UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Sintetizador de Merchandise: translação de estilo na produção de Merchandise em escala

Eric Koji Yang Imai

Trabalho de Conclusão de Curso
MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Sintetizador de Merchandise: translação de estilo na produção de Merchandise em escala

USP - São Carlos

2022

Eric Koji Yang Imai

Sintetizador de Merchandise: translação de estilo na produção de Merchandise em escala

Trabalho de conclusão de curso apresentado ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientadora: Prof. Dr. Adenilson da Silva Simão

USP - São Carlos

2022

Dedico este trabalho aos meus familiares e amigos, pela compreensão e suporte nos grandes momentos de alegria e superação.

AGRADECIMENTOS

Agradeço ao professor do ICMC Adenilson da Silva Simão pelo acompanhamento ao longo deste processo de produção de tese de conclusão de curso, que além dos conhecimentos transmitidos contribuiu de forma êmpática em um momento de inúmeras demandas profissionais e pessoais, além MBA. Sem este tato e empatia, com toda certeza, haveria outras dificuldades diante deste desafio final.

De forma paralela, deixo os meus agradecimentos a minha família e amigos próximos que sempre estiveram ao meu lado para me apoiar nos momentos de superação. A atividade acadêmica sempre trouxe consigo a necessidade de algumas abdicações e são essas pessoas que sempre fizeram a diferença no dia a dia.

Por fim, dedico a este último parágrafo a Escola Politécnica da USP que em minha gradução sempre me motivou ao aprendizado e me capacitou para abraçar desafios cada vez maiores. O bacharelado foi uma passagem de extrema importância no meu amadurecimento acadêmico e profissional.

“*É por isso que os filósofos nos alertam para não ficarmos satisfeitos com o mero aprendizado, mas para adicionar a prática e depois o treinamento. Pois com o passar do tempo esquecemos o que aprendemos e acabamos fazendo o oposto, e temos opiniões opostas do que deveríamos.*”

(EPITETO)

RESUMO

Já diziam que a criatividade e o processo de criação da arte e design era algo inerênte e único ao ser humano, atividade que tecnologia nenhuma poderia replicar. Porém estudos veem mostrando que redes neurais generativas adversativas conseguem produzir criatividade a partir de translação de estilo entre domínios distintos de imagem. Dito isto, este projeto de conclusão de curso buscou explorar esta tecnologia a ponto de se identificar uma arquitetura que pudesse agregar no mercado de merchandise, gerando em escala inúmeras peças distintas para uma marca. Durante o processo de descoberta foram analisadas arquiteturas como VAE, CycleGAN, UNIT, entre outras, onde a escolhida para o fim desejado foi a arquitetura FUNIT dado características que facilitariam a obtenção de dados, flexibilidade, generalização e multimodalidade.

Palavras-chave: Rede Neural Generativa Adversativa, Translação de estilo, FUNIT.

ABSTRACT

Many people said that creativity and the process of creating art and design were something inherent and unique to human beings, an activity that no technology could replicate. However, studies are showing that generate adversarial networks could produce creativity from different image domains. Therefore, this final paper explores this technology to the point of identifying an architecture that could aggregate in merchandise market, generating innumerable distinct garments for a brand. During the discovery process, the architectures of Cycle GAN, UNIT, et cetera, were analyzed and FUNIT was chosen for the desired purpose because of the characteristics that would facilitate the data needy, flexibility, generalization and multimodality.

Keywords: Adversarial Generative Neural Network, Style translation, FUNIT.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ETA	Estimated Completion Time
FUNIT	Few-Shot Unsupervised Image-to-Image
GAN	Generative Adversarial Networks
MUNIT	Multimodal Unsupervised Image-to-Image Translation
UNIT	Unsupervised Image-to-Image Translation
VAE	Variationl Autoenconder
VGG	Visual Geometry Group

SUMÁRIO

1. INTRODUÇÃO

1.1 Contextualização do problema

O termo *design* no campo artístico possui origem recente, surgiu apenas há poucos séculos atrás, no Brasil só adquiriu o seu significado orientado ao planejamento recentemente, dado ter sido traduzido de forma literal por muitos anos como um sinônimo do verbo desenhar. Ele se difundiu pelo mundo com o manifesto "De Stij" de artistas no ano de 1917 em resposta a Revolução Industrial Inglêsa e hoje o *design* se refere não só ao ato de projetar no campo artístico, mas se tornou um termo que transborda esses limites e se integrou ao vocabulário popular (A origem do termo . Ifd. Disponível em: https://www.ifd.com.br/design/a-origem-do-termo-design/?quad_cc/. Acesso em: 02/09/2022).

Apesar do termo ter sido cunhado no contemporâneo, não é de hoje que o ser humano praticou o *design*. A criação artística sempre foi algo inerente a nossa espécie, sendo as primeiras pinturas rupestres datadas quarenta e cinco mil anos e meio atrás chegando a atualidade com obra famosas de Design Digital atrelados a criptoativos, como o Nyan Cat de saraj00n. Para qualquer um destes exemplos ao longo da história da humanidade, foi necessário uma grande dose de criatividade, uma característica que distingue o *Homo Sapiens* de outras espécies e que há muito tempo foi considerado uma das habilidades que faz do ser humano insubstituível quando comparado as novas tecnologias.

1.2 Justificativa

De acordo com a IBISWorld, o mercado global de design gráfico movimentou um faturamento de U$ 45,8 bilhões em 2021 e possui projeções de crescimento sólidos para os próximos anos (Designerd.com, 2022). Estes resultados se devem principalmente as mudanças de mercado e comportamentais da era digital focada em conteúdo e a sua aceleração com a chegada da pandemia global de Covid-19. O ambiente dos últimos anos cada vez mais necessita de profissionais na área do design, principalmente voltado no meio tecnológico, assim como a sua criatividade e metodologia de trabalho. Dito isto, um trabalho que permeie um mercado com ganho de importância comercialmente e socialmente se faz atrativo.

De maneira complementar, para o desenvolvimento de um projeto que possui como foco central o design é impossível não se atravessar a neurologia e a filosofia. Esta intersecção

natural de áreas de pesquisa possui potencial em contribuir para outros estudiosos, sejam eles de inteligência artificial ou não.

1.3 Motivação

O trabalho de conclusão de curso é uma oportunidade. O projeto escolhido foi planejado partindo desta premissa, de ser um momento singular em que posso praticar o ócio criativo com o suporte de um grande profissional na área e colher inúmeros proveitos desta experiência. Dito isto, o favoritismo de uma arquitetura GAN (Generative adversarial network) que replica estilos de design em objetos ficou bastante claro.

Primeiramente, a própria arquitetura, partindo da premissa de atingimento de resultados satisfatórios, já se mostra como um pilar de motivação. A solução pode muito bem se tornar um protótipo de produto funcional para um mercado em ascensão, como apontado em seções anteriores o mercado de design veem crescendo e já movimenta valores financeiros representativos

Com os resultados da solução como algo proveitoso e utilitário, todo o percurso para se chegar nesta realidade também se torna, visto que para isto há a necessidade de aprendizado em alguns campos de conhecimento que não possuo experiência, como arquiteturas GANs e processamento de imagem, ambas técnicas que não tive contanto previamente. Por fim, destaco que durante todo o percurso será exercitado uma temática de cunho filosófico que naturalmente gera curiosidade e por consequência interesse, que é automatizarmos a imaginação, algo que ao longo de todos os anos foi considerado algo inerente e único ao ser humano.

