

CENTRO DE ESTUDOS E SISTEMAS AVANÇADOS DE RECIFE
Graduação em Ciência da Computação

SRGame - Dataset de super-resolução de imagens para jogos

Lucas Andrade Tenorio

Monografia de Graduação do Curso em Ciência da
Computação (CESAR School)

SECRETARIA DE GRADUAÇÃO DA CESAR-SCHOOL;

Data de Depósito:

Assinatura: _____

Lucas Andrade Tenorio

SRGame - Dataset de super-resolução de imagens para jogos

Monografia apresentada ao Programa de Graduação em Ciência da Computação do Centro de Estudos e Sistemas Avançados do Recife – CESAR School, como parte dos requisitos para obtenção do título de Graduando em Ciência da Computação.
EXEMPLAR DE DEFESA

Área de Concentração: Ciência da Computação

Orientador: Prof. Me. Eldrey Seolin Galindo

**CESAR School – Recife-PE
Novembro de 2022**

Lucas Andrade Tenorio

SRGame - Super-resolution image dataset for games

Graduation Monograph submitted to the Graduation Program in Computer Science of the Center for Advanced Job Studies and Systems – CESAR School, in partial fulfillment of the requirements for the degree of Graduate Graduate Program in Computer Science. *EXAMINATION BOARD PRESENTATION COPY*

Concentration Area: Computer Science

Advisor: Prof. Me. Eldrey Seolin Galindo

CESAR-School – Recife-PE
November 2022

*Este trabalho é dedicado aos Gamers
e todas as pessoas que são minimamente conectadas ao mundo Geek.*

AGRADECIMENTOS

- Gostaria de agradecer a minha família por todo apoio e paciência.
- Gostaria de agradecer minha profunda gratidão ao meu orientador Mestre Eldrey Galindo, pela oportunidade e confiança ao me aceitar como orientando e me apoiando, orientando e principalmente pela paciência ao longo deste trabalho.
- Gostaria de agradecer a todos os colaboradores que ajudaram na construção da base de imagens deste trabalho.
- Gostaria de agradecer a todos os membros da CESAR School School que trabalhão no desenvolvimento, na educação, qualificação e pesquisa, ajudando os alunos a ingressar no mercado.

*“Não brinque com o nerd de sua turma,
pois um dia ele pode ser seu chefe.”*
(Bill Gates)

RESUMO

TENORIO. **SRGame - Dataset de super-resolução de imagens para jogos.** 2022. 50 p. Monografia (Graduação em Ciência da Computação) – Ciências da Computação, CESAR School, Recife - PE, 2022.

O processamento de imagem usando técnicas de machine learning está se destacando cada vez mais, tanto na academia quanto no mercado, isso se deve ao seu sucesso e ao seu rápido desenvolvimento, assim revolucionando o desenvolvimento sobre super-resolução de imagens. Com isso, o crescimento atingiu também áreas como medicina, astronomia e na área de jogos. Contudo, especificamente na área de jogos, as redes recentes não disponibilizam o *dataset* para estudo e aplicação. Para isso, neste trabalho será apresentado a criação de um *dataset* voltado para super-resolução de imagens para jogos com o objetivo de ser disponibilizado publicamente, podendo ser utilizado em redes voltadas para super-resolução em jogos, com isso foram geradas mais de 2400 imagens entre 9 jogos diferentes.

Palavras-chave: Super-resolução, Aprendizagem profunda, Processamento de imagens, Jogos, Dataset.

ABSTRACT

TENORIO. **SRGame - Super-resolution image dataset for games.** 2022. 50 p. Monografia (Graduação em Ciência da Computação) – Ciências da Computação, CESAR School, Recife - PE, 2022.

Image processing using machine learning techniques is becoming increasingly prominent, both in academia and in the market, this is due to its success and rapid development, thus revolutionizing the development of super-resolution images. With that, fuel growth also in areas such as medicine, astronomy and in the games area. However, specifically in the area of games, recent networks do not make the dataset available for study and application. For this, in this work the creation of a dataset aimed at super-resolution of images for games will be presented and with that more than 2400 images were generated between 9 different games.

Keywords: Super resolution, Deep Learning, Image Processing, Games, Dataset.

LISTA DE ILUSTRAÇÕES

Figura 1 – Doom de 1993, onde mostra o upscale das texturas do lado direito em comparação com o esquerdo	24
Figura 2 – Escala de alta resolução de imagens	28
Figura 3 – Imagem do <i>dataset</i> do The Elder Scrolls V: Skyrim em tamanho original (1920x1080p)	32
Figura 5 – Imagem do jogo <i>Skyrim</i> em resolução original (1920x1080).	39
Figura 7 – Imagem do jogo <i>Dead Cells</i> em resolução original (1920x1080).	40

LISTA DE TABELAS

Tabela 1 – Lista do Top 10 jogos mais jogados.	33
Tabela 2 – Dados obtidos com os datasets.	37
Tabela 3 – Resultados obtidos por PSNR / SSIM.	38

LISTA DE ABREVIATURAS E SIGLAS

CNN	<i>Convolutional Neural Network</i>
COD	<i>Call of Duty Warzone</i>
CS	<i>Counter Strike</i>
DL	<i>Deep Learning</i>
DLDSR	<i>Deep Learning Dynamic Super-Resolution</i>
EDSR	<i>Enhanced Deep Residual Networks for Single Image Super-Resolution</i>
EESR	<i>Edge Enhance super-resolution</i>
GPU	<i>Graphics processing unit</i>
LoL	<i>League of Legends</i>
PSNR	Peak Signal Noise Ratio
PUBG	<i>Player Unknown Battlegrounds</i>
RCAN	<i>Residual Channel Attention Network</i>
SISR	Single image super-resolution
SR	super-resolução
SSIM	Structural Similarity Index

SUMÁRIO

1	INTRODUÇÃO	23
1.1	Base de dados	25
1.2	Objetivo Geral	25
1.3	Objetivo Específico	25
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	Processamento de imagens	27
2.1.1	<i>Resolução</i>	27
2.1.2	<i>Interpolação</i>	28
2.1.3	<i>Bicubic</i>	28
2.1.4	<i>Blur</i>	29
2.1.5	<i>Sal e pimenta</i>	29
2.2	Dataset	29
2.2.1	<i>Div2k e Urban100</i>	30
3	METODOLOGIA	31
3.1	Coleta da base	31
3.1.1	<i>Coleta da Atronach Detector Data Base</i>	31
3.1.2	<i>Coleta de imagens de usuários</i>	32
3.2	Downscale	32
3.3	Upscale	34
3.4	Metricas Avaliativas	34
3.5	Criação do dataset	35
4	RESULTADOS	37
5	CONCLUSÃO	41
	REFERÊNCIAS	43
ANEXO A	FORMULÁRIO DO GOOGLE FORMS	47
ANEXO B	FOLDER TREE	49



