THE UNIVERSITY OF TEXAS AT AUSTIN

Vision Transformer-Assisted Analysis of Neural Image Compression and Generation

Master's Thesis Report

PRELIMINARY DRAFT v1.0

Official Code Repositories:
https://github.com/kliment-slice/thesis-code
https://github.com/kliment-slice/thesis-latex

Author:
Kliment Minchev

May 3, 2022

Contents

1 Introduction to Vision Transformers (ViT) 1

1.1 Motivation . . . 2
1.2 Brief History . . . 3
1.3 Principles of Operation . . . 4
1.4 Mathematical Formulation . . . 4
1.5 Implementations . . . 6
1.6 Computational Constraints . . . 6

2 Background Review: Transformers and Neural Image Compression 7

2.1 "An Image is Worth 16x16 Words" . . . 8
2.2 "End-to-End Image Compression with Transformers" . . . 9
2.3 Image Generation with GANs . . . 9
2.4 First Principles of Neural Image Compression . . . 9
2.5 Commonly Used Metrics for Image Quality . . . 9

3 ViT-based Assessment of Neural Image Compression 11

3.1 Generative Image Compression and Generation . . . 12
3.2 Output and Visual Inspection . . . 13
3.3 ViT-Scores . . . 14
3.4 Established IQA Metrics . . . 16
3.5 GAN-Related Quantitative Metrics . . . 19
3.6 Summary of Results . . . 19

4 Discussion 21

4.1 Results and Optimization . . . 22
4.2 Present and Future of Image Transformers . . . 23

4.3 Training and Cost Estimates 23

5 Summary **25**

5.1 Key Contributions 26

5.2 Takeaways 26

5.3 Acknowledgments 26

5.4 Closing Remarks 27

Bibliography **27**

Appendix **31**

List of Figures

1.1 Historical Usage of ViT in Image Tasks 2
1.2 Vector Representation 5
1.3 Key/Query Vector Proximity 6

3.1 Input Images 12
3.2 Latent Space Representation 13
3.3 GAN Training Process 13
3.4 Neural Compression and Generation 14
3.5 "Kliment" Established Metrics 16
3.6 "Logo" Established Metrics 17
3.7 "Bevo" Established Metrics 17
3.8 "Tower" Established Metrics 18

List of Tables

3.1 ViT-Scores of Generated Images 15
3.2 BRISQUE 18
3.3 Summary of Results 19

Executive Summary

This work investigates a novel application of a Vision Transformer (ViT) as a quality assessment reference metric for generated images after neural image compression. The Vision Transformer is a revolutionary implementation of the Transformer attention mechanism (typically used in language models) to object detection in digital images. The ViT architecture is designed to output a classification probability distribution against a set of training labels. Thus, it is a suitable candidate for a new method for quantitative assessment of generated images based on object-level deviations from the original pre-compression image. The metric will be referred to as a ViT-Score. This approach complements other comparative measurement techniques based on per-pixel discrepancies (Mean Squared Error, MSE) or structural comparison (Structural Similarity Index, SSIM). This study proposes an original end-to-end deep learning framework for neural image compression, latent vector representation, regeneration, and image quality analysis using state-of-the-art model architectures. Neural image compression and generation is achieved using a Generative Adversarial Network (GAN). Results from this work demonstrate that a ViT-Score from a Vision Transformer is capable of assessing the quality of a neurally compressed image. Moreover, this methodology provides valuable insights when measuring image quality. It can be used in addition to established perceived quality metrics for compressed and generated images such as SSIM and Frechet Inception Distance (FID).

Chapter 1

Introduction to Vision Transformers (ViT)

This chapter presents the reader with an introduction to Vision Transformers (ViT).

It covers the motivation as to why ViT, or a future development inspired by it, will have a profound impact on the future of image compression, analysis, and generation. This chapter presents evidence that a Transformer, or perhaps an evolved deep learning model with a similar architecture (i.e. generalizable and highly overparameterized) can be superior in compressing and evaluating the latent feature space of a digital image compared to present-day technologies.

This section then reviews the brief history of Transformer usage in deep learning. These generalized architectures are now dominating state-of-the-art language models, as they are extremely efficient in packing information within a one dimensional vector.

This introduction will then proceed to describe the principles of operation of a ViT. Then, proceed with a mathematical formulation. Finally, it will cover currently available implementations in the form of pre-trained models and conclude with an explanation on the computational and financial constraints of training such demanding architectures.

1.1 Motivation

Transformers are presently considered to hold a great promise for the future of Deep Learning as a step towards Artificial General Intelligence. Due to their architecture, they are more generalizable, less prone to overfitting, and able to learn highly complex representations. The Transformer architecture has already been proven to make obsolete Recurrent Neural Networks (RNNs) in natural language models. Furthermore, the Vision Transformer (ViT) has outperformed certain Convolutional Neural Networks (CNNs) in image classification tasks. [Dosovitskiy et al., 2021]

Figure 1.1 below shows an increase in the popularity of research related to Vision Transformers.

