THE UNIVERSITY OF TEXAS AT AUSTIN

Vision Transformer-Assisted Analysis of Neural Image Compression and Generation

Master's Thesis Report

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

Author:
Kliment Minchev

May 6, 2022

Contents

List of Figures

List of Tables

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 image quality based on object-level deviations from the original pre-compression image. The metric is 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, reconstruction, and image quality analysis using state-of-the-art model architectures. Neural image compression and reconstruction is achieved using a Generative Adversarial Network (GAN). Results from this work demonstrate that a ViT-Score is capable of assessing the quality of a neurally compressed image. Moreover, this methodology provides valuable insights when measuring GAN output quality and can be used in addition to other relevant perceived quality metrics such as SSIM or 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 summarizes the brief history of Transformer usage in deep learning. These generalized architectures are dominating state-of-the-art language models, as they are extremely efficient in packing relevant information within a one dimensional vector.

This introduction then proceeds to describe the principles of operation of a ViT, followed by its mathematical formulation. This section is followed by an overview of currently available implementations in the form of pre-trained models. The following section engages the reader in a thought-provoking review of the computational and financial constraints of training such demanding architectures. Finally, the chapter concludes with a speculation on the potential cost of training the next generation of successful generative image Transformers.

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.

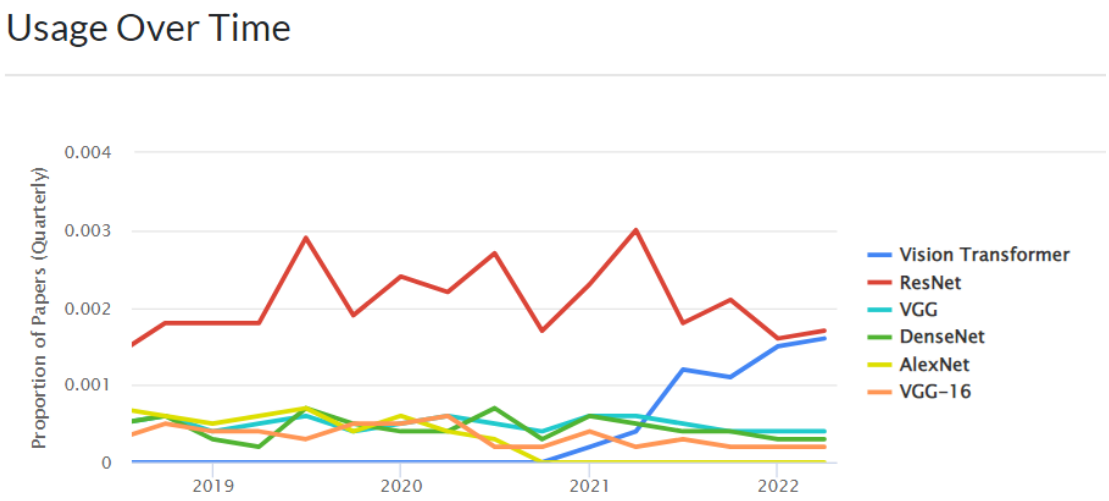


Figure 1.1: As of 2022, the usage of a Vision Transformer (ViT) in image tasks matches the usage of ResNets and has outnumbered any other popular CNN architecture. [PapersWithCode, 2022b]

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 explores 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, contributions from this thesis can be viewed as providing a stepping stone towards an end-to-end Transformer-based image compression and reconstruction 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 used 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 many other applications, processes and autofills every single English-based Google user search query as of 2021 . [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 for applications 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]

As shown in Figure 1.2 below, 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]

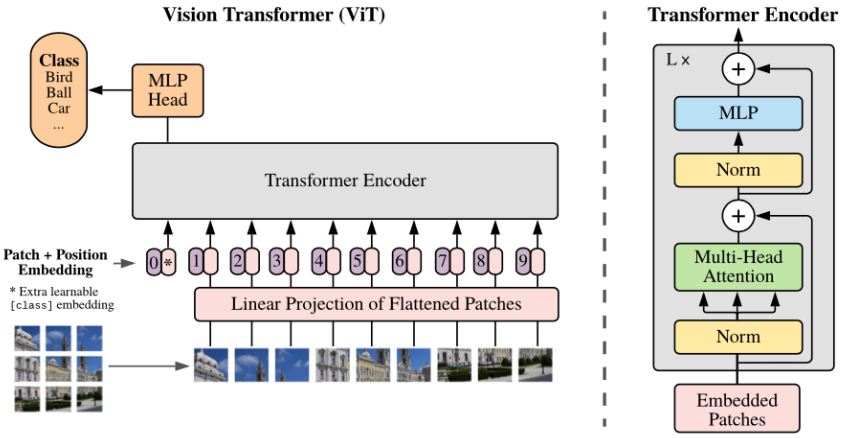


Figure 1.2: The Vision Transformer (ViT) architecture. [Google-Research, 2022]

A 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. This 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 ViT can be thought of 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

As shown in Figure 1.3 below, sets of parallel attention layers at each token are called multi-head attention. This approach varies what to pay attention to, for example,at the different objects in an image (or in natural langue, different verbs in a sentence). The multi-head attention is composed of Key-Value pairs coming from the encoding part of the source image (i.e. the input embedding) and Queries from the output embedding (i.e. encoding part of target image).

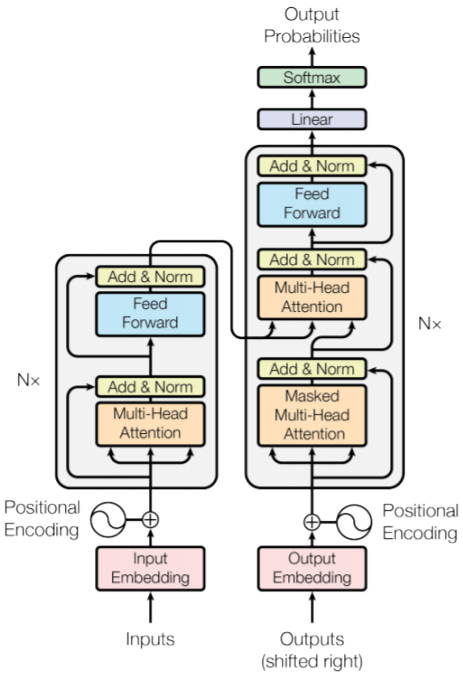


Figure 1.3: The Attention mechanism architecture. [Google-Research, 2022]

1.4 Mathematical Formulation

1.4.1 Attention

In its full formulation, Attention is a function of vectors representing Queries, Keys, and Values, labeled as (Q, K, V). Attention equals the dot product (QK^T) of Keys and Queries respectively, softmaxed over the square root of dimensions and multiplied by the Values vector.

To provide further intuition:

Values are what is most interesting in the source image, i.e. attributes or structural features. In text, a Value could be important adjectives before each keyword, which provide emphasis in a given input sentence. Keys, on the other hand, index (or address) those Values (e.g. name, type, weight). Each Key has an associated Value. Queries are built by the encoder of the target image

and prompt the network to find closest available information (Key and its corresponding Value).

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

Thus, the overall dynamic is that a Query is pegged against a Key to locate a certain Value.

1.4.2 Softmax

The Softmax function is defined as a normalized exponential function. A sequence of variables is mapped into exponentials and divided by the sum of all exponentials. Thus, the large numbers become almost ones and small numbers near zeros. Softmax is similar to the maximum function, with a key difference that Softmax is differentiable.

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

Thus, a Softmax of an inner product of each Key with Query vector normalizes to a probability distribution over all Values. Neural networks typically utilize a softmax in their last layer over all the classification labels. This yields the top classification by probability. Using the softmax function, a certain Key will stand out (in magnitude) vs the rest.

1.4.3 Vector Similarity

Vector proximity between embeddings represents the similarity between objects in images, or word connotations in sentences.

