UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

UMA ANÁLISE DO USO DE INFORMAÇÕES MULTIESCALA NO MAPEAMENTO DA PSNR PARA PONTUAÇÃO PERCEPTUAL

LUAN ASSIS GONÇALVES

DM: 41/2019

UFPA / ITEC / PPGEE
Campus Universitário do Pará
BELÉM - PARÁ
2019

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

LUAN ASSIS GONÇALVES

UMA ANÁLISE DO USO DE INFORMAÇÕES MULTIESCALA NO MAPEAMENTO DA PSNR PARA PONTUAÇÃO PERCEPTUAL

Dissertação submetida à Banca Examinadora do Programa de Pós-Graduação em Engenharia Elétrica da UFPA para obtenção do grau de Mestre em Engenharia Elétrica.

UFPA / ITEC / PPGEE
Campus Universitário do Pará
BELÉM - PARÁ
2019

UMA ANÁLISE DO USO DE INFORMAÇÕES MULTIESCALA NO MAPEAMENTO DA PSNR PARA PONTUAÇÃO PERCEPTUAL

Este trabalho foi julgado adequado em 18/11/2019 para a obtenção do Grau de Mestre em Engenharia Elétrica, aprovado em sua forma final pela banca examinadora que atribui o conceito _____.

Prof. Dr. Fabrício José Brito Barros
ORIENTADOR

Prof. Dr. Ronaldo de Freitas Zampolo
COORIENTADOR

Prof. Dr. Aldebaro Barreto da Rocha Klautau Júnior.
MEMBRO DA BANCA EXAMINADORA

Prof. Dr. Bianchi Serique Meiguins.
MEMBRO DA BANCA EXAMINADORA

Prof(a). Dr(a). Maria Emília de Lima Tostes
COORDENADORA DO PROGRAMA DE
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Aos anjos em minha vida

***Helena** e **Antônio**.*

Agradecimentos

Agradeço a Deus pela minha vida, pela família que tenho, por me dar forças em momentos de fraqueza e por permitir a realização de sonhos como esta etapa que se encerra.

Agradeço aos meus pais, Antônio e Helena, por serem meus maiores apoiadores e por acreditarem em mim mais do que eu mesmo acredito. Pelos ensinamentos diretos e indiretos, de onde pude formular minhas essências moral e intelectual. Por me apoiarem incondicionalmente, mesmo que significasse a anulação de si mesmos.

Agradeço aos meus irmãos e amigos Camilo e Yuri, por serem meus companheiros de jornadas e se fazerem presentes quando mais precisei.

Agradeço aos meus amigos de engenharia: Wederson Medeiros, Raphael Navegantes, Leidiane Castro, José Fiel, Charles Ferreira e outros conhecidos no caminho, por terem colaborado de alguma forma, pela amizade durante esta caminhada.

Agradeço, especialmente, a minha namorada Adriane Costa e família: Joaquim Costa, Anete Costa, Raimundo Rosa, Terezinha Rosa, Susete Rosa, Juliete Rosa e os demais, por todo o carinho, consideração,

Por fim, agradeço, profundamente, aos professores Evaldo Gonçalves Pelaes, Fabrício José Brito Barros e Ronaldo de Freitas Zampolo. Meu eterno agradecimento pelos ensinamentos, apoio e companhia durante essa jornada.

“Não é o conhecimento, mas o ato de aprender, não a posse mas o ato de chegar lá, que concede a maior satisfação.”
Carl Friedrich Gauss.

Resumo

A previsão da qualidade visual é crucial nos sistemas de imagem e vídeo. Métricas de qualidade de imagem com base no erro quadrático médio prevalecem em diversas aplicações, apesar de apresentarem baixa correlação com a percepção visual humana, devido à sua simplicidade matemática. As últimas realizações na área sustentam que o uso de redes neurais convolucionais (CNN) para avaliar a qualidade visual perceptiva é uma tendência clara. Resultados em outras aplicações, como detecção de desfoque e remoção de chuva, indicam que a combinação de informações de diferentes escalas melhora o desempenho da CNN. No entanto, até onde sabemos, a melhor maneira de incorporar informações em várias escalas na caracterização da qualidade visual ainda é uma questão em aberto. Assim, neste trabalho, investigamos a influência do uso de informações em várias escalas para prever a qualidade perceptual de imagens. Especificamente, propomos uma rede densa de fluxo único que estima um parâmetro espacialmente variável da função logística usada para mapear valores de métricas objetivas de qualidade visual para as notas subjetivas de qualidade visual através da imagem de referência. O método proposto alcançou uma redução de 36,37% e 69,45% para o número de parâmetros e de operações de ponto flutuante por segundo, respectivamente, e seu desempenho é comparado com o estado da arte, usando um banco de dados de imagens disponível publicamente.

Palavras-chave:Redes neurais convolucionais, *Differential mean opinion score*, informações multiescala, relação sinal-ruído de pico, avaliação de qualidade visual.

Abstract

The prediction of visual quality is crucial in image and video systems. For this task, image quality metrics based on the mean squared error prevail in the field, due to their mathematical straightforwardness, even though they do not correlate well with the visual human perception. Latest achievements in the area support that the use of convolutional neural networks (CNN) to assess perceptual visual quality is a clear trend. Results in other applications, like blur detection and de-raining, indicate the combination of information from different scales improves the CNN performance. However, to the best of our knowledge, the best way to embody multi-scale information in visual quality characterization is still an open issue. Thus, in this work, we investigate the influence of using multi-scale information to predict the perceptual image quality. Specifically, we propose a single-stream dense network that estimates a spatially-varying parameter of a logistic function used to map values of a objective visual quality metric to subjective visual quality scores through the reference image. The proposed method achieved a reduction of 36.37% and 69.45% for the number of parameters and floating-point operations per second, respectively, and its performance is compared with a competing state-of-the-art approach by using a public image database.

Keywords: Convolution neural network, differential mean opinion score, multi-scale information, peak signal-to-noise ratio, visual quality assessment

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

CNN Convolutional neural networks

DMOS Differential mean opinion score

FC Fully connected

FLOPs Floating-point operations per second

FSIM Feature-similarity

HVS Human visual system

HaarPSI Haar wavelet-based perceptual similarity index

IQM Image quality assessment

JP2K Joint photographics experts group 2000

JPEG Joint photographics experts group

LCC linear correlation coefficient

LIVE Laboratory for image & video engineering

LR Learning rate

MDN multi-stream dense network

MLP Multi-layer perceptron

MN Multi-stream network

MOS Mean option score

MS-SIM Multiscale structural similarity

MSE Mean squared error

PSNR Peak signal-to-noise ratio

RNA Redes neurais artificiais

SDN Single-stream dense network

SNR Signal-to-noise ratio

SROCC Spearman rank ordered correlation coefficient

SSIM Structural similarity index

paPSNR Perceptual adapted peak signal-to-noise ratio

Sumário

Capítulo 1

Introdução

1.1 Contextualização

1.1.1 Justificativa

A predição da qualidade visual é uma característica estratégica para sistemas de processamento de informações visuais (imagens e vídeos). Testes psicofísicos continuam sendo a abordagem mais confiável para se obter a qualidade visual. Esse procedimento, no entanto, é complexo, caro e impraticável em muitas situações, impulsionando o desenvolvimento de métodos e métricas para caracterização da qualidade visual.

De acordo com a quantidade de informações disponíveis da imagem de referência, as métricas de qualidade visual (*visual quality metrics*, VQM) podem ser classificadas como de referência completa (a imagem de referência está disponível), referência reduzida (apenas algumas informações da imagem de referência estão disponíveis) e sem referência (a imagem de referência é completamente desconhecida) [9]. A simplicidade do erro médio quadrático (*mean squared error*, MSE) motivou sua adoção (e de métricas derivadas do MSE) em várias situações, porém o MSE não possui um alto grau de correlação com a percepção visual humana [10].

Os avanços na microeletrônica e no processamento de sinais favoreceram o desenvolvimento de dispositivos de exibição e imageamento acessíveis. Paralelamente, aplicações complexas de vídeo (principalmente serviços de *streaming* e comunicações pessoais) levaram a considerar modelos matemáticos que tentam imitar, pelo menos parcialmente, o sistema visual humano.

Embora as realizações notáveis nas últimas duas décadas de pesquisa em avaliação da

qualidade da imagem, modelos práticos e confiáveis para representar a qualidade visual subjetiva ainda desafiam a comunidade de pesquisa.

Recentemente, as métricas de qualidade de imagens (*image quality metrics*, IQMs) baseadas em redes neurais convolucionais (*convolutional neural networks*, CNNs) ganharam muita atenção. Kang *et al.* [11] propuseram uma IQM sem referência, onde primeiro dividiram a imagem do teste em segmentos não sobrepostos e, em seguida, avaliaram a qualidade de cada segmento usando uma CNN. Os autores assumiram que a qualidade visual pode variar ao longo dos segmentos. No final, o índice de qualidade de toda a imagem é estimado reunindo todos os índices de qualidade dos segmentos. Bosse et al. [12] projetou uma CNN profunda para avaliar a qualidade da imagem de teste para condições de referência completa e sem referência. Seus resultados superaram as abordagens de ponta da época. Em [2], os autores usaram a mesma rede que em [12] para estimar o parâmetro de deslocamento de uma função para mapear a razão sinal-ruído de pico (*peak signal-to-noise ratio, PSNR*) para índices de qualidade subjetivos.

Vários trabalhos sobre CNNs [13, 14, 15, 16] sugerem que combinar informações de núcleos convolucionais de diferentes tamanhos (diferentes campos receptivos) fornece uma melhor representação do sinal de entrada. Yang *et al.* [13] utilizaram diferentes campos receptivos, variando o fator de dilatação, para refinar a detecção e extração de pingos de chuva de imagens. Também relacionado com a remoção de chuva, os autores em [14] assumiram que a densidade da chuva impacta no resultado, levando-os a usar três redes com variações no campo receptivo para classificar a densidade da chuva antes da eliminação da mesma. Finalmente, resultados apresentados em [15] e[16] sugerem que informações em diferentes escalas são importantes para detecção de borramento.

Com base nos artigos mencionados, investigamos a influência da combinação de informações de diferentes escalas na avaliação da qualidade visual. Especificamente, propomos uma rede *single-stream* (SDN) responsável por inserir informações perceptuais no PSNR.

A performance da rede proposta é comparada com estratégias do estado da arte, usando um banco de dados público LIVE (Laboratory for Image & Video Engineering) [17].

1.1.2 Objetivos

Esta dissertação tem como objetivos:

- Analisar a influência de informações de diferentes escalas no processo de avaliação de qualidade visual de imagens estáticas.

- Analisar diferentes formas de inserção de informações multiescala em estruturas de redes convolucionais de forma diminuir a complexidade (número de parâmetros e quantidade de operações de ponto flutuante por segundo) do trabalho de Bosse *et al.* [2].

1.1.3 Contribuições

As contribuições deste trabalho são as seguintes: (a) é realizado um estudo sobre a importância do uso de informações de diferentes escalas em uma CNN aplicada à previsão de qualidade subjetiva de imagem; (b) é proposta uma uma versão simplificada do trabalho apresentado por Bosse *et al.* [2] através da utilização de uma rede densa *single-stream* que combina informações de diferentes escalas para atribuir contexto perceptual ao PSNR; e (c) é disponibilizado o código fonte, utilizado neste trabalho, no GitHub para os interessados.

1.2 Organização do trabalho

O restante deste trabalho está organizado da seguinte forma:

- Capítulo 2: neste capítulo, é feita uma introdução dos métodos de avaliação de qualidade de imagens bem como é feita uma descrição do método de adaptação perceptual do PSNR proposto em [2];

- Capítulo 3: apresenta alguns conceitos relacionados a aprendizado de máquina utilizados neste trabalho. Os fundamentos teóricos de Redes Neurais Artificiais e aprendizado profundo são abordados bem como algumas estruturas fundamentais de redes e seus aspectos teóricos associados;

- Capítulo 4: são detalhadas a técnica proposta e as estruturas de rede utilizadas;

- Capítulo 5: o procedimento experimental é definido e, por fim, os resultados obtidos são apresentados e discutidos;

- Capítulo 6: a partir da análise detalhada dos resultados, serão apresentadas as conclusões da técnica analisada, destacando vantagens e desvantagens observadas.