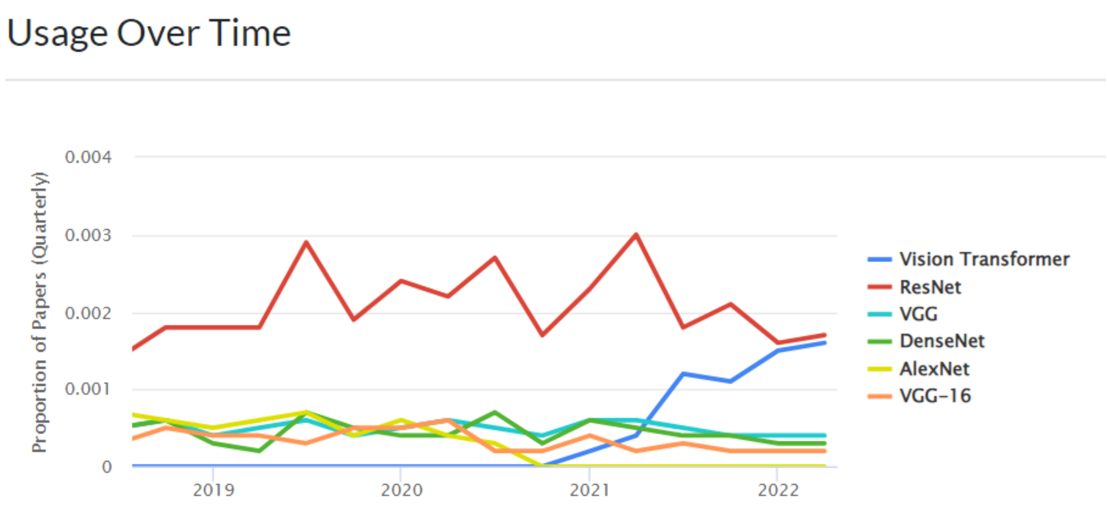


Figure 1.1: As of 2022, the usage of a Vision Transformer (ViT) in image tasks matches the usage of ResNets and has outnumbered any other popular CNN architecture. [PapersWithCode, 2022]

Figure 1.1 was produced by PapersWithCode, a popular academic research aggregator. For the past three years, ResNets, the most popular architecture in image processing and computer vision, has dominated the proportion of academic research in object detection. In 2022, Vision Transformer research popularity has reached that of ResNets and exceeded any other major category.

In the zeitgeist of Vision Transformer research, this thesis will explore a ViT-assisted metric related to image compression. This metric can provide additional insights to GAN output quality and the latent space (contextual) preservation of a variety of input images.

Thus, this work can be viewed as a stepping stone towards an end-to-end Transformer-based image compression and regeneration framework.

1.2 Brief History

1.2.1 Attention and Language Models

"Attention Is All You Need" is a seminal research publication by a team of Google researchers, which kickstarted the Transformer revolution in Deep Learning in 2017. It proposes a novel architecture, which models long-range dependencies in sequential (text) data, by arranging a set of self-attention layers.

A self-attention layer is what the model uses to focus on different elements of the input sequence simultaneously. For example, it can be ysed to compute the distance (relationship) between every word in a given sentence. [Vaswani et al., 2017]

Examples of implementations of text-based Transformers are BERT by Google and GPT-3 by OpenAI. BERT, among other things, as of 2021 processes and autofills every single English-based Google user search query. [Nayak, 2022] GPT-3, on the other hand, revolutionized text generation in 2020, demonstrating the ability to generate extremely cohesive textual output.

Most Transformers are used in language modeling and Natural Language Processing (NLP). Thus, they are often benchmarked against Recurrent Neural Networks (RNNs, and specifically Long Short-Term Memory, LSTM architecture). LSTMs rely on hidden states to pass information along sequentially during the encoding and decoding process for each word token. However, they typically fall short learning long-range dependencies.

1.2.2 Attention in Vision Tasks

The attention mechanism is capable of focusing on objects found anywhere on an input image. It operates within a single network layer compared to Convolutional Neural Networks (CNNs), where the variable size convolution kernels scan across the different layers of the architecture. [Dosovitskiy et al., 2021]

Tokenization happens at the pixel level, i.e. each pixel attends to each other pixel in the grid. This becomes computationally intensive, on the order of $(n^2)^2$, where n denotes width of a square image. To resolve this, the input image is broken down into square blocks of equal size, referred to as image patches. Then, each image patch is unrolled into a one-dimensional sequence $(nx1)$ and indexed with a positional embedding in a table for future reference and retrieval purposes. The embeddings enter the Transformer and finally, a feed forward classifier, in the form of a Multilayer Perceptron (MLP) makes the classification prediction, yielding a probability distribution.[Dosovitskiy et al., 2021]

Transformer, in a way, is a generalization of a feed forward network, but instead of fixed connections weights in an MLP, each connection weight (i.e. attention) is computed ad hoc. That makes the Transformer, unlike the MLP, permutation invariant. That is, it would not know where certain information is coming from, unless there are additional learnable sequential positional embeddings, i.e. index the image patches.

1.3 Principles of Operation

Continuing from the previous section, a good way to think of a ViT is as a generalization of an MLP, which itself is a generalization of a CNN. The ViT happens to learn very similarly to a CNN, which represents the latent space as filters carrying principal components.

In principle, CNNs have good inductive priors and can learn any function. However, they promote locality, i.e. nearing pixels are probability-wise considered most important. This may easily not be desired, especially in the key applications of object detection and, in the future, image compression.

The encoding process indexes embeddings. For instance, certain key words in a sentence or objects in image blocks are mapped in a reference lookup table. The Decoder outputs Keys at each step. These vectors represent hidden states, which are being passed on into each next iteration of the Transformer. The last layer, expectedly, uses a Softmax architecture to normalize and map the potential output classes to a probability distribution.

Multi-Head Attention

Sets of parallel attention layers at each token are called multi-head attention (to vary what to pay attention to: e.g. at verbs in the sentence or different objects in an image). The multi-head attention is composed of Key-Value pairs coming from the encoding part of the source sentence or image (i.e. the input embedding) and Queries from the output embedding (i.e. encoding part of target sentence or image). Now that we explained relevant transformer components, we can see how it applies to 2D signals, i.e. image matrices for classification purposes in image recognition.

1.4 Mathematical Formulation

So in its full formulation, Attention is a function of queries, keys and values vectors labeled capital (Q,K,V). It equals the dot product (QK^T) of keys and queries respectively, softmaxed over the square root of dimensions and multiplied by Values. So: Values - are what is most interesting in the source (sentence or image), e.g. attributes or features (like keyword adjectives or perhaps structural features in an image). Keys - index (or address) those values (name, type, weight). Each key has an associated value. Queries - are built by the encoder of the target sentece or image and prompt the network to find information.

