Low Bandwidth Video-Chat Compression using Deep Generative Models

Maxime Oquab*[*], Pierre Stock*[*], Oran Gafni*, Daniel Haziza*, Tao Xu*, Peizhao Zhang*, Onur Celebi*, Yana Hasson†, Patrick Labatut*, Bobo Bose-Kolanu*, Thibault Peyronel*, Camille Couprie*
* Facebook, † INRIA, work achieved during internship at Facebook AI Research

{qas, pstock, oran, dhaziza, xutao, stzpz, celebio, plabatut, bobobose, peyronel, coupriec}@fb.com, yana.hasson@inria.fr

Abstract

To unlock video chat for hundreds of millions of people hindered by poor connectivity or unaffordable data costs, we propose to authentically reconstruct faces on the receiver's device using facial landmarks extracted at the sender's side and transmitted over the network. In this context, we discuss and evaluate the benefits and disadvantages of several deep adversarial approaches. In particular, we explore quality and bandwidth trade-offs for approaches based on static landmarks, dynamic landmarks or segmentation maps. We design a mobile-compatible architecture based on the first order animation model of Siarohin et al. In addition, we leverage SPADE blocks to refine results in important areas such as the eyes and lips. We compress the networks down to about 3 MB, allowing models to run in real time on iPhone 8 (CPU). This approach enables video calling at a few kbits per second, an order of magnitude lower than currently available alternatives.

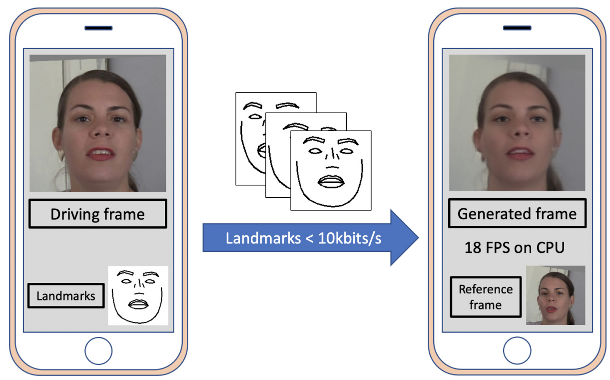


Figure 1. We propose to authentically reconstruct faces in real-time on mobile devices using a stream of compressed facial landmarks extracted from *driving* or target frames. The identity of the sender is transmitted in one shot to the receiver at the beginning of the call through a *reference* or *source* frame. This approach is compatible with end-to-end encryption (E2EE).

1. Introduction

For many smartphone users around the world, video-calling remains unavailable or unaffordable. These users are driven out of this fundamental connectivity experience by the prohibitive cost of data plans or because they depend on outdated technologies and infrastructures. For instance, networks might suffer from congestion, poor coverage, power fluctuations and datarate limits – 2G networks allow for a maximum of 30 kbits/s. However, with current technologies, an acceptable video-call quality requires at least a stable 200 kbits/s connection.

Meanwhile, the research in generative models has now come to a point where the quality of synthetic faces are sometimes indistinguishable from real videos [14]. To name a few, we may cite Deep video portraits [18], X2Face [45], FSGAN [28], Neural Talking Heads [49], the Bilayer model [48] and the First Order Model (FOM)[35]. This unprecedented performance can now be exploited to the benefit of higher quality video calls. However, there remain important challenges to address before generative models can offer ultra-low data-rate video-calling. In particular, to unlock duplex video-calling for users with last-mile connectivity issues or limited data plans, models need to be light and fast enough to run on mobile handsets. In addition, to deliver a more seamless and authentic experience, the models should adapt to the current appearance of the user without additional training. In this work, we focus on identifying the best generative strategy compatible with real-time inference on device. We discuss the following approaches:

- The Bilayer model [48], where the face is reconstructed from a stream of landmarks and a reference frame sent once.
- The SegFace model, a novel architecture based on SPADE [30] which requires sending an initial face embedding and a stream of semantic segmentation maps.
- The FOM [35], which requires sending ten landmarks, their related motion matrices, and a reference frame.

*Contributed equally

Analysing the FOM in depth, we observe that only sending the landmarks compressed with Huffman coding (no motion matrices) achieves sufficient quality and leads to an outstanding data-rate reduction. Compared to other approaches, this model allows for good identity and background preservation. Our contributions are the following:

- We provide the first comparative analysis of leading generative approaches for the specific use-case of enabling ultra-low data-rate video calling.
- We develop a strong baseline leveraging the SPADE architecture and segmentation maps.
- We propose a novel warping based approach building on FOM and leveraging SPADE blocks to refine important face attributes such as eyes and lips.
- While previous approaches were tested on specialized hardware (servers, mobile GPU), we provide first real-time results on mobile CPU.

In parallel to our work [29] were recently developed two generative approaches to video chat compression [44], that are like us both building on the FOM approach, but present important differences: First, Maxine [1, 44] is computationally intensive, employing highly dimensional feature deformations ($128 \times 128 \times 32 \times 64$, 128 times larger than ours). Consequently, the approach cannot be ran on device like ours, creating a problem of two-way applicability in a real low bandwidth use-case: The landmarks of a person A with a poor connection can be compressed and reconstructed on server allowing a receiver to see the video, but the person A will not be able to download a reconstructed video from the receiver: the bandwidth savings are limited to upload savings. Second, [19] applies the FOM as is, and suggests an algorithm to optimally select frames to compress, which is an orthogonal direction to our work.

