2024 하계 세미나

Image compression with text information

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
김현빈

Contents

- Introduction
  - Image Compression
    - Pipeline of traditional Image Compression: JPEG
  - Neural Image Compression
  - Neural Image Compression with text information
    - Background: CLIP
- Paper Review
  - Towards image compression with perfect realism at ultra-low bitrates (ICLR 2024)
  - Neural Image Compression with Text-guided Encoding for both Pixel-level and Perceptual Fidelity (ICML 2024)

Introduction

- Image Compression?
  - 이미지를 효율적으로 저장하고 전송하기 위해 이미지 파일 크기를 줄이는 과정
  - 손실 압축(Lossy Compression)과 무손실 압축(Lossless Compression)으로 구분
    - 무손실 압축(Lossless Compression)
      - 원본 이미지의 데이터를 잃지 않으면서 파일 크기를 줄이는 방법. Ex) PNG
    - 손실 압축(Lossy Compression) – Today's topic
      - 일부 데이터를 버리고 이미지의 시각적 품질을 유지하면서 파일 크기를 줄이는 방법
      - 영구적으로 데이터가 손실됨. Ex) JPEG

JPEG vs PNG*

Introduction

- Image Compression in Real-World Application
  - Send high-quality videos efficiently → Low latency, better experience
    - Video Streaming
      - YouTube, Netflix, Tiktok, Twitch, Meta
    - Video conference
      - Zoom, Skype, Webex, MS teams
    - Website
      - Naver, Google, Kakao
      - Aliexpress, Amazon, Baemin
    - Gaming: Download game faster
      - Steam, Nexon, Smilegate
  - Optimize storage space by compressing images
    - Cloud Storage
      - Google Drive and Dropbox

Pipeline of traditional Image Compression: JPEG

1) http://www.xway.kr/pds2/26694
2) https://mvje.tistory.com/86?category=1033082

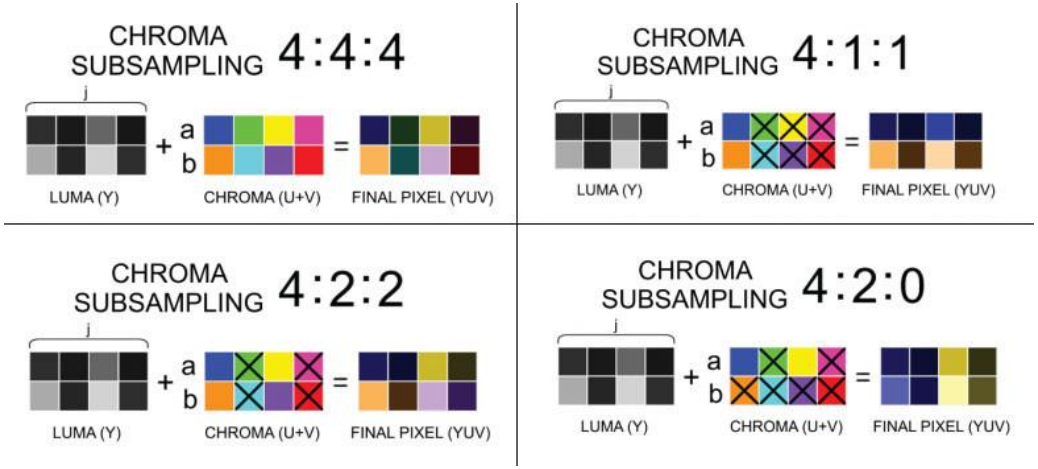
Pipeline of traditional Image Compression: JPEG

- 색공간변경 (RGB to YCbCr) + Chroma Subsampling (Downsampling)
- DCT (Discrete Cosine Transformation)
- Quantization
- Encoding
  - Zigzag Scanning
  - Huffman coding
- Decoding & Inverse transform

Pipeline of traditional Image Compression: JPEG

색공간 변경 + Downsampling → DCT → Quantization → Entropy encoding → Decoding + Inverse transform

- 색공간변경 (RGB to YCbCr) + Chroma Subsampling (Downsampling)
  - 색공간 변경 (RGB to YCbCr)
    - Y: Brightness, Cb,Cr: Chroma
  - Chroma Subsampling (Downsampling)
    - 사람의 눈은 색상 정보보다 밝기 정보에 더 Y 성분은 유지하고, Cb, Cr 성분만 Subsampling 하여 1차 압축을 수행

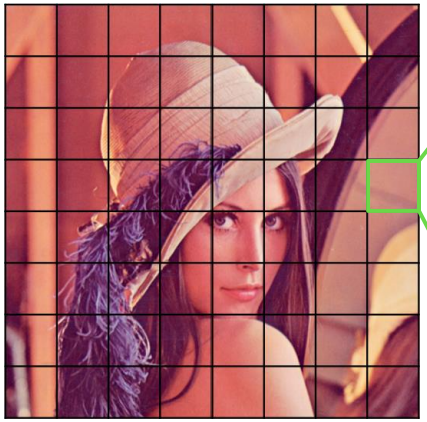


Chroma Subsampling

Pipeline of traditional Image Compression: JPEG

색공간 변경 + Downsampling → **DCT** → Quantization → Entropy encoding → Decoding + Inverse transform

- DCT (Discrete Cosine Transformation)
  - 이미지를 8x8 혹은 16x16 블록으로 분할하고, 각 블록에 대해 DCT 변환을 수행함
  - DCT를 통해 공간정보를 주파수정보(x방향, y방향 주파수성분의 크기)로 변환함
  - 주파수별 정보
    - Low Frequency: 이미지의 배경, 넓고 균일한 색상 영역, 부드러운 그라데이션 등을 표현
    - High Frequency: 이미지의 가장자리, 세밀한 텍스처, 빠른 색상 변화 등을 표현



Subsampled Image

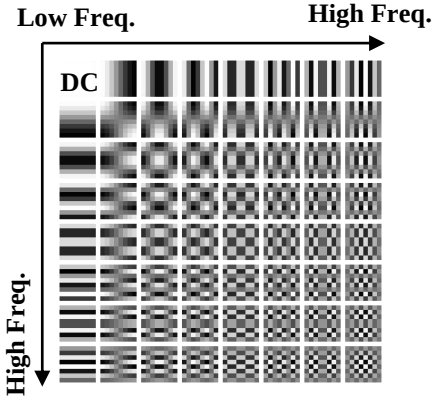
139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

Image Block (8x8)

DCT

1259.6	-1	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1	-1
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	0.4

DCT Applied



1) http://www.xway.kr/pds2/26694
2) https://mvje.tistory.com/86?category=1033082

Pipeline of traditional Image Compression: JPEG

- Quantization
  - Low Frequency는 작은 수로 나누고, High Frequency는 큰 수로 나눈 후 반올림
  - 사람은 고주파에는 둔하고, 저주파에는 더 민감하기 때문에, 고주파 정보를 일부 제거하는 방법으로 압축을 진행
  - 색상이 변화하는 등의 변화가 고주파 영역 정보인데, 이것이 소실되며 block artifact가 발생하는 원인

1259.6	-1	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1	-1
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	0.4

DCT Applied

÷

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Quantization Table

→

-79	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantized applied

÷ elementwise division

Pipeline of traditional Image Compression: JPEG

색공간 변경 + Downsampling → DCT → Quantization → **Entropy encoding** → Decoding + Inverse transform

- Entropy Coding
  - Zigzag Scanning
    - 지그재그 방향으로 스캐닝하여 1차원 벡터로 변환
  - Huffman encoding
    - 자주 등장하는 수의 정보량을 줄이는 Huffman 부호화를 적용하여 데이터를 추가적으로 압축. 이 때는 정보의 손실이 발생하지 않음

-79	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantized applied

Zigzag Scanning

[-79, 0, -2, -1, -1, -1, 0, 0, -1, -1]

