Framework para reconstrução de imagens baseado em técnicas de InPainting

Trabalho de Conclusão de Curso

Engenharia da Computação

Fábio Alexandre Magalhães de Holanda e Silva
Orientador: Bruno Fernandes

Universidade de Pernambuco
Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação

Fábio Alexandre Magalhães de Holanda e Silva

Framework para reconstrução de imagens baseado em técnicas de InPainting

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, Dezembro de 2016.

POLI
ESCOLA POLITÉCNICA
DE PERNAMBUCO

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 6 de 6 de 2017, às 10:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente **Fábio Alexandre Magalhães de Holanda e Silva**, orientado pelo professor **Bruno José Torres Fernandes**, sob título **Framework para reconstrução deimagens baseado em técnicas deInPainting**, a banca composta pelos professores:

Sérgio Murilo Maciel Fernandes
Bruno José Torres Fernandes

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

☒ Aprovada ☐ Aprovada com Restrições* ☐ Reprovada

e foi-lhe atribuída nota: 10,0 (dez)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 07 dias para entrega da versão final da monografia a contar da data deste documento.

SÉRGIO MURILO MACIEL FERNANDES

BRUNO JOSÉ TORRES FERNANDES

* Este documento deverá ser encadernado juntamente com a monografia em versão final.

AGRADECIMENTOS

Agradeço à Deus, pelas inúmeras oportunidades que vem me dando todos esses anos de faculdade e pela força que me ofereceu para enfrentar cada desafio.

Agradeço à minha mãe, que se fez presente em cada momento da minha vida, incluse na fase acadêmica. Apoiando-me em cada decisão tomada, e sempre me educando da melhor forma possível.

Agradeço à minha futura esposa, por trilhar este caminho junto comigo, me ajudando em tudo que precisava, e principalmente acreditando no meu potencial.

E por fim, aos meus professores que me fizeram um amante da computação, passando seus conhecimentos da melhor forma possível, com ressalva ao meu orientador, Bruno Fernandes, que me mostrou uma área da qual não imaginava que tinha este facínio.

RESUMO

Devido ao avanço tecnológico das câmeras digitais e computadores, a utilização de ferramentas para edição de imagens se tornou algo bastante comum na rotina das pessoas, presente principalmente em softwares profissionais e até mesmo em redes sociais. Porém, a complexidade das imagens e suas variações tornaram algumas operações de edição custosas, tanto computacionalmente quanto manualmente. Baseando-se neste cenário, este trabalho propõe um framework de reconstrução de imagens, visto que esta operação é bastante usada, porém nada prática, nem tão pouco automatizada, e com resultados pouco satisfatórios. Este algoritmo aborda conceitos de métricas de similaridade, modelos de regressão e processamento digital de imagens. Para avaliar o desempenho do algoritmo proposto foram realizados testes comparativos com os algoritmos mais conhecidos da literatura. Nessa análise, o modelo proposto obteve o melhor resultado em todos os testes.

ABSTRACT

Due to the technological advancement of digital cameras and computers, the use of tools for image editing has become common in the routine of people, present mainly in professional software and even in social networks. However, the complexity and variations of the images have made some editing operations costly, both computationally and manually. Based on this scenario, this work proposes a framework for image reconstruction, since this operation is widely used, but nothing practical and not even automated, and with unsatisfactory results. This algorithm uses concepts of similarity metrics, regression models and digital image processing. To evaluate the performance of the proposed algorithm, we performed comparative tests with the most known algorithms in the literature. In this analysis, the proposed model obtained the best result in all tests.

SUMÁRIO

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

ÍNDICE DE EQUAÇÕES

1 INTRODUÇÃO

Neste trabalho de conclusão de curso foi desenvolvido um framework baseado em comitê de máquinas, para reconstrução de imagens que leva em consideração suas características, de modo que através dessas características se possa reconstruir a imagem pela utilização do algoritmo de reconstrução mais adequado para a mesma. O modelo proposto utiliza um repositório com os algoritmos de reconstrução mais conhecidos da literatura, métricas de similaridade para extração da qualidade da reconstrução, modelos de regressão para previsão da qualidade da reconstrução para todos os algoritmos e um comitê de máquinas.

Este capítulo apresenta a introdução desta monografia, e está organizado em 3 seções. Na Seção 1.1, é apresentada a motivação para a realização deste trabalho, bem como o problema abordado pelo mesmo. Em seguida, na Seção 1.2, são apresentados os objetivos gerais e específicos e a hipótese de como o problema pode ser solucionado. Por fim, na Seção 1.3, é descrita a estrutura do restante da monografia.

1.1 Motivação e Caracterização do Problema

Atualmente uma das técnicas de processamento de imagem mais difundidas na literatura chama-se *Region Filling* [1]. Consiste no processo de preenchimento de uma região, definida na imagem, sendo utilizado por alguns softwares de edição de imagens para remover objetos e reconstruir a imagem de tal forma que não seja perceptível a remoção. Este tipo de processo tem sido abordado por diversas técnicas, sendo o *InPainting* uma das mais utilizadas [2]. O *InPainting* pode ser implementado para gerar grandes regiões na imagem através de uma amostra de textura que pertence a mesma, ou seja, esta abordagem funciona repetindo padrões da textura com base em sua amostra, analisando as áreas vizinhas, ou até mesmo pode ter sua implementação focada em pequenos espaços na imagem analisando estruturas lineares que compõem esta área, isto é, analisando o contorno de objetos.

Para cada técnica, na literatura, já foram implementados diversos algoritmos, como Efros *et al*. [3], baseado em síntese de textura, onde foi proposto um processo de cópia de blocos diretamente da textura na amostra, pela observação da semelhança dos pixels vizinhos da região com a imagem de entrada. Basicamente,

estas áreas que possuem vizinhos semelhantes aos da parte removida seriam copiados para esta região.

