Generating Images from Audio Input

Sai Krishna Revanth Vuruma
University of South Carolina
svuruma@email.sc.edu

***Abstract*—Generating images from audio input is a growing research field with applications in education, entertainment and many more areas. Most state-of-the-art research methods use an intermediate text representation when converting data from audio to image or video. I built a network that leverages the encoder architecture from Autoencoder to convert the input audio signal into a latent space representation. This representation is then fed to a generator network, instead of a decoder network, which then upsamples it into an image. The network was trained on audio-image pairs taken from a Multi-modal Scene Classification dataset and evaluated. The generated images from the network had a Structural Similarity Index score of 0.463 and a Histogram Similarity Score of 0.999 with the ground truth images, indicating that there is better than moderate similarity between the generated and real images. The eye test supports this argument due to the generated images being grainy and pixels with low luminance not being generated as expected.**

I. Introduction

About 80-90% of sensory information is acquired through the eyes, highlighting the significance of visual content in human perception and cognition. This is followed by hearing which accounts for about 10% of the information we ingest, with the other senses making up the rest. In terms of data, visual input is represented in images or videos, whereas audio in sound signals, waveforms and so on.

While the field of AI has made significant strides in generating visual content from text and audio, the task of directly generating videos from audio is relatively new. This project aims to bridge this gap by developing a deep learning model capable of synthesizing realistic images from raw audio input.

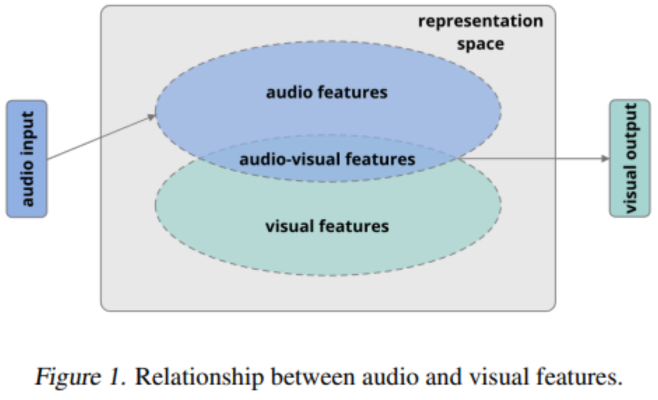


Fig. 1: Intersection between audio and visual data [1]

As shown in Fig 1, there is an intersection between audio and visual features that needs to be exploited in order to achieve true multi-modality. This would eliminate the bias of intermediary modality and also pave the way for identifying more granular relationships between audio and visual information.

To achieve that, I propose a hybrid architecture that combines two component networks from Autoencoder and Generative Adversarial Networks that can take audio signals as input and output an image corresponding to it. I used a Multi-modal Scene Classification dataset that contains audio-image pairs taken from different environments like forests, beaches, cities, etc. to train this architecture. The audio data serves as the input to the model, whereas the images will be used to evaluate the images outputted by the model. The results indicate that the network is able to achieve moderate performance in generating images that closely resemble the expected output with respect to similarity metrics and human evaluation.

II. Background & Related Work

Audio-to-Video generation has been gaining a lot of popularity lately with advances in Deep Learning. For common audio-to-image or audio-to-video tasks like language translation, most methods used a text intermediary for generating visual output, there are some architectures used in latest literature that closely resemble the approach I proposed.

[1] used a Variational Autoencoder (VAE) network takes MFCC features from audio data and generates videos from that data. They also used a discriminator network to critique and improve the performance of the generator. The performance of their framework was tested on two benchmark datasets that contained audio recordings of different users uttering numbers from the MNIST dataset. The full methodology they used is shown in Fig 2.