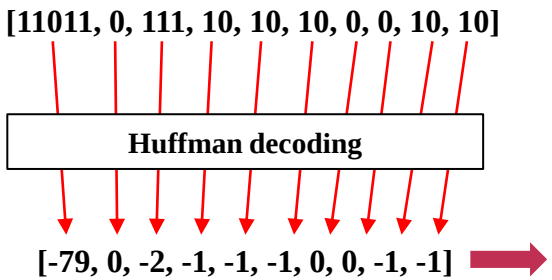
Huffman encoding

[11011, 0, 111, 10, 10, 10, 0, 0, 10, 10]

1) http://www.xway.kr/pds2/26694
2) https://mvje.tistory.com/86?category=1033082

Pipeline of traditional Image Compression: JPEG

- Decoding & Inverse transform
  - 앞서 수행했던 과정의 역과정을 수행하면서, 이미지를 복원
    - Huffman decoding
    - Dequantization
    - I-DCT(Inverse - Discrete Cosine Transformation)



-79	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantized applied

De-quantization

1264	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	0	0	0	0	0	0
-14	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

DCT applied

I-DCT

Original Block

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

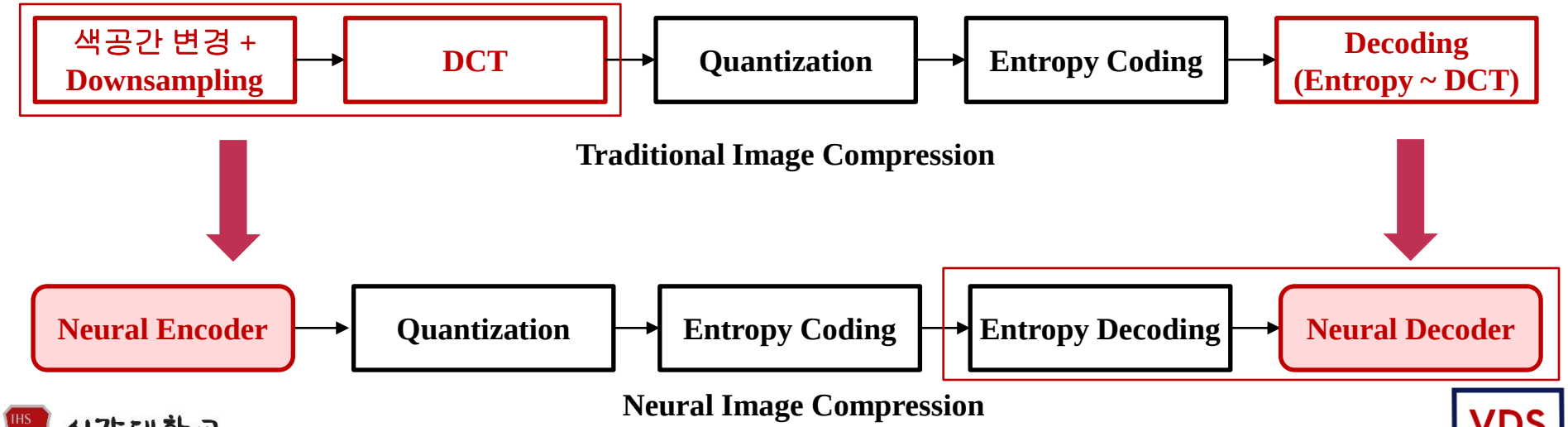
142	143.9	146.9	150	152.3	153.5	153.8	153.7
148.7	150.2	152.6	154.9	156.4	156.8	156.4	156
157	158	159.5	160.6	160.7	159.9	158.7	157.8
161.6	162.2	162.9	162.9	162	160.2	158.1	156.8
161.6	162	162.4	162.1	160.7	158.4	156.1	154.6
160.1	160	161.2	161.1	159.9	157.9	155.8	154.4
159.7	160.6	161.5	162	161.5	160.1	158.4	157.2
160.2	161.2	162.6	163.6	163.6	162.6	161.3	160.4

Reconstructed Block

Neural Image Compression

Pipeline of Neural Image Compression

- Neural Image Compression
  - 현재 많이 사용하는 Neural Image (lossy) Compression method는 기본적인 JPEG 압축 알고리즘과 동일한 파이프라인을 가짐
  - 기존에 DCT를 쓰던 Transform 과정을 Artificial Neural Network로 대체함
    - Neural network가 이미지로부터 중요한 특징들을 추출하도록 설계되고 학습되어, color subsampling + DCT보다 좀 더 유의미한 정보를 추출할 수 있을 것이라 기대되기 때문
      - AE, VAEs, GAN

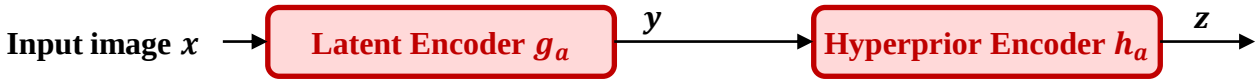


Pipeline of Neural Image Compression

- Neural Image Compression

Neural Encoder

- Latent Encoder
  - 원본 image x를 latent representation y (≈feature)로 변환
- Hyperprior Encoder
  - Latent representation으로부터 hyperprior 추출
    - Hyperprior: Decoding 과정 중에 latent representation을 좀 더 잘 모델링하기 위한 부가 정보

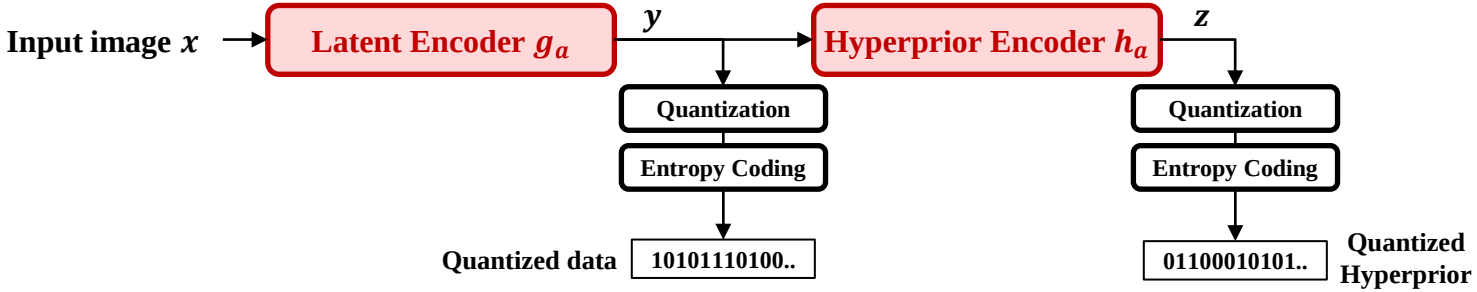


1) BALLÉ, Johannes, et al. Variational image compression with a scale hyperprior. In: *International Conference on Learning Representations*. 2018.

Pipeline of Neural Image Compression

- Neural Image Compression

  - Quantization
    - Latent representation, hyperprior에 quantization을 수행하여 데이터를 압축
  - Entropy Coding
    - 양자화된 latent representation과 hyperprior를 효율적으로 압축
      - Huffman Coding: 계산 복잡도가 낮지만, 압축률이 arithmetic coding에 비해 낮음
      - Arithmetic Coding: 계산 복잡도가 높지만, 압축률이 높음



