ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10887775

# Compgen: Synthesis and Generation of Faces From Edgemaps

*Ruban Vishnu Pandian*[1*] *Abhiram Rao Gorle*[1*] *Karthick Krishna M*[1*]
*Nambi Seshadri*[1,2] *R. David Koilpillai*[1]

[1]Indian Institute of Technology Madras, Chennai, India
[2]University of California San Diego, USA.

## ABSTRACT

We address the problem of synthesis and generation of faces from edgemaps, motivated by extreme low bit-rate facial compression and the need for robust source-channel coding over noisy channels. Three approaches for image reconstruction are proposed. In the first, a deep learning-based encoder-decoder creates a latent space representation of the original image. An Edgemap-to-Latent Mapper (ELM) network maps the input edgemap to this latent space, with the final image reconstructed using a pre-trained compressive decoder. The second approach retrains the compressive decoder to reconstruct images from the ELM network's output. The third approach jointly trains the ELM network and decoder, enabling direct reconstruction from the edgemap. This end-to-end framework achieves reasonable reconstruction fidelity. We also examine the impact of additive channel noise on edgemap transmission under low SNR conditions, demonstrating that even with significant noise, a DNN-based joint denoiser and edgemap decoder can reconstruct images. At extremely low SNRs, where edgemaps are highly corrupted, the network also exhibits generative capabilities, producing plausible images.

***Index Terms***— Image compression, generation, residual network, edgemap, coding with side information

## 1. INTRODUCTION

Artificial image generation is the process of creating realistic images using machine learning techniques. This idea gained traction with the advent of Variational Autoencoders (VAEs) [1] and Generative Adversarial Networks (GANs) [2]. Since then, numerous models have been proposed to address various aspects of image synthesis, including PixelCNN [3], BigGAN [4], and noise conditional score networks [5]. A major breakthrough occurred with the onset of diffusion models [6] for unsupervised learning, which have also been effectively applied to image generation tasks [7]. The integration of transformer architectures [8] into these models has further advanced the field, culminating in the development of latent diffusion models [9], where transformer blocks are utilized in the reverse diffusion process.

Our work draws inspiration from key ideas in speech synthesis, particularly linear prediction-based speech synthesis and coding [10], diffusion models for image generation, and joint source channel coding (JSCC) techniques for robust transmission over noisy channels. In this paper, we introduce a CNN-based residual network model **Compgen**, designed to perform image compression-cum-generation by leveraging contextual information. In particular, the proposed model takes an image edgemap as input and generates a corresponding high-fidelity output image.

### 1.1. Code-Excited Linear Prediction

In [11], a parametric model for speech synthesis is proposed, wherein the vocal tract is represented as a time-varying linear filter excited by pulses for voiced speech and noise for unvoiced speech. Drawing inspiration from this framework, we conceptualize edgemaps as an analogous excitation source for image synthesis, driving a deep neural network to generate corresponding images, thereby replacing the predictive filter from speech synthesis. While our current approach employs a fixed deep network for image synthesis, future directions could explore spatially adaptive parameterization of the model.

### 1.2. Latent Diffusion Model

Diffusion models operate by progressively adding noise to data and training a reverse diffusion model to denoise it, thereby learning the underlying data distribution. Latent diffusion models extend this principle to latent images instead of original input images but suffer from the complexity of the U-Net architecture and high sampling time in the reverse diffusion process.

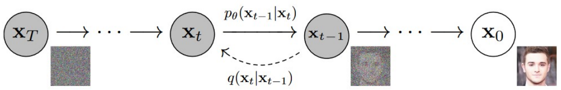

$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$

**Fig. 1**: Diffusion Process

The forward diffusion process is mathematically described as a series of random Gaussian transformations:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I); \quad x_t = x_{t-1} + n_t$$

where $q(.|.)$ denotes the conditional distribution of the forward diffusion process, $x_{t-1}$ is the previous image, $x_t$ is the current image and $\beta_t$ is a variance hyperparameter. It can be observed that the transformation can be modelled as a noise addition transform where $n_t$ is a noise matrix with distribution $\mathcal{N}((\sqrt{1-\beta_t}-1)x_{t-1}, \beta_t I)$. Our model is motivated by this observation, using contextual information

[*] - Equal Contribution

about the target image to generate it from a noisy input. Specifically, the edgemap of the target image is corrupted with significant noise, and the model synthesizes the output image using this noise-edgemap input.

