

# Compressão de Imagens com Perdas usando Redes Neurais

Raphael Soares Ramos

Universidade de Brasília

*raphael.soares.1996@gmail.com*

11 de julho de 2019

# Overview I

- 1 Introdução
  - Definições
  - Motivação
  - Compressão de Imagens
  - Hipótese
  - Objetivos
- 2 Fundamentação Teórica
  - JPEG
  - Redes Neurais
- 3 Metodologia
  - Bases de Dados
  - Modelos Desenvolvidos
- 4 Experimentos
  - JPEG
  - Modelo 1

# Overview II

- Modelo 2
- Modelo 3
- JPEG x Modelo 3

## 5 Conclusão

# Introdução

# Codificação de Dados

- Transformação feita nos dados para atingir um certo objetivo.
- Compressão (redução do comprimento) vs. Criptografia (proteger sigilo ou integridade do que os dados significam).

# Compressão de Dados

- Processo de codificar uma determinada informação utilizando uma menor representação.
- Arte ou ciência de representar informação de forma compacta [Sayood et al. 2017].
- Representações compactas são criadas identificando e usando estruturas existentes nos dados para que seja possível extrair redundâncias nos dados [Sayood et al. 2017].

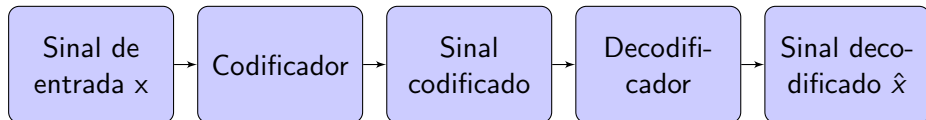
# Tipos de Compressão

- Sem perdas (Lossless):

- $x = \hat{x}$

- Com perdas (Lossy):

- $x \neq \hat{x}$



# Porquê comprimir

- Geração e uso cada vez maior de dados digitais (muitos são redundantes e irrelevantes para determinadas aplicações).
- Representar digitalmente 1 segundo de vídeo sem compressão usando o formato CCIR 601 requer mais de 20 *megabytes* de armazenamento ou 160 megabits para transmissão [Sayood et al. 2017].
- Imagem Monocromática com resolução  $512 \times 512$ :
$$Taxa = \frac{512 \cdot 512 \cdot 8}{10^6} = 2.097 Mbits.$$
- Aumentar capacidade de armazenamento e de transmissão de um sistema.
- Para serviços de streaming de mídia como Netflix, não usar compressão não é uma opção.



# Medidas de Desempenho em Compressão de Imagens

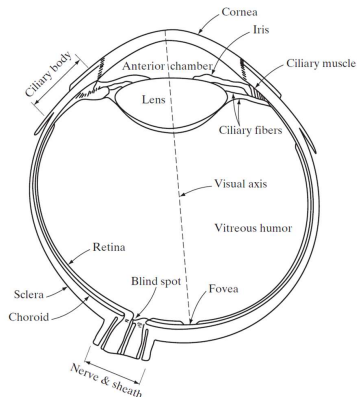
- Representar com o menor número possíveis de bits, preservando a qualidade e a inteligibilidade necessárias à sua aplicação.
- Taxa vs. Distorção.
- *Mean Squared Error* é uma medida de distorção.

$$MSE = \frac{1}{n} \sum_{n=1}^N (x(n) - \hat{x}(n))^2 \quad (1)$$

# Algoritmos de Compressão de Imagens

- Exploram características imperfeitas da nossa percepção e propriedades estatísticas para fornecer resultados superiores quando comparados com métodos de compressão de dados genéricos.
- Examinada durante anos por pesquisadores e times como o *Joint Pictures Experts Group*.
- Alguns métodos de compressão:
  - 1 JPEG [G. K. Wallace et al. 1993]
  - 2 JPEG2000 [C. Christopoulos et al. 2000]
  - 3 BPG [Bellard et al. 2019]
  - 4 WebP [Google 2019]