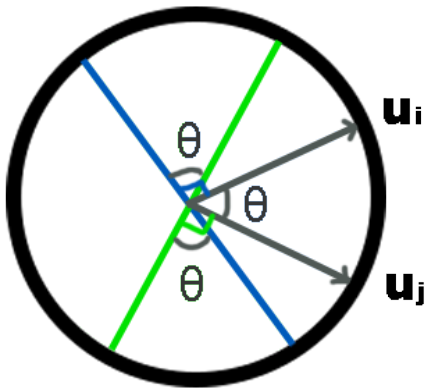


Figure 1.4: Vector representation on a unit circle.

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 dimensional spaces, most vectors would be orthonormal to each other and $cos(90) = 0$. But if Key and Query vectors

are similar or align, they'd have a large dot product. The larger the similarity, the larger the dot product. The Query vector is computed with each Key in the surrounding vector space and softmaxed to select the one Key with the highest dot product. The selected Key in space has an associated Value. In applications, that Value would correspond to a match with a labeled object in an image, or perhaps the next word in a generated sentence.

A depiction of the vector space is shown in Figure 1.3 below.

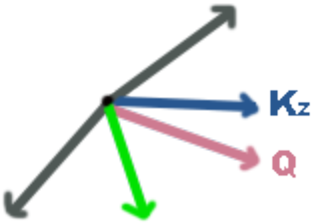


Figure 1.5: Vector proximity shows the closest Key vector to a given Query vector in space.

1.5 Implementations

This section reviews several notable ViT implementations of interest. Most developments have been made open source. Howerver, some of the highest quality implementations are still closed source and operated under a license or payment wall.

1.5.1 Open Source

The academic research aggregator PapersWithCode lists 96 open source implementations of the original ViT from "An Image is Worth 16x16 Words" discussed in Chapter 2. The publication was made for ICLR 2021. [PapersWithCode, 2022a]

The original model from the team of Google researchers, written using TensorFlow, has 72,567 stars in its GitHub repository. The second highest implementation by Hugging Face, written using PyTorch, has 61,820 stars.

Pre-trained Model Used

For the purpose of this thesis, a PyTorch implementation was used trained on ImageNet-21k and fine tuned on ImageNet-1k. The ViT has 9,696 stars on its GitHub repository, ranking as the 5th highest rated implementation. It was chosen due to its reliability, project maturity, and interfaceability. [PapersWithCode, 2022a]

A python pip package is also available on PyPI from the same "lucidrains" implementation, which

can be installed via:

```
"pip install pytorch_pretrained_vit"
```

Other pre-trained models are also availabe on GitHub and Google Colab. Some suffer from a lower GitHub star rating, are not well packaged (not object-oriented), or perhaps deviate too far from the original implementation of "An Image is Worth 16x16 Words".

1.5.2 Closed Source

Several modified implementations targeting specific generative applications have been made by Hugging Face and OpenAI. OpenAI has expanded from its success from GPT-3, a text generation Transformer, to visual applications. Transformers and Vision Transformers can be used to find context in image objects from the latent space of an input image. CLIP, Image GPT, DALL.E, and DALL.E2 by OpenAI bridge the gap between textual information and images. DALL.E 2, the highest quality image Transformer to date, was released in April 2022. It is trained on 250M image-text pairs to be able to generate coherent images from textual description. [OpenAI, 2022]

Most of OpenAI's Transformer developments (e.g. GPT-3, DALL.E 2) have not been open sourced. Google's ViT and BERT, however, have been.

1.6 Computational Constraints for Training

The main reason for using a pre-trained ViT is that the Transformer architecture is so generalizable and overparameterized, that the computational requirements for Time, Memory, and Cost of training are beyond the scope of this thesis.

For example, GPT-3 was trained on 45TB of text data (e.g. all of Wikipedia included). It has 175B parameters and 96 attention layers. [Li, 2022]

One flavor of GPT-3 (of many) alone cost $4.6M to train using the most advanced Nvidia datacenter GPUs in a cluster.

The original ViT implementation from "An Image is Worth 16x16 Words" took 2.5k core-days of training time. [Dosovitskiy et al., 2021]

Training the Next Generation of Image GPT

It would require several orders of magnitude more resources to train a Vision Transformer that matches the quality of GPT-3 for generative text applications.

The training data would need to be similar to all Google images on the internet (or at least a respectable representative subset). It would probably cost on the order of $50M in training alone, since image matrices are two-dimensional sequences of data. Contextual complexity also rises exponentially for images compared to text.

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]
- Overview of Image Generation with GANs
- "First Principles of Deep Learning and Compression" [Ehrlich, 2022]
- 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 in 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 in detail, this section focuses on outlining the outcomes from this publication. Refer to Figures 1.2 and 1.3 from Chapter 1, which were originally developed for this paper.

The Self-Attention mechanism, detailed in section 1.2.2, is key to defining where the Transformer is able to find a certain object on an image. Notice the self-attention map plotted on top of the original images in Figure 2.1 below:

Figure 2.1: The attention map is plotted and superimposed on the original images from the paper. [Dosovitskiy et al., 2021]

"An Image is Worth 16x16 Words" completely discards the notion of convolutions. The team used an image patch-based approach by breaking the image down to square blocks called patches. These patches are used as embeddings to a Transformer to classify images.

Comparison between the position embeddings using cosine similarity points the transformer into forming the object detection attention map. Figure 2.2 below shows a representation.

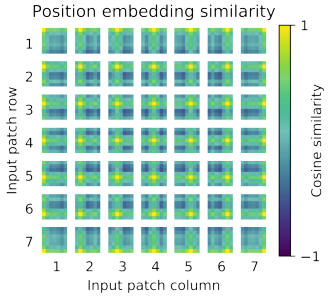


Figure 2.2: The image patches are compared to each other using cosine similarity to yield an indexed embedding map guiding the self-attention mechanism. [Dosovitskiy et al., 2021]

Typically, such object detection tasks in this field of research are dominated by Convolutional Neural Networks (CNNs).

However, the ViT model achieves a top-1 accuracy (matching highest class probability only) of 77.3% on ImageNet, which compares to the accuracy of state-of-the-art CNNs.

According to the paper, the pre-trained ViT on the JFT-300M dataset beats all ResNets on all datasets. Furthermore, it takes significantly less computational resources to train. [Dosovitskiy et al., 2021]

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.

Some of the results from the original publication are shown in Figure 2.3 below.

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21K (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Figure 2.3: Results summary from the original ViT by Google. It shows the ViT performance on popular classification benchmarks. [Dosovitskiy et al., 2021]

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

A recent publication in Association for the Advancement of Artificial Intelligence (AAAI) 2022 is claiming to have achieved an end-to-end Transformer-based model for image compression and analysis.

"Towards End-to-End Image Compression and Analysis with Transformers" redesigns the ViT to classify images from compressed features. [Bai et al., 2022]

The research team replaces the patches and embeddings with a CNN-based lightweight encoder. The compressed latent space features from the encoder are fed into the Transformer. The Transformer proceeds to classify the image without any regeneration or reconstruction. The framework includes a feature aggregation module, which blends the compressed and intermediate Transformer features. These features are then passed onto a Deconvolutional Neural Network and the image is regenerated. As a result, long-term dependencies from the Transformer self-attention mechanism are preserved.

Figure 2.4 below demonstrates the exact architecture used in the experiment.