### 1.3. Source and Channel Coding

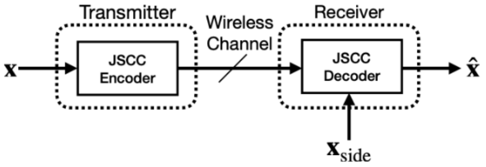


**Fig. 2**: Joint Source-Channel Coding With Side Information

Our work additionally draws inspiration from image transmission over noisy channels, and in particular the problem of combined source and channel coding. While we show noisy input results in generative capability at larger noise powers, we also observe that the same architecture is capable of reconstructing the original image with reasonable fidelity at lower noise powers and hence our approach could be suitable for low bit rate image compression using edgemaps [12] that are derived from the source for transmission over noisy channels [13], [14].

## 2. APPROACH

### 2.1. Problem Formulation

We consider the previously described problem in the presence of additive white Gaussian noise (AWGN), where the input is the edgemap of our original image. Let $x$ denote our original image. Our problem can be presented as: a) Considering the edgemap $z$ of $x$, b) Corrupting this with noise $n$, c) Inputting this into the **Compgen** pipeline and producing the reconstructed $\hat{x}$. In our work, we define PSNR as the following:

$$\text{PSNR} = \max_i 10 \log_{10} \frac{z_i^2}{\sigma_n^2}$$

where $z_i$ is the intensity of the edgemap $z$ at pixel location $i$ and $\sigma_n^2$ is the variance of AWGN noise $n$.

### 2.2. Compression Model

Images have a lot of correlated information within them and are hence typically compressed for efficient storage and processing. In recent years, DNN-based autoencoder networks have been widely used for data compression [15]. Our model architecture builds upon the architecture from [16]. Slight adjustments are made to this architecture to obtain the image compression module shown in Fig. 3.

Blocks with text of the form **[k(x) n(y) s(z)]** denote a convolutional layer with kernel size '$x$', number of output channels '$y$' and stride '$s$'. The *ReLU* and *Sigmoid* are the usual activation functions. The number of residual blocks used is a hyperparameter we call **num_res**. Similarly, the number of multi-residual blocks present in a residual network block is another hyperparameter **num_mult**.

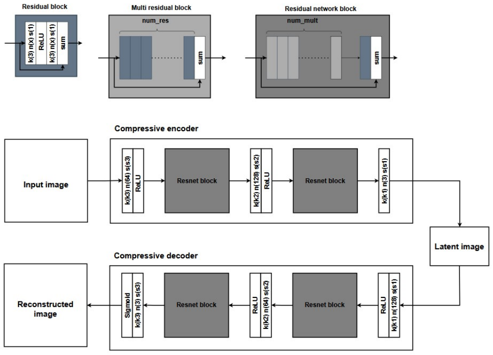


**Fig. 3**: Compression Model Architecture

The compressive encoder-decoder pair is trained to produce a meaningful latent representation of the image (represented as $y$).

### 2.3. Edgemap-to-Latent Mapper (ELM)

The Edgemap-to-Latent Mapper considers the noisy input edgemap and produces the reconstructed latent image. The ELM model is as shown in Fig. 4).

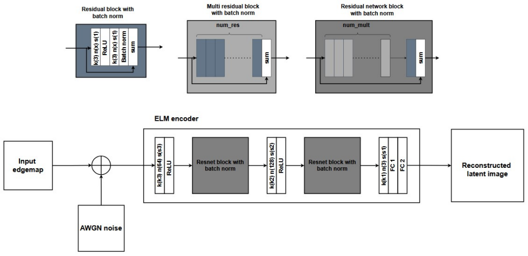

Residual block with batch norm
k(3) n(x) s(1)
ReLU
k(3) n(x) s(1)
Batch norm
sum
Multi residual block with batch norm
num_res
sum
Residual network block with batch norm
num_mult
sum
ELM encoder
Input edgemap
AWGN noise
k(k3) n(64) s(s3)
ReLU
Resnet block with batch norm
k(k2) n(128) s(s2)
ReLU
Resnet block with batch norm
k(k1) n(3) s(s1)
FC 1
FC 2
Reconstructed latent image

**Fig. 4**: ELM Model Architecture

A batch normalization layer is incorporated into the residual building block to enhance the generative model's ability to capture the dataset's structure and generalize effectively to test data. Additionally, fully connected layers, denoted as FC1 and FC2 are included to improve the edgemap-to-latent mapping capability of the model. With the compression and ELM models in place, the overall compression (cum-generator) model can be assembled.

