**UNICAMP – UNIVERSIDADE ESTADUAL DE CAMPINAS**

**FT – FACULDADE DE TECNOLOGIA**

**RELATÓRIO TÉCNICO: DESENVOLVIMENTO DE SOFTWARE VISUALIZADOR DE IMAGENS USANDO TÉCNICAS DE SUPER RESOLUÇÃO**

**FÁBIO AUGUSTO ALVES DINIZ**

**LIMEIRA - SP**

**2021**

**FÁBIO AUGUSTO ALVES DINIZ**

**RELATÓRIO TÉCNICO: DESENVOLVIMENTO DE SOFTWARE VISUALIZADOR DE IMAGENS USANDO TÉCNICAS DE SUPER RESOLUÇÃO**

Relatório Técnico realizado para o Trabalho de Conclusão de curso apresentado ao Curso de Graduação em Sistemas de Informação da Universidade Estadual de campinas.

Orientador: João Roberto Bertini Junior

**LIMEIRA – SP**

**2021**

**AGRADECIMENTOS**

Agradeço primeiramente a minha família por terem me dado apoios durante os momentos que mais precisei, a Deus por ter me dado saúde e força para superar todas as dificuldades. Obrigado a todos os professores que eu tive durante a minha graduação, o conhecimento obtido em cada disciplina cursada foi necessário para a conclusão desse projeto, e principalmente ao Prof. João Bertini pela paciência e pela oportunidade me dada para a realização desse trabalho. Meus agradecimentos aos amigos e a todos que fizeram parte da minha formação, seja indireta ou diretamente.

**RESUMO**

Apesar da tecnologia estar muito avançada e as câmeras fotográficas tirarem fotos praticamente sem defeitos, em determinados ambientes e com uma luminosidade fraca essas fotos podem sair com alguns serrilhados, e se for necessária uma ampliação de alguma parte da foto pode ser que ela não saia nítida. Portanto, o projeto de desenvolvimento de um software que utilizando de técnicas de Super Resolução fará com que essas imagens fiquem melhores visualmente. Esse tipo de software pode ser usado para ajudar em investigações policiais como reconhecimento facial, na medicina entre outras áreas.

Este relatório tem como objetivo mostrar a minha experiência durante o processo de desenvolvimento de software, usando Python como linguagem, e um algoritmo de Super Resolução.

**Palavras chaves:** Super Resolução, Desenvolvimento de Software, Python, Deep Learning, Inteligência Artificial

**LISTA DE FIGURAS**

[**Figura 1 - Interface gráfica do sistema 9**](#_Toc60410899)

[**Figura 2 - Diagrama de Casos de Uso 10**](#_Toc60410900)

[**Figura 3 - Diagrama de Sequência 11**](#_Toc60410901)

**SUMÁRIO**

[1 Introdução 5](#_Toc76453871)

[2 O que é *Deep Learning* 6](#_Toc76453872)

[3 Super-Resolução 7](#_Toc76453873)

[4 Projeto 8](#_Toc76453874)

[4.1 Análise e Levantamento de Requisitos 8](#_Toc76453875)

[4.2 Modelagem de interface gráfica 8](#_Toc76453876)

[4.3 Desenvolvimento do Sistema 9](#_Toc76453877)

[5 Testes e Resultados 11](#_Toc76453878)

[6 Requisitos de sistema 13](#_Toc76453879)

[6.1 Dependências 13](#_Toc76453880)

[7 Tutorial 14](#_Toc76453881)

[8 Trabalhos Futuros 15](#_Toc76453882)

[Referências: 16](#_Toc76453883)

# Introdução

*Deep learning* ou Aprendizado profundo é uma área de pesquisa que atualmente está sendo bem ativa, uma grande quantidade de aplicações estão obtendo sucesso, reconhecimento de fala, como a Siri, Cortana e Alexa e reconhecimento facial e até mesmo em diagnósticos no setor de saúde e identificação de fake news.

Rede Neural Artificial mostrou grande potencial na área de *Machine Learning*. A técnica é feita aplicando uma série de camadas que atuam de maneira análoga a um neurônio, *Deep Learning* é a aplicação de uma quantidade massiva de camadas de processamento em um algoritmo de rede neural.

Atualmente um dos maiores entretenimento das pessoas é tirar foto para postar nas redes sociais e atrair mais seguidores, porém dependendo da câmera fotográfica utilizada, a imagem que é entregue para o usuário pode não ter a qualidade que ele deseja e por isso ele pode querer editar essa foto para que fique melhor. Visando esse objetivo, existe um uma técnica de programação chamada Super Resolução.

