ENHANCED SHARED EXPERIENCES IN HETEROGENEOUS NETWORK WITH GENERATIVE AI

Neeraj Kumar[1,2], Ankur Narang[1], Brejesh Lall[2], Nitish Kumar Singh[3]
[1]Hike Private Limited, India, [2]IIT Delhi, India

NOTE: Corresponding author: Neeraj Kumar, neerajku@hike.in

Abstract – *COVID-19 has made the immersive experiences such as video conferencing, virtual reality/augmented reality, the most important modes of exchanging information. Despite much advancement in the network bandwidth and codec techniques, the current system still suffers from glitches, lags and poor video quality, especially under unreliable network conditions. In this paper, we propose the method of a video streaming pipeline to provide better video quality under erratic network conditions. We propose an environment where the participants can interact with each other through video conferencing by only sending the audio in the network. We propose a Multimodal Adaptive Normalization (MAN)-based architecture to synthesize a talking person video of arbitrary length using as input: an audio signal and a single image of a person. The architecture uses multimodal adaptive normalization, keypoint heatmap predictor, optical flow predictor and class activation map-based layers to learn movements of expressive facial components and hence generates a highly expressive talking-head video of the given person. We demonstrate the effectiveness of proposed streaming that dynamically controls the Quality of Experience (QoE) as per the requirements.*

Keywords – Audio to video generation, deep learning architecture, dynamic QoE control, GAN, multimodal adaptive normalization, video streaming pipeline

1. INTRODUCTION

The ongoing COVID-19 pandemic has forced people to work, learn, and communicate remotely on an unprecedented scale. With more people in quarantine and isolation, the demand for low latency applications, such as video streaming, online games, and teleconferencing has soared to the point that it has prompted some countries to look at ways to curb streaming data to avoid overwhelming the Internet. Several large companies have already announced that this unintended pilot on remote teleworking might become the norm.

Immersive media is likely to further exacerbate the issues related to bandwidth and latency (even in the new generation 5G networks), since all next-generation media types, either omnidirectional (360 degree) or multiview or three-dimensional, impose bandwidth requirements and latency requirements that vastly surpass those of traditional media.

With the emergence of 5G networks, ultrafast, ultra-reliable, and high bandwidth capable edge becomes an attractive option to media services developers. For immersive media, 5G is a crucial enabling technology, since its targeted key performance indicators stipulated by the architecture documents are essential to providing good Quality of Experience (QoE) for the users. With the 5G network, a videoconferencing pipeline in erratic conditions can still be challenging and advancements will be made to lower the latency and network bandwidth and provide better user experience.

A lot of work has been done on the development and optimization of novel video codecs to enhance the quality of video streaming. Various codecs have been developed to reduce the amount of streamed data while maintaining as much information as possible in the network.

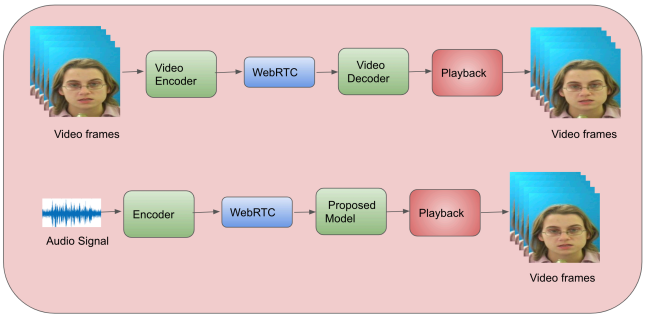


Fig. 1 – Top: Typical video streaming pipeline. In the typical system, the input video is encoded using video codecs and sent to the receiver which decodes it in the form of a lossy reconstruction that preserves most of the video features at a pixel level. Bottom : Proposed streaming pipeline where the audio signal is sent through a general-purpose WebRTC DataChannel and at the receiver side, the proposed model converts the audio into the video signal.

In a typical system (Fig. 1), the data is first read from a video source and compressed. The compressed data is sent over a network to the receiving end, where a decoding algorithm reconstructs a representation of the original feed from the streamed data. Since most of the codecs are lossy, the reconstruction process at the receiver end does not create the original feed but sufficiently close to the original with some distortions. The compression techniques utilize the fact that not all the information contained within a video frame is

equally important and prioritize the preservation of more important aspects of a feed over others in the compression/decompression process. Despite these advancements, a lot of work needs to be done in order to give an enhanced videoconferencing experience in unreliable network conditions [1] such as glitches, lags, low internet bandwidth, etc.

In this paper, we propose the audio driven video conferencing methodology that helps in improving the video quality in odd network scenarios. In the proposed method, we have used a GAN-based approach at the receiver's end to generate video with enhanced quality under unreliable conditions. One of the possible concerns of this methodology is that it shifts the burden from communication bandwidth to increased computation at the receiver end. The use of a GAN-based [2] approach can increase the latency resulting in the lag of video during streaming. But with the rapid improvement of hardware capabilities in mobiles and personal computers, this is unlikely to be a major obstacle. With the recent development of NVIDIA Maxine project [3], such hurdles can be resolved and results into the practical system that provides immense gains over the conventional methods.

Given an arbitrary image and an audio sample, we propose multimodal adaptive normalization in the proposed architecture to generate realistic videos. We have built the architecture based on [4] to show how multimodal adaptive normalization helps in generating highly expressive videos using the audio and person's image as input. The proposed GAN architecture consists of generator and discriminator. The generator has two major components, namely multimodal adaptive normalization framework and class activation attention map. A multimodal adaptive normalization framework feeds various features such as optical flow/keypoint heatmaps, single image, audio melspectrogram, pitch and energy of the audio frames to the generator to produce realistic and expressive video. A class activation attention map helps the generator to produce global features such as eyes, nose, lips, etc and local features such as movements of facial action units properly which will increase the video quality. The discriminator used in the proposed method is multiscale with a class activation attention layer to discriminate fake and real frames at the global and local level.
Our main contributions are :

- The proposed speech driven facial video synthesis architecture is a GAN-based approach that consists of a generator and discriminator in Section 4. The generator incorporates the multimodal adaptive normalization framework (Fig. 9), optical flow/keypoint predictor and class activation map-based attention layer to generate the expressive videos. The discriminator uses multiscale patchGAN-based discriminator along with a class activation map-based layer to classify fake or real images.

- We have shown how the Quality of Experience (QoE) in videoconferencing has improved in low bandwidth networks by the proposed architecture in Section 7.2.2. The proposed videoconferencing pipeline helps in controlling the QoE based on the compute resource, bandwidth availability and importance of the speaker in the videoconference. It can further be used in data privacy by synthesizing the video on person or avatar. Noisy audio can be handled by the proposed model and generates the expressive output and gives a high quality of experience.

- Various experimental (Section 7.2) and ablation studies (Section 7.3) have shown that the proposed multimodal adaptive normalization is flexible in building the architecture with various networks such as 2DConvolution, partial2D convolution, attention, LSTM, Conv1D for extracting and modeling the mutual information.

- The proposed multimodal adaptive normalization-based architecture for video synthesis using audio and a single image as an input has shown superior performance on multiple qualitative and quantitative metrics such as Structural Similarity Index (SSIM), Peak Signal to Noise Ratio (PSNR), Cumulative Probability of Blur Detection (CPBD), Word Error Rate (WER), blinks/sec and Landmark Distance (LMD) in tables 1, 2, 3 and 4. The generated videos are given at [1].

2. BACKGROUND

2.1 Audio to video generation

Audio to video generation is an active area of research due to its wide range of applications such as for the entertainment industry, education, healthcare and many more. Computer Generated Imagery (CGI) has become an important part of the entertainment industry due to its ability to produce high quality results in a controllable manner.

Facial animation is an important part of CGI as it is capable of conveying a lot of information not only about the character but also about the scene in general. The generation of realistic and expressive animation is highly complex due to its multiple properties such as lip synchronization with audio, movements of a facial action units for expressiveness and natural eye blinks. Facial synthesis in CGI is traditionally performed using face capture methods, which have seen drastic improvements over the past years and can produce faces that exhibit a high level of realism. However, these approaches require expensive equipment and significant amounts of labour. In order to drive down the cost and time required to produce high quality, researchers are looking into automatic

[1] https://sites.google.com/view/itu2021

face synthesis using machine learning techniques.

Machine learning methods could simplify the video generation process by automatically producing it from the audio. Such methods could be applied in post-production of film making to achieve better lip synchronization. They can be applied in the education sector to teach students in a more realistic manner that can help reduce the cost of teaching. Apart from that, such techniques can be used to generate parts of the face that are occluded or missing in a scene. This technology can improve band-limited visual telecommunications by either generating the entire visual content based on the audio or filling in dropped frames.

2.2 Facial video

Facial video generation is a complex problem. It has several properties which make the video realistic.

- Semantic consistency - The facial features such as eyes, nose, lips, etc. should be consistent among each other.
- Temporal consistency - Video consists of several frames. Each frame should be temporal smoother with its previous and next frames, so that there should not be any jitters, spikes or holes in the video.
- Expressiveness - This property makes the video more realistic and natural. Properties such as movement of facial action units, lip synchronization with the audio and the blinking of eyes make the video more realistic and visually appealing.

While generating the video from audio, the predicted videos should inhibit such properties. Optical flow and a keypoint heatmap help in making the video semantically and temporally consistent as well as more expressive.

2.2.1 Optical flow

Optical flow is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of object or camera. It is a 2D vector field where each vector is a displacement vector showing the movement of points from the first frame to the second. Optical flow has many applications in areas such as structure from motion [5], video compression [6] and video generation. The optical flow helps in achieving the temporally smoother videos. Fig. 2 shows the optical flows between the two consecutive frames of any videos. The optical flow gives the temporal as well as spatial information based on the movement of the intensity values of the frames.

2.2.2 Keypoint heatmap

Facial landmark detection is a well-studied topic in the field of computer vision with many applications such as face verification [7], face recognition [8], and facial attribute inference [9]. The high variability of shapes, poses, lighting conditions, and possible occlusions makes it a particularly challenging task even today. Such variabilities can be captures using the facial landmark keypoints. We detect the landmark keypoints around the cheeks, nose, eyes, lips to capture the movement of face while speaking or giving expressions using deep learning techniques. The heatmap of keypoints helps in giving a coarser view of these keypoint locations. Such heatmaps help the model to focus on the regions around the lips, noise, eyes and cheeks such that it captures the expressiveness of the image. Fig. 3 shows the landmark points of the images on the upper part of the image. The lower part shows the heatmap of the keypoints which gives the information about the expressiveness of the images.

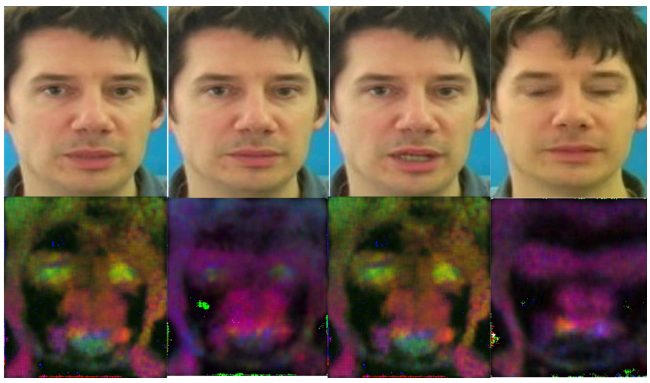


Fig. 2 – Top: Frames of the video. Bottom: Optical flow of the video.