INTRODUÇÃO

Processamento de imagens utilizando técnicas de aprendizagem profunda, ou no inglês *Deep Learning* (DL), vem se destacando no âmbito acadêmico e de mercado, devido ao seu sucesso e seu rápido desenvolvimento. Atualmente existem duas vertentes sobre criação de redes de DL, elas podem ser criadas com um propósito geral onde a rede busca a generalização ou para um propósito específico onde a rede é focada em objetivo único. Existem vários tópicos que podem ser abordados dentro dessa área, como: classificação de imagens (CAI; GAO; ZHAO, 2020); reconhecimento facial (WANG; DENG, 2021); análise de imagens do setor médico (CAI; GAO; ZHAO, 2020); verificação de impressão digital (SHAO *et al.*, 2018); reconhecimento de padrões das pessoas (BAI *et al.*, 2021); mosaico Automático; Comprimir imagem e/ou vídeo (MA *et al.*, 2020); super-resolução de imagens (GALINDO; PEDRINI, 2019; WATSON, 2020a; TAI; YANG; LIU, 2017) e entre outros. A super-resolução (SR), consiste em reconstruir imagens em alta resolução oriundas de imagens com baixa resolução, tentando sempre minimizar os ruídos e borrões gerados, diminuir o desbotamento das cores ou até tentar melhorar a nitidez dos objetos da imagem em baixa resolução (GALINDO; PEDRINI, 2019). Diante disso, existem duas vertentes de SR, uma denominada de *multi super-resolution* (XIAO *et al.*, 2020), que consiste na SR com base em várias imagens e a outra denominada *single super-resolution*, que consiste na aplicação de SR com base em uma única imagem (GALINDO; PEDRINI, 2019).

Ao longo do tempo algumas redes convolucionais foram desenvolvidas para gerar imagens em SR tais como: *Edge Enhance super-resolution* (EESR) (GALINDO; PEDRINI, 2019), *Residual Channel Attention Network* (RCAN) (ZHANG *et al.*, 2018), *Enhanced Deep Residual Networks for Single Image Super-Resolution* (EDSR) (LIM *et al.*, 2017) e várias outras redes, essas redes tem como o objetivo serem genéricas, ou seja, converter qualquer imagem em baixa resolução para alta resolução. Com o decorrer do tempo, SR vem ganhando cada vez mais espaço no mercado de diferentes formas, sejam elas na parte biomédica com estudos voltados para reconhecimento de digitais, íris, movimentos, face e etc (NGUYEN *et al.*, 2013), ou sejam voltadas para a parte de multimídia, em vídeos ou imagens (LI *et al.*, 2020). A área de multimídia



Figura 1 – Doom de 1993, onde mostra o upscale das texturas do lado direito em comparação com o esquerdo

vem ganhando muito espaço, em especial junto ao mercado de jogos, que está fazendo muito sucesso com rápido desenvolvimento tecnológico. Em 2021 foram desenvolvidos jogos com poder gráfico muito avançado, ao compararmos os jogos entre os dessa época com os de 1996, geração de console do nintendo 64 ou popularmente conhecido como nintendinho, fica mais evidente ainda esse avanço tecnológico, por exemplo, a figura 1 possui duas versões de várias imagens, onde a versão do lado esquerdo é a imagem do jogo original e a do lado direito é uma imagem após a utilização de uma rede de super-resolução, é notável que a textura do lado direito está mais limpa e aparenta estar menos pixelada.

Esse avanço tecnológico vem muito por parte das empresas que produzem *Graphics processing unit* (GPU) voltadas para jogos, como: NVIDIA, AMD e Microsoft, com o intuito de melhorar o desempenho dos jogos e com isso a SR entrou nesse mercado. Elas já desenvolvem algumas soluções para vários jogos com otimização ruim ou que possuem dificuldade de processar na máquina devido à alta qualidade gráfica. Dentro do mercado de jogos, virou uma tendência jogos *online*, que vem crescendo bastante (GOMEZ, 2022), por exemplo: *League of Legends* (LoL); *Counter Strike* (CS); *Dota2*; *Player Unknown Battlegrounds* (PUBG); *APEX Legends*; *Fortnite*; *Call of Duty Warzone* (COD); e dentre outros, para isso é necessário um computador que precise rodar os jogos sem que ocorra uma lentidão na renderização para não impactar o jogo negativamente. Por conta disso muitos empresas estão trabalhando com Redes Neurais Convolucionais, do inglês *Convolutional Neural Network* (CNN), voltadas para SR em múltiplas imagens ou frames de vídeo, por exemplo: Atualmente a NVIDIA, pioneira nessa tecnologia, se utiliza de uma rede de Deep Learning, denominada de “*Deep Learning Dynamic*

Super-Resolution (DLDSR)”, para processamento dos frames e geração de imagens em SR, onde utiliza como entrada para sua rede convolucional, frames em baixa resolução para produzir uma versão em alta resolução ([WATSON, 2020b](#)).

1.1 Base de dados

Single image super-resolution (SISR) para imagens de jogos é uma área em desenvolvimento e que tangibiliza as áreas de interesse de grandes empresas, por este motivo, muitas das descobertas nessa área são consideradas segredo de mercado como citado no artigo de A. Watson “*Deep Learning Techniques for Super-Resolution in Video Games*”, onde mesmo a NVIDIA e AMD tendo implantado suas versões nas placas de vídeos mais novas, ainda não disponibilizaram a arquitetura ou seu funcionamento. Atualmente os *datasets* de imagens que treinam essas redes ainda são um mistério e quando procurados na internet são encontradas poucas bases e aquelas sendo encontradas não tem a mesma qualidade que as bases mais tradicionais de SR, exemplo: DIV2K ([AGUSTSSON; TIMOFTE, 2017](#)) Set5 ([BEVILACQUA *et al.*, 2012](#)) e Urban100 ([HUANG; SINGH; AHUJA, 2015](#)). Atualmente é possível encontrar uma base de imagens de SR do jogo *Elder Scrolls V: Skyrim* e duas bases de vídeos para SR, uma do jogo *Quake* e outra do jogo *Doom*, todos os *datasets* serão mais comentados no tópico 3. Com isso o intuito desse trabalho é voltado para gerar um *dataset* de SR com imagens únicas para jogos.

1.2 Objetivo Geral

Criar um *dataset* de SR específico de jogos, visando utilização para treinamento e estudos na área da SR para jogos.