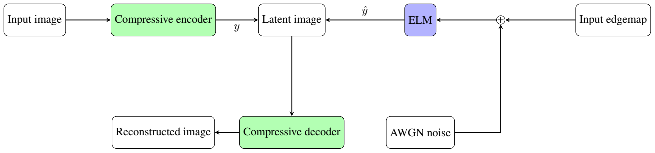


**Fig. 5**: Compgen Pipeline

The compressive encoder and decoder serve as the source encoder and decoder, respectively, while the ELM model functions as a denoiser. It processes noisy edgemaps to produce latent space representations for the pre-trained decoder, approximating the true output $y$ with $\hat{y}$. The complete pipeline, **Compgen**, is illustrated in Fig. 5, with the relevant notations for the training outline provided in Table 1.

**Table 1**: Notations

| Description | Notation |
|---|---|
| Input image | $x$ |
| Reconstructed image | $\hat{x}$ |
| Latent image by compressive encoder | $y$ |
| Latent image by ELM model | $\hat{y}$ |
| Distortion measure | $d(x, \hat{x})$ |
| Input sketch/edgemap | $z$ |
| AWGN noise matrix | $n$ |
| ELM model function | $h(.)$ |
| Compressive encoder function | $f(.)$ |
| Compressive decoder function | $g(.)$ |

## 3. TRAINING OUTLINE

### 3.1. Step 1: Compression Model Training

In step 1, the compression encoder and decoder are jointly trained to obtain a meaningful latent space. Residual blocks with a kernel size of 3 and stride 1 are used to maintain consistent input-output dimensions for addition at the end. After training, the parameters of the compression model are frozen and used as pre-trained blocks in subsequent steps. The training process minimizes a distortion measure between input and reconstructed images, conducted over 50 epochs with a learning rate of 1e-3, a batch size of 100, and the ADAM optimizer. The distortion measure is:

$$y = f(x), \;\; \hat{x} = g(y), \;\; d(x, \hat{x}) = 0.5(1 - \text{SSIM}(x, \hat{x}))$$

where SSIM is the commonly used *structural similarity index*. In step 2, the ELM model is trained to map the input sketch to the latent space constructed in the previous step.

### 3.2. Approach 1: Relying Solely on ELM Training

In this approach, the ELM model is trained to reconstruct the latent image similar to the one produced by the pre-trained compressive encoder from step 1. To do so, it is trained by minimizing the following loss function:

$$y = f(x), \;\; \hat{y} = h(z + n), \;\; d(y, \hat{y}) = \text{MSE}(y, \hat{y})$$

where $y$ is the latent image produced by the compressive encoder, $\hat{y}$ is the latent image produced by the ELM model and MSE is the mean squared error loss. The output of the ELM model is passed through the pre-trained compressive decoder from step 1. Training is done for 40 epochs with a learning rate of 1e-3, batch size of 100 with the ADAM optimizer.

### 3.3. Approach 2: Retraining the Compressive Decoder

The reconstructed latent space in 3.2 may not be perfect. To address this, our second approach extends the first by initially freezing the parameters of the ELM model from step 2. In a subsequent step (say step 3), the compressive decoder is retrained to improve the reconstruction of the original image. The output from the ELM model is then passed through the retrained compressive decoder from step 3 to obtain the final reconstructed image.

### 3.4. Approach 3: End-to-end Joint Optimization

In this approach, the goal is to directly synthesize the reconstructed image from the noisy input edgemap through end-to-end training of the combined decoder (or synthesizer), which integrates the ELM model and compressive decoder from the generation pipeline. This single-step process eliminates the need for a distinct delineation between the ELM model and decoder.

In the first approach, the onus of producing a meaningful latent image rested solely on the ELM model. In contrast, this variation passes the latent image produced by the ELM model through the compressive decoder to produce (generate) the final image.

$$y = f(x), \;\; \hat{y} = h(z + n), \;\; \hat{x} = g(\hat{y})$$

Similar to step 1, an SSIM-based distortion measure is used as the distortion measure. However, the compressive decoder is now updated alongside so basically a joint update of both the ELM model and the compressive decoder components is done. Training is done for 40 epochs with a learning rate of 1e-3, batch size of 100 with the ADAM optimizer.