1.4 Questões de pesquisa

Dentro do desafio de se estruturar uma arquitetura GAN que estilize objetos, é necessário definir questões cruciais a serem respondidas, não apenas pelo fato de ser um método que auxilia no planejamento, execução e validação de um projeto, mas por serem parâmetros de suma importância para um trabalho de conclusão de curso dado a sua natureza acadêmica. Esclarecer os parâmetros e condições em que a arquitetura a ser desenvolvida se mostra funcional, assim como os que não se aproximam são os verdadeiros produtos da pesquisa. Dado as diretrizes, para cada etapa de projeto

Possuir uma base bem estruturada e em abundância para o projeto em foco é um dos fatores mais importantes para o seu sucesso. Entretanto, a sua estruturação e coleta é

normalmente bastante custosa e não necessariamente estes fatores serão cumpridos de forma integral para o treinamento das redes neurais a serem arquitetadas, desta forma definir a relação da quantidade de dados necessários para o bom sucesso e as suas dimensões de imagem (altura, largura e profundidade) são de enorme importância.

Da mesma forma que a definição da base de dados dos objetos é de suma importância para o bom funcionamento do projeto, entender os parâmetros do ruído a serem introduzidos também possui sua criticidade.

A arquitetura e seus parâmetros de treinamento, são o *core* do projeto a ser estruturado no trabalho de conclusão de curso. Apesar de já sabido que será utilizando uma arquitetura de rede adversária generativa é necessário se definir como serão os detalhes desta, quantidade de camadas, neurônios e conexões e os métodos de ativação.

Com a premissa de uma arquitetura de rede adversária generativa já escolhida é necessário explorar e definir o modelo de treinamento desta rede a fim de se obter a melhor performance possível.

1.5 Objetivos

O projeto de conclusão de curso possui por objetivo principal mesclar objetos com o design de outras peças de forma harmoniosa. A pesar de um dos critérios de sucesso do produto ser de cunho subjetivo, também será considerado resultados quantitativos relativos a generalização e qualidade de borda dos modelos, para assim definir uma arquitetura ideal a ser utilizada para o propósito de produção de merchandise em escala.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Fundamentos

A área de aprendizado de máquina reúne uma variedade de conceitos com certo grau de complexidade. O planejamento da arquitetura sem um embasamento teórico sólido é um fator que pode prejudicar o desenvolvido do produto final, dito isto, esta sessão é dedicada a definir os fundamentos teóricos necessários para o projeto organizadas de forma hierárquica conceitual.

2.1.1 Rede Neural Convolucional

O aprendizado de máquina é um subcampo da engenharia e da ciência da computação na inteligência artificial. O seu conceito foi introduzido em 1959 por Arthur Samuel como um campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados. Os algoritmos que seguem este conceito utilizam de dados amostrais para definir padrões e gerar valor futuros. (Canaltech, 2022).

As redes neurais convolucionais foram desenvolvidas principalmente para o tratamento matricial no campo do aprendizado de máquina, frequentemente utilizado no processamento de imagens. A sua arquitetura de tipo *feed-foward* usa uma variação de perpectrons multicamadas segmentada em uma camada convolucional, uma camada ReLU, uma camada de agrupamento (*pooling*) e uma camada de *dropout*. (Data Science Academy, 2022).

2.1.2 VGG

A arquitetura VGG foi desenvolvida por Simonyan e Zisserman, sendo ela uma rede neural com uma série de camadas convolucionais e muito utilizada para o aumento de profundidade no processamento de imagens. Ela possui diferentes configurações variando a quantidade de camadas. Na figura 1 podem ser visualizado as suas diferentes variações com os seus detalhes de arquitetura.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figura 1: arquitetura convNet (VGG)

Fonte: SIMONYAN; ZISSERMAN, 2014

2.1.3 Autoencoder

Os Autoencoders, uma técnica de machine learning não supervisionada, foram introduzidos pela primeira vem na década de 1980, sendo um dos seus principais contribuintes Geoffey Hinton. Esta técnica possui como cerne que informações de altas dimensionalidades possuem redundâncias e podem ser minimizadas em vetores latentes de baixa dimensionalidade a partir de técnicas de redução de dimensionalidade como o PCA (Principal Component Analysis). Entretanto, para o campo de conhecimento de geração de imagens utilizaremos desta etapa para posteriormente transformar o produto da redução de dimensionalidade para uma versão de dimensionalidade igual a inicial JPEG, conjuntamente a manipulações de resultado.

A sua arquitetura é formada por um encoder, uma camada escondida e um decoder. O primeiro possui por função reduzir a dimensionalidade do dado inicial nas variáveis latentes de baixa dimensionalidade. Por fim o decoder aumenta a dimensionalidade a partir das variáveis latentes.

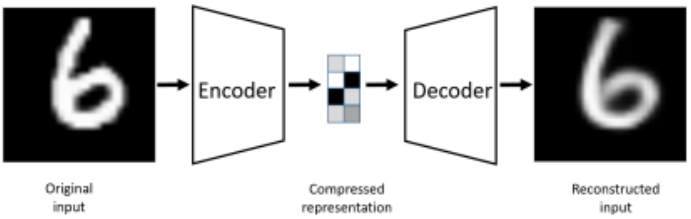


Figura 2:Estrutura Autoencoder

Fonte: DOR BANK, 2020

2.1.4 Variational Autoencoder (VAE)

Desenvolvido em 2014, um modelo que a amostragem é feita a partir de uma distribuição parametrizada gaussiana na construção das variáveis latentes. Passo adicional entre o encoder e o decoder, o encoder produz a média e a variância a partir de uma distribuição gaussiana como variáveis latentes N(0,1)*variancia + média

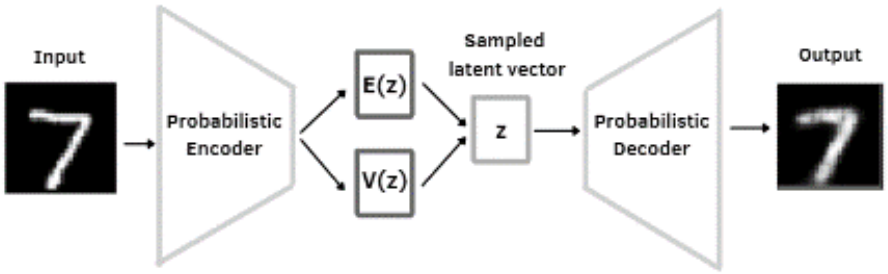


Figura 3: arquitetura VAE

Fonte: EUGENIA ANELLO, 2021

O erro é propagado para ajustar os parâmetros das redes reunais. Este modelo busca adaptar o modelo de autoeconder para funções generativas evitando o overfiting.

2.1.5 Rede adversária generativa

A arquitetura de rede adversária generativa pode ser dividida em duas redes neurais, sendo a primeira a geradora, que a partir de ruídos gera novas instâncias de dados, enquanto a segunda, discriminadora, avalia a autenticidade dos dados sintetizados a partir do padrão aprendido de dados reais. Dito isto, a fim de se esboçar novas referências que possuem alto grau de semelhança com dados reais a rede discriminadora possui por objetivo atingir 50% de discretização de dados reais ou gerados, ou seja, não consegue distinguir estes.

