Efficient Storage of Image Sets with Deep Learning

Kartik Sinha
Georgia Institute of Technology
ksinha45@gatech.edu

Katherine Stevo
Georgia Institute of Technology
kstevo@gatech.edu

Vivek Vijaykumar
Georgia Institute of Technology
vivekvjk@gatech.edu

Abstract

Inter-image compression has become increasingly relevant in the age of cloud storage. With an ever-growing collection of images in the cloud, it becomes all the more important to efficiently compress images, or possibly compress a set of images that may share some redundancy. Existing works have tried to target this problem by leveraging low level frequencies, SIFT features, etc. extracted from these images. However, there has been a lack of advancement in this area with modern deep learning techniques. With the ability of deep neural network models to learn strong latent representations of images, we plan on leveraging neural networks and modern deep learning techniques to act as feature extractors for finding representative images in a set to perform image set compression on. The motivation for this compression is to reduce the disk space utilization in storing these images. Taken in aggregate, we increase the efficiency of storing large collections of image data. ***Our method provides 7$\times$ reduction in memory footprint, beating baseline methods by*** 9.6% ***and current image-set compression approaches by*** 8.8%***.***

1. Introduction

The last decade has seen an increased use of images on the internet, and the storage of photos on cloud services. With the magnitude of data present online, there is a need to efficiently store them in an accessible manner. Hence, image compression has been a field that has received vital attention during this time span. However, there are instances when images contain lots of redundant information with others in a given set (e.g. photo album, images of similar landscapes or monuments, etc.). The idea of image set compression aims to tackle this problem by leveraging this repeated information across images in a set to efficiently encode and compress them together.

This paper addresses the following task objective: Given a collection of m images we wish to represent these images by information contained within a set of representative images m', where $|m'| < |m|$ from which the original images may be perceptually recovered via prediction schemes. Random access time should be reasonably low to reduce image loading times. This has wide applicability for personal cloud-based or on-device image storage, which empirically contain sets of images with high inter-image redundancy.

Current approaches to image-set compression utilize classical techniques such as SIFT features [13] and dense correspondences [21]. However, deep learning approaches [17] have shown impressive performance for intra-coding images. In the case of video compression, autoencoder methods such as [6] match or outperform standard codecs such as H.264 and H.265 (HEVC) [18].

Inspired by these works, we hope to apply the strong modeling power of deep neural networks to the image-set compression tasks. We design, implement, and analyze results for the following methods in this paper:

- Using CLIP [9] and DINOv2 [8] features in place of SIFT features to compute pair-wise similarity scores for images in the set.
- Evaluate compression of non-overlapping patches as image frames of the original image.
- Evaluate compression performance after using metrics such as LPIPS to sparsify image embeddings to create video compression opportunities.

2. Related Works

2.1. Existing Image Set Compression Methods

Existing image set compression techniques [2, 11, 12, 14, 15, 19] take advantage of a pipeline that first create a cost graph based on similarity between images in a set, computed via SIFT [5] features. From here, they aim to create a pseudo-video which is an ordered view of these images

that attempts to minimize the total cost of transitions between images. Hence, they construct a minimum-spanning tree (MST) from this cost graph and topologically sort the MST to create a directed acyclic graph to use as a pseudo-video. These works have shown that using this pseudo video with video compression techniques such as HEVC [18] have shown promise and have been effective at solving this problem.

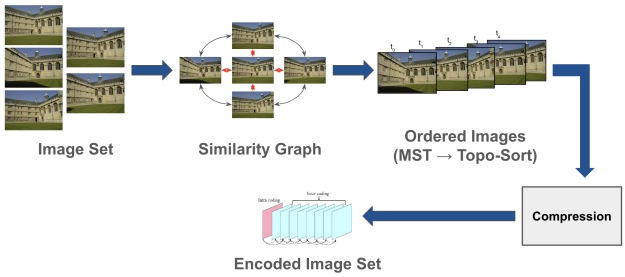

Figure 1. **Image-Set Compression Pipeline**. A similarity graph is constructed via a similarity metric yielding an MST that may be ordered as a pseudo-video subsequently compressed into a bitstream by video compression frameworks.

2.2. VCT: A Video Compression Transformer

Recent existing methods for video compression attempt to use a hybrid of hand-crafted and learned compression techniques, but the Video Compression Transformer (VCT) proposed in [7] takes the approach of trying to train a transformer to learn complete end-to-end video compression with no other augmentations. They found that removing the hand-crafted elements of video compression led to significantly improved results. The VCT architecture consists of an encoder-decoder model which takes in the encoded previous two frames and the current frame to predict the compressed representation of the current frame using a probability mass function so that one can encode more frequently occurring values with fewer bits.