Super Resolução é uma técnica que aumenta e melhora detalhes em uma imagem, ao pegar uma imagem com baixa resolução conseguimos utilizar de técnicas de *Deep Learning* e transformá-la em uma imagem de alta resolução.

Ao pegar uma imagem de alta resolução e aplicar uma função de degradação, teremos uma imagem de baixa resolução, e sabendo qual a função de degradação usada poderemos realizar o processo inverso nessa imagem, mas se tentar utilizar uma outra foto de baixa resolução e aplicar o processo inverso da função de degradação pode ser que a imagem não saia como o esperado, pois não saberemos a função de degradação que foi utilizada nela.

# O que é *Deep Learning*

*Deep Learning (DL)* vem sendo utilizada em diversas áreas de estudos como: Reconhecimento de imagens, de áudio e para reconhecimento facial. Hoje várias corporações vêm utilizando dessa técnica em seus projetos, como a Apple e a Siri, Microsoft e a Cortana, Samsung com seus desbloqueios de tela por reconhecimento facial, entre outras.

O *Deep Learning* surgiu como um paradigma que trata a dificuldade de arquiteturas utilizadas frequentemente, as RNAs ou máquina de vetores de suporte, que possuem alta dimensão de dados. Esta é proporcional ao volume em que os dados são inseridos, sendo mais difundidos quanto maior a dimensionalidade.

A diferença entre *Deep Learning* e *Machine Learning* está na função f(.), onde técnicas que não utilizam Aprendizado profundo são frequentemente chamadas de ‘superficiais’ ou ‘rasas’, pois buscam uma única função a partir de um grupo de parâmetros, gerando um resultado desejado. Porém para o *Deep Learning* existem técnicas que aprendem a função a partir de uma composição de funções, onde, para cada função o índice fi(.) o índice l refere-se a uma ‘camada’.

Sendo assim, um dos princípios da DL é gradativamente aprender com as representações dos dados. Estes algoritmos aprofundam a aprendizagem através de uma continuação de funções que transformam vetores e mapeiam todos os espaços até alcançar o resultado desejado.

# Super-Resolução

Super-Resolução (SR) pode ser definida como a obtenção de uma imagem de maior resolução a partir de uma imagem de baixa resolução, como por exemplo ter a entrada de uma imagem de dimensões 125p x 120p e gerar uma imagem de saída com a resolução 500p x 480p.

Um dos métodos para aumentar a resolução de imagens é utilizar a SR. O que torna a SR viável é justamente os deslocamentos em nível de subpixel.

O método de super-resolução baseado em filtragem de Wiener combina interpolação não-uniforme e restauração através de simples operações de somas ponderadas dos pixels de imagens deslocadas de baixa resolução, sendo a determinação dos coeficientes de ponderação detalhada neste capítulo.

Wavelets são funções matemáticas que permitem dividir um sinal em diferentes escalas, e estudar cada componente com a resolução igual à sua escala. Nossa percepção do universo utiliza o conceito de escala, onde cada observação realizada é feita em uma escala adequada para a compreensão dos diferentes detalhes de que precisamos. O procedimento de análise por wavelet consiste em adotar uma função base, denominada wavelet mãe, que serve como protótipo para todas as funções usadas no processo.

Há duas versões da transformada wavelet, a contínua e a discreta. A contínua é usada principalmente na análise e caracterização de detalhes de sinais, enquanto a discreta é mais apropriada para compressão de dados e reconstrução de sinais.

# Projeto

## Análise e Levantamento de Requisitos

O primeiro passo do projeto foi levantar e analisar os principais requisitos para o desenvolvimento do software. Foi perguntado para algumas pessoas que gostam de tirar fotos e foi dito que era necessário apenas um software que selecionasse uma imagem, melhorasse a qualidade e resolução dela e que salvasse no local selecionado.

## Modelagem de interface gráfica

Após o levantamento e análise dos requisitos foi feita a modelagem da interface gráfica, por se tratar de um sistema simples, que não teria tantas funcionalidades envolvidas, foi pensada em uma interface mais limpa e intuitiva para que o usuário consiga realizar seu objetivo sem precisar de muito esforço.