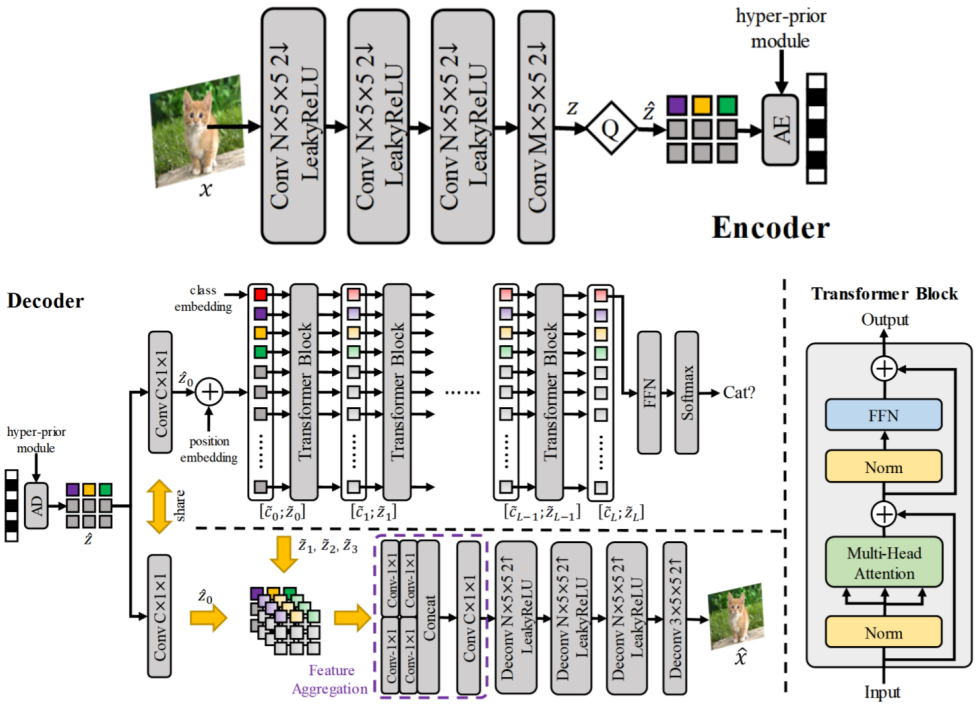


Figure 2.4: The encoder and decoder architecture of the End-to-End Image Compression ViT. [Bai et al., 2022]

2.3 Overview of Image Generation with GANs

Generative Adversarial Networks (GANs) have long been considered to be the most exciting innovation in Deep Learning. In a typical GAN architecture, two CNNs are competing against each other. One network is called the Generator, whose purpose is to generate an image. The other network is called the Discriminator, whose purpose is to determine how realistic the generated output by the Generator is.

The Discriminator communicates feedback to the Generator after each epoch through a loss function. The Discriminator is trained to minimize the loss, while the Generator is maximizing it. The Generator iteratively learns to create new synthetic images resembling real source input image. Then, the Discriminator is trained to differentiate between the real input data and generated synthetic data. The desired outcome is that a Generator would be able to fool the Discriminator that its synthetic output is real. [Pathmind, 2022]

In essence, the Discriminator provides feedback to the Generator on its performance, while simultaneously achieving incremental improvements in its own discernibility.

As a result, after each epoch, the Generator is expected to produce increasingly more realistic data, according to a quantitative or qualitative (visual inspection) metric. In turn, the Discriminator becomes better at catching synthetic output, thereby having both networks compete against each other to refine the final generated image output.

Figure 2.4 below illustrates the architecture of a common GAN. Starting from random noise, the Generator is iteratively trained to produce a quality generated image. That image is then evaluated by the Discriminator whether it is real or synthetic.

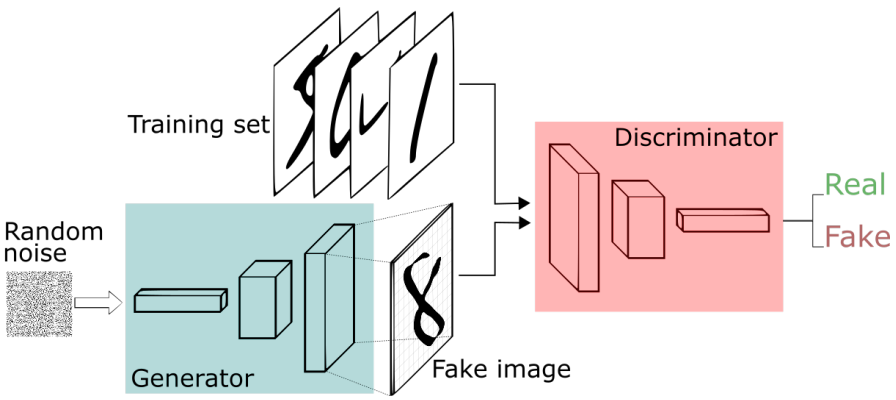


Figure 2.5: The architecture of a simple Generative Adversarial Network. [Pathmind, 2022]

Several common GAN applications include generating missing data (versus deterministic interpolation), synthetic data, image-to-image translation, medical imaging, and art.

2.4 First Principles of Neural Image Compression

The explosion in popularity of Deep Learning methods as universal function approximators is expected to revolutionize image compression. The key reason is that neural networks could achieve orders of magnitude higher compression than lossy deterministic methodologies. They can extract relevant information from the latent space of the image and are extremely efficient at packing it into a condensed version. The resulting compressed representations would require much less disk space or bandwidth for transmission.

Presently, efficient storage and transmission of image and video is still a developing and costly domain. Thus, a Deep Learning approach to multimedia compression would, theoretically, enable higher compression ratios and an increase in visual quality. This approach would train a model to learn a compression function directly from data. Referred to as "Learned Multimedia Compression", this approach involves computing a compressed representation of an input image. It uses separate models for both the encoder and decoder. [Ehrlich, 2022]

Finally, a Deep Learning-based method could learn key steps of established deterministic compression algorithms. This approach would serve as an improvement to the existing framework of popular compression techniques. The most popular image compression technique is the Joint Photographic Expert Group format, JPEG, created in 1992. In short, it operates by using a Discrete Cosine Transform (DCT) to convert image information from the spatial to frequency domain. Then, the compression algorithm discards high-frequency information, as it is not relevant to human visual perception. [Ehrlich, 2022]

Figure 2.6 illustrates JPEG compression at various thresholds, along with the compression ratios of each compressed image listed underneath.

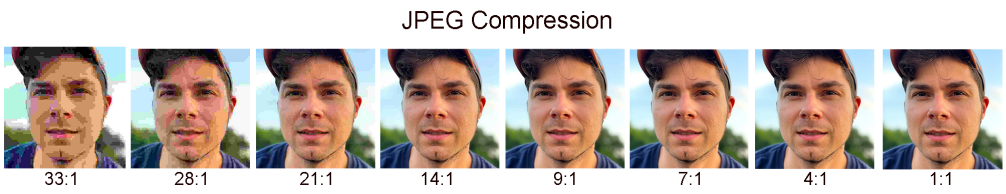


Figure 2.6: A neural compression model would have to outperform JPEG in compression ratio vs reconstructed image quality. [Pathmind, 2022]

An example of a Deep Learning-based compression model could be to improve the compression fidelity (output quality, compression ratio, and speed) of JPEG. To prevent information loss, a model can be trained to predict non-linearities in the compressed image related to blurring or noise. Furthermore, a model could learn image correction maps to correct distortions caused by JPEG compression. It can be trained to learn the latent feature space from example pairs of corrected JPEG compressed images and their ground-truth originals.

2.5 Metrics for Image Quality

Image Quality Assessment (IQA) is a methodology to evaluate distortions, aberrations, or degradations in perceived image quality. To evaluate the quality of a reconstructed image and benchmark the ViT-Score, this thesis will use the following metrics:

- Structural Similarity Index (SSIM)
- Mean Squared Error (MSE)
- Peak Signal-to-Noise Ratio (PSNR)
- Frechet Inception Distance (FID)
- Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE)

SSIM

SSIM is a measure of the similarity between two images, on an interval $[-1,1]$.

The higher SSIM value indicates a more similar image to the reference.