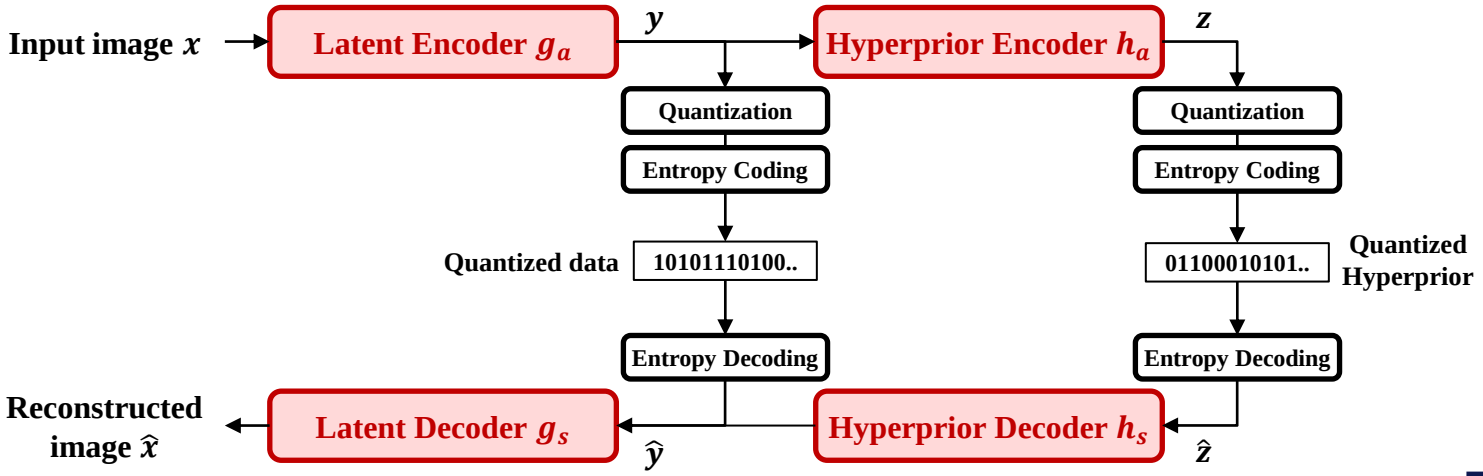
Pipeline of Neural Image Compression

- Neural Image Compression

Neural Encoder → **Quantization** → **Entropy Coding** → **Entropy Decoding** → **Neural Decoder**

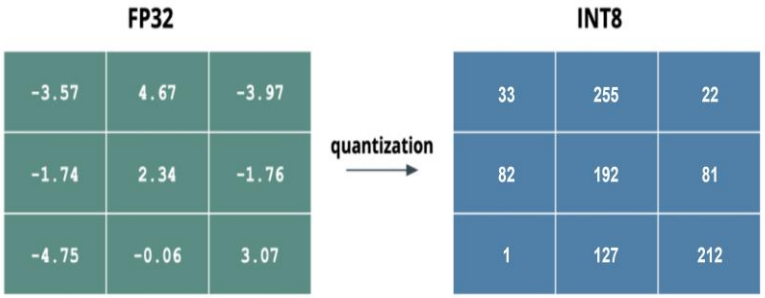
- Hyperprior Decoder
  - Decoding하여 얻은 hyperprior로부터 부가정보에 대한 latent representation을 복원

- Latent Encoder
  - Decoding하여 얻은 latent representation과 hyperprior를 바탕으로 image $\hat{x}$ 를 복원

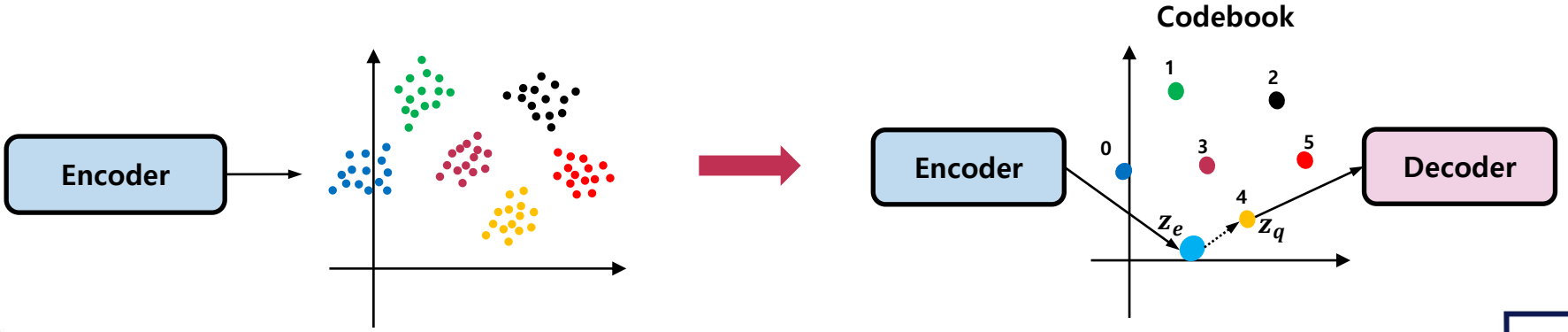


Background

- Quantization
  - 연속적인 값의 집합을 이산적인 값의 집합으로 변환하는 과정
  - Type of Quantization
    - Scalar Quantization
      - 개별 값을 독립적으로 양자화
      - Weight (Model)에 적용하여 경량화 가능



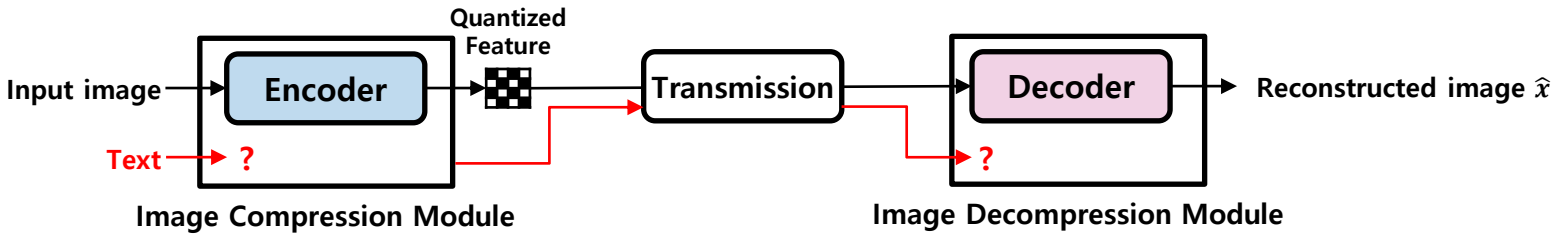
    - Vector Quantization
      - 어떤 값의 집합을 하나의 대표 값으로 양자화 (ex: Clustering - Centroid)
      - 어떤 벡터 V=(v1,v2,v3, …)를 Codebook에 존재하는 벡터 중 가장 가까운 벡터의 index i로 변환하여 저장



Neural Image Compression with text information

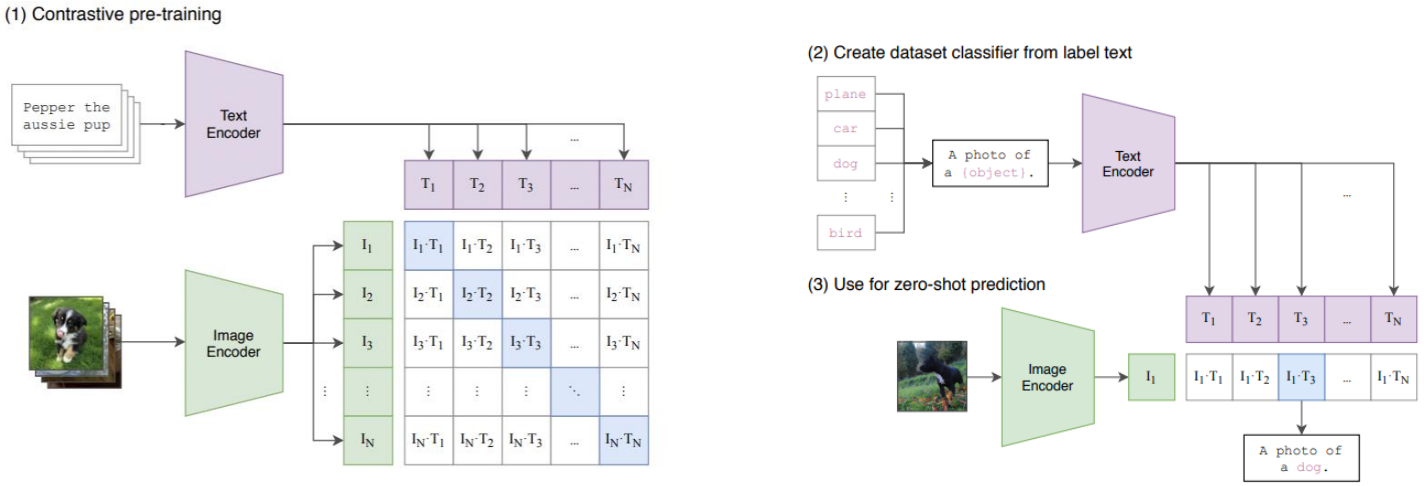
Neural Image Compression with text information

