12.10.24

# Generating Images from Audio

Sai Vuruma

CSCE883 Machine Learning

# Index

# Introduction

- Video generation from audio inputs is a cutting-edge research field that has applications in various domains from movies to education.
- It is said that most of the information that we acquire is visual – through the eyes.

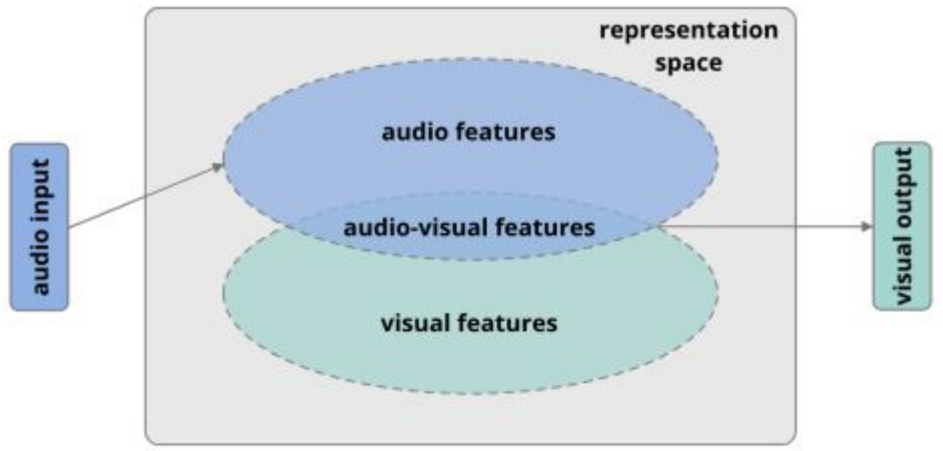



*Figure 1.* Relationship between audio and visual features.

# Literature

- Proposed in 2021 by Zelaszczyk et al.
- This network takes MFCC features from audio data and generates videos from that data.
- The use of discriminator network has improved the performance of the generator.

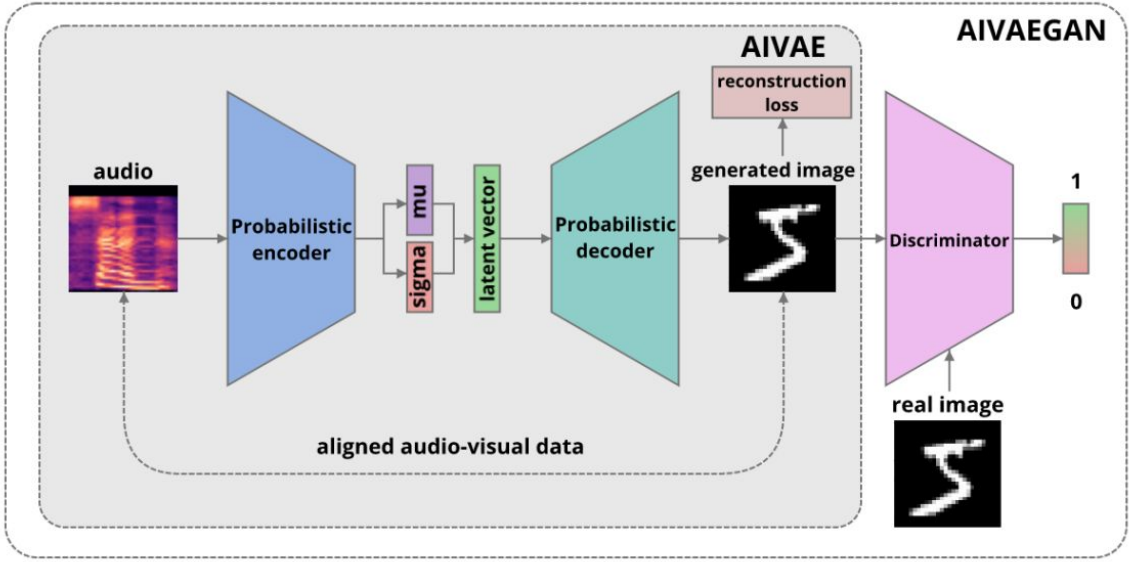



*Figure 2.* AIVAE and AIVAEGAN architectures.

# Literature

- Proposed in 2024 by Xing et al.
- The network leverages ImageBind to create a shared latent space.
- This latent space representation is used to establish connections between multiple generative models that usually target just one modality.

