de estilos. Além disso, as métricas *FID* e *KID* demonstraram que o modelo conseguia gerar imagens de alta qualidade. Por fim, a última grande conclusão retirada foi o aumento da consistência na geração *txt2img*. Embora esse não fosse o foco do trabalho desenvolvido, houve um melhor alinhamento entre o estilo desejado e o *prompt* textual fornecido. Os autores referem ainda que o desempenho da solução aplicada sofreu alterações devido ao dataset utilizado, uma vez que este apresentava algumas inconsistências artísticas.

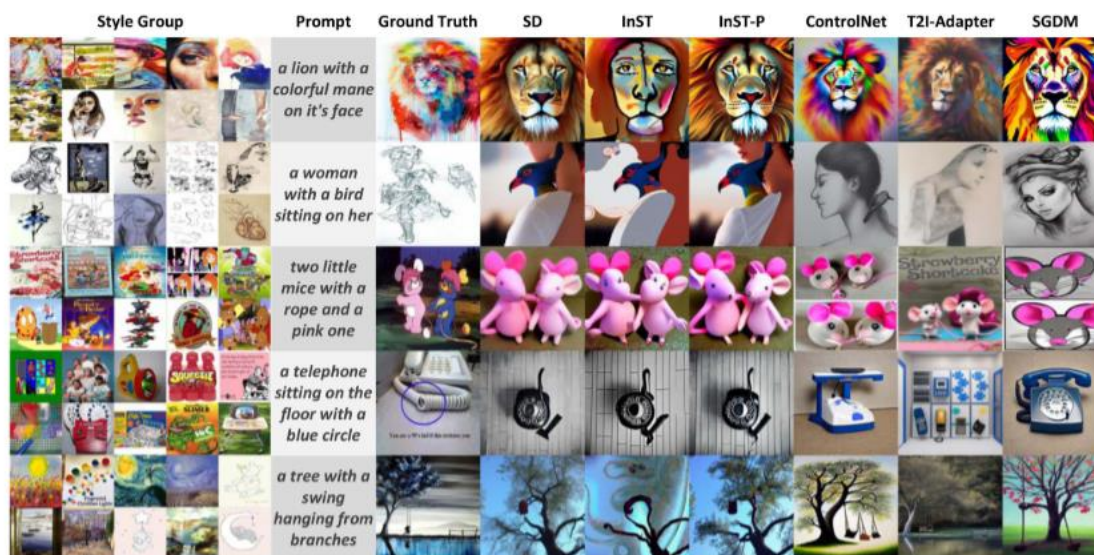


Figura 16 – Exemplos de imagens geradas pelos diversos elementos utilizados no modelo apresentado, juntamente das imagens utilizadas para o estilo e prompt textual (Retirado de: [44])

No artigo [46], os autores propõem um modelo baseado em Stable Diffusion de nome *DEPM (Diffusion-Enhanced PatchMatch)* que tem como objetivo realizar uma transferência de estilo sem a necessidade de ajustes ou pré-treino.

O *DEPM*, consegue capturar as características estilísticas e ao mesmo tempo preservar detalhes finos nas imagens permitindo assim uma transferência de estilos arbitrária, flexível e eficiente, destacando-se pela sua não necessidade de ajustes finos. Para alcançar este o objetivo os autores utilizaram métodos inovadores tais como o *PM (PatchMatch)* e o *WCT (Whitening and Colour Transform)*.

O *PatchMatch* é essencialmente uma técnica que ajuda na transferência de estilo através da aplicação de *patches* nas imagens, ou seja, a imagem é dividida em pequenos blocos (ou *patches*) para depois ser feita uma comparação de cada bloco obtido com o estilo alvo a que se pretende aplicar a transferência, assim, para cada *patch*, vai ser encontrado o pedaço mais próximo com base na textura ou padrões encontrados. Após este processo ser aplicado a todos os patches, começa a fase de substituição, onde cada um vai ser trocado pelo seu pedaço correspondente da imagem de estilo. Por fim os pedaços substituídos são combinados novamente, de forma a originar uma nova versão da imagem original ajustada à imagem do estilo escolhido. Esta técnica demonstrou ser muito eficaz a transportar detalhes de estilo desejado, mas, uma vez que altera a ordem dos pixels, pode trazer problemas futuros a alguns modelos de difusão.

Já o *WCT (Whitening and Coloring Transform)* tenta resolver o problema que surge da utilização do *PatchMatch*. Para o fazer, divide o processo em duas etapas: a primeira: o branqueamento (ou *Whitening*) é responsável por remover as características estilísticas da imagem original, esta etapa é importante pois permite criar uma base limpa para aplicação dos efeitos estilísticos da imagem de referência. A segunda etapa denominada de Colorização (ou *Coloring*) vai reintroduzir os detalhes do estilo da imagem de referência através da matriz de correlação que vai ser criada através da análise da imagem do estilo no espaço latente.

Este trabalho evidenciou resultados experimentais com desempenhos muito positivos incluindo uma superior transformação das cores, melhor preservação de conteúdo e uma alta qualidade na transferência de estilo. O sistema apresentado demonstra uma grande flexibilidade para diferentes aplicações criativas permitindo um controle sobre o equilíbrio entre o conteúdo e estilo. Um dos aspetos mais notáveis no trabalho é a sua capacidade de efetuar a transferência de estilo sem a necessidade de *prompts* textuais, mas mantendo a sua opção para a modificação

das imagens. Esta nova abordagem representa um avanço significativo na área, oferecendo uma solução mais eficiente e flexível.

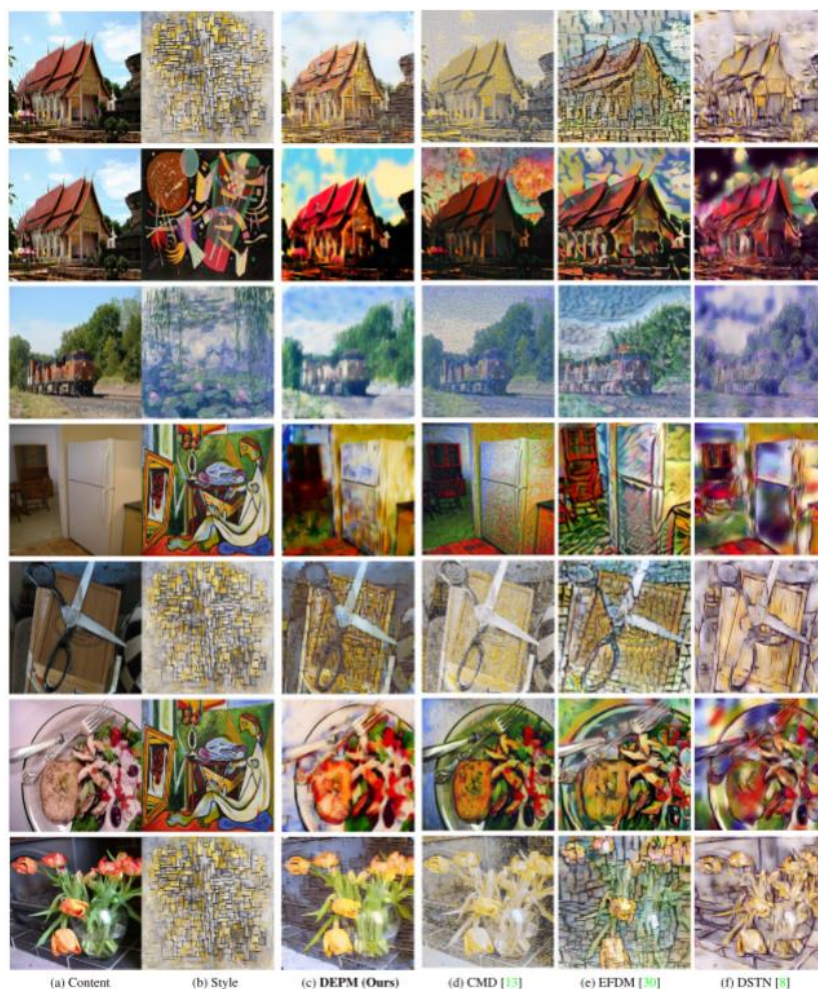


Figura 17 – Comparação da aplicação de estilos entre diversos modelos (Retirado de: [46])

Em [47] os autores fazem referência à dificuldade da escrita chinesa para pessoas que não estão familiarizadas com esta cultura e os seus símbolos ideográficos. Assim, basearam-se na tecnologia *Stable Diffusion* com adaptações *LoRa* (*Low-Rank Adaptation*) para que seja possível redesenhar caracteres chineses com base nos seus significados. Ou seja, os utilizadores poderão compreender e aprender a semântica de cada carácter de uma forma mais rápida e intuitiva.

O modelo de difusão utilizado nesta solução foi adaptado para a tarefa de redesenhar os caracteres chineses que tem como entradas sequências de escrita introduzidas pelos utilizadores. Para o bom funcionamento desta abordagem foram utilizados ficheiros *SVG* para permitir uma fragmentação precisa dos caracteres

introduzidos, permitindo assim ao modelo interpretar todas as linhas de forma individual.

Após a seleção das partes pretendidas, é inserido um *prompt* que orienta o modelo a conseguir transformar o traço do caractere chinês, em uma figura que se assemelha ao significado do símbolo com o uso de técnicas de difusão latente e adaptações *LoRa* para o ajuste dos parâmetros. Estando este trabalho numa fase inicial, o foco dos autores é principalmente garantir a qualidade do redesenho dos traços, preservando assim a estrutura do caractere e a sua legibilidade.

Este artigo demonstra que os modelos de difusão, mais propriamente *Stable Diffusion*, podem ter um grande sucesso a adaptar figuras permitindo que pessoas não nativas possam aprender chinês com mais facilidade. Os autores após testarem o modelo desenvolvido, comprovaram um aumento de cerca de 12% na precisão do reconhecimento em caracteres redesenhados.

Além disso a integração de *SVG* e possibilidade de utilização de *prompts* permitiu um controlo refinado dos resultados de reconstrução dos caracteres mantendo a sua estrutura e significado. Fazem referência a algumas principais dificuldades como a necessidade de *prompts* muito específicos e dificuldade em abranger todas as possibilidades de caracteres da língua. Apesar de tudo, continua sendo uma solução inovadora que junta tecnologias que dão uso a modelos de difusão com a área educacional e preservação de cultura para o transporte de estilos alternativos à escrita chinesa.

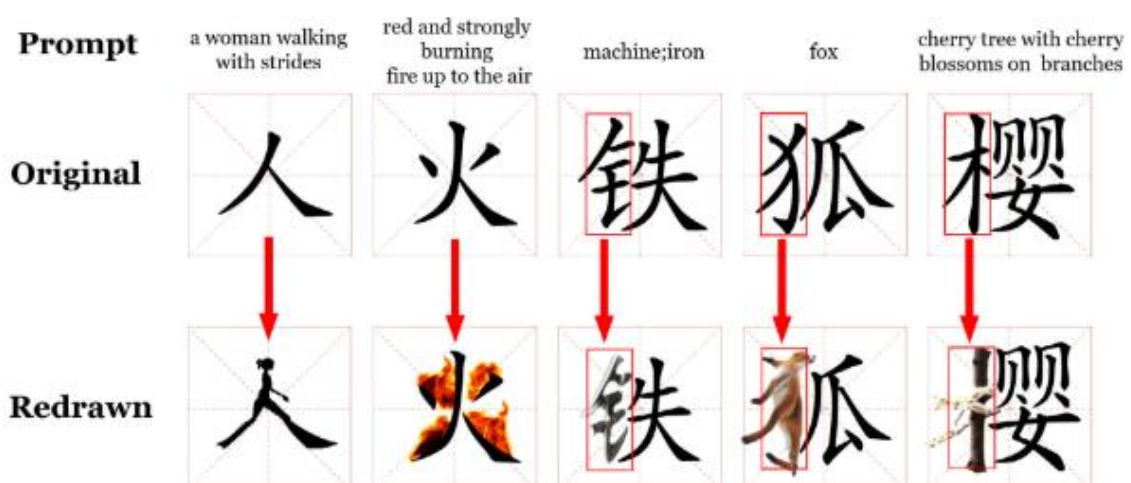


Figura 18 – Exemplo da aplicação do modelo apresentado (Retirado de: [47])

Em [48] os autores fizeram uma pesquisa para introduzir uma nova metodologia de transferência de estilo, através do uso do modelo base *Stable Diffusion* juntamente com incorporações *LoRa* e *ControlNet*. O principal objetivo deste estudo é o teste da geração de imagens de plantas de diferentes espécies e em diferentes fases do seu crescimento para facilitar a criação de *datasets* para outros modelos de inteligência artificial que tenham o objetivo de fazer análises de fetotipagem (processo de observação das características físicas de uma planta).

A proposta apresentada pelos autores tem a seguinte arquitetura: Uso de *LoRa* para ajudar a aprendizagem do domínio alvo, neste caso plantas, em que as suas estruturas têm rosetas. Dois módulos *ControlNet* que através da sua operação em paralelo conseguem identificar com maior precisão as bordas do elemento desejado com o uso de um algoritmo de nome *Canny*, com esta estrutura é garantida a preservação detalhada da geometria original da planta, para depois, começar o processo de transferência de estilo e gerar assim um conjunto de novas imagens realistas.

Deste trabalho é possível retirar alguns destaques, como a sua capacidade de trabalhar com quantidades reduzidas de dados, os autores dizem que cerca de 10-30 imagens são suficientes para um treino eficaz. Esta abordagem apresentou também resultados superiores aos de tecnologias anteriores, superando assim algumas dificuldades dos métodos convencionais que têm como base o uso de *GANs* e *CycleGANs*, tendo assim uma maior consistência e capacidade de adaptação em diversos cenários de utilização.

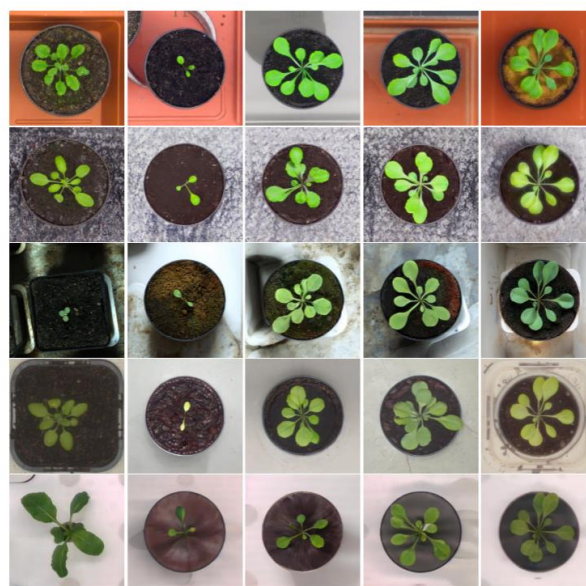


Figura 19 – Exemplo de imagens geradas pelo modelo apresentado, a primeira coluna representa a imagem utilizada como treino, enquanto o resto são diferentes outputs do modelo (Retirado de: [48])

Com o artigo [49] os autores apresentam o *MRStyle*, um framework que inova na área transferência de estilo focada na cor. Através de dois tipos diferentes de prompt, consegue alterar as cores de uma imagem fornecida. O primeiro *prompt* representa a imagem original que queremos editar, enquanto no segundo prompt vai uma descrição textual do estilo desejado. Para a parte da imagem a ser transformada, os autores utilizaram um modelo de nome *IRStyle*, que, através de tabelas de consulta 3D otimizadas, faz uma rede de mapeamento dupla e um *pipeline* com aprendizagem híbrida, conseguindo assim alcançar uma eficiência computacional muito elevada, mantendo a consistência e qualidade das imagens, mesmo em cenários onde existe muita variação de cor. Já nos prompts textuais, é utilizado o *TRStyle*, que em conjunto de um modelo de *Stable Diffusion*, alinha as características do texto apresentado à imagem fornecida, permitindo assim a customização das cores, minimizando artefactos desnecessários causados pelo uso destes modelos.

O principal desafio deste trabalho foi unificar as informações relativas aos estilos presentes nas imagens com a descrição textual fornecida. Para ultrapassar esta dificuldade, os autores focaram-se em treinar o modelo que efetua a transferência de estilo. De seguida, é feito um ajuste na descrição textual. Segundo os autores esta é a primeira abordagem que mistura a transferência de estilo via imagem e via texto em simultâneo, demonstrando-se assim um trabalho muito positivo por inovar dentro do campo da transferência de estilo com o uso de tecnologias que usam como base modelos de difusão.

Além disso o *MRStyle* destaca-se também pela sua elevada capacidade lidar com transferências estilísticas em cenários abertos, permitindo transformações eficientes. O uso de um *pipeline* híbrido combina supervisão pareada e não pareada, que, segundo os autores, essencial para melhorar a qualidade e eficiência do *framework* desenvolvido, garantindo que os resultados respeitem as características originais da imagem, assim como, corretamente, alterar a cor de forma que o resultado seja uma fusão entre ambos os inputs recebidos pelo utilizador. Quando comparado a métodos, o *MRStyle* apresenta vantagens significativas como: maior precisão na transferência de estilo, uma presença reduzida de artefactos visuais produzidos, e melhorias na eficiência do uso de memória e tempos de execução reduzidos mesmo sendo uma abordagem que está preparada para trabalhar com imagens de alta resolução e descrições textuais complexas.

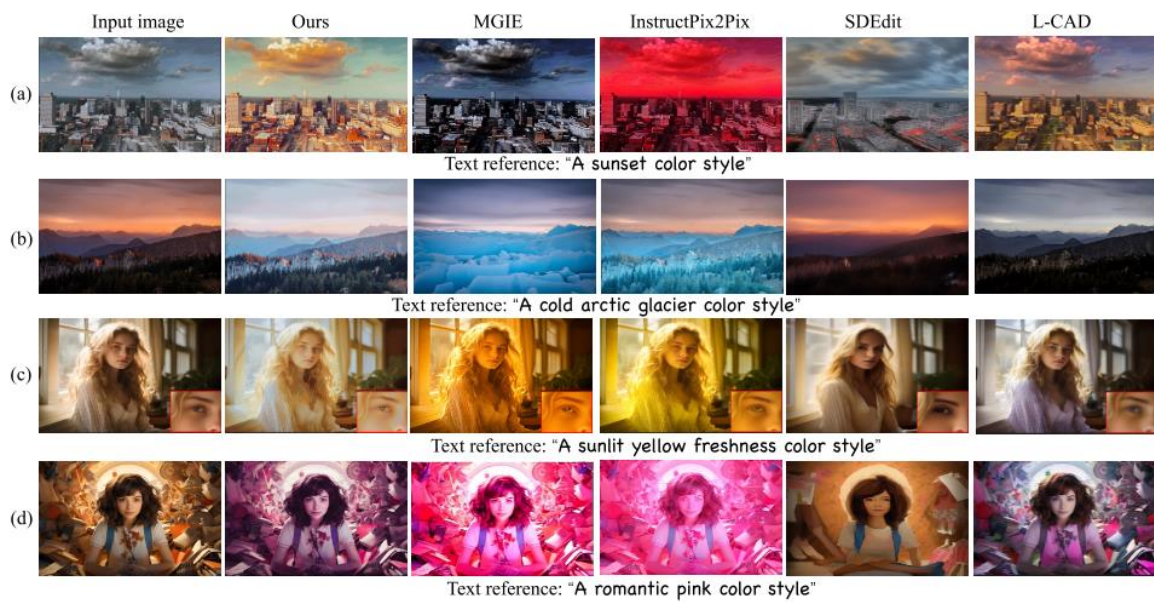


Figura 20 — Comparação entre outros modelos, acompanhada pela imagem de input e descrição textual do estilo (Retirado de [49])

Com [50] é apresentada uma nova abordagem para realizar transferência de estilo com uso de modelos de difusão pré-treinados, esta estratégia destacou-se de outras metodologias principalmente pela sua elevada capacidade de gerar imagens estilizadas, altamente realistas, com foco na estrutura original da imagem, este equilíbrio era algo muito difícil de alcançar em outras técnicas analisadas pelos autores.

A principal inovação que separa esta abordagem das outras, está no sistema de controlo criado para separar o processo de difusão em duas dimensões. O modelo apresentado divide a rede *U-Net* em três camadas distintas, a grosseira (*coarse*), a moderada (*moderate*) e a fina (*fine*) para conseguir capturar os detalhes estilísticos com maior precisão, através destas divisões, conseguem minimizar a probabilidade de erros na deteção do estilo desejado. O processo feito pelos modelos de difusão também foi alterado, e, segundo os autores, alterar a quantidade de etapas na remoção de ruído trouxe vantagens, nesta abordagem, foram usados 1000 passos divididos em dez etapas distintas que permitem ao modelo ganhar controlo granular sobre a evolução da imagem. Durante ambas estas fases, o sistema apresentado aplica diferentes controlos em cada fase. Na fase inicial, o *prompt* textual controla a parte grosseira da rede *U-Net* que preserva as características principais da imagem, já na fase intermédia o foco fica com os detalhes estruturais da imagem através da camada moderada, e, por fim, são feitos os retoques finos e aplicações minuciosas do estilo desejado na última e mais detalhada camada que irá tornar a transição entre a imagem original e o estilo mais suave e natural.

A principal limitação desta alternativa foi o tempo de geração de cada imagem quando comparada a outros métodos passados que utilizam *GANs*, uma vez que, a divisão por etapas e a criação de vários novos parâmetros atrasou o processo, apesar de resultar em melhorias positivas na imagem final. Os hiper parâmetros criados também se demonstraram uma fonte alguns problemas, uma vez que este modelo necessita de um ajuste cuidadoso dos mesmos para maximizar a qualidade dos resultados.

Esta solução representa uma superação significativa nas dificuldades apresentadas por outras soluções que tentam fazer semelhante, pois maior parte dependia de *prompts* globais ou divisões em etapas mais simples que as apresentadas pelos autores. Este novo sistema que divide o *prompt*, permite não só apenas ao modelo aprender melhor o estilo desejado, mas também tornar o seu armazenamento mais otimizado.

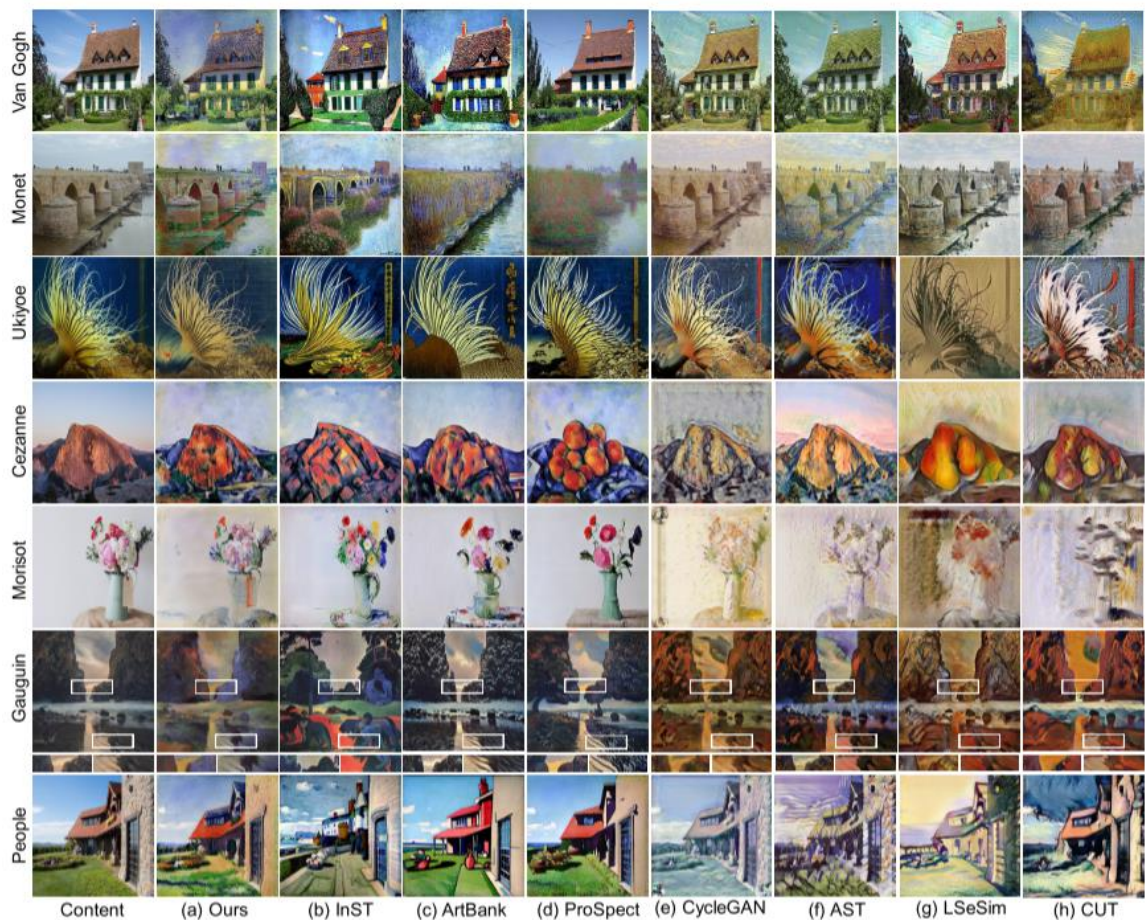


Figura 21 — Comparação dos resultados obtidos com outros modelos, as colunas B-D representam os modelos de *Stable Diffusion* estudados pelos autores, e as colunas E-H representam outras técnicas como o uso de *GANs* (Retirado de [50])

Os autores de [51] apresentam o *DiffStyler*, uma ferramenta criada com o objetivo de realizar transferências de estilos de maneira eficaz. A abordagem apresentada baseia-se na utilização de modelos de difusão, mais especificamente *Stable Diffusion* que, com o uso de *LoRa*, permite de uma maneira mais eficaz capturar o estilo desejado.

Para alcançar este objetivo, os autores exploraram algumas técnicas presentes nos modelos *txt2img* como: inversão *DDIM*, mecanismo responsável por reconstruir as imagens fornecidas do espaço dos pixels para o espaço latente permitindo modificações sem perder a semântica original, a injeção e características e mecanismos de atenção *LoRa*, que irá permitir a combinação dos estilos desejados, ou seja, manter os elementos estruturais da imagem original enquanto é aplicada detalhes do estilo desejado.

O treino *LoRa*, é um método muito eficiente que permitiu treinar um modelo pré-treinado mediante matrizes de baixa complexidade, isto, essencialmente, permitiu capturar os atributos do estilo desejado sem haver a necessidade de alterações grandes dos pesos originais do modelo utilizado, ao invés disso, são feitas pequenas alterações de forma que seja possível aprender o estilo apresentado. Por fim a remoção ruído por meio de máscaras obtidas pelo modelo *FastSAM*, ajudaram a preservar certas áreas da imagem original, de modo que haja um equilíbrio entre o novo estilo inserido na imagem e os seus elementos originais.

Após os testes realizados, esta abordagem demonstrou uma consistência notável nas imagens obtidas, ajudando a fundamentar a viabilidade do uso de modelos de *Stable Diffusion* juntamente com *LoRa*. Concluindo, o *DiffStyler* representa uma significativa contribuição na área da transferência de estilos com modelos de difusão, permitindo a aprendizagem de estilos via uma única referência e integrando o estilo eficazmente numa imagem fornecida.

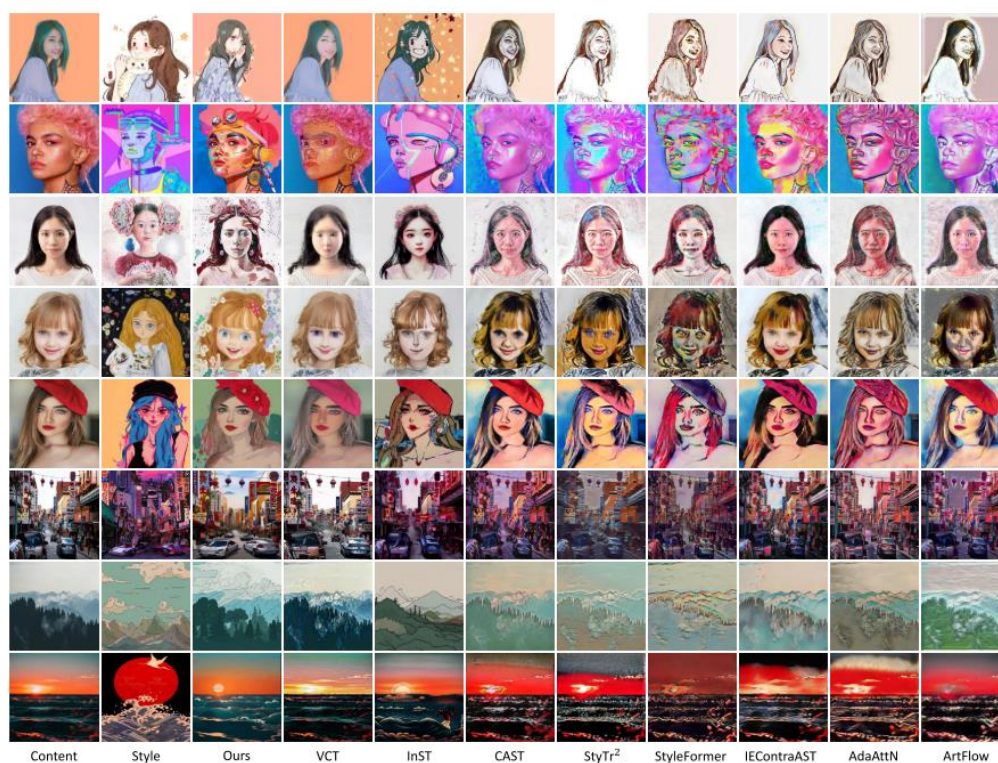


Figura 22 — Exemplos de imagens geradas por outros modelos e pelo modelo apresentado pelos autores, com base num *input* (*Content*) e num estilo (*Style*) submetidos (Retirado de: [51])

Para sintetizar a análise realizada, é possível visualizar na Tabela 3 as principais características analisadas em cada artigo

Tabela 3 - Descrição dos Atributos

Atributo	Descrição
Referência	Número de referência do artigo
Tema	Nome ou tema principal do artigo
Ano	Ano em que estudo/artigo foi realizado
Técnicas Utilizadas	Métodos, frameworks, modelos e algoritmos aplicados pelos autores para realizar os objetivos do artigo.
Resultados	Contribuições significativas do artigo, como, os principais resultados obtidos e as suas dificuldades.

Na tabela 3 é possível visualizar os principais tópicos da cada um dos artigos analisados consoante os parâmetros da tabela 3.

Tabela 4 - Resumo das características analisadas em cada artigo

Ref	Tema	Ano	Técnicas Utilizadas	Resultados
[40]	Geração de Avatares Animados com GAN	2023	GAN (DCGAN), StyleRig, SEAN	Alta flexibilidade
[41]	Text2QR: Geração de QR Codes Estéticos	2023	Stable Diffusion, QR Aesthetic Blueprint (QAB), SELR	Equilíbrio entre arte e legibilidade dos códigos QR
[42]	Modelo de Difusão com Alta Eficiência de Estilo	2024	Stable Diffusion, refinamento de U-Net, ajuste de ruído	Alta precisão na transferência de estilos em pinturas famosas
[43]	Combinação de IES e BO para Estilo Personalizado	2023	IES, BO, biomorphs	Alta personalização com geração interativa de designs
[44]	SGDM: Personalização Sem Treino Prévio	2024	VGG19, matrizes de Gram	Alta personalização de estilo em text-to-image
[45]	DEPM: Transferência de Estilo com PatchMatch	2023	PatchMatch, WCT, Stable Diffusion	Preservação de detalhes finos e flexibilidade
[46]	AIGC para Redesenho de Caracteres Chineses	2024	Stable Diffusion, LoRA, SVG	Aumento de 12% no reconhecimento dos caracteres redesenhados
[47]	Geração de Plantas para Fenotipagem	2024	Stable Diffusion, LoRA, ControlNet,	Treino eficaz, sem a necessidade de um grande dataset

			algoritmo <i>Canny</i>	
[48]	<i>MRStyle</i> : Transferência de Estilo via Texto e Imagem	2024	<i>IRStyle</i> , <i>TRStyle</i> , <i>Stable Diffusion</i>	Ganho na precisão da transferência de estilo e redução de artefactos
[49]	Nova Abordagem de Estilo com Difusão Pré- Treinada	2023	Divisão <i>U- Net</i> em <i>coarse</i> , <i>moderate</i> , <i>fine</i> ; 1000 passos de difusão	Imagens realistas com preservação estrutural
[50]	<i>DiffStyler</i> : Transferência Eficaz de Estilo	2024	<i>Stable Diffusion</i> , <i>LoRA</i> , <i>DDIM</i> , <i>FastSAM</i>	Aprendizagem eficiente de estilos com preservação estrutural

3.3 Discussão dos Resultados

Os avanços analisados nos modelos de difusão, mais concretamente, o *Stable Diffusion*, têm transformado o cenário dentro da área da inteligência artificial generativa. Estes modelos têm se demonstrado extremamente versáteis e capazes de gerar imagens que correspondem às mais diversas realidades. Através deste subcapítulo pretendemos explorar os principais atributos que contribuíram para estes resultados a partir das conclusões a retirar de cada um dos artigos.