O fator que torna a utilização do *Region Filling* um processo raro, não trivial, e com resultados não satisfatórios, se dá pela composição da imagem, sua aleatoriedade e sua complexidade nas texturas, fazendo com que um algoritmo tenha bons resultados para uma determinada classe de imagens, mas para outras não tenha [4]. Isso inviabiliza, e até, impede que algumas aplicações de edição de imagens sejam utilizadas, fazendo com que o trabalho de remoção e preenchimento seja extremamente custoso, tanto computacionalmente, quanto para o usuário.

1.2 Objetivos e Metas

Dado o problema da utilização do *Region Filling,* este trabalho tem como objetivo desenvolver um framework de comitê de máquinas [5] para a reconstrução de imagens que, através de um modelo de regressão [6], defina qual o melhor algoritmo para uma determinada imagem. Será utilizado um repositório de algoritmos de reconstrução e métricas de similaridade [7], para definir a qualidade destes algoritmos em cada imagem de uma base de dados. A partir disso, *n* modelos de regressão serão treinados utilizando como dados, a imagem, os algoritmos e o desempenho de cada um para a determinada imagem. Após o treinamento destes modelos, o framework fará uso do conceito de comitê de máquinas e com isso irá definir qual o melhor algoritmo para reconstrução, dada uma imagem qualquer.

1.3 Organização do Documento

Este trabalho está organizado em 5 capítulos. No Capítulo 2, serão revistos alguns fundamentos que foram explorados, começando com conceitos e técnicas para reconstrução de uma imagem. Em seguida, serão apresentadas as métricas de similaridade que vão ser utilizadas e suas definições. Ainda neste capítulo, serão mostradas as definições e conceitos de regressão estatística e comitê de máquinas. Já os processos e métodos utilizados para a construção do algoritmo desenvolvido neste trabalho serão apresentados no Capítulo 3. No Capítulo 4, serão apresentados os resultados do comparativo realizado entre o algoritmo proposto e os descritos na literatura. Por fim, no Capítulo 5, serão discutidas as principais conclusões desse trabalho, como também os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Reconstrução de imagens

A reconstrução de partes danificadas de uma imagem e até a remoção e preenchimento de objetos da mesma, são práticas que puderam ser aprimoradas com o poder de processamento dos computadores modernos [1]. Hoje se tornou uma prática comum, a aplicação desses conceitos para edição de imagens utilizando softwares especializados como o *PhotoShop.* Reconstrução de imagem refere-se ao processo de reconstruir partes perdidas de uma imagem [1], processo este que é utilizado por diversos algoritmos na literatura, sendo o *Region Filling* uma das suas principais técnicas*.*

Region Filling consiste em um algoritmo morfológico que preenche buracos em uma imagem com base em seus vizinhos [1]. A implementação do mesmo pode ser feita com base em diferentes aspectos, sendo o *InPainting* [2] uma das interpretações mais utilizadas na literatura e que ganhou força com o surgimento de câmeras digitais e digitalização de fotos impressas, onde processos de remoção de pessoas, objetos, arranhões e até texto de uma foto se tornaram rotineiras.

Este processo pode ser implementado utilizando a estrutura, a textura e a combinação de ambas. A primeira implementação consiste em analisar as bordas da região que será preenchida, ou seja, analisa os vizinhos e as estruturas geométricas do espaço estudado, e então define quais pixels serão utilizados para preencher o buraco. Já a segunda implementação precisa de uma amostra da imagem a ser preenchida. Os vizinhos da região são comparados com a amostra e, com isso, é copiado o pixel da amostra para o espaço da imagem. A terceira implementação por sua vez é uma combinação dos dois métodos, onde as estruturas da vizinhança são continuadas, porém as escolhas dos pixels para preenchimento são feitas por meio da amostra. Diferentemente de outras técnicas baseadas no *Region Filling,* o *InPainting* é utilizado para propósito geral, podendo preencher grandes ou pequenas regiões de uma imagem [2].

Nesta seção são apresentadas as definições e conceitos básicos dos trabalhos relacionados e das técnicas de *Inpainting* que serão abordadas. Além disso, serão apresentados detalhes dos algoritmos utilizados para comparação com o algoritmo

proposto. A seguir, serão apresentados os algoritmos de *InPainting* que serão utilizados neste trabalho.

2.1.1 Image InPainting

Inicialmente, Bertalmio *et al.* [8] propuseram um dos primeiros algoritmos baseados neste tipo de técnica, onde o mesmo observa a estrutura da área circundante da região e, através disso, é dado continuidade as linhas de contorno. Basicamente o algoritmo continua preenchendo as linhas da fronteira, fazendo com que o *gap* seja preenchido através de um número predeterminado de iterações através de uma constante calculada [8]. Na equação 2.1 podemos ver o cálculo do algoritmo proposto por Bertalmio, onde $\overrightarrow{\delta L^n}(i,j)$ é uma medida calculada que muda a informação do pixel estudado na direção $\overrightarrow{N}^n(i,j)$. Este processo é mantido até que que a condição de parada seja satisfeita.

$$I_t^n(i,j) = \overrightarrow{\delta L^n}(i,j) \cdot \overrightarrow{N}^n(i,j),$$

Equação 2.1. Equação do algoritmo de Bertalmio.

2.1.2 Exemplar-Based Image Inpainting

Recentemente, Criminisi *et al.* [4] propuseram um algoritmo baseado na combinação dos dois métodos do *InPainting*, unindo as vantagens das duas abordagens. Sabendo que a síntese de textura é essencial para o preenchimento e que este processo é altamente dependente da propagação da estrutura. O algoritmo necessita de uma amostra da imagem para poder começar sua execução, que por sua vez é dividida em 3 etapas [4]. A primeira é responsável pelo processamento das prioridades de cada bloco da amostra, levando em consideração a continuação das bordas e os pixels vizinhos. A equação 2.2 mostra o cálculo da prioridade que é utilizado pelo algoritmo de reconstrução, onde o *C(p)* é a confiança calculada e *D(p)* é o valor do pixel.

$$P(\mathbf{p}) = C(\mathbf{p})D(\mathbf{p}).$$

Equação 2.2. Equação para o cálculo da prioridade do algoritmo de Criminisi.