2. Related work

2.1. Face compression before deep learning

The idea of face-specific video compression is not novel and appeared with classical computer vision tools, for instance morphings using Delaunay triangulations, Eigenfaces, or 3D models. The first reference we found on the topic is the work of Lopez et al. [25] that proposes to encode only pose parameters of a 3D head model, which is projected to reproduce a video sequence.

Previous work [21] use PCA to model the current frame as a linear combination of three basis frames sent prior to the call. The authors rely on known control points on the face boundaries and landmarks. The principal drawback of the approach is the presence of triangulation artefacts, even when a large number of control points is used. The achieved bandwidth is 1500 bits/frame. Similar usage of Eigenspaces are suggested in [36, 38, 39]. Among these proposals using Eigenspaces, one claims an extremely low bit-rates achievement of 100 bits/s [37]. However the proposed solution is hard to scale, as it requires storing personal galleries of face images to reconstruct videos at the receiver side.

2.2. Deep compression

The emergence of Generative Adversarial Networks (GANs) stimulated the application of deep learning to video compression. Super-resolution has been an active field of research leveraging GANs for image and video compression. There have been a number of research works tackling this problem [5, 9, 40]. However, for compressing faces, these reconstructions methods are limited to restoring personal traits from low level images and only work well for limited upscaling factors (around $2\times$ in resolution). The power of GANs for lossy image compression started to be demonstrated in the Generative compression work of Santukar et al. [34], using an auto-encoder combined with adversarial training. The state-of-the-art has since improved with the Extreme Learned Image Compression work of Agustsson et al. [2], thanks to a multi-scale architecture and the usage of semantic segmentation information, among other tricks used by the authors. The work of Liu et al. [23] surveys deep learning-based approaches for general purpose video compression. Among them, Learned Video Compression [31] demonstrates for the first time the superior capacity of an end-to-end machine learning approach over standard codecs. By focusing on faces only, we can lower the bandwidth, improve the quality and compress models compared to using more generic methods. Therefore, we review next deep face videos reconstruction approaches and their adequacy to video chat compression.

2.3. Deep talking head approaches

3D based approaches produce realistic avatars which can be animated in real-time [6]. However, such methods require to capture a set of images of the user (a few dozens) to build their personal face model. PAGAN [26] generates key face expression textures that can be deformed and blended in real-time on mobile from a single frame. However, the reconstruction of certain features, notably the hair, is still problematic in 3D model-based approaches. Deep video portraits [18] is handling this issue using a rendering-to-video translation network, but the approach needs about a thousand images per subject for training. Stimulated by advancements in face swapping pipelines [20, 45], a number of deep generative re-enactment approaches arose. Contrary to warping based re-enactment [3], learning faces reconstructions enables extra robustness in presence of large head angles. The Face Swapping GAN [28] relies on several steps: landmarks extraction, segmentation, interpolation and inpainting. This complex pipeline may result in ro-

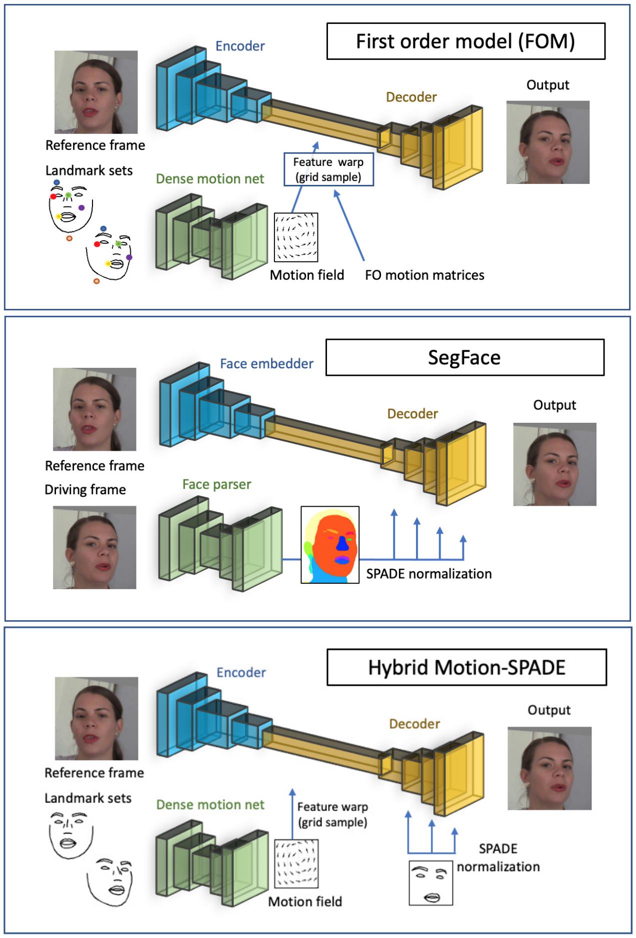


Figure 2. Deep generative approaches discussed in this study. We detail two novel architectures, SegFace, and Hybrid Motion-SPADE that builds on existing FOM [35] model. For all models, we assume the generation is performed by the encoder-decoder pair on the receiver device, while the emitter sends a reference frame (or several) at the beginning of inference, and streams a series of landmarks or segmentation maps.

bustness issues and limited bandwidth gain due to the need of sending both compressed segmentations and landmarks. Similarly, the vid2vid approach ([43], [42]) requires sending a "sketch" (edge map) for each frame in order to re-enact a face, which has a relatively high bandwidth cost. In the next section, we discuss and compare the learning based approaches which yield the most promising results in terms of bandwidth, visual quality and inference time.