O *Stable Diffusion* tem ganho bastante destaque nos últimos anos no que toca a transferência de estilos, demonstrando melhores resultados que estratégias até então como as *GANs* (redes generativas adversárias) tornando-se assim a escolha acertada para o transporte do estilo do bordado de Castelo Branco. Abordagens analisadas como o caso do *DiffStyler* [51], mostram como a utilização de modelos de difusão com outras técnicas auxiliares como *LoRa's*, pode trazer resultados muito positivos, permitindo uma aprendizagem rápida e eficiente de um estilo ao mesmo tempo que mantendo a estrutura e semântica da imagem original fornecida, oferecendo um alto nível de consistência e realismo.

Os trabalhos realizados em [42] e [47] demonstram também uma muito positiva prova da capacidade de juntar à transferência de estilos uma utilidade fora do universo artístico. Em [48] também é possível vermos um cenário onde o foco

principal não está na criação de imagens para fins artísticos, mas sim utilizando a ideia da transferência de estilos para criar imagens realistas o suficiente para serem utilizadas como parte de um *dataset*, onde, caso não utilizassem este tipo de técnicas, iram ter dificuldades para atingir um número satisfatório de imagens.

Apesar disso, pelos trabalhos analisados, é possível analisar que o forte deste tipo de modelos está nas adaptações artísticas, como é possível ver pelos artigos [42], [43], [44], [45], [48], [49] e [50] sendo nestes trabalhos explorado a vertente artística destes modelos, desde a transferência arbitrária de estilos, até à transferência de estilos das mais conhecidos como estilo de pinturas de artistas famosos para a outras imagens fornecidas como *input*. Uma análise a estes artigos, mostrou-nos que todas as abordagens analisadas, demonstraram-se ser superiores aos métodos utilizados até então em diversas métricas, como, tempo de execução, flexibilidade, redução quantidade de imagens para treino e capacidade de manter os elementos principais da imagem fornecida.

3.4 Conclusões

O estudo da análise dos artigos permitiu-nos verificar e perceber os avanços realizados na área da transferência de estilos e personalização de imagens via inteligência artificial generativa, com o uso de modelos de difusão. Estes avanços tendem sempre em focar na criação de soluções mais eficientes e flexíveis, enquanto mantém um nível bastante positivo de realismo e fidelidade às imagens fornecidas. Um dos principais pontos em comum com os estudos analisados é o uso de pequenas adaptações e inovações nos modelos de difusão, tais como *Stable Diffusion*, o uso de *ControlNet* e *LoRa* que quando utilizados, elevam os resultados esperados oferecendo uma maior precisão na transferência estilo. Além disso, a utilização destes avanços torna os modelos muito mais flexíveis e preparados para diferentes estilos, tornando assim estas soluções extremamente adaptáveis a outros meios assim como preparadas para receber não só uma imagem para efetuar a transferência de estilo, mas também *prompts* textuais que permitem ao utilizador refinar os resultados ou alterar as características mais pequenas.

Apesar de tudo, foi ainda possível verificar algumas das mais frequentes dificuldades destas técnicas, a necessidade de ajustes finos mediante *prompts* textuais foi uma limitação em vários trabalhos, o tempo de geração de imagens e os desafios na otimização de alguns parâmetros dificultaram a geração de resultados consistentes.

4. Transferência de Estilo com *Stable Diffusion*

Neste capítulo, pretendemos entrar mais a fundo na transferência de estilo, através da introdução de alguns conceitos fundamentais para o entendimento desta técnica. Vai também ser feita uma referência a alguns problemas e questões a serem levantadas com a aplicação de técnicas como esta, assim como uma breve evolução da transferência de estilos.

4.1 O que é Transferência de Estilo

A transferência de estilo é uma técnica desenvolvida há perto de duas décadas, baseia-se na manipulação de um conjunto de dados, normalmente imagens ou vídeos. O objetivo deste processo é criar amostras de dados utilizando como base outros dados, combinando essencialmente as características estilísticas de uma imagem com o conteúdo de outra. Uma das suas principais dificuldades, é ter ao seu dispor uma quantidade suficientemente ampla de dados para conseguir capturar todas as nuances e detalhes de um dado estilo, já que a qualidade dos resultados obtidos vai depender diretamente com a qualidade e flexibilidade do conjunto de treino. Outra dificuldade muito comum, é a dificuldade na distinção entre o estilo e o conteúdo da imagem, o que pode levar a uma incorreta seleção das características do estilo, criando assim inconsistências e resultados menos positivos [52].

4.2 Implicações Éticas e Legais

A transferência de estilo, quando junta de ferramentas poderosas de inteligência artificial, pode levantar sérias questões éticas e legais. Como, por exemplo, ao treinarmos um modelo para aprender o estilo de um dado artista, podemos obter resultados tão convincentes que é natural surgir a dúvida de se a obra gerada foi feita pelo autor em questão, ou não, criando assim problemas sérios, e situações onde ocorre uma violação da lei da propriedade. Além disso, há o risco de plágio, pois obras geradas com transferência de estilos, embora muitas vezes tragam consigo fusões novas de estilos e criação de arte nunca vista, acabam, por natureza, carregar características identificáveis das imagens originais.

Outro grande problema muito discutido na atualidade está relacionado com o uso de tecnologias como esta para a manipulação de imagens. Vivendo nós na era das notícias falsas onde existe sempre a necessidade de verificar os factos, estas tecnologias abrem espaço para a criação de imagens com os seus conteúdos visuais alterados ou então até mesmo a imitação de documentos oficiais com mensagens alteradas.

Em suma, apesar de ser uma tecnologia revolucionária dentro dos campos criativos e artísticos, é importante haver uma responsabilização pelo conteúdo criado e atenção cuidadosa aos seus utilizadores e intenções.

4.3 Técnicas Tradicionais de Transferência de Estilo

As primeiras técnicas de transferência de estilo baseadas em renderização conhecidas como *NPR* (*Non-Photorealistic Rendering*) foram criadas em 2001 por *Hertzmann*, sendo desenvolvidas visando a geração de imagens estilizadas que imitavam os estilos mais tradicionais de arte como desenhos e pinturas, ao invés de ir atrás do fotorrealismo. Estas abordagens faziam comparações pixel a pixel para a detecção de contornos e aplicações de filtros simples que tinham como principal objetivo, remover algumas imperfeições criadas. Ao aplicar estas técnicas, o objetivo é alterar a aparência de uma dada figura e computacionalmente aplicar efeitos estilísticos que remetessem a formas de arte tradicionais.

Estes métodos clássicos seguem frequentemente um modelo de “um para um”, no qual todo o processo de transferência ocorre tendo apenas como base duas imagens, uma onde o estilo vai ser extraído e outra onde vai ser aplicado. Estas abordagens embora simples, tinham grandes dificuldades quando o estilo desejado era rico em detalhes uma vez que a simples análise dos contornos não iria permitir capturar todos os detalhes do estilo [53].

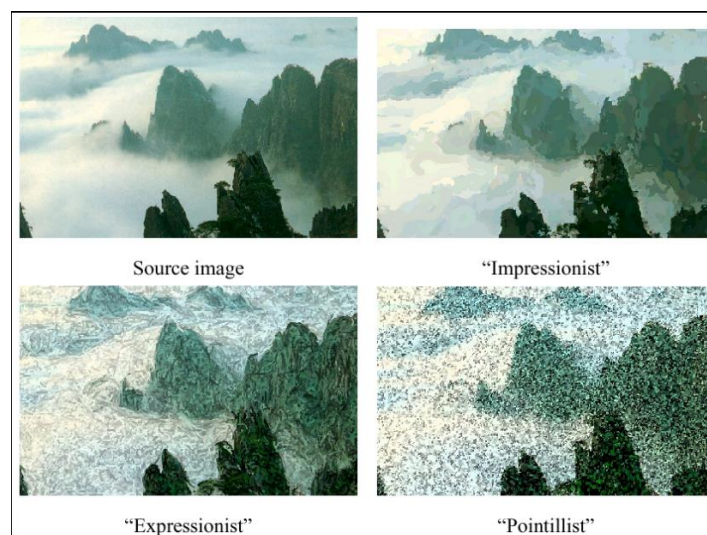


Figura 23 — Exemplo do uso do *NPR* para três estilos distintos de pinturas (Retirado de: [54])

4.4. Aplicação de Transferência de Estilo com *Stable Diffusion*

Modelos que utilizam *Stable Diffusion* tem capacitado cada vez mais a aplicação de transferência de estilo no dia a dia das pessoas por meio de aplicações que incorporam este tipo de modelos. Permitindo assim, o transporte de estilos para novas realidades, como, por exemplo, aplicar o estilo de famoso pintor numa foto nossa [46].

A principal vantagem ao utilizar este tipo de modelos difusão é a sua capacidade em gerar resultados muito positivos sem a necessidade de grandes conjuntos de dados para treino, a sua eficiência, flexibilidade e capacidade de fazer ajustes finos e alta customização do conteúdo de cada foto.

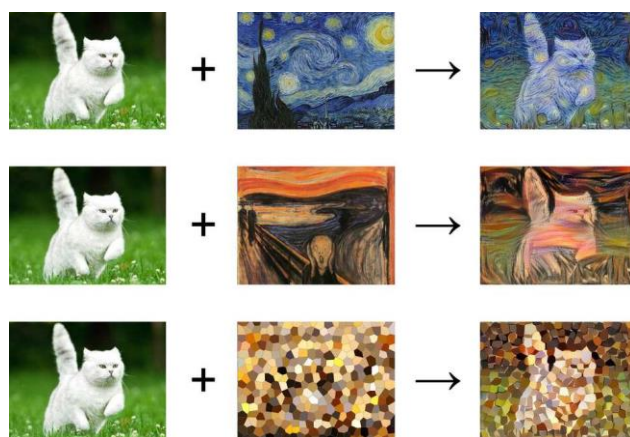


Figura 24 — Exemplos de transferências de estilo com *Stable Diffusion* (Retirado de: [55])

5. Caso de Estudo: Bordado de Castelo Branco

Neste capítulo, temos como objetivo explorar com mais detalhe o caso de estudo a ser analisado, assim como documentar a criação do *dataset*, passando pelos pontos mais pertinentes como a recolha, tratamento e dificuldades deste processo.

5.1 História

O bordado de Castelo Branco foi e continua a ser um dos símbolos mais característicos da região de Castelo Branco, é feito a partir de colchas de linho bordadas com fio de seda, com desenhos influenciados por estilos e técnicas tradicionais, tornando-se assim conhecido a partir do meio do século XVI. Este tipo de arte ficou fixa na região de Castelo Branco devido à sua cultura do linho que vinha do fato de amoreira se desenvolver com grande facilidade nesta região [56].

A maioria dos elementos das representações bordadas têm significados, como, por exemplo, os cravos e rosas representarem o homem e a mulher, corações o amor, gavinhas para a amizade entre outros. Uma das variadas características do tema do bordado de Castelo Branco, é que ele se propaga pela cidade quer nas ruas, edifícios e calçadas assumindo um dos maiores símbolos da cidade [57].

O bordado de Castelo Branco torna-se diferente de todos os outros, pois a simbologia embutida nele é própria e original, atualmente ainda se produz algumas peças, apesar do processo de criação das mesmas ser, por vezes, bastante difícil.

Algumas dessas peças estão expostas localmente em museus, como por exemplo, o Museu Francisco Tavares Proença Júnior ou então também visíveis em feiras ou eventos como a Feira Internacional de Turismo de Madrid, uma vez que o município de Castelo Branco marcou a sua presença fundamentalmente por causa do bordado [57].



Figura 25 – Exemplo do Bordado de Castelo Branco

5.2 Criação do *Dataset*

Com o objetivo de ensinar o estilo do bordado de Castelo Branco ao nosso modelo, era de grande importância conseguir criar um *dataset* que represente com a máxima clareza e diversidade possível os elementos do bordado. O *dataset* utilizado foi feito com base nos conteúdos digitais disponíveis na *internet*, que retratavam parcialmente ou completamente os elementos estilísticos do bordado.

5.2.1 Recolha das Imagens

No desenvolvimento desta fase do projeto, encontrámos algumas dificuldades técnicas que fizeram com que os resultados obtidos não fossem os mais satisfatórios.

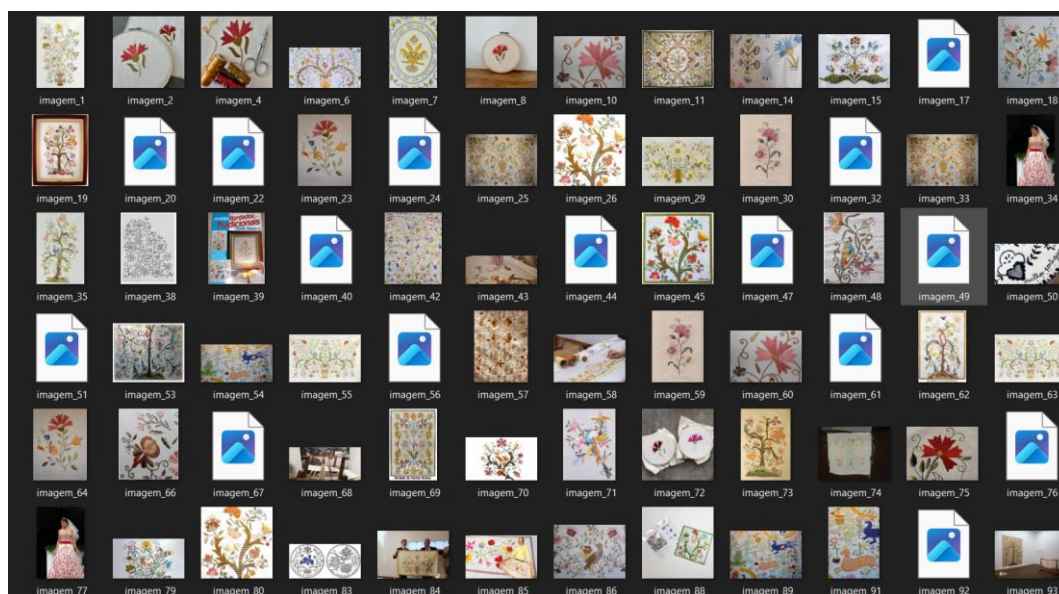


Figura 26 – Seleção das imagens para o *dataset*

Foram recolhidas 200 (duzentas) fotografias, todas com origem na pesquisa via internet com o termo ‘Bordado de Castelo Branco’. Para a obtenção destas imagens, optámos pela utilização da *API do Google Custom Search JSON API* [58] que tem como principal objetivo, criar, com base numa *query*, um ficheiro *JSON* com os *URLs* de todas as imagens obtidas através da pesquisa, conseqüentemente, através da biblioteca *requests* disponível nativamente com a linguagem *Python*, conseguimos transferir cada uma das imagens retornadas pela pesquisa feita.

5.2.2 Desafios

Após termos as imagens obtidas, enfrentámos alguns desafios, principalmente no que toca na estrutura das fotos, uma vez que nem todas têm a mesma luminosidade, ângulo, e até mesmo a estrutura em volta do bordado serem diferentes, o facto de algumas das fotos terem sido duplicadas ou então não estarem no formato correto tornou o seu acesso impossível. Na Figura 27 é possível visualizar um de muitos exemplos de fotos não tão satisfatórias obtidas pelo processo de recolha anteriormente descrito.



Figura 27 – Exemplo de uma foto obtida

5.2.3 Preparação dos Dados

De forma a minimizar os desafios que surgiram pela falta da qualidade das imagens foi feita uma escolha, dentro de todos os resultados obtidos, de forma a excluir as situações mais extremas, onde seria muito difícil de extrair apenas os elementos presentes no bordado. Para todas as outras, com o objetivo de aumentar significativamente o tamanho do nosso dataset, começamos por recortar as

imagens de forma a ficar apenas com as partes mais relevantes. Após os recortes, optámos por efetuar diversas rotações e espelhamentos de forma a multiplicar as imagens obtidas e maximizar assim a informação que o modelo poderia aprender com cada uma das imagens obtidas.

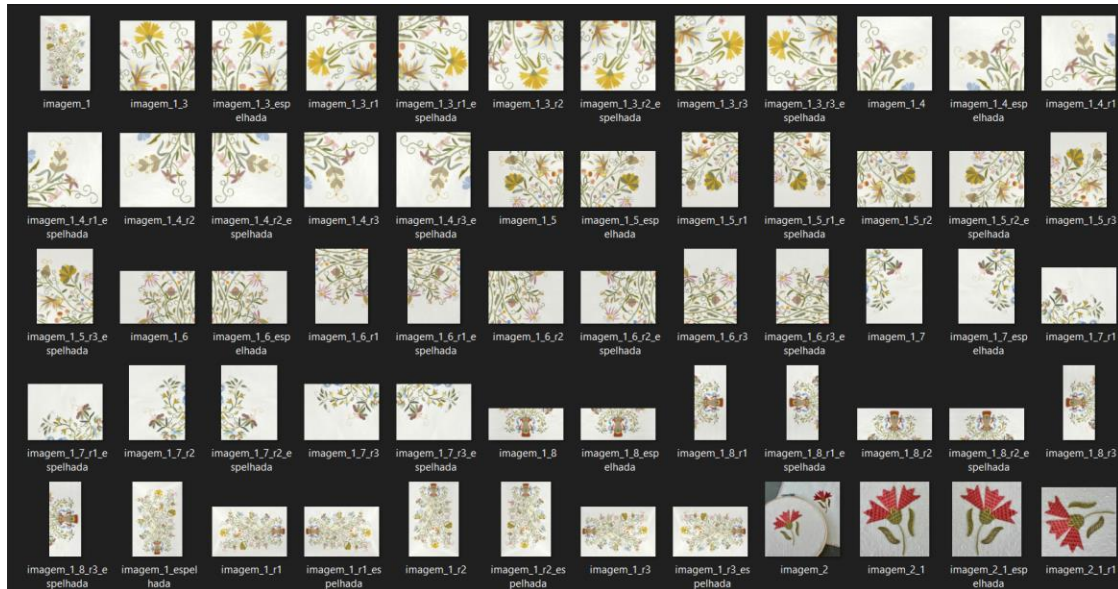


Figura 28 – Seleção das imagens para o dataset após recortes e rotações

Com este processo conseguimos transformar as 200 (duzentas) imagens obtidas para 1785 (mil setecentos e oitenta e cinco), classificadas como Bordado de Castelo Branco.

Por fim, com o objetivo de criar um *standard* de tamanhos, convertemos todas as imagens para um tamanho igual, aceite de forma generalizada, com diversos modelos de *Stable Diffusion*, tornando assim todas as imagens no tamanho 512x512.

6. Ferramentas Utilizadas

Com este capítulo pretendemos abordar as mais predominantes ferramentas utilizadas para o desenvolvimento deste projeto.

6.1 Python

Python [59] é uma linguagem de programação de alto nível que devido à sua fácil sintaxe permite aos desenvolvedores escreverem o seu código de forma clara e concisa. É, também, uma das ferramentas mais populares dentro do campo da inteligência artificial devido ao seu amplo suporte e número extenso de módulos e bibliotecas que dão suporte a esta área.

6.2 Google Colab

O *Google Colab* [60] é uma plataforma baseada em nuvem que permite a execução de código remotamente, diretamente do navegador. Esta ferramenta foi fundamental para o desenvolvimento do nosso projeto, especialmente na secção de treino do modelo utilizado, uma vez que, permitiu realizá-lo rapidamente devido à alta capacidade de computação das máquinas disponibilizadas, assim como ter um ambiente controlado onde o controlo de versões de cada um dos módulos necessários não é um problema.

6.3 Hugging Face

Hugging Face [61] é uma plataforma com foco na comunidade de Machine Learning e ciência de dados, muito semelhante com o *GitHub*, onde o foco principal está na disponibilização de ferramentas, modelos e *datasets open source*. Neste projeto o seu uso está relacionado com a interligação com as plataformas *Google* e o seu fácil uso junto do *Google Colab*.

6.3 Birme

Birme [62] é uma aplicação web que permite o redimensionamento e recorte em massa de imagens. É uma plataforma gratuita, que facilita o processo de ajuste do tamanho de um grande número de imagens, tornando-se assim importante para este projeto, uma vez que a normalização do tamanho das imagens tornou a aprendizagem mais fácil.

6.4 Automatic1111

O *Automatic1111* [5] é uma plataforma open source disponível no *GitHub*, que oferece uma interface gráfica simples e intuitiva para o uso de modelos *Stable Diffusion*. Esta permite o uso de vários tipos de *prompts*, assim como a alteração minuciosa de diversos parâmetros. Além disso, suporta uma ampla diversidade de extensões abrindo o caminho para o uso de ferramentas como *ControlNET* e *LoRA's*.

6.5 Dreambooth

Dreambooth [63] é uma técnica para o treino para os modelos de difusão que permite a inserção de um novo conceito ao modelo através de um treino com foco na geração *txt2img* onde, através das imagens fornecidas, com a devida descrição dos seus conteúdos, torna possível ensinar qualquer novo estilo ou termo desejado. No desenvolvimento deste projeto, juntamente com o *Google Colab* foi permitido fazer o treino do estilo do bordado de Castelo Branco.

6.6 GitHub

O *GitHub* [64] é um sistema de alojamento de código e controlo de versões, que, através do *Git*, permite guardar e armazenar alterações feitas no código e, posteriormente, o seu alojamento *online*. A sua capacidade em criar diferentes ramificações e registo de bugs torna esta uma plataforma versátil para os desenvolvedores que procuram suporte às suas criações.

6.7 Dart

O *Dart* [65] é uma linguagem de programação multiplataforma desenvolvida pela *Google*, com o propósito de criar aplicações eficientes em diversos ambientes, como *web*, *móvel*, *desktop* e servidor. Através do seu sistema de tipagem estática, o *Dart* permite que os desenvolvedores escrevam código seguro, fácil de manter e escalável. Esta foi a linguagem escolhida para o desenvolvimento do protótipo, permitindo a construção de uma solução rápida eficiente e multiplataforma.

6.8 Flutter

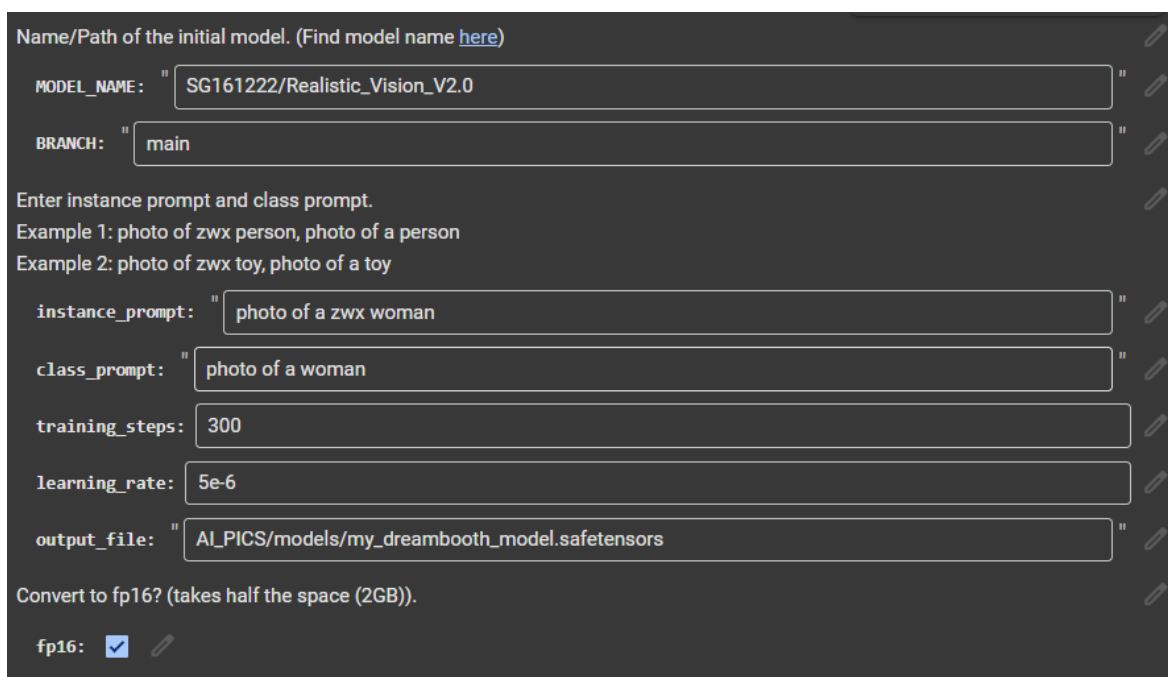
O *Flutter* [66] é uma *framework* de desenvolvimento criado pela *Google*, utilizado para construir aplicações nativas para dispositivos móveis, *web* e *desktop* a partir de um único código. Através da sua arquitetura baseada em widgets e da linguagem de programação *Dart*. Com este *framework* conseguimos construir uma interface simples e intuitiva para o nosso protótipo.

7. Resultados Obtidos pelo primeiro modelo

Neste capítulo pretendemos documentar todo o processo do trabalho experimental realizado, desde o treino do primeiro modelo, até às escolhas de parâmetros nas gerações de imagens, assim como uma pequena demonstração de resultados, através das duas principais formas de utilização destes modelos, a geração de imagens através de *txt2img* e com a transferência de estilo a ser aplicada via *img2img*.

7.1 Treino Realizado

Para a fase de treino optámos por utilizar um *notebook*, disponibilizado pela *Dreambooth*, na plataforma *Google Colab*. Apesar de ser um *notebook* já antigo, e não suportado, demonstrou-se ser das únicas opções que conseguia de maneira eficaz treinar um modelo de *Stable Diffusion* pré treinado, um novo elemento estilístico do qual passamos a demonstrar a figura abaixo.



The image shows a screenshot of a Jupyter Notebook interface for training a Dreambooth model. The parameters are as follows:

- MODEL_NAME:** "SG161222/Realistic_Vision_V2.0"
- BRANCH:** "main"
- instance_prompt:** "photo of a zwx woman"
- class_prompt:** "photo of a woman"
- training_steps:** 300
- learning_rate:** 5e-6
- output_file:** "AI_PICS/models/my_dreambooth_model.safetensors"
- fp16:**

Figura 29 – Parâmetros do *notebook* de treino da *Dreambooth*

Estes parâmetros permitem configurar diversas definições do processo de treino do modelo tais como: Escolher o modelo base de *Stable Diffusion* que queremos utilizar através de *MODEL_NAME*, este campo irá receber o nome do modelo com base na plataforma *Hugging Face*, permitindo assim escolher um qualquer modelo disponível dentro da dada plataforma. Em *BRANCH* é nos permitida escolher uma versão do modelo dentro da plataforma, em quase todos os casos, o uso do *branch main* é o recomendado, por se tratar da última versão estável.

Com *instance_prompt* é onde nos permite explicar ao modelo o conceito que vai aprender através de uma pequena descrição, é aqui que palavras-chave como “Bordado de Castelo Branco” vão ser utilizadas, permitindo assim ao modelo ganhar a capacidade de identificar e gerar imagens do bordado. Já em *class_prompt*, permite associar a *instance_prompt* uma classe mais geral, que ajuda na preservação das características gerais enquanto aprende um conceito mais específico, no nosso caso, utilizámos como *instance_prompt* = “Bordado de Castelo Branco” e *class_prompt* = “Bordado”. Na figura a baixo é possível verificar estes parâmetros.

```

MODEL_NAME: " SG161222/Realistic_Vision_V2.0
BRANCH: " main

Enter instance prompt and class prompt.
Example 1: photo of zwx person, photo of a person
Example 2: photo of zwx toy, photo of a toy

instance_prompt: " Foto do Bordado de Castelo Branco
class_prompt: " Foto do Bordado
training_steps: 2500
learning_rate: 5e-6
output_file: " AI_PICS/models/modelo_bordadoCasteloBranco.safetensors

Convert to fp16? (takes half the space (2GB)).
fp16: 

```

Figura 30 — Parâmetros de treino

Após estabelecer conexão com a máquina virtual encarregue de realizar o nosso treino, é nos permitido escolher os ficheiros das imagens com as quais iremos fazer o treino.

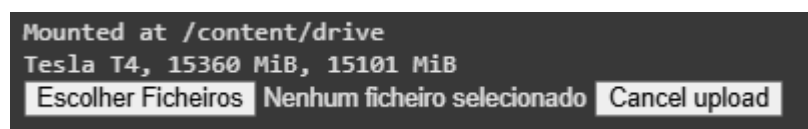


Figura 31 — Escolha das imagens de treino

Após escolher as imagens de treino começa o processo de *upload* do *dataset*.

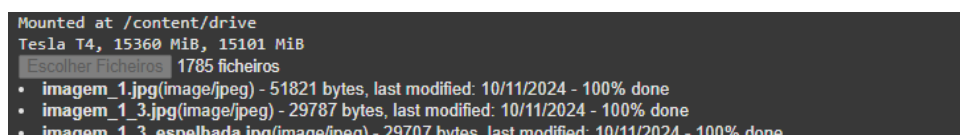


Figura 32 — Upload do *dataset* para treino

Com este notebook, serão utilizadas técnicas do *DreamBooth* para realizar o *fine-tuning* de um modelo pré-treinado através da associação de um *token* textual único. Este, irá representar o novo tema a ser treinado, neste caso, o bordado de

Castelo Branco. Esta técnica permite ao modelo aprender a gerar imagens coerentes com base no novo conceito, mas sem comprometer a sua capacidade de geração de imagens prévia. Esta abordagem é especialmente importante, pois evita a catástrofe do esquecimento, um problema recorrente em redes neuronais profundas, onde a modificação dos pesos pode levar à perda da capacidade de gerar imagens de outros tópicos. Desta forma, ao contrário de outros métodos de *fine-tuning*, o *overfitting* é evitado, permitindo assim ao modelo manter a sua flexibilidade criativa e generalização [67].

Após o processo de upload e treino for concluído o ficheiro que origina do *notebook* vai ser enviado para o *Google Drive*, em forma de *safetensors*, possibilitando assim o seu uso em aplicações como o *A1111*, ferramenta escolhida como demonstra a figura abaixo.

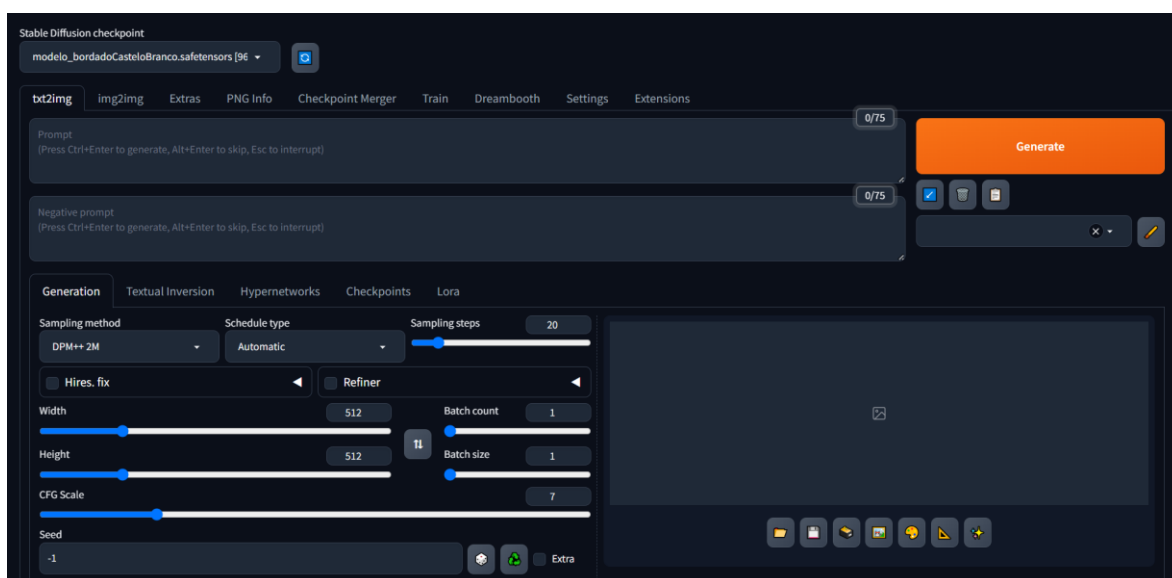


Figura 33 — Interface gráfica do A1111

Vai ser a partir desta interface gráfica que a geração das imagens vai ser feita, dentro desta interface temos a possibilidade de modificar alguns parâmetros do quais passamos a explorar:

Parâmetros Básicos – Estes são os parâmetros mais simples, que não alteram de maneira significativa o resultado e apenas têm como função alterar as propriedades básicas das imagens geradas. São estes: *Width* (Largura), *Height* (Altura), *Batch Count* (número de *batches* de imagens que vão ser gerados) e *Batch Size* (número de imagens geradas por cada *batch*).