Após calcular as prioridades de cada bloco o algoritmo executa a etapa seguinte, que consiste napropagação de textura e estrutura da informação. O bloco com maior prioridade é escolhido para realizar o processo de preenchimento, onde

por sua vez, são selecionados os pixels com maior similaridade. Este preenchimento é feito diretamente, diferente de algumas técnicas que usam o esmaecimento do mesmo. E por último é recalculado a medida de confiança para o algoritmo iterar mais uma vez esses processos.

2.1.3 Inpaint Nans

Este algoritmo é especializado em remoção de ruídos e em pequenos elementos, os quais podem estar espalhados por toda a imagem [9]. Através de uma triangulação dos dados e de cálculos algébricos, o *Inpaint Nans* é implementado por 5 abordagens diferentes, variando seu desempenho de acordo com o tamanho dos buracos da imagem, sendo o quinto método a melhor escolha para restauração de grandes *gaps*. Desenvolvido por D'Errico, em *MatLab,* este algoritmo é fornecido gratuitamente como extensão.

2.1.4 Hybrid Sparse Representations

Definir texturas e bordas não é algo trivial, pois muitas vezes não estão evidentes na imagem, principalmente se tratando de imagens capturadas por câmeras com características e qualidades diferentes [1]. Este é um dos principais problemas encontrados pelos algoritmos de reconstrução. Pensando nisso, *Li et al.*[10] propôs um algoritmo baseado em representações esparsas sob um modelo bayesiano.

A ideia básica por trás deste modelo é reconhecer a dependência estatística entre os pixels conhecidos e os ausentes em uma imagem, que podem ser descritos por vários modelos concorrentes M_k . A equação 2.3 mostra esta dependência que o pixel ausente possui em relação ao pixel presente.

$$p(x_m|x) = \sum_{k=1}^{K} a_k p(x_m|x, M_k),$$

Equação 2.3. Equação da dependência estatística.

Onde x e x_m são os pixels presentes e ausentes na imagem, respectivamente, e $a_k = p(M_k|x)$ denota a próxima probabilidade para o modelo. Através de simplificações matemáticas, chega-se a equação vista na equação 2.4, que calcula esta probabilidade, e que irá ser usada para estimar o valor do pixel recorrente.

$$a_k = \frac{p(x|M_k)}{\sum_{l=1}^{K} p(x|M_l)}$$

Equação 2.4. Equação da probabilidade do modelo.

2.2 Métricas de similaridade

Quando se aborda a reconstrução de imagem, temos como resposta diversos algoritmos, com performances que variam de muito bom à péssimo. Este desempenho se dá pela aleatoriedade e formação das imagens, as quais podem variar desde a sua textura, ao padrão da estrutura que as compõem [1]. Em virtude disto, se tornou necessária uma medida que calculasse a qualidade de uma reconstrução. Esta medida chama-se métrica de similaridade, e é aplicada em muitos ramos da engenharia [11]. Atualmente as medidas mais usadas para calcular a qualidade das reconstruções de imagem chamam-se PSNR *(Peak signal-to-noise ratio) e* SSIM *(Structural similarity).*

Nesta seção são apresentados os algoritmos de similaridades que serão abordados, bem como suas características e diferenças.

2.2.1 PSNR

Peak signal-to-noise ratio (Relação sinal-ruído de pico) [11], mas popularmente conhecido como PSNR é um termo bastante utilizado na engenharia para medir a relação entre a máxima energia de um sinal e o ruído que afeta sua fiel representação. Este cálculo é expresso em decibéis, relaciona um maior valor de potência de um sinal pela potência do ruído [11]. Neste trabalho, isso pode ser resumido como a relação da imagem original com a sua restauração. Para definir o algoritmo, é necessário o uso de outra técnica bastante conhecida na literatura, chamada de MSE (*Mean Squared Error*) ou simplesmente erro médio quadrático. Na Equação 2.5 podemos analisar o cálculo do erro médio quadrático, utilizado para calcular o PSNR, que é apresentado na Equação 2.6.

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} ||I(i,j) - K(i,j)||^2$$

Equação 2.5. Cálculo do erro médio quadrático.

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right)$$

Equação 2.6. Cálculo do PSNR.

Onde o *M* e *N* são as dimensões da imagem, o *I* e *K* são as imagens originais e a reconstruída respectivamente, já o *MAX* indica o valor máximo de pixel possível em uma imagem.

2.2.2 SSIM

Structural similarity (Similaridade estrutural) [12], ou simplesmente SSIM é um método para medir a similaridade entre duas imagens. A comparação só pode ser feita caso que uma das imagens tenha qualidade perfeita. Em outras palavras, a medição é baseada numa imagem inicial sem redução de qualidade ou distorção. Esta técnica foi projetada para melhorar os métodos tradicionais como o PSNR, que demonstrava inconsistência com a percepção visual humana [12].

A principal diferença entre essas técnicas está na abordagem em que o PSNR tem em estimar o erro absoluto. Já o SSIM baseia-se na percepção da informação estrutural da imagem, onde os pixels possuem fortes interdependências, que por sua vez trazem informações importantes sobre a estrutura do objeto [12].

O algoritmo do SSIM percorre toda a imagem aplicando seu cálculo para cada janela da mesma, este cálculo está presente na equação 2.7.

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Equação 2.7. Cálculo do *SSIM.*

Onde μ_x e μ_y são as médias da imagem original e restaurada, σ_x^2 e σ_y^2 são as variâncias e por último temos c_1 e c_2 que possuem o papel de estabilizar a divisão.

