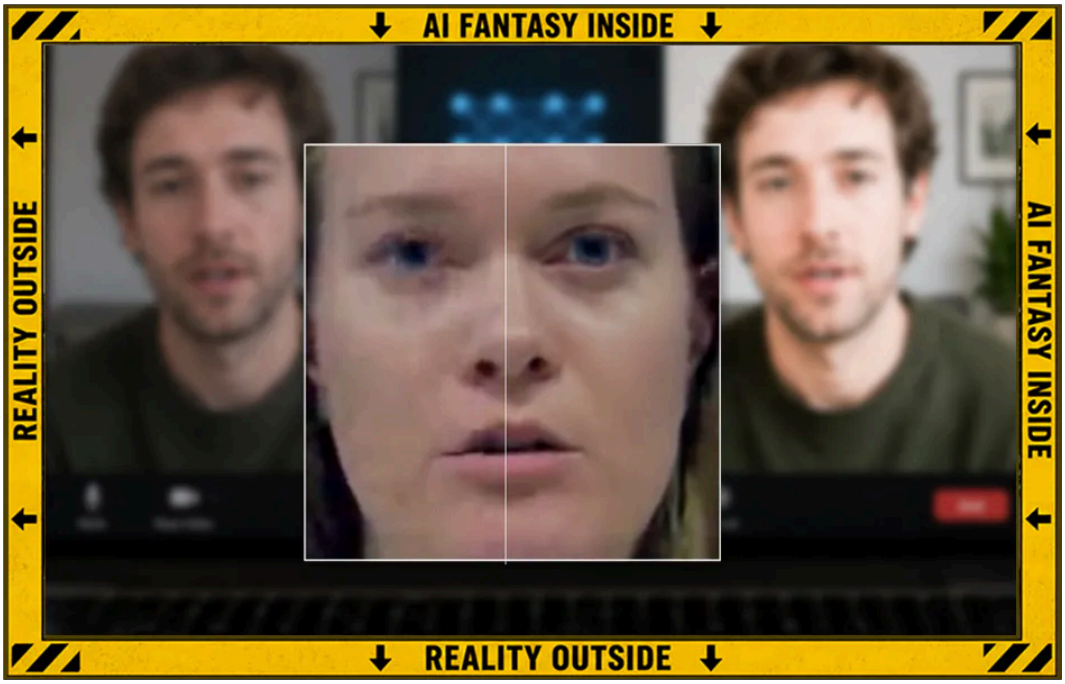
Fixing Blurry Video Calls in Real Time Using Only CPU

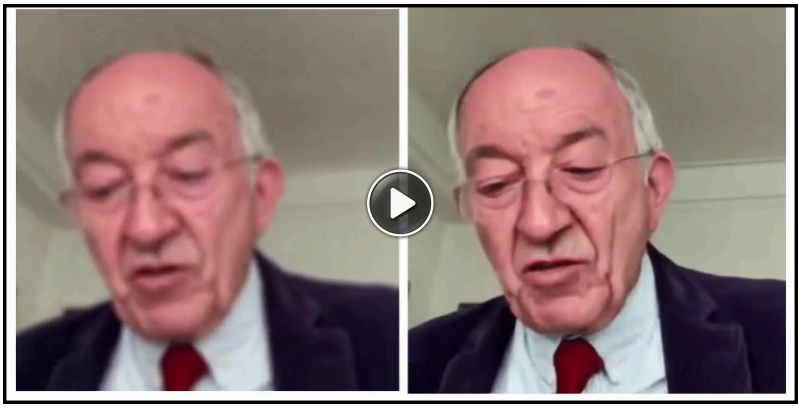
Originally published at https://www.unite.ai/fixing-blurry-video-calls-in-real-time-using-only-cpu/



A new method creates an AI model of your face in seconds, to fix blurry video calls in real time — running entirely on a basic laptop CPU, without the need for powerful hardware, or cloud processing.