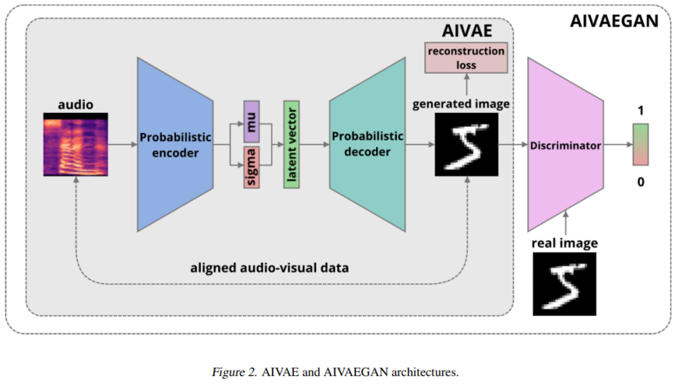


Fig. 2: Model Architecture [1]

Meta's ImageBind [2] is a multi-modal Large Language Model that has state-of-the-art performance in identifying different modalities of the same data, especially audio and visual. [3] leveraged ImageBind to create a shared latent space to establish connections between multiple generative models that usually target just one modality. Their method can enable generation across multiple modalities video-to-audio, audio-to-video, joint video-audio generation among others. This indicates the quality of the multi-modal embedding space that they were able to create using ImageBind. The full methodology used by this study is shown in Fig 3.

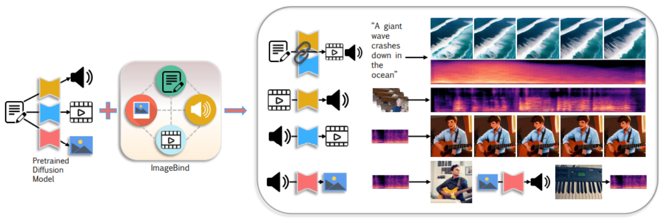

"A giant wave crashes down in the ocean"
Pretrained Diffusion Model
ImageBind

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.

Fig. 3: Framework [3]

OneShotA2V [4] leverages curriculum learning to generate videos of a talking person by inputting an unseen image of the person and the audio they will be speaking. The architecture consists of one generator and three discriminator modules that evaluate different aspects of the generated output like frame consistency, temporal consistency and audio-video synchronization. They trained the model in phases using different loss values like blink loss, temporal adversarial loss, etc. to achieve synchronization across lip movements, blinking patterns, etc. The architecture used for the model is shown in Fig 4.

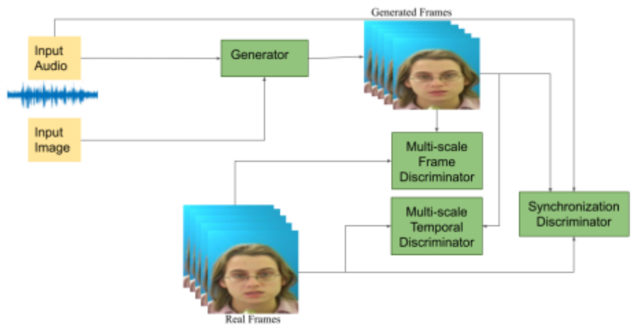

Generated Frames
Input Audio
Generator
Input Image
Multi-scale Frame Discriminator
Multi-scale Temporal Discriminator
Synchronization Discriminator
Real Frames

Figure 1. Model for generating robust and high-quality videos. This uses deep speech audio features to be fed into SPADE Generator and 2 discriminators i.e frame discriminator which is a multi-scale discriminator for frame generation and another discriminator for better lip synchronization.

Fig. 4: OneShotA2V [4]

In visial scene generation tasks, studies such as [5] have also used latent representations and generative methods for converting audio to video. AudioToken [6] adapted text-conditioned diffusion models to generate images conditioned on the audio inputs. Whereas [7] leveraged audio and semantic consistency with generative networks to achieve improved performance in generating images from audio descriptions. And [8] used a cross-model approach combining two models with generative networks for intersensory generation.

III. Dataset

A. *Multi-Modal Scene Classification*

The dataset used for this project was curated in this study [9]. It consists of over 17000 audio-image pairs extracted from 37 different video sources recorded in 8 different environments. The videos were sliced into frames every one second and then the visual (image) and audio data were isolated and extracted separately. The images are all sized 256x256 and the audio data is available as MFCC features. The dataset is accessible on Kaggle [10].

B. *Data Pre-processing*

1) Audio: Mel-frequency Cepstral Coefficients (MFCC) are a set of features that describe the overall shape and spectrum of an audio signal's spectrum. 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). They are often used in speech recognition and can be interpreted as a representation of timbre, meaning the quality of a sound that makes it unique. Studies [11], [12] have used MFCCs to represent audio signals as well as the research presented in the Literature section.