Capítulo 2

Avaliação de qualidade de imagens

"Uma imagem vale mais que mil palavras", este ditado expressa a importância das informações visuais para os seres humanos. Não é de se espantar que haja grande esforço na condução de pesquisas no desenvolvimento de métodos automáticos de avaliação de qualidade visual de imagens e vídeos.

O conhecimento das características do sistema visual humano (*human visual system*, HVS) é fundamental nesse esforço. Embora o conhecimento sobre o HVS esteja longe de ser pleno, o que já se conhece vem permitindo a concepção de métricas (como SSIM [18], MS-SSIM [19], FSIM [20], e HaarPSI [21]) de melhor desempenho que o das métricas baseadas no MSE em relação à percepção da qualidade visual.

2.1 Avaliações subjetiva e objetiva

A melhor forma de conseguir avaliar a qualidade de informações visuais é através da opinião de observadores. A nota média de opiniões (*mean opinion score*, MOS), avaliação subjetiva obtida a partir de um grupo de observadores, é a melhor forma de avaliar a qualidade de uma imagem se se pretende que essa avaliação esteja de acordo com a percepção humana. Porém, esse método de avaliação demanda tempo e é pouco prático no uso, elaboração, e teste de produtos ou algoritmos.

Frequentemente, o MSE é utilizado pelos métodos de avaliação de qualidade de imagens, devido a sua fácil implementação e baixo custo computacional. No entanto, esses métodos apresentam baixo nível de correlação com as avaliações subjetivas, pois o MSE não leva em consideração características espaciais [9].

As pesquisas sobre modelos e métricas perceptuais para avaliação visual buscam contornar os inconvenientes da avaliação subjetiva e das métricas baseadas em MSE. Há duas abordagens possíveis para o desenvolvimento desses simuladores perceptuais: *bottom-up* e *top-down*.

A abordagem *bottom-up* estuda o funcionamento de cada elemento do sistema visual humano relevante para a percepção de qualidade, combinando-os em um sistema computacional.

A abordagem *top-down* considera o HVS como uma caixa preta, onde comportamentos hipotéticos são implementados e ajustados. Os pontos atrativos desta abordagem são que o único conhecimento prévio necessário é a relação entre a entrada e a saída do sistema, e a simplicidade de implementação.

2.2 Classificação das métricas de avaliação de qualidade visual

Os diferentes métodos para avaliação de qualidade visual podem ser classificados em três categorias, com base na disponibilidade da imagem de referência:

- Referência completa: há uma imagem de referência (considerada sem distorção) para avaliar uma imagem de teste. Proporciona resultados mais precisos em relação a similaridade e fidelidade entre as duas imagens;
- Sem referência: utilizada quando não é possível ter acesso à imagem referência; a avaliação da imagem de teste deve ser feita as "cegas". Por isso esta categoria também é conhecida como de referência cega. As técnicas desta categoria são as que possuem resultados de menor correlação com dados experimentais subjetivos;
- Referência reduzida ou parcial: neste caso, apenas algumas características da imagem de referência estão disponíveis. As características correspondentes da imagem de teste são calculadas e comparadas com as da imagem de referência no processo de avaliação de qualidade.

Outra classificação possível seria em relação à utilização das características do HVS. Neste caso, existem duas categorias:

- Não perceptuais: não utilizam características do HVS nas suas formulações e têm como virtude a baixa complexidade computacional. Porém, possuem baixa correlação com as

avaliações subjetivas.

- Perceptuais: são formulações matemáticas inspiradas em características fisiológicas e psicovisuais da visão que representam de forma automática a percepção humana perante uma representação visual. Normalmente, apresentam maior correlação com a percepção humana.

A seguir, serão apresentadas métricas não perceptuais e perceptuais de especial interesse para este trabalho.

2.2.1 Não perceptuais

2.2.1.1 Erro quadrático médio (Mean Square Error - MSE)

O MSE é uma métrica bastante utilizada que consiste no valor esperado do quadrado do erro,

$$MSE = E\left[(y(i,j) - x(i,j))^2\right] \tag{2.1}$$

em que y é a imagem de teste, x é a imagem de referência e (i, j) determina a posição dos pixels.

2.2.1.2 Razão Sinal-Ruído (Signal-to-Noise Ratio - SNR)

Quantifica o quanto um sinal foi distorcido através da razão entre a energia do sinal e a energia do erro associado à imagem distorcida.

A SNR é comumente medida em dB da seguinte forma,

$$SNR = 10\log_{10}\frac{\sum_{i,j}[x(i,j)]^2}{\sum_{i,j}[x(i,j) - y(i,j)]^2} \tag{2.2}$$

2.2.1.3 Razão Sinal-Ruído de Pico (Peak Signal-to-Noise Ratio - PSNR)

É normalmente utilizada para medir a qualidade da reconstrução da imagem ou vídeo após uma compressão com perdas.

A PSNR é medida em dB da seguinte forma,

$$PSNR = 10\log_{10}\frac{K^2}{MSE} \tag{2.3}$$

onde K representa o valor máximo que um pixel pode atingir. No caso de uma imagem de 8 bits/pixel o valor de K é 255.

2.2.2 Perceptuais

2.2.2.1 Índice de Similaridade Estrutural (Structural Similarity Index - SSIM)

O SSIM talvez seja a função de aproximação perceptual mais utilizada, devido à sua baixa complexidade computacional e maior correlação com dados subjetivos em relação a outras abordagens. Esta métrica de referência completa avalia o quanto a estrutura da imagem de teste é diferente da estrutura da imagem de referência.

O SSIM é definido como

$$SSIM = \left[l(x,y)^{\alpha} c(x,y)^{\beta} s(x,y)^{\gamma}\right] \tag{2.4}$$

em que $\alpha > 0$, $\beta > 0$ e $\gamma > 0$ são responsáveis pelo ajuste da importância relativa das componentes de luminância $l(x,y)$ (brilho medido por um instrumento apropriado), contraste $c(x,y)$ (diferença de brilho entre um objeto e seu entorno próximo) e estrutura $s(x,y)$ (estrutura dos objetos presentes em uma imagem) definidas como

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{2.5}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \tag{2.6}$$

$$s(x,y) = \frac{2\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \tag{2.7}$$

em que μ_i e σ_i são a média e o desvio padrão da imagem i ($i = x, y$); σ_{xy} é a covariância entre as imagens de referência e teste, x e y, respectivamente. Os termos c_1, c_2 e c_3 são inseridos com o objetivo de evitar instabilidades numéricas.

Esta métrica é aplicada localmente, deslocando, horizontalmente e verticalmente, uma janela de tamanho $B \times B$ sobre a imagem. Pode-se calcular a média do SSIM (MSSIM) para que se tenha uma índice de qualidade geral da imagem, em que M é o número de janelas e $SSIM_j$ é o SSIM associado à j-ésima janela.

$$MSSIM = \frac{1}{M}\sum_{j=1}^{M} SSIM_j \tag{2.8}$$

2.3 Função de mapeamento entre métricas perceptuais e dados subjetivos

A relação entre os dados subjetivos (Q_p) e as métricas de qualidade (Q_c) não é linear. A Figura 2.1 apresenta um exemplo típico, em que Q_c é representado pelo PSNR e a diferença das opiniões média (DMOS, *difference mean opinion score*) representa Q_p. Devido ao efeito de saturação, a função de regressão logística de 4 parâmetros é comumente utilizada para mapear os valores de Q_c em Q_p [2, 22, 12, 23]:

$$\hat{Q}_p = a + \frac{b-a}{1+e^{-c(Q_c-d)}} \tag{2.9}$$

onde $\hat{Q}_p$ é a predição de Q_p; a e b são os limites superior e inferior da métrica, respectivamente; o parâmetro c controla a inclinação da curva de mapeamento; e d desloca a curva horizontalmente.

Os parâmetros a e b são definidos pelo design do experimento psicofísico. Depois da coleta dos valores de Q_p e dos cálculos de Q_c, um processo de otimização estima os valores de c e d.

As abordagens convencionais admitem que os parâmetros c e d são constantes para todo o conjunto de dados. Porém, recentemente, os trabalhos [2, 12, 11] consideram a variabilidade desses parâmetros não apenas entre as imagens do conjunto de teste, mas também dentro de uma dada imagem.

2.4 PSNR perceptualmente adaptado

Os resultados apresentados em [2] indicam que a predição da qualidade perceptual é mais sensível ao deslocamento (d) que à inclinação (c) da função de mapeamento (Equação 2.9). A relevância do d motivou a definição de um PSNR adaptado que incorpora esse parâmetro de deslocamento:

$$paPSNR = 10\log_{10}\frac{C^2}{MSE} - d \tag{2.10}$$

$$= 10\log_{10}\frac{C^2}{10^{\frac{d}{10}}MSE} \tag{2.11}$$

$$= 10\log_{10}\frac{C^2}{paMSE} \tag{2.12}$$

em que $paPSNR$ e $paMSE$ representam as versões perceptualmente adaptadas do PSNR e MSE, respectivamente; e C é o valor máximo do sinal (255 para uma imagem de 8 bits/pixel).

Por sua vez, o MSE é dado por

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [s(x,y) - \hat{s}(x,y)]^2 \tag{2.13}$$

em que $s(x,y)$ e $\hat{s}(x,y)$ são as imagens de referência e distorcida, respectivamente; e M e N são as dimensões da imagem.

Estatísticas de imagens naturais são localmente estruturadas e não estacionárias, então a qualidade percebida não varia apenas entre diferentes imagens mas também espacialmente em uma mesma imagem [24, 25]. Considerando a variabilidade espacial da percepção de distorções, o parâmetro d pode assumir diferentes valores em diferentes posições $d(x,y)$ [11, 12, 2], dando origem a uma nova versão perceptualmente adaptada do MSE (paMSE):

$$paMSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} 10^{\frac{d(x,y)}{10}} [s(x,y) - \hat{s}(x,y)]^2 \tag{2.14}$$

Este simples esquema de ponderação dá contexto espacial ao parâmetro de deslocamento (d), levando a um $paPSNR$ mais representativo em termos de qualidade subjetiva.

Na prática, a imagem de teste é dividia em blocos não sobrepostos, em que os pixels dentro de um mesmo bloco compartilham do mesmo parâmetro de deslocamento.

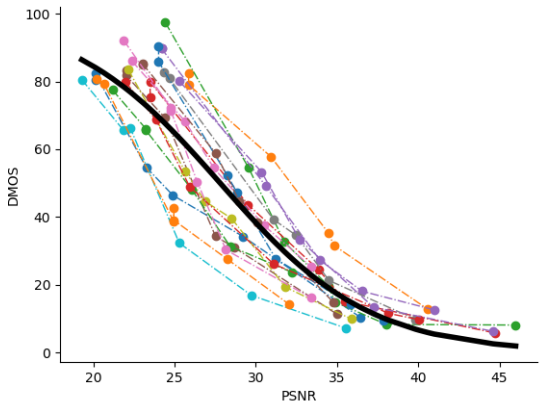

Figura 2.1: PSNR vs. DMOS para o conjunto de imagens JPEG da base de dados LIVE [1]. As curvas coloridas indicam pares PSNR-DMOS para uma imagem de referência. A curva na cor preta representa a regressão dos valores de DMOS de todas as imagens JPEG (Adaptado de [2]).

Capítulo 3

Aprendizado de máquina

3.1 Redes neurais artificiais

Redes Neurais Artificiais (RNA) são modelos computacionais de aprendizagem de máquina inspirados em redes neurais biológicas, originados em 1943 no trabalho de McCulloch e Pitts [26]. RNAs são muito utilizadas em tarefas de reconhecimento de padrões, como por exemplo, reconhecimento de fala e de objetos, e identificação de células cancerígenas.

Cada neurônio de uma RNA pode ser representado conforme a Figura 3.1 em que $x = [x_1, x_2, ..., x_n]$ representa o sinal de entrada, $w = [w_1, w_2, ..., w_n]$ é o vetor de pesos sinápticos, b_k é o *bias* associado ao neurônio a fim de ajudá-lo a se adaptar de melhor forma possível e T é uma função (não-linear) de ativação e y denota a saída do neurônio.

