UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

Dissertação de Mestrado

Visualização de Camadas Intermediárias de Redes Neurais Convolucionais de Transformação de Imagem

Églen da Veiga Protas

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Computação

Orientador: Prof. Dr. Paulo Lilles Jorge Drews Junior
Co-orientador: Prof. Dr. Adriano Velasque Werhli

Rio Grande, 2017

Ficha catalográfica

P967v Protas, Églen da Veiga.
Visualização de camadas intermediárias de redes neurais convolucionais de transformação de imagem / Églen da Veiga Protas. – 2017.
148 p.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-graduação em Engenharia da Computação, Rio Grande/RS, 2017.
Orientador: Dr. Paulo Lilles Jorge Drews Junior.
Coorientador: Dr. Adriano Velasque Werhli.

1. Redes Neurais 2. Visualização 3. *Deep Learning* 4. Restauração de imagens 5. Maximização da ativação I. Drews Junior, Paulo Lilles Jorge II. Werhli, Adriano Velasque III. Título.

CDU 004.92

Catalogação na Fonte: Bibliotecário Me. João Paulo Borges da Silveira CRB 10/2130

UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

Dissertação de Mestrado

Visualização de Camadas Intermediárias de Redes Neurais Convolucionais de Transformação de Imagem

Églen da Veiga Protas

Banca examinadora:

Prof. Dr. Ricardo Matsumura Araujo

Profª. Drª. Sílvia Silva da Costa Botelho

Prof. Dr. Paulo Lilles Jorge Drews Junior
Orientador

AGRADECIMENTOS

À Agência Nacional do Petróleo, Gás Natural e Biocombustível - ANP e à Petrobras, pelo apoio financeiro prestado através do Programa de Recursos Humanos - PRH, sem o qual a realização deste trabalho não seria possível.

Aos meus colegas e professores do NAUTEC, que me deram todo o apoio necessário durante a realização deste projeto.

À minha família, pelo apoio, paciência e compreensão.

RESUMO

PROTAS, Églen da Veiga. **Visualização de Camadas Intermediárias de Redes Neurais Convolucionais de Transformação de Imagem**. 2017. 148 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande, Rio Grande.

As Redes Neurais Convolucionais são um modelo de aprendizado supervisionado que nos últimos anos tem se tornado o estado da arte em diversas aplicações da área de visão computacional, como reconhecimento de carácteres, classificação de imagens e detecção de objetos. Apesar do grande poder deste modelo, ele possui algumas desvantagens, entre elas a dificuldade de se compreender como os seus diversos parâmetros se relacionam para chegar a resposta final. Recentemente, algumas técnicas de visualização foram desenvolvidas com o objetivo de auxiliar na compreensão do funcionamento interno de uma rede neural, e o conhecimento obtido através da aplicação destas técnicas foi utilizado para melhorar o desempenho das arquiteturas em questão. Estas técnicas, porém, foram desenvolvidas para e aplicadas em redes de classificação de imagens. O objetivo deste trabalho é estudar os métodos de visualização existentes e avaliar a sua aplicação em redes neurais destinadas a problemas de transformação de imagem, que são aqueles onde a entrada e a saída são uma imagem, geralmente do mesmo tamanho. Foram utilizadas como estudo de caso redes neurais relacionadas aos problemas de estimativa de profundidade, remoção de névoa e restauração de imagens subaquáticas. A aplicação de métodos de visualização nestes modelos permitiu uma melhor compreensão sobre os mesmos, que pode ajudar no desenvolvimento de arquiteturas melhores e mais eficientes.

Palavras-chave: Redes Neurais, Visualização, Deep Learning, Restauração de Imagens, Maximização da Ativação.

ABSTRACT

PROTAS, Églen da Veiga. **Visualization of Intermediate Layers of Image Transformation Convolutional Neural Networks**. 2017. 148 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande, Rio Grande.

Convolutional Neural Networks are a supervised learning model that in recent years has become the state-of-the-art in many fields of computer vision, such as character recognition, image classification and object detection. Despite its power, this model has some disadvantages, among them the difficulty in understanding how the model parameters are related in order to reach the network's final answer. Recently, some visualization techniques have been developed with the objective of helping the understanding of the inner working of a neural network, and the knowledge obtained through the application of these techniques was used to improve the performance of the architectures in question. Those techniques, however, were developed for and applied to image classification networks. The objective of this work is to study existing visualization methods and evaluate their application in neural networks intended to image transformation problems, i.e., problems where both the input and output are images, usually of the same size. Convolutional Neural Networks related to the problems of depth estimation, image dehazing and underwater image restoration were used as case studies. The application of visualization methods in these models allowed a better understanding of them, which may help in the development of better and more efficient architectures.

Keywords: Convolutional Neural Networks, Deep Learning, Visualization, Image Restoration, Activation Maximization.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

BReLu	Bilateral Rectified Linear Unit
CAM	Campo Aleatório de Markov
CPPN	Compositional Pattern-Producing Network
DCP	Dark Channel Prior
ELU	Exponential Linear Unit
GPU	Graphics Processing Unit
ILSVRC	ImageNet Large Scale Visual Recognition Competition
LReLu	Leaky Rectified Linear Unit
MLP	Multilayer Perceptron
PReLu	Parametric Rectified Linear Unit
ReLu	Rectified Linear Unit
RMS	Root Mean Square
RN	Rede Neural
RNC	Rede Neural Convolucional
SGD	Stochastic Gradient Descent
SSDA	Stacked Sparse Denoising Autoencoders
UDCP	Underwater Dark Channel Prior

SUMÁRIO

1 INTRODUÇÃO

Nos últimos anos Redes Neurais (RNs) tem sido aplicadas a vários problemas da área de visão computacional, como classificação de imagens [36], reconhecimento de caracteres [38], detecção de objetos [53], remoção de ruído [30] e colorização de imagens em preto e branco [78]. Apesar dos princípios básicos por trás do funcionamento das RNs serem conhecidos desde os anos 80, durante muitos anos a grande quantidade de recursos computacionais necessária para o seu treinamento dificultou a aplicação prática de RNs em problemas reais. Está situação, porém, tem se modificado nos últimos anos, devido a recentes avanços em hardware, como o surgimento de computação de propósito geral em GPUs, que proporcionou uma grande redução no tempo de treinamento de RNs com arquiteturas complexas. Outro fator que contribuiu para a popularização das RNs foi a disponibilidade de grandes conjuntos de dados rotulados, como o ImageNet [12], que permitiu o treinamento de redes neurais destinadas à classificação de imagens.

Redes Neurais são uma família de modelos computacionais biologicamente inspirados, com uma organização baseada na estrutura cerebral de animais. Elas consistem em uma série de unidades de processamento, denominadas neurônios, que implementam funções não lineares de suas entradas. Os neurônios são organizados em camadas, que são interconectadas entre si. O processamento de uma entrada por uma rede neural se da pela passagem da entrada por diversas camadas de neurônios, até a camada de saída, que fornece a resposta final. Em geral, quanto maior o número de camadas, maior o poder da rede, e maior o seu custo computacional.

Uma das características mais importantes das redes neurais é a utilização de aprendizado de máquina, que é o ajuste automático de parâmetros através da observação de um conjunto dados de treinamento. Em uma rede neural, os parâmetros de cada neurônio são aprendidos automaticamente através de um processo de treinamento. O primeiro passo deste processo consiste em propagar uma entrada através da rede e comparar a resposta obtida com a resposta correta. A diferença entre as duas respostas é calculada através de uma função de erro. Com base no resultado desta função de erro calcula-se o gradiente, que é um vetor onde cada posição representa a contribuição individual de um parâmetro da rede para o resultado da função de erro. O cálculo do gradiente normalmente é reali-

zado através de um método chamado *backpropagation* [55]. Os parâmetros da rede então são ajustados através de um método de otimização, como por exemplo gradiente descendente estocástico. Esse processo é repetido iterativamente até a sua convergência, que ocorre quando não se consegue mais reduzir o erro da rede. No caso de redes complexas o treinamento pode demorar muito tempo e demandar grandes recursos computacionais. Para que a rede seja treinada corretamente também é necessário um grande conjunto de dados de entrada rotulados, ou seja, onde a resposta esperada é conhecida.

As Redes Neurais Convolucionais (RNCs) são redes neurais onde as conexões entre camadas são organizadas como em uma operação de convolução. Todos os neurônios de uma RNC estão associados a uma posição espacial específica, e cada neurônio está conectado apenas com os neurônios da camada anterior que estão em uma posição espacial próxima. As camadas de uma RNC estão organizadas em planos, que são chamados de *feature maps*. Todos os neurônios de um mesmo *feature map* compartilham o mesmo conjunto de parâmetros. Desta forma, cada *feature map* equivale à aplicação uma operação de convolução sobre o resultado da camada anterior. Estas características permitem uma redução no número de parâmetros da rede, o que facilita o treinamento de redes muito profundas.

Nos últimos anos RNCs se tornaram o estado da arte em muitas aplicações de visão computacional. O vencedor das categorias de classificação de imagens e localização de objetos da edição 2012 da competição ILSVRC (ImageNet Large Scale Visual Recognition Competition) foi a arquitetura de Rede Neural Convolucional AlexNet, proposto em [36]. Desde então o vencedor das tarefas de classificação de imagens, localização e detecção de objetos de todas as edições da ILSVRC tem sido uma rede neural convolucional. Atualmente o estado da arte em classificação de imagens é a arquitetura Inception-ResNet [66].

1.1 Visualização de Redes Neurais Convolucionais

Apesar de serem amplamente utilizadas na resolução de vários problemas da área de visão computacional, a compreensão sobre o funcionamento das RNCs e a razão do seu desempenho ainda é limitada. Muitos dos avanços recentes na área foram alcançados empiricamente, sem levar em conta as características do problema específico que se deseja resolver. A grande quantidade de parâmetros e a presença de relações não lineares entre eles dificultam bastante a análise de uma rede neural. As estratégias que cada rede utiliza dependem da natureza do problema ao qual ela é aplicada, da sua arquitetura, e da forma como ela foi treinada. Um melhor conhecimento sobre o funcionamento interno de um modelo específico ajuda na previsão de como ele vai se comportar em situações inesperadas, além de contribuir para o desenvolvimento de soluções mais eficientes.

Uma forma de compreender o funcionamento de uma rede neural é descobrir o que é

computado em suas camadas intermediárias. Em redes que recebem imagens como entrada, isto pode ser feito através de métodos de visualização. Em geral, algoritmos de processamento de imagens trabalham com a detecção de *features*, características da imagem que são consideradas relevantes para a resolução do problema em questão. Partindo do princípio que as redes neurais utilizam estratégias semelhantes, a maioria dos métodos de visualização busca descobrir quais *features* a rede aprendeu a detectar em cada uma das suas camadas. Um dos métodos de visualização mais simples é a visualização direta dos *kernels* de convolução. Este método, porém, só gera resultados relevantes para os *kernels* que são aplicados diretamente sobre imagens RGB. Um outro método simples é a visualização direta, em forma de imagem, das ativações produzidas por uma determinada entrada em uma camada da rede predefinida. Outra estratégia frequentemente utilizada é determinar quais as entradas de um determinado conjunto (o conjunto de treinamento, por exemplo) produzem as maiores ativações em um *feature map* específico. Existem ainda métodos mais avançados, que utilizam otimização por gradiente ascendente para sintetizar uma imagem que produz uma ativação muito alta em um *feature map* previamente selecionado. Cada um destes métodos possui as suas vantagens e desvantagens. A melhor forma de se analisar uma rede neural é utilizar diferentes métodos de visualização em conjunto.

1.2 Problemas de Transformação de Imagem

Problemas de transformação de imagem são aqueles onde a entrada e a saída são uma imagem, como por exemplo remoção de ruído, colorização de imagens monocromáticas, remoção de névoa e restauração de imagens subaquáticas. Apesar de terem alguns aspectos em comum, problemas de classificação e transformação de imagens são consideravelmente diferentes. Por esta razão, arquiteturas de RNC de classificação não podem ser aplicadas diretamente em problemas de transformação de imagem. Em uma rede de classificação, a camada de saída possui um número de neurônios igual ao número de categorias possíveis, enquanto em uma rede de restauração é necessário um neurônio para cada canal de cor de cada pixel da imagem de saída. As diferenças, porém, não estão limitadas à camada de saída. As arquiteturas de classificação costumam utilizar redução de dimensão através de *max pooling*, que é uma operação que descarta parte da informação de entrada considerada irrelevante para o resultado final. Já em alguns problemas transformação de imagem, como na remoção de ruído e na remoção de névoa, praticamente toda informação de entrada é importante, e por isso não pode ser descartada. Devido a estas diferenças, o desenvolvimento de RNCs destinadas a problemas de transformação de imagem não é uma tarefa tão simples quanto a adaptação de uma arquitetura que apresenta bons resultados na tarefa de classificação.

1.3 Objetivos

Até agora, a grande maioria dos estudos associados à visualização de RNCs foi aplicada a redes de classificação de imagens. Devido às diferenças entre os problemas, os resultados destes estudos não necessariamente são válidos para redes de transformação de imagem. O objetivo deste trabalho é estudar os métodos de visualização presentes na literatura e avaliar a aplicação destes métodos em redes neurais convolucionais destinadas a problemas de transformação de imagem.

Outro objetivo é descobrir quais *features* estão presentes nas camadas intermediárias das arquiteturas estudadas. No caso das redes de remoção de névoa e restauração de imagens subaquáticas, são esperados *features maps* relacionados a características utilizadas como indicadores de profundidade nos métodos tradicionalmente utilizados para tratar estes problemas. Nas redes de remoção de névoa, estas características incluem o contraste local [68], o valor mínimo entre os três canais de cor [23] e a diferença entre os canais valor e saturação no espaço de cor HSV [79]. Também busca-se por um mapa de transmissão da imagem nas camadas intermediárias da rede. No caso das redes de restauração de imagens subaquáticas, além dos *features* presentes em redes de remoção de névoa são esperados *feature maps* relacionados à presença ou ausência da cor vermelha, que é um indicador de distância em ambientes subaquáticos [14]. Outra hipótese é que o problema de restauração seja tratado como um problema de classificação, ou seja, que a rede seja capaz de identificar texturas específicas e, com base nos exemplos apresentados durante o treinamento, determinar que aparência estas texturas devem ter na ausência de turbidez. Caso esta hipótese se confirme, devem ser encontrados *feature maps* dedicados à detecção das texturas mais comuns no conjunto de treinamento.

1.4 Estudos de Caso

São apresentados cinco estudos de caso neste trabalho: Uma rede de estimativa de profundidade [37], uma rede de estimativa do mapa de transmissão de imagens com névoa [5], uma rede de restauração de imagens subaquáticas baseada na arquitetura de estimativa de profundidade apresentada em [16], uma rede residual de remoção de névoa e uma rede residual de restauração de imagens subaquáticas.

Em cada um destes modelos são aplicadas diferentes técnicas de visualização, com o objetivo de compreender melhor como os seus parâmetros se relacionam para chegar ao resultado final. O conhecimento adquirido com os resultados é utilizado para identificar potenciais problemas nas arquiteturas e encontrar formas de torna-las mais eficientes.

1.5 Contribuições

Neste trabalho são apresentadas as seguintes contribuições:

- A avaliação da aplicação de métodos de visualização de redes neurais em redes neurais de transformação de imagem
- Uma combinação de métodos de regularização para maximização da ativação que melhora a interpretabilidade dos resultados
- O método de visualização por inversão da rede, desenvolvido para redes de transformação de imagem

1.6 Organização do Texto

No capítulo 2 são apresentados os métodos de visualização de redes neurais convolucionais mais populares. No capítulo 3 são apresentados alguns problemas de transformação de imagem, assim como arquiteturas de redes neurais aplicadas a resolução destes problemas. No capítulo 4 explica-se a metodologia deste trabalho, e no capitulo 5 são apresentados os resultados. O capítulo 6 apresenta um breve resumo dos resultados obtidos a as conclusões tiradas com base neles. No apêndice A são apresentados alguns métodos tradicionais para a solução do problema de *dehazing*, e no apêndice B são apresentadas algumas técnicas para restauração de imagens subaquáticas.

2 TRABALHOS RELACIONADOS

Nos últimos anos Redes Neurais Convolucionais têm sido o estado da arte em várias aplicações da área de visão computacional. Apesar disso, a compreensão sobre como elas funcionam ainda é limitada, principalmente em relação às computações realizadas nas camadas intermediárias. As RNCs podem ser consideradas como um sistema caixa preta, já que podem ser analisadas em termos de suas entradas e saídas, sem a necessidade de conhecimento do seu funcionamento interno. A análise do funcionamento interno de uma RNC é muito difícil devido ao grande número de partes não lineares que interagem entre si. O tamanho das RNCs modernas, que podem ter até dezenas de milhões de parâmetros, dificulta ainda mais esta análise. Uma melhor compreensão sobre o funcionamento interno das RNCs poderia levar ao desenvolvimento de arquiteturas mais poderosas. Isso motivou o desenvolvimento de várias técnicas para a visualização de redes neurais.

2.1 Métodos de Visualização Direta

Uma abordagem é a visualização direta dos pesos. Em redes convolucionais que recebem imagens como entrada os pesos pode ser visualizados como filtros de convolução. A principal desvantagem desta técnica é que ela só gera resultados de fácil interpretação quando aplicada aos pesos da primeira camada, que recebem imagens como entrada, e por isso tem um ou três canais na terceira dimensão e, logo, podem ser visualizados na forma de imagens [17]. Os pesos das camadas mais profundas normalmente são aplicados a entradas com muito mais de três dimensões, o que impossibilita a sua visualização em forma de imagem. A visualização direta dos *kernels* convolucionais é comum na literatura. Os filtros da primeira camada costumam ter a forma de detectores de traços em redes treinadas para classificar dígitos desenhados à mão e de detectores de bordas em redes treinadas para classificação de imagens naturais [17].

Uma outra abordagem é a visualização direta da ativação das camadas intermediárias. Em [75] a visualização direta das ativações de uma rede de classificação mostrou que existem camadas especializadas na detecção de estruturas específicas, como flores, frutas, texto e faces de pessoas e animais. Apesar de estes conceitos não corresponderem

a nenhuma das mais de mil classes que a rede foi treinada para identificar, a rede aprendeu a identifica-los porque eles representam informações parciais úteis na identificação da classe da imagem. Uma desvantagem desta técnica é que a sua aplicação em camadas *fully-connected* não produz resultados espacialmente informativos, já que a ordem das entradas nessas camadas é irrelevante [75]. Uma outra potencial desvantagem desta técnica é que para se compreender a função de todos os canais de todas as camadas da rede poderia ser necessário visualizar as ativações produzidas por todo o conjunto de treinamento. É possível que alguns dos *feature maps* da rede tenham se especializado em detectar estruturas específicas que estão presentes em apenas alguns exemplos de treinamento, o que dificultaria muito a descoberta da função desses *feature maps* através da visualização de imagens aleatórias.

Outro método de visualização é mostrar as imagens do conjunto de treinamento ou do conjunto de teste que causam a maior ativação em um neurônio ou *feature map* específico.

2.2 Deconvolução

A Deconvolução [76] é um método que destaca quais posições de uma imagem particular são responsáveis por causar a ativação de um *feature map* específico. O método consiste em projetar a ativação dos *features* no espaço dos pixels de entrada usando uma Rede Deconvolucional [77].

Uma Rede Deconvolucional (deconvnet) pode ser vista como uma Rede Convolucional que utiliza os mesmos componentes (convolução, *pooling*), mas ao inverso. O resultado disso é que, ao invés de mapear pixels para *features*, ela faz o oposto. As Redes Deconvolucionais foram propostas em [77] como uma forma de se realizar aprendizado supervisionado, mas foram aplicadas em [76] como uma forma de testar uma Rede Convolucional já treinada.

Para se examinar uma Rede Convolucional, uma Rede Deconvolucional é ligada em cada uma das camadas que se deseja visualizar, criando um caminho contínuo de volta aos pixels da imagem. O processo de visualização é iniciado mostrando uma imagem para a Rede Convolucional e calculando os *features* através das suas camadas. Para se examinar uma dada ativação, todas as outras ativações da camada são zeradas e os *feature maps* são passados como entrada para a camada da deconvnet ligada a ela. Esses *feature maps* então passam por operações de (i) *unpooling*, (ii) ReLu e (iii) convolução transposta para que a ativação da camada anterior que deu origem à ativação selecionada possa ser reconstruída. Esse processo é repetido até se chegar ao espaço de pixels da imagem de entrada. O diagrama de uma rede deconvolucional é apresentado na Figura 1.

Unpooling: É o inverso da operação de *max pooling*. Apesar da operação de *max pooling* não ser reversível, é possível se obter uma aproximação da sua inversão guardando a posição que gerou a ativação máxima em cada região de *pooling*. Em uma de-

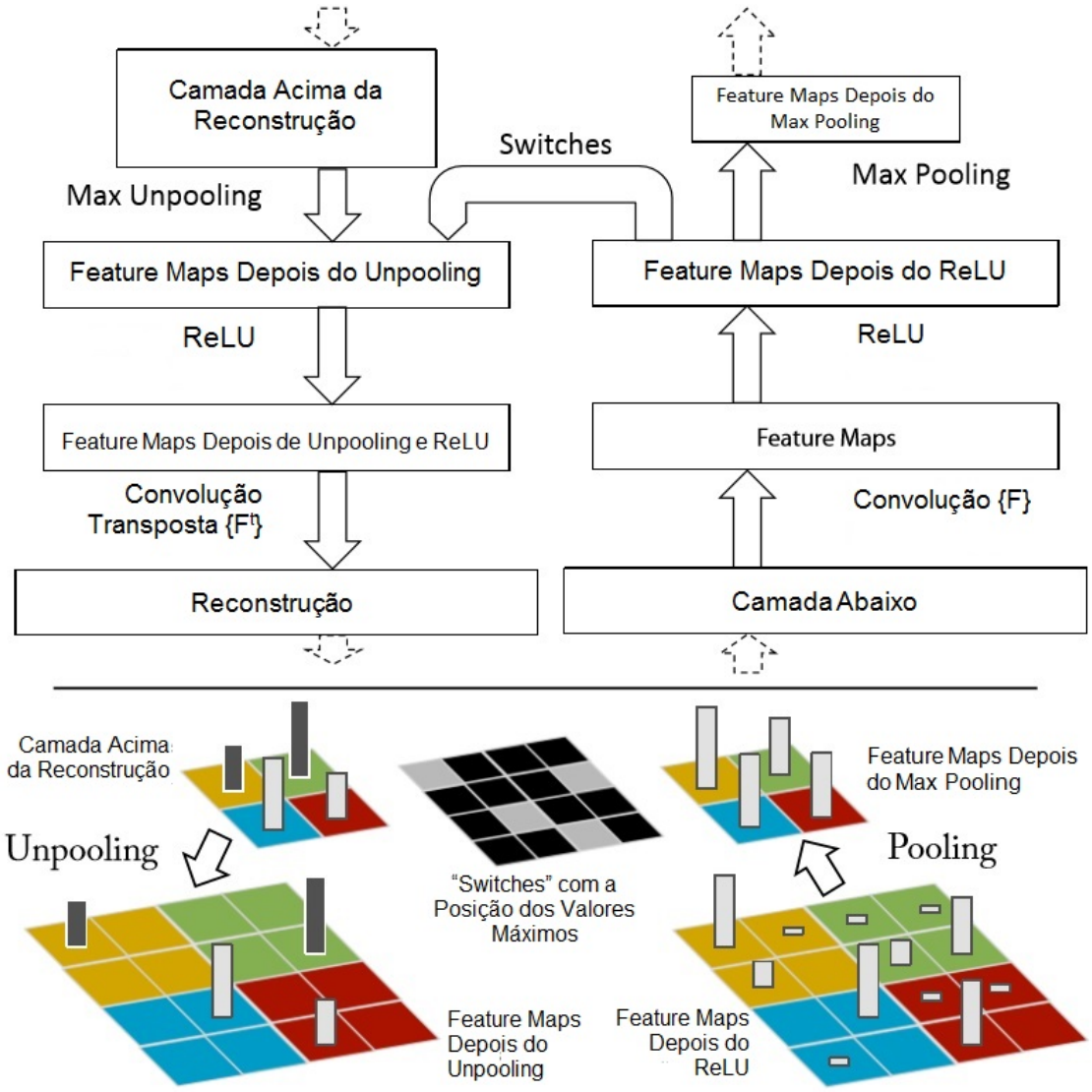


Figura 1: Diagrama da reconstrução por deconvolução. Topo: Uma camada deconvolucional (à esquerda) ligada a uma camada convolucional (à direita). A rede deconvolucional reconstrói uma versão aproximada dos *feature maps* da camada convolucional anterior. Em baixo: Uma ilustração da operação de *unpooling*, usando *switches* que guardam a posição do máximo local em cada região de *pooling* (zonas coloridas) de uma rede convolucional. Fonte: [76].

Figura 2: Visualização através de deconvolução dos *features* de uma rede neural convolucional de classificação treinada. Para as camadas 2-5 são mostradas as 9 maiores ativações de um subconjunto aleatório de *feature maps* encontradas nos dados de validação, projetadas no espaço dos pixels através de uma rede deconvolucional. Para cada *feature map* também são mostrados os *patches* correspondentes. Pode-se perceber um exagero das partes discriminativas da imagem, como por exemplo olhos e narizes de cachorros. Fonte: [76]. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

convnet a operação de *unpooling* é realizada colocando a reconstrução da camada superior na posição apropriada, preservando assim a estrutura que gerou o estímulo.

Relu: As redes convolucionais normalmente utilizam a não-linearidade ReLu para garantir que a ativação dos *feature maps* seja sempre positiva. Para se obter reconstruções válidas, que também devem ser positivas, em cada camada, o sinal reconstruído é passado através de uma não-linearidade ReLu. O uso de funções de ativação diferentes provavelmente iria requerer alterações nesta etapa, já que seria necessário encontrar uma função que revertesse o efeito da função de ativação utilizada.

Convolução transposta: Uma Rede Convolucional utiliza filtros aprendidos durante o treinamento para realizar uma operação de convolução sobre os *feature maps* da camada anterior. Para inverter esta operação a deconvnet utiliza versões transpostas dos mesmos filtros, mas aplicadas aos *feature maps* reconstruídos pelas operações de *unpooling* e ReLu, não no resultado da operação de convolução original. Na prática, isto significa inverter cada filtro na horizontal e na vertical.

A reconstrução obtida de uma única ativação lembra uma pequena parte da imagem de entrada original, com estruturas que tem um peso de acordo com a sua contribuição para a ativação do *feature*. Isso implicitamente mostra quais partes da imagem de entrada são discriminativas, ou seja, quais partes são importantes para a tomada de decisão da rede. Um exemplo de visualização por deconvolução é apresentado na Figura 2.

Uma desvantagem deste método é a sua complexidade de implementação, que requer a alteração de toda a estrutura da rede. Além disso, pode ser necessária uma grande quantidade de imagens para se determinar quais características são responsáveis pela ativação de um *feature map* específico.

2.3 Maximização da Ativação

A Maximização da Ativação, introduzida em [17], é uma forma de gerar uma imagem que maximiza uma função arbitrária, que pode ser a ativação de um neurônio da rede ou a ativação média de um *feature map*.

Na Maximização da Ativação a busca pela ativação máxima de um neurônio é tratada como um problema de otimização. Seja θ a notação dos parâmetros da rede (pesos e bias) e seja $h_{ij}(\theta, \mathbf{x})$ a ativação de uma dada unidade i de uma dada camada j na rede; h_{ij} é uma função de θ e da entrada $\mathbf{x}$. Assumindo θ como fixo (por exemplo, os parâmetros de uma rede já treinada), o método pode ser visto como a busca por

$$\mathbf{x}^* = \underset{\mathbf{x}\ s.t.\ \|\mathbf{x}\|=\rho}{\arg\max}\ h_{ij}\left(\theta, \mathbf{x}\right).$$

Isto, em geral, é um problema e otimização não convexo, porém é possível buscar por um mínimo local. A maneira mais fácil de se fazer isso é realizar uma otimização por

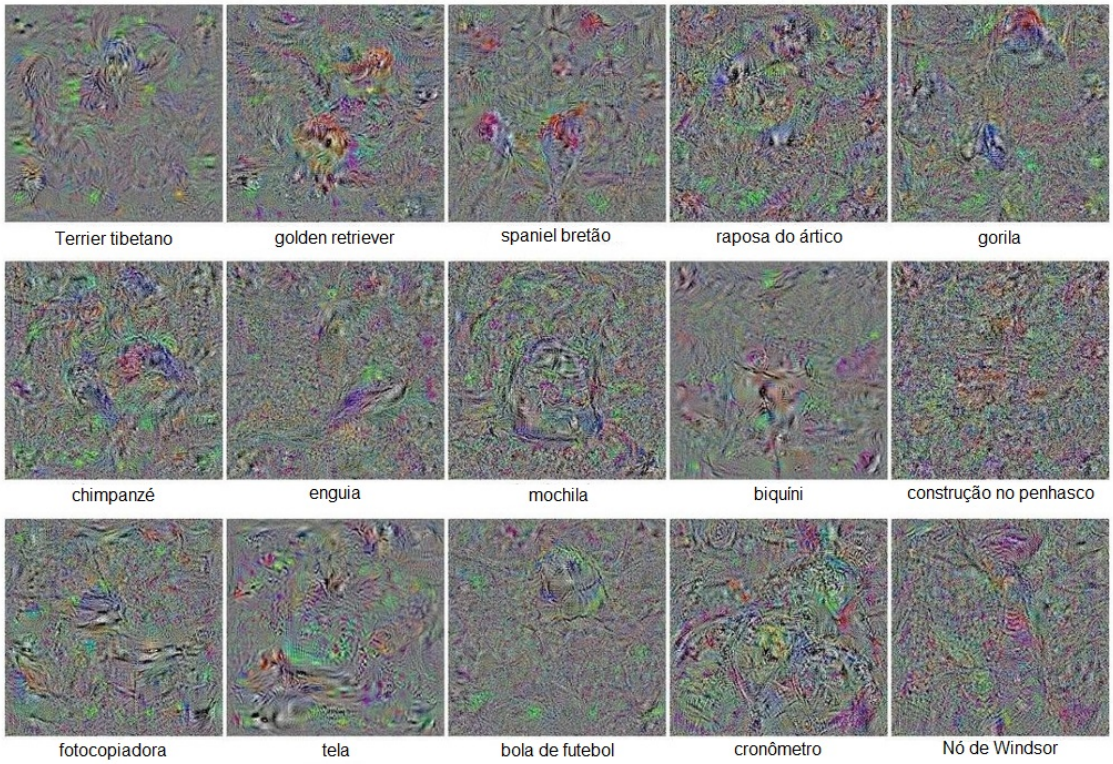


Figura 3: Maximização da ativação de neurônios da camada de saída de uma rede neural de classificação. Cada resultado é uma imagem sintetizada para maximizar o escore de uma classe de saída. Fonte: [47]. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

gradiente ascendente no espaço da imagem, ou seja, computar o gradiente de $h_{ij}(\theta, \mathbf{x})$ e mover $\mathbf{x}$ na direção deste gradiente.

Essa técnica de otimização, chamada em [17] de maximização da ativação, pode ser aplicada em qualquer rede onde o gradiente de $h_{ij}(\theta, \mathbf{x})$ pode ser computado. Como todas as técnicas de gradiente descendente ela envolve uma escolha de hiperparâmetros: a taxa de aprendizagem e o critério de parada, que pode ser o número de iterações.

Métodos baseados em gradiente também podem ser utilizados como um alternativa à deconvolução. Segundo [62], a reconstrução por deconvolução da ativação produzida em um neurônio n por uma entrada $\mathbf{x}$ é similar ou equivalente ao gradiente da ativação de n em relação à $\mathbf{x}$.

A maximização da ativação é de fácil implementação, porém ela possui algumas desvantagens. Quando aplicada em RNCs de classificação para maximizar a ativação que indica a probabilidade da imagem de entrada pertencer uma determinada classe ela, costuma gerar resultados de difícil interpretação, que não lembram imagens naturais [62, 47, 75]. As imagens sintetizadas possuem padrões que lembram a classe esperada, porém estes padrões se apresentam de forma desorganizada, sem estrutura global. Também podem ser observados muitos pixels com valores extremos e padrões de alta frequência. Alguns exemplos de visualização por maximização da ativação são apresentados na Figura 3.

2.3.1 Regularização

A maximização da ativação tende a gerar resultados de difícil interpretação, que não lembram imagens reais. Em [75] são apresentadas técnicas de regularização que buscam amenizar este problema. A regularização é um processo que busca estimular um determinado comportamento no resultado de uma operação. Normalmente ela é implementada através de operadores que penalizam ou beneficiam determinadas características da entrada.

Formalmente, considere uma imagem $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$, onde $C = 3$ é o número de canais de cores, e a altura H e a largura W são valores fixos. Quando esta imagem é apresentada a uma rede neural, ela produz uma ativação $a_i(\mathbf{x})$ em um neurônio i, onde i é um índice que cobre todos os neurônios em todas as camadas da rede. É definida uma função de regularização parametrizada $R_\theta(\mathbf{x})$ que penaliza a imagem de várias formas. O problema de otimização pode ser apresentado como a busca por uma imagem $\mathbf{x}^*$ onde

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \left(a_i(\mathbf{x}) - R_\theta(\mathbf{x}) \right).$$

Na prática, a formulação usada em [75] é um pouco diferente. A regularização é definida por um operador $r_\theta(\cdot)$ que mapeia $\mathbf{x}$ para uma versão um pouco mais regularizada de si mesmo. Esse método de otimização pode ser facilmente implementado alternando-se entre passos na direção do gradiente de $a_i(\mathbf{x})$ e passos na direção dada por r_θ. Com um passo na direção do gradiente de tamanho η, cada iteração desse processo aplica a atualização:

$$\mathbf{x} \leftarrow r_\theta \left(\mathbf{x} + \eta \frac{\partial a_i}{\partial x} \right).$$

Quatro métodos de regularização são apresentados em [75], todos desenvolvidos para amenizar os diferentes problemas frequentemente encontrados nos resultados da maximização da ativação sem regularização.

L_2 decay: L_2 decay é um método de regularização comum que penaliza valores grandes. Ele é implementado como $r_\theta(\mathbf{x}) = (1 - \theta_{\text{decay}}) \cdot \mathbf{x}$. A regularização por L_2 decay tende a prevenir que um pequeno número de pixels com valores extremos domine a imagem. Estes pixels isolados com valores extremos não costumam ocorrer em imagens naturais e podem prejudicar a visualização.

Filtro Gaussiano: A sintetização de imagens por maximização da ativação costuma gerar imagens com informação de alta frequência. Estas imagens provocam uma alta ativação, porém elas não são realistas nem interpretáveis [47]. Uma forma de penalizar estas informações de alta frequência é a regularização por filtro Gaussiano. Filtro Gaussiano é um processo que borra a imagem, produzindo um efeito similar a uma imagem desfocada. Em [75] a regularização por filtro Gaussiano é implementada como

$r_\theta(\mathbf{x}) = \mathrm{GaussianBlur}(\mathbf{x}, \theta_{\mathrm{b\_width}})$. A operação de convolução com um filtro Gaussiano tem um custo computacional maior que o dos outros métodos de regularização, por isso foi utilizado um outro hiperparâmetro, $\theta_{\mathrm{b\_every}}$. Este hiperparâmetro controla a frequência com que o filtro é aplicado, ou seja, a operação de filtragem ocorre a cada $\theta_{\mathrm{b\_every}}$ iterações. Filtrar uma imagem múltiplas vezes com um filtro Gaussiano de tamanho pequeno é equivalente a filtrar uma única vez com um filtro maior, e o efeito será similar mesmo que a imagem mude um pouco durante o processo de otimização, logo a utilização de filtros pequenos permite uma redução no custo computacional sem limitar a expressividade da regularização.

Corte de pixels com módulo baixo: As duas primeiras formas de regularização tendem a suprimir informações de alta frequência e grande amplitude, logo a aplicação de ambas leva a um $\mathbf{x}^*$ com valores relativamente pequenos e com variações relativamente suaves. A imagem, porém, ainda tende a ter pixels com valor diferentes de zero por toda a parte. Mesmo que alguns dos pixels em $\mathbf{x}^*$ mostrem o objeto principal ou o padrão que causa a ativação do neurônio em questão, o gradiente em relação a todos os outros pixels em $\mathbf{x}^*$ geralmente é diferente de zero, o que faz com que esses pixels formem algum tipo de padrão que contribui um pouco para aumentar a ativação do neurônio escolhido. Esse comportamento prejudica a visualização do objeto principal, e por isso deseja-se evita-lo, fazendo com que todas as regiões da imagem que não são necessárias para uma alta ativação tenham um valor de exatamente zero. Isso pode ser feito através de um operador $r_\theta(\mathbf{x})$ que calcula o módulo de cada pixel (sobre os canais de verde, vermelho e azul) e zera todos os pixels com módulo baixo. O ponto de corte é definido pelo hiperparâmetro $\theta_{\mathrm{n\_pct}}$, que é especificado como uma porcentagem de todos os módulos dos pixels em $\mathbf{x}$.

Corte de pixels com pouca contribuição: Uma alternativa um pouco mais sofisticada ao corte de pixels com módulos baixos é o corte de pixels que tem uma pequena contribuição para a ativação. A forma mais direta de se calcular a contribuição de um pixel para a ativação é medir o quanto a ativação muda quando o valor do pixel é zerado, ou seja, a contribuição pode ser calculada como $|a_i(\mathbf{x}) - a_i(\mathbf{x}_{-j})|$, onde $\mathbf{x}_{-j}$ é $\mathbf{x}$ com o pixel de índice j zerado. Este método, porém, é proibitivamente lento, pois requer que a ativação seja recalculada para cada pixel da imagem. Uma maneira de se aproximar esse processo é linearizar $a_i(\mathbf{x})$ ao redor de $\mathbf{x}$. Neste caso a contribuição de cada dimensão de $\mathbf{x}$ pode ser estimada como a multiplicação elemento a elemento de $\mathbf{x}$ pelo seu gradiente. A contribuição aproximada de cada pixel é calculada como o valor absoluto da soma das contribuições aproximadas dos seus três canais de cor, ou seja, $|\sum_c \mathbf{x} \circ \nabla_{\mathbf{x}} a_i(\mathbf{x})|$. O operador $r_\theta(\mathbf{x})$ é definido como uma operação que zera os pixels com uma contribuição abaixo da porcentagem $\theta_{\mathrm{c\_pct}}$.

Todos estes métodos de regularização resultam em imagens um pouco mais interpretáveis quando aplicados individualmente, como mostrado na Figura 4, porém os resultados são melhores quando diferentes métodos são aplicados em conjunto [75].

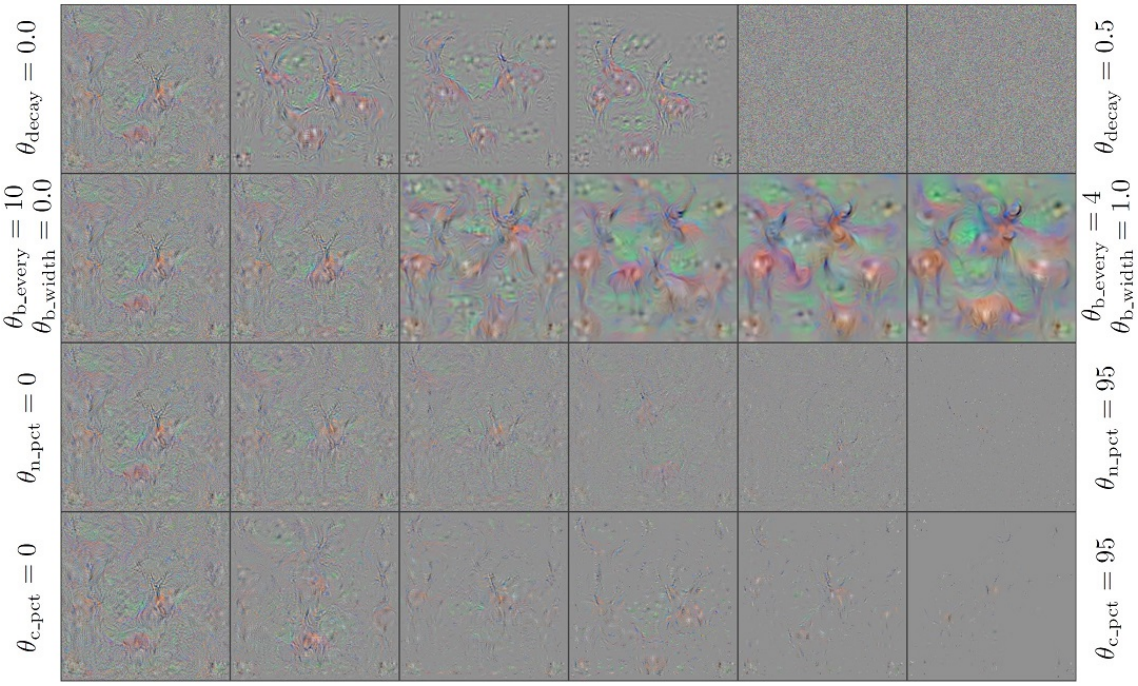


Figura 4: Efeitos de cada um dos métodos de regularização apresentados em [75], quando aplicado individualmente. Fonte: [75]. Esta figura é melhor visualizada em cores.

A qualidade dos resultados depende muito da combinação de hiperparâmetros utilizada. Em [75] foram testadas 300 combinações de hiperparâmetros geradas aleatoriamente. A combinação que gerou os resultados considerados mais interpretáveis pelos autores foi: $\theta_{\text{decay}} = 0.0001$, $\theta_{\text{b\_width}} = 1.0$, $\theta_{\text{b\_every}} = 4$, $\theta_{\text{n\_pct}} = 0$ e $\theta_{\text{c\_pct}} = 0$. Alguns resultados produzidos com esta combinação de hiperparâmetros são apresentados na Figura 5. Com esta combinação de hiperparâmetros, o corte de pixels com módulo baixo e o corte de pixels com pouca contribuição estão desligados. A regularização por L_2 decay está presente, porém de forma muito suave. A única técnica de regularização que é aplicada de forma significativa neste caso é o filtro Gaussiano. Isso leva a crer que, de todas as técnicas de regularização apresentadas em [75], o filtro Gaussiano é a mais eficiente.

2.4 Visualização Multifacetada

Os detectores de *features* de uma rede neural precisam reconhecer que imagens muito diferentes podem representar o mesmo conceito (por exemplo, um detector de pimentões precisa reconhecer pimentões verdes, vermelhos e laranjas como a mesma classe). Pode-se dizer que cada um destes tipos de imagem, que são muito diferentes entre si, mas representam a mesma classe, é uma *faceta* da classe. O método da maximização da ativação não leva isso em conta, e por isso não é capaz de mostrar as múltiplas facetas que causam a ativação do mesmo neurônio. O resultado disso é que a maximização da ativação tenta gerar múltiplas facetas da mesma classe simultaneamente, na mesma imagem, o que gera resultados de difícil interpretação. O algoritmo Visualização Multifacetada

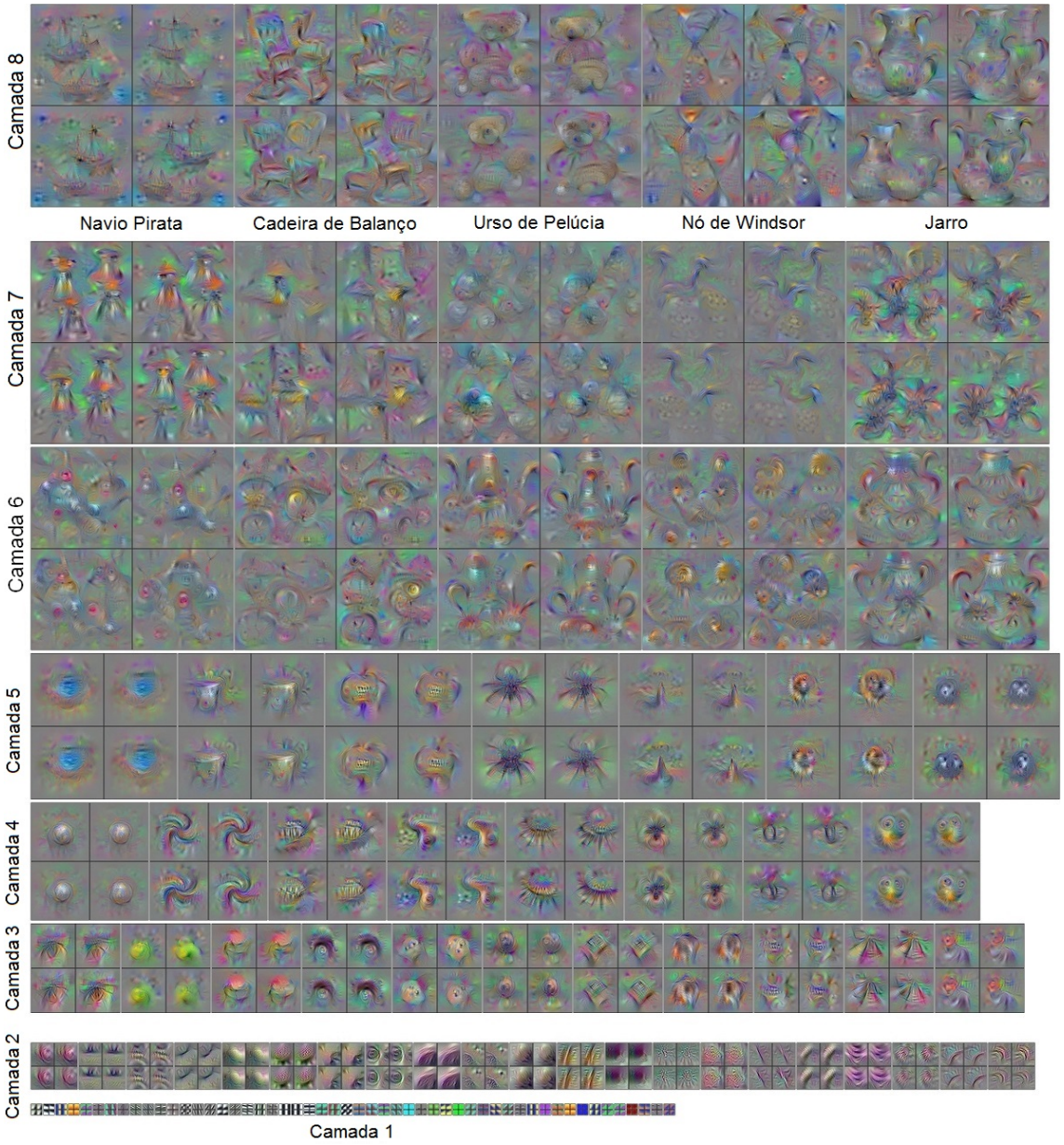


Figura 5: Visualização gerada por maximização da ativação com regularização de alguns *feature maps* das oito camadas de uma rede neural convolucional de classificação. Para cada canal são mostrados os resultados de 4 inicializações aleatórias diferentes. As camadas 1 a 5 são camadas convolucionais, as camadas 6 a 8 são camadas *fully-connected*. Podem ser reconhecidos *features* importantes de objetos em diferentes escalas, como bordas, cantos, rodas, olhos, faces, garrafas, etc. As visualizações apresentam um aumento da complexidade e da variação nas camadas mais profundas. Os parâmetros de regularização utilizados são $\theta_{\text{decay}} = 0.0001$, $\theta_{\text{b\_width}} = 1.0$, $\theta_{\text{b\_every}} = 4$, $\theta_{\text{n\_pct}} = 0$ e $\theta_{\text{c\_pct}} = 0$. Fonte: [75]. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

[48] busca aliviar este problema, permitindo a visualização das diferentes facetas de cada neurônio.

A ideia do método parte da observação que no conjunto de treinamento da base de dados ImageNet cada classe tem múltiplos grupos de imagens que refletem diferentes facetas da classe. Por exemplo, na classe pimentão podem ser encontrados pimentões de diferentes cores, sozinhos ou em grupos, cortados ou inteiros, etc. Os autores de [48] supuseram que a maximização da ativação da classe pimentão é difícil porque a faceta a ser reconstruída não é especificada, o que pode resultar em diferentes áreas da imagem sendo otimizadas para reconstruir diferentes facetas da classe. Os autores também supuseram que se a maximização da ativação for inicializada com a imagem média de uma única faceta, ao invés da imagem média de todo o conjunto de dados ou de uma imagem aleatória, a chance da otimização reconstruir aquela faceta seria aumentada.

Basicamente, a ideia do algoritmo é (1) colocar todas as imagens da classe em um espaço bidimensional usando PCA [51, 33] e t-SNE[41], (2) utilizar k-means clustering para encontrar k tipos de imagens, (3) para cada k criar uma imagem média $\mathbf{x}_0$ com as m imagens mais perto do centroide do cluster e (4) executar a maximização da ativação usando $\mathbf{x}_0$ como imagem inicial. Em [48] os melhores resultados foram obtidos com $k = 10$ e $m = 15$.