# Elementos da Percepção Visual



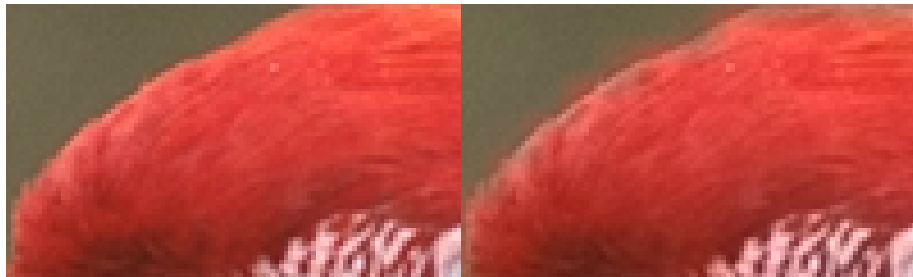
**Figura:** O olho humano é capaz de discriminar o brilho de uma imagem muito mais facilmente do que a sua informação de cor, pois existem cerca de 120 à 150 milhões de bastonetes distribuídos sobre a superfície da retina contra apenas 6 à 7 milhões de cones. Fonte: [Zaghetto et. al 2018]

# Elementos da Percepção Visual



**Figura:** Imagem original 512x768 à esquerda (retirada de [Kodak](#)) sem compressão e imagem (direita) gerada a partir da imagem original com dimensionalidade nos canais de croma reduzida por um fator de 8 nas duas direções. Economia total de 2.1875 Kbytes para armazenamento!

# Elementos da Percepção Visual



**Figura:** Versão com zoom da Figura 2. Aqui é possível notar artefatos existentes devido à perda de informação.

# Hipótese

- Nos últimos anos redes neurais profundas se tornaram a base dos resultados do estado da arte para diversas áreas
- Deseja-se verificar se modelos baseados em *autoencoders* convolucionais são competitivos com o clássico codec (codificador e decodificador) *JPEG*.

# Objetivos gerais e específicos

- Foi proposto um *autoencoder* convolucional que estenda a estrutura básica de um *autoencoder*, gerando uma representação binária para a imagem.
- Estudar propostas de compressão de imagens na literatura.
- Avaliar o desempenho dos modelos propostos e do *JPEG* em imagens de baixa resolução espacial.

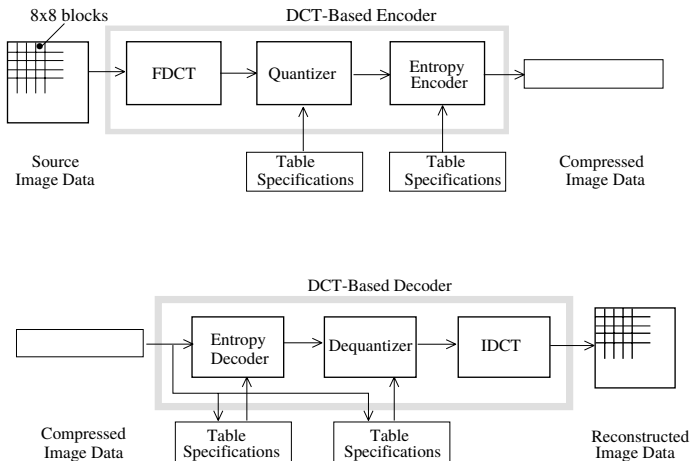
## Fundamentação Teórica



# Formato de Arquivos

- Arquivos comprimidos pelo método *JPEG* são normalmente descritos no formato *JPEG File Interchange Format (JFIF)*, que é uma limitação do padrão *JPEG* completo (muitos espaços de cores e modos de operação).
  - 1 Primeiro é feita conversão do espaço de cor para YCbCr.
  - 2 É usado um fator para reduzir a quantidade de pixels nos componentes de crominância. Normalmente é usado um fator de 2 nas duas direções.