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}}V)$$

The overall dynamic: a Query is pegged against a Key to locate a certain Value.

The softmax is basically a normalized exponential function: sequence of Variables is mapped into exponentials and divided by the sum of all the exponentials. Thus, the large numbers become almost ones and small numbers near zeros, like the maximum function, but this one is differentiable.

$$\sigma(Z_i) = \frac{e^{z_j}}{\sum_{j=1}^{K} e^{z_j}} \text{ for } i = 1, \ldots, K \; and z = (z_1, \ldots, z_K) \in \mathbb{R}^K$$

So a softmax of an inner product of each key with query vector normalizes to a probability distribution over all Values (very similar to using a softmax in the last layer of a NN over all the labels to yield the top classification pick).

To understand vector proximity between embeddings, e.g. similarity in objects for images, or words in sentences.

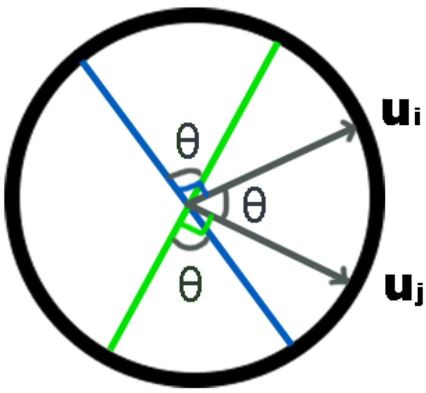


Figure 1.2: Vector representation on a unit circle.

The dot product of Keys and Queries yields an angle between both vectors (e.g. u_i and u_j in Figure 1.2 above) to measure how similarly aligned they are. In high dimension, most vectors would be orthonormal and $cos(90) = 0$. But if Key and Query align, they'd have a large dot product. Each key in space has an associated value (the pair). The Query vector is computed with each key and softmaxed to select one Key with the highest dot product. With softmax, a certain Key will stand out (in magnitude) vs the rest.

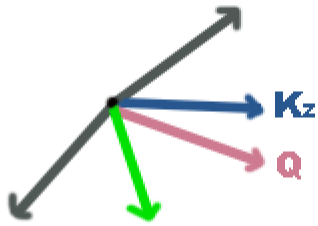


Figure 1.3: Vector proximity shows the closest Key vector to a given Query vector.

So the formulation of each proximity is the dot product of vectors K with Q: $sum(\langle K_z|Q\rangle)$

1.5 Implementations

1.5.1 Open Source Pre-trained Models

Open Source On GitHub. Original one by Google in tensorflow. Hugging Face in PyTorch. For the purpose of this thesis, we will use a PyTorch implementation trained on ImageNet-21k and fine tuned on ImageNet-1k.

1.5.2 Closed Source

OpenAI (DALL.E 2) released in April 2022, trained on 250M image-text pairs to be able to generate images from textual description. [OpenAI, 2022] Not much information, not open source like google's ViT or BERT.

1.6 Computational Constraints

Time, memory, cost

Text transformers perform extremely well on orders of magnitude smaller size training examples. GPT-3 trained on 45TB of text data (Wikipedia included), has 175B parameters and 96 attention layers. [Li, 2022] $4.6M using a Nvidia advanced datacenter GPU grade cloud cluster.

Several orders of magnitude more for image data.

Need a ViT on all internet, to cost $100M in training alone. To train a ViT on the whole TACC Frontera at 20k teraflops (top10) or Stampede at 10k tflops (top25), it would take respectively about a minute and 2 minutes. tpu v3 is 420 teraflops * 2500 = 1M Tflops

Chapter 2

Background Review: Transformers and Neural Image Compression

This chapter serves to present the audience with a literature review of seminal academic publications and relevant background information relating Transformers to image compression and generation.

Necessary formulations of Generative Adversarial Networks (GANs), Image Compression, and Image Quality Assessment (IQA) are also provided to aid the reader's understanding of this thesis project.

Among much of the domain knowledge available, the reader would find interest in the takeaways offered from:

- "An Image is Worth 16x16 Words" [Dosovitskiy et al., 2021]
- "Towards End-to-End Image Compression and Analysis with Transformers" [Bai et al., 2022]
- Image Generation with GANs
- "First Principles of Deep Learning and Compression" [Ehrlich, 2022]
- Review of commonly used Image Quality Assessment (IQA) metrics [Documentation, 2022]

2.1 "An Image is Worth 16x16 Words"

Transformers for Image Recognition at Scale

This seminal academic work was published for the International Conference on Learning Representations (ICLR) 2020 by a team from Google. It presents the first Vision Transformer (ViT) for object detection trained on the ImageNet dataset, a rather large natural images dataset common to research on image classification.

Since Chapter 1 already explained and formulated the ViT, this section will focus on outlining the outcomes from this publication.

"An Image is Worth 16x16 Words" completely discards the notion of convolutions. The team used an image patch-based approach as embeddings to a Transformer to classify ImageNet images, as opposed to using Convolutional Neural Networks (CNNs), which dominate this field of research. The ViT model achieves a top-1 accuracy of 77.3% on ImageNet, which compares to the accuracy of state-of-the-art CNNs.

Compared to its convolutional counterpart (the ResNet), the ViT cost 75% less resources to train and outperformed on ImageNet accuracy by 1%. ViT uses approximately 2-4x less compute to attain the same performance (averaged over 5 test datasets).

The total number of parameters is on the order of 100M.

2.2 "End-to-End Image Compression with Transformers"

Unique positional encoding can also be achieved using trigonometric representation. For example, a full sentence from text or perhaps a row of pixels from an image could be represented by the various periods of a sinusoid. Thus, the exact location of each token would be unique.