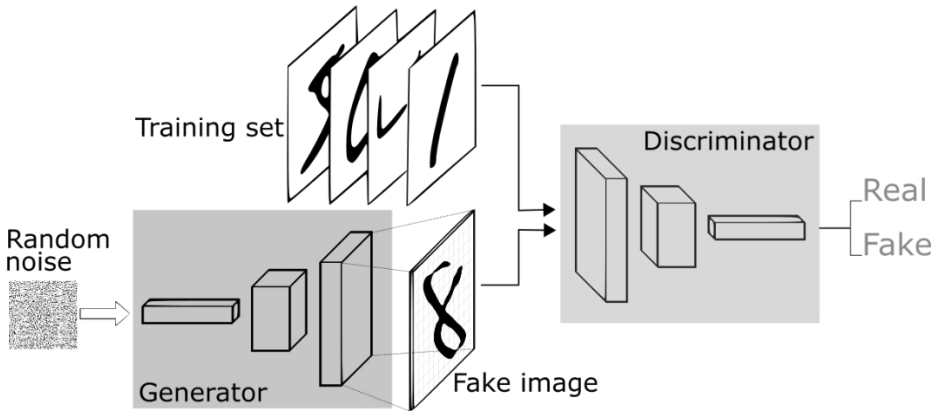


Figura 4: Arquitetura GAN

Fonte: THALLES SILVA, 2020

2.1.6 Pix2Pix

A arquitetura Pix2Pix é constituída por uma rede neural generativa adversativa de arquitetura U-Net para o módulo gerador, a partir de imagens pareadas, ou seja, um método de aprendizado supervisionado.

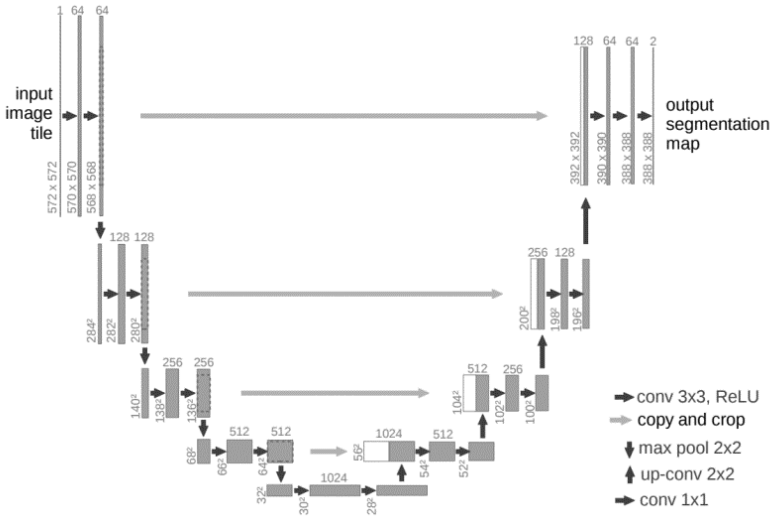


Figura 5: Estrutura U-Net

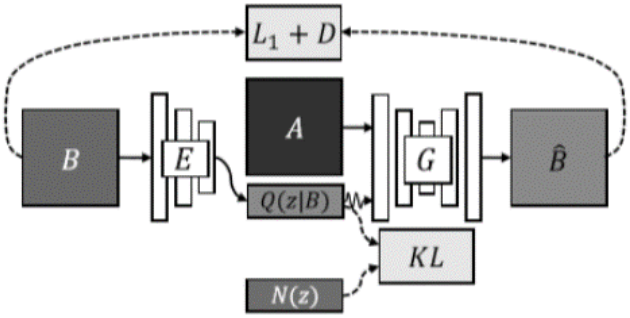
Fonte: RONNEBERGER; FISCHER; BROX, 2015

Esta arquitetura possui este nome dado o seu formato de “U” onde cada conjunto de camadas de convolução do bloco gerador direciona o seu resultado para o bloco discriminador para adicionar de forma latente o contexto dos dados utilizados no aprendizado (ISOLA et al., 2016).

2.1.7 BicycleGAN

Dentro das arquiteturas de translação de estilo, a arquitetura BicycleGAN é uma multimodal e supervisionada, ela necessita de dados rotulados e pareados para o seu treinamento. A sua estrutura é formada por dois componentes, cVAE-GAN (Conditional Variational AutoEncoder GAN) e cLR-GAN(Conditional Latent Regressos GAN).

O componente cVAE-GAN possui a função de codificar a imagem verdade B em um espaço latente usando o codificador E (U-NET), sendo este resultado utilizado na etapa posterior de decodificação junto a imagem pareada A para reconstruir a imagem verdade B.



$$G^*, E^* = \arg\min_{G,E}\max_{D} \quad \mathcal{L}_{\text{GAN}}^{\text{VAE}}(G, D, E) + \lambda \mathcal{L}_1^{\text{VAE}}(G, E) + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}(E)$$

where

$$\mathcal{L}_1^{\text{VAE}}(G) = \quad \mathbb{E}_{\mathbf{A},\mathbf{B}\sim p(\mathbf{A},\mathbf{B}),\mathbf{z}\sim E(\mathbf{B})} ||\mathbf{B} - G(\mathbf{A}, \mathbf{z})||_1$$

$$\mathcal{L}_{\text{GAN}}^{\text{VAE}} = \mathbb{E}_{\mathbf{A},\mathbf{B}\sim p(\mathbf{A},\mathbf{B})}[\log(D(\mathbf{A}, \mathbf{B}))] + \mathbb{E}_{\mathbf{A},\mathbf{B}\sim p(\mathbf{A},\mathbf{B}),\mathbf{z}\sim E(\mathbf{B})}[\log(1 - D(\mathbf{A}, G(\mathbf{A}, \mathbf{z})))]$$

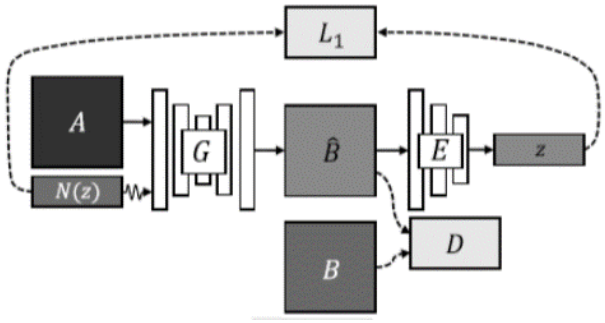
$$\mathcal{L}_{\text{KL}}(E) = \mathbb{E}_{\mathbf{B}\sim p(\mathbf{B})}[\mathcal{D}_{\text{KL}}(E(\mathbf{B})|| \mathcal{N}(0, I))],$$

where $\mathcal{D}_{\text{KL}}(p||q) = -\int p(z) \log \frac{p(z)}{q(z)} dz.$

Figura 6: cVAE-GAN

Fonte: JUN-YAN ZHU; RICHARD ZHANG; DEEPAK PATHAK; TREVOR DARRELL; ALEXEI A. EFROS; OLIVER WANG; ELI SHECHTMAN, 2018.

O segundo componente, cLR-GAN, utilizando a imagem pareada A busca gerar a imagem B^ para ser comparado com a imagem real B e codificado para replicar o espaço latente antes construído pelo cVAE-GAN (JUN et al., 2017).



$$G^*, E^* = \arg\min_{G,E}\max_{D} \quad \mathcal{L}_{\text{GAN}}(G, D) + \lambda_{\text{latent}}\mathcal{L}_1^{\text{latent}}(G, E)$$

$$\mathcal{L}_{\text{GAN}}(G, D) = \mathbb{E}_{\mathbf{A},\mathbf{B}\sim p(\mathbf{A},\mathbf{B})}[\log(D(\mathbf{A}, \mathbf{B}))] + \mathbb{E}_{\mathbf{A}\sim p(\mathbf{A}),\mathbf{z}\sim p(\mathbf{z})}[\log(1 - D(\mathbf{A}, G(\mathbf{A}, \mathbf{z})))]$$

$$\mathcal{L}_1^{\text{latent}}(G, E) = \mathbb{E}_{\mathbf{A}\sim p(\mathbf{A}),\mathbf{z}\sim p(\mathbf{z})}||\mathbf{z} - E(G(\mathbf{A}, \mathbf{z}))||_1$$

Figura 7: *cLR-GAN*

Fonte: JUN-YAN ZHU; RICHARD ZHANG; DEEPAK PATHAK; TREVOR DARRELL; ALEXEI A. EFROS; OLIVER WANG; ELI SHECHTMAN, 2018.