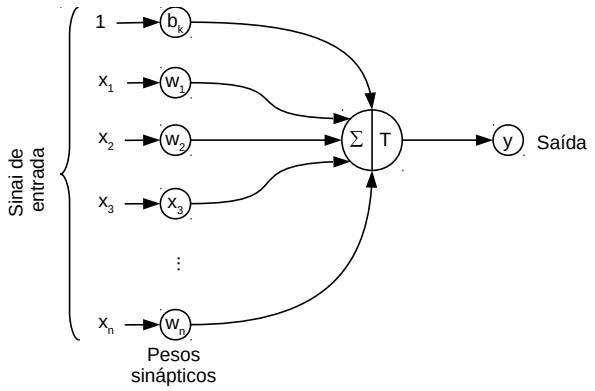

Figura 3.1: Neurônio artificial (Fonte: O autor).

Um neurônio artificial pode ser descrito segundo a Equação 3.1.

$$y_k = T\left(\sum_{i=1}^{m} x_i w_{ki} + b_k\right) \tag{3.1}$$

3.1.1 Funções de ativação

O papel das funções de ativação não se limita a possibilitar que uma rede possa resolver problemas não lineares. Elas também são responsáveis por limitar as saídas de neurônios e acelerar o processo de aprendizagem. Dentre as funções mais utilizadas, podemos citar (Figura 3.2):

- Sigmoide: geralmente utilizada na camada de saída de uma rede em aplicações que exijam resultados no intervalo $[0, 1]$ (Equação 3.2).

$$\varphi(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

- Tangente hiperbólica: limita os valores de saída do neurônio no intervalo $[1, -1]$ e acelera o processo de aprendizado quando comparada com a função sigmoide (Equação 3.3).

$$\varphi(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3.3}$$

- ReLU (*rectified linear unit*): mais utilizada em camadas escondidas, por ser capaz de acelerar o aprendizado e por evitar que o gradiente diminua ou cresça de forma descontrolada (Equação 3.4).

$$\varphi(x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{if } x \leq 0. \end{cases} \tag{3.4}$$

- Leaky ReLU: gera resultados melhores que todas as outras funções no controle do gradiente e aceleração de aprendizado. Porém, tem um maior custo computacional associado (Equação 3.5).

$$\varphi(x) = \begin{cases} x & \text{if } x > 0, \\ 0,1x & \text{if } x \leq 0. \end{cases} \tag{3.5}$$

Não há critérios rígidos na escolha das funções de ativação a serem utilizada em uma RNA, sendo o desempenho na aplicação pretendida o fator determinante.

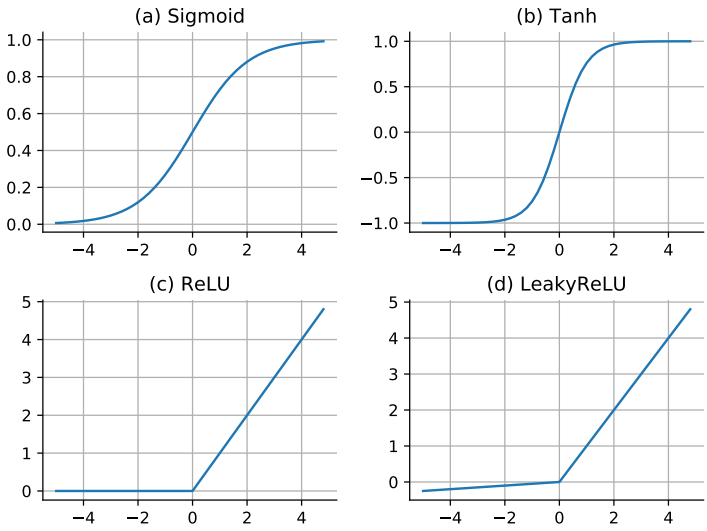

Figura 3.2: Gráfico de algumas funções de ativação: **(a)** Sigmoide; **(b)** Tanh; **(c)** ReLU; **(d)** LeakyReLU.

3.1.2 Perceptron multicamadas

As redes neurais do tipo perceptron são formadas por apenas um neurônio desenvolvidas para solução de problemas linearmente separáveis. Como grande parte dos problemas existentes não são linearmente separáveis surgiu uma arquitetura mais robusta chamada Perceptron multicamadas (*Multi-layer perceptron*, MLP).

Diferentemente da perceptron convencional, a perceptron multicamadas (Figura 3.3) é composta por um conjunto de camadas de neurônios diretamente alimentadas, chamadas: camada de entrada, camadas escondidas (ou ocultas) e camada de saída.

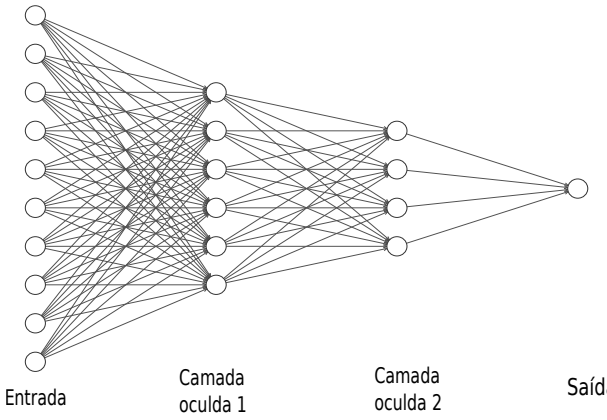

Figura 3.3: Exemplo de MLP com 10 neurônios na camada de entrada, duas camadas ocultas (6 e 4 neurônios, respectivamente) e 1 neurônio na camada de saída (Fonte: O autor).

3.1.3 Função de custo

A função de custo mapeia um evento ou valores de uma ou mais variáveis num número real representando alguma figura de mérito associada ao evento ou conjunto de variáveis. Um problema de otimização procura minimizar uma função de perda. O cálculo é geralmente feito através da média de alguma função que permita avaliar o quão distante se está de uma resposta aceitável para um problema (Equação 3.6)

$$L = \frac{1}{N} \sum_{i} E(\hat{Y}_i, Y_i) \tag{3.6}$$

em que E é a função de custo, e $\hat{Y}_i$ e Y_i representam a resposta da rede e o vetor de referência, respectivamente.

3.1.4 Método do gradiente descendente

O gradiente descendente é um método numérico usado em otimização a fim de minimizar alguma função. Esta minimização é realizada modificando os pesos e os limiares de ativação com o objetivo de encontrar o mínimo local da função perda.

A atualização dos pesos e dos *bias* é feita de acordo com as equações 3.7 e 3.8, respectivamente,

$$w_k \leftarrow w_{k-1} - \alpha \Delta_w L \tag{3.7}$$

$$b_k \leftarrow b_{k-1} - \alpha \Delta_b L \tag{3.8}$$

em que o parâmetro α das equações representa a taxa de aprendizado (*learning rate*, LR). Esta taxa é então multiplicada pela derivada da função perda (L) com relação a cada peso e *bias* (Δ_w e Δ_b).

3.2 Aprendizado profundo

3.2.1 Motivação biológica

No contexto de aprendizagem de máquina, redes neurais convolucionais (*Convolutional Neural Networks*, CNN) referem-se a um conjunto de estruturas variantes das redes perceptron multicamadas (*Multilayer Perceptron*, MLP), largamente aplicadas em processamento de imagens.

Assim como as redes MLPs, as redes CNNs são inspiradas em processos biológicos. A estrutura de conectividade de seus neurônios é baseada no córtex visual dos animais, em que cada neurônio responde a uma pequena porção do campo visual (campo receptivo). Os campos receptivos dos neurônios se sobrepõem parcialmente para que todo o campo visual seja coberto [3]. Matematicamente, esse processo é representado por convoluções.

A popularização das CNNs é devida à necessidade de poucos parâmetros, quando comparado às MLPs convencionais, para o processamento de informações visuais e por demandarem pouco ou, até mesmo, nenhum pré-processamento no sinal de entrada em algumas aplicações.

3.2.2 Camadas

As CNNs possuem, essencialmente, duas estruturas básicas: camadas convolucionais e *pooling*. Tais estruturas são responsáveis por inserirem graus de invariância temporal ou espacial.

3.2.2.1 Camadas convolucionais

Uma das limitações das redes MLP é o alto custo computacional associado ao processamento de sinais de alta dimensionalidade, como imagens. Para que uma MLP possa processar informações visuais faz-se necessário um número grande de parâmetros, dificultando a utilização da mesma. Por isso, camadas convolucionais passaram a ser utilizadas para a extração de características, em que cada neurônio está ligado a uma pequena porção da imagem (campo receptivo) e todos os neurônios compartilham o mesmo conjunto de pesos de forma que a saída de uma camada convolucional é a resposta de um sistema linear e invariante (convolução), conforme as equações 3.9 e 3.10 para os casos 1D e 2D, respectivamente, em que f é o sinal de entrada e g é o *kernel* de convolução.

$$f[n] * g[n] = \sum_{v=-\infty}^{\infty} f[v]g[n-v] \tag{3.9}$$

$$f[m,n] * g[m,n] = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f[u,v]g[m-u,n-v] \tag{3.10}$$

Considerando um conjunto de N entradas de C_{in} canais e dimensões $H_{in} \times W_{in}$ $(N, C_{in}, H_{in}, W_{in})$, pode-se determinar as dimensões da saída de uma camada convolucional

através das seguintes equações:

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times P[0] - D[0] \times (K[0] - 1) - 1}{S[0]} + 1 \right\rfloor \tag{3.11}$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times P[1] - D[1] \times (K[1] - 1) - 1}{S[1]} + 1 \right\rfloor \tag{3.12}$$

em que $P[i]$ representa o número de zeros adicionados nas extremidades da entrada no sentidos horizontal ($i = 0$) e vertical ($i = 1$); K representa as dimensões dos kernels de convolução; D representa o espaçamento (fator de dilatação) entre os elementos dos kernels, conforme a Figura 3.4; e S representa o tamanho dos passos (*stride*) da convolução, conforme exemplificado na Figura 3.5.

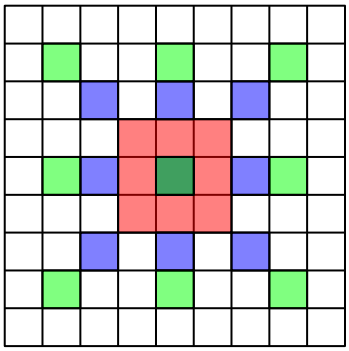

Figura 3.4: Representação de diferentes fatores de dilatação em uma camada convolucional. Em vermelho, azul e verde têm-se dilatações de 0, 1 e 2, respectivamente (Fonte: O autor).

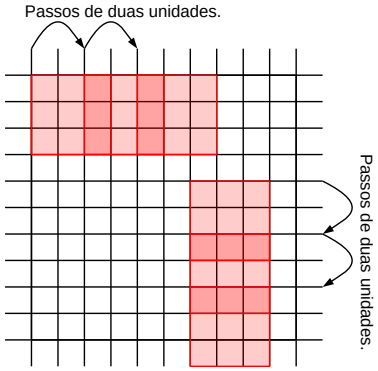

Figura 3.5: Representação de uma convolução 3×3 com passos de duas unidades (Fonte: O autor).

Comumente, as camadas convolucionais, assim como as camadas densas nas redes MLP, são seguidas por uma função não linear para que a rede seja capaz de solucionar problemas não lineares.

3.2.2.2 Camadas pooling

Objetivando a redução de dimensionalidade do volume processado e controle de *overfitting*, costuma-se inserir periodicamente camadas *pooling* entre conjuntos de camadas convolucionais. Sua função é reduzir progressivamente o tamanho espacial da sua entrada e, consequentemente, redução do custo computacional.

O *pooling* também é aplicado em pequenas porções da entrada, geralmente com filtros 2×2 e com deslocamento de duas unidades, conforme a Figura 3.6. Essa redução pode ser feita com base nos valores máximos, mínimos, médios e até mesmo funções mais elaboradas.