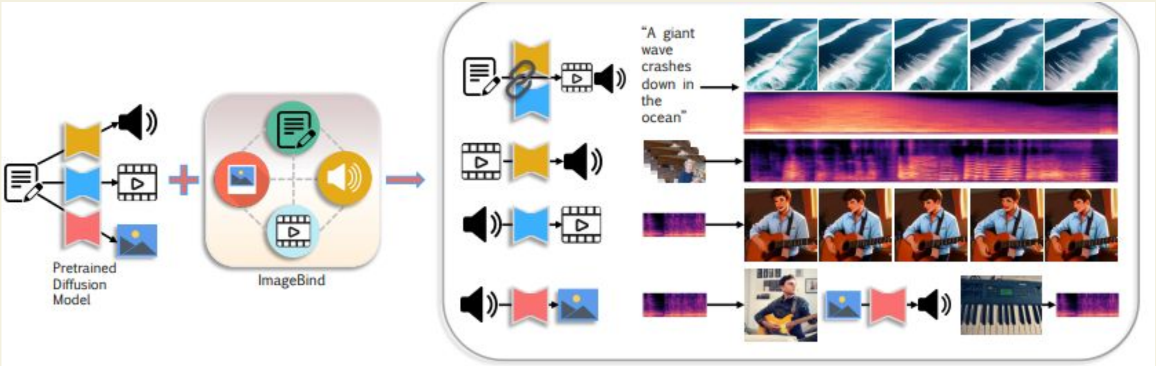
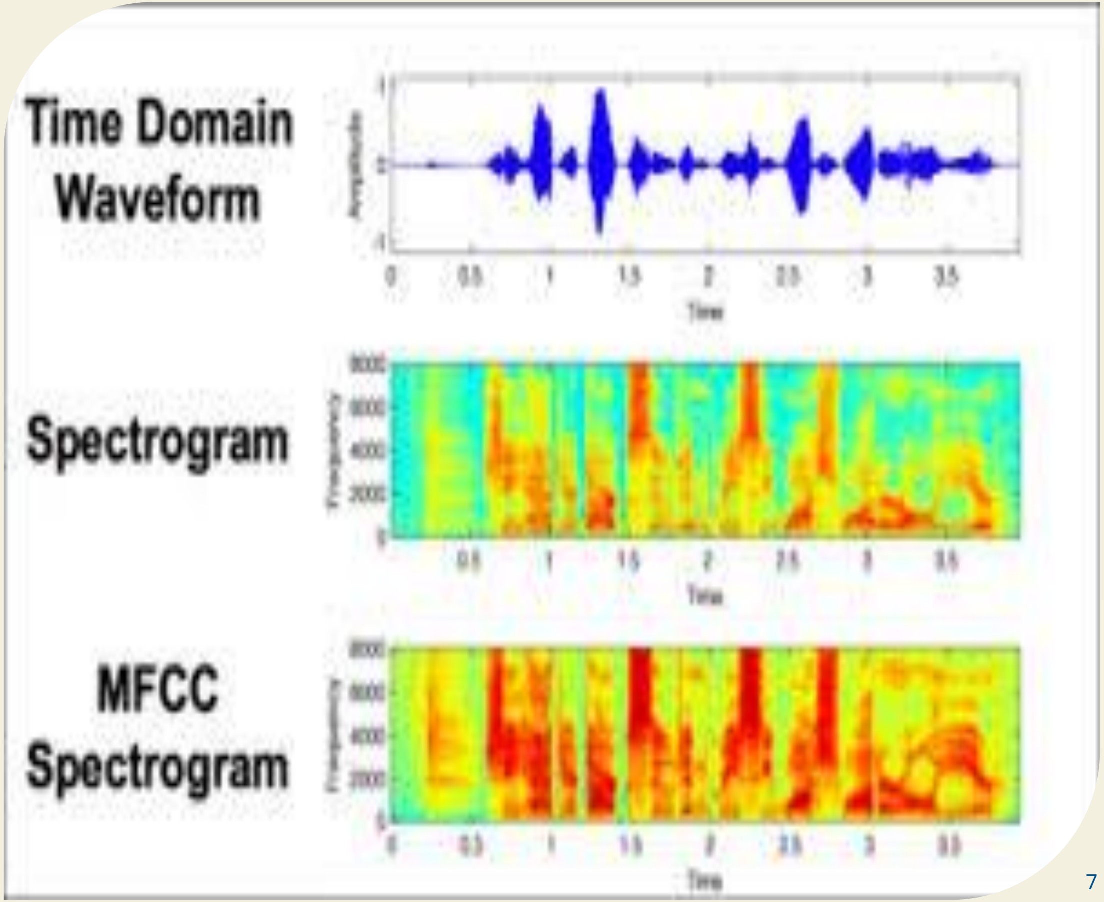



Figure 1. **Overview.** Our approach is versatile and can tackle four tasks: joint video-audio generation (Joint-VA), video-to-audio (V2A), audio-to-video (A2V), and image-to-audio (I2A). By leveraging a multimodal binder, e.g., pretrained ImageBind, we establish a connection between isolated generative models that are designed for generating a single modality. This enables us to achieve both bidirectional conditional and joint video/audio generation.