A Visualização Multifacetada também pode ser aplicada em neurônios das camadas intermediárias. Para se fazer isto, basta substituir as imagens da classe pelas top n imagens do conjunto de dados que causaram a maior ativação no neurônio que se deseja visualizar.

A Visualização Multifacetada produz imagens diferentes que causam a ativação do mesmo neurônio. Essas imagens mostram o mesmo objeto em diferentes cores e visto de diferentes ângulos. Existem casos onde as facetas são completamente diferentes, como na classe cinema, apresentada na Figura 6, onde uma faceta mostra o interior do cinema, com as fileiras de cadeiras e a tela, enquanto outra mostra a fachada externa. A diferença entra as facetas tende a ser maior nos neurônios das camadas mais profundas, o que indica que estes neurônios representam conceitos mais abstratos em relação aos neurônios das camadas mais rasas. As imagens geradas por Visualização Multifacetada também tendem a ter cores mais naturais e estrutura global mais consistente em relação às imagens geradas com Maximização da Ativação pura [48].

2.5 Algoritmos Evolutivos

Em [47] é proposta a utilização de algoritmos evolutivos [19] para gerar imagens que causam uma alta ativação em um determinado neurônio. Algoritmos evolutivos são uma técnica de otimização baseada na teoria da evolução de Darwin. Eles contêm uma população de organismos (no caso, imagens) que alternadamente passam por etapas de seleção (apenas os melhores são mantidos) e perturbação aleatória (mutação e/ou cruzamento). A

Figura 6: Visualização de diferentes facetas da classe cinema. Fonte: [48].

seleção dos organismos a serem mantidos depende de uma função de avaliação, que no caso é a ativação produzida pela imagem em um neurônio da rede.

Em [47] são testados algoritmos evolutivos com dois tipos diferentes de codificação [64, 7], que é a representação da imagem em forma de genoma.

2.5.1 Codificação Direta

Na codificação direta cada pixel da imagem é representado por um número inteiro, para imagens em escala de cinza, ou por três (um para cada canal RGB ou HSV), para imagens coloridas. Cada valor é inicializado com um valor aleatório entre 0 e 255. Esses números sofrem mutação independente: primeiro se determina quais número sofrerão mutação, através de uma taxa que inicia em 0.1 (cada número tem uma chance de 10% de sofrer mutação) e é reduzida pela metade a cada mil gerações. Os números escolhidos para sofrer mutação são alterados por um operador de mutação polinomial [11] com uma força de mutação fixa de 15.

Quando aplicado à RNC AlexNet [36] o algoritmo evolutivo com codificação direta teve dificuldades de gerar imagens que são classificadas pela rede com alta confiança para a maioria das classes. Apesar disso, a evolução foi capaz de gerar imagens que são classificadas com confiança superior a 99% para 45 classes. Estas imagens contêm alguns *features* que estão associados à classe que elas deveriam representar, mas apesar disso são praticamente irreconhecíveis para humanos que não foram informados sobre a classe em que elas foram classificadas pela rede [47].

2.5.2 Codificação Indireta

A codificação indireta tende a produzir imagens regulares, ou seja, imagens que contém padrões compreensíveis, com simetria e repetição. Em [47], as imagens são codifi-

cadas em forma de *compositional pattern-producing networks* (CPPNs). Esta codificação permite a geração imagens complexas e regulares, que podem lembrar objetos naturais ou artificiais.

Uma CPPN recebe a posição (x, y) de um pixel como entrada e da como saída um único valor (para imagens em escala de cinza) ou uma tupla de três valores (para imagens coloridas). Como em uma rede neural a função computada pela CPPN depende do número de neurônios dela, de como estes neurônios estão conectados e do peso entre eles. Cada nodo da CPPN pode ser uma função de ativação pertencente a um conjunto de funções possíveis, como seno, sigmoide, Gaussiana e linear. São essas que funções podem dar regularidade geométrica à imagem. Por exemplo, a passagem da entrada x por uma função Gaussiana ira gerar simetria entre esquerda e direita, e a passagem da entrada y por uma função seno ira gerar repetição vertical. A evolução determina a topologia, os pesos e as funções de ativação em cada CPPN da população. As CPPNs são inicializadas sem nenhum nodo oculto e os nodos são adicionados com o tempo, o que encoraja a evolução a procurar por imagens simples e regulares antes de adicionar complexidade.

O algoritmo evolutivo com codificação indireta consegue gerar imagens com alta confiança para a maioria das classes da AlexNet. Estas imagens, apesar de não serem reconhecíveis para humanos, frequentemente contêm *features* da classe alvo. Por exemplo, uma imagem gerada para maximizar a classe estrela do mar tem o azul da água e o laranja da estrela do mar, a imagem da classe bola de beisebol tem costuras vermelhas em um fundo branco, a imagem da classe controle remoto tem um grid de botões, etc. Para muitas das imagens produzidas é possível compreender porque a rede a identificou como parte de uma determinada classe, desde que a classe dada pela rede seja conhecida. Isso ocorre porque a evolução só precisa produzir *features* que são únicos, ou *discriminativos*, da classe, e não uma imagem que contém todos os *features* típicos da classe. Diferentes execuções do algoritmo evolutivo produzem diferentes tipos de imagem para muitas classes, o que mostra que para cada classe existem diferentes *features* discriminativos que a evolução pode explorar. Alguns resultados da visualização por algoritmos evolutivos com codificação por CPPN são apresentados na Figura 7.

2.6 Conclusão

Neste capítulo foram apresentadas algumas das técnicas mais frequentemente empregadas na visualização de camadas intermediárias de redes neurais. Apesar de todas as técnicas apresentadas terem sido desenvolvidas para a visualização de redes neurais de classificação, algumas delas podem ser utilizadas para auxiliar na compreensão de arquiteturas de transformação de imagem. A visualização direta pode ser utilizada para se ter uma ideia de que tipos de *feature maps* estão presentes nas camadas intermediárias destas redes, o que pode dar uma ideia de quais *features* estão relacionados com o problema que

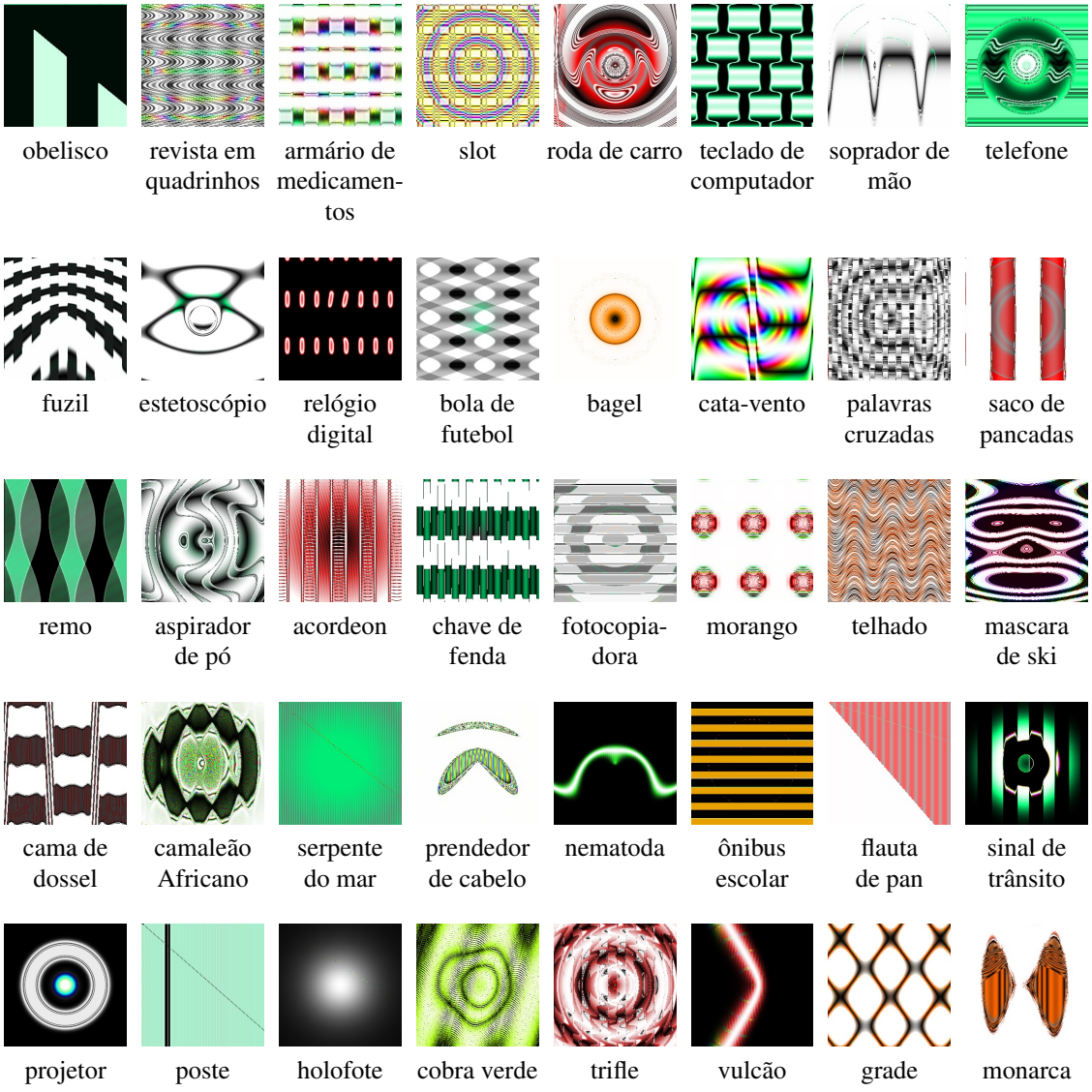


Figura 7: Imagens geradas por um algoritmo evolutivo com codificação indireta por CPPNs para maximizar a ativação de neurônios da camada de saída de uma rede neural de classificação. Fonte: [47].

está sendo tratado. A maximização da ativação também pode ser utilizada para o mesmo propósito, já que ela produz imagens de entrada que provocam uma alta ativação em um *feature map* específico. A visualização por deconvolução mostra quais estruturas da imagem provocam a ativação de um determinado *feature map*. Apesar de bastante útil no problema de classificação, a aplicação desta técnica em redes de transformação de imagem não parece ser muito promissora. A razão disto é que soluções para problemas de transformação de imagem normalmente se baseiam em *features* mais simples, como cor e contraste, e não em estruturas complexas. Outra desvantagem deste método é que ele precisa de uma implementação específica para cada arquitetura de rede neural. A visualização por algoritmos evolutivos com codificação por CPPNs é um método extremamente complexo, que produz padrões regulares, que apresentam algum tipo de simetria. Estes padrões podem ser úteis para a interpretação de redes de classificação, mas dificilmente podem produzir resultados interpretáveis para redes de transformação de imagem, onde regularidade e simetria aparentemente são muito menos importantes.

3 PROBLEMAS DE TRANSFORMAÇÃO DE IMAGEM

Problemas de transformação de imagem são aqueles onde a entrada e a saída são uma imagem. Apesar de estes problemas estarem de certa forma relacionados o problema de classificação, onde a saída é uma única categoria que representa a imagem inteira, eles são consideravelmente diferentes, e por isso não podem ser resolvidos com os mesmos métodos. Neste capítulo são apresentados alguns problemas de transformação de imagem e arquiteturas de redes neurais desenvolvidas para resolvê-los.

3.1 Restauração de Imagens com Ruído

O objetivo da restauração é melhorar uma dada imagem em um sentido predefinido. Tradicionalmente a restauração busca reconstruir ou recuperar uma imagem degradada utilizando conhecimento prévio sobre o fenômeno de degradação. As técnicas de restauração tradicionais são orientadas à modelagem do fenômeno de degradação e da aplicação do processo inverso para se recuperar a imagem original. Essas abordagens normalmente envolvem a formulação de um critério de qualidade que produz uma estimativa ótima do resultado desejado.

Uma operação relacionada é o melhoramento de imagens. Apesar de existir uma certa sobreposição entre as duas, o melhoramento é um processo altamente subjetivo, enquanto a restauração é um processo muito mais objetivo. As técnicas de melhoramento basicamente são processos heurísticos desenvolvidos para manipular uma imagem com o objetivo de aproveitar os aspectos físicos e psicológicos do sistema visual humano. Por exemplo, o aumento de contraste é considerado uma técnica de melhoramento porque é baseado principalmente no aspecto agradável que ele pode proporcionar ao espectador, enquanto o aumento da nitidez através da aplicação de uma função de *deblur* é considerado uma técnica de restauração [21].

3.1.1 Modelo dos Processos de Degradação e Restauração

Em [21] a degradação é modelada como uma função que, junto com termo de ruído adicional, opera sobre uma imagem de entrada $f(x, y)$ para produzir uma imagem degra-

dada $g(x, y)$:

$$g\left(x, y\right) = H\left[f\left(x, y\right)\right] + \eta(x, y)$$

Dados $g\left(x, y\right)$, algum conhecimento sobre a função de degradação H, e algum conhecimento sobre o termo de ruído adicional $\eta(x, y)$, o objetivo da restauração é obter uma estimativa $\hat{f}(x, y)$ da imagem original. Deseja-se que a estimativa seja a mais próxima possível da imagem original. Em geral, quanto mais se sabe sobre H e η, menor a distância entre $\hat{f}(x, y)$ e $f(x, y)$.

Se H é um processo *linear*, *espacialmente invariante*, pode-se mostrar que a imagem degradada é dada no *domínio espacial* por:

$$g\left(x, y\right) = h\left(x, y\right) * f\left(x, y\right) + \eta(x, y)$$

onde $h\left(x, y\right)$ é a representação espacial da função de degradação e o símbolo "$*$" representa a operação de convolução. Como a convolução no domínio espacial é equivalente à multiplicação no domínio da frequência, a representação da equação 3.1.1 no domínio da frequência é:

$$G\left(u, v\right) = H\left(u, v\right) F\left(u, v\right) + N\left(u, v\right)$$

onde os termos em letras maiúsculas são a transformada de Fourier dos termos correspondentes na equação 3.1.1. A função de degradação $H\left(u, v\right)$ é chamada de *função de transferência óptica* (FTO), um termo derivado da análise de Fourier de sistemas ópticos. No domínio espacial, refere-se a $h(x, y)$ como *função de espalhamento de ponto* (FEP), um termo que vem da aplicação de $h(x, y)$ em um ponto de luz para se obter as características da degradação para qualquer tipo de entrada.

Como a degradação por uma função de degradação linear, invariante no espaço H pode ser modelada como uma convolução, às vezes refere-se ao processo de degradação como "a convolução da imagem com uma FEP ou FTO". Da mesma forma, o processo de restauração às vezes é chamado de *deconvolução*.

3.1.2 Aplicação das RNs na Restauração de Imagens

Os métodos de restauração tradicionais se baseiam na modelagem do fenômeno de degradação e no desenvolvimento de um processo inverso. Essas soluções normalmente são algoritmos complexos, desenvolvidos manualmente para tratar um tipo de ruído específico. O desenvolvimento de tais algoritmos exige um conhecimento sobre o fenômeno responsável pela geração do ruído. O principal problema destas abordagens é que elas se baseiam em suposições que são simplificações da realidade, e que não são verdadeiras em todos os casos. Além disso, elas normalmente são limitadas a um fenômeno especifico, ou seja, o algoritmo desenvolvido para tratar um tipo de ruído específico não funciona bem

no tratamento de outros. As Redes Neurais são uma alternativa aos métodos tradicionais que apresenta uma maior robustez em relação a estes problemas. As RNs são capazes de inferir o processo de restauração através dos dados, sem a necessidade de um grande conhecimento prévio sobre o fenômeno responsável pela formação do ruído. Além disso, uma única RN pode ser treinada para tratar diferentes tipos de ruído simultaneamente, o que garante um maior poder de generalização.

3.1.2.1 Remoção de Ruído Gaussiano

Em [30] uma Rede Neural Convolucional é utilizada na tarefa de remoção de ruído Gaussiano de variância desconhecida. Os outros métodos de restauração disponíveis na época, como BLS-GSM [52] e FoE [54], assumiam que a distribuição do ruído tinha variância conhecida, o que pode ser uma desvantagem, já que esta informação nem sempre está disponível no mundo real.

A RNC utilizada tem 5 camadas ocultas e 24 *feature maps* por camada. Também foram treinados outros modelos, com 4 camadas ocultas cada, destinados à restauração de imagens corrompidas com um ruído Gaussiano de distribuição conhecida e específica. As redes foram treinadas com *patches* de tamanho 6×6, com um *mini-batch* de 6 *patches* extraídos de 6 imagens escolhidas aleatoriamente do conjunto do treinamento. O processo de geração de ruído está integrado ao treinamento. A cada iteração as imagens degradadas são produzidas através da aplicação de uma função de geração de ruído sobre imagens do conjunto de treinamento, que são consideradas como livres de ruído. O processo de treinamento utilizado consiste em treinar cada camada de forma incremental. Inicialmente é treinada uma rede com uma única camada oculta, por 30 épocas. Os pesos desta camada então são copiados para uma nova rede, de duas camadas ocultas, que é treinada por mais 30 épocas. Este processo é repetido até que todas as camadas da rede sejam treinadas. Este procedimento, segundos os autores, reduz o tempo de treinamento e aumenta o poder de generalização da rede.

Os modelos treinados para restaurar imagens com uma distribuição de ruído específica apresentaram um desempenho superior aos métodos BLS-GSM e FoE em todos os níveis de ruído testados. O modelo treinado para restaurar imagens com nível de ruído desconhecido apresentou um desempenho comparável aos outros modelos de RN. Em geral, quanto maior o ruído, melhor o desempenho das RNs em relação aos outros métodos. Uma comparação entre os resultados de diferentes métodos de remoção de ruído é apresentada na Figura 8.

3.1.2.2 Inpainting

Inpainting é o processo de reconstruir partes perdidas ou deterioradas de uma imagem. A diferença entre inpainting e remoção de ruído Gaussiano é que o inpainting busca restaurar regiões maiores, que podem formar padrões complexos, onde o valor original

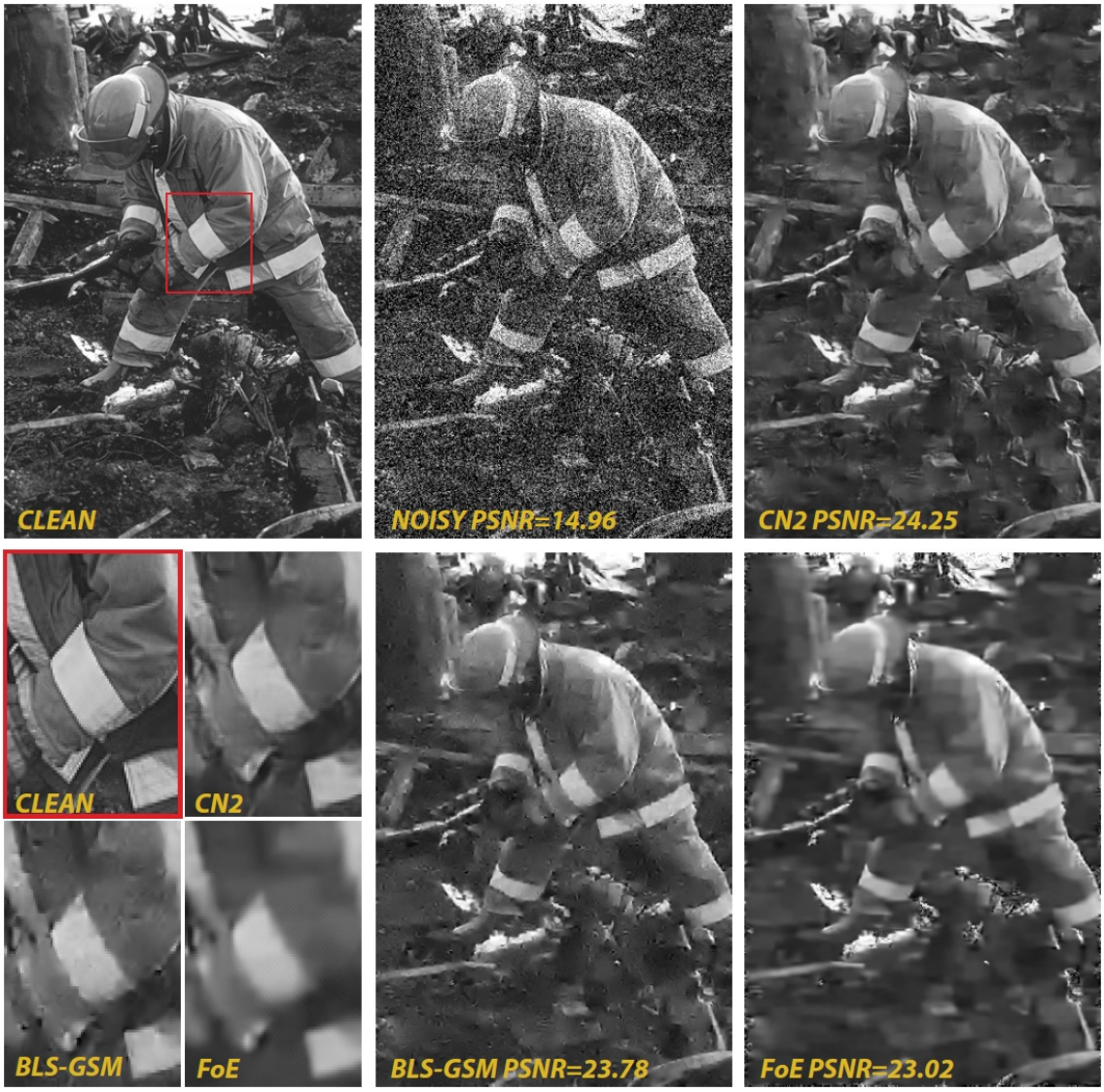


Figura 8: Comparação entre os resultados de restauração por rede neural convolucional e pelos métodos BLS-GSM [52] e FoE [54]. Fonte: [30].

dos pixels não está presente, enquanto a remoção de ruído Gaussiano se concentra na restauração de pixels isolados. Um exemplo de inpainting é a remoção de um texto que está escrito sobre uma imagem.

Os algoritmos tradicionais de inpanting, como o KSVD [42], precisam receber como entrada uma máscara, que informa quais regiões da imagem estão corrompidas e precisam ser reescritas. Em [73] é apresentada uma rede neural que detecta e reconstrói as regiões corrompidas da imagem automaticamente, sem a necessidade de uma máscara, além de restaurar os pixels corrompidos por ruído gaussiano. A arquitetura utilizada, chamada Stacked Sparse Denoising Autoencoders (SSDA), é baseada no modelo Denoising Autoencoder [71].

O modelo foi treinado utilizando dados simulados, gerados a partir de imagens naturais coletadas da internet. As imagens corrompidas do conjunto de treinamento foram produzidas através da adição de texto de sobre as imagens originais. Foram utilizadas diferentes fontes, entre os tamanhos 18-pix e 36-pix.

Os autores consideram que, quando aplicado na remoção de ruído Gaussiano, o modelo SSDA gera bordas mais claras e preserva melhor os detalhes das texturas em relação aos algoritmos BLS-GSM [52] e KSVD. Apesar disso a diferença na relação sinal-ruído de pico, um critério frequentemente utilizado para avaliar a qualidade da restauração, entre os três algoritmos é estatisticamente insignificante. Na tarefa de inpainting o SSDA mostrou um desempenho comparável ao algoritmo KSVD, mesmo com a desvantagem de ser um modelo de inpainting "cego", onde as regiões que precisam ser restauradas não são conhecidas previamente. Um comparativo entre os resultados do modelo SSDA e do algoritmo KSVD é apresentado na Figura 9.

3.1.2.3 Remoção de Ruídos Diversos

Em [3] é estudada a aplicação de RNs na restauração de imagens corrompidas por diversos tipos de ruídos, como ruído Gaussiano, ruído "sal e pimenta", ruído em "listras" e artefatos de compressão JPEG. A rede neural utilizada é do tipo Multilayer Perceptron (MLP), que é a arquitetura clássica de Redes Neurais, onde todas as camadas são *fully-connected*. A justificava para o uso de MLPs no lugar de RNCs, que são mais frequentemente utilizadas em tarefas de visão computacional, é que o MLP pode ser considerado um aproximador universal de funções [27], enquanto as RNCs têm restrições quanto às classes de funções que podem ser aprendidas [3].

A arquitetura que obteve os melhores resultados usa *patches* de tamanho 17×17 como entrada e tem 4 camadas ocultas, com 2047 neurônios cada. Várias redes com esta arquitetura foram treinadas, cada uma com o seu próprio conjunto de dados de treinamento, para restaurar imagens corrompidas pelos diferentes tipos de ruído. Os resultados foram comparados com os dos algoritmos tradicionalmente utilizados para restaurar imagens corrompidas pelo tipo de ruído correspondente. Na remoção de ruído Gaussiano os re-

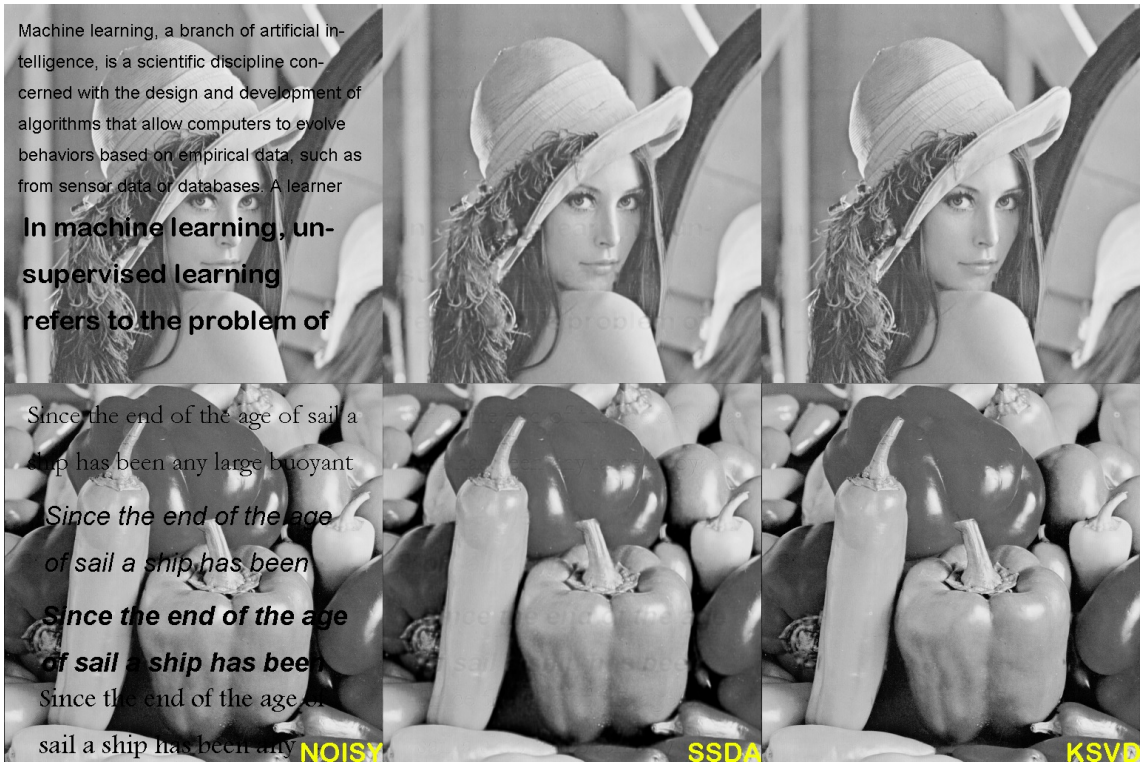


Figura 9: Comparação entre os resultados de inpaiting do modelo SSDA apresentado em [73] e do algoritmo KSVD. Fonte: [73].

sultados foram comparáveis aos do algoritmo BM3D [10]. Na restauração de ruído em "listras" o MLP testado mostrou um desempenho superior ao BM3D. Na restauração de ruído do tipo "sal e pimenta" os resultados do MLP foram superiores aos obtidos com filtro mediana de tamanho 5×5. Na restauração de artefatos provocados por compressão JPEG o MLP se mostrou superior à técnica de reaplicação de JPEG [49].

3.1.2.4 Remoção de Sujeira e Pingos de Chuva

Em [15] é apresentada uma Rede Neural capaz de restaurar fotografias tiradas através de uma janela coberta com manchas de sujeira ou pingos de chuva. A restauração deste tipo de imagem é um problema bastante diferente da remoção de outros tipos de ruído, como o ruído Gaussiano, já que os artefatos resultantes não são limitados a pixels isolados e tem uma estrutura característica. Abordagens tradicionais, como o filtro mediana e o filtro bilateral, não são capazes de remover este tipo de artefato.

A arquitetura utilizada é uma Rede Neural Convolucional com 2 camadas ocultas. A primeira camada aplica 512 *kernels* de tamanho $16 \times 16 \times 3$, a segunda aplica 512 *kernels* de tamanho $1 \times 1 \times 512$ e a camada final 3 *kernels* de tamanho $8 \times 8 \times 512$. Durante o treinamento a rede recebe *patches* de tamanho 64×64 como entrada e produz *patches* de tamanho 56×56 na saída.

Para se ensinar a rede a remover artefatos provocados por sujeira foram utilizadas imagens com manchas de sujeira simuladas. Estas imagens foram criadas com a sobre-

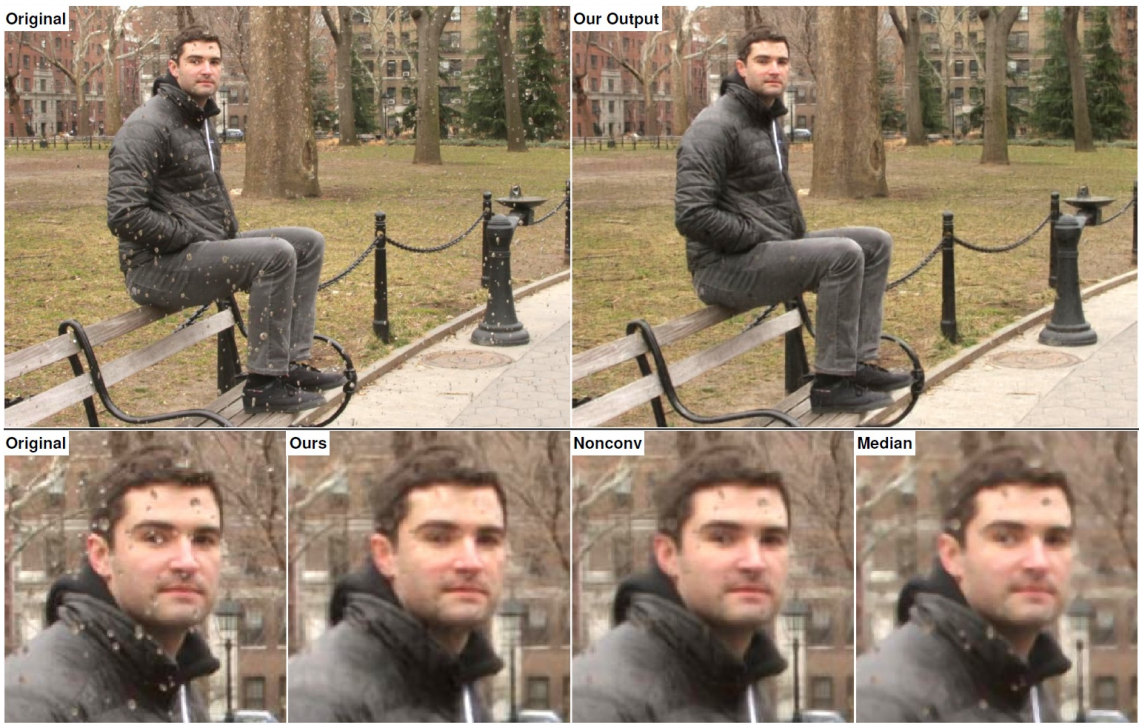


Figura 10: Comparativo entre os resultados da restauração pelo método apresentado em [15], por uma rede não convolucional e pelo filtro mediana. Fonte: [15].

posição de um padrão de sujeira de cor aleatória sobre uma imagem limpa. Os padrões de sujeira utilizados foram obtidos através de fotografias de painéis de vidro sujos tiradas em laboratório.

A simulação realista de ruído provocado por gotas de chuva não foi possível porque as gotas de água provocam a refração da luz ao seu redor, o que é um efeito difícil de se simular. Para se ensinar a rede a tratar este tipo de ruído foi necessária a criação de um conjunto de treinamento composto por fotografias reais da mesma cena com e sem degradação. O efeito da chuva foi simulado com a colocação de um vidro anti-reflexivo molhado na frente da câmera. Para limitar as diferenças provocadas por movimento entre as imagens com e sem ruído foram fotografados apenas objetos estáticos.

O modelo apresentado foi comparado com outros métodos de restauração: uma rede não convolucional similar à apresentada em [3], filtro mediana, filtro bilateral [70] e BM3D [10]. Os resultados apresentados tanto na remoção de sujeira quanto na remoção de pingos de chuva foram considerados superiores aos de todos os métodos concorrentes. A rede foi capaz de remover a maioria dos artefatos sem causar perda de detalhes na imagem. Apesar disso a qualidade da restauração caiu consideravelmente quando foram testadas imagens com artefatos com características diferentes daqueles presentes no conjunto de treinamento. Alguns resultados do método proposto são apresentados na Figura 10.

3.1.2.5 Deconvolução

Em [74] é apresentada uma arquitetura de Rede Neural Convolucional destinada a realizar a deconvolução de imagens. A deconvolução é o processo que busca reverter diversos tipos de degradação, como *blur* (perda de nitidez da imagem), saturação, ruído da câmera e artefatos de compressão.

O processo de degradação é modelado como

$$\hat{y} = \psi_b \left[\phi \left(\alpha x * k + n \right) \right],$$

onde αx representa a imagem nítida latente. A notação $\alpha \geq 1$ indica que αx pode ter valores que excedem o alcance dinâmico dos sensores da câmera, e por isso são ceifados. k é o *kernel* de convolução conhecido, normalmente chamado de função de espalhamento de ponto (FEP), n é um modelo do ruído aditivo da câmera. $\phi(\cdot)$ é a função de ceifamento que modela a saturação, definida como $\phi(z) = min(z, z_{max})$, onde z_{max} é o alcance máximo. $\psi_b[\cdot]$ é um operador de compressão não linear (JPEG, por exemplo).

Mesmo com $\hat{y}$ e k conhecidos, a restauração de αx é intratável, por causa da perda de informação provocada pelo ceifamento [74]. Logo, o objetivo do modelo proposto é restaurar $\hat{x}$, onde $\hat{x} = \phi(\alpha x)$.

Segundo os autores a deconvolução pode ser aproximada por uma convolução com um *kernel* inverso $k^{\dagger}$ suficientemente grande. A utilização de *kernels* grandes, porém, aumenta consideravelmente o número de parâmetros da rede, o que dificulta o treinamento. Para resolver este problema os autores usam a decomposição de *kernels*, que consiste na decomposição de um *kernel* 2D em um soma ponderada de *kernels* 1D.

A arquitetura proposta é composta pela concatenação de dois módulos, uma rede de deconvolução e uma rede de remoção de ruído baseada na arquitetura proposta em [15]. A rede de deconvolução proposta possui duas camadas ocultas. A entrada é um *patch* de tamanho 184×184. A primeira camada aplica 38 *kernels* de tamanho 121×1, o que resulta em um *feature map* h_1 de tamanho $64 \times 184 \times 38$. A segunda camada h_2, de tamanho $64 \times 64 \times 38$, é gerada através da aplicação um 38 *kernels* de tamanho 1×121, um em *feature map* h_1. A saída, de tamanho 64×64, é gerada com a aplicação de um *kernel* $1 \times 1 \times 38$ em h_2. A rede de remoção de ruído tem duas camadas ocultas com 512 *feature maps* cada. A imagem de entrada é convoluída com 512 *kernels* de tamanho 16×16 para gerar a ativação da primeira camada oculta. A segunda camada oculta aplica 512 *kernels* de tamanho $1 \times 1 \times 512$, e a camada final aplica um *kernel* de tamanho $8 \times 8 \times 512$. Os dois módulos da rede são concatenados com a combinação da última camada da rede de convolução com a entrada da rede de remoção de ruído. Isso é feito com a união do *kernel* $1 \times 1 \times 38$ com os 512 *kernels* 16×16, o que resulta em 512 *kernels* de tamanho $16 \times 16 \times 38$. Não é utilizada nenhuma função de não linearidade na união entre os dois módulos. A arquitetura completa é apresentada na Figura 11.

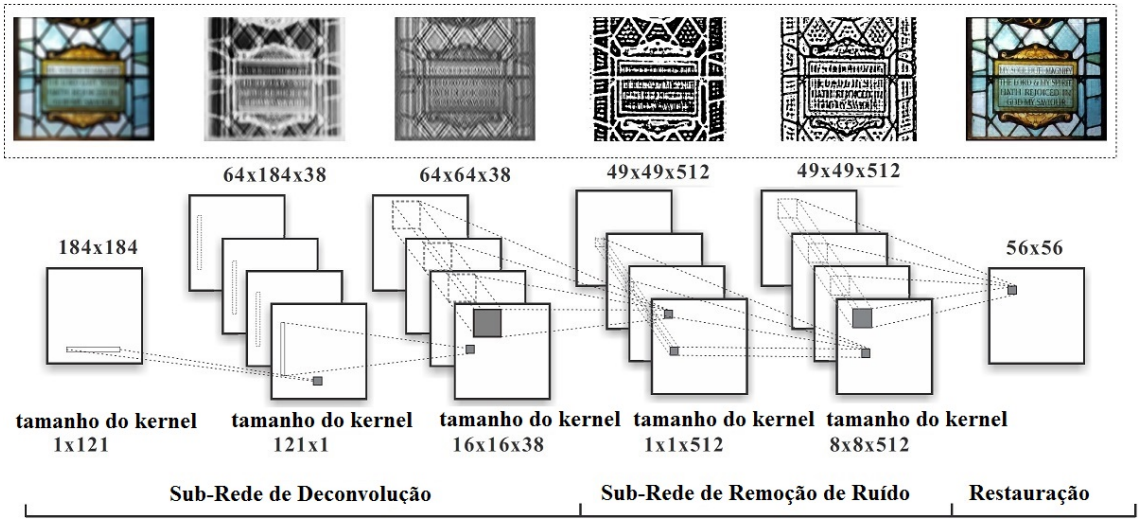


Figura 11: Arquitetura da rede neural de deconvolução apresentada em [74].

O treinamento dos dois módulos da rede é realizado separadamente. A rede de convolução é inicializada através da decomposição do *kernel* inverso $k^{\dagger}$, o que tende a produzir resultados melhores que a inicialização aleatória. Depois que o treinamento individual dos dois módulos está concluído eles são unidos e treinados em conjunto, o que resulta em um ganho de precisão adicional.

A arquitetura proposta é comparada com abordagens de estado da arte em deconvolução, obtendo uma relação sinal-ruído de pico média superior a todos os outros métodos no conjunto de testes apresentado. Os autores consideram que os resultados obtidos são de alta qualidade.

3.2 Colorização de Imagens

A colorização de imagens é o processo de restauração das cores em uma imagem monocromática. Os processos de captura de imagens monocromáticas, assim como a conversão de imagens coloridas, levam em conta apenas a luminância em cada ponto da imagem. A informação de cor é completamente perdida. O resultado disto é que muitas cores completamente diferentes podem resultar em um mesmo tom de cinza. Isso torna a colorização um problema indeterminado, que admite muitas soluções diferentes para cada ponto da imagem. A restauração adequada das cores é altamente dependente do contexto. Por exemplo, em uma paisagem espera-se que o céu seja azul e a grama seja verde, mesmo que ambos sejam representados com um mesmo tom de cinza. Algumas possibilidades de restauração, como uma grama azul e um céu vermelho, não podem ser consideradas aceitáveis, apesar de serem tecnicamente válidas, porque não correspondem às cores observadas no mundo real. Já na colorização objetos artificiais, como carros e peças de roupa, praticamente todos os resultados podem ser considerados aceitáveis, já que no mundo real esses objetos possuem uma grande diversidade de cores.

As abordagens de colorização tradicionais necessitam de intervenção humana para lidar com a ambiguidade do problema, sendo necessário que o usuário diga qual a cor correspondente a cada segmento da imagem. Existem ainda métodos que realizam a colorização utilizando como base uma imagem de referência fornecida pelo usuário [72]. Em [78] é apresentada uma rede neural convolucional que realiza a colorização de imagens sem nenhuma interferência do usuário. O objetivo da rede apresentada não é recuperar as cores reais da imagem, e sim gerar resultados plausíveis, com cores vibrantes, capazes de enganar observadores humanos.

A arquitetura utilizada é uma Rede Neural Convolucional com 8 blocos de camadas convolucionais. Os 5 primeiros blocos de camadas convolucionais são inicializados com os pesos da rede de classificação VGGNet [63], com algumas modificações na arquitetura.

Para o treinamento de uma rede neural de colorização é muito importante a utilização de uma função de erro adequada. O uso de funções de erro tradicionais, como a distância L_2, tende a produzir resultados com cores acinzentadas. Isto ocorre porque a forma mais fácil de se minimizar a distância L_2 entre a saída e o resultado desejado é sempre prever a média entre as cores de saída possíveis. Restaurações com cores mais vivas tendem a apresentar, em média, um erro muito maior, mesmo quando estas restaurações são perfeitamente plausíveis. Por está razão, em [78] a colorização é tratada como um problema de classificação densa. Cada pixel é classificado individualmente, onde as classes possíveis representam diferentes cores. As imagens são tratadas no espaço de cores CIE Lab. O espaço ab é divido em regiões de tamanho 10×10, e cada uma dessas regiões representa uma classe diferente. As classes que caem fora da gama de cores (ou seja, cores que não podem ser representadas) são descartadas, resultando em 313 classes válidas ($Q = 313$). Para uma dada entrada $\mathbf{X}$, a rede aprende uma função de mapeamento $\hat{\mathbf{Z}} = \mathcal{G}(\mathbf{X})$ para uma distribuição de probabilidade sobre as possíveis cores $\hat{\mathbf{Z}} \in [0,1]^{H \times W \times Q}$, onde H e W são as dimensões da entrada e Q é o número de intervalos no espaço ab.

Para se comparar a predição $\hat{\mathbf{Z}}$ com o resultado desejado, é definida uma função $\mathbf{Z} = \mathcal{H}_{gt}^{-1}(\mathbf{Y})$, que converte a cor verdadeira no vetor $\mathbf{Z}$. A função de erro L é definida como:

$$L(\hat{\mathbf{Z}}, \mathbf{Z}) = -\sum_{h,w} v(\mathbf{Z}_{h,w}) \sum_{q} \mathbf{Z}_{h,w,q} \log(\hat{\mathbf{Z}}_{h,w,q})$$

onde $v(\cdot)$ é um termo utilizado para reequilibrar a função de erro com base na raridade das classes de cor. Pare se obter uma imagem colorida, a distribuição de probabilidade $\hat{\mathbf{Z}}$ é mapeada para valores de cor $\hat{\mathbf{Y}}$ com a função $\hat{\mathbf{Y}} = \mathcal{H}(\hat{\mathbf{Z}})$.

Os resultados gerados pela rede foram capazes de confundir observadores humanos em 20% dos casos, onde as imagens coloridas pela rede foram consideradas mais realistas que as imagens originais. Este índice é superior ao de todos os outros métodos de colorização testados. A rede também foi capaz de apresentar resultados convincentes na colorização de fotografias antigas, que possuem características de baixo nível considera-

velmente diferentes das de fotografias modernas, que foram utilizadas no treinamento da rede. Alguns destes resultados são apresentados na Figura 12.

3.3 Estimativa de Mapa de Profundidade

A estimativa do mapa profundidade consiste em determinar a distância dos diferentes pontos da imagem em relação à posição do observador. O mapa de profundidade tem aplicações em muitas áreas, como modelagem 3D e robótica. As soluções tradicionais para este problema utilizam sensores especializados, como a tecnologia LIDAR. Uma alternativa a esses sensores são as câmeras estereoscópicas, que capturam duas imagens da mesma cena de posições diferentes, o que permite que a profundidade seja calculada de forma determinística. A estimativa de profundidade através de uma única imagem é consideravelmente mais complicada. Existe uma grande ambiguidade no problema, provocada principalmente pela dificuldade de se determinar a escala dos objetos.

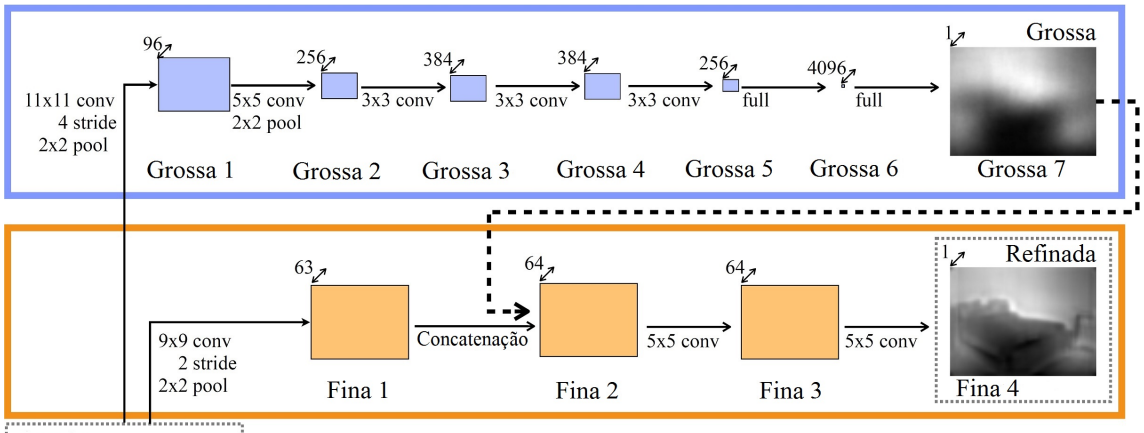
3.3.1 Estimativa de Profundidade com Rede Neural Multi-Escala

Em [16] é apresentada uma arquitetura de rede neural que busca estimar o mapa de profundidade de uma cena utilizando apenas uma única imagem RGB. A arquitetura proposta possui dois componentes, uma rede de escala grossa, que estima a profundidade da cena a um nível global, e uma rede de escala fina, que refina a estimativa nas regiões locais. As duas redes são aplicadas à imagem de entrada, e a saída de rede de escala grossa é utilizada como um *feature map* adicional da primeira camada da rede de escala fina. A arquitetura completa é apresentada na Figura 13.

A rede de escala grossa tem a função de fazer uma estimativa grosseira da profundidade global da cena. As camadas mais profundas desta rede são do tipo *fully-connected*, o que significa que o campo de visão de cada neurônio inclui a imagem inteira, enquanto as camadas mais rasas são camadas convolucionais projetadas para combinar informações de diferentes partes da imagem em uma pequena região através de *max pooling*. Desta forma a rede é capaz de utilizar informações de toda a cena para estimar a profundidade em um único ponto. Estas informações globais, como pontos de fuga, alinhamento da cena e posição dos objetos, são necessárias quando se utiliza uma única imagem para se estimar a profundidade. A rede de escala grossa contém 5 camadas de convolução e *max pooling* que são utilizadas na extração de *features*, seguidas por 2 camadas *fully-connected*. Todas as camadas ocultas utilizam a função de ativação ReLu. A camada de saída utiliza uma função de ativação linear. As camadas convolucionais passam por uma etapa de pré-treinamento, onde elas são treinadas para classificar as imagens do conjunto de dados ImageNet [12], o que gera uma pequena vantagem em relação à inicialização aleatória. A saída da rede tem $1/4$ da resolução da imagem de entrada.

A rede de escala fina é responsável por refinar a estimativa da rede de escala grossa.

Figura 12: Resultados da colorização de imagens antigas por rede neural convolucional. Fonte: [78].



		Grossa					Fina
Camada	entrada	1	2,3,4	5	6	7	1,2,3,4
Tamanho (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Tamanho (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Relação à entrada	/1	/8	/16	/32	–	/4	/4

Figura 13: Arquitetura da rede neural convolucional de estimativa de mapa de profundidade apresentada em [16].

Esta rede é composta apenas por camadas convolucionais, além de uma camada de *pooling* aplicada aos *feature maps* da primeira camada convolucional. O campo de visão de cada neurônio de saída é uma área de 45×45 pixels da imagem de entrada. A saída da rede de escala grossa também é utilizada como um *feature map* de baixo nível, que serve como entrada adicional para a segunda camada convolucional. Todas as camadas ocultas utilizam a função de ativação ReLu. A última camada convolucional utiliza uma função de ativação linear, já que a sua saída é a profundidade da imagem em um ponto, que pode ser negativa. Os pesos da rede de escala grossa são mantidos fixos durante o treinamento da rede de escala fina.

