Facial Image Reconstruction

Connor Killingbeck, Ethan Kim, Khoa Nguyen, Sam Vousden

Introduction and Motivation

Since the inception of the internet and the sharing of files, compression algorithms have provided an effective way to optimise the transfer of data. With ever increasing file sizes due to increased complexity and internet speeds continue to skyrocket, a nice equilibrium has been maintained so far. However, in today's ever competitive technological landscape, the ability to be faster is of high value. This is especially true with images, a notoriously large file type that has become more and more transferred as the years go on.

In this paper, we will look into the ability to increase the speed of file transfer by reducing a file's size using Neural Networks, hoping to increase the speed in which technologies that use file transfering operate.

Problem Description

The Problem here can be boiled down simply into the following: Lossy Compression using Neural Networks for the purposes of Facial Image Compression. The Key is to find a means in which we can reduce images down into lossy alternatives that are a fraction of the size of the original, so that we may deconstruct faces into a significantly smaller form and then recreate them for smaller file transfers. Unlike compression schemes such as PNG, information in the original image can be lost, hence "lossy" compression. Since we are boiling an image down into numbers and then reconstructing it again, we can not preserve everything, and as such, details will be left behind. Since we are attempting to reconstruct images of faces, this offers a unique challenge and lower leeway for failure. The Human face is a complex structure that is heavily reliant on the dimensions and proportions of its facial components. When deconstructing and reconstructing an image, even slight deviations in the jaw length, nose position, eye distance, skin contrast, lighting, hair style, ect will cause the reconstructed image to have less to even no resemblance to the original, or may even put the image inside of the "uncanny valley". Simple items such as basic everyday objects do not suffer from such scenarios to the same extent, as their simple silhouette and visual profile makes it so if there are slight deviations, as long as the overall reconstruction is close to its original profile, it more likely than not appears to be acceptable. This is part of the challenge, as getting the network to identify key facial structures and patterns particular to each individual is a necessity in order to have the reconstruction bear any resemblance to the original.

Contribution and Workload Distribution

- Khoa: Skeleton Code, Variational Autoencoder skeleton code, Generative Adversarial Network training loop, hyper-parameter tuning the Generative Adversarial Network model. Paper contributions: Implementation section of the Deep Convolutional Generative Adversarial Network model and Future Works and Improvement. General consultation and proofreading.

- Connor: Completion and Hyperparameter Tuning of Variational AutoEncoder, Paper Contributions Include Introduction, Problem Description, Alongside Related Works, Metrics Assistance, and Dataset Breakdown. Slideshow Creator, Video Editor, Demo Man.
- Ethan: Evaluation of Variational AutoEncoder, Sourcing of Masked Images. Paper contributions: Evaluation Metric Tables, Discussion "Masking Efficacy", Conclusion.
- Sam: Discriminator for DCGAN, helped hyperparameter-tune DCGAN, demo code, Slideshow Presenter.

Related Works

Previous Works that attempted similar tasks to our include the (Hou, 2017), in which a Variational Autoencoder was used with a distinguished loss function. A pre-trained VGG network was employed to use the outcomes found at the 19th layer as a feature for the perceptual loss. The goal was to use a facial boundary map to deconstruct and then reconstruct the images. Their success with this method was mixed.

A new, more experimental method for the compression and reconstruction of images is a VAE mixed together with Vector Quantization, where the vectors are learned over the course of training. The prototype vectors found in this network that have been determined by Vector Quantization are learned throughout the course of the training. One learned, a hierarchical level based system is employed where levels of representations and complex prior distributions of the latent space are used like self-attention layers. The network essentially considers all past successes and failures to train further and achieve a greater result (Vahdat, Kautz, 2020).

Other common, albeit less effective approaches include the fusion of VAE with an accompanying Gaussian Mixture Model. This approach attempts to learn about adversarial loss to accompany the reconstruction loss, hoping to replace any element wise errors with errors that would instead be seen as more feature wise. The combination aimed to create a more complex version than standard single Guassian distributions in the hope that the increased complexity would be further reflected in the ability to detect better face features such as poses, age, and complexions (Qian et al., 2019).

In more recent works, the (Toledo, Antonelo, 2021) paper attempted to combine all of these with an additional face mask component. The goal of the face mask was to help the training of VAEs for face reconstruction, by restricting the learning to the pixels selected by the face mask. This is ideal, as the background information is not present to clutter up the training and potentially throw off the results. Background information is often out of focus, blurry, or being obstructed by the subject in the foreground. This often leads to bad results as models tend to view this mishmashing of background colours as easy ways to mitigate loss, for example, a model may learn that just putting green in the background will mitigate loss between two images, since one takes place in an area filled with vegetation. As this is the