# Can we directly convert audio signals to images without the text intermediate?

# Mel-frequency Cepstral Coefficients

- It is a way to represent the short-term power spectrum of a sound which helps machines understand and process human speech more effectively.
- MFCCs quantify the self-similarity of the high-pass filtered signal at different time scales (musical pitch removed, robust to bandwidth reduction).



# MM Scene Classification

- Consists of 17K 256*256 images from 8 different environments.
- 37 video sources were processed and sliced into frames every second.
- Corresponding audio is also sliced and the MFCC features are extracted.

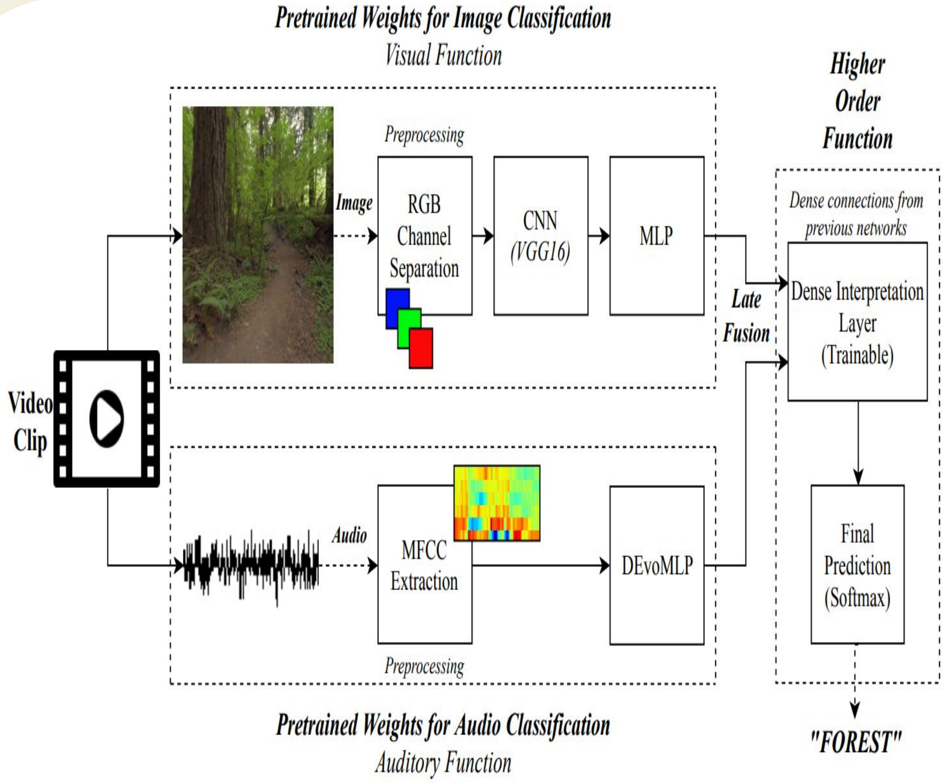



Fig. 2: Overview of the multi-modality network. Pre-trained networks without softmax activation layer take synchronis images and audio segments as input, and classify based on interpretations of the outputs of the two models.


Fig. 3: Example of extracted data from a five second timeline. Each second, a frame is extracted from the video along with the accompanying second of audio.

# Data Cleaning: Audio

- Signal data is already available in MFCC. No feature extraction needed.
- High quality, no missing values.
- Normalized before passing to the network.
- 104 MFCCs available for each row (audio-image pair).