A escala global é um fator de ambiguidade na estimativa de profundidade. Grande parte do erro nas métricas de avaliação pixel a pixel pode ser atribuída ao erro na estimativa da profundidade média da cena. Isso motivou o uso de uma função de erro invariante a escala, que mede a relação entre os pontos na cena sem levar em conta a escala global absoluta. A função de erro utilizada durante o treinamento é dada por

$$L\left(y, y^*\right) = \frac{1}{n}\sum_i d_i^2 - \frac{\lambda}{n^2}\left(\sum_i d_i\right)^2$$

onde y é a estimativa do mapa de profundidade, y^* é o mapa de profundidade real, n é o número de pixels da imagem, i é o índice do pixel, $d_i = \log y_i - \log y_i^*$ e $\lambda \in [0, 1]$. A saída da rede é $\log y$, ou seja, a última camada estima a profundidade logarítmica. O uso de $\lambda = 0$ equivale à distância l_2 pixel a pixel, enquanto $\lambda = 1$ torna o erro invariante a escala. Os autores usam $\lambda = 0.5$, o que produz boa estimativa da escala absoluta e

Figura 14: Resultados da estimativa de profundidade por rede neural convolucional multi-escala. Da esquerda para a direita: A imagem de entrada, a saída da rede de escala grossa, a saída da rede de escala fina e o mapa de profundidade real. Fonte: [16].

melhora um pouco a qualidade da saída.

A rede foi treinada e testada utilizando os conjuntos de dados NYU Depth [61], composto de cenas interiores gravadas com uma câmera Microsoft Kinect, e KITTI [20], composto por cenas externas capturadas com câmeras e um escâner LIDAR montados em um carro. Em ambos os conjuntos de dados o modelo apresentado se mostrou superior a todos os métodos concorrentes, em todas as métricas de avaliação utilizadas. Alguns resultados do modelo são apresentados na Figura 14.

3.3.2 Estimativa de Profundidade com Rede Neural Residual

Em [37] é proposta a utilização de uma RNC com conexões residuais para estimativa de profundidade com base em uma única imagem. A arquitetura utilizada é baseada na rede de classificação ResNet-50 [25], com as camadas *fully-connected* substituídas por operações de aumento de resolução que permitem uma saída com aproximadamente metade da resolução de entrada.

Basicamente, a arquitetura da rede de estimava de profundidade é composta por duas partes. A primeira corresponde às camadas convolucionais da arquitetura ResNet-50, sem as camadas *fully-connected* e a camada de saída que da o resultado da classificação. Esta primeira parte da rede inclui sucessivas operações de redução de dimensão, o que resulta em um grande número de *feature maps* de baixa resolução (10×8 para uma imagem de entrada de 304×228). A segunda parte é composta por um sequencia de convoluções e operações de *unpooling* (o reverso da operação de *max pooling*), que aumentam a resolução dos *feature maps* enquanto reduzem o número de canais de cada camada. Ao final

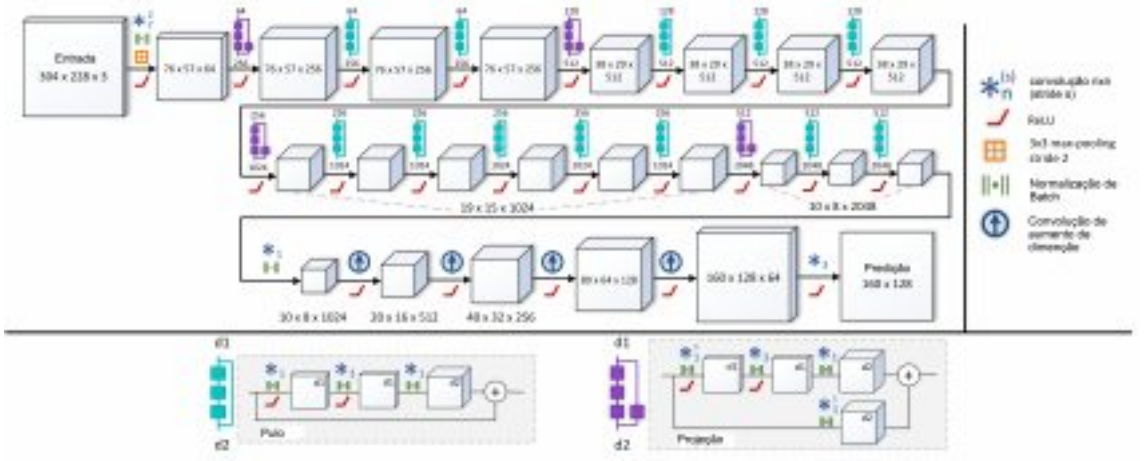


Figura 15: Arquitetura da rede neural residual de estimativa de profundidade proposta em [25].

destas operações, o resultado é um mapa de profundidade que possui aproximadamente a metade do tamanho da imagem de entrada (160×128 para uma entrada de 304×228). A arquitetura completa é apresentada na Figura 15.

No momento do treinamento, a primeira parte da rede é inicializada com os pesos do modelo ResNet-50, pré-treinado na tarefa de classificação com o conjunto de dados do ILSVRC [56]. Os pesos da segunda parte da rede são inicializados aleatoriamente, de uma distribuição normal com média zero e variância 0.01.

A função de treinamento utilizada é a função Huber reversa, ou berHu [80], dada por

$$\mathcal{B}(x) = \begin{cases} |x| & |x| \leq c, \\ \frac{x^2+c^2}{2c} & |x| > c. \end{cases}$$

A função de erro Berhu é igual à distância $L_1(x) = |x|$ quando $x \in [-c, c]$ e igual à distância L_2 fora deste intervalo. A cada passo de treinamento, durante o calculo de $\mathcal{B}(\tilde{y} - y)$ utiliza-se $c = \frac{1}{5}\max_i(|\tilde{y}_i - y_i|)$, onde i é um índice percorre todos os pixels do *batch* atual. O uso da função de erro Berhu permitiu uma melhora nos resultados em relação à função L_2.

A arquitetura proposta foi testada nos conjunto de dados NYU-Depth [61] e Make3D [57]. Em ambos os casos, a rede foi treinada utilizando apenas os dados do conjunto de treinamento correspondente. Para aumentar o volume de dados de treinamento, foram utilizadas técnicas como rotação. mudança de escala, alteração de cor e espelhamento das entradas. A avaliação dos resultados mostrou que a arquitetura proposta superou o estado da arte em ambos os conjuntos de dados, segundo várias métricas diferentes. Alguns resultados para imagens do conjunto de dados Make3D são mostrados na Figura 16.

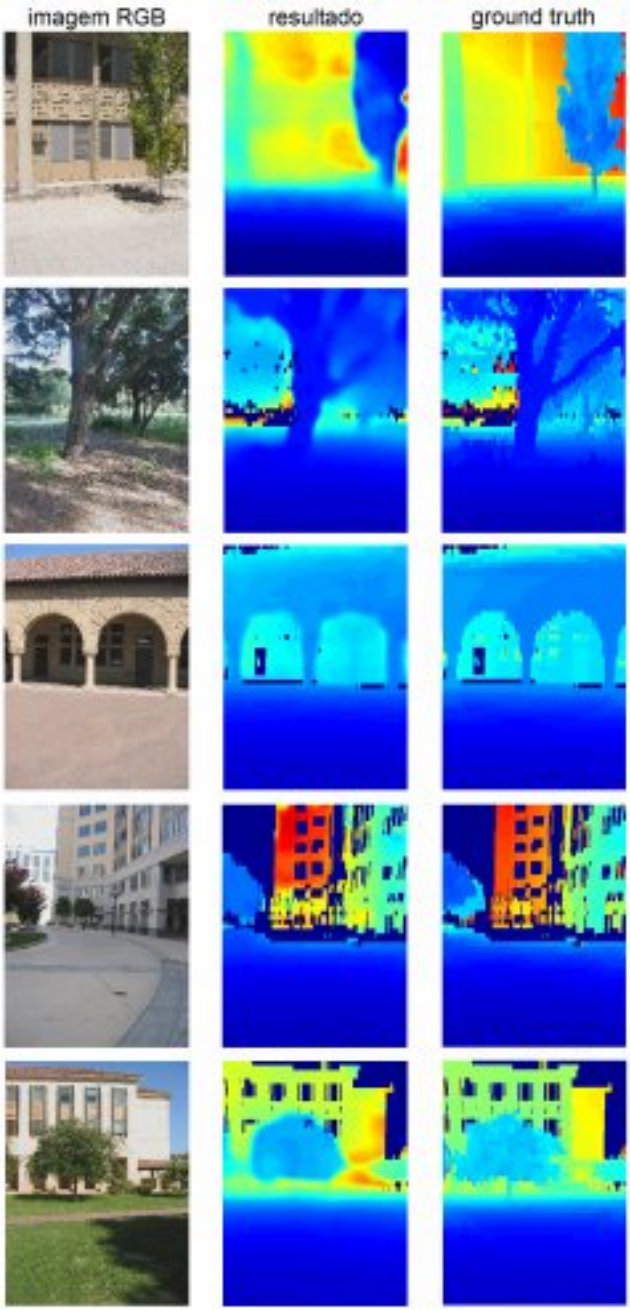


Figura 16: Resultados de [25] para estimativa de profundidade de imagens do conjunto de dados Make3D [57].

3.4 Remoção de Névoa

A aquisição de imagens em áreas abertas é prejudicada quando existe a presença de partículas no ar. Estas partículas absorvem, refletem e desviam os raios de luz, reduzindo a quantidade de informação que chega até a câmera e consequentemente prejudicando a qualidade da imagem. Os efeitos resultantes, que aumentam de intensidade conforme a distância, incluem o embaçamento da imagem, a perda de contraste e a perda de intensidade das cores, que ficam misturadas com a cor da luz ambiente. Em condições atmosféricas normais estes efeitos só são perceptíveis a longas distâncias, porém quando

a quantidade de partículas no ar é muito grande eles podem ser percebidos a distâncias muito menores. Exemplos de situações onde isto ocorre são fenômenos atmosféricos, como chuva e neblina, tempestades de areia e na presença de muita poluição no ar. Estes meios com grande quantidade de partículas que interferem na propagação da luz são chamados meios participativos.

A remoção de névoa, frequentemente chamada de *dehazing* (da palavra *haze*, nevoeiro em ingles), é a operação de restauração de imagens capturadas em meios participativos, que busca remover os efeitos resultantes e recuperar a imagem original. A restauração destas imagens possui muitas aplicações, como segurança, navegação de veículos e fotografia. Os efeitos provocados pelos meios participativos também prejudicam a atuação de muitos algoritmos de visão computacional. *Dehazing* é um problema complicado, pois a intensidade dos efeitos provocados pelos meios participativos é proporcional à profundidade. Esta informação de profundidade, porém, raramente está presente, o que motivou o desenvolvimento de abordagens alternativas. Uma destas abordagens são os métodos baseados em polarização [59, 60], que realizam a restauração utilizando duas ou mais imagens capturadas com diferentes graus de polarização. Uma outra abordagem é a utilização de um mapa de profundidade obtido através da captura de múltiplas imagens da mesma cena em diferentes condições climáticas [45, 46]. O problema destas abordagens é que elas necessitam de dados que nem sempre estão disponíveis. Existem casos em que o único dado disponível é uma única imagem RGB, o que introduz uma grande ambiguidade no problema. Por exemplo, não é possível saber se um determinado ponto da imagem é branco por influência da luz atmosférica ou se o objeto presente naquele ponto é realmente branco. Nestes casos é necessário o uso de heurísticas, que fazem muitas suposições sobre as características da cena e do meio onde ela foi capturada.

3.4.1 Modelo de Formação da Imagem

Um modelo da formação de imagens em meios participativos, frequentemente utilizado em aplicações de visão computacional, é apresentado em [43] e [29]. Segundo este modelo, a imagem é formada pela superposição linear de três componentes: *backscatter*, *forward scattering* e componente direto. A formação da imagem é dada por:

$$\mathbf{I} = E_d + E_{fs} + E_{bs},$$

onde E_d é o componente direto, E_{fs} é o componente de *forward scattering*, E_{bs} é o componente de *backscatter* e $\mathbf{I}$ é a imagem final.

O componente direto se refere à parte do sinal original que chega até a câmera. Este sinal sofre um processo de *atenuação*, que é provocado pela absorção e pelo espalhamento da luz pelas partículas do meio. O componente *forward scattering* é o efeito de embaçamento da imagem, provocado pelo espalhamento dos raios de luz em pequenos ângulos.

O componente de *backscatter* se refere ao efeito provocado por raios de luz provenientes de fontes externas, como a luz do sol, que são refletidos pelas partículas do meio em direção ao plano da imagem. Este efeito, que tem a aparência característica de um "véu" sobre a imagem, provoca uma redução no contraste e contribui para a atenuação do sinal original. O componente de *forward scattering* tem um efeito relativamente pequeno na imagem final, e por isso pode ser ignorado [58], o que permite uma simplificação do modelo:

$$\mathbf{I} = E_d + E_{bs}.$$

3.4.1.1 Componente Direto

O componente direto $E_d(x)$, também chamado de *atenuação direta* [68], é definido como:

$$E_d\left(x\right) = \mathbf{J}\left(x\right) t\left(x\right),$$

onde x é uma posição no espaço 2D, $\mathbf{J}\left(x\right)$ é o sinal original livre de degradação, e $t\left(x\right)$ é a transmissão, que é uma medida da quantidade de luz que não é absorvida ou desviada e chega até a câmera. A transmissão é dada por:

$$t\left(x\right) = e^{-\beta d(x)}, \tag{1}$$

onde β é o coeficiente e atenuação do meio e d é a distância entre a origem do sinal e o observador. Nesta equação supõe-se que β é constante para diferentes comprimentos de onda. Esta suposição é comum em métodos que lidam com meios onde as partículas são grandes em relação às ondas de luz, como neblina, fumaça, etc. Além disso, β é considerado constante em todas as posições da imagem [68].

O sinal original da imagem, também chamado de radiância da cena, corresponde à parte da luz atmosférica que é refletida em direção ao observador, ou seja:

$$\mathbf{J}\left(x\right) = \mathbf{L}_{\infty}\rho\left(x\right),$$

onde $\mathbf{L}_{\infty}$ é a luz atmosférica global e ρ e a refletância de um objeto na imagem. Geralmente supõe-se que $\mathbf{L}_{\infty}$ é constante, ou seja, independente da posição x [68].

3.4.1.2 Componente de Backscattering

O componente de *backscattering*, também chamado de *airlight* [68], ou *luz ambiente*, é dado por

$$E_{bs}\left(x\right) = \mathbf{L}_{\infty}\left(1 - t\left(x\right)\right).$$

Neste caso $\mathbf{L}_{\infty}$ é considerada natural e uniforme. Desta forma é possível considerar que todas as fontes de luz provenientes de fora do plano observado tem origem no mesmo ponto [8].

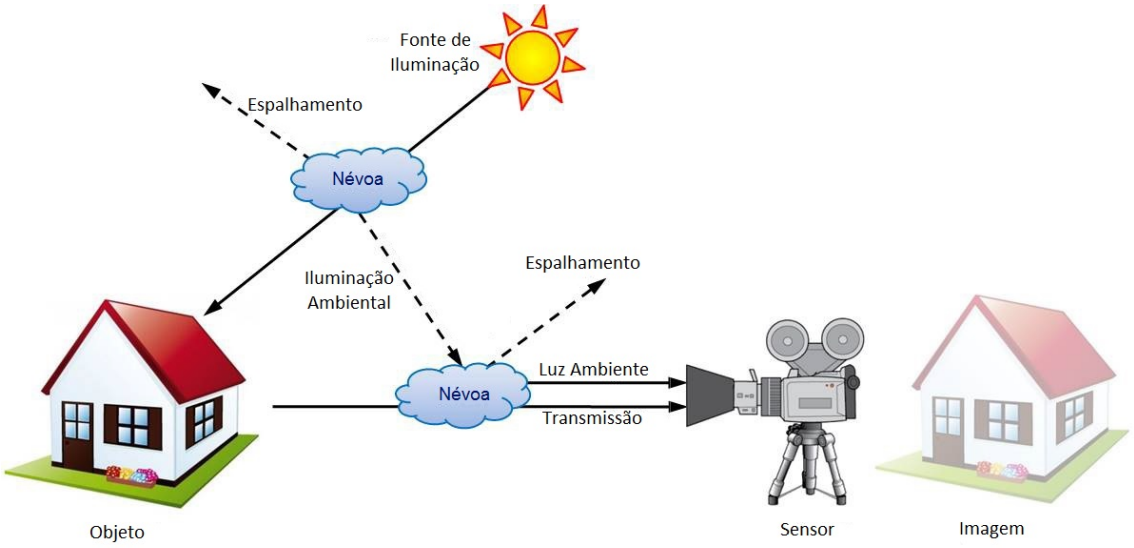


Figura 17: Modelo de formação de imagem em condições de neblina. A atenuação da transmissão $\mathbf{J}(x)\,t(x)$ causada pela redução da energia refletida leva a uma baixa intensidade de brilho. A luz ambiente $\mathbf{L}_\infty(1-t(x))$ formada pelo espalhamento da luz atmosférica aumenta o brilho e reduz a saturação. Fonte: [5].

3.4.1.3 Modelo Final

O modelo final da formação de imagens em meios participativos frequentemente utilizado em métodos de *dehazing*, como [23] e [68], é dado por

$$\mathbf{I}(x) = \mathbf{L}_\infty \rho(x)\, e^{-\beta d(x)} + \mathbf{L}_\infty \left(1 - e^{-\beta d(x)}\right), \tag{2}$$

onde x representa um ponto na imagem, $\mathbf{I}$ é a imagem final, ρ é a refletância da cena, β é o coeficiente de atenuação do meio, d é a distância para o observador e $\mathbf{L}_\infty$ é a luz atmosférica. Nesta equação $\mathbf{I}$, $\mathbf{L}_\infty$ e ρ são vetores de cores (RGB), enquanto as outras variáveis são escalares. Neste modelo o sinal da imagem original é atenuado exponencialmente em função da distância d. A imagem também é degradada pelo componente aditivo de *backscattering*, que cresce exponencialmente em função da distância.

A operação de *dehazing* busca reconstruir a imagem original $\mathbf{J}(x)$, dada por $\mathbf{L}_\infty \rho(x)$. Nos casos onde a única informação disponível sobre a cena é uma única imagem $\mathbf{I}(x)$ este é um problema indeterminado, já que se tem uma única equação com 4 variáveis desconhecidas ($\rho(x)$, β, $d(x)$ e $\mathbf{L}_\infty$). Para se resolver este problema é necessário fazer suposições sobre a natureza de $\mathbf{J}$ e $\mathbf{L}_\infty$. Essas suposições permitem a elaboração de métodos capazes de gerar soluções aproximadas para o problema de *dehazing*. Essas soluções aproximadas tendem a ter uma qualidade satisfatória quando as suposições utilizadas são verdadeiras para a imagem de entrada.

3.4.2 DehazeNet

Os métodos de *dehazing* tradicionais utilizam uma série de suposições sobre as características das imagens afetadas por neblina para lidar com a ambiguidade inerente ao problema. Estas suposições, porém, são baseadas em estatísticas, e não são verdadeiras em todos os casos. Por essa razão, estes métodos tendem a falhar em alguns casos, onde as suposições nas quais eles são baseados não são verdadeiras. Para aliviar este problema em [5] é proposta a DehazeNet, uma arquitetura de Rede Neural Convolucional que busca realizar a estimativa do mapa de transmissão de uma imagem afetada por neblina. A principal vantagem da DehazeNet em relação aos métodos de *dehazing* tradicionais é que ela é capaz de aprender a realizar a extração de *features* automaticamente, através de aprendizado supervisionado, sem depender de suposições ou restrições elaboradas manualmente.

3.4.2.1 Arquitetura

A DehazeNet é uma Rede Neural Convolucional, composta por camadas de convolução e *pooling*, com funções de ativação não linear aplicadas depois de algumas destas camadas. A arquitetura da DehazeNet é baseada em alguns princípios e suposições dos métodos de *dehazing* tradicionais. Suas camadas estão organizadas para estimar o mapa de transmissão através de quatro operações sequenciais: extração de *features*, mapeamento em múltipla escala, extremos locais e regressão não linear. Um diagrama da arquitetura é apresentado na Figura 18.

3.4.2.1.1 Extração de Features

A extração de *features* relacionados à presença da névoa é uma parte importante dos algoritmos de *dehazing* tradicionais. A extração densa destes *features* é equivalente a uma operação de convolução da imagem de entrada com os filtros apropriados, seguida por um mapeamento não linear. Na DehazeNet o mapeamento não linear para redução de dimensão é realizado pela função de ativação Maxout [22]. Quando usada em RNCs, a Maxout gera um novo *feature map* através de uma operação de máximo pixel a pixel sobre k *feature maps*. A primeira camada da DehazeNet, baseada na função Maxout, é definida por:

$$F_1^i = \max_{j \in [1,k]} f_1^{i,j}(x), \; f_1^{i,j} = W_1^{i,j} * \mathbf{I} + B_1^{i,j},$$

onde $\mathcal{W}_1 = \{W_1^{p,q}\}_{(p,q)=(1,1)}^{(n_1,k)}$ representa os filtros, $\mathcal{B}_1 = \{B_1^{p,q}\}_{(p,q)=(1,1)}^{(n_1,k)}$ representa as variáveis de bias, e $*$ denota a operação de convolução. Existem n_1 *feature maps* na saída da primeira camada, ou seja, $i \in [1, n_1]$. $W_1^{i,j} \in \mathbb{R}^{3 \times f_1 \times f_1}$ é um de um total de $k \times n_1$ filtros de convolução, onde 3 é o número de canais na imagem de entrada $\mathbf{I}(x)$, e f_1 é o tamanho espacial do filtro. As unidades de Maxout mapeiam cada um dos vetores kn_1-dimensionais em um vetor n_1-dimensional, e extraem os *features* relevantes através de aprendizado automático, sem utilizar heurísticas, como ocorre nos outros métodos.

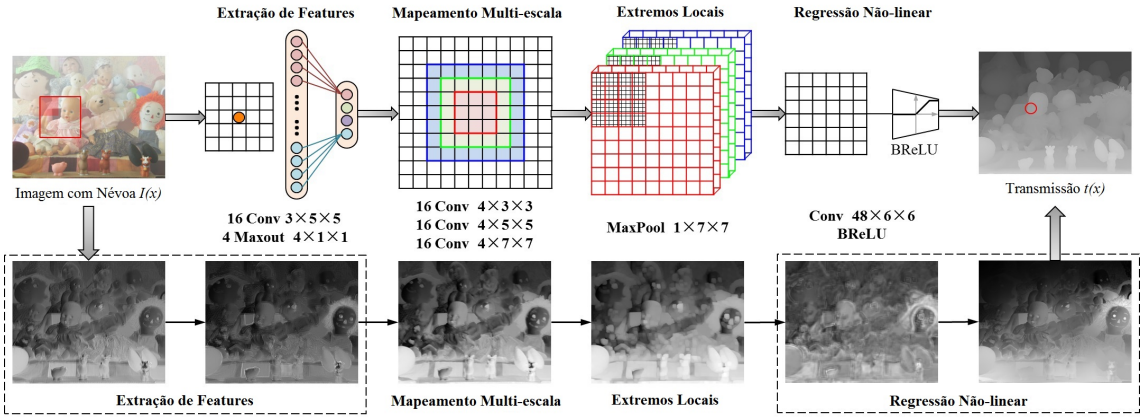


Figura 18: Arquitetura da DehazeNet. Fonte: [5].

3.4.2.1.2 Mapeamento em Múltipla Escala

O mapeamento em múltipla escala consiste na extração de *features* da imagem de entrada em várias escalas espaciais. Esta técnica é efetiva na obtenção de invariância à escala [5]. Na DehazeNet o mapeamento em múltipla escala é implementado na segunda camada, através da execução em paralelo de operações de convolução com filtros de tamanhos 3×3, 5×5 e 7×7. O mesmo número de filtros é usado nas três escalas. A saída da segunda camada é dada por:

$$F_2^i = W_2^{\lceil i/3 \rceil, (i \backslash 3)} * F_1 + B_2^{\lceil i/3 \rceil, (i \backslash 3)},$$

onde $\mathcal{W}_2 = \{W_2^{p,q}\}_{(p,q)=(1,1)}^{(3,n_2/3)}$ e $\mathcal{B}_2 = \{B_2^{p,q}\}_{(p,q)=(1,1)}^{(3,n_2/3)}$ contêm n_2 pares de parâmetros que estão divididos em 3 grupos. n_2 é a dimensão de saída da segunda camada e $i \in [1, n_2]$ é o índice dos *feature maps* de saída. $\lceil \rceil$ é a operação de teto e $\backslash$ denota a operação de resto.

3.4.2.1.3 Extremos Locais

Na arquitetura clássica das RNCs [38], a sensibilidade local é evitada considerando-se o máximo da vizinhança de cada pixel. O uso de extremos locais também está de acordo com a suposição de que a transmissão é localmente constante, geralmente utilizada para evitar ruído na estimativa da transmissão. Por estas razões, a operação de extremos locais é utilizada na terceira camada da DehazeNet:

$$F_3^i(x) = \max_{y \in \Omega(x)} F_2^i(y),$$

onde $\Omega(x)$ é uma vizinhança $f_3 \times f_3$ com centro em x, e a dimensão de saída da terceira camada é $n_3 = n_2$. Ao contrário da operação de *max pooling*, que geralmente reduz a resolução dos *feature maps*, a operação de extremos locais utilizada na DehazeNet é aplicada em cada pixel do *feature map*, preservando a resolução da entrada.

3.4.2.1.4 Regressão Não Linear

As funções de ativação mais frequentemente utilizadas em redes neurais são as funções Sigmoides e ReLu. As funções sigmoides tendem a ser afetadas pelo problema do desaparecimento de gradientes [26]. A função ReLu não é afetada por este problema com a mesma intensidade, porém ela foi desenvolvida para o problema de classificação, e por isso não é perfeitamente adequada para problemas de regressão, como a restauração de imagens. A utilização de ReLu pode provocar saturação na saída, especialmente na última camada, porque na restauração de imagens os valores de saída da última camada devem ficar limitados a um intervalo pequeno, com limite superior e inferior. Por esta razão, em [5] é proposta a função de ativação *Bilateral Rectified Linear Unit* (BReLu), dada por:

$$f(y) = \begin{cases} t_{min} & \text{para } y < t_{min} \\ y & \text{para } t_{min} \leq y < t_{max} \\ t_{max} & \text{para } y \geq t_{max} \end{cases}$$

O *feature map* da quarta camada da DehazeNet, baseada na função BReLu, é definido como:

$$F_4 = \min\left(t_{max}, \max\left(t_{min}, W_4 * F_3 + B_4\right)\right).$$

$\mathcal{W}_4 = \{W_4\}$ contém um filtro de tamanho $n_3 \times f_4 \times f_4$, $\mathcal{B}_4 = \{B_4\}$ é o bias, e t_{min} e t_{max} são os valores limite da função BReLu ($t_{min} = 0$ e $t_{max} = 1$ em [5]). O gradiente desta função de ativação é dado por:

$$\frac{\partial F_4(x)}{\partial F_3} = \begin{cases} \dfrac{\partial F_4(x)}{\partial F_3} & \text{para } t_{min} \leq F_4(x) < t_{max} \\ 0 & \text{para outros casos} \end{cases}$$

As quatro camadas descritas acima são unidas para formar uma RNC, onde os filtros e os bias são os parâmetros a serem aprendidos.

3.4.2.2 Relação com os Métodos de Dehazing Tradicionais

Na DehazeNet, o *feature map* da primeira camada F_1 é projetado para extrair *features* relacionados com a presença de névoa. Por exemplo, se os pesos W_1 formam um filtro de inversão (matriz esparsa com o valor -1 no centro de um canal) e $B_1 = 1$, a saída máxima do *feature map* é equivalente ao mínimo dos canais de cor, o que é similar ao dark channel [23]. Da mesma forma, quando os pesos formam um filtro com 1 no centro e -1 nas outras posições, F_1 é equivalente ao contraste máximo [68]; quando $\mathcal{W}_1$ inclui filtros passa-tudo e filtros de inversão, F_1 é similar aos *feature maps* máximo e mínimo, que são operações atômicas da transformação do espaço de cor RGB para o espaço HSV, o que leva a extração de *features* relacionados com a atenuação de cor [79] e disparidade de matiz [2]. Em conclusão, quase todos os *features* relevantes ao problema de *dehazing*

podem ser extraídos da primeira camada da DehazeNet ao final de um aprendizado bem sucedido. A função de ativação Maxout pode ser considerada a aproximação de uma função convexa arbitrária. Em [5] o máximo entre quatro *feature maps* é utilizado para aproximar uma função convexa.

Objetos de cor branca possuem altos valores de brilho e baixos valores de saturação, características similares às de regiões onde existe a presença de grande concentração de névoa. Por essa razão, praticamente todos os modelos de estimativa de névoa tendem a considerar os objetos brancos como distantes, o que resulta em uma estimativa de transmissão imprecisa. Com base na suposição de que a profundidade da cena é localmente constante, operações de extremos locais são frequentemente utilizadas para contornar este problema [68, 23, 79]. Na DehazeNet, os erros de estimativa locais são removidos através das operações de máximo local realizadas na terceira camada da rede. O termo de atenuação direta $\mathbf{J}(x)\,t(x)$ pode ser muito próximo de zero quando a transmissão $t(x)$ é próxima de zero, o que torna a restauração da cena $\mathbf{J}(x)$ sensível a ruído. A função BReLu restringe os valores de transmissão entre t_{min} e t_{max}, o que alivia o problema do ruído. A utilização da função BReLu é equivalente às restrições de limite superior e inferior utilizada nos métodos de *dehazing* tradicionais [23, 79].

3.4.2.3 Treinamento

Devido à dificuldade de se obter pares de imagens naturais com e sem degradação por neblina ou efeitos similares, em [5] a DehazeNet é treinada com um conjunto de dados de treinamento sintético. Os pares de imagens de treinamento são sintetizados com base em duas suposições: a) o conteúdo da imagem é independente da transmissão (o mesmo conteúdo pode aparecer e qualquer profundidade); b) a transmissão é localmente constante (pixels em uma pequena área tendem a ter profundidades parecidas). Segundo estas suposições, uma transmissão arbitrária pode ser atribuída a cada *patch* individual da imagem. Dado um *patch* livre de névoa $J^P(x)$, a luz atmosférica $\mathbf{L}_\infty$, e uma transmissão aleatória $t \in (0,1)$, um *patch* com névoa é sintetizado como $I^P(x) = J^P(x)\,t + \mathbf{L}_\infty(1-t)$. Para reduzir a incerteza no aprendizado, a luz atmosférica $\mathbf{L}_\infty$ é definida como 1. O treinamento é realizado com *patches* de tamanho 16×16, extraídos de imagens livres de névoa coletadas da internet.

A DehazeNet é treinada por aprendizado supervisionado, que busca encontrar a relação $\mathcal{F}$ entre os valores RGB e a transmissão. Os parâmetros da rede $\Theta = \{\mathcal{W}_1, \mathcal{W}_2, \mathcal{W}_4, \mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_4\}$ são encontrados através da minimização da função de erro entre o *patch* de treinamento $I^P(x)$ e a transmissão verdadeira correspondente t. A função de erro utilizada é o erro quadrático médio:

$$L(\Theta) = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathcal{F}\left(I_i^P;\Theta\right) - t_i\right\|^2$$

A rede é treinada com o algoritmo gradiente descendente estocástico.

3.4.2.4 Restauração

Assim como ocorre em outros métodos, os mapas de transmissão produzidos pela DehazeNet tendem a apresentar artefatos de bloco, provocados pela operação de máximo local realizada na terceira camada da rede, e por isso precisam passar por uma etapa de processamento adicional. Em [5] o algoritmo guided filter [24] é utilizado para suavizar o mapa de transmissão. Com base no mapa de transmissão refinado é possível realizar a restauração da imagem, através do mesmo procedimento utilizado nos métodos tradicionais. A luz atmosférica $\mathbf{L}_\infty$ pode ser estimada como correspondente à cor do pixel com maior intensidade em $\mathbf{I}\,(x)$ entre os 0.1% pixels de menor transmissão. Dadas a transmissão $t\,(x)$ e a luz atmosférica $\mathbf{L}_\infty$, a imagem restaurada $\mathbf{J}\,(x)$ pode ser recuperada através da equação 2.

3.4.2.5 Resultados

Em [5] a DehazeNet é comparada com diversos métodos de *dehazing*, com base em várias métricas diferentes. Nos testes com imagens sintéticas, a DahezeNet apresentou o melhor desempenho na grande maioria dos casos, em todas as métricas utilizadas, sendo superada apenas pelo método baseado na atenuação de cor [79] nos testes onde o coeficiente de atenuação atmosférica β é muito baixo. Nos testes com imagens reais, os autores consideram que a DehazeNet foi capaz de localizar as regiões de céu e produzir restaurações que preservam a cor original destas regiões, o que é um problema para alguns métodos de *dehazing*, como [23]. Os autores também consideram que a restauração das outras regiões da imagem é de boa qualidade. Uma comparação qualitativa entre os resultados da DehazeNet e de vários outros métodos é apresentada na Figura 19.

3.5 Restauração de Imagens Subaquáticas

A restauração de imagens subaquáticas está muito relacionada com a operação de *dehazing*. Assim como a degradação provocada por fenômenos atmosféricos, a degradação provocada pela água é frequentemente modelada através da equação 2 [6, 13, 8]. Apesar disto, existem algumas diferenças importantes entre os dois fenômenos. Uma delas é que a densidade de partículas em meios subaquáticos é muito maior que na atmosfera, e por isso o efeito de degradação ocorre de forma muito mais intensa, em distâncias muito menores. Uma outra diferença, ainda mais importante, é que na água a absorção dos raios de luz pelas partículas do meio produz um efeito muito mais significativo na atenuação da radiância da cena, enquanto na atmosfera a atenuação é provocada em sua maior parte pelo espalhamento dos raios de luz. A absorção dos raios de luz, ao contrario do espalhamento, é dependente do seu comprimento de onda, ou seja, da cor da luz.

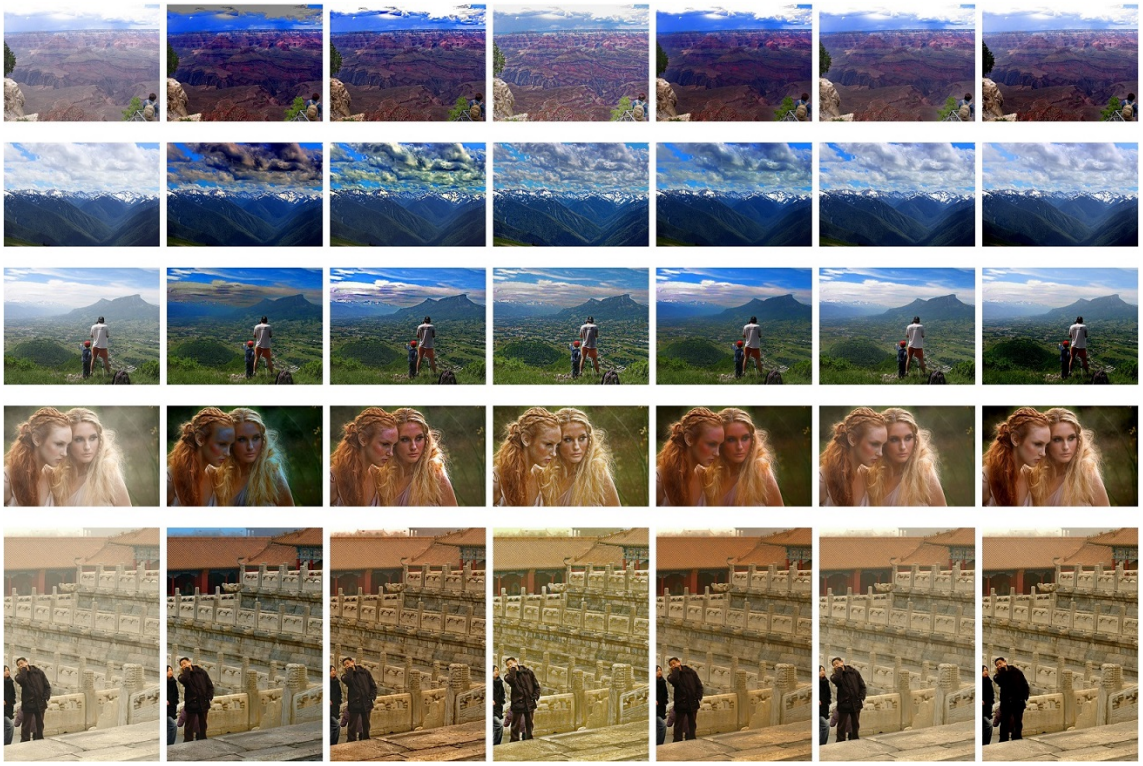


Figura 19: Comparação qualitativa entre os resultados de vários métodos de *dehazing*. Da esquerda para a direita: a imagem original, ATM [65], BCCR [44], FVR [69], DCP [23], CAP [79], e DehazeNet [5]. Fonte: [5].

A atenuação da cor vermelha é muita mais intensa do que a das outras cores [6], o que da às imagens subaquáticas um tom azul ou esverdeado, dependendo das características da água. Muitos dos métodos de *dehazing* são baseados em suposições que só são verdadeiras para imagens capturadas ao ar livre, onde a luz ambiente é branca ou cinza, e por isso a eficácia destes métodos cai drasticamente quando eles são aplicados a imagens subaquáticas. Para resolver este problema foram desenvolvidos métodos de restauração específicos para imagens subaquáticas, que levam em conta as características particulares das imagens capturadas neste meio. Neste trabalho duas redes neurais destinadas à restauração de imagens subaquáticas são utilizadas como estudo de caso: um modelo baseado na rede multi escala apresentada em [16] e outro inspirado na arquitetura Resnet [25].

3.6 Conclusão

Neste capítulo foram apresentadas arquiteturas de redes neurais aplicadas a diversos problemas de transformação de imagem. Em todos os casos as redes neurais apresentaram resultados tão bons quanto ou melhores que os dos métodos tradicionalmente aplicados aos problemas correspondentes. Estes resultados provam o potencial das redes neurais em problemas de reconstrução de imagem, porém na maior parte eles foram obtidos empiricamente, sem o conhecimento de como a rede trata o problema. A utilização de métodos

de visualização poderia ajudar na compreensão do funcionamento destes modelos, o que poderia levar a resultados ainda melhores.

Entre os problemas apresentados neste capítulo, são de interesse especial para este trabalho os problemas de remoção de névoa e restauração de imagens subaquáticas. Estes são problemas complexos, para os quais ainda não existem algoritmos que conseguem produzir soluções satisfatórias de maneira eficiente. Por esta razão, a aplicação de redes neurais à solução destes problemas é interessante, já que ela pode levar a melhores resultados com uma maior eficiência computacional. Uma das motivações para a realização deste trabalho é o desenvolvimento de ferramentas que podem auxiliar na compreensão e no aperfeiçoamento de arquiteturas de rede neural convolucional destinadas à resolução destes problemas.

4 METODOLOGIA

Neste trabalho técnicas de visualização de camadas intermediárias de redes neurais são aplicadas a redes neurais convolucionais de transformação de imagem. São utilizadas técnicas de visualização disponíveis na literatura, originalmente desenvolvidas para visualização de redes neurais de classificação e aqui adaptadas a redes de transformação de imagem. Além disso, é apresentado o método de visualização por inversão da rede, desenvolvido especialmente para as redes de transformação de imagem estudadas aqui.

Além dos métodos de visualização, são apresentadas redes de *dehazing* e restauração de imagens subaquáticas. Apesar de não apresentarem os resultados desejados em todas as situações, estas redes produzem restaurações satisfatórias para algumas imagens, e por isto podem ser utilizadas para estudo de caso.

4.1 Técnicas de Visualização

4.1.1 Visualização Direta

Apesar de ser uma das técnicas de visualização mais simples, a visualização direta da ativação produzida pelas imagens de entrada nos *feature maps* pode proporcionar informações importantes sobre o funcionamento da rede. Por exemplo, pode ser possível que alguns *feature maps* apresentem ativação alta nas regiões da imagem onde existem determinadas estruturas, o que indicaria que a presença destas estruturas é importante para determinar a saída da rede. Por esta razão, a visualização direta da ativação dos *feature maps* de diversas camadas da rede será utilizada neste trabalho.

Um outro método de visualização direta também utilizado aqui é a apresentação das 10 imagens do conjunto de treinamento que produzem as maiores ativações médias em *feature maps* específicos. Este método, em conjunto com outras técnicas, como a maximização da ativação, pode revelar quais as características são responsáveis pela ativação dos *feature maps* em questão.

4.1.2 Maximização da Ativação

A maximização da ativação [17] é uma técnica de simples implementação, que permite descobrir que tipo de padrão provoca a ativação máxima de um determinado *feature map* da rede. Nas redes de classificação, a maximização da ativação tende a produzir padrões que lembram determinados objetos, como rodas de carro, garrafas, lâmpadas, faces de cachorros, etc. O problema de restauração de imagens é muito diferente do problema de classificação, e por isso é possível que a aplicação de maximização da ativação a redes de transformação de imagem produza resultados diferentes. Neste trabalho a maximização da ativação é utilizada para descobrir a quais padrões as redes de transformação de imagem estudadas são sensíveis.

Os resultados produzidos pela maximização da ativação tendem a apresentar padrões de alta frequência, além de outros artefatos, que dificultam a sua interpretação. Este problema pode ser amenizado por técnicas de regularização, que influenciam a maximização da ativação a produzir resultados mais semelhantes a imagens naturais. Das técnicas de regularização apresentadas em [75], a única que melhorou consideravelmente a interpretabilidade dos resultados foi o filtro Gaussiano. Por este motivo, neste trabalho a maximização da ativação é regularizada por filtro Gaussiano, utilizando os mesmos parâmetros que produziram os melhores resultados em [75]. As outras técnicas de regularização não são aplicadas, pois elas não beneficiam ou até mesmo prejudicam a interpretação dos resultados.

4.1.2.1 Normalização de Gradiente por Pirâmide Laplaciana

Em geral, as técnicas de regularização da maximização da ativação são aplicadas às imagens de entradas, entre os passos de otimização. Uma alternativa é aplicar regularização diretamente ao gradiente produzido a cada iteração. Uma forma de regularização que pode ser aplicada ao gradiente para tornar os resultados da otimização mais interpretáveis é a normalização por pirâmide Laplaciana.

A pirâmide Laplaciana [4] é uma técnica de codificação desenvolvida para permitir o armazenamento de imagens em um espaço reduzido. Neste método as imagens são armazenadas em uma estrutura em forma de pirâmide, onde o topo é uma versão de baixa resolução da imagem original e os níveis inferiores são a diferença entre a imagem original e uma versão suavizada da mesma.

Seja $g_0(ij)$ a imagem original, e $g_1(ij)$ o resultado da aplicação de um filtro passa-baixa, como o filtro gaussiano, em g_0. A diferença $L_0(ij)$ é dada por

$$L_0(ij) = g_0(ij) - g_1(ij)$$

Ao invés de se codificar g_0, codifica-se L_0 e g_1. Isto resulta em uma redução no espaço de armazenamento necessário, já que L_0 pode ser representada com muito menos bits que

g_0, e g_1 pode ser armazenada com uma taxa de amostragem reduzida, já que foi filtrada com um filtro passa-baixa. Uma maior taxa de compressão pode ser atingida com mais iterações deste processo. A aplicação de um filtro passa-baixa na imagem g_1 resulta em uma imagem g_2 e uma segunda imagem de diferença é obtida: $L_2(ij) = g_1(ij) - g_2(ij)$. Repetindo-se este processo diversas vezes se obtém uma sequencia de imagens L_0, L_1, L_2, ..., L_n, onde cada uma tem, em cada dimensão, a metade do tamanho da imagem anterior.

Formalmente, a pirâmide Laplaciana é definida como uma sequência de imagens de erro L_0, L_1, ..., L_n, onde cada imagem é a diferença entre dois níveis de uma pirâmide Gaussiana. Logo, para $0 \leq 1 < N$,

$$L_l = g_l - \text{EXPANDE}(g_{l+1})$$

onde EXPANDE$(\cdot)$ é uma operação de aumento de dimensão, que expande a imagem ao tamanho do nível anterior. Como não existe uma imagem g_{N+1} para ser utilizada no calculo de L_N, utiliza-se $L_N = g_N$.

A decodificação de uma pirâmide Laplaciana pode ser realizada através da inversão dos passos da sua geração. A imagem L_N é expandida uma única vez e adicionada a L_{N-1}, que então é expandida e adicionada a L_{N-2}, e assim por diante, até que o nível 0 e alcançado e a imagem g_0 é recuperada.

$$g_N = L_N$$

e para $l = N-1, N-2, ..., 0$,

$$g_l = L_l + \text{EXPANDE}(g_{l+1}).$$

Quando o gradiente é decomposto em uma pirâmide Laplaciana, cada um dos níveis passa a conter os componentes de uma faixa de frequência. Os níveis mais altos da pirâmide guardam as informações de baixa frequência, enquanto as altas frequências ficam armazenadas nos níveis mais baixos. A normalização de gradiente por pirâmide Laplaciana é um processo onde cada um destes níveis é normalizado de forma independente, antes de a pirâmide ser decodificada. O resultado desta operação é que os níveis mais altos, que normalmente têm valores dentro de uma escala reduzida, ficam em uma escala mais próxima da dos níveis mais baixos, que costumam apresentar valores maiores. Em outras palavras, a normalização de gradiente por pirâmide Laplaciana realça as baixas frequências do gradiente, o que leva a otimização a produzir imagens com menos informações de alta frequência e mais informações de baixa frequência, tornando assim os resultados mais interpretáveis.

A implementação do algoritmo de normalização de gradiente por pirâmide Laplaciana

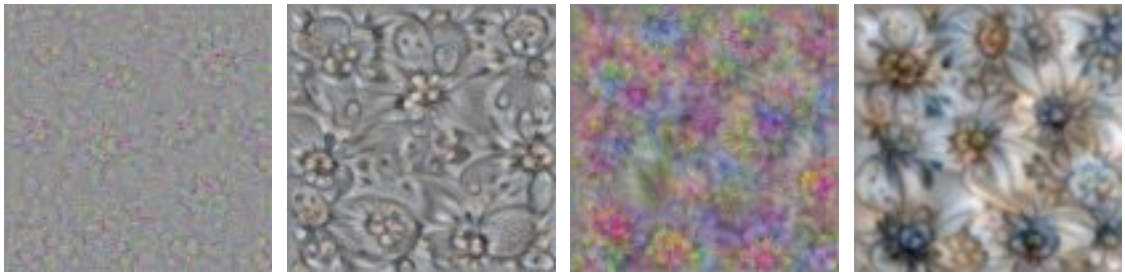
utilizada neste trabalho é descrita no algoritmo 1. O parâmetro N define o número de níveis da pirâmide. A operação de redução de dimensão é realizada através da aplicação de um filtro Gaussiano 5×5 com *stride* (distância, em pixels, entre os pontos de aplicação do *kernel* convolucional) 2 nas dimensões x e y. A aplicação de expansão é uma convolução transposta utilizando o mesmo filtro, multiplicado por 4, com *stride* 2 nas dimensões x e y, o que resulta em uma imagem com o dobro tamanho.

Algoritmo 1 Normalização de gradiente por pirâmide Laplaciana.

Entrada: gradiente g
Saída: gradiente normalizado g_{norm}
início
  $g_0 = g$;
  para $l \leftarrow 0$ ***até*** $N - 1$ **faça**
    $g_{l+1} \leftarrow$ CONVOLUÇÃO$(g_l, Gauss)$;
    $L_l =\leftarrow g_l -$ CONVOLUÇÃO_TRANSPOSTA$(g_{l+1}, 4 * Gauss)$;
  fim
  $L_N \leftarrow g_N$;
  para $l \leftarrow 0$ ***até*** N **faça**
    $\sigma_l \leftarrow \sqrt{\overline{L_l^2}}$;
    $L_l = L_l / \max(\sigma_l, 10^{-10})$;
  fim
  $g_N \leftarrow L_N$;
  $l \leftarrow N - 1$;
  enquanto $l \geq 0$ **faça**
    $g_l \leftarrow$ CONVOLUÇÃO_TRANSPOSTA$(g_{l+1}, 4 * Gauss) + L_l$;
    $l = l - 1$;
  fim
  $g_{norm} \leftarrow g_0$;
  retorna g_{norm};
fim

Uma comparação entre diferentes métodos de regularização é apresentada na Figura 20, onde são mostrados os resultados da maximização da ativação para o canal 139 da camada mixed4d_3x3_bottleneck_pre_relu da rede de classificação GoogLeNet [67] utilizando diferentes métodos de regularização. O *feature map* em questão aparentemente é um detector de flores. Quando nenhum método de regularização é utilizado (20a), a maximização da ativação resulta em uma imagem com padrões coloridos de alta frequência, que dificultam a interpretação dos resultados. A regularização por filtro Gaussiano (20b) deixa as flores mais visíveis, com contornos mais definidos, porém a imagem resultante é predominantemente cinza. A normalização de gradiente por pirâmide Laplaciana com 5 níveis (20c), por outro lado, resulta em uma imagem extremamente colorida, porém o contorno das flores não é tão bem definido e existe uma quantidade considerável de ruído. Finalmente, a junção dos dois métodos (20d) resulta em uma imagem onde o contorno das

flores é realçado sem que as cores sejam completamente diluídas. Devido aos resultados apresentados, considerou-se que a combinação entre a regularização por filtro Gaussiano e a normalização de gradiente por pirâmide Laplaciana produz as imagens que podem ser mais facilmente interpretadas. Logo, este método de regularização é utilizado para gerar todos os resultados de maximização da ativação apresentados neste trabalho.