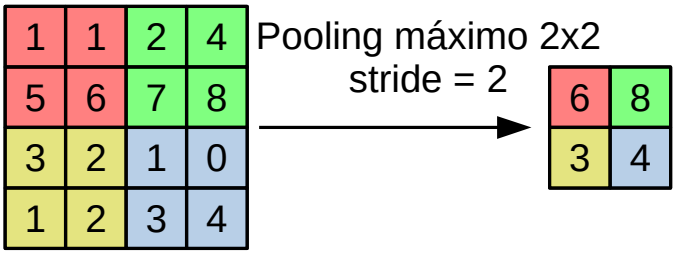

Figura 3.6: Tipo comum de pooling, pooling máximo, que desliza uma janela, como uma convolução, e preserva o maior valor na janela como saída (Fonte: O autor).

3.2.2.3 Camadas completamente conectadas

Comumente, após um conjunto de camadas convolucionais e *pooling* segue-se uma MLP convencional responsável pela tarefa de classificação ou regressão. Porém, seu vetor de características de entrada não é determinado previamente pelo usuário ma sim pelo conjunto de camadas convolucionais e *pooling* que antecedem a MLP.

3.2.3 Sub-ajuste e sobre-ajuste

O que diferencia o aprendizado de máquina de técnicas de otimização é a capacidade de treinar modelos capazes de generalizar exemplos desconhecidos.

A avaliação de desempenho é feita através da utilização de duas bases de dados: uma conhecida e (base de treino) outra desconhecida (base de teste). O objetivo é que minimizar o erro de ambas as bases. Quando os dois erros não caminham juntos surgem os problemas de sub-ajuste (*underfitting*) e sobre-ajuste (*overfitting*).

O sub-ajuste ocorre quando o modelo não é capaz de encontrar uma relação entre os dados de treinamento. Este fenômeno é evidenciado pela não redução do erro de treinamento.

O sobre-ajuste ocorre quando o modelo não é capaz de generalizar para exemplos desconhecidos. Este fenômeno é evidenciado pela redução do erro de treinamento e não redução do erro de teste.

3.2.4 Dropout

O *dropout* é uma técnica que objetiva minimizar o fenômeno de *sobre-ajuste* 3.2.3 (técnica de regularização). Consiste na remoção aleatória de neurônios de uma camada por iteração. Em cada iteração, os neurônios são reativados e são submetidos ao mesmo processo aleatório. Como consequência, ocorre um aumento da capacidade de generalização da rede, já que um neurônio não poderá depender da presença de um conjunto seleto de neurônios [27]

3.2.5 Arquiteturas

Apesar das CNNs serem bastante atrativas por proporcionarem uma redução considerável do número de parâmetros associada a uma pequena redução de performance [3], até o momento não há uma forma definida de construção de arquiteturas que possa ser generalizada. Em resposta a essa realidade, pesquisas por novas arquiteturas de redes têm sido impulsionadas por desafios como ImageNet LSVRC (do inglês *Large Scale Visual Recognition Competition*). Nesta seção, são apresentadas algumas arquiteturas de rede de fundamental importância para esta dissertação.

3.2.5.1 LeNet

A LeNet foi introduzida por LeCun *et al.* [3] em 1998 para reconhecimento de caracteres manuscritos e digitados. Sua objetividade e simplicidade o tornaram um exemplo amplamente usado para o ensino de CNNs.

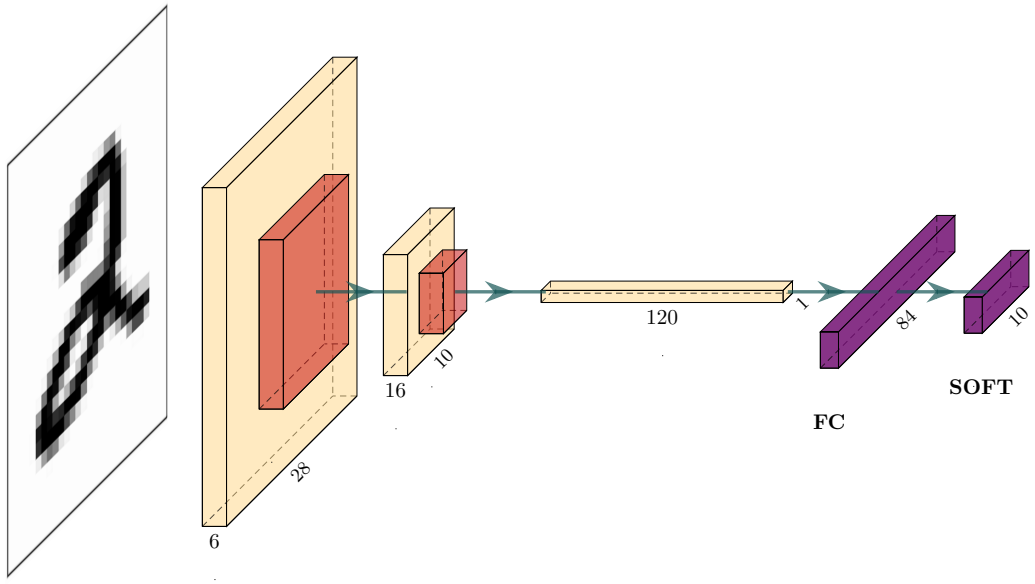

Figura 3.7: Arquitetura LeNet-5 [3]. Mais detalhes na Tabela B.1 do Apêndice B (Fonte: O autor).

A arquitetura da rede LeNet-5, Figura 3.7, foi desenvolvida para aceitar, como entrada, imagens em escala de cinza com tamanho 32×32 e é composta de dois conjuntos de camadas convolucionais e *pooling*, seguidos de uma camada convolucional responsável por vetorizar o sinal, duas camadas completamente conectadas e um classificador *softmax*.

Os resultados apresentados por LeCunn *et al.* [3] mostram que a rede LeNet-5 superou o estado da arte tanto com quanto sem o aumento artificial da base de dados.

3.2.5.2 AlexNet

O trabalho apresentado por Krizhevsky *et al.* [4] destacou-se, em 2012, ao apresentar uma estrutura mais profunda (AlexNet) alcançando as taxas de erro top-1 e top-5 ($37,5\%$ e $17,0\%$, respectivamente) do desafio LSVRC-2010, e taxa de erro top-5 ($15,3\%$) do desafio LSVRC-2012.

Para a redução de *overfitting* nas camadas completamente conectadas, a técnica de regularização, recém criada, *dropout* [27] foi utilizada.

Algumas camadas implementam métodos de normalização local, desenvolvidos pelo próprio autor, para aumentar o poder de generalização do modelo. Dado que $a^i_{x,y}$ representa a atividade do neurônio i na posição (x, y) submetida a função não-linear ReLU, a resposta normalizada é dada por

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{min(j=0,i-n/2)}^{min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta} \tag{3.13}$$

em que n é o numero de kernels adjacentes e as constantes k, n, α e β são hiperparâmetros cujos valores utilizados foram $k = 2$, $n = 5$, $\alpha = 10^{-4}$ e $\beta = 0,75$.

A arquitetura da rede Alexnet, Figura 3.8, é composta de cinco camadas convolucionais em que as camadas 1, 2 e 5 são seguidas por *pooling* máximo e as camadas 2 e 3 são normalizadas.

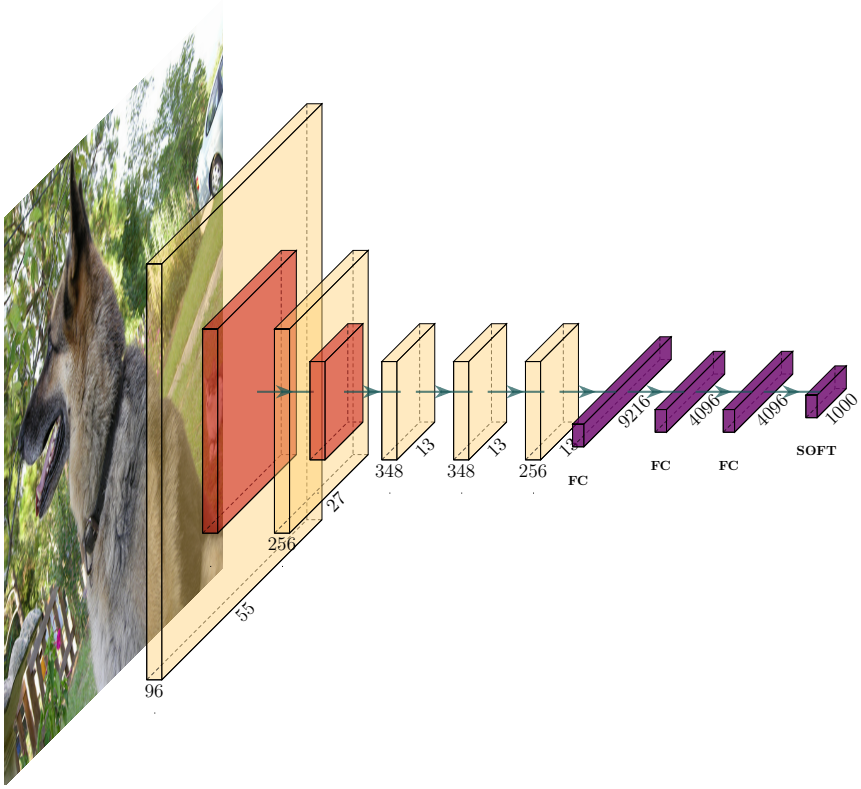

Figura 3.8: Arquitetura AlexNet [4]. Mais detalhes na Tabela B.2 do Apêndice B. (Fonte: O autor).

3.2.5.3 VGG

Em 2014, Simonyan *et al.* [5] propuseram a utilização de apenas *kernels* de convolução de tamanho mínimo (3×3) a fim de diminuir o número de parâmetros de uma rede e aumentar sua capacidade discriminativa da função de decisão através da inserção de mais não-linearidade. Nessa arquitetura, utiliza-se a ideia de substituição de camadas convolucionais 5×5 por duas camadas 3×3, de uma camada 7×7 por três 3×3 e assim por diante. Essa substituição, além de proporcionar uma diminuição do número de parâmetros da rede, insere mais não-linearidades, aumentando o poder de adaptação da rede. Tal processo é exemplificado na Figura 3.9. Este

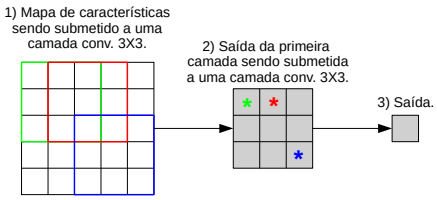

Figura 3.9: Agrupamento de camadas convolucionais 3×3 para o aumento do campo receptivo. Neste caso, duas camadas convolucionais 3×3 geram um campo receptivo 5×5 (Fonte: O autor).

estudo deu origem a uma estrutura de rede, submetida pelo time "VGG", com profundidades variando de 16 a 19 camadas sem a necessidade de normalização local, que garantiu os primeiro e segundo lugares para as tarefas de localização e classificação no desafio ImageNet 2014.

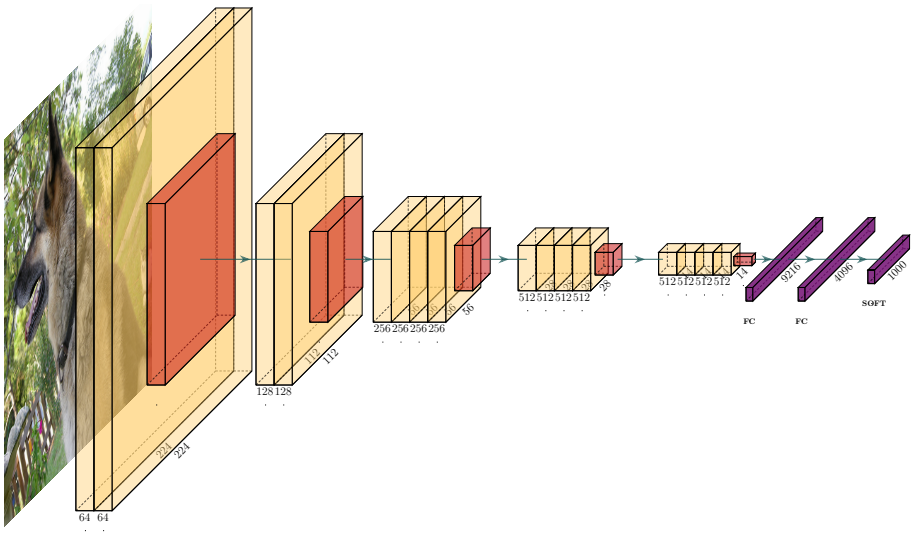

Figura 3.10: Arquitetura VGG-19 [5]. Mais detalhes na Tabela B.3 do Apêndice B (Fonte: O autor).