3. Generative models

In this section, we first briefly describe two recent face animation algorithms as well as SPADE. We then introduce our two model contributions, namely SegFace and Hybrid Motion-SPADE, see Fig. 2. For our self-reenactment task, the goal is to generate frames based on (i) one fixed *reference* or *source* frame and (ii) position information (e.g. landmarks) from a stream of *driving* or *target* frames (see Figure 1).

3.1. Background

3.1.1 Bilayer model

The Bilayer synthesis approach [48] decouples high frequencies and low frequencies generation, does not require fine-tuning step as [49], and leads to visually appealing and sharp results. In our observations (see Figure 5), the identity preservation suffers from a stronger uncanny valley effect. In terms of bandwidth, the bilayer approach would require sending 68 compressed landmarks.

3.1.2 First order model for image animation (FOM)

The "First Order Model" approach of Siarohin et al. [35] deforms a reference source frame to follow the motion of a driving video. FOM follows an encoder-decoder architecture with a motion transfer component:

- A landmark extractor is learned using an equivariant loss, without explicit labels.
- Two sets of ten learned landmarks are computed for the source and driving frames.
- A dense motion network uses the landmarks and the source frame to produce a dense motion field and an occlusion map.
- The encoder encodes the source frame.
- The resulting feature map is warped using the dense motion field (using a differentiable grid-sample operation [17]), then multiplied with the occlusion map.
- The decoder generates an image from the warped map.

The networks are trained end-to-end on video frames, using perceptual losses, and are then optionally fine-tuned with an adversarial net. The self-supervised landmarks do not necessarily match precise locations of the face. Instead, they correspond to point coordinates optimized to achieve the best deformation of the source frame. [35] describes how to improve motion estimation in landmark areas by estimating Jacobian matrices to model motion in their neighborhood.

3.1.3 SPADE

SPADE blocks, introduced in [30], are normalization layers that incorporate spatial information from semantic segmentation maps. They are typically used to generate new images from layouts.

3.2. Our contributions: from segmentation to landmarks for video chat compression

3.2.1 SegFace

This approach builds upon SPADE [30]. Unlike MaskGAN [22], we propose to use a face descriptor computed on a source frame, and decode it by conditioning on face segmentation maps from a driving frame. It follows an encoder-decoder architecture described as follows:

- A face descriptor is computed on a source frame.
- This face descriptor is given to a decoder network, that applies SPADE normalization blocks at each layer using the face segmentation maps of the driving frame, ensuring all parts of the face are correctly placed.

The decoder network is trained using VGGFace2 face embeddings [7], and segmentation maps from [47] as inputs. Its objective during training is to reconstruct the same source frame. The optimization is done using losses from [30], and the face perceptual loss from [15]. This method operates on independent frames, and thus allows to use high-resolution training data, leading to high quality results. Training is achieved using CelebA [24] and Flickr-Faces-HQ datasets.

Bandwidth The model requires a segmentation map labeled for 15 categories (eyes, hairs, ears etc.). Sending compressed segmentation maps would require 18/25 kbits/s at resolutions $48\times$/$64\times$, knowing that there is a trade-off between the resolution of the transmitted segmentation maps and the quality of the generated faces. We do not build on this method further for low-bandwidth video-chat because the cost of running a face parser inference step and the bandwidth requirements are too high. The SegFace implementation, however, allows us to observe that the generated images respect the segmentation map labels almost perfectly. We build on this property with our Hybrid Motion-SPADE approach, after introducing some variants of the First order animation model.

3.2.2 FOM Variants

First, our implementation does not use the Jacobian component, as we do not observe a strong effect the quality of the results. We refer to the resulting model as "Motion Net (MN-10)" as it no longer uses first order approximation anymore and employs a set of ten landmarks.

Second, we explore using off-the-shelf facial landmarks extraction to replace the unsupervised landmarks. In this case, we only stream 20 or 68 compressed landmarks.

Third, we explore a combined strategy employing both 10 self-supervised landmarks and 20 supervised ones, that we note MN-10+20. We introduce a fourth variant in Section 3.2.3 below.

3.2.3 Hybrid Motion-SPADE model

Important quality criteria for compressed video-chat include a good synchronization between the lips and the speech, and a good rendering of the eyes and eyebrows; therefore, it is crucial to generate these facial parts precisely.

We propose an improvement over the FOM-based Motion Net method, by adding SPADE normalization layers in the upsampling blocks of the decoder network (in the last step of the FOM approach). We draw polygons for the eyes, eyebrows, lips and inner mouth using 60 extracted face landmarks, and use these as semantic maps for SPADE.

