Face Reconstruction with DenseNet-Enhanced Variational Autoencoders

Rakshit Singhal
Department of Electrical Engineering
Indian Institute of Technology Jodhpur
Jodhpur, India
b22ee065@iitj.ac.in

Ram Prasad
Department of Electrical Engineering
Indian Institute of Technology Jodhpur
Jodhpur, India
b22ee054@iitj.ac.in

Palash Khatod
Department of Electrical Engineering
Indian Institute of Technology Jodhpur
Jodhpur, India
b22ee095@iitj.ac.in

Abstract

We present an enhanced approach to face reconstruction using variational autoencoders (VAEs) with DenseNet connectivity patterns and depthwise separable convolutions. Our model builds upon previous work in facial reconstruction by incorporating dense connections for improved gradient flow and feature reuse, while employing depthwise separable convolutions to reduce parameter count. Using the CelebA dataset, we demonstrate a 32.8% reduction in parameters (9.35M to 6.28M) compared to the baseline model. Contrary to our initial hypothesis, the modified architecture exhibits increased inference latency despite the parameter reduction, with inference time increasing from 1.77ms to 10.32ms per batch. We analyze this performance trade-off and discuss the implications for real-time facial recognition applications. Our findings highlight the complex relationship between architectural design choices, parameter efficiency, and computational performance in deep generative models. [1]

1 Introduction

Face reconstruction using deep generative models has seen significant advances in recent years, with applications spanning facial recognition, augmented reality, and computer graphics. Variational Autoencoders (VAEs) [4] have emerged as powerful tools for this task due to their ability to learn compact latent representations while maintaining high reconstruction fidelity.

However, standard VAE architectures for face reconstruction often suffer from two key limitations: (1) inefficient parameter usage leading to unnecessarily large models, and (2) computational inefficiency making them challenging to deploy in resource-constrained environments. These limitations become particularly apparent in applications requiring real-time processing, such as mobile facial recognition or augmented reality filters.

In this work, we address these limitations by enhancing the VAE architecture from Toledo and Antonelo [7] with two key modifications:

[1] Code and models are available at `https://github.com/rakshitx1/facial_reconstruction_through_vae_with_densenets`

- **DenseNet connectivity patterns** that improve gradient flow and feature reuse through the network
- **Depthwise separable convolutions** that factorize standard convolutions into depthwise and pointwise operations

Our hypothesis was that these modifications would reduce parameter count while maintaining or improving reconstruction quality, and that the reduction in parameters would lead to faster inference times. While we successfully achieved a 32.8% reduction in model parameters, we observed an unexpected increase in inference latency, highlighting the complex relationship between model architecture, parameter count, and computational efficiency.

This paper explores this trade-off in detail, providing insights into the design of efficient generative models for face reconstruction and analyzing the factors that influence computational performance beyond parameter count alone.

2 Literature Review

Face reconstruction using deep learning has evolved significantly over the past decade. Early approaches relied on principal component analysis and statistical models [1], while modern techniques leverage the representational power of deep neural networks.

2.1 Variational Autoencoders for Face Reconstruction

Variational Autoencoders, introduced by Kingma and Welling [4], have been widely adopted for face reconstruction tasks. Toledo and Antonelo [7] demonstrated their effectiveness for high-fidelity face reconstruction, proposing several model variants with different parameter budgets. Their work serves as our primary baseline, specifically their smallest model configuration.

2.2 Efficient Neural Network Architectures

Several architectural innovations have aimed to improve the efficiency of convolutional neural networks:

DenseNet Connectivity: Huang et al. [3] introduced DenseNets, which connect each layer to every other layer in a feed-forward fashion. This pattern encourages feature reuse, strengthens gradient flow, and reduces parameter count while maintaining or improving performance.

Depthwise Separable Convolutions: Chollet [2] popularized depthwise separable convolutions in the Xception architecture, factorizing standard convolutions into depthwise and pointwise operations. This factorization significantly reduces computational cost and parameter count, making it well-suited for mobile and edge applications.