The audio data in the dataset is already available in the required format i.e. extracted MFCC features. A total of 104 MFCCs were extracted for each audio-image pair. The data is of high quality and has no missing values or values that need to be dropped.

2) Images: Certain classes in the dataset like 'CITY' have a lot of variance from frame to frame. The elements from an image are distinctly different from the next image and hence all classes except the 'BEACH' class were dropped. Within the 'BEACH' class, there are some images that contain multiple people at the beach shown in Fig 6. Due to the significant differences such images from the class, the model was not able to converge even after many epochs. Hence, they were excluded as well.

Most of the images from the 'BEACH' class were consistent and there is good correlation from frame to frame i.e. image to image. A total of 480 image-audio pairs were chosen for training and testing the network, all from the 'BEACH' class. The data was split in a 3:1 ratio - so 360 pairs for training and the remaining 120 for testing. A sample image from the training sample is shown in Fig 7.

IV. Model

The network architecture I have chosen for this problem is a hybrid approach between an Autoencoder and a Generative

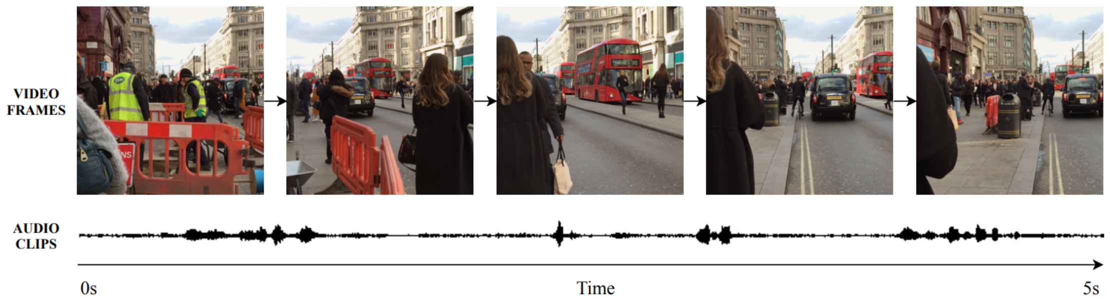


Fig. 3: Example of extracted data from a five second timeline. Each second, a frame is extracted from the video along with the accompanying second of audio.

Fig. 5: Dataset Extraction [9]

Fig. 6: Excluded Data: Images

Fig. 7: Included Data: Images

Adversarial Network (GAN). To recap, an Autoencoder has two components: an encoder and a decoder. The encoder network ingests input data and fits it into a lower or latent space representation. This lower dimensional input is then fed into the decoder that reconstructs the original data. On the other hand, GANs also have two components: a generator and a discriminator. The discriminator network is trained to distinguish between real and fake data, while the generator is trained to generate fake samples from random noise with an intention to fool the discriminator into misclassifying fake samples. Both networks work against each other in an adversarial fashion, and after training we end up with a generator that can generate high quality fake samples that closely resemble the real data.

Considering the task at hand, the aim is to find the common ground between audio and visual representations of data. Combining the two architectures mentioned above, I built an encoder-decoder model that can ingest extracted MFCC features from audio signals and generate images. The encoder is a set of five fully connected layers that progressively convert the audio features into the latent space representation. This representation is then fed to the generator (decoder) network for further processing. The encoder architecture is shown in Table I. The activation function used is Leaky ReLU.

Before the latent representation is passed to the generator network, it is reshaped to match the number of dimensions of the expected output i.e. four = batch size, channels, height, width. This reshaped tensor is the input to the generator. The generator (decoder) network is a set of eight 2D deconvolutional layers that progressively upsample the latent space

TABLE I: Encoder Architecture

Layer	Dimensions	Activation
Fully Connected	input x 1024	Leaky ReLU
Fully Connected	1024 x 768	Leaky ReLU
Fully Connected	768 x 512	Leaky ReLU
Fully Connected	512 x 256	Leaky ReLU
Fully Connected	256 x 100	Leaky ReLU

representation into an image tensor. All the deconvolutional layers use a kernel size of 4 and a stride of 2 units. So the tensor dimensions (height, width) are doubled as it passes through each layer and we end up with a 256x256 tensor that will be fine-tuned to closely represent the expected output. The decoder architecture is shown in Table II. Each deconvolutional layer is followed by a 2D Batch Normalization layer and uses Leaky ReLU activation, except the output layer. The output layer i.e. the final deconvolutional layer uses Tanh activation and it outputs 3 channels (one each for RGB).