2.3 Modelos de regressão

Modelos que procuram prever informações relevantes a uma aplicação estão se tornando cada vez mais presentes no nosso dia-a-dia. Podemos ver exemplos disto, em barragens, climatologia, economia e outros, onde sistemas são capazes de estimar um valor de uma determinada variável com base em dados antigos e com isso tomar alguma decisão.

No momento, os modelos de regressão são as únicas técnicas estatísticas usadas para esse fim [6]. Eles podem ser definidos como modelos matemáticos que relacionam o comportamento de uma variável *Y* com outra *X.* A forma como se dá esta relação irá definir o tipo de modelo, podendo ser linear ou não linear. É através desta relação que é possível simular o comportamento futuro destas variáveis e com isto prever seus valores, ou seja, extrapolar as relações, já observadas no passado [6]. Como por exemplo, prever a população futura de uma cidade simulando a tendência de crescimento da população no passado.

A seguir serão abordados os conceitos de regressão linear e não linear, e por fim será apresentado o algoritmo de regressão usado neste trabalho.

2.3.1 Modelos de regressão linear

Sistemas lineares possuem uma reta como relação entre suas variáveis, e é esta mesma definição que é usada em regressão linear. Nós temos as mesmas variáveis, *Y* e *X,* porém elas somente se relacionam através de uma reta, chamada *reta de regressão* [6]. Esta reta é usada para analisar os futuros valores, seguindo a mesma relação do sistema. Isto significa que os valores observados neste sistema podem ser simulados através da equação apresentada na equação 2.8.

$$Y = \alpha + \beta X$$

Equação 2.8. Equação linear.

2.3.2 Modelos de regressão não linear

Diferente da regressão linear, estes modelos possuem uma função definida por uma combinação não linear dos parâmetros, onde uma ou mais variáveis do sistema são independentes [13]. Esses dados são ajustados geralmente pelo método dos mínimos quadrados ou por algum método de aproximação sucessiva.

Voltando ao mesmo contexto, as variáveis *X e Y,* desta vez, se relacionam de forma independente, onde seus valores não estão ligados proporcionalmente, isto faz com que técnicas de regressão linear não funcionem em problemas não lineares, exigindo novos algoritmos [13]. A principal forma de lidar com este problema é convertendo um sistema não linear em um linear através de transformação linear, e com isto aplicando as mesmas técnicas, como podemos ver nas equações 2.9 e 2.10, onde mostram um exemplo de transformação linear utilizando logaritmo neperiano.

$$y = ae^{bx}$$

Equação 2.9. Equação não linear (regressão exponencial).

$$y/a = e^{bx}$$

$$\ln(y/a) = bx * \ln(e)$$

$$\ln(y) - \ln(a) = bx$$

$$\ln(y) = bx + \ln(a)$$

Equação 2.10. Transformação linear.

2.3.3 Support Vector Regression (SVR)

O *Support Vector Machine* (SVM) [14], muito conhecido na literatura por seu modelo de classificação, também pode ser usado como um método de regressão, mantendo todas as principais características que identificam o algoritmo, este método é chamado de *Support Vector Regression* (SVR). A ideia básica deste modelo é mapear um conjunto de dados em um espaço multidimensional através de um mapeamento não linear e então realizar uma regressão linear neste espaço transformado [15].

O SVR utiliza os mesmos princípios que a SVM para classificação, com apenas algumas pequenas modificações. A principal mudança está na saída do algoritmo, onde em classificação tínhamos um valor inteiro, discreto, enquanto na regressão possuiria um número real, contínuo. Isto faz com que o algoritmo fique mais complexo e custoso. Diante disso é usado uma margem de tolerância para simplificar as

operações e estimar a saída. Apesar de algumas diferenças, a ideia principal sempre é a mesma, minimizar o erro.

2.4 Comitê de máquinas

A agregação de mais de uma máquina de aprendizado na produção de uma única solução computacional para um determinado problema é chamado comitê de máquinas [5]. Seu princípio é baseado na seleção do melhor candidato através de uma medida de desempenho.

As máquinas de comitê podem ser classificadas em duas categorias: as estruturas estáticas e as dinâmicas [5]. A primeira possui uma saída produzida por respostas de várias redes que são combinadas por um mecanismo que não envolve o sinal de entrada, por isso a designação estática. Por outro lado, as dinâmicas, possuem entradas que estão diretamente envolvidas na atuação do mecanismo que combina as respostas.

Na Figura 2.1 pode-se ver uma representação do comitê de máquinas, onde *x(p)* representa o sinal de entrada, que será dado a todas as máquinas de aprendizado (Expert 1, Expert 2 e etc). Essas máquinas, por sua vez, respondem com os valores $y_k(p)$ que são direcionados a um processo de combinação. Este processo dependerá da aplicação final para qual o comitê foi desenvolvido. E por fim será obtido como resposta o valor *y(p).*

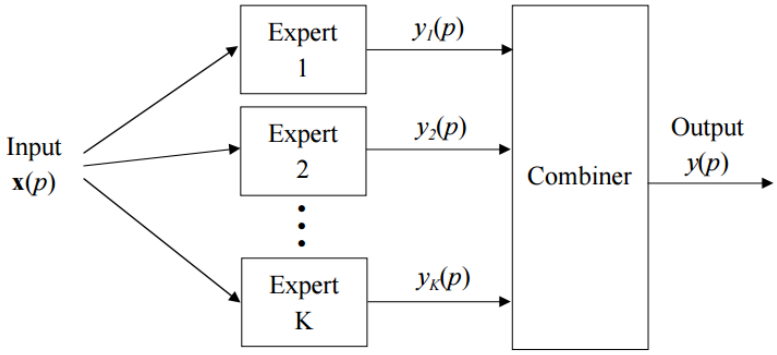


Figura 2.1. Representação de um comitê de máquinas que possui *x(p)* como entrada e *y(p)* como saída.

3 MODELO PROPOSTO

O desenvolvimento do modelo proposto está baseado no comitê de máquinas e faz uso dos algoritmos mais populares da literatura para extração das métricas de similaridade. Além disso, são usados *n* modelos de regressão para a predição da melhor abordagem para reconstruir uma imagem.