(a) Sem regularização (b) Filtro Gaussiano (c) Pirâmide Laplaciana (d) Filtro Gaussiano e pirâmide Laplaciana

Figura 20: Resultados da maximização da ativação para o canal 139 da camada mixed4d_3x3_bottleneck_pre_relu da rede de classificação GoogLeNet [67] utilizando diferentes métodos de regularização.

4.1.3 Inversão da Rede

A visualização por inversão da rede é um método desenvolvido especialmente para redes de transformação de imagem. Inspirado pela maximização da ativação, este método utiliza otimização por gradiente descendente para encontrar a imagem de entrada que produz em uma rede a saída mais próxima de uma imagem específica.

Seja $\mathbf{x}$ uma imagem, $F(\mathbf{x})$ a saída da rede F quando recebe $\mathbf{x}$ como entrada, e Y uma imagem arbitrária nas mesmas dimensões de $F(\mathbf{x})$. A inversão da rede pode ser vista como um problema de otimização que busca a imagem $\mathbf{x}^*$ onde

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \left(|F(\mathbf{x}) - Y| \right).$$

Assim como na maximização da ativação, esta busca é realizada através otimização por gradiente descendente. A inversão da rede é um processo iterativo, onde a cada iteração se da um passo na direção inversa do gradiente da distância L_1 entre a saída da rede e a imagem alvo em relação à imagem entrada da rede. O processo se encerra após um número de iterações predefinido, que deve ser grande o suficiente para a convergência da otimização.

Assim como a maximização da ativação, a inversão da rede está sujeita a ocorrência de ruído. A otimização pode resultar em imagens compostas quase que exclusivamente por ruído, mas que produzem na saída da rede uma imagem muito parecida com a imagem alvo. Um exemplo desta situação é apresentado na Figura 21. Este problema pode

ser amenizado com a utilização das mesmas técnicas de regularização utilizas na maximização da ativação. Em geral, a combinação de regularização por filtro Gaussiano e normalização de gradiente por pirâmide Laplaciana produz bons resultados. Apesar disso, a necessidade da utilização de regularização depende da própria rede. Em alguns casos, os melhores resultados são obtidos quando a processo de otimização converge livremente, sem nenhum tipo de regularização. Uma outra forma de reduzir a ocorrência de ruído é inicializar o processo de otimização com uma imagem completamente cinza, e não uma imagem aleatória, como normalmente é feito na maximização da ativação [17, 75].

(a) Resultado da otimização (b) Saída da rede (c) Imagem alvo

Figura 21: Resultados da visualização por inversão da rede sem regularização para uma rede de *dehazing*. A otimização resulta em uma imagem com uma quantidade extrema de ruído (a). A resposta da rede para esta imagem (b), porém, é muito parecida com a imagem alvo (c).

Um dos parâmetros mais importantes da visualização por inversão da rede é o tamanho do passo dado a cada iteração da otimização. Além de afetar o número de iterações necessárias para a convergência, o tamanho do passo de otimização tem um impacto significativo sobre a qualidade do resultado. Em geral, passos de otimização maiores produzem imagens que, quando transformadas pela rede, apresentam uma estrutura global próxima da imagem alvo, porém sem muita definição, enquanto passos de otimização menores resultam em imagens cujas transformações possuem detalhes com melhor definição, mas apresentam artefatos globais consideráveis. Exemplo de como o tamanho de passo influencia o resultado da visualização são apresentados nas figuras 22 e 23.

A visualização por inversão da rede pode fornecer informações importantes sobre o que foi aprendido pela rede. Uma das características da otimização por gradiente descendente é que ela só considera as características da entrada que tem influência no resultado final. Deste foram, é possível ter uma ideia de quais *features* a rede utiliza como base para tomar decisões. A inversão da rede também pode mostrar quais são as entradas "preferidas pela" rede, ou seja, o tipo de entrada ao qual a rede esta melhor adaptada.

Todos os resultados de visualização por inversão da rede apresentados neste trabalho foram obtidos utilizando um tamanho de passo de 10^{-4}. Este valor foi escolhido porque,

(a) Tamanho de passo: 10^{-3}

(b) Tamanho de passo: 10^{-4}

(c) Tamanho de passo: 10^{-5}

Figura 22: Resultados da visualização por inversão da rede para uma rede de *dehazing*, utilizando diferentes tamanhos de passo. Da esquerda para a direita, a imagem alvo, o resultado da otimização, e a saída da rede para o resultado da otimização. Em todos os casos foram utilizados os mesmos métodos de regularização: aplicação de filtro Gaussiano de raio 1 a cada 4 iterações e normalização de gradiente por pirâmide Laplaciana de 5 níveis. Em (a), a saída da rede para o resultado da otimização apresenta uma estrutura global próxima da imagem alvo, porém baixa definição. Em (b), a saída da rede apresenta uma melhor definição, porém alguns artefatos podem ser observados, principalmente no canto superior esquerdo da imagem. Em (c), a saída da rede apresenta artefatos consideráveis e uma aparência desfocada, possivelmente provocada por uma regularização excessivamente forte.

(a) Tamanho de passo: 10^{-3}

(b) Tamanho de passo: 10^{-4}

(c) Tamanho de passo: 10^{-5}

Figura 23: Resultados da visualização por inversão da rede para uma rede de *dehazing*, utilizando diferentes tamanhos de passo. Da esquerda para a direita, a imagem alvo, o resultado da otimização, e a saída da rede para o resultado da otimização. Em todos os casos foram utilizados os mesmos métodos de regularização: aplicação de filtro Gaussiano de raio 1 a cada 4 iterações e normalização de gradiente por pirâmide Laplaciana de 5 níveis. Em (a), a saída da rede para o resultado da otimização apresenta uma estrutura global próxima da imagem alvo, porém baixa definição. Além disso, a toalha que deveria ser verde tem a cor azul. Em (b), a textura da madeira pode ser vista com mais nitidez e a cor da toalha é mais próxima da imagem alvo, porém alguns artefatos podem ser observados, principalmente no canto inferior direito da imagem. Em (c), a saída da rede apresenta uma aparência mais suave, porém algumas regiões da imagem apresentam uma coloração azulada.

após alguns testes, considerou-se que em geral ele produz a melhor relação entre qualidade dos resultados e tempo necessário para a convergência. Os métodos de regularização utilizados variam de acordo com a rede. Em algumas redes é necessária uma regularização forte para evitar a ocorrência de ruído, enquanto em outras os melhores resultados são obtidos quando a otimização converge sem nenhum tipo de normalização. A quantidade de iterações necessárias para a convergência do processo de otimização também varia, dependendo da rede e dos métodos de regularização utilizados.

4.2 Estudos de Caso

Nesta seção são apresentadas as arquiteturas de rede neural de transformação de imagem estudadas neste trabalho. Estas arquiteturas estão relacionadas com problemas complexos, como estimativa de mapa de profundidade, remoção de névoa e restauração de imagens subaquáticas.

4.2.1 DehazeNet

Um dos passos mais importantes nos métodos tradicionais de restauração de imagens degradadas por meios participativos é a estimativa do mapa da transmissão. Em [5] é proposta a DehazeNet, uma Rede Neural Convolucional que realiza a estimativa do mapa de transmissão de uma cena degradada por névoa através de uma única imagem de entrada. Diferente do modelo apresentado em [5], que foi treinado para estimar a transmissão de imagens com névoa, a rede visualizada aqui foi treinada para estimar a transmissão de imagens subaquáticas.

4.2.2 RNC Multi-Escala de Restauração de Imagens Subaquáticas

Como o coeficiente de atenuação atmosférica normalmente é considerado constante dentro de uma imagem, o mapa de transmissão está diretamente relacionado com o mapa de profundidade. Uma arquitetura de RNC que realiza a estimativa do mapa de profundidade de uma imagem é proposta em [16]. Como esta arquitetura é capaz de estimar o mapa de profundidade de uma imagem, é razoável se supor que ela também possa estimar o mapa de transmissão de uma imagem subaquática. Como a estimativa do mapa de transmissão é a etapa mais importante do processo de restauração de imagens subaquáticas, também pode-se supor que a arquitetura em questão possa ser utilizada como uma arquitetura de restauração *end-to-end*. Com base nestas suposições, uma versão modificada da arquitetura apresentada em [16] foi treinada para restaurar imagens subaquáticas. Os resultados da visualização deste modelo são apresentados neste trabalho.

4.2.3 RNC Residual de Estimativa de Profundidade

Em [37] é apresentada uma arquitetura residual de estimativa de profundidade capaz de produzir resultados superiores aos da arquitetura multi-escala apresentada em [16]. Como a arquitetura em questão é diretamente baseada na rede de classificação ResNet-50 [25], o seu estudo permite avaliar como uma mesma arquitetura se comporta quando aplicada a problemas diferentes, levando assim a uma melhor compreensão da relação entre os problemas de transformação de imagem e classificação.

4.2.4 Dehaze Resnet 12

Uma das arquiteturas utilizadas como estudo de caso neste trabalho é a Dehaze Resnet 12, uma rede neural convolucional de remoção de névoa. A arquitetura da Dehaze Resnet 12 é inspirada na rede de classificação ResNet-34 [25]. Existem, porém, algumas diferenças importantes entre os dois modelos. No caso do problema de *dehazing*, a saída esperada é uma imagem com a mesma resolução da imagem de entrada. Logo, é importante que nenhuma informação presente na entrada seja perdida, e por esta razão a arquitetura em questão não utiliza *pooling* ou qualquer outra operação de redução de dimensão. Como todos os *feature maps* da rede possuem a mesma resolução da imagem de entrada, é necessária a redução do número de *feature maps* em cada camada para manter o tempo de processamento e o uso de memória em níveis razoáveis. Outra característica da rede é que ela é muito mais rasa (possui menos camadas) que as arquiteturas que são o estado da arte em classificação de imagens.

A rede é composta por blocos residuais semelhantes aos usados na arquitetura ResNet-34. Em cada um desses blocos, a entrada passa por duas operações de convolução consecutivas, ambas com 64 filtros de tamanho $3 \times 3 \times 64$. Cada uma destas operações é seguida de uma camada de normalização de *batch*, com uma conexão residual entre a entrada do bloco e a saída da segunda camada de normalização de *batch*. Finalmente, a função de ativação ReLu é aplicada à saída da conexão residual. Um diagrama dos blocos utilizados é apresentado na Figura 24.

A rede propriamente dita possui a seguinte estrutura: A entrada é uma imagem de tamanho $W \times H \times 3$, sobre a qual se aplica uma operação de convolução com 64 filtros de tamanho $7 \times 7 \times 3$, seguida por uma camada de normalização de *batch*. Os *feature maps* resultantes (chamados de **conv1**) passam por doze blocos residuais consecutivos (cujas saídas são chamadas **residual1-residual12**). À saída do último bloco residual se aplica uma convolução com três filtros de tamanho $7 \times 7 \times 64$, resultando em uma imagem de saída de tamanho $W \times H \times 3$. Finalmente, aplica-se a função de ativação BReLU à saída desta última camada convolucional. Um diagrama da arquitetura da rede é apresentado na Figura 25. O resultado da aplicação desta arquitetura em algumas imagens reais é mostrado na Figura 26.

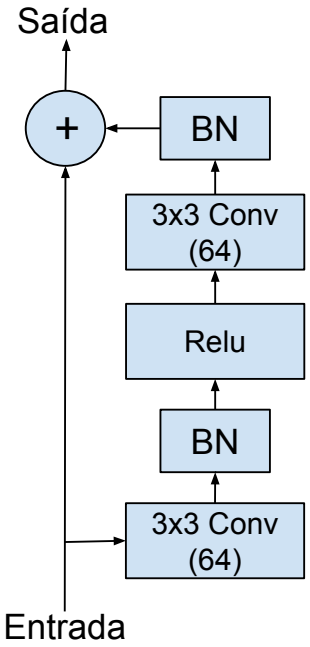


Figura 24: Arquitetura dos blocos residuais utilizados na rede Dehaze Resnet 12.

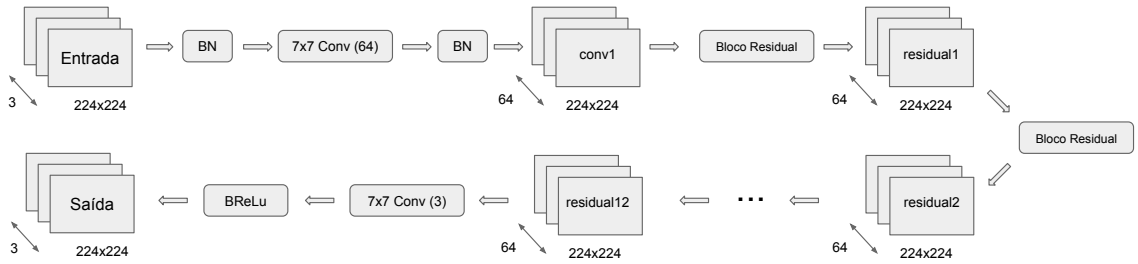


Figura 25: Diagrama da arquitetura Dehaze Resnet 12.

4.2.4.1 Treinamento

Um dos principais problemas enfrentados no treinamento de redes de *dehazing* é a aquisição de dados de treinamento. Capturar duas imagens da mesma cena, com e sem névoa, sob as mesmas condições de iluminação é uma tarefa extremamente difícil. A utilização de dados simulados, criados através da adição de névoa simulada a imagens reais, é uma forma de contornar este problema. Para se adicionar degradação simulada a uma imagem limpa são necessários o mapa de profundidade da cena, a luz ambiente e o coeficiente de atenuação do meio. Para o treinamento da arquitetura Dehaze Resnet 12 é utilizado um conjunto de dados composto por pares de imagens RGB e seus respectivos mapas de profundidade, além de um conjunto de imagens de turbidez, que são *patches* extraídos de imagens reais que contém apenas regiões da imagem onde a transmissão é próxima de zero, utilizadas como referência para o cálculo da luz ambiente e dos coeficientes de atenuação do meio. Estes dados são utilizados para simular imagens com névoa de acordo com o modelo descrito na equação 2. O conjunto de dados de trei-

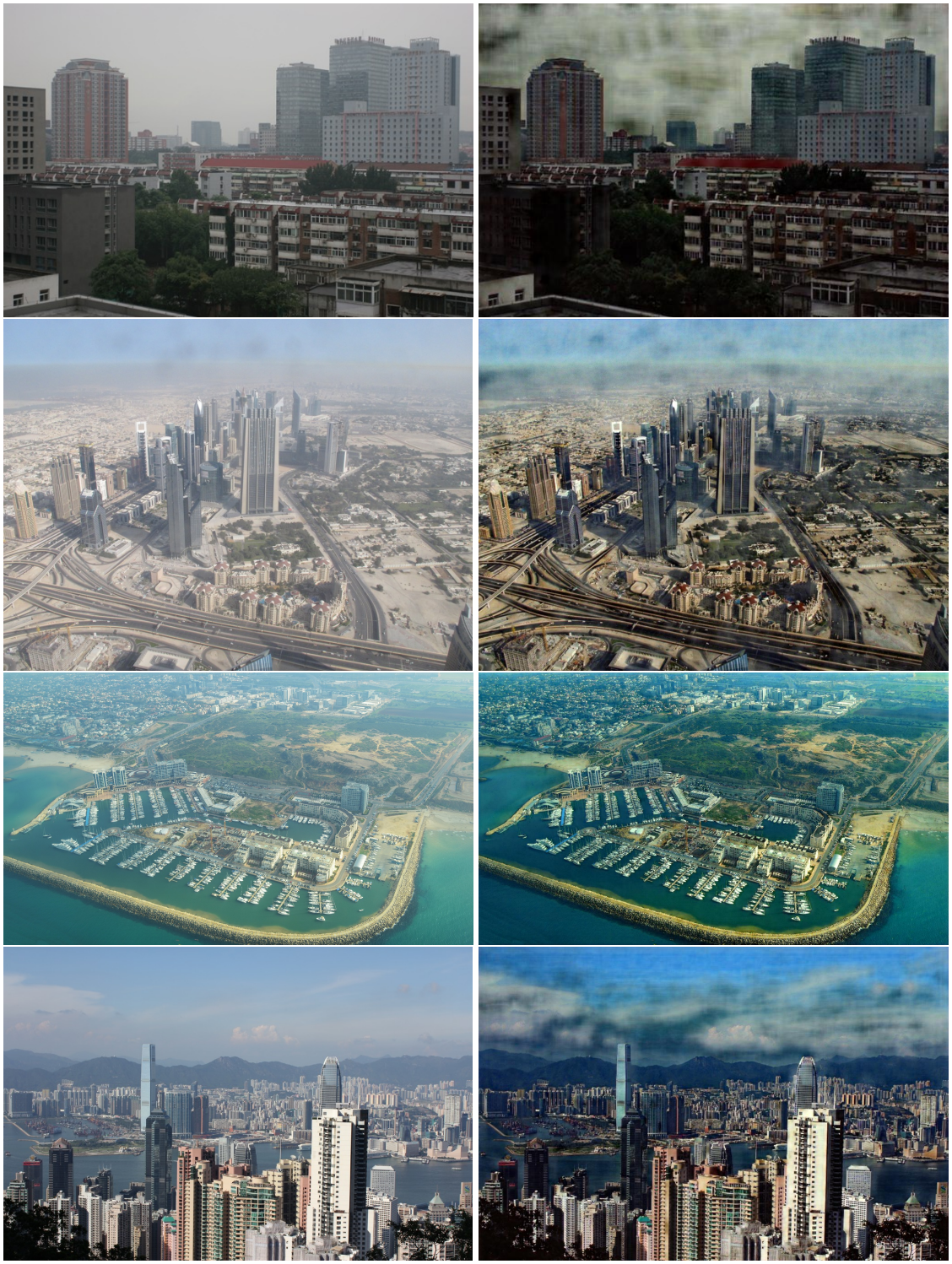


Figura 26: Resultados da rede de remoção de névoa Dehaze Resnet 12. As imagens à esquerda são as entradas da rede, as imagens à direita são o resultado da restauração.

namento utilizado é composto por 8732 *patches* de tamanho 224×224, extraídos dos *datasets* NYU-Depth [61] e B3DO [31], e 4 imagens de turbidez, resultando em um total de 34928 exemplos de treinamento possíveis. A rede é treinada utilizando o algoritmo de otimização Adam com uma taxa de aprendizagem de 10^{-4}, por 240 épocas, onde cada

época consiste em 8732 exemplos de treinamento. A função de erro utilizada é a *perceptual loss* [32], baseada na camada conv22 da rede de classificação VGG16 [63]. A rede é implementada utilizando framework tensorflow [1].

4.2.5 Underwater Resnet 12

A Underwater Resnet 12 é uma versão da Dehaze Resnet 12 destinada à restauração de imagens subaquáticas. As duas redes possuem exatamente a mesma arquitetura, a diferença entre elas está no treinamento. As imagens de turbidez utilizadas no treinamento da Underwater Resnet 12 foram extraídas de imagens subaquáticas, o que da aos dados simulados características parecidas com as de imagens capturadas alguns metros abaixo da superfície do mar. A rede foi treinada com quatro imagens de turbidez, cada uma contendo um tom de azul diferente. Uma outra diferença é que o conjunto de imagens de treinamento foi incrementado com *patches* extraídos de imagens de ambientes internos capturadas com o sensor Kinect 2.0, resultando em um total de 15021 pares de treinamento e 836 pares de validação, cada um composto por uma imagem RGB e seu respectivo mapa de profundidade. A rede foi treinada por 120 épocas, onde cada época consiste em 15021 exemplos de treinamento. Com exceção destas diferenças, o processo de treinamento é o mesmo utilizado na Dehaze Resnet 12. Alguns resultados da aplicação da Underwater Resnet 12 em imagens subaquáticas reais são apresentados na Figura 27.

Figura 27: Resultados da rede de restauração de imagens subaquáticas Underwater Resnet 12. As imagens à esquerda são as entradas da rede, as imagens à direita são o resultado da restauração.

5 RESULTADOS

Neste capítulo são apresentados os resultados da aplicação de técnicas de visualização de redes neurais nas arquiteturas selecionadas como estudo de caso. Todos os métodos de visualização utilizados, assim como as arquiteturas de rede neural estudas, foram implementados com o framework tensorflow [1].

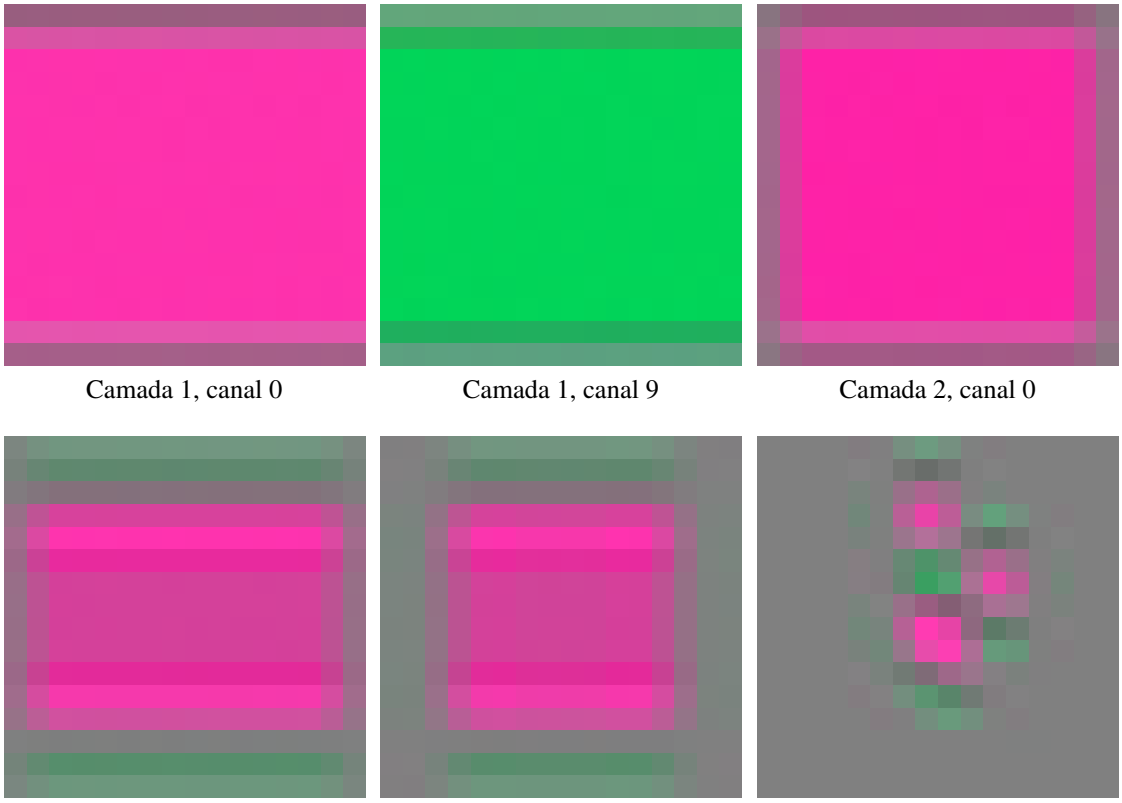
5.1 DehazeNet

A visualização por maximização da ativação foi aplicada a uma versão modificada da arquitetura DehazeNet, treinada para estimar o mapa de transmissão de imagens subaquáticas. Alguns dos resultados obtidos são apresentados na Figura 28.

Em praticamente todos os casos, a maximização da ativação gerou imagens muito semelhantes para todos os *feature maps* dentro de uma mesma camada da rede. A exceção é o canal 9 da primeira camada, que é praticamente o oposto dos outros canais. Assim como em [75], houve um aumento na complexidade dos padrões apresentados nas camadas mais profundas. Apesar disto, não é possível identificar nos resultados nenhum padrão que lembra alguma estrutura do mundo real. Uma característica comum à praticamente todos os resultados é a presença de um tom de cor magenta, com valores muito altos no canal de cor vermelho. Isto pode significar que a rede considera a presença da cor vermelha como um indicativo de alta transmissão, o que está de acordo com a suposição apresentada em [8].

5.2 RNC Multi-Escala de Restauração de Imagens Subaquáticas

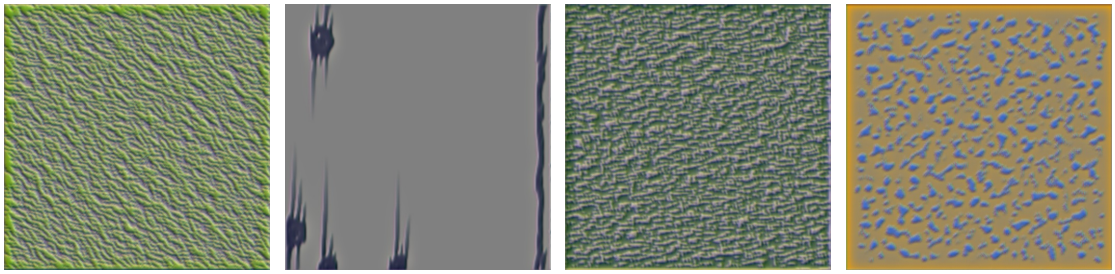
Técnicas de visualização foram aplicadas a uma versão modificada da arquitetura de estimativa de profundidade apresentada em [16]. A principal diferença da rede estudada aqui para a arquitetura original é que ela foi treinada para reconstruir imagens subaquáticas. Os resultados da maximização da ativação de alguns de seus *feature maps* são apresentados nas figuras 29 e 30. A visualização direta da ativação produzida por imagens de treinamento nos *feature maps* da primeira camada convolucional da rede é apresentada nas figuras 31, 32 e 33.



Camada 1, canal 0 | Camada 1, canal 9 | Camada 2, canal 0

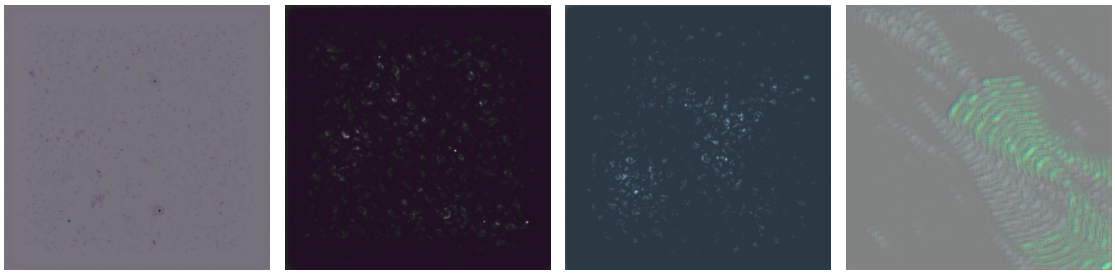
Camada 3, canal 0 | Camada 4, canal 0 | Camada 5, canal 0

Figura 28: Resultados da maximização da ativação das camadas convolucionais da arquitetura DehazeNet. Esta figura é melhor visualizada em cores.



Canal 7 Canal 31 Canal 57 Canal 63

Figura 29: Maximização da ativação dos *feature maps* da primeira camada convolucional da rede multi-escala de restauração de imagens subaquáticas. Esta figura é melhor visualizada em cores.

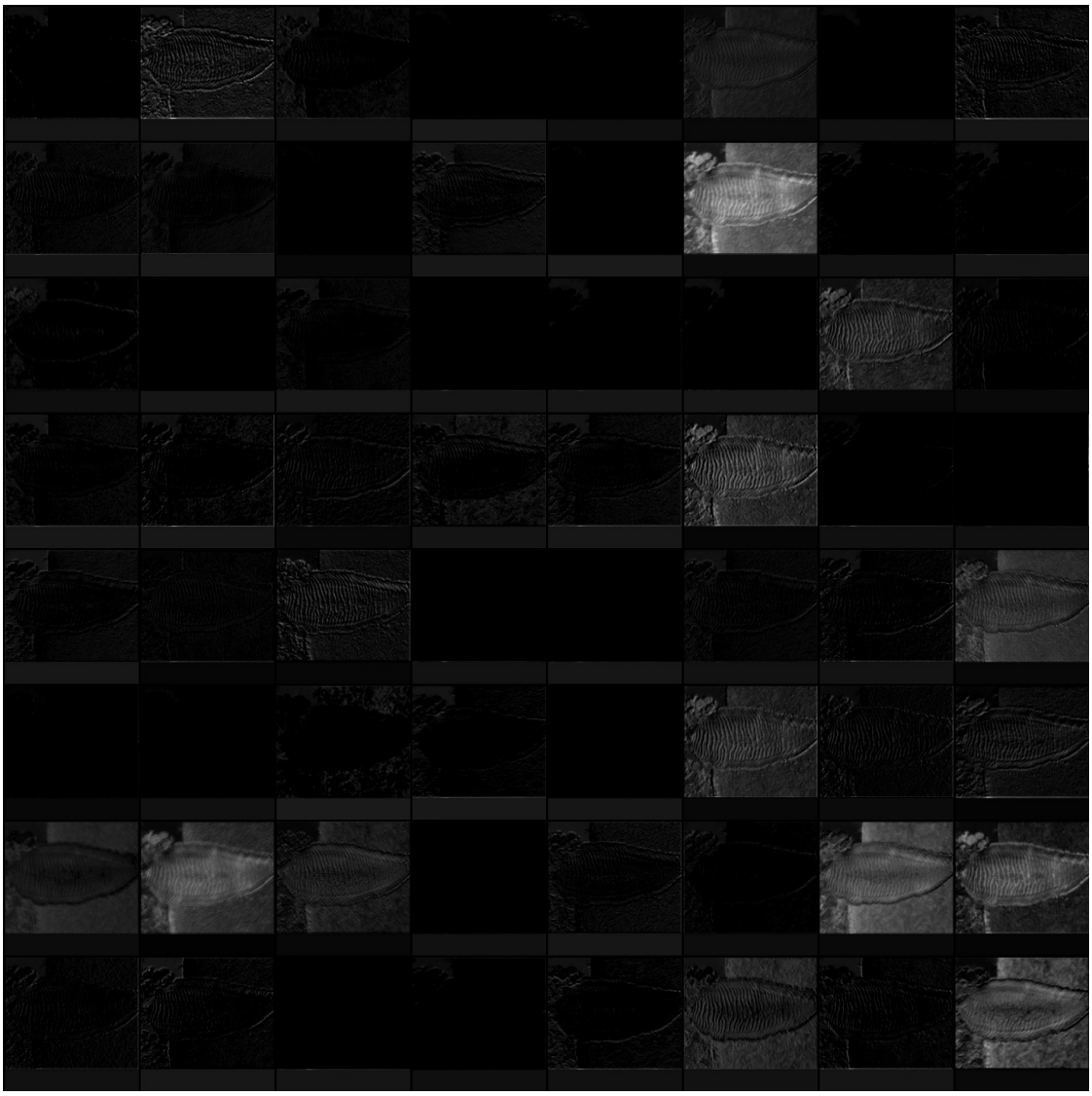


Canal 6 Canal 8 Canal 11 Canal 30

Figura 30: Maximização da ativação dos *feature maps* de uma camada convolucional intermediaria da rede multi-escala de restauração de imagens subaquáticas. Esta figura é melhor visualizada em cores.

Os resultados mostram que existe uma grande variação entre os *feature maps* de uma mesma camada. A visualização direta dos *feature maps* da primeira camada convolucional apresentou diversas ativações diferentes para cada imagem de entrada, incluindo algumas que indicam a presença detectores de bordas em diferentes orientações. A maximização da ativação produziu diferentes texturas de variadas cores na primeira camada convolucional e padrões variados em uma camada mais profunda.

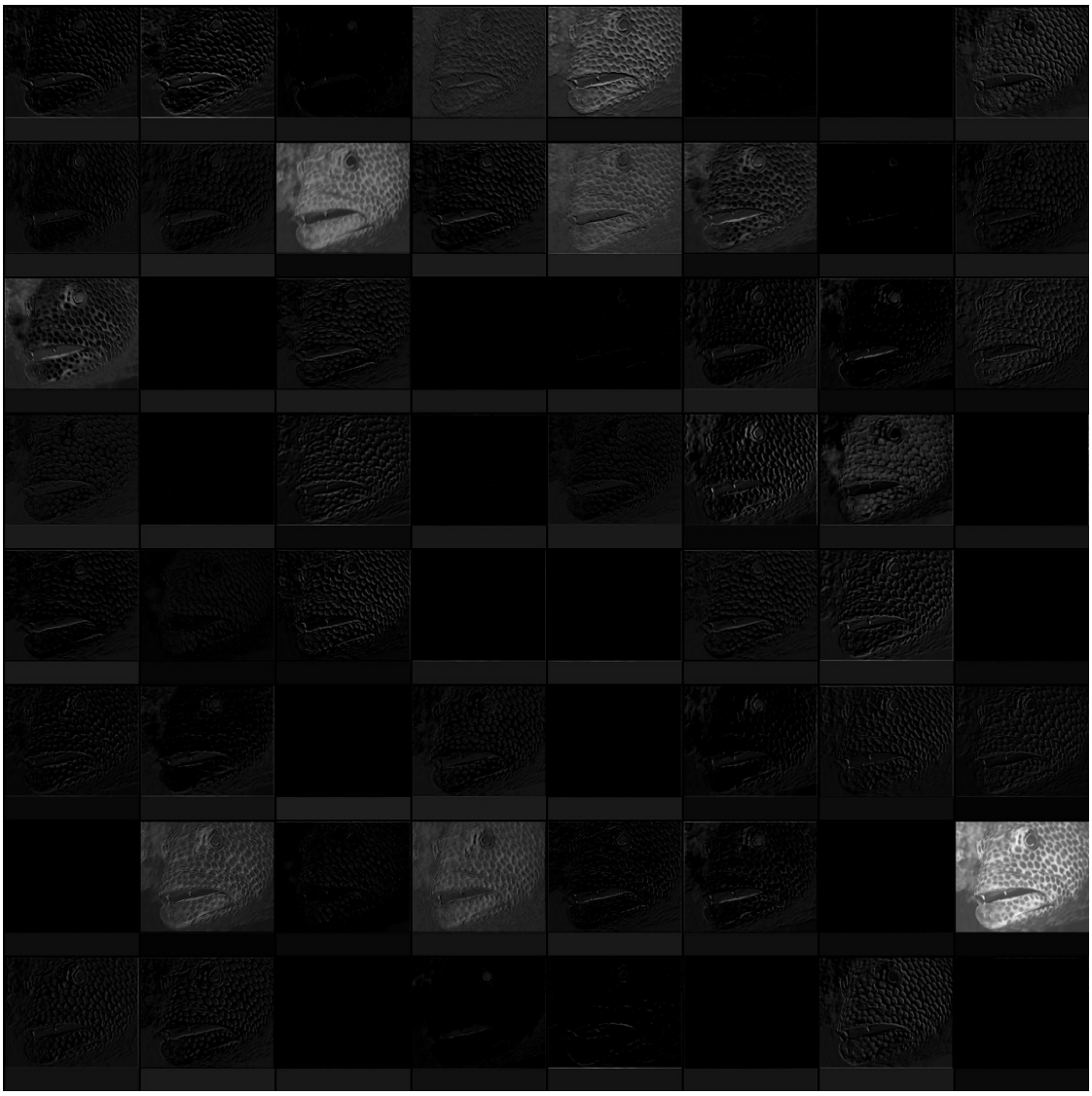
Entrada Saída Ground Truth



Feature maps

Figura 31: Visualização direta dos *feature maps* da primeira camada convolucional da rede multi-escala de restauração de imagens subaquáticas.

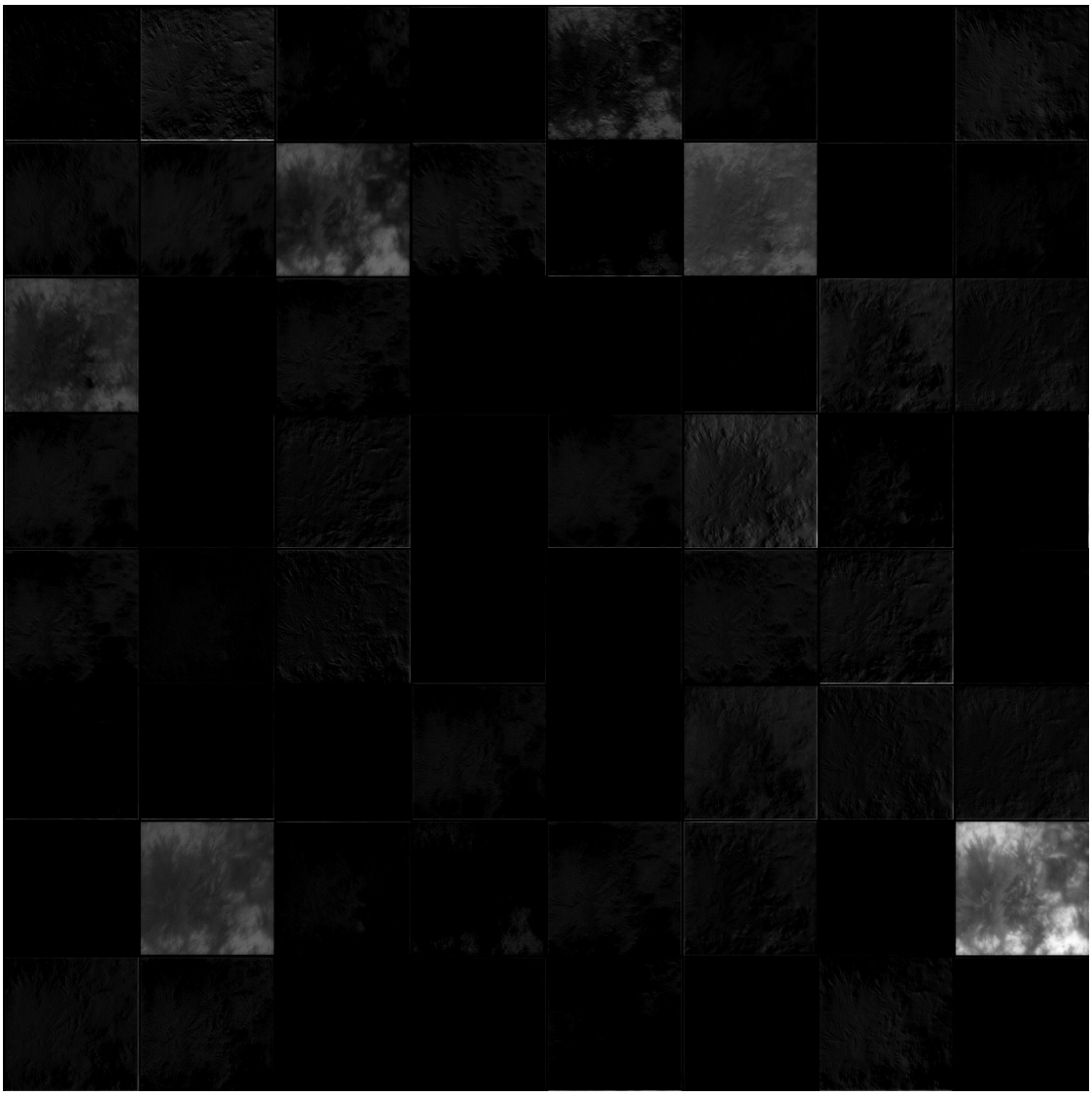
Entrada Saída Ground Truth



Feature maps

Figura 32: Visualização direta dos *feature maps* da primeira camada convolucional da rede multi-escala de restauração de imagens subaquáticas.

Entrada Saída Ground Truth



Feature maps

Figura 33: Visualização direta dos *feature maps* da primeira camada convolucional da rede multi-escala de restauração de imagens subaquáticas.

5.3 RNC Residual de Estimativa de Profundidade

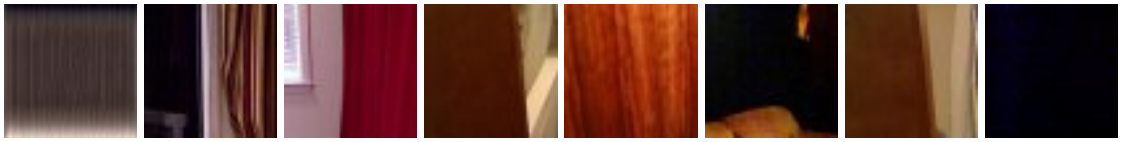
A estimativa do mapa de profundidade está diretamente relacionada com a estimativa do mapa de transmissão, e consequentemente com as operações de *dehazing* e restauração de imagens subaquáticas. Por este motivo, a rede de estimativa de profundidade proposta em [37] é estudada aqui.

Os resultados da maximização da ativação de alguns canais da rede são apresentados na figura 34. Eles mostram diferentes padrões multicoloridos, que se tornam mais complexos e diversificados com o aumento da profundidade na rede. Alguns destes padrões lembram texturas presentes em ambientes internos, como madeira ou tecido, ou até mesmo estruturas mais complexas. Não foram identificadas estruturas normalmente observadas nos resultados da aplicação do método em redes de classificação, como detectores de gatos, cachorros e flores. Isto sugere que o treinamento para estimativa de profundidade foi capaz de sobrescrever os detectores de *features* de alto nível aprendidos durante o pré-treinamento na tarefa de classificação.

A figura 35 mostra as 7 entradas de um conjunto de imagens composto por *patches* extraídos dos *datasets* NYU-Depth [61] e B3DO [31] que produzem as maiores ativações médias em alguns *feature maps* da rede. Em 35a, a maximização da ativação resulta em uma textura que lembra madeira. Como esperado, as entradas que produzem as maiores ativações neste *feature map* contém madeira ou texturas que parecem madeira. Em 35c, a otimização resulta em um grid preto sobre um fundo branco, e a entrada que produz a maior ativação média contém exatamente isto. As entradas que produzem as ativações mais próximas também contêm estruturas similares e fundos predominantemente brancos. Em 35d, a maximização da ativação resulta no que pode ser descrito como uma estante marrom preenchida com livros amarelos, e 5 das 7 entradas que produzem as maiores ativações contêm estantes com livros. Existem casos, porém, onde o resultado da maximização da ativação e as entradas que produzem as maiores ativações não têm quase nada em comum. Um exemplo desta situação é apresentado em 35d, onde a otimização resultou em uma imagem que lembra uma plantação, mas as entradas que produzem as maiores ativações não contém nenhuma planta, o que é esperado, já que o *dataset* NYU-Depth, usado nos experimentos e no treinamento da rede, é composto por cenas internas e praticamente não contém imagens de plantas. Uma possível explicação para isto é que este *feature map* em particular foi herdado do pré-treinamento na tarefa de classificação. Outra possibilidade é que a aparência de planta do resultado da otimização é uma simples coincidência.

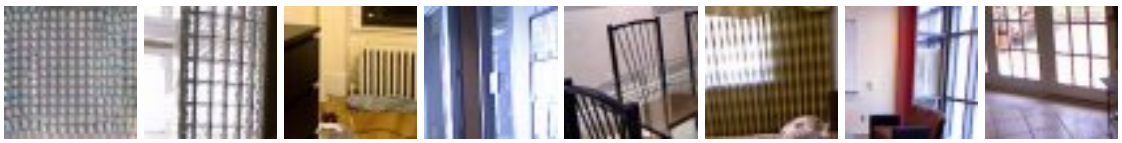
Ainda não está claro como estes detectores de texturas e estruturas ajudam a rede a estimar a profundidade de uma cena. Uma possibilidade é a ocorrência de *overfitting* devido à complexidade da arquitetura ResNet-50 e ao tamanho reduzido do *dataset* NYU-Depth. A rede pode ter aprendido que, no conjunto de treinamento, cada textura ou estrutura

Figura 34: Resultados da maximização da ativação da rede de estimativa de profundidade proposta em [37]. Cada linha mostra alguns canais de uma mesma camada. As camadas são, de cima para baixo: pool1, res2a_relu, res2b_relu, res2c_relu, res3a_relu, res3b_relu, res3d_relu, res4b_relu, res4c_relu, res4d_relu, res4e_relu, res4f_relu, res5a_relu, res5b_relu e res5c_relu. Esta figura é melhor visualizada digitalmente, em cores e com zoom.



(a) camada res2a_relu, canal 224

(b) camada res4d_relu, canal 236



(c) camada res4e_relu, canal 525

(d) camada res4f_relu, canal 650

Figura 35: Entradas que produzem as maiores ativações médias em alguns *feature maps* da rede de estimativa de profundidade proposta em [37]. Em cada linha a imagem mais à esquerda é o resultado da maximização da ativação, as outras imagens são as 7 entradas que produzem as maiores ativações médias.

particular sempre ocorre na mesma profundidade.

Os resultados mostram que alguns *feature maps* permanecem praticamente inalterados através de muitas camadas da rede, como mostrado na figura 36. A provável causa deste fenômeno é a presença de conexões residuais, que são ligações diretas entre canais correspondentes de camadas diferentes. Também existem casos onde um *feature map* "desaparece" (a maximização da ativação do canal resulta em uma imagem completamente preta) em uma camada e reaparece praticamente inalterado na próxima.

Figura 36: Alguns *feature maps* permanecem praticamente inalterados através de muitas camadas. Da esquerda para a direita, resultados da maximização da ativação para o canal 81 das camadas res4a_relu, res4b_relu, res4c_relu, res4d_relu, res4e_relu e res4f_relu da rede de estimativa de profundidade.

Pode-se observar que nas camadas finais o número de canais "mortos" (canais onde a maximização da ativação resulta em uma imagem preta) aumenta drasticamente. Na

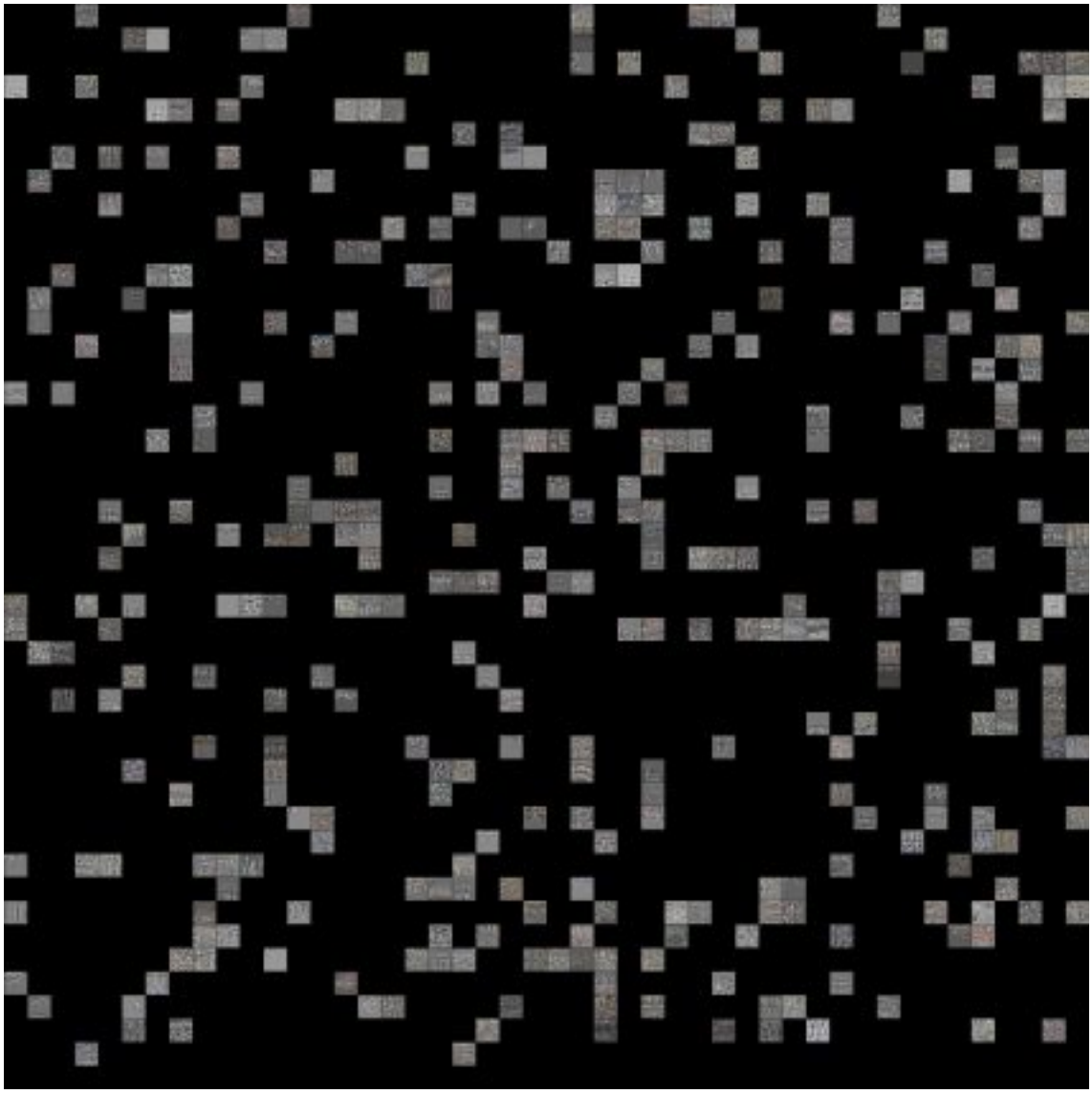


Figura 37: Resultados da maximização da ativação da camada res5c_relu da arquitetura de estimativa de profundidade. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Pode-se observar que em mais da metade dos canais a otimização resulta em imagens completamente pretas. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

camada res5c_relu mais da metade dos *feature maps* estão "mortos", como pode ser visto na Figura 37. Nestes canais o gradiente da ativação média do *feature map* em relação à imagem de entrada fica em zero e a otimização nunca sai da imagem inicial. A razão mais provável para este comportamento é a ausência de conexões ativas entre a imagem de entrada e os *feature maps* em questão. Isto pode ser considerado como um sinal de que o número de *feature maps* nas camadas mais profundas é muito grande para este problema em particular, pelo menos com o conjunto de treinamento utilizado.