# Funcionamento

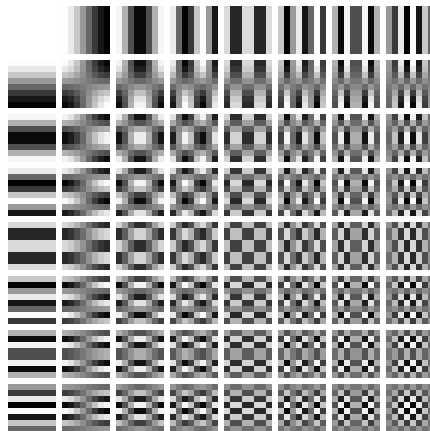


**Figura:** Diagrama geral que ilustra o funcionamento (codificação e decodificação) do método de compressão JPEG. Fonte: [G. K. Wallace et al. 1993]

# Transformada Discreta de Cossenos (DCT)

- Cada bloco  $8 \times 8$  pode ser replicado por 64 ( $8 \times 8$ ) ondas de cossenos.
- Analisa as frequências dos valores originais da imagem ao longo de cada linha e coluna usando um conjunto de ondas de cossenos oscilando em diferentes frequências e amplitudes. Cada bloco é representado usando estas ondas.

# Transformada Discreta de Cossenos (DCT)



**Figura:** 64 ondas base de cossenos que produzem qualquer imagem 8x8. A DCT-II irá calcular os coeficientes - a contribuição de cada uma dessas ondas que somadas irão recriar a imagem perfeitamente. Fonte: [Wikimedia Commons 2015]

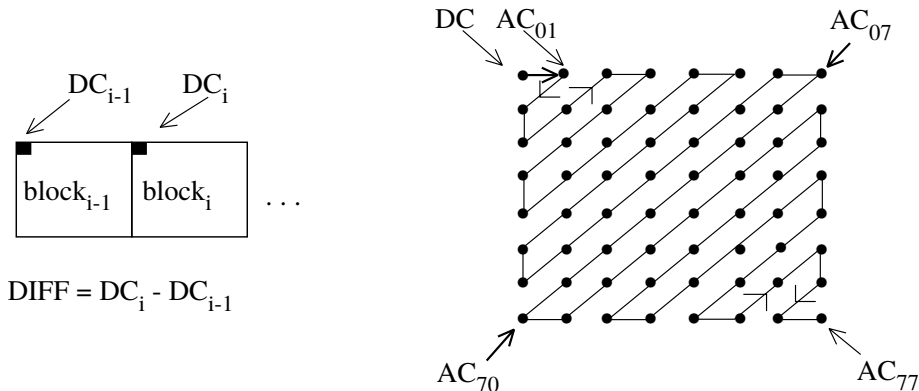
# Quantização

- Para o *encoder* será usada a seguinte equação:

$$\left[ F^Q(u, v) = \frac{F(u, v)}{Q(u, v)} \right] \quad (2)$$

- 1 Parte em que há maior perda de informação da imagem.
  - 2 Cada um dos 64 coeficientes DCT  $F(u, v)$  são uniformemente quantizados em conjunto com uma tabela de quantização de 64 elementos, que é definida pelo nível de qualidade escolhido para a aplicação.
  - 3 Remove altas frequências definindo altos valores para  $Q(u, v)$  (passo de quantização)
- No final, para cada bloco 8 por 8 da imagem original, existirá uma matriz quantizada.

# Codificação



**Figura:** Codificação dos Direct Current (DC) coeficientes (esquerda) e sequência zig-zag usada para codificar todos os coeficientes (direita).

Fonte: [G. K. Wallace et al. 1993]

# O problema com o JPEG



Figura: Imagem de texto com e sem compressão. Fonte: [Mike Pound et. al 2015]

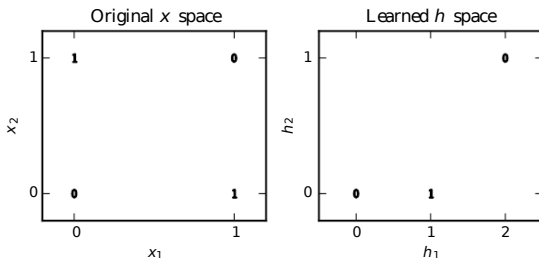
# Conceitos Básicos

- Para muitas tarefas em inteligência artificial, é necessário extrair *features* (características) dos dados.
  - ① As vezes é difícil saber quais *features* devem ser extraídas.
  - ② Solução: aprender a representação dos dados (*representation learning*).
- *Deep learning* resolve o problema de *representation learning* introduzindo representações que são expressadas em termos de outras representações mais simples.