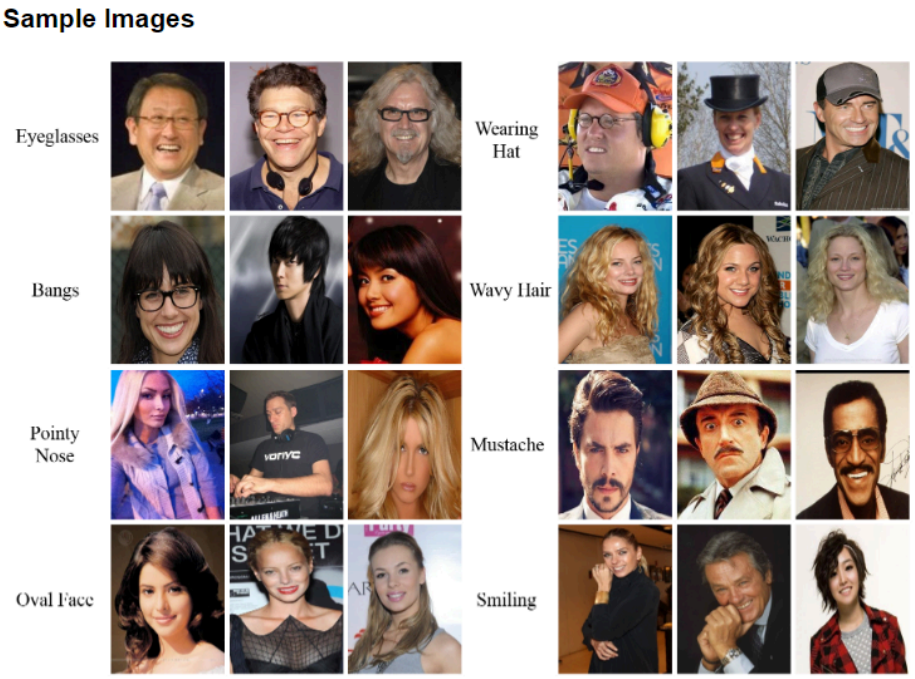
most recent paper on the subject, we will consider this as the state-of-the-art model and act as the benchmark for our models' performance.

Additionally, we will make additional comparisons to common and popular lossy compression techniques used by other picture formats. The Joint Photographic Experts Group format, also known as JPEG, offers good size reduction when it comes to compression of image files to a smaller size but suffers from large losses to detail and image quality. This similarity will allow for useful comparisons between our model and standard, algorithmic file compression methods. (Santa-Cruz, Ebrahimi, 2000)

Dataset

A common dataset for facial images is CelebA, also known as CelebFaces Attributes Dataset and proposed by Liu et al. (2018), is a large-scale face attributes dataset with more than 200K different celebrity images, and was created and curated by the University of Hong Kong. Each image has around 40 attribute annotations, such as "Eyeglasses", "Wearing Hat", and "Wavy Hair" just to name a few. The Dataset offers a large variety of poses, background clutters, noise, lighting conditions, and facial angles. The Dataset features multiple ethnic races, as well as:

- Over 10,000 Identities
- Over 200,000 individual face images
- Over 1,000,000 landmark locations, 5 per image
- Over 8,000,000 total binary attributes

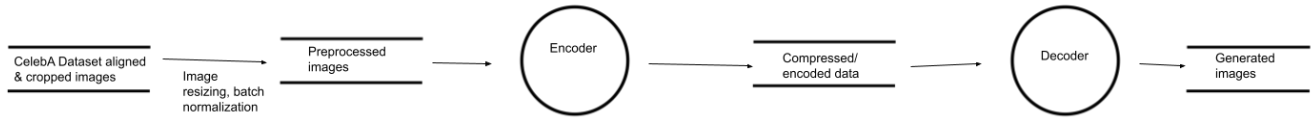


Specifically we are using the "cropped & aligned" variation of the dataset, which features only faces that are cropped out and resized to 178x218 in the Joint Photographic Experts Group (JPEG) format (Wallace, 1992). We chose this specific variation as opposed to

the full, uncompressed dataset due to its relatively small size of 1.8 Gigabytes, which allows it to be used on our computers which possess small storage space. However, it is important to note that JPEG is not a lossless compression technique. In other words, the images we are training and evaluating the model on are not raw images of faces, and can contain artifacts caused by the compression itself. However, experiments carried out by Schachner et al. (2023) showed that JPEG compression largely does not affect facial recognition. We theorize that such a method would retain the most identifiable facial structures, allowing our model to still effectively reach the goal of the project.

Implementations

Preprocessing



The data is loaded from the dataset and preprocessed using TensorFlow Keras image_dataset_from_directory, which loads and processes the images for the encoder. The images from the dataset are resized to 64x64 pixels and the data is normalized by a scale of 1/255. The dataset is split with a ratio of 10:1 for training and validation. The shuffle parameter is set to false to ensure consistency between runs.

Convolutional Variational Auto-encoder (CVAE)

a) Experimental setup:
   List your PC spec here, or describe the compute cluster we're using.
b) Model architecture**:**