Parâmetros Avançados — Estes são os parâmetros que vão ser utilizados pelo modelo, influenciando assim diretamente o resultado. Sendo eles: O modelo em si (é possível selecionar diferentes modelos no canto superior esquerdo da figura

anterior), *Prompts* Negativos e Positivos (permitem modificação dos resultados a partir de *prompts* textuais, sendo os positivos as características a gerar, e os negativos características a evitar). *Sampling Method* (este parâmetro vai definir qual o algoritmo que vai ser utilizado para o processo de remoção de ruído sendo o *DPM ++ 2 Karras* o recomendado [68]). Já *Schedule Type* e *Sample Steps* controlam o processo oposto, permitindo ao utilizador alterar a forma com que a adição de ruído é feita [69]. *Hires* e *Refiner* são extensões adicionais do *A1111* que permitem o *upscaling* das imagens e possibilitar a criação das mesmas com múltiplos modelos em simultâneo, respetivamente. *CFG_Scale* representa o peso dos *prompts* textuais, onde valores altos, como 30 tornam o modelo mais restrito às características dos prompts, e valores mais baixos como 5 dão mais liberdade e criatividade ao modelo. Por fim, *seed*, permite manter a consistência dos resultados entre execuções, caso seja escolhida uma *seed* fixa os outputs serão sempre os mesmos enquanto uma *seed* com um valor de -1 cria resultados novos com cada execução.

7.2 Escolha dos *prompts*

Nesta secção, pretendemos justificar a escolha dos *prompts* utilizados. Esta é uma decisão fundamental para a qualidade dos resultados, pois os *prompts* têm um grande impacto na forma como o modelo gera as imagens. Assim, uma seleção adequada é essencial para a obtenção de imagens de qualidade.

Os *prompts* positivos escolhidos foram: “Foto do Bordado de Castelo Branco”. Este é um prompt expectável, uma vez que o treino foi realizado com base no termo “Bordado de Castelo Branco”, tornando-o necessário. Os *prompts* “têxtil”, “feito à mão” e “alto detalhe” têm como objetivo melhorar a qualidade da imagem e transmitir a minuciosidade do bordado nas imagens geradas. Já “simétrico” e “fundo branco” referem-se a características específicas do Bordado de Castelo Branco, garantindo que estas estejam presentes nas imagens.

Quanto aos *prompts* negativos, foram escolhidos “desfigurado”, “feio”, “mau”, “imaturo” e “anime” como uma série de descritores genéricos para evitar resultados indesejados. Os termos “3D”, “preto e branco” e “fundo negro” foram incluídos para corrigir algumas limitações do treino, garantindo que os resultados sejam coloridos e num fundo branco. Adicionámos ainda “arquitetura” e “castelos” para mitigar um erro do modelo, que, devido à presença do termo “Castelo Branco” em “Bordado de Castelo Branco”, tendia a inserir um castelo na cor branca em várias imagens.

7.3 Exemplos de Imagens geradas através de *txt2img*

Neste subcapítulo, pretendemos demonstrar as capacidades do modelo treinado num registo *txt2img*, onde serão enviados dois *prompts* textuais, e o modelo gerará uma imagem com base nestes. Serão apresentadas dez imagens geradas com os mesmos prompts e parâmetros.



Figura 34 – Primeira imagem gerada



Figura 35 – Segunda imagem gerada



Figura 36 – Terceira Imagem gerada

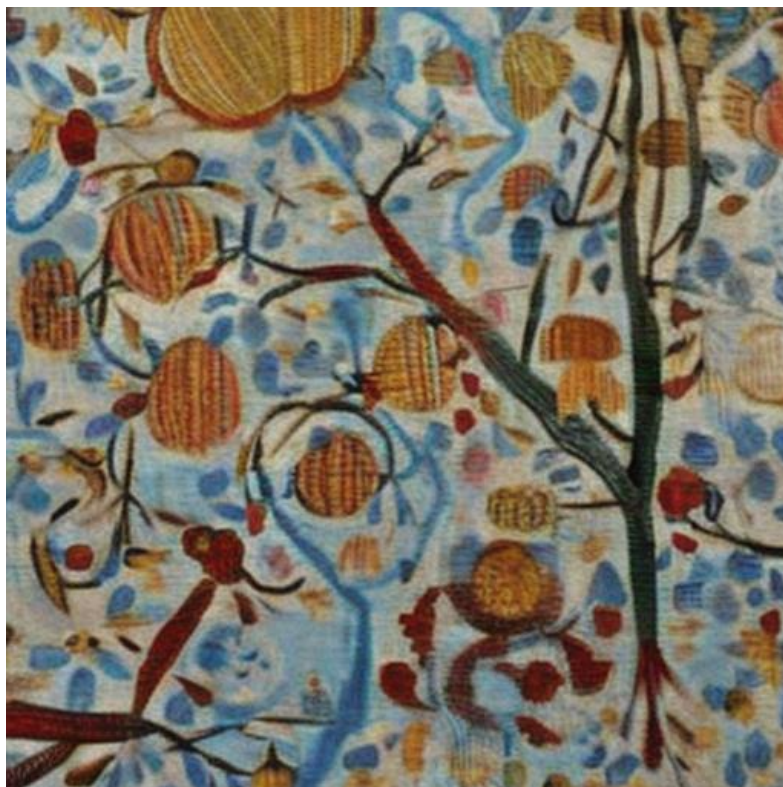


Figura 37 – Quarta Imagem gerada



Figura 38 – Quinta Imagem gerada



Figura 39 – Sexta Imagem gerada



Figura 40 – Sétima Imagem gerada



Figura 41 – Oitava Imagem gerada



Figura 42 – Nona Imagem gerada



Figura 43 – Décima Imagem gerada

Foto do bordado de Castelo Branco, têxtil, feito à mão, alto detalhe, simétrico, fundo branco
Negative prompt: Desfigurado, feio, mau, imaturo, anime, 3D, preto e branco, fundo negro, arquitetura, castelos
Steps: 20, Sampler: DPM++ 2M, Schedule type: Karras, CFG scale: 7, Seed: 1317325874, Size: 512x512, Model hash: 963c5aa9aa, Model: modelo_bordadoCasteloBranco, Version: v1.10.1

Figura 44 – Parâmetros para as imagens geradas anteriormente

7.4 Exemplos de Imagens geradas através de *img2img*

Neste subcapítulo pretendemos demonstrar as capacidades do modelo treinado, agora num registo *img2img*, com a demonstração de dez exemplos onde vão ser enviados dois prompts textuais (iguais aos anteriores) juntamente com uma imagem a transformar e o modelo vai gerar uma imagem com base nestes. Nestes 10 exemplos, 6 deles são de imagens genéricas para teste, com elementos mais simples e com menor complexidade, enquanto as últimas 4 são de fotos tiradas por um dos autores com o objetivo de testar uma utilização comum com fotografias reais.



Figura 45 — Transferência de estilo com uma flor

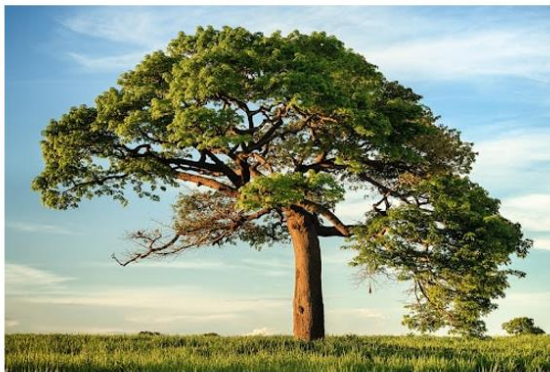


Figura 46 — Transferência de estilo com uma árvore



Figura 47 — Transferência de estilo com uma paisagem



Figura 48 — Transferência de estilo com um rosto



Figura 49 — Transferência de estilo com uma casa

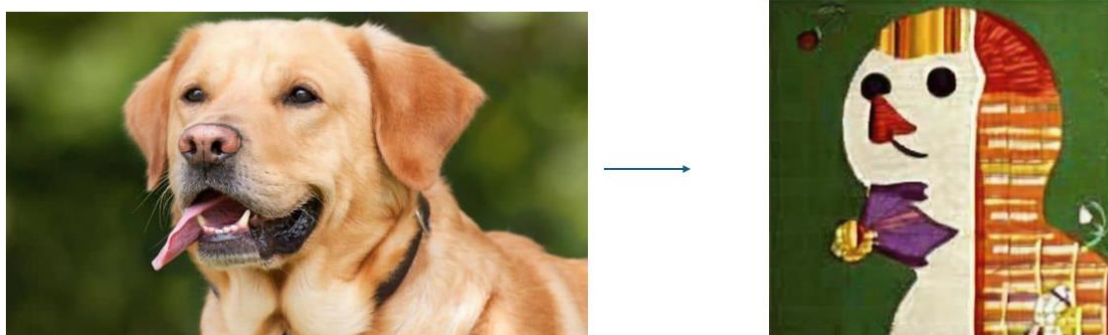


Figura 50 — Transferência de estilo com um cão



Figura 51 — Transferência de estilo com um camelo



Figura 52 — Transferência de estilo com uma águia

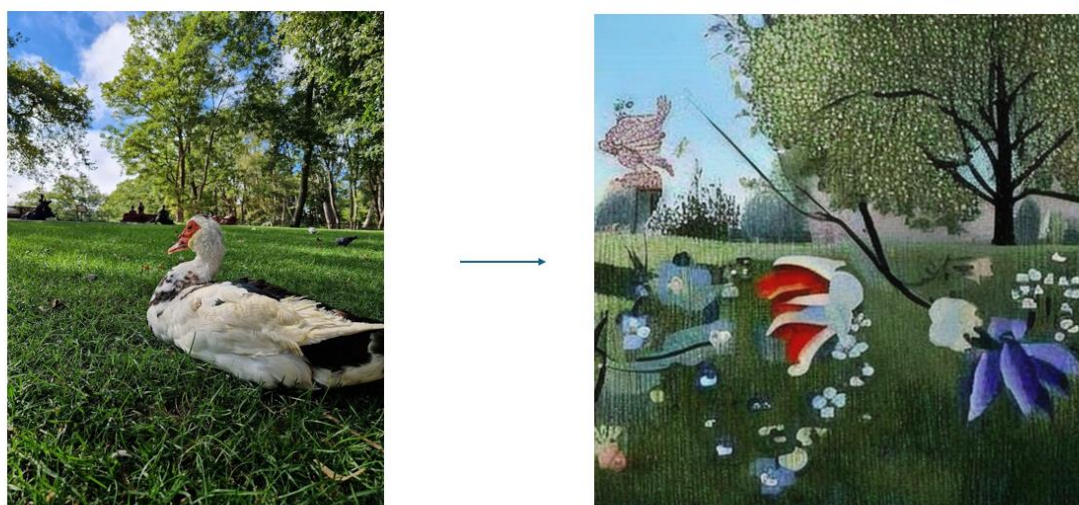


Figura 53 — Transferência de estilo com um ganso



Figura 54 — Transferência de estilo com uma chita

7.5 Avaliação dos Resultados Obtidos pelo Primeiro Modelo

Com base nas imagens obtidas no subcapítulo anterior, podemos concluir que, embora os resultados sejam interessantes, não coincidem totalmente com a natureza do Bordado de Castelo Branco. Este problema é mais evidente no registo *img2img*, onde o modelo apresenta dificuldades em distinguir o conteúdo original da imagem dos elementos estilísticos do bordado. Apesar disso, é possível identificar características que remetem à sua estrutura, como os cravos e flores, apesar de, muitas vezes, desfigurados.

No registo *txt2img*, os resultados foram consistentes, apesar de algumas falhas na clareza dos elementos gerados. Destacam-se a Figura 37 e a Figura 42, que apresentaram melhor alinhamento com o estilo esperado. No entanto, foi notada a inserção de tons de azul que não pertencem à estrutura tradicional do Bordado de Castelo Branco.

Entre os exemplos analisados num registo *img2img*, destacam-se a árvore (Figura 46) e o rosto de um homem (Figura 48), que apresentaram os melhores resultados na distinção entre o estilo e o conteúdo original. Estas imagens demonstram um equilíbrio eficaz entre os elementos estilísticos e a preservação da forma original, resultando em imagens visualmente mais apelativas.

No entanto, ao utilizar fotografias complexas, observou-se um aumento significativo na dificuldade do modelo em realizar a transferência de estilo. Nos quatro últimos exemplos, que incluem fotos reais, o modelo teve dificuldades em reconhecer a silhueta dos objetos, o que levou a resultados distantes das imagens originais. Este problema sugere que um refinamento no *dataset* ou no ajuste dos parâmetros pode ser necessário para melhorar os resultados futuros.

8. Melhorias

Nesta secção apresentam-se as principais melhorias realizadas para enriquecer os resultados dos modelos de modo a incidirem melhor com o estilo do Bordado de Castelo Branco. Ao longo do projeto foi evidente que a qualidade do *dataset* utilizado é fundamental para o sucesso, quer da geração de novas imagens, como também para a transferência de estilos. Assim, seguindo os passos naturais após a conclusão de Projeto I, foi essencial rever algumas das escolhas feitas anteriormente e explorar a possibilidade de melhorias através de escolha / criação de um *dataset* novo.

A revisão dos parâmetros utilizados durante o treino com o *DreamBooth* é também uma etapa importante para melhorar os resultados obtidos, uma vez que, o treino com base nos *tokens* corretos irá facilitar a concessão de novos conceitos, o que, devido à complexidade do estilo do bordado de Castelo Branco, poderá aumentar significativamente a qualidade dos resultados.

Adicionalmente, o uso da interface *A1111*, poderá possibilitar extrair mais dos modelos, a tentativa de otimizar vários parâmetros, através de experimentação, pode ajudar a aumentar a consistência dos *outputs*. A exploração de parâmetros como *cfg_scale*, *denoising_strength*, e o número de *steps* juntamente com o uso de uma *seed* fixa conseguiremos analisar em detalhe o impacto de cada um deles, assim como realizar um estudo de quais colocam em maior destaque os detalhes do estilo do Bordado de Castelo Branco.

Com base nestas melhorias, definimos como estratégia, a criação de dois novos modelos, o primeiro, com foco no teste com um *dataset* de treino menor, mas sem a aplicação de rotações ou recortes, o principal objetivo será a avaliação da capacidade de, através do *DreamBooth*, ensinar um novo conceito partindo de menos imagens, permitindo assim o foco nas imagens como um todo e absorção completa das imagens na sua máxima qualidade. Já o segundo modelo, vai utilizar o *dataset* original que deu origem ao primeiro modelo (com recortes e rotações), mas com parâmetros diferentes, tendo como foco a aprendizagem de um estilo, ao invés de um bordado, como foi feito no semestre passado.

8.1 Primeiro modelo – modelo_melhores

O primeiro modelo desenvolvido no âmbito das melhorias propostas teve como principal objetivo avaliar a capacidade do *DreamBooth* ensinar um novo conceito utilizando um dataset de treino reduzido, composto exclusivamente pelas imagens de maior qualidade e representatividade do Bordado de Castelo Branco por completo.

Neste modelo, foram excluídas imagens com rotações e recortes, de modo a preservar a integridade estética das imagens selecionadas, permitindo que o foco esteja na aprendizagem do estilo como um todo. Desta forma, pretendeu-se maximizar o potencial de aprendizagem do modelo ao garantir que este absorva as características fundamentais do bordado a partir de exemplos completos e de elevada fidelidade, com o objetivo de alcançar outputs com maior precisão e consistência com o estilo do Bordado de Castelo Branco.

8.1.1 Novo *Dataset*

A seleção adequada do *dataset* é uma etapa fundamental para garantir o sucesso do processo de treino do modelo. Nesse sentido, optámos por reduzir o volume de dados, priorizando a qualidade das imagens presentes no nosso dataset. Assim, seleccionámos as melhores imagens disponíveis a partir do *dataset* original, transferindo o foco da quantidade para a qualidade. Este procedimento tem como objetivo garantir que o modelo seja treinado com exemplos mais relevantes, nítidos e informativos relativamente ao estilo desejado.

O processo de filtragem das imagens envolveu a eliminação dos exemplos com baixa resolução, falta de nitidez, enquadramento incorreto ou, ainda, imagens em que o bordado de Castelo Branco não constituía o elemento central (por exemplo, um vestido com padrões semelhantes, mas sem o bordado em destaque). Desta seleção resultaram 76 imagens que melhor representam o bordado permitindo a construção de uma base sólida que reduz inconsistências.

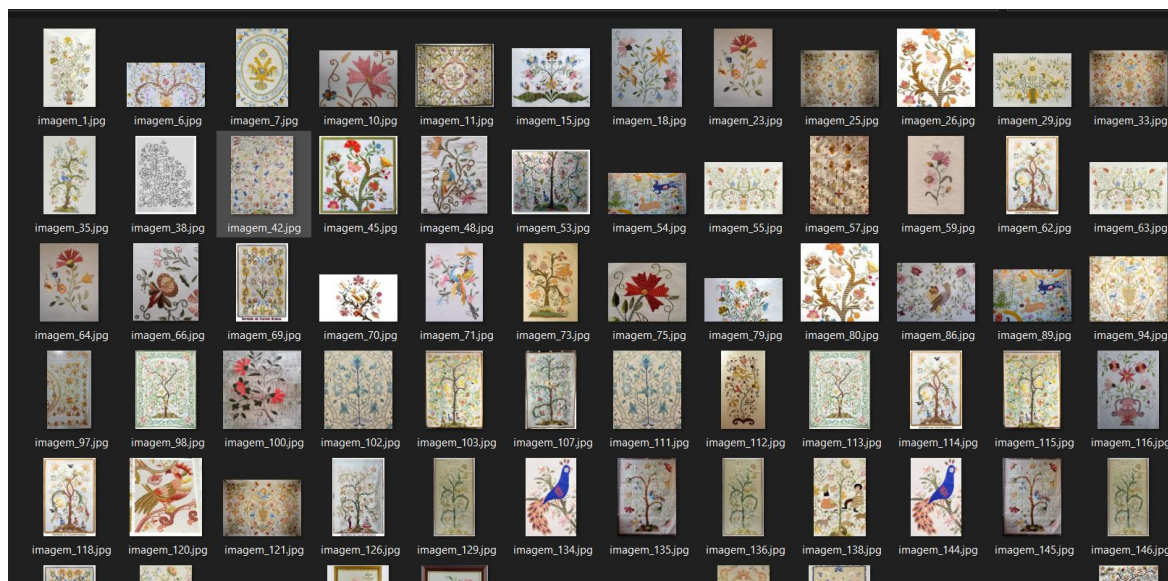


Figura 55 — Excerto do *dataset* utilizado para o modelo_melhores

O principal objetivo da redução do tamanho do *dataset* é permitir ao modelo extrair o máximo de informação possível sobre cada imagem do bordado, dando particular atenção ao seu estilo como um todo. Desta forma, o modelo não apenas aprende separadamente os diferentes elementos do bordado, mas também é capaz de os combinar de forma coerente.

Esta abordagem foi adotada em resultado do estudo do estado da arte realizado no semestre anterior, no qual diversos artigos analisados referiam que, com apenas 20 a 50 imagens, os modelos já conseguiam absorver informação suficiente para gerar outputs satisfatórios.

8.1.2. Parâmetros de treino

Para o processo de treino com o *DreamBooth*, optámos por manter a maioria dos parâmetros inalterados com o objetivo de assegurar a consistência dos resultados e permitir uma comparação rigorosa entre os diferentes modelos desenvolvidos ao longo do projeto. A escolha de manter os valores padrão para parâmetros como a taxa de aprendizagem e o número de passos de treino ajuda a isolar o impacto das alterações introduzidas no *dataset* e nas configurações dos *prompts*, facilitando assim a análise do efeito de cada mudança na qualidade dos outputs gerados.

A única diferença significativa no processo de treino deste modelo em relação ao anterior foi uma alteração nas *strings* de *instance_prompt* e *class_prompt*. Esta estratégia foi adotada com o objetivo de corrigir alguns problemas identificados no

semestre anterior, nomeadamente a tendência do modelo em associar o conceito de bordado de Castelo Branco a imagens literais de castelos ou elementos arquitetónicos brancos. Assim, em vez de utilizar o *prompt* “Foto do Bordado de Castelo Branco” para treinar e gerar imagens, decidimos utilizar a expressão “Foto do Bord4do de C4stelo Br4nco”, substituindo os caracteres ‘a’ por ‘4’.

Esta alteração, aparentemente simples, tem um impacto relevante na forma como se vão construir os *prompts*. Ao substituir as vogais ‘a’ por números, criamos um identificador único e distinto para o conceito de bordado, afastando o modelo de associações indesejadas com imagens de castelos ou objetos brancos. Desta forma, o modelo passa a focar-se nas características visuais e estilísticas do bordado em si, aprendendo a reconhecer os padrões, texturas e cores específicas do Bordado de Castelo Branco, sem confusões com outros elementos semânticos.

8.2 Segundo modelo – modelo_styles

Já no segundo modelo a abordagem seguida teve como foco a melhoria das *strings* que conduzem a aprendizagem. O objetivo principal foi avaliar a capacidade de melhorar o modelo treinado apenas alterando estes parâmetros. Esta abordagem tem como objetivo avaliar a capacidade de, com as mesmas imagens, aprender melhor um conceito se for introduzido de forma diferente ao modelo.

Nesta abordagem treinamos o modelo para aprender um estilo específico, ou seja, o invés de usar as *strings* “Foto do ...” optámos por utilizar “Estilo do ...”. Desta forma iríamos permitir ao modelo aprender especificamente o estilo do bordado e não a capacidade de reproduzir fotos do mesmo. Esta tentativa tenta melhorar a capacidade do modelo de gerar bons resultados numa vertente *txt2img*, já que, poderia facilitar a junção com outros conceitos já que conceber uma foto de um Bordado de Castelo Branco com um bulldog pode ser mais desafiador que um bulldog representado no estilo do bordado.

8.2.1. Dataset utilizado

Para este modelo, decidimos manter o *dataset* original utilizado no primeiro modelo treinado em Projeto I. Esta escolha foi importante para garantir a comparação justa entre os resultados obtidos pelos diferentes modelos. Ao não alterar os dados de treino, conseguimos isolar o efeito das mudanças realizadas nos parâmetros de treino, assegurando que qualquer diferença observada nos outputs fosse resultado exclusivo da nova estratégia e não de variações no conteúdo das imagens. Desta forma, criámos condições controladas para avaliar o verdadeiro impacto da alteração das *strings* utilizadas no treino, permitindo uma análise mais precisa do quanto a nova abordagem, focada no estilo em vez da

fotografia. Além disso, esta consistência no dataset permitiu que a comparação entre o modelo `styles` e os modelos anteriores fosse direta e objetiva, funcionando como um ponto de referência sólido para medir as melhorias com esta estratégia.

8.2.2. Parâmetros de treino

Esta etapa representa a principal inovação em relação aos modelos anteriores: apesar de mantermos idênticos parâmetros como *training_steps* e *learning_rate*, introduzimos uma alteração fundamental nas *strings* de *instance_prompt* e *class_prompt*. O objetivo é reorientar o foco do treino realizado no modelo para que, em vez de buscar a reprodução fiel de imagens do bordado, passamos a priorizar a aprendizagem do seu estilo visual. Tal como feito no modelo `melhores`, mantivemos o treino com base na designação Bord4do de C4stelo Br4nco para, tal como referido anteriormente, evitar a inclusão literal de castelos na cor branca e criando assim um identificador único que referencie o treino realizado sem interferir com outros conceitos pré-existentes.

Esta reformulação tem como objetivo permitir ao modelo aplicar as características estilísticas aprendidas em diversos contextos, transcendendo a replicação fotográfica. A nova abordagem permite melhorar a geração de imagens com objetos ou cenários completamente novos como o caso de um animal ou uma paisagem.

A motivação para esta mudança surgiu de uma análise crítica ao desempenho do modelo original. Observámos que, ao combinar conceitos (ex: "Foto de um X em bordado de castelo branco") havia uma ausência de um conector claro entre os elementos gerando resultados inconsistentes. A nova formulação ("Estilo do Bordado de Castelo Branco") resolve esta limitação ao estabelecer explicitamente o *estilo* como entidade transferível.

8.3 Melhorias nos *Prompts*

A otimização dos *prompts* utilizados na geração das imagens representa uma etapa fundamental para a melhoria dos resultados obtidos pelos modelos de *Stable Diffusion*. Com o objetivo de maximizar a fidelidade ao estilo do Bordado de Castelo Branco e garantir resultados visualmente ricos e detalhados, fizemos uma revisão cuidada aos *prompts* utilizados no modelo original, tornando-os mais descritivos, específicos e alinhados com o estilo desejado. Esta abordagem implicou alterações tanto nos *prompts* positivos como nos negativos, de modo a orientar o modelo para a produção de imagens mais consistentes e de melhor qualidade.

Relativamente aos *prompts* positivos, introduzimos um primeiro *token* muito importante, “*A majestic X in the style of bordado de Castelo Branco*”, que se tornou o elemento principal dos nossos *prompts*, onde X representa o elemento pedido. Esta *string* foi concebida para ativar o identificador único criado durante o treino do modelo, enquanto transmite ao modelo a luxúria e detalhes coloridos, características marcantes do bordado. Palavras como “*majestic*” contribuem para a criação de imagens com maior impacto visual, facilitando também a combinação do estilo com outros conceitos, quando se pretende aplicar o padrão do bordado a novos contextos.

Adicionámos também uma série de *tokens* adicionais, como “*ultra high detail*”, “*colorful silk thread look*”, “*handcrafted texture*”, “*linen background*”, “*elegant symmetry*”, “*museum-quality artwork*”, “*full of vibrant patterns and ornaments*” e “*super high quality*”. Estes termos reforçam a atenção do modelo para aspetos como o fio de seda, a textura artesanal, a simetria elegante e a riqueza ornamental, orientando a geração para uma estética mais próxima do Bordado de Castelo Branco.

De forma paralela, de modo a filtrar resultados indesejados e artefactos visuais recorrentes, foram introduzidos novos *tokens* nos *prompts* negativos, mantendo alguns como “*ugly*”, “*disfigured*”, “*malformed*” e “*immature*”, mas acrescentando outros termos que visam suprimir dificuldades específicas deste tipo de modelos, tais como “*poorly drawn*”, “*low quality*”, “*blurry*”, “*lowres*”, “*bad anatomy*”, “*bad proportions*”, “*extra limbs*”, “*missing limbs*”, “*bad hands*” e “*bad eyes*”. Para além destes, incluímos *tokens* que ajudam a corrigir problemas recorrentes identificados no modelo original, nomeadamente o uso incorreto de cores e a inclusão indesejada de castelos nas imagens geradas, através de palavras como “*black and white*” e “*castles*”. Adicionalmente, para reforçar a importância da cor, da qualidade da imagem e da representação 2d, foram adicionados os *tokens* “*3d*”, “*sketch*” e “*monochrome*”.

A utilização conjunta destes *prompts* positivos detalhados e *prompts* negativos específicos é fundamental para direcionar o modelo na produção de imagens que não só lembram ao estilo do Bordado de Castelo Branco, mas também apresentam uma elevada qualidade nos detalhes, minimizando erros comuns na geração de imagens com inteligência artificial e aumentando a consistência entre os *outputs*. Esta abordagem permite um controlo mais rigoroso sobre a estética final das imagens, contribuindo para resultados mais alinhados com os objetivos do projeto.

8.3.1. Como melhorar os *prompts* utilizados

Nos modelos de *Stable Diffusion* não existe uma fórmula exata para criar *prompts* perfeitos, mas a experiência demonstra que a escolha criteriosa dos mesmos pode melhorar substancialmente a qualidade das imagens geradas. Para este projeto, a otimização dos *prompts* centrou-se na descrição detalhada do estilo do Bordado de Castelo Branco, com a inclusão de *tokens* como “handcrafted” e “high detail”, e na exclusão de elementos indesejados através de *prompts* negativos claros.

A experimentação com diferentes combinações de *prompts*, sempre mantendo os restantes parâmetros constantes, permitiu identificar quais os *tokens* que mais contribuem para a fidelidade estilística e a riqueza visual das imagens. A adaptação dos *prompts* para inglês e a criação de um identificador único (“bord4do de C4stelo Br4nco”) facilitaram a ativação consistente do conceito pretendido.

Assim, a melhoria dos *prompts* utilizados baseia-se numa abordagem iterativa, exploratória e adaptativa, que valoriza a clareza das instruções de forma a garantir resultados mais alinhados e consistentes com o estilo desejado.

8.3.2. Estudo do impacto dos *prompts* textuais em *txt2img*

Para demonstrar de forma clara e objetiva a influência da correta seleção de *prompts* na geração de imagens, serão apresentados resultados comparativos utilizando a mesma *seed* com diferentes configurações textuais no ambiente *txt2img*. Esta análise irá incluir todos os modelos treinados no projeto, permitindo visualizar concretamente o impacto dos *prompts* na estrutura das imagens geradas.

Na figura a baixo estão os parâmetros que deram origem à primeira imagem feita pelo modelo original utilizando os *prompts* sugeridos em projeto I.

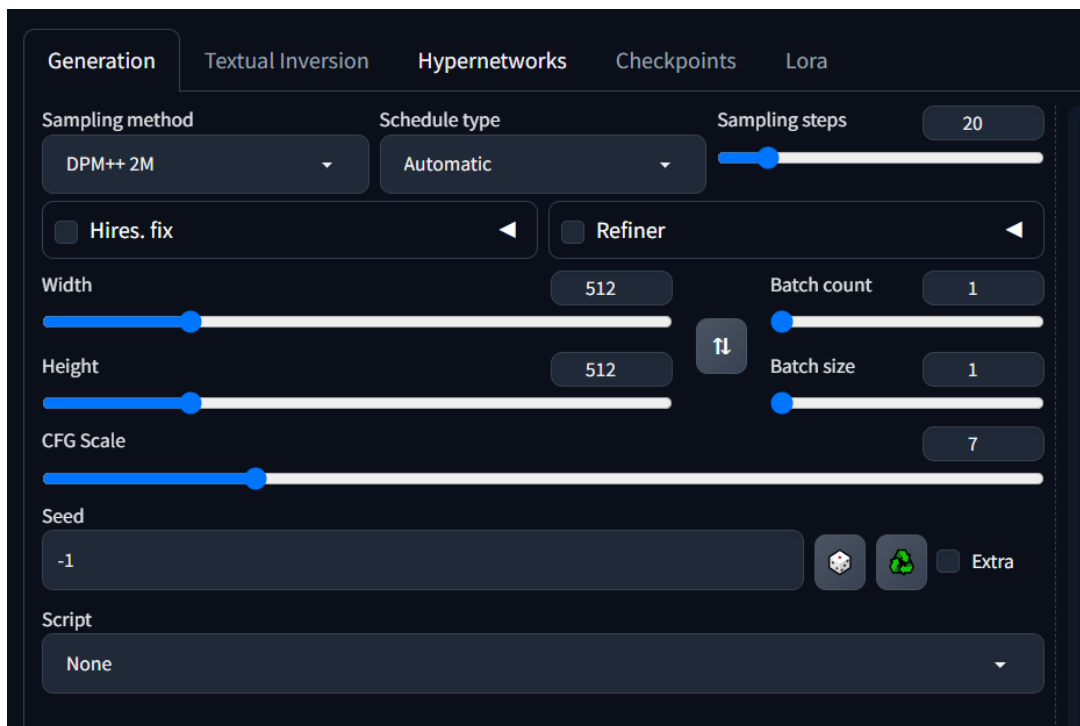


Figura 56 — Parâmetros para a criação da imagem

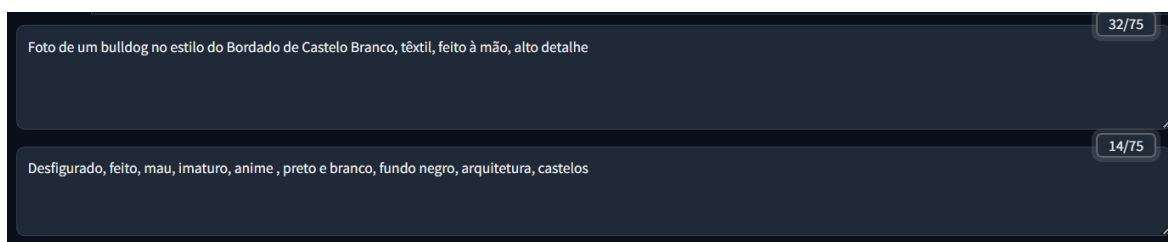


Figura 57 — Prompts utilizados na criação da imagem



Figura 58 — Imagem gerada pelo modelo original

Como é possível verificar o modelo original não consegue conceber facilmente a junção destes dois elementos, consequentemente, não conseguindo nem fazer um desenho satisfatório de um bulldog, nem incorporar corretamente com os restantes elementos do bordado.

De forma a criar uma comparação justa entre o *output* gerado pelo modelo original com os *prompts* originais vamos avaliar a qualidade das imagens geradas pelos modelos mais recentes com os mesmos *prompts*, mas adaptando-os para o novo identificador único (bord4do de C4stelo Br4nco) e traduzindo para inglês. Os antigos *prompts* adaptados aos novos modelos ficaram: positivo = “*A bulldog in the style of bord4do de C4stelo Br4nco, textile, handcrafted, high detail*” negativo = “*disfigured, ugly, malformed, immature, black and white, black background, architecture, castles*” como é possível confirmar na figura abaixo.