# Conceitos Básicos

- Redes neurais são algoritmos de *deep learning* capazes de fazer previsões aprendendo uma função que relaciona as características dos dados às respostas observadas/desejadas
  - 1 Aproximadores universais de funções [Nielsen et. al 2015].
  - 2 Funções de ativações permitem aprendizado de funções mais complexas. Mais comum é a *ReLU*.



**Figura:** Um modelo linear aplicado diretamente à entrada original não pode implementar a função XOR. Fonte: [Goodfellow et. al 2015]

# Taxa de Aprendizagem

$$lr = baselr + (maxlr - baselr) \max(0, 1 - |itr/step - 2cycle + 1|) \gamma^{itr} \quad (3)$$

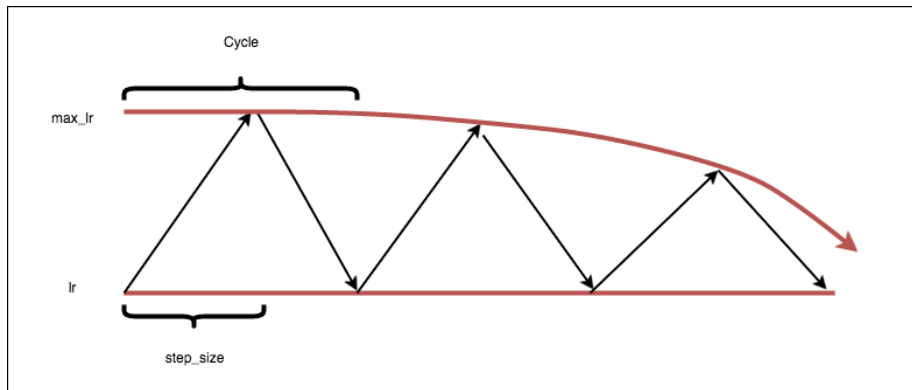


Figura: Política exp\_range de *learning rate* cíclica definida por [Leslie et. al 2017].

Fonte: [Brad Kenstler et. al 2015]

# Convoluções

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (4)$$

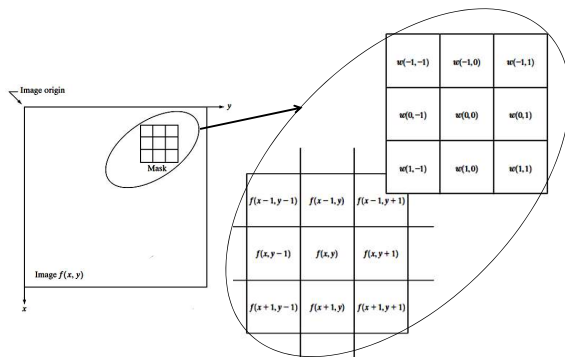


Figura: Operação de convolução. Fonte: [Zaghetto et. al 2018]

# Redes Neurais Convolucionais

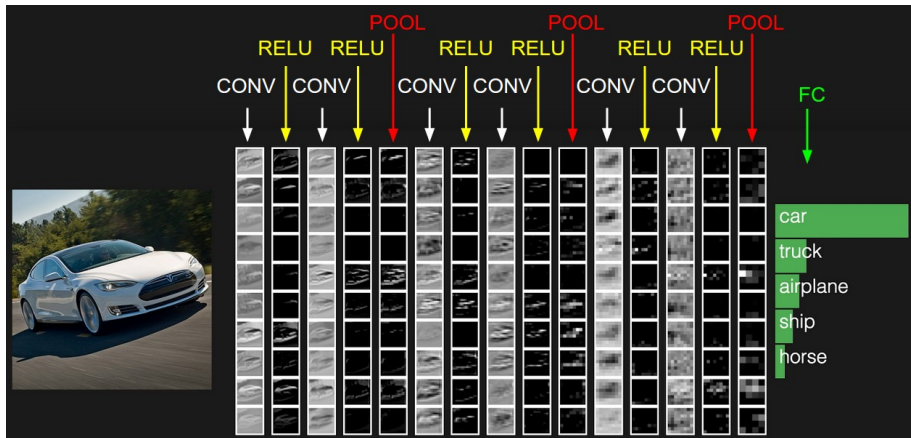


Figura: Ilustração de uma rede neural convolucional.

Fonte: [CS231n Course Materials 2019]

# Autoencoder

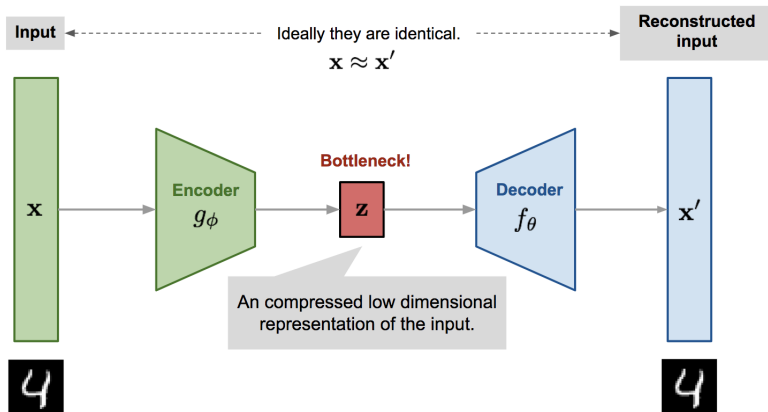


Figura: Ilustração de um *autoencoder*. Fonte: [Lilain Weng et.al 2018]

# Compressão de Imagens com Taxa Variável usando Redes Neurais Recorrentes [Toderici et. al 2015]

- Função de binarização  $b(x)$ :

$$b(x) = x + \epsilon \in \{-1, 1\}, \epsilon \sim \begin{cases} 1 - x & \text{com probabilidade } \frac{1+x}{2}, \\ -1 - x & \text{com probabilidade } \frac{1-x}{2}, \end{cases} \quad (5)$$

- Binarizador:

$$B(x) = b(\tanh(W^{bin}x + b^{bin})) \quad (6)$$

- Em tempo de teste  $b$  é substituída por:

$$b^{inf}(x) = \begin{cases} -1, & \text{se } x < 0, \\ 1, & \text{caso contrário.} \end{cases} \quad (7)$$

- Autoencoder:  $D(B(E(x)))$

# Metodologia

# Bases de Dados

- Foram pegas imagens de CLIC, DIV2K e EYE.
- Foram gerados 6,231,440 *patches* com 32 pixels de largura e altura. Estes *patches* foram divididos em 5 bases:
  - 1 **BD0**: formada por 1248978 de *patches* que pertencem ao grupo dos 20% com menor entropia;
  - 2 **BD1**: formada por 1251421 de *patches* que pertencem ao grupo dos que estão na faixa 40% à 60% (porcentagem dada pelo *patch* com maior entropia);
  - 3 **BD2**: formada por 1248725 de *patches* que pertencem ao grupo dos 20% com maior entropia;
  - 4 **BD3**: formada por 1247033 de *patches* pegos de forma aleatória. Correspondem à 20% do total.
  - 5 **BD4**: formada por 1246698 de *patches*. 20% do total retirados aleatoriamente dos 50% de *patches* com maior entropia.



# Bases de Dados

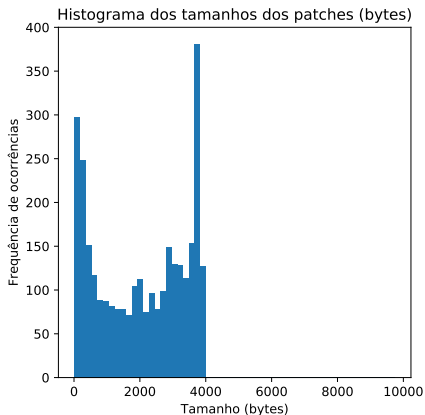


Figura: Histograma de todas as bases de dados.

# Bases de Dados

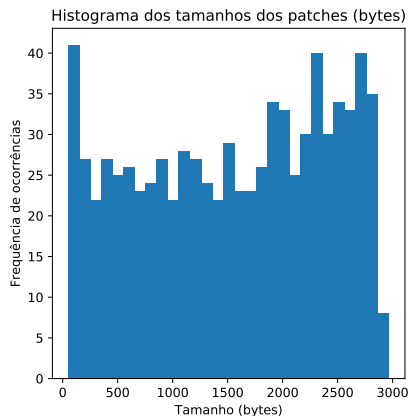


Figura: Histograma da BD2.

# Modelo 1

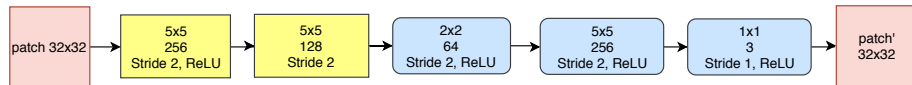


Figura: Ilustração do Modelo 1

# Modelo 2

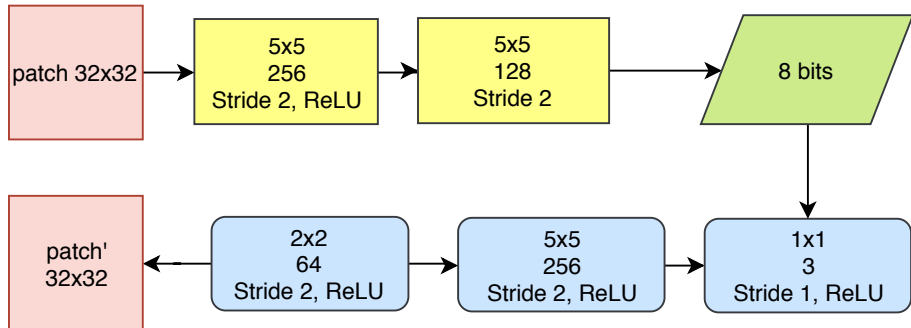


Figura: Ilustração do Modelo 2.

# Modelo 3

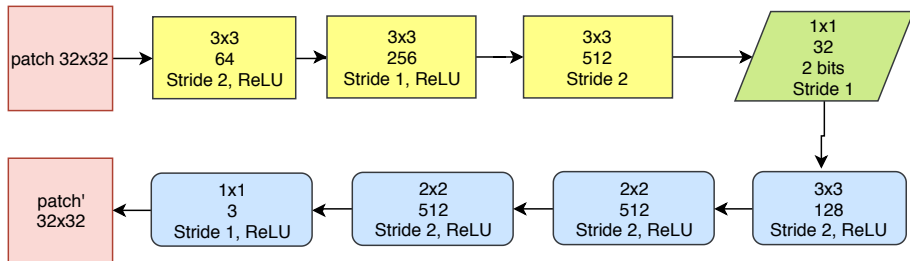


Figura: Ilustração do Modelo 3.

# Experimentos

# Desempenho JPEG

**Tabela:** Tabela contendo médias obtidas pelo **JPEG** em cada uma das bases de teste utilizadas

Bases	BPP	PSNR	SSIM	MSSIM	Quality
<b>CLIC Mobile test (patches 32)</b>	8	44.23	0.98	0.99	94.06
<b>CLIC Mobile test</b>	2	39.58	0.96	0.99	88.69
<b>Kodak</b>	2	36.77	0.95	0.99	85.91
<b>BD0</b>	8	57.85	0.99	0.99	99.18
<b>BD1</b>	8	42.94	0.97	0.99	95.94
<b>BD2</b>	8	32.31	0.95	0.99	83.69
<b>BD3</b>	8	43.84	0.96	0.99	93.94
<b>BD4</b>	8	36.72	0.96	0.99	89.68