2.1.8 CycleGAN

A arquitetura CycleGAN é uma rede neural generativa adversativa que não necessita de dados pareados para o seu treinamento, diferente das arquiteturas BycicleGAN e Pix2Pix para realizar transferência de estilo. A sua estrutura é formada por dois geradores e dois discriminadores que são utilizados para realizar uma translação de estilo de forma bidirecional.

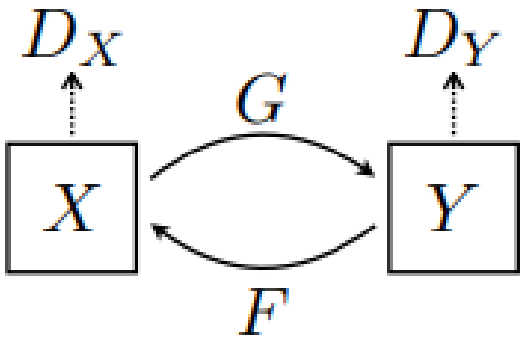


Figura 8: Fluxograma de uma arquitetura CycleGAN

Fonte: Zhu et al., 2017

O aprendizado inicia-se a partir do gerador G que translaciona o estilo da imagem X para a Y e é direcionado ao discriminador D_y e ao gerador G. O Gerador G utiliza da imagem falsa Y para transladar novamente para o domínio de estilo X e calcular o erro a partir do discriminador D_x e realizar os ajustes de pesos necessários. Por fim a imagem falsa do domínio Y é comparada com a imagem real do domínio Y no discriminador D_Y para se identificar se ambas são falsas ou reais dentro do domínio Y e assim calculando o erro L1 (ZHU et al., 2017).

2.1.9 Matriz GRAM

A Matriz Gram no contexto da transferência de estilo é um produto que corresponde a correlação entre cada filtro de características utilizada em uma arquitetura VGG-19. Esta técnica possui por objetivo extrair o estilo de imagens separando-os do seu conteúdo, desta forma, podem ser transferidas para um conteúdo distinto da sua original. O seu cálculo é ilustrado na imagem abaixo, cujas variáveis Nc, Nh e Nw correspondem respectivamente ao número do filtro, valores das variáveis de atributo horizontais e Nw valores das variáveis de atributo verticais de imagem.

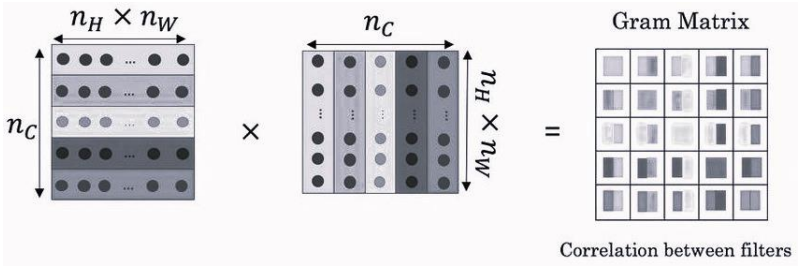


Figura 9: Matriz Gram

Fonte: HIEN; HUY; NGUYEN, 2021

Um aspecto de suma importância é que a utilização da Matriz Gram complementar a uma arquitetura VGG deve ser aplicada conjuntamente a um erro quadrado como perda para o treinamento, dado que diferente disto, as texturas tendem a aparecer como ruído.

2.1.10 UNIT (Unsupervised Image-To-Image Translation)

UNIT é uma arquitetura de translação de estilo não supervisionado que parte da premissa de que para transferir o estilo as imagens utilizadas para extração de conteúdo e estilo

dividam o mesmo espaço latente. Ou seja, a premissa de que imagens de diferentes domínios de estilo possam compartilhar um mesmo espaço latente z.

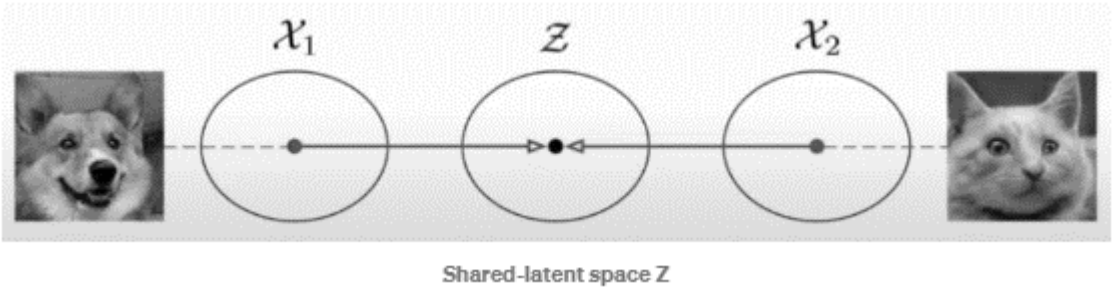


Figura 10: fluxograma de compartilhamento de domínio de espaço latente de uma UNIT

Fonte: SIK-HO TSANG, 2021

A sua arquitetura é constituída de um par de VAE { E,G } e por uma GAN {G, D}. As imagens x1 e x2 são codificas no espaço latente Z e decodificas a partir do gerador G de forma que os pesos de cada etapa entre as imagens são restringidas entre si, posteriormente são geradas novamente e discretizadas pelo bloco D, a fim de se obter o resultado final (LIU; BREUEL; KAUTZ, 2017).

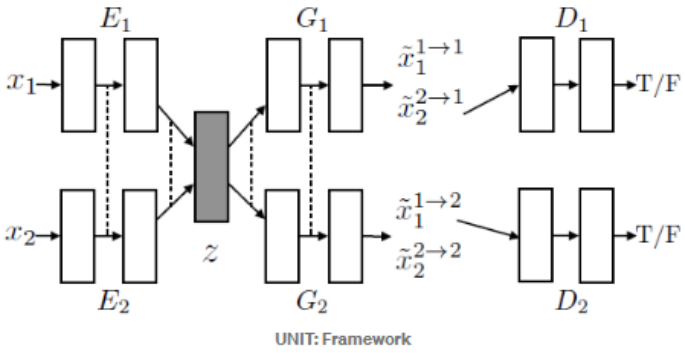


Figura 11: *arquitetura UNIT*

Fonte: MING-YU LIU; THOMAS BREUEL; JAN KAUTZ, 2018

O treinamento da VAE e GAN são realizadas de forma conjunta a partir da fórmula de perda:

$$\min_{E_1,E_2,G_1,G_2} \max_{D_1,D_2} \mathcal{L}_{\text{VAE}_1}(E_1,G_1) + \mathcal{L}_{\text{GAN}_1}(E_2,G_1,D_1) + \mathcal{L}_{\text{CC}_1}(E_1,G_1,E_2,G_2)$$
$$\mathcal{L}_{\text{VAE}_2}(E_2,G_2) + \mathcal{L}_{\text{GAN}_2}(E_1,G_2,D_2) + \mathcal{L}_{\text{CC}_2}(E_2,G_2,E_1,G_1).$$

2.1.11 MUNIT (Multimodal Unsupervised Image-to-Image Translation)

A arquitetura MUNIT foi estruturada pela NVIDIA com o objetivo de transladar estilos, partindo da premissa de que duas imagens distintas, após sua decomposição de conteúdo e estilo, dividam domínios de estilo diferentes mas conteúdos semelhantes. Para isto, a partir de uma distribuição gaussiana os domínios de estilos são cruzados para o domínio do conteúdo, como apresentado na imagem abaixo com as suas perdas no aprendizado de máquina associados (HUANG et al.;, 2018).