2.3. DVC

Deep Video Compression [6] improves upon traditional encoding approaches such as H.264 and H.265 by replacing or augmenting canonical compression modules such as motion estimation and compensation, residual transform and quantization, and entropy coding with CNN-based auto-encoder deep neural networks.

3. Method

Following current approaches [22], we design an image-set compression pipeline with the following stages — (1) Process an input of an arbitrarily-sized image dataset into clusters of similar, highly redundant images. (2) For each such group of images, compute a pairwise similarity metric M_S. (3) With the n^2 scores generated, we compute the MST of the fully-connected graph K_n, wherein each image is represented by a vertex and the edges are weighted by $M_S(u, v)$, where $u, v \in G_V$. (4) We arbitrarily root the MST at a vertex v and designate this as our starting image to topologically sort this MST into an image sequence S. (5) S is then processed as a pseudo-video input to a video compression framework to obtain the final compressed bitstream B. In such a formalization, the coding efficiency is predominantly determined by the image coding order, i.e., order of frames in the pseudo-video.

3.1. Global Image Features

3.1.1 Pre-trained DNN Feature Extractors

In order to construct a cost graph, where edges indicate distance between images, we look into leveraging features extractors via off-the-shelf models to obtain image embeddings that we can compare. Specifically, we experiment with both DINOv2 [8] and CLIP [9], which have been shown to be strong feature image extractors. These models have been trained on a large corpus of images, thus making them well suited to serve as global descriptors and strong image feature extractors. For every pair of images in an image set, we compute the L2 distance between their embeddings and use this as the cost of the edge in the fully-connected graph, K_n.

3.2. Patching

As an orthogonal approach, taking inspiration from the ViT [3] and Masked Auto-Encoder [4], we design a patching-based pair-wise similarity and pseudo-video generation approach for subsequent video compression. We propose subdividing the input image into equally-sized non-overlapping smaller patches. We hypothesize that by increasing the space for total permutations we create more opportunities for optimization and redundancy across the generated patches. As a result of patching, we also hope to inject a degree of spatial invariance as local objects that are centered in the resulting patches may be matched/coded with other instances of that object (not necessarily always appearing in the same region) across an image set by being permuted into successive frames. However, we note that one potential drawback of this approach is reduced intra-coding efficiency due to the reduction in per-frame resolution.

3.3. Sparsifying Embedding Space with LPIPS

Leveraging encoders to obtain latent space embeddings for images, although effective, suffers from a fundamental drawback. These models are highly effective in capturing important semantic features in images. Thus when prompted with a set of highly similar images, e.g., multiple views of the same scene, the encoding process yields