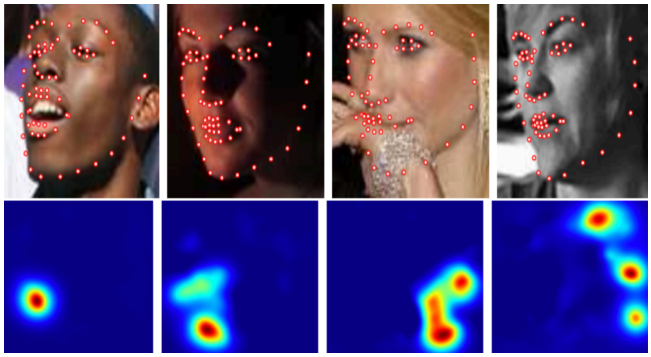


Fig. 3 – Top: facial keypoints. Bottom: keypoint heatmap of the face.

2.3 Audio

Modeling audio is a complex problem. Several aspects of the synthesized speech, such as a speaker's voice, speaking style/prosody and noise comes into play to better incorporate the audio into modeling. The range of prosody in the dialogue must encompass a large range of human conversation, from neutral expression to extremely emotional, while always sounding perfectly natural. Here, prosody refers to the variation of several speech related phenomena such as intonation, stress, rhythm and style of the speech. Traditionally, prosody modeling is based on schematizing and labeling prosodic phenomena and developing rule-based systems or statistical models from

the derived data. However, the prosodic attributes are difficult and time consuming to annotate. The prosody of speech is best captured by pitch, energy and melspectrogram of the audio frames. Such features help the deep learning model to incorporate natural and expressive audio to meet the end tasks such as generation of expressive video.

2.3.1 Pitch

Pitch is the fundamental frequency of an audio waveform, and is an important parameter in the analysis and synthesis of speech and music. Normally only voiced speech and harmonic music have well-defined pitch. But we can still use pitch as a low-level feature to characterize the fundamental frequency of any audio waveform. The typical pitch frequency for human speech is between 50 and 450 Hz, whereas the pitch range for music is much wider.

2.3.2 Energy

Energy models the excitation pattern on the basilar membrane by simulating the acoustic signal transformations in the ear according to the perceptual model of the human auditory system. Short-term speech energy is closely related with activation or arousal dimension of the emotion, its usage in the conventional features contributes to the classification of emotions.

2.3.3 Melspectrogram

A melspectrogram is a spectrogram where the frequencies are converted to the mel scale.This mel scale is constructed such that sounds of equal distance from each other on the mel scale, also "sound" to humans as they are equal in distance from one another. In contrast to the Hz scale, where the difference between 500 and 1000 Hz is obvious, whereas the difference between 7500 and 8000 Hz is barely noticeable.

2.4 Generative adversarial network

The Generative Adversarial Network (GAN) [10] consists of the generative model and discriminative model. The GAN framework naturally takes up a game-theoretic approach. The word "adversarial" is chosen as the two networks, i.e., generator and discriminator are in constant conflict and compete with each other. The generative model can be thought of as analogous to a team of counterfeiters, trying to create money similar to the real ones while the discriminator acts as police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods by constantly giving knowledge and feedback until the counterfeits are indistinguishable from the genuine articles.
The generative model generates samples by passing random noise through a multilayer perceptron, and the discriminative model is also a multilayer perceptron. We can train both models using only the highly successful

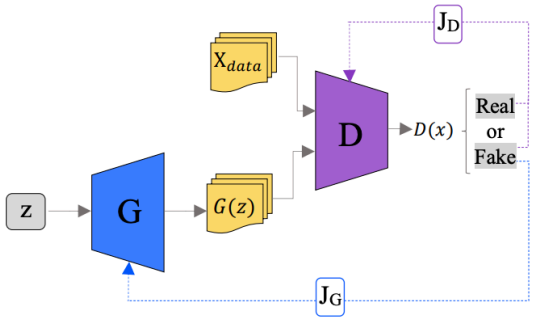


Fig. 4 – Architecture of Generative Adversarial Network (GAN)

back propagation and dropout algorithms and sample from the generative model using only forward propagation.

Fig. 4 shows the general architecture of GAN. To learn the generator's distribution p_g over data x, we define a prior on input noise variables $p_z(z)$, then represent a mapping to data space as $G(z;\theta_g)$, where G is a differentiable function represented by a multilayer perceptron with parameters θ_g. We also define a second multilayer perceptron $D(x;\theta_d)$ that outputs a single scalar. $D(x)$ represents the probability that x came from the data rather than p_g. We train D to maximize the probability of assigning the correct label to both training examples and samples from G. We simultaneously train G to minimize $\log(1-D(G(z)))$: In other words, D and G play the following two-player minimax game with value function $V(G,D)$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))].$$

2.5 Normalization

The normalization framework has become the integral part of neural network training. It has gained success due to many reason such as a higher learning rate, faster training, regularization effects, smoothing of loss landscape, etc. The variants of normalization are discussed in the following subsections. One of the first normalization architecture proposed was batch normalization [11] which helps the deep learning community understand the effect of normalization.

2.5.1 Batch normalization

In traditional deep networks, a too-high learning rate may result in the gradients that explode or vanish, as well as getting stuck in poor local minima. Batch normalization [11] helps address such issues. By normalizing activations throughout the network, it prevents small changes to the parameters from amplifying into larger and suboptimal changes in activations in gradients; for instance, it prevents the training from getting stuck in the saturated regimes of nonlinearities.

The Dropout [12] is typically used to reduce overfitting but in a batch-normalized network it can be either removed or reduced in strength and helps in better generalization of the network. Batch normalization reduces the photometric distortions because batch normalized networks train faster and observe each training example fewer times, we let the trainer focus on more "real" images by distorting them less.

Equation *(4)* is the batch normalized output with input($x_1 \cdots x_n$) used to calculate the mean (Equation *(1)*) and variance (Equation *(2)*) which is used to get the normalized output ($\hat{x}_1 \cdots \hat{x}_n$) (Equation *(3)*). Need of normalization occurs as distribution invariance assumption is not satisfied at local level. Without normalization, the model has to run more steps for parameters to adapt. Use of scale (γ) and bias (β) in Equation *(4)* gives flexibility to work with normalized input and also with scaled normalized input, if there is a need, thus increasing the representation power.

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \qquad (1)$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \qquad (2)$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \qquad (3)$$

$$y_i = \gamma \hat{x}_i + \beta \qquad (4)$$

2.5.2 Variants of normalization

Variants of normalization have been used to capture various information such as style, texture, shape, etc. Instance Normalization (IN) [13] is a representative approach which was introduced to discard instance-specific contrast information from an image during style transfer. Inspired by this, adaptive instance normalization [14] provided a rational interpretation that IN performs a form of style normalization, showing that by simply adjusting the feature statistics, namely the mean and variance of a generator network, one can control the style of the generated image. IN dilutes the information carried by the global statistics of feature responses while leaving their spatial configuration only, which can be undesirable depending on the task at hand and the information encoded by a feature map. To handle this, Batch-Instance Normalization(BIN) [15] normalizes the styles adaptively to the task and selectively to individual feature maps. It learns to control how much of the style information is propagated through each channel of features leveraging a learnable gate parameter. For style transfer across the domain, UGATIT [16] has used adaptive instance and Layer Normalization (LN) [17] which adjusts the ratio of IN and LN to control the amount the style transfers from one domain to other domains.

For style transfer tasks, a popular methodology is trying the denormalization to the learned affine transformation that is parameterized based on a separate input image (the style image). SPADE [18] makes this denormalization spatially sensitive. SPADE normalization boils down to "conditional batch normalization which varies on a per-pixel basis". In world-consistent video to video synthesis [19], they have used optical features and semantic maps in the normalization to learn the affine parameters to generate the realistic and temporally smoother videos.

We have proposed multimodal adaptive normalization to incorporate the higher-order statistics of multimodal fea- tures (image and audio) through affine parameters of nor- malization i.e. scale (γ) and shift (β) .

3. RELATED WORK

There have been many years of research on video codecs for various applications such as AV1 [20] and VVC [21] codecs. Researchers are working on improving the codes using machine learning techniques either by end to end approaches or working on specific parts of video streaming pipelines.

In one of the approaches , face detection/mesh extraction [22, 23, 24, 25] and on body pose tracking [26, 27, 28], focusing on both 3D and 2D meshes, generally based on neural networks are used to encode the video streams and sent to the data channel. The final video is then reconstructed back by using body pose along with mesh at the receiver side to make the video streaming pipelines in erratic network conditions.

There was some work on video compression and reconstruction based on facial landmarks in [29, 30], which are promising in extremely low bitrates, but did not demonstrate real-time conferencing capabilities.

3.1 Audio to realistic video generation

The earliest methods for generating videos relied on Hidden Markov Models which captured the dynamics of audio and video sequences. Simons and Cox [31] used the Viterbi algorithm to calculate the most likely sequence of the mouth shape given particular utterances. Such methods are not capable of generating quality videos and lack emotions.

3.1.1 Phoneme and visemes generation of videos

Phoneme and viseme-based approaches have been used to generate videos. Real-Time Lip Sync for Live 2D Animation [32] has used an LSTM-based approach to generate live lip synchronization on 2D character animation.

Some of these methods target rigged 3D characters or meshes with predefined mouth blend shapes that correspond to speech sounds [33, 34, 35, 36, 37, 38] which have primarily focused on mouth motions only and show a finite number of emotions, blinks, facial action units movements.

3.1.2 Deep learning techniques for video generation

CNN-based architectures for audio to video generation: A lot of work has been done on CNN to generate realistic videos given an audio and static image as input. [39](Speech2Vid) has used encoder-decoder architecture to generate realistic videos. They have used L1 loss between the synthesized image and the target image. Our approach has used multimodal adaptive normalization in GAN-based architecture to generate realistic videos.

Synthesizing Obama: Learning lip sync from audio [38] is able to generate quality videos of Obama speaking with accurate lip-sync using RNN-based architecture. They can generate only a single person video whereas the proposed model can generate videos on multiple images in GAN-based approach.

GAN-based architectures for audio to video generation: Temporal Gan [40] and generating videos with scene dynamics [41] have done the straightforward adaptation of GANs for generating videos by replacing 2D convolution layers with 3D convolution layers. Such methods are able to capture temporal dependencies but require constant length videos. The proposed model is able to generate videos of variable length with a low word error rate.

Realistic Speech-Driven Facial Animation with GANs (RSDGAN) [42] used a GAN-based approach to produce quality videos. They used identity encoder, context encoder and frame decoder to generate images and used various discriminators to take care of different aspects of video generation. The proposed method has used multimodal adaptive normalization along with class activation layers and an optical flow predictor and keypoint heatmap predictor in the GAN-based setting to generate expressive videos.

The X2face [43] model uses a GAN-based approach to generate videos given a driving audio or driving video and a source image as input. The model learns the face embeddings of source frame and driving vectors of driving frames or audio bases which generate the videos. In X2face, the video is processed at 1fps whereas the model generate the video at 25fps. The quality of output video is not good as compared to our proposed method with audio as an input.

MoCoGAN [44] uses RNN-based generator with separate latent spaces for motion and content. A sliding window approach is used so that the discriminator can handle variable-length sequences. This model is trained to generate disentangled content and motion vectors such that they can generate audios with different emotions and content. Our approach uses multimodal adaptive normalization to generate expressive videos.
[45] extracts the expression and pose from an audio signal and a 3D face is reconstructed on the target image. The model renders the 3D facial animation into video frames using the texture and lighting information obtained from the input video. Then they fine-tune these synthesized frames into realistic frames using a novel memory-augmented GAN module. The proposed approach uses multimodal adaptive normalization with predicted optical flow/keypoint heatmap as an input to learn the movements and facial expressions on the target image with audio as an input. CascadedGAN [46] have used the L-GAN and T-GAN for motion (landmark) and texture generation. They have used a noise vector for blink generation. Model Agnostic Meta Learning (MAML) [47] is used to generate the videos on an unseen person image. The proposed method has used multimodal adaptive normalization to generate realistic videos.

[48] uses an Audio Transformation network (AT-net) for audio to landmark generation and a visual generation network for facial generation. [49] uses audio, identity encoder and a three-stream GAN discriminator for audio, visual and optical flow to generate lip movement based on input speech. [50] enables arbitrary-subject talking face generation by learning disentangled audiovisual representation through an associative-and-adversarial training process. [51] uses a generator that contains three blocks: (i) Identity Encoder, (ii) Speech Encoder, and (iii) Face Decoder. It is trained adversarially with a visual quality discriminator and pretrained architecture for lip audio synchronization. [49, 50, 51] are limited to lip movements whereas the proposed method uses multimodal adaptive normalization to generate different facial action units of an expressive video. [52] uses Asymmetric Mutual Information Estimator (AMIE) to better express the audio information into generated video in talking face generation. They have AIME to capture mutual information to learn the cross-modal coherence whereas we have used multimodal adaptive normalization to incorporate multimodal features into our architecture to generate the expressive videos. [4] have used deep speech features into the generator architecture with spatially adaptive normalization layers in it along with lip frame discriminator, temporal discriminator and synchronization discriminator to generate realistic videos. They have limited blinks and lip synchronization whereas the proposed method used multimodal adaptive normalization to capture the mutual relation between audio and video to generate expressive video.

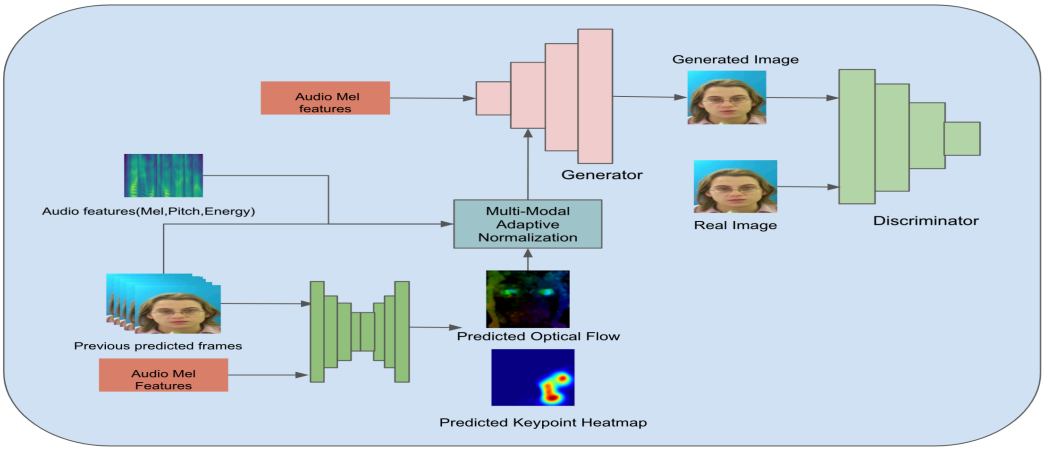


Fig. 5 – Proposed architecture for audio to video synthesis

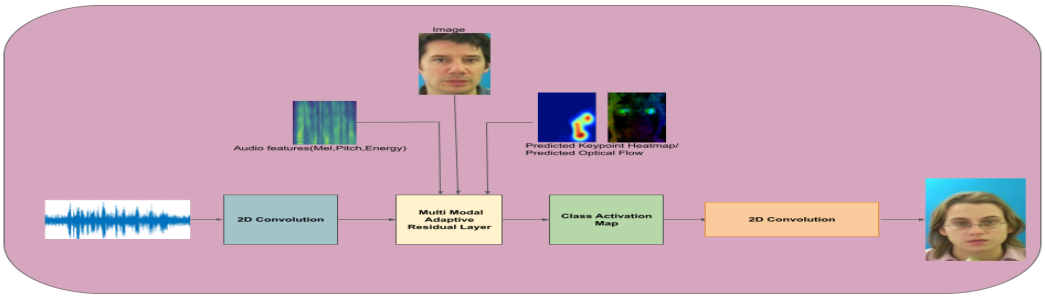


Fig. 6 – Generator architecture

4. ARCHITECTURAL DESIGN OF SPEECH DRIVEN VIDEO SYNTHESIS

Given an arbitrary image and an audio sample, the proposed method is able to generate speech synchronized realistic video on the target face. The proposed method uses multimodal adaptive normalization technique to generate realistic expressive videos. The proposed architecture is GAN-based which consists of a generator and a discriminator; see Fig. 5.

The architecture consists of 4 important subparts i.e. Generator, Discriminator, Multimodal Adaptive Normalization and Features Extractor Modules. The role of the generator is to generate realistic video frames (Fig. 6). The discriminator distinguishes between real and fake images and helps the generator to produce more realistic images (Fig. 14). The multimodal adaptive normalization provides necessary information/features i.e. pitch, energy and Audio Melspectrogram Features (AMF) from audio domain & static image and Optical Flow (OF)/facial Keypoint Heatmap (KH) features from video domain to the generator (figures 7, 10, 11). The feature extractor modules consists of various predictor modules such optical flow predictor, keypoint heatmap predictors, pitch, energy and audio melspectrogram extractors that extract necessary features such as Optical Flow (OF)/facial Keypoints Heatmap (KH), pitch, energy and melspectrogram features which go into the normalization framework.

4.1 Generator

Fig. 6 shows the generator architecture to generate realistic images. It consists of convolution layers, several layers having multimodal adaptive normalization-based Resnet [53] block (MANResnet) along with a class activation map layer. Fig. 7 shows the residual architecture around Multimodal Adaptive Normalization (MAN) along with 2d convolution and Relu [54] activation layers. The audio and video features namely person's image, predicted optical flow/predicted keypoint heatmap, melspectrogram features, pitch and energy go into the multimodal adaptive normalization network. Figures 10 and 11 show the multimodal adaptive normalization architecture which takes various features of audio and video domain and calculates the affine parameters i.e, scale, γ and a shift, β for normalization.

Class Activation Map (CAM)-based layer: This layer is employed as a layer of generator to capture the global and local features of the face. In class activation map [55], the concatenation of adaptive average pooling and adaptive max pooling of feature map create the CAM features which capture global and local facial features respectively. It helps the generator to focus on the image regions that

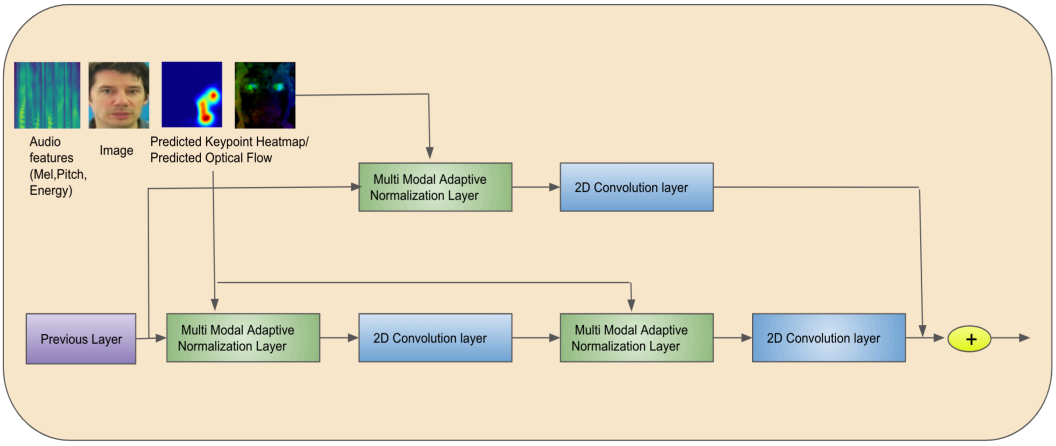


Fig. 7 – Multimodal adaptive normalization residual architecture

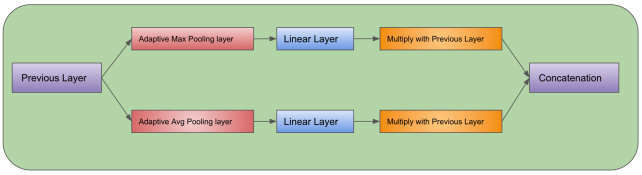


Fig. 8 – Class activation map layer architecture in generator

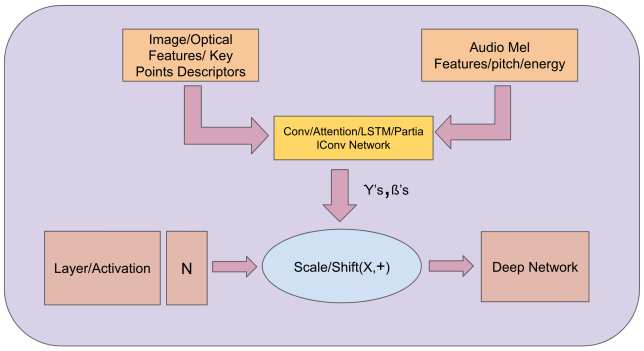


Fig. 9 – Higher level architecture of multimodal adaptive normalization

are more discriminative such as eyes, mouth and cheeks (Fig. 8).

4.2 Multimodal adaptive normalization

Fig. 9 shows the higher-level architectural design of multimodal adaptive normalization. The affine parameters i.e, scale, γ and a shift, β are typically used to learn the higher-order statistics of image features corresponding to style, texture, etc. to generate the required output as depicted in various previous work [13, 56, 18, 16, 19, 15]. We are the first ones to propose how affine parameters help to learn the higher-order statistics of multiple domains. The respective affine parameters i.e. γ and β are dynamically controlled by learnable parameters, ρ' s whose sum will be 1 constrained by the softmax function (Equation *(6)*). The idea behind using multimodal adaptive normalization is that various features in the multimodal domain are correlated. Multimodal adaptive normalization opens the non-trivial path to capture the mutual dependence between various domains. Generally, various encoder architectures [42] are used to convert the various features of multiple domains into latent vectors, and then the concatenated vectors are fed to the decoder to model the mutual dependence and generate the required output. The proposed multimodal adaptive normalization helps in reducing the number of model parameters required to incorporate multimodal mutual dependence into the architecture.

In the multimodal adaptive normalization where we have used the pitch, energy and Audio Melspectrogram Features (AMF) (Figure 11) from audio domain & static image and Optical Flow (OF)/facial Keypoints Heatmap (KH) features from video domain (Figure 10) in the normalization to compute the different affine parameters in multimodal adaptive normalization setup. Multimodal adaptive normalization gives the flexibility of using various architectures namely 2D convolution, partial convolution and attention model for video related features and 1D convolution and the LSTM layer for audio features, as shown in Table 6.

(Equation *(5)*) shows the combined equation of the multimodal adaptive normalized output where x_{IN} is the instance normalized with mean and variance calculated across batch and channel.Various γ's and β's are modeled and linearly combined under an equation. The parameter ρ's is used to combine these parameters (Equation *(6)*). The value of ρ's is constrained to the range of [0, 1] by using the softmax function (Equation *(6)*).

$$y = \rho_1(\gamma_{Image}x_{IN} + \beta_{Image}) + \rho_2(\gamma_{OF/KH}x_{IN} + \beta_{OF/KH}) + \rho_3(\gamma_{AMF}x_{IN} + \beta_{AMF}) + \rho_4(\gamma_{pitch}x_{IN} + \beta_{pitch}) + \rho_5(\gamma_{energy}x_{IN} + \beta_{energy}) \quad (5)$$

$$\rho_1 + \rho_2 + \rho_3 + \rho_4 + \rho_5 = 1 \quad (6)$$

4.3 Feature extractor modules

This section consists of various feature extractor modules which extract the various features such as pitch, energy

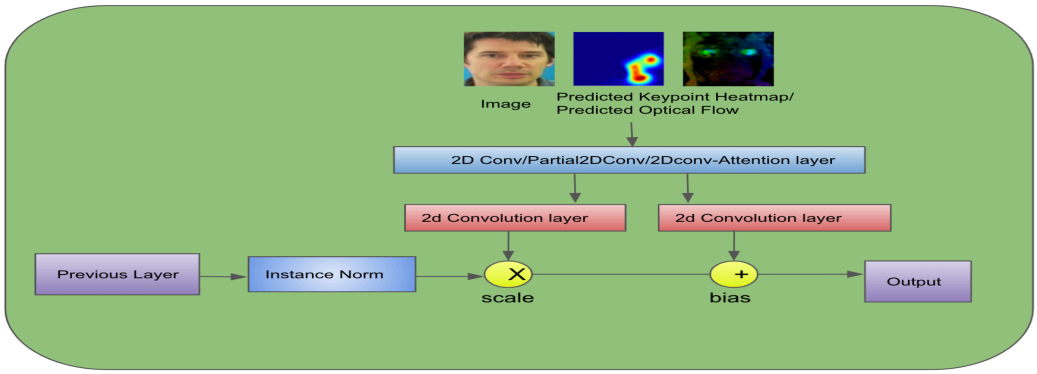


Fig. 10 – Architecture to calculate the affine parameters from video features in multimodal adaptive normalization

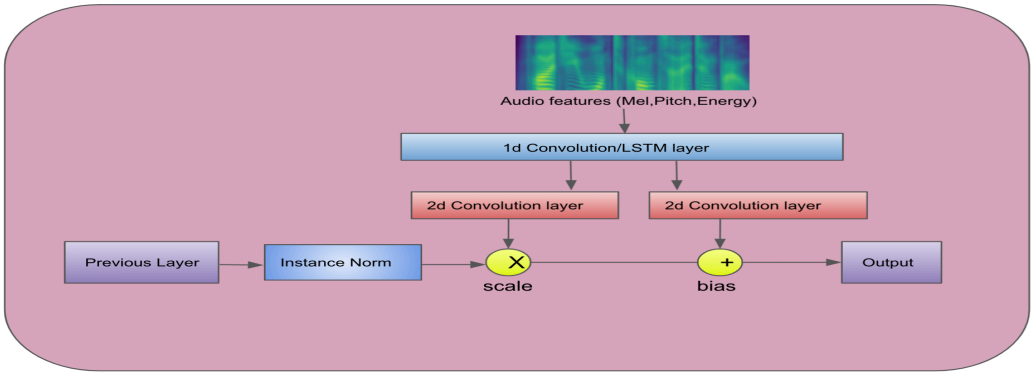


Fig. 11 – Architecture to calculate the affine parameters from audio features in multimodal adaptive normalization

and Audio Melspectrogram Features (AMF) from audio domain & static image and Optical Flow (OF)/facial Keypoints Heatmap (KH) features from video domain. There are various feature extractor modules such as keypoint heatmap predictor which predicts the keypoint heatmap having the information of various facial parts, e.g., upper left eyelid and nose bridge. Another module is Optical Flow Predictor which predicts the optical flow of the frame needed to maintain the temporal consistency in any video.

Fig. 7 shows that block level diagram of how the various features of the audio and video domain go into the multimodal adaptive normalisation through the affine parameters i.e, scale, γ and a shift, β .Fig. 10 shows that the video features such as predicted optical flow or keypoint heatmaps and single image go into a few layers of 2D convolution/2D partial convolution/2D convolution-attention layers to generate the corresponding γ's and β's and then go to the normalization layers of the generator. Fig. 11 shows that various features such as pitch, energy and melspectrogram go to the various layers of 1D convolution/LSTM to generate the corresponding γ's and β's.

4.3.1 Keypoint heatmap predictor

The predictor model is based on Hourglass architecture [57] that, from the input image, estimates K heatmaps $H_K \in [0, 1]^{H \times W}$, one for each keypoint, each of which represents the contour of a specific facial part, e.g., upper left eyelid and nose bridge. It captures the spatial configuration of all landmarks, and hence it captures pose, expression and shape information. We have used a pretrained model[2] to calculate the ground truth of heatmap and have applied mean square error loss between predicted heatmaps and ground truth. Fig. 12 shows the architectural diagram the of the keypoint heatmap predictor which takes the previous 5 frames and melspectrogram of audio signal and feed it to the model to predict the heatmaps of the frames. Hourglass architecture [57] is used after two layers of convolution which helps in generating better facial heatmaps.

In the experiments, we have used the 15 channel heatmaps and input and output sizes are (15, 96, 96). We have done the joint training of keypoint predictor architecture along with the generator architecture and fed the output of keypoint predictor architecture in the multimodal adaptive normalization network to learn the affine parameters and have optimized it with mean square error loss with the output of a pretrained model. The input of the keypoint predictor model is the previous 5 frames along with 256 audio mel spectrogram features which are concatenated along the channel axis. This is optimization with mean square error loss with a pretrained facial keypoint detection model.

[2] https://github.com/raymon-tian/hourglass-facekeypoints-detection

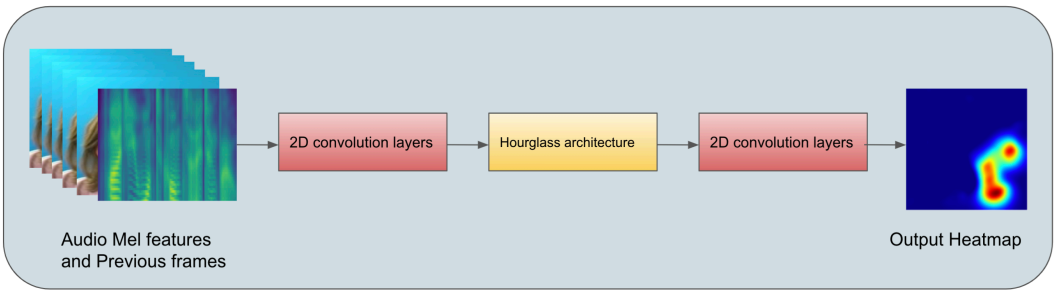


Fig. 12 – Keypoint heatmap predicted architecture

4.3.2 *Optical flow predictor*

The architecture is based on an encoder-decoder model (Fig. 13)to predict the optical flow of the next frame. We are giving the previous frames and current audio mel-spectrogram as an input to the model with KL loss and reconstruction loss. The pretrained model is then used in the generator to calculate the affine parameters. The input of the optical flow is previous 5 frames along with 256 audio melspectrogram features and is jointly trained along with the generator architecture and is optimized with mean square loss with the actual optical loss.

4.3.3 *Pitch extractor*

We extracted the pitch contour, F_0 using PyWorldVocoder tool [58] and quantized each frame to 256 possible values and encode them into a sequence of one-hot vectors as a pitch vector.

4.3.4 *Energy extractor*

We compute L2-norm of the amplitude of each Short-Time Fourier Transform (STFT) frame as the energy given by (Equation *(8)*) and then we add it to the expanded hidden sequence coming similar to pitch.

$$X(m,k) = \sum_{n=0}^{N-1} x[n-mH]w[n]exp(-2\pi ikn/N) \qquad (7)$$

where X(m,k) is the STFT of raw waudio waveform x[n] with window w[n] and m is the frame index , $k \in [0:K]$ and for every frame,m there are $K+1$ spectral vectors.

$$Energy(m) = \left(\sum_{k=0}^{K}(|x(m,k)|)^2\right)^{1/2} \qquad (8)$$

4.3.5 *Audio melspectrogram extractor*

We transfer the raw waveform into melspectrograms by setting the frame size and hop size to 1024 and 256 with respect to the sample rate of 22050 Hz.

4.4 Multiscale frame discriminator

We have used multiscale frame discriminator [59] to distinguish the fake and real image at the finer and coarser level. The class activation map-based layer is also used to distinguish the real or fake image by visualizing local and global attention maps. We have applied the adversarial loss (Equation *(14)*) on the information from the CAM output, n_{D_t} at different scale of the discriminator so that it will help the generator and discriminator to focus on local and global features and help in generating a more realistic image. This multiscale frame discriminator is based on Pix2PixHD [60].

$$L_{\text{cam}} = E_{y\sim P_t}[\log(n_{D_t}(y))] + E_{x\sim P_s}[\log(D(1-n_{D_t}(G(x))))] \qquad (9)$$

5. LOSSES

The proposed method is trained with different losses to generate realistic videos as explained below.

5.1 Adversarial loss

Adversarial loss is used to train the model to handle adversarial attacks and ensure the generation of high quality images for the video. The loss is defined as:

$$L_{\text{GAN}}(G,D) = E_{x\sim P_d}[\log(D(x))] + E_{z\sim P_z}[\log(D(1-G(z)))] \qquad (10)$$

where G tries to minimize this objective against an adversarial D that tries to maximize.

5.2 Reconstruction loss

Reconstruction loss [61] is used on the lower half of the image to improve the reconstruction in the mouth area. L1 loss is used for this purpose as described below:

$$L_{\text{RL}} = \sum_{n\epsilon[0,W]*[H/2,H]} (R_n - G_n) \qquad (11)$$

where, R_n and G_n are the real and generated frames respectively.

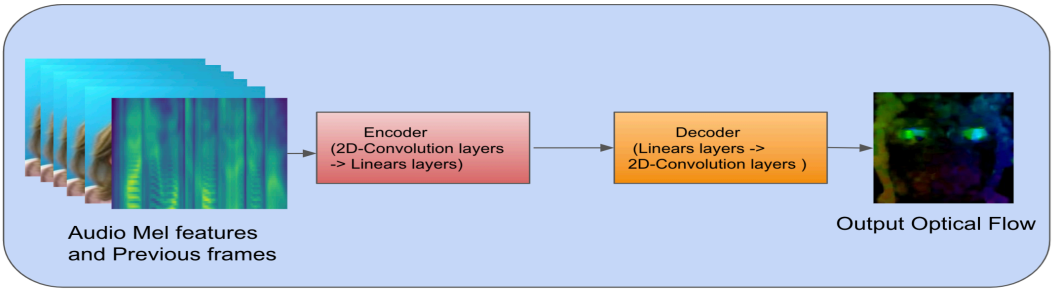


Fig. 13 – Optical flow predictor architecture

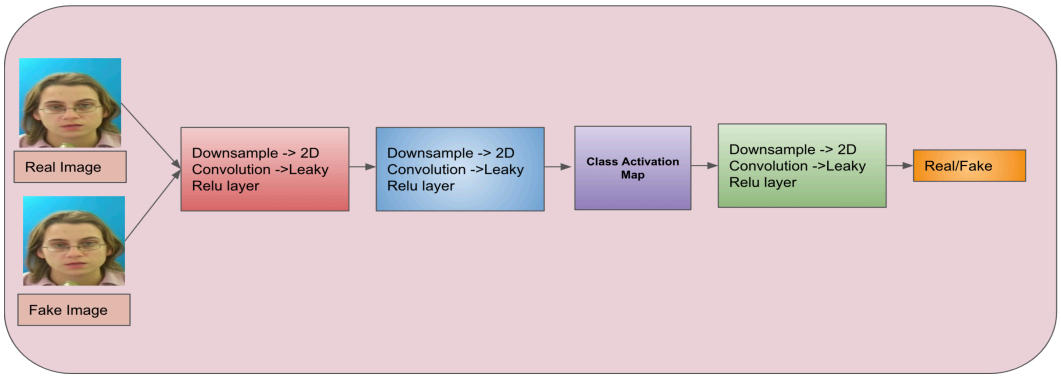


Fig. 14 – Discriminator architecture

5.3 Feature loss

Feature-matching loss [59] ensures the generation of natural-looking high quality frames. We take the L1 loss between generated images and real images for different scale discriminators and then sum it all. We extract features from multiple layers of the discriminator and learn to match these intermediate representations from the real and the synthesized image. This helps in stabilizing the training of the generator. The feature matching loss, $L_{FM}(G,D_k)$ is given by:

$$L_{\text{FM}}(G, D_{\text{k}}) = E_{(\text{x,z})} \sum_{n=1}^{T} [\frac{1}{N_{\text{i}}} ||D_{\text{k}}^{(i)}(x) - D_{\text{k}}^{(i)}(G(z))||_1] \quad (12)$$

where, T is the total number of layers and N_i denotes the number of elements in each layer.

5.4 Perceptual loss

The perceptual similarity metric is calculated between the generated frame and the real frame. This is done by using features of a VGG19 [62] model trained for ILSVRC classification and VGGFace [63] data set. The perceptual loss [64], (L_{PL}) is defined as:

$$L_{\text{PL}} = \lambda \sum_{n=1}^{N} [\frac{1}{M_{\text{i}}} ||F^{(i)}(x) - F^{(i)}(G(z))||_1] \quad (13)$$

where, λ is the weight for perceptual loss and $F^{(i)}$ is the ith layer of VGG19 network with M_i elements of VGG layer.

5.5 Class activation loss

We have used the class activation-based adversarial loss in the generator and discriminator which helps the model to learn local and global facial features and helps in cheek movement, blinks as well as image reconstruction.

$$L_{\text{cam}} = E_{\text{y}\sim \text{P}_{\text{t}}}[\log(n_{\text{D}_{\text{t}}}(y))] + E_{\text{x}\sim \text{P}_{\text{s}}}[\log(D(1 - n_{\text{D}_{\text{t}}}(G(x))))] \quad (14)$$

where n_{D_t} is the class activation-based logits from the real and fake image.

5.6 Mean square loss

We have optimized the keypoint heatmap predictor and optical flow predictor using mean square loss between the generated keypoint heatmap and pretrained model [65] and generated optical flow and ground truth farneback [66] optical flow output.

6. QUALITY OF EXPERIENCE (QOE)

In order to avoid the spectators from quitting, thus increasing the revenue , the proposed model is able to control the quality of experience. We derive our QoE model from [67]. Using subjective Mean Opinion Score (MOS) measurements, they derive QoE as a second degree function of the image PSNR and Frame Rate (FR), fitted to the MOS:

$$QoE = -8.97 + 0.056 \cdot \text{FR} + 0.41 \cdot \text{PSNR} - 0.0038 \cdot \text{PSNR}^2 - 0.001 \cdot \text{FR}^2 + 0.00079 \cdot \text{FR} \cdot \text{PSNR} \quad (15)$$

Knowing the average PSNR and frame size, we use this model to calculate each receiver's QoE at present and estimate their QoE in the future for different profiles.
The total QoE for each receiver, which aims to reflect their satisfaction with the whole video streaming experience, will be a function of the individual QoE corresponding to each player.
The QoE metric has several advantages:

- Due to erratic network connectivity or low bandwidth, the Quality of Experience (QoE) can be low. With the proposed model we can significantly improve the QOE by sending the audio signal and synthesizing the video at the receiver's end, thus improving the PSNR.
- The proposed video streaming pipeline helps in dynamically using the proposed video generation architecture when the quality of experience goes below the threshold PSNR level. It thus gives the flexibility to control the QOE based on the compute resource, bandwidth availability and importance of the speaker in the video conference.

7. EXPERIMENTS

7.1 Implementation details

7.1.1 Data sets

We have used the GRID [68], LOMBARD GRID [69], Crema-D [70] and VoxCeleb2 [71] data sets for the experiments and evaluation of different metrics.

GRID: GRID [68] is a large multi-talker audiovisual sentence corpus to support joint computational-behavioral studies in speech perception. In brief, the corpus consists of high quality audio and video (facial) recordings of 1000 sentences spoken by each of the 34 talkers (18 male, 16 female). Sentences are of the form "put red at G9 now".

LOMBARD GRID: Lombard GRID [69] is a bi-view audiovisual Lombard speech corpus that can be used to support joint computational-behavioral studies in speech perception. The corpus includes 54 talkers, with 100 utterances per talker (50 Lombard and 50 plain utterances). This data set follows the same sentence format as the audiovisual GRID corpus, and can thus be considered as an extension of that corpus.

CREMA-D: CREMA-D [70] is a data set of 7,442 original clips from 91 actors. These clips were from 48 male and 43 female actors between the ages of 20 and 74 coming from a variety of races and ethnicities (African American, Asian, Caucasian, Hispanic, and Unspecified). Actors spoke from a selection of 12 sentences. The sentences were presented using one of six different emotions (Anger, Disgust, Fear, Happy, Neutral, and Sad) and four different emotion levels (Low, Medium, High, and Unspecified).

VOXCELEB2: VoxCeleb2 [71] is a very large-scale audio-visual speaker recognition data set collected from open-source media. Voxceleb2 contains over 1 million utterances for over 6,000 celebrities, extracted from videos uploaded to YouTube. The data set is fairly gender balanced, with 61 % of the speakers male.

7.1.2 Preprocessing steps

Videos are processed at 25fps and frames are resized into 256X256 size and audio features are processed at 16khz. The ground truth of optical flow is calculated using the farneback optical flow algorithm [66]. To extract the keypoint heatmaps, we have used the pretrained hourglass face keypoint detection [65]. Every audio frame is centered around a single video frame. To do that, zero padding is done before and after the audio signal and use the following formula for the stride.

$$stride = \frac{\text{audio sampling rate}}{\text{video frames per sec}}$$

We extract the pitch, F0 using using PyWorldVocoder [72] from the raw waveform with the frame size of 1024 and hop size of 256 sampled at 16khz to obtain the pitch of each frame and compute the L2-norm of the amplitude of each STFT frame as the energy. We quantize the F0 and energy of each frame to 256 possible values and encode them into a sequence of one-hot vectors as p and e respectively and then feed the value of p, e and 256 dimensional melspectrogram features in the proposed normalization method.

7.1.3 Metrics

To quantify the quality of the final generated video, we use the following metrics. Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM), Cumulative Probability Blur Detection (CPBD) and Average Content Distance (ACD). PSNR, SSIM, and CPBD measure the quality of the generated image in terms of the presence of noise, perceptual degradation, and blurriness respectively. ACD [44] is used for the identification of the speaker from the generated frames by using OpenPose [73]. Along with image quality metrics, we also calculate Word Error Rate (WER) using pretrained LipNet architecture [74], Blinks/sec using [75] and Landmark Distance (LMD) [76] to evaluate our performance of speech recognition, eye-blink reconstruction and lip reconstruction respectively.

1. PSNR- Peak Signal to Noise Ratio: It computes the peak signal to noise ratio between two images. The higher the PSNR the better the quality of the reconstructed image.

2. SSIM- Structural Similarity Index: It is a perceptual metric that quantifies image quality degradation. The larger the value the better the quality of the reconstructed image.

3. CPBD- Cumulative Probability Blur Detection: It is a perceptual based no-reference objective image sharpness metric based on the cumulative probability of blur detection developed at the image.

4. WER- Word error rate: It is a metric to evaluate the performance of speech recognition in a given video. We have used LipNet architecture [74] which is pretrained on the GRID data set for evaluating the WER. On the GRID data set, Lipnet achieves 95.2 percent accuracy which surpasses the experienced human lipreaders.

5. ACD- Average Content Distance ([44]): It is used for the identification of speakers from the generated frames using OpenPose [73]. We have calculated the Cosine distance and Euclidean distance of representation of the generated image and the actual image from Openpose. The distance threshold for the OpenPose model should be 0.02 for Cosine distance and 0.20 for Euclidean distance [77]. The lesser the distances the more similar the generated and actual images.

6. LMD - Landmark Distance ([76]): To ensure realistic and accurate lip movement, ensuring good performance on speech recognition we use this metric. We calculate the landmark points [78] on both real and generated images at the scale of 256*256 and use the lip region points i.e., points 49-68 and call then as LR and LF respectively. LR refers to lip region from ground truth image and LF corresponds to lip region from generated/fake image. T is the number of frames. Then, we calculate the euclidean distance between each corresponding pairs of landmarks on LR and LF. The LMD is defined as:

$$LMD = \frac{1}{T} * \frac{1}{P} \sum_{t=1}^{T} \sum_{p=1}^{P} ||LR_{t,p} - LF_{t,p}|| \qquad (16)$$

7. Blinks/sec: To capture the blinks in the video, we are calculating the blinks/sec so that we can better understand the quality of animated videos. Fig. 15 shows the 6 points which are used to calculate the Eye Aspect Ratio (EAR) given in Equation *(17)*. We have used SVM and eye landmarks along with Eye Aspect Ratio (EAR) used in Real-Time Eye Blink Detection using Facial Landmarks [75] to detect the blinks in a video.

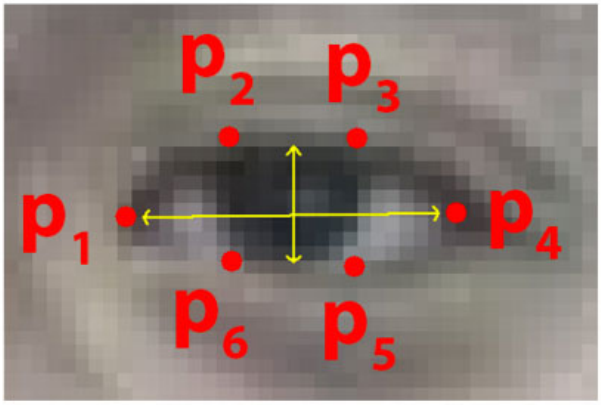


Fig. 15 – Description of the 6 eye points

$$EAR = \frac{||p2 - p6|| + ||p3 - p5||}{||p1 - p4||} \qquad (17)$$

7.1.4 *Training and inference*

Our model is implemented in pytorch and takes approximately 7 days to run on 4 Nvidia V100 GPUs for training. In the training stage, the model is trained with a multiscale frame discriminator with adversarial loss (Equation *(6)*), class activation map-based loss (Equation *(14)*) and feature matching loss (Equation *(12)*). The generator is trained with adversarial loss (Equation *(6)*), class activation map-based loss (Equation *(14)*), reconstruction loss (Equation *(11)*), perceptual loss (Equation 5.4), and key-point predictor/optical flow-based mean square error loss are also used to ensure generation of natural-looking, high quality frames.
We have taken the Adam optimizer [79] with learning rate = 0.002 and β_1= 0.0 and β_2 = 0.90 for the generator and discriminators.

7.2 Implementation results

7.2.1 *Quantitative results*

Tables 1,2,3,4 compare the proposed method with its competitors and shows better SSIM, PSNR, CPBD, Word Error Rate (WER), blinks/sec and LMD on GRID [68], Crema-D [70], GRID-Lombard [69] and Voxceleb2 [71] data sets, suggesting highly expressive and realistic video synthesis. The proposed method has shown superior results on most of the metrics in all the mentioned data sets.

7.2.2 *QoE metric*

We have computed the QoE metric for various data sets using Equation *(15)*. For our experiments we have taken the 25fps for synthesizing the video. Table 5 shows the QoE metric for various data sets when synthesizing the video from audio using the proposed method. The higher the QoE metric is, the better the model is. We can dynamically control the QoE based on the need of the video conferencing and during erratic network conditions.

Table 1 – Comparison of the proposed method(MAN-keypoint and MAN-optical) with other previous works for GRID data set

Method	SSIM↑	PSNR↑	CPBD↑	WER↓	ACD-C↓	ACD-E↓	blinks/sec	LMD↓
FOMM[80]	0.833	26.72	0.214	38.21	0.004	0.088	0.56	**0.718**
OneShotA2V[4]	0.881	28.571	0.262	27.5	0.005	0.09	0.15	0.91
RSDGAN[42]	0.818	27.100	0.268	**23.1**	-	$1.47x10^{-4}$	0.39	-
Speech2Vid[39]	0.720	22.662	0.255	58.2	0.007	$1.48x10^{-4}$	-	-
ATVGnet[48]	0.83	**32.15**	-	-	-	-	-	1.29
X2face[43]	0.80	29.39	-	-	-	-	-	1.48
CascadedGAN[46]	0.81	27.1	0.26	23.1	-	$1.47x10^{-4}$	0.45	-
MAN-optical	**0.908**	29.78	**0.272**	23.7	**0.005**	**$1.41x10^{-4}$**	0.45	**0.77**
MAN-keypoint	0.887	29.01	0.269	25.2	0.006	$1.41x10^{-4}$	0.48	0.80

Table 2 – Comparison of the proposed method(MAN-keypoint and MAN-optical) with other previous works for CREMA-D data set

Method	SSIM↑	PSNR↑	CPBD↑	WER↓	ACD-C↓	ACD-E↓	blinks/sec	LMD↓
FOMM[80]	0.654	20.74	0.186	NA	0.007	0.12	-	1.041
OneShotA2V[4]	0.773	24.057	0.184	NA	0.006	0.96	-	0.632
RSDGAN[42]	0.700	23.565	0.216	NA	-	$1.40x10^{-4}$	-	-
Speech2Vid[39]	0.700	22.190	0.217	NA	0.008	$1.73x10^{-4}$	-	-
MAN-optical	0.826	27.723	0.224	NA	0.004	$1.62x10^{-4}$	-	0.592
MAN-keypoint	**0.841**	**28.01**	**0.228**	NA	**0.003**	**$1.38x10^{-4}$**	-	**0.51**

Table 3 – Comparison of the proposed method(MAN-keypoint and MAN-optical) with other previous works for GRID Lombard data set

Method	SSIM↑	PSNR↑	CPBD↑	WER↓	ACD-C↓	ACD-E↓	blinks/sec	LMD↓
FOMM[80]	0.804	22.97	0.381	NA	0.003	0.078	0.37	1.09
OneShotA2V[4]	0.922	28.978	0.453	NA	0.002	0.064	0.1	0.61
Speech2Vid[39]	0.782	26.784	0.406	NA	0.004	0.069	-	0.581
MAN-optical	0.895	26.94	0.43	NA	0.001	0.048	0.21	0.588
MAN-keypoint	**0.931**	**29.62**	**0.492**	NA	**0.001**	**0.046**	**0.31**	textbf0.563

Table 4 – Comparison of the proposed method(MAN-keypoint and MAN-optical) with other previous works for VOXCELEB2 data set

Method	SSIM↑	PSNR↑	CPBD↑	WER↓	ACD-C↓	ACD-E↓	blinks/sec	LMD↓
OneShotA2V[4]	0.698	20.921	0.103	NA	0.011	0.096	0.05	0.72
MAN-optical	0.714	21.94	0.118	NA	0.008	0.067	0.21	0.65
MAN-keypoint	**0.732**	**22.41**	**0.126**	NA	**0.004**	**0.058**	**0.28**	**0.47**

Table 5 – Average QoE on proposed method

Method	QoE ↑
MAN-optical(GRID)	**1.232**
MAN-keypoint(GRID)	1.074
MAN-optical(GRID-Lombard)	0.624
MAN-keypoint(GRID-Lombard)	**1.20**
MAN-optical(CREMA-D)	0.797
MAN-keypoint(CREMA-D)	**0.860**
MAN-optical(VoxCeleb2)	-0.595
MAN-keypoint(VoxCeleb2)	**-0.472**

7.2.3 Qualitative results

Expressive aspect: Fig. 16 displays the lip synchronized frames of a speaker speaking the word 'bin' and 'please' as well as the blinking of the eyes. Fig. 17 shows the comparison of the proposed model with previous work [52] where the proposed model shows better image reconstruction and lip synchronization. The generated videos are given at [3].

Fig. 16 – Top: The speaker speaking the word 'bin' , Middle : The speaker speaking the word 'please', Bottom: The speaker blinking his eyes

[3] https://sites.google.com/view/itu2021

Fig. 17 – Top: Actual frames of voxceleb2 [71] data set , Middle : Predicted frames from proposed method, Bottom: Predicted frame from [52]

Architecture analysis: Fig. 18 shows the optical flow map and class activation-based heatmaps at different expressions of the speakers while speaking. The optical flow map has a different color while speaking and the opening of eyes as compared to closing of mouth and the blinking of eyes. The CAM-based heatmap shows the attention regions in the heatmap which captures the local as well as global features during video generation. The bottom part of the figure shows the predicted keypoints from the keypoint predictor calculated using the max operator to find the coordinates of the maximum value in each predicted heatmap (15, 96, 96).

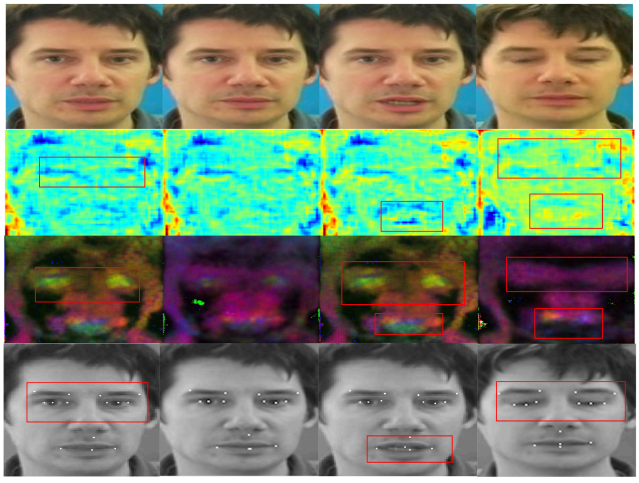


Fig. 18 – Top: The speaker with different expressions, Middle1 : CAM-based attention map, Middle2: Predicted optical flow from the optical flow generator architecture, Bottom: Predicted Key-points from Keypoint predictor architecture

Eye blinks: The average human blink rate of 0.28 blinks/second, especially when considering that the blink rate increases to 0.4blinks/second during conversation. Fig. 19 shows the sharp decline in the eye aspect ratio [75] at the centre which justifies the generation of blinks in the predicted videos. Table 1 shows the blinks/sec of 0.45 on the GRID data set.

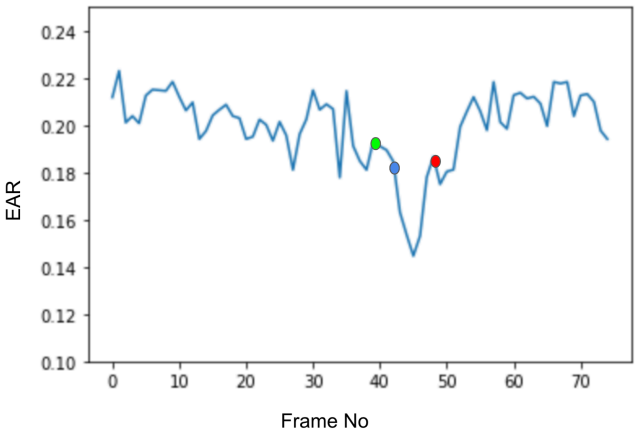


Fig. 19 – A blink is detected at the location where a sharp drop occurs in the EAR signal (blue dot). We consider the start (green dot) and end (red dot) of the blink to correspond to the peaks on either side of the blink location (Color figure online).

Comparison with video to video synthesis architecture: We have compared the proposed method with First Order Motion Model (FOMM) for image animation [80] on the GRID data set which generates the video sequences so that an object in a source image is animated according to the motion of a driving video. The comparison is done to see how effectively driving audio signals instead of driving video helps in reconstructing the expressive video as shown in Fig. 20. Tables 1, 2, 3 compare the various metrics between FOMM and the proposed model and show better image reconstruction metrics (SSIM, PSNR, CPBD,LMD) and WER but FOMM has more blinks/sec as compared to the proposed method. The reason for better WER is a limited number of utterances in the GRID data set and faster speaking style of the speaker which the proposed method is better able to capture as compared to FOMM.

7.3 Ablation study

To study the effectiveness of the proposed model and its novel multimodal adaptive normalization approach. We have shown that multimodal adaptive normalization is flexible to incorporate the various architecture shown in Section 7.3.1 and its effectiveness in the generation of realistic videos.We have also studied the incremental effect of audio and video features such as optical flow, melspectrogram, pitch and energy in Section 7.3.2.

Fig. 20 – Top: Actual frames of speaker of GRID data set. Middle: Predicted frames from proposed method with keypoints predicted from keypoint predictor. Bottom: Predicted frames from the FOMM method [80]

7.3.1 Network analysis in multimodal adaptive normalization

We have done the ablation study on three architectures, namely 2D convolution, partial 2D convolution [81, 82] and 2D convolution+Efficient Channel Attention (ECA) [83] for extracting video features and two architectures namely 1D convolution and LSTM for audio features as shown in Fig. 10 and Fig. 11 to study its effect on multimodal adaptive normalization with optical flow predictor in the proposed method. Table 6 shows that 2DConv+ECA+LSTM has improved the reconstruction metrics such as SSIM, PSNR and CPBD as well as word error rate and blinks/sec as compared to other networks. The image quality reduced with the use of partial 2D convolution which demonstrates that since the predicted optical flow is dense, holes in the optical flow has some spatial relation with other regions which are better captured by other networks.

Table 6 – Ablation study of different networks of multimodal adaptive normalization on GRID data set

Method	SSIM↑	PSNR↑	CPBD↑	blinks/sec	WER↓
2DConv+1dConv	0.875	28.65	0.261	0.35	25.6
Partial2DConv+1dConv	0.803	28.12	0.256	0.15	29.4
2DConv+ECA+1dConv	0.880	29.11	0.263	0.42	23.9
2DConv+LSTM	0.896	29.25	0.086	0.260	24.1
Partial2DConv+LSTM	0.823	28.12	0.258	0.12	28.3
2DConv+ECA+LSTM	0.908	29.78	0.272	0.45	23.7

7.3.2 Incremental effect of multimodal adaptive normalization

We study the incremental effect of multimodal adaptive normalization of the proposed model with the Optical Flow Predictor (OFP) and 2DConv+ECA+LSTM combination in multimodal attention normalization on a GRID data set. Table 7 shows the impact of the addition of melspectrogram features, pitch, predicted optical flow in multimodal adaptive normalization. The base model consists of generator and discriminator architecture with a static image in the adaptive normalization.

Table 7 – Incremental study of multimodal adaptive normalization on GRID data set

Method	SSIM↑	PSNR↑	CPBD↑	blinks/sec	WER↓
Base Model(BM)	0.776	27.99	0.213	0.02	57.9
BM + OFP+mel	0.878	28.43	0.244	0.38	27.4
BM + OFP+mel+pitch	0.881	28.57	0.264	0.41	24.1
BM+OFP+mel+pitch+energy	0.908	29.78	0.272	0.45	23.7

8. PSYCHOPHYSICAL ASSESSMENT

Results are visually rated (on a scale of 5) individually by 25 persons, on three aspects, lip synchronization, eye blinks and eyebrow raises and quality of video on a GRID data set. The subjects were shown anonymous videos at the same time for the different audio clips for side-by-side comparison. Table 8 clearly shows that MAN-based proposed architecture performs significantly better in quality and lip synchronization which is of prime importance in videos.

Table 8 – Psychophysical evaluation (in percentages) based on users rating on GRID daatset

Method	Lip-Sync↑	Eye-blink↑	Quality↑
MAN	91.8	**90.5**	**79.6**
OneShotA2V[4]	90.8	88.5	76.2
RSDGAN[42]	**92.8**	90.2	74.3
Speech2Vid[39]	90.7	87.7	72.2

9. TURING TEST

To test the naturalism of the generated videos we conduct an online Turing test on a GRID data set [4]. Each test consists of 20 questions with 10 fake and 10 real videos. The user is asked to label a video real or fake based on the aesthetics and naturalism of the video. Approximately 300 user data is collected and their score of the ability to spot fake video is displayed in Fig. 21.

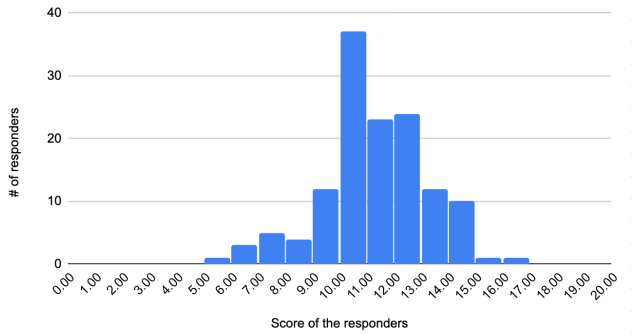


Fig. 21 – Distribution of user scores for the online Turing test

[4] https://forms.gle/DM1DRcTToQFvUpTa7

10. CONCLUSIONS AND FUTURE WORK

We have seen that the proposed video streaming pipeline with multimodal adaptive normalization-based architecture to generate the video helps in reducing the network bandwidth in unreliable Internet conditions. The proposed video streaming pipeline can control the quality of experience based on the compute resource and bandwidth availability. It helps in data privacy by synthesizing the video on the avatar of that person.

Although this implementation provides a proof of concept for the underlying idea, further work is needed to implement a full body low latency, low bandwidth video streaming environment to further enhance the quality of experience. With the rapid improvement of hardware capabilities in mobiles and personal computers, this is unlikely to be a major obstacle. As evidenced by the recent announcement of the NVIDIA Maxine project [3], hurdles are surmountable and these ideas can be translated into a practical system that provides immense gains over the conventional methods.

REFERENCES

[1] Sadjad Fouladi, John Emmons, Emre Orbay, Catherine Wu, Riad S. Wahby, and Keith Winstein. "Salsify: Low-Latency Network Video through Tighter Integration between a Video Codec and a Transport Protocol". In: *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*. Renton, WA: USENIX Association, Apr. 2018, pp. 267–282. ISBN: 978-1-939133-01-4. URL: https://www.usenix.org/conference/nsdi18/presentation/fouladi.

[2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[3] Sid Sharma. "AI Can See Clearly Now: GANs Take the Jitters Out of Video Calls". In: *NVIDIA Blog* (Aug. 2020).

[4] Neeraj Kumar, Srishti Goel, Ankur Narang, and Mujtaba Hasan. "Robust One Shot Audio to Video Generation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2020.

[5] Johannes L. Schönberger and Jan-Michael Frahm. "Structure-from-Motion Revisited". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 4104–4113.

[6] S. Ma, Xinfeng Zhang, Chuanmin Jia, Zhenghui Zhao, S. Wang, and Shanshe Wang. "Image and Video Compression With Neural Networks: A Review". In: *IEEE Transactions on Circuits and Systems for Video Technology* 30 (2020), pp. 1683–1698.

[7] Chaochao Lu and X. Tang. "Surpassing Human-Level Face Verification Performance on LFW with GaussianFace". In: *AAAI*. 2015.

[8] Z. Huang, Xiaowei Zhao, S. Shan, R. Wang, and X. Chen. "Coupling Alignments with Recognition for Still-to-Video Face Recognition". In: *2013 IEEE International Conference on Computer Vision* (2013), pp. 3296–3303.

[9] N. Kumar, P. Belhumeur, and S. Nayar. "FaceTracer: A Search Engine for Large Collections of Images with Faces". In: *ECCV*. 2008.

[10] I. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, S. Ozair, Aaron C. Courville, and Yoshua Bengio. "Generative Adversarial Nets". In: *NIPS*. 2014.

[11] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: (Feb. 2015).

[12] Nitish Srivastava, Geoffrey E. Hinton, A. Krizhevsky, Ilya Sutskever, and R. Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *J. Mach. Learn. Res.* 15 (2014), pp. 1929–1958.

[13] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Instance Normalization: The Missing Ingredient for Fast Stylization". In: (July 2016).

[14] Xun Huang and Serge Belongie. "Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization". In: Oct. 2017, pp. 1510–1519. DOI: 10.1109/ICCV.2017.167.

[15] Hyeonseob Nam and Hyo-Eun Kim. *Batch-Instance Normalization for Adaptively Style-Invariant Neural Networks*. May 2018.

[16] Junho Kim, Minjae Kim, Hyeon-Woo Kang, and Kwanghee Lee. *U-GAT-IT: Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization for Image-to-Image Translation*. July 2019.

[17] Jimmy Ba, Jamie Kiros, and Geoffrey Hinton. "Layer Normalization". In: (July 2016).

[18] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. "Semantic Image Synthesis with Spatially-Adaptive Normalization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

[19] Arun Mallya, Ting-Chun Wang, Karan Sapra, and Ming-Yu Liu. *World-Consistent Video-to-Video Synthesis*. July 2020.

[20] Peter de Rivaz and Jack Haughton. "Av1 bitstream & decoding process specification". In: *The Alliance for Open Media* (2018), p. 182.

[21] *Versatile Video Coding (VVC)*. https://jvet.hhi.fraunhofer.de/. Accessed: 2020-10-27.

[22] Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, and Matthias Grundmann. "Blazeface: Sub-millisecond neural face detection on mobile gpus". In: *arXiv preprint arXiv:1907.05047* (2019).

[23] Yury Kartynnik, Artsiom Ablavatski, Ivan Grishchenko, and Matthias Grundmann. "Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs". In: *arXiv preprint arXiv:1907.06724* (2019).

[24] Adrian Bulat and Georgios Tzimiropoulos. "How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks)". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1021–1030.

[25] Ivan Grishchenko, Artsiom Ablavatski, Yury Kartynnik, Karthik Raveendran, and Matthias Grundmann. "Attention Mesh: High-fidelity Face Mesh Prediction in Real-time". In: *arXiv preprint arXiv:2006.10962* (2020).

[26] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. "BlazePose: On-device Real-time Body Pose tracking". In: *arXiv preprint arXiv:2006.10204* (2020).

[27] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. "Deep high-resolution representation learning for human pose estimation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019, pp. 5693–5703.

[28] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. "Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 269–286.

[29] Peter Eisert and Bernd Girod. "Analyzing facial expressions for virtual conferencing". In: *IEEE Computer Graphics and Applications* 18.5 (1998), pp. 70–78.

[30] Peter Eisert. "MPEG-4 facial animation in video analysis and synthesis". In: *International Journal of Imaging Systems and Technology* 13.5 (2003), pp. 245–256.

[31] A. Simons and Stephen Cox. "Generation of mouthshapes for a synthetic talking head". In: *Proceedings of the Institute of Acoustics, Autumn Meeting* (Jan. 1990).

[32] FirstName Alpher, FirstName Fotheringham-Smythe, and FirstName Gamow. "Can a machine frobnicate?" In: *Journal of Foo* 14.1 (2004), pp. 234–778.

[33] Andreea Stef, Kaveen Perera, Hubert Shum, and Edmond Ho. "Synthesizing Expressive Facial and Speech Animation by Text-to-IPA Translation with Emotion Control". In: Dec. 2018, pp. 1–8. DOI: 10.1109/SKIMA.2018.8631536.

[34] Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. "Audio-driven facial animation by joint end-to-end learning of pose and emotion". In: *ACM Transactions on Graphics* 36 (July 2017), pp. 1–12. DOI: 10.1145/3072959.3073658.

[35] Sarah Taylor, Moshe Mahler, Barry-John Theobald, and Iain Matthews. "Dynamic units of visual speech". In: July 2012, pp. 275–284.

[36] Pif Edwards, Chris Landreth, Eugene Fiume, and Karan Singh. "JALI: an animator-centric viseme model for expressive lip synchronization". In: *ACM Transactions on Graphics* 35 (July 2016), pp. 1–11. DOI: 10.1145/2897824.2925984.

[37] Wesley Mattheyses and Werner Verhelst. "Audiovisual speech synthesis: An overview of the state-of-the-art". In: *Speech Communication* 66 (Nov. 2014). DOI: 10.1016/j.specom.2014.11.001.

[38] Supasorn Suwajanakorn, Steven Seitz, and Ira Kemelmacher. "Synthesizing Obama: learning lip sync from audio". In: *ACM Transactions on Graphics* 36 (July 2017), pp. 1–13. DOI: 10.1145/3072959.3073640.

[39] Joon Son Chung, Amir Jamaludin, and Andrew Zisserman. "You said that?" In: *British Machine Vision Conference*. 2017.

[40] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. "Temporal Generative Adversarial Nets with Singular Value Clipping". In: Oct. 2017. DOI: 10.1109/ICCV.2017.308.

[41] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. "Generating Videos with Scene Dynamics". In: (Sept. 2016).

[42] Konstantinos Vougioukas, Stavros Petridi, and Maja Pantic. "End-to-End Speech-Driven Facial Animation with Temporal GANs". In: *Journal of Foo* 14.1 (2004), pp. 234–778.

[43] O. Wiles, A.S. Koepke, and A. Zisserman. "X2Face: A network for controlling face generation by using images, audio, and pose codes". In: *European Conference on Computer Vision*. 2018.

[44] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. "MoCoGAN: Decomposing motion and content for video generation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 1526–1535.

[45] R. Yi, Zipeng Ye, J. Zhang, H. Bao, and Yongjin Liu. "Audio-driven Talking Face Video Generation with Learning-based Personalized Head Pose". In: *arXiv: Computer Vision and Pattern Recognition* (2020).

[46] Dipanjan Das, Sandika Biswas, Sanjana Sinha, and Brojeshwar Bhowmick. "Speech-Driven Facial Animation Using Cascaded GANs for Learning of Motion and Texture". In: (Oct. 2019).

[47] Chelsea Finn, P. Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *ICML*. 2017.

[48] Lele Chen, Ross Maddox, Zhiyao Duan, and Chenliang Xu. *Hierarchical Cross-Modal Talking Face Generationwith Dynamic Pixel-Wise Loss*. May 2019.

[49] Lele Chen, Zhiheng Li, Ross Maddox, Zhiyao Duan, and Chenliang Xu. "Lip Movements Generation at a Glance". In: July 2018.

[50] Hang Zhou, Y. Liu, Z. Liu, Ping Luo, and X. Wang. "Talking Face Generation by Adversarially Disentangled Audio-Visual Representation". In: *AAAI*. 2019.

[51] K Prajwal, Rudrabha Mukhopadhyay, Vinay Namboodiri, and C Jawahar. *A Lip Sync Expert Is All You Need for Speech to Lip Generation In The Wild*. Aug. 2020.

[52] Hao Zhu, Huaibo Huang, Y. Li, A. Zheng, and R. He. "Arbitrary Talking Face Generation via Attentional Audio-Visual Coherence Learning." In: *arXiv: Computer Vision and Pattern Recognition* (2020).

[53] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.

[54] Alireza M. Javid, Sandipan Das, M. Skoglund, and S. Chatterjee. "A ReLU Dense Layer to Improve the Performance of Neural Networks". In: *ICASSP*. 2021.

[55] B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba. "Learning Deep Features for Discriminative Localization." In: *CVPR* (2016).

[56] Dmitry Nikitko. *stylegan-encoder. https://github.com/Puzer/stylegan-encoder*. 2019.

[57] Gary Storey, Ahmed Bouridane, Richard Jiang, and Chang-Tsun Li. "Atypical Facial Landmark Localisation with Stacked Hourglass Networks: A Study on 3D Facial Modelling for Medical Diagnosis". In: Jan. 2020, pp. 37–49. ISBN: 978-3-030-32582-4. DOI: 10.1007/978-3-030-32583-1_3.

[58] PyWORLD. "https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder". In: (2019).

[59] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[60] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[61] Yanchun Li, Nanfeng Xiao, and Wanli Ouyang. "Improved Generative Adversarial Networks with Reconstruction Loss". In: *Neurocomputing* 323 (Oct. 2018). DOI: 10.1016/j.neucom.2018.10.014.

[62] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *arXiv 1409.1556* (Sept. 2014).

[63] Wang Mei and Weihong Deng. "Deep Face Recognition: A Survey". In: (Apr. 2018).

[64] Alexandre Alahi Justin Johnson and Li Fei-Fei. "Perceptual Losses for Real-Time Style Transfer and Super-Resolution". In: (2016).

[65] facial keypoint detection. "https://github.com/raymon-tian/hourglass-facekeypoints-detection". In: (2017).

[66] Gunnar Farnebäck. "Two-Frame Motion Estimation Based on Polynomial Expansion". In: vol. 2749. June 2003, pp. 363–370. DOI: 10.1007/3-540-45103-X_50.

[67] Saman Zadtootaghaj, Steven Schmidt, and Sebastian Möller. "Modeling gaming QoE: Towards the impact of frame rate and bit rate on cloud gaming". In: *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2018, pp. 1–6.

[68] FirstName Alpher and FirstName Fotheringham-Smythe. "Frobnication revisited". In: *Journal of Foo* 13.1 (2003), pp. 234–778.

[69] Najwa Alghamdi, Steve Maddock, Ricard Marxer, Jon Barker, and Guy J. Brown. *A corpus of audio-visual Lombard speech with frontal and profile view, The Journal of the Acoustical Society of America 143, EL523 (2018); https://doi.org/10.1121/1.5042758*. 2018.

[70] Houwei Cao, David Cooper, Michael Keutmann, Ruben Gur, Ani Nenkova, and Ragini Verma. "CREMA-D: Crowd-sourced emotional multimodal actors dataset". In: *IEEE transactions on affective computing* 5 (Oct. 2014), pp. 377–390. DOI: 10.1109/TAFFC.2014.2336244.

[71] J. S. Chung, A. Nagrani, and A. Zisserman. "VoxCeleb2: Deep Speaker Recognition". In: *INTERSPEECH*. 2018.

[72] PyWorldVocoder. "https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder". In: (2017).

[73] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields". In: *arXiv preprint arXiv:1812.08008*. 2018.

[74] Yannis M Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas. "LipNet: End-to-End Sentence-level Lipreading". In: *GPU Technology Conference* (2017). URL: https://github.com/Fengdalu/LipNet-PyTorch.

[75] T. Soukupová and Jan Cech. "Real-Time Eye Blink Detection using Facial Landmarks". In: 2016.

[76] Lele Chen, Zhiheng Li, Ross K. Maddox, Zhiyao Duan, and Chenliang Xu. "Lip Movements Generation at a Glance". In: (2018). arXiv: 1803.10404 [cs.CV].

[77] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris Metaxas. *Learning to Forecast and Refine Residual Motion for Image-to-Video Generation*. 2018.

[78] D.E. King. "Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research". In: (2009).

[79] Diederik Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations* (Dec. 2014).

[80] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. "First Order Motion Model for Image Animation". In: (2020). arXiv: 2003.00196 [cs.CV].

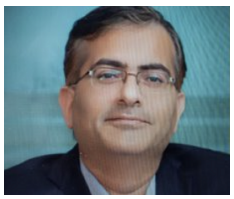
[81] Guilin Liu, Kevin J. Shih, Ting-Chun Wang, Fitsum A. Reda, Karan Sapra, Zhiding Yu, Andrew Tao, and Bryan Catanzaro. "Partial Convolution based Padding". In: *arXiv preprint arXiv:1811.11718*. 2018.

[82] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. "Image Inpainting for Irregular Holes Using Partial Convolutions". In: *The European Conference on Computer Vision (ECCV)*. 2018.

[83] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. *ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks*. Oct. 2019.

AUTHORS

Neeraj Kumar is currently working as Senior Machine Learning Scientist at Hike Private Limited, India. He is also a PHD student at Indian Institute of Technology, Delhi, India and completed his B-tech(ECE) from Indian Institute of Technology, Kharagpur, India. His current area of interest is multimodal AI along with the foundation and theoretical aspect of machine learning. He has published papers in top conferences such as CVPR, Interspeech & IJCAI.



Dr. Ankur Narang is currently the VP of AI technologies at Hike Private Limited. He holds a B.Tech. & Ph.D. from IIT Delhi in CSE and has 40+ publications in top international computer science machine learning conferences and journals, along with 15 granted US patents. He was one amongst Top 10 data scientists in India in 2017 (Analytics India Mag) in recognition of solid scientific and industry contributions to the field of data science and artificial intelligence. In 2018, he was given the Top 50 Analytics Award at the MachineCon conference in recognition of exemplary leadership and contributions to ML/AI (Analytics India Magazine). He was also conferred Top 100 Innovative CIO Award in 2019, for distinguished leadership in innovative technologies based digital transformation (CIO Axis). In 2002, he was awarded Sun Microsystem's prestigious "Innovation Leadership Award" for significant contributions to the Hardware Acceleration Project.

Dr. Brejesh Lall is a professor at Electrical Engineering Department at IIT Delhi and has contributed to research & teaching in the general area of signal processing. He is the head of Bharti School of Telcom Technology and Management, and the coordinator of two centers of excellence, viz. Airtel IIT Delhi Centre of Excellence in Telecommunications and Ericsson IIT Delhi 5G Center of Excellence. He is also the incharge of an IoT laboratory that he set up in collaboration with Samsung. Besides this, he is the NCC co-ordinator of IIT Delhi. The areas in which he has been publishing and doing sponsored research are centered on signal processing. The areas include object representation, tracking and classification, odometry, depth map generation, representation and rendering. He is also exploring vector sensor-based underwater acoustic communications, and performance issues in molecular communications. He has mentored 5 startups, in the areas of virtualization, geofencing, UAV based solutions and recommendations and data mining. He actively participates in building and deploying technology. He has also served as an expert in numerous government and private agencies in aspects related to signal processing.

Nitish Kumar Singh is currently working in the area of speech, text and vision. He has completed his post graduation from Manipal Institute of Technology, Bangalore.