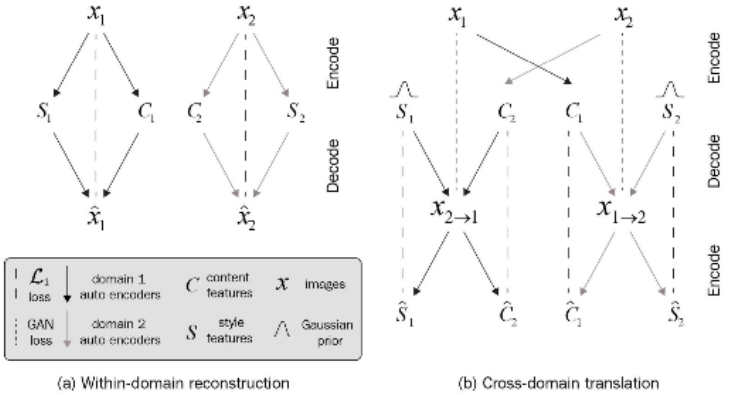


Figura 12: estrutura lógica MUNIT

Fonte: HUANG et al., 2018

Lloss:

$$\mathcal{L}_{\text{recon}}^{c_1} = \mathbb{E}_{c_1 \sim p(c_1), s_2 \sim q(s_2)}[\|E_2^c(G_2(c_1, s_2)) - c_1\|_1]$$
$$\mathcal{L}_{\text{recon}}^{s_2} = \mathbb{E}_{c_1 \sim p(c_1), s_2 \sim q(s_2)}[\|E_2^s(G_2(c_1, s_2)) - s_2\|_1]$$

Ganloss:

$$\mathcal{L}_{\text{GAN}}^{x_2} = \mathbb{E}_{c_1 \sim p(c_1), s_2 \sim q(s_2)}[\log(1 - D_2(G_2(c_1, s_2)))] + \mathbb{E}_{x_2 \sim p(x_2)}[\log D_2(x_2)]$$

Total Loss:

$$\min_{E_1, E_2, G_1, G_2} \max_{D_1, D_2} \mathcal{L}(E_1, E_2, G_1, G_2, D_1, D_2) = \mathcal{L}_{\text{GAN}}^{x_1} + \mathcal{L}_{\text{GAN}}^{x_2} +$$
$$\lambda_x(\mathcal{L}_{\text{recon}}^{x_1} + \mathcal{L}_{\text{recon}}^{x_2}) + \lambda_c(\mathcal{L}_{\text{recon}}^{c_1} + \mathcal{L}_{\text{recon}}^{c_2}) + \lambda_s(\mathcal{L}_{\text{recon}}^{s_1} + \mathcal{L}_{\text{recon}}^{s_2})$$

$$\mathrm{AdaIN}(z, \gamma, \beta) = \gamma \left(\frac{z - \mu(z)}{\sigma(z)} \right) + \beta$$

Adicionalmente a arquitetura do MUNIT também pode ser observada no esquema abaixo com as suas estruturas lógicas.

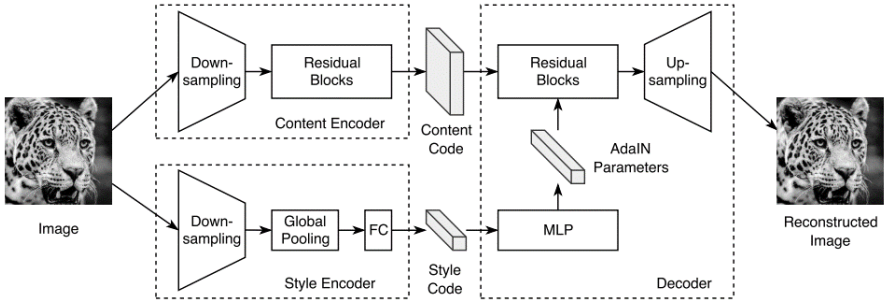


Figura 13: arquitetura MUNIT

Fonte: HUANG et al., 2018

2.1.12 FUNIT

A arquitetura FUNIT surgiu com o objetivo de preencher algumas necessidades de projeto que apareceram em modelos anteriores. Modelos como MUNIT não possuem a aptidão de generalização, ou seja, a patir dos dados de treinamento o algoritmo não consegue gerar novos resultados a partir de uma nova classe distinta de dados. Dessa forma, em busca de uma generalização no processamento de novos dados o modelo FUNIT treina uma base de dados separadas em classes para realizar a translação e assim, quando fornecido dados de uma nova classe antes não apresentada, ainda há resultados sólidos como apresentado o fluxograma da figura 12: estrutura lógica FUNIT (LIU et al., 2019).

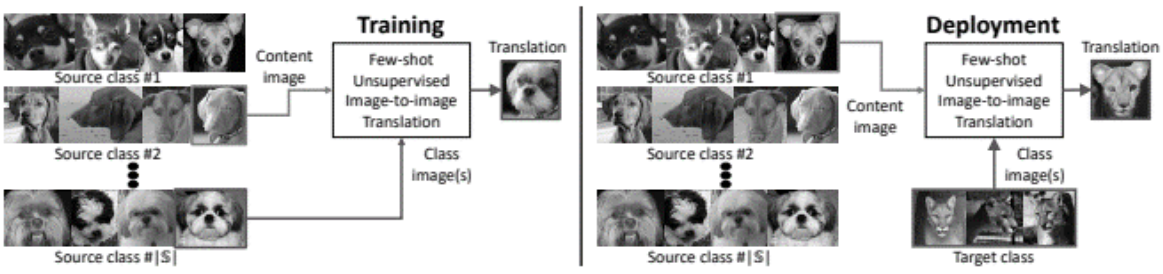


Figura 14: estrutura lógica FUNIT

Fonte: XUN HUANG et al, 2019

A arquitetura FUNIT pode ser exemplificada abaixo em um teste com diferentes tipos de animais para domínios de estilo e um para o domínio de conteúdo. Como resultado podemos observar um animal de mesma posição que o domínio de conteúdo em um estilo de um segundo domínio, no caso utilizado para transladar o estilo.

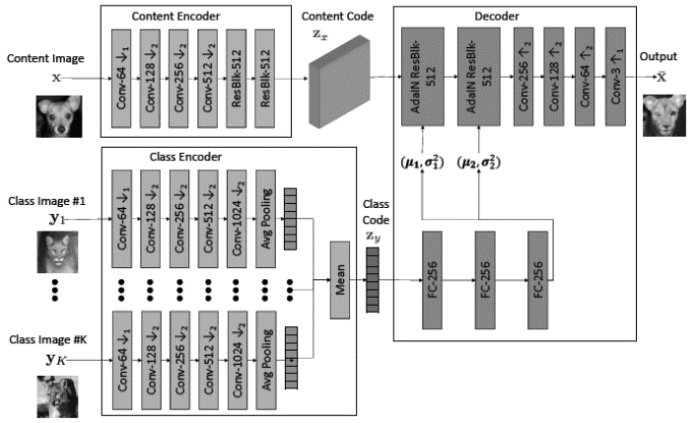


Figura 15: arquitetura FUNIT

Fonte: BARRAS; CHASSOT; SILVA, 2021

2.2 Estado da arte

A tecnologia de redes adversárias generativas nas translação de estilo vem ganhando espaço e cada vez mais veem se tornando foco de pesquisas de inúmeras empresas e universidades. Esta ambientação impulsionou resultados no campo da arte e design de produto. Entretanto, apesar de inúmeros uma arquitetura já comparadas entre si no quesito de multimodalidade e qualidade para alguns estudos de caso, ainda não foram essencialmente analisadas na dimensão de produção no mercado de arte em escala, centro principal deste trabalho de conclusão de curso.