1.3 Objetivo Específico

1. Criar um *dataset* de SR específico de jogos.
2. Aumentar a quantidade de *Low Resolution Images* com várias métricas.
3. Utilizar o upscale com bicubic para gerar uma base de comparação das métricas PSNR e SSIM.



FUNDAMENTAÇÃO TEÓRICA

Será descrito neste capítulo quais assuntos são importantes para nosso trabalho, mais especificamente, uma breve descrição sobre inteligência artificial, aprendizagem profunda, redes neurais convencionais, processamento de imagens e datasets.

2.1 Processamento de imagens

Ao longo do tempo, com o advento dos computadores e da tecnologia, é possível criar imagens oriundas de outras, graças a isso foi criado uma área para estudo e aplicações desses casos, essa área é denominada de Processamento de imagens ou Processamento digital de imagens.

O processamento de imagem é qualquer forma de processamento de sinal para a qual a entrada é uma imagem. O processamento digital de imagens é o estudo da representação e manipulação de imagens utilizando um computador, com isso é possível visualizar, analisar e processar essas imagens em muitos tipos de dados ([KOUR et al., 2012](#)). Processamento de imagens pode ser dividido em vários tópicos, porém para desenvolvimento e construção desse trabalho será citado focado em conceitos de Resolução.

2.1.1 *Resolução*

Para o conceito de resolução as imagens elas são divididas em uma matriz bidimensional, onde possui N linhas e M colunas e o ponto gerado por essas coordenadas é conhecido como pixel. Portanto, para determinar a resolução de uma imagem digital é o número de pixels representados no número de linhas vezes o número de colunas, ou seja, se uma imagem possui 1920 colunas e 1080 linhas, ela possui 1920 pixels para cada 1080 linhas. E quando nos referimos a resolução de imagem 2k e 4k, estamos dizendo que a imagem possui dois mil pixels em uma linha, imagens em 4K possui 4 mil pixels em uma linha e assim por diante, como representa a figura [2](#).

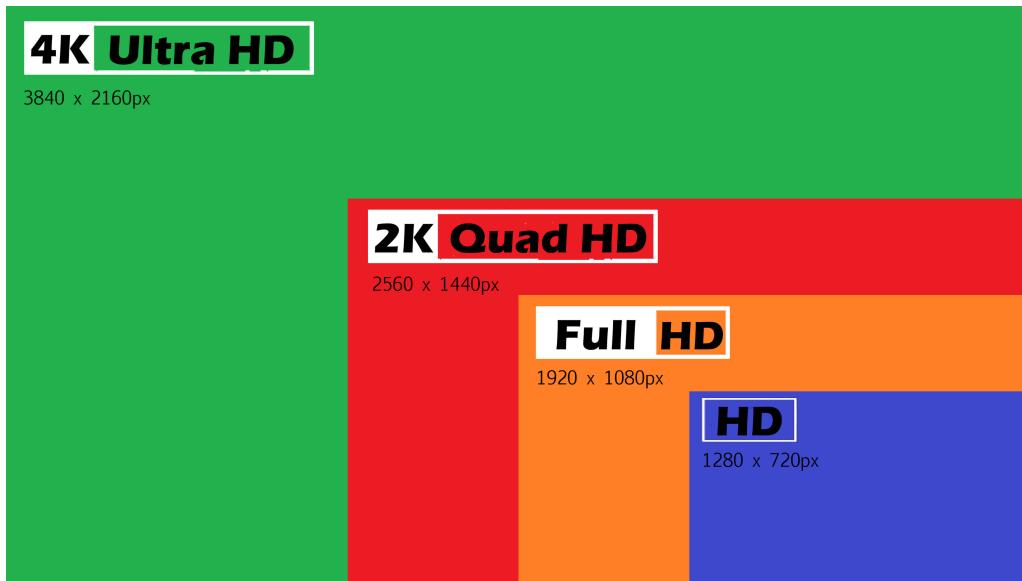


Figura 2 – Escala de alta resolução de imagens

2.1.2 *Interpolação*

A interpolação é um método matemático que permite estimar o valor de uma variável em um ponto intermediário de um conjunto de dados conhecidos. Isso é útil ao ter um conjunto de valores conhecidos, mas se deseja estimar o valor de uma variável em um ponto específico que não faz parte desse conjunto.

A ideia de utilizar interpolação em SR é usar os pixels da imagem original como pontos de referência e, em seguida, usar um método de interpolação para preencher os espaços entre esses pixels com novos pixels estimados. Isso permite aumentar a resolução da imagem sem adicionar ruído ou distorção.

2.1.3 *Bicubic*

O método *bicubic* é um método de interpolação que usa uma função cúbica para aproximar o valor de uma variável em um determinado ponto. Ele pode ser usado para aproximar valores de uma função de duas variáveis em um plano ou de uma função de três variáveis no espaço (LI; QI; WAN, 2019). A principal diferença entre o método bicúbico e outros métodos de interpolação, como a interpolação de vizinhos mais próximos, é que bicúbica usa uma função de três variáveis para estimar os valores de uma função de duas variáveis (uma imagem bidimensional). Por exemplo, se um pixel tem um valor de 50 e os pixels ao lado do pixel tem valores de 60 e 40, então o valor do pixel em uma imagem bicubica será de 50. No entanto, se o valor do pixel em uma imagem é de 60 e os pixels ao lado tem valores de 40 e 20, o valor do pixel na imagem bicubica será de 40.

2.1.4 Blur

Outro método que pode ser usado para *downscale* de imagens é o de *blur*, consistido em aplicar, antes da redução de dimensionalidade, um filtro de suavização que usa duas amostras de pixels vizinhos para estimar o valor do pixel no meio. Esse filtro possui diferentes algoritmos de implementação. A redução ocorre com o *Blur* seguido da função de degradação baseada na equação do desvio padrão.

Minimizar a perda de informação na redução de dimensionalidade é um dos principais motivos para o uso de técnicas de *blur* e *downscale*, principalmente o *blur* com o filtro de Gauss.

Por exemplo, considerando duas amostras de pixels vizinhos, A1 e A2, o valor do pixel no meio é estimado como: $A_1 + (A_2 - A_1) / 2$, onde A1 é o valor do pixel na posição (x1, y1) e A2 é o valor do pixel na posição (x2, y2) e (x1, y1) e (x2, y2) são as coordenadas dos pixels vizinhos. O valor do pixel no meio é estimado como a média ponderada dos valores dos pixels vizinhos. A1 é pesado com 0,5 e A2 é pesado com 0,5.