3.2.5.4 ResNet

Considerando $\mathcal{H}(X)$ como uma função de mapeamento a ser representada por um conjunto de camadas (não, necessariamente, toda a rede), em que X é a entrada da primeira dessas camadas. Segundo o trabalho publicado por He *et al.* [6] é mais fácil otimizar o mapeamento residual ($\mathcal{F}(X)$) do que o original, $\mathcal{H}(x)$. Logo, a estrutura de rede apresentada em seu trabalho utiliza o mapeamento residual, como é mostrado na Equação 3.14, para acelerar o aprendizado e reduzir problemas relacionados ao gradiente em estrutura profundas.

$$\mathcal{F}(X) = \mathcal{H}(x) - x. \tag{3.14}$$

Isso pode ser incorporado na estrutura da rede através de "atalhos" (Figura 3.11) que não adicionam parâmetros extras ou custo computacional [6].

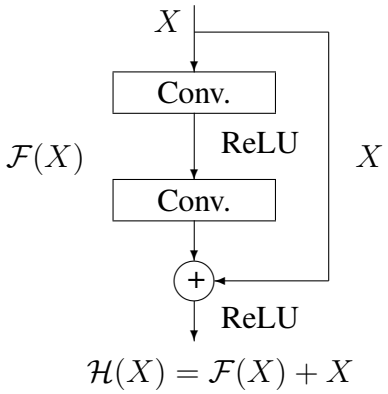

Figura 3.11: Bloco residual (Adaptado de He *et al.* [6]).

Tais "atalhos" funcionam como um mapeamento de identidade, e sua saída é adicionada à saída do conjunto de camadas empilhadas. A rede pode ser treinada fim-a-fim através de algum algoritmo de otimização, como o SGD (do inglês *stochastic gradient descent*).

Essa metodologia possibilitou a criação de redes com 152 camadas (8$\times$ mais profundas que as redes "VGG" [5]) que garantiram os primeiros lugares para as tarefas de detecção e localização para o desafio ILSVRC 2015, e detecção e segmentação para o desavio COCO 2015.

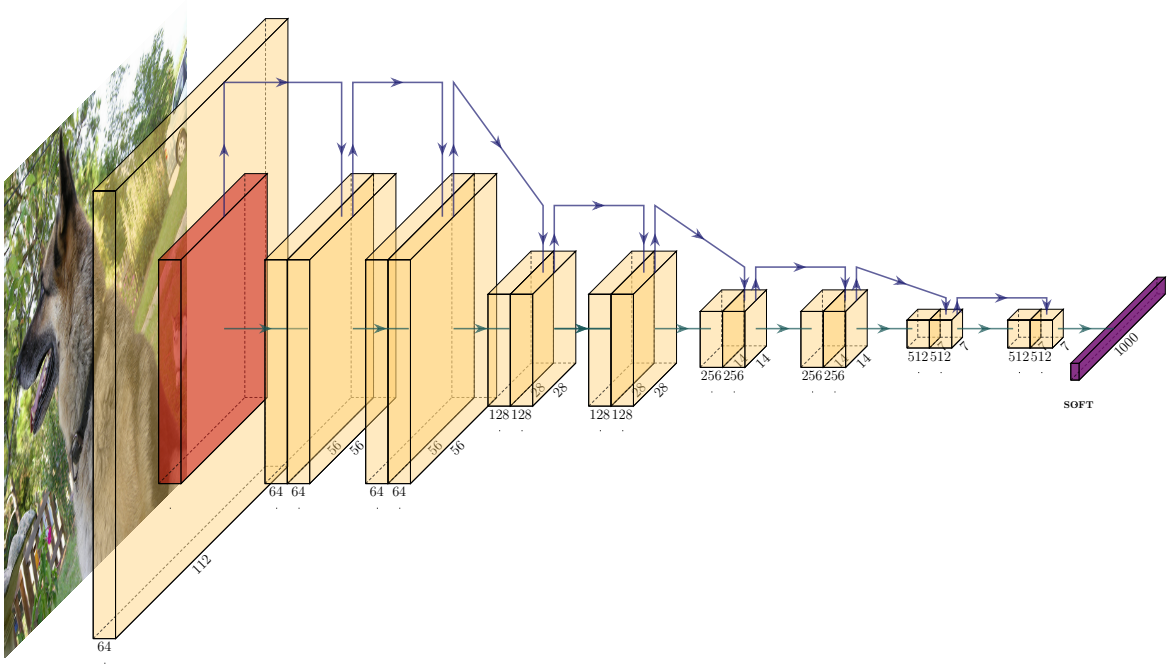

Figura 3.12: Arquitetura ResNet-18 [6]. Mais detalhes na Tabela B.4 do Apêncide B (Fonte: O autor).

3.2.5.5 DenseNet

Em 2016, Huang *et al.* [7] apresentou uma nova estrutura de rede denominada rede convolucional densamente conectada. Sua rede é composta por estruturas densas em que a saída de cada camada está conectada à entrada de todas as camadas a sua frente. Enquanto uma rede convolucional convencional com L camadas possui L conexões, uma rede densa possui $\frac{L(L+1)}{2}$ conexões diretas.

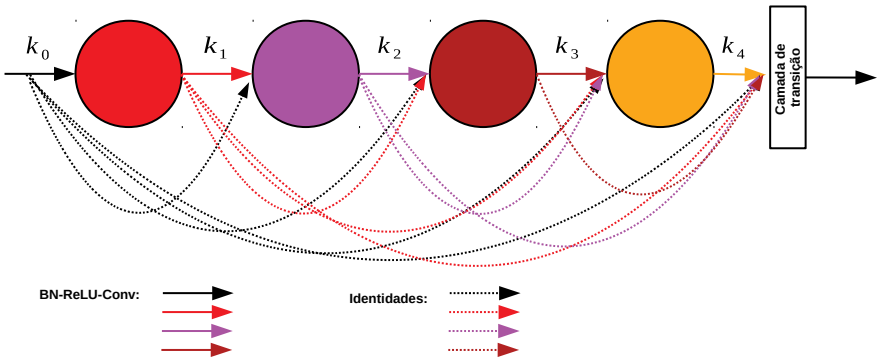

Figura 3.13: Bloco denso (Fonte: O autor).

Considerando que há uma concatenação de saídas de camadas para formar a entrada de uma camada a frente e que cada camada gera uma saída de k canais, o número de canais do sinal de entrada de uma camada l é dado por

$$k_l = k_0 + K \times (l - 1) \tag{3.15}$$

em que k_0 é o número de canais na camada de entrada e k é o fator de crescimento, conforme ilustrado na Figura 3.13.

O uso de estruturas densas é atrativo por vários motivos: redução do problemas com crescimento ou diminuição do gradiente; facilitação da propagação de características extraídas; e redução do número de parâmetros.

Nos testes apresentados por Huang *et al.* [7], a DenseNet foi capaz de superar o estado da arte na área com menor número de parâmetros que todas as estruturas apresentadas anteriormente.

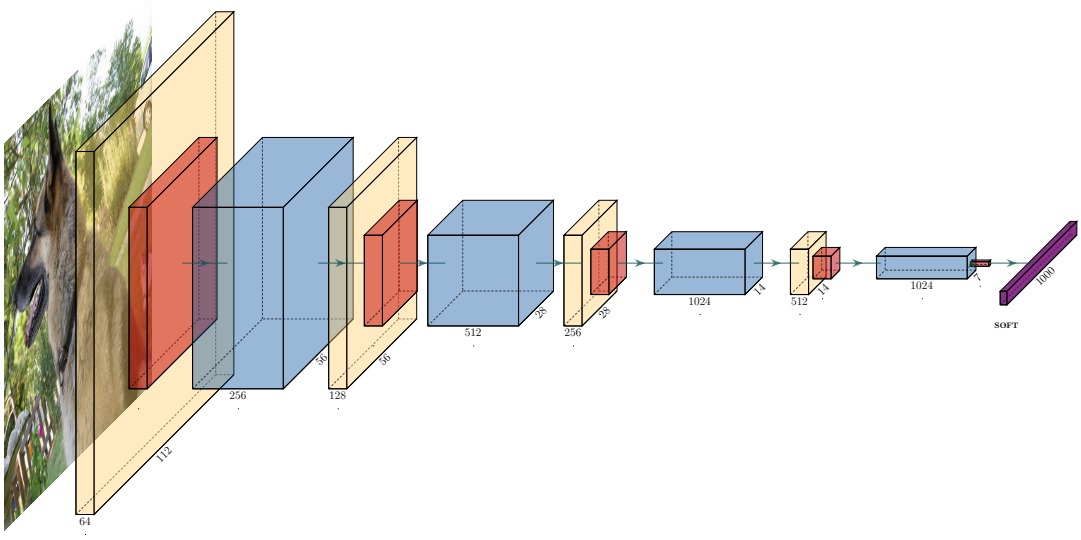

Figura 3.14: Rede densa de 121 camadas [7]. Mais detalhes na Table B.5 do Apêndice B (Fonte: O autor).

Capítulo 4

Desenvolvimento do sistema

4.1 Técnica proposta

Este trabalho estende a técnica proposta por Bosse *et al.* [2], a fim de inserir informações de múltiplas escalas. A justificativa está nas melhorias que a inserção desse tipo de informação trouxe para algumas aplicações, tais como remoção de chuva e detecção de borramento [13, 14, 15, 16].

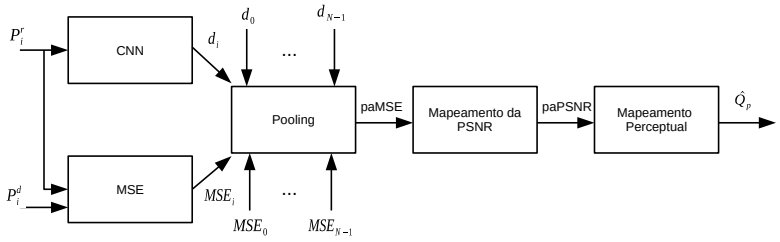

Figura 4.1: Uma visão geral do método proposto: P_i^r e P_i^t representam o i-ésimo segmento das imagens de referência e teste, respectivamente; d_i denota o parâmetro de deslocamento da função logística do i-ésimo segmento; MSE_i é o valor de erro quadrático médio do i-ésimo segmento; paPSNR é a PSNR adaptada para a imagem de teste; e $\hat{Q}_p$ é estimativa de qualidade visual (Adapado de [2]).

A Figura 4.1 apresenta o funcionamento geral do sistema em questão, em que uma rede convolucional é utilizada para estimar os parâmetros de deslocamento (d_i) para cada um dos N segmentos (*patches*) da imagem de teste. Em [28], uma análise da influência de diferentes tamanhos dos *patches* (8, 16, 32, 64 e 128) é apresentada em que se conclui que este aspecto

pouco influencia no resultado final. Logo, para facilitar a comparação do método proposto com o estado da arte, o tamanho dos *patches* utilizado foi de 32×32 pixels.

A rede utilizada é inspirada na arquitetura VGG [5] e é composta por 12 camadas, conforme detalhado na Tabela 4.1.

Tabela 4.1: Arquitetura de rede utilizada por Bosse *et al.* [2].