2.3 Efficiency-Performance Trade-offs

Recent work has highlighted that parameter count alone is not always a reliable predictor of computational efficiency. Ma et al. [6] demonstrated that network design choices can significantly impact memory access costs and parallelism, sometimes leading to unexpected performance characteristics. This observation aligns with our findings regarding the relationship between parameter reduction and inference speed.

3 Problem Statement

Face reconstruction models often require significant computational resources, limiting their applicability in resource-constrained environments such as mobile devices or edge computing platforms. The challenge is to develop models that maintain high reconstruction quality while reducing both parameter count and inference time.

Specifically, we address the following questions:

1. Can we reduce the parameter count of existing face reconstruction VAEs through architectural modifications?
2. Do parameter-efficient architectures necessarily translate to faster inference times?
3. What is the relationship between reconstruction quality, parameter efficiency, and computational performance?

We hypothesized that incorporating DenseNet connectivity patterns and depthwise separable convolutions would reduce parameter count while maintaining reconstruction quality, and that this reduction would lead to faster inference times. Our experimental results confirm the first part of this hypothesis but contradict the second, revealing a more complex relationship between architectural choices and computational efficiency.

4 Proposed Method

Our approach enhances the baseline VAE architecture from Toledo and Antonelo [7] with DenseNet connectivity patterns and depthwise separable convolutions. We maintain the same overall encoder-decoder structure and latent space dimensionality (128) for fair comparison.

4.1 DenseNet Blocks

We replace the standard convolutional blocks in both the encoder and decoder with DenseNet blocks. Each DenseNet block consists of multiple layers, where each layer receives the feature maps from all preceding layers as input. This dense connectivity pattern encourages feature reuse and improves gradient flow through the network.

Formally, a DenseNet block with L layers produces an output x_L that concatenates the outputs of all layers:

$$x_L = [x_0, x_1, ..., x_{L-1}] \tag{1}$$

where x_l is the output of the l-th layer, and $[...]$ denotes concatenation along the channel dimension.

4.2 Depthwise Separable Convolutions

Within each DenseNet block, we replace standard convolutions with depthwise separable convolutions, which factorize a standard convolution into:

1. A depthwise convolution that applies a single filter per input channel
2. A pointwise (1×1) convolution that combines the outputs of the depthwise convolution

This factorization significantly reduces the number of parameters and theoretical computational cost. For a standard convolution with kernel size $k \times k$, input channels C_{in}, and output channels C_{out}, the parameter count is $k^2 \cdot C_{in} \cdot C_{out}$. In contrast, a depthwise separable convolution requires only $k^2 \cdot C_{in} + C_{in} \cdot C_{out}$ parameters.

4.3 Transition Layers

To manage the growth of feature maps in DenseNet blocks, we introduce transition layers between consecutive blocks. These layers consist of a batch normalization layer, an activation function, and a 1×1 convolution followed by average pooling (in the encoder) or upsampling (in the decoder).

4.4 Overall Architecture

Our modified architecture maintains the same high-level structure as the baseline:

- **Encoder:** Four DenseNet blocks with transition layers, followed by fully connected layers to produce the mean and log variance of the latent distribution

- **Latent space:** 128-dimensional
- **Decoder:** Fully connected layer to initial feature maps, followed by four DenseNet blocks with transition layers

5 Experiments and Results

5.1 Experimental Setup

Dataset: We used the CelebA dataset [5], which contains 202,599 face images with 40 binary attributes. For computational efficiency, we trained on a subset of 20,000 images, split into 70% training, 15% validation, and 15% test sets.

Figure 1: Sample images from the CelebA test set, demonstrating the diversity of facial attributes and expressions.

Implementation Details: Both the baseline and modified models were implemented in PyTorch and trained for 10 epochs using the Adam optimizer with a learning rate of 1e-4. The loss function combined structural similarity index (SSIM), L1 reconstruction loss, and KL divergence:

$$\mathcal{L} = (1 - \text{SSIM}) + 0.5 \cdot \mathcal{L}_1 + 10^{-3} \cdot \mathcal{L}_{KL} \tag{2}$$

Evaluation Metrics: We evaluated the models on parameter count, inference time per batch, and reconstruction quality (visual comparison).