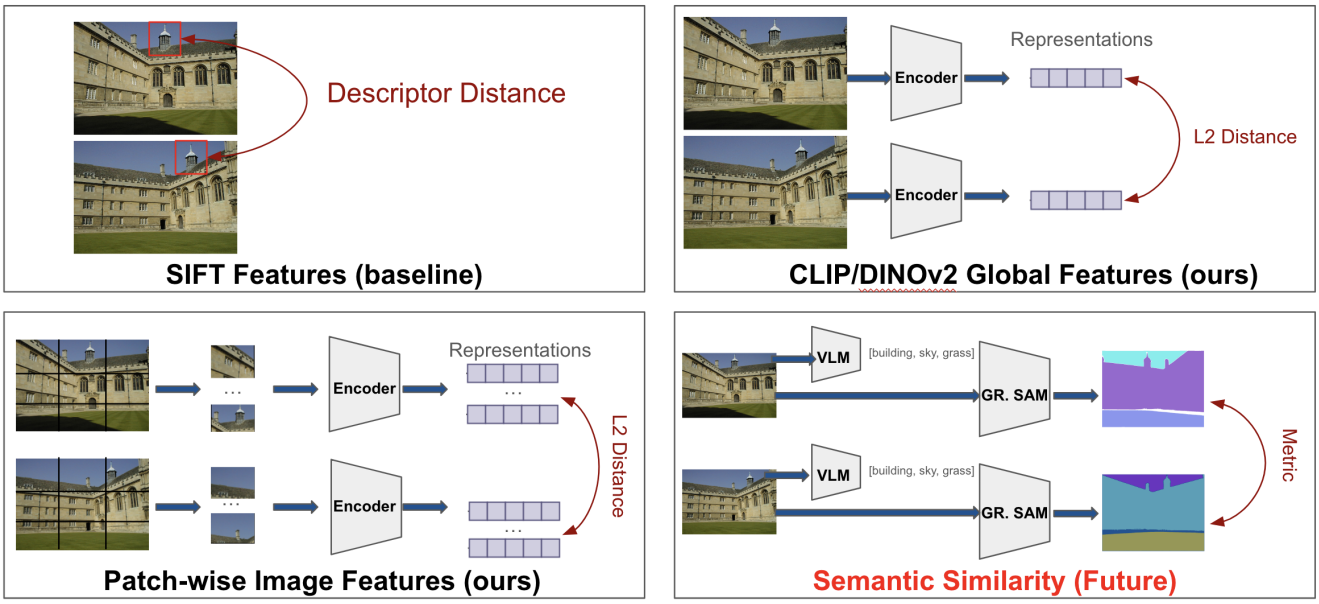

Figure 2. Summary of our designed approaches to compute pair-wise image similarity metrics. *(Top-left)* SIFT feature descriptor baseline. Euclidean distance between *(Top-right)* CLIP/DINOv2 image embeddings. *(Bottom-left)* Same as top-right but the input images are subdivided into non-overlapping patches. *(Bottom-right)* Future work delving into using similarity metrics grounded in semantic feature space over pixel space.

highly similar embeddings that are clustered together in feature space.

To address this limitation, we augment our latent embeddings by weighting (multiplying) it with the LPIPS (Learned Perceptual Image Patch Similarity) score [20] as a means to inject additional similarity information into our pairwise distances matrix prior to MST construction. LPIPS is a similarity metric that is calculated between pairs of images where a score of 0 denotes the same image and a score 1 denotes an entirely dissimilar image. Each image is split into patches and then the distance between the activations for each of the patches is used to form the LPIPS score for the image pair. We used the LPIPS metric that has AlexNet as a backbone. By multiplying our latent embeddings with the LPIPS score of the respective image pairs, this is effectively pushing dissimilar images farther apart and similar images closer together.

3.4. Video Compression

Video compression traditionally follows a predict-transform architecture [16] as detailed below. Input frames are divided into blocks (e.g. 8x8).

Motion estimation: Estimate motion between current frame and previous reconstructed frame to obtain motion vector.

Motion compensation: Predict frame by copying pixels from previous frame based on motion vector and calculate residual, the difference between ground truth and predicted frames.

Transform and quantization: Apply transform (e.g., DCT) to residual for better compression followed by quantization.

Inverse transform: Reconstruct residual from its quantized counterpart.

Entropy coding: Encode motion vectors and quantized result into bitstream via an entropy coding method (e.g. CABAC).

Frame reconstruction: Reconstruct frame by adding predicted frame and reconstructed residual. The decoder performs motion compensation, inverse quantization, and frame reconstruction based on encoded bitstream.

4. Experiments

We evaluate our approach on three datasets (1) Wadham College [1] (2) Tower Bridge subset of the Photo-Tourism dataset [2] and (3) "Hallway", our curated collection of images depicting various viewpoints of a hallway scene.

The Wadham College dataset was a dataset collected for multi-view construction consisting of 5 images with redundant features of a building from slighly different angles.

The Tower Bridge subset of the Photo-Tourism dataset is a set of 100 images of varying quality over a large timespan. This becomes a challenging benchmark, as there may

[1] Available at https://www.robots.ox.ac.uk/ vgg/data/mview/

[2] Available at https://phototour.cs.washington.edu/

Method	File Size on Disk (KB) ↓	PSNR ↑	SSIM ↑
CLIP/DINO (w/ LPIPS) + MST + HEVC (ours)	**113** (+9.6%)	34.932	0.916
CLIP+MST+HEVC (ours)	117 (+6.4%)	35.043	0.917
DINO+MST+HEVC (ours)	120 (+4.0%)	35.101	0.918
SIFT+MST+HEVC	124 (+0.8%)	35.173	0.920
Random+HEVC	125	**35.276**	**0.921**
Random+patches+HEVC	163	-	-
Original	794	-	-

Table 1. Compression performance of evaluated methods on the Wadham College Dataset. Relative percent improvement calculated over Random+HEVC baseline.

be less redundant information between images, and thus constructing a strong distance cost graph is vital.

The "Hallway" dataset is a dataset we curated with a set of 5 images of a hallway from slightly varying angles and positions.

4.1. Patch-wise Image Features

We experiment with the approach outlined in 3.2, dividing the input images into non-overlapping patches of sizes 64, 128, and 256 pixels. We limit our study to the Wadham College Dataset.

4.2. SIFT Baseline

A popular current method for image set compression is to use Scale Invariant Feature Transform as purposed in [14]. The code for this paper was not released so we had to reproduce their method locally ourselves. Following [14], we calculated the key point descriptor distances between all images pairs to determine an ordering of the images in each dataset that was then passed into our video generator for HEVC compression.