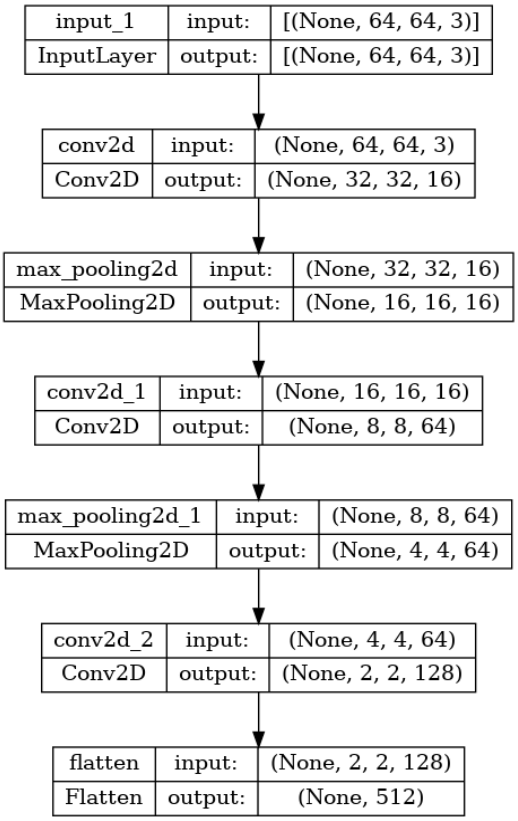


Figure 1: The Convolutional Variational Auto-encoder Generator model diagram for the encoder

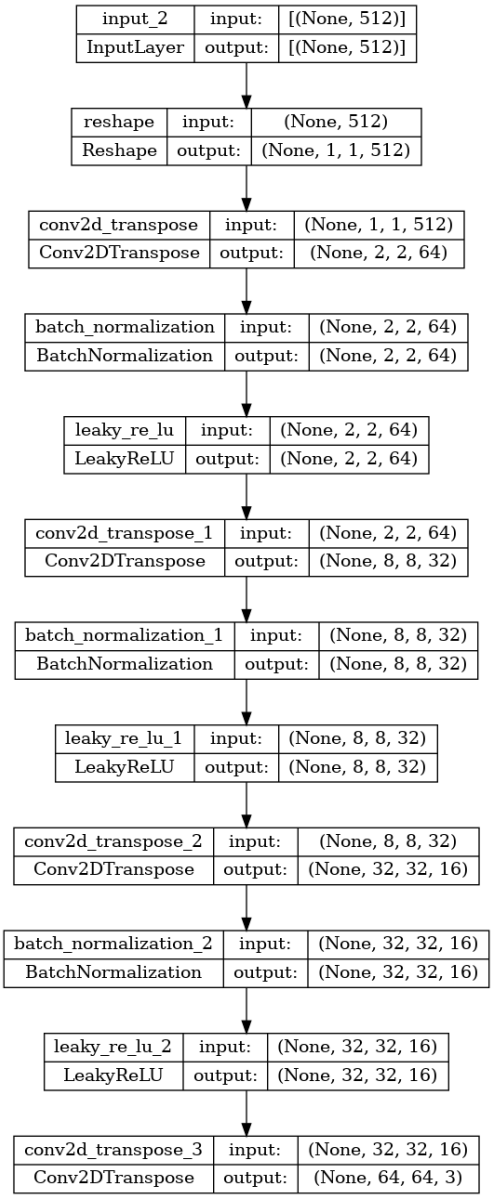


Figure 2: The Convolutional Variational Auto-encoder Generator model diagram for the decoder

The architecture used is based on the Variational Auto-encoder from Kingma, D. & Welling M. (2022). The encoder is a convolutional neural network that takes data from the 64x64 input image and encodes it into a latent variable with 512 data points. The decoder is a convolutional neural network that inputs the latent variable and decodes it into a 64x64 image. The decoder mirrors the encoder in its architecture, as it is ideally designed to reverse the process of the encoder.

c) Training Loop:

The model is trained to minimise L1 loss between the input data and the data reconstructed from the latent variable. L1 loss also known as Mean Absolute Error or MAE.

Assume a prediction matrix *y* of size *n*, and the ground truth matrix *x* of the same size, MAE is:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Although the Variational Auto-encoder consists of two models: the encoder as seen in figure 1 and the decoder as seen in figure 2, the entire setup was trained as a single model using backpropagation for simplicity.

Deep Convolutional Generative Adversarial Network (DCGAN)

a) Experimental setup:

The computer used to generate the final model is a desktop personal computer with the Ryzen 3700x CPU and RTX 3070 Ti running the Ubuntu operating system version 22.04.1 LTS. Software-wise, Tensorflow version 2.14.0 is used alongside CuDNN version 8600 to utilise the GPU for faster training time. The specific versions for all packages used were included in the submitted code in the "requirements.txt" file in pip-friendly format.

b) Model architecture:

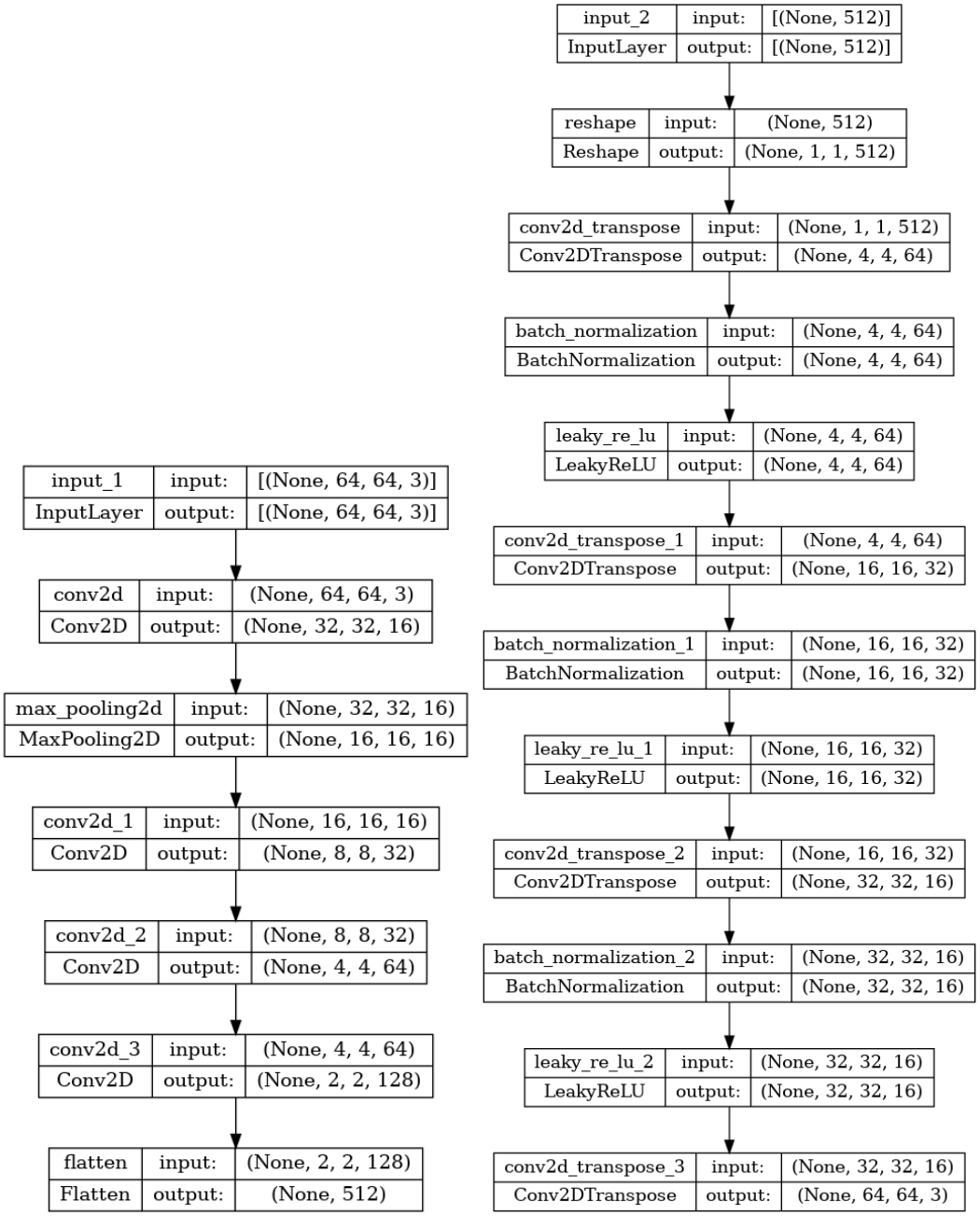


Figure 3: The Deep Convolutional Neural Networks Generator model diagram with encoder (left) and decoder (right)

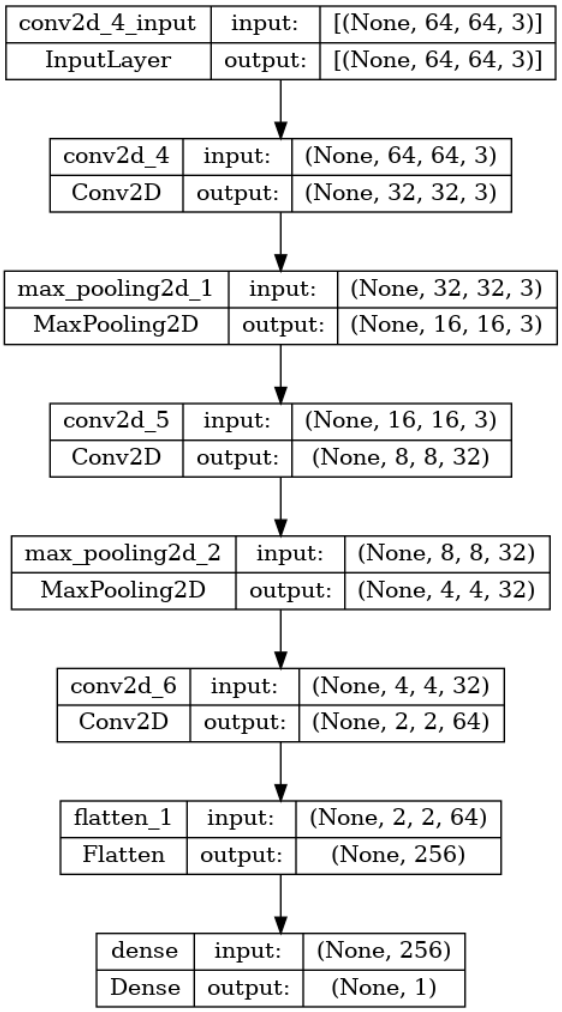


Figure 4: The Deep Convolutional Neural Networks Discriminator model diagram

The model architecture is based on the concept of Generative Adversarial Networks (GANs) as proposed by Goodfellow et al. (2014), which is very popular for generative tasks. Such a framework includes at least two models: the generator and the discriminator. The generator, as its name would suggest, generates data from an input, while the discriminator attempts to classify whether the data generated is real or generated. In practice, the GAN architecture is usually implemented as an extension of a VAE network, as the introduced discriminator often allows the generated data from such a network to appear more natural. This can be seen in our results section.