O modelo fará uso de um repositório de algoritmos de restauração de imagens, os quais conterão um modelo de regressão dedicado a sua abordagem. Estes modelos de regressão serão treinados com uma base de imagens exclusiva ao treinamento, e com as medidas de similaridade, que indicarão o desempenho dos algoritmos para uma imagem. Após este treinamento, o modelo proposto, baseado no comitê de máquinas, receberá uma imagem, que será dada a todos os modelos de regressão, que por sua vez, responderão com a predição das medidas de desempenho. Essas medidas serão usadas para indicar a melhor proposta de restauração para a imagem de entrada.

Na Seção 3.1 serão descritos os processos realizados para extração das medidas de qualidade, medidas estas que serão utilizadas na fase de treinamento dos modelos de regressão, a ser abordado na Seção 3.2. E por último, na Seção 3.3, será mostrado o fluxo de execução e o pseudocódigo do algoritmo implementado.

3.1 Extração das métricas de similaridade

O objetivo deste processo é adquirir as informações relativas ao desempenho de cada algoritmo para uma dada imagem em um determinado conjunto de treinamento. Essas informações serão utilizadas para o treinamento do modelo de regressão [6]. Essa etapa consiste em 3 fases: a primeira é a formação de um *gap* em cada imagem da base; em seguida são executados todos os algoritmos de restauração do repositório para cada imagem; e, por fim, será calculado o desempenho desses algoritmos.

Todas as imagens, da base de treinamento, passarão por um processo de remoção. Essa remoção será no centro da imagem, com dimensões de 60 x 60 *pixels.* Posteriormente essas imagens serão reconstruídas pelos algoritmos de reconstrução, e com os resultados obtidos nesta etapa, serão calculados as medidas de similaridade utilizando os algoritmos PSNR e SSIM [11, 12].

A seguir serão apresentados os algoritmos de reconstrução que compõem o repositório, e suas devidas configurações que foram utilizadas para o desenvolvimento deste modelo:

- *Hybrid Sparse Representations,* de Li [10]:
    - Este algoritmo recebe como parâmetro uma lista de números inteiros configurada para receber os valores 15, 31, 30 e 2. Além disso, ele recebe um número inteiro que representa as interações que o algoritmo irá realizar, para este parâmetro o valor 2 foi escolhido.
- *Inpaint Nans,* de D'Errico [9]:
    - A implementação deste algoritmo é feito por 5 técnicas. Essas abordagens são escolhidas através do único parâmetro que ele possui, sendo o valor 5 escolhido para este trabalho.
- *Exemplar-Based Image Inpainting,* de Criminisi *et al.* [4]:
    - A função de restauração deste algoritmo não recebe nenhum tipo de parâmetro de configuração.
- *Image Inpainting,* de Bertalmio *et al.* [8]:
    - Assim como o algoritmo anterior, esse também não necessita de configurações extras.

3.2 Treinamento dos modelos de regressão

Na segunda etapa do modelo proposto estão presentes os processos de treinamento do modelo e regressões da base. Nesta fase é fundamental para o algoritmo estimar as taxas de desempenho, e através disso, usar a lógica de comitê de máquinas para escolher a melhor abordagem para restauração de imagem [5]. Como essas medidas são contínuas, torna-se inviável uma análise por categorias discretas, por isso se fez necessário o uso de uma análise de regressão, fazendo com que as máquinas de aprendizado utilizados neste comitê sejam modelos de regressão [13].

Para cada algoritmo de restauração contido no repositório, será gerado um modelo de regressão. O modelo de regressão usado foi o SVR [15], por ser extremamente robusto em espaços com muitas dimensões, já que ele não depende da dimensão do espaço das entradas. Como esse algoritmo irá receber imagens de

200 x 200 *pixels* essa característica é fundamental. Este modelo foi implementado em *python* pelo pacote *open source* scikit-learn.

Na fase de treinamento, os modelos de regressão receberam como entrada suas configurações iniciais, que são o tipo de função e a margem de tolerância do erro, os quais foram configurados como *linear* e *1e3,* respectivamente. Além disso, o modelo recebe como parâmetro a base de imagens destinada ao treinamento e suas respectivas métricas. Após a fase de treinamento o modelo receberá uma imagem como entrada e como saída apresentará a previsão da medida de similaridade para o determinado algoritmo. Essa medida será o índice que indicará a qualidade da reconstrução para esta imagem caso ela tenha sido reconstruída usando este algoritmo, ou seja, será uma previsão dos valores do PSNR e SSIM [11, 12].

3.3 Fluxo de execução do modelo proposto

O algoritmo proposto se comporta como um comitê de máquinas de regressão [5], com o propósito de melhorar as reconstruções de imagens. Através do treinamento das máquinas de aprendizado, é realizado uma comparação dos resultados obtidos com os modelos de regressão e, com isso, é definida a melhor abordagem para reconstrução de uma imagem.

Este modelo foi desenvolvido utilizando a linguagem de programação *python,* e seu fluxo de execução é iniciado com o treinamento do algoritmo, descrito na seção anterior. Após treinado o algoritmo receberá como parâmetro a imagem que será reconstruída e retornará sua reconstrução. Internamente essa imagem será dada como entrada para os modelos de regressão de todos os algoritmos do repositório. Os modelos, por sua vez, retornarão os índices de qualidade [7]. Esses índices serão comparados, e um decisor escolherá o algoritmo que obteve o melhor desempenho [5]. Com isto a imagem será dada a este algoritmo, que por sua vez responderá com a imagem reconstruída.