A visualização por inversão da rede desta arquitetura não produziu nenhum resultado significativo. Varias combinações de parâmetros foram testadas, mas nenhuma delas foi

capaz de produzir imagens com características reconhecíveis ou com mapas de profundidade parecidos com o da imagem alvo. Foram realizados vários testes, utilizando como imagem alvo diferentes mapas de profundidade obtidos através da aplicação da rede neural a imagens de ambientes internos extraídas do *dataset* NYU-Depth. Em todos os casos, o processo de otimização resultou em imagens sem estrutura global, compostas por padrões em forma de onda. A estimativa do mapa de profundidade destas imagens normalmente é uma imagem sólida, com um único valor uniforme, que não lembra em nada o mapa de profundidade utilizado como imagem alvo. Estes resultados não fornecem nenhuma informação que pode ser utilizada para compreender como a rede realiza a estimativa do mapa de profundidade, e, por isto, não são apresentados neste trabalho. Uma possível explicação para o comportamento apresentado é que, devido ao tamanho da arquitetura da rede e à complexidade do problema, a otimização sempre acaba convergindo para um mínimo local "ruim".

5.4 Dehaze Resnet 12

Os resultados da aplicação da maximização da ativação na arquitetura Dehaze Resnet 12 são apresentados nas figuras 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49 e 50. Em comparação com redes de classificação, os resultados da otimização apresentam menor complexidade e diversidade. Os resultados das três primeiras camadas mostram uma boa diversidade de cores, porém com o aumento da profundidade todos os *feature maps* começam a tomar a mesma forma: padrões de cor ciano claro com formas quadradas ou retangulares sobre um fundo cinza. Esta uniformidade sugere que todos estes *feature maps* são redundantes, e as formas retangulares sugerem a presença de detectores de bordas. A visualização direta das entradas que produzem as maiores ativações médias nestes *feature maps* parece confirmar esta teoria. As camadas iniciais apresentam boa diversidade, com alguns *feature maps* sensíveis a cores específicas, alguns sensíveis a bordas horizontais, e outros sensíveis a bordas verticais. Nas camadas mais profundas, porém, a grande maioria dos *feature maps* são, aparentemente, detectores de bordas. Para quase todos os canais da camada residual12 as imagens que produzem as maiores ativações médias são as mesmas. As características em comum entre estas imagens são a predominância da cor branca e a presença de muitas bordas. As ativações produzidas por estas imagens nos *feature maps* da última camada oculta destacam as suas bordas, principalmente as verticais. As entradas que produzem as maiores ativações médias em alguns *feature maps* de diferentes camadas são apresentadas na Figura 51.

Na Figura 52 é apresentada a visualização direta das ativações dos canais 20, 32, 40 e 58 da última camada oculta da Dehaze Resnet 12 para uma imagem com névoa real. Apesar de estes canais apresentarem as maiores ativações com os mesmos *patches* do conjunto de treinamento e de seus resultados de maximização da ativação serem muito

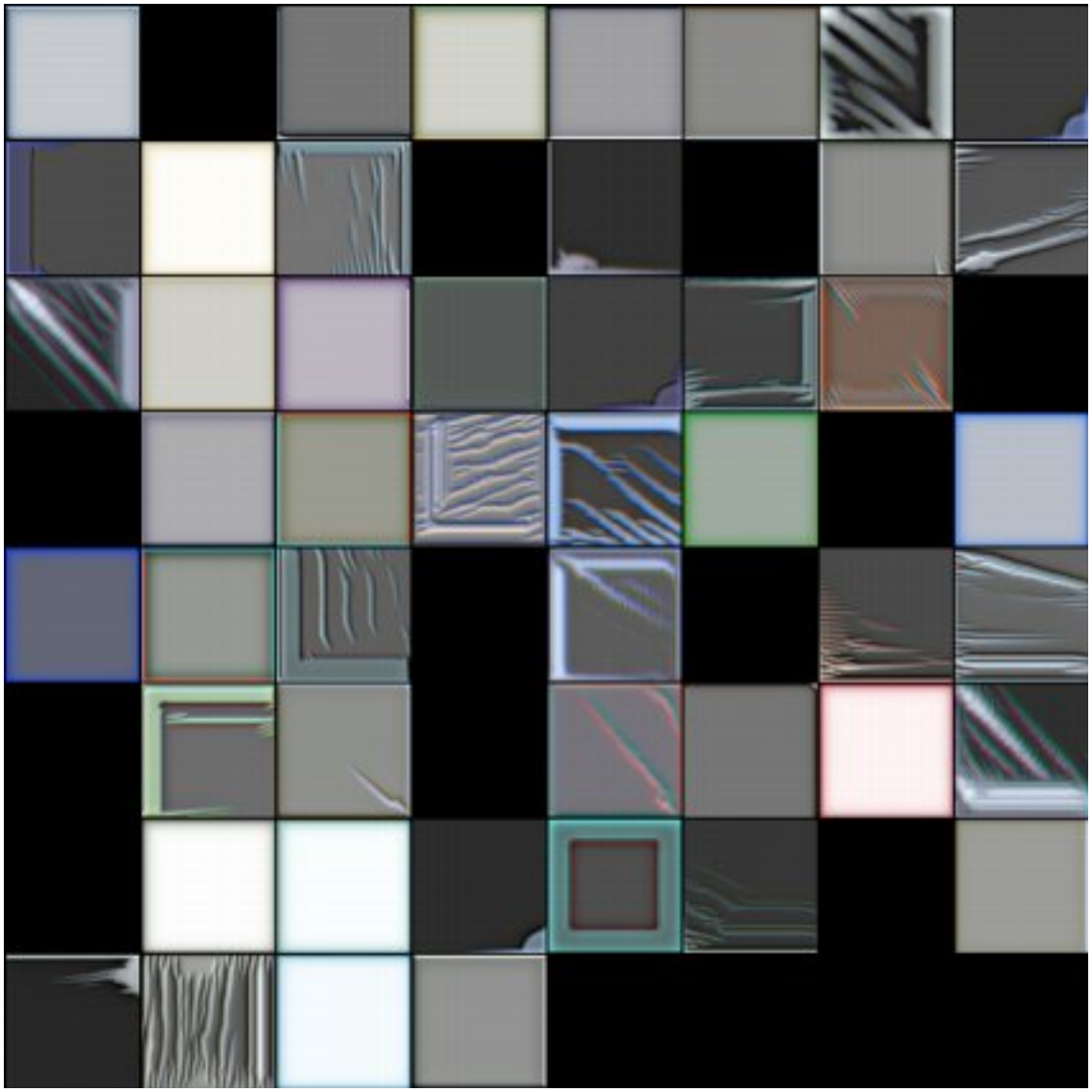


Figura 38: Resultados da maximização da ativação da camada conv1 da arquitetura Dehaze Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

parecidos, neste caso as suas ativações são muito diferentes. As ativações do canal 58 coincidem perfeitamente com as bordas da imagem, e estão concentradas na região mais próxima da câmera, onde existe menos névoa e o contraste é maior. As ativações do canal 20, por outro lado, estão concentradas na região onde existe uma maior intensidade de névoa, mas não incluem a região do céu. As ativações do canal 32 parecem representar a intensidade do componente de *backscatter*, incluindo as regiões de céu. O canal 40 parece detectar a cor branca, mas também é sensível a bordas. Estes resultados mostram que a redundância nas últimas camadas pode não ser tão grande quanto a maximização da ativação e a visualização das entradas que produzem as maiores ativações podem sugerir, e que *feature maps* que apresentam ativações muito parecidas em alguns casos podem

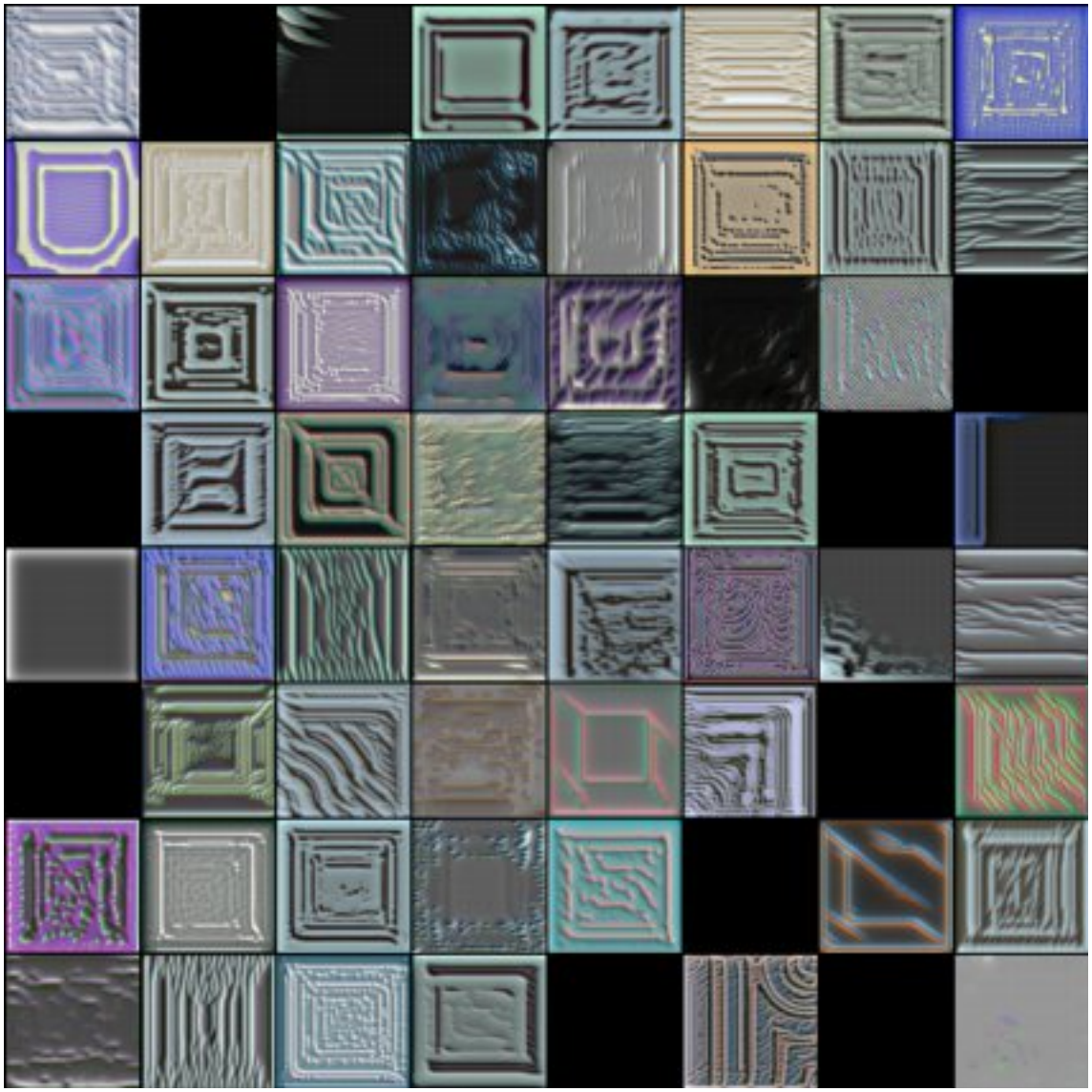


Figura 39: Resultados da maximização da ativação da camada residual1 da arquitetura Dehaze Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

apresentar ativações completamente diferentes em outros.

Uma possível explicação para a presença de tantos detectores de bordas na rede é que eles sejam utilizados como estimativa de contraste. Se este for o caso, a rede pode estar utilizando o contraste local pra estimar a transmissão, como ocorre em [68], e utilizando esta estimativa para determinar o quão vivas devem ser as cores na imagem de saída. Esta teoria explica as manchas presentes nos resultados apresentados na Figura 26, pois variações no contraste local podem produzir diferentes estimativas de transmissão para pontos que estão na mesma profundidade e possuem a mesma cor.

Apesar de ter sido treinada com um conjunto de treinamento semelhante ao da rede de estimativa de mapa de profundidade apresentada em [37], a Dehaze Resnet 12 apresenta

Figura 40: Resultados da maximização da ativação da camada residual2 da arquitetura Dehaze Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

muito menos diversidade e complexidade em seus *feature maps*. Uma possível explicação para isto é que o treinamento da Dehaze Resnet 12 foi inicializado com pesos aleatórios, enquanto a rede de estimativa de profundidade foi inicializada com os pesos de uma rede de classificação treinada com o conjunto de dados do ILSVRC [56]. Isto sugere que um pré-treinamento com um conjunto de dados maior, mesmo que em um problema diferente, pode ajudar a rede a desenvolver estratégias alternativas para resolver um mesmo problema, o que a torna mais robusta.

Inicialmente, a complexidade dos *feature maps* presentes na rede aumenta de acordo com a profundidade da camada, porém a partir de um certo ponto ocorre uma estagnação. Isto sugere que a adição de mais camadas à rede não traria nenhum benefício. Ainda, estes

Figura 41: Resultados da maximização da ativação da camada residual3 da arquitetura Dehaze Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

resultados sugerem que talvez seja possível remover algumas camadas sem nenhum impacto negativo na qualidade da restauração. Nas camadas finais, muitos dos *feature maps* aparentemente possuem a mesma função. Esta redundância, teoricamente, torna a rede menos eficiente sem proporcionar nenhum benefício à qualidade dos resultados. Desta forma, a redução do número de canais nas camadas finais da arquitetura aparentemente é uma alternativa viável para a redução do tempo de teste que não prejudica a qualidade dos resultados.

Os resultados da visualização por inversão da rede são apresentados na Figura 53. Mesmo sem nenhum tipo de regularização, a inversão da rede produziu resultados que lembram muito uma versão com névoa da imagem alvo. O nível de ruído produzido nos

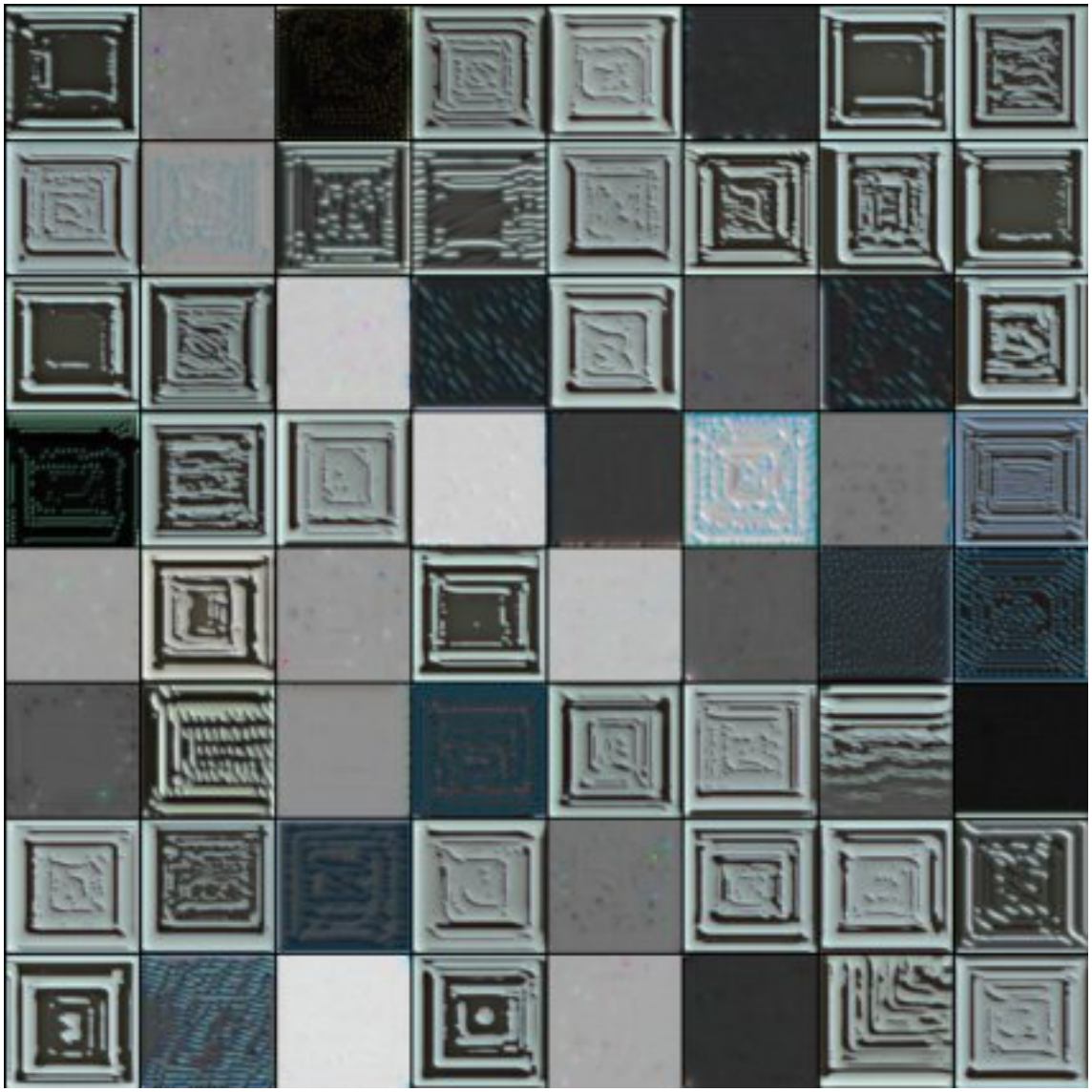


Figura 42: Resultados da maximização da ativação da camada residual4 da arquitetura Dehaze Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

resultados da otimização pode ser considerado baixo, e deve-se em parte ao ruído presente nas imagens originais. Todos os *patches* utilizados como imagem alvo pertencem ao conjunto de validação, e logo não foram utilizados no treinamento da rede. A utilização de *patches* do conjunto de treinamento tende a produzir resultados ainda melhores. Na maioria dos casos apresentados a restauração do resultado da otimização é praticamente impossível de se diferenciar da imagem alvo. Em geral os resultados da otimização apresentam baixo contraste, enquanto a restauração destas imagens é praticamente perfeita, o que indica que a rede pode restaurar imagens com grande quantidade de turbidez, pelo menos quando estas imagens são parecidas com as utilizadas durante o treinamento. Os resultados da otimização possuem coloração acinzentada, mesmo nas regiões onde a ima-

Figura 43: Resultados da maximização da ativação da camada residual5 da arquitetura Dehaze Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

gem alvo é branca. Isso indica que a rede espera névoa de cor cinza escura. Logo, pode-se esperar que a restauração não apresente bons resultados quando a turbidez possui uma cor diferente, como branco. Uma possível solução para este problema seria treinar a rede com névoa mais clara ou com níveis de turbidez mais baixos.

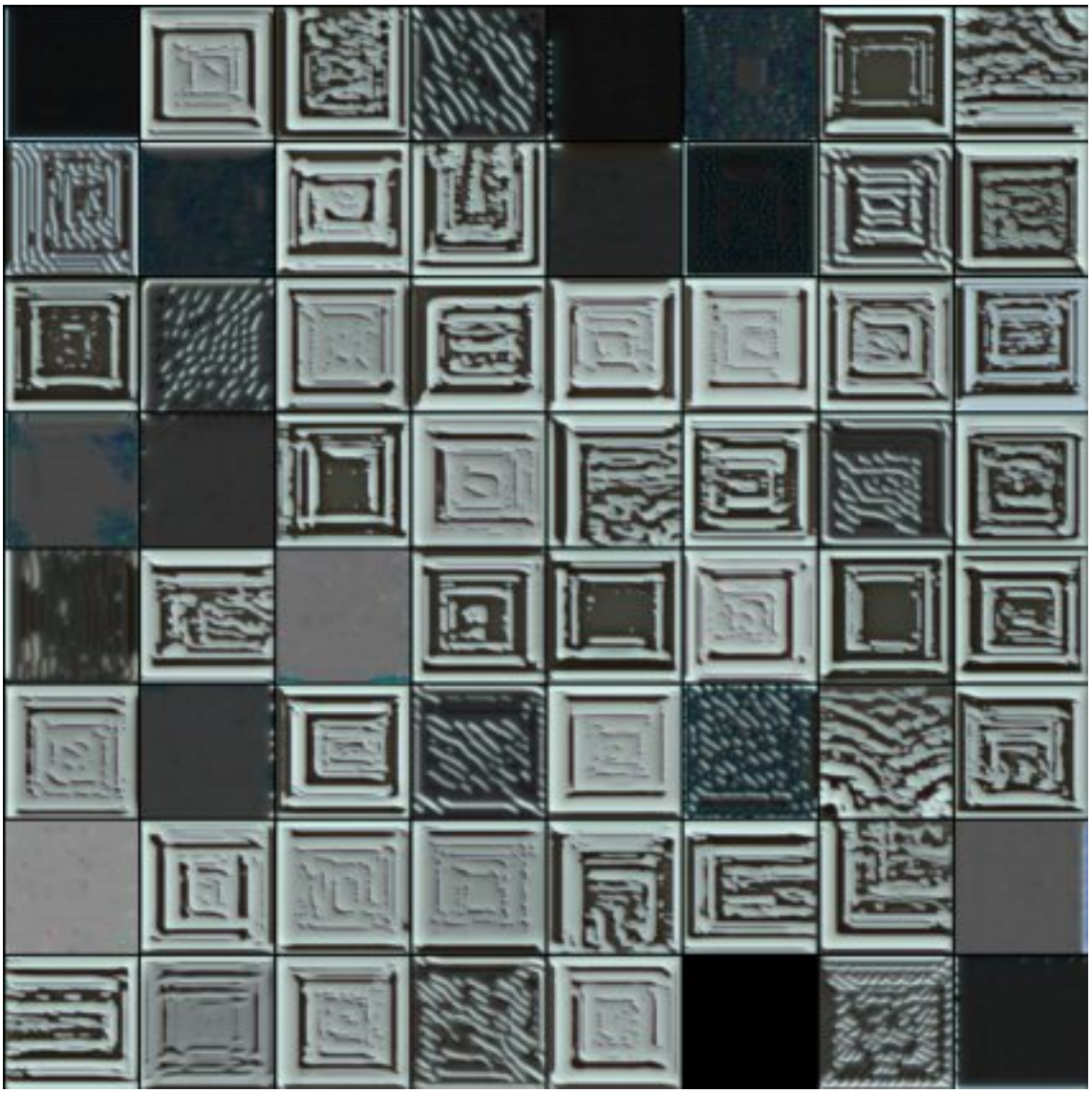


Figura 44: Resultados da maximização da ativação da camada residual6 da arquitetura Dehaze Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

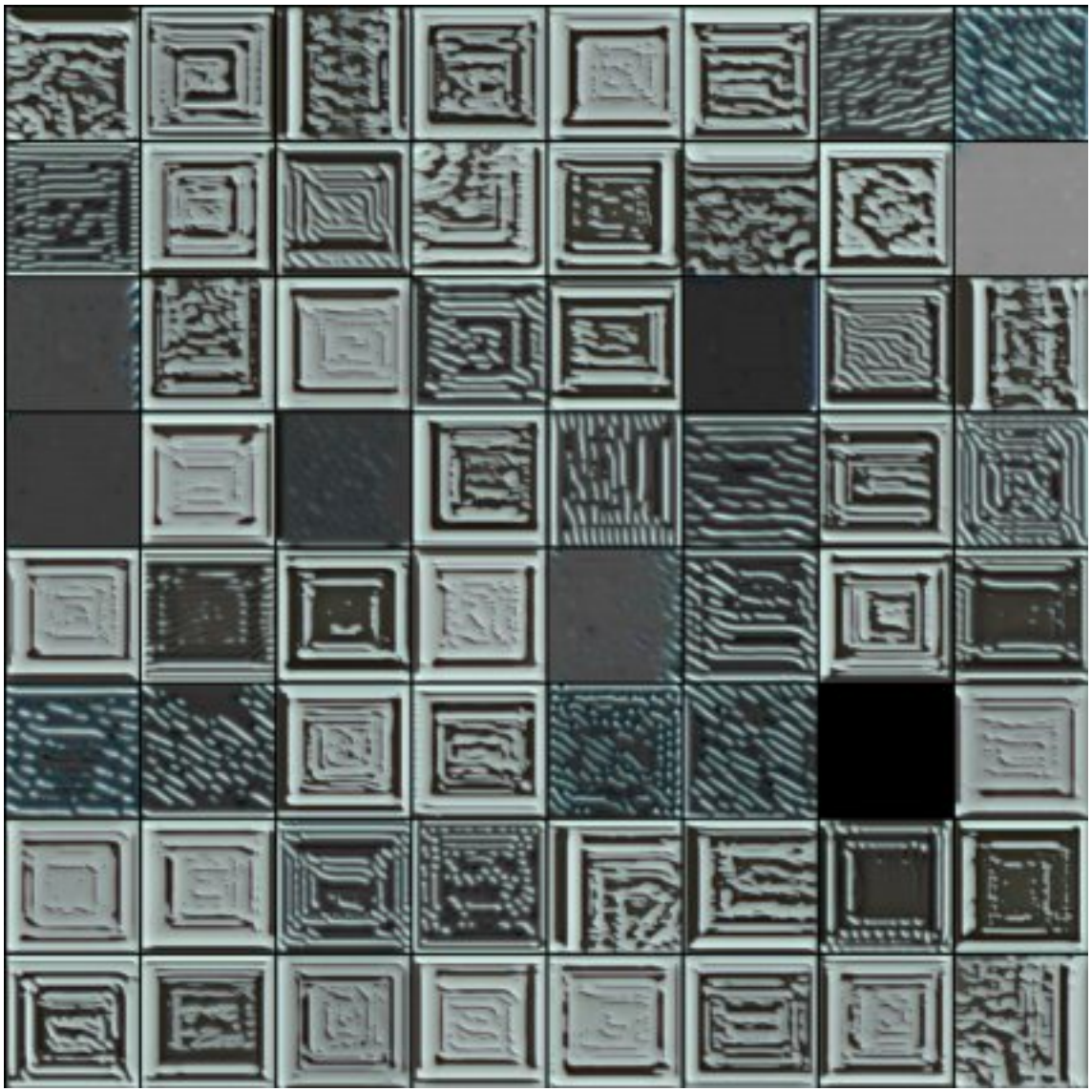


Figura 45: Resultados da maximização da ativação da camada residual7 da arquitetura Dehaze Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

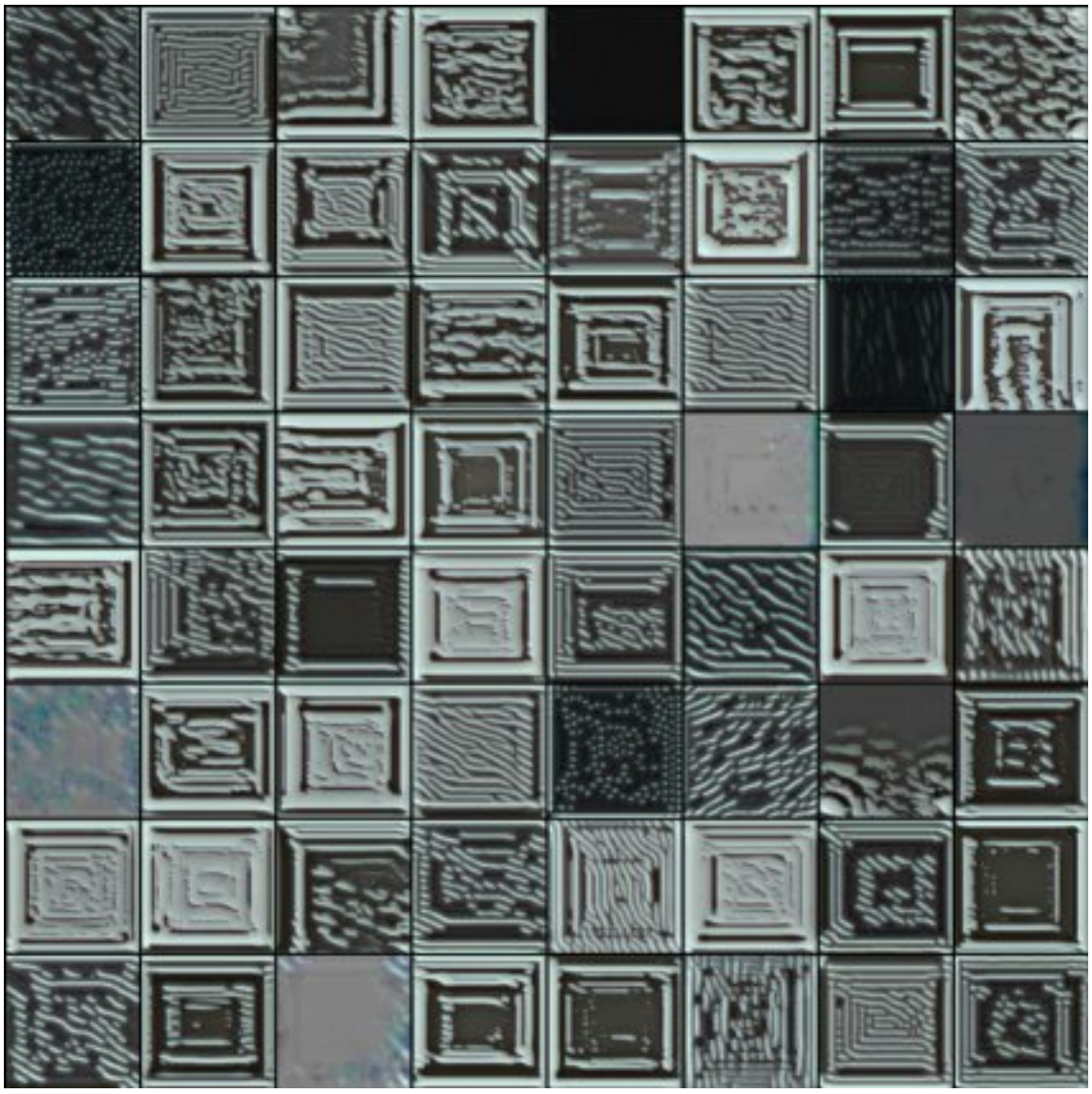


Figura 46: Resultados da maximização da ativação da camada residual8 da arquitetura Dehaze Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

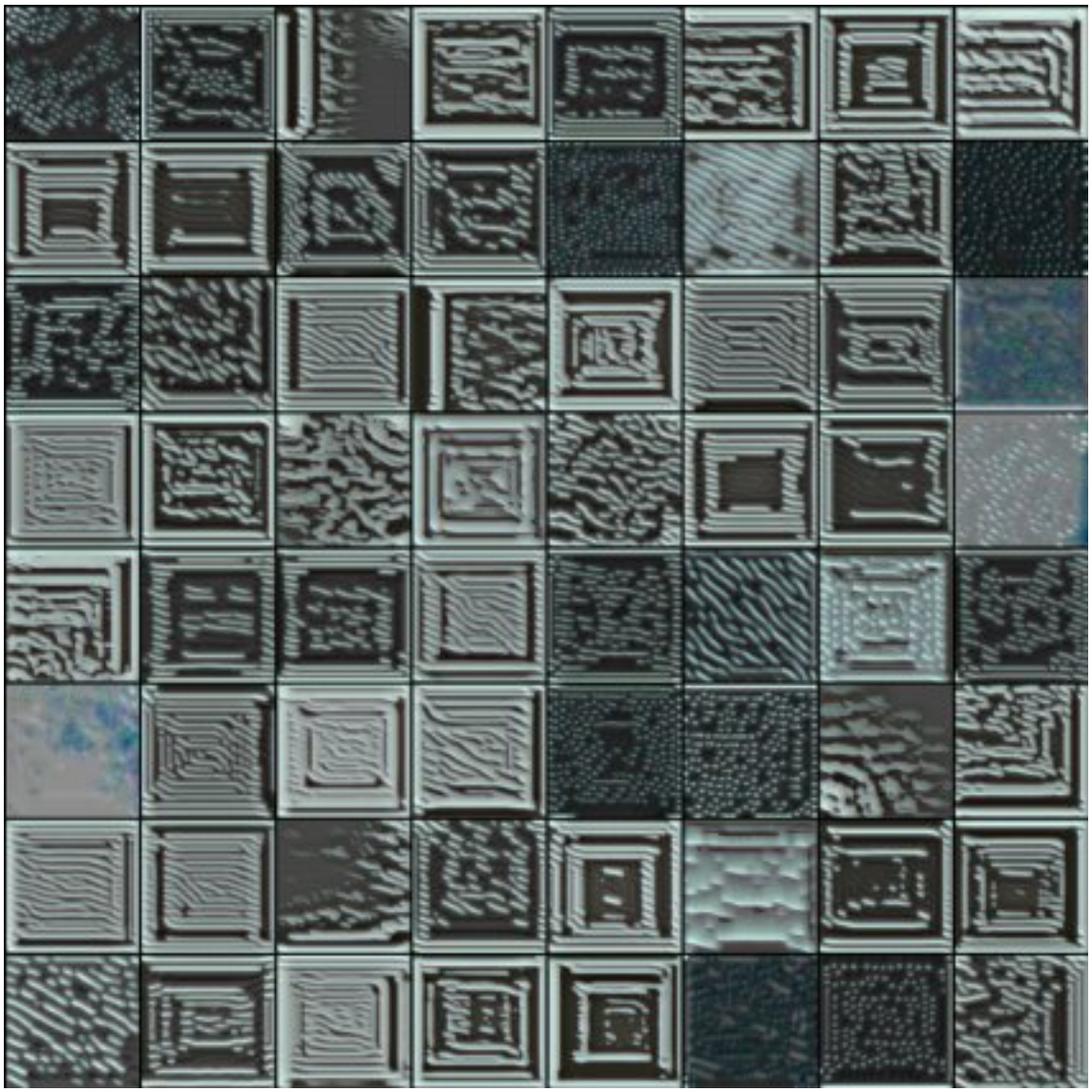


Figura 47: Resultados da maximização da ativação da camada residual9 da arquitetura Dehaze Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

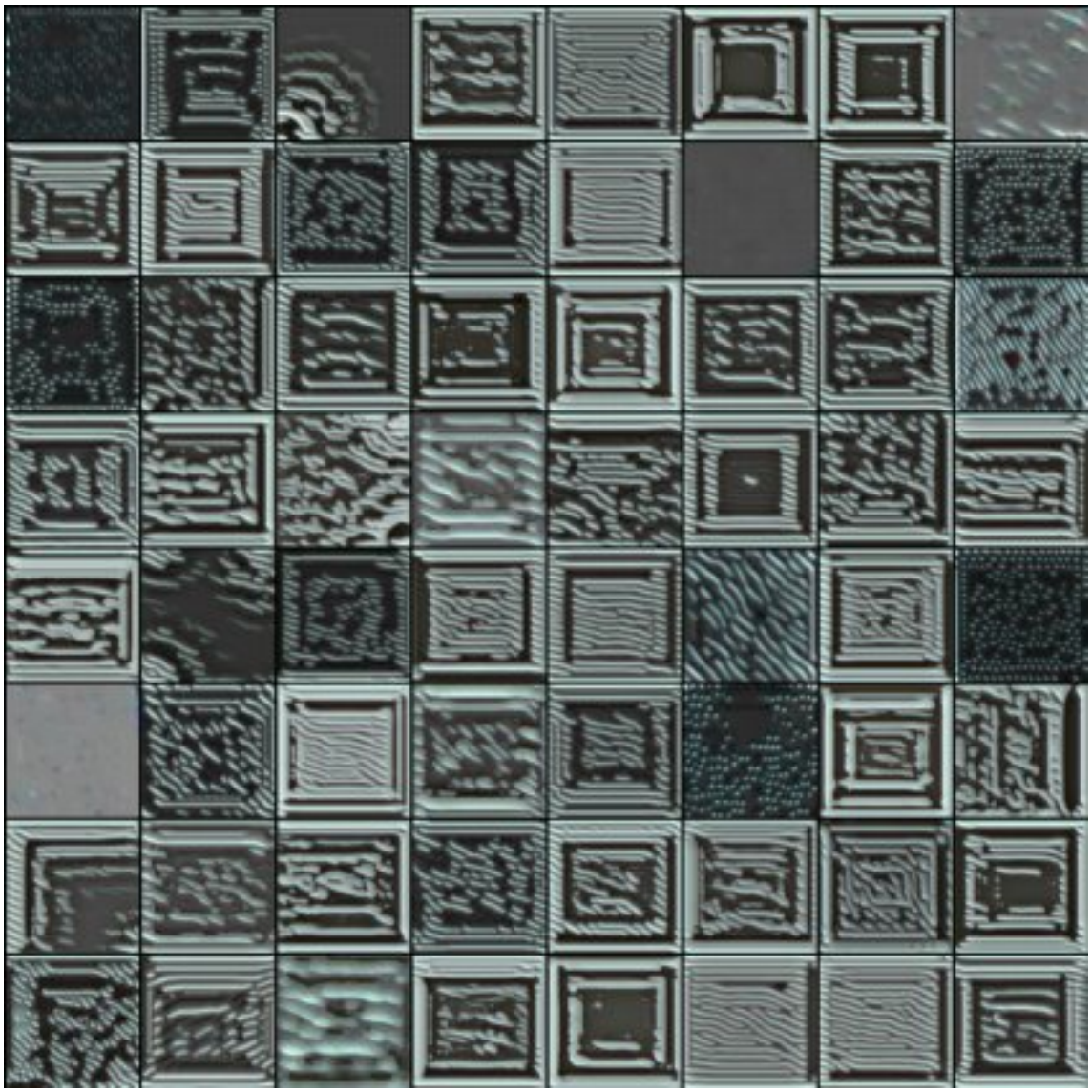


Figura 48: Resultados da maximização da ativação da camada residual10 da arquitetura Dehaze Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

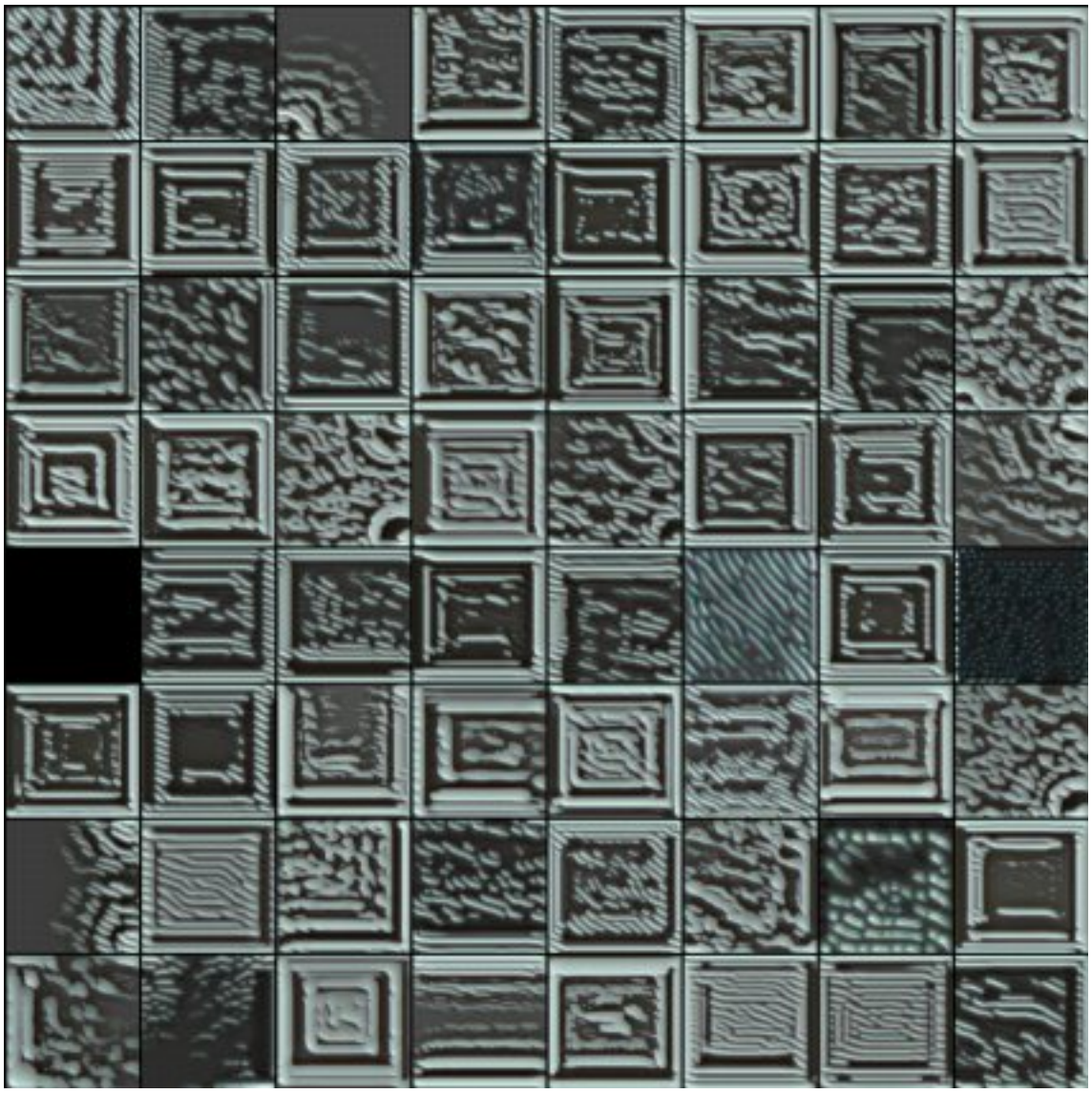


Figura 49: Resultados da maximização da ativação da camada residual11 da arquitetura Dehaze Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

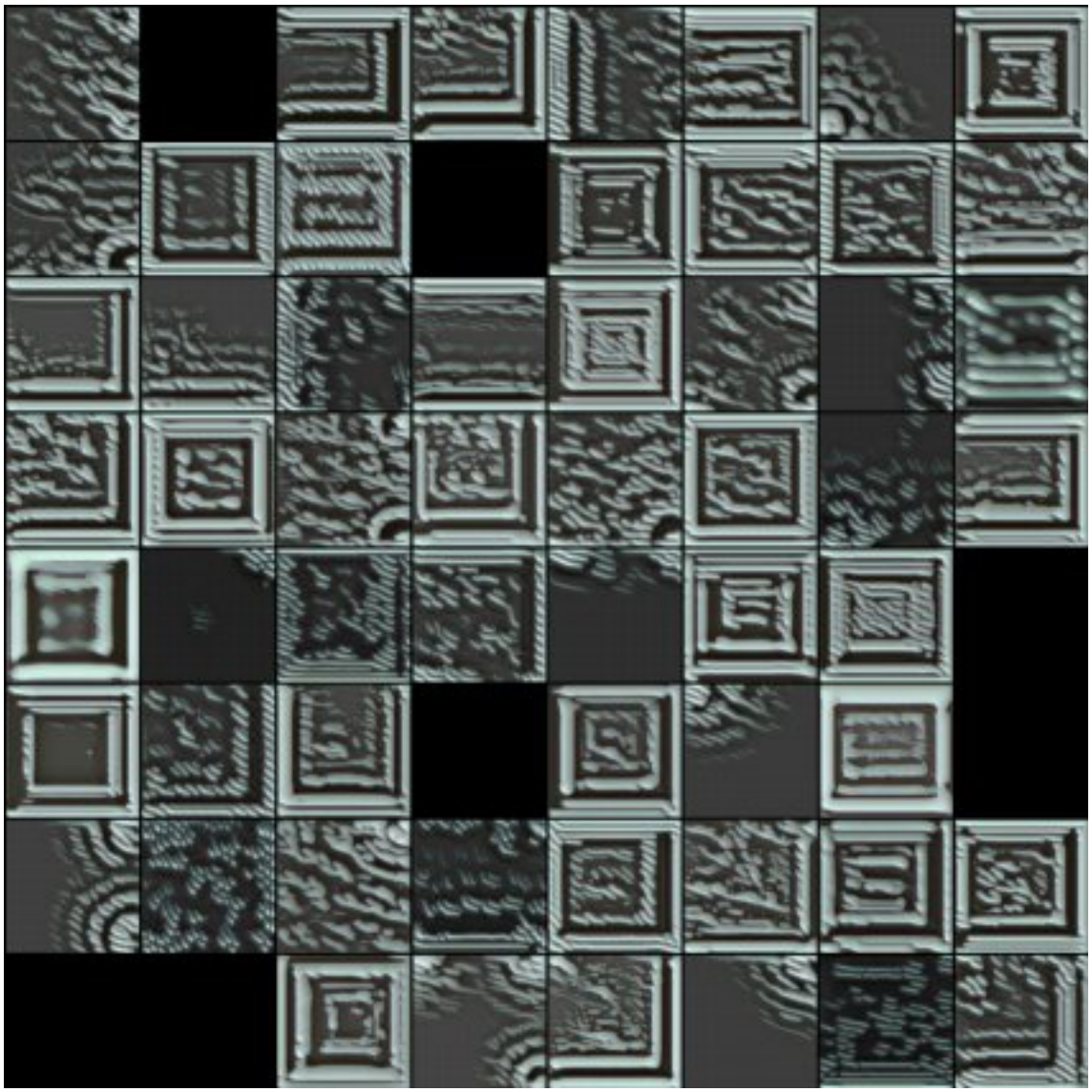
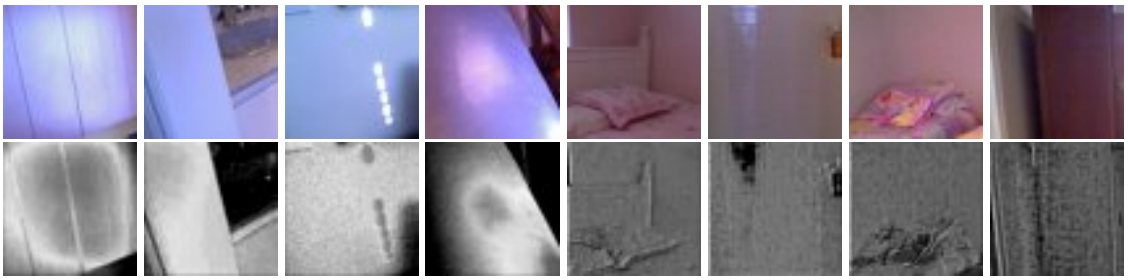
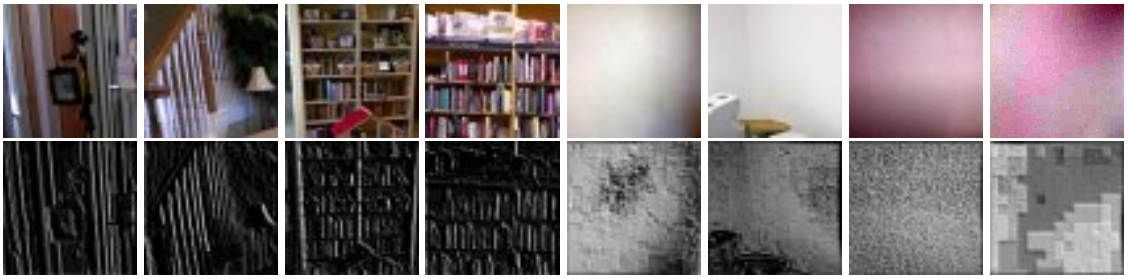


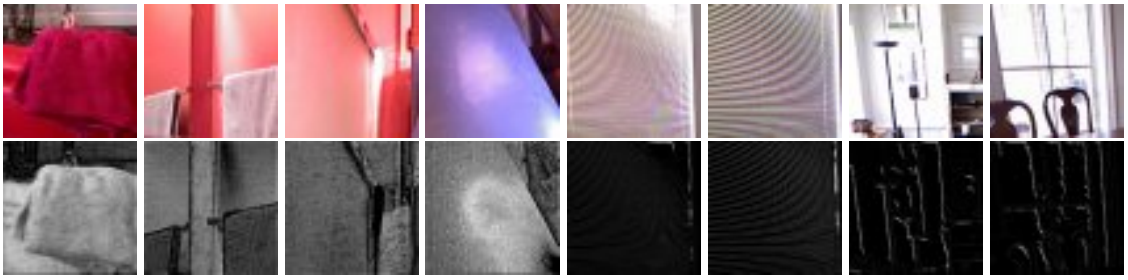
Figura 50: Resultados da maximização da ativação da camada residual12 da arquitetura Dehaze Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.