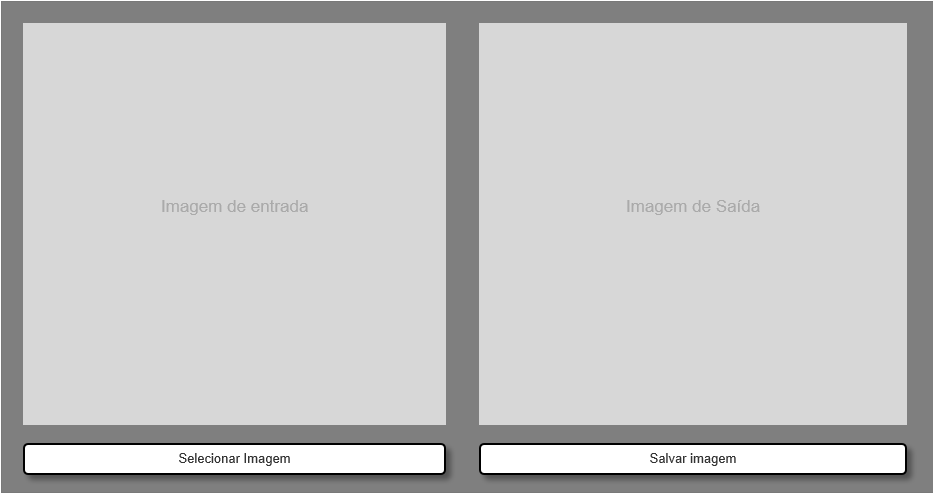
 Na figura 1 abaixo, é apresentado como é foi desenhada a tela do sistema.

Figura 1 - Interface gráfica do sistema

## Desenvolvimento do Sistema

Para o processo de implementação e desenvolvimento a partir dos requisitos passados pelos clientes, foi necessário que houvesse, inicialmente, um planejamento e para isso foi criado um diagrama de casos de uso e um diagrama de sequência. Com esse planejamento foi possível passar para a parte de desenvolvimento do *software.*

Esses diagramas são uteis para facilitar na hora da programação, e para que os usuários saibam quais recursos foram implantados no sistema.

A Figura 2 e 3 abaixo mostram o diagrama de Casos de uso e o diagrama de sequências.

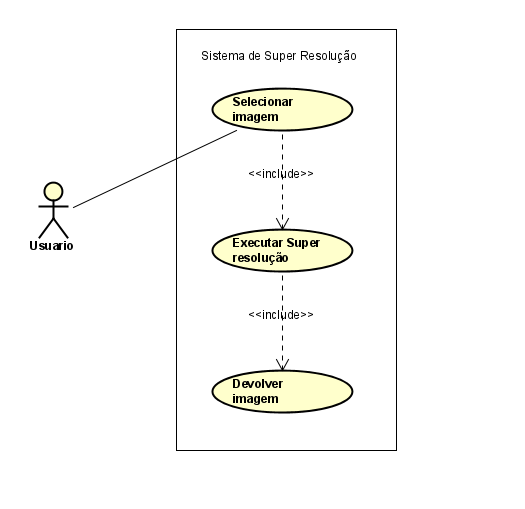


Figura 2 - Diagrama de Casos de Uso

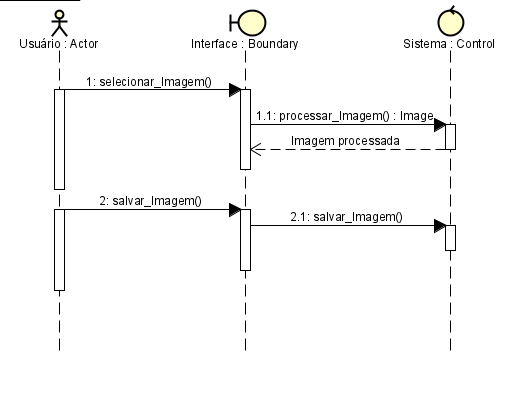


Figura 3 - Diagrama de Sequência

A maior dificuldade do desenvolvimento do sistema foi encontrar um algoritmo de Super resolução que resultasse em uma saída satisfatória para o usuário, e que esse tipo de algoritmo normalmente necessita de um computador com especificações não tão acessíveis ao público, fazendo com que dependendo do tamanho da foto, a máquina não tenha memória gráfica o suficiente para processar a imagem.

Após algumas pesquisas foi encontrado o algoritmo ESRGAN (*Enhanced Super-Resolution Generative Adversarial Networks*) e nele foi trabalhado para que os requisitos do sistema fossem atribuídos.