4.3. Random Ordering Baseline

The output of the SIFT, CLIP, and DINO methods is an ordering of the images which make up the video passed into HEVC. To ensure that our purposed methods were effective and better than using an arbitrary ordering of the images when creating the video for HEVC, we compressed each dataset using a random ordering of the images.

5. Results

The compression performance of the aforementioned proposed approaches on the Wadham College Dataset are detailed in 1. We compare against two main baselines (1) using SIFT Features to compute pair-wise similarity and (2) using a random permutation of image-set frames as a pseudo-video input to HEVC. See 3 and 4 for results on the Tower Bridge and Hallway datasets respectively.

Patch Size (px)	Size on Disk (KB)
64	156
128	139
256	139
No Patches (using best method)	113
Original	794

Table 2. Compression performance on Wadham College Dataset after subdividing input image into patches prior to MST construction and HEVC steps.

LPIPS. The latent embeddings that were weighted using LPIPS resulted in an improvement over the non-LPIPS weighted embeddings for temporally local images such as in the Wadham and Hallway datasets. LPIPS was able to pick up on the fine grained differences across temporally local images which nicely complemented the CLIP emeddings. Using LPIPS slightly worsened results for the Photo-Tourism dataset. Intuitively, we believe that LPIPS didn't improve compression of the Photo-Tourism dataset because the images in that dataset are taken across many different years, cameras, and environments (cloudy, sunny, rainy) which LPIPS was more sensitive to than other features like viewing angle. The addition of LPIPS caused the ordering to become one based on the quality of the image versus an ordering that was indicative of the point of view present in the image which is why we observed degraded performance for the Photo-Tourism dataset.

Patching. Compression performance on Wadham College using patched input images are shown in 2. We observe that patches yielded a sub-optimal result in comparison to our best method without patches (CLIP+LPIPS). Due to this, we do not evaluate the patch method on the Tower Bridge and Hallways datasets. We hypothesize that this may be caused by the reduction in the resolution of the image sequence processed by HEVC due to sub-division which consequently makes intra-coding of key-frames suffer.

Method	File Size on Disk (KB) ↓	PSNR ↑	SSIM ↑
CLIP/DINO (w/ LPIPS) + MST + HEVC (ours)	2123.163 (-0.50%)	**38.029**	**0.947**
CLIP+MST+HEVC (ours)	**2100.554** (+0.57%)	37.976	<u>0.946</u>
DINO+MST+HEVC (ours)	<u>2102.885</u> (+0.46%)	<u>38.006</u>	0.946
SIFT+MST+HEVC	2107.569 (+0.24%)	37.983	0.946
Random+HEVC	2112.652	-	-
Original	36500	-	-

Table 3. Compression performance of evaluated methods on the Tower Bridge Dataset. Relative percent improvement calculated over Random+HEVC baseline.

Method	File Size on Disk (KB) ↓	PSNR ↑	SSIM ↑
CLIP/DINO (w/ LPIPS) + MST + HEVC (ours)	**20.812** (+1.44%)	45.386	0.9831
CLIP+MST+HEVC (ours)	21.981 (– 4.09%)	**45.451**	**0.9834**
DINO+MST+HEVC (ours)	**20.812** (+1.44%)	45.386	0.9831
SIFT+MST+HEVC	22.228 (– 5.26%)	45.362	<u>0.9834</u>
Random+HEVC	21.116	<u>45.412</u>	0.9834
Original	92	-	-

Table 4. Compression performance of evaluated methods on our curated Hallway Dataset. Relative percent improvement calculated over Random+HEVC baseline.

6. Limitations

Our work has several limitations in the pipeline which we discovered throughout experimentation.

- The pseudo-video order given by a topological sort of the MST may not be optimally suited for this problem. Though prior works use this methodology, through observation, we noticed that given an MST, the order of images may ensure a smooth transition between image frames in the final ordering, even though the tree-like structure from the MST signifies this. Instead, in future work we look to explore leveraging traveling salesman approximation algorithms, with the relaxed constraint where we can revisit a node if necessary to find a smallest cost path.
- Our current methodology looks to optimize the ordering of the images in a set for the HEVC algorithm, however we have not yet explored whether these optimal orderings apply for other video compression algorithms like DVC. We also reported results on non-traditional image set compression datasets (except for Wadham College), however this was due to the lack of accessibility of these datasets used to benchmark previous approaches.
- Reproducibility of prior works was also an issue due to the lack of open-source works, hence our method to recreate prior works via SIFT features. The model weights for VCT were not released, making reproduction difficult. We reached out to the authors for weights to no response. For those reasons, we were not able to explore this further as it's currently infeasible to train the model ourselves due to limited access to GPUs.