The dense motion network receives (i) a downsampled reference frame with (ii) the positions of N landmarks for that frame, and (iii) the positions of the same landmarks for a driving frame. It outputs a motion field M and an occlusion map O. The encoder network outputs a feature map F_s. The decoder warps F_s with the result of the dense motion network M and multiplies it element-wise with the occlusion map O, to obtain F_w. Then, F_w is processed by a stack of five residual blocks and three upsampling blocks that apply the SPADE normalization using a set of 60 landmarks.

Training is performed with a multiscale perceptual loss (based on a VGG-19 architecture) with a weight $\lambda_p = 10$ in addition to an equivariance loss with a weight $\lambda_{eq} = 1$ for the unsupervised landmark detector when applicable, following the procedure described in [35].

Bandwidth The necessary segmentation maps are obtained by plotting the polygons of the facial landmarks extracted using a landmark detector (see Figure 2), rather than running a face segmentation network. Moreover, landmark coordinates are inexpensive to transmit, while rasterized segmentation maps are more difficult to compress, especially at higher resolutions. In terms of bandwidth, this approach requires sending $N+60$ compressed landmarks. We experiment with $N = 10, 20$, and 30.

4. Compression

In this section, we explain different strategies to make architectures – and in particular our novel hybrid Motion-SPADE – compatible with low-bandwidth video calls on mobile. We first detail the architectures and then the compression aspects for the models and the bandwidth. Results are displayed in Table 1.

4.1. Mobile architectures

Base blocks We rely on the open-source FbNet family of architectures [12, 41, 46] to design mobile-capable models for our Motion Net and Motion-SPADE approaches. These networks typically build on blocks combining 1×1 point-wise and 3×3 depth-wise convolutions [33] that require less floating-point operations than traditional 3×3 convolutions found in residual blocks. We provide further architecture details in Figure 3.

Model variant	Inputs	FPS	#Params	#FLOPS	int8 size	Raw BW	Compressed BW
Motion Net	10 U	18	2.9 M	1411 M	3.1 MB	3.9 kbits/s	1.4 kbits/s
Motion Net	20 L	19	2.3 M	1293 M	2.5 MB	7.8 kbits/s	2.2 kbits/s
Motion Net	10 U + 20 L	14	3.0 M	1505 M	3.4 MB	11.7 kbits/s	3.6 kbits/s
Motion SPADE	10 U	16	2.9 M	1198 M	3.2 MB	27.3 kbits/s	8.0 kbits/s
Motion SPADE	20 L	19	2.3 M	1029 M	2.5 MB	41.2 kbits/s	8.8 kbits/s
Motion SPADE	10 U + 20 L	13	3.0 M	1292 M	3.4 MB	35.3 kbits/s	10.2 kbits/s

Table 1. Comparison of our approaches running on mobile in terms of compression for both model size and stream."10 U" (resp. "20 L") means that 10 unsupervised landmarks (resp. 20 facial landmarks) are used as inputs to the dense motion network. SPADE variants require 60 extra facial landmarks to draw the facial label maps. Notes: the "int8 size" is the full combined size of the models. The number of frames per second (FPS) is measured for the whole int8-quantized pipeline running on an iPhone 8 CPU, including landmark detection, grid-samples and face alignment. The #FLOPS count is for the dense motion, decoder, and unsupervised landmark extractor networks. The bandwidth (BW) is measured at 25 FPS, without (Raw BW) and with Huffman encoding (Compressed BW).

Mobile SPADE normalization blocks When applicable, we perform a SPADE normalization after the last 1×1 point-wise convolution, with kernel sizes of 1×1, and 32 hidden channels. We have found these parameters to provide a good trade-off between speed and quality while preserving the fidelity of the SPADE approach.

4.2. Landmark stream compression

We compress the landmarks with Huffman encoding [16]. In this approach, the landmark displacements are first binarized into 32 bins plus one sign bit, and we encode the bin index with Huffman coding. This compression leads to an average rate of 90 bits/frame for 20 landmarks, hence 2.2 kbits/s at 25 FPS (see Table 1 for details). For reference, bandwidth requirements for audio are around 10 kbits/s, while the AV1 video codec (not widely hardware-supported to date) aims at 30 kbits/s [11]. Therefore, we did not explore other variants such as Arithmetic Coding [32] since the audio part takes most of the bandwidth of a call with the proposed approach.

4.3. Model quantization

We rely on `int8` post-training quantization. This technique simply consists in uniformly quantizing both weights and activations over 8 bits, thus reducing the model size by a factor 4. Moreover, `int8` models traditionally benefit from a $\times 2-3$ speed-up compared to their `fp32` counterparts for both server and mobile CPUs. The scale and zero-point parameters[1] of the quantized layers are calibrated after training using a few batches of training data. When not properly calibrated, we found that the decoder generates an image with a small amount of grain or noise, resulting in a loss of visual quality. To compress the Motion based models, we only rely on `int8` since the non-compressed models are already small. The models are converted to TorchScript. Results are displayed in Table 1.

[1]The affine transform coefficients that allow converting an 8-bit quantized tensor (integer-valued in $[0, 255]$) to its floating-point counterpart.

4.4. Mobile inference

We use CoreML to deploy our models on mobile. We report the frame rates on various phones in Table 2. On the iPhone 11 pro, the battery CPU overhead is 0.47, and the battery NPE overhead is 0.05 (0.05% of battery is consumed every minute during a call).

	iPhone 12 pro	iPhone 11 pro	iPhone 8	Samsung S9
CPU (NE)	38 (125)	31 (80)	23	18

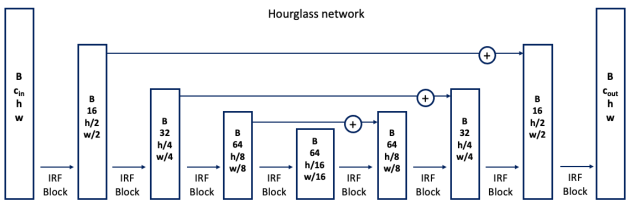
Table 2. FPS of Motion SPADE 20L model (Dense motion and decoder parts) on various phones. NE: Neural Engine.

4.5. Implementation details

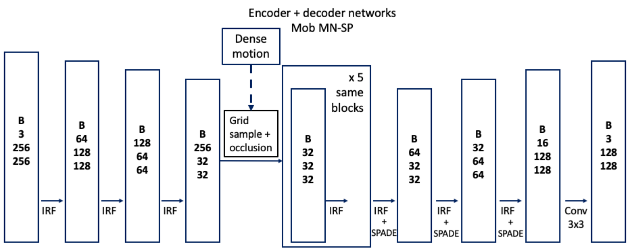
Our mobile models are trained on the DFDC dataset [14] rather than the VoxCeleb [27] dataset, in contrast to the original work of [35] (though we provide evaluation numbers for comparison and reference). We split different identities following a 90%-10% ratio, resulting in a total of 21899 training videos, and 2369 validation videos. We choose DFDC in this work because the videos are higher-quality and not cropped as tight, allowing for different face alignment procedures: (i) cropping around the face, or (ii) cropping after rotation using the facial landmarks such that the eyes are horizontally aligned. We have notably found that for smaller Motion Net models, this alignment makes the task easier and improves the results. The alignment procedure is reproduced on mobile at inference time to match the training distribution. We perform training on 8 GPUs using a Distributed Data Parallel pipeline in Pytorch, with a batch size of 48, for 265K steps. We use the Adam optimizer with learning rate 2.10^{-4} on all networks in all experiments.

5. Experiments

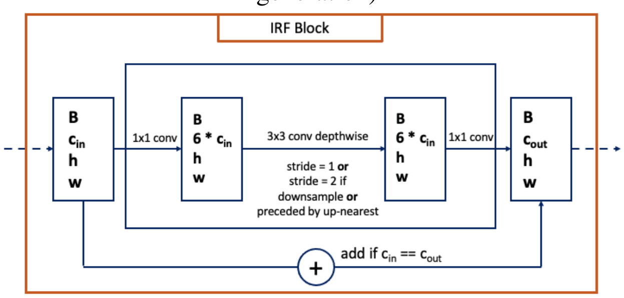
Section 5.2 presents ablation study of FOM and our Hybrid SPADE approach. In section 5.3, we compare these



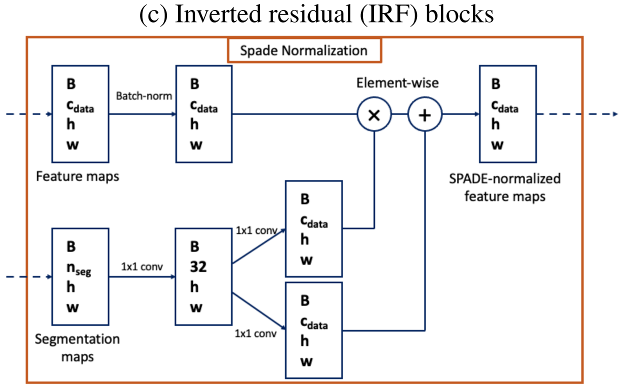
(a) Mobile architectures (dense motion net and landmark extractor)



(b) Generator architecture (Input: driving frame, output: generation)



(c) Inverted residual (IRF) blocks



(d) SPADE blocks used in Motion-SPADE

Figure 3. Schemes of mobiles architectures for the Motion Net ((a), (c)) and Motion-SPADE approaches ((a), (b), (c), (d)).

results to alternative Segface and bilayer approaches. Finally, we conduct a human study in section 5.4 to quantify improvements brought by our hybrid-SPADE approach.

5.1. Evaluation metrics

We evaluate the models using the perceptual LPIPS [50] and multi-scale LPIPS-like metrics employed in [35], that we name msVGG. Second, as argued in [8], the cosine similarity CSIM computed between features of the pre-trained face embedding network ArcFace [13] is one of the most effective metric to assess quality of talking heads models, we therefore report it. Finally, we quantify facial landmarks mismatch by running a landmark detector on true and generated videos and computing the Mean Square Error between each pair of landmarks. This metric is classically referred to as the Normalized Mean Error (NME) of head pose [4].

	FOM adv	FOM w/o adv	MN
msVGG $\downarrow$	**85.6**	87.5	87.9
LPIPS $\downarrow$	**0.226**	0.233	0.236
NME $\downarrow$	**0.51**	0.53	0.54
CSIM $\uparrow$	**0.83**	0.81	0.82

Table 3. Ablation study for FOM on VoxCeleb2-28. MN: FOM without first order approximation nor adversarial fine-tuning.

5.2. Quality evaluation: ablation studies

For evaluation, we assembled a set of 28 videos of diverse persons in terms of gender, age, skin color from the validation set of VoxCeleb2 [10], and a similar set of 50 videos from the validation split of the DFDC dataset [14].

We begin our analysis of the FOM by computing the quality of reconstruction without first order motion approximation and without adversarial training in Table 3. While it is clear that the adversarial fine-tuning boosts the performance, we experiment without it in the remaining of our ablation study around this model to reduce training time for each model. Removing the first order approximation only slightly degrades the LPIPS but not the msVGG perceptual metric. Interestingly, the CSIM metric which is the one supposed to best reflect the identify preservation, is slightly increased by dropping this component. A second observation is that the fidelity of facial landmarks to the target video is negatively affected by this removal. Since the drop of performance induced by discarding first order motion approximation leads to important bandwidth savings and limited loss in performance, we conduct our experiments without it. We refer to this approach as the Motion Net approach. Next, we explore the replacement of the self-supervised landmarks of the Motion Net approach by off-the-shelf landmarks from a state-of-the art detector. Results appear in Table 4. Note that the results presented in this table are obtained by our re-implementation of the MotionNet approach, and are slightly better than these of Table 3 obtained with the original code. We compare in Table 4 different variants of the Motion Net approach, using 20 input landmarks, 68 input landmarks, self-supervised landmarks with dense architectures and with mobile architectures. All these dense architecture employ a latent space of $256 \times 64 \times 64$, and were trained on VoxCeleb. Using standard facial landmarks instead of unsupervised motion landmarks degrades the scores of perceptual metrics, but

Figure 4. Qualitative results using Motion based variants on mobile architectures, using a $32 \times 32 \times 256$ latent space. Each model generates the face given the fixed source frame and the landmarks of the target frame. All models run in real-time on an iPhone 8.

	LPIPS ↓	NME ↓	CSIM ↑
Dense MN-10 U	0.221	0.59	0.83
Dense MN-20 L	0.242	0.50	0.80
Dense MN-68 L	0.240	0.49	0.81
Mob MN-10 U	0.225	0.52	0.79
Mob MN-20 L	0.244	0.48	0.78
Mob MN-10 U + 20 L	0.218	0.46	0.80
Mob M-SPADE-10 U	0.217	0.47	**0.81**
Mob M-SPADE-20 L	0.242	**0.44**	0.79
Mob M-SPADE-10 U + 20 L	**0.215**	0.46	**0.81**

Table 4. Evaluation results for Motion Net approaches without adversarial fine-tuning on the VoxCeleb2-28 video subset. Mob : Mobile models. Dense models (64×64 latent space) are trained on VoxCeleb. Mobile models (32×32) are trained on the DFDC aligned dataset. U: unsupervised landmarks; L: facial landmarks.

improve NME. Using 68 landmarks is only very slightly improving the quality over 20. With mobile architectures, we reduce the latent space to $256 \times 32 \times 32$. In addition to using 10 motion landmarks or 20 landmarks alone, combining these two sets helps boost all quality metrics. Finally, we observe that adding the SPADE blocks preserves the perceptual quality and brings a large improvement in NME.

	Dense models (on server)			Mobile
	Bilayer	SegFace	FOM	MS20L
msVGG* ↓	68.6	84.4	58.7	*57.9*
LPIPS* ↓	0.200	0.304	**0.153**	0.167
NME* ↓	0.55	0.55	0.50	**0.44**
CSIM* ↑	0.85	0.76	**0.87**	0.84
kbits/s ↓	9.7	18	**4.0**	8.8

Table 5. Comparison of Bilayer, SegFace (48×48), FOM adv, in terms of quality / bandwidth (kbits/s with 25 fps) trade-offs on VoxCeleb2-28. * : Metrics were computed using ground truth backgrounds. We also include our best mobile model, Motion-SPADE-20L (MS20L) in the comparison.

5.3. Quantitative comparative evaluation

We compare the quality/bandwidth trade-off of different dense face animation approaches in Table 5. As the Bilayer and Segface approaches do not generate backgrounds, we pasted original backgrounds to provide a fair evaluation. We observe that FOM leads to better LPIPS and CSIM metrics. Interestingly, our mobile results have better NME and msVGG scores than the original FOM dense approach.

5.4. Qualitative evaluation and human study

Fig. 4 illustrates the quality performance reached on Mobile. The last line displays a challenging case for the motion based approach. We note that the quality of results degrades

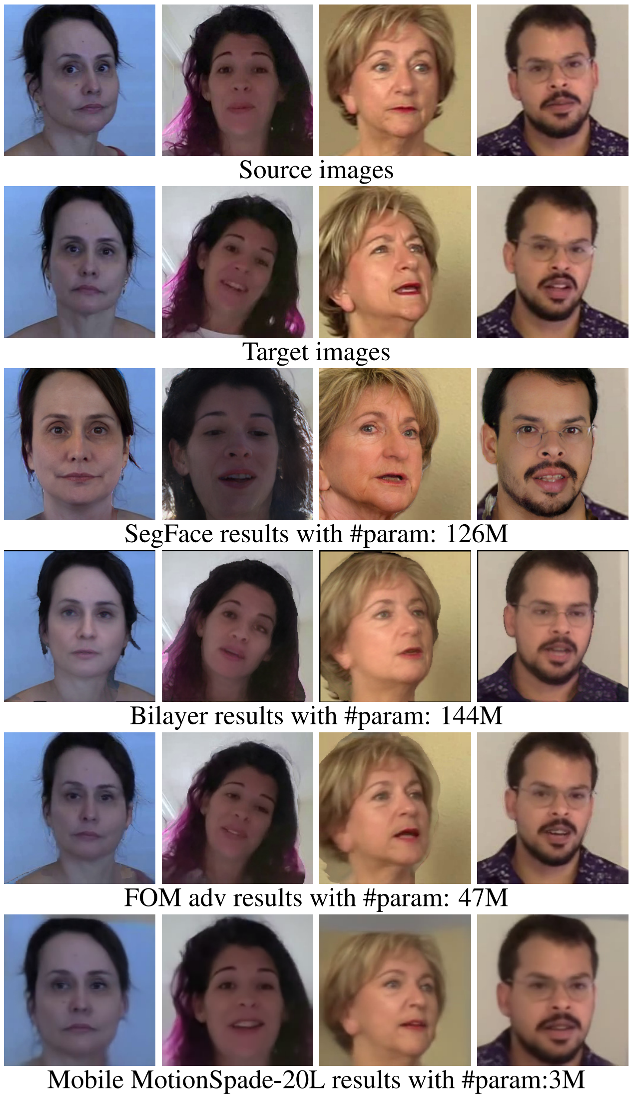


Figure 5. Comparison of different results using SegFace (48×48), Bilayer, and FOM adv. Each model generates faces using fixed source frames and facial information (e.g. landmarks) of driving frames. We pasted ground truth backgrounds for a fair evaluation of dense models, rows 3 to 5. The last row showcases the results using our Mobile Motion-SPADE that runs at 18 FPS on an iPhone 8, whereas other models run on server, have at least $10\times$ more parameters and are not necessarily compatible with low-bandwidth video calling. The alignment procedure may differ between the models, leading to different head centering.

in presence of large head rotation. Still, Motion-SPADE results are visually close to the targets, particularly it renders lips and teeth better. The H264 compression results are displayed given a bandwidth of 9 kbit/s, to be compared to the ones of the Motion-SPADE 20 model that runs the fastest on mobile. This illustrates that at this bandwidth, video transmission is hardly possible using standard codecs, whereas our mobile approach would make the video call

Dense models			
model	human identity score	human expression score	
Bilayer	3.70±0.046	3.62±0.046	
SegFace	3.11±0.047	3.00±0.051	
FOM adv	**3.99**±0.042	**4.00**±0.041	
Overall human ratings of Mobile models			
	MN-10	MN-20	MN-10+20
no SPADE	3.44$_{\pm 0.034}$	3.40$_{\pm 0.034}$	3.46$_{\pm 0.034}$
with SPADE	**3.50**$_{\pm 0.034}$	3.46$_{\pm 0.035}$	3.45$_{\pm 0.034}$

Table 6. Quality assessment of models on DFDC-50 - Human study. Average scores (Higher is better) with confidence intervals.

possible. Fig. 5 compares results obtained using SegFace, Bilayer, and FOM with adversarial finetuning. We observe skin tones/lightning differences between targets and SegFace results and distortions of personal traits. The Bilayer, and FOM models are qualitatively better. In the last column, we observe a promising comparison with a Mobile Motion-SPADE model, given its real-time performance and possible orthogonal improvements using adversarial training. Table 6 provides a quality assessment of different models by human raters. Participants rated images produced by the different models by comparing them in terms of identity and expression preservation, on a scale from 1 to 5. In a first round of evaluations, we display side by side the main dense models results, and in the second round, Motion Net and Motion-SPADE results using six different mobile architectures. We collect in each case 500 pairwise evaluations, each from five different participants. For dense models results, human scores seem to agree with metrics, ranking FOM first. Mobile models results differences are more subtle, but the Hybrid Motion-SPADE-10 landmarks model is preferred. The addition of SPADE blocks brings significant improvement in most cases.

6. Conclusions

We designed a novel low bandwidth face video compression approach, the first one that is able to run in real time on mobile. Our hybrid architecture takes advantage of high fidelity to the target image thanks to the warping principle, and enhances the quality of important attributes with SPADE blocks. Only exploiting polygons induced segments allows our approach to improve quality without high transmission cost. The bandwidth required to send a video using this approach is lower than the one required for sending audio. There are a number of interesting research directions to improve the quality of the generations, beyond using adversarial losses, e.g. generating large head rotation movements, hands.

References

[1] Ai can see clearly now: Gans take the jitters out of video calls (nvidia maxine blog post), Oct 2020. 2

[2] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *ICCV*, 2019. 2

[3] Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F Cohen. Bringing portraits to life. *Transactions on Graphics*, 36(6):1–13, 2017. 2

[4] 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 *ICCV*, 2017. 6

[5] Adrian Bulat and Georgios Tzimiropoulos. Super-fan: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with gans. In *CVPR*, 2018. 2

[6] Chen Cao, Hongzhi Wu, Yanlin Weng, Tianjia Shao, and Kun Zhou. Real-time facial animation with image-based dynamic avatars. *Transactions on Graphics*, 35(4), 2016. 2

[7] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face & Gesture Recognition*, 2018. 4

[8] Lele Chen, Guofeng Cui, Ziyi Kou, Haitian Zheng, and Chenliang Xu. What comprises a good talking-head video generation?: A survey and benchmark. *arXiv 2005.03201*, 2020. 6

[9] Yu Chen, Ying Tai, Xiaoming Liu, Chunhua Shen, and Jian Yang. Fsrnet: End-to-end learning face super-resolution with facial priors. In *CVPR*, 2018. 2

[10] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018. 6

[11] Dave Citron. “four new google duo features to help you stay connected”, April 2020. 5

[12] Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Bichen Wu, Zijian He, Zhen Wei, Kan Chen, Yuandong Tian, Matthew Yu, Peter Vajda, et al. Fbnetv3: Joint architecture-recipe search using neural acquisition function. *arXiv 2006.02049*, 2020. 4

[13] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 6

[14] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset. *arXiv 2006.07397*, 2019. 1, 5, 6

[15] Oran Gafni, Lior Wolf, and Yaniv Taigman. Live face de-identification in video. In *CVPR*, 2019. 4

[16] David A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the Institute of Radio Engineers*, 40(9):1098–1101, 1952. 5

[17] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NeurIPS*, pages 2017–2025, 2015. 3

[18] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *Transactions on Graphics*, 37(4), 2018. 1, 2

[19] Goluck Konuko, Giuseppe Valenzise, and Stphane Lathuilire. Ultra-low bitrate video conferencing using deep image animation. *arXiv 2012.00346*, 2020. 2

[20] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. In *ICCV*, 2017. 2

[21] Ioannis Koufakis and Bernard F Buxton. Very low bit rate face video compression using linear combination of 2D face views and principal components analysis. *Image and Vision computing*, 17(14):1031–1051, 1999. 2

[22] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *CVPR*, 2020. 3

[23] Dong Liu, Yue Li, Jianping Lin, Houqiang Li, and Feng Wu. Deep learning-based video coding: A review and a case study. *Computing Surveys*, 53(1):1–35, 2020. 2

[24] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 4

[25] Ricardo Lopez and Thomas S Huang. Head pose computation for very low bit-rate video coding. In *International Conference on Computer Analysis of Images and Patterns*, 1995. 2

[26] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. pagan: real-time avatars using dynamic textures. In *SIGGRAPH Asia*, page 258, 2018. 2

[27] Arsha Nagrani, Joon Son. Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017. 5

[28] Yuval Nirkin, Yosi Keller, and Tal Hassner. FSGAN: Subject agnostic face swapping and reenactment. In *CVPR*, 2019. 1, 2

[29] Maxime Oquab, Pierre Stock, Oran Gafni, Daniel Haziza, Tao Xu, Peizhao Zhang, Onur Celebi, Yana Hasson, Patrick Labatut, Bobo Bose-Kolanu, Thibault Peyronel, and Camille Couprie. Low bandwidth video-chat compression using deep generative models. *arXiv 2012.00328*, 2020. 2

[30] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 1, 3, 4

[31] Oren Rippel, Sanjay Nair, Carissa Lew, Steve Branson, Alexander Anderson, and Lubomir Bourdev. Learned video compression. In *ICCV*, 2019. 2

[32] Jorma Rissanen and Glen G. Langdon. Arithmetic coding. *IBM Journal of Research and Development*, 23(2):149–162, 1979. 5

[33] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 4

[34] Shibani Santurkar, David Budden, and Nir Shavit. Generative compression. In *Picture Coding Symposium*, 2018. 2

[35] Aliaksandr Siarohin, Stphane Lathuilire, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *NeurIPS*, 2019. 1, 3, 4, 5, 6

[36] Ulrik Söderström. *Very low bitrate facial video coding: based on principal component analysis*. PhD thesis, Tillämpad fysik och elektronik, 2006. 2

[37] Le-Hung Son, Ulrik Söderström, and Haibo Li. Ultra low

bit-rate video communication: video coding= pattern recognition, 2006. 2

[38] Luis Torres and Daniel Prado. A proposal for high compression of faces in video sequences using adaptive eigenspaces. In *ICIP*, 2002. 2

[39] Mihran Tuceryan and Bruce E Flinchbaugh. Model based faced coding and decoding using feature detection and eigenface coding, Mar. 28 2000. US Patent 6,044,168. 2

[40] Evgeniya Ustinova and Victor S. Lempitsky. Deep multi-frame face hallucination for face identification. *arXiv*, 1709.03196, 2017. 2

[41] Alvin Wan, Xiaoliang Dai, Peizhao Zhang, Zijian He, Yuandong Tian, Saining Xie, Bichen Wu, Matthew Yu, Tao Xu, Kan Chen, Peter Vajda, and Joseph E. Gonzalez. Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions, 2020. 4

[42] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. In *NeurIPS*, 2019. 3

[43] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018. 3

[44] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. *arXiv 2011.15126*, 2020. 2

[45] Olivia Wiles, A. Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *ECCV*, 2018. 1, 2

[46] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search, 2019. 4

[47] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, 2018. 4

[48] Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. Fast bi-layer neural synthesis of one-shot realistic head avatars. *ECCV*, 2020. 1, 3

[49] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. *ICCV*, 2019. 1, 3

[50] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6