## 4. EXPERIMENTS AND RESULTS

We use the aligned and cropped images of the CelebA [17] dataset for training. Along with this dataset, the SKSF [18] face sketch dataset is later used for inference. In step 1 of training, the compression model is trained till convergence and the final compression model was able to achieve a mean SSIM score of about 93-94%. This model serves as the pre-trained compressive model in the all approaches.

### 4.1. Results Using Approach 1

The ELM model is trained to reconstruct the latent image as explained in the training outline. The results are shown in Fig. 6 and in Fig. 8 with the following order (left to right): original image, noisy edgemap input, latent image given by compressive encoder, latent image reconstructed by ELM mapper, final reconstructed image.

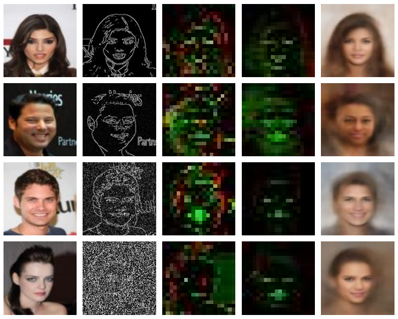


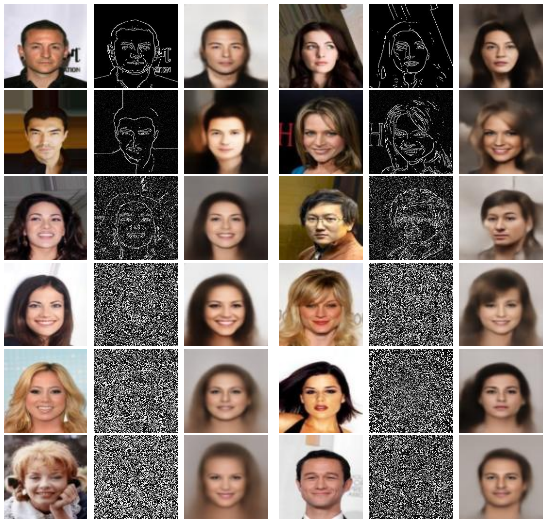
**Fig. 6**: Results for First Approach at PSNRs = 50, 25, 10, 0 dB

### 4.2. Results Using Approach 2

The compressive decoder is retrained to yield a better reconstruction of the original image as discussed in the training outline. The results are shown in Fig. 7a and in Fig. 8 with the following order (left to right): Input image, noisy edgemap, reconstructed final image.

### 4.3. Results Using Approach 3

In this step, both the ELM model and the compressive decoder are jointly updated to reconstruct the final image as explained in the training outline. The results at different PSNRs are shown in Fig. 7b and in Fig. 8 with the following order (left to right): Input image, noisy edgemap, reconstructed final image.



(a) Results for the Second Approach

(b) Results for the Third Approach

**Fig. 7**: Comparison of Results for the Second and Third Approaches at PSNRs = 50, 25, 10, 0, -2, -5 dB

It is to be noted that a power normalized edgemap-noise mixture is used in all the cases. Consider $z_{\text{noisy}} = z + n$ to be the noisy input edgemap and $\sigma_n^2$ to be the noise power. At high PSNRs, $z$ would dominate over $n$ and hence, the scale of the values of $z_{\text{noisy}}$ would be $[0, 1]$ since $z$ operates in that range. However, at low PSNRs, to maintain the same power level, we therefore modify $z_{\text{noisy}}$ as $z_{\text{noisy}} = \frac{z+n}{\sigma_n}$.

### 4.4. On Generative Capabilities

We clearly see that the models described in approach 2 and 3 demonstrate superior performance and are also able to operate convincingly at negative PSNRs. Consequently, we select approach 3 as our baseline model for conditional generation, setting the PSNR to -5 dB for all subsequent results. We also test the model on different types of hand-drawn sketches instead of our original edgemaps. Further experiments, including image generation from hand-drawn sketches and unconditional generation, are also conducted.