| | mfcc_1 | mfcc_2 | mfcc_3 | mfcc_4 | mfcc_5 | mfcc_6 | mfcc_7 | mfcc_ |
|---|---|---|---|---|---|---|---|---|
| nages/be | 15.078205 | 2.3052775 | -4.19941 | -20.0551 | 14.721057 | -30.1465 | 20.333356 | -8.3 |
| images/be | 15.753304 | 1.1929291 | -3.3422 | -12.3244 | 17.318377 | -33.9805 | 17.502614 | -11.6 |
| images/be | 14.547205 | -6.23738 | -3.99396 | -18.1746 | 2.6801125 | -22.081 | 5.0624365 | -13.8 |
| images/be | 13.368157 | -5.27076 | -3.61985 | -12.9804 | 5.4177610 | -28.4173 | 8.2326021 | -17.0 |
| images/be | 15.029634 | -0.64473 | -9.13927 | -21.3585 | 8.258333 | -36.1396 | 19.831567 | -9.69 |
| images/be | 16.481006 | -8.19752 | -8.60729 | -18.2895 | -8.28646 | -42.0258 | -16.3422 | -29.4 |
| images/be | 16.121613 | 0.2376866 | -5.16319 | -10.2325 | 22.852460 | -31.1039 | 23.971666 | -11.4 |
| images/be | 14.884372 | -2.01201 | -4.72965 | -15.7121 | 16.725091 | -31.2331 | 9.5733917 | -15. |
| images/be | 13.472135 | -4.81872 | -0.68176 | -12.0563 | 12.731065 | -16.444 | 11.600003 | -6.33 |
| images/be | 13.940906 | -1.54964 | 2.1928959 | -10.1544 | 8.0735146 | -20.8231 | 3.8054374 | -3.53 |
| images/be | 13.474925 | 1.3711314 | -2.35127 | -11.9984 | 11.570591 | -23.1967 | 14.568225 | -19.1 |
| images/be | 14.013072 | -1.61105 | 0.4222981 | -13 | -1.83068 | -25.3943 | 2.0042004 | -16.3 |
| images/be | 16.575443 | -6.66466 | -1.46552 | -7.31345 | 1.7883212 | -16.3092 | 2.7226581 | -8.01 |
| images/be | 14.942908 | 0.480329 | -8.52036 | -20.4603 | 19.509018 | -42.0803 | 11.513085 | -32.3 |
| images/be | 15.821529 | -8.20423 | -9.06703 | -19.2314 | -4.42991 | -24.3391 | -1.27796 | -14.2 |
| images/be | 16.057667 | -5.66334 | -2.3814 | -9.07029 | 7.6130163 | -19.6383 | 6.751176 | -10.7 |
| images/be | 16.502019 | -3.68365 | -1.33799 | -4.82326 | 0.3142908 | -17.7198 | 11.930281 | -3.09 |
| images/be | 16.061906 | -2.28078 | -8.07023 | -19.8597 | 7.0995502 | -35.7178 | 13.225196 | -20.9 |
| images/be | 16.031937 | -3.51665 | -4.80891 | -11.5375 | 12.745104 | -30.9441 | 8.1416728 | -19.5 |
| images/be | 15.604706 | -2.27774 | -12.3306 | -22.3806 | 20.526177 | -39.2241 | 20.705806 | -13 |
| images/be | 15.730225 | -4.68477 | -0.69719 | -10.1527 | 7.9970620 | -9.17936 | 15.082384 | |

# Data Cleaning: Images

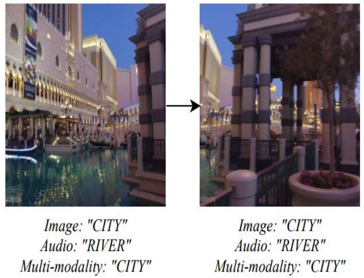



Fig. 9: An example of confusion of the audio model, which is corrected through multi-modality. In both examples, the audio of a City is incorrectly classified as the “RIVER” environment due to the sounds of a fountain and flowing water by the audio classification network.

**EXCLUDED**

Images are very diverse, high variance between each frame.


Fig. 8: An example of confusion of the vision model, which is corrected through multi-modality. In the second frame, the image of hair is incorrectly classified as the “FOREST” environment through Computer Vision.

**EXCLUDED**

Images are very diverse, high variance between each frame.

**EXCLUDED**

Lot of people present in the image, model is unable to learn the basics like sand, sea, etc.