O pseudocódigo que detalha este processo pode ser visto na descrição do Algoritmo 1, a seguir. Nele, é utilizado algumas funções auxiliares como as funções de busca da base de dados as imagens de treinamento e teste que estão descritas nas linhas 1 e 2. A variável *modeloRestauracao* é uma lista formada por uma estrutura que contêm os algoritmos de restauração e os modelos de regressão. Essa estrutura possui uma interface chamada *restaurar* que por sua vez executa a função de

restauração do algoritmo, e a função *treinamento* que vai treinar o modelo de regressão. Podemos ver na linha 8 a função *ssim* que retorna a medida de similaridade da restauração que foi realizada, e com isto é concatenado na lista de medidas que vão ser utilizadas no treinamento. Por fim é executado a função *melhorAbordagem* que usa o conceito de comitê de máquinas para escolher a melhor abordagem para a determinada imagem. O fluxo de execução deste algoritmo está descrito na Figura 3.1.

Algoritmo 1. Pseudocódigo do modelo proposto baseado em comitê de máquinas.

```
1.  imagensTreinamento = getImagensTreinamento()
2.  imagensTeste = getImagensTeste()
3.  numeroAlgoritmoRepositorio = modeloRestauracao.quantidade()
4.  para i=0 ate numeroAlgoritmosRepositorio faca
5.    para j=0 ate imagensTreinamento.quantidade() faca
6.      imagemOriginal = imagensTreinamento[j]
7.      imagemRestaurada = modeloRestauracao[i].restaurar(ImagemOriginal)
8.      labelTreinamento.add(ssim(ImagemOriginal,imagemRestaurada)
9.    fim para
10.   modeloRestauracao[i].treinamento(imagensTreinamento,labelTreinamento)
11. fim para
12. para i=0 ate imagensTeste.quantidade faca
13.   melhor = melhorAbordagem(imagensTeste[i])
14.   melhor.restaurar(imagens.Teste[i])
15. fim para
```

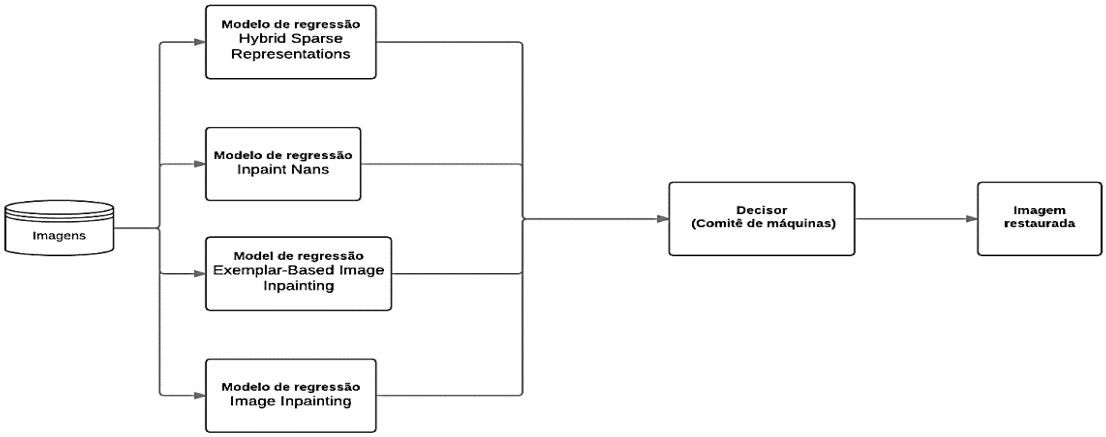


Figura 3.1. Fluxograma do algoritmo proposto.

4 EXPERIMENTOS

Este capítulo tem como objetivo apresentar o ambiente e o cenário de experimento, bem como suas configurações e construções. Além disso, serão discutidos os resultados de desempenho obtidos com o modelo proposto neste trabalho, em comparação aos algoritmos mais conhecidos da literatura. Os resultados deste trabalho mostram que combinar comitê de máquinas com modelos de regressão que processem imagens é uma solução viável para resolver problemas não linearmente separáveis. Além de facilitar a utilização de ferramentas de edição de imagem, que porventura fazem uso de algoritmos de reconstrução. Com isso, usuários de softwares como *PhotoShop* podem ter seus trabalhos otimizados e automatizados, evitando esforço humano para realizar a reconstrução da imagem e melhorando os resultados que eram alcançados com algoritmos anteriores.

Na Seção 4.1 serão apresentados, o ambiente de experimento, sua construção e configuração. Este ambiente será usado para realizar os procedimentos experimentais que estão descritos, juntamente com os resultados, na Seção 4.2.

4.1 Construção da base de imagens

Para que o modelo de regressão consiga prever com maior eficiência as medidas de similaridade, uma base de dados contendo o maior número de imagens, com estruturas diferentes, precisa ser formada [6]. É baseado nisto, que a construção do ambiente de experimento foi desenvolvido.

Para essa etapa foram utilizados algoritmos implementados em *MatLab* e em *Java.* Todas as imagens geradas foram redimensionadas para uma largura e altura de 200 *pixels* e convertidas para escala de tom de cinza, visto que alguns algoritmos de reconstrução funcionam com imagens monocromáticas.

As estruturas, texturas e estatísticas de uma imagem foram agrupados em 4 grupos distintos, sendo o primeiro formado por imagens geradas através de algoritmos sintetizadores de formas geométricas com múltiplas combinações. Esses algoritmos geram imagens a partir de uma constante, que é aplicada de formas diferentes, dependendo do tipo de figura que irá ser gerada. Por exemplo, o raio de figuras circulares ou o fator de replicação de pontos pretos e brancos. Essa constante, para todas as imagens deste grupo, é um múltiplo de 10 variando de 10 à 100. Já o segundo

conjunto é composto por fractais que foram gerados pelo algoritmo de Newton Fractal. Este algoritmo utiliza polinômios para gerar imagens cujas formas gráficas identificam regiões de atratores. Para a geração deste grupo foi alterado a ordem polinomial do método, de 1 à 10, para produzir imagens distintas. O terceiro conjunto é formado por imagens geradas por algoritmo de síntese de textura baseado em *patch,* desenvolvido neste trabalho, onde foi necessário a utilização de 4 amostras de textura para construir o conjunto. E o último contém imagens de diversas texturas reais, retiradas de outras bases. Os 4 grupos totalizam 567 imagens distintas.