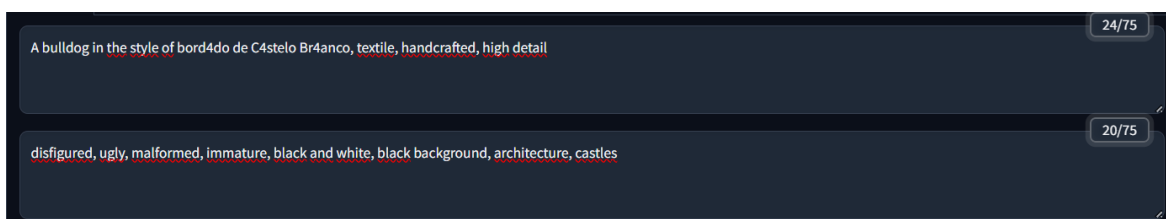


Figura 59 — Antigos *prompts* ajustados aos novos modelos



Figura 60 — Imagem gerada pelo modelo_styles com os *prompts* adaptados



Figura 61 — Imagem gerada pelo modelo_melhores com os *prompts* adaptados

Mesmo os outputs dos novos modelos não sendo extremamente satisfatórios, é possível confirmar que, quer a presença do animal (com dificuldades no modelo_melhores), quer a inclusão dos elementos do bordado de Castelo Branco foram mais bem conseguidos que o modelo original, partindo da mesma base nos *prompts*. Apesar da falta de simetria em ambos os resultados, podemos retirar a conclusão de que, para este caso, o modelo_styles conseguiu incorporar melhor o bulldog na imagem, enquanto, no modelo_melhores, os elementos do bordado foram melhor conseguidos tendo maior dificuldade na inclusão do animal.

O próximo passo é verificar qual o resultado de todos os modelos, mas agora com os novos *prompts*. Na figura a baixo estão os prompts utilizados para a geração das próximas imagens. Os restantes parâmetros irão permanecer inalterados.

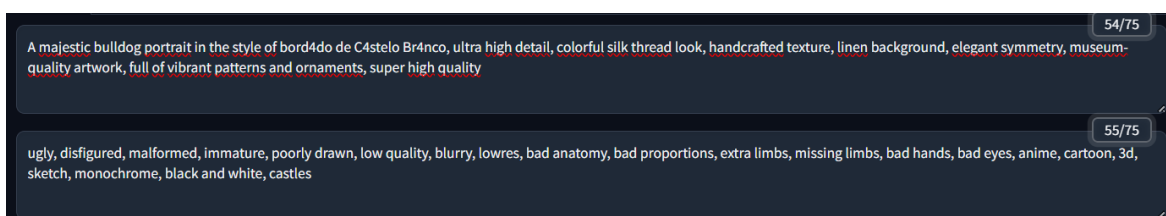


Figura 62 — *Prompts* das próximas imagens a serem geradas



Figura 63 — Imagem gerada pelo modelo_melhores

O modelo_melhores não conseguiu corretamente juntar ambos os conceitos, desta vez, já nem sendo possível ver qualquer elemento que faça lembrar a silhueta do animal pedido no *prompt* positivo. Apesar da dificuldade de geração da imagem, neste caso específico, o modelo parece ter assimilado melhor alguns dos elementos e cores do bordado apesar da falta de simetria.

Na figura a baixo está o output do modelo_styles com os *prompts* e parâmetros anteriores.



Figura 64 — Imagem gerada pelo modelo_styles

O modelo_styles conseguiu de forma mais satisfatória a inclusão do animal, incluindo diversos detalhes como uma expressão facial mais correta, detalhes como sombras ligeiras na zona das orelhas e olhos muito mais bem conseguidos. Quanto à parte do bordado, deixou um pouco a desejar, embaralhando muito os elementos do mesmo, não conseguindo uma simetria agradável, e um incorreto uso da cor causado pelo uso em demasia das cores verde e azul.

Para finalizar os testes com os novos *prompts* realizámos uma adaptação dos mesmos para coincidirem com o treino realizado no modelo original, ou seja, fizemos de volta a passagem de “Bord4do de C4stelo Br4nco” para “Bordado de Castelo Branco” já que este foi o identificador do treino do modelo original. Após as adaptações os *prompts* ficaram como é possível ver na figura abaixo.

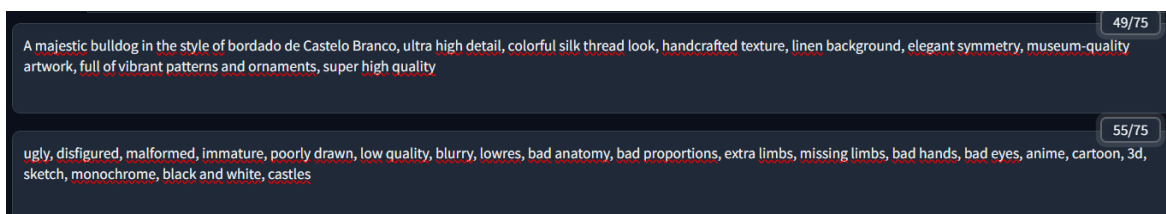


Figura 65 — Adaptação dos *prompts* novos ao modelo original



Figura 66 — Imagem gerada pelo modelo original com os *prompts* adaptados

A imagem gerada pelo modelo original ficou longe dos restantes outputs, não conseguindo incluir qualquer elemento com aparência a um bulldog, apesar disto, aparenta ter incluído mais elementos corretos do bordado com um equilíbrio agradável destacando um uso muito mais correto da cor do que nos modelos mais recentes.

De todos os modelos a versão *styles* foi a que gerou o *output* mais satisfatório já que este modelo conseguiu corretamente incluir o elemento pedido na imagem, apesar de ter perdido alguma essência do bordado de Castelo Branco com alguma distorção dos seus emblemáticos símbolos. Os cravos não aparecem na sua totalidade e apesar de haver alguma simetria presente, as figuras aparentam estar todas embaralhadas.

Para analisar melhor os efeitos dos *prompts* nas abordagens *txt2img* vamos realizar mais alguns testes, só que agora com apenas alguns dos *tokens*, para compreendermos quais são os que têm maior peso na qualidade dos outputs. No próximo teste iremos mostrar como seriam os resultados ao utilizar a mesma *seed* e o mesmo modelo, mas desta vez, sem os *tokens* que fazem alusão à qualidade da imagem. Na figura a baixo estão os prompts que sobraram após essa remoção.

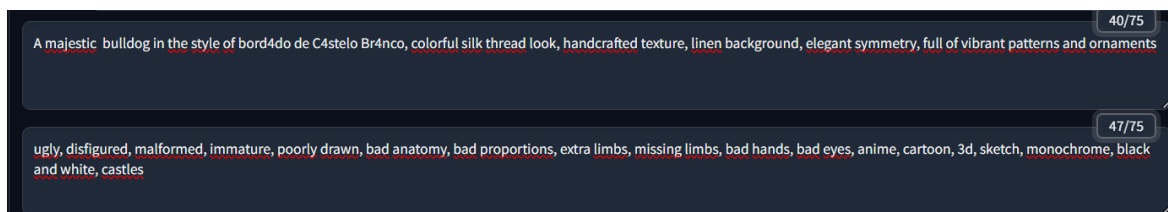


Figura 67 — Prompts sem a referência à qualidade da imagem



Figura 68 — Output obtido sem referência à qualidade nos *prompts*

Apesar de subtil, existe uma diferença entre ambos os *outputs*, especialmente quanto ao uso da cor no bordado. No primeiro teste (Figura 64) o modelo conseguiu de forma bastante satisfatória utilizar as cores corretas até conseguindo incluir alguns detalhes como sombras junto às orelhas do animal já na versão sem *prompts* que referem a qualidade o modelo acabou por se usar a cor verde no fundo, assim como uma perda de qualidade bastante grande nos contornos do animal, não sendo possível distinguir tão facilmente ver as várias partes do animal.

De seguida iremos testar se os *prompts* positivos que descrevem textualmente o bordado têm ou não alguma influência na qualidade da imagem gerada pelo modelo_styles. Para isso, removemos os *tokens* que fazem referência às características do bordado como “*colorful silk thread look*”, “*handcrafted texture*”, “*linen background*”, “*elegant symmetry*” e “*full of vibrant patterns and ornaments*”

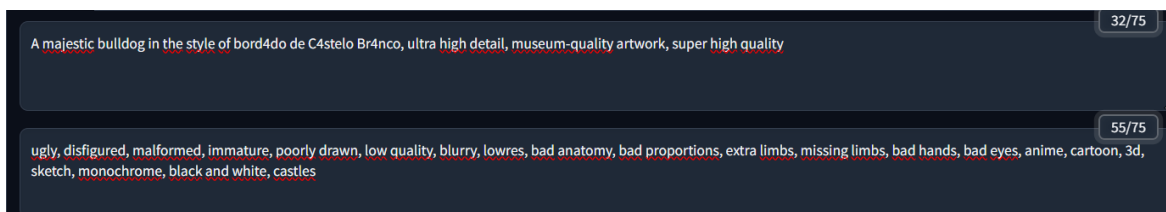


Figura 69 — *Prompts* sem descrição do estilo do bordado de Castelo Branco



Figura 70 — *Output* sem descrição do estilo do bordado de Castelo Branco

A remoção dos *tokens* que explicam textualmente o estilo do bordado têm um impacto significativo na qualidade da imagem, sem eles, o modelo parece embaralhar ainda mais os elementos do bordado, assim como uma ainda mais aparente falta de simetria causada pela inclusão de uma flor. O uso da cor permaneceu incorreto com a presença da cor verde sendo a mais predominante.

Concluindo, avaliar o impacto isolado de cada *token* nos *prompts* revelou-se bastante desafiador. Quando testada a remoção de grupos de *tokens*, observámos que essa exclusão não provocou alterações significativas nos resultados obtidos.

Isto indica que o modelo tende a responder mais ao conjunto dos *prompts* do que a cada termo individualmente, reforçando a importância de uma construção contextual.

9. Parâmetros dentro do A1111

A correta escolha de parâmetros em modelos de *Stable Diffusion* é essencial para gerar imagens alinhadas com estilos complexos, como o bordado de Castelo Branco. Dentro do *A1111*, o *sampling method* e o *scheduler* destacam-se como elementos críticos: o primeiro controla o processo de remoção de ruído durante a difusão reversa, enquanto o segundo controla a progressão do ruído ao longo das etapas de geração. A eficiência e qualidade dos resultados dependem diretamente destes componentes, influenciando o tempo de processamento e a fidelidade estilística das imagens finais [70].

Outros parâmetros menos complexos como o número de *steps* e o valor usado em *cfg_scale* permitem um ajuste fino das imagens geradas. Os valores de *cfg_scale*, variam normalmente entre 5-15 permitindo ao modelo ganhar ou perder aderência ao *prompt* dependendo do quão alto o valor seja. Já a variação do valor escolhido em *steps* é uma forma direta de aumentar a computação efetuada para cada imagem, o aumento desta irá consequentemente levar também a um aumento do tempo necessário para cada imagem em troca de maior qualidade.

9.1 O que fazem?

O *sampling method* atua como o algoritmo central que converte uma imagem ruidosa em uma imagem, definindo a estratégia de *denoising* em cada etapa da difusão reversa. Métodos como *Euler*, *DDIM* ou *DPM++* variam em velocidade, estabilidade e capacidade de preservar nuances complexas, sendo a sua escolha determinante para o equilíbrio entre eficiência computacional e qualidade do output. Esta seleção tem um impacto importantíssimo para a coerência estilística, pois diferentes algoritmos podem priorizar detalhes realistas ou interpretações criativas do mesmo *prompt* [70].

Por sua vez, o *scheduler* organiza a dinâmica de adição e remoção de ruído, definindo a taxa de ruído removido ao longo dos *steps* e moldando a transição entre caos e estrutura. Parâmetros como o número de etapas e o tipo de curva (*linear*, *exponencial* ou *Karras*) afetam a suavidade dessa evolução, influenciando a estabilidade das imagens geradas e equilíbrio entre elementos artísticos. A otimização conjunta com o *sampling method* permite ajustes precisos no *pipeline*, essenciais para garantir a inclusão das características únicas do bordado de Castelo Branco [70].

9.2 Como melhorar?

Tal como no estudo dos *prompts* textuais, não existe uma estratégia universalmente correta para a otimização destes parâmetros. A abordagem mais eficaz depende sempre de uma série de fatores, incluindo o modelo base utilizado, o tipo de treino efetuado e as características específicas do estilo desejado para a imagem final.

No nosso caso, adotámos uma metodologia experimental, testando uma ampla variedade de combinações de parâmetros, tais como *sampling method*, *scheduler*, número de *steps* e *cfg_scale*, para identificar aquelas que proporcionam resultados visuais mais satisfatórios e alinhados com o estilo do bordado de Castelo Branco. Este processo envolveu a realização de múltiplos testes, comparação sistemática dos resultados e ajustes, permitindo-nos compreender como cada parâmetro influencia a qualidade e a fidelidade estilística das imagens geradas.

9.3 Como os parâmetros influenciam o resultado em *txt2img*

A experimentação de parâmetros no *A1111* é uma etapa fundamental para otimizar a geração de imagens alinhadas com o estilo do Bordado de Castelo Branco. Os testes apresentados nesta secção concentraram-se em quatro principais elementos: *sampling method*, *scheduler*, *steps* e *cfg_scale*, cujas combinações influenciam diretamente na fidelidade estilística coerência visual.

Os testes iniciais foram realizados com o objetivo de identificar o *sampling_method* mais adequado ao estilo pretendido. Para garantir a consistência dos resultados e permitir uma comparação justa entre todas as configurações, mantivemos constantes a *seed*, os *prompts* e o modelo base utilizado (neste caso, o modelo *styles* devido à sua superioridade). O parâmetro *scheduler* foi igualmente fixado, assumindo o valor *Uniform*, de forma a minimizar a sua influência na análise dos restantes parâmetros, embora outros valores venham a ser explorados numa fase posterior deste capítulo. Esta abordagem permitiu-nos isolar e compreender de forma mais precisa o impacto específico da escolha do *sampling_method* nos resultados obtidos.

Os *prompts* selecionados para estes testes mantiveram o tema recorrente nos restantes de testes e resultados, pedindo ao modelo que gerasse uma imagem de um bulldog no estilo do bordado de Castelo Branco. Esta opção surgiu para garantir a consistência entre os resultados dos diferentes testes realizados ao longo do projeto. Ainda que não se faça uma comparação direta entre todos os testes, a uniformidade dos *prompts* permite acompanhar de forma mais clara o progresso e

a evolução das melhorias. É possível visualizar na imagem abaixo os prompts escolhidos para os testes.

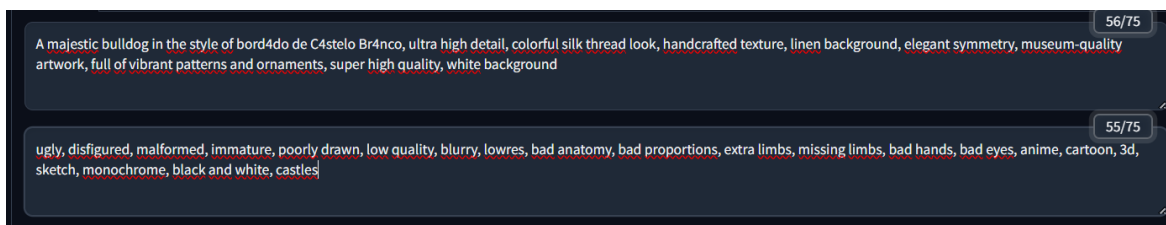


Figura 71 — Prompts para o estudo dos parâmetros

9.3.1. Estudo do parâmetro *sampling_method*

Segue em baixo (Figuras 72 a 87) o resultado do estudo da seleção dos diversos *sampling_methods*, do lado esquerdo da cada imagem irá estar o output gerado pelo modelo, enquanto do lado direito estarão os parâmetros escolhidos.

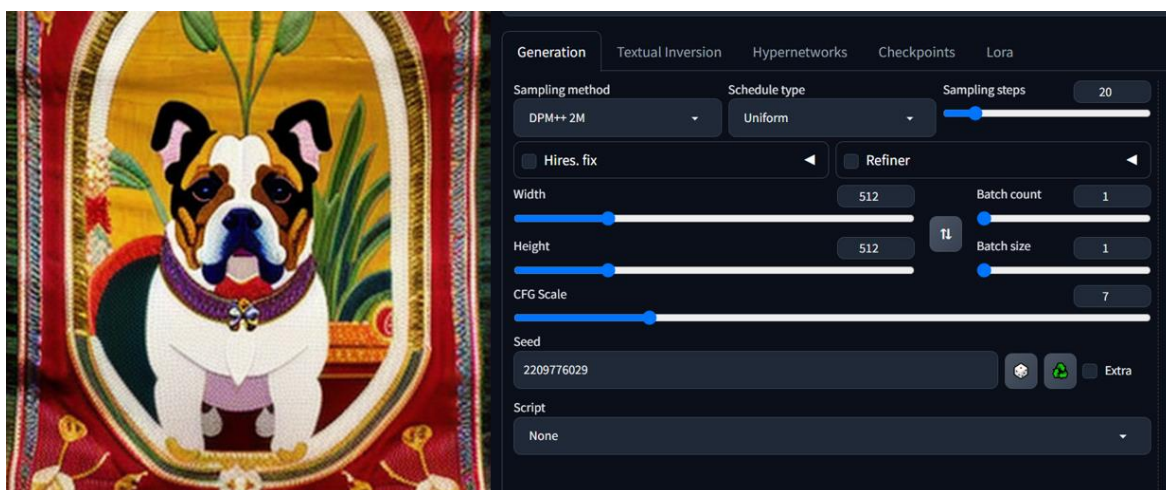


Figura 72 — Teste para *DPM++2M*

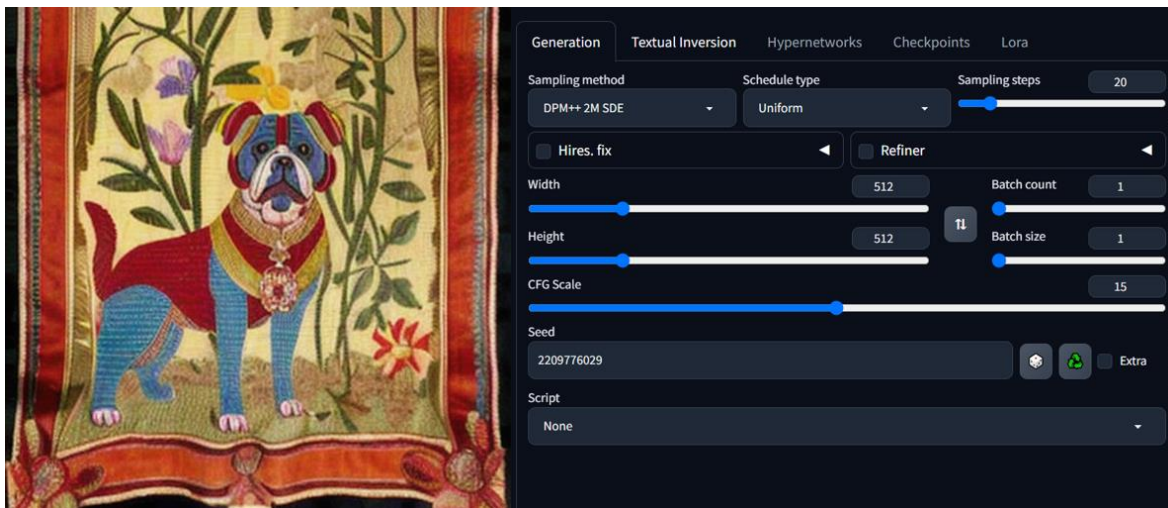


Figura 73 — Teste para *DPM++ 2M SDE*

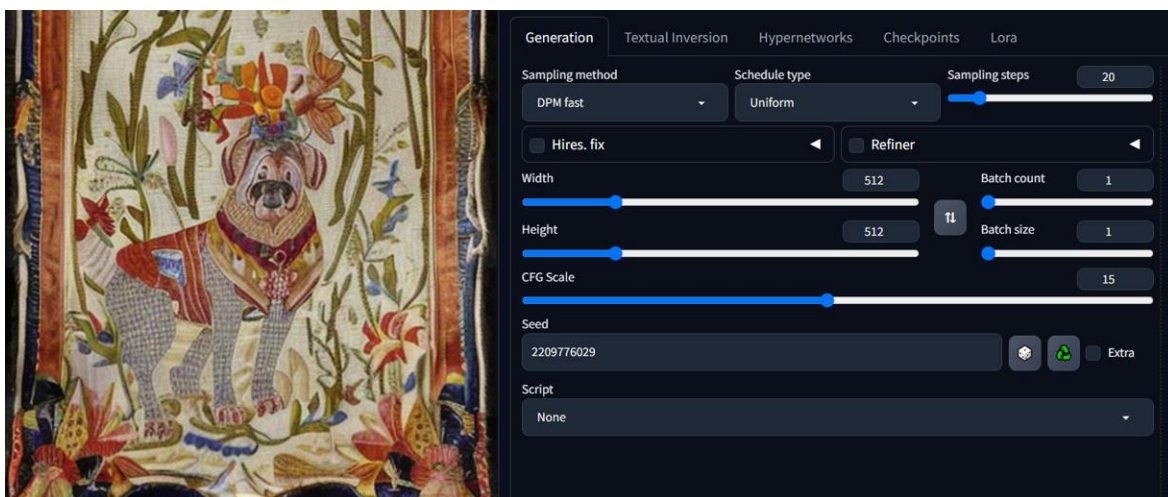


Figura 74 — Teste para *DPM fast*

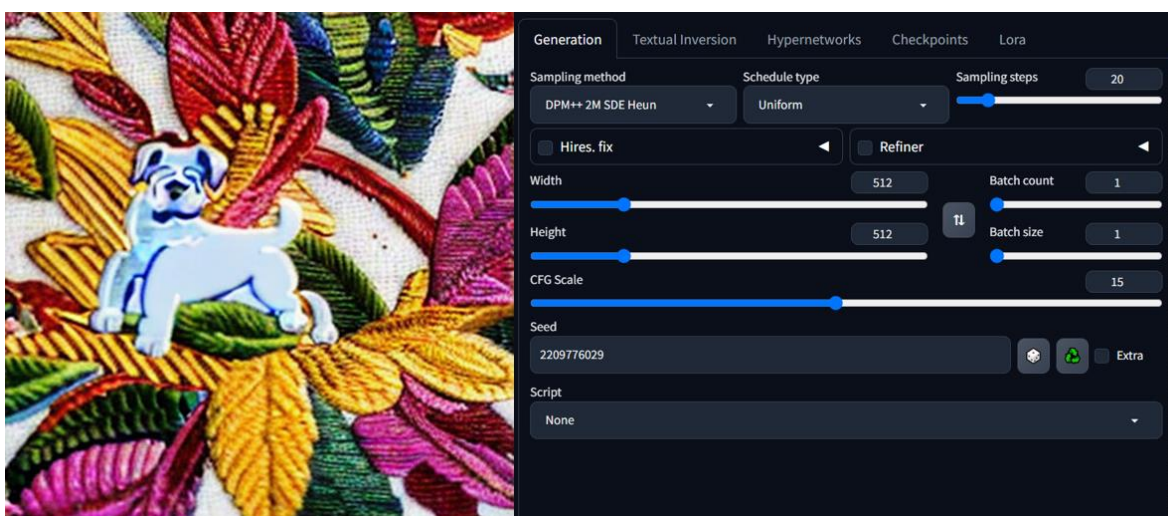
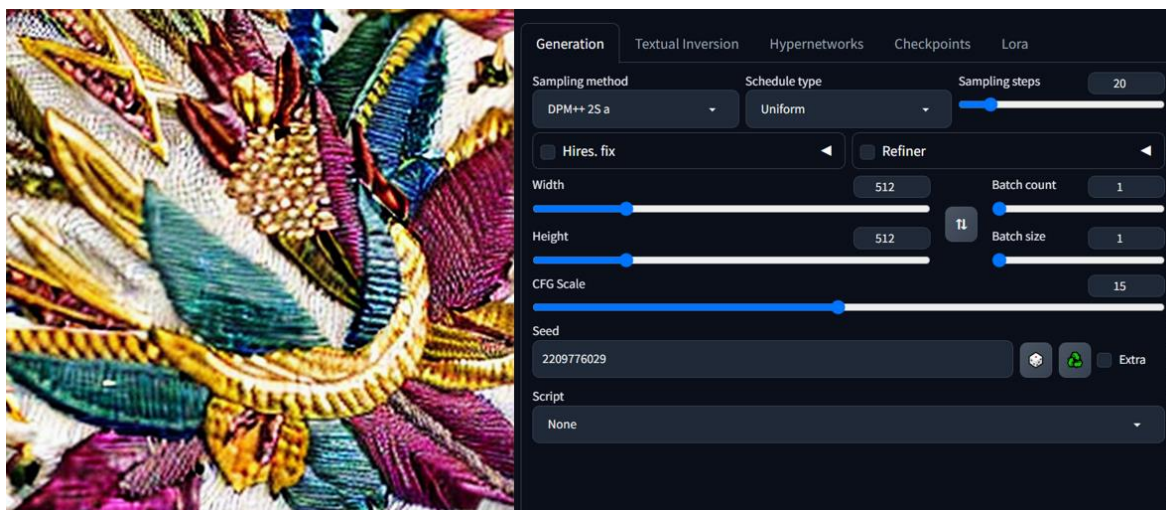
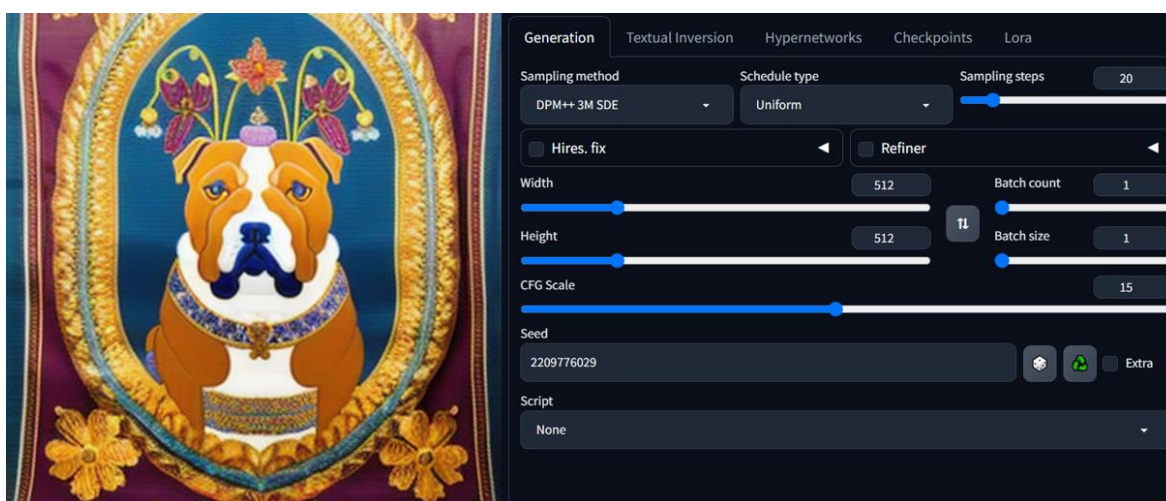
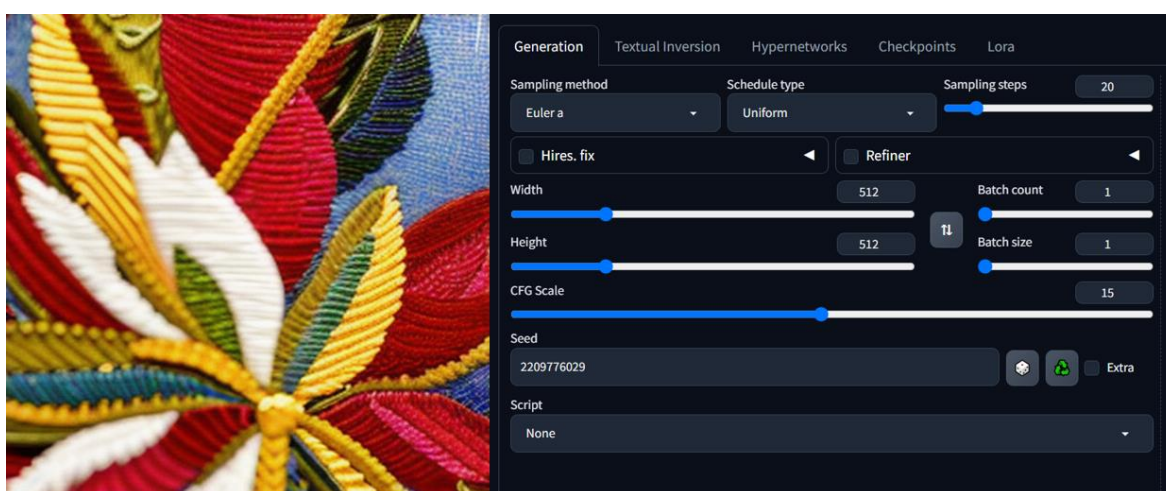


Figura 75 — Teste para *DPM ++2M SDE Heun*

Figura 76 — Teste para *DPM ++ 2S a*Figura 77 — Teste para *DPM ++ 3M SDE*Figura 78 — Teste para *Euler a*