# Data Cleaning: Images

INCLUDED

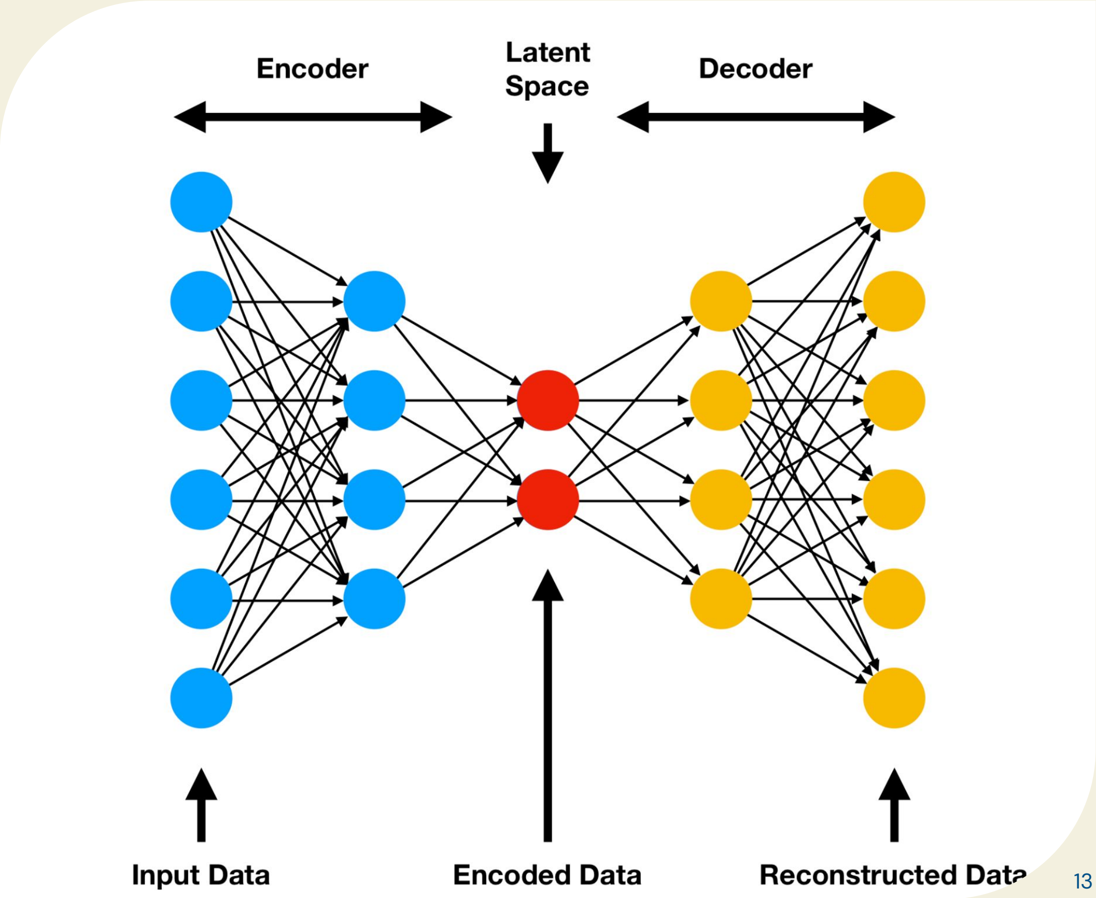
INCLUDED

INCLUDED

Images from the ‘BEACH’ class. Very consistent from frame to frame, limited noise in the images.

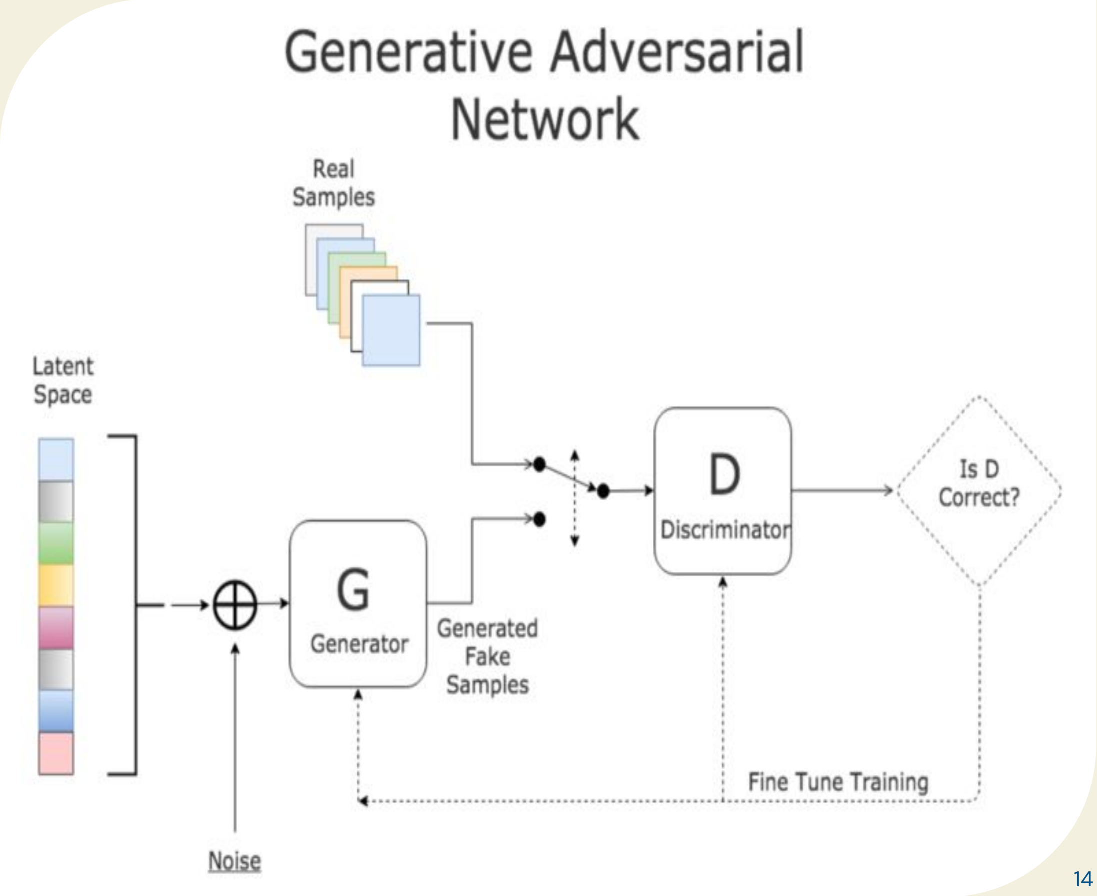
# Autoencoder

- Two components: encoder and decoder.
- Encoder compresses input data into a latent space representation.
- Decoder ingests that representation and tries to reconstruct the original data.
-