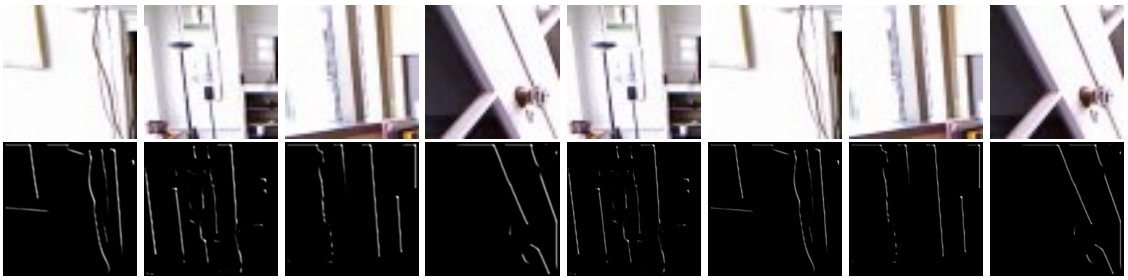
(a) camada conv1, canal 07 (b) camada residual1, canal 12



(c) camada residual2, canal 14 (d) camada residual3, canal 06



(e) camada residual6, canal 48 (f) camada residual8, canal 37



(g) camada residual10, canal 02 (h) camada residual12, canal 63

Figura 51: Visualização de alguns *feature maps* da rede Dehaze Resnet 12. Acima, os *patches* do conjunto de treinamento que produzem as maiores ativações médias no *feature map*. Abaixo, as respectivas ativações. Esta figura é melhor visualizada em cores.

(a) entrada

(b) resultado da restauração

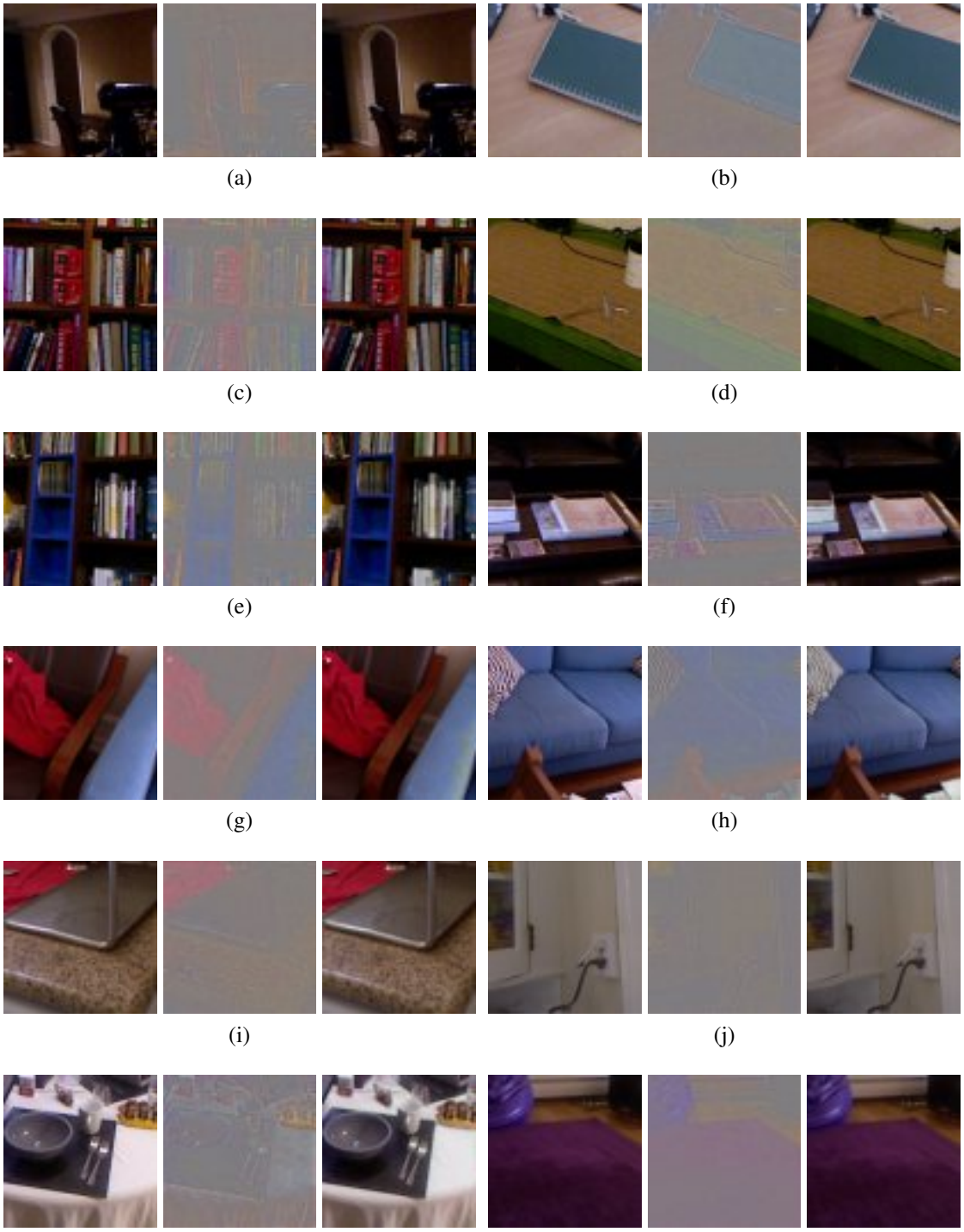
(c) ativação no canal 20 da camada residual12

(d) ativação no canal 32 da camada residual12

(e) ativação no canal 40 da camada residual12

(f) ativação no canal 58 da camada residual12

Figura 52: Visualização direta das ativações produzidas por uma entrada em alguns *feature maps* da última camada da rede Dehaze Resnet 12.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Figura 53: Resultados da visualização por inversão da rede para a arquitetura Dehaze Resnet 12. Todos os resultados foram gerados com inicialização em imagem cinza, passo de otimização de 10^{-4} e 50 mil iterações. Não foi utilizado nenhum tipo de regularização. Em cada item são apresentados, da esquerda para direita, a imagem alvo, o resultado da otimização e a saída da rede para o resultado da otimização. Esta figura é melhor visualizada em cores.

5.5 Underwater Resnet 12

Os resultados da aplicação da maximização da ativação na arquitetura Underwater Resnet 12 são apresentados nas figuras 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65 e 66. Assim como na Dehaze Resnet 12, os resultados são padrões coloridos, com formas abstratas, que não lembram nenhuma estrutura ou textura do mundo real. Na primeira camada convolucional (Figura 54), os resultados são, em geral, bastante simples. Para a maioria dos canais, a otimização resultou em uma imagem preta ou cinza escura no centro e com cores vivas nas bordas. A partir da camada residual1 (Figura 55) os resultados se tornam mais complexos, apresentando diferentes cores e formas, incluindo padrões retangulares e texturas que lembram ondas. As camadas seguintes não apresentam nenhum aumento significativo na complexidade dos resultados. A diversidade, porém, vai caindo com o aumento da profundidade das camadas na rede, até que nas camadas finais praticamente todos os canais apresentam resultados com cores e formas semelhantes, assim como ocorre na Dehaze Resnet 12. Em comparação com a última camada oculta da Dehaze Resnet 12 (Figura 50), os resultados da última camada oculta de Underwater Resnet 12 (Figura 66) apresentam menos padrões retangulares e mais padrões em forma de onda. Uma outra diferença está na cor dos resultados. Na rede de *dehazing*, os padrões apresentados têm uma cor ciano claro, quase branco, enquanto na Underwater Resnet 12 eles são azuis. Esta diferença provavelmente está relacionada com a cor da turbidez simulada utilizada durante o treinamento.

Os *paches* de treinamento com turbidez simulada que produzem as maiores ativações em alguns *feature maps* da última camada oculta da Underwater Resnet 12 são apresentados na Figura 67. Para a maioria dos canais, os resultados são muito parecidos com os do canal 10 (67b), que aparentemente se especializou em detectar um tom específico de azul. Outros canais (67c, 67e) parecem ter aprendido a detectar outras cores, também relacionadas com a presença de turbidez. Alguns canais, como o canal 37 (67d), parecem ser detectores de bordas que só funcionam em alguns tons de azul. As maiores ativações do canal 07 (67a) apresentam a cor roxa, mas o que está sendo detectado, na verdade, é a intensidade do canal de cor vermelho, como mostrado na Figura 68. Segundo [8], em ambientes subaquáticos uma alta intensidade no canal vermelho indica uma alta transmissão. Uma possível interpretação para estes resultados é que cada *feature map* da camada residual12 estima o mapa de transmissão utilizando uma estratégia deferente, e que todas estas estimativas são combinadas na camada de saída para dar origem à imagem restaurada.

As ativações produzidas por uma imagem subaquática real em alguns canais da última camada oculta da Underwater Resnet 12 são apresentadas na Figura 69. As ativações do canal 10 parecem coincidir com as regiões da imagem onde existem cores associadas à presença de turbidez. Isto corresponde aos resultados da visualização dos *patches* de

entrada que produzem as maiores ativações no canal 10, apresentados na Figura 67b, que mostram que o *feature map* é sensível à cor azul. As ativações do canal 15, por outro lado, estão concentradas nas regiões da imagem onde a turbidez é menor. As ativações do canal 37 destacam as regiões da imagem onde existe contraste entre cores claras e cores escuras, o que está de acordo com os resultados apresentados na Figura 67d. As ativações do canal 40 parecem confirmar que ele se trata de um detector de bordas.

Alguns resultados da visualização por inversão da rede para a Underwater Resnet 12 são apresentados na Figura 70. Para reduzir o nível de ruído, foi utilizada regularização de gradiente por pirâmide Laplaciana. A utilização de regularização por filtro gaussiano não foi necessária. Todos os *patches* utilizados como imagem alvo pertencem ao conjunto de validação, e logo não foram utilizados no treinamento da rede. Na maioria dos casos, a otimização resultou em imagens que lembram uma versão esverdeada da imagem alvo, com contraste reduzido. Esta coloração verde é inesperada, já que a rede foi treinada apenas com turbidez simulada de cor azul. Os resultados também apresentam, em média, uma turbidez muito menor que a aplicada às imagens durante o treinamento. A razão disto é que a otimização procura imagens cuja saída seja idêntica à imagem alvo, o que só é possível em condições de baixa turbidez, já que quando a turbidez é muito alta a imagem é degradada de uma forma que algumas características, como a intensidade do canal de cor vermelho, não podem ser recuperadas. Em geral, os resultados da visualização por inversão da sugerem que a rede funciona melhor em condições de baixa turbidez.

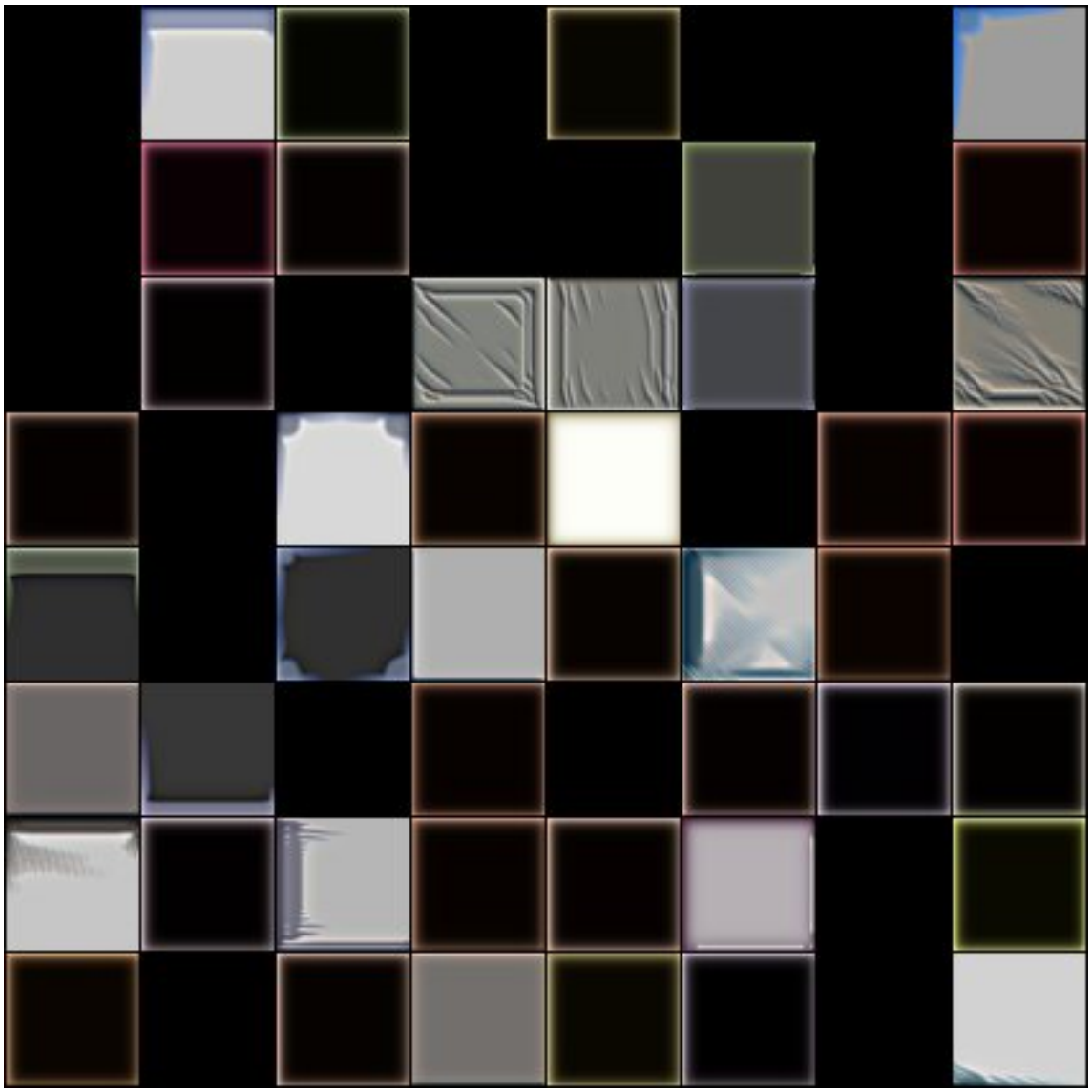


Figura 54: Resultados da maximização da ativação da camada conv1 da arquitetura Underwater Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

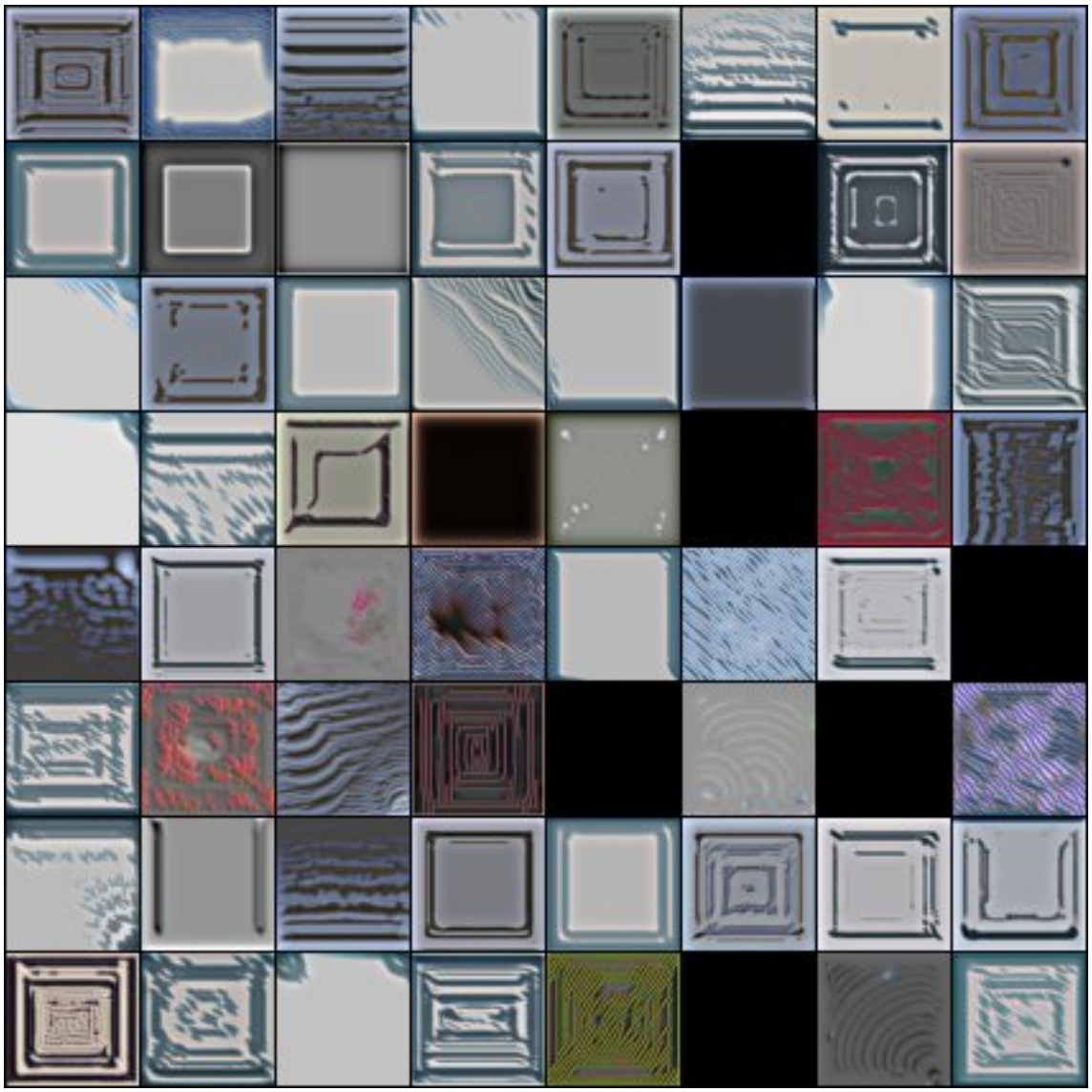


Figura 55: Resultados da maximização da ativação da camada residual1 da arquitetura Underwater Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

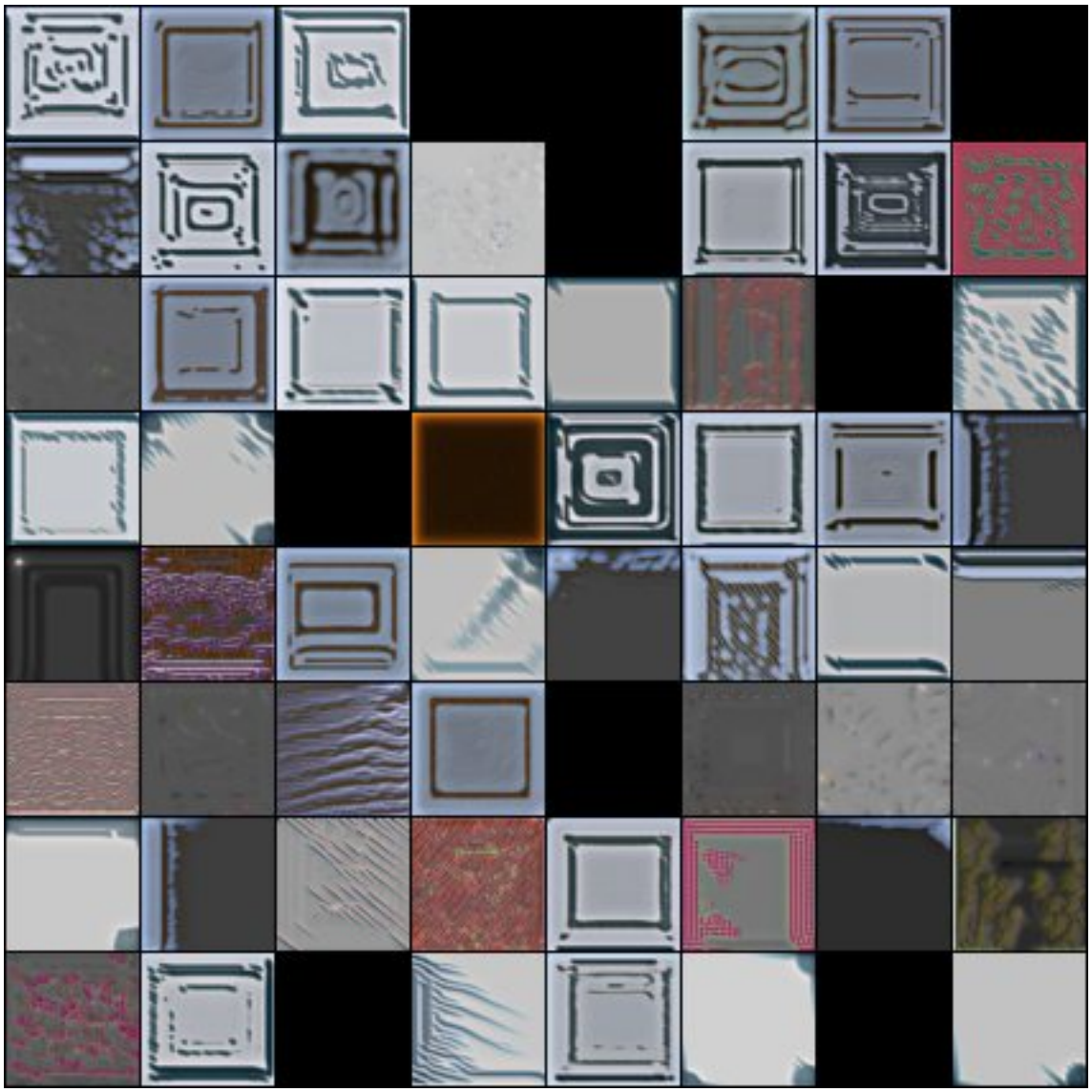


Figura 56: Resultados da maximização da ativação da camada residual2 da arquitetura Underwater Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

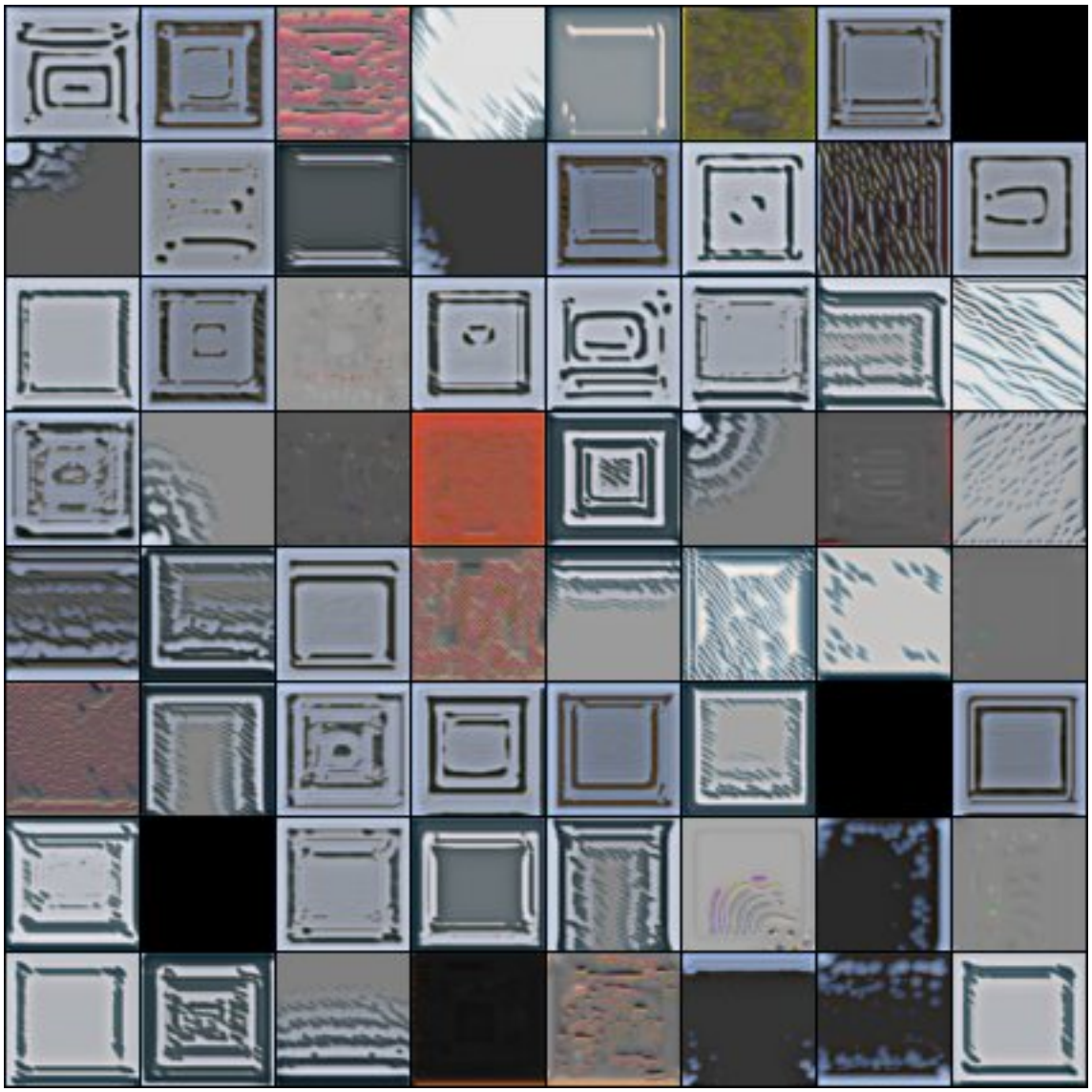


Figura 57: Resultados da maximização da ativação da camada residual3 da arquitetura Underwater Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

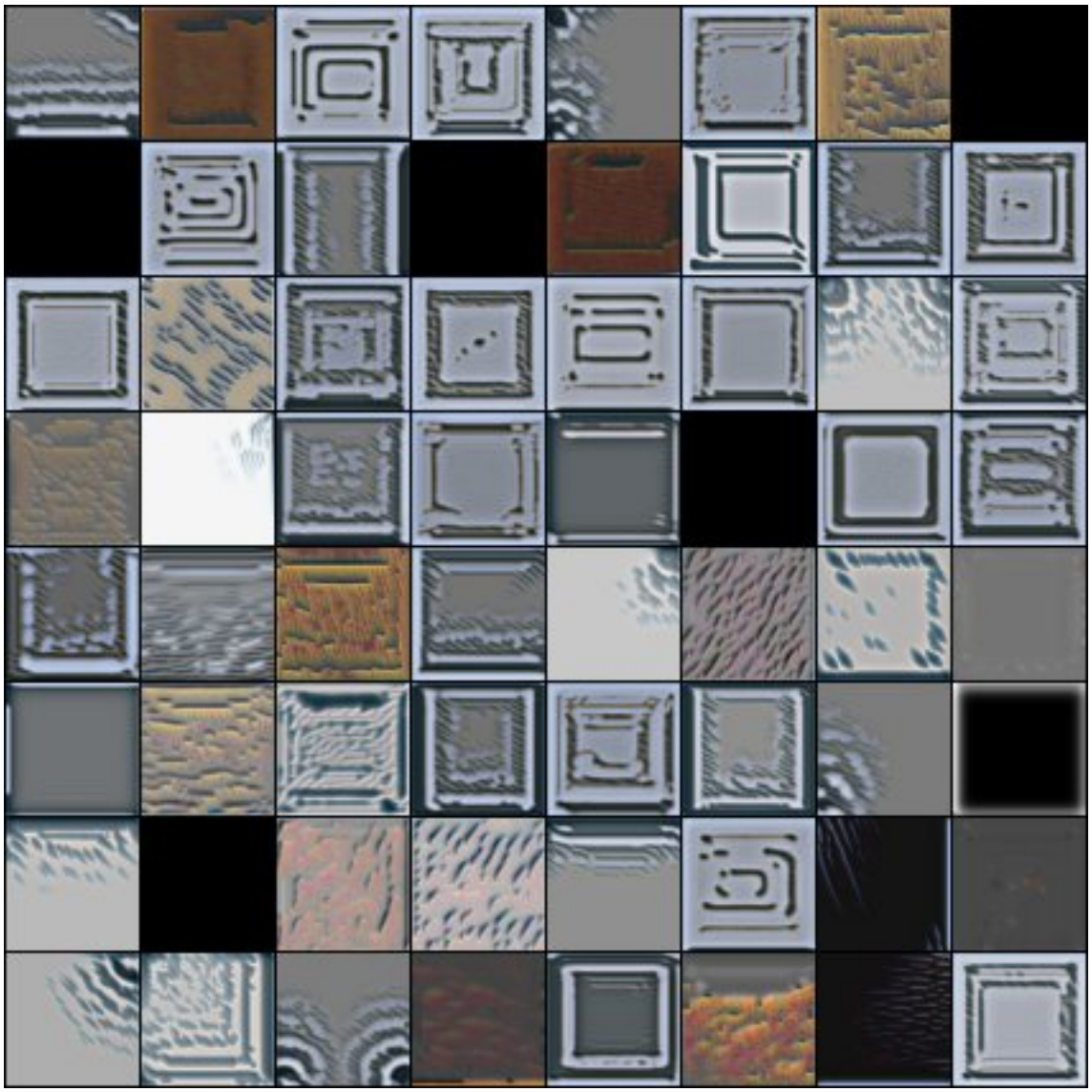


Figura 58: Resultados da maximização da ativação da camada residual4 da arquitetura Underwater Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

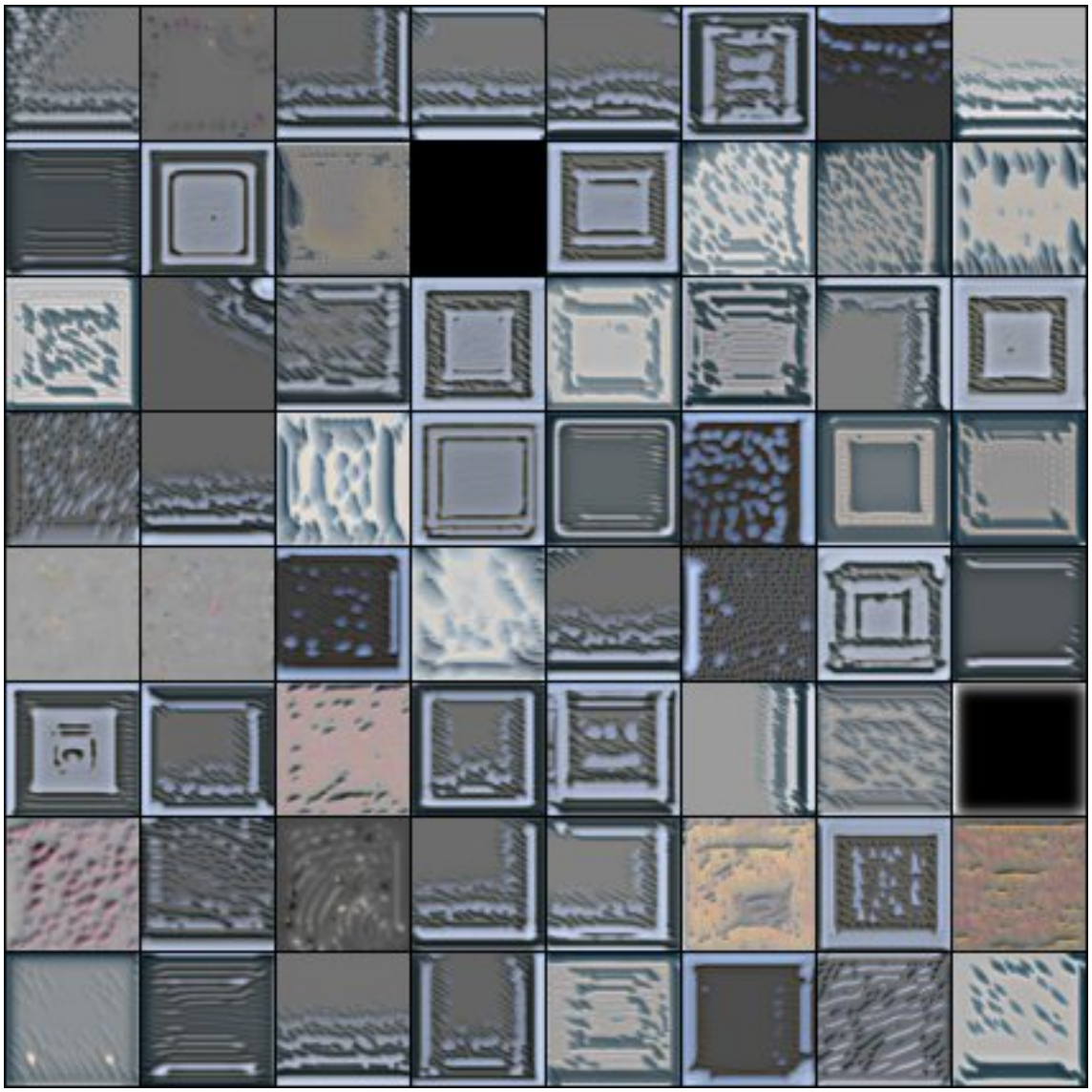


Figura 59: Resultados da maximização da ativação da camada residual5 da arquitetura Underwater Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

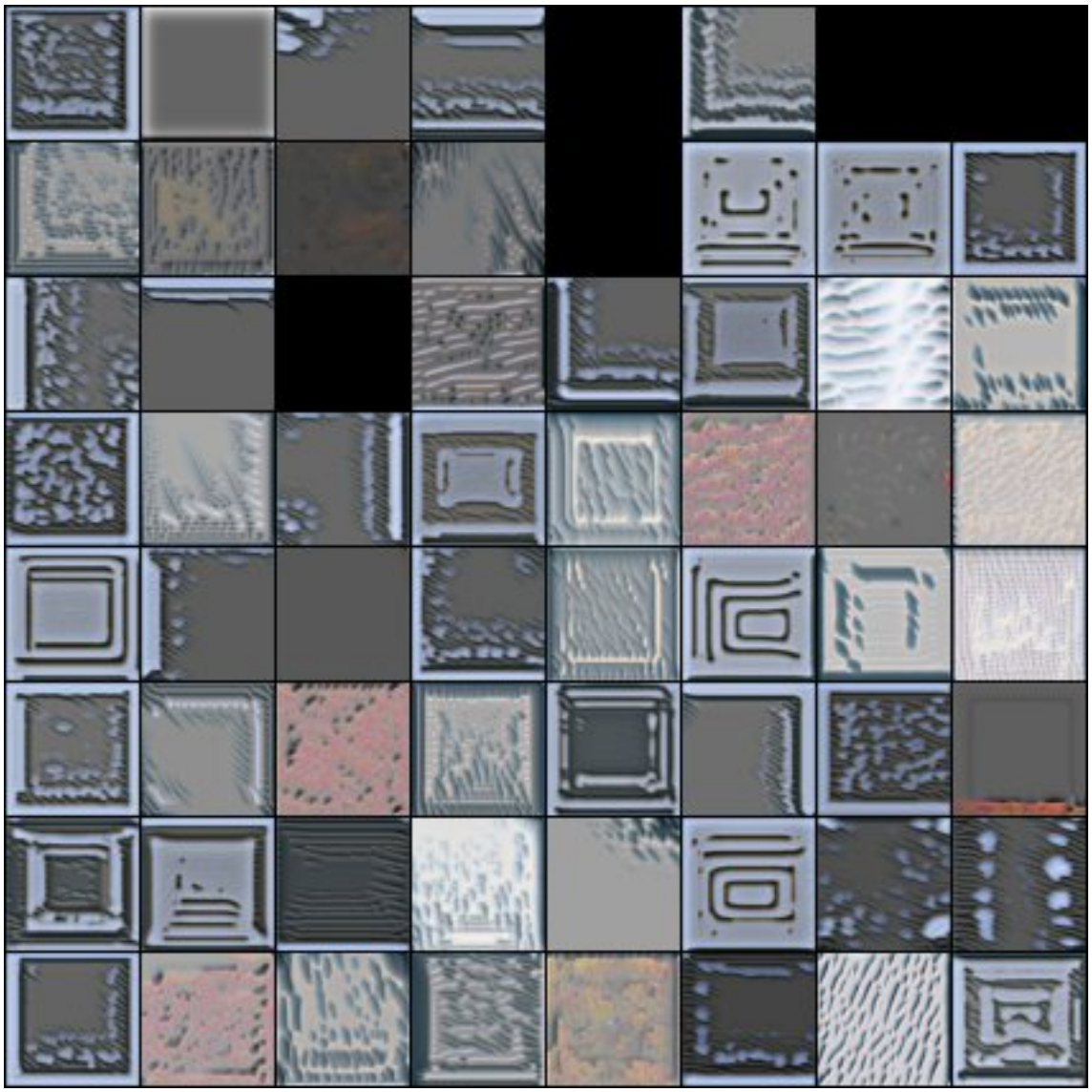


Figura 60: Resultados da maximização da ativação da camada residual6 da arquitetura Underwater Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

Figura 61: Resultados da maximização da ativação da camada residual7 da arquitetura Underwater Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

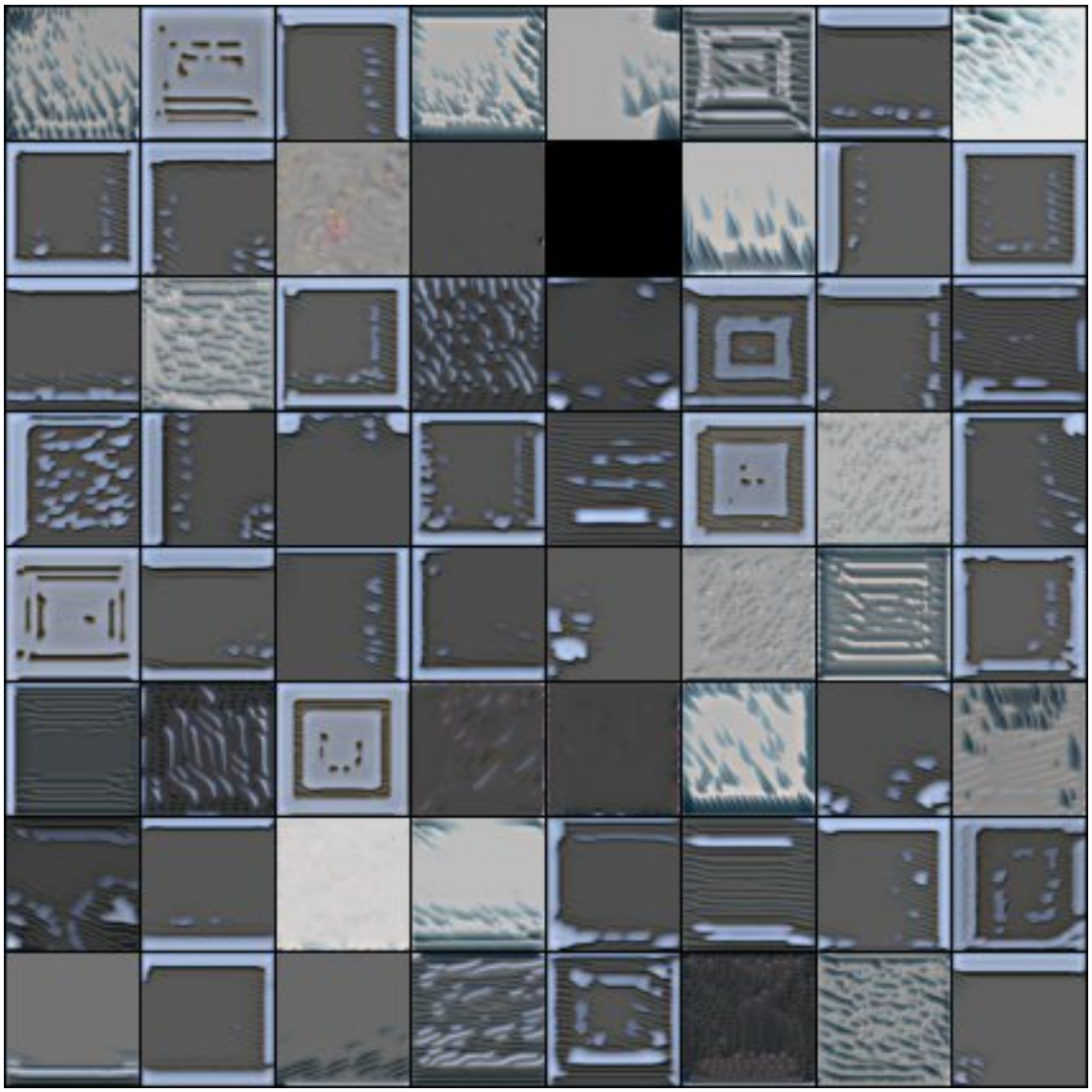


Figura 62: Resultados da maximização da ativação da camada residual8 da arquitetura Underwater Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

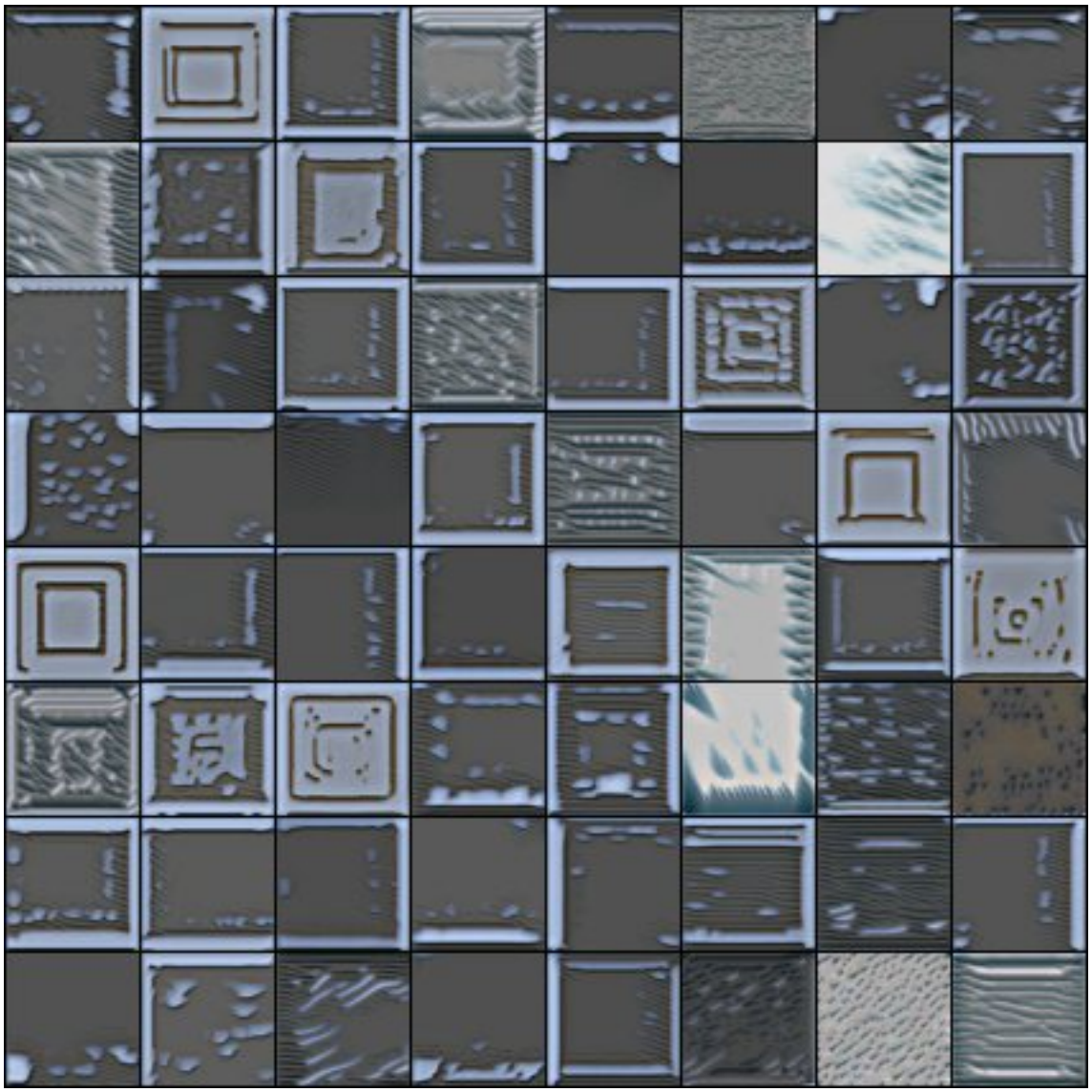


Figura 63: Resultados da maximização da ativação da camada residual9 da arquitetura Underwater Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

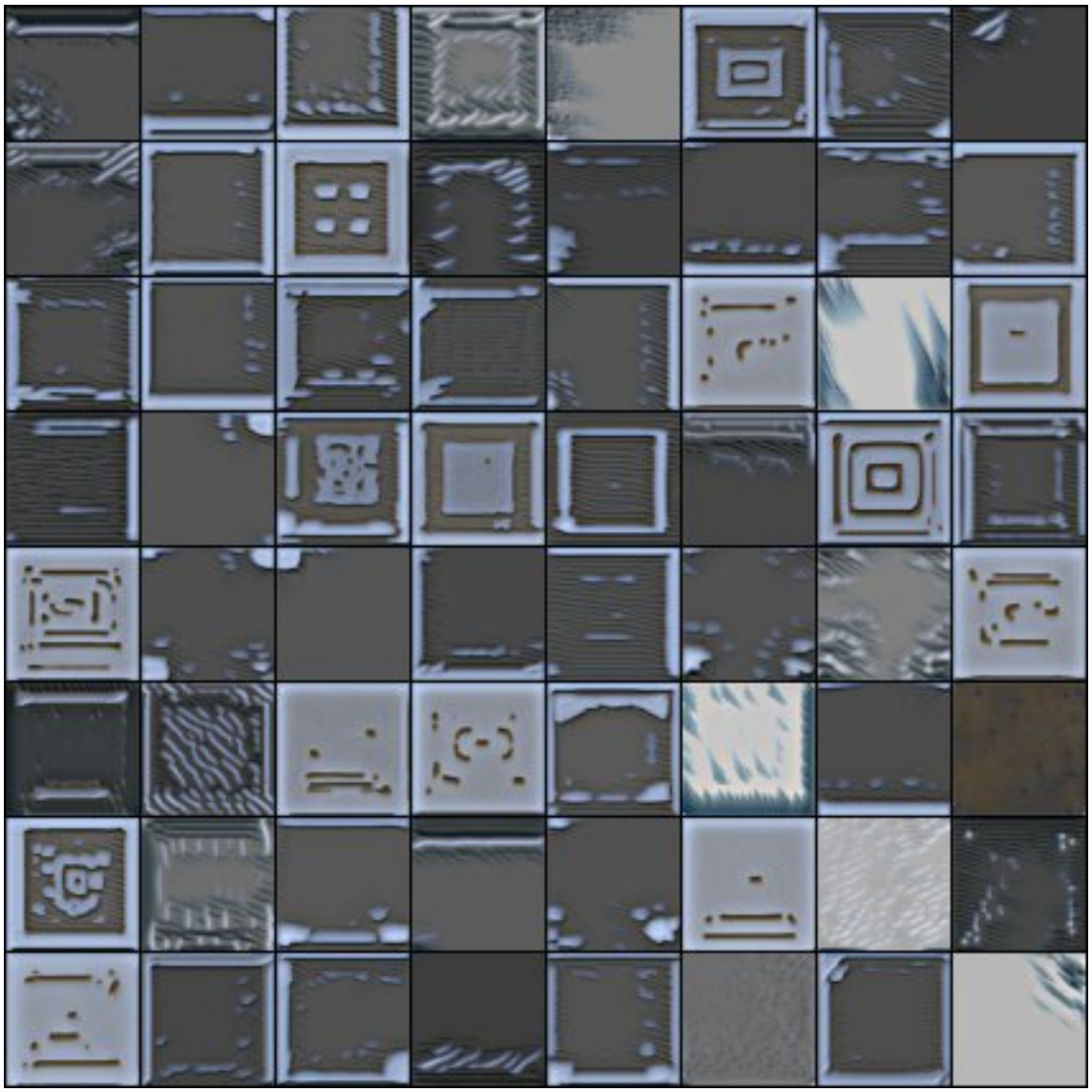


Figura 64: Resultados da maximização da ativação da camada residual10 da arquitetura Underwater Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