# Treino e Teste em todas as BDs

**Tabela:** Tabela contendo o valor da *PSNR*, em decíbeis, dos testes do **Modelo 1** com o uso do otimizador *Adam* e *learning rate* fixa. As linhas denotam a base de treino utilizada. As colunas denotam as bases de teste usadas para avaliação do modelo. O índice “todas” se refere ao uso de todas as imagens de todas as bases **BD** para treino

Treino (linhas) x Teste (colunas)	BD0	BD1	BD2	BD3	BD4
<b>BD0</b>	49.66	38.07	26.34	38.05	31.13
<b>BD1</b>	47.57	44.08	34.54	42.66	38.69
<b>BD2</b>	<b>53.69</b>	<b>51.16</b>	<b>44.07</b>	<b>50.06</b>	<b>47.14</b>
<b>BD3</b>	50.62	47.88	39.76	46.59	43.25
<b>BD4</b>	47.30	46.98	41.10	45.63	43.75
<b>Todas</b>	46.77	46.35	43.94	45.88	45.08



# Resultados

**Tabela:** Tabela contendo o valor da *PSNR*, em decíbeis, dos testes do Modelo 1 com o uso do otimizador *Adam* e *learning rate* fixa.

Treino (linhas) x Teste (coluna)	CLIC Mobile test
<b>BD0</b>	34.77
<b>BD1</b>	44.11
<b>BD2</b>	48.97
<b>BD3</b>	45.34
<b>BD4</b>	42.42
<b>Todas</b>	51.27
<b>Todas + CLIC Mobile train</b>	<b>55.78</b>
<b>Todas + Clic Mobile train + Clic Professional train</b>	47.26
<b>CLIC Mobile Train</b>	46.63

# Resultados

**Tabela:** Tabela contendo os resultados do Modelo 2 para as métricas visuais *PSNR*, *SSIM* e *MS-SSIM* a uma taxa nominal de 8 bits por pixel

Bases de Treino e Teste	BPP	PSNR	SSIM	MS-SSIM	Épocas
<b>CLIC Mobile test</b>	8	35.03	0.94	0.98	30
<b>BD1</b>	8	35.26	0.93	0.98	30
<b>BD2</b>	8	29.10	0.95	0.98	30

# Resultados

**Tabela:** Tabela contendo os melhores resultados do Modelo 3

<b>Bases</b>	<b>BPP</b>	<b>PSNR</b>	<b>SSIM</b>	<b>MS-SSIM</b>
<b>CLIC Mobile test (patches 32)</b>	2	33.75	0.92	0.97
<b>Kodak (patches 32)</b>	2	31.46	0.88	0.96
<b>BD0</b>	2	40.24	0.97	0.99
<b>BD1</b>	2	35.00	0.91	0.98
<b>BD2</b>	2	27.53	0.91	0.97
<b>BD3</b>	2	33.27	0.91	0.97
<b>BD4</b>	2	30.16	0.90	0.97

# Curva Distorção vs. Taxa

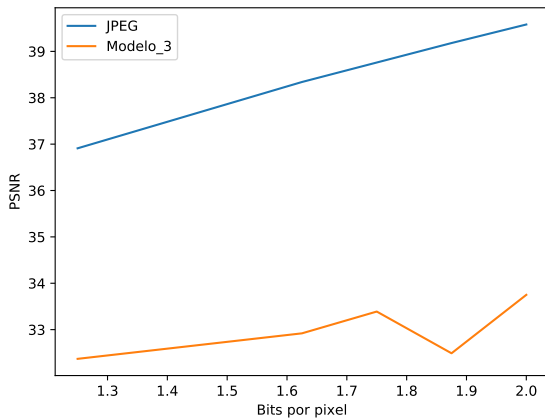


Figura: Comparação Modelo 3 e *JPEG* na métrica PSNR.