- Image Compression with text information
  - Neural Image Compression에서 text information을 추가하여 성능을 개선
  - 아래의 두 가지 방향에 대한 논문을 금일 소개하고자 함
    - 텍스트 정보를 활용하여 이미지의 semantic information을 추가함으로써, bitrate가 극단적으로 낮아져도 아티팩트(block, ringing)를 최소화하고 품질을 유지
    - 텍스트 기반 이미지 압축을 수행하여 지각적 품질을 높이면서 픽셀 단위의 정확도 유지



Neural Image Compression with text information

- Background
  - CLIP
    - Image embedding과 text embedding 간의 contrastive learning을 통해 모델을 학습
    - 자연어를 guidance로 활용하여, image encoder가 좀 더 의미 있는 feature를 추출하도록 학습
    - Large-scale dataset으로 학습하여 강력한 representation 능력을 갖고 있어, zero-shot vision task에서 활용 가능



Paper Review

Towards image compression with perfect realism at ultra-low bitrates (ICLR 2024)

Marlène Careil, Matthew J. Muckley , Jakob Verbeek , Stéphane Lathuilière

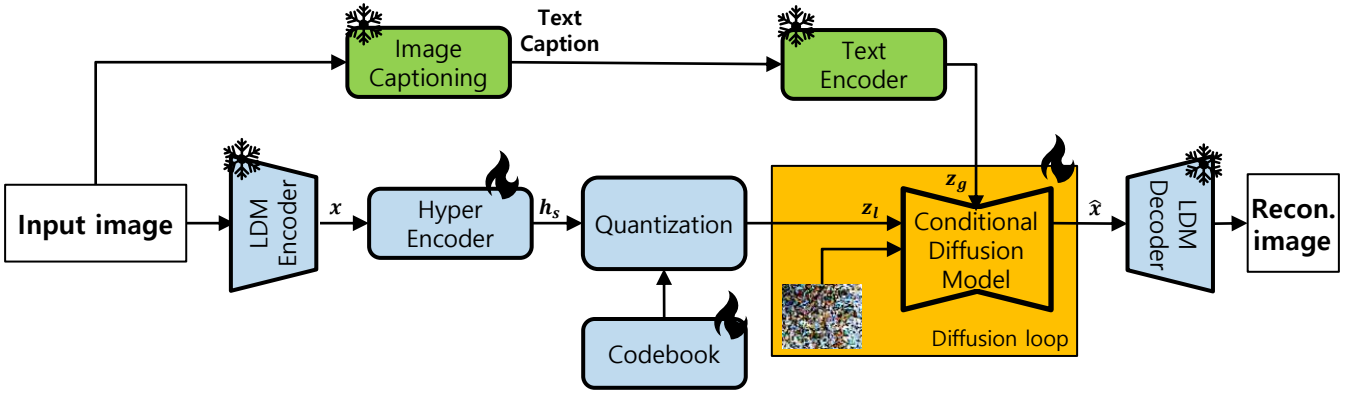
1) Careil, Marlene, et al. "Towards image compression with perfect realism at ultra-low bitrates." *The Twelfth International Conference on Learning Representations*. 2023.

Introduction

- Motivation
  - 기존 image codec의 문제
    - Typical image codecs optimize the balance between bitrate (compression ratio) and distortion
    - In traditional image compression methods, lower bitrates usually result in a significant degradation of visual quality due to compression artifacts (e.g. block, ringing effects)
  - Diffusion Models
    - 압축된 데이터에서 세부 정보를 복원하는 데 뛰어난 능력을 가지고 있음
    - Latent diffusion 기반 모델은 text를 기반으로 이미지를 생성할 수 있으며, 뿐만 아니라 diffusion 과정에 추가적인 정보를 제공하여 압축된 데이터를 재구성하는데 도움을 줄 수 있음

Method

- Encoding local and global context
  - Local spatial encoding - Encode image to get local context
    - Encode the image to capture local context
      - LDM encoder: Image (512 × 512) → Feature $\boldsymbol{x}$ (4 × 64 × 64)
      - Hyper encoder: $\boldsymbol{x} \rightarrow \boldsymbol{h_s}$ (h × w), 보다 더 작은 크기의 feature로 압축
      - Vector quantization: $\boldsymbol{x} \rightarrow \boldsymbol{z_q}(= \boldsymbol{z_l})$ ($\sim log_2 V$, V: size of codebook)
  - Global encoding with image captioning
    - Use an image caption model to generate text captions from the image
      - Image caption model (BLIP-2 or IDEFICS): Image → Text captions
      - Text encoder: Text caption → $\boldsymbol{z_g}$



Method

- Decoding with a diffusion model
  - Feed local and global feature to diffusion model
    - Upsample local feature $\boldsymbol{z_l}$ and concatenation with the latent features $\boldsymbol{x_t}$ of the current time step
    - Pass the global feature $\boldsymbol{z_g}$ to the diffusion model as the condition through cross-attention layers of the pre-trained diffusion model

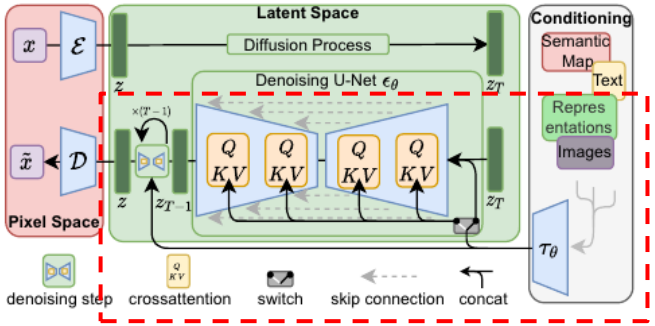


Figure 3. We condition LDMs either via concatenation or by a more general cross-attention mechanism. See Sec. 3.3

  - Classifier-free guidance
    - To enable classifier-free guidance, we train PerCo by dropping the text-conditioning in 10% of the training iterations
    - When dropping the text-conditioning we use a constant learned text-embedding instead

$$\hat{\epsilon}_\theta = \epsilon_\theta(\boldsymbol{x}_t, (\boldsymbol{z}_l, \emptyset), t) + \lambda_s \left(\epsilon_\theta(\boldsymbol{x}_t, (\boldsymbol{z}_l, \boldsymbol{z}_g), t) - \epsilon_\theta(\boldsymbol{x}_t, (\boldsymbol{z}_l, \emptyset), t)\right)$$

Method

- Loss setting
  - Traditional loss setting in neural compression
    - Linear combination of rate (compression ratio) and a distortion term.
    - We use fixed size latent vector $\boldsymbol{z}$(size of codebook). Therefore, $\boldsymbol{L_R}$ can be considered fixed

$$\mathcal{L}_{\mathrm{RD}} = \mathbb{E}_{P_{\boldsymbol{x}}}[\mathbb{E}_{P_{\boldsymbol{z}|\boldsymbol{x}}}\mathcal{L}_{\mathrm{R}}(\boldsymbol{z}) + \lambda\mathcal{L}_{\mathrm{D}}(\hat{\boldsymbol{x}}(\boldsymbol{z}), \boldsymbol{x})]$$

  - Vector quantization loss
    - Codebook learned using the vector quantization loss
    - We update codebook with EMA, and thus the first term in Equation (4) is no longer used