2.3 Image Generation with GANs

Generative models. GANs used to be the coolest thing until Transformers came along, but nonetheless still great. Typically, GANs pit 2 CNNs against each other, one called Generator to generate an image and one called Discriminator to determine how fake the image is compared to the input, relaying feedback to improve the Generator after each epoch. A GAN pits two Neural Networks (e.g. Convolutional) one against the other to generate new content (e.g. an image). The structure of a GAN is foundational to its design. As Fig.1 shows, the Generator CNN is shown on the left of the generated output. The Generator iteratively learns to create new synthetic data resembling real source data (e.g. regular or 360° images). On the right of the generated output is the Discriminator CNN. The Discriminator is trained to differentiate between real and synthetic data. In essence, the Discriminator gives the Generator feedback on its performance while simultaneously achieving incremental improvement in discernibility. Thus, each epoch the Generator produces more and more realistic data and the Discriminator iteratively improves its own ability to differentiate between real and synthetic data.

2.4 First Principles of Neural Image Compression

Deterministic, Probabilistic GANs JPEG/MPEG

2.5 Commonly Used Metrics for Image Quality

Metrics for the generated output image are: a BRISQUE score (reference-less) [5], Mean Squared Error (MSE) and Structural Similarity Index (SSIM). PSNR. GANs use FID and Inception score. BRISQUE [Mittal et al., 2012] Divisive normalization, Gaussian properties from natural scene statistics Process results in Gaussian unless it's distorted BRISQUE too simple for user-generated content (UGC)

Chapter 3

ViT-based Assessment of Neural Image Compression

Now that we explained relevant transformer components, we can see how it applies to 2D signals, i.e. image matrices for classification purposes in image recognition. The Vision Transformer or ViT, was very recently published for ICLR 2021 by a google team for a ImageNet trained transformer. Tokenization happens at pixel level, so each pixel would have to attend to each other pixel in the grid, which becomes too heavy to compute, on the order of. To resolve that, the image is broken down into blocks of equal size, a 16x16 subset of the image called image patches. Then, unroll each image patch into a sequence (256x1), and index it with a positional embedding in a table. All of that is then fed into a standard Transformer, like from Attention is all you need. Finally, a feed forward classifier (MLP) makes the classification prediction, voila image recognition. The total number of parameters is on the order of 100M.

3.1 Generative Image Compression and Generation

(vineeth)

3.1.1 Architecture

SGD optimizer for GAN

3.1.2 Sample Images Used

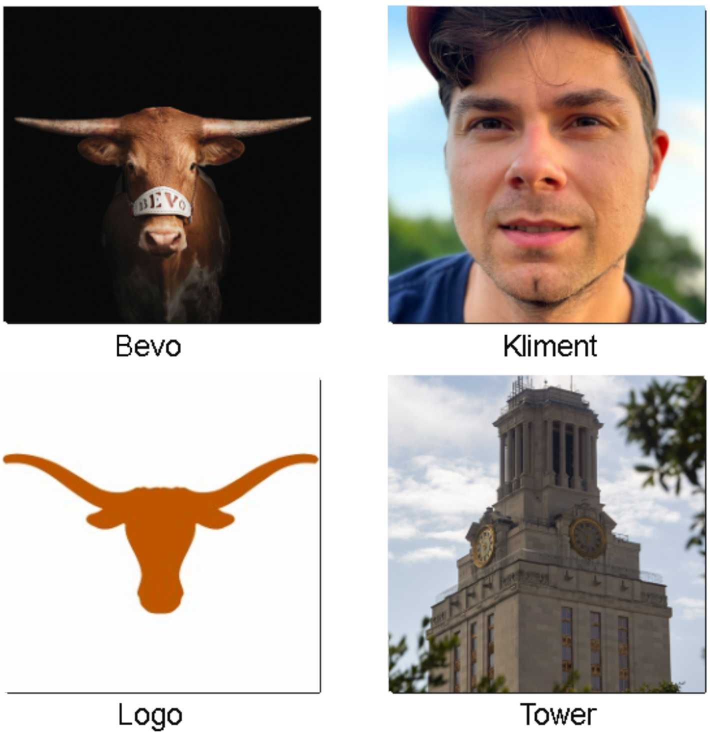


Figure 3.1: Input Images used in this project (512x512):
"Bevo": The University of Texas mascot, a famous longhorn bull.
"Kliment": A face portrait of the author.
"Logo": The Texas Longhorns logo.
"Tower": The University of Texas Tower, the Main Building on campus.

3.1.3 Latent space vector representation during compression

$nx1$ vector, where n corresponds to the height or width (in pixels) of a square input image. In the case of all input images used, the latent vector is of size $512x1$, since the input images are of size $512x512$.

hard to decipher but here is a zoomed in visual of the first This is what the GAN architecture compresses the full image to (1.54kb vs original size 409kb). The GAN then rebuils to 242kb.

Figure 3.2: Visual representation of the first 10 pixels in the most compressed version of the original image.

3.2 Output and Visual Inspection

The generative process from the GAN was designed to output an image at a specified epoch.