Camada		Profundidade	Dimensões	Dimensões do kernel	Stride	Ativação
Entrada	Imagem	1	32x32	-	-	-
1	Convolução	32	32x32	3x3	1	ReLU
2	Convolução	32	32x32	3x3	1	ReLU
	Pooling máximo	32	16x16	2x2	2	-
3	Convolução	64	16x16	3x3	1	ReLU
4	Convolução	64	16x16	3x3	1	ReLU
	Pooling máximo	64	8x8	2x2	2	-
5	Convolução	128	8x8	3x3	1	ReLU
6	Convolução	128	8x8	3x3	1	ReLU
	Pooling máximo	128	4x4	2x2	2	-
7	Convolução	256	4x4	3x3	1	ReLU
8	Convolução	256	4x4	3x3	1	ReLU
	Pooling máximo	256	2x2	2x2	2	-
9	Convolução	512	2x2	3x3	1	ReLU
10	Convolução	512	2x2	3x3	1	ReLU
	Pooling máximo	512	1x1	2x2	2	-
11	FC	-	512	-	-	ReLU
Saída	FC	-	1	-	-	Softmax

Fonte: o autor.

4.1.1 Arquiteturas avaliadas

Inspirado no sucesso de métodos que usam informações de escalas diferentes para a remoção de chuva [13, 14] e detecção de borramento [15, 16], foram analisadas três estruturas para predizer o parâmetro de deslocamento na função (Equação 2.9) que mapeia uma métrica

de qualidade (PSNR) para as notas de qualidade visual (DMOS): MDN (*multi-stream dense network*), SND (*single-stream dense network*) e MN (*multi-stream network*).

4.1.1.1 Multi-stream dense network

Na Figura 4.2 a arquitetura completa da rede proposta MDN é apresentada, a qual consiste de três estruturas densas com campos receptivos de tamanhos diferentes [14], referidas como Dense (3×3), Dense (5×5) e Dense (7×7), em azul, verde e laranja, respectivamente. Cada estrutura densa gera uma resposta de dez canais as quais são concatenadas para compor a entrada de um regressor. O regressor estima o parâmetro de deslocamento de um dado segmento de imagem e possui a seguinte estrutura: `Conv(30,64,3)`–`Conv(64,24,3)`–`FC(24576,512)`–`FC(512,1)`.

`Conv(`x`,`y`,`z`)` representa uma rede camada convolucional com a ativação ReLU (*(rectified linear unit)*), com entrada de x canais, saída de y canais e as dimensões dos kernels de convolução iguais a $z \times z$. Por fim, `FC(`m`,`n`)` representa uma camada completamente conectada de m entradas, n saídas, e função de ativação ReLU.

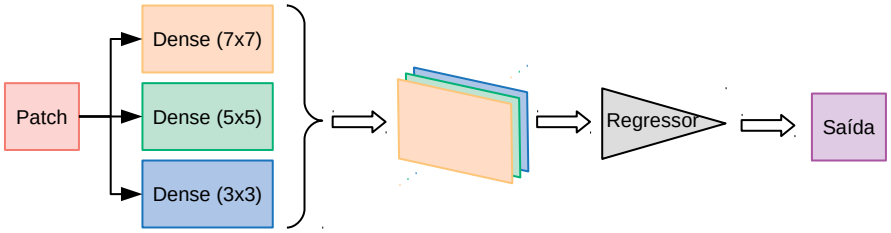

Figura 4.2: Uma visão geral da rede MDN. Segmentos da imagem de teste, com tamanho 32×32, são submetidos a três estruturas densas com diferentes campos receptivos (3×3, 5×5, 7×7). As saídas dessas três estruturas densas são concatenadas para serem usadas como entrada para um regressor (Fonte: O autor).

Cada estrutura densa é composta por seis blocos densos, seguidos por blocos de transição (Figura 4.3). Nesse trabalho foram utilizados três tipos de blocos de transição (Apêndice C.1).

Um bloco denso é composto por um agrupamento denso de 4 camadas convolucionais com fator de crescimento $k = 16$ e cada estrutura densa tem diferentes combinações de camadas de transição:

- Dense (7×7:) três camadas de transição para baixo e três camadas de transição para cima.
- Dense (5×5:) duas camadas de transição para baixo, duas camadas de transição sem amostragem e duas camadas de transição para cima.
- Dense (3×3:) uma camada de transição para baixo, quatro camadas de transição sem amostragem e uma camada de transição para cima.

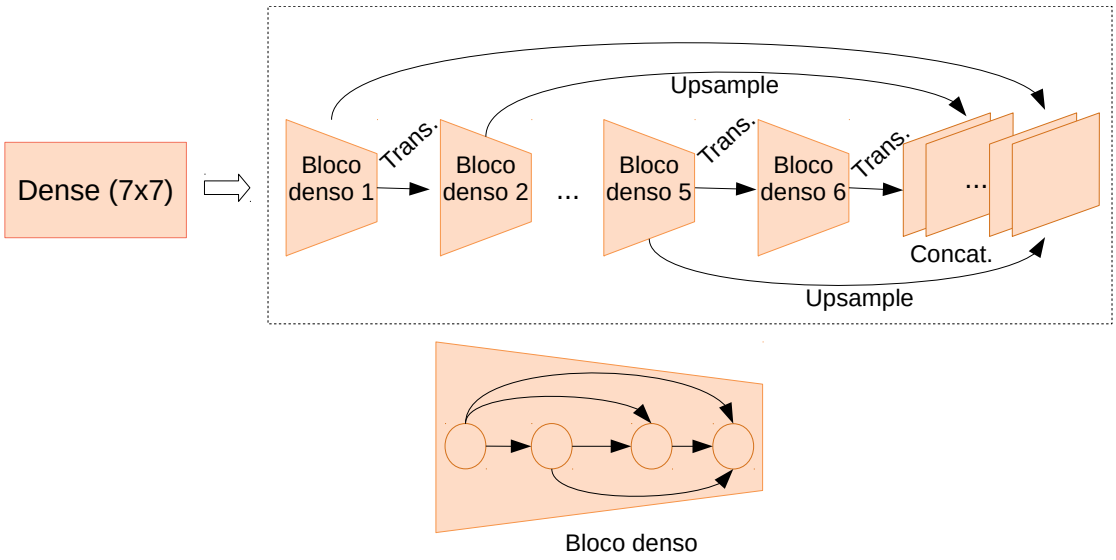

Figura 4.3: Detalhes da estrutura densa (7×7) (Fonte: O autor).

4.1.1.2 Single-stream dense network

Como cada estrutura densa da rede MDN 4.1.1.1 retorna uma concatenação de saídas de camadas em diferentes profundidades, foi utilizada apenas uma estrutura Dense (3×3) para explorar informações de diferentes escalas. A saída da estrutura Dense (3×3) é usada como entrada para um regressor estruturado da seguinte forma: `Conv(10,32,3)–Conv(32,24,3)–FC(24576,512)–FC(128,1).`

4.1.1.3 Multi-stream network

Esta rede é uma versão simplificada da rede MDN 4.1.1.1. Sua entrada passa por uma camada convolucional, cuja saída é submetida a três ramos paralelos com diferentes fatores de

dilatação (1, 2 e 3).

Cada estrutura densa gera uma resposta de dez canais as quais são concatenadas para compor a entrada de uma camada convolucional e sua saída será a entrada do mesmo regressor usado na rede SDN 4.1.1.2, conforme detalhado na Figura 4.4.

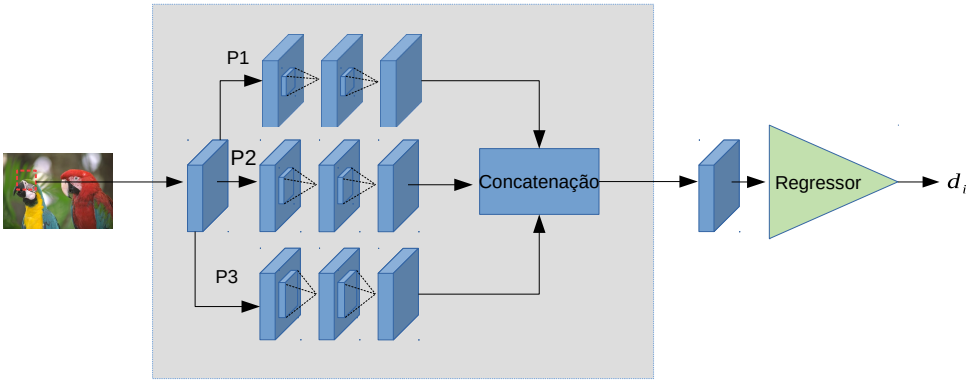

Figura 4.4: Uma visão geral da rede MN. Segmentos da imagem de teste, com tamanho 32×32, são submetidos a três estruturas convolucionais com diferentes fatores de dilatação (1, 2 e 3). As saídas dessas três estruturas convolucionais são concatenadas para compor a entrada de uma camada convolucional e sua saída será a entrada de um regressor (Fonte: O autor).

Capítulo 5

Procedimento experimental e resultados

5.1 Materiais e métodos

5.1.1 Implementação do sistema

Para a realização dos experimentos apenas ferramentas livres foram utilizadas. Tanto a técnica apresentada em [2], quanto a proposta foram implementadas em Python [29] com auxílio da biblioteca Pytorch [30] e estão disponíveis em repositório público online.[1]

5.1.2 Base de dados

Para comparação de desempenho entre as técnicas, foi usada a base de dados pública LIVE [17], composta por 29 imagens de referência e 779 imagens de teste (versões das imagens de referência). As imagens de teste foram obtidas por diferentes níveis de 5 tipos distorções: compressão JP2K, compressão JPEG, adição de ruído gaussiano do tipo branco, borramento gaussiano, e desvanecimento do tipo Rayleigh. Para garantir que as imagens de referência usadas nos testes e na validação não tenham sido vistas pelas redes durante o estágio de treinamento, o conjunto de dados LIVE foi dividido em 29 subconjuntos de acordo com as imagens de referência, das quais 6 subconjuntos são escolhidos aleatoriamente para teste, outros 6 subconjuntos para validação e os 17 subconjuntos restantes para treinamento. Cada imagem de teste está associada a uma avaliação de qualidade subjetiva, obtida experimentalmente de acordo com as recomendações estabelecidas em [22, 31].

[1] https://github.com/LuanAGoncalves/DeepVisualQualityPrediction

5.1.3 Procedimento experimental

Neste trabalho, levamos em consideração apenas a estimativa do parâmero de deslocamento a partir da imagem de referência, com ($\gamma = \gamma^*$) e sem ($\gamma = 1$) fator de escala.

Os resultados apresentados por Bosse *et al.* [2] foram reproduzidos e comparados com os resultados obtidos utilizando as três redes propostas: *multi-stream dense network* (MDN), *single-stream dense network* (SDN) e *multi-stream network (MN)*. Tal análise é feita através dos coeficientes de correlação de Pearson e de Spearman para avaliar as relações linear e monotônicas entre as notas de qualidade visual e suas predições.

Adicionalmente, analisamos a complexidade da rede utilizada por Bosse *et al.* e das três redes apresentadas nesse trabalho através da comparação entre a quantidade de parâmetros das redes e seus receptivos números de operações de ponto flutuante por segundo (do inglês FLoating-point Operations Per Second, FLOPs).

5.1.4 Treinamento

Embora MSE seja comumente usado como função de perda para a tarefa de regressão, o erro absoluto médio (MAE) provou ser menos sensível para *outliers*. A função de perda adotada em [2] é indireta regressão do parâmetro de deslocamento (d) expresso na Equação 2.9 e representado na Figura 5.1.

$$\frac{1}{I}\sum_{i=1}^{I}|\hat{Q}_p^i - Q_p^i| \tag{5.1}$$

em que i é representa um determinado *patch*, e I representa o número total de *patches* utilizados para calcular o MAE.

A Equação 5.2 mostra outra função de perda, que requer que cada *patch* tenha o seu próprio d previamente calculado (Figura 5.2).

$$\frac{1}{I}\sum_{i=1}^{I}|\hat{d}_i - d_i| \tag{5.2}$$

Nos nossos experimentos, testamos ambos os critérios (equações 5.1 e 5.2). No entanto, as simulações deste trabalho consideram apenas o critério de regressão direta (Equação5.2) pois o mesmo gerou melhores resultados.