# Generative Adversarial Networks

- Two components: generator, discriminator.
- Generator takes random noise as input and produces fake samples.
- Discriminator ingests fake samples and must determine if they are real or fake.
- Both networks work against each other adversarially.



# Encoder

- The encoder takes the mfcc features from the audio input and processes it into the latent space.
- Fully connected layers + Leaky ReLU to get a good latent rep.
- The reparameterization step is excluded from the encoder architecture.

```
self.encoder = nn.Sequential(
    nn.Linear(input_dim, 1024),
    nn.LeakyReLU(0.2),
    nn.Linear(1024, 768),
    nn.LeakyReLU(0.2),
    nn.Linear(768, 512),
    nn.LeakyReLU(0.2),
    nn.Linear(512, 256),
    nn.LeakyReLU(0.2),
    nn.Linear(256, latent_dim),
    nn.LeakyReLU(0.2),
```

# Decoder (Generator)

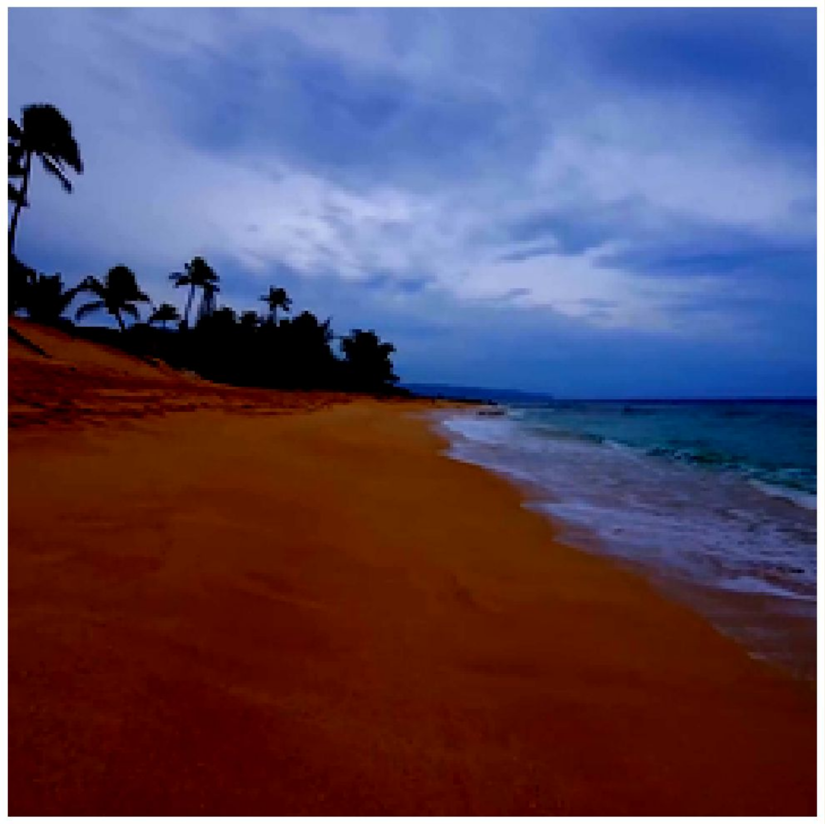
- The generator takes the latent representation from the encoder and produces images.
- Using conv transpose, the linear output from the encoder is upsampled to match the target image size.
- Using Tanh activation on the output layer to get data in [-1, 1].

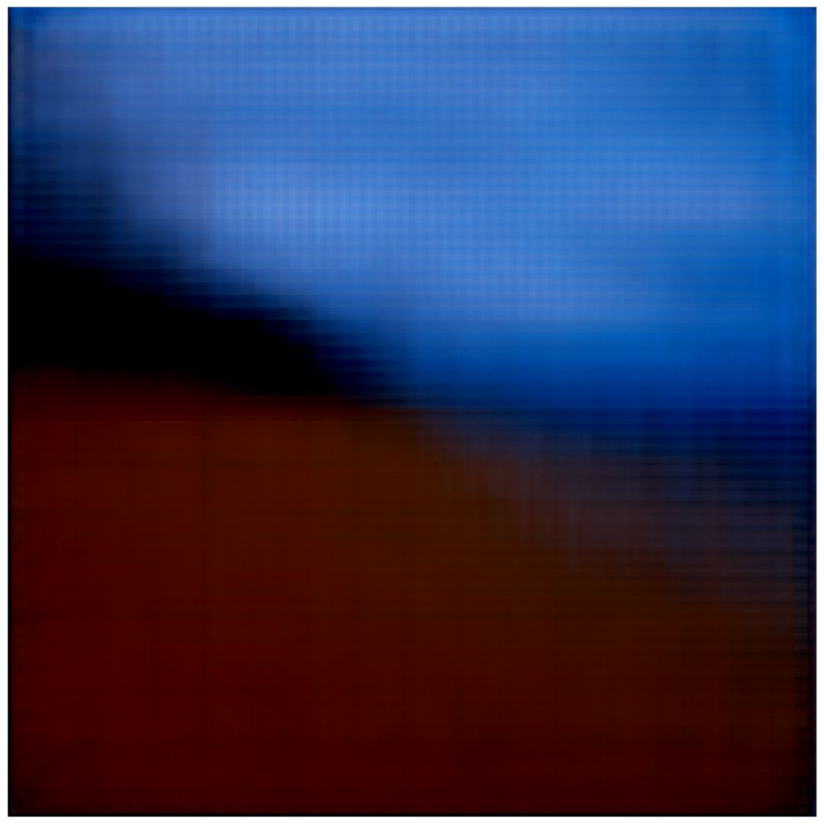
```
coder
elf.decoder = nn.Sequential(
    nn.ConvTranspose2d(latent_dim, 256, kernel_size=4, stride=2, padding=1),
    nn.BatchNorm2d(256),
    nn.LeakyReLU(0.2),
    nn.ConvTranspose2d(256, 128, kernel_size=4, stride=2, padding=1),
    nn.BatchNorm2d(128),
    nn.LeakyReLU(0.2),
    nn.ConvTranspose2d(128, 64, kernel_size=4, stride=2, padding=1),
    nn.BatchNorm2d(64),
    nn.LeakyReLU(0.2),
    nn.ConvTranspose2d(64, 32, kernel_size=4, stride=2, padding=1),
    nn.BatchNorm2d(32),
    nn.LeakyReLU(0.2),
    nn.ConvTranspose2d(32, 16, kernel_size=4, stride=2, padding=1),
    nn.BatchNorm2d(16),
    nn.LeakyReLU(0.2),
    nn.ConvTranspose2d(16, 8, kernel_size=4, stride=2, padding=1),
    nn.BatchNorm2d(8),
    nn.LeakyReLU(0.2),
    nn.ConvTranspose2d(8, 4, kernel_size=4, stride=2, padding=1),
    nn.BatchNorm2d(4),
    nn.LeakyReLU(0.2),
    nn.ConvTranspose2d(4, channels, kernel_size=4, stride=2, padding=1),
    nn.Tanh(),
)
```

# Training

- The model receives the mfcc features of the audio signal as input. The generator generates images as output. The images from the database are used as the ground truth for comparison.
- Loss function = MSE * batch_size
- Train data = 360 audio-image pairs
- Test data = 120 audio-image pairs

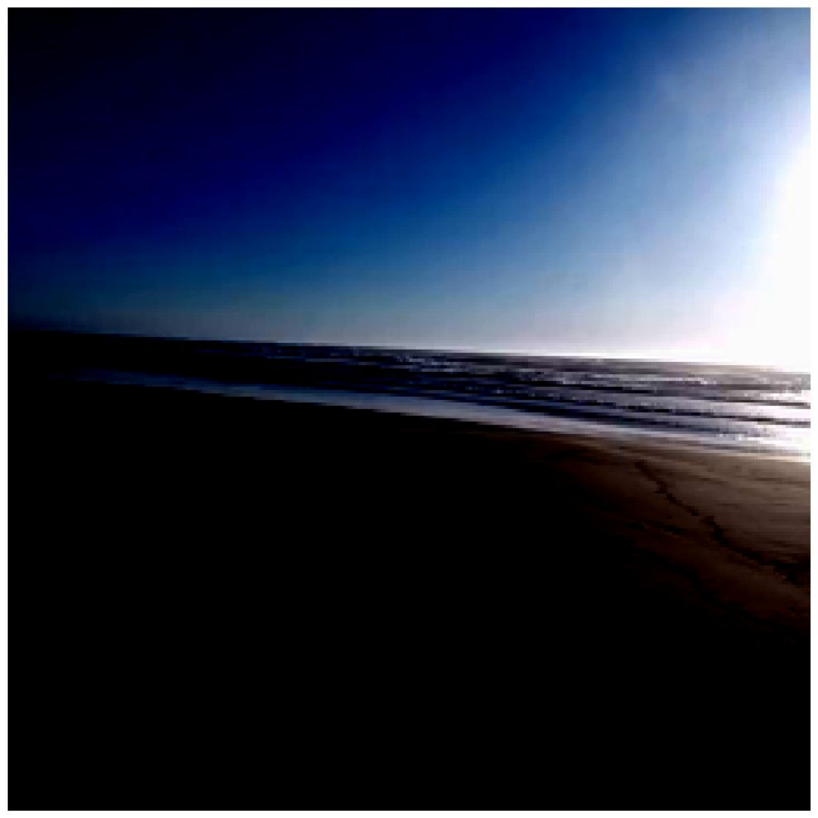
```
def forward(self, x):
    #mu, logvar = self.encode(x)
    #z = self.reparameterize(mu, logvar)
    #z = z.view(z.size(0), -1, 1, 1)

    x = self.encode(x)
    x = x.view(x.size(0), -1, 1, 1)
    x = self.decode(x)

    return x
```