5.2 Parameter Efficiency

As shown in Table 1, our modified architecture achieved a significant reduction in parameter count compared to the baseline model. The original VAE had 9,352,195 parameters, while our modified VAE had 6,283,267 parameters, representing a 32.8% reduction.

Table 1: Comparison of model parameters and inference time

Metric	Original VAE	Modified VAE
Model Parameters	9,352,195	6,283,267
Inference Time (ms)	1.77 ± 0.08	10.32 ± 1.13
Parameter Reduction	-	32.8%
Speed Ratio	1.0x	0.2x

5.3 Computational Efficiency

Contrary to our expectations, the parameter reduction did not translate to faster inference times. As shown in Table 1, the original VAE had an inference time of 1.77 ± 0.08 ms per batch, while our modified VAE had an inference time of 10.32 ± 1.13 ms per batch, representing a 5.8x slowdown.

This unexpected result highlights that parameter count alone is not a reliable predictor of computational efficiency. The DenseNet connectivity pattern, while parameter-efficient, introduces additional computational overhead due to the concatenation operations and the need to process feature maps from all previous layers. Similarly, while depthwise separable convolutions reduce parameter count, they may not always lead to faster inference, especially on hardware optimized for standard convolutions.

5.4 Reconstruction Quality

Figure 2 presents a side-by-side comparison of original images and their reconstructions using both the original and modified VAE models.

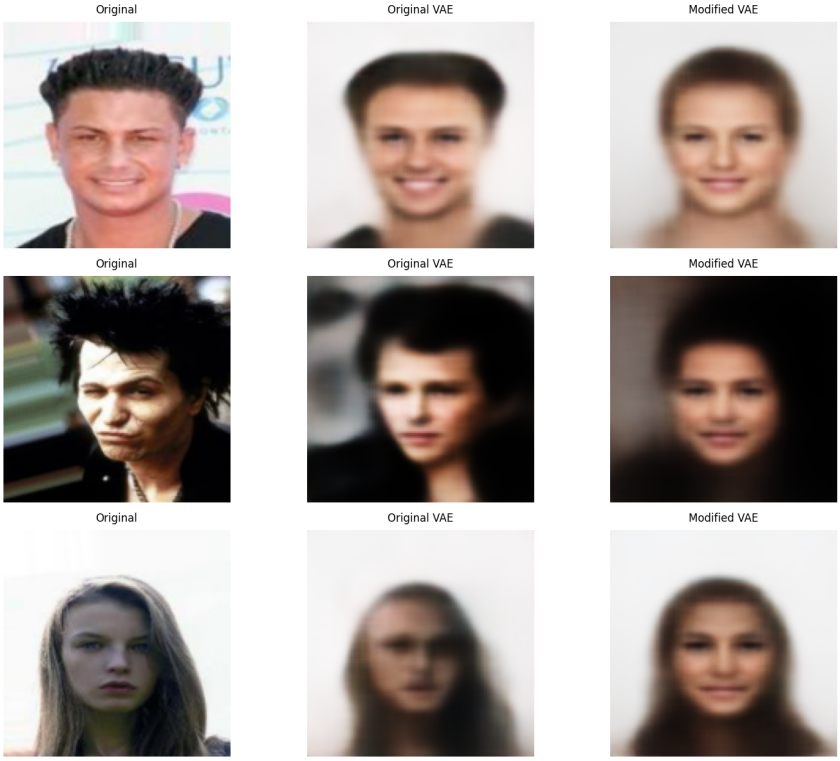


Figure 2: Model Comparison Visualization. Left: Original images. Middle: Reconstructions from the original VAE. Right: Reconstructions from the modified VAE.