Para a fase de treinamento do comitê de máquinas foram destinados 70% da base de imagens geradas. Após os modelos de regressão estarem treinados, serão postos os 30% restantes da base para a fase de testes.

4.2 Resultados

Os testes foram realizados utilizando as linguagens *Python* e *MatLab.* Para cada algoritmo analisado foram dados 30% das imagens geradas na etapa descrita na seção anterior, como entrada. Após a reconstrução, foram calculados os seus desempenhos utilizando as métricas de similaridade, PSNR e SSIM [11, 12].

Analisando a Tabela 4.1 e 4.2, é possível observar que o algoritmo proposto em seu perfeito funcionamento e treinamento, sempre escolherá a melhor abordagem para restauração. Na primeira tabela, é possível notar que alguns algoritmos se saem melhores que outros em determinadas condições. Através do comitê de máquinas, o algoritmo proposto consegue identificar a melhor solução para a determinada imagem, e com isso seus resultados são sempre melhores, como podemos ver nos resultados encontrados. Na Figura 4.1 e a Tabela 4.2, pode-se perceber que o algoritmo deste trabalho acompanha as melhores soluções, ou seja, na média ele vai se sair melhor que os demais algoritmos, por sempre ser uma das duas melhores abordagens.

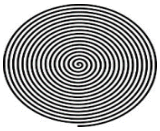
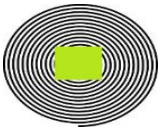
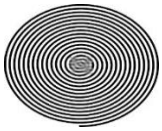
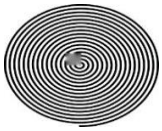
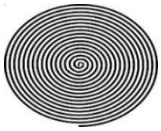
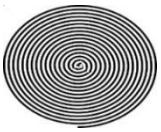
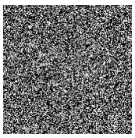
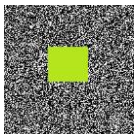
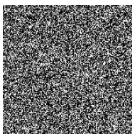
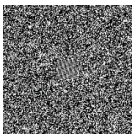
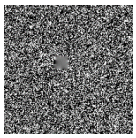
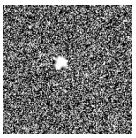
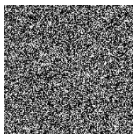
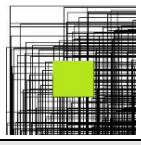
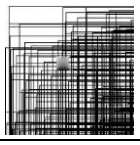
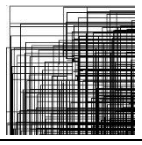
Original	Danificada	Exemplar Based	Hybrid Sparse	Inpaint Nans	Image Inpainting	Este trabalho
						
		PSNR:0.979	PSNR:0.988	PSNR:0.956	PSNR:0.956	PSNR:0.988
						
		PSNR:0.989	PSNR:0.988	PSNR:0.994	PSNR:0.944	PSNR:0.994
						
		PSNR:0.978	PSNR:0.994	PSNR:0.960	PSNR:0.963	PSNR:0.994

Tabela 4.1. Relação dos resultados obtidos com os algoritmos estudados em comparação com o algoritmo proposto, utilizando algumas amostras da base de imagens.

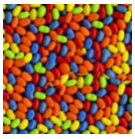
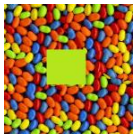
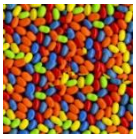
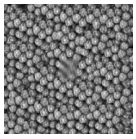
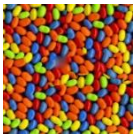
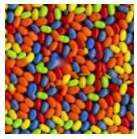
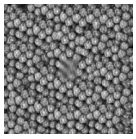
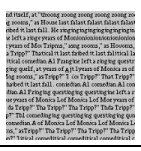
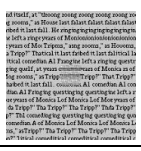
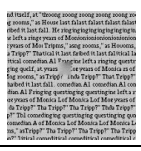
Original	Danificada	Exemplar Based	Hybrid Sparse	Inpaint Nans	Image Inpainting	Este Trabalho
						
		PSNR:0.997	PSNR:0.997	PSNR:0.985	PSNR:0.997	PSNR:0.997
						
		PSNR:0.999	PSNR:0.994	PSNR:0.985	PSNR:0.998	PSNR:0.999
						
		PSNR:0.994	PSNR:0.980	PSNR:0.986	PSNR:0.954	PSNR:0.999

Tabela 4.1(Continuação). Relação dos resultados obtidos com os algoritmos estudados em comparação com o algoritmo proposto, utilizando algumas amostras da base de imagens.

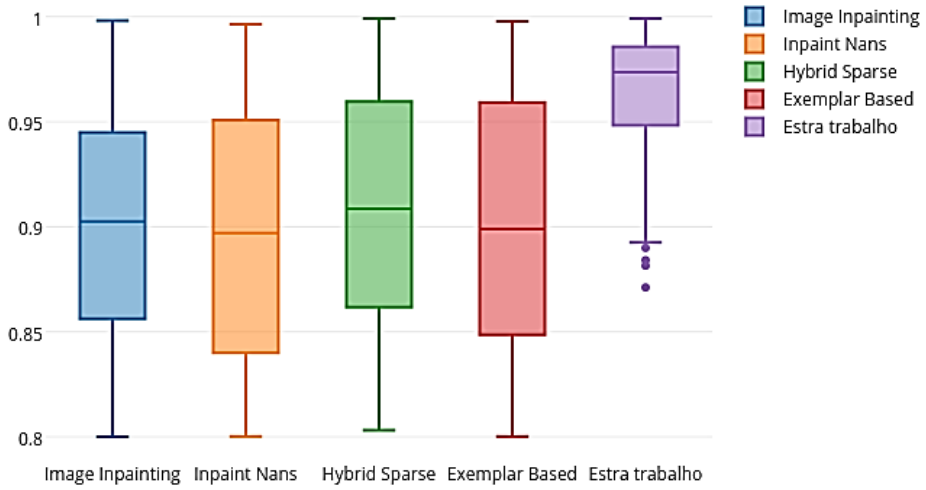


Figura 4.1. Comparação dos desempenhos (SSIM) obtidos pelos algoritmos, utilizando a base de testes.