It is based on the computing and combining three components: Luminance, Contrast, and Structure. Luminance measures image brightness. Contrast measures how dark and bright pixels are distributed in an image. Structure captures edge definition. [Wang et al., 2004]

The exact mathematical formulation is:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + (k_1L)^2) + (2\sigma_{xy} + (k_2L)^2)}{(\mu_x^2 + \mu_y^2 + (k_1L)^2)(\sigma_x^2 + \sigma_y^2 + (k_2L)^2)}$$

Where, given a windows x and y of size (N,N):

μ_x is the average of x' μ_y is the average of y; σ_x^2 the variance of x, σ_y^2 the variance of y;

σ_{xy} the covariance of x and y; L is the pixel value dynamic range, i.e. $2^{\#bits\_per\_pixel} - 1$;

and by default, given as constants are $k_1 = 0.01$ and $k_2 = 0.03$ [Wang et al., 2004]

MSE

MSE is a measure of the average squared error between an input and reference matrix (i.e. image). The lower the MSE value, the better the image quality.

The mathematical formulation is:

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij} - y_{ij})^2$$

Where: m,n is the rows and columns of the input image, x_{ij} and y_{ij} are pixel values from the input and generated image respectively. [Documentation, 2022]

PSNR

PSNR is the ratio between the maximum possible signal intensity value and the corrupting noise present in the same signal. The higher the PSNR value, the better the image quality. [Documentation, 2022]

The mathematical formulation becomes:

$$PSNR(x,y) = \frac{10\log_{10}[\max(\max(x),\max(y))]^2}{|x-y^2|}$$

FID

Commonly used in analyzing GAN output quality, the Frechet Inception Distance (FID) compares two multidimensional Gaussian distributions.

Given are:

$\mathcal{N}(\mu,\Sigma)$, representing neural network features of the GAN generated image.

and

$\mathcal{N}(\mu_w,\Sigma_w)$, representing the same features from the real images in a training dataset. [Brownlee, 2022]

Thus, the mathematical formulation is:

$$FID = ||\mu-\mu_w||_2^2 + tr(\Sigma+\Sigma_w-2(\Sigma^{1/2}\Sigma_w\Sigma^{1/2})^{1/2})$$

BRISQUE

BRISQUE represents a referenceless image quality methodology where 0 is the perfect score and 100 worst. This score could be interpreted as the tendency of an image to be photorealistic. The score is derived from the Gaussian properties of natural scene statistics found in images. Image quality is evaluated for the presence of corruption caused by blurs or graininess. An image with no distortions often has a score below 5. [Mittal et al., 2012]

BRISQUE may be too simple to evaluate user-generated content, but is nonetheless useful in comparing it to the ViT-Score.

Chapter 3

ViT-based Assessment of Neural Image Compression

This chapter presents the core contributions from this thesis. Four input images are used to demonstrate the modest ability of a Generative Adversarial Network (GAN) to compress and reconstruct an image from its latent space representation. The section explains the architecture of choice and presents the output from the GAN. The audience is presented with visualizations of the training process, compressed latent vector, and the resulting Compression Ratio (CR) values achieved. The reader is encouraged to visually inspect the output. The novel ViT-Score and its mathematical formulation are presented. Along with the ViT-Score, the reconstructed output images are also evaluated using the following image quality metrics: SSIM, MSE, PSNR, FID, BRISQUE (refer to Section 2.5).

Finally, a summary table presents all results from this chapter for quick reference.

3.1 Architecture

3.1.1 Main Components

The main architectural components of this project are a Generative Adversarial Network (GAN) and a Vision Transformer (ViT). An input image is compressed into a latent vector form, which is then used to generate a synthetic version of the original image. The GAN iteratively improves its ability to generate a realistic image, which resembles the input (refer to Section 2.3 for an overview of GANs). A pre-trained "PGAN" was used, further described in Section 3.1.3.

A pre-trained Vision Transformer (ViT) was used for object detection in both the input and reconstructed images. A custom method then identifies the number of matching labels between the images. Finally, a ViT-Score is computed.

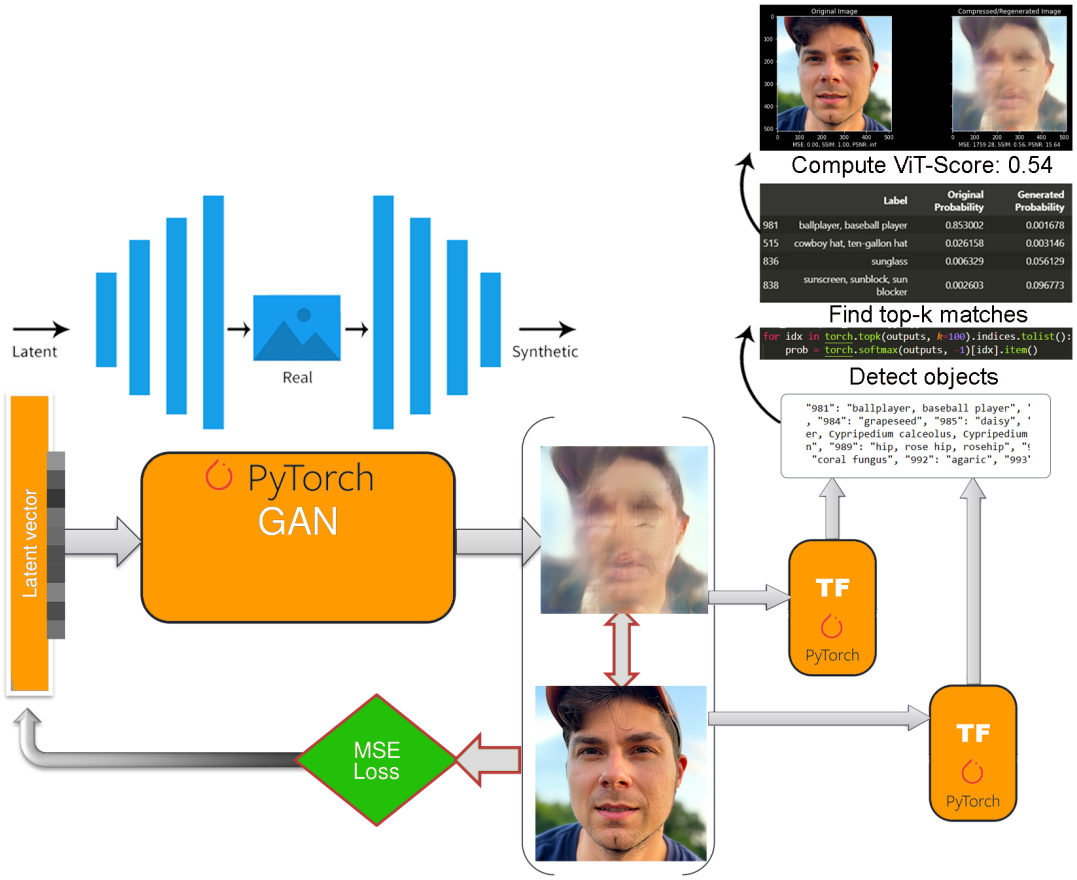


Figure 3.1: The full ViT-Scoring engine. A GAN compresses, then reconstructs an input image from its latent vector. Then, a Vision Transfomer (ViT) detects objects in the input and generated images. Finally, the ViT-Score is computed from the number of matching labels.

3.1.2 Image Compression Diagram

Stochastic Gradient Descent (SGD) optimizer was used for the GAN to find the most optimal latent vector representation of the input image. Mean Squared Error (MSE) was used as a Loss function to improve the GAN performance.

Similar to established image compression methodologies, the compression and reconstruction process is divided into a Compressor and Decompressor parts.

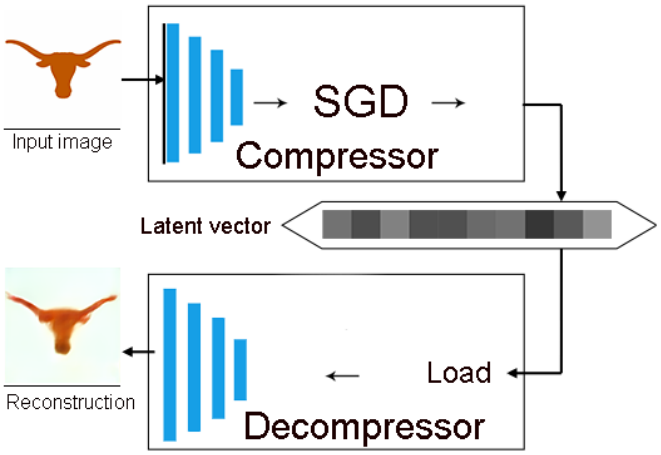


Figure 3.2: The Compressor-Decompressor uses the GAN Generator from Figure 3.1 above to generate the output image using most optimal latent vector representation.