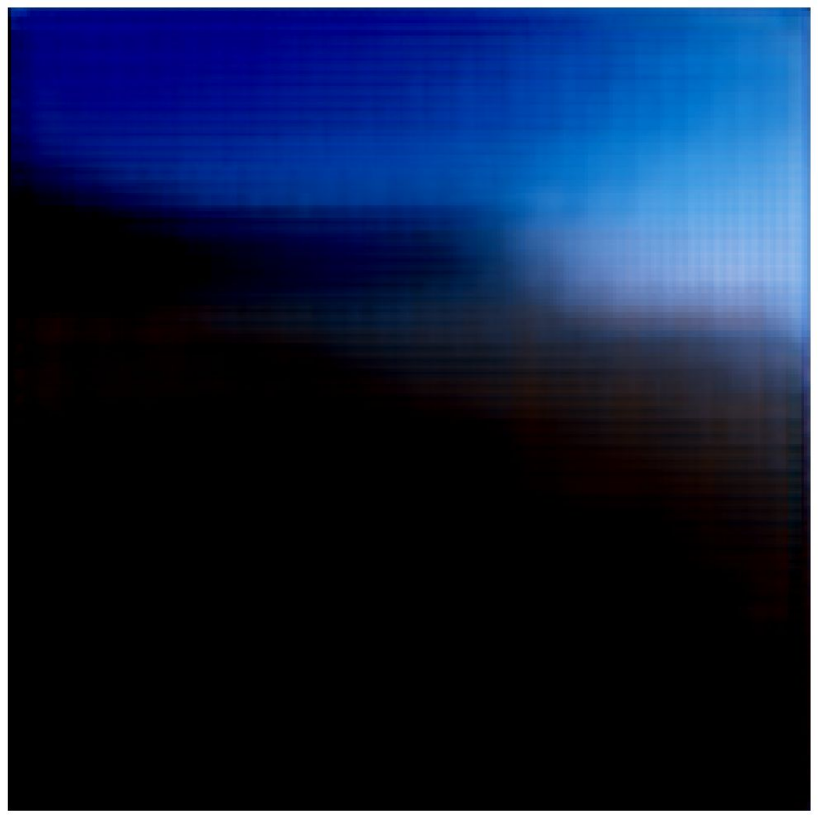




# Results: Target vs Generated

# Results: Target vs Generated





# Results: Target vs Generated

# Evaluation Metrics

Test Loss = 4.691

- MSE*batch_size
- Slightly higher than train_loss of 2.838.
- Indicates reasonable performance and generalization.

SSIM = 0.463

- Structural Similarity Index
- Compares luminance, contrast, texture and structure between two images.
- More aligned with human perception.
- -1 if dissimilar and 1 if identical.

HSS = 0.999

- Histogram Similarity Score
- The correlation between two histograms given by the Pearson coefficient.
- -1 if they are dissimilar i.e. inverse correlation and 1 if they are identical i.e. perfect correlation.

# Future Work

## Video generation

Process sequential audio input to generate video frames.

## Loss function

Include reconstruction loss, similarity scores and frame-to-frame cohesion.

## Dataset

Experiment with the AVSpeech to identify relationships between speaker's voice and appearance.

## Architecture

Include discriminator to improve the quality of generation.

# References


Bird, Jordan J., et al. "Look and listen: A multi-modality late fusion approach to scene classification for autonomous machines." 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020.

https://www.kaggle.com/datasets/birdy654/scene-classification-images-and-audio

Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems 27 (2014).

https://www.geeksforgeeks.org/mel-frequency-cepstral-coefficients-mfcc-for-speech-recognition/

https://medium.com/@derutycsl/intuitive-understanding-of-mfccs-836d36a1f779

https://www.ibm.com/think/topics/encoder-decoder-model

# Thank you!