In our implementation of a Deep Convolutional GAN, the generator architecture is highly reminiscent of the CVAE model described previously. However, some changes were made to it to improve GAN performance. For example, it has more than double the parameters: 973,443 parameters vs 443,491. This is because earlier testing showed the discriminator overpowering the generator early on and causing an early convergence where the generator has yet to be able to generate proper facial structures.

The discriminator architecture in this case is just a simple Convolutional Neural Network meant to classify generated images and real images. The model itself was tuned specifically for use in a GAN setup, rather than individually as the success of a GAN is dependent on the generator and discriminator learning at the stable rate to not overpower one

another as described by Goodfellow et al. (2014). Accordingly, the model's depth and parameter count was specifically tuned via trial and error in order to not overpower or be overpowered by the generator.

c) Training Loop:

A traditional GAN setup would train the generator to maximise the discriminator loss and minimise the generator loss (Saxena, Cao, 2021). Because the generator follows the exact same architecture as the previously talked about CVAE model, the generator is trained in almost the exact same way, with the only difference being the addition of the discriminator loss in the equation.

The discriminator loss is given as Binary Cross-entropy loss, which is popular for binary classification tasks. Assume the probability of a point in the prediction vector p being 1 is y and, similarly, the probability of a point in the ground truth vector q being 1 is y hat, the Binary Cross-entropy between p and q is:

$$H(p,q) = -\sum_i p_i \log q_i = -y \log \hat{y} - (1-y) \log(1-\hat{y}).$$

However, in this project, we find that our GAN setup suffers from model collapse as described by Saxena and Cao (2020). Case in point, the generator would only attempt to trick the discriminator instead of trying to recreate the original image. We theorize this is because of the small values returned by the generator due to normalization of the image as mentioned in preprocessing. As the image's data only contain floating point values in the [0, 1] range, the generator loss returned during training can be very small, even if the images generated look nothing like the original. Because of this, we scaled the generator loss by a factor of 10 to give it more weight in training. This largely resolved the issue.

Another potential solution for this issue as discussed by Toledo and Antonelo (2021) is to de-normalize the images before evaluating loss by scaling all pixels by a factor of 255. This allows loss to be evaluated on data in the [0, 255] range, and by extension making the loss value a lot more significant. In addition, background masking can be implemented during training to allow the loss to be focused on just the face instead of the background, which, ultimately, is what we care about. This is significant as we will show later on in Results. This method is the main thesis of Toledo and Antonelo's 2021 paper and is shown to be very successful.

Training Variables and Testing

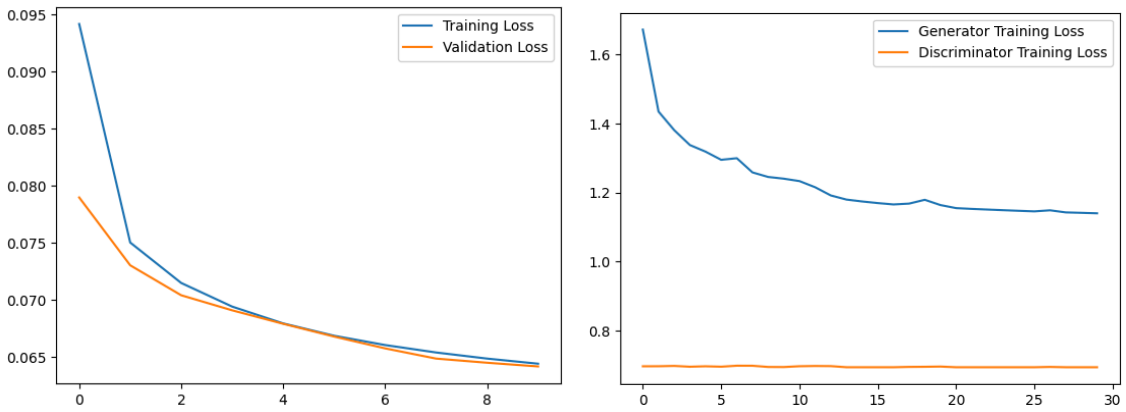


Figure 3: Training loss over time for CVAE (left) and DCGAN (right)

We trained the Conv-VAE model for 10 epochs and the DCGAN setup for 30 epochs on the preprocessed data in batches. Specifically, we used 100 images per batch to allow computers with less memory to run it. Moreover, we also split the data into training and evaluation sets with a 90-10 split. No early stopping condition was implemented, the number of epochs was decided upon via observation of the loss curve seen in Figure 3. For validation specifically of the CVAE model, we run the model on a number of examples in the test set and calculate the average MAE score (the lower the better). As for the DCGAN model, we validate the model visually by comparing the original and the reconstructed image generated every epoch. We focus on comparing the visual quality of the reconstructed image and the original as well as the identifiability of the reconstructed face, which we believe is the most important quality to achieve our goal.

Results

Metrics

In order to evaluate the quality of the generated images with respect to the images they were meant to mimic, two conventional methods would be L1 (Mean Squared Error or MSE) and L2 (Mean Absolute Error or MAE). L1's and L2's are calculated with the formulas:

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Whereas yi is the predicted value, xi is the true value, and n is the total number of data points.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Whereas n is the number of data points, Yi is the observed values, and Ŷi is the predicted observations.

Wang and Bovik (2009) cited MSE's simplicity and cheap computation cost as the main reasons behind its popularity. Moreover, due to its popularity, it is considered a standard for comparing algorithms. On the other hand, MAE is also commonly used alongside MSE. Unlike MSE, MAE does not increase punishment based on the error itself, which allows it to be more useful when outliers are present. Experiments by Zhao et al. (2016) have also shown that MAE can outperform MSE in certain tasks.

Beyond this, we also employed the Structural Similarity Index (SSIM) as well as its extension Multiscale Structural Similarity Index (MS-SSIM) to quantify the perceptual quality of an image (Wang et al. 2003). These metrics often complement MSE and MAE as they often do not correlate well with the perceived quality of images.

Toledo and Antonelo (2021) also incorporated two other evaluation paradigms: Learned Perceptual Image Patch Similarity (LPIPS) and the Visual Information Fidelity (VIF). However, VIF requires authorization from its authors, while LPIPS require the training of an extra model, which lies beyond the scope of this project. Fortunately, these metrics are still new and still relatively uncommon so our evaluation will not suffer significantly.

Evaluation

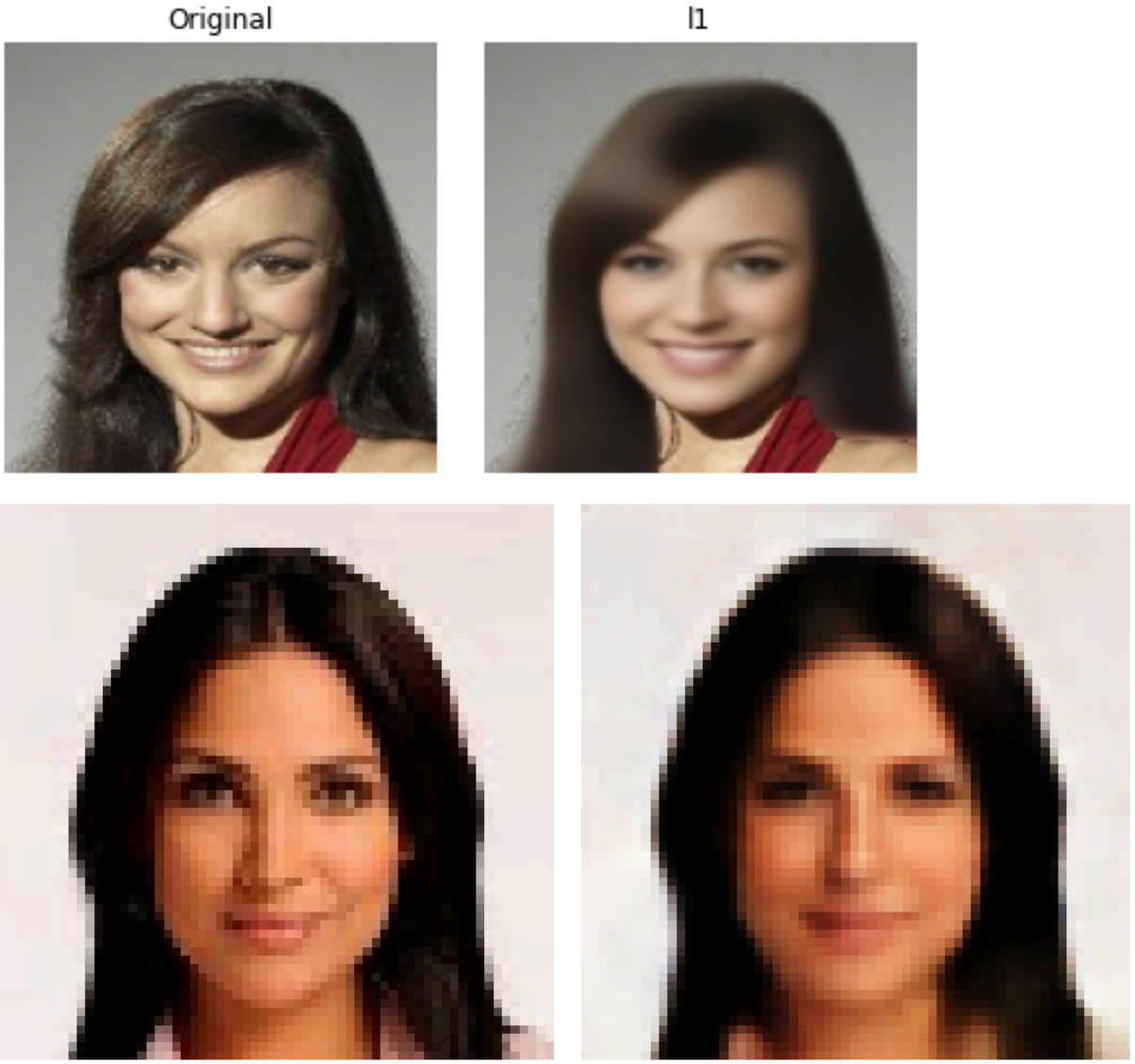
Our evaluation schema consisted of 6 methods, 4 of them use the models we have built with and without masking, the final is standard JPEG compression algorithm with a quality retention of .85 with and without masking. These hypotheses are denoted with the subscript of the model architecture and "Mask". The motivation behind this is to understand the quality of specifically the faces and the limits of encoding efficiency. The images from Toledo and Antonelo (2021) are presented first, then the CVAE and DC-GAN model after in said order.