3.2.1 Training Process

Below is a demonstration of the GAN learning process at every 250 epochs:

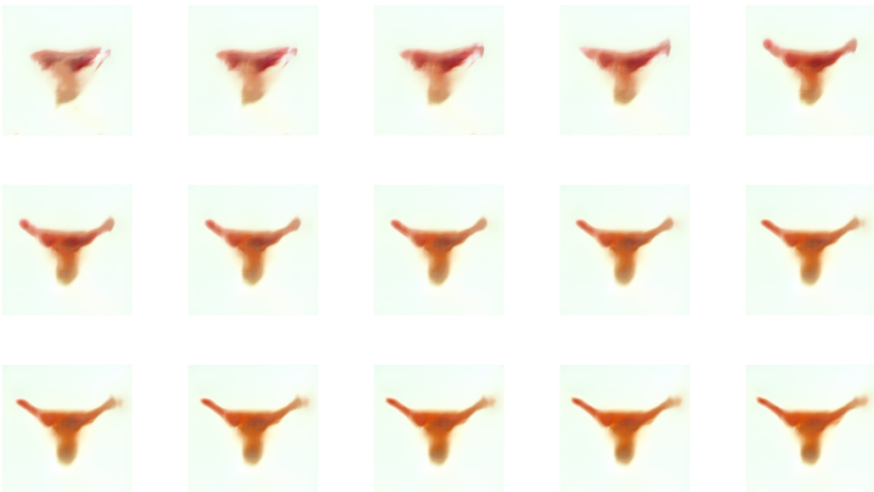


Figure 3.3: The GAN learns to compress and generate the Texas Longhorns logo.

Due to the cutting edge nature of the technologies used, a natural performance asymptote was observed. The GAN was able to reconstruct certain input images better than others. After a certain iteration, as usual, the GAN was unable to further learn how to compress, represent, and regenerate some input images. Typically, once a GAN reaches this stage, it learns from random noise and generation performance decreases.

3.2.2 Generated Results

Figure 3.4 below shows the resulting images from the neural compression GAN.

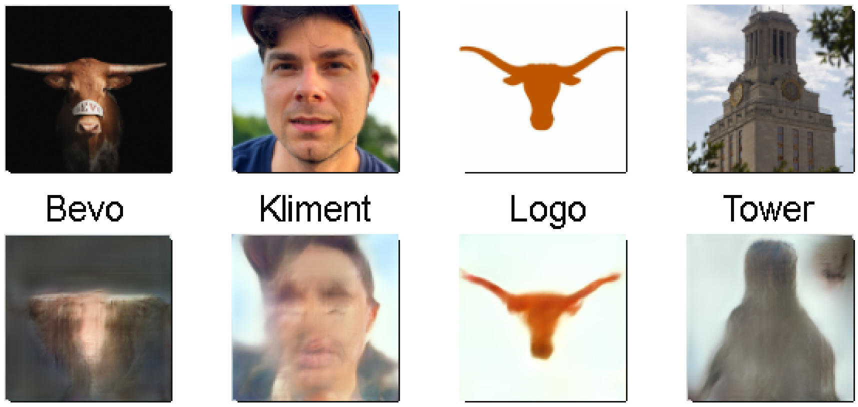


Figure 3.4: After sufficient training, the GAN outputs a regenerated version of the original image from a latent space vector representation.

The GAN was able to respectably regenerate "Kliment" and "Logo", especially if the resolution were to be lowered (e.g. 32x32, such as the CIFAR-10 dataset).

However, the GAN was unable to perform as well for "Bevo" and "Tower". It learned random noise and generation performance decreased.

3.3 ViT-Scores

The Vision Transformer-Assisted ViT Score is an original development from this thesis.

It is an attempt to measure the quality of a generated image after neural compression.

The ViT-score is in the open interval $(0,1)$ with 0 being poor and extremely dissimilar from the original and 1 being excellent and fully similar to original.

Mathematically, the endpoint values of the interval are unattainable by probabilistic models such as the GAN.

3.3.1 Mathematical Formulation

The following is a mathematical representation explained in further detail.

$$ViT_{score} = \frac{argmax_{A' \subset A, |A'|=k} \sum_{a \in A'} a}{k}$$

where $\sum_{a \in A'} a = \{m \in I_{input}\} \cap \{n \in I_{generated}\}$

and m are the top-K labels in the input image I_{input}

and n are the top-K labels in the generated image $I_{generated}$.

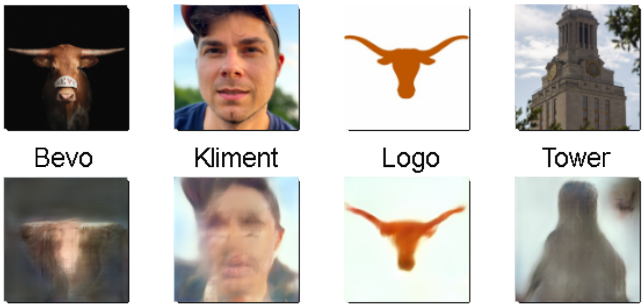
This overly elaborate mathematical notation is an attempt at describing:

"From the full set of trained ViT labels, we find the top-K number of intersecting labels between the original and generated images. Then, we divide that by K"

For example, of the top-100 labels found in the original image, identify the set of labels also found in the generated image. Then, divide that number of intersecting labels by the total number of 100 labels.

3.3.2 ViT-Scores from Resulting Images

Following Figure 3.7, the ViT-scores for the GAN generated images after neural compression are as follows:



Bevo	0.14
Kliment	0.54
Logo	0.29
Tower	0.03

Table 3.1: ViT-Scores demonstrate a somewhat expected quality assessment.

"Kliment" leads with a ViT-score of 0.54, which is understandable as the GAN generated a face (although smudgy) and was rather able to recreate the scenery structurally.

"Logo" generation seems structurally excellent and the ViT-score is 0.29, which is considered a good score for this particular GAN architecture and training.

"Bevo" barely preserves the original shape at ViT-score of 0.14, while **"Tower"** is incomprehensible and barely resembles the original at ViT-score of 0.03.

Overall, the ViT-score does a good job of measuring image quality.