TABLE II: Generator (Decoder) Architecture

Layer	Dimensions	Activation
2D Deconvolution	100 x 256	
2D Batch Normalization	256	Leaky ReLU
2D Deconvolution	256 x 128	
2D Batch Normalization	128	Leaky ReLU
2D Deconvolution	128 x 64	
2D Batch Normalization	64	Leaky ReLU
2D Deconvolution	64 x 32	
2D Batch Normalization	32	Leaky ReLU
2D Deconvolution	32 x 16	
2D Batch Normalization	16	Leaky ReLU
2D Deconvolution	16 x 8	
2D Batch Normalization	8	Leaky ReLU
2D Deconvolution	8 x 4	
2D Batch Normalization	4	Leaky ReLU
2D Deconvolution	4 x 3	Tanh

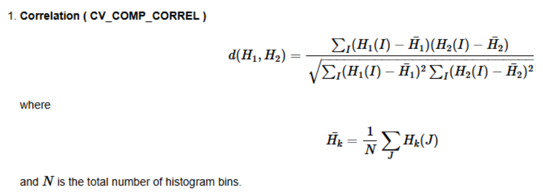
This network is trained for 100 epochs with a batch size of 64 on the MM Scene Classification dataset discussed in the previous section. The audio data serves as the input to the model and the images will be used to calculate the loss and evaluate the model performance. The Adam optimizer and a learning rate of 0.0625 were used for the optimization process. The loss function is calculated as loss = Mean Squared Error (MSE) * batch_size. The MSE is calculated between the generator output (generated images) and expected output (target images).

V. Evaluation

A. Metrics

In addition to the test loss, the network's performance will be judged on the following criteria,

- **Similarity Metrics:** To quantify the similarity between the generated and real images, I used two similarity metrics: Histogram Similarity Score (HSS) and Structural Similarity Index (SSIM). HSS compares the histograms of two images and calculates the similarity using the selected metric. The metric I chose is correlation so the HSS score is an estimation of the correlation between the two histograms given by the Pearson coefficient. The formula for calculating HSS is shown in Fig 8.



1. **Correlation (CV_COMP_CORREL)**

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_I (H_1(I) - \bar{H}_1)^2 \sum_I (H_2(I) - \bar{H}_2)^2}}$$

where

$$\bar{H}_k = \frac{1}{N} \sum_J H_k(J)$$

and N is the total number of histogram bins.

Fig. 8: Formula for calculating HSS

SSIM on the other hand, is more closer to how humans perceive images. It compares luminance, texture, contrast and structure between two images [13]. The formula for calculating SSIM is given by Equation (1), where x,y are the two images, mu and sigma stand for mean and variance, c1, c2 are two variables used to stabilize the division with weak denominator.

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \qquad (1)$$

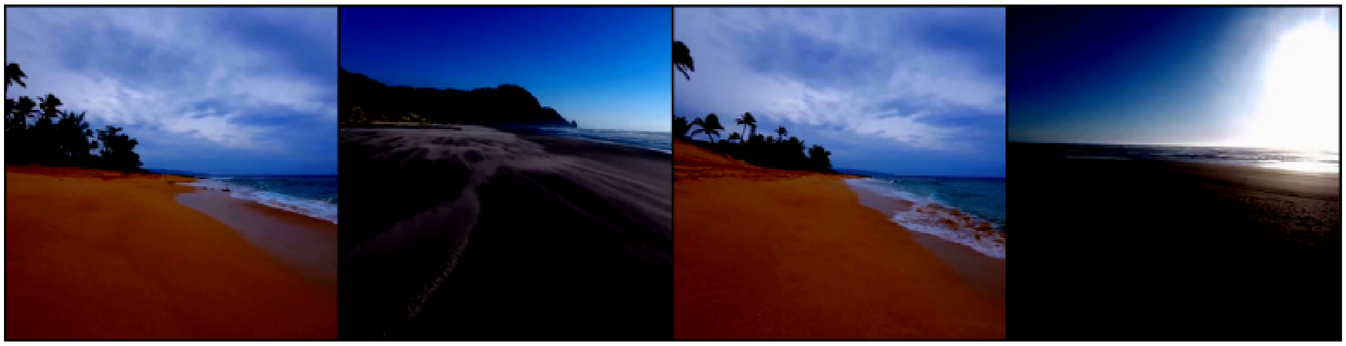
- **Human Evaluation:** To assess the quality of the generated images, human evaluation or eye test will also be performed. Points of interest: distinguishing features, texture, performance on low luminance pixels among others.