2.1.5 Sal e pimenta

O método *salt and pepper* é o método mais simples para ruído de imagens, esse método é bastante comum no âmbito de SR e pode ser simulado para validação. Esse método consiste em atribuir aos pixels valores exatos, ou seja, atribui 0 aos pixels que devem se tornar preto e 1 aos pixels que deveram ser brancos. Esse processo de atribuição é feito a cada pixel da imagem ou é feito de forma aleatória, ou seja, cada imagem pode possuir uma quantidade aleatória de pixels brancos e pretos ([FENGQING et al., 2013/11](#)).

2.2 Dataset

Com os avanços da inteligência artificial e com o surgimento da Aprendizagem Profunda, a construção de um conjunto de dados se tornou ainda mais importante, pois é esse conjunto que alimenta os modelos que estão cada vez mais exigindo quantidade e qualidade dos dados em um dataset ([GEBRU et al., 2018](#)).

Esse conjunto de dados também é conhecido como dataset. O conceito de dataset, de acordo com Chapman et al ([CHAPMAN et al., 2019](#)), é uma coleção de observações relacionadas organizadas para um propósito particular, por exemplo, um conjunto de dados pode conter imagens, gráficos ou documentos, além dos dados numéricos tradicionais, esse conceito será utilizado como referência para este trabalho. Porém, há uma grande discussão na comunidade científica com a relação ao conteúdo dentro dele e dependendo da comunidade em particular pode variar.

2.2.1 Div2k e Urban100

Existem diversos datasets para SR e alguns deles são grandes fonte de dados para treinamento de suas redes, por exemplo, o Div2k ([AGUSTSSON; TIMOFTE, 2017](#)) e Urban100([HUANG; SINGH; AHUJA, 2015](#)) estão presentes em diversos artigos de SR como: ([GALINDO; PEDRINI, 2019](#)), ([CHAPMAN *et al.*, 2019](#)), ([WANG *et al.*, 2018](#)), ([WAN *et al.*, 2020b](#)) e ([WAN *et al.*, 2020a](#)). O Div2k possui um total de 1000 imagens em resolução 2k, para criação do par de imagens em baixa resolução foi dividido imagens em 2x, 3x e 4x para duas métricas diferentes, Bicubic e um método desconhecido, totalizando 6 datasets em baixa resolução. O segundo dataset genérico que auxiliou a criação do nosso foi Urban100, ele possui um total de 100 imagens de cenários urbanos, a resolução deste dataset é de 512x322 em downscale de 2 vezes e sua imagem original possui 1024x644.



METODOLOGIA

Este trabalho tem o propósito de construir um *dataset* de SR para jogos, então foi feita a coleta de imagens de jogos com tamanhos iguais ou similares. Em seguida, foram aplicados os métodos de downscale e upscale para obter as imagens de menor e maior resolução. Para avaliar as imagens, foram utilizadas as métricas PSNR e SSIM.

Neste capítulo será explicado como foi feita a coleta de imagens para a construção do *dataset*, quais métodos e métricas foram utilizadas para o upscaling e downscaling das imagens e validação das imagens.

3.1 Coleta da base

Como citado na seção 1.1, não foram encontradas bases de imagens de jogos para SR, desta forma este trabalho buscou a construção de uma base de imagens de jogos através de duas formas de aquisição de imagens distintas, a primeira foi através da coleta de imagens de bases de jogos já existentes, mas foram construídas com outros propósitos como classificação de objetos, a segunda forma de coleta das imagens consistiu em um formulário de aquisição onde os voluntários poderiam enviar imagens de acordo com algumas especificações, essas duas formas serão melhor detalhadas nas subseções a seguir.

3.1.1 Coleta da Atronach Detector Data Base

A base de imagens Atronach Detector Data Base consiste em uma base de imagens do jogo *The Elder Scrolls V: Skyrim*, essa foi disponibilizada em um github e visa a classificação das imagens de acordo com características do jogo (magias e ações). Ele possui um total de 2400 imagens onde elas são divididas em 1200 imagens para treino, 600 imagens para teste e 600 imagens para validação. Como não existe um *dataset* com imagens em Low Resolution das imagens originais, foi utilizado algumas métricas para construção desse *dataset*, onde será



Figura 3 – Imagem do *dataset* do The Elder Scrolls V: Skyrim em tamanho original (1920x1080p)

comentado na seção 3.2. Um exemplo de imagem adquirida por este dataset é a figura 3, onde o tamanho original da imagem é 1920x1080 pixels.

3.1.2 Coleta de imagens de usuários

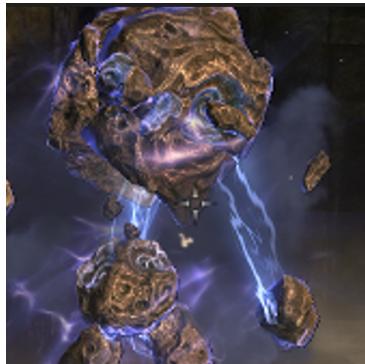
Visando construir uma base de imagens mais homogênea com relação aos jogos e as características gráficas deles, foi realizado uma aquisição de imagens através de um formulário do Google Forms, o formulário completo pode ser encontrado em anexo A, essa coleta durou 3 semanas. Entretanto, os voluntários possuíam algumas restrições para envio das imagens, foi recomendado que eles tirassem *screenshots* na resolução de 1920x1080 pixels, de jogos que eles estão jogando atualmente e que não possuíssem muito texto nas imagens. Após o fechamento do formulário foi adquirido um total de 253 imagens entre 8 jogos diferentes, onde alguns desses jogos estão na lista de jogo mais jogados de 2022 ([MOREIRA, 2022](#)), por exemplo, o *League of Legends*, representado na tabela 1.

3.2 Downscale

Como o *dataset* criado nesse trabalho possui apenas imagens em alta resolução, se faz necessária a criação de imagens em baixa resolução gerando assim um par de imagens, para isso, foi necessário utilizar um algoritmo de *downscale* nas imagens. A construção do algoritmo de *downscale* se baseou em dois *datasets* já conhecidos e amplamente utilizados na área de SR, mas que contém imagens genéricas, sendo eles o Div2k([AGUSTSSON; TIMOFTE, 2017](#)) e o Urban100 ([HUANG; SINGH; AHUJA, 2015](#)).

Jogos	Quantidade de Players simultaneos
Roblox	199 Milhões
Minecraft	140 Milhões
PUBG	100 Milhões
Call of Duty: Warzone	100 Milhões
Apex Legends	100 Milhões
Free fire	50 Milhões
Hearthstone	30 Milhões
Fortnite	12.3 Milhões
League of Legends	8 Milhões
Genshin Impact	8 Milhões

Tabela 1 – Lista do Top 10 jogos mais jogados.