Though users in under-resourced countries are most affected by low-quality video chat images, it is frequently a 'first world' problem as well – for instance, if one of the participants in a chat is using an over-taxed Wi-Fi hotspot, or some other process or person in a participant's household is dominating the available bandwidth; or even if that person is visiting a country with poor connectivity.

Since the problem is common, a certain amount of attention has been devoted to it in the research literature, with proposed solutions offered via deblocking, denoising, super-resolution, and various other methods. The VQFR system from 2022 restores degraded facial images by replacing low quality image features with higher quality representations drawn from a learned codebook; 2021's GFP-GAN leverages generative facial priors to improve LQ images; and *RTFVE: Realtime Face Video Enhancement* uses high-quality reference images to perform face restoration:



CLICK TO PLAY VIDEO ON SITE

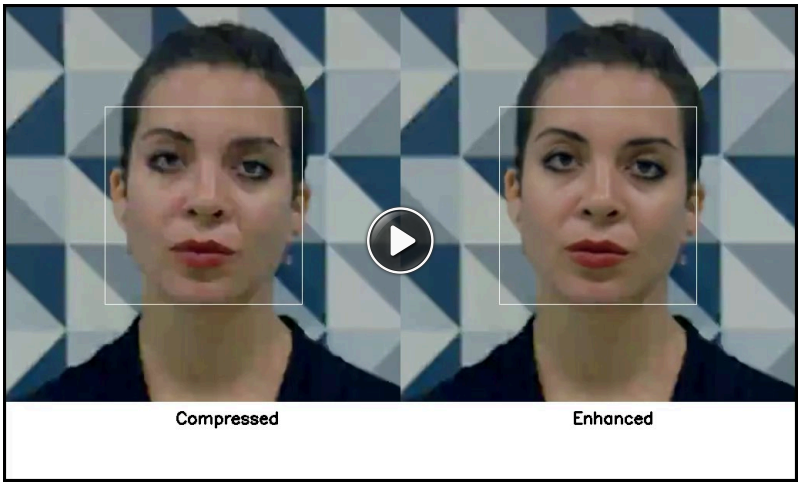
Click to play, as necessary. *From the 2025 RTFVE face enhancement project, GPU-accelerated improvement of faces.* Source

From the point of view of the under-resourced user, most of these solutions are not viable, since they offload the work to a GPU that may not be available in that user's setup. However, using the (far less capable) CPU for computer vision tasks is normally an offline process, meaning that one would need to wait for the CPU to finish processing before the output would be available.

This is unacceptable for the video chat scenario, which is resource-hungry but also, often, resource-starved: any truly democratic facial improvement system would need to run on the CPU at a minimum of 24fps at a reasonable resolution; and this is a lot to ask.

Facing Up

This demanding scenario is met by a new research offering from the US, which is capable, the authors claim, of training a model in less than 100 seconds from sparse frames of the viewer's face, with the final model then running as an interstitial layer between the user and the conference, via official API layers in video-conferencing applications such as Zoom:



CLICK TO PLAY VIDEO ON SITE

Click to play as necessary. *From the project page for FSFVE, sample improvements to webcam-scenario faces, using a lightweight (3.5MB) model trained in less than two minutes.* Source

The paper states:

> *'The FSFVE model can be trained either just before a videocall, or at the start of a call. Only a few high quality images of the subject are required; e.g. 5 to 30 images making it few-short learning. Just before the call, or at the beginning of the call, a few seconds of high quality video of the user can be obtained; for instance 3 seconds, providing 3∗24=72 frames assuming a frame rate of 24 frames/second (FPS).*
>
> *'This high quality video can quickly be re-encoded to a low quality video based on the settings for the videocall, and then with the low and high quality frames, the model is trained.'*

One notable feature of the new system is its flexibility in regard to who 'pays' for the processing of the model; if the viewer themselves for some reason cannot provide the CPU power for the training, this can be offloaded to the correspondent, through sending a small number of high quality frames, with the model thus trained 'remotely' to the resource-starved user.