3. METODOLOGIA E PROPOSTA

3.1 Metodologia

Dado ser o primeiro projeto que se é utilizado da tecnologia de processamento de imagens e redes generativas adversativas, a abordagem de forma exploratória se tornou a mais óbvia a ser a utilizada a fim de se ganhar eficiência e de se maximizar o aprendizado sobre o assunto, que é um dos principais objetivos citados inicialmente. Para isto, foi seguido uma metodologia de diamante duplo para atacar o problema.

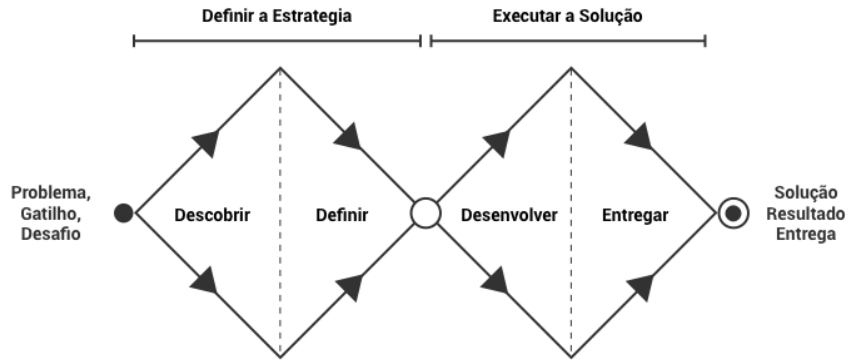


Figura 16: Diagrama Diamante Duplo

Fonte: HENRIQUE CARVALHO, 2019

Na etapa de descobrimento foi utilizado da Internet para se entender todas as atuais tecnologias que poderiam resolver a problemática proposta. A partir desta etapa, foi encontrada algumas soluções e conteúdos que poderiam direcionar o aprendizado da temática, sendo a escolhida a ser seguida como via principal o livro Hands-On Image Generation with TensorFlow: a pratical guide to generating images and videos using deep learning do autor Soon Yau Cheong. Este livro passa pela evolução dos modelos de forma temporal, com os seus elementos positivos e negativos para cada uma das abordagens de problema.

Desta forma, a partir desta diretriz foi analisado diferentes arquiteturas a fim de se chegar a conclusão da que possui melhores resultados dentro de uma realidade de recursos limitados, sejam eles de mão de obra, temporais ou de processamento. Ao longo desta diretriz foi testado algumas arquiteturas a fim de se entender, além de se obter conhecimento, os resultados produzidos. Escolhido o modelo final, a sua arquitetura foi executada, testada e adaptada para se chegar no estado de produto final.

3.2 Identificação do problema

A grande dificuldade de projeto foi encontrar um modelo que solucionasse o problema proposto e ao mesmo tempo não ultrapassasse os limites de recursos existentes. Durante este processo de descoberta, foi mapeado possíveis arquiteturas que poderiam ser utilizadas no papel da translação de estilo e algumas escolhidas a serem testadas a fim de se entender se os seus resultados solucionariam o objetivo proposto. Para esta seleção, foi interseccionado as possíveis soluções e os requisitos de projeto imaginados inicialmente, desta forma, como resultado, a matriz a seguir foi desenhada.

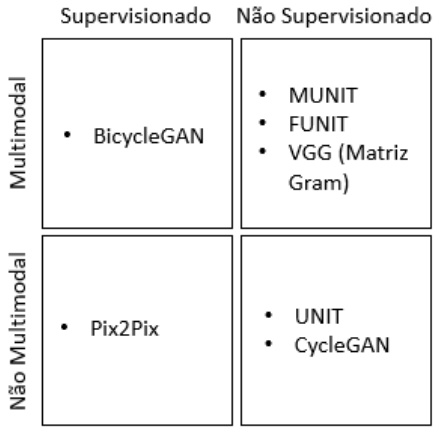


Figura 17: Classificação de Arquiteturas

Com base nas características de cada arquitetura mapeada foram escolhidas as arquiteturas que se apresentavam não supervisionadas e multimodais, ou seja, soluções que não necessitam de dados previamente classificados e que possuem resultados com maior variabilidade possível, assim preenchendo os requisitos de projeto. Nos tópicos a seguir foram apresentados os modelos testados de forma cronológica a as suas respectivas conclusões a partir dos seus resultados, no processo de escolha final de arquitetura que possui por fim a produção de merchandise em escala.

De forma paralela, um aspecto de enorme importância é que estudos anteriores já compararam parcela das arquiteturas sendo apenas a arquitetura BicycleGAN, quando testada em adereços de moda, de melhor qualidade que as arquiteturas não supervisionadas e multimodais testadas. Desta forma, apesar dos melhores resultados, esta arquitetura não se encaixa nos requisitos de projeto apresentados inicialmente, apesar de também multimodal.

	edges → shoes		edges → handbags	
	Quality	Diversity	Quality	Diversity
UNIT [15]	37.4%	0.011	37.3%	0.023
CycleGAN [8]	36.0%	0.010	40.8%	0.012
CycleGAN* [8] with noise	29.5%	0.016	45.1%	0.011
MUNIT w/o $\mathcal{L}^x_{\text{recon}}$	6.0%	0.213	29.0%	0.191
MUNIT w/o $\mathcal{L}^c_{\text{recon}}$	20.7%	0.172	9.3%	0.185
MUNIT w/o $\mathcal{L}^s_{\text{recon}}$	28.6%	0.070	24.6%	0.139
MUNIT	50.0%	0.109	50.0%	0.175
BicycleGAN [11]†	56.7%	0.104	51.2%	0.140
Real data	N/A	0.293	N/A	0.371

† Trained with paired supervision.

Quantitative evaluation on edges → shoes/handbags

Tabela 1: Comparativo de arquiteturas de translação de estilo

Fonte: HUANG et al., 2018

3.3 Proposta

Ao passo que os estudos de soluções viáveis para o problema proposto foi sendo realizado, também foi se entendendo o escopo a ser realizado para o projeto. Este fim se caracteriza por testar as tecnologias dentro do escopo dos requisitos de projeto, no caso uma rede neural GAN atrelada a Matriz Gram, arquitetura MUNIT e FUNIT, com o objetivo de se entender se elas se mostram viáveis para a produção de merchandise em escala.

No tópico de projeto as tecnologias são executadas em ambientes controlados ou de projetos paralelos a fim de serem comparadas para uma definição de uma arquitetura de melhor uso para a finalidade em alvo deste projeto de conclusão de curso.

4. O PROJETO

4.1 Transferência neural de estilo

A partir da técnica do uso da Matriz Gram foi extraído o conteúdo de uma imagem, no caso de uma camiseta, e o estilo de duas imagens, a primeira um quadro do Salvador Dalí e o segundo uma logo marca da Budweiser. Se foi realizado dois testes a fim de se analisar se formatos escritos interfeririam na textura ou seriam projetados de forma integral. O código utilizado foi embasado no apresentado no capitulo cinco do livro Hands-On Image Generation with TensorFlow: a pratical guide to generating images and videos using deep learning, este

pode ser encontrado no endereço https://github.com/PacktPublishing/Hands-On-Image-Generation-with-TensorFlow-2.0/blob/master/Chapter05/ch5_neural_style_transfer.ipynb.