As showing in Figure 3.2 above, the Compressor is composed of the GAN Generator and an input image to find and return the latent vector using Stochastic Gradient Descent (SGD).

The Decompressor then uses the GAN Generator and latent vector from the Compressor to produce the final reconstructed image.

Following the Decompressor generating an output image, the pre-trained ViT then detects objects from a dictionary of labels it has been trained on. The ViT outputs a probability distribution, i.e. a probability for each label summing to unity (Softmax, refer to Section 1.4.2). The final output of the script developed for this project is the ViT-Score, and other relevant image quality metrics (e.g. SSIM, MSE, etc) to evaluate the generated output image.

A PyTorch implementation of the ViT was used, trained on ImageNet-21k and fine tuned on ImageNet-1k (refer to Section 1.5.1).

Python and Python libraries were used throughout the implementation of the architectures described above (see Appendix A for details).

3.1.3 The GAN

A standard Generator-Discriminator GAN architecture, called a "PGAN" (Progressive Growing of GAN), was used from a pre-trained model. The model was trained on three major datasets: "celebaHQ", "fashionGen", and "DTD". "CelebaHQ" is a dataset of celebrity faces. "FashionGen" is a set of fashion objects such as clothing and accessory items. "DTD" (Describable Textures Dataset) is a collection of textures such as checkered patterns, foods, and animal fur. [Facebook-Research, 2022]

Figure 3.3: The pre-trained "PGAN" by Facebook-Research, implemented in PyTorch. It is trained on three diverse datasets of objects.[Facebook-Research, 2022]

The presumption is that since a "PGAN" is trained on these three diverse datasets, it should, in principle, be able to compress and reconstruct any input image. Its output however, may vary significantly, depending on the complexity of the input.

3.1.4 Latent space vector representation

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

The exact meaning of the vector is hard to decipher. Figure 3.4 below shows a sample of the compressed image vector representation:

Figure 3.4: Visual representation of the first 10 pixels of the most compressed form of the original image.

The figure above demonstrates what the GAN architecture compresses the full image to. In this case, an original image of size 409 kilobytes was compressed to a 1.54 kilobyte image vector.

The resulting Compression Ratio (CR) is 266 : 1. Such a Compression Ratio is an order of magnitude higher than deterministic compression algorithms would ever be able to achieve. [Ehrlich, 2022]

The GAN then proceeds to rebuild the image to 242kb. The final Compression Ratio is 1.69.

3.1.5 Sample Input Images Used

Four images of size 512x512, in PNG format are used in this study. These images represent distinct types of image to be compressed, containing diverse features, in order to challenge the model architecture.

For example, one image ("Logo") contains a white background, while another ("Bevo") a black background. The two completely dissimilar images, "Logo" and "Bevo", also happen to contain similar shapes, representing a graphic and a living Texas longhorn bull. "Bevo" contains an animal head and face. The image "Kliment", a portait of the author, contains a human head and face with trees and clouds in the background.

The image "Tower" is a building, the University of Texas at Austin Main Tower, containing trees in the foreground, but clouds in the background.

This set of four images was carefully chosen to represent similarities and dissimilarities in order to test the robustness of the ViT-Score and capacity of the Vision Transformer itself.

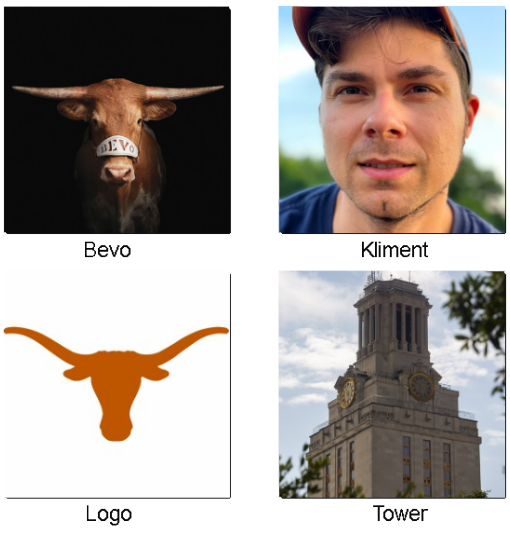


Figure 3.5: Input Images used in this project (512x512):

"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.

3.2 Outputs and Visual Inspection

The generative process from the GAN was designed to output an image at a specified epoch. Along with manual stopping, criteria for early stopping was set as well.

Moreover, learning rate reduction on plateau was used. This technique reduces the learning rate hyperparameter as the loss stops reducing.

The generated images are output for the user to monitor the GAN learning process.

3.2.1 Training Process

Below is a demonstration of the GAN training to compress and reconstruct "Logo".

Samples are collected at every 250 epochs:

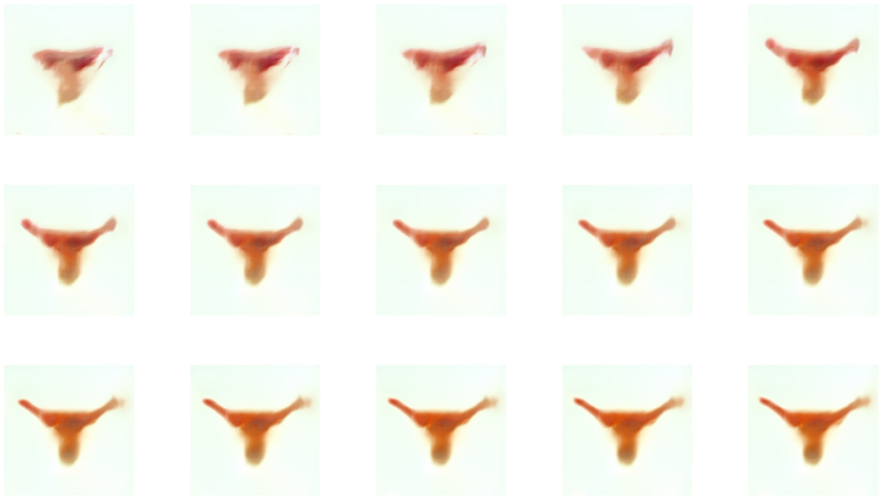


Figure 3.6: The GAN learns to compress and generate the Texas Longhorns logo. It starts from a blob, then 3500 epochs later, it is able to produce a completely recognizable version of "Logo".

Due to the novel nature of the technologies used, a natural performance asymptote was observed. The "PGAN" itself is trained on diverse data, yet this data is not able to capture the complexity of all potential input images. Thus, 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. Losses start increasing and output image quality degrades.

3.2.2 Generated Results

Figure 3.7 below shows the resulting images from neurally compressing and reconstructing all four sample input images.

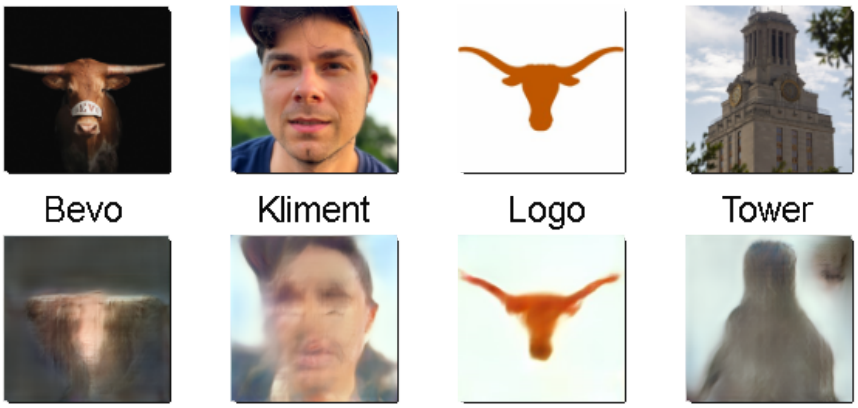


Figure 3.7: After sufficient training, the "PGAN" outputs a regenerated version of the original image from its latent vector representation. All four reconstructions resemble their inputs, to a varying degree of success.