Alternately, the training can be offloaded systematically to a server for training. The authors observe:

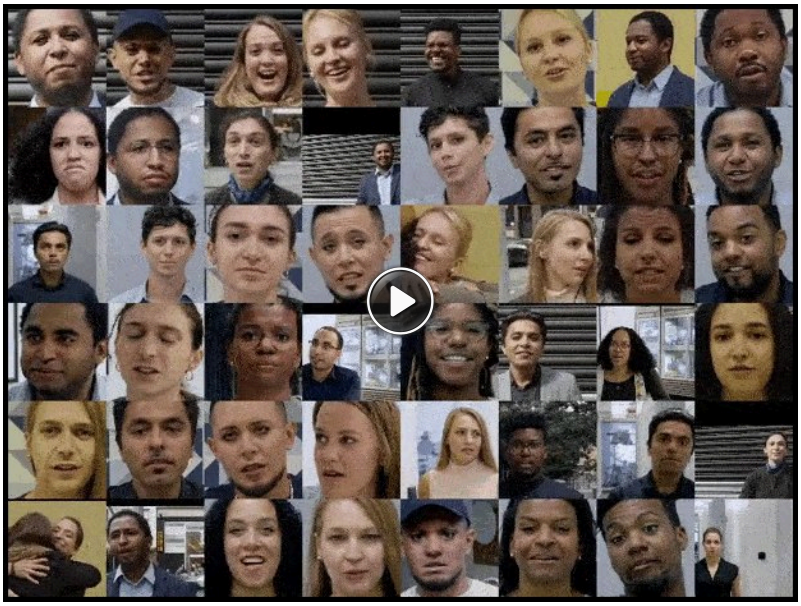
> *'The server can be used for cases where the sending and receiving devices are too slow for training (or even if not, to relieve them of the task).*
>
> *'In any case, the size of the model is relatively small (about 3.5 MB) as are a few high quality frames, and once training is complete, the model can run in real-time on a typical laptop CPU.'*

In tests, with models trained against baselines and prior works (including the aforementioned RTFVE, which forms the basis of the new work), FSFVE consistently outperformed the unenhanced compressed video, a CPU-optimized ResNet, and the modified RTFVE-0 model, while still running in real time on a CPU.

The new paper is titled *FSFVE: Few Shot Compressed Face Video Enhancement*, and comes complete with a demo and a GitHub repo.

Method

FSFVE was trained on the open FaceForensics++ dataset*, which consists of 363 1080p videos of actors talking, from which the authors extracted the *talking against a wall* category, as being nearest in style to a typical video call:



CLICK TO PLAY VIDEO ON SITE

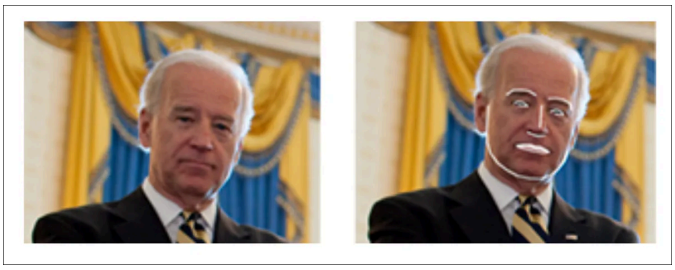
Click to play, as necessary. *Examples from the FaceForensics++ dataset.* Source

This subset comes to 27 videos of around 972 frames each – mostly frontal views, but, necessarily, some side views too.

To generate low-quality videos designed to simulate video call connections issues, the authors compressed the dataset using the H264 and H265 video codecs. Compression levels were set to a CRF range spanning 36, 40, and 44 (considering that zero is lossless and 51 is extremely blocky).