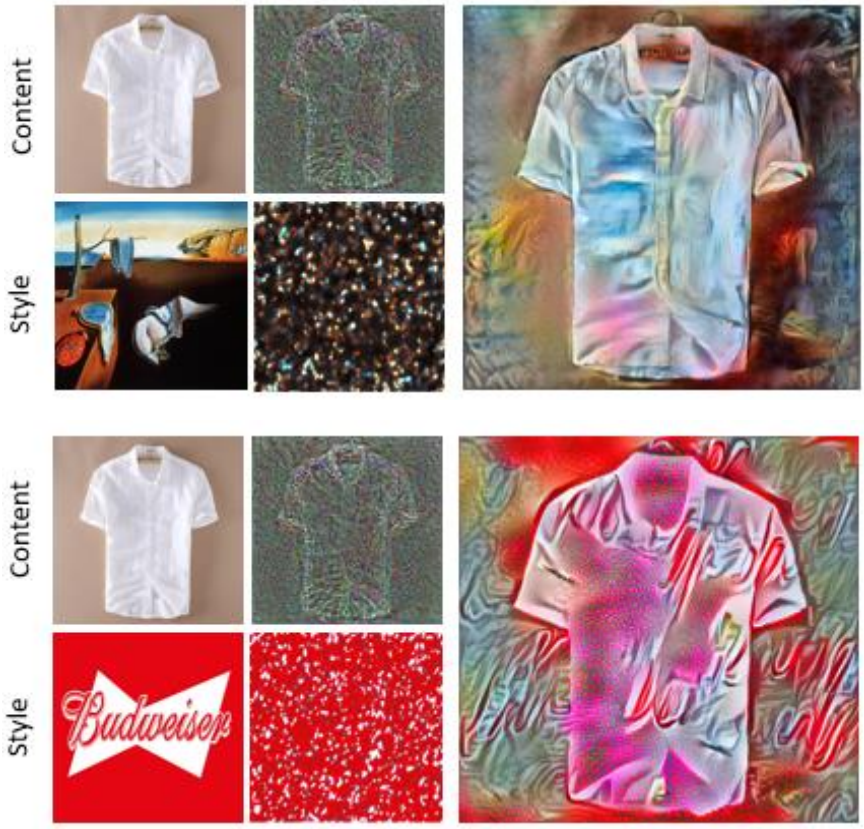


Figura 18: Resultados GAN (Matriz Gram)

Na utilização de tecnologia apresentada é visível que não conseguimos obter resultados expressivos de novos merchandises. Esta conclusão vem do fato de que a técnica utilizada não nos possibilita o controle espacial, de textura e de cor, ela simplesmente projeta todos estes elementos diretamente no conteúdo extraído. Entretanto, esta experiência se mostrou de enorme importância dado que os seus conceitos são utilizados em arquiteturas mais complexas e de resultados mais expressivos, como a arquitetura Multimodal Unsupervised Image-to-Image Translation.

4.2 MUNIT

Para a arquitetura ser testada foi utilizado o esqueleto de código encontrado no repositorio guithub https://github.com/eriklindernoren/PyTorch-GAN com algumas alterações em seu corpo de código. Ao longo da sua implementação, a partir da base de dados edges2shoes, a sua execução apresentou um ETA acima de quatrocentos dias a partir dos

recursos de processaemento disponíveis, desta forma foi utilizado os resultados a partir da execução do código em seu corpo original realizadas pelo autor sa fim dos seus resultados serem analisados para o propósito de produção de merchandise em escala.

Figura 19: Resultados MUNIT

Fonte:< https://github.com/eriklindernoren/PyTorch-GAN>

Os resultados obtidos pelo autor são notáveis no quesito de borda e multimodalidade, porém é valido ressaltar a sua necessidade de paridade entre dados, no caso de conteúdo e estilo. Paralelamente, parte da premissa que as imagens a serem transladadas possuem conteúdos semelhantes, apesar de domínios de estilo distintos, prejudicando o potencial de escalabilidade, requisito de projeto necessário.

4.3 FUNIT

Ao longo do projeto de conclusão de curso, a fim de se executar testes relativos a produção de merchandise em uma arquitetura FUNIT, foi evidênciado a necessidade de um processador NVIDIA DGX1 (8-V100, 32GB) para o tempo de processamento se apresentar viável. Como não houve o encontro de outras opções semelhantes ou disponíveis, as conclusões relativas a arquitetura em questão foram feitas a partir de resultados de experimento do próprio paper.

	Setting	Top1-all ↑	Top5-all ↑	Top1-test ↑	Top5-test ↑	DIPD ↓	IS-all ↑	IS-test ↑	mFID ↓
Animal Faces	CycleGAN-Unfair-20	28.97	47.88	38.32	71.82	1.615	10.48	7.43	197.13
	UNIT-Unfair-20	22.78	43.55	35.73	70.89	1.504	12.14	6.86	197.13
	MUNIT-Unfair-20	38.61	62.94	53.90	84.00	1.700	10.20	7.59	158.93
	StarGAN-Unfair-1	2.56	10.50	9.07	32.55	1.311	10.49	5.17	201.58
	StarGAN-Unfair-5	12.99	35.56	25.40	60.64	1.514	7.46	6.10	204.05
	StarGAN-Unfair-10	20.26	45.51	30.26	68.78	1.559	7.39	5.83	208.60
	StarGAN-Unfair-15	20.47	46.46	34.90	71.11	1.558	7.20	5.58	204.13
	StarGAN-Unfair-20	24.71	48.92	35.23	73.75	1.549	8.57	6.21	198.07
	StarGAN-Fair-1	0.56	3.46	4.41	20.03	1.368	7.83	3.71	228.74
	StarGAN-Fair-5	0.60	3.56	4.38	20.12	1.368	7.80	3.72	235.66
	StarGAN-Fair-10	0.60	3.40	4.30	20.00	1.368	7.84	3.71	241.77
	StarGAN-Fair-15	0.62	3.49	4.28	20.24	1.368	7.82	3.72	228.42
	StarGAN-Fair-20	0.62	3.45	4.41	20.00	1.368	7.83	3.72	228.57
	FUNIT-1	17.07	54.11	46.72	82.36	1.364	22.18	10.04	93.03
	FUNIT-5	33.29	78.19	68.68	96.05	1.320	22.56	13.33	70.24
	FUNIT-10	37.00	82.20	72.18	97.37	1.311	22.49	14.12	67.35
	FUNIT-15	38.83	83.57	73.45	97.77	1.308	22.41	14.55	66.58
	FUNIT-20	**39.10**	**84.39**	**73.69**	**97.96**	**1.307**	**22.54**	**14.82**	**66.14**
North American Birds	CycleGAN-Unfair-20	9.24	22.37	19.46	42.56	1.488	25.28	7.11	215.30
	UNIT-Unfair-20	7.01	18.31	16.66	37.14	1.417	28.28	7.57	203.83
	MUNIT-Unfair-20	23.12	41.41	38.76	62.71	1.656	24.76	9.66	198.55
	StarGAN-Unfair-1	0.92	3.83	3.98	13.73	1.491	14.80	4.10	266.26
	StarGAN-Unfair-5	2.54	8.94	8.82	23.98	1.574	13.84	4.21	270.12
	StarGAN-Unfair-10	4.26	13.28	12.03	32.02	1.571	15.03	4.09	278.94
	StarGAN-Unfair-15	3.70	11.74	12.90	31.62	1.509	18.61	5.25	252.80
	StarGAN-Unfair-20	5.38	16.02	13.95	33.96	1.544	18.94	5.24	260.04
	StarGAN-Fair-1	0.24	1.17	0.97	4.84	1.423	13.73	4.83	244.65
	StarGAN-Fair-5	0.22	1.07	1.00	4.86	1.423	13.72	4.82	244.40
	StarGAN-Fair-10	0.24	1.13	1.03	4.90	1.423	13.72	4.83	244.55
	StarGAN-Fair-15	0.23	1.05	1.04	4.90	1.423	13.72	4.81	244.80
	StarGAN-Fair-20	0.23	1.08	1.00	4.86	1.423	13.75	4.82	244.71
	FUNIT-1	11.17	34.38	30.86	60.19	1.342	67.17	17.16	113.53
	FUNIT-5	20.24	51.61	45.40	75.75	1.296	74.81	22.37	99.72
	FUNIT-10	22.45	54.89	48.24	77.66	1.289	75.40	23.60	98.75
	FUNIT-15	23.18	55.63	49.01	78.70	1.287	**76.44**	23.86	98.16
	FUNIT-20	**23.50**	**56.37**	**49.81**	**78.89**	**1.286**	76.42	**24.00**	**97.94**

Tabela 2: Comparação de Performance FUNIT

Fonte: BARRAS; CHASSOT; SILVA, 2021

A arquitetura FUNIT comparada a CycleGAN, UNIT e MUNIT se apresenta com melhores resultados em situações de poucos dados, mesmo em casos de maiores quantidades de classes utilizadas no treinamento. De forma adicional esta arquitetura possui vantagens dado que desempenha em ambientes de domínios de estilo não contempladas pela base de treinamento, se mostrando a melhor opção para produção de merchandise em escala.

5. CONCLUSÃO

Ao longo do projeto foi testado três arquiteturas que cumpriam os requisitos de projeto levantados, ser multimodal e não supervisionado. Como conclusão o uso de uma rede neural, conjuntamente a técnica de matrizx gram, se mostrou de difícil controle espacial, coloração e textura, de forma paralela a arquitetura MUNIT, apesar de não supervisionada ainda necessita de dados pareados para o seu treinamento e que as imagens dividam um domínio de conteúdo semelhante. Desta forma, a solução FUNIT se mostrou a mais adequada para a produção de merchandise em escala, apesar de não executado dado limitação de recursos de processamento, para isto foi criado uma arquitetura teórica como proposta para este fim.

Para o treinamento se é utilizado vestimentas de inúmeras marcas distintas como classes e uma dessas será utilizada para transladar o domínio de estilo, no caso da figura 19 as camisetas de marca Budsweiser.

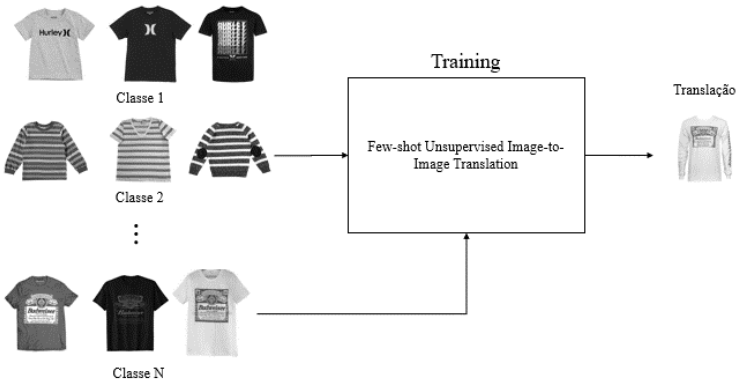


Figura 20: Treinamento FUNIT – Merchandise

Após o treinamento da rede neural, a sua execução é realizada a partir de um novo conjunto de images que será utilizada para o seu estilo ser transladado para o conteúdo aprendido. A figura 20 exemplifica este processo a partir da translação de camisetas da marca Red Bull.

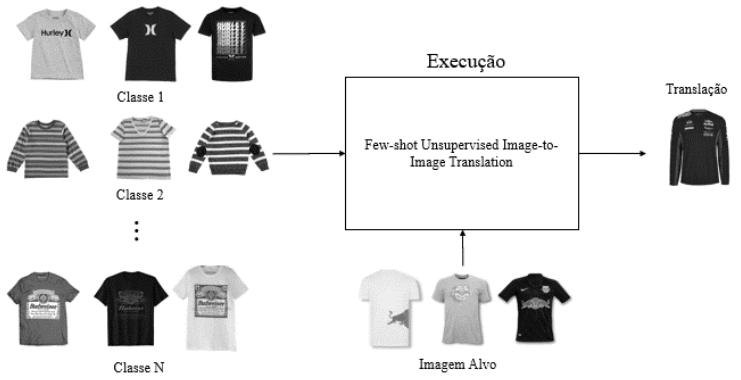


Figura 21: Execução FUNIT - Merchandise

Como resultado temos uma camiseta de manga comprida da marca de estilo transladada. Esta arquitetura e formato de treinamento fazem desta rede neural um ferramental com uma ampla flexibilidade para ser utilizada na produção de outras camisetas de diversas marcas a serem transladadas. Dessa forma, a produção de novas coleções de merchandise e arte se tornam escaláveis sem a necessidade de um agente intermediário para a criação de novas peças de design.

6. REFERÊNCIAS

A origem do termo . Ifd. Disponível em: https://www.ifd.com.br/design/a-origem-do-termo-design/?quad_cc/. Acesso em: 02/09/2022.

Introdução as Redes Neurais Convolucionais. Deeplearningbook. Disponível em: https://www.deeplearningbook.com.br/introducao-as-redes-neurais-convolucionais/ . Acesso em: 02/09/2022.

JUN-YAN ZHU; RICHARD ZHANG; DEEPAK PATHAK; TREVOR DARREL; ALEXEI A. EFROS; OLIVER WANG; ELI SHECHTMAN, Toward Multimodal Image-to-Image Translation, v.4, 2017.

JUN-YAN ZHU; TAESUNG PARK; PHILLIP ISOLA; ALEXEI A. EFROS, Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, v.1, 2017.

LUCAS BARRAS; SAMUEL CHASSOT; DANIEL FILIPE NUNES SILVA,
Unsupervised Image to Image Translation, 2021

MING-YU LIU; THOMAS BREUEL; JAN KAUTZ, Unsupervised Image-to-Image Translation Networks, v.1, 2017.

MING-YU LIU; XUN HUANG; ARUN MALLYA; TERO KARRAS; TIMO AILA; JAAKKO LEHTINEN; JAN KAUTZ, Few-shot Unsupervised Image-to-Image Translation, 2019.

OLAF RONNEBERGER; PHILIPP FISCHER; THOMAS BROX, U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015.

PHILLIP ISOLA; JUN-YAN ZHU; TINGHUI ZHOU; ALEXEI A. EFROS, v.1, Image-to-Image Translation with Conditional Adversarial Networks, 2016.

SOON YAU CHEONG, Hands-On Image Generation with TensorFlow: A practical guide to generating image and vides using deep learning, 2020.

Você sabe o que é machine learning? Entenda tudo sobre esta tecnologia. Canaltech. Disponível em: https://canaltech.com.br/inovacao/voce-sabe-o-que-e-machine-learning-entenda-tudo-sobre-esta-tecnologia-104100/. Acesso em: 02/09/2022

XUN HUANG; MING-YU LIU; SERGE BELONGIE; JAN KAUTZ, Multimodal Unsupervised Image-to-Image Translation, v.1, 2018.

3 áreas do desing que ainda têm muito para crescer - e como você pode se preparar para isso. Designerd.com. Disponível em: https://www.designerd.com.br/3-areas-do-design-que-ainda-tem-muito-para-crescer-e-como-voce-pode-se-preparar-para-isso/#:~:text=De%20acordo%20com%20a%20IBISWorld,1%2C45%20mil%20por%20segundo. Acesso em: 02/09/2022.