# Curva Distorção vs. Taxa

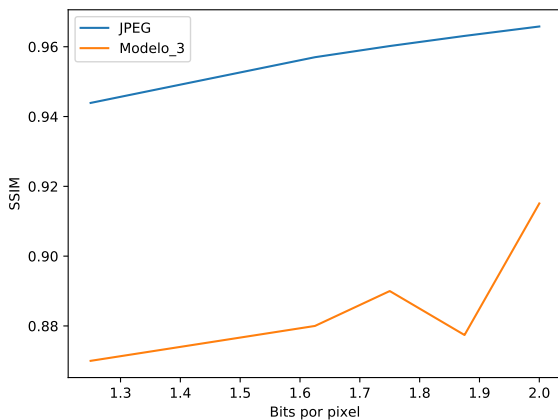


Figura: Comparação Modelo 3 e *JPEG* na métrica SSIM.

# Conclusão

# Limitações do Trabalho

- Custo computacional superior aos codecs clássicos.
- Algoritmos flexíveis, porém quantização é uma operação não diferenciável o que dificulta treinamento de redes neurais.

# Trabalho de Graduação 2

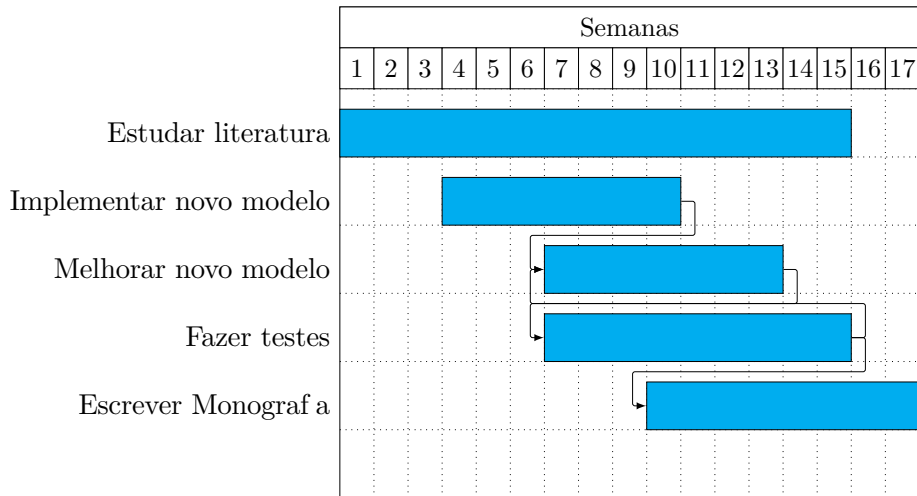


Diagrama de Gantt



# References I



Khalid Sayood

Introduction to Data Compression

Morgan Kaufmann



Gregory K. Wallace

The JPEG Still Picture Compression Standard

In: IEEE Transactions on Consumer Electronics



C. Christopoulos and A. Skodras and T. Ebrahimi

The JPEG2000 Still Image Coding System: An Overview

In: IEEE Transactions on Consumer Electronics



Fabrice Bellard

BPG image format

In: BPG

# References II



WebP

Compression Techniques

In: [Compression](#)



Michael A. Nielsen

Neural Networks and Deep Learning

In: [Determination Press](#)



Ian Goodfellow and Yoshua Bengio and Aaron Courville

Deep Learning

In: [MIT Press](#)




Alexandre Zaghetto

Introduction to Image Processing

In: [GitHub Repository](#)


# References III

 [Wikimedia Commons](#)  
Wikimedia Commons, The Free Media Repository

In: [DCT Figure](#)

 [Mike Pound - Computerphile](#)  
The Problem with JPEG - Computerphile

In: [Computerphile Video](#)

 [Leslie N. Smith](#)  
Cyclical Learning Rates for Training Neural Networks

In: [Proceedings - IEEE Winter Conference on Applications of Computer Vision](#)

 [Brad Kenstler](#)  
Cyclical Learning Rate (CLR)

In: [GitHub Repository](#)

# References IV



Lilian Weng

From Autoencoder to Beta-VAE

In: [Blog](#)



CS231n

CS231n Convolutional Neural Networks for Visual Recognition

In: [CS231n](#)



Toderici, George and Vincent, Damien and Johnston, Nick and Jin Hwang, Sung and Minnen, David and Shor, Joel and Covell, Michele

Variable Rate Image Compression with Recurrent Neural Networks

In: [ICLR 2016](#)