Each frame was cropped to a 256 × 256 region around the face using the BlazeFace face detector, after which facial keypoints were used to align the face by locating the eyes and applying an affine warp (which rotates, scales and shifts the face into a consistent position), so that the eyes remain level, and in approximately the same position across all frames.

This was undertaken with the Face Recognition library:



An example of extracting facial lineaments from an image with the Face Recognition Python library. Source

Because the model is designed to learn from only a handful of frames from a single video call, the training frames needed to capture as much facial variation as possible. Therefore k-means clustering was used instead of simpler selection methods (i.e., random sampling or fixed intervals).

Each cropped and aligned frame was passed through the aforementioned RTFVE face encoder to generate a numerical representation, after which k-means clustering grouped similar frames into 30 clusters. This clustering process was repeated five times to obtain the best grouping, and the frame nearest each cluster center selected for training. This process produced a diverse set of 30 images for each video.

Training and Tests

The models were trained for 100 epochs, which is rather low, considering the very small number of images involved. However, as the authors observe, an overfitted model is actually *desirable* in this scenario, even if it cannot capture all possible eventualities, such as uncommon facial expression, poses, or the hiding/obfuscation of facial features. The priorities, instead, are moderate flexibility in generalization, and speed of training.

The RAdam optimizer was used at a learning rate of 10^{-4}.

For initial tests, the system was compared against the original compressed video; a lightweight ResNet configured for real-time CPU inference; and the aforementioned RTFVE – the only other real-time CPU face enhancement system known.

To ensure a fair comparison, RTFVE was modified to remove its dependence on high quality reference images during inference while preserving a similar speed and parameter count, producing a variant called *RTFVE-0*. The ResNet and RTFVE-0 models were trained using the L1 loss function. To ensure a fair comparison, all models used the same RAdam optimizer and training settings, with a separate instance-specific model trained from 30 frames, for each video.

To speed up training, a custom frequency-domain focal loss (emphasizing the most visually important image details during training) was introduced, to give greater weight to the low-frequency DCT components (the image information after conversion into frequency values), which have the greatest influence on perceived image quality.

The weighting was derived from the JPEG luminance quantization matrix, causing the model to prioritize the most visually important image structures during training.

Quantitative Tests

Both *technical reconstruction accuracy* (distortion) and *perceived visual quality* were measured for the tests. Distortion was measured by Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM); and perceptual quality by Learned Perceptual Image Patch Similarity (LPIPS):

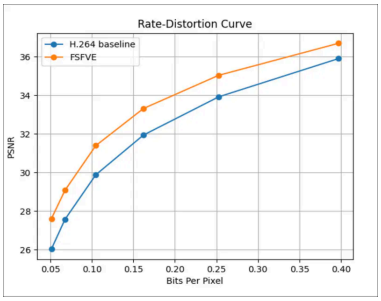
Codec	CRF	Model	PSNR (↑)	SSIM (↑)	LPIPS (↓)
H.264	36	H.264 (baseline)	33.9231	0.9097	0.0979
		ResNet	33.9568	0.9134	0.1759
		RTFVE-0	34.0147	0.9133	0.1401
		FSFVE (Ours)	35.0399	0.9257	0.1037
	40	H.264 (baseline)	31.9403	0.8837	0.1388
		ResNet	32.0716	0.8893	0.2301
		RTFVE-0	32.1601	0.8887	0.1801
		FSFVE (Ours)	33.3174	0.9062	0.1225
	44	H.264 (baseline)	29.8786	0.8505	0.1935
		ResNet	30.1967	0.8618	0.2789
		RTFVE-0	30.1563	0.8566	0.2328
		FSFVE (Ours)	31.3923	0.8771	0.1467
H.265	36	H.265 (baseline)	33.5402	0.9068	0.1118
		ResNet	33.4028	0.9099	0.1962
		RTFVE-0	33.6326	0.9116	0.1487
		FSFVE (Ours)	34.6758	0.9232	0.1112
	40	H.265 (baseline)	31.5015	0.8783	0.1552
		ResNet	31.6465	0.8869	0.2358
		RTFVE-0	31.6737	0.8844	0.1911
		FSFVE (Ours)	32.8555	0.9011	0.1287
	44	H.265 (baseline)	29.4523	0.8444	0.2182
		ResNet	29.7738	0.8586	0.2835
		RTFVE-0	29.6721	0.8517	0.2503
		FSFVE (Ours)	30.8846	0.8708	0.1559