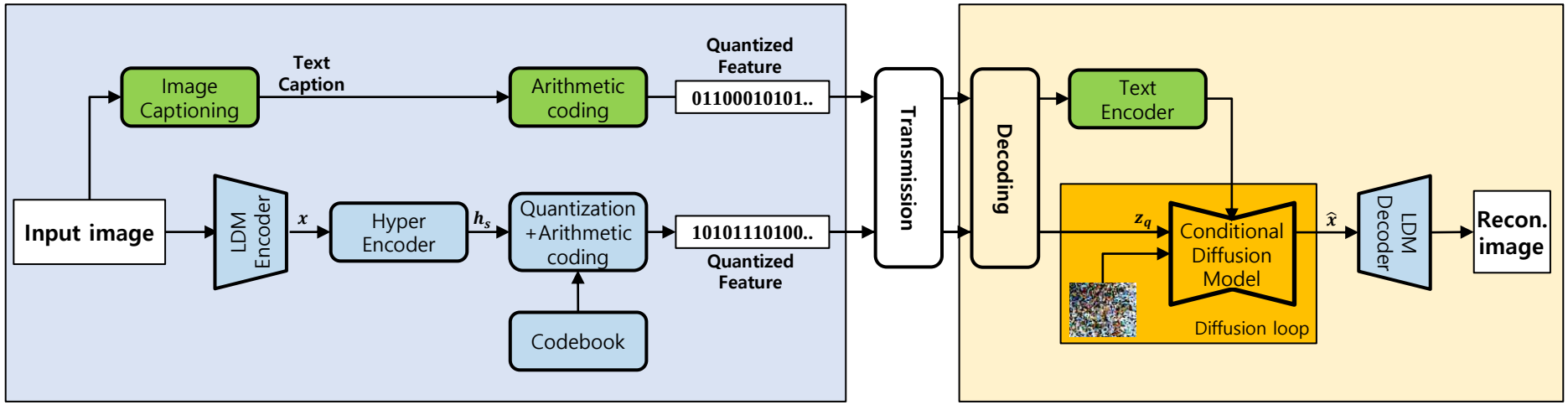
$$\mathcal{L}_{\mathrm{VQ}} = \mathbb{E}_{\boldsymbol{h}_s}\left[\|sg(\boldsymbol{h}_s) - \boldsymbol{z}_q\|_2^2 + \|sg(\boldsymbol{z}_q) - \boldsymbol{h}_s\|_2^2\right]$$

  - Distortion loss in diffusion process
    - For every diffusion step t, we compute an estimation and minimize error between estimation and gt

$$\mathcal{L}_{\mathrm{Diff}}^t \propto \mathbb{E}_{P_{\boldsymbol{x}}}\mathbb{E}_{P_{\boldsymbol{z},\boldsymbol{x}_t|\boldsymbol{x}}}\mathop{\mathbb{E}}_{\epsilon\sim\mathcal{N}(0,1)}\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, \boldsymbol{z}, t)\|_2^2.$$

Method

- Full pipeline of the model



Experiment

- Metrics
  - Quantify image quality
    - FID, KID
  - Measure distortion
    - LPIPS, MS-SSIM
  - CLIP score
    - Measure global alignment of reconstructed samples with ground truth caption (Higher is better)

- Dataset
  - Kodak
  - MS-COCO

Experiment

- Comparison with SOTA
  - PerCo는 낮은 bitrate(<0.04 bpp)에서 SOTA에 비해 FID 및 KID가 낮으며, bitrate와 관계없이 일관적인 성능을 보이는 것을 확인
  - CLIP 및 mIoU 지표에서도 다른 모든 방법에 비해 일관적으로 높은 성능을 보임

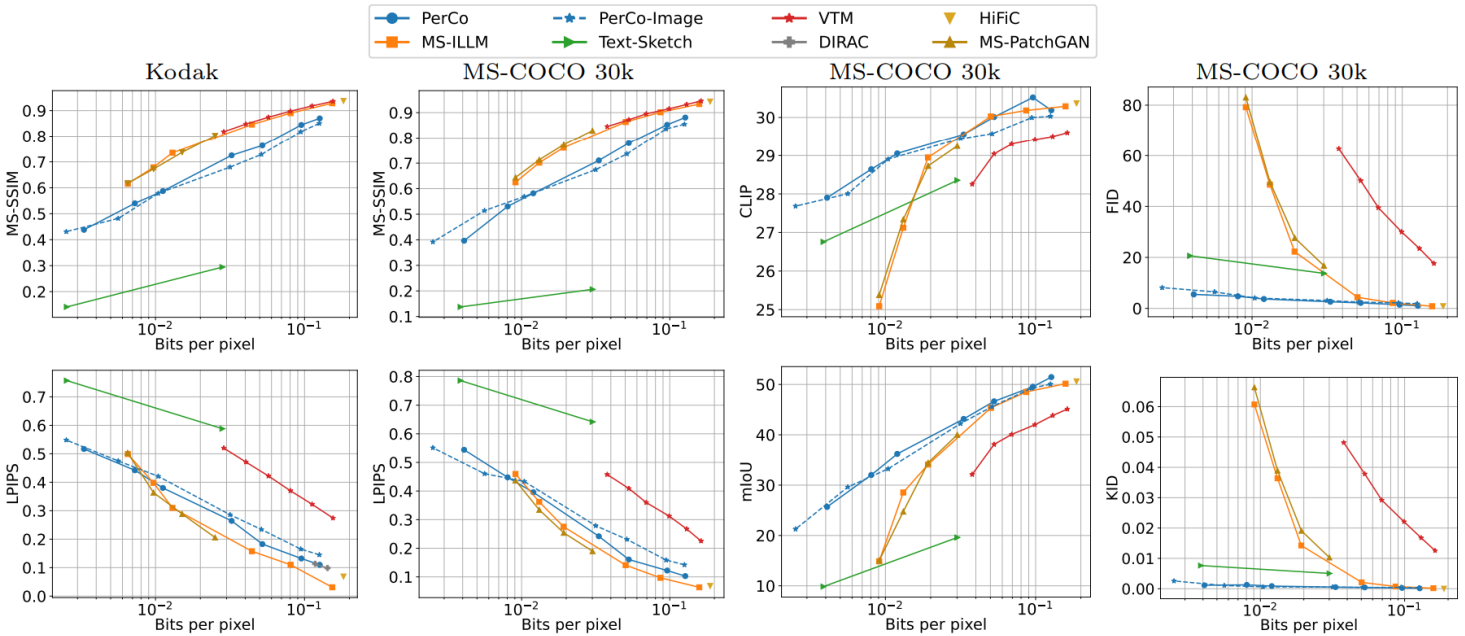


Figure 3 Evaluation of PerCo and other image compression codecs on Kodak and MS-COCO 30k.

Experiment

- Ablation studies
  - Diversity in reconstructions
    - 이상적으로는 여러 번 샘플링해도 원본 입력 이미지와 동일하게 복원해야 함
    - 모델 용량, 학습 데이터, bitrate의 제한으로 인해 이는 발생하지 않음
    - 특히 낮은 bitrate 에서는 원본 샘플에 대한 정보가 적어, 샘플 간의 다양성이 더 커짐