7. Future Work

There are several extensions of our work that we'd like to explore. For example, in larger datasets, images may vary in quality and scenes drastically. Hence, we'd like to split up large image sets into smaller ones via unsupervised clustering, and then perform our pipeline on these smaller clusters. We believe this will lead to better pseudo-videos and hence better compression. We'd also like to add additional metrics such as PSNR vs. BPP and perceptual loss to align with metrics presented in some prior works.

Another idea which we want to explore is leveraging semantic similarity between images in a set to inject more information into the cost graph. The proposed pipeline is in Figure 2, labeled as "Semantic Similarity". The goal would be to leverage off the shelf segmentation models trained on an open-vocabulary to obtain segmentation masks for free. Specifically, in our proposed pipeline, we'd use a vision language model, such as GPT-4 [1] to list the classes of objects observed in an image. From here, we'd pass in this list of classes along with the image to a segmentation model such as Grounded SAM [10], which will output segmentation masks. Then we'd be able to construct similarity metrics leveraging the masks, such as those based on pixel-wise class statistics, or mIoU between segmentation masks for

pairs of images. We'd inject this information into an existing cost graph to hopefully add more information into the pipeline.

Finally, we plan to evaluate additional deep learning compression models such as DVC and substitute it into our pipeline to fully modernize the pipeline and leverage the recent advances of deep neural networks to realize the full potential of image-set compression.

8. Conclusion

In this paper, we propose a novel image set compression pipeline leveraging modern deep learning techniques to act as feature extractors to find representative images to perform compression on. By leveraging strong off-the-shelf backbones like CLIP and DINOv2 as feature extractors and injecting metrics such as LPIPS between image pairs in our distance cost graph, we were able to achieve a 7x reduction in memory footprint, beating baseline methods by 9.6% and current image-set compression approaches by 8.8%. We hope that future work looks to leverage the recent advancements in deep learning to further improve the image set compression pipeline.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 5

[2] Chia-Hsin Chan, Bo-Hsyuan Chen, and Wen-Jiin Tsai. Local feature-based photo album compression by eliminating redundancy of human partition. In *Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part I 13*, pages 143–158. Springer, 2017. 1

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 2

[4] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. 2

[5] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 1

[6] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework, 2019. 1, 2

[7] Fabian Mentzer, George Toderici, David Minnen, Sung-Jin Hwang, Sergi Caelles, Mario Lucic, and Eirikur Agustsson. Vct: A video compression transformer, 2022. 2

[8] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2

[10] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 5

[11] Lina Sha, Wei Wu, and Bingbing Li. Novel image set compression algorithm using rate-distortion optimized multiple reference image selection. *IEEE Access*, 6:66903–66913, 2018. 1

[12] Lina Sha, Wei Wu, and Bingbing Li. Low-complexity and high-coding-efficiency image deletion for compressed image sets in cloud servers. *IEEE Transactions on Cloud Computing*, 11(1):608–619, 2021. 1

[13] Zhongbo Shi and Xiaoyan Sun. Photo album compression for cloud storage using local features. *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, 4:17–28, 2014. 1

[14] Zhongbo Shi, Xiaoyan Sun, and Feng Wu. Multi-model prediction for image set compression. In *2013 Visual Communications and Image Processing (VCIP)*, pages 1–6. IEEE, 2013. 1, 4

[15] Zhongbo Shi, Xiaoyan Sun, and Feng Wu. Photo album compression for cloud storage using local features. *IEEE Journal on emerging and selected topics in circuits and systems*, 4(1): 17–28, 2014. 1

[16] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012. 3

[17] George Toderici, Sean M. O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks, 2016. 1

[18] Thomas Wiegand. Wd3: Working draft 3 of high-efficiency video coding. In *JCT-VC 5th Meeting, March 2011*, 2011. 1, 2

[19] Wei Wu and Lina Sha. Image subset union for compressed image sets in cloud servers. *IEEE Transactions on Cloud Computing*, 2022. 1

[20] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. 3

[21] Yabin Zhang, Weisi Lin, and Jianfei Cai. Dense correspondence based prediction for image set compression. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1240–1244, 2015. 1

[22] Ruobing Zou, Oscar C. Au, Guyue Zhou, Sijin Li, and Lin Sun. Image set modeling by exploiting temporal-spatial correlations and photo album compression. In *Proceedings of*

The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference, pages 1–4, 2012.