(a) Recorte do jogo *Skyrim* 2 vezes reduzida pelo Bicubic(b) Recorte do jogo *Skyrim* 2 vezes reduzida pelo Bicubic + Blur(c) Recorte do jogo *Skyrim* 2 vezes reduzida pelo Bicubic + Blur

A base de dados Div2K foi aplicado o algoritmo Bicubic e Unknown para criação do *dataset* em low resolution a métrica utilizada pelo Urban100 ([HUANG; SINGH; AHUJA, 2015](#)) foi o borrão ou *blur* em inglês.

Diante disso, foi utilizado algumas dessas métricas para criação do nosso *dataset* em baixa resolução, a primeira métrica é o Bicubic apresentado por um corte do jogo *Skyrim* na figura 4a, a segunda é o Bicubic + *blur*, figura 4b e a terceira o Bicubic + Sal e Pimenta ([FENGQING et al., 2013/11](#)) apresentado na figura 4c, onde cada métrica possui imagens reduzidas em 2 vezes, 3 vezes e 4 vezes.

A utilização do método de interpolação bicubic, se deu por, além de sua utilização no Div2k, ser um método mais preciso que os métodos de interpolação lineares e quadráticos e também por sua alta velocidade de processamento ([LI; QI; WAN, 2019](#)). Contudo, este trabalho também utilizou o Urban100 e ele se caracterizou pela funcionalidade de *blur*, por isso a intenção de utilizar ao método bicubic + blur e com isso a junção das métricas utilizadas nos dois bancos, Div2k e Urban100. A última métrica utilizada foi a junção do método bicubic + sal e pimenta, devido a isso essa camada extra de sal e pimenta gera falhas na imagem, pixeis que não tem

informação, forçando as redes a não apenas aumentarem a imagem, mas aprenderam a completar pequenas "lacunas" ruído a mais nas imagens em baixa resolução.

3.3 Upscale

Após a redução das imagens do *dataset* original para criação de *datasets* em baixa resolução, foi necessária a criação de imagens em alta-resolução, permitindo assim a geração de uma base de métricas comparativas, utilizando como entrada as imagens em baixa resolução geradas. Para essa etapa, foi realizado em cada imagem um aumento de resolução passando como parâmetro de método apenas o *bicubic*. Após a realizado a etapa de *Upscale* nas imagens, haverá uma comparação da imagem criada com a imagem original, utilizando métricas onde será avaliado a qualidade da imagem. Esses métodos observarão ruídos que são gerados, luminância da imagem, contraste e a textura dela, o resultado gerado é um número que indica a semelhança da imagem original com a imagem criada. Isso é importante porque, ao comparar os resultados da imagem original com a imagem obtida, é possível avaliar a eficácia das técnicas de SR em termos de qualidade da imagem.

3.4 Metricas Avaliativas

O intuito deste trabalho para criação de um *dataset* de SR voltado para jogos pode ser estudado pela comunidade acadêmica, então é necessário garantir que os resultados comparativo das imagens ocorra e que seja disponibilizado. Diante disso, a validação é importante para garantir que elas apresentem boa qualidade e semelhança com as imagens originais, sendo comumente utilizado para avaliar as imagens o Peak Signal Noise Ratio (PSNR) e o Structural Similarity Index (SSIM).

O PSNR é um indicador de qualidade de imagem que mede a relação entre o sinal de ruído e o sinal de referência. É uma medida de similaridade entre duas imagens e é geralmente expressa em termos de decibéis (dB), quanto maior o PSNR, melhor a qualidade da imagem ([HORÉ; ZIOU, 2010](#)).

O cálculo do PSNR envolve a comparação de um pixel específico do sinal de ruído com o pixel correspondente do sinal de referência. A diferença entre os dois pixels é calculada e, em seguida, é aplicada uma função para converter essa diferença em um valor de dB.

$$\text{PSNR} = 20 \log_{10} (\max I_{(i,j)}) - 10 \log_{10} \quad (3.1)$$

Enquanto o SSIM (Structural Similarity Index) é um método de avaliação de imagens criado para medir a qualidade de uma imagem. Ele considera a luminância, o contraste e a textura das imagens para calcular um valor que representa a semelhança estrutural das imagens e varia

de 0 (imagens totalmente diferentes) a 1 (imagens idênticas). Ele também é robusto a alterações de compressão e de codificação, o que o torna um método de avaliação de imagens muito útil (HORÉ; ZIOU, 2010).

$$\text{SSIM}(\text{HR}, \text{SR}) = \frac{(2\mu_{\text{HR}}\mu_{\text{SR}} + C_1)(2\sigma_{\text{HS-R}} + C_2)}{(\mu_{\text{HR}}^2 + \mu_{\text{SR}}^2 + C_1)(\sigma_{\text{HR}}^2 + \sigma_{\text{SR}}^2 + C_2)} \quad (3.2)$$

3.5 Criação do dataset

Como citado nos tópicos 3.1.1 e 3.1.2, o *dataset* do Atronach Detector Data Base possui um total de 2400 imagens somente do jogo *The Elder Scrolls V: Skyrim* e o *dataset* coletado pelo formulário do Google possui 253 imagens de diversos jogos, onde dessas diversas imagens de jogos o jogo que mais obteve quantidade de imagens foi o *Genshin Impact* com 114, juntando os dois *dataset* totalizou 2653 imagens. Após a junção de todas as imagens um única pasta nomeada de *original_images*, foi realizado uma verificação de resolução, onde o padrão definido para a resolução de cada imagem deveria ser 1920x1080 pixels. Entretanto, foi notado que o total de imagens caiu para 2441, onde o *dataset* do *Skyrim* teve 141 imagens com resolução diferente do padrão e do *dataset* coletado foram retiradas 71 imagens por estarem fora do padrão, essas imagens foram salvas em uma pasta chamada *fake_image*, encontrado na árvore B.

Em seguida, foi efetuado o *downscale* da imagem, utilizando os métodos citados no tópico 3.2, tendo a finalidade de criar imagens em baixa resolução para serem utilizadas por redes de SR voltada para jogos. O processamento de imagens para alta resolução, referenciado no tópico 3.3, teve a finalidade de comparação utilizando as métricas avaliativas citadas no tópico 3.4.