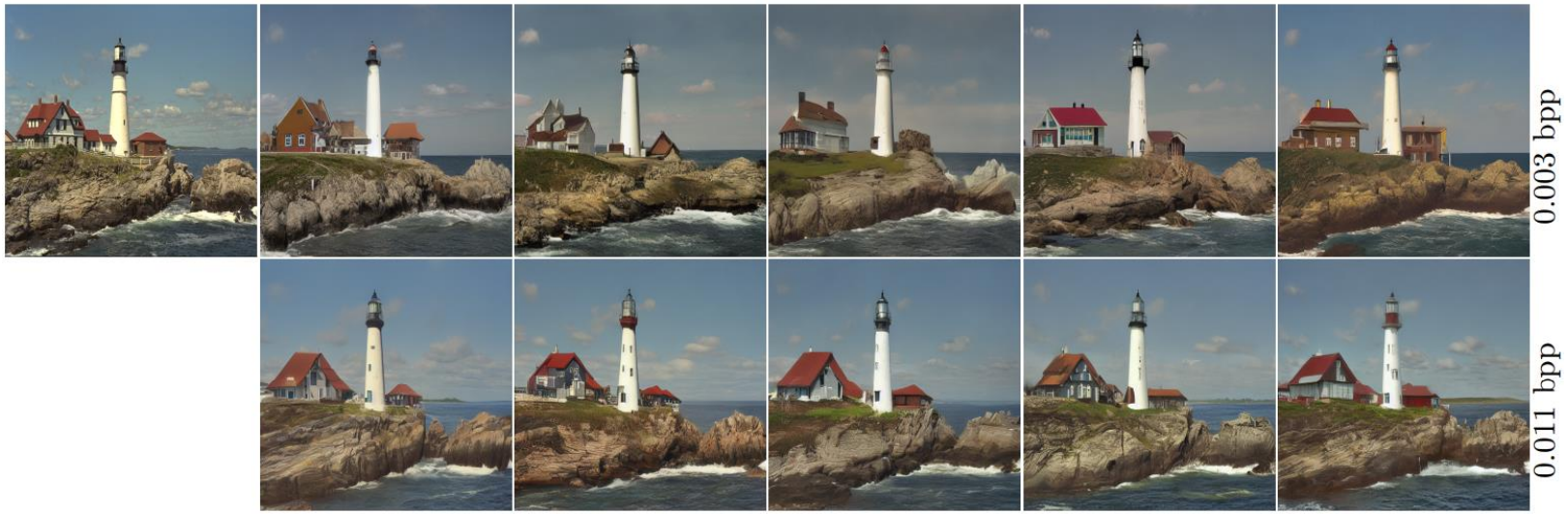
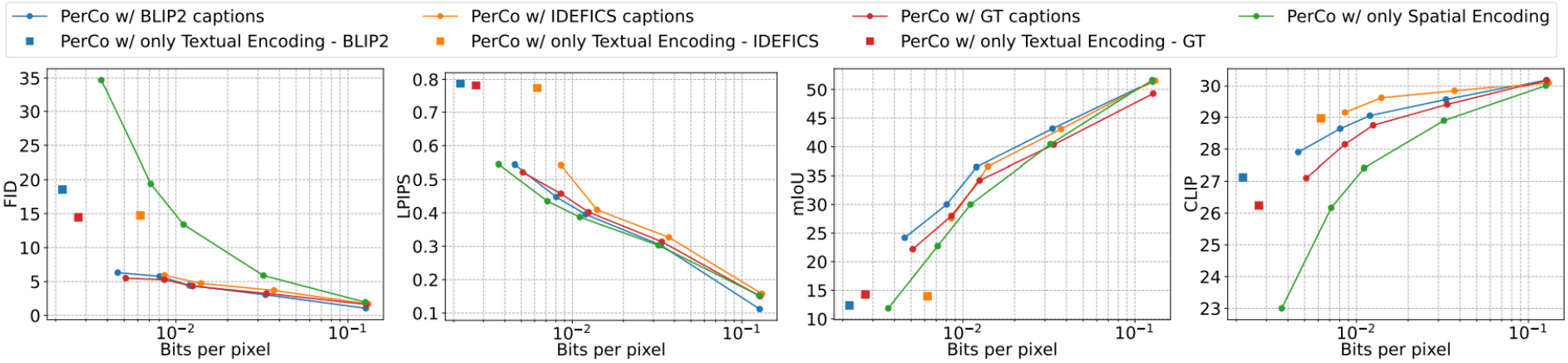


Figure 6 Reconstructions of a Kodak image at 0.003 and 0.011 bpp. Best viewed zoomed in.

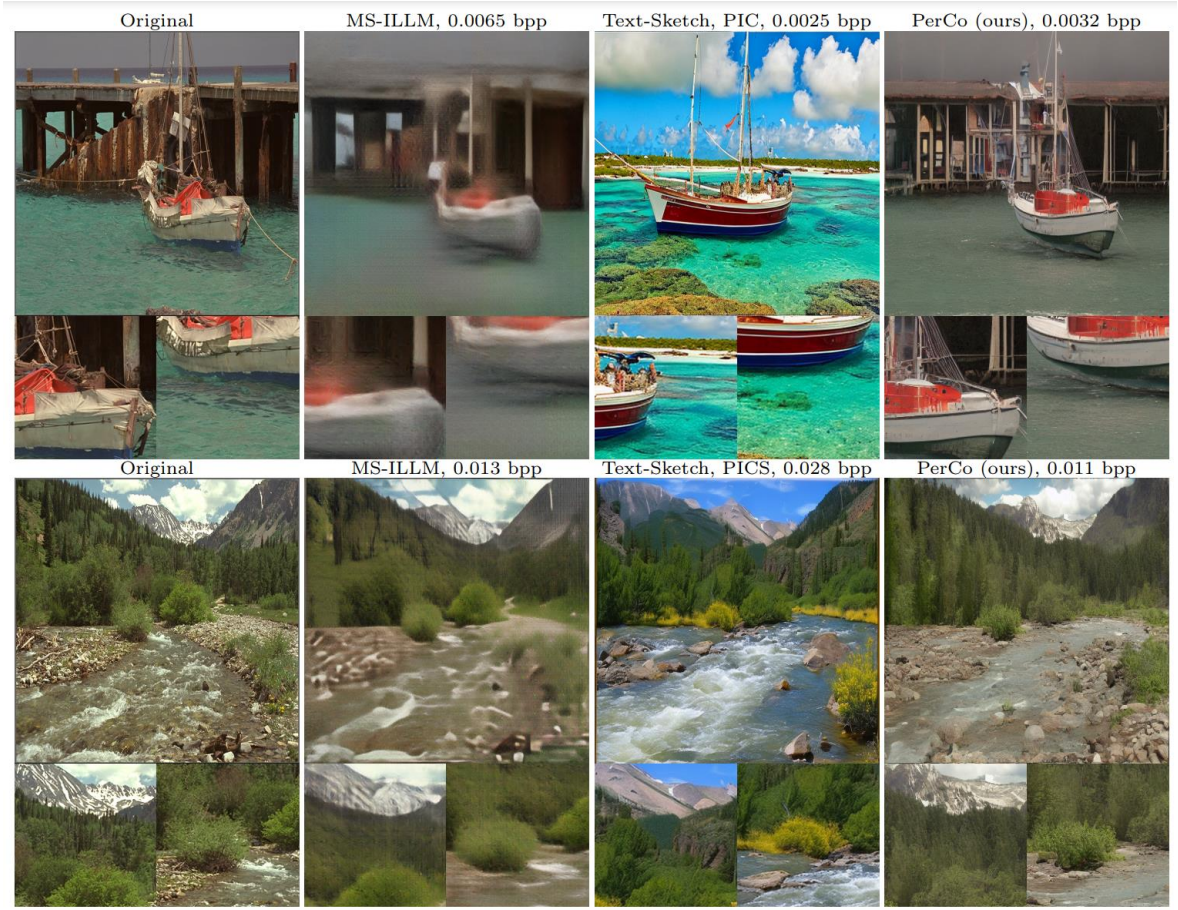
Experiment

- Ablation studies
  - Conditions of modality
    - PerCo를 평가할 때 text 및 spatial(image) condition을 별도로 사용하여 기여도를 분석
    - Generated caption: BLIP-2, IDEFICS & GT Caption
    - Spatial Encoding만 사용하는 경우, FID와 CLIP 점수가 떨어짐
    - Text Encoding 사용하는 경우, LPIPS와 mIoU에서 성능이 낮은 것을 확인
    - Caption에 상관없이 FID는 거의 유사하며, BLIP은 LPIPS, mIOU에서, 자세한 설명을 하는 IDEFICS는 CLIP 점수가 높은 것을 확인



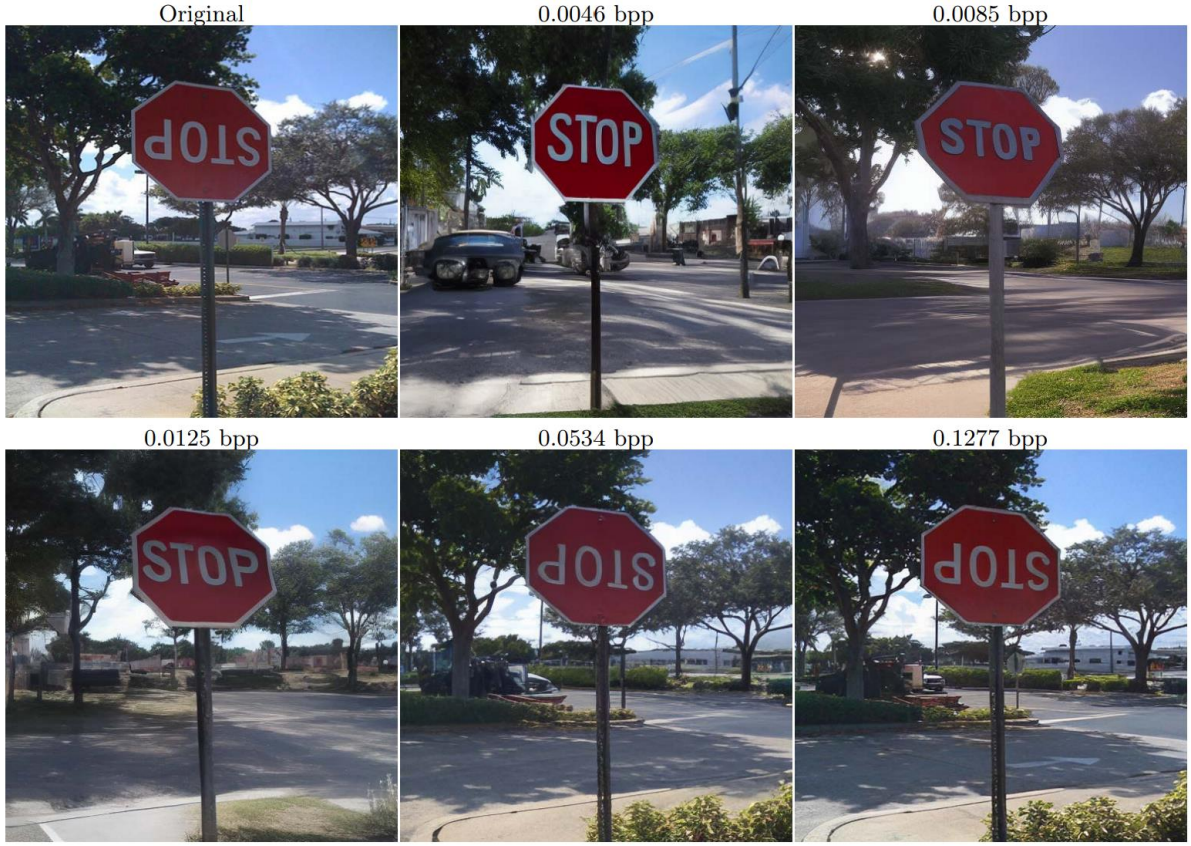
Experiment

- Qualitative Results
  - 극단적으로 낮은 bitrate에서도, 타 모델에 비해 품질을 유지하면서 원본의 semantic information을 최대한 유지하는 경향을 확인할 수 있음



Experiment

- Qualitative Results
  - Bitrate 변화에 따른 결과 비교, 극단적으로 bitrate가 낮아도 image quality는 높지만, 원본 이미지의 정보 손실이 많이 발생한 것을 확인



Paper Review

Neural Image Compression with Text-guided Encoding for both Pixel-level and Perceptual Fidelity (ICML 2024)

Hagyeong Lee, Minkyu Kim, Jun-Hyuk Kim, Seungeon Kim, Dokwan Oh, Jaeho Lee

Introduction

- Motivation
  - Neural Image Compression with text information 방법은 복원된 이미지의 perceptual quality를 향상시켰지만, pixel 단위의 정확도가 매우 떨어져 사용이 제한적임
  - 기존 방법론의 경우 decoding 과정 위주로 text 정보를 주어 압축하면서 발생한 손실된 정보 중 global semantic information을 text로 채워주면서 성능을 개선함
  - 하지만 text로부터 얻은 information이 원본 이미지와 상충되거나, 다르게 작용하는 경우가 있어 pixel 단위의 정확도가 떨어짐

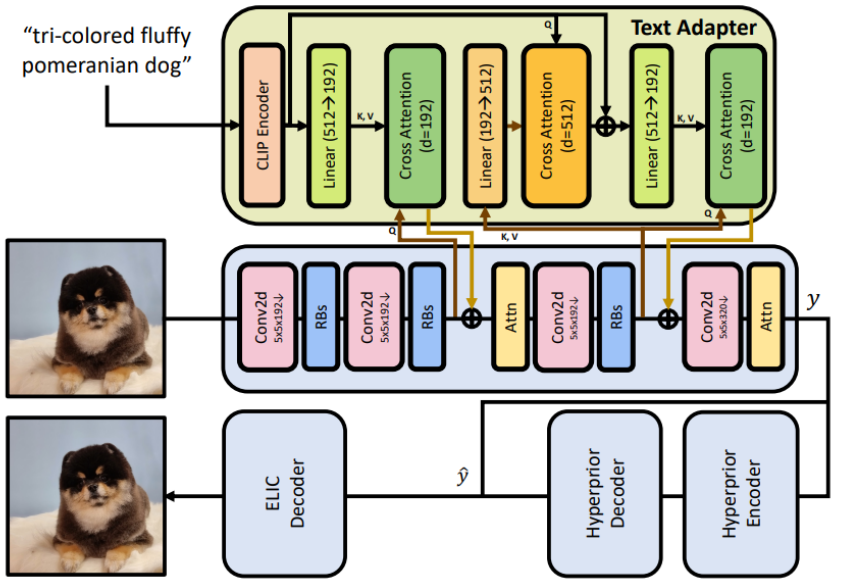
Method - TACO

- Text information injection – baseline인 ELIC encoder에 text adapter를 추가
  - CA1: Text → Image - Updates image features based on text tokens
    - Text caption으로부터 clip text embedding을 추출한 후, linear layer를 거친 후 CA 수행
  - CA2: Image → Text - Updates text tokens based on image features
  - CA3: Text → Image - Updates image features based on updated text tokens
    - CA (Cross Attention)
- Loss: Joint Image -Text Loss
  - Standard loss
    - Rate: r(·)
    - Distortion: MSE d(·, ·), LPIPS(·, ·)

$$r(\hat{y}) + \lambda \cdot d(x, \hat{x}) + k_p \cdot \mathrm{LPIPS}(x, \hat{x}) + k_j \cdot L_j(x, \hat{x}, c)$$

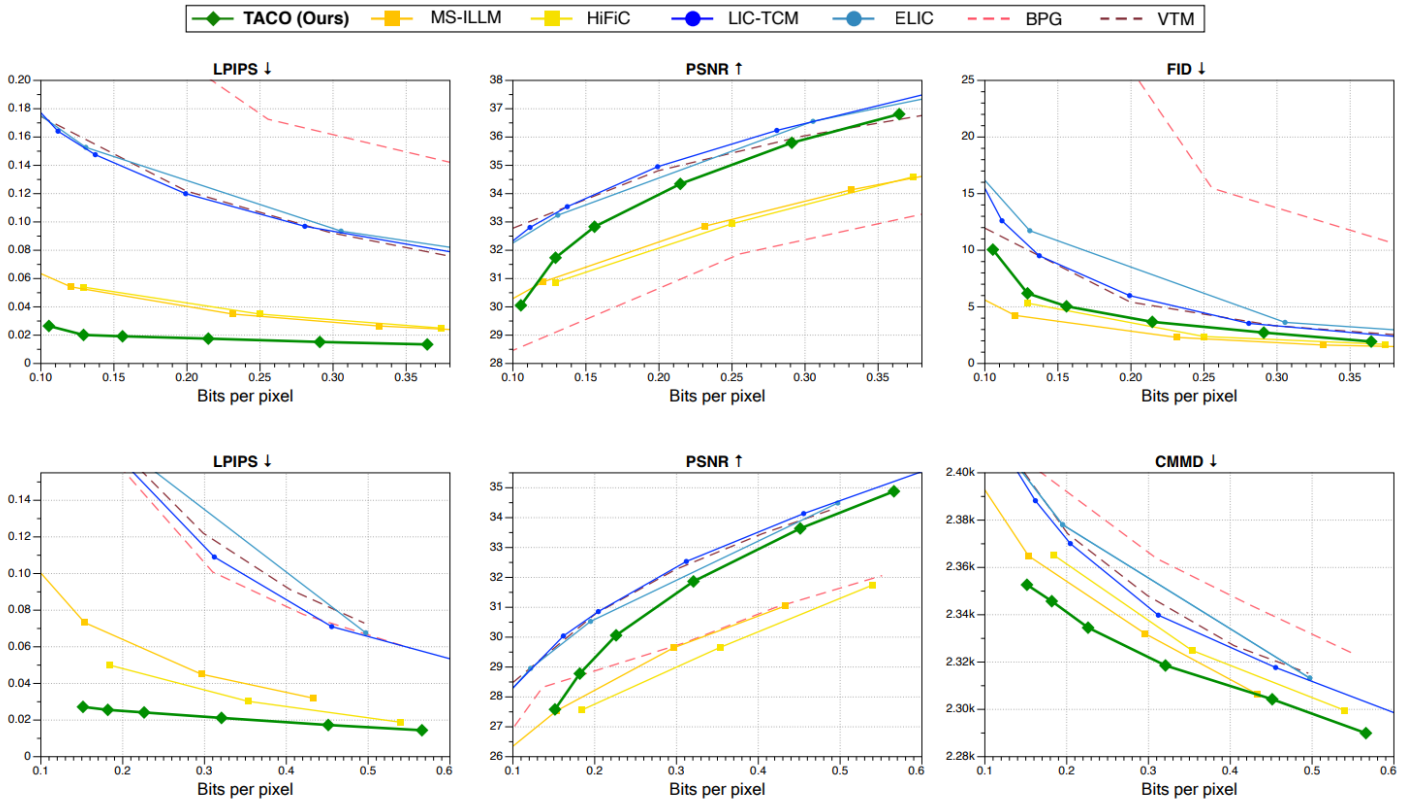
  - Joint image-text loss
    - reconstructed image to be semantically close to the given text and the original image

$$L_j(x, \hat{x}, c) = L_{con}(f_I(\hat{x}), f_T(c)) + \beta \cdot \|f_I(x) - f_I(\hat{x})\|_2.$$



Experiment

- Compare with Image Compression Codecs
  - Dataset – CLIC(위), Kodak (아래)
    - SOTA 대비 PSNR, FID(or CMMD)가 비슷하면서, LPIPS에서 높은 성능을 보이는 것을 확인



Experiment

- Ablation studies
  - Effect of Text Adapter (ELIC + Joint Image -Text Loss) & Joint Image -Text Loss
    - Text Adapter와 JIF 모두 성능에 영향을 미치는 것을 확인
  - Dataset: Kodak

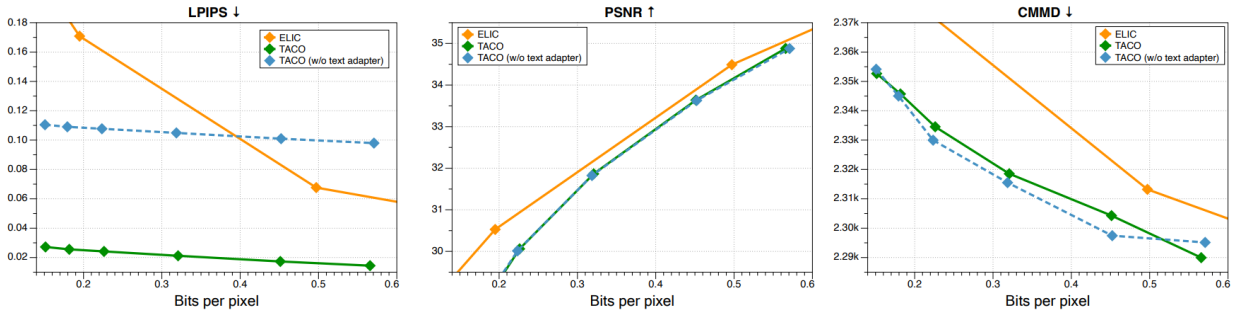


Figure 10. **Without text adapter.** We observe both CMMD and LPIPS substantially degrade, while there is a tiny gain in PSNR.

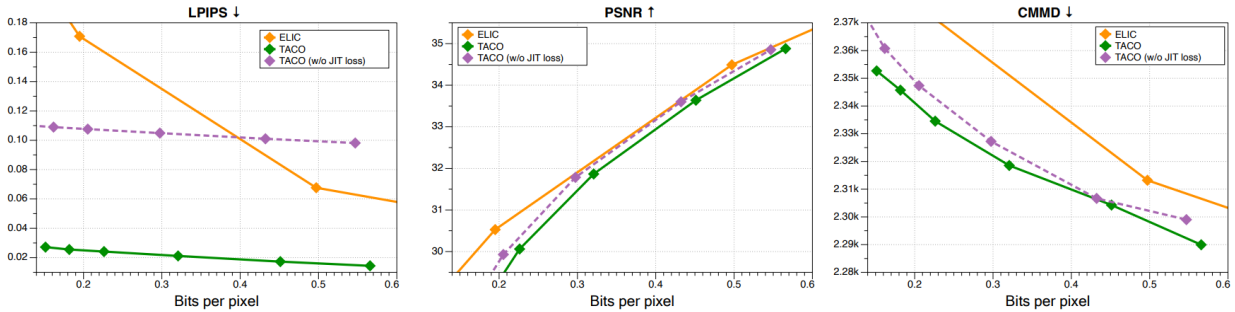


Figure 11. **Without joint image text loss.** We observe LPIPS severely degrades, while PSNR remains similar, with a tiny gain in CMMD.

Experiment

- Ablation studies
  - Caption dependency
    - Human and GPT-4가 가장 높은 성능을 보이나, 큰 차이가 나지 않음
  - Computational cost, memory efficiency
    - Baseline인 ELIC에 비해 computing time이 10%, 파라미터의 경우 약 65M 증가하였으나, 타 모델과 여전히 경쟁력이 있는 모델임을 볼 수 있음

	Human (Chen et al., 2015)	OFA (Wang et al., 2022)	BLIP-2 (Li et al., 2023)	GPT-4 (Achiam et al., 2023)
LPIPS	0.0435	0.0435	0.0435	0.0435
PSNR	27.42	27.41	27.41	27.42

Caption Dependency

	Enc. (ms)	Enc.@High (ms)	Dec. (ms)	Total (ms)
LIC-TCM	112.07	960.99	125.26	237.33
MS-ILLM	70.39	800.45	53.74	124.14
ELIC	71.35	793.41	102.07	173.42
TACO	78.60 (+10.2%)	832.07 (+4.8%)	102.98	181.58

Computational cost

Modules	Parameters (M)
ELIC	36.93
LIC-TCM	45.41
HiFiC/MS-ILLM	181.72
TACO	101.75

Memory Efficiency

Experiment

- Qualitative Results

Conclusion

- 전통적인 image compression
  - Chroma Subsampling → DCT → Quantization → Encoding → Decoding
- Neural image compression
  - Neural encoder → Quantization → Encoding → Decoding → Neural defcoder

- Neural Image Compression with text information
  - PerCo
    - Diffusion model에 text condition, image information을 주어 복원시킴으로써 bitrate가 극단적으로 낮아져도 아티팩트(block, ringing)를 최소화하고 품질을 유지
    - 하지만, bitrate가 낮을 때 의미적으로 같지만 원본과 다르게 복원되는 경우가 발생
  - TACO
    - 기존 neural image compression model에 text adapter를 추가하여 연산량 및 메모리에 큰 영향을 미치지 않고, 픽셀 단위 정확도를 해치지 않으면서 의미 정보를 풍부하게 유지

감사합니다