Exemplar Based	**Hybrid Sparse**	**Inpaint Nans**	**Image Inpainting**	**Este trabalho**
0.90711	0.91411	0.90412	0.89241	0.96314

Tabela 4.2. Média dos desempenhos (SSIM) obtidos utilizando as imagens da base de teste.

5 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho de conclusão de curso foi proposto um framework para reconstrução de imagens, baseado no comitê de máquinas que utiliza modelos de regressão, métricas de similaridade e um conjunto de algoritmos de reconstrução.

Neste capítulo, serão feitas considerações finais sobre os resultados encontrados com o comparativo feito à algoritmos bem conhecidos da literatura. Na Seção 5.1 será discutido as contribuições deste trabalho, já na Seção 5.2 serão apresentados os trabalhos futuros.

5.1 Contribuições e conclusões

A necessidade pela edição de imagens é algo que vem acompanhando o avanço da tecnologia, com o surgimento de câmeras e computadores cada vez mais potentes. Hoje, o mercado de *Designers* e de ferramentas que realizam estas edições é imensa. Porém, algumas edições são muito custosas, tanto computacionalmente, quanto fisicamente para o usuário [1, 4]. Isso acontece com remoção de objetos e reconstrução de imagens, fazendo com que o usuário perca muito tempo para realizar este tipo de operação, onde muitas vezes o resultado não é satisfatório.

Neste trabalho foi desenvolvido um framework baseado no comitê de máquinas para reconstrução de imagem, onde o mesmo consegue prever qual a melhor abordagem para reconstruir uma dada imagem a partir de regressores previamente treinados, concluindo que é possível predizer um fator através de um modelo de regressão para maximizar a performance do sistema. Isso permite que a média de restaurações seja maior do que os algoritmos presentes na literatura, visto que ele sempre apresentará a melhor técnica. Este algoritmo usa modelo de regressão e métricas de similaridade para definir as características da imagem que será reconstruída, e assim definir qual melhor método de restauração.

A partir de testes realizados, foi demonstrado que a eficácia do algoritmo proposto é consideravelmente superior aos demais da literatura. Onde o mesmo sempre está entre as melhores abordagens, sendo na média o melhor algoritmo para reconstrução. Em todos os testes ele obteve o melhor resultado.

5.2 Trabalhos Futuros

Como trabalhos futuros desta proposta, pode ser analisado o comportamento do desempenho deste algoritmo realizando combinações dos algoritmos de restauração do repositório, baseado nas estruturas e padrões das imagens que são dadas como entrada. Por exemplo, se para uma determinada imagem dois algoritmos obtiveram bons desempenhos, sendo estes algoritmos distintos, com abordagens diferentes, a combinação dessas duas técnicas podem resultar em uma recuperação melhor. Além disso, essas imagens podem ser mapeadas pelas suas características e através disso determinar quais são as melhores abordagens e cálculos para uma restauração mais eficaz.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Ashikhmin, M. Synthesizing natural textures. *ACM Symposium on Interactive 3D Graphics*, p. 217-226, 2001.

[2] Bertalmio, M.; Ballester, C.; Sapiro, G.; Caselles, V. Image inpainting. *ACM Conf. Comp. Graphics (SIGGRAPH*), p. 417-424, 2000.

[3] Efro, A.; Leung, T. Texture synthesis by non-parametric sampling. *Int. Conf. Computer Vision*, p. 1033-1038, 1999.

[4] Criminisi, A.; Perez, P.; Toyama, K. Region Filling and Object Removal by Exemplar-Based Image Inpainting. *Conf. Comp. Vision Pattern Rec*, 2003.

[5] Aparecido, C. Comitê de Máquinas: Uma abordagem unificada empregando máquinas de vetores-suporte. Tese de pós-graduação, Engenharia da Computação, Universidade Estadual de Campinas, 2004.

[6] Norman, R.; Smith, H. Applied Regression Analysis. 3rd, 1998.

[7] Hore, A.; Ziou, D. Image Quality Metrics: PSNR vs. SSIM. *Pattern Recognition (ICPR)*, 2010.

[8] Bertalmio, M.; Sapiro, G.; Vese, L.; Osher, S. Simultaneous structure and texture image inpainting. *Conf. Comp. Vision Pattern Rec,* 2003.

[9] D'Errico, J. Interpolates (& extrapolates) NaN elements in a 2d array. Pacote de funções MatLab, 2012.

[10] Li, X. Regularized Image Recovery via Hybrid Sparse Representations: a Deterministic Annealing Approach. Departamento de ciência da computação e engenharia elétrica da West Virginia University, 2011.

[11] Huynh-Thu, Q.; Ghanbari, M. Scope of validity of PSNR in image/video quality assessment. Electronics Letter, Volume 44, p. 800-8001, 2008.

[12] Ndajah, P.; Kikuchi, H.; Yukawa, M.; Watanabe, H.; Muramatsu, S. SSIM Image Quality Metric for Denoised Images. Departamento de engenharia elétrica e eletrônica da Niigata University, 2010.

[13] Steve, R. Support Vector Machines for Classification and Regression. Departamento de eletrônica e ciência da comutação da University of Southampton, 1998.

[14] Joachims, T. Making large-scale SVM learning practical. University of Dortmund , 1998.

[15] Drucker, H.; Chris, J.; Kaufman, L.; Smola, A.; Vapnik, V. Support Vector Regression Machines. Departamento de engenharia eletrônica, Monmouth University, 1996.