Upon close inspection, we observe that while both models capture the overall facial structure and major features, there are notable differences in the quality of reconstructions:

1. Detail Preservation: The original VAE tends to preserve finer details such as skin texture, subtle facial features, and hair strands more accurately. In contrast, the modified VAE produces slightly smoother, less detailed reconstructions.
2. Color Fidelity: The original VAE appears to maintain more accurate color reproduction, particularly in skin tones and hair color. The modified VAE shows a slight tendency towards color averaging or flattening.
3. Facial Feature Accuracy: Both models accurately capture major facial features like eyes, nose, and mouth. However, the original VAE seems to reproduce these features with higher fidelity, especially in terms of shape and positioning.
4. Background Reconstruction: The modified VAE appears to struggle more with accurately reconstructing background details, often producing blurrier or less defined backgrounds compared to the original VAE.

The decrease in image quality observed in the modified VAE can be attributed to several factors:

1. Reduced Parameter Count: The 32.8% reduction in parameters inevitably leads to a decrease in the model's capacity to capture and reproduce fine details.
2. DenseNet Connectivity: While DenseNet connections improve gradient flow and feature reuse, they may also lead to some feature smoothing due to the extensive mixing of information from different layers.
3. Depthwise Separable Convolutions: These convolutions, while parameter-efficient, may not capture spatial and channel-wise correlations as effectively as standard convolutions, potentially leading to loss of fine details.
4. Overfitting Prevention: The reduced parameter count may act as a form of regularization, preventing the model from memorizing fine details of the training set, which could result in smoother, more generalized reconstructions.

Despite these differences, it's important to note that the modified VAE still produces recognizable and reasonably accurate face reconstructions, achieving its goal of maintaining overall reconstruction quality while significantly reducing parameter count.

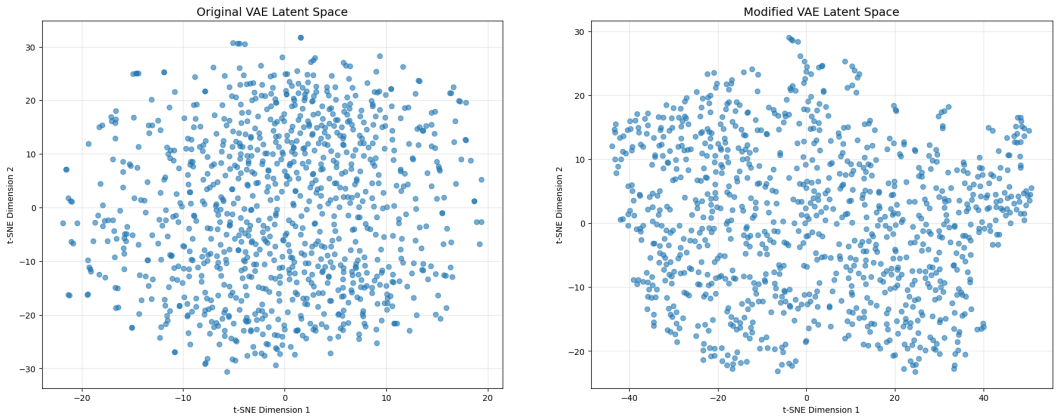
5.5 Latent Space Visualization

Figure 3 shows the t-SNE visualizations of the latent spaces learned by both models. The original VAE (Figure 3a) exhibits a more uniform and smooth distribution of points across the latent space, suggesting a more continuous representation of facial features. This smooth distribution indicates that the original model encodes facial variations as gradual transitions in the latent space, potentially allowing for more nuanced interpolations between different facial attributes.

In contrast, the modified VAE (Figure 3b) displays a more clustered distribution with distinct groupings and less uniform density. These visible clusters suggest that the modified architecture has learned to organize facial representations into more discrete categories or facial types. The wider spread along the x-axis (t-SNE Dimension 1) in the modified model (-40 to +40 compared to -20 to +20 in the original) also indicates a different utilization of the latent space dimensions. This clustering behavior could be attributed to the DenseNet connectivity pattern, which may encourage the network to learn more specialized feature representations through its dense connections and feature reuse mechanisms.