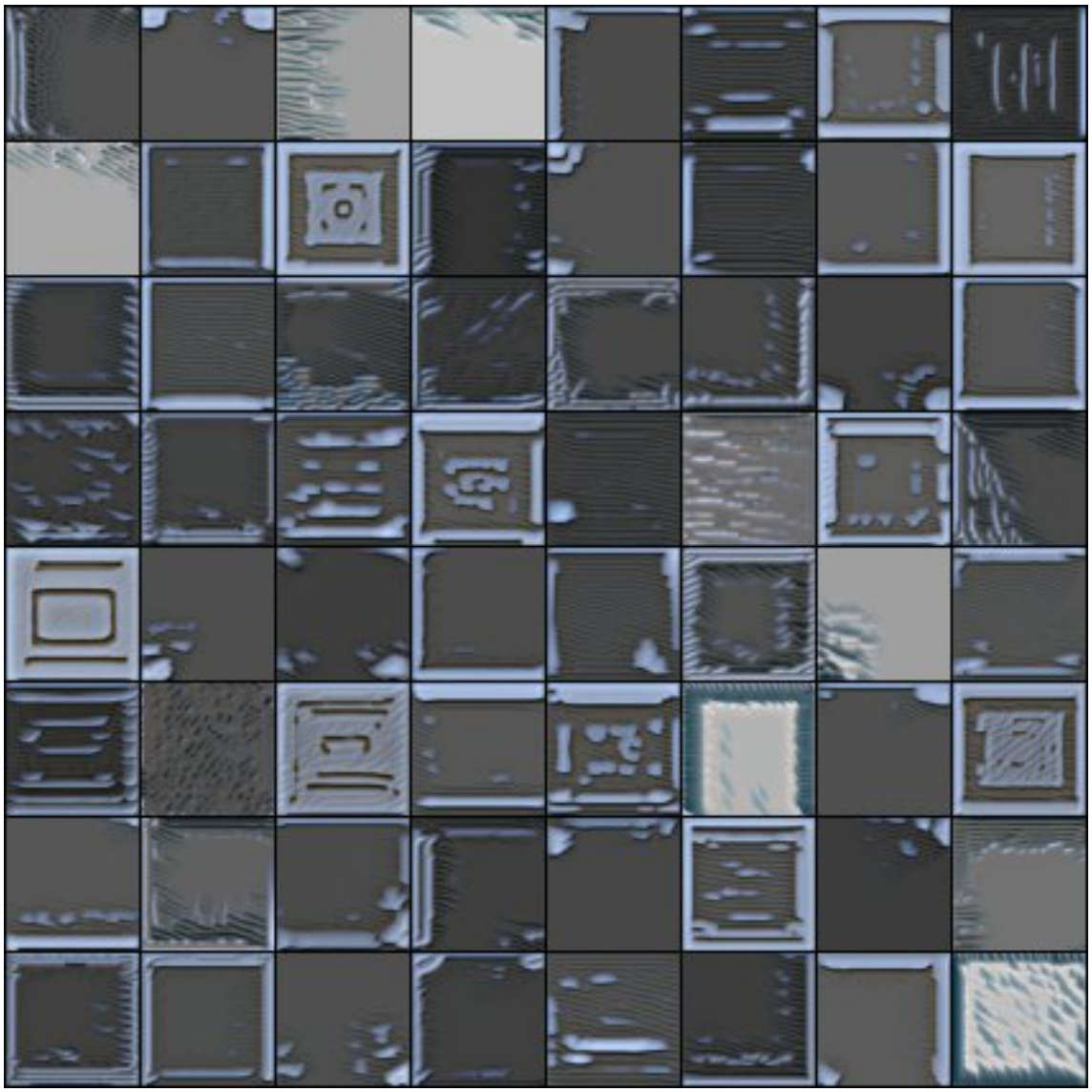


Figura 65: Resultados da maximização da ativação da camada residual11 da arquitetura Underwater Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

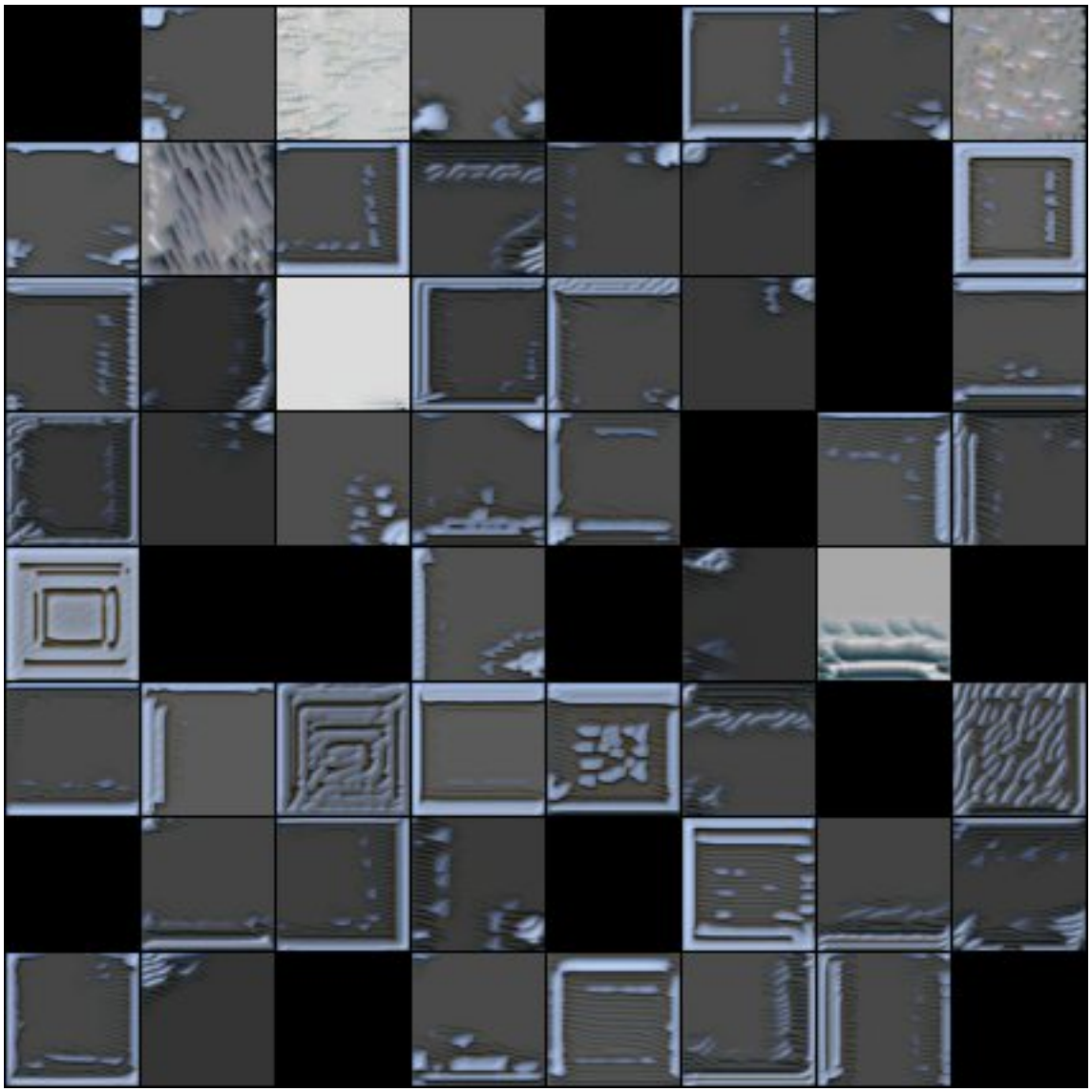
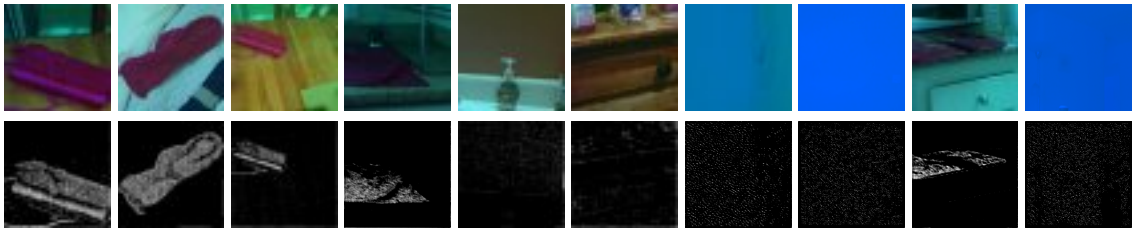
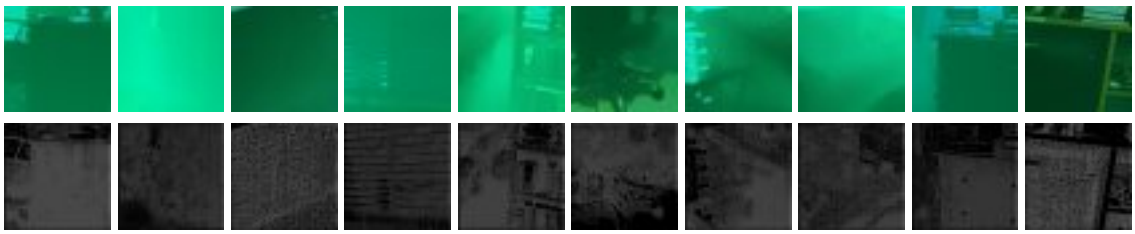


Figura 66: Resultados da maximização da ativação da camada residual12 da arquitetura Underwater Resnet 12. Os resultados são apresentados na forma de um grid onde cada imagem é o resultado de um canal diferente. Esta figura é melhor visualizada digitalmente, em cores e com zoom.

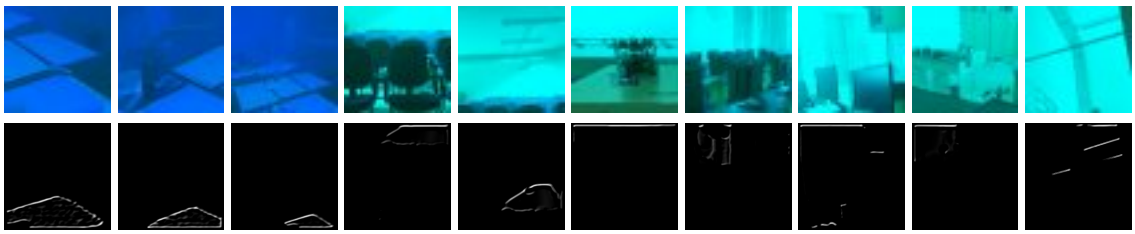


(a) camada residual12, canal 07

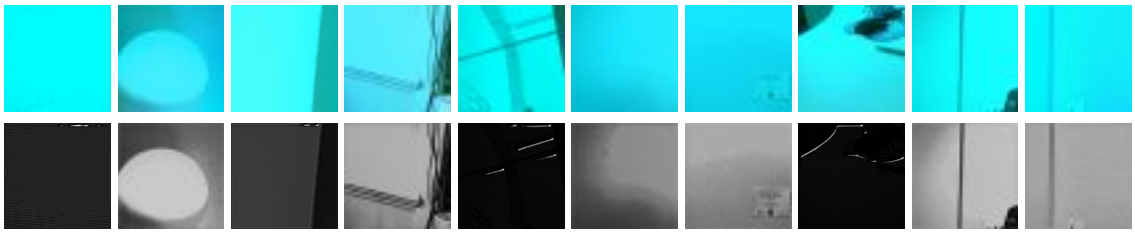
(b) camada residual12, canal 10



(c) camada residual12, canal 18



(d) camada residual12, canal 37



(e) camada residual12, canal 47

Figura 67: Visualização de alguns *feature maps* da rede Underwater Resnet 12. Acima, os *patches* do conjunto de treinamento que produzem as maiores ativações médias no *feature map*. Abaixo, as respectivas ativações. Esta figura é melhor visualizada em cores.

(a) camada residual12, canal 07

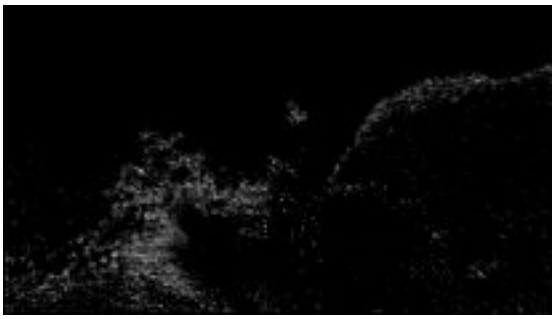
Figura 68: Acima, os *patches* limpos (sem turbidez simulada) do conjunto de treinamento que produzem as maiores ativações médias no canal 07 da última camada oculta da rede Underwater Resnet 12. Abaixo, as respectivas ativações. Esta figura é melhor visualizada em cores.

(a) entrada

(b) resultado da restauração

(c) ativação no canal 10 da camada residual12

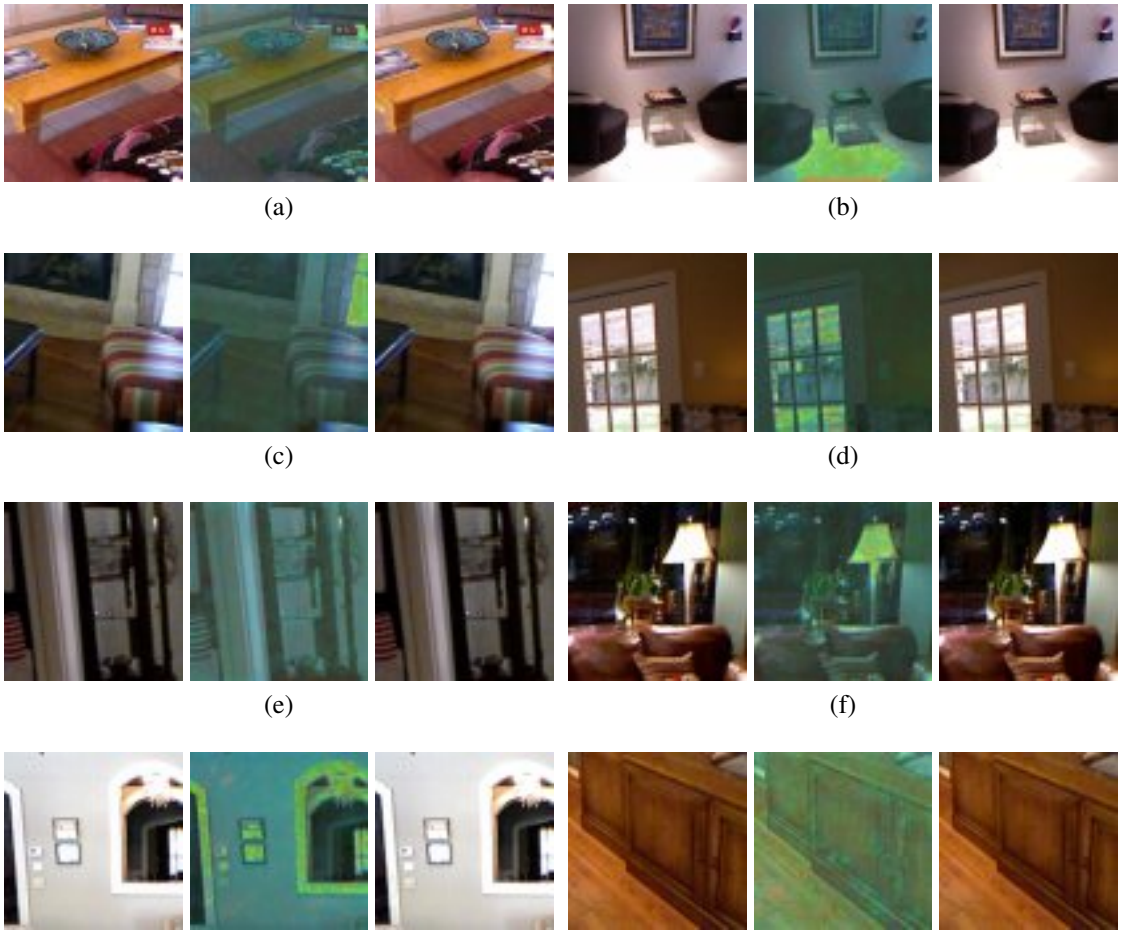
(d) ativação no canal 15 da camada residual12



(e) ativação no canal 37 da camada residual12

(f) ativação no canal 40 da camada residual12

Figura 69: Visualização direta das ativações produzidas por uma entrada em alguns *feature maps* da última camada da rede Underwater Resnet 12.



(a) (b) (c) (d) (e) (f) (g) (h)

Figura 70: Resultados da visualização por inversão da rede para a arquitetura Underwater Resnet 12. Todos os resultados foram gerados com inicialização em imagem cinza, passo de otimização de 10^{-4} e 50 mil iterações. O único método de regularização utilizado foi a normalização de gradiente por pirâmide Laplaciana. Em cada item são apresentados, da esquerda para direita, a imagem alvo, o resultado da otimização e a saída da rede para o resultado da otimização. Esta figura é melhor visualizada em cores.

6 CONCLUSÃO

Neste trabalho técnicas de visualização de redes neurais foram aplicadas a arquiteturas destinadas à resolução de problemas de transformação de imagem. Foram utilizadas como estudo de caso redes relacionadas aos problemas de estimativa de mapa de profundidade, remoção de névoa e restauração de imagens subaquáticas. A utilização dos métodos de visualização permitiu um maior conhecimento sobre as redes aos quais eles foram aplicados, mas não foi suficiente para que o funcionamento dos modelos em questão fosse compreendido por completo. Uma das razões disto é que a maioria das técnicas utilizadas neste trabalho foi desenvolvida para redes de classificação de imagens, que é um problema muito diferente dos problemas abordados aqui. Na tarefa de classificação a rede precisa decidir a qual classe uma imagem pertence, o que pode ser feito com base na presença ou na ausência de determinados *features*. Isto favorece o uso de técnicas como a maximização da ativação, que mostra quais estruturas a rede aprendeu a detectar. No caso das redes de transformação de imagem o resultado não necessariamente depende da detecção de *features*, e por isto técnicas como a maximização da ativação e a visualização das imagens que produzem as maiores ativações médias em cada *feature map* podem não ser as mais indicadas para a compreensão destas redes.

Apesar das diferenças entre as duas classes de problemas, alguns problemas de transformação de imagem podem ser tratados como problemas de classificação. No caso da remoção de névoa, por exemplo, em teoria a rede poderia ser capaz de identificar certas texturas através de meios turvos e, com base nos exemplos apresentados durante o treinamento, determinar a aparência destas texturas na ausência de turbidez. Esta abordagem é utilizada na rede de colorização de imagens apresentada em [78], onde a colorização é explicitamente tratada como um problema de classificação densa, ou seja, a rede é treinada para classificar cada pixel em classes que correspondem a intervalos no espaço de cor CIE Lab. Porém, ao que tudo indica isto não ocorre nas redes de restauração estudadas neste trabalho. Não existe nenhuma evidência de que as redes em questão aprenderam a detectar texturas específicas.

Outra possibilidade é que as redes neurais realizem a restauração através de um processo similar ao utilizado nos métodos tradicionais. Nas redes estudadas neste trabalho

não foram encontrados *feature maps* com ativações similares ao mapa de transmissão da imagem de entrada, utilizado na grande maioria dos métodos de *dehazing* e restauração de imagens subaquáticas tradicionais. Ainda assim, é possível que as redes possuam estimativas do mapa de transmissão distribuída através de diferentes *feature maps*. A arquitetura Dehaze Resnet 12 possui muitos detectores de bordas nas suas camadas mais profundas. Estes detectores de bordas podem estar sendo utilizados para estimar o contraste local, que é utilizado como estimativa do mapa de transmissão em alguns métodos de restauração, como [68] e [9]. A Underwater Resnet 12 possui *feature maps* especializados na detecção de cores associadas à presença de alta turbidez, assim como um *feature map* que detecta a intensidade da cor vermelha, que pode ser utilizada para estimar o mapa de transmissão em ambientes subaquáticos [8].

Entre os métodos utilizados, a visualização por inversão da rede se mostrou o mais eficiente, justamente por ter sido desenvolvido especificamente para redes de transformação de imagem. Este método mostra qual é a entrada necessária para se obter uma saída específica. Desta forma, é possível identificar qual o tipo de entrada "preferido" da rede, ou seja, para quais entradas a rede tende a apresentar os melhores resultados.

Apesar do conhecimento adquirido durante este trabalho, o funcionamento das redes de *dehazing* e restauração de imagens subaquáticas utilizadas como estudo de caso ainda não é completamente conhecido. Para uma melhor compreensão destes modelos é necessário o desenvolvimento de métodos de visualização mais eficientes, que levam em conta as características dos problemas de transformação de imagem.

Trabalhos futuros incluem o desenvolvimento de novos métodos de visualização, voltados especificamente para arquiteturas de transformação de imagem, além de uma melhor utilização dos métodos existentes. Outra possibilidade é a aplicação de métodos estatísticos, como a avaliação da ativação média para diferentes tipos de entrada e análise de componentes principais destas ativações. As técnicas apresentadas neste trabalho também podem ser aplicadas a outros estudos de caso, como a rede de super-resolução apresentada em [39]. Ainda, os métodos de visualização apresentados podem ser aplicados nos discriminadores utilizados em *Generative Adversarial Networks* (GANs).

REFERÊNCIAS

[1] ABADI, M. et al. *TensorFlow: large-scale machine learning on heterogeneous systems*. 2015. Software available from tensorflow.org. Disponível em: <http://tensorflow.org/>.

[2] ANCUTI, C. O. et al. A fast semi-inverse approach to detect and remove the haze from a single image. In: *Proceedings of the 10th Asian Conference on Computer Vision - Volume Part II*. Berlin, Heidelberg: Springer-Verlag, 2011. (ACCV'10), p. 501–514. ISBN 978-3-642-19308-8. Disponível em: <http://dl.acm.org/citation.cfm?id=1965992.1966032>.

[3] BURGER, H. C.; SCHULER, C. J.; HARMELING, S. Image denoising: can plain neural networks compete with bm3d? In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. [S.l.: s.n.], 2012. p. 2392–2399. ISSN 1063-6919.

[4] BURT, P.; ADELSON, E. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, v. 31, n. 4, p. 532–540, Apr 1983. ISSN 0090-6778.

[5] CAI, B. et al. Dehazenet: an end-to-end system for single image haze removal. *CoRR*, abs/1601.07661, 2016. Disponível em: <http://arxiv.org/abs/1601.07661>.

[6] CARLEVARIS-BIANCO, N.; MOHAN, A.; EUSTICE, R. M. Initial results in underwater single image dehazing. In: *OCEANS 2010 MTS/IEEE SEATTLE*. [S.l.: s.n.], 2010. p. 1–8. ISSN 0197-7385.

[7] CLUNE, J. et al. On the performance of indirect encoding across the continuum of regularity. *IEEE Transactions on Evolutionary Computation*, v. 15, n. 3, p. 346–367, June 2011. ISSN 1089-778X.

[8] CODEVILLA, F. et al. Underwater single image restoration using dark channel prior. In: *2014 Symposium on Automation and Computation for Naval, Offshore and Subsea (NAVCOMP)*. [S.l.: s.n.], 2014. p. 18–21.

[9] CODEVILLA, F. et al. Single image restoration for participating media based on prior fusion. *CoRR*, abs/1603.01864, 2016. Disponível em: <http://arxiv.org/abs/1603.01864>.

[10] DABOV, K. et al. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, v. 16, n. 8, p. 2080–2095, Aug 2007. ISSN 1057-7149.

[11] DEB, K.; KALYANMOY, D. *Multi-objective optimization using evolutionary algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001. ISBN 047187339X.

[12] DENG, J. et al. Imagenet: a large-scale hierarchical image database. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. [S.l.: s.n.], 2009. p. 248–255. ISSN 1063-6919.

[13] DREWS-JR, P. L. et al. Transmission estimation in underwater single images. In: *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*. [S.l.: s.n.], 2013. p. 825–830.

[14] DREWS-JR, P. L. et al. Underwater depth estimation and image restoration based on single images. *IEEE Computer Graphics and Applications*, v. 36, n. 2, p. 24–35, Mar 2016. ISSN 0272-1716.

[15] EIGEN, D.; KRISHNAN, D.; FERGUS, R. Restoring an image taken through a window covered with dirt or rain. In: *Proceedings of the 2013 IEEE International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 2013. (ICCV '13), p. 633–640. ISBN 978-1-4799-2840-8. Disponível em: <http://dx.doi.org/10.1109/ICCV.2013.84>.

[16] EIGEN, D.; PUHRSCH, C.; FERGUS, R. Depth map prediction from a single image using a multi-scale deep network. *CoRR*, abs/1406.2283, 2014. Disponível em: <http://arxiv.org/abs/1406.2283>.

[17] ERHAN, D. et al. *Visualizing higher-layer features of a deep network*. [S.l.], 2009. Also presented at the ICML 2009 Workshop on Learning Feature Hierarchies, Montréal, Canada.

[18] FINLAYSON, G. D.; TREZZI, E. Shades of gray and colour constancy. In: SOCIETY FOR IMAGING SCIENCE AND TECHNOLOGY. *Color and imaging conference*. [S.l.], 2004. v. 2004, n. 1, p. 37–41.

[19] FLOREANO, D.; MATTIUSSI, C. *Bio-inspired artificial intelligence: theories, methods, and technologies*. [S.l.]: The MIT Press, 2008. ISBN 0262062712, 9780262062718.

[20] GEIGER, A. et al. Vision meets robotics: the kitti dataset. *Int. J. Rob. Res.*, Sage Publications, Inc., Thousand Oaks, CA, USA, v. 32, n. 11, p. 1231–1237, set. 2013. ISSN 0278-3649. Disponível em: <http://dx.doi.org/10.1177/0278364913491297>.

[21] GONZALEZ, R. C.; WOODS, R. E. *Digital image processing (3rd edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN 013168728X.

[22] GOODFELLOW, I. J. et al. Maxout networks. In: *Proceedings of the 30th International Conference on Machine Learning (ICML'13)*. [S.l.: s.n.], 2013. p. 2356–2364.

[23] HE, K.; SUN, J.; TANG, X. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, v. 33, n. 12, p. 2341–2353, dez. 2011. ISSN 0162-8828. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2010.168>.

[24] HE, K.; SUN, J.; TANG, X. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 35, n. 6, p. 1397–1409, June 2013. ISSN 0162-8828.

[25] HE, K. et al. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 770–778.

[26] HOCHREITER, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, World Scientific Publishing Co., Inc., River Edge, NJ, USA, v. 6, n. 2, p. 107–116, abr. 1998. ISSN 0218-4885. Disponível em: <http://dx.doi.org/10.1142/S0218488598000094>.

[27] HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. *Neural Netw.*, Elsevier Science Ltd., Oxford, UK, UK, v. 2, n. 5, p. 359–366, jul. 1989. ISSN 0893-6080. Disponível em: <http://dx.doi.org/10.1016/0893-6080(89)90020-8>.

[28] HUIYING, D.; SHAN, J. Haze removal for single image based on physical model and guided filtering algorithm. In: *The 27th Chinese Control and Decision Conference (2015 CCDC)*. [S.l.: s.n.], 2015. p. 4988–4992. ISSN 1948-9439.

[29] JAFFE, J. Computer modeling and the design of optimal underwater imaging systems. *Oceanic Engineering, IEEE Journal of*, v. 15, n. 2, p. 101–111, Apr 1990. ISSN 0364-9059.

[30] JAIN, V.; SEUNG, S. Natural image denoising with convolutional networks. In: KOLLER, D. et al. (Ed.). *Advances in Neural Information Processing Systems 21*. Curran Associates, Inc., 2009. p. 769–776. Disponível

em: <http://papers.nips.cc/paper/3506-natural-image-denoising-with-convolutional-networks.pdf>.

[31] JANOCH, A. et al. A category-level 3d object dataset: putting the kinect to work. In: ____. *Consumer Depth Cameras for Computer Vision: Research Topics and Applications*. London: Springer London, 2013. p. 141–165. ISBN 978-1-4471-4640-7. Disponível em: <http://dx.doi.org/10.1007/978-1-4471-4640-7_8>.

[32] JOHNSON, J.; ALAHI, A.; FEI-FEI, L. Perceptual losses for real-time style transfer and super-resolution. In: ____. *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*. Cham: Springer International Publishing, 2016. p. 694–711. ISBN 978-3-319-46475-6. Disponível em: <http://dx.doi.org/10.1007/978-3-319-46475-6_43>.

[33] JOLLIFFE, I. *Principal component analysis*. [S.l.]: Springer Verlag, 1986.

[34] KINDERMANN, R.; SNELL, J. L. *Markov random fields and their applications*. AMS, 1980. Disponível em: <http://www.ams.org/online_bks/conm1/>.

[35] KRATZ, L.; NISHINO, K. Factorizing scene albedo and depth from a single foggy image. In: *2009 IEEE 12th International Conference on Computer Vision*. [S.l.: s.n.], 2009. p. 1701–1708. ISSN 1550-5499.

[36] KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F. et al. (Ed.). *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012. p. 1097–1105. Disponível em: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.

[37] LAINA, I. et al. Deeper depth prediction with fully convolutional residual networks. In: IEEE. *3D Vision (3DV), 2016 Fourth International Conference on*. [S.l.], 2016. p. 239–248.

[38] LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, Nov 1998. ISSN 0018-9219.

[39] LEDIG, C. et al. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016. Disponível em: <http://arxiv.org/abs/1609.04802>.

[40] LEVIN, A.; LISCHINSKI, D.; WEISS, Y. A closed form solution to natural image matting. In: *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*. Washington, DC, USA: IEEE Com-

puter Society, 2006. (CVPR '06), p. 61–68. ISBN 0-7695-2597-0. Disponível em: <http://dx.doi.org/10.1109/CVPR.2006.18>.

[41] MAATEN, L. van der; HINTON, G. E. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, v. 9, p. 2579–2605, 2008.

[42] MAIRAL, J.; ELAD, M.; SAPIRO, G. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, v. 17, n. 1, p. 53–69, Jan 2008. ISSN 1057-7149.

[43] MCGLAMERY, B. L. A computer model for underwater camera systems. In: . [s.n.], 1980. v. 0208, p. 221–231. Disponível em: <http://dx.doi.org/10.1117/12.958279>.

[44] MENG, G. et al. Efficient image dehazing with boundary constraint and contextual regularization. In: *2013 IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2013. p. 617–624. ISSN 1550-5499.

[45] NARASIMHAN, S. G.; NAYAR, S. K. Contrast restoration of weather degraded images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 25, n. 6, p. 713–724, June 2003. ISSN 0162-8828.

[46] NAYAR, S. K.; NARASIMHAN, S. G. Vision in bad weather. In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. [S.l.: s.n.], 1999. v. 2, p. 820–827 vol.2.

[47] NGUYEN, A. M.; YOSINSKI, J.; CLUNE, J. Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. *CoRR*, abs/1412.1897, 2014. Disponível em: <http://arxiv.org/abs/1412.1897>.

[48] NGUYEN, A. M.; YOSINSKI, J.; CLUNE, J. Multifaceted feature visualization: uncovering the different types of features learned by each neuron in deep neural networks. *CoRR*, abs/1602.03616, 2016. Disponível em: <http://arxiv.org/abs/1602.03616>.

[49] NOSRATINIA, A. Enhancement of jpeg-compressed images by re-application of jpeg. *Journal of VLSI signal processing systems for signal, image and video technology*, v. 27, n. 1, p. 69–79, 2001. ISSN 0922-5773. Disponível em: <http://dx.doi.org/10.1023/A:1008167430544>.

[50] PANG, J.; AU, O. C.; GUO, Z. Improved single image dehazing using guided filter. *Proc. APSIPA ASC*, p. 1–4, 2011.

[51] PEARSON, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, v. 2, n. 6, p. 559–572, 1901.

[52] PORTILLA, J. et al. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, v. 12, n. 11, p. 1338–1351, Nov 2003. ISSN 1057-7149.

[53] REN, S. et al. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. Disponível em: <http://arxiv.org/abs/1506.01497>.

[54] ROTH, S.; BLACK, M. J. Fields of experts: a framework for learning image priors. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. [S.l.: s.n.], 2005. v. 2, p. 860–867 vol. 2. ISSN 1063-6919.

[55] RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Neurocomputing: Foundations of research. In: ANDERSON, J. A.; ROSENFELD, E. (Ed.). Cambridge, MA, USA: MIT Press, 1988. cap. Learning representations by back-propagating errors, p. 696–699. ISBN 0-262-01097-6. Disponível em: <http://dl.acm.org/citation.cfm?id=65669.104451>.

[56] RUSSAKOVSKY, O. et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, v. 115, n. 3, p. 211–252, 2015.

[57] SAXENA, A.; SUN, M.; NG, A. Y. Make3d: learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 31, n. 5, p. 824–840, May 2009. ISSN 0162-8828.

[58] SCHECHNER, Y. Y.; KARPEL, N. Recovery of underwater visibility and structure by polarization analysis. *IEEE Journal of Oceanic Engineering*, v. 30, n. 3, p. 570–587, July 2005. ISSN 0364-9059.

[59] SCHECHNER, Y. Y.; NARASIMHAN, S. G.; NAYAR, S. K. Instant dehazing of images using polarization. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. [S.l.: s.n.], 2001. v. 1, p. I–325–I–332 vol.1. ISSN 1063-6919.

[60] SHWARTZ, S.; NAMER, E.; SCHECHNER, Y. Y. Blind haze separation. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. [S.l.: s.n.], 2006. v. 2, p. 1984–1991. ISSN 1063-6919.

[61] SILBERMAN, N. et al. Indoor segmentation and support inference from rgbd images. In: ____. *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 746–760. ISBN 978-3-642-33715-4. Disponível em: <http://dx.doi.org/10.1007/978-3-642-33715-4_54>.

[62] SIMONYAN, K.; VEDALDI, A.; ZISSERMAN, A. Deep inside convolutional networks: visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013. Disponível em: <http://arxiv.org/abs/1312.6034>.

[63] SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. Disponível em: <http://arxiv.org/abs/1409.1556>.

[64] STANLEY, K. O.; MIIKKULAINEN, R. A taxonomy for artificial embryogeny. *Artif. Life*, MIT Press, Cambridge, MA, USA, v. 9, n. 2, p. 93–130, abr. 2003. ISSN 1064-5462. Disponível em: <http://dx.doi.org/10.1162/106454603322221487>.

[65] SULAMI, M. et al. Automatic recovery of the atmospheric light in hazy images. In: *Computational Photography (ICCP), 2014 IEEE International Conference on*. [S.l.: s.n.], 2014. p. 1–11.

[66] SZEGEDY, C. et al. Inception-v4, inception-resnet and the impact of residual connections on learning. In: *ICLR 2016 Workshop*. [s.n.], 2016. Disponível em: <https://arxiv.org/abs/1602.07261>.

[67] SZEGEDY, C. et al. Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 1–9. ISSN 1063-6919.

[68] TAN, R. T. Visibility in bad weather from a single image. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. [S.l.: s.n.], 2008. p. 1–8. ISSN 1063-6919.

[69] TAREL, J. P.; HAUTIèRE, N. Fast visibility restoration from a single color or gray level image. In: *2009 IEEE 12th International Conference on Computer Vision*. [S.l.: s.n.], 2009. p. 2201–2208. ISSN 1550-5499.

[70] TOMASI, C.; MANDUCHI, R. Bilateral filtering for gray and color images. In: *Computer Vision, 1998. Sixth International Conference on*. [S.l.: s.n.], 1998. p. 839–846.

[71] VINCENT, P. et al. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, JMLR.org, v. 11, p. 3371–3408, dez. 2010. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=1756006.1953039>.

[72] WELSH, T.; ASHIKHMIN, M.; MUELLER, K. Transferring color to greyscale images. *ACM Trans. Graph.*, ACM, New York, NY, USA,

v. 21, n. 3, p. 277–280, jul. 2002. ISSN 0730-0301. Disponível em: <http://doi.acm.org/10.1145/566654.566576>.

[73] XIE, J.; XU, L.; CHEN, E. Image denoising and inpainting with deep neural networks. In: BARTLETT, P. et al. (Ed.). *Advances in Neural Information Processing Systems 25*. [s.n.], 2012. p. 350–358. Disponível em: <http://books.nips.cc/papers/files/nips25/NIPS2012_0184.pdf>.

[74] XU, L. et al. Deep convolutional neural network for image deconvolution. In: GHAHRAMANI, Z. et al. (Ed.). *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014. p. 1790–1798. Disponível em: <http://papers.nips.cc/paper/5485-deep-convolutional-neural-network-for-image-deconvolution.pdf>.

[75] YOSINSKI, J. et al. Understanding neural networks through deep visualization. *CoRR*, abs/1506.06579, 2015. Disponível em: <http://arxiv.org/abs/1506.06579>.

[76] ZEILER, M.; FERGUS, R. Visualizing and understanding convolutional networks. In: ____. *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*. Cham: Springer International Publishing, 2014. p. 818–833. ISBN 978-3-319-10590-1. Disponível em: <http://dx.doi.org/10.1007/978-3-319-10590-1_53>.

[77] ZEILER, M. D.; TAYLOR, G. W.; FERGUS, R. Adaptive deconvolutional networks for mid and high level feature learning. In: *2011 International Conference on Computer Vision*. [S.l.: s.n.], 2011. p. 2018–2025. ISSN 1550-5499.

[78] ZHANG, R.; ISOLA, P.; EFROS, A. A. Colorful image colorization. *CoRR*, abs/1603.08511, 2016. Disponível em: <http://arxiv.org/abs/1603.08511>.

[79] ZHU, Q.; MAI, J.; SHAO, L. A fast single image haze removal algorithm using color attenuation prior. *IEEE Transactions on Image Processing*, v. 24, n. 11, p. 3522–3533, Nov 2015. ISSN 1057-7149.

[80] ZWALD, L.; LAMBERT-LACROIX, S. The BerHu penalty and the grouped effect. *ArXiv e-prints*, jul. 2012.

APÊNDICE A ALGORITMOS DE REMOÇÃO DE NÉVOA

A.1 Restauração Baseada no Contraste

Em [68] é apresentado um método de *dehazing* para imagens capturadas ao ar livre, durante o dia. O método apresentado necessita apenas de uma única imagem como entrada, e pode ser aplicado a imagens coloridas ou em escala de cinza. O método é baseado em duas observações básicas: primeiro, imagens limpas (sem degradação) têm um maior contraste em relação a imagens degradadas por neblina; segundo, a luz ambiente cuja variação depende principalmente da distância tende a ser uniforme. Essas duas observações são utilizadas para desenvolver uma função de energia de um Campo aleatório de Markov (CAM) [34], que pode ser otimizada por várias técnicas, como corte de grafos e propagação de crença.

Dada uma imagem de entrada, o primeiro passo do método é estimar a luz atmosférica, da qual se pode obter a cromaticidade da luz. Essa cromaticidade é usada para se remover a cor da luz atmosférica da imagem de entrada. O próximo passo é o cálculo de um *custo de dados* (data cost) e um *custo de uniformidade* (smoothness cost) para cada pixel. O custo de dados é calculado com base no contraste de um pequeno *patch* da imagem de entrada. O custo de uniformidade é calculado da diferença dos *labels* de dois pixels vizinhos, onde os labels são idênticos aos valores de luz ambiente. Os custos de dados e de uniformidade são utilizados para construir CAMs completos que podem ser otimizados usando métodos de inferência existentes, o que produz os valores estimados de luz ambiente. Com base nesta estimativa da luz ambiente é possível calcular a atenuação direta, que representa a cena com visibilidade aprimorada. O objetivo deste método não é recuperar as cores originais da imagem, e sim produzir uma imagem com maior contraste, o que melhora a visibilidade.

O método apresentado foi capaz de recuperar grande parte do contraste em algumas imagens capturas em condições de neblina intensa, tornando possível a visualização de detalhes que são quase imperceptíveis nas imagens originais, como pode ser observado na Figura 71. Apesar disso os resultados não são perfeitos. As imagens produzidas por este método costumam apresentar artefatos, além de uma saturação de cores maior do que

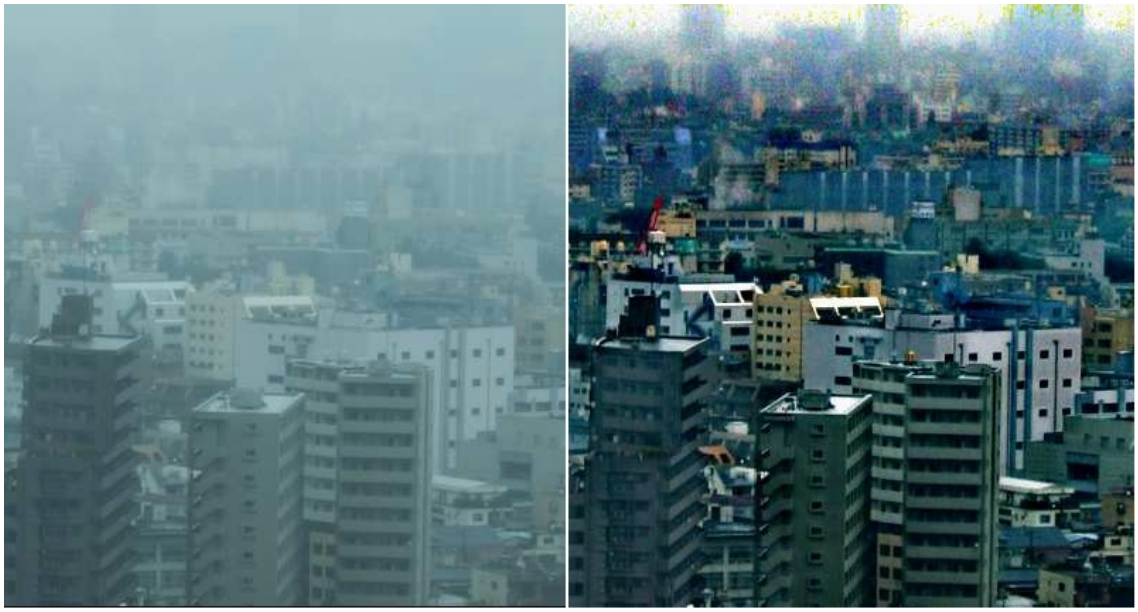


Figura 71: Resultado do método de *dehazing* proposto em [68].

a das imagens capturadas em condições atmosféricas ideais. A supersaturação das cores ocorre porque o método tende a subestimar a transmissão [23].

A.2 Dark Channel Prior

Dark channel prior [23] é uma suposição sobre as características de imagens naturais que pode ser utilizada na restauração de imagens capturadas em condições de neblina. A dark channel prior (DCP) é baseada na observação que, em imagens de cenas exteriores capturadas em dias claros, em cada *patch* que não faz parte do céu pelo menos um canal de cor tem uma intensidade muito baixa em alguns pixels. Em outras palavras, a intensidade mínima nestes *patches* tem um valor muito baixo. Formalmente, para uma imagem **J**, define-se:

$$J^{dark}(x) = \min_{c \in \{r,g,b\}} \left(\min_{y \in \Omega(x)} \left(J^c(y) \right) \right),$$

onde J^c é um canal de cor de **J** e $\Omega(x)$ é um *patch* local com centro em x. Segundo as observações dos autores de [23], com exceção da região do céu, a intensidade de J^c é baixa e tende a ser zero se **J** é uma imagem ao ar livre sem degradação por névoa. J^{dark} é chamado de *dark channel*, ou canal escuro, de **J**, e a observação descrita acima é chamada de *dark channel prior*.

Segundo [23], as baixas intensidades no dark channel são causadas por três fatores: a) sombras. Por exemplo, as sombras dos carros, prédios e o interior de janelas em paisagens urbanas, ou as sombras de folhas, arvores e pedras em imagens do campo; b) objetos ou superfícies coloridas. Qualquer objeto (como plantas verdes, flores vermelhas ou a superfície azul da água) que tem um valor baixo em um dos canais de cor vai resultar

em valores baixos no dark channel; c) objetos ou superfícies escuras. Por exemplo, uma pedra ou um tronco de arvore. As imagens naturais ao ar livre normalmente são coloridas e cheias de sombras, logo o dark channel destas imagens tende a ser muito escuro.

Para verificar a qualidade da dark channel prior os autores coletaram 5000 imagens da internet, removeram manualmente a região do céu e calcularam o dark channel usando um tamanho de *patch* de 15×15. 75% dos pixels de dark channel apresentaram o valor zero em um dos canais, e em 90% dos pixels a intensidade foi menor que 25. Segundos os autores estas estatísticas dão suporte à dark channel prior.

Por causa do componente aditivo da luz ambiente, uma imagem com névoa é mais brilhante que uma versão sem névoa da mesma imagem onde a transmissão t é baixa. Isso significa que o dark channel de uma imagem com névoa terá uma intensidade maior nas regiões onde a névoa é mais densa. Visualmente, a intensidade do dark channel é uma aproximação da espessura da névoa. Esta propriedade por ser utilizada para estimar a transmissão e a luz atmosférica.

A.2.1 Estimativa da Luz Atmosférica

Uma forma de estimar a luz atmosférica $\mathbf{L}_\infty$ é selecionar o pixel da imagem com maior intensidade, como é feito em [68]. Esta abordagem, porém, é falha, já que em imagens reais o pixel mais brilhante pode estar em um carro ou prédio branco. Uma abordagem mais robusta é proposta em [23]. O primeiro passo desta abordagem é selecionar os top 0.1% pixels da imagem com o dark channel mais claro. Entre os pixels selecionados o mais brilhante é escolhido como luz atmosférica. O pixel escolhido não necessariamente é o pixel mais brilhante da imagem inteira.

A.2.2 Estimativa da Transmissão

Um método para a estimativa da transmissão através da dark channel prior é apresentado em [23]. Neste método supõe-se que a luz atmosférica $\mathbf{L}_\infty$ é conhecida. Também supõe-se que a transmissão em um *patch* local $\Omega(x)$ é constante. Denota-se a transmissão do *patch* por $\tilde{t}(x)$. A aplicação da operação de mínimo na equação 2 resulta em:

$$\min_{y \in \Omega(x)} \left(I^c(y)\right) = \tilde{t}(x) \min_{y \in \Omega(x)} \left(J^c(y)\right) + \left(1 - \tilde{t}(x)\right) L_\infty^c. \tag{3}$$

A operação de mínimo é aplicada nos três canais de cor de forma independente. Esta equação é equivalente a:

$$\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{L_\infty^c}\right) = \tilde{t}(x) \min_{y \in \Omega(x)} \left(\frac{J^c(y)}{L_\infty^c}\right) + \left(1 - \tilde{t}(x)\right). \tag{4}$$

Aplicando a operação de mínimo entre os três canais de cor na equação acima se obtém:

$$\min_c \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{L_\infty^c} \right) \right) = \tilde{t}(x) \min_c \left(\min_{y \in \Omega(x)} \left(\frac{J^c(y)}{L_\infty^c} \right) \right) + \left(1 - \tilde{t}(x)\right). \tag{5}$$

De acordo com a dark channel prior, o dark channel J^{dark} do sinal livre de névoa $\mathbf{J}$ deve tender a zero:

$$J^{dark}(x) = \min_c \left(\min_{y \in \Omega(x)} \left(J^c(y) \right) \right) = 0. \tag{6}$$

Como L_∞^c é sempre positivo, isso leva a:

$$\min_c \left(\min_{y \in \Omega(x)} \left(\frac{J^c(y)}{L_\infty^c} \right) \right) = 0 \tag{7}$$

A transmissão $\tilde{t}(x)$ pode ser estimada colocando-se a equação 7 na equação 5:

$$\tilde{t}(x) = 1 - \min_c \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{L_\infty^c} \right) \right). \tag{8}$$

$\min_c \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{L_\infty^c} \right) \right)$ é o dark channel da imagem normalizada $\frac{I^c(y)}{L_\infty^c}$, o que fornece diretamente uma estimativa da transmissão.

A dark channel prior não é válida para as regiões da imagem que mostram o céu. Felizmente, a cor do céu normalmente é muito similar à luz atmosférica $\mathbf{L}_\infty$ nas imagens com névoa e tem-se:

$$\min_c \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{L_\infty^c} \right) \right) \to 1, \ \tilde{t}(x) \to 0,$$

nas regiões de céu. Como o céu está no infinito e tende a ter uma transmissão zero, a equação 8 trata tanto das regiões de céu quanto das outras regiões. Não existe a necessidade de separar as regiões de céu previamente.

Na prática, nem mesmo nos dias claros a atmosfera está completamente livre de partículas, e por isso o efeito de névoa sempre pode ser observado a longas distâncias. A remoção completa da névoa resulta em imagens com uma aparência não natural e provoca a perda da sensação de profundidade. Por essa razão pode-se optar por manter uma pequena quantidade de névoa na imagem para objetos distantes através da introdução de um parâmetro constante ω $(0 < \omega \leq 1)$ na equação 8:

$$\tilde{t}(x) = 1 - \omega \min_c \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{L_\infty^c} \right) \right). \tag{9}$$

Uma propriedade desejável desta modificação é que ela mantém mais névoa para objetos distantes. O valor de ω depende da aplicação. O valor usado em [23] é 0.95.