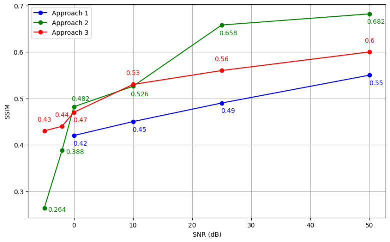
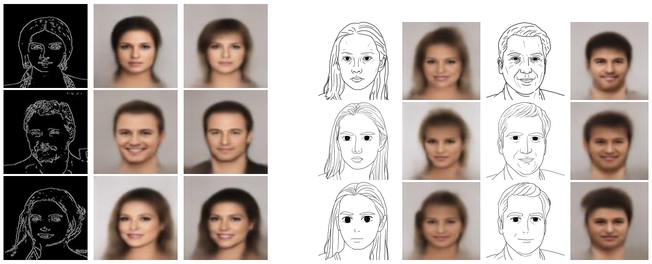


**Fig. 8**: Comparative Performance of all Approaches



(a) Face Generation From Edgemaps

(b) Face Generation From Different Sketch Styles

**Fig. 9**: Comparative Results of Face Generation Approaches

### 4.5. Unconditional Generation Results

To evaluate the generative capabilities of our model at low SNR, we tested it with pure noise inputs, excluding any edgemap information. The results, shown in 10, while less perceptually satisfying compared to edgemap-aided outputs, demonstrate the model's ability to generate novel faces from pure noise realizations.

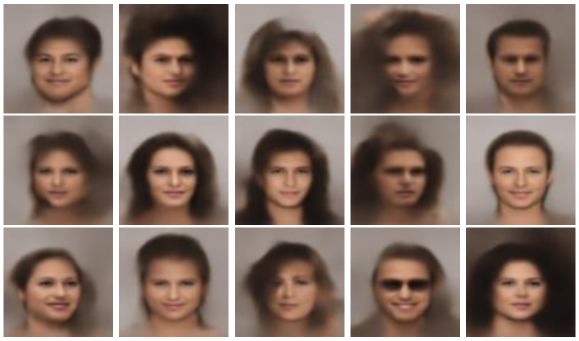


**Fig. 10**: Unconditional Generation

## 5. CONCLUSION

In this work, we present an edgemap-aided image compression and generation model, drawing inspiration from classical information theory and latent diffusion models. Our experiments demonstrate both the reconstructive and generative capabilities of the model. Notably, despite being trained exclusively on machine-generated edgemaps, the model performs effectively on hand-drawn sketches of varying styles, showcasing its versatility.

A key objective of this study was to evaluate the model's ability to accurately reconstruct original images from their corresponding edgemaps. The highest average SSIM score achieved was approximately 70%, suggesting scope for further improvement. Future work will focus on enhancing reconstruction performance by incorporating additional features such as color and effectively utlizing the residual image. Moreover, we plan to develop a deep joint source-channel coding (JSCC) model that processes edgemaps through a joint source-channel encoder, introduces noise, and subsequently utilizes the pre-trained decoder described in this paper.

## 6. REFERENCES

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

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[3] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al., "Conditional image generation with pixelcnn decoders," *Advances in neural information processing systems*, vol. 29, 2016.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.

[5] Yang Song and Stefano Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in neural information processing systems*, vol. 32, 2019.

[6] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. PMLR, 2015, pp. 2256–2265.

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10684–10695.

[10] B. Atal and S L Hanaver, "Speech analysis and synthesis by linear prediction of the speech wave.," *The Journal of the Acoustical Society of America*, vol. 50 2, pp. 637–55, 1971.

[11] M. Schroeder and B. Atal, "Code-excited linear prediction(celp): High-quality speech at very low bit rates," in *ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1985, vol. 10, pp. 937–940.

[12] Eric Lei, Yiğit Berkay Uslu, Hamed Hassani, and Shirin Saeedi Bidokhti, "Text + sketch: Image compression at ultra low rates," 2023.

[13] Eirina Bourtsoulatze, David Burth Kurka, and Deniz Gunduz, "Deep joint source-channel coding for wireless image transmission," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 567–579, Sept. 2019.

[14] Selim F. Yilmaz, Ezgi Ozyilkan, Deniz Gunduz, and Elza Erkip, "Distributed deep joint source-channel coding with decoder-only side information," 2024.

[15] Sonain Jamil, Md Jalil Piran, MuhibUr Rahman, and Oh-Jin Kwon, "Learning-driven lossy image compression: A comprehensive survey," *Engineering Applications of Artificial Intelligence*, vol. 123, pp. 106361, 2023.

[16] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool, "Conditional probability models for deep image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4394–4402.

[17] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[18] Kwan Yun, Kwanggyoon Seo, Chang Wook Seo, Soyeon Yoon, Seongcheol Kim, Soohyun Ji, Amirsaman Ashtari, and Junyong Noh, "Stylized face sketch extraction via generative prior with limited data," in *Computer Graphics Forum*. Wiley Online Library, 2024, p. e15045.