The GAN was able to respectably regenerate "Kliment" and "Logo". "Logo" is completely recognizable as the Texas Longhorns logo, especially if the resolution is to be lowered (e.g. 32x32). In "Kliment", the model was successful at reconstructing a human head and the background features.

Nevertheless, the GAN was unable to perform as well for "Bevo" and "Tower". It learned random noise and generation performance decreased. Though "Bevo" does resemble a living Longhorn bull when viewed next to its original, a standalone image would be confusing to a viewer, as well as a Deep Learning-based object detector. "Tower" is clearly the worst output, since the generated image only loosely matches a blurry version of the original shape.

3.2.3 Compression Ratios (CR)

Compression Ratios (CR) are calculated as:

$$CR = \frac{original\_size}{compressed\_size}$$

However, it is important to note that the CR can technically be computed as the ratio of the original image size to the latent vector size. Since the latent vector compresses to about 1.5 kilobytes , the resulting compression ratio would always be around on the order of $100:1$ to $1000:1$.

This latent vector CR does not necessarily help with evaluating a transmitted image. For the purpose of this section, the size of the GAN reconstructed image is used as a more conservative measurement of compression.

This conservative measure of performance would also demonstrate model superiority over established compression techniques.

Bevo	0.75 *(failed to compress)*
Kliment	1.69
Logo	0.12 *(failed to compress)*
Tower	1.85

Table 3.1: Compression ratios of generated images.

The GAN failed to compress two of the images, "Bevo" and "Logo". Understandably, "Logo" is actually a vector graphic with a clear background, with an input file size of only 13.4 KB. The GAN had to create the white background, which it inefficiently stores as information, costing extra bandwidth and disk space. "Kliment" and "Tower" were compressed to half their original sizes. "Kliment" is a suitable candidate to demonstrate the capacity of this generative model. The image has a high Compression Ratio (CR), while also maintaining a high ViT-Score. After all, the most popular application and training set of most GANs, included in this pre-trained model (celebaHQ dataset), is indeed human faces.

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 this interval are unattainable by the ViT-Score. Probabilistic models such as this GAN, would generate enough similarities, though never an exact match.

The ViT-Score can be considered a contextual measure. It captures a higher abstraction of pixel arrangement based on objects detected under a certain probability. Such measures can be insightful, especially when evaluating generative models.

3.3.1 Mathematical Formulation

The following is a mathematical representation later explained in further detail.

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

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

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

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

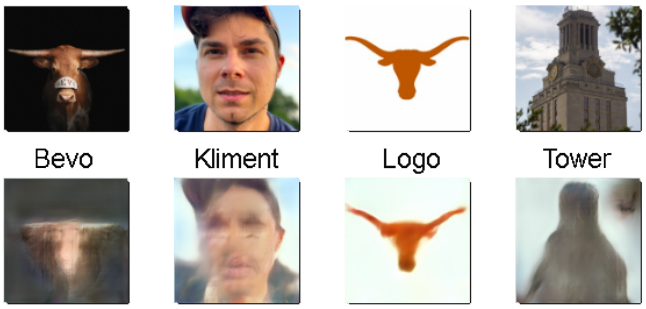
This overly elaborate mathematical notation is an attempt at describing:

"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:



Resulting generated images.

ViT-Scores

Bevo	0.14
Kliment	0.54
Logo	0.29
Tower	0.03

Table 3.2: 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

Refer to Section 2.5 "Metrics for Image Quality" for detailed definitions of each measurement technique.

3.4.1 "Kliment"

Figure 3.8: The image is structurally reconstructed well. Shortcomings are found in the facial features within the face.

This image achieves a ViT-Score of 0.54, SSIM of 0.56, MSE of 1,759.28, and a PSNR of 15.64 with a Compression Ratio of 0.59. Major shortcomings have less to do with structure, and more with details and features.

3.4.2 "Logo"

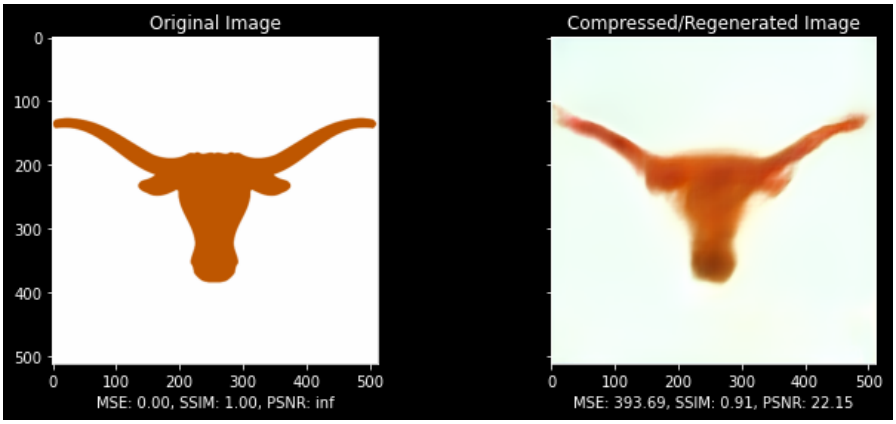


Figure 3.9: The GAN is able to reconstruct an almost perfectly recognizable version of the original logo.

The generated logo is extremely well identifiable. This image achieves a ViT-Score of 0.29, SSIM of 0.91, MSE of 393.69, and PSNR of 22.15. However, it fails to compress less than its original size. The GAN had to reconstruct the white background, which comes out off-white.

3.4.3 "Bevo"

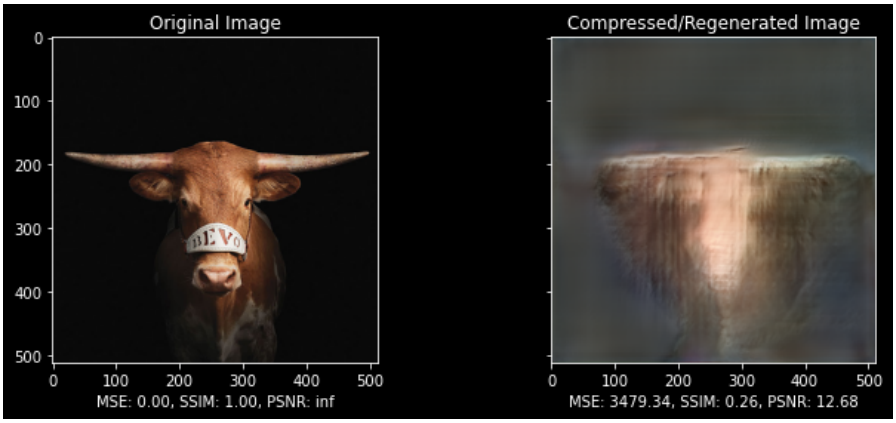


Figure 3.10: The GAN was unable to reconstruct details inside the longhorn, yet the structure of the image is decently rebuilt. One could possibly identify the animal from the generated image.

"Bevo" achieves a ViT-Score of 0.14, SSIM of 0.26, MSE of 3,479.34, and PSNR of 12.68 while failing to compress at all.

3.4.4 "Tower"

Figure 3.11: The GAN completely failed at reconstructing a comprehensible image.

This reconstruction is the lowest quality of all four images and we rightfully expect to have the lowest scores on all metrics. Both quantitatively and qualitatively, the generated image is considered incomprehensible. It has the lowest ViT-Score at 0.03, SSIM of 0.53, MSE of 1,315.75, and PSNR of 16.90. The fact that it does compress is irrelevant, since the image lost structural, textural, and feature detail information.