B. Results

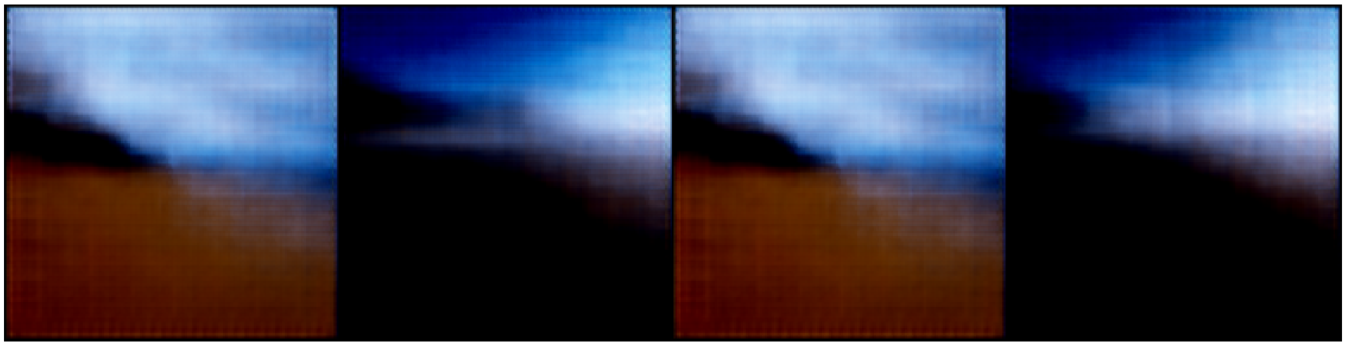
The network had an average HSS score of 0.999 on the test set. The key point to note here is that the histogram score is frequency and intensity based. The high score indicates that the network is able to generate images that are matching most pixel intensities from the target images. This does not fully capture how similar the generated image is to the target image.

The network had an average SSIM score of 0.463 on the test set. As mentioned earlier, this metric is closer to how humans perceive images and is a better choice for quantifying image similarity. The moderate score indicates that the network is predicting images that have some resemblance to the target images, and there is room for improvement.

The network reported a loss of 4.691 on the test set which is slightly higher than the train loss indicating moderate performance and generalization. Comparing the network's predictions on the test set with the target images using the eye test, it is clear that the generated images are similar but not quite matching the expected output. The texture is grainy and no distinguishing features like the trees in the background are being captured. Another observation is that the network is unable to generate pixels that have low luminance as seen in Fig 9. The figure shows a side-by-side comparison between the expected output (real images) and actual output (generated images).



(a) Real Images



(b) Generated Images

Fig. 9: Real vs Generated Images

C. Limitations

One of the major limitation of the study is the small sample size used for training and testing the network. Generating data in a different modality to the input data is a complex task and the network should have access to significantly more data points.

In addition, class diversity within the train-test sample can also be increased. Currently, the sample only contains audio-image pairs from videos taken at the beach. Adding other environments to the sample will improve the robustness of the results and the training process. Both will be addressed in future works.

VI. Conclusion & Future Scope

In conclusion, I built a network that leverages the encoder architecture from Autoencoder and the generator architecture from Generative Adversarial Networks. It is trained on a sample taken from the Multimodal Scene Classification dataset curated in [9]. The encoder in the network ingests MFCC features extracted from audio signals and converts it into a latent space representation. This representation is fed to the generator which then upsamples it into an image. The network had a SSIM score of 0.463 and a HSS score of 0.999 indicating that it is able to generate images that reasonably match the expected output. From the eye test, the images are still grainy and pixels with low luminance aren't being generated properly. This can be attributed to the small training and testing sample used for the experiments, which is a major limitation of the study.