# Testes e Resultados

Para os testes e resultados que aqui serão apresentados, foi utilizado um computador com as seguintes configurações:

* Placa de vídeo: Nvidia GTX 1660 Super – 6GB VRAM
* Processador: Intel i5 9600KF
* Memória RAM: 16GB

Acima é possível ver na esquerda, a imagem de entrada de um babuíno com resolução 125x120 e na direita a saída com resolução 500x480.

Link para vídeo demonstrativo:

<https://drive.google.com/file/d/1Zb6iWY91q3Do0_rVHtOx8MafFLb-Maxd/view?usp=sharing>

Cachorro na areia da praia

Descrição gerada automaticamenteCachorro na areia da praia

Descrição gerada automaticamente Nestas imagens acima é possível ver uma imagem de um cachorro sentado na areia da praia, na esquerda a entrada com resolução 251x280 e a saída na direita de resolução 1004x1120

# Requisitos de sistema

Para o software funcionar corretamente, é necessária uma placa gráfica de no mínimo 6 GB de memória de vídeo, mas para alguns tipos de imagens é necessário pelo menos 8 GB de VRAM.

Por conta do algoritmo de Super-Resolução utilizar a plataforma de computação paralela desenvolvida pela Nvidia, é recomendado que o computador que seja executado o programa contenha uma placa da Nvidia que tenha essa plataforma.

Para os testes e desenvolvimento desse projeto foi utilizada uma Placa de vídeo Nvidia GTX 1660 Super, processador Intel I5 9600kf, 16GB de memória RAM.

## Dependências

* Python 3
* PyTorch 10+ (CUDA 7.5+, se instalado com CUDA)
* Pacotes Python: pip install numpy opencv-python.

# Tutorial

1 - Ao abrir o aplicativo abrirá a tela principal dele onde terá um botão para escolher qual imagem que será aplicada a Super Resolução.

2 – Na tela que aparecer, vá até a pasta em que a imagem está salva e selecione-a.

3 – Após selecionar, o processo de Super Resolução será iniciado.

4 – Quando terminar irá aparecer uma outra tela para que seja escolhida a pasta ao qual deseja salvar a nova imagem, após selecionar é só escolher um nome para o arquivo e salvar.

5 – A imagem estará pronta na pasta em que foi escolhida no passo 4.

# Trabalhos Futuros

Muitas linhas podem ser seguidas, seria interessante fazer mais experimentos, com aumento da resolução da imagem por diferentes fatores e uso de um número reduzido de imagens, para investigar as limitações dos algoritmos implementados. O efeito da utilização de mais níveis de decomposição não foi estudado neste trabalho e poderia ser vantajoso em alguns casos. Uma análise da complexidade computacional dos métodos com diferentes parâmetros, tais como número de imagens LR e fator de aumento de resolução, poderia ser realizada.

# Referências:

[1] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. *Fast and robust multiframe super resolution*. IEEE *Transactions on Image Processing*, 13(10):1327–1344, 2004.

[2] S. Borman and R. L. Stevenson. *Super-Resolution from Image Sequences-A Review. Midwest Symposiumon Circuitsand* Systems, páginas 374–378, 1998.

[3] C. E. Duchon. *Lanczos Filtering in One and Two Dimensions. In Journal of Applied Meteorology,* volume 18, páginas 1016–1022. 1979.

[4] K. Simonyan and A. Zisserman. *Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations* (ICLR), 2015.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. *Going deeper with convolutions*. In IEEE *Conference on Computer Vision and Pattern Recognition* (CVPR), páginas 1–9, 2015.

[6] Christopher Thomas – Deep learning based super resolution without using GAN, Disponível em: <https://towardsdatascience.com/deep-learning-based-super-resolution-without-using-a-gan-11c9bb5b6cd5> acesso: 10/10/2019.

[7] Pacheco, César Augusto Rodrigues, Pereira, Natasha Sophie. Deep Learning Conceitos e Utilização nas Diversas Áreas do Conhecimento, 2018

[8] Copeland,Michel – What’s the Difference Between Artificial Intelligence, Machine Learning and Deep Learning, Disponível em <https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/> acesso: 20/05/2020.

[9] Park, S. C., Park, K., Kang, M.G. M. Super-resolution image reconstruction: a technical overview. IEEE Signal Processing Magazine. V. 20, n. 3, p. 21-26, 2003.

[17] Takemura, E. S., Algoritmos para Super-Resolução de imagens baseados nas filtragens de Wiener e adaptativa usando a transformada Wavelet, 2010.