3.4.5 "BRISQUE"

The Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) metric is also defined in Section 2.5. BRISQUE measures distortions, with the score approaching 0 is a good score and approaching 100 is a bad score. This methodology could help evaluate an image as being more photorealistic than not. In terms of GAN output quality, this compares to a camera captured image with quality corruption caused by blurs or graininess.

	Original	Generated	Delta
Bevo	32.9214	39.5535	6.6
Kliment	-8.3593	44.3570	52.7
Logo	102.9010	97.1844	5.7
Tower	14.5973	52.8363	38.2

Table 3.3: BRISQUE Scores of original and generated images.

Expectedly, the BRISQUE values for the generated images are always higher than their original counterparts. That is, image quality always worsens after compression. "Logo" is not a photo-realistic 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. Largest Deltas (changes from original to generated) are also found in the images with the highest Compression Ratios (CRs), "Kliment" and "Tower".

A MATLAB implementation of BRISQUE was used to compute these scores.

3.4.6 GAN-Related Quantitative Metrics

There are two prominent evaluation metrics for the performance of a Generative Adversarial Network (GAN). Specifically, the Frechet Inception Distance (FID) and Inception Score (IS). Both measurements serve to evaluate the synthetic nature of the generated output.

The FID score, defined in Section 2.5, is 0 if there is no difference between the two multidimensional Gaussian distributions compared.

The **FID** scores are presented in Table 3.1 below:

Bevo	1,241,999.901
Kliment	549,089.491
Logo	81,105.162
Tower	331,171.556

Table 3.4: The Frechet Inception Distance (FID) score for all test images.

The FID score can be used as a Loss functions as well, embedded within the architecture of the GAN. However, it provides notoriously incosistent results, and is not as robut as MSE, which was used as the Loss function of choice for this project. None of the FID scores above provide much insight into the capacity of the GAN to generate a realistic image.

3.5 Summary of Results

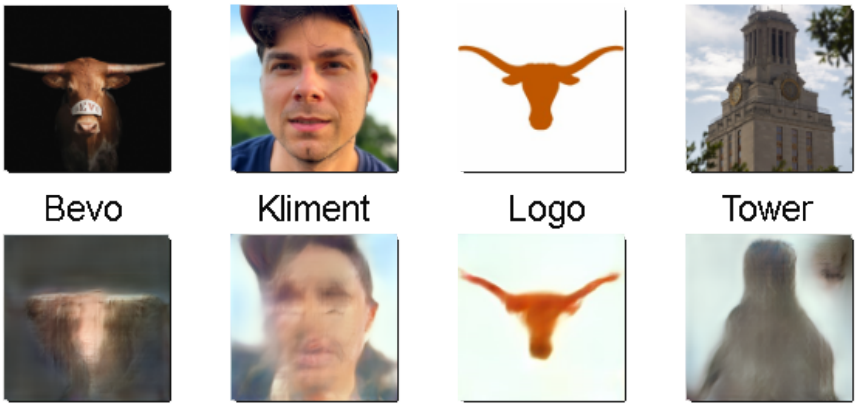


Image	ViT-Score	SSIM	MSE	PSNR	FID	BRISQUE	CR
Bevo	0.14	0.26	3,479.34	12.68	1,241,999.901	39.5535	0.75
Kliment	0.54	0.56	1,759.28	15.64	549,089.491	44.3570	1.69
Logo	0.29	0.91	393.69	22.15	81,105.162	97.1844	0.12
Tower	0.03	0.53	1,315.75	16.90	331,171.556	52.8363	1.85

Table 3.5: ViT-Scores provide an insightful quality assessment compared to established methods.

Chapter 4

Discussion

This chapter includes a critical discussion the results from Chapter 3. The reader is presented with a thorough evaluation of shortcomings and potential improvements to the architecture and components used in this project. Possible optimization improvements are also discussed with regards to Loss functions, methods, training datasets, and input images. Finally, based on the results shown, the reader is invited to a captivating discussion of the present and future of image-based Transformers.

4.1 Discussion of Results

The results summarized in Section 3.5 demonstrate that a ViT-Score is capable of:

- providing higher abstraction (object-level) insights
- comparing similar images for aberrations and distortions
- assessing the quality of a GAN generated image
- complementing existing metrics as a useful measure of image quality
- providing a novel approach to Deep Learning-based compression evaluation

"Tower" was the most incomprehesible generated image. The GAN failed to reconstruct a recognizable object, which resulted in the lowest ViT-Score among all four input images. This score is a testament for the valuable insights the ViT-Score contributes.

"Logo" ViT-Score could be higher, as are other metrics. The ViT object detector potentially struggled with finding the exact labels that would match such a unique symbollic graphic. Furthermore, the original input can be a transparent (backgroundless) vector graphic. If the backgrounds are to be ignored, perhaps all scores would increase.

Some of the metrics such as PSNR, FID, and BRISQUE seem to lack significant contributions in their assessment. They have the tendency to be unable to capture synthetically induced aberrations, which are obvious to the human eye. BRISQUE fairly consistently ranks the images in their photorealism. "Logo" is indeed a non-realistic synthetic symbol. The reconstruction of "Bevo" reminds the reader of an oil painting, yet BRISQUE ranks it most realistic with a low Delta. BRISQUE Deltas (from Table 3.3) can be insightful as well, since the smallest changes in score determine a level of consistency in evaluation.

Furthermore, the generated "Kliment" and "Tower" seem to have a similar PSNR, yet "Kliment" is a clearly defined object, whereas "Tower" is the least recognizable one.

In fact, in the case of "Kliment" and "Tower", the *ViT-Score proves to be more insightful than an established metric such as PSNR.*

Since features are typically found in the deeper layers of the GAN network, it is substantiated that a lot of structural information can be compressed into the 1-dimensional latent vector. The latent space is hard to decipher, but a Transformer is efficient in packing and unpacking information from it. The Compression Ratios associated with a fixed size latent vector can be useful in low bandwidth transmission scenarios.

4.2 Improvements

4.2.1 ViT-Score

The ViT-Score (defined in Section 3.3.1) could be further improved or experimented with. For the purpose of this project, the ViT-Score was based on how many of the top-100 labels match between the input and generated images. Experiments with the top-k value could yield a more optimal k, since 100 was chosen arbitrarily.

The ViT-Score can also take into account probability of each label found (included in the script output, shown in Figure 3.1). However, while the label probability is stable when working with corrupted images, it is rather unstable when working with generated images. Nonetheless, much like the SSIM (defined in Section 2.5), an elaborate regularized equation could include the output probabilities per detected label. In turn, this could improve the ViT-Score.

Intuitively, the ViT and ViT-score can also be used as a Loss function in the model architecture.

4.2.2 Compression Engine

The pre-trained "PGAN" used in the compression engine needs to be trained on more data inspired by human perception. An important note is that the Fashion-Gen dataset contains 293,008 high resolution images of size 1360x1360. The DTD texture dataset, includes 5640 images, with 120 images in 47 categories of varied resolution between 300x300 and 640x640. There is an obvious dataset imbalance, as the pre-trained model used has had more exposure to certain classes of images than other. [Facebook-Research, 2022]

Thus, it can be concluded that the natural performance limit to the capacity of this pre-trained model comes from its training sets. This applies to both the pre-trained "PGAN" and the pre-trained ViT. In a more capable iteration of this project, a GAN and ViT trained on all internet images, or a sizable, diverse, and representative subset, is necessary.

Evaluating output quality from Generative Adversarial Networks (GANs) is still a developing field, which uses non-Deep Learning-adapted assessment methods. It is expected that new and more capable methods of evaluating generated output will emerge. Furthermore, using these future GAN metrics as Loss functions inside the compression engine could significantly improve the GAN generated output.

Finally, unique positional encoding in the ViT can be achieved using trigonometric representation (e.g. periods of a sinusoid). This could be extremely useful when scaling the approach to elaborate high-resolution input images with lots of objects. For example, the delta between the original and generated images could be used to find and fine tune discrepancies between rows of pixels. Thus, as the exact location of each object (token) is known, deviations may be used to penalize the output and steer the Generator into improving its output. [Dosovitskiy et al., 2021]

4.3 Optimization

Some of the optimization techniques used in this work include:

- Learning rate reduction on plateau, hyperparameters for patience, threshold, and eps
- SGD, hyperparameters for learning rate and momentum
- Varying input images

One of the most valuable techniques, which aided the GAN in improving its output quality, was reducing the learning rate as the model trains. Stochastic Gradient Descent (SGD) was used as an optimizer for the GAN. Perhaps substituting SGD with Adam could yield better results on certain image types, though probably not on average.

A major opportunity to optimize the compression engine would be to experiment with Loss functions. Mean Squared Error (MSE) was used throughout this study, however, SSIM or potentially a GAN specific evaluator such as FID could achieve better results (refer to Section 2.5 for definitions). Additionally, experimenting with input image types to cater to what the generative model is best trained on could yield much better results as well. Other options to experiment with include introducing regularization during training such as residual dropout and label smoothing.

When analyzing the latent space vector, it is possible to engage an adapted Transformer model. This overparameterized model can create maps from the latent space to the spatial representation of the image. This generalizable approach can steer the GAN to train faster, compress better, and output higher quality images. Finally, Transfer learning or combinations of deterministic and probabilistic methods can combat slow training times (as mentioned in Section 2.4).

4.4 Present and Future of Image Transformers

4.4.1 Status Quo

According to Figure 1.1 and Section 1.1, there is a clearly defined increase in interest in using Vision Transformer-based models for applications in image processing and computer vision. Furthermore, there has been a significant increase in relevant recent publications related to the topic of this thesis. A staggering thirteen (13) publications in 2022 thus far support the vision of tying Vision Transformers (ViT) to image compression. [Arxiv-Query, 2022]

A clear statement must be made that the present day is still an early stage in the Deep Learning and Artificial Intelligence evolutionary process. Many of the models have yet to be trained on sizable amounts of image data. Hence, the current models are still suffering from underperformance in tasks that are easy to achieve by human perception.

Finally, the existing approaches to image compression, analysis, and generation may not reflect the most efficient ones, but rather the most scalable and universally accepted. [Ehrlich, 2022]

4.4.2 Future Developments

The future of pre-trained GAN and ViT models will include extremely wide and overparameterized feature sets. Training will be done on sizable sets of images from the internet to mimic the development of the text-based language model Transformers such as GPT-3 (see Section 1.6). An extremely large dataset (on the order of petabytes of data) featuring diverse and representative images will be compiled and used for training by a major technology company or AI initiative.

A coveted and highly desirable Deep Learning-based image compression technique will emerge, which will succeed the use of JPEG and other classical compression techniques.

For the purpose of this thesis project, a Convolutional Neural Network (CNN) based Generative Adversarial Network (GAN) was used as a placeholder for a more generalizable alternative. Perhaps, it could also be Transformer or Vision Transformer-based. In fact, a publication in Computer Vision and Pattern Recognition (CVPR) 2021 dubbed "TransGAN: Two Pure Transformers Can Make One Strong GAN" was proposed by a team from The University of Texas at Austin. [Jiang et al., 2021]

It demonstrates that the Generator and Discriminator in a GAN could be replaced with Transformers free of convolutions. These TransGANs were able to match or exceed the performance of similar CNN-based GANs. [Jiang et al., 2021]

In the future, GANs will improve immensely, as the approach proposed by Ian Goodfellow in 2014 is still only in its first generation as a highly promising generative technique.

To conclude the discussion on the future of neural image compression, it can be said that all developments will organically extend into neural video compression.

Chapter 5

Summary

This chapter serves to summarize this thesis report.

It provides a review of the 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 report concludes with acknowledgments to key contributors to the success of this project and closing remarks.

5.1 Key Contributions

The main merit of this thesis project is introducing the ViT-Score (see Section 3.3). It is a Vision Transformer-assisted metric for evaluating the performance of neural image compression. It computes a score to compare the reconstructed output from a Generative Adversarial Network (GAN) to the original input image.

The ViT-Score conclusively contributes to evaluating image quality by quantifying human perception of recognizable objects in the generated image. The metric can also provide additional insights to understanding the latent space (contextual) preservation of an input image. Finally, the ViT-Score can most likely verify that a GAN has generated a comprehensible image.

This work can also be viewed as contributing towards:

- an end-to-end Deep Learning-based approach to image compression and reconstruction
- abstracted evaluation of GAN generated images
- promoting generalizable and overparameterized models in Deep Learning
- experimentation with Transformers while still a nascent technology
- expansion of human consciousness through investigation of evolving techniques in Artifical Intelligence (AI)

5.2 Summary

Throughout this report, the reader was presented with all relevant background knowledge necessary to grasp the key contributions listed.

A Vision Transformer (ViT) was used to evaluate the capacity of a GAN to compress and generate an image of choice based on object-level similarities with the original input image.

The new metric, referred to as a ViT-Score, was able to capture and assess the quality of the output images and provide valuable insights. The ViT-Score performed well, comparing in capacity to established image quality metrics such as SSIM, MSE, and PSNR.

5.3 Takeaways

The future of image compression technology will be based on a Deep Learning methodology. Due to their generalizability, excellent performance in 1-dimensional data (text), and proven ability to scale to 2-dimensions, Transformers are an excellent choice of architecture to use in image compression and reconstruction.

A Vision Transformer (ViT)-Assisted metric related to image compression can provide additional insights to understanding latent feature space and its preservation through lossy compression.

Such a metric could be used as a Loss function, embedded within the architecture. It could be useful as a standalone evaluation metric.

It may cost on the order of $100M and several years to develop, but an end-to-end deep learning-based approach to image compression will be achieved.

Finally, such image compression technology can be extended to video and video compression in further developments.

5.4 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 Dr. 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 (GANs).

CSE 382, Foundations of Machine Learning taught by Professor Dr. 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 Texas Advanced Computing Center (TACC) provided free access to advanced High-Performance Computing (HPC) resources, which were used throughout the experimentation process in this thesis.

5.5 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 has been made available as open source to the general public in the form of a GitHub repository.

Bibliography

Arxiv-Query. Advanced search | arxiv e-print repository. https://arxiv.org/search/advanced?advanced=&terms-0-operator=AND&terms-0-term=vision+transformer+image+compression&terms-0-field=all&classification-physics_archives=all&classification-include_cross_list=include&date-filter_by=specific_year&date-year=2022&date-from_date=&date-to_date=&date-date_type=submitted_date&abstracts=show&size=50&order=-announced_date_first, 2022. Accessed: 2022-04-30.

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.

J. Brownlee. How to implement the frechet inception distance (fid) for evaluating gans. https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/, 2022. Accessed: 2022-04-30.

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.

Facebook-Research. Pytorch gan zoo. https://github.com/facebookresearch/pytorch_GAN_zoo, 2022. Accessed: 2022-04-30.

Google-Research. Vision transformer and mlp-mixer architectures. https://github.com/google-research/vision_transformer, 2022. Accessed: 2022-04-30.

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. An image is worth 16x16 words: Transformers for image recognition at scale. https://paperswithcode.com/paper/an-image-is-worth-16x16-words-transformers-1, 2022a. Accessed: 2022-04-30.

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

Pathmind. A beginner's guide to generative adversarial networks (gans). https://wiki.pathmind.com/generative-adversarial-network-gan, 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.

Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

APPENDIX

A. Software Stack

Python, PyTorch MATLAB for BRISQUE LaTeX to generate this PDF 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
```

B. Hardware

NVIDIA GTX 1650Ti CUDA 11 GPU, Local machine

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
```