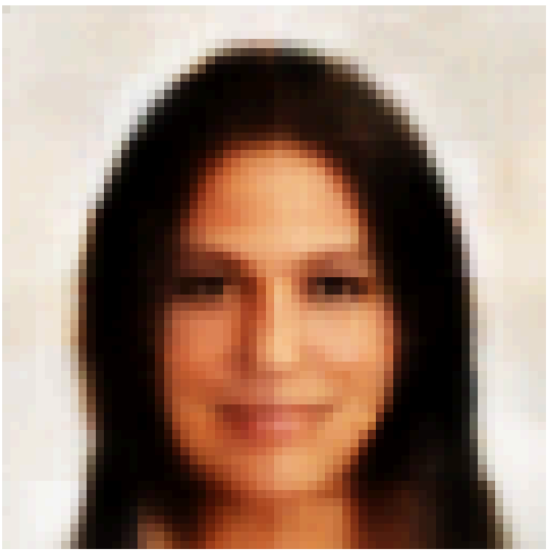
Table 1: Metrics for Compression Methods

Hypothesis	1 - SSIM	1 - MSSSIM	l_1	l_2
H_{CVAE}	0.307	0.067	0.439	0.475
$H_{CVAE+Mask}$	0.215	0.045	0.323	0.559
H_{DCGAN}	0.417	0.150	0.642	0.943
$H_{DCGAN+Mask}$	0.274	0.090	0.456	1.050
H_{JPEG}	0.004	0.000	0.003	0.000
$H_{JPEG+Mask}$	0.004	0.000	0.323	0.559

Table 2: Hypotheses results from Toledo and Antonelo (2021)

	1 - SSIM	1 - MSSSIM	LPIPS	1 - VIF	l_1	l_2
H_1	0.311	0.082	0.145	0.529	0.508	0.732
H_2	0.338	0.090	0.162	0.560	0.546	0.781
H_3	0.343	0.095	0.151	0.541	0.534	0.774
H_4	0.373	0.103	0.173	0.574	0.587	0.826
H_5	0.321	0.093	0.146	0.537	0.585	0.836
H_6	0.314	0.085	0.157	0.550	0.601	0.852
H_7	0.322	0.082	0.146	0.527	0.501	0.694
H_8	0.377	0.100	0.172	0.572	0.577	0.797
H_9	0.307	0.077	0.143	0.523	0.487	0.682
H_{10}	0.359	0.095	0.168	0.566	0.562	0.782



Discussion

Cause for higher MSE loss in masked images

Comparing the results, it is observed that the application of masking notably increases the MSE, while simultaneously decreasing the MAE, SSIM, MS-SSIM metrics. This phenomenon can be attributed to several factors. Firstly, the absence of a simplistic background, which is generally easier to predict, can augment MSE. In cases where the background is a uniform colour, such as white, and contrasts with an average facial representation, the model might achieve deceptively lower error rates. However, when masking is employed in evaluation, the model's focus is redirected exclusively to facial features, which are inherently more complex and detailed, including elements like wrinkles, freckles, and structural asymmetries. This refocusing of the models learning on facial features by removal of background is similarly mentioned in Toledo and Antonelo (2021) where their architecture is described as being specifically built to ignore background information and focus on important facial features.

Additionally, the process of masking could inadvertently omit crucial facial features due to partial obstructions in the original image, such as hands covering parts of the face, hats casting shadows, or hair partially obscuring the face. This facet of masking is commonly referred to as aliasing, where a mask omits or includes distortion or irregular features that interferes with an image's predictability. Jonscher, M et al. (2022) touches on this subject where an emphasis is made on the need for masking that does not cause any aliasing. The omissions caused by aliasing can lead to a loss of important contextual information, potentially impacting the model's ability to accurately reconstruct facial features and thus contributing to a higher MSE.

Comparison to original papers model

In comparison to the original paper's model, the model proposed in this paper is able to compress images while retaining higher quality. In all shared metrics between papers our model consistently results in significantly lower MAE, MSE, SSIM, and MSSIM. Although our model beats their model in performance, our smaller initial image size reduces the overall amount of pixels needed to be compressed for the final image. Our model reduces a 64x64 image to 512, while the original paper reduces a 128x128 to 512. This smaller compression ratio means our model can learn more features from the original image that can be satisfactorily reproduced in the output images. Our model exchanges image quality for image quality.

Comparison to current image compression standards

Lossy image compression is ubiquitously associated with JPEG. In order to provide a pragmatic view on artificial neural network methods of image compression such as the CVAE and DCGAN models used in this paper, JPEG image compression with a quality of .85, was compared with our models. It is evident that JPEG compresses the image with much less structural loss than any model presented in this paper or Toledo and Antonelo (2021). While considering the effectiveness of JPEG image compression based on our evaluation it is important to note how masking affects the MAE and MSE of any lossy image compression. Even with a standardised and proven method of compression, the application of masking will inherently skew MAE and MSE evaluations of imagery. This reinforces the need for alternative metrics, such as, for evaluating how similar reconstructed images are to the human visual system as described in Hang Zhao et al. (2015).

Future Works and Improvements

Although our models exceeded initial expectations, it still cannot compare to that of Toledo and Antonelo (2021). There are a few possible improvements that can be made to the DCGAN model, which we deem vastly superior to the CVAE model in terms of quality.

1. Incorporating masking to the training method. This was proven to be highly effective and the main contribution from Toledo and Antonelo.
2. Exploring other loss functions. Currently we are using L1 loss, however, with more time, it might be beneficial to explore L2, as well as SSIM and MSSIM.
3. Investigate the discriminator’s peculiar behaviour. As seen in the training loss history chart, the discriminator features an almost completely flat loss curve that does not change over time. Although there is a good chance this does not affect the performance of the model, it is worth investigating.
4. Explore higher compression ratios. We want to hit a ratio close to that of Toledo and Antonelo’s model to better compare performances.
5. Train existing models on raw, uncompressed data. The only reason this was not done already is because of time and storage constraints of the system we possess.

Conclusion

The findings presented in this paper indicate that both the CVAE and DCGAN are capable of significantly reducing image size while preserving aspects of perceptual quality found in the original image. With the CVA model in particular being excellent at image quality preservation while simultaneously compressing the image. In comparison to the model presented in (Toledo, Antonelo, 2021), our model results in higher compression, albeit at the cost of feature loss.

The application of facial masking in testing highlighted the importance of focusing the face contained in the image in order to improve image reconstruction. Although masking presented increases in MSE, SSIM, MS-SSIM and MAE were all thoroughly increased with the application of masking.

The higher quality reconstructions that masking produced indicates room for future improvement. Implementing masking in the training, as well as exploring alternative loss functions and compression ratio could provide further improvement to the model. Further, these tunings could provide insights into broader application towards image compression technology.

In conclusion, our work contributes to understanding the application of CVAE and DCGAN models in image compression, suggesting that neural networks hold promising capability in the efficient compression of images.

References

Abadi, Martín, Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … others. (2016). Tensorflow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (‘OSDI’ 16) (pp. 265–283).

Hou, X., Shen, L., Sun, K., and Qiu, G. (2017). Deep feature consistent variational autoencoder. In 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 1133–1141. IEEE.

Jonscher, M., Seiler, J., Richter, T., & Kaup, A. (2022). Reducing Randomness of Non-Regular Sampling Masks for Image Reconstruction. arXiv:2204.04065v1 [eess.IV]. Friedrich-Alexander-Universität Erlangen-Nürnberg.

Kingma, D. & Welling M. (2022). Auto-Encoding Variational Bayes. Retrieved September 2023. https://arxiv.org/pdf/1312.6114.pdf

Liu, Z., Luo, P., Wang, X., and Tang, X. (2018). Large-scale celebfaces attributes (celeba) dataset. Retrieved August, 15(2018):11.

Qian, S., Lin, K.-Y., Wu, W., Liu, Y., Wang, Q., Shen, F., Qian, C., and He, R. (2019). Make a face: Towards arbitrary high fidelity face manipulation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10033–10042.

Santa-Cruz, D. & Ebrahimi, Touradj. (2000). An analytical study of JPEG 2000 functionalities. 2. 49 - 52 vol.2. 10.1109/ICIP.2000.899222.

Saxena, Divya, and Jiannong Cao. “Generative Adversarial Networks (GANs).” ACM Computing Surveys, vol. 54, no. 3, June 2021, pp. 1–42, https://doi.org/10.1145/3446374.

Schachner, S., Rathgeb , C., Tapia, J., & Busch, C. (2023). Effect of lossy compression algorithms on face image quality and recognition. https://arxiv.org/pdf/2302.12593

Toledo, R., & Antonelo, E. (n.d.). Face Reconstruction with Variational Autoencoder and Face Masks. Retrieved September 19, 2023. https://arxiv.org/pdf/2112.02139.pdf

Vahdat, A. and Kautz, J. (2020). NVAE: A deep hierarchical variational autoencoder. In Neural Information Processing Systems (NeurIPS).

Wallace, G. K. (1992). The JPEG still picture compression standard. IEEE Transactions on Consumer Electronics, vol. 38, no. 1, pages 18-34. doi: 10.1109/30.125072.

Wang, Z. and Bovik, A. C. (2009). Mean squared error: Love it or leave it? a new look at signal fidelity measures. IEEE signal processing magazine, 26(1):98–117.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4):600–612.

Wang, Z., Simoncelli, E. P., and Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. In The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, volume 2, pages 1398–1402. Ieee.

Zhao, H., Gallo, O., Frosio, I., and Kautz, J. (2016). Loss functions for image restoration with neural networks. IEEE Transactions on computational imaging, 3(1):47–57.

Zhao, H. (2015). Loss Functions for Image Restoration with Neural Networks. arXiv:1511.08861.