Os mapas de transmissão gerados com este método são relativamente bons, mas con-

tém alguns efeitos de bloco, que ocorrem porque a transmissão nem sempre e constante dentro de um *patch* [23]. Para amenizar estes efeitos o mapa de transmissão resultante é refinado com o algoritmo soft matting [40].

A.2.3 Relação com o Problema de Matting

O problema de matting consiste na extração do objeto que está em primeiro plano em uma imagem. Os métodos de matting recebem como entrada uma imagem **I**, que é composta por uma imagem de primeiro plano F e uma imagem de fundo B. A cor do pixel x é modelada como uma combinação linear das cores nas posições correspondentes de F e B [40]:

$$\mathbf{I}(x) = \alpha(x) F(x) + (1 - \alpha(x)) B(x),$$

onde $\alpha(x)$ é a opacidade do pixel no primeiro plano. Os algoritmos de matting, em geral, buscam determinar o valor de α em todos os pontos da imagem, com base em uma estimativa inicial fornecida pelo usuário.

Substituindo-se a cor do primeiro plano F pela radiância da cena **J**, a cor de fundo B pela luz atmosférica $\mathbf{L}_\infty$, e o termo de opacidade α pela transmissão t, a equação resultante é:

$$\mathbf{I}(x) = t(x) \mathbf{J}(x) + (1 - t(x)) \mathbf{L}_\infty(x).$$

Considerando-se a luz atmosférica $\mathbf{L}_\infty$ como constante em todos os pontos da imagem, esta equação é equivalente ao modelo de formação de imagens em meios participativos descrito pela equação 2. Devido a esta semelhança entre os modelos, em [23] é proposta a utilização do algoritmo soft matting [40] para refinar o mapa de transmissão inicial, estimado através da dark channel prior.

O mapa de transmissão refinado é obtido através da execução do algoritmo soft matting com a imagem degradada **I** como imagem de entrada e o mapa de transmissão $\tilde{t}$ como estimativa inicial de α. A matriz α resultante é equivalente a uma estimativa da transmissão da cena, que possui maior fidelidade às estruturas da imagem em relação à estimativa inicial $\tilde{t}$. Um exemplo dos resultados deste processo é apresentado na Figura 72. A principal desvantagem deste método é a alta complexidade do algoritmo soft matting, que resulta em um longo tempo de execução e consequentemente impossibilita a sua aplicação em tempo real [28]. Por esta razão, em muitos casos [28, 50, 79, 5] o soft matting é substituído pelo algoritmo guided filter [24], que produz resultados comparáveis com um custo computacional muito menor [50].

A.2.4 Recuperação da Radiância da Cena

Com o mapa de transmissão, a radiância da cena **J** pode ser recuperada de acordo com a equação 2. Porém o termo de atenuação direta $\mathbf{J}(x) t(x)$ pode ser muito próximo de zero quando a transmissão $t(x)$ é próxima de zero. Nestes casos, a recuperação direta da

Figura 72: Da esquerda para direita: A imagem de entrada, o mapa de transmissão estimado por DCP, o mapa de transmissão refinado por soft matting, o resultado final da restauração. Fonte: [23].

Figura 73: Comparação entre os resultados de *dehazing* por contraste (meio) [68] e por DCP (direita) [23].

radiância da cena **J** fica sujeita a ruído. Por essa razão a transmissão $t\left(x\right)$ deve ser restrita a um limite inferior t_0, o que significa que uma pequena quantidade de névoa é mantida nas regiões onde a neblina é mais intensa. Assim, a expressão final para recuperação da radiância da cena e dada por:

$$\mathbf{J}\left(x\right) = \frac{\mathbf{I}\left(x\right) - \mathbf{L}_{\infty}}{\max\left(t\left(x\right), t_0\right)} + \mathbf{L}_{\infty}$$

Um valor típico para t_0 é 0.1. O sinal **J** geralmente não é tão brilhante quanto a luz atmosférica, e por isso as imagens restauradas com este método tendem a ficar escuras.

A.2.5 Resultados

Segundo os autores, o método apresentado é capaz de recuperar detalhes e cores vivas até mesmo em regiões da imagem com névoa muito densa. Os mapas de profundidade estimados são nítidos e consistentes com as imagens de entrada. Ao contrario da restauração baseada em contraste [68], que pode apresentar resultados com cores supersaturadas, o método baseado na dark channel prior recupera as estruturas da imagem sem sacrificar a fidelidade das cores. Uma comparação entre os resultados dos dois métodos é apresentada na Figura 73. Uma desvantagem da restauração por DCP é que ela tende a superestimar a transmissão [79], o que pode resultar em cores com baixa saturação.

A.3 Atenuação de Cor

Em [79] é proposto um método de estimativa da transmissão baseado na atenuação das cores em imagens degradas por névoa. O método é baseado na observação que em uma imagem com névoa o brilho e a saturação variam bruscamente com a mudança na concentração da névoa.

A explicação para esta variação, segundo [79], está no modelo de formação da imagem degradada. Em uma condição de ausência de névoa, os elementos da cena refletem a energia da fonte de iluminação (luz do sol, luz difusa no céu, luz refletida pelo solo), e a maior parte desta energia chega até a câmera. Por essa razão as imagens capturadas ao ar livre nestas condições geralmente apresentam cores vivas. Já em condições de neblina a situação se torna muito mais complexa. A atenuação direta, causada pela redução da energia refletida diretamente pela cena, produz uma baixa intensidade de brilho. Por outro lado, a luz ambiente branca ou cinza, formada pelo espalhamento da iluminação ambiente, aumenta o brilho e reduz a saturação. Como a luz ambiente exerce maior influência na maioria dos casos, as regiões da imagem com névoa são caracterizadas por um alto brilho e uma baixa saturação. Quanto maior a densidade da névoa, maior a influência da luz ambiente. Isto permite que a diferença entre o brilho e a saturação seja utilizada para estimar a concentração de névoa em cada ponto da imagem. Um exemplo desta situação é apresentado na Figura 74.

A.3.1 Modelo Linear

Como a diferença entre o brilho e a saturação é uma representação aproximada da concentração de névoa, o seguinte modelo linear é definido:

$$d\left(x\right) = \theta_0 + \theta_1 v\left(x\right) + \theta_2 s\left(x\right) + \varepsilon\left(x\right), \tag{10}$$

onde x é uma posição da imagem, d é a profundidade da cena, v é o componente de brilho, s é o componente de saturação, θ_0, θ_1, θ_2 são coeficientes lineares desconhecidos, $\varepsilon\left(x\right)$ é o erro aleatório do modelo, e ε pode ser considerada uma imagem aleatória. É usada uma densidade Gaussiana para ε, com média 0 e σ^2 variável. De acordo com a propriedade da distribuição Gaussiana:

$$d\left(x\right) \sim p\left(d\left(x\right)|x, \theta_0, \theta_1, \theta_2, \sigma^2\right) = N(\theta_0 + \theta_1 v + \theta_2 s, \sigma^2).$$

Os coeficientes θ_0, θ_1, θ_2 e σ^2 podem ser determinados através de um método de aprendizado supervisionado. Os valores encontrados em [79] são $\theta_0 = 0.121779$, $\theta_1 = 0.959710$, $\theta_2 = -0.780245$ e $\sigma = 0.041337$.

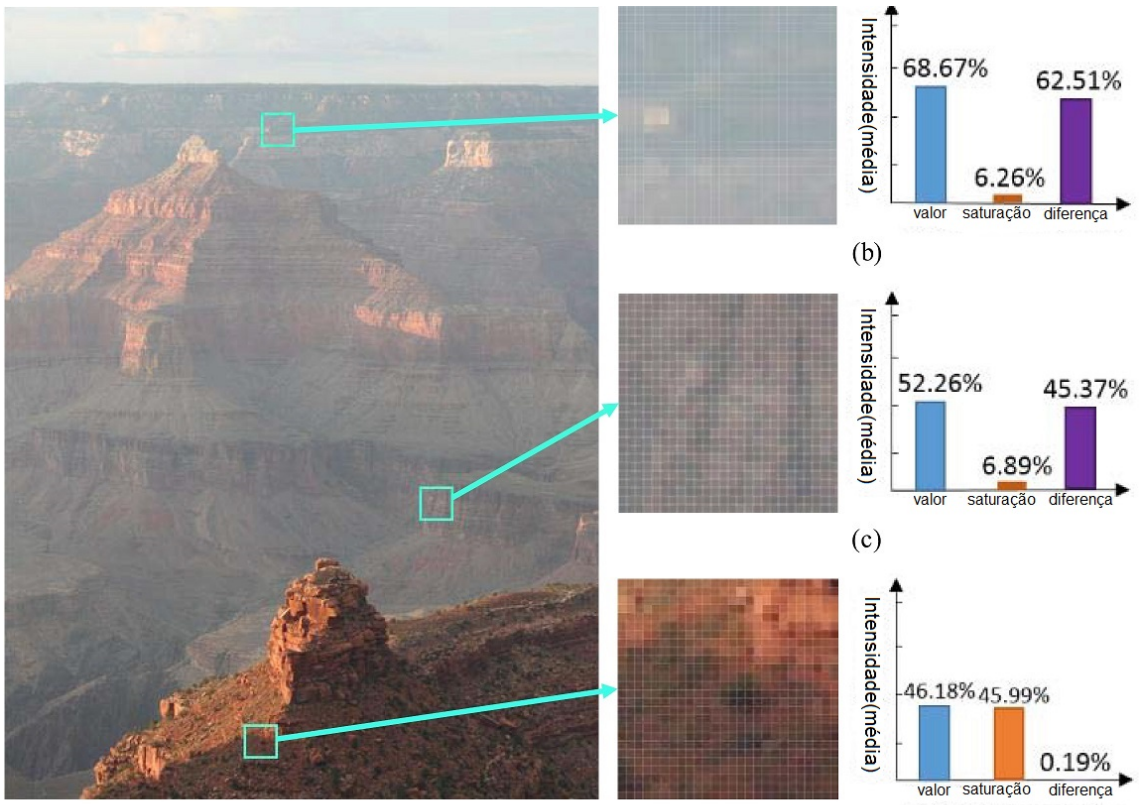


Figura 74: A concentração de névoa está positivamente correlacionada com a diferença entre o brilho e a saturação. (a) Uma imagem com névoa. (b) Um *patch* com névoa densa e o seu histograma. (c) Um *patch* com concentração moderada de névoa e seu histograma. (d) Um *patch* livre de névoa e seu histograma. Fonte: [79].

A.3.2 Estimativa da Profundidade

Com a estimativa dos coeficientes θ_0, θ_1 e θ_2 é possível determinar o mapa de profundidade de uma imagem com neblina através da equação 10. Esse modelo, porém, pode falhar em algumas situações particulares. Por exemplo, objetos brancos em uma imagem normalmente têm altos valores de brilho e baixos valores de saturação, e por isso são considerados pelo modelo como distantes. Esse erro na classificação pode provocar uma estimativa de profundidade imprecisa em alguns casos. Para contornar este problema é necessário considerar a vizinhança de cada pixel. Baseando-se na hipótese de que a profundidade e localmente constante, o mapa de profundidade é estimado como:

$$d_r(x) = \min_{y \in \Omega_r(x)} d(y),$$

onde $\Omega_r(x)$ é uma vizinhança de tamanho $r \times r$ com centro em x, e d_r é o mapa de profundidade com escala r. O mapa de profundidade resultante tende a ter um efeito de blocos, dependendo do tamanho da vizinhança, e por isso precisa ser refinado. Isso pode ser feito com o algoritmo guided filter [24], que é uma alternativa muito mais rápida ao soft matting [40].

A.3.3 Estimativa da luz Atmosférica

O método utilizado para estimar a luz atmosférica utilizado em [79] é similar ao apresentado em [23]. Com o mapa de profundidade já determinado, o primeiro passo é selecionar os top 0.1% pixels com a maior profundidade correspondente. Dentre estes pixels aquele que tem a maior intensidade na imagem degradada **I** corresponde à luz atmosférica.

A.3.4 Restauração da Cena

Com a profundidade da cena d é a luz atmosférica $\mathbf{L}_\infty$ conhecidas, é possível estimar a transmissão t de acordo com a equação 1 e recuperar a radiância da cena **J** com a equação 2. Por conveniência, a equação 2 pode ser reescrita como:

$$\mathbf{J}(x) = \frac{\mathbf{I}(x) - \mathbf{L}_\infty}{t(x)} + \mathbf{L}_\infty = \frac{\mathbf{I}(x) - \mathbf{L}_\infty}{e^{-\beta d(x)}} + \mathbf{L}_\infty.$$

Para evitar a geração de muito ruído, a transmissão $t(x)$ é restrita entre 0.1 e 0.9. Desta forma a expressão final para restauração da radiância da cena **J** é dada por:

$$\mathbf{J}(x) = \frac{\mathbf{I}(x) - \mathbf{L}_\infty}{\min\{\max\{e^{-\beta d(x)}, 0.1\}\}} + \mathbf{L}_\infty,$$

onde **J** é a cena livre de névoa que se deseja restaurar.

O coeficiente de espalhamento β representa a capacidade de uma unidade de volume de atmosfera de espalhar a luz em todas as direções. Em outras palavras, β determina

indiretamente a intensidade da operação de *dehazing*. O use de um valor pequeno de β leva a uma transmissão baixa, e o resultado correspondente contém névoa nas regiões mais distantes. O uso de um valor de β muito grande pode resultar em uma superestimação da transmissão. Desta forma, um valor de β moderato é necessário para o tratamento de imagens com regiões de névoa densa. Na maioria dos casos $\beta = 1.0$ é suficiente [79].

A.3.5 Resultados

Um exemplo de restauração baseada na atenuação de cor é apresentado na Figura 75. Segundo os seus autores, o método proposto em [79] é superior a todos os outros métodos de *dehazing* baseados em uma única imagem disponíveis na época, incluindo os métodos baseados no contraste [68] e na dark channel prior [23]. Comparados aos métodos existentes, a restauração baseada na atenuação de cor é livre de supersaturação, principalmente nas regiões da imagem que contém objetos brancos.

A.4 Disparidade de Matiz

Em [2] é proposto um método de *dehazing* baseado na disparidade de matiz. O método proposto busca generalizar a abordagem baseada na dark channel prior [23].

A.4.1 Detecção de Névoa

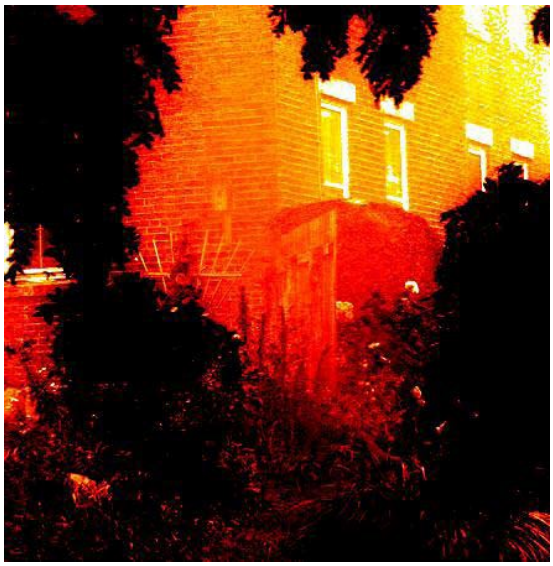
O algoritmo de detecção de névoa proposto em [2] é uma operação pixel a pixel, desenvolvida com base na dark channel prior. O primeiro passo deste método é a criação de uma imagem *semi-inversa* $\mathbf{I}_{si}(x) = \left[I^r_{si}, I^g_{si}, I^b_{si}\right]$. Esta imagem pode ser obtida através substituição dos valores RGB de cada pixel x em uma operação de máximo entre o valor inicial de cada canal de cor e o seu inverso:

$$
\begin{aligned}
I^r_{si}(x) &= \max[I^r(x), 1 - I^r(x)] \\
I^g_{si}(x) &= \max[I^g(x), 1 - I^g(x)] \\
I^b_{si}(x) &= \max[I^b(x), 1 - I^b(x)]
\end{aligned}
$$

onde $I^r(x)$, $I^g(x)$ e $I^b(x)$ representam os canais RGB do pixel x. Como esta operação mapeia todos os pixels da imagem semi-inversa $\mathbf{I}_{si}$ no intervalo $[0.5, 1]$, a renormalização dos valores é necessária.

A razão para a disparidade de matiz são as características das imagens descritas pela dark channel prior. Nas áreas livres de névoa pelo menos um canal é caracterizado por valores pequenos, que serão substituídos pelo seu inverso pela operação. Nas regiões de céu ou com névoa todos os canais possuem valores altos, e por isso a operação de máximo irá resultar nos mesmos valores da imagem original. Desta forma, através da comparação direta entre valores de matiz da imagem inversa e da imagem original, pode-

Imagem degradada por névoa



Mapa de profundidade

Mapa de transmissão

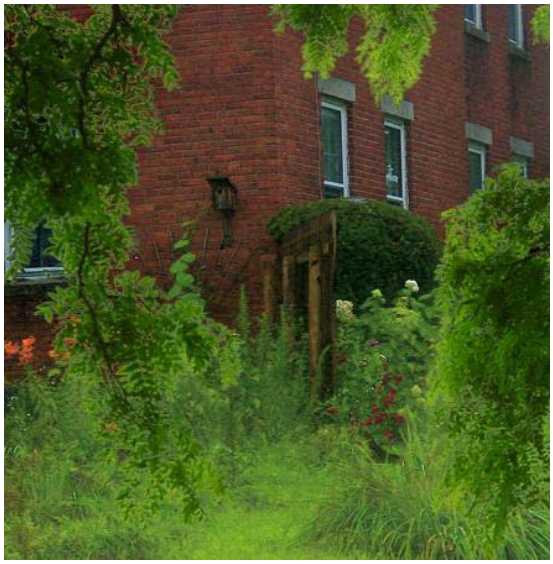


Imagem restaurada

Figura 75: Restauração por Atenuação de Cor. Fonte: [79].

se determinar os pixels que precisam ser restaurados mantendo-se uma aparência de cor similar à original.

A operação de semi-inversão produz uma imagem onde as áreas com névoa são apresentadas com contraste aumentado, enquanto as áreas não afetadas aparecem como o inverso da imagem original. Para se identificar as regiões afetadas pela névoa, é calculada a diferença entre os canais de matiz da imagem original $\mathbf{I}$ e da imagem inversa $\mathbf{I}_{si}$. Os pixels onde esta diferença é menor que um limite predefinido τ são marcados como pixels com névoa. O valor de τ facilita a seleção dos pixels que apresentam um aspecto similar tanto na imagem original quanto na imagem semi-inversa. O valor padrão utilizado em [2] é $\tau = 10°$. Em [2] a imagem é convertida para o espaço de cor CIE $L^*c^*h^*$, onde a informação de matiz é representada pelo canal h^*. Alguns exemplos de detecção de névoa por disparidade de matiz são apresentados na Figura 76.

Segundos os autores, esta técnica simples é capaz de estimar as regiões afetadas por névoa com precisão aceitável. Para checar a validade desta observação os autores coletaram mais de 2400 imagens de ambientes ao ar livre da internet, todas capturadas durante o dia, e as dividiram em três categorias: imagens sem céu livres de névoa, imagens de céu e imagens com névoa. A variação de matiz destas imagens foi avaliada utilizando a estratégia proposta. A conclusão principal foi que as imagens livres de neblina são caracterizadas por variações significantes de matiz na grande maioria dos pixels, enquanto nas outras duas categorias a variação é consideravelmente menor.

A.4.2 Estimativa da Luz Atmosférica

Um método para determinação da luz ambiente é proposto em [2]. O primeiro passo deste método é mascarar as regiões da imagem onde a névoa é mais espessa, o que é feito com o mesmo método utilizado para determinar as regiões afetadas por névoa, porém com a intensidade da imagem semi-inversa aumentada por um fator ξ (com um valor padrão de $\xi = 0.3$). Nas imagens onde o céu está presente, a máscara resultante é na maior parte composta pela região do céu, o que reduz o espaço de busca. A extração da luz atmosférica $\mathbf{L}_\infty$ é realizada através da determinação do pixel mais brilhante na região positiva (não mascarada) da imagem. O valor de $\mathbf{L}_\infty$ é extraído do pixel correspondente na imagem original $\mathbf{I}$.

A.4.3 Restauração em Camadas

Em [2] é proposto um método de restauração baseado em camadas, que busca preservar a maior quantidade de detalhes possível mantendo uma velocidade de execução suficiente para aplicação em tempo real. O algoritmo é inicializado com a criação de várias novas imagens $\mathbf{I}_i$, com $i \in [1, k]$ e k camadas, nas quais se remove uma porção

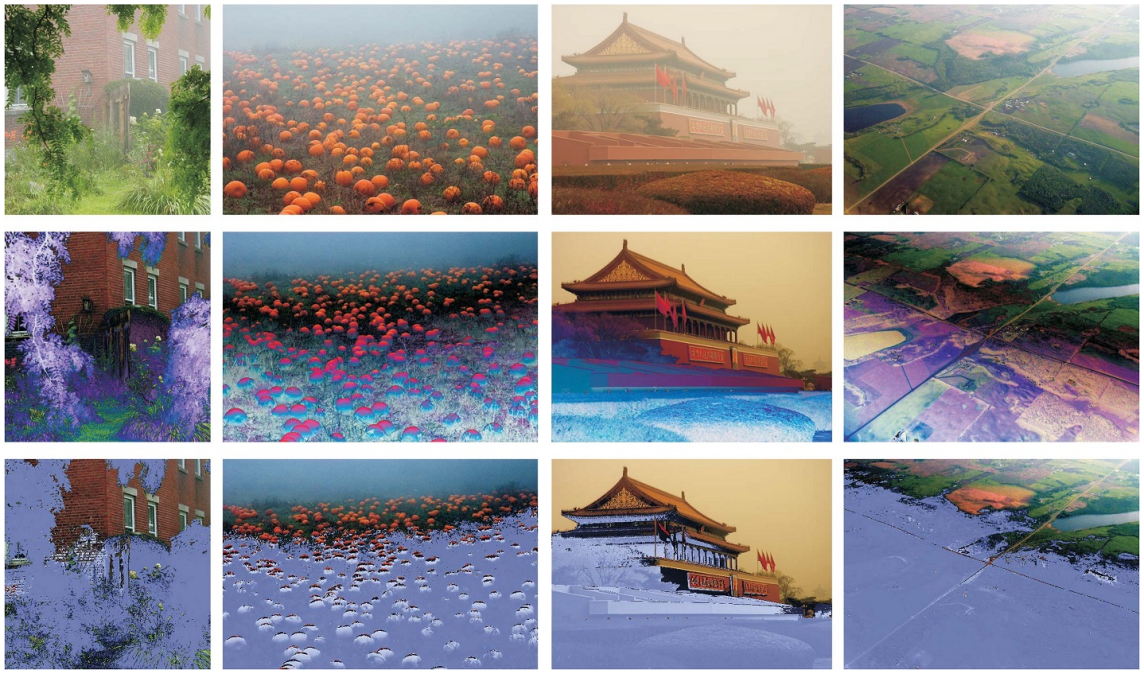


Figura 76: Detecção de névoa por disparidade de matiz. A primeira linha mostra a imagem original degradada por névoa $\mathbf{I}$. A segunda linha mostra a imagem semi-inversa $\mathbf{I}_{si}$. Na terceira linha, os pixels identificados como livres de névoa são marcados com uma máscara azul. Nestas regiões, a intensidade da cor azul é proporcional à disparidade de matiz. Fonte: [2].

crescente da constante de luz atmosférica $\mathbf{L}_\infty$ da imagem inicial $\mathbf{I}$:

$$\mathbf{I}_i = \mathbf{I} - c_i \cdot \mathbf{L}_\infty,$$

com um fator de contribuição da luz ambiente c_i que aumenta iterativamente. O parâmetro c_i é um valor escalar no intervalo $[0, 1]$, cujo valor depende do número de camadas.

Após a aplicação da operação de detecção de névoa em $\mathbf{I}_i$, apenas os pixels com um valor de disparidade de matiz suficientemente baixo são marcados como parte da camada $\mathcal{L}_i$. Na ausência de informação sobre a geometria da cena, a discretização da imagem em k camadas distintas permite que sejam estimados os valores de c_i que correspondem às camadas de profundidade mais dominantes da cena. Por exemplo, quando a cena contém dois objetos localizados em profundidades diferentes, o mapa de transmissão será caracterizado por dois valores dominantes (já que a luz ambiente está correlacionada com a distância). Por fim, estas camadas são unidas em uma única imagem composta, livre de névoa. Para suavizar a transição entre as diferentes camadas, o número de camadas extraídas k precisa ser de pelo menos 5, ou mais. Em [2] são usadas 5 camadas para geração dos resultados, com os valores de c_i: [0.2;0.4;0.6;0.8;1]. Para se obter uma imagem restaurada $\mathbf{I}_0$ as camadas são misturadas em uma ordem decrescente, de acordo com a contribuição da luz ambiente. Para evitar o surgimento de artefatos indesejáveis, provocados por pequenas descontinuidades, cada camada contribui para a próxima com uma

pequena porcentagem, de acordo com a equação:

$$\mathbf{I}_0 = \sum_{i=1}^{k} \mathcal{X}_i \mathcal{L}_i,$$

onde o parâmetro escalar $\mathcal{X}_i$ é o peso da contribuição dos pixels da camada, que aumenta exponencialmente de acordo com o número da camada.

A.4.4 Resultados

Segundos os autores, o método de estimativa apresentado é capaz de gerar máscaras de detecção de névoa mais detalhadas que as geradas por métodos baseados em *patch*, como [23]. Os autores também consideram que a restauração em camadas apresenta resultados tão bons quanto ou melhores que as técnicas de *dehazing* concorrentes, sendo capaz de restaurar a cena original, mantendo até mesmo pequenos detalhes, além de preservar a cor dos objetos da cena. A principal vantagem deste método, porém, está no seu tempo de execução. Uma implementação em CPU leva aproximadamente 0.013 segundos para processar uma imagem de tamanho 600×800, o que permite a aplicação do método em tempo real. A restauração baseada na disparidade de matiz é comparada com outros métodos de remoção de névoa na figura 77.

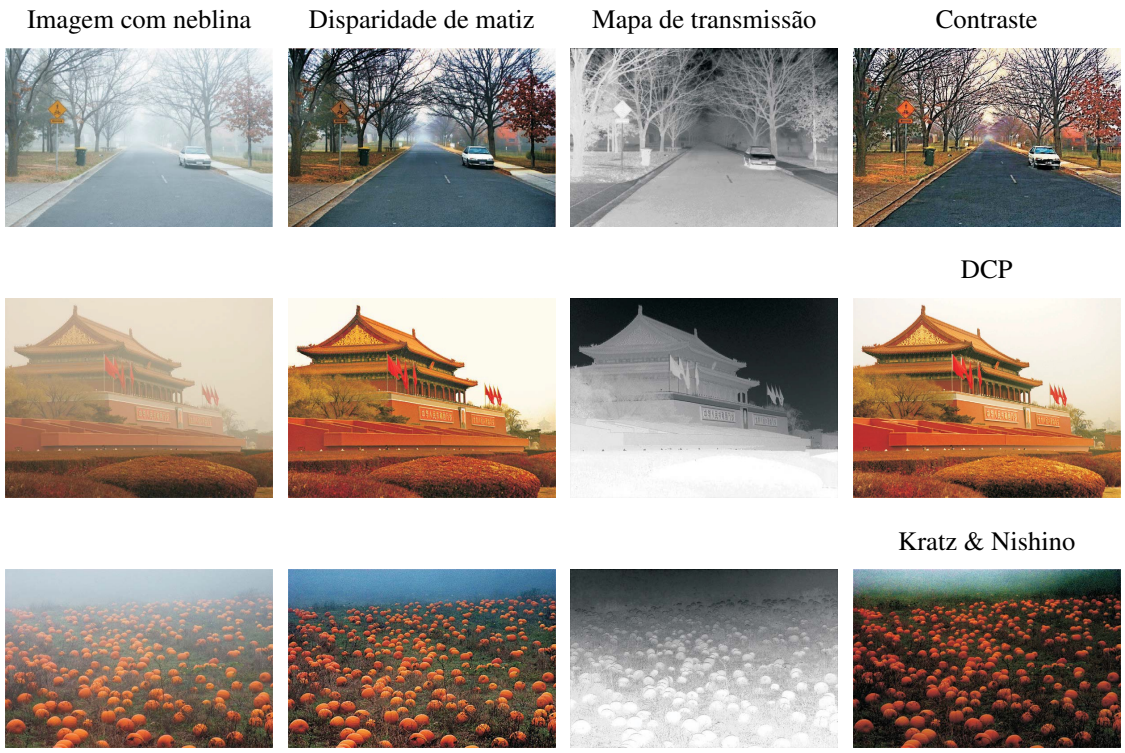


Figura 77: Da esquerda para direita: a imagem degradada, restauração por disparidade de matiz [2], estimativa de transmissão por disparidade de matiz, e os resultados correspondentes obtidos pela restauração por contraste [68], DCP [23] e pelo método proposto em [35]. Fonte: [2]

APÊNDICE B MÉTODOS DE RESTAURAÇÃO DE IMAGENS SUBAQUÁTICAS

B.1 Diferença Entre os Canais de Cor

Em [6] é proposto um método de estimativa da transmissão que explora o fato de que, no meio aquático, a taxa de atenuação do canal de cor vermelho é maior do que a dos canais verde e azul.

B.1.1 Estimativa da Transmissão

O método proposto realiza a estimativa da transmissão através da comparação da intensidade máxima do canal de cor vermelho com a intensidade máxima dos canais verde e azul, dentro de um pequeno *patch* da imagem. O primeiro passo é encontrar a diferença entre a intensidade máxima do canal vermelho e a intensidade máxima dos outros canais:

$$D\left(x\right) = \max_{x\in\Omega, c\in r} I_c\left(x\right) - \max_{x\in\Omega, c\in\{b,g\}} I_c\left(x\right).$$

Na equação acima, $I_c\left(x\right)$ é um pixel x em um canal de cor $c \in \{r, g, b\}$ da imagem, e Ω se refere a um *patch* da imagem. Para se encontrar a transmissão estimada $\tilde{t}$, os valores de D são ajustados de modo que a maior diferença entre os canais de cor, que representa o ponto da imagem mais próximo do observador, tenha o valor de 1:

$$\tilde{t}\left(x\right) = D\left(x\right) + \left(1 - \max_{x} D\left(x\right)\right).$$

Como o cálculo é realizado por *patches*, o resultado desta estimativa inicial é um mapa de transmissão grosseiro, de baixa resolução. Assim como em [23], esta estimativa inicial é refinada com o algoritmo soft matting [40], o que resulta em um mapa de transmissão de melhor qualidade, mais fiel às estruturas da imagem.

Finalmente, para se manter uma aparência mais realista nos resultados e evitar a acentuação do ruído nas regiões onde a degradação é mais intensa, utiliza-se um parâmetro ω

para restringir a estimativa da transmissão a um limite inferior:

$$\tilde{t} = \begin{cases} \tilde{t} & para\, \tilde{t} \geq \omega \\ \omega & para\, \tilde{t} < \omega \end{cases}.$$

Em [6], são utilizados valores de ω entre 0.75 e 0.95. O tamanho de *patch* Ω varia entre 20 e 60, dependendo do tamanho da imagem (imagens maiores requerem um tamanho de *patch* maior).

B.1.2 Estimativa da Luz Ambiente

Para se estimar a luz ambiente $\mathbf{L}_\infty$, encontra-se o pixel com a menor transmissão estimada, que representa o ponto da imagem mais distante da câmera. A luz ambiente estimada é dada pela cor da imagem nesta posição:

$$\tilde{\mathbf{L}_\infty} = \mathbf{I}\left(\arg\min_x \tilde{t}\,(x)\right).$$

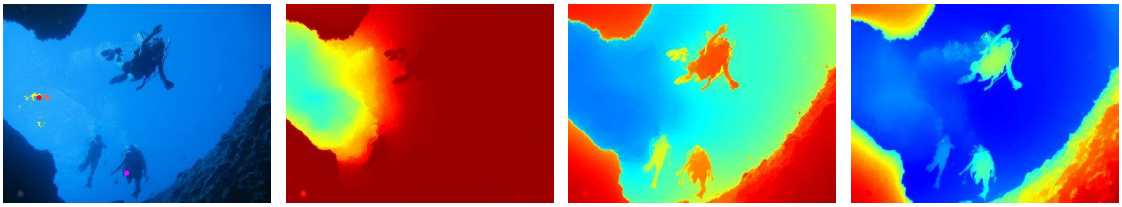
Um requisito importante para a estimativa da luz ambiente é que a imagem contenha uma região onde a névoa é completamente opaca, ou seja, o componente direto é zero. Isto normalmente ocorre acima do horizonte, onde apenas a coluna de água é visível.

B.1.3 Resultados

Em [6] o método proposto é comparado com técnicas de regularização de histograma, frequentemente utilizadas para aumentar o contraste de imagens. Segundo os autores, o método apresentado é superior, sendo capaz de recuperar uma maior quantidade de detalhes da imagem. O método também é testado em [13], onde em alguns casos apresenta uma estimativa incorreta da luz ambiente, e em outros resulta em uma superestimação da transmissão.

B.2 Underwater Dark Channel Prior

A dark channel prior é baseada na suposição de que nos pontos onde a profundidade tende ao infinito, e consequentemente existe muita concentração de névoa, o valor mínimo entre os três canais de cor é alto. Esta suposição, apesar de geralmente válida para imagens degradadas por neblina ou fenômenos similares, não é verdadeira para imagens capturas em meios subaquáticos. Devido à absorção desigual dos diferentes comprimentos de onda, o canal de cor vermelho tende a apresentar valores baixos em diversas situações, principalmente nas regiões da imagem onde a contribuição do componente direto é muito pequena, e por isso normalmente domina o dark channel da imagem. Isso resulta na superestimação da transmissão em muitos casos, como mostrado em [13]. Tendo em vista este problema, em [13] é proposta a *Underwater Dark Channel Prior* (UDCP), uma



Entrada | DCP [23] | UDCP [13] | Diferença entre os canais de cor [6]

Figura 78: Estimativa de transmissão de uma imagem subaquática por diferentes métodos. Fonte: [13].

variação da DCP tradicional que leva em conta as características da propagação da luz em meio aquático.

A UDCP faz uma estimativa da transmissão com base no mínimo entre os canais de cor verde e azul, desconsiderando o canal de cor vermelho:

$$J^{UDCP}(x) = \min_{y \in \Omega(x)} \left(\min_{c \in G,B} J^c(y) \right).$$

A constante de luz ambiente $\mathbf{L}_\infty$ é estimada como equivalente à cor do ponto da imagem com o maior valor no underwater dark channel. A transmissão $\tilde{t}$ é estimada como:

$$\tilde{t}(x) = 1 - \min_{y \in \Omega(x)} \left(\min_{c \in G,B} \frac{I^c(y)}{L_\infty^c} \right).$$

Esta estimativa da transmissão é realizada por *patches*, o que resulta em um efeito de blocos. Como proposto em [23], estes efeitos podem ser eliminados através da aplicação de um algoritmo de matting, como soft matting [40] ou guided filter [24].

Em [13], a UDCP é comparada com outros métodos de restauração de imagens subaquáticas, incluindo a aplicação direta da DCP e o método proposto em [6]. A UDCP foi capaz de estimar corretamente a luz ambiente em todos os casos de teste, enquanto os outros métodos falharam em alguns casos. A estimativa de transmissão através da UDCP também apresentou resultados plausíveis em todos os casos de teste apresentados, enquanto a DCP e o método de [6] falharam completamente em alguns casos. Uma comparação entre os diferentes métodos de estimativa de transmissão é apresentada na Figura 78.

B.3 Dark Channel Prior com Inversão do Canal Vermelho

Em [8] é proposta uma outra adaptação da Dark Channel Prior a ambientes subaquáticos. O método proposto é baseado na suposição de que a presença da cor vermelha indica uma alta transmissão. Esta suposição é válida apenas para imagens subaquáticas, já que em imagens capturadas fora da água a luz ambiente costuma ter a cor branca, o que resulta

em altos valores no canal vermelho quando a transmissão é baixa.

No método proposto, o cálculo do Dark Channel depende da cor predominante da água, que pode ser determinada automaticamente através de uma análise de histograma. O Dark Channel J^w é dado por:

$$J^w(x) = \begin{cases} \min\limits_{y\in\Omega(x)} \left(\min\left(J^{invR}(y), J^B(y)\right)\right) & \text{para azul} \\ \min\limits_{y\in\Omega(x)} \left(\min\left(J^{invR}(y), J^G(y)\right)\right) & \text{para verde} \\ \min\limits_{y\in\Omega(x)} \left(\min\left(J^{invR}(y), J^G(y), J^B(y)\right)\right) & \text{para ciano} \end{cases}$$

onde o canal $invR$ é dado por $1 - R$. Com base em J^w, pode-se estimar a transmissão $\tilde{t}$ e recuperar a radiância **J** através do mesmo procedimento utilizado em [23].

B.4 Restauração por Veil Difference Prior

Em [9] é proposto um método para restauração de imagens capturadas em meios participativos gerais. Ao contrário dos métodos de restauração anteriores, que foram desenvolvidos especialmente para restaurar imagens capturadas em um meio específico, o método proposto pode ser aplicado a qualquer tipo de meio participativo, incluindo neblina, tempestades de areia e água. No método proposto, a transmissão é obtida através da união de duas estimativas diferentes: uma baseada no contraste, a outra baseada na *Veil Difference Prior*.

B.4.1 Estimativa de Transmissão pela Veil Difference Prior

Em [9], é proposta a suposição de que a cor da imagem tende a ser mais próxima da luz ambiente quando a imagem é afetada por turbidez.

Assumindo-se que todos os pixels de um *patch* $\Omega(x)$, com centro em x, tem a mesma distância em relação à câmera, a transmissão é definida como:

$$t_v(x) = \frac{\max\limits_{c\in\{r,g,b\}} \left(\max\limits_{y\in\Omega(x)} \left(|I^c(y) - L^c_\infty|\right)\right)}{\max\limits_{c\in\{r,g,b\}} \left(\max\limits_{y\in\Omega(x)} \left(|J^c(y) - L^c_\infty|\right)\right)}, \tag{11}$$

onde a transmissão $t_v(x)$ pode ser interpretada como a perda de distinção entre a imagem limpa **J** e a imagem turva **I** provocada pela luz ambiente $\mathbf{L}_\infty$. Escolhe-se o pixel do canal de cor onde existe a maior diferença, já que é ele que contém a maior quantidade de informação. Esta expressão, porém, não pode ser utilizada diretamente, já que a imagem limpa **J** não é conhecida.

Segundo os autores, imagens livres de névoa tendem a apresentar uma diferença significativa para a luz ambiente em pelo menos um dos canais de cores. Com base nesta

observação, a *Veil Difference Prior* é definida como:

$$\max_{c\in\{r,g,b\}}\left(\max_{y\in\Omega(x)}\left(\left|J^{c}\left(y\right)-L_{\infty}^{c}\right|\right)\right)=\max_{c\in\{r,g,b\}}\left(\max\left(1-L_{\infty}^{c},L_{\infty}^{c}\right)\right). \tag{12}$$

De acordo com a *Veil Difference Prior*, a equação 11 pode ser reescrita como:

$$t_{v}\left(x\right)=\frac{\max\limits_{c\in\{r,g,b\}}\left(\max\limits_{y\in\Omega(x)}\left(\left|I^{c}\left(y\right)-L_{\infty}^{c}\right|\right)\right)}{\max\limits_{c\in\{r,g,b\}}\left(\max\left(1-L_{\infty}^{c},L_{\infty}^{c}\right)\right)}. \tag{13}$$

Segundo [9], a *Veil Difference Prior* pode ser considerada uma generalização da Dark Channel Prior [23]. Considerando-se a luz atmosférica como branco puro, $\mathbf{L}_{\infty}=[1,1,1]$, a equação 12 se torna $\max\left(1-J^{c}\left(x\right)\right)=1$ para $c\in\{r,g,b\}$, que é equivalente à Dark Channel Prior.

B.4.2 Estimativa de Transmissão por Contraste

Segundo [23], os histogramas das imagens onde existe muita turbidez tendem a apresentar uma distribuição de valores em um intervalo reduzido, com uma grande concentração ao redor da cor da luz ambiente. Isto significa uma redução no contraste global da imagem. Com base nesta observação, supõe-se que o contraste de um *patch* da imagem reduz proporcionalmente ao aumento da turbidez.

A estimativa de transmissão por contraste é definida por:

$$t_{c}\left(x\right)=\frac{\max\limits_{c\in\{r,g,b\}}\left(\max\limits_{y\in\Omega(x)}\left(I^{c}\left(y\right)\right)-\min\limits_{y\in\Omega(x)}\left(I^{c}\left(y\right)\right)\right)}{\max\limits_{c\in\{r,g,b\}}\left(\max\limits_{y\in\Omega(x)}\left(J^{c}\left(y\right)\right)-\min\limits_{y\in\Omega(x)}\left(J^{c}\left(y\right)\right)\right)}, \tag{14}$$

esta relação representa a perda de contraste provocada pela turbidez. Como o contraste da imagem limpa não é conhecido, pode-se supor que ele tem o maior valor possível:

$$\max_{c\in\{r,g,b\}}\left(\max_{y\in\Omega(x)}\left(J^{c}\left(y\right)\right)-\min_{y\in\Omega(x)}\left(J^{c}\left(y\right)\right)\right)=1. \tag{15}$$

Isto é verdadeiro para alguns *patches* da imagem. Como este não é o único indicador de transmissão, esta suposição é razoável neste caso [9]. Desta forma, a transmissão por contraste é calculada como:

$$t_{c}\left(x\right)=\max_{c\in\{r,g,b\}}\left(\max_{y\in\Omega(x)}\left(I^{c}\left(y\right)\right)-\min_{y\in\Omega(x)}\left(I^{c}\left(y\right)\right)\right). \tag{16}$$

B.4.3 Estimativa de Transmissão Final

O uso de um único indicador de transmissão, como a cor ou o contraste, é *decisivo* mas não *suficiente* [9]. Por exemplo, não é possível saber se um pixel possui uma determinada cor por influência da luz ambiente ou se aquela é a cor original da cena, da mesma forma que não se pode saber se a ausência de contraste em um *patch* é característica da cena ou foi provocada pela atenuação do meio.

Já quando um *patch* apresenta uma alta transmissão, os autores acreditam que esta informação está relacionada com o sinal da imagem, e não com a luz ambiente. Desta forma, considera-se que uma estimativa alta de transmissão tem uma menor probabilidade de estar errada. Assim, a transmissão final é calculada como o máximo entre a transmissão da *Veil Difference Prior* e a transmissão do contraste:

$$t\left(x\right) = \max\left(t_v\left(x\right), t_c\left(x\right)\right).$$

Nos testes realizados, as duas metodologias de estimativa apresentaram contribuição significativa para a transmissão final. A estimativa pela *Veil Difference Prior* apresentou valores maiores nas regiões mais próximas da imagem, enquanto a transmissão nas regiões intermediárias foi dominada pela estimativa por contraste.

B.4.4 Restauração da Imagem

O primeiro passo para a restauração da imagem é a estimativa da luz ambiente $\mathbf{L}_\infty$. Em [9], a luz ambiente é estimada através do algoritmo *shades-of-gray* [18]. As imagens capturadas em meios participativos costumam apresentar ruído, que pode prejudicar a estimativa da transmissão. Por esta razão, antes de se realizar a estimativa da transmissão a imagem é filtrada com o algoritmo filtro bilateral [70], que tende a eliminar pontos de alta frequência isolados sem afetar as bordas.

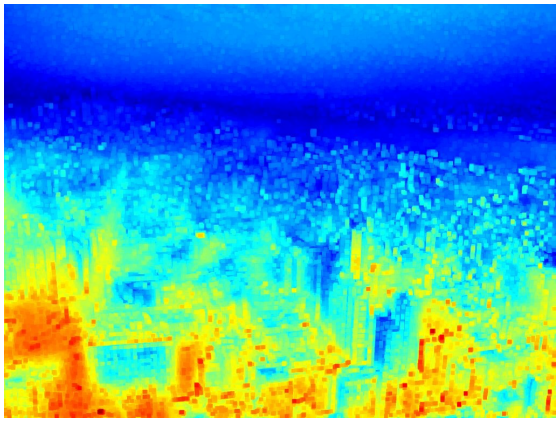
Com a luz ambiente $\mathbf{L}_\infty$ conhecida, é possível estimar o mapa de transmissão através do contraste e da *Veil Difference Prior*. Como em ambos os métodos a estimativa é realizada por *patches*, o mapa de transmissão resultante tende a apresentar artefatos de bloco, e por isso é refinado com o algoritmo soft matting [40], como proposto em [23].

Para se obter a imagem restaurada, isola-se a refletância $\rho\left(x\right)$ na equação 2:

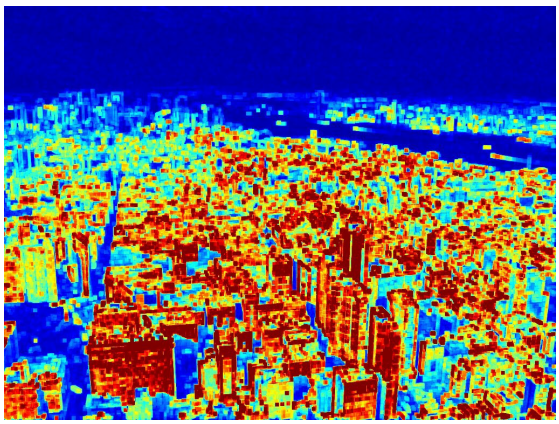
$$\rho\left(x\right) = \frac{I^c\left(x\right) - L^c_\infty + L^c_\infty t\left(x\right)}{\max\left(t^c_0, L^c_\infty t\left(x\right)\right)},$$

onde o parâmetro t^c_0 é a transmissão mínima. Este parâmetro é utilizado para evitar a recuperação de ruído nas regiões da imagem onde praticamente não existe nenhuma informação a ser recuperada. São utilizados três parâmetros de mínimo, t^r_0, t^g_0 e t^b_0, normalmente definidos entre 0.1 e 0.2.

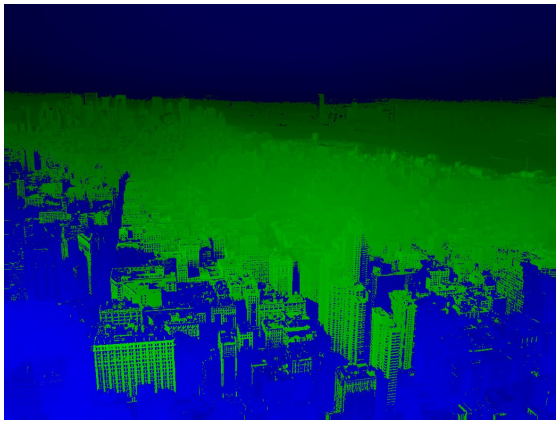
(a) Imagem de entrada



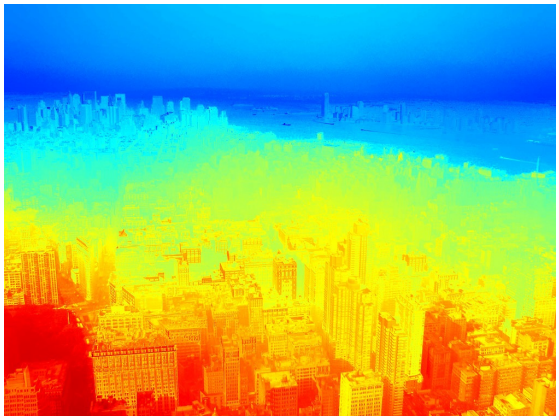
(b) $t_v\,(x)$



(c) $t_c\,(x)$



(d) Contribuições



(e) Transmissão refinada

(f) Imagem restaurada

Figura 79: Resultados do método proposto em [9]. (a) A imagem de entrada. (b) Transmissão estimada por *Veil Difference Prior*. (c) Transmissão estimada pelo contraste. (d) Contribuição de cada estimativa de transmissão para o resultado final, onde verde é $t_c\,(x)$ e azul $t_v\,(x)$. (e) Transmissão refinada por soft matting. (f) A imagem restaurada. Fonte: [9].

B.4.5 Resultados

O método proposto foi comparado com outros métodos de restauração, utilizando imagens subaquáticas e cenas externas capturas em dias de neblina, obtendo desempenho

de estado da arte na maioria dos casos de teste [9]. Uma vantagem do método proposto em relação a outros métodos, como [13] e [8], é que ele é capaz de recuperar a cor original da cena sem a necessidade da aplicação de técnicas adicionais, como balanço de branco. Um exemplo de restauração por *Veil Difference Prior* é apresentado na Figura 79.