In the future, I would like to work on the following:

- **Video Processing:** The original idea for the project was to generate videos from audio signals but I had infrastructure issues which forced me to work with images. I am excited to test this architecture with videos.
- **Dataset:** As mentioned earlier, one of the major limitations of the study is the dataset being too small for the task at hand. A larger and more diverse dataset like AVSpeech [14] can be considered to improve the network's performance.
- **Network Architecture:** Adding a Discriminator to the network architecture to fine-tune the generators output to be better. This is especially desirable when dealing with videos. The addition of ImageBind is also an interesting direction - the model has excellent results in identifying different modalities of the same data and its embeddings are top quality as seen in [3].
- **Loss Function:** Currently the loss function only looks at the mean squared error. To improve the quality and sharpness of the generated images/videos, it is imperative that items like reconstruction loss, similarity scores and frame-to-frame cohesion are factored into the loss function.

CONTRIBUTIONS

I am the sole contributor to this project. The full dataset is available at [10]. The code can be found in this Colab notebook. https://colab.research.google.com/drive/19GDE_VEJhTq2igvfnKOWCZiQj1RxzxJx?usp=sharing. The code and sample used for this study can be found at this Git repohttps://github.com/sai-vuruma/ml_project.

REFERENCES

[1] M. Żelaszczyk and J. Mańdziuk, "Audio-to-image cross-modal generation," 2021. [Online]. Available: https://arxiv.org/abs/2109.13354

[2] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra, "Imagebind: One embedding space to bind them all," 2023. [Online]. Available: https://arxiv.org/abs/2305.05665

[3] Y. Xing, Y. He, Z. Tian, X. Wang, and Q. Chen, "Seeing and hearing: Open-domain visual-audio generation with diffusion latent aligners," 2024. [Online]. Available: https://arxiv.org/abs/2402.17723

[4] N. Kumar, S. Goel, A. Narang, and M. Hasan, "Robust one shot audio to video generation," 2020. [Online]. Available: https://arxiv.org/abs/2012.07842

[5] K. Sung-Bin, A. Senocak, H. Ha, A. Owens, and T.-H. Oh, "Sound to visual scene generation by audio-to-visual latent alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6430–6440.

[6] G. Yariv, I. Gat, L. Wolf, Y. Adi, and I. Schwartz, "Audiotoken: Adaptation of text-conditioned diffusion models for audio-to-image generation," 2023. [Online]. Available: https://arxiv.org/abs/2305.13050

[7] P.-T. Yang, F.-G. Su, and Y.-C. F. Wang, "Diverse audio-to-image generation via semantics and feature consistency," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2020, pp. 1188–1192.

[8] L. Chen, S. Srivastava, Z. Duan, and C. Xu, "Deep cross-modal audio-visual generation," 2017. [Online]. Available: https://arxiv.org/abs/1704.08292

[9] J. J. Bird, D. R. Faria, C. Premebida, A. Ekárt, and G. Vogiatzis, "Look and listen: A multi-modality late fusion approach to scene classification for autonomous machines," 2020. [Online]. Available: https://arxiv.org/abs/2007.10175

[10] J. J. Bird, "Scene classification: Images and audio," 2020. [Online]. Available: https://www.kaggle.com/datasets/birdy654/scene-classification-images-and-audio

[11] B. Logan *et al.*, "Mel frequency cepstral coefficients for music modeling." in *Ismir*, vol. 270, no. 1. Plymouth, MA, 2000, p. 11.

[12] Z. K. Abdul and A. K. Al-Talabani, "Mel frequency cepstral coefficient and its applications: A review," *IEEE Access*, vol. 10, pp. 122 136–122 158, 2022.

[13] J. Nilsson and T. Akenine-Möller, "Understanding ssim," 2020. [Online]. Available: https://arxiv.org/abs/2006.13846

[14] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation," *ACM Transactions on Graphics*, vol. 37, no. 4, p. 1–11, Jul. 2018. [Online]. Available: http://dx.doi.org/10.1145/3197517.3201357