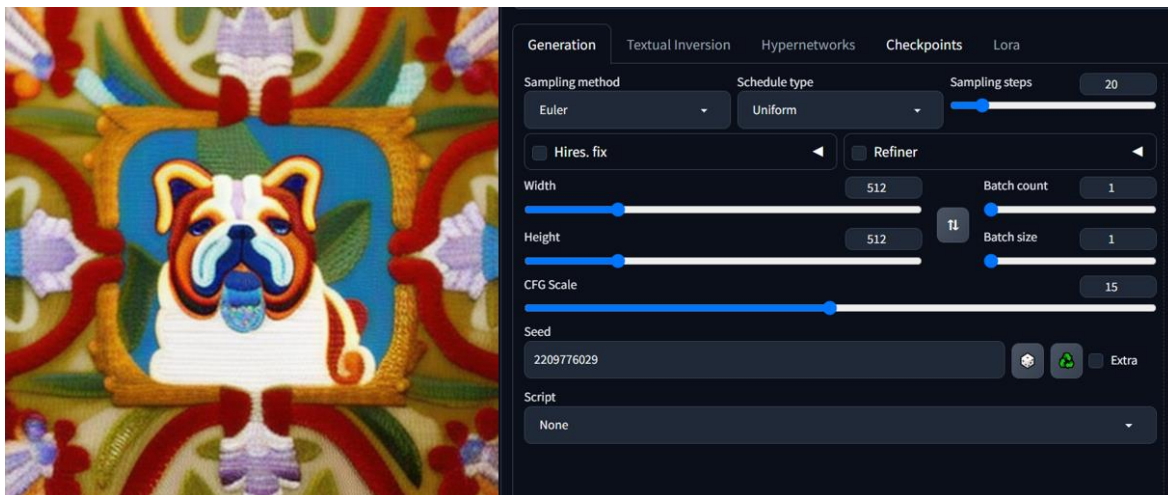


Figura 79 — Teste para *Euler*

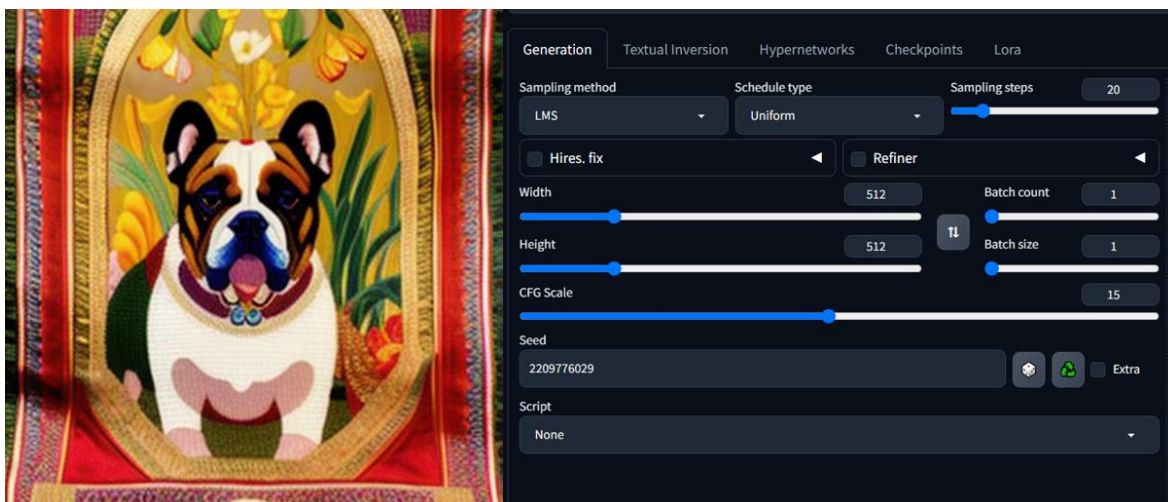


Figura 80 — Teste para *LMS*

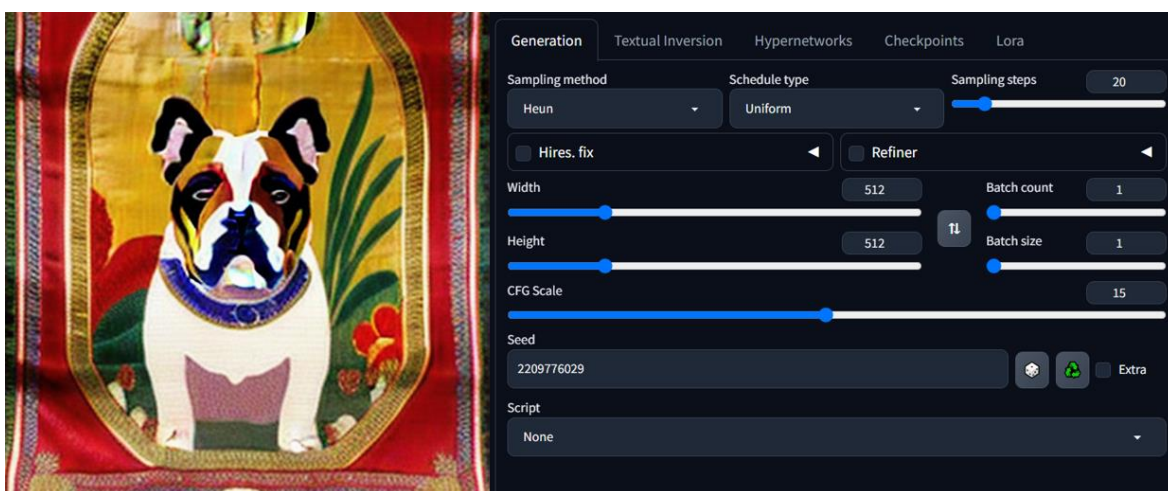
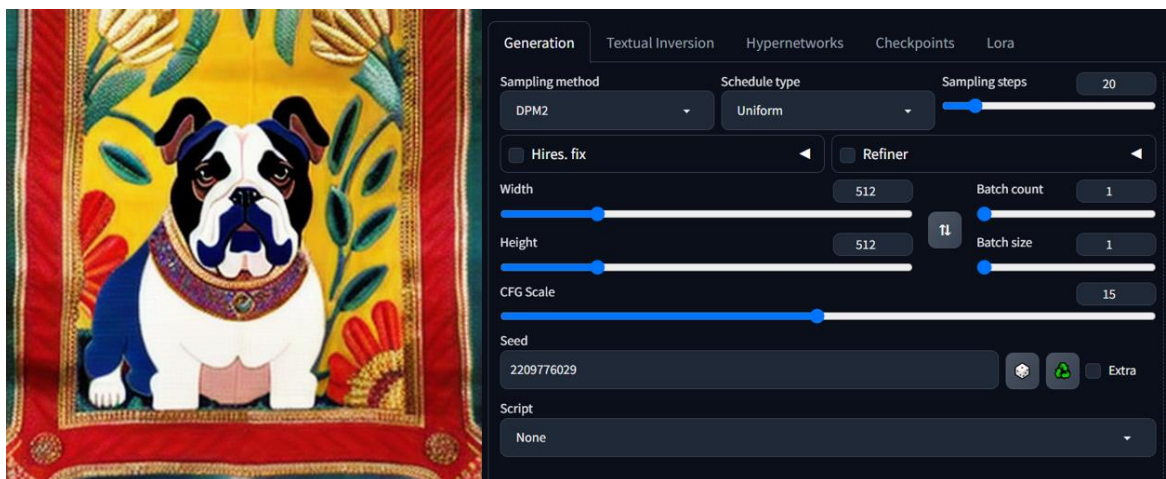
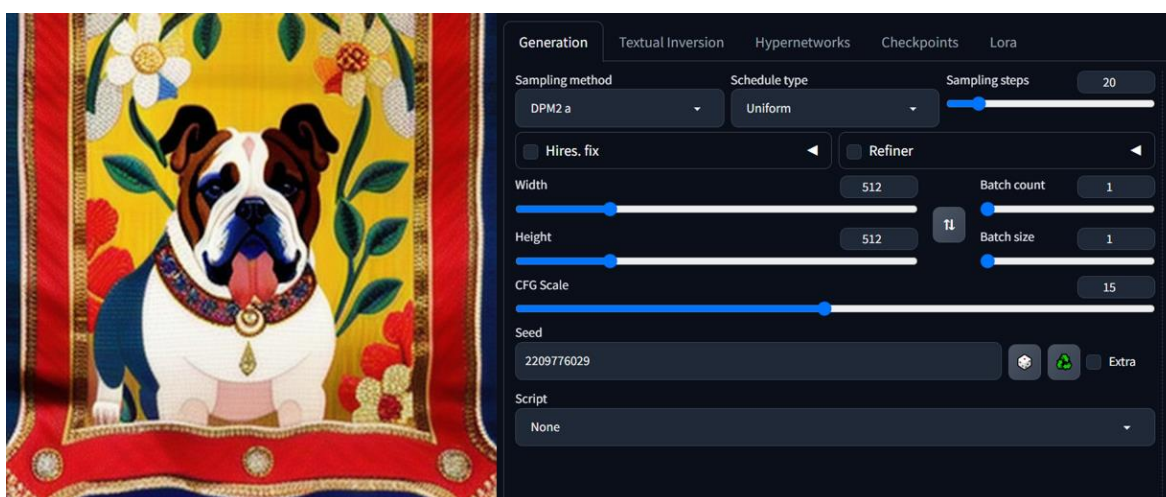
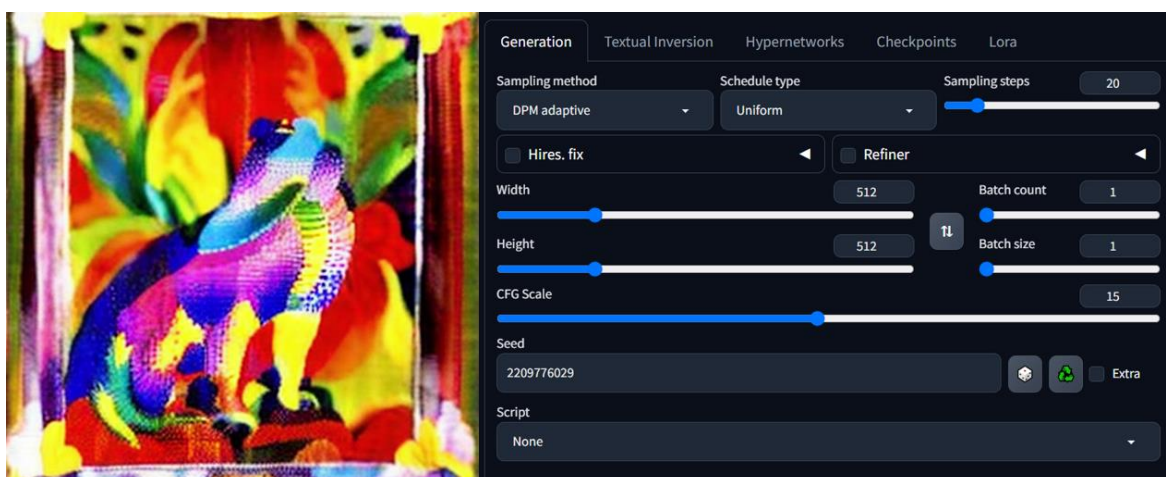


Figura 81 — Teste para *Heun*

Figura 82 — Teste para *DPM2*Figura 83 — Teste para *DPM2 a*Figura 84 — Teste para *DPM adaptive*

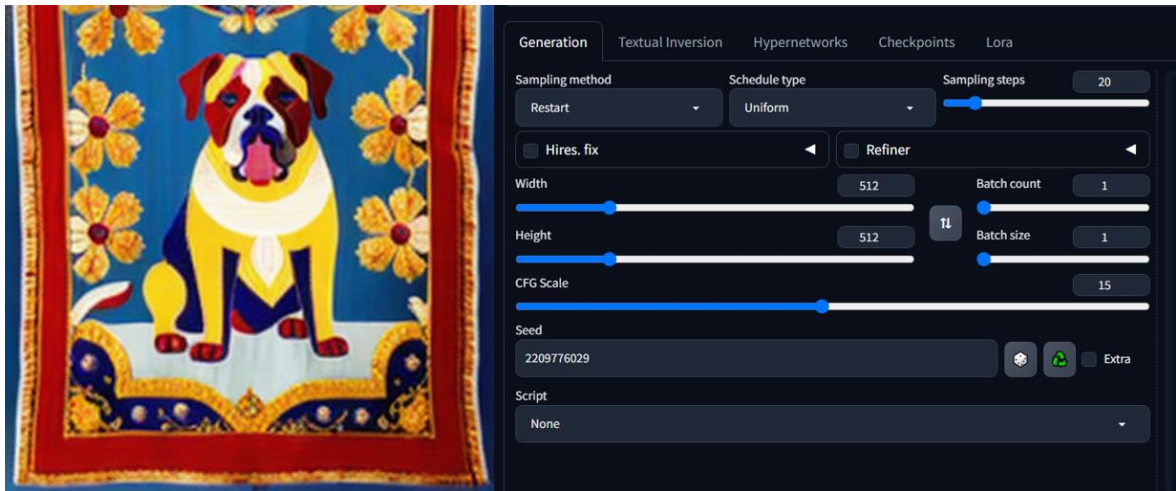


Figura 85 — Teste para *restart*

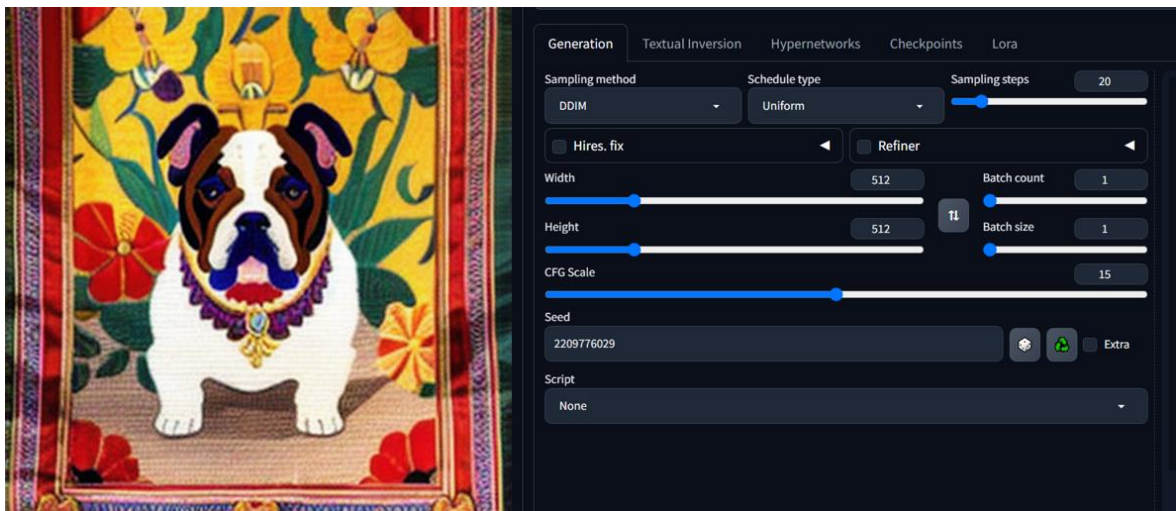


Figura 86 — Teste para *DDIM*

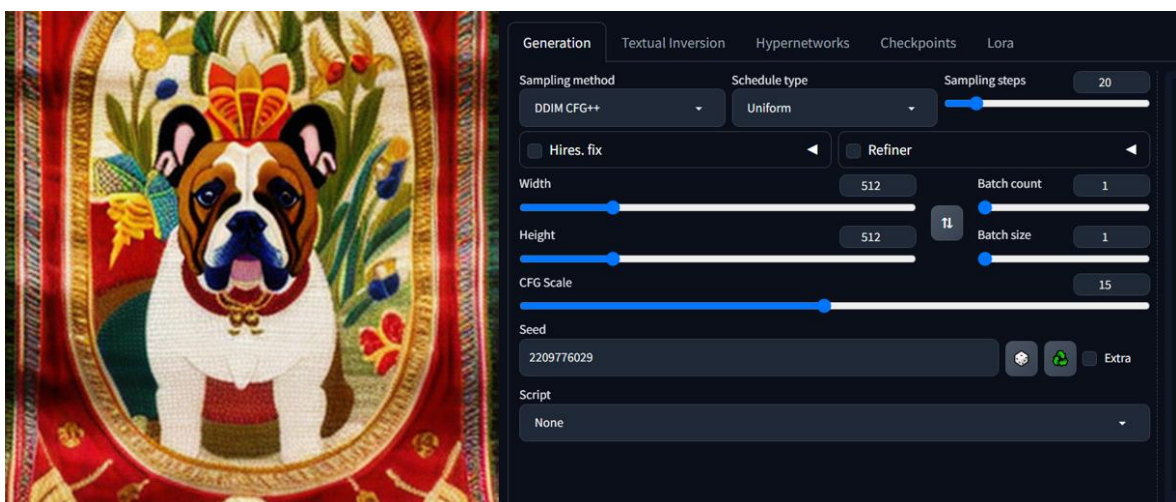
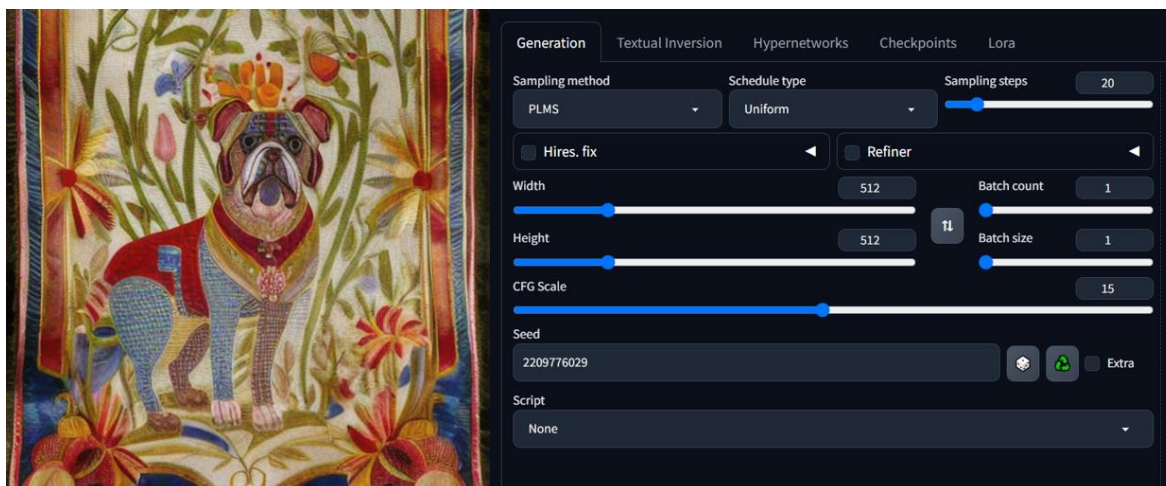
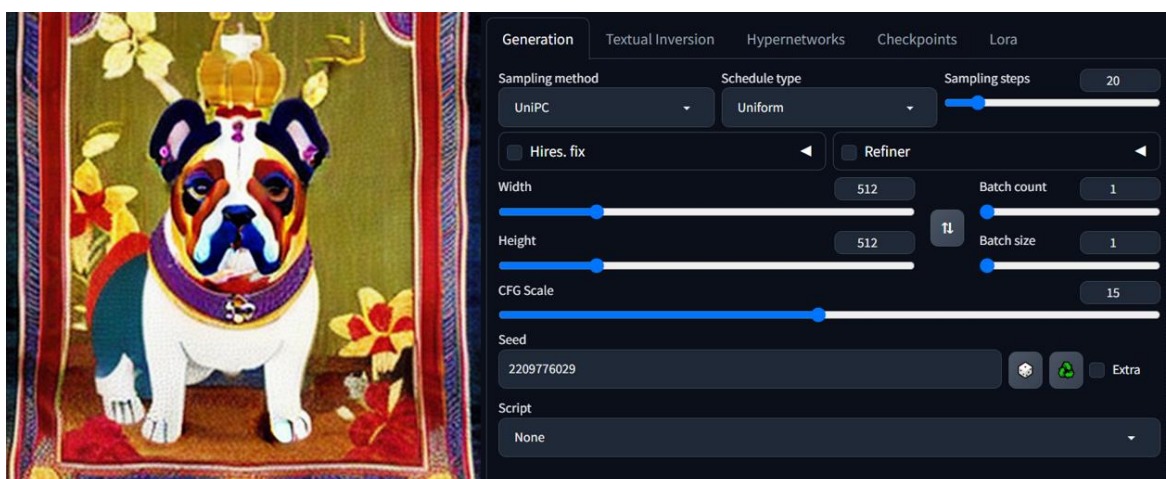
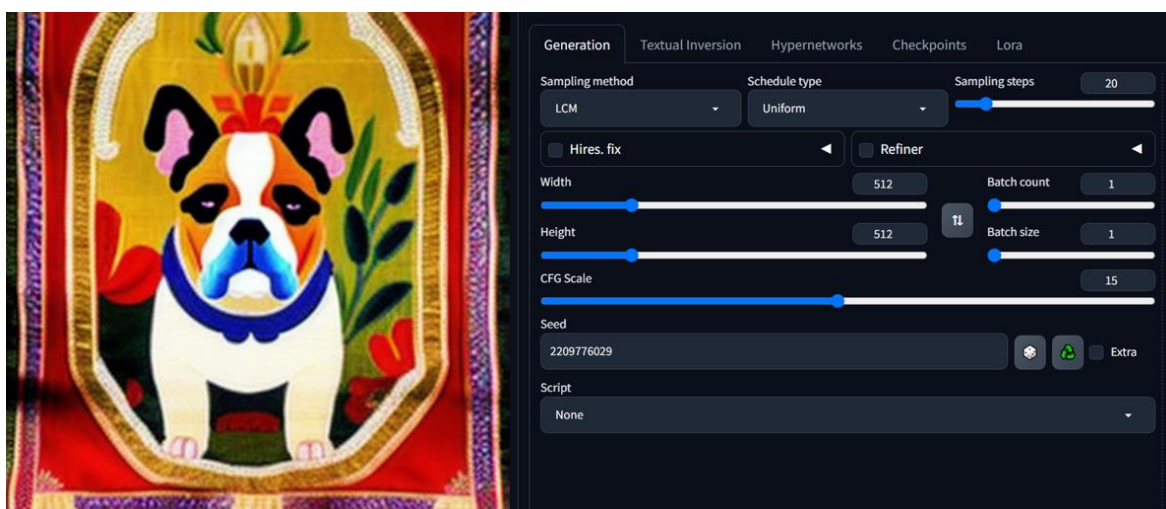


Figura 87 — Teste para *DDIM CFG ++*

Figura 88 — Teste para *PLMS*Figura 89 — Teste para *UniPC*Figura 90 — Teste para *LCM*

Ao analisar os outputs produzidos, foi possível identificar três principais categorias de imagens: as que apresentaram resultados positivos, aquelas com desempenho mediano e, por último, as que não atingiram resultados satisfatórios o suficiente.

A partir da análise dos resultados, identificamos que determinados parâmetros influenciam significativamente a qualidade das imagens geradas do estilo do bordado de Castelo Branco. Os parâmetros que produziram imagens de melhor qualidade e maior aderência ao estilo foram: *DPM++2M*, *DPM++2M SDE*, *DPM++3M SDE*, *LMS*, *DPM2*, *DPM2 a* e *DDIM CFG++*.

Por outro lado, os parâmetros considerados intermédios, geraram imagens com resultados satisfatórios, mas não tão positivos quanto os anteriores, incluem: *DDIM*, *restart*, *DPM fast*, *Heun*, *PLMS*, *UniPC*, *LCM*.

Finalmente, os parâmetros classificados como negativos, responsáveis por outputs menos fiéis ao estilo ou de qualidade inferior, foram: *DPM adaptive*, *Euler*, *Euler a*, *DPM++25 a* e *DPM++2M SDE Heun*.

Dentre os resultados mais positivos, destacam-se os obtidos com os parâmetros *DPM++2M* e *LMS*, cujos desempenhos superaram os demais *sampling_methods* avaliados. No caso do *DPM++2M*, a representação do bulldog é bastante fiel e detalhada, além de apresentar um uso de cores bastante positivo. Apesar da ausência do fundo branco tradicional, o resultado é visualmente equilibrado e bem executado. No entanto, observa-se uma presença limitada dos elementos característicos do bordado, bem como a ausência de simetria.

Já com o uso do *LMS*, os resultados foram ainda mais expressivos. A qualidade na representação do animal manteve o nível, havendo ainda um aumento significativo na fidelidade e na quantidade dos elementos típicos do bordado. Os detalhes apresentam uma qualidade superior e, em especial na região acima da cabeça do bulldog, nota-se uma leve simetria, que dá maior harmonia à imagem. Além disso, o uso da cor manteve-se muito positivo e, em comparação ao *DPM++2M*, mostrou-se superior em todos os aspetos, reforçando a excelência do resultado obtido.

No entanto, após os resultados positivos com o método *LMS*, decidimos testar num *batch* de 4 novas imagens com *seeds* aleatórias para determinar se o caso foi uma exceção ou se, realmente, é a escolha acertada. Na figura a baixo está o

resultado da geração de 3 imagens com exatamente os mesmos parâmetros e *prompts* para avaliar a consistência deste *sampling_method*.



Figura 91 — 4 Imagens geradas com *LMS* com *seeds* aleatórias

Como se pode observar na figura acima, o *sampling_method LMS* não produziu resultados positivos quando testado com uma *seed* aleatória, uma vez que, em nenhuma das quatro imagens, foi possível identificar claramente o animal solicitado. Apesar dessa limitação, é inegável a qualidade do uso da cor e a presença de diversos elementos característicos do bordado nas imagens geradas.

Para garantir uma comparação totalmente justa, a figura abaixo apresenta o mesmo pedido processado por *DPM++2M*.

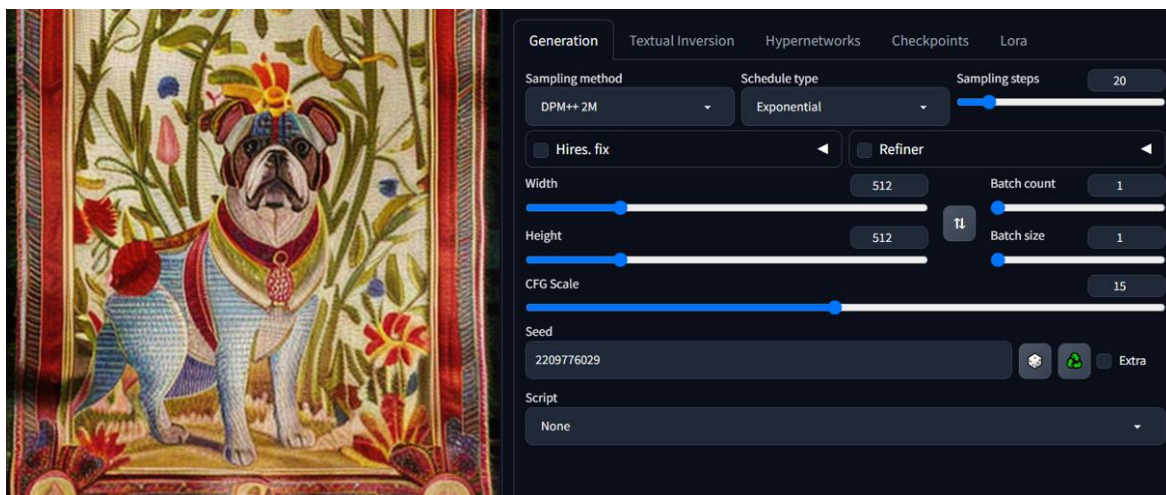
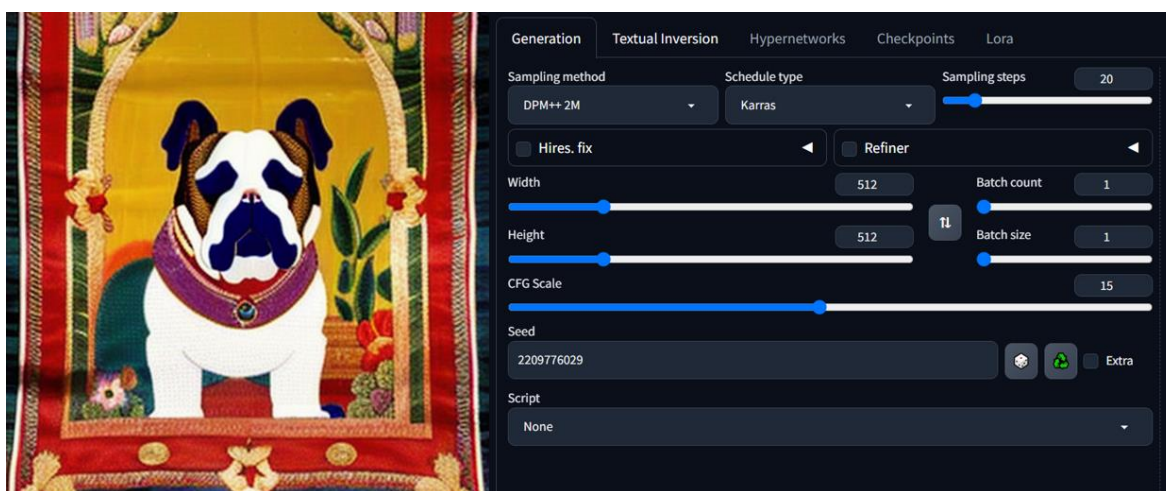
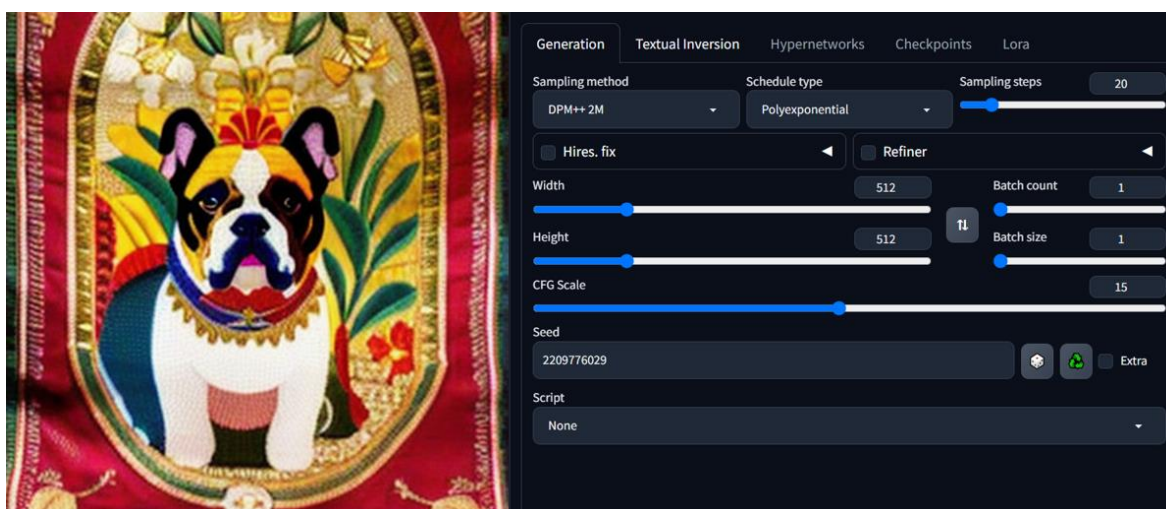


Figura 92 — 4 Imagens geradas com *DPM ++2M* com *seeds* aleatórias

Pela análise da última figura, é possível concluir que, embora a qualidade do bordado não alcance o nível obtido pelo método *LMS*, a inserção do animal na imagem demonstrou-se bastante positiva. Ao contrário dos resultados anteriores, aqui foi possível identificar claramente a presença do animal em três das quatro imagens, o que indica uma melhoria significativa na clareza e na representação do elemento principal no *prompt*.

9.3.2. Estudo do parâmetro *schedule type*

A próxima etapa consistirá na busca pelo *Schedule type* que maximize a qualidade das imagens geradas. Nos próximos testes, será utilizado o *sampling method DPM++2M* como referência, uma vez que este apresentou os melhores resultados até o momento.

Figura 93 — Teste com *Exponential*Figura 94 — Teste com *Karras*Figura 95 — Teste com *Polyexponential*