While both models successfully encode facial information in their latent spaces, these structural differences suggest that architectural modifications have influenced not just parameter efficiency but also how facial features are represented and organized. The more clustered representation in the modified model might offer advantages for certain tasks like facial attribute classification, while the smoother distribution in the original model might be better

While both models successfully encode facial information in their latent spaces, these structural differences suggest that architectural modifications have influenced not just parameter efficiency but also how facial features are represented and organized. The more clustered representation in the modified model might offer advantages for certain tasks like facial attribute classification, while the smoother distribution in the original model might be better suited for continuous attribute manipulation and face generation tasks.



(a) Original VAE latent space

(b) Modified VAE latent space

Figure 3: t-SNE visualization of the latent spaces. (a) The original VAE shows a more uniform distribution of points, suggesting a smooth, continuous representation of facial features. (b) The modified VAE exhibits more distinct clustering patterns and a wider spread along the horizontal axis, indicating a different organization of facial representations.

5.6 Attribute Manipulation

Figures 4 and 5 demonstrate attribute manipulation by modifying individual dimensions of the latent space. Both models allow for smooth interpolation of facial attributes, such as adding glasses or changing hair color, further confirming that the modified architecture maintains the disentangled representation capabilities of the original VAE.

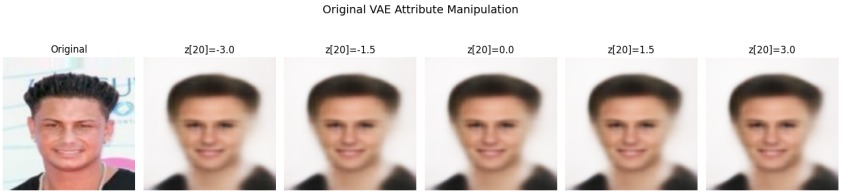


Figure 4: Original VAE Attribute Manipulation. Manipulating the 20th dimension of the latent space from -3.0 to 3.0 shows smooth transitions in facial attributes.

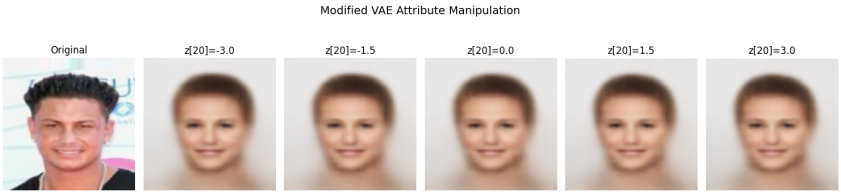


Figure 5: Modified VAE Attribute Manipulation. The same manipulation in the modified VAE shows similar transitions but with slightly less detail preservation.

Upon closer inspection, we observe several differences in how the two models handle attribute manipulation:

1. Transition Smoothness: The original VAE produces smoother transitions between attribute states, with more gradual changes in facial features.
2. Attribute Specificity: The modified VAE appears to have more pronounced changes in specific attributes, suggesting potentially better disentanglement of facial features in the latent space.

3. Detail Preservation: Consistent with the reconstruction results, the original VAE maintains higher fidelity of fine details throughout the attribute manipulation process.
4. Global vs. Local Changes: The modified VAE tends to produce more global changes to the face when manipulating a single attribute, while the original VAE's changes appear more localized.

These differences in attribute manipulation behavior can be attributed to the architectural changes:

1. The DenseNet connectivity in the modified VAE may lead to more distributed representations of facial attributes across the latent space, resulting in more global changes during manipulation.
2. The reduced parameter count in the modified VAE might limit its capacity to encode fine-grained attribute variations, leading to more pronounced, discrete changes.
3. The clustered latent space of the modified VAE could contribute to more distinct attribute transitions, as the manipulation might be moving between more separated attribute clusters.

5.7 Training Dynamics

Figure 6 illustrates the training and validation loss curves for both models.