3.4 Established IQA Metrics

3.4.1 "Kliment"

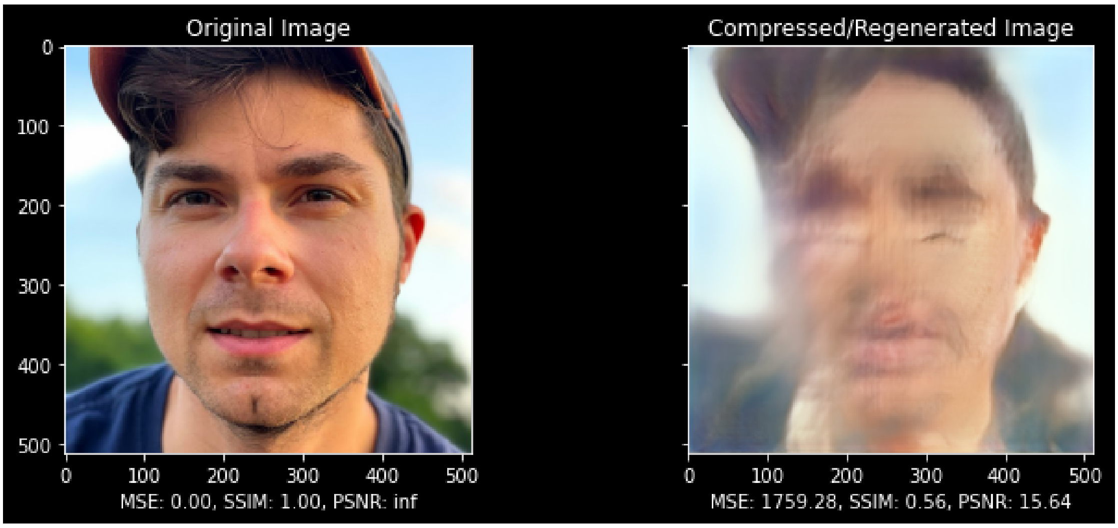


Figure 3.5: After sufficient training, the GAN outputs a regenerated version of the original image from a latent space vector representation.

3.4.2 "Logo"

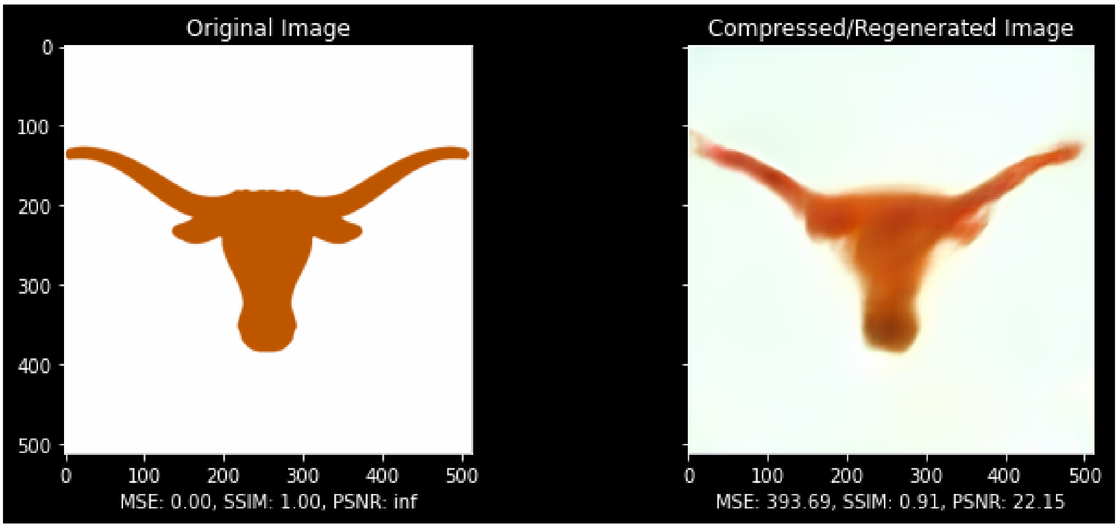


Figure 3.6: After sufficient training, the GAN outputs a regenerated version of the original image from a latent space vector representation.

3.4.3 "Bevo"

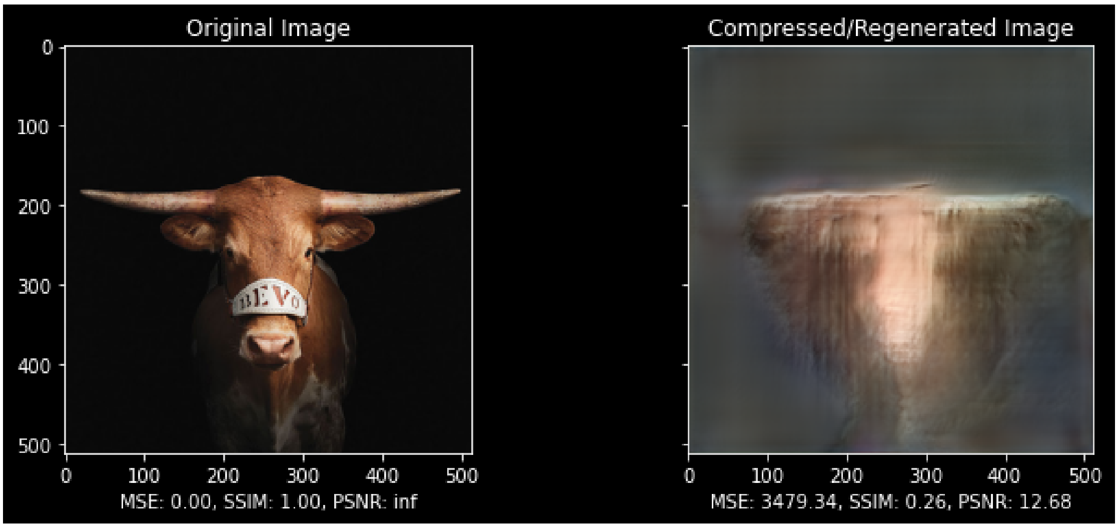


Figure 3.7: After sufficient training, the GAN outputs a regenerated version of the original image from a latent space vector representation.

3.4.4 "Tower"

Figure 3.8: After sufficient training, the GAN outputs a regenerated version of the original image from a latent space vector representation.