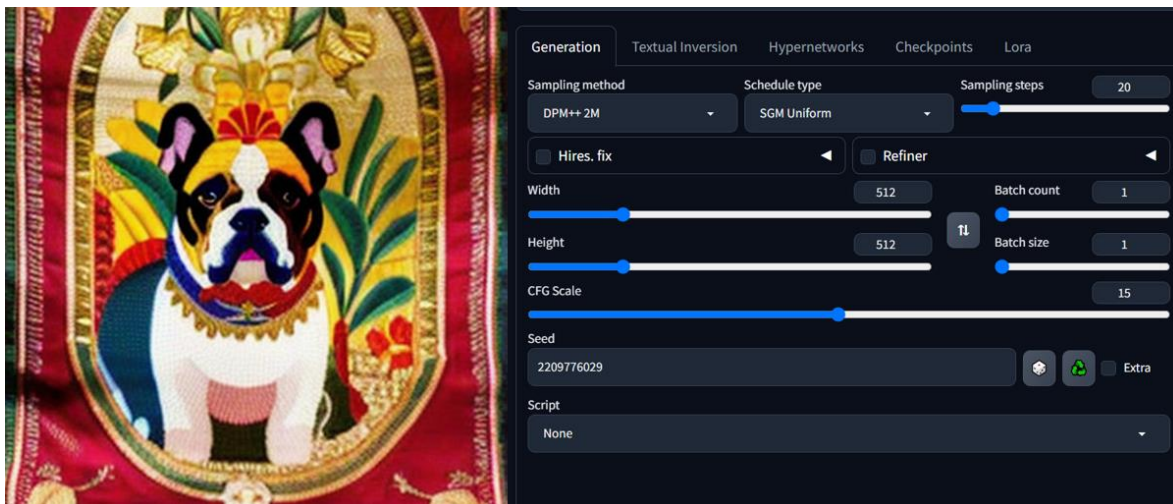


Figura 96 — Teste com *SGM Uniform*

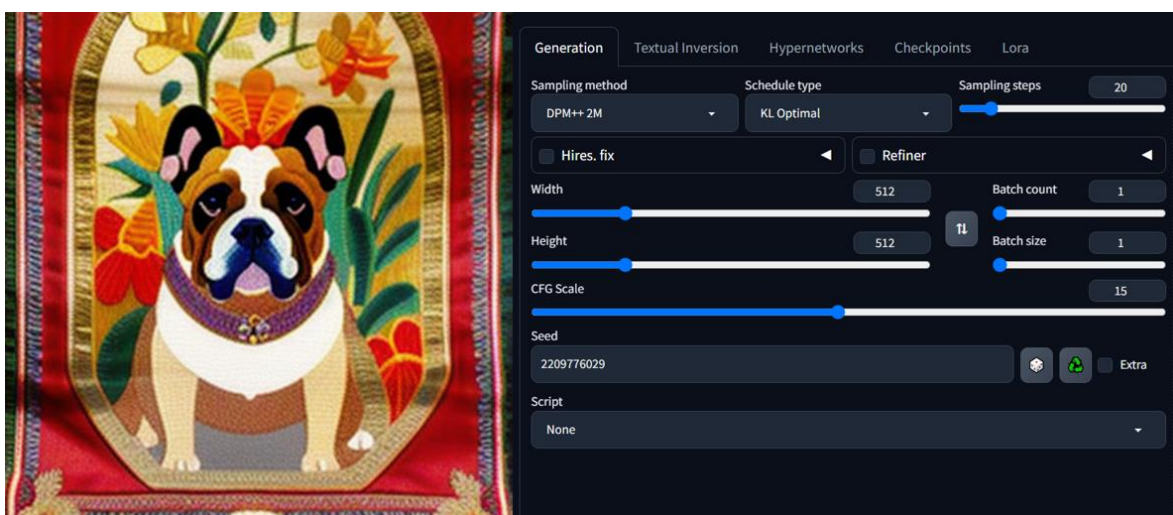


Figura 97 — Teste com *KL Optimal*

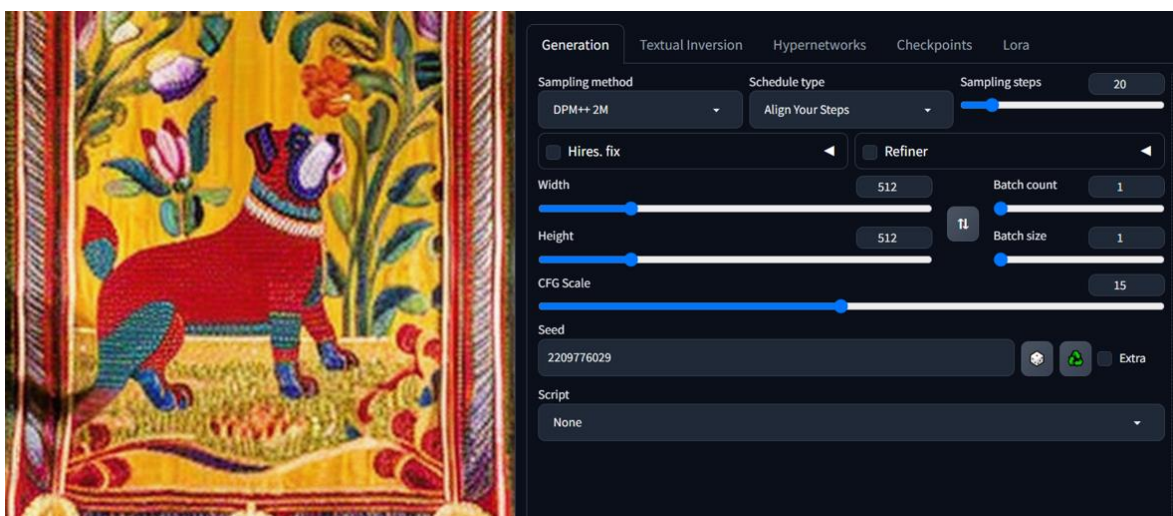
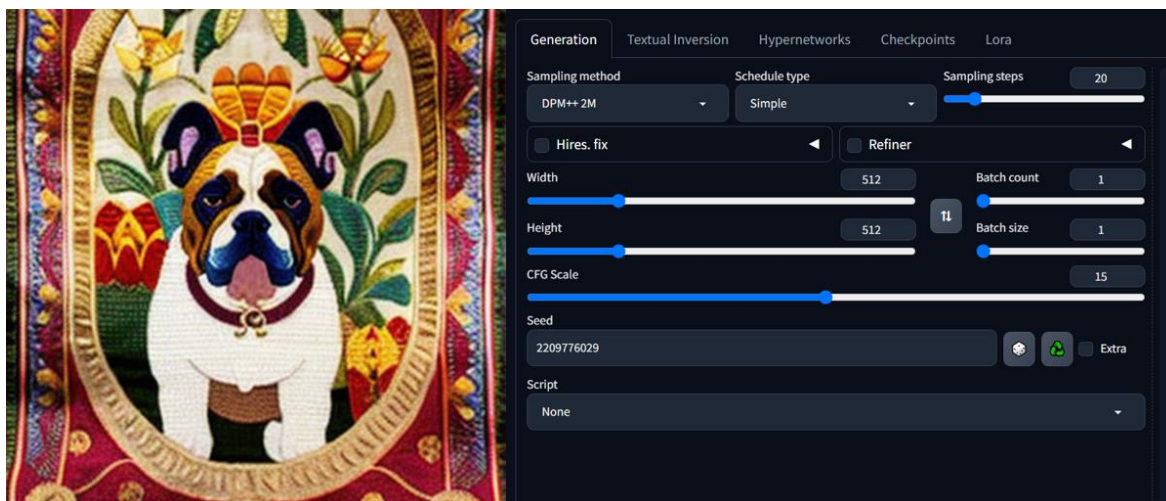
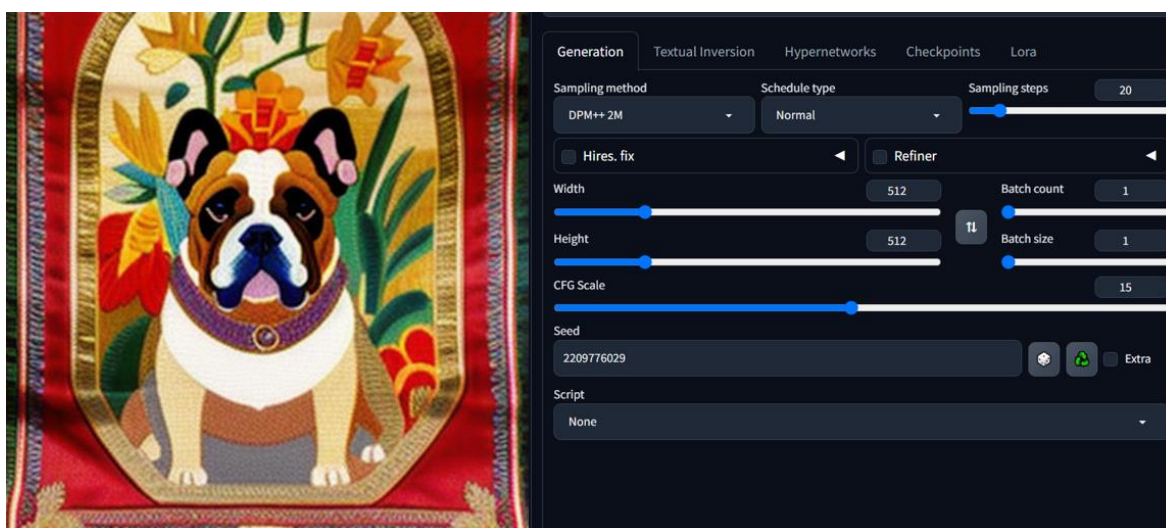
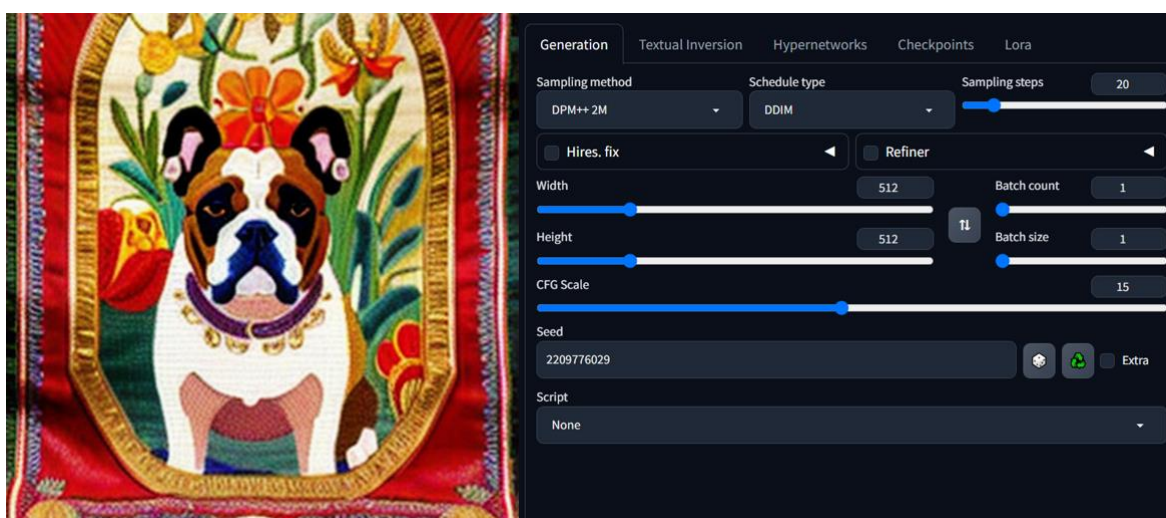


Figura 98 — Teste com *Align Your Steps*

Figura 99 — Teste com *Simple*Figura 100 — Teste com *Normal*Figura 101 — Teste com *DDIM*

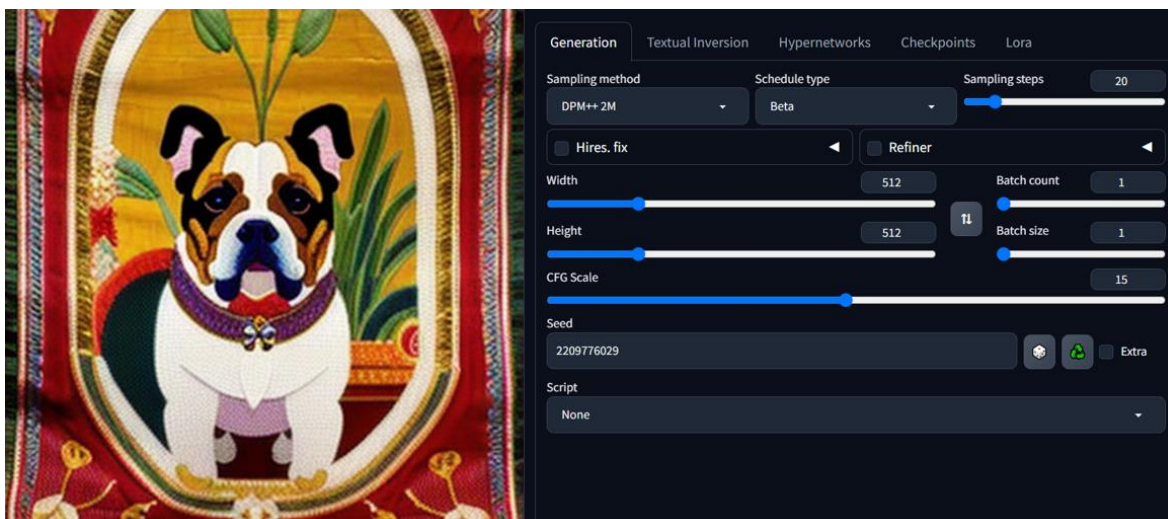


Figura 102 — Teste com *Beta*

Como na análise anterior, os resultados foram divididos em três grupos principais: resultados negativos, intermédios e positivos. Embora a diversidade entre os resultados não tenha sido muito grande, organizámos os métodos da seguinte forma: os resultados positivos foram obtidos com *exponential*, *kl optimal*, *simple*, *normal*, *DDIM* e *Beta*. Já os resultados intermédios corresponderam a *polyexponential* e *sgm uniform*. Por fim, os resultados negativos foram identificados utilizando o *karras* e *align your steps*.

Entre os resultados positivos, merecem destaque os métodos *kl optimal*, *simple* e *normal*, que se destacaram por produzirem *outputs* satisfatórios. Embora não apresentem diferenças significativas na representação dos símbolos típicos do bordado, esses métodos destacaram-se pelo uso da cor, contribuindo para uma qualidade visual superior.

Para garantir que tais resultados não sejam apenas exceções, será realizada uma nova rodada de testes, gerando mais quatro exemplos para cada um desses métodos e, adicionalmente, o *automatic*, para avaliar se a não seleção deste parâmetro melhora os resultados obtidos. Assim como nas análises anteriores, todos os exemplos serão produzidos a partir de uma *seed* aleatória, permitindo uma avaliação mais robusta e confiável dos resultados obtidos.



Figura 103 — 4 Imagens geradas com *kl optimal* com seeds aleatórias



Figura 104 — 4 Imagens geradas com *simple* com seeds aleatórias



Figura 105 — 4 Imagens geradas com *normal* com *seeds* aleatórias



Figura 106 — 4 Imagens geradas com *automatic* com *seeds* aleatórias

Ao analisar os outputs obtidos, concluímos que nenhuma das opções de *Schedule Type* apresentou a consistência desejada, ainda que tenham gerado algumas imagens satisfatórias. Em todos os grupos de imagens avaliados, existiu pelo menos uma imagem considerada bastante positiva, evidenciando que cada método possui pontos fortes e limitações distintas.

No caso do *KL Optimal*, os outputs destacam-se pelo alto nível de detalhe, mesmo em imagens de dimensões reduzidas (512 x 512 *pixels*). Contudo, observa-se um excesso de sobreposição entre os diferentes elementos, o que dificulta a distinção entre eles. O bulldog aparece claramente visível em apenas duas das quatro imagens, enquanto numa delas nota-se uma tentativa falhada de inserir o quadrúpede.

Quanto ao *Schedule Type simple*, a maioria dos resultados foi negativa: três das quatro imagens geradas não atingiram um padrão satisfatório. Embora alguns elementos individuais sejam claros, a composição geral das imagens acabou por não ser positiva. O animal está presente em todas as imagens, mas com níveis de qualidade bastante variados. Por exemplo, na segunda imagem (canto superior direito da Figura 104), o bulldog apresenta boa qualidade e detalhe, enquanto na terceira (canto inferior esquerdo da mesma figura), a presença do animal é muito menos perceptível.

No *Schedule Type normal*, os resultados também não se aproximaram do estilo do bordado, com destaque para o uso excessivo da cor vermelha. Além disso, verificou-se grande inconsistência na qualidade da representação do bulldog. Na primeira imagem (canto superior esquerdo da Figura 105), o animal é apresentado com bastante detalhe, mas nas demais houve dificuldade em torná-lo perceptível. Especialmente na terceira imagem (canto inferior esquerdo da Figura 105), não foi possível identificar qualquer elemento que lembrasse o animal solicitado.

Por fim, com o *Schedule Type Automatic*, os resultados apresentam algumas semelhanças com o tipo *normal*, especialmente no uso de tons vermelhos e sobreposição dos elementos. Apesar da alta densidade de detalhes, os motivos do bordado foram bem representados e nota-se até uma introdução de simetrias, o que confere um aspeto visual mais positivo às imagens. Contudo, o principal problema está na inconsistência da inclusão do bulldog onde apenas em uma das quatro imagens o animal é claramente visível, enquanto nas demais não houve sequer uma tentativa de representá-lo.

Da análise dos *schedule types*, conclui-se que não existe uma escolha completamente ideal. Dentre os resultados obtidos, o mais positivo foi o *Automatic*, que, embora tenha apresentado maior dificuldade na inclusão do bulldog do que o método *simple*, destacou-se por produzir imagens visualmente mais densas e ricas em detalhes. Acima de tudo, o método *Automatic* demonstrou maior coerência com o estilo e a estética característicos do bordado de Castelo Branco, aproximando-se mais do resultado desejado em termos de fidelidade ao padrão tradicional.

9.3.3. Estudo dos parâmetros *cfg_scale*

Para próxima etapa para estudo dos parâmetros do *A1111*, serão utilizadas as configurações mais eficientes identificadas até o momento (*Schedule type: automatic e sampling_method: DPM++2M*) na análise dos efeitos do *cfg_scale*. O objetivo é determinar os valores que maximizam a qualidade das imagens geradas, otimizando assim os parâmetros e *prompts* estabelecidos.

A análise irá começar pelo parâmetro *cfg_scale*, que determina diretamente o grau de aderência ao *prompt* textual. Este valor regula o equilíbrio entre a fidelidade ao *prompts* e a liberdade criativa do modelo. Este ajuste requer cuidado especial porque valores excessivamente baixos tendem a omitir elementos essenciais pedidos, enquanto valores elevados frequentemente introduzem artefactos visuais devido à rigidez na interpretação do *prompt*. Embora a escala permita valores entre 1 e 30, a faixa ideal situa-se tipicamente entre 7 e 15, onde se observa o melhor compromisso entre precisão e qualidade visual.

Na nossa metodologia de testes, vamos avaliar múltiplos valores em incrementos de 5, começando no valor mais baixo possível (1) até chegar ao máximo (30), permitindo uma análise comparativa do impacto deste parâmetro. Esta abordagem sistemática facilitará a identificação do valor ótimo que maximiza a eficácia dos *prompts* sem comprometer a integridade visual das imagens geradas.

Para os testes, iremos manter os *prompts* assim como a *seed* dos testes originais, de forma a permitir a comparação direta entre todos os resultados.



Figura 107 — Teste com *cfg_scale* = 1



Figura 108 — Teste com *cfg_scale* = 5



Figura 109 — Teste com *cfg_scale* = 10



Figura 110 — Teste com *cfg_scale* = 15



Figura 111 — Teste com *cfg_scale* = 20



Figura 112 — Teste com *cfg_scale* = 25



Figura 113 — Teste com *cfg_scale* = 30

Após análise dos resultados, concluímos que os valores de *cfg_scale* entre 5 e 15 produzem os outputs mais satisfatórios. Observando-se um padrão claro, quanto maior o valor de *cfg_scale*, maior a distância das imagens em relação ao estilo do bordado. Especificamente a partir do valor 20, o modelo começou a omitir progressivamente elementos característicos do mesmo e a comprometer a qualidade da representação do bulldog, atingindo o limite negativo no valor máximo (30), onde detalhes cruciais como os olhos do animal foram completamente excluídos.

Para determinar o valor específico que maximiza a qualidade dentro da faixa ideal (5-15), iremos gerar quatro outputs adicionais com *seed* aleatória, com foco nos seguintes valores: 7, 9, 11 e 13. Esta abordagem irá permitir uma comparação dos efeitos finos do parâmetro, identificando o ponto ótimo entre fidelidade ao *prompt* e qualidade.



Figura 114 — Teste com *cfg_scale* = 7 e *seed* aleatória



Figura 115 — Teste com *cfg_scale* = 9 e *seed* aleatória



Figura 116 — Teste com *cfg_scale* = 11 e *seed* aleatória



Figura 117 — Teste com *cfg_scale* = 13 e *seed* aleatória

A partir da análise dos resultados obtidos, concluímos que nenhum dos valores testados de *cfg_scale* pode ser considerado claramente superior. No entanto, destacaram-se, por motivos distintos, os valores 7 e 13, que apresentaram resultados inferiores aos demais na faixa considerada ideal.

Utilizando o valor 7, observou-se uma clara dificuldade na representação do animal solicitado, embora tenha havido uma leve inclusão dos elementos característicos do bordado. Por outro lado, com o valor 13, a representação do animal foi mais satisfatória, embora algumas dificuldades no uso das cores. Neste caso, os detalhes do bordado foram melhor destacados, sobretudo nos cantos da imagem, mas a presença de um losango na cor vermelha acabou por comprometer a aproximação ao estilo do bordado de Castelo Branco.

A partir da análise do parâmetro *cfg_scale*, concluímos que os valores entre 9 e 11 são os mais adequados para o estilo pretendido. Nesta faixa, é possível alcançar um equilíbrio ideal entre elementos característicos do bordado e a qualidade na representação dos *prompts* solicitados, garantindo imagens alinhadas com os objetivos do projeto.

9.3.4. Estudo do parâmetro *steps*

Por fim, para concluir o estudo dos parâmetros do *A1111*, vamos analisar como os parâmetros escolhidos reagem ao aumento e diminuição do parâmetro *steps*. Para o estudo do mesmo, vamos analisar em incrementos de 10 para analisar a qualidade das imagens geradas, assim como o tempo de geração de cada uma delas.

Como parâmetros para a geração destas imagens, iremos utilizar a *seed* original e, para o resto dos parâmetros, iremos utilizar aqueles que tiveram melhores resultados até então (*sampling_method: DPM ++2M*, *Schedule type: Automatic* e *cfg_scale: 10*). Os valores de *steps* testados serão em incrementos de 25 desde o valor mais baixo (1), até ao mais alto (150).

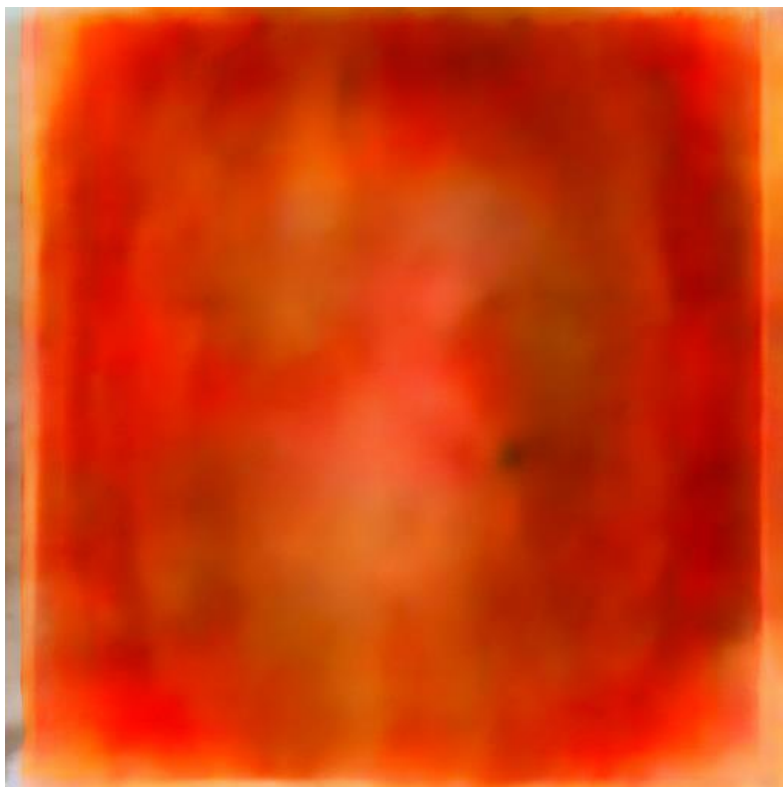


Figura 117 — Teste com *steps* = 1 – Tempo de geração = 3.3 sec



Figura 118 — Teste com *steps* = 25 – Tempo de geração = 34.7 sec



Figura 119 — Teste com *steps* = 50 – Tempo de geração = 1 min 7.7 sec



Figura 120 — Teste com *steps* = 75 – Tempo de geração = 1 min 40.5 sec



Figura 121 — Teste com *steps* = 100 – Tempo de geração = 2 min 20.1 sec



Figura 122— Teste com *steps* = 125 – Tempo de geração = 2 min 47.2 sec



Figura 123— Teste com *steps* = 150 – Tempo de geração = 3 min 25.7 sec

Ao analisar os resultados obtidos com a experimentação do parâmetro *steps*, concluímos que não compensa a utilização de valores superiores a 100, pois não se verificam melhorias relevantes na representação dos elementos da imagem. Especificamente, os valores de 125 e 150 *steps* não apresentam alterações que justifiquem o aumento no tempo de geração, que pode chegar a mais 1 minuto e 15 segundos em comparação com valores inferiores, sendo possível obter resultados semelhantes com 50 *steps* a menos.

Já com valores mais baixos, como 50 *steps*, a qualidade das imagens continua a ser satisfatória, embora possa haver alguma confusão entre os elementos presentes, como é possível ver na Figura 120, onde flores localizadas no canto inferior direito acabam por ser interpretadas como um vaso à medida que o número de *steps* aumenta. De qualquer forma, tendo em conta o objetivo de manter os tempos de geração curtos e garantir um equilíbrio estável entre qualidade e eficiência, optamos por utilizar o valor de 50 *steps* para as gerações *txt2img*, já que oferece uma boa relação entre desempenho visual e rapidez de processamento, sem comprometer significativamente a clareza dos elementos principais da imagem.

A experimentação dos parâmetros do *A1111* permitiu identificar combinações que maximizam a fidelidade ao estilo do Bordado de Castelo Branco, sem comprometer a qualidade visual ou a coerência dos elementos solicitados. Entre os *sampling methods*, o *DPM++2M* destacou-se pela consistência na representação do animal solicitado, enquanto o LMS apresentou resultados promissores em termos de detalhamento e uso da cor, mas com menor estabilidade com o variar da *seed*.

Por sua vez, a análise dos *schedule types* revelou que o *Automatic* oferece o melhor equilíbrio entre densidade de detalhes e fidelidade ao estilo, mesmo que a inclusão do animal nem sempre seja garantida com clareza.

Já na testagem dos parâmetros *cfg_scale* e *steps* demonstrou que valores intermédios (entre 9 e 11 para *cfg_scale* e até 100 para *steps*) proporcionam os melhores resultados, equilibrando a fidelidade ao *prompt*, qualidade visual e eficiência. Valores extremos destes parâmetros tendem a piorar a qualidade das imagens, seja por omissão de elementos essenciais, seja pela introdução de artefactos visuais ou aumento do tempo de geração. A abordagem sistemática adotada, com múltiplos testes e comparações diretas, permitiu isolar o impacto de cada parâmetro e seleccionar as configurações mais adequadas para o objetivo proposto.

9.4 Como os parâmetros influenciam o resultado em *img2img*

Numa abordagem *img2img*, o único novo parâmetro que tem influência no resultado e ainda não foi estudado é *denoising_strength*. Este parâmetro é o principal responsável por determinar o peso da imagem original, ou seja, controla o equilíbrio entre manter as características da imagem de entrada e introduzir as alterações sugeridas pelo *prompt*.

Quando o valor de *denoising_strength* é baixo, o resultado permanece muito próximo da imagem inicial, com poucas modificações. Por outro lado, valores mais elevados permitem que o modelo altere mais significativamente a imagem, tornando-a mais alinhada com o *prompt*, mas também aumentando o risco de perder detalhes originais essenciais. Assim, o estudo deste parâmetro é fundamental para obter o impacto desejado na transformação das imagens, garantindo um equilíbrio entre inovação e preservação do conteúdo original.

Devido à natureza das transformações *img2img* é impossível chegar a um valor perfeito, já que cada imagem tem o seu nível de detalhe, luminosidade e conteúdos

diferentes. De qualquer forma, para o estudo deste parâmetro iremos utilizar a foto da figura abaixo.



Figura 124 — Imagem base para estudo dos parâmetros *img2img*

9.4.1. Estudo do parâmetro *denoising_strength*

O parâmetro *denoising_strength* pode variar entre 0-1, para o estudo do melhor valor neste intervalo, vamos testar com incrementos 0,25 e analisar os resultados no *output* gerado. Os *prompts* para as gerações *img2img* estão na figura abaixo. Os restantes parâmetros serão os alcançados pelo estudo da vertente *txt2img*. Adicionámos também a opção *restore_faces* para garantir a qualidade quando as imagens têm rostos, que, apesar de não aplicável no exemplo estudado, pode ajudar na inclusão de outros elementos como os olhos.

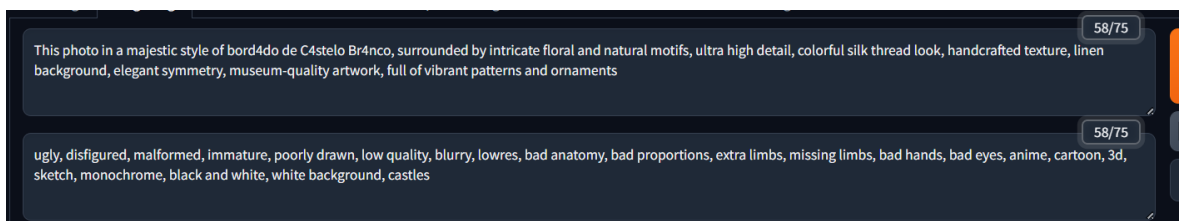


Figura 125 — Prompts para as gerações *img2img*



Figura 126 — Teste para *denoising_strength* = 0,25

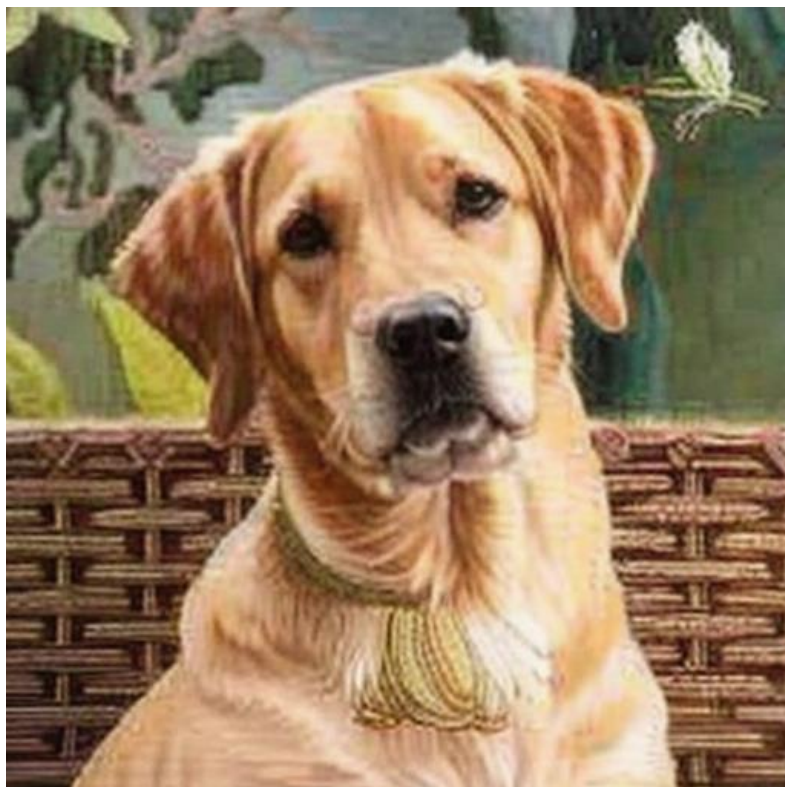


Figura 127 — Teste para *denoising_strength* = 0,50



Figura 128 — Teste para *denoising_strength* = 0,75



Figura 129 — Teste para *denoising_strength* = 1

Com base nos resultados obtidos, concluímos que os melhores valores para o parâmetro *denoising_strength* situam-se entre 0,50 e 0,75. Contudo, considerando que os outputs ainda não têm o nível de satisfação desejado, iremos aprofundar a exploração deste intervalo, procurando identificar valores que permitam um equilíbrio ideal entre os elementos originais da imagem e a incorporação dos motivos e padrões característicos do bordado de Castelo Branco. Desta forma, pretendemos obter resultados visualmente mais coerentes e fiéis ao estilo pretendido.

Nos próximos testes iremos alterar o valor de *denoising_strength* em incrementos menores, de 0.05, dentro do intervalo que melhor resultado gerou nos últimos testes.



Figura 130 — Teste para *denoising_strength* = 0,55

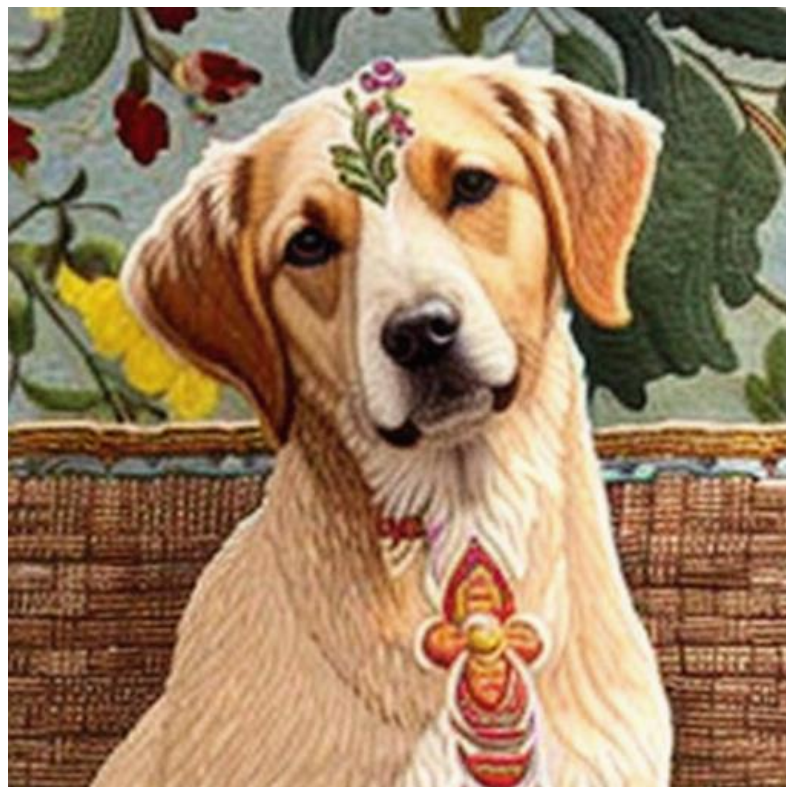


Figura 131 — Teste para *denoising_strength* = 0,60



Figura 132 — Teste para *denoising_strength* = 0,65



Figura 133 — Teste para *denoising_strength* = 0,70

Como foi possível verificar, os melhores valores para o parâmetro *denoising_strength* situam-se entre 0.60 e 0.65. Para finalizar o estudo deste parâmetro e determinar se um ponto intermédio oferece resultados melhores, para isso, iremos realizar um último teste utilizando o valor de 0.625. Desta forma, pretendemos avaliar se este equilíbrio entre os dois extremos permite obter um *output* superior, que harmonize de forma mais eficaz a preservação dos elementos da imagem original com a incorporação do estilo do bordado de Castelo Branco.



Figura 134 — Teste para *denoising_strength* = 0,625

Concluindo o estudo do parâmetro *denoising_strength*, consideramos que o valor mais adequado a ser utilizado é 0,60. Este foi o único que permitiu alcançar um equilíbrio satisfatório entre a inclusão do animal solicitado e a presença do estilo característico do bordado de Castelo Branco. Com este ajuste, é possível de forma satisfatória tanto os elementos da imagem original quanto os padrões e motivos do bordado, resultando numa composição visualmente positiva.

10. Exemplos gerados com os novos modelos

Neste capítulo, procuramos verificar a versatilidade e a robustez do modelo, aliadas aos efeitos dos novos parâmetros otimizados, por meio da geração de uma série de imagens que vão além do teste padrão com o bulldog. Ao invés de restringir a avaliação a um único *prompt*, decidimos explorar diferentes temas e solicitações, a fim de testar a capacidade do modelo em responder a contextos mais avançados e complexos.

O objetivo central é demonstrar como o modelo se comporta diante de novos desafios, avaliando não apenas a fidelidade à representação dos elementos solicitados, mas também a qualidade dos bordados gerados em novas situações. Desta forma, pretendemos confirmar se a combinação dos parâmetros selecionados é capaz de manter a excelência visual e a coerência estilística do bordado de Castelo Branco, mesmo quando os *prompts* são variados e exigentes.

Os testes realizados vão incidir sobre ambas as possibilidades de geração pretendendo avaliar a capacidade do modelo num contexto *txt2img* e também *img2img*.

10.1 Exemplos *txt2img*

Nesta secção, iremos apresentar os resultados obtidos através da vertente *txt2img*, utilizando diversos *prompts* distintos organizados em grupos de quatro imagens cada. Esta abordagem permite avaliar de forma mais robusta a consistência e a qualidade das imagens geradas pelo modelo. Desta forma, conseguimos obter uma visão mais completa e fiável acerca do desempenho do modelo.

Os *prompts* serão divididos em três grupos temáticos principais. O primeiro grupo focar-se-á em animais, com o objetivo de avaliar a capacidade do modelo em adaptar diferentes espécies ao estilo do bordado de Castelo Branco.

O segundo grupo será dedicado a objetos do quotidiano, permitindo testar a habilidade do modelo em incorporar elementos mais simples, mas de difícil interpretação no estilo do bordado.

Por fim, o terceiro grupo abordará elementos da natureza, com especial atenção para flores e plantas, dada a riqueza tradicional dos motivos florais no bordado de Castelo Branco.

Esta seleção de temas permitirá avaliar a capacidade do modelo em responder a *prompts* textuais variados, gerando imagens que respeitam e valorizam as suas características únicas.

No primeiro grupo, iremos realizar a geração de imagens para cada um dos quatro animais diferentes: um cavalo, um peixe, uma borboleta e uma águia. Assim iremos conseguir uma boa representação de animais, desde terrestres, aquáticos e aéreos.

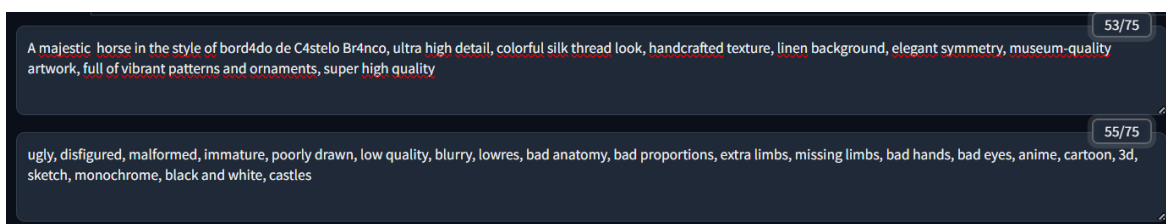


Figura 135 — *Prompts* para a geração de imagens com um cavalo



Figura 136 — Resultados da criação de imagens com um cavalo

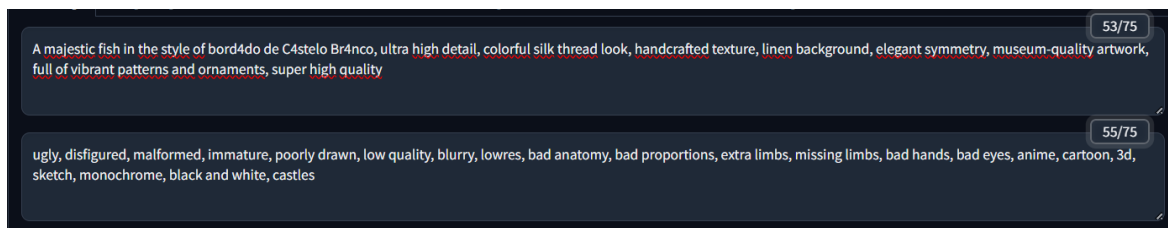


Figura 137 — Prompts para a geração de imagens com um peixe



Figura 138 — Resultados da criação de imagens com um peixe

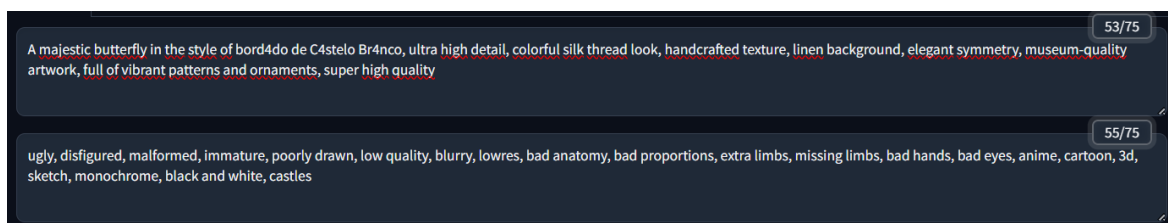


Figura 139 — Prompts para a geração de imagens com uma borboleta



Figura 140 — Resultados da criação de imagens com uma borboleta

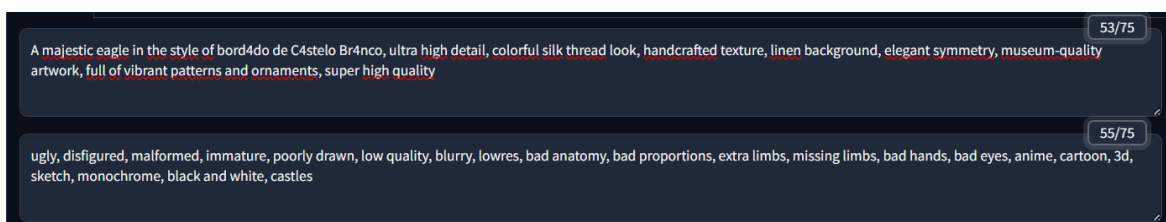


Figura 141 — Prompts para a geração de imagens com uma águia



Figura 142 — Resultados da criação de imagens com uma águia

O segundo grupo, dedicado a objetos do quotidiano, será composto por quatro conjuntos distintos de imagens, cada um deles representando um objeto do dia-a-dia. Para isso, serão geradas imagens de uma cadeira, um lápis, uma tesoura e um teclado, permitindo avaliar a capacidade do modelo em adaptar elementos simples ao estilo do bordado de Castelo Branco.

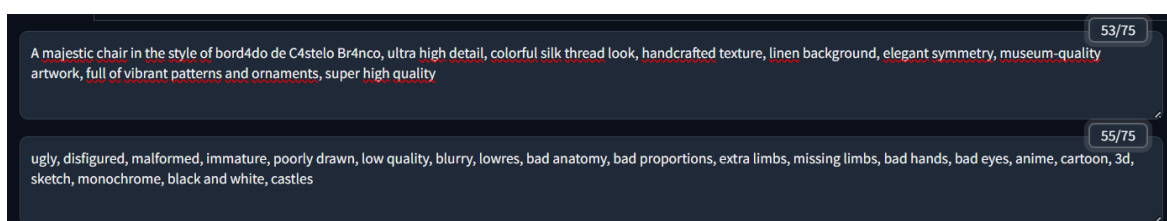


Figura 143 — Prompts para a geração de imagens com uma cadeira



Figura 144 — Resultados da criação de imagens com uma cadeira

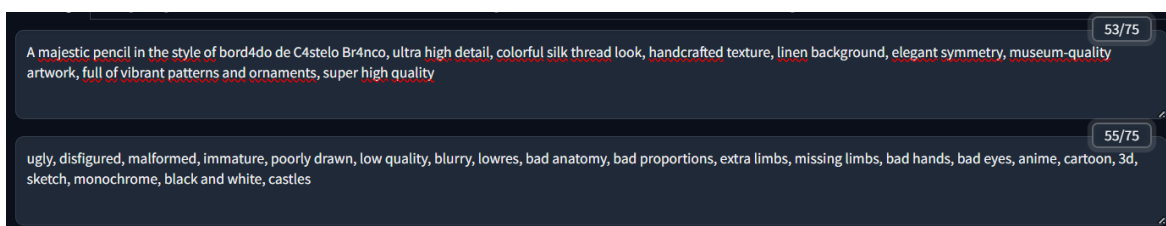


Figura 145 — Prompts para a geração de imagens com um lápis



Figura 146 — Resultados da criação de imagens com um lápis

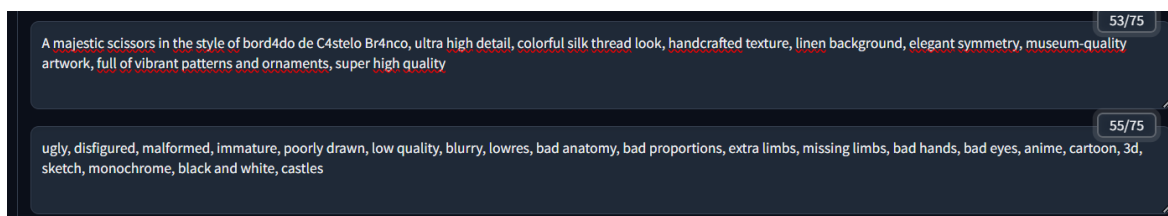


Figura 147 — Prompts para a geração de imagens com uma tesoura



Figura 148 — Resultados da criação de imagens com uma tesoura

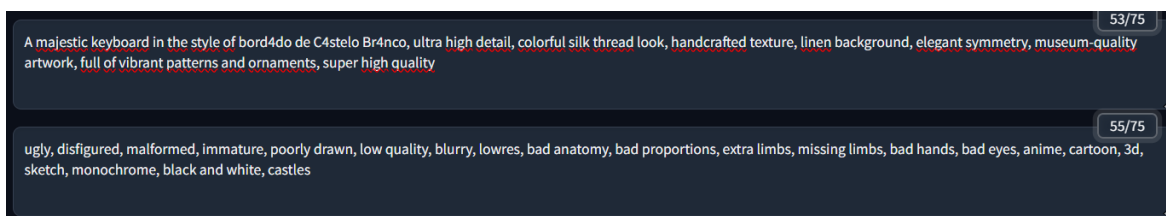


Figura 149 — Prompts para a geração de imagens com um teclado



Figura 150 — Resultados da criação de imagens com um teclado

O terceiro grupo, dedicado a objetos da natureza, será composto por quatro conjuntos distintos de imagens, cada um deles representando um objeto da natureza. Para isso, serão geradas imagens de uma rosa, uma pedra, uma árvore e uma paisagem.

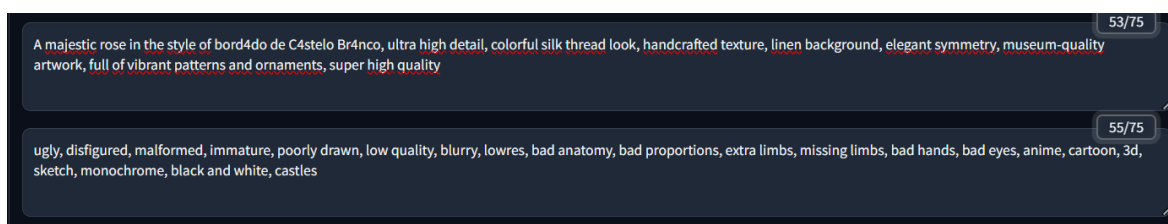


Figura 151 — Prompts para a geração de imagens com uma rosa



Figura 152 — Resultados da criação de imagens com uma rosa

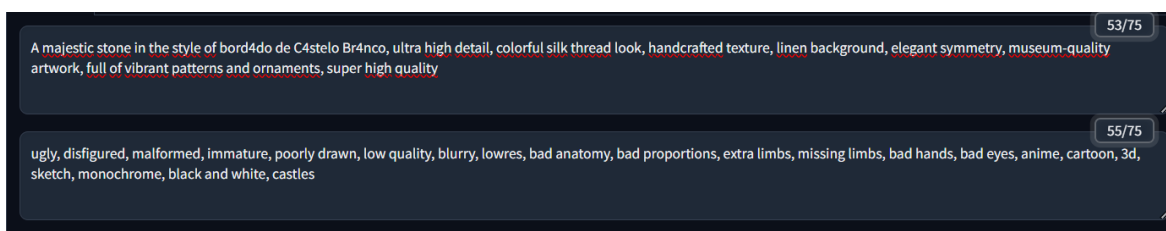


Figura 153 — Prompts para a geração de imagens com uma pedra



Figura 154 — Resultados da criação de imagens com uma pedra

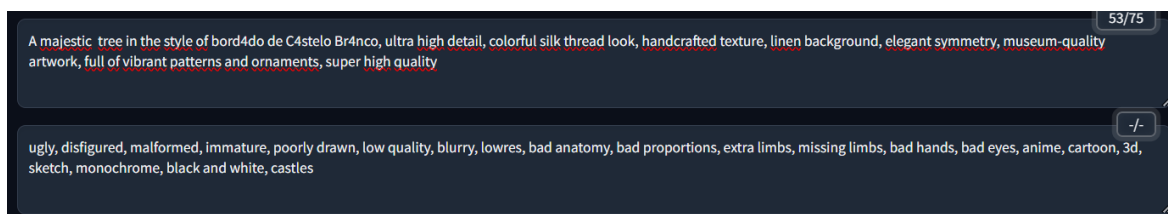


Figura 155 — Prompts para a geração de imagens com uma árvore

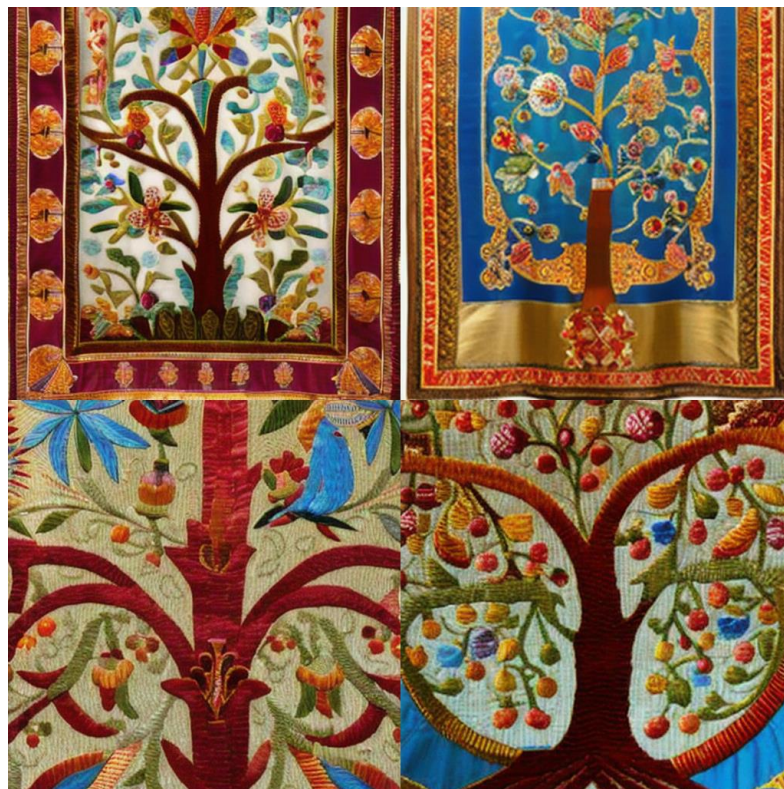


Figura 156 — Resultados da criação de imagens com uma árvore

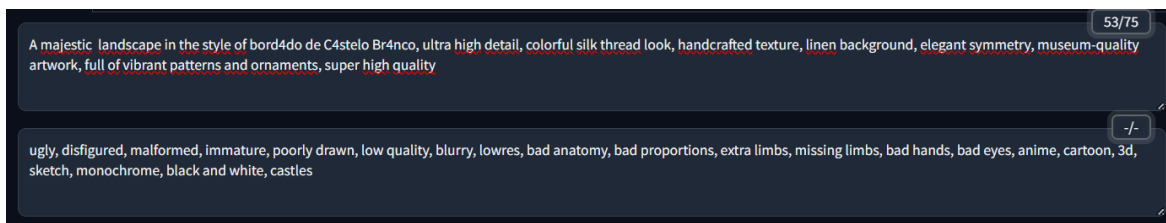


Figura 157 — Prompts para a geração de imagens com uma paisagem



Figura 158 — Resultados da criação de imagens com uma paisagem

O primeiro grupo analisado, relativo à representação de animais, foi o que mais se destacou, conseguindo, de forma geral, reproduzir de maneira positiva e reconhecível os animais solicitados. Os resultados foram particularmente satisfatórios para o cavalo e borboleta, que aparecem em praticamente todas as imagens geradas de forma irrepreensível, mantendo as características do estilo do bordado de Castelo Branco. No caso do peixe, como era de esperar, o modelo enfrentou algumas dificuldades em duas das quatro imagens geradas não foi possível identificar claramente o animal, e mesmo nas restantes duas, a sua presença não é tão evidente como nos outros exemplos, o que demonstra que elementos menos familiares ao padrão tradicional do bordado podem ser mais desafiadores para o modelo. Já no que diz respeito à águia, o desempenho foi igualmente positivo, sendo bastante evidente a inclusão do animal nas imagens, embora, numa delas, o bico tenha ficado próximo da asa.

O grupo dos objetos do quotidiano foi o que apresentou resultados menos positivos. Apesar de as imagens geradas manterem um nível de qualidade elevado, já não é possível identificar claramente os objetos solicitados nos prompts. No entanto, observam-se semelhanças longínquas em alguns casos, por exemplo, no output gerado com o objetivo de incluir um lápis, os caules de algumas flores foram claramente reforçados, sugerindo uma tentativa de representar um objeto visualmente semelhante a um lápis. No caso do teclado, é possível notar um reforço das linhas, o que aproxima a figura de um retângulo, embora não seja suficiente

para uma identificação clara. A tentativa de representar a tesoura também é visível na forma como as folhas aparecem no caule de uma das flores. Já quando se pediu ao modelo para gerar a imagem de uma cadeira, as dificuldades foram ainda maiores, e desta vez não foi possível identificar qualquer elemento que remetesse para o objeto pedido.

Por fim, o grupo com foco em elementos da natureza apresentou um desempenho intermédio. O ponto alto foi, sem dúvida, a representação da rosa, cujas quatro imagens geradas foram consideradas bastante satisfatórias, resultado esperado dada a proximidade deste elemento com os padrões tradicionalmente presentes no bordado. No caso da árvore, o resultado foi igualmente positivo, sendo a árvore facilmente reconhecível e bem integrada nos elementos do bordado. Por outro lado, os resultados foram menos positivos para a pedra e a paisagem. A ambiguidade destes conceitos criou dificuldades evidentes ao modelo, que não conseguiu gerar imagens claras ou reconhecíveis destes elementos, o que sugere que a especificidade do *prompt* têm um impacto significativo na qualidade do resultado.

Da análise das imagens geradas, conclui-se que, apesar das melhorias significativas obtidas, ainda existem desafios, especialmente quando os *prompts* são ambíguos ou distantes ao estilo do bordado. O principal ponto forte das imagens produzidas está na sua qualidade e equilíbrio visual, sendo possível distinguir com clareza a maioria dos elementos, graças à inclusão de simetrias e padrões florais característicos do bordado de Castelo Branco. No entanto, a representação de objetos e elementos menos tradicionais continua a ser um desafio, indicando que o modelo beneficia de *prompts* mais específicos.

10.2 Exemplos *img2img*

Neste subcapítulo, apresentaremos um conjunto de resultados obtidos através da abordagem *img2img*. Para assegurar uma avaliação do desempenho do modelo, optámos por utilizar fotografias reais tiradas por um dos autores, o que permitirá observar como o sistema responde a desafios comuns deste tipo de imagens. Entre estes desafios incluem-se imperfeições como falta de foco, variações de cor provocadas pela iluminação e a presença de elementos estranhos ao bordado de Castelo Branco. Desta forma, será possível analisar a robustez do modelo perante situações práticas e avaliar a sua capacidade de adaptação a diferentes contextos visuais.

Nas figuras apresentadas abaixo, é possível visualizar os parâmetros e *prompts* selecionados para as demonstrações. Estes foram escolhidos por terem gerado os melhores resultados ao longo da fase de testes.

Para cada imagem base serão geradas quatro novas imagens, para, tal como nas abordagens *txt2img*, melhor visualizar a consistência do modelo. Para também simular o funcionamento da aplicação, iremos realizar cada teste 3 vezes, para diferentes valores de *steps* desde 20, 35 e 50, simulando diferentes escolhas dentro do protótipo.

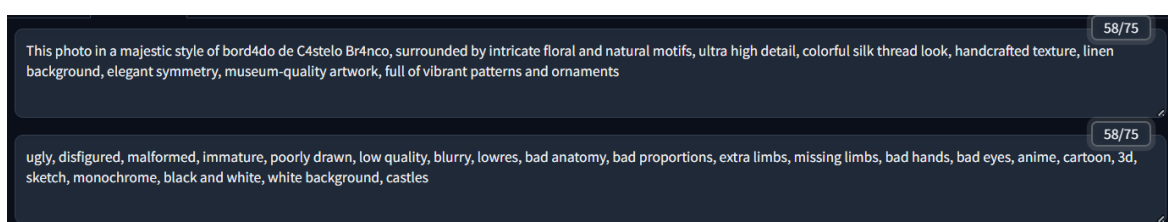


Figura 159 — Prompts para as gerações *img2img*



Figura 160 — Imagem base para o primeiro teste

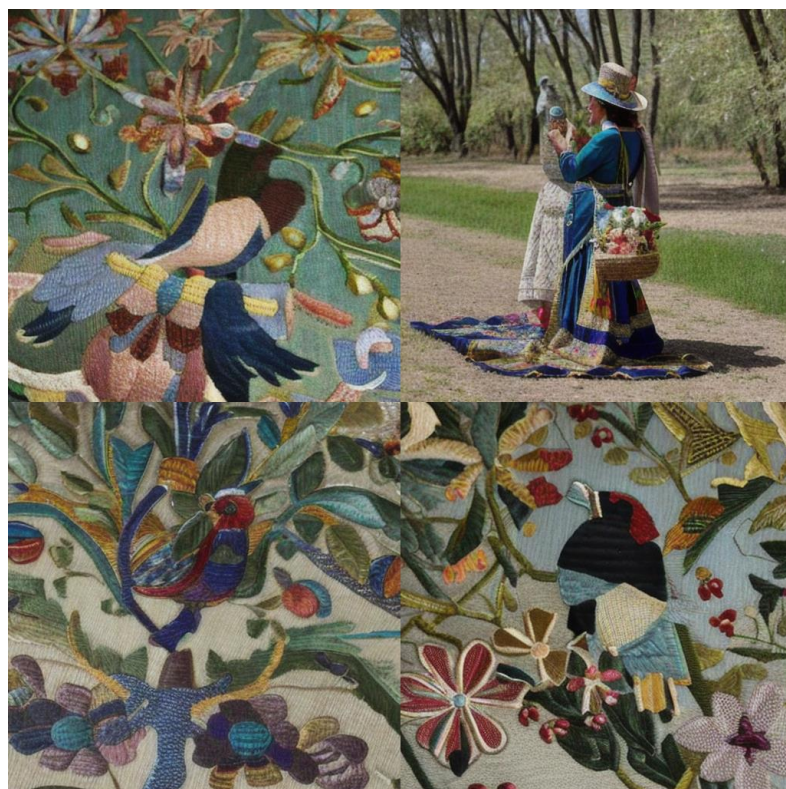


Figura 161 — *Output* gerado para *steps* = 20

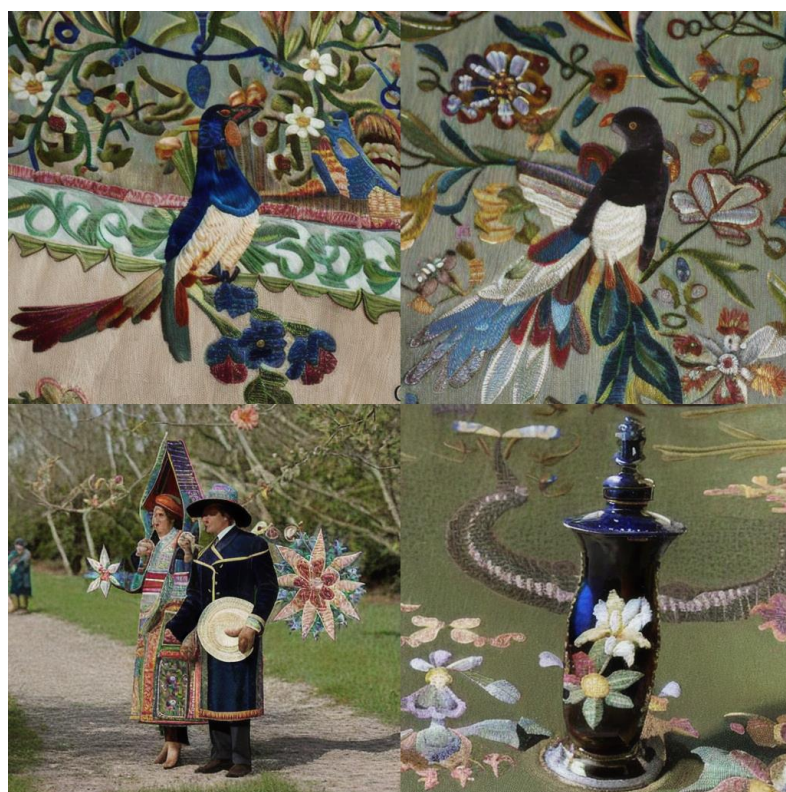


Figura 162 — *Output* gerado para *steps* = 35

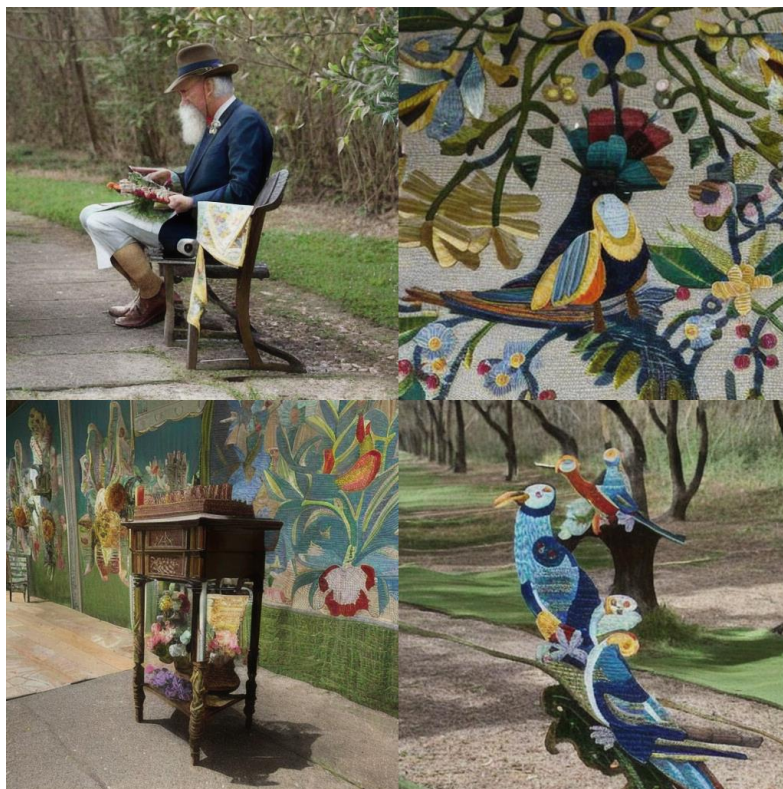


Figura 163 — *Output* gerado para *steps* = 50



Figura 164 — Imagem base para o segundo teste



Figura 165 — *Output* gerado para *steps* = 20



Figura 166 — *Output* gerado para *steps* = 35



Figura 167 — Output gerado para steps = 50



Figura 168 — Imagem base para o terceiro projeto



Figura 169 — *Output* gerado para *steps* = 20



Figura 170 — *Output* gerado para *steps* = 35



Figura 171 — *Output* gerado para *steps* = 50

A principal conclusão que se retira dos resultados obtidos é que o modelo ainda não consegue transferir de forma eficaz o estilo do bordado de Castelo Branco para fotografias reais. A presença de elementos estranhos ao universo do bordado como edifícios, carros e motas revela-se um desafio significativo, impedindo o modelo de realizar uma integração equilibrada entre os motivos tradicionais do bordado e os elementos presentes nas imagens originais.

No primeiro exemplo, o modelo conseguiu distinguir o elemento principal da fotografia do respetivo fundo. No entanto, ao tentar aplicar o estilo do bordado, não foi capaz de preservar as características fundamentais do objeto principal. Em quatro das imagens geradas, o modelo adotou uma abordagem excessivamente realista, transformando a mota em elementos completamente distintos, tais como uma cadeira, um senhor de idade sentado numa cadeira e até um móvel semelhante a uma mesa de cabeceira. Este comportamento evidencia a dificuldade do modelo em interpretar e adaptar objetos complexos ao estilo pretendido.

No segundo exemplo, testámos a capacidade do modelo em identificar um animal presente numa imagem com condições de luminosidade desafiante e com falta de foco. Neste caso, o modelo não conseguiu distinguir adequadamente os elementos, resultando em outputs negativos. As imagens geradas apresentaram uma predominância da cor branca, herdada da imagem original, mas sem a presença clara do cão. Em alguns casos, o modelo chegou a gerar imagens que

lembram loiça decorada com motivos inspirados no bordado de Castelo Branco, mas sem qualquer relação com o animal presente no *input*.

Por fim, na última imagem, o objetivo era avaliar a transformação de uma paisagem com múltiplos elementos distintos. A imagem original continha um palco e uma carrinha, elementos que não foram devidamente mantidos nos outputs gerados. O palco foi frequentemente convertido num edifício, com variações significativas no estilo arquitetónico. Quanto à carrinha, o modelo não a identificou em nenhuma das imagens, estando totalmente ausente nos resultados gerados. Em apenas duas das doze imagens geradas, observou-se uma tentativa de aplicar o estilo do bordado de Castelo Branco, mas mesmo nestes casos, o resultado ficou muito distante do *input* original.

Estes exemplos evidenciam que as capacidades *img2img* do modelo ainda não estão totalmente preparadas para transformar imagens do quotidiano no estilo do bordado de Castelo Branco, sobretudo quando confrontadas com elementos complexos ou ambientes pouco convencionais.

11. Criação de um protótipo para o BordadoBOT

Um dos principais objetivos deste projeto envolvia a criação de uma aplicação protótipo que utilizasse os modelos treinados para permitir aos utilizadores gerar imagens no estilo do Bordado de Castelo Branco de forma intuitiva e acessível. Esta aplicação, foi concebida para ser multiplataforma, o que garante uma ampla acessibilidade a diferentes tipos de utilizadores, independentemente do sistema operativo ou do dispositivo utilizado.

A aplicação foi pensada para simplificar o acesso dos utilizadores às funcionalidades avançadas de geração e estilização de imagens, normalmente associadas a interfaces complexas e exigências técnicas elevadas. Através de uma interface gráfica simples e intuitiva, os utilizadores podem facilmente inserir *prompts* descritivos, selecionar o modelo desejado e visualizar os resultados das suas criações. Além disso, a integração dos modelos treinados com Stable Diffusion foi realizada através de uma *API* remota, o que permite aliviar as limitações de *hardware* dos dispositivos móveis.

11.1 Tecnologias Escolhidas

A escolha do *Dart/Flutter* como linguagem para a construção do protótipo vem essencialmente pela familiaridade com a ferramenta por já termos tido um contacto inicial na unidade curricular de Informática Móvel.

Esta é uma ferramenta multiplataforma que permite o desenvolvimento tanto para sistemas operativos de computador (*Windows*, *Linux* e *MacOS*), quanto para dispositivos móveis (*Android* e *iOS*) assim como apps web nativas. *Dart* é uma linguagem de programação, criada pela *Google*, que dá prioridade ao desempenho, segurança e produtividade enquanto juntamente com o *Flutter*, *framework* mantida pela mesma empresa, tem um foco na construção de *UI's* altamente personalizáveis e responsivas, permitindo assim o desenvolvimento de aplicações nativas para várias plataformas que apenas se baseiam numa base de código, sendo o seu principal lema “*Build apps for any screen*”.

Torna-se assim uma excelente opção para a construção de um protótipo porque oferece um desenvolvimento multiplataforma com alto desempenho, permitindo ver mudanças e alterações rápidas nos simuladores com o recurso *Hot Reload* e *Hot Restart*, estas funcionalidades aceleram a velocidade de desenvolvimento e promovem a experimentação. A escolha do *Dart/Flutter* foi uma forma de expandir horizontes através do uso de uma tecnologia da qual apenas tivemos uma

introdução teórica, mas também pelas inúmeras vantagens que oferece em campos como a produtividade, desempenho e versatilidade.

11.2 Arquitetura

O nosso protótipo para o BordadoBot baseia-se numa aplicação capaz de fazer e receber pedidos com um servidor a correr a interface *Automatic1111*, no nosso caso é um computador portátil que recebe e gere os pedidos JSON feitos pela aplicação usando HTTP para enviar instruções e receber imagens geradas. O *Automatic1111* fornece uma *API REST* caso seja executado no modo *api*, e que, juntamente com o modo *listen*, vai esperar por pedidos na porta 7860.

Estes pedidos permitem a configuração de parâmetros como o número de *steps*, resolução da imagem, modelo a utilizar, entre outros, e o computador responde com uma imagem gerada que é codificada em *Base64*. Este é um método de codificação que converte dados binários, no nosso caso imagens, para texto, ou seja, em vez de enviar uma imagem como um arquivo (.jpg ou .png), vai enviar o seu conteúdo binário da imagem transformado para uma *string* de texto para facilitar o transporte nas mensagens HTTP.

Esta arquitetura de comunicação tem como principais vantagens a facilidade de configuração pelas chamadas serem feitas com HTTP simples, sem a necessidade de sistemas complexos para os pedidos pois tal sistema já vem embutido na instalação base do *A1111*. Por outro lado, na nossa implementação a principal desvantagem é a necessidade do servidor *A1111* estar na mesma rede que o dispositivo a correr o protótipo, não sendo possível fazer pedidos em redes diferentes. Outro ponto relevante é a incapacidade de cancelar pedidos via HTTP, o que, em casos mais extremos, pode levar a que o utilizador fique demasiado tempo à espera pela sua imagem caso haja imprevistos ou alguma falha do lado do *A1111*.

11.3 Explicação das Interfaces e uso do protótipo

Para garantir a simplicidade e funcionalidade do protótipo, é fundamental que as interfaces sejam intuitivas e acessíveis. O objetivo do BordadoBOT é servir uma grande e diversificada audiência, incluindo utilizadores sem experiência técnica em inteligência artificial. Por este motivo, o design da interface foi concebido para minimizar a complexidade e reduzindo o número de passos necessários para gerar uma imagem no estilo do Bordado de Castelo Branco.