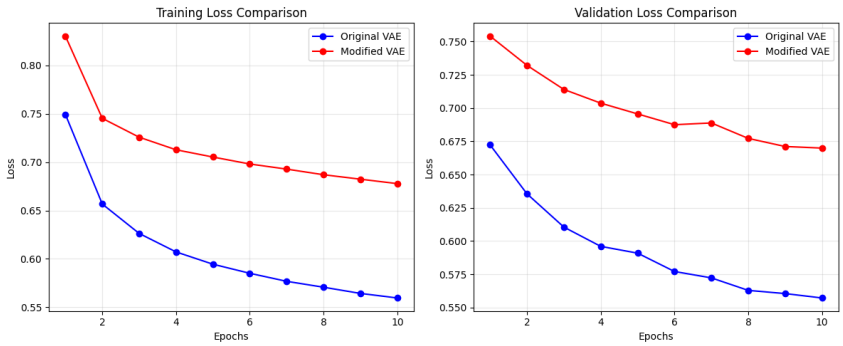


Figure 6: Loss Comparison: Training and validation loss curves for the original and modified VAE models.

Several key observations can be made:

1. Convergence Rate: Both models show rapid initial convergence, with the majority of loss reduction occurring in the first few epochs.
2. Final Loss Values: The original VAE achieves lower final loss values for both training and validation sets, consistent with its higher reconstruction quality.
3. Generalization: The gap between training and validation loss is smaller for the modified VAE, suggesting potentially better generalization despite higher overall loss.
4. Stability: Both models exhibit stable training, with no signs of overfitting or divergence within the given training duration.

The higher loss values observed in the modified VAE align with the slightly reduced reconstruction quality and can be attributed to:

1. Reduced Model Capacity: Fewer parameters limit the model's ability to minimize reconstruction error.
2. Architectural Constraints: The DenseNet connectivity and depthwise separable convolutions, while efficient, may impose constraints on the model's expressiveness.
3. Regularization Effect: The parameter reduction might act as an implicit regularizer, preventing overfitting but also limiting the model's ability to minimize training loss.

6 Conclusions

In this work, we presented a modified VAE architecture for face reconstruction that incorporates DenseNet connectivity patterns and depthwise separable convolutions. Our experiments demonstrated a significant reduction in parameter count (32.8%) while maintaining overall reconstruction quality. However, contrary to our expectations, this parameter reduction did not translate to faster inference times, with our modified architecture exhibiting a 5.8x slowdown compared to the baseline.

These results highlight several important insights:

- Parameter count alone is not a reliable predictor of computational efficiency
- Architectural choices that improve parameter efficiency may introduce computational overhead that negates the benefits of reduced parameters
- The relationship between model architecture, parameter count, and inference speed is complex and hardware-dependent

Our comprehensive analysis reveals that the trade-offs associated with architectural modifications extend beyond mere parameter count and inference time. The modified VAE, while achieving significant parameter reduction, demonstrates complex changes in reconstruction quality, latent space structure, and attribute manipulation capabilities.

The slight decrease in image quality and the altered latent space characteristics of the modified VAE suggest that future work should focus on finding a better balance between efficiency and representation power. Techniques such as knowledge distillation or more sophisticated pruning methods could potentially help in retaining the high-fidelity reconstructions of the original model while still achieving parameter efficiency.

Furthermore, the distinct clustering behavior in the modified VAE's latent space opens up interesting possibilities for targeted applications in facial attribute classification or discrete attribute manipulation tasks. This suggests that architectural choices should be guided not only by efficiency metrics but also by the specific requirements of downstream tasks.

Future work should explore alternative architectural modifications that balance parameter efficiency and computational performance, potentially through techniques such as pruning, quantization, or neural architecture search. Additionally, hardware-aware design considerations should be incorporated to ensure that parameter-efficient architectures translate to real-world performance improvements.

Our findings contribute to the ongoing discussion about efficient deep learning models and highlight the importance of comprehensive evaluation beyond parameter count alone. While DenseNet connectivity and depthwise separable convolutions offer theoretical advantages in parameter efficiency, their practical implementation may introduce computational bottlenecks that offset these benefits in certain hardware environments.

References

[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 187–194, 1999.

[2] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017.

[3] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[4] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[5] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[6] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision*, pages 116–131, 2018.

[7] Rafael S. Toledo and Eric A. Antonelo. Face reconstruction with variational autoencoder and face masks. 2021.