Performance of FSFVE against the original compressed video, a CPU-optimized ResNet, and the modified RTFVE-0 across H.264 and H.265 video at three compression levels, with higher PSNR and SSIM indicating better reconstruction quality, and lower LPIPS indicating better perceptual quality.

FSFVE consistently outperformed both the original compressed video, as well as the competing CPU-based enhancement models, across every compression level.

With H.264 video, PSNR increased by 1.11dB, 1.37dB, and 1.51dB over the baseline at CRF values of 36, 40, and 44, with corresponding improvements in SSIM and lower LPIPS scores at the two higher compression settings (indicating better perceived image quality).

Similar gains were recorded with H.265, suggesting that the approach remains effective across both major video codecs. As shown below, the improvement became more pronounced as compression increased, indicating that the method performed particularly well under the low-bandwidth conditions that it was designed to address:



PSNR as bitrate increases for the original H.264 video and the enhanced output. The widening gap at lower bitrates shows that FSFVE delivers its largest quality gains under the most heavily compressed, bandwidth-constrained conditions.

The CPU-optimized ResNet and RTFVE-0 provided only modest improvements over the compressed baseline, whereas FSFVE exceeded both by more than 1.1dB, while maintaining real-time performance at 30.24 FPS, despite containing around a scant one million parameters.

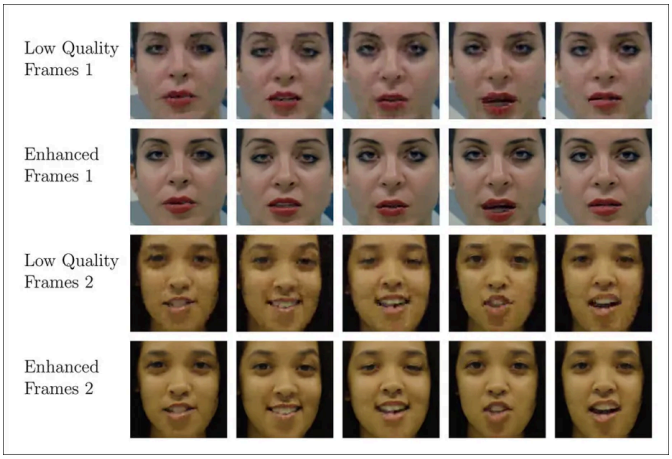
As shown below, performance improved steadily as more training frames were provided. Even with only five training images, FSFVE improved PSNR by more than 0.75dB over the compressed baseline, while ten training frames were sufficient to exceed a 1dB improvement – indicating that the approach remained effective with very limited training data:

Train Size	5	10	20	30
FSFVE (Ours)	30.6335	30.9452	31.0902	31.3923

Effect of training set size on PSNR for unseen H.264 video frames at CRF 44. Even five training images improved quality over the compressed baseline, with performance increasing steadily as more frames were available.

Qualitative Tests

In the results below, for the qualitative phase of testing, we see heavily compressed H.264 video frames at CRF 44 with the corresponding FSFVE output:



Comparison of heavily compressed H.264 video frames at CRF 44 with the corresponding FSFVE output. The enhanced results, the paper asserts, reduce compression artifacts, restore facial structure, and produce smoother, more natural skin while preserving the subject's identity and expression. Please refer to the source PDF and original videos for better examples.

The authors maintain that the compressed images exhibit prