2024/11/12 multimedia

- shindo

Image Coding For Machine Via Analytics-Driven Appearance Redundancy Reduction

(ICIP 2024)

画像のコントラスト低減を用いた Image Coding for Machines。画像のコントラストを低減させたとしても、画像認識モデルには影響がない場合があることが以前の研究で判明している。

画像のコントラストを低減させることで、エントロピーを抑えることが出来、画像圧縮には有効であると考えられる。そこで、以下図のような End-to-end の学習フレームワークを提案している。

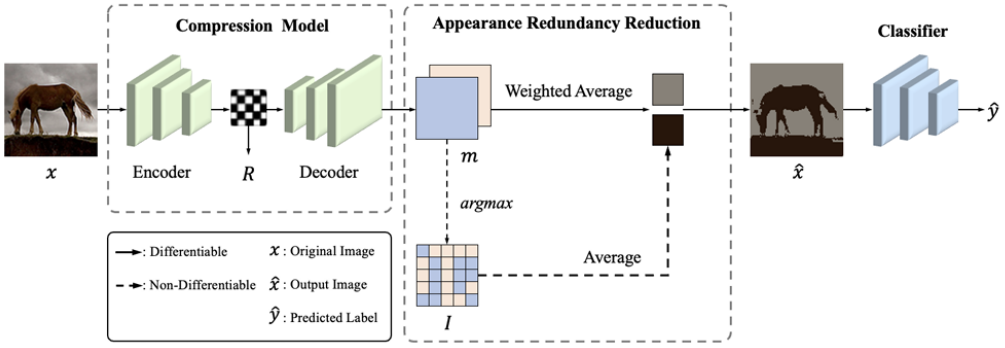

Fig. 1: Architecture of the proposed appearance redundancy reduction-based ICM framework.

Appearance Redundancy Reduction (ARR) は画像のカラー階調を減少させるためのモジュールである。compression model の出力を、CNNを通じて、HxWxCに変換する。Cの数が階調の数になる。この特徴量は分類器の出力のように扱うことが出来、図中のIのように、カラーマップを出力することが出来る。これに基づいて、C個の階調を持った画像を生成することが出来る。上の図の場合、論文中の実装の場合、C＝２である。

つまり、画像は２値画像としてエンコード・デコードされるため、エントロピーを下げることが出来、小さい情報量に圧縮することが出来る。復号画像の一例と、その認識精度は次の図のとおりである。

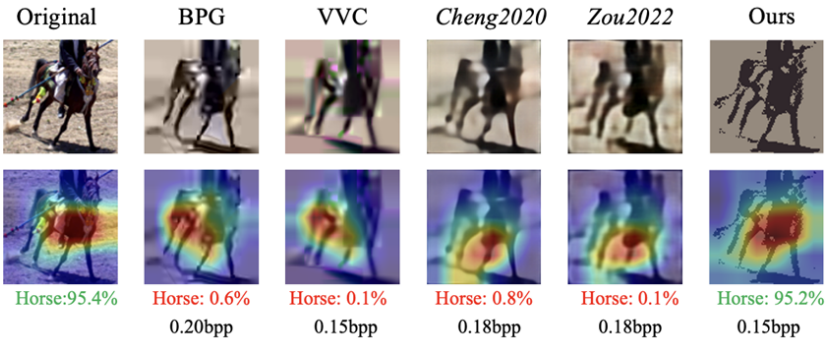

Fig. 3: Visual comparison of different image codecs regarding semantic information preservation.

後段の認識タスクについては、ほとんど画像分類を用いている。後段に物体検出モデルを用いた場合の結果もあるが、レートのかなり小さい場合で実験している。そのため比較手法・提案手法ともに、検出精度がかなり低い。

一方で、画像分類のための画像圧縮手法としては、小さいモデルでかなりの精度が達成できる。

- yenan

Seeing and Hearing: Open-domain Visual-Audio Generation with Diffusion Latent Aligners

(CVPR 2024)

ImageBind: One Embedding Space To Bind Them All

(CVPR 2023)

task: Joint video and audio generation, video to audio, audio to video, image to audio

pre-trained generative AI を組み合わせることで、上記タスクの性能を改善する手法の提案。

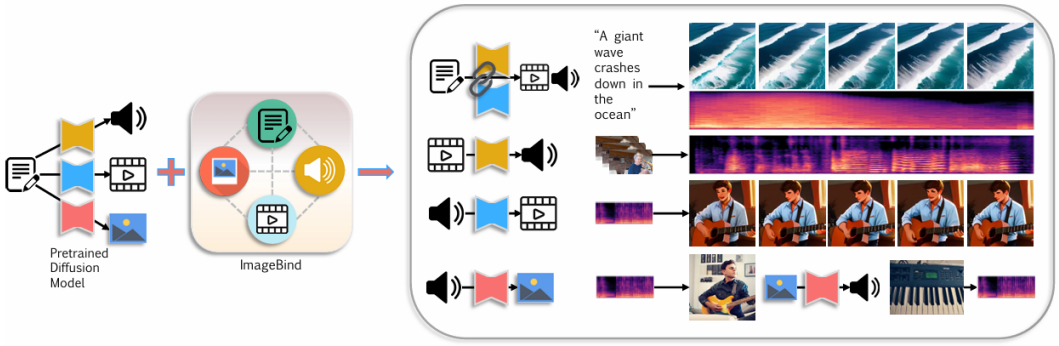

Figure 1. **Overview.** Our approach is versatile and can tackle four tasks: joint video-audio generation (Joint-VA), video-to-audio (V2A), audio-to-video (A2V), and image-to-audio (I2A). By leveraging a multimodal binder, e.g., pretrained ImageBind, we establish a connection between isolated generative models that are designed for generating a single modality. This enables us to achieve both bidirectional conditional and joint video/audio generation.

pre-trained diffusion models (生成モデル達)を pre-trained ImageBind により接続することで、上記タスクを実現。具体的には、各生成モデルのノイズ除去過程におけるノイズを調整することで達成できる。ノイズ調整手法は以下の通り。

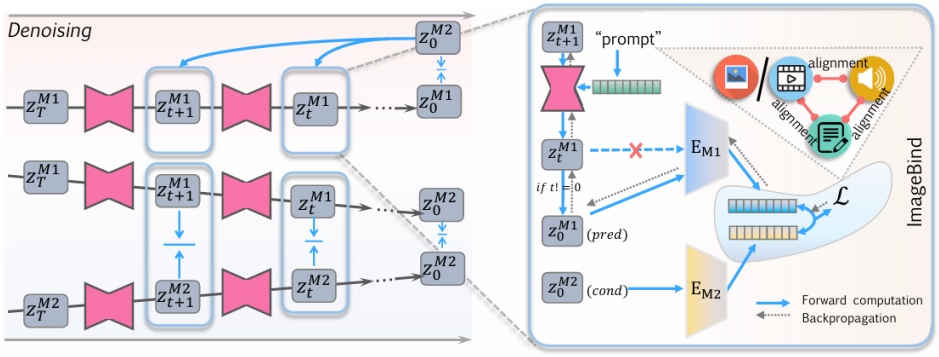

Figure 2. **The proposed diffusion latent aligner.** During the denoising process of generating one specific modality (visual/audio), we adopt the condition information (audio/video) to guide the denoising process. By leveraging the pretrained ImageBind model, we calculate the distance of the generative latent $\mathbf{z}_t^{M_1}$ with the condition $\mathbf{z}_0^{M_2}$ in the shared embedding space of ImageBind. Then we backpropagate the distance value to obtain the gradient of $\mathbf{z}_t^{M_1}$ with respect to the distance.

モダリティ２の生成潜在特徴量をconditionとして使用し、Encoder M2に入力。モダリティ１のnoisy潜在特徴量をEncoder M1に入力。各モダリティからの特徴量の差を小さくするように調整する。

その際、conditionとしては、各モダリティの生成潜在特徴量を用いるのと同時に、text promptも利用することで、より良い生成結果が得られることを確認。

Algorithm 1 Multimodal guidance for joint-VA generation

Require: Learning rate λ_1, λ_2, optimization steps N, warmup steps K, prompt p

1: $\mathbf{y} = \textsc{Emb}(p)$
2: **for** $t = T$ to 0 **do**
3: $\quad \mathbf{z}_t^v \leftarrow \textsc{Denoise}(\mathbf{z}_{t+1}^v, \mathbf{y})$
4: $\quad \mathbf{z}_t^a \leftarrow \textsc{Denoise}(\mathbf{z}_{t+1}^a, \mathbf{y})$
5: $\quad$ **if** $t < K$ **then**
6: $\quad\quad$ **for** $n = 0$ to N **do**
7: $\quad\quad\quad \tilde{\mathbf{z}}_0^v = \frac{1}{\sqrt{\bar{\alpha}_t^v}} (\mathbf{z}_t^v - \sqrt{1-\bar{\alpha}_t^v}\epsilon_t^v)$
8: $\quad\quad\quad \tilde{\mathbf{z}}_0^a = \frac{1}{\sqrt{\bar{\alpha}_t^a}} (\mathbf{z}_t^a - \sqrt{1-\bar{\alpha}_t^a}\epsilon_t^a)$
9: $\quad\quad\quad \mathbf{e}_a, \mathbf{e}_v, \mathbf{e}_p = \textsc{ImageBind}(\tilde{\mathbf{z}}_0^a, \tilde{\mathbf{z}}_0^v, p)$
10: $\quad\quad\quad \mathcal{L}_{\text{joint-va}} = \mathcal{F}(\mathbf{e}_v, \mathbf{e}_p) + \mathcal{F}(\mathbf{e}_v, \mathbf{e}_a) + \mathcal{F}(\mathbf{e}_a, \mathbf{e}_p)$
11: $\quad\quad\quad \mathbf{z}_t^v = \mathbf{z}_t^v - \lambda_1 \nabla_{\mathbf{z}_t^v} \mathcal{L}_{\text{joint-va}}$
12: $\quad\quad\quad \mathbf{z}_t^a = \mathbf{z}_t^a - \lambda_1 \nabla_{\mathbf{z}_t^a} \mathcal{L}_{\text{joint-va}}$
13: $\quad\quad\quad \mathbf{y} = \mathbf{y} - \lambda_2 \nabla_{\mathbf{y}} \mathcal{L}_{\text{joint-va}}$
14: $\quad\quad$ **end for**
15: $\quad$ **end if**
16: **end for**
17: **return** $\mathbf{z}_0^v, \mathbf{z}_0^a$

アルゴリズムは上の図の通りである。10行目で各モダリティの潜在特徴量から、特徴差分を計算。

11,12行目で、ノイズを調整し、各モダリティの入力を考慮する。

- takabe

Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature

(CVPR 2024)

3DGSやNeRFなどの3D空間表現手法は、空間の色を表現する手法として有効であるが、空間中の(物体等の)意味を理解することが難しい。

意味を理解した上でのレンダリングが出来れば、より効率よく空間を復号できる可能性があるし、画像認識タスクへの応用が利く可能性もある。

そこで、2D image 入力の認識モデル(SAMやLSeg)の特徴量蒸留を用いて、3DGSの空間意味理解を試みる。

学習プロセスは下の図のとおりである。

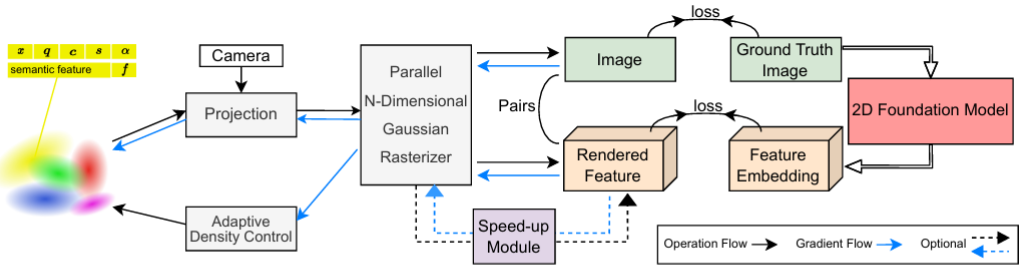

Figure 2. **An overview of our method.** We adopt the same 3D Gaussian initialization from sparse SfM point clouds as utilized in 3DGS, with the addition of an essential attribute: the *semantic feature*. Our primary innovation lies in the development of a Parallel N-dimensional Gaussian Rasterizer, complemented by a convolutional speed-up module as an optional branch. This configuration is adept at rapidly rendering arbitrarily high-dimensional features without sacrificing downstream performance.

図における緑色の損失は、元の3DGS学習の損失である。オレンジ色の損失が、提案されている損失である。2D Foundation model はSAMやLSegなどのセグメンテーションモデルであり、これらの認識器を用いて特徴量の蒸留を行う。つまり、ガウシアンを利用して、セグメンテーションマップを出力する(ような)学習をさせるというもの。

セグメンテーションマップ出力のためには、一般的な3DGSにおけるガウシアンのパラメータに加えて、semantic feature用のパラメータを、各ガウシアンに持たせる必要がある。このパラメータ量の増加と、2D Foundation model の導入による、計算量の大幅増加が課題となっており、speed-up moduleが加えられているが、それでもかなりきつい。

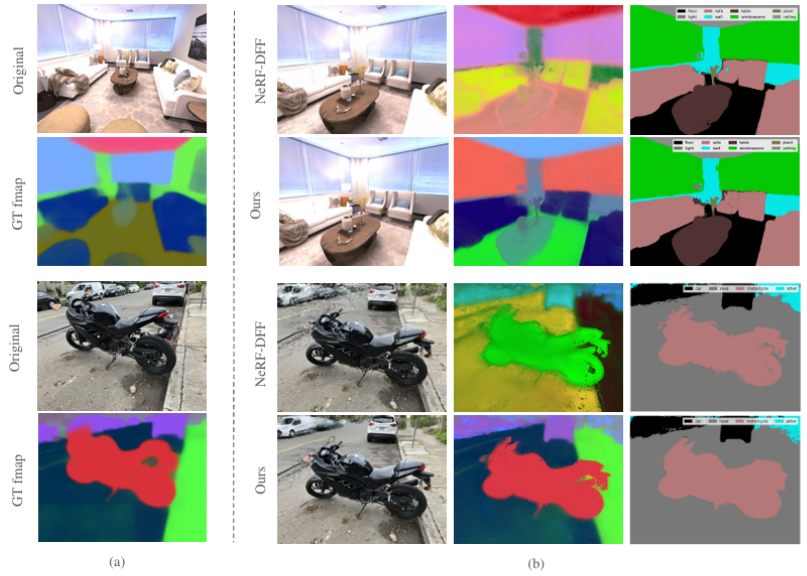

Figure 3. **Novel view semantic segmentation (LSeg) results on scenes from Replica dataset [43] and LLFF dataset [29].** (a) We show examples of original images in training views together with the ground-truth feature visualizations. (b) We compare the qualitative segmentation results using our Feature 3DGS with the NeRF-DFF [21]. Our inference is **1.66**× faster when rendered feature $dim = 128$. Our method demonstrates more fine-grained segmentation results with higher-quality feature maps.

視聴用のレンダリング結果と、セグメンテーションマップの出力結果は、上の図のようになっている。定量的な評価は以下の通り。

Metrics	PSNR(±s.d.)↑	SSIM(±s.d.)↑	LPIPS(±s.d.)↓
Ours (w/ speed-up)	**37.012** (±0.07)	**0.971** (±5.3e-4)	**0.023** (±2.9e-4)
Ours	36.915 (±0.05)	0.970 (±5.7e-4)	0.024 (±1.1e-3)
Base 3DGS	36.133 (±0.06)	0.965 (±1.5e-4)	0.033 (±1.2e-3)

Table 1. **Performance on Replica Dataset.** (average performance for 5K training iterations, speed-up module rendered feature $dim = 128$). Boldface font represents the preferred results.

Metrics	mIoU↑	accuracy↑	FPS↑
Ours (w/ speed-up)	0.782	**0.943**	**14.55**
Ours	**0.787**	**0.943**	6.84
NeRF-DFF	0.636	0.864	5.38

Table 2. **Performance of semantic segmentation on Replica dataset compared to NeRF-DFF.** (speed-up module rendered feature $dim = 128$). Boldface font represents the preferred results.