RESULTADOS

A construção do *dataset* realizado neste trabalho teve como resultado um montante de 2441 imagens, todas em formato .PNG, onde 2259 imagens foram adquiridas do *dataset* de *Skyrim*, comentado mais detalhadamente no tópico 3.1.1 e 182 imagens do *dataset* coletado no tópico 3.1.2. Inicialmente o *dataset* do jogo *Skyrim* possuía 2400, porém devido a algumas imagens não estarem na resolução padrão definida para este trabalho (1920x1080) foram excluídas. Devido ao mesmo motivo de não estarem na mesma resolução, o *dataset* coletado inicialmente possuía 253 imagens sendo removidas 71 imagens, diminuindo para 182. Com isso, este *dataset* foi nomeado de SRGame, tendo como base do seu nome super-resolução e jogos. A tabela 2 demonstra a essa quantidade inicial e final de imagens a partir da coleta dos datasets, além de mostrar o aproveitamento das bases utilizadas para a construção do SRGame, o dataset final obteve um aproveitamento maior que 90%. Essas imagens resultantes foram armazenadas na pasta *original_images*, enquanto as que foram jogadas fora estão dentro da pasta *fake_images*.

Em seguida, foi efetuado o *downscale* da imagem, utilizando os métodos citados no tópico 3.2. Após o processamento das imagens, a organização da pasta de imagens em baixa resolução, denominada de *LR Images*, foram geradas 3 subpastas para cada método utilizado,

Dataset	Jogos	Imagens Inicial	Imagens Final	Aproveitamento
Atronach Detector DB	1	2400	2259	95%
Coleta com usuários	8	253	182	71%
SRGame	9	2653	2441	92%

Tabela 2 – Dados obtidos com os datasets.

Metodo	2x	3x	4x
Bicubic	34.9575 0.9682	31.1117 0.9378	28.8890 0.9069
Bicubic + Blur	29.7337 0.9075	28.1631 0.8744	26.9835 0.8480
Bicubic + Salt and Pepper	23.0012 0.7544	19.7189 0.6606	16.9640 0.5558

Tabela 3 – Resultados obtidos por PSNR / SSIM.

onde essas subpastas são a quantidade de vezes que a imagem foi reduzia, ou seja, a pasta *LR Images/2x* possui 2441 imagens em baixa resolução como método de *downscale* o *bicubic* e que essas imagens diminuíram sua resolução em 2 vezes, saindo de 1920x1080 para 960x540. A organização da pasta para o processamento de imagens para alta resolução é similar ao de baixa resolução, isso pode ser demonstrado na árvore [B](#), onde estão armazenadas na pasta *HR Images*.

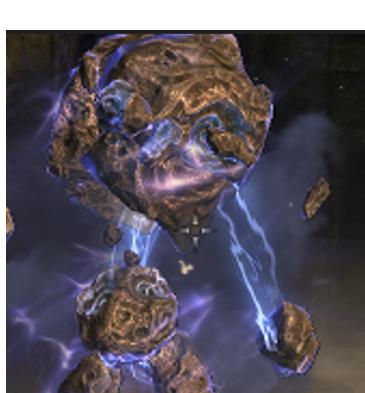
Com a criação do SRGame de SR em alta resolução é possível a utilização de métricas para avaliar a boa qualidade e semelhança com as imagens originais. Então no tópico [3.4](#) foram apresentados algumas métricas para avaliação e elas geraram resultados. A tabela [3](#) mostra esses resultados, sendo assim um comparativo dos métodos de *downscale*, utilizados no tópico [3.2](#), com a quantidade de vezes que a imagem foi reduzida.

Foi apresentado, conforme a tabela [3](#), que o método *bicubic* apresentou melhor resultado em comparação com os demais. Isto é devido a uma menor quantidade de ruídos gerados, pois como no *bicubic + blur* ainda existe um borrão que distorce a imagem, então o resultado será menor. Isso também ocorre com o método *bicubic + sal e pimenta* e nele ainda é pior em comparação com o de *blur*, pois como os pontos brancos e pretos são gerados na imagem é causado uma perda de informação da cor do pixel que estava no lugar, gerando assim um ruído ainda maior.

Quando comparamos os resultados entre a quantidade de vezes que a imagem foi reduzida, há uma grande diminuição dos resultados das métricas em todos os métodos, quando a imagem ela foi reduzida mais vezes. Entretanto, o resultado de mais relevância comparativa está entre as imagens da pasta *bicubic_sp/4x*, onde a perca de cores devido ao método sal e pimenta é enorme. Este resultado da pasta *bicubic_sp/4x* ocorreu provavelmente por causa da diminuição da imagem em 4 vezes, ou seja, diminui a resolução de 1920x1080 para 480x270 utilizando o método *bicubic*, resultando em uma grande redução na quantidade de pixels e quando aplicado o método sal e pimenta essa perda de pixels ainda é maior devido troca de pixels por preto e branco.



Figura 5 – Imagem do jogo *Skyrim* em resolução original (1920x1080).



(a) Ponteiro do jogo *Skyrim* 2 vezes reduzida pelo Bicubic



(b) Ponteiro do jogo *Skyrim* 2 vezes reduzida pelo Bicubic + Blur



(c) Ponteiro do jogo *Skyrim* 2 vezes reduzida pelo Bicubic + Blur

Analizando as imagens **6a**, **6b** e **6c** que estão duas vezes reduzida e utilizando os métodos de downscale *Bicubic*, *Bicubic + blur* e *Bicubic + sal e pimenta*, respectivamente, com a imagem original, figura 5, é notório que as imagens elas possuem um ruído gerado, principalmente quando se trata da imagem **6b**, onde degrada os raios do personagem, fazendo assim perder o brilho da imagem. Realizando um comparativo com a resolução das imagens após o *Downscale*, é notório que quanto maior a redução de dimensionalidade maior a quantidade de ruído que será produzido, por exemplo, barra de vida da imagem original do jogo *Dead Cells*, representado na figura 7 é nítido que a vida do personagem é 248, porém quando a resolução das imagens devido ao *downscale* vai ficando cada vez menor, esse numero vai ficando menos legível.

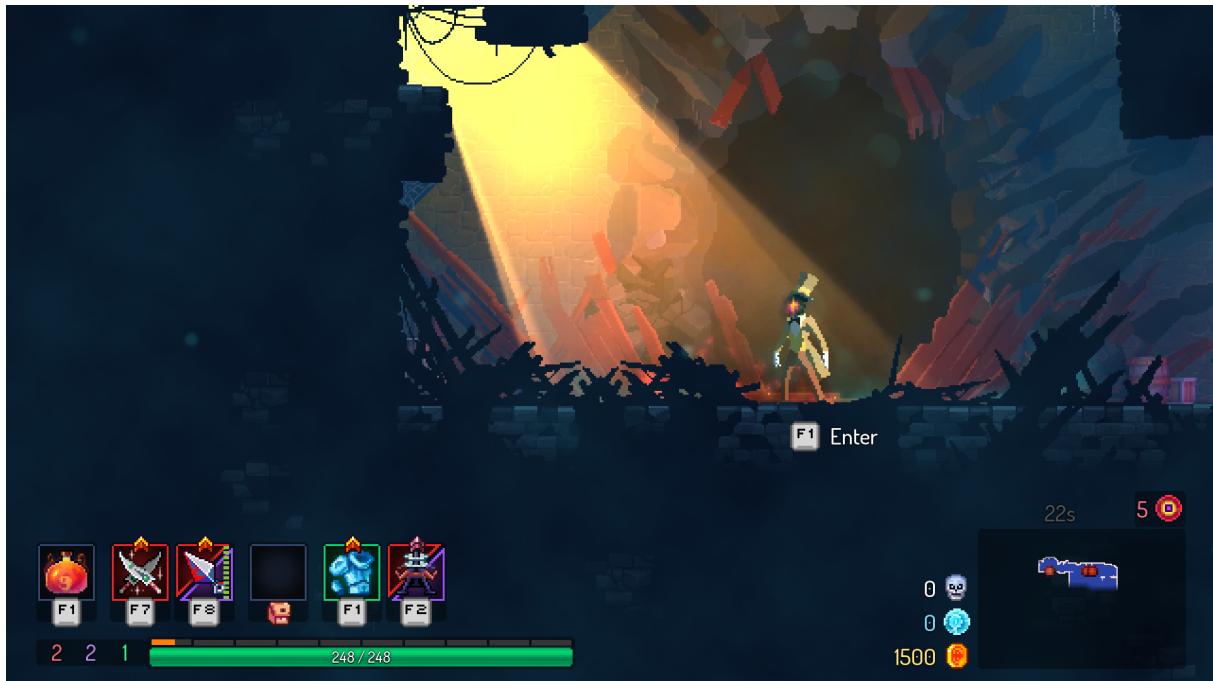
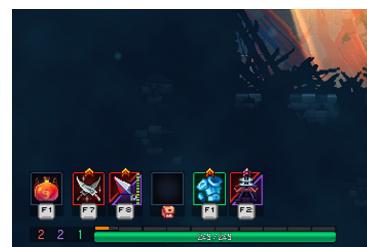


Figura 7 – Imagem do jogo *Dead Cells* em resolução original (1920x1080).



(a) Barra de vida do jogo *Dead Cells* em 2x



(b) Barra de vida do jogo *Dead Cells* em 3x



(c) Barra de vida do jogo *Dead Cells* em 4x



CONCLUSÃO

Super-resolução com imagens únicas é um problema típico na área de processamento de imagens, podendo afetar diversas áreas de conhecimento, como medicina e segurança. Com o avanço do processamento de imagens, SR também ganhou espaço e referência na área de entretenimento, mais precisamente na área de jogos, onde a NVIDIA é pioneira neste ramo. Entretanto, a aquisição de imagens que treinam essas redes ainda são um mistério e quando procurados na internet são encontradas poucas bases. O intuito desse trabalho é voltado para gerar um *dataset* de super-resolução com imagens únicas para jogos. Este trabalho resultou em uma grande base de dados adquirida de duas formas, uma foi através da coleta de imagens do jogo *The Elder Scrolls V: Skyrim*, mas ela foi construída com outros propósitos como classificação de objetos e a outra forma de coleta das imagens consistiu em um formulário de aquisição onde os voluntários poderiam enviar imagens de acordo com algumas especificações, obtendo um total de 253 imagens. Entretanto, como o dataset criado nesse trabalho possui apenas imagens em alta resolução, foi necessário a criação de imagens em baixa resolução. Diante disso, foi utilizado algumas dessas métricas para criação do dataset em baixa resolução, Bicubic, Bicubic + blur e o Bicubic + Sal e Pimenta, onde cada métrica possui imagens reduzidas em 2 vezes, 3 vezes e 4 vezes. Com os resultados obtidos, conseguimos fazer um dataset onde existem imagens com pouco ruído, normalmente imagens utilizando apenas o método *bicubic* e imagens com 2 vezes aumentada na resolução, também foi gerado imagens com muitos ruídos, foram utilizadas para construção dessas imagens o método *bicubic* + sal e pimenta. Portanto, o dataset possui os dois extremos, podendo fazer que as redes que irão utilizá-lo seja treinada para o melhor e pior caso.

Para planos futuros foi pensado em aumentar a quantidade de imagens do *dataset* e aumentar a quantidade de jogos que estão sendo mais jogados no mundo, balancear a quantidade de imagens dos jogos e depois utilizá-lo em alguma rede de SR. Entretanto, o ideal para utilizar uma rede de SR seria utilizar uma específica para jogos, portanto foi pensado em criar uma rede de SR voltada para jogos.

REFERÊNCIAS

- AGUSTSSON, E.; TIMOFTE, R. Ntire 2017 challenge on single image super-resolution: Dataset and study. In: **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops**. [S.l.: s.n.], 2017. Citado nas páginas [25](#), [30](#) e [32](#).
- BAI, X.; WANG, X.; LIU, X.; LIU, Q.; SONG, J.; SEBE, N.; KIM, B. Explainable deep learning for efficient and robust pattern recognition: A survey of recent developments. **Pattern Recognit.**, Elsevier BV, v. 120, n. 108102, p. 108102, dez. 2021. Citado na página [23](#).
- BEVILACQUA, M.; ROUMY, A.; GUILLEMOT, C.; MOREL, M. line A. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: **Proceedings of the British Machine Vision Conference**. [S.l.]: BMVA Press, 2012. p. 135.1–135.10. ISBN 1-901725-46-4. Citado na página [25](#).
- CAI, L.; GAO, J.; ZHAO, D. A review of the application of deep learning in medical image classification and segmentation. **Ann. Transl. Med.**, AME Publishing Company, v. 8, n. 11, p. 713, jun. 2020. Citado na página [23](#).
- CHAPMAN, A.; SIMPERL, E.; KOESTEN, L.; KONSTANTINIDIS, G.; IBÁÑEZ, L.-D.; KACPRZAK, E.; GROTH, P. Dataset search: a survey. **The VLDB Journal**, Springer Science and Business Media LLC, v. 29, n. 1, p. 251–272, ago. 2019. Disponível em: <<https://doi.org/10.1007/s00778-019-00564-x>>. Citado nas páginas [29](#) e [30](#).
- FENGQING, Q.; LILAN, C.; LIHONG, Z.; WANAN, Y. ulti-image super resolution reconstruction with pepper and salt noise. In: **Proceedings of 3rd International Conference on Multimedia Technology(ICMT-13)**. Atlantis Press, 2013/11. p. 401–407. ISBN 978-90-78677-89-5. ISSN 1951-6851. Disponível em: <<https://doi.org/10.2991/icmt-13.2013.49>>. Citado nas páginas [29](#) e [33](#).
- GALINDO, E.; PEDRINI, H. Image super-resolution improved by edge information. In: **2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)**. [S.l.]: IEEE, 2019. Citado nas páginas [23](#) e [30](#).
- GEBRU, T.; MORGNSTERN, J.; VECCHIONE, B.; VAUGHAN, J. W.; WALLACH, H. M.; III, H. D.; CRAWFORD, K. Datasheets for datasets. **CoRR**, abs/1803.09010, 2018. Disponível em: <<http://arxiv.org/abs/1803.09010>>. Citado na página [29](#).
- GOMEZ, N. **Top 10 Most Popular Online Games In 2022 | Updated Ranking**. 2022. Disponível em: <<https://sportsbrowser.net/most-popular-online-games/>>. Citado na página [24](#).
- HORÉ, A.; ZIOU, D. Image quality metrics: Psnr vs. ssim. In: **2010 20th International Conference on Pattern Recognition**. [S.l.: s.n.], 2010. p. 2366–2369. Citado nas páginas [34](#) e [35](#).