3.4.5 "BRISQUE"

Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) where approaching 0 is a good score and approaching 100 is a bad score, the BRISQUE referenceless image quality methodology This score could be interpreted as the image being more photorealistic than not. In terms of quality, this compares to a camera captured image with quality corruption caused by blurs or graininess. An image with no distortions often has a score below 5.

	Original	Generated
Bevo	32.9214	39.5535
Kliment	-8.3593	44.3570
Logo	102.9010	97.1844
Tower	14.5973	52.8363

Table 3.2: BRISQUE Scores of original and generated images.

Expectedly, the BRISQUE values for the generated images are always higher than their original counterparts. "Logo" is not a photorealistic image to begin with, so it is understandable that the BRISQUE value is high at 102.9. None of the generated images would pass BRISQUE as photorealistic and free of distortions.

Loss functions as well (MSE loss was used in GAN)

3.5 GAN-Related Quantitative Metrics

(FID score, inception score (IS)) Can be loss functions as well (MSE was used) FID is Frechet Inception Distance. 0 if there is no difference between the images. 81105.162 for logo and logo_GAN. 331171.556 for tower and towerGAN 549089.491 for kimbo and kimboGAN 1241999.901 for bevo and bevoGAN

3.6 Summary of Results

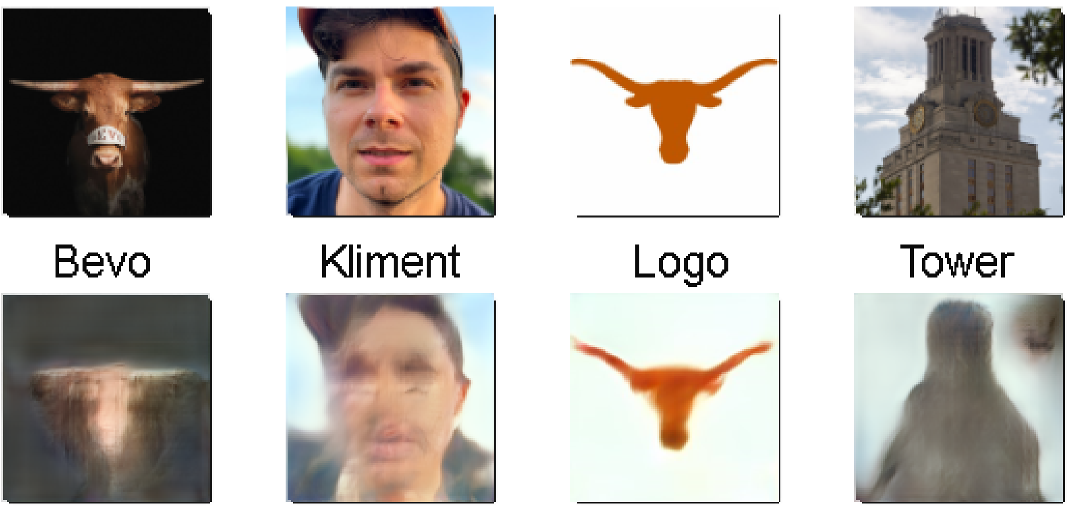


Image	ViT-Score	SSIM	MSE	PSNR	FID	BRISQUE
Bevo	0.14	0.26	3,479.34	12.68	1,241,999.901	39.5535
Kliment	0.54	0.56	1,759.28	15.64	549,089.491	44.3570
Logo	0.29	0.91	393.69	22.15	81,105.162	97.1844
Tower	0.03	0.53	1,315.75	16.90	331,171.556	52.8363

Table 3.3: ViT-Scores demonstrate a somewhat expected quality assessment.

Chapter 4

Discussion

Increase in relevant recent publications (13 alone in 2022 thus far) supports the vision of tying vision transformers to image compression.

4.1 Results and Optimization

We will review the merits from this work.

4.1.1 Results

Features are in deep layers of the GAN network. Latent space is hard to decipher ViT score definition: how many of the top100 labels match Can take into account probability of label (included in ViT). While label probability is stable when working with corrupted images (demo), unstable when working with generated images.

4.1.2 Potential Improvements to Architecture

Analyze latent space vector with transformer model (not a ViT, but a transformer adaptation) Steer the GAN faster into training to compress Slow computational times Need a GAN trained on all images, not just ImageNet or Celeb Need ViT trained on all images Natural limit to capacity of this model comes from training sets.

4.1.3 Optimization

Experiment and change loss functions (MSE was used, can use a GAN specific loss like FID)

Most valuable technique: (reducing learning_rate as the model trains) changing input images to cater to what the generative model is trained on. SGD optimizer for GAN

A lot of options still not figured out. regularization during training: residual dropout, label smoothing

4.2 Present and Future of Image Transformers

Status quo of Transformers in Image Processing, Compression, Analysis, and Generation Coveted Deep learning based Image compression In the deep learning/AI evolutionary process, still too early. Models have not been trained on enough image data.

GAN model only trained on finite set (ImageNet, CIFAR-10, Celeb-HQ faces etc) and resolution. Need to train GAN on all images ever.

"TransGAN: Two Pure Transformers Can Make One Strong GAN" [Jiang et al., 2021] NeurIPS 2021

Goal is to replace Generator and Discriminator in a GAN with Transformers free of convolutions.

4.3 Training and Cost Estimates

Need a ViT on all internet to cost 100M A 512-core TPU v3 pod costs $384/hr to use commercially on GCP. 2.5k core-days means training the ViT cost 24hrs * $384 * 5 of them = $46k. That's for one of many ViT flavors. To train a ViT on the whole TACC Frontera at 20k teraflops (top10) or Stampede at 10k tflops (top25), it would take respectively about a minute and 2 minutes. tpu v3 is 420 teraflops * 2500 = 1M Tflops

(like GPT-3 trained on all internet text, Vision T trained on all google images)

($100M+), GPT-3 cost $10M-$20M

Chapter 5

Summary

This chapter serves to conclude this thesis.

It provides a summary of original contributions made by the author while studying and experimenting with Vision Transformers (ViT) and Neural Image Compression, as well as the broader scientific domains of Machine and Deep Learning and Digital Image Processing.

A summary of key takeaways is provided for the audience.

The author concludes by acknowledging key contributors to the project and serves closing remarks.

5.1 Key Contributions

(experimentation with Vision Transformers, nascent field) Main merit of this thesis: ViT-score, a ViT-Assisted metric for evaluating the performance of a neural image compression Generative Adversarial Network.

This thesis explores a Vision Transformer (ViT)-Assisted metric related to image compression, which can provide additional insights to GAN output quality and the input latent space (contextual) preservation.

Furthermore, evaluating output quality from Generative Adversarial Networks (GANs) is still a developing field using non-Deep Learning-adapted assessment methods. For the purpose of this thesis, a GAN was used as a placeholder for a future, coveted, and highly desirable Deep Learning-based image compression mechanism.

Thus, this work can be viewed as a stepping stone towards an end-to-end Transformer-based image compression and regeneration.

5.2 Takeaways

A Vision Transformer (ViT)-Assisted metric related to image compression can provide additional insights to the latent space (contextual) preservation. Thus, this work can be viewed as a stepping stone towards an end-to-end Transformer-based image compression and regeneration.

It will take 100M dollars and a lot of work from a giant tech company, but future is near. The next major image compression methodology will be deep learning-based. The next image quality assessment will be deep learning-based. Transformers are ideal fit, highly generalizable, highly performant, hard to train.

5.3 Acknowledgments

The author would like to express gratitude towards several individuals and organizations from The University of Texas at Austin campus.

The major inspiration for this project was gathered from two courses taught by the reviewers of this thesis.

EE 371Q, Digital Image Processing taught by Professor Alan C. Bovik was the class where the author learned about Image Compression, Image Quality Assessment, and completed a term project on Generative Adversarial Networks.

CSE 382, Foundations of Machine Learning taught by Professor Rachel A. Ward was the class where the author learned key concepts used throughout this thesis and completed a term project on Vision Transformers (ViT).

Further acknowledgments are made to the Laboratory for Image and Video Engineering (LIVE) at the University of Texas at Austin for providing a source for project inspiration and insights.

Finally, the author would like to express gratitude to The Texas Advanced Computing Center (TACC). TACC provided free access to advanced High-Performance Computing (HPC) resources, which were used throughout the experimentation process in this thesis.

5.4 Closing Remarks

This thesis is written as a graduation requirement for the degree of Master of Science in Computational Science, Engineering, and Mathematics awarded by the Oden Institute at The University of Texas at Austin.

All code and knowledge is available as open source to the general public.

Bibliography

Y. Bai, X. Yang, X. Liu, J. Jiang, Y. Wang, X. Ji, and W. Gao. Towards end-to-end image compression and analysis with transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

S.-I. Documentation. Module: metrics - skimage. https://scikit-image.org/docs/stable/api/skimage.metrics.html, 2022. Accessed: 2022-04-30.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

M. Ehrlich. The first principles of deep learning and compression, 2022. URL https://arxiv.org/abs/2204.01782.

Y. Jiang, S. Chang, and Z. Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems*, 34, 2021.

C. Li. Openai's gpt-3 language model: A technical overview. https://lambdalabs.com/blog/demystifying-gpt-3/, 2022. Accessed: 2022-04-30.

A. Mittal, A. K. Moorthy, and A. C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012. doi: 10.1109/TIP.2012.2214050.

P. Nayak. Understanding searches better than ever before. https://blog.google/products/search/search-language-understanding-bert/, 2022. Accessed: 2022-04-30.

OpenAI. Dall.e 2. https://openai.com/dall-e-2/, 2022. Accessed: 2022-04-30.

PapersWithCode. Vision transformer explained. https://paperswithcode.com/method/vision-transformer, 2022. Accessed: 2022-04-30.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural*

Information Processing Systems, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

APPENDIX

5.4.1 Technologies used

Python, PyTorch MATLAB for BRISQUE LaTeX to generate this PDF

5.4.2 GPU, Local machine

NVIDIA GTX 1650Ti CUDA 11

Project dependencies (requirements.txt)

```
kiwisolver==1.3.1
matplotlib==3.2.0
matplotlib-inline==0.1.3
numpy==1.22.3
opencv-python==4.4.0.46
packaging==21.3
pandas==1.4.2
pickleshare==0.7.5
Pillow==8.0.1
pytorch-pretrained-vit==0.0.7
pywin32==303
pyzmq==22.3.0
regex==2020.11.13
scikit-image==0.18.1
scipy==1.5.4
torch==1.7.1+cu110
torchvision==0.8.2+cu110
```

5.4.3 TACC, Stampede2, job submission process

TACC Job submissions

```
#!/bin/bash

#SBATCH -J run_model         # Job name
#SBATCH -o logs/job.%j.out   # Name of stdout output file (%j expands to jobId)
#SBATCH -e logs/job.%j.err   # error file
#SBATCH -p gtx               # Queue name
#SBATCH -N 1                 # Total number of nodes requested (16 cores/node)
#SBATCH -n 1                 # Total number of tasks requested
#SBATCH -t 24:00:00          # Run time (hh:mm:ss) - 24 hours
#SBATCH -A Automatic-Assessment

module load python3
module load cuda/10.0
module load cudnn/7.6.2

cd /work/29369/kliment/
```

```
date

model_path="/model/model.1.pkl"

python3 main.py --data_path ./data/

date
```

TACC srun/idev

```
cd $WORK2
idev -m 30
module load python3

squeue
python3 transformer.py --data_path
```