Para que seja possível a utilização da regressão direta do parâmetro de deslocamento (d) é necessário encontrar um parâmetro de inclinação (c) através de uma regressão logística a partir de todo o conjunto de treinamento e admitir que todos os *patches* de uma imagem

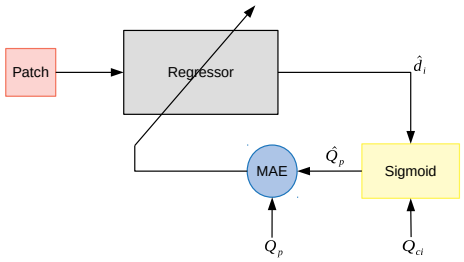

Figura 5.1: Regressão indireta do parâmetro d através da minimização do MAE entre d e sua previsão (Fonte: O autor).

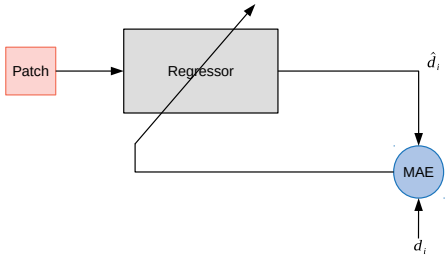

Figura 5.2: Regressão direta do parâmetro d através da minimização do MAE entre a qualidade perceptual e a qualidade prevista (Fonte: O autor).

herdam a qualidade perceptual da imagem. Desta forma é possível encontrar o parâmetro de deslocamento associado a um determinado *patche*.

Os modelos foram treinados por 50 épocas com o otimizador Adam [32] padrão a uma taxa de aprendizado constante de 10^{-3}. Após o treinamento a rede com melhor desempenho foi selecionada para a fase de teste. Uma iteração de treinamento consiste em avaliar o MAE (Equação 5.1) para um lote, que por sua vez é composto 32 *patches* em escala de cinza do tamanho 32×32, selecionadas aleatoriamente a partir de uma única imagem de referência. Uma etapa de validação acontece a cada 30 iterações de treinamento. Os *patches* de validação foram escolhidos aleatoriamente no início do processo de treinamento e permaneceram fixos até o final das 50 épocas de treinamento. Em uma única época, todas as imagens dos 17 subconjuntos de treinamento foram usadas para atualizar os parâmetros da rede. Para cada rodada de validação, todas as imagens dos 6 subconjuntos de validação foram usadas. Para o teste, todos os *patches* de todas as imagens de teste foram usados para avaliar o desempenho da rede. Nossos resultados foram relatados como a média de mais de 30 divisões aleatórias do conjunto de dados em subconjuntos de treinamento, validação e teste.

5.2 Resultados

A Tabela 5.1 apresenta uma comparação entre os métodos propostos e o estado da arte. Em uma comparação geral dos métodos de predição de qualidade visual vemos que as correlações de Pearson e de Spearman dos métodos de estimativa da qualidade visual propostos ($\text{MDN}_{\gamma=1}$, $\text{SDN}_{\gamma=1}$, $\text{MN}_{\gamma=1}$) e do método de referência ($\text{paPSNR}_{\gamma=1}$ [2]) são inferiores a todas as abordagens que extraem características a partir da imagem distorcida, mesmo assim apresentam resultados superiores ao PSNR padrão. Em uma análise focada nos métodos baseados na otimização do parâmetro de deslocamento, os métodos propostos não apresentaram nenhum ganho.

Tabela 5.1: Comparação entre as técnicas propostas e o estado da arte para referência completa: os maiores valores dos coeficientes de correlação de Pearson (*linear correlation coefficient*, LCC) e de Spearman (*Spearman rank order correlation coefficient*, SROCC) estão em vermelho; as técnicas propostas estão em negrito; as correlações apresentadas são valores médios de 30 treinamentos.

Extração de características a partir da imagem distorcida	Técnica	LCC	SROCC
Sim	HaarPSI	0,967	0,900
	SSIM [18]	0,945	0,948
	MS-SSIM [19]	**0,969**	**0,966**
	FSIM [20]	0,960	0,963
Não	PSNR	0,872	0,876
	$\text{paPSNR}_{\gamma=1}$ [2]	0,905	0,923
	MDN$_{\gamma=1}$	0,908	0,925
	SDN$_{\gamma=1}$	0,908	0,920
	MN$_{\gamma=1}$	0,909	0,925

Fonte: o autor.

A Tabela 5.2 mostra uma análise comparativa de complexidade entre os métodos considerados: nos métodos propostos há uma redução de aproximadamente 29% na quantidade de parâmetros em relação ao método apresentado por Bosse *et al.* [2]. Embora, a rede SDN tenha estatísticas similares as apresentadas pelas demais redes ela apresentou menores quantidades de

parâmetros e FLOPs, justificando a sua utilização no restante do trabalho.

Tabela 5.2: Análise de complexidade entre o método apresentado por Bosse *et al.* [2] e os métodos propostos (*). A menor complexidade e o menor número de parâmetros estão em negrito.

CLASSIFICADORES	NÚMERO DE PARÂMETROS	FLOPS
paPSNR [2]	4,979E+06	6,764E+07
MN*	3,174E+06	3,514E+07
SDN*	**3,168E+06**	**2,066E+07**
MDN*	3,511E+06	1,133E+08

Fonte: o autor.

Na Tabela 5.3, é sumarizado o desempenho da abordagem apresentada ($SDN_{\gamma=1}$) e comparado com o estado da arte para diferentes distorções com a base LIVE em termos de correlação de Spearman (SROCC). Assim como $paPSNR_{\gamma=1}$, $SDN_{\gamma=1}$ supera os resultados alcançados pelo PSNR em em quase todas as distorções. Apesar de a extração de características ser feita na imagem de referência, os resultados provenientes do método proposto são relativamente próximos dos métodos baseados nas imagens distorcidas.

Tabela 5.3: Comparativo entre o estado-da-arte e o método proposto(*). Cada valor representa o valor médio de SROCC para 30 treinamentos para as diferentes distorções presentes na base de dados LIVE

	JP2K	JPEG	AWGN	GB	FF
PSNR	0,895	0,881	0,985	0,782	0,891
SSIM [18]	0,961	0,976	0,969	0,952	0,956
MS-SIM [19]	0,962	0,981	0,973	0,954	0,947
FSIM [20]	0,972	0,984	0,972	0,971	0,952
HaarPSI [21]	0,968	0,983	0,985	0,967	0,951
$paPSNR_{\gamma=1}$ [2]	0,949	0,963	0,981	0,929	0,930
$SDN^{*}_{\gamma=1}$	0,952	0,955	0,973	0,921	0,930

Fonte: o autor.

Como a extração de características é feita com base na imagem de referência, a rede não tem informações a respeito da distorção presente na imagem de teste. Isso implica em uma

dependência do tipo de distorção. Tal dependência é comumente modelada pela inserção de fatores de escala específicos. A efetividade desse método pode ser atestada na Tabela 5.4.

Tabela 5.4: Comparação do método proposto (SDN) com o proposto por Bosse *et al.* [2] com fatores de escala otimizados para cada tipo de distorção. Os resultados apresentados são valores médios de 30 treinamentos.

CLASSIFICADORES	LCC	SROCC
paPSNR$_{\gamma=\gamma^*}$ [2]	0,930	0,935
SDN$_{\gamma=\gamma^*}$	0,928	0,933

Fonte: o autor

5.3 Discussão

Todos os métodos propostos (MDN, SDN e NM) alcançaram resultados similares aos alcançados pelo método de referência [2] com uma redução de, no mínimo, 29% no número de parâmetros da rede como mostram as tabelas 5.1 e 5.2, respectivamente. Apesar da redução de parâmetros, a utilização das redes SDN e NM proporcionaram reduções substanciais no número de FLOPs.

A rede que alcançou os melhores resultados foi a SDN, com uma redução de 36.37% e 69.45% para os números de parâmetros e FLOPS, respectivamente. Apesar dos resultados gerados pela SDN serem similares aos apresentados por Bosse *et al.* [2], observa-se uma redução de complexidade computacional na técnica proposta.

É importante notar que os métodos de adaptação perceptual do PSNR obtiveram uma diferença de aproximadamente 2% entre os coeficientes de Pearson e Speaman, o que é pouco comum nos demais métodos. Isso pode ser explicado através da dependência dos tipos de distorção que as métricas de qualidade visual que extraem características a partir das imagens de referência tem, como pode ser atestado através da tabela 5.3, que apresenta uma maior variação de valores de correlação de Spearman através da utilização da rede SDN quando comparada com valores obtidos por outros métodos (como SSIM, MS-SIM e FSIM), e da tabela 5.4, que apresenta a eficácia da utilização de fatores de escalas otimizados para cada tipo de distorção.

Capítulo 6

Conclusão

Nesta dissertação, analisamos a influência das informações multi-escala nas CNNs para estimação do parâmetro de deslocamento de uma função de mapeamento da qualidade visual. Embora Bosse *et al.* [2] sugira que as imagens distorcidas são mais adequadas para predizer a qualidade perceptual este trabalho está focado na extração de características a partir da imagem de referência, o que é mais difícil.

Baseado nos resultados que a utilização de informações multi-escala trouxeram para tarefas como a remoção de chuva [13, 14] e detecção de borramento [15, 16] supomos que este seria um aspecto interessante a ser explorado para a predição da qualidade de imagens. Para isso, propomos três estruturas de redes (MDN, SDN, e NM) com diferentes estratégias para explorar informações multi-escala. Todas possuem uma redução de aproximadamente 30% no número de parâmetros. A melhor rede (SDN) alcançou uma redução de 36.37% e 69.45% para o número de parâmetros e FLOPS, respectivamente, com resultados similares aos apresentados por Bosse *et al.* [2].

Apêndice A

A.1 Convenções de figuras

Neste apêndice apresentamos as convenções utilizadas para a representação de arquiteturas de redes apresentadas na Subseção 3.2.5 do Capítulo 3.1.

Considerações gerais

Por questões de simplicidade, as arquiteturas apresentadas neste trabalho não representam as camadas em si mas sim as dimensões de suas saídas. As saídas provenientes de diferentes tipos de camadas são discriminadas por cores diferentes.

É importante observar que, por questões de espaço, nem sempre é possível obedecer a escala que representam as dimensões reais das saídas provenientes de cada camada.

Maiores detalhes, referentes aos parâmetros utilizados nas camadas de cada arquitetura, são apresentados nas tabelas do apêndice B.

Representações de camadas e blocos

Conforme mencionado anteriormente, cada camada é representada pela sua saída em que sua cor determina o tipo da camada em questão. A convenção de cores utilizada foi:

- Convolucionais: bege.
- Completamente conectadas: vinho.
- Pooling: laranja.

No caso das redes densas (Figura 3.14) cada bloco denso é representado por sua saída como um todo. Para tal representação, a cor azul foi utilizada.

Conexões

Neste trabalho as conexões entre diferentes camadas também possuem peculiaridades que são representadas por diferentes cores e formas. As convenções utilizadas foram:

- Conexões entre camadas sucessivas: setas horizontais em verde.
- Conexões entre camadas não sucessivas de mesmas dimensões: setas de três segmentos, na cor roxa, com o topo horizontal.
- Conexões entre camadas não sucessivas de dimensões diferentes: setas de três segmentos, na cor roxa, com o topo diagonal.

Nota: No caso das conexões entre camadas não sucessivas de dimensões diferentes, é necessário realizar um *downsampling* na camada de origem para que seja possível concatenar as camadas de origem e destino da seta.

Apêndice B

B.1 Arquiteturas de rede

Tabela B.1: Arquitetura da LeNet-5 [3].

Camada		Profundidade	Dimensões	Dimensões do kernel	Stride	Ativação
Entrada	Imagem	1	32x32	-	-	-
1	Convolução	6	28x28	5x5	1	tanh
2	Pooling médio	6	14x14	2x2	2	tanh
3	Convolução	16	10x10	5x5	1	tanh
4	Pooling médio	16	5x5	2x2	2	tanh
5	Convolução	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Saída	FC	-	10	-	-	softmax

Fonte: O autor.

Tabela B.2: Arquitetura da AlexNet [4].

Camada		Profundidade	Dimensões	Dimensões do kernel	Stride	Ativação
Entrada	Imagem	3	224x224	-	-	-
1	Convolução	96	55x55	11x11	4	ReLU
	Pooling Máximo	96	27x27	3x3	2	ReLU
2	Convolução	256	27x27	5x5	1	ReLU
	Pooling máximo	256	13x13	3x3	2	ReLU
3	Convolução	348	13x13	3x3	1	ReLU
4	Convolução	348	13x13	3x3	1	ReLU
5	Convolução	256	13x13	3x3	1	ReLU
	Pooling máximo	256	6x6	3x3	2	ReLU
6	FC	-	9216	-	-	ReLU
7	FC	-	4096	-	-	ReLU
8	FC	-	4096	-	-	ReLU
Saída	FC	-	1000	-	-	Softmax

Fonte: O autor.

Tabela B.3: Arquitetura da VGG-19 [5].

Camada		Profundidade	Dimensões	Dimensões do kernel	Stride	Ativação
Entrada	Imagem	3	224x224	-	-	-
1	Convolução	64	224x224	3x3	1	ReLU
2	Convolução	64	224x224	3x3	1	ReLU
	Pooling máximo	64	112x112	3x3	1	ReLU
3	Convolução	128	112x112	3x3	1	ReLU
4	Convolução	128	112x112	3x3	1	ReLU
	Pooling máximo	128	56x56	3x3	1	ReLU
5	Convolução	256	56x56	3x3	1	ReLU
6	Convolução	256	56x56	3x3	1	ReLU
7	Convolução	256	56x56	3x3	1	ReLU
8	Convolução	256	56x56	3x3	1	ReLU
	Pooling máximo	256	28x28	3x3	1	ReLU
9	Convolução	512	28x28	3x3	1	ReLU
10	Convolução	512	28x28	3x3	1	ReLU
11	Convolução	512	28x28	3x3	1	ReLU
12	Convolução	512	28x28	3x3	1	ReLU
	Pooling máximo	512	14x14	3x3	1	ReLU
13	Convolução	512	14x14	3x3	1	ReLU
14	Convolução	512	14x14	3x3	1	ReLU
15	Convolução	512	14x14	3x3	1	ReLU
16	Convolução	512	14x14	3x3	1	ReLU
	Pooling máximo	512	7x7	3x3	1	ReLU
17	FC	-	4096	-	-	ReLU
18	FC	-	4096	-	-	ReLU
19	FC	-	1000	-	-	softmax

Fonte: O autor.

Tabela B.4: Arquiteturas do tipo ResNet [6].

Camada	Saída	18 camadas	34 camadas	50 camadas	101 camadas	152 camadas
conv1	112x112	Conv. 7x7, 64, stride = 2				
conv2_x	56x56	Pooling máximo 3x3, stride = 2				
		$\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \\ 1\times1, & 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \\ 1\times1, & 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \\ 1\times1, & 256 \end{bmatrix} \times 3$
conv3_x	28x28	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \\ 1\times1, & 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \\ 1\times1, & 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \\ 1\times1, & 512 \end{bmatrix} \times 8$
conv4_x	14x14	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \\ 1\times1, & 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \\ 1\times1, & 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \\ 1\times1, & 1024 \end{bmatrix} \times 36$
conv5_x	7x7	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \\ 1\times1, & 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \\ 1\times1, & 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \\ 1\times1, & 2048 \end{bmatrix} \times 3$
	1x1	Pooling médio, 1000 FC, softmax				

Fonte: O autor.

Tabela B.5: Arquiteturas do tipo DenseNet [7] com fator de crescimento $k = 32$.

Camada	Saída	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolução	112x112	Conv. 7x7, stride = 2			
Pooling	56x56	Pooling máximo 3x3, stride = 2			
Bloco denso (1)	56x56	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 6$	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 6$	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 6$	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 6$
Camada de transição	56x56	Conv. 1x1			
(1)	28x28	Pooling médio 2x2, stride = 2			
Bloco denso (2)	28x28	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 12$	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 12$	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 12$	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 12$
Camada de transição	28x28	Conv. 1x1			
(2)	14x14	Pooling médio 2x2, stride = 2			
Bloco denso (3)	14x14	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 24$	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 32$	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 48$	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 64$
Camada de tansição	14x14	Conv. 1x1			
(3)	7x7	Pooling médio 2x2, stride = 2			
Bloco denso (4)	7x7	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 16$	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 32$	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 32$	$\begin{bmatrix} \text{Conv.} & 1\times1 \\ \text{Conv.} & 3\times3 \end{bmatrix} \times 48$
Camada de	1x1	Pooling médio global 7x7			
Classificação		1000 FC, softmax			

Fonte: O autor.

Apêndice C

C.1 Blocos de transição

No trabalho desenvolvido por Jégou *et al.* [8], uma abordagem de segmentação semântica de imagens através de redes densas foi apresentada. A estrutura em questão, representada na Figura C.1, é composta por blocos densos e camadas de transição. As camadas de transição são utilizadas entre blocos densos para promover o aumento ou redução de dimensionalidade.

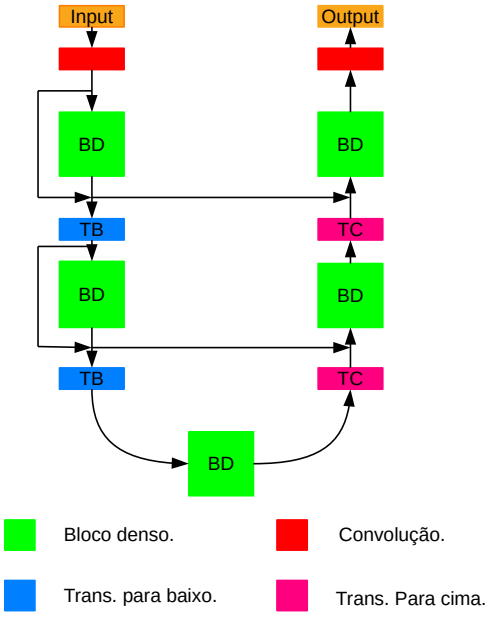

Figura C.1: Estrutura de rede utilizada em [8] (Adaptada).

No trabalho em questão, duas camadas de transição foram utilizadas: transição para cima e transição para baixo. A transição para cima é feita através de uma convolução transposta 3×3

com passo igual a 2; e a transição para baixo é composta de uma convolução 1×1, uma camada de *dropout* com $p = 0,2$ e uma camada de *pooling* máximo 2×2.

Tabela C.1: Bloco de transição para cima.

Transição para cima
Convolução transposta 3×3
Stride = 2

Fonte: O autor.

Tabela C.2: Bloco de transição para baixo.

Transição para baixo
Normalização de batch
ReLU
Convolução 1×1
Dropout $p = 0,2$
Max pooling 2×2

Fonte: O autor.

No nosso trabalho, camadas de transição sem amostragem (C.3) foram utilizadas, conforme o trabalho de Zhang *et al.* [14].

Tabela C.3: Bloco de transição sem amostragem.

Transição sem amostragem
Normalização de batch
ReLU
Convolução 1×1
Dropout $p = 0,2$

Fonte: O autor.

Referências Bibliográficas

[1] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *Trans. Img. Proc.*, vol. 15, no. 11, pp. 3440–3451, Nov. 2006. [Online]. Available: http://dx.doi.org/10.1109/TIP.2006.881959

[2] S. Bosse, S. Becker, Z. V. Fisches, W. Samek, and T. Wiegand, "Neural Network-Based Estimation of Distortion Sensitivity for Image Quality Prediction," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, oct 2018, pp. 629–633. [Online]. Available: https://ieeexplore.ieee.org/document/8451261/

[3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. [Online]. Available: http://ieeexplore.ieee.org/document/726791/

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[5] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Nov 2015, pp. 730–734.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[7] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2261–2269.

[8] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 11–19.

[9] Z. Wang and A. C. Bovik, "Modern Image Quality Assessment," *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 2, no. 1, pp. 1–156, jan 2006. [Online]. Available: http://www.morganclaypool.com/doi/abs/10.2200/S00010ED1V01Y200508IVM003

[10] B. Girod, "Chapter what's wrong with mean-squared error?" *"Digital Images and Human Vision,"*, pp. 207–220, 1993.

[11] L. Kang, P. Ye, Y. Li, and D. Doermann, "Convolutional Neural Networks for No-Reference Image Quality Assessment," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2014, pp. 1733–1740. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909620

[12] S. Bosse, D. Maniry, K.-R. Müller, T. Wiegand, and W. Samek, "Deep Neural Networks for No-Reference and Full-Reference Image Quality Assessment," *IEEE Transactions on Image Processing*, dec 2016. [Online]. Available: http://arxiv.org/abs/1612.01697http://dx.doi.org/10.1109/TIP.2017.2760518

[13] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep Joint Rain Detection and Removal from a Single Image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, sep 2016. [Online]. Available: http://arxiv.org/abs/1609.07769

[14] H. Zhang and V. M. Patel, "Density-aware Single Image De-raining using a Multi-stream Dense Network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, feb 2018. [Online]. Available: http://arxiv.org/abs/1802.07412

[15] R. Huang, W. Feng, M. Fan, L. Wan, and J. Sun, "Multiscale blur detection by learning discriminative deep features," *Neurocomputing*, vol. 285, pp. 154–166, apr 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231218300602

[16] L. Gillibert, T. Chabardès, and B. Marcotegui, "Local multiscale blur estimation based on toggle mapping for sharp region extraction," *IET Image Processing*, vol. 12, no. 12, pp. 2138–2146, dec 2018. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-ipr.2017.0095

[17] H. Sheikh, M. Sabir, and A. Bovik, "A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3440–3451, nov 2006. [Online]. Available: http://ieeexplore.ieee.org/document/1709988/

[18] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, apr 2004. [Online]. Available: http://ieeexplore.ieee.org/document/1284395/

[19] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. IEEE, 2003, pp. 1398–1402. [Online]. Available: http://ieeexplore.ieee.org/document/1292216/

[20] Lin Zhang, Lei Zhang, Xuanqin Mou, and D. Zhang, "FSIM: A Feature Similarity Index for Image Quality Assessment," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, aug 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5705575/

[21] R. Reisenhofer, S. Bosse, G. Kutyniok, and T. Wiegand, "A Haar wavelet-based perceptual similarity index for image quality assessment," *Signal Processing: Image Communication*, vol. 61, pp. 33–43, feb 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0923596517302187

[22] "ITU-T Rec. P.910 (04/2008) Subjective video quality assessment methods for multimedia applications," 2009. [Online]. Avai-

lable: https://www.semanticscholar.org/paper/ITU-T-Rec.-P.910-(04{%}2F2008)-Subjective-video-quality/6100e92fe43838214c927640c1d7567b9e85841e

[23] J. Bossu, N. Hautière, and J.-P. Tarel, "Rain or Snow Detection in Image Sequences Through Use of a Histogram of Orientation of Streaks," *International Journal of Computer Vision*, vol. 93, no. 3, pp. 348–367, jul 2011. [Online]. Available: http://link.springer.com/10.1007/s11263-011-0421-7

[24] A. J. Bell and T. J. Sejnowski, "The "independent components" of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327–3338, dec 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0042698997001211

[25] D. L. Ruderman, "The statistics of natural images," *Network: Computation in Neural Systems*, vol. 5, no. 4, pp. 517–548, 1994. [Online]. Available: https://doi.org/10.1088/0954-898X_5_4_006

[26] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[28] S. Bosse, S. Becker, K.-R. Müller, W. Samek, and T. Wiegand, "Estimation of distortion sensitivity for visual quality prediction using a convolutional neural network," *Digital Signal Processing*, vol. 91, pp. 54–65, 2019.

[29] G. Van Rossum and F. L. Drake Jr, *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[30] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[31] "Methodology for the subjective assessment of the quality of television pictures," 2012. [Online]. Available: https://www.itu.int/rec/R-REC-BT.500-13-201201-I/en

[32] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *3rd International Conference for Learning Representations, San Diego, 2015*, dec 2014. [Online]. Available: http://arxiv.org/abs/1412.6980