HUANG, J.-B.; SINGH, A.; AHUJA, N. Single image super-resolution from transformed self-exemplars. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2015. p. 5197–5206. Citado nas páginas 25, 30, 32 e 33.

KOUR, A.; YV, V.; MAHESHWARI, V.; PRASHAR, D. A review on image processing. **International Journal of Electronics Communication and Computer Engineering**, v. 4, p. 2278–4209, 01 2012. Citado na página 27.

LI, X.; WU, Y.; ZHANG, W.; WANG, R.; HOU, F. Deep learning methods in real-time image super-resolution: a survey. **J. Real Time Image Process.**, Springer Science and Business Media LLC, v. 17, n. 6, p. 1885–1909, dez. 2020. Citado na página 23.

LI, Y.; QI, F.; WAN, Y. Improvements on bicubic image interpolation. In: **2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)**. [S.l.: s.n.], 2019. v. 1, p. 1316–1320. Citado nas páginas 28 e 33.

LIM, B.; SON, S.; KIM, H.; NAH, S.; LEE, K. M. Enhanced deep residual networks for single image super-resolution. **CoRR**, abs/1707.02921, 2017. Disponível em: <<http://arxiv.org/abs/1707.02921>>. Citado na página 23.

MA, S.; ZHANG, X.; JIA, C.; ZHAO, Z.; WANG, S.; WANG, S. Image and video compression with neural networks: A review. **IEEE Trans. Circuits Syst. Video Technol.**, Institute of Electrical and Electronics Engineers (IEEE), v. 30, n. 6, p. 1683–1698, jun. 2020. Citado na página 23.

MOREIRA, S. **28 jogos mais jogados do mundo (2022)**. 2022. Disponível em: <<https://www.psverso.com.br/listas/jogos-mais-jogados-do-mundo/>>. Citado na página 32.

NGUYEN, K.; FOOKES, C.; SRIDHARAN, S.; DENMAN, S. Feature-domain super-resolution for iris recognition. **Computer Vision and Image Understanding**, Elsevier BV, v. 117, n. 10, p. 1526–1535, out. 2013. Disponível em: <<https://doi.org/10.1016/j.cviu.2013.06.010>>. Citado na página 23.

SHAO, W.; LUO, H.; ZHAO, F.; MA, Y.; ZHAO, Z.; CRIVELLO, A. Indoor positioning based on fingerprint-image and deep learning. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 6, p. 74699–74712, 2018. Citado na página 23.

TAI, Y.; YANG, J.; LIU, X. Image super-resolution via deep recursive residual network. In: **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.]: IEEE, 2017. Citado na página 23.

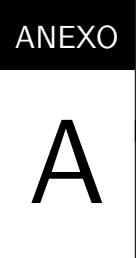
WAN, Z.; ZHANG, B.; CHEN, D.; ZHANG, P.; CHEN, D.; LIAO, J.; WEN, F. Bringing old photos back to life. **CoRR**, abs/2004.09484, 2020. Disponível em: <<https://arxiv.org/abs/2004.09484>>. Citado na página 30.

_____. Old photo restoration via deep latent space translation. **CoRR**, abs/2009.07047, 2020. Disponível em: <<https://arxiv.org/abs/2009.07047>>. Citado na página 30.

WANG, M.; DENG, W. Deep face recognition: A survey. **Neurocomputing**, Elsevier BV, v. 429, p. 215–244, mar. 2021. Citado na página 23.

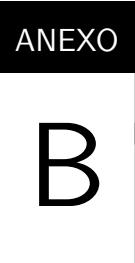
WANG, X.; YU, K.; WU, S.; GU, J.; LIU, Y.; DONG, C.; LOY, C. C.; QIAO, Y.; TANG, X. ESRGAN: enhanced super-resolution generative adversarial networks. **CoRR**, abs/1809.00219, 2018. Disponível em: <<http://arxiv.org/abs/1809.00219>>. Citado na página 30.

- WATSON, A. Deep learning techniques for super-resolution in video games. dez. 2020. Citado na página 23.
- _____. **Deep Learning Techniques for Super-Resolution in Video Games**. arXiv, 2020. Disponível em: <<https://arxiv.org/abs/2012.09810>>. Citado na página 25.
- XIAO, L.; NOURI, S.; CHAPMAN, M.; FIX, A.; LANMAN, D.; KAPLANYAN, A. Neural supersampling for real-time rendering. **ACM Trans. Graph.**, Association for Computing Machinery (ACM), v. 39, n. 4, ago. 2020. Citado na página 23.
- ZHANG, Y.; LI, K.; LI, K.; WANG, L.; ZHONG, B.; FU, Y. Image super-resolution using very deep residual channel attention networks. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. Citado na página 23.



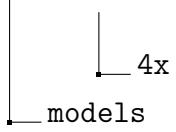
FORMULÁRIO DO GOOGLE FORMS

<<https://drive.google.com/file/d/1-9PYupfDb5Wqw4N9BsQyiZD2Kb9Ki6SD/view?usp=sharing>>



FOLDER TREE

```
original_images
├── fake_images
└── HR_Images
    ├── Bicubic
    │   ├── 2x
    │   ├── 3x
    │   └── 4x
    ├── Bicubic_blur
    │   ├── 2x
    │   ├── 3x
    │   └── 4x
    └── Bicubic_sp
        ├── 2x
        ├── 3x
        └── 4x
└── LR_Images
    ├── Bicubic
    │   ├── 2x
    │   ├── 3x
    │   └── 4x
    ├── Bicubic_blur
    │   ├── 2x
    │   ├── 3x
    │   └── 4x
    └── Bicubic_sp
        ├── 2x
        └── 3x
```





c.e.s.a.r
sch&ol