A navegação na aplicação foi pensada para ser direta. O utilizador inicia o processo selecionando o modo desejado através de uma *AppBar*, onde terá duas opções: “Texto → Imagem” e “Imagem → Imagem” (Figura 172). Logo a baixo da seleção do modo o utilizador terá, em ambas as opções, a possibilidade de escolher qual modelo vai usar para a criação da imagem tendo como opções disponíveis: `modelo_original`, `modelo_melhores` e `modelo_styles` juntamente de uma breve descrição dos mesmos como é possível ver na Figura 173.



Figura 172 — *AppBar* para a seleção do modo

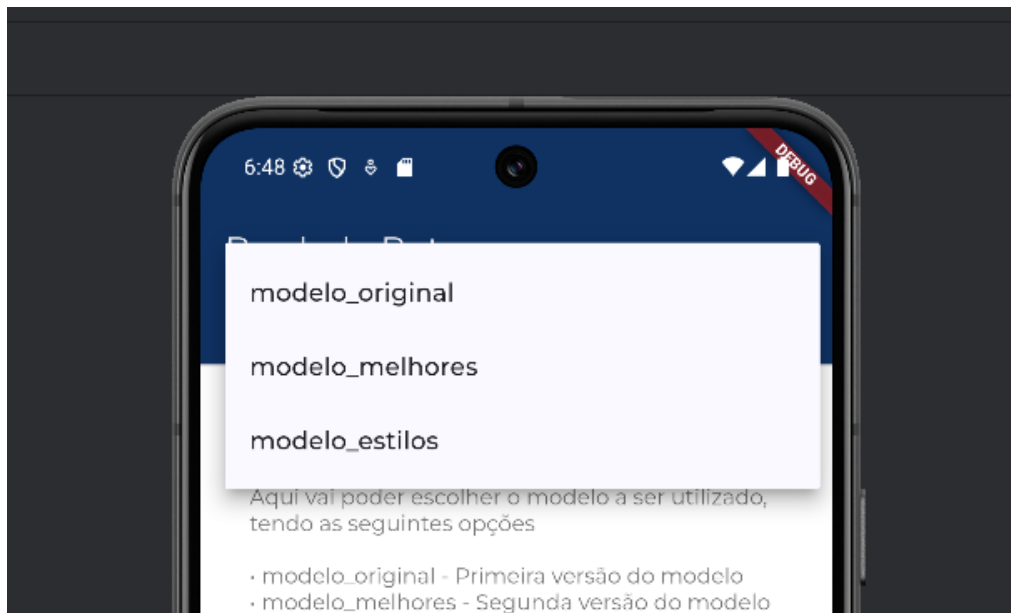


Figura 173 — Exemplo da seleção de modelos

Na aba “Texto → Imagem”, seguem-se dois campos para a introdução dos *prompts* positivos e negativos. Além destes campos, o utilizador pode ajustar o parâmetro ‘Criatividade’ do modelo, que corresponde ao valor de `cfg_scale` enviado à API do *A1111*. Este parâmetro quantifica a importância dos *prompts* textuais na geração da imagem:

- **Baixa:** O modelo segue os *prompts* de forma mais restrita, tentando garantindo a presença dos conteúdos dos *prompts* (*cfg_scale* = 12).
- **Média:** Utiliza o valor ótimo estudado (*cfg_scale* = 10), seguindo moderadamente os *prompts*.
- **Alta:** O modelo tem maior liberdade criativa (*cfg_scale* = 7), mas ainda considera os *prompts* inseridos.

Após o preenchimento dos parâmetros, o utilizador clica no botão 'Gerar Imagem' e o pedido JSON é enviado ao servidor A1111.



Figura 174 — Exemplo da interface da *tab txt2img*

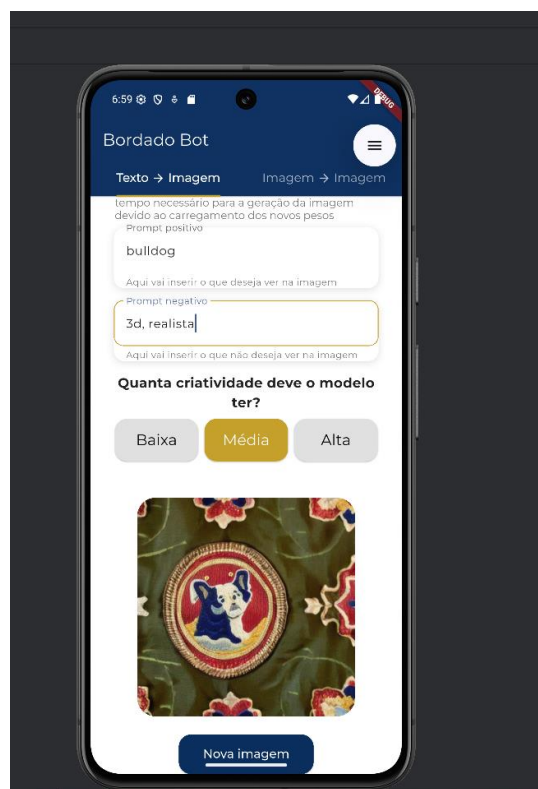


Figura 175 — Exemplo do uso da aplicação no modo *text2img*

Na aba “Imagem → Imagem”, mantém-se o seletor de modelo e a respetiva descrição. A principal diferença é a substituição dos campos de *prompt* por um botão ‘Selecionar Imagem Base’, que permite ao utilizador escolher a imagem a ser transformada (Figura 176).

Neste modo, o utilizador pode personalizar o número de *steps* e o valor de *denoising_strength* através de dois *sliders*. O primeiro destinado para *steps* varia de 20-50, influenciando diretamente a qualidade e o tempo de processamento da imagem: quanto maior o número de passos, maior a qualidade, mas também o tempo de espera. Já o segundo tem o objetivo de permitir ao utilizador alterar o valor de *denoising_strength*, que varia entre 0.5 e 0.75, janela de valores ótimos estudada anteriormente.

Após preencher os campos necessários, o utilizador clica em ‘Processar Imagem’ e o pedido JSON é enviado ao servidor *A1111*, iniciando a transformação.

Esta abordagem pretende que até utilizadores inexperientes consigam gerar imagens de forma autónoma, promovendo uma experiência de utilização positiva e alinhada com os objetivos do projeto.

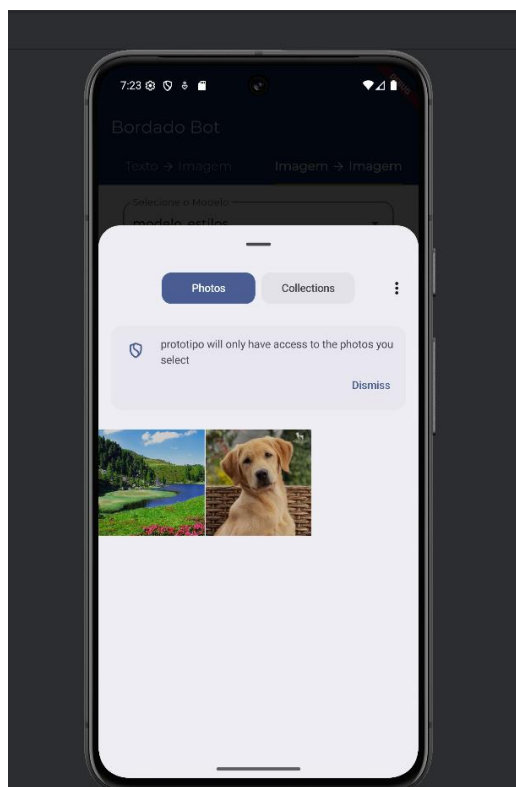
Figura 176 — Exemplo da interface da *tab img2img*

Figura 177 — Exemplo de como carregar uma imagem

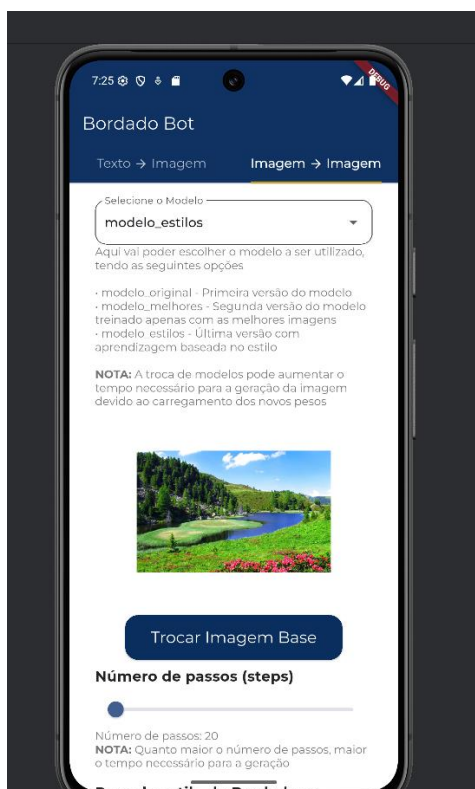


Figura 178 — Exemplo do uso do modelo *img2img*



Figura 179 — Continuação do exemplo de uso do modelo *img2img*

11.4 Como os *prompts* são geridos?

De modo a tornar a experiência de uso consistente e os resultados do protótipo apelativos, foi utilizada uma funcionalidade chave dentro do *A1111* de nome *styles*. Esta funcionalidade permite colocar o texto inserido do utilizador dentro de um *prompt* pré feito, desta forma, quando for escrever o que deseja ver no estilo do Bordado de Castelo Branco, não precisa de se preocupar com todos os *tokens* extra a usar, pois, independentemente do texto que inserir este vai ser incluindo dentro dos *prompts* estudados ao longo deste projeto. A figura a baixo exemplifica este funcionamento visualmente.

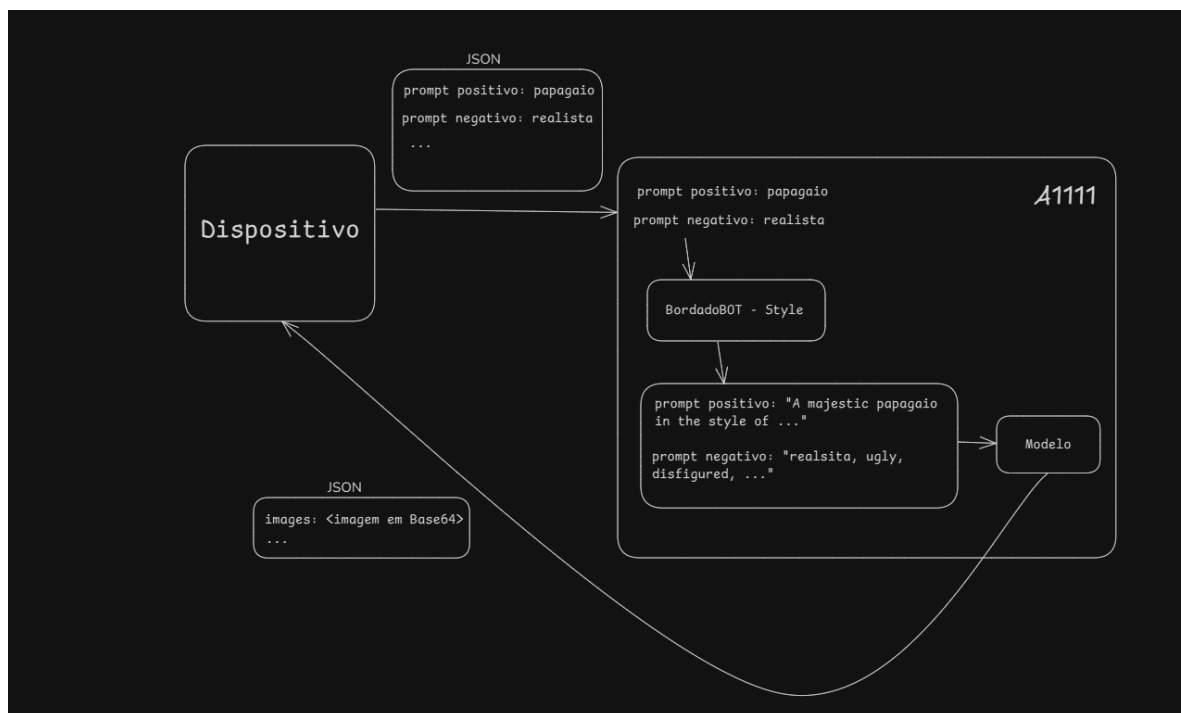


Figura 180 — Funcionamento do protótipo em *txt2img* visualmente

Embora esta abordagem seja excelente para manter a standardização dos resultados, surge aqui um problema relevante relacionado com a língua que o utilizador faz o seu *prompt*. Como maior parte dos modelos de *Stable Diffusion* são treinados usando maioritariamente palavras do dicionário inglês, a junção e mistura de termos em português vai dificultar a geração de resultados satisfatórios. É possível ver nas figuras seguintes onde vai ser pedido ao modelo para gerar uma imagem de um papagaio no estilo do Bordado de Castelo Branco.



Figura 181 — Imagem gerada pelos *prompts* em português



Figura 182 — Imagem gerada pelos mesmos *prompts*, mas em inglês

Como é possível verificar existe uma diferença claríssima na qualidade do *output* gerado, apesar da versão em português até conseguir fazer um resultado satisfatório como uma imagem de um bordado sem um tema específico, não existe qualquer elemento que remeta para um papagaio como foi pedido no *prompt*.

De modo a minimizar este inconveniente, adicionámos na interface do *txt2img* uma indicação clara na secção “NOTAS”, informando os utilizadores de que os resultados tendem a ser mais satisfatórios quando os *prompts* são inseridos em inglês. Esta recomendação tem como objetivo orientar o utilizador para práticas que maximizem a qualidade das imagens geradas pelo modelo.

12. Conclusões

Este projeto constitui um marco relevante na junção entre tecnologia contemporânea e património cultural, ao utilizar modelos de inteligência artificial, particularmente uso de *Stable Diffusion*, para reinterpretar e preservar o estilo do bordado de Castelo Branco. O trabalho desenvolvido evidenciou que os modelos de difusão possuem capacidade de captar e reproduzir elementos distintivos do bordado, tais como padrões florais, simetrias e cores vibrantes, possibilitando a geração de imagens que reinterpretem esta arte tradicional de forma inovadora. Mesmo com limitações, especialmente no que diz respeito à qualidade e quantidade das imagens disponíveis para treino, o modelo inicial demonstrou resultados iniciais promissores, sobretudo em cenários de *txt2img*, nos quais as imagens geradas exibem uma influência clara do estilo característico do bordado de Castelo Branco.

No entanto, ao longo do projeto foram identificados desafios significativos, com destaque para a dificuldade em distinguir o conteúdo original da imagem dos elementos estilísticos do bordado em cenários *img2img*. O modelo inicial revelou limitações na preservação da semântica e estrutura das imagens fornecidas, o que, por vezes, resultou em artefactos visuais ou distorções que comprometem a fidelidade ao estilo desejado.

Com base nesses desafios, avançámos para o treino de dois novos modelos, explorando diferentes estratégias, nomeadamente a alteração das *strings* que orientam o treino e a seleção cuidada do *dataset* utilizado. Deste processo resultaram dois novos modelos distintos: o modelo_melhores, treinado com um dataset reduzido, e o modelo_styles, focado na aprendizagem específica do estilo desejado. Após avaliação comparativa entre todos, concluiu-se que o modelo_styles apresentou resultados claramente superiores, sendo selecionado para testes exaustivos de parâmetros e *prompts*, com o objetivo de extrair o máximo potencial das imagens geradas.

Após a otimização dos parâmetros, avançámos para o desenvolvimento de um protótipo multiplataforma, capaz de comunicar com um servidor *A1111*, permitindo assim a utilização dos modelos treinados por um público mais vasto. A criação deste protótipo representou uma oportunidade de aprofundar conhecimentos práticos com *Flutter/Dart* que até então tinham apenas sido exploradas apenas em contexto teórico noutra unidade curricular. O protótipo foi concebido tendo em conta a acessibilidade, permitindo que qualquer utilizador possa experimentar a transformação das suas próprias imagens para o estilo típico do bordado de Castelo Branco, ainda que os resultados obtidos não tenham sido completamente satisfatórios em todos os casos.

Apesar da evidente melhoria nos resultados obtidos após a otimização dos parâmetros, ainda não foi possível alcançar resultados considerados completamente satisfatórios. No modo *txt2img*, o modelo apresenta um desempenho bastante positivo, conseguindo integrar conceitos específicos e elementos que evocam o estilo tradicional do bordado de Castelo Branco, embora alguns detalhes estilísticos característicos ainda não sejam totalmente inseridos.

Já no modo *img2img*, todos os modelos avaliados enfrentaram dificuldades significativas, tal como já observado em Projeto I. A distinção dos elementos em fotografias reais continua a ser um grande desafio, assim como a incorporação adequada desses elementos com os padrões do bordado. Estas limitações resultam em resultados muito inconsistentes, sendo que a utilização de fotografias reais com baixa qualidade ou problemas de iluminação agravam as dificuldades.

Em síntese, este projeto demonstrou que a aplicação de inteligência artificial, como o *Stable Diffusion*, é viável e promissora para reinterpretar e preservar o estilo do bordado de Castelo Branco, captando padrões estéticos tradicionais e integrando-os em novas imagens através de uma transferência de estilos. Embora tenham sido alcançados resultados positivos, especialmente no modo *txt2img*, persistem desafios técnicos na distinção e fusão com os elementos do bordado em fotografias reais, evidenciando a necessidade de maior refinamento dos *datasets* e dos modelos, bem como da otimização contínua dos parâmetros para a criação das imagens.

A criação de um protótipo multiplataforma acessível ampliou o alcance da tecnologia, tornando possível a experimentação prática do público em geral, e reforçando o potencial da inteligência artificial como ferramenta de valorização e inovação do património cultural, mesmo que ainda existam limitações a superar para garantir resultados consistentes e plenamente satisfatórios em todos os cenários.

12.1 Desafios e limitações

Ao longo do desenvolvimento deste projeto, enfrentámos uma série de dificuldades e contratempos que acabaram por ter um impacto direto na qualidade dos resultados obtidos.

O primeiro e talvez o maior desafio foi a escassez de imagens de boa qualidade disponíveis *online* que retratassem o Bordado de Castelo Branco de forma fiel e detalhada. Esta limitação obrigou-nos a trabalhar com um dataset inferior ao

esperado, o que comprometeu a capacidade do modelo de aprender e reproduzir corretamente o estilo pretendido, uma vez que a falta de exemplos variados e em alta resolução dificultou o treino do modelo.

Outro fator que prejudicou a obtenção de resultados mais satisfatórios foi o *hardware* utilizado para o desenvolvimento e teste do projeto. Todo o processo de geração de imagens foi realizado num portátil equipado com apenas 4 GB de VRAM dedicada, o que é considerado baixo para a obtenção de resultados que aproveitem todo o potencial destas tecnologias.

Embora o equipamento se mantenha funcional, já não possui a capacidade gráfica necessária para gerar imagens com a máxima qualidade possível localmente, especialmente em resoluções mais elevadas. Por esta razão, todas as imagens tiveram de ser geradas em tamanho fixo de 512x512 pixels, uma dimensão muito abaixo do considerado padrão para fotografias ou imagens atuais.

Em suma, todas estas adversidades técnicas tiveram um impacto real na qualidade das imagens produzidas pelo *A1111*, evidenciando a importância de contar com recursos adequados para projetos futuros nesta área.

12.2 Trabalho futuro

Embora a nossa iteração do BordadoBOT fique concluída com este documento, gostaríamos imenso de ver este projeto continuar a ser desenvolvido em anos futuros. O campo da inteligência artificial está em constante e acelerada evolução, trazendo novas abordagens, modelos e técnicas capazes de superar as limitações identificadas ao longo deste trabalho. Estamos certos de que, com o avanço da tecnologia, será possível alcançar resultados ainda mais positivos e próximos do estilo pretendido.

Além disso, acreditamos que futuras versões poderão beneficiar de *hardware* mais potente, para enriquecer tanto o processo de treino quanto a qualidade dos *outputs* gerados. Desta forma, não só aperfeiçoar a capacidade de reprodução do estilo tradicional, mas também abrir portas para novas aplicações criativas, contribuindo para a preservação e divulgação deste património cultural português.

Referências

- [1] Y. Zuo *et al.*, “Towards Multi-View Consistent Style Transfer with One-Step Diffusion via Vision Conditioning,” Nov. 2024, [Online]. Available: <http://arxiv.org/abs/2411.10130>
- [2] L. A. Gatys, M. Bethge, A. Hertzmann, and E. Shechtman, “Preserving Color in Neural Artistic Style Transfer,” Jun. 2016, [Online]. Available: <http://arxiv.org/abs/1606.05897>
- [3] J. Huang, M. Yan, Y. Liu, and S. Chen, “Color-SD: Stable Diffusion Model Already has a Color Style Noisy Latent Space,” in *Proceedings - IEEE International Conference on Multimedia and Expo*, IEEE Computer Society, 2024. doi: 10.1109/ICME57554.2024.10687653.
- [4] A. T. Carceller, “The ARTificial Revolution: Challenges for redefining Art Education in the paradigm of generative artificial intelligence,” *Digital Education Review*, no. 45, pp. 84–90, Jul. 2024, doi: 10.1344/der.2024.45.84-90.
- [5] “A1111.” Accessed: Jan. 29, 2025. [Online]. Available: <https://github.com/AUTOMATIC1111/stable-diffusion-webui>
- [6] M. Kaur, A. K. Shukla, and S. Kaur, “An Introduction to Machine Learning in a Nutshell,” in *Proceedings of the 2021 10th International Conference on System Modeling and Advancement in Research Trends, SMART 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 17–22. doi: 10.1109/SMART52563.2021.9676315.
- [7] Z. Qiu, H. Zhao, and S. Wang, “Applications and Challenges of Artificial Intelligence in Aerospace Engineering,” in *2023 6th International Conference on Artificial Intelligence and Big Data, ICAIBD 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 970–974. doi: 10.1109/ICAIBD57115.2023.10206205.
- [8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 2022, pp. 10674–10685. doi: 10.1109/CVPR52688.2022.01042.
- [9] A. Horvath, “Stable Diffusion with Continuous-time Neural Networks,” Institute of Electrical and Electronics Engineers (IEEE), Sep. 2024, pp. 1–4. doi: 10.1109/cnna60945.2023.10652658.
- [10] “Artificial Intelligence’s Use and Rapid Growth Highlight Its Possibilities and Perils.” Accessed: Dec. 07, 2024. [Online]. Available: <https://www.gao.gov/blog/artificial-intelligences-use-and-rapid-growth-highlight-its-possibilities-and-perils>

-
- [11] B. Liu, "'Weak AI' is Likely to Never Become 'Strong AI', So What is its Greatest Value for us?," Mar. 2021, [Online]. Available: <http://arxiv.org/abs/2103.15294>
- [12] P. Bory, S. Natale, and C. Katzenbach, "Strong and weak AI narratives: an analytical framework," *AI Soc*, 2024, doi: 10.1007/s00146-024-02087-8.
- [13] B. J. Copeland, "Early AI in Britain: Turing et al.," *IEEE Annals of the History of Computing*, vol. 45, no. 3, pp. 19–31, Jul. 2023, doi: 10.1109/MAHC.2023.3300660.
- [14] L. Li, N. N. Zheng, and F. Y. Wang, "On the Crossroad of Artificial Intelligence: A Revisit to Alan Turing and Norbert Wiener," *IEEE Trans Cybern*, vol. 49, no. 10, pp. 3618–3626, Oct. 2019, doi: 10.1109/TCYB.2018.2884315.
- [15] L. Cao, "A New Age of AI: Features and Futures," *IEEE Intell Syst*, vol. 37, no. 1, pp. 25–37, 2022, doi: 10.1109/MIS.2022.3150944.
- [16] S. K. Puli and P. Usha, "Transforming Healthcare: Advancements, Applications, and Future Directions of Machine Learning," in *2024 10th International Conference on Smart Computing and Communication, ICSCC 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 502–506. doi: 10.1109/ICSCC62041.2024.10690530.
- [17] . IEEE Staff, *2009 International Conference on Information Engineering and Computer Science*. I E E E, 2009.
- [18] "Machine Learning vs Traditional Programming." Accessed: Jan. 22, 2025. [Online]. Available: <https://www.avenga.com/magazine/machine-learning-programming/>
- [19] "Types of Machine Learning." Accessed: Jan. 22, 2025. [Online]. Available: <https://www.javatpoint.com/types-of-machine-learning>
- [20] C. Enyinna Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning."
- [21] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014, [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [22] "What is a neural network?" Accessed: Jan. 25, 2025. [Online]. Available: <https://www.cloudflare.com/learning/ai/what-is-neural-network/>
- [23] D. H. Hagos, R. Battle, and D. B. Rawat, "Recent Advances in Generative AI and Large Language Models: Current Status, Challenges, and Perspectives," *IEEE Transactions on Artificial Intelligence*, 2024, doi: 10.1109/TAI.2024.3444742.
- [24] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing [Review Article]," Aug. 01, 2018, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/MCI.2018.2840738.

- [25] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep Unsupervised Learning using Nonequilibrium Thermodynamics," Mar. 2015, [Online]. Available: <http://arxiv.org/abs/1503.03585>
- [26] "Review: Denoising Diffusion Probabilistic Models (DDPM)." Accessed: Jan. 30, 2025. [Online]. Available: <https://andychungyatming.github.io/2022/12/21/Review-DDPM/>
- [27] "An Introduction to Diffusion Models and Stable Diffusion." Accessed: Jan. 30, 2025. [Online]. Available: <https://blog.marvik.ai/2023/11/28/an-introduction-to-diffusion-models-and-stable-diffusion/>
- [28] Y. Liu, J. Yue, S. Xia, P. Ghamisi, W. Xie, and L. Fang, "Diffusion Models Meet Remote Sensing: Principles, Methods, and Perspectives," Apr. 2024, doi: 10.1109/TGRS.2024.3464685.
- [29] I. J. Goodfellow *et al.*, "Generative Adversarial Networks," Jun. 2014, [Online]. Available: <http://arxiv.org/abs/1406.2661>
- [30] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," Dec. 2013, [Online]. Available: <http://arxiv.org/abs/1312.6114>
- [31] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear Independent Components Estimation," Oct. 2014, [Online]. Available: <http://arxiv.org/abs/1410.8516>
- [32] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models." [Online]. Available: <https://github.com/CompVis/latent-diffusion>
- [33] "What is a variational autoencoder?" Accessed: Jan. 30, 2025. [Online]. Available: <https://www.ibm.com/think/topics/variational-autoencoder>
- [34] R. Azad *et al.*, "Medical Image Segmentation Review: The Success of U-Net," *IEEE Trans Pattern Anal Mach Intell*, 2024, doi: 10.1109/TPAMI.2024.3435571.
- [35] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," May 2015, [Online]. Available: <http://arxiv.org/abs/1505.04597>
- [36] A. Radford *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," Feb. 2021, [Online]. Available: <http://arxiv.org/abs/2103.00020>
- [37] "Training A Diffusion Model - Stable Diffusion Masterclass."
- [38] "The state of AI in early 2024: Gen AI adoption spikes and starts to generate value," 2024.
- [39] "IEEE Xplore." Accessed: Dec. 07, 2024. [Online]. Available: <https://ieeexplore.ieee.org/Xplore/home.jsp>
- [40] "B-On." Accessed: Dec. 07, 2024. [Online]. Available: <https://www.b-on.pt/>

-
- [41] H. Wu, S. Lim, and B. Xiao, "Animated Avatar Generation Technology Research Based on Deep Convolutional Generative Adversarial Network integrated with Self-attention and Spectral Normalization," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3482989.
- [42] G. Wu, X. Liu, J. Jia, X. Cui, and G. Zhai, "Text2QR: Harmonizing Aesthetic Customization and Scanning Robustness for Text-Guided QR Code Generation," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2024, pp. 8456–8465. doi: 10.1109/CVPR52733.2024.00808.
- [43] M. N. Everaert, M. Bocchio, S. Arpa, S. Susstrunk, and R. Achanta, "Diffusion in Style," in *Proceedings of the IEEE International Conference on Computer Vision*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 2251–2261. doi: 10.1109/ICCV51070.2023.00214.
- [44] Y. D. Rueda-Arango, D. Rojas-Velazquez, A. V. Gorelova, J. Garssen, A. Tonda, and A. Lopez-Rincon, "Image Generation with Interactive Evolutionary System using Bayesian Optimization," in *International Conference on Human System Interaction, HSI*, IEEE Computer Society, 2024. doi: 10.1109/HSI61632.2024.10613596.
- [45] Y. Xu, X. Xu, H. Gao, and F. Xiao, "SGDM: An Adaptive Style-Guided Diffusion Model for Personalized Text to Image Generation," *IEEE Trans Multimedia*, 2024, doi: 10.1109/TMM.2024.3399075.
- [46] M. Hamazaspian and S. Navasardyan, "Diffusion-Enhanced PatchMatch: A Framework for Arbitrary Style Transfer with Diffusion Models," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, IEEE Computer Society, 2023, pp. 797–805. doi: 10.1109/CVPRW59228.2023.00087.
- [47] L. Wang *et al.*, "GlyphGenius: Unleashing the potential of AIGC in Chinese character learning," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3464562.
- [48] Z. K. J. Hartley, R. J. Lind, M. P. Pound, and A. P. French, "Domain Targeted Synthetic Plant Style Transfer using Stable Diffusion, LoRA and ControlNet," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Jun. 2024, pp. 5375–5383. doi: 10.1109/CVPRW63382.2024.00546.
- [49] J. Huang, Y. Gao, Z. Jie, Y. Zhong, X. Han, and L. Ma, "MRStyle: A Unified Framework for Color Style Transfer with Multi-Modality Reference," Sep. 2024, [Online]. Available: <http://arxiv.org/abs/2409.05250>
- [50] Z. Zhang *et al.*, "Towards Highly Realistic Artistic Style Transfer via Stable Diffusion with Step-aware and Layer-aware Prompt," Apr. 2024, [Online]. Available: <http://arxiv.org/abs/2404.11474>

- [51] S. Li, “DiffStyler: Diffusion-based Localized Image Style Transfer,” Mar. 2024, [Online]. Available: <http://arxiv.org/abs/2403.18461>
- [52] J. Liao, “A Study on Neural Style Transfer Methods for Images,” in *Proceedings - 2022 2nd International Conference on Big Data, Artificial Intelligence and Risk Management, ICBAR 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 60–64. doi: 10.1109/ICBAR58199.2022.00019.
- [53] M. Bogacz and M. Iwanowski, “Towards many-to-one neural style transfer method,” in *International Conference on Human System Interaction, HSI*, IEEE Computer Society, Jul. 2021. doi: 10.1109/HSI52170.2021.9538780.
- [54] “Non-Photorealistic Rendering Pen-and-ink Illustrations Painterly Rendering Cartoon Shading Technical Illustrations.”
- [55] “How to style transfer your own images.” Accessed: Jan. 28, 2025. [Online]. Available: <https://xebia.com/blog/how-to-style-transfer-your-own-images/>
- [56] “História do bordado de Castelo Branco.” Accessed: Jan. 07, 2025. [Online]. Available: https://pt.wikipedia.org/wiki/Bordado_de_Castelo_Branco
- [57] “Bordado de Castelo Branco.” Accessed: Jan. 07, 2025. [Online]. Available: <https://www.cm-castelobranco.pt/municipe/noticias/detalhe-noticia/?id=18993>
- [58] “Google Custom Search JSON API.” Accessed: Jan. 20, 2025. [Online]. Available: <https://developers.google.com/custom-search/v1/overview?hl=pt-br>
- [59] “Python.”
- [60] “Google Colab.” Accessed: Jan. 29, 2025. [Online]. Available: <https://colab.research.google.com/>
- [61] “Hugging Face.” Accessed: Jan. 29, 2025. [Online]. Available: <https://huggingface.co/>
- [62] “Brime.” Accessed: Jan. 29, 2025. [Online]. Available: <https://www.birme.net/>
- [63] “Dreambooth.” Accessed: Jan. 29, 2025. [Online]. Available: <https://dreambooth.github.io/>
- [64] “GitHub.” Accessed: Jan. 30, 2025. [Online]. Available: <https://github.com/>
- [65] “Dart.” Accessed: Jun. 28, 2025. [Online]. Available: <https://dart.dev/>
- [66] “Flutter.” Accessed: Jun. 28, 2025. [Online]. Available: <https://flutter.dev/>
- [67] “DreamBooth.” Accessed: Feb. 01, 2025. [Online]. Available: <https://huggingface.co/docs/diffusers/training/dreambooth>
- [68] “Know these Important Parameters for stunning AI images.” Accessed: Jan. 26, 2025. [Online]. Available: https://stable-diffusion-art.com/know-these-important-parameters-for-stunning-ai-images/#Sampling_methods

- [69] “Basic usage of Stable Diffusion web UI (v1.9.0) Text-to-image section.” Accessed: Jan. 26, 2025. [Online]. Available: <https://www.digitalcreativeai.net/en/post/how-to-use-stable-diffusion-web-ui-text-to-image>
- [70] Z. Ma *et al.*, “Efficient Diffusion Models: A Comprehensive Survey from Principles to Practices,” Oct. 2024, doi: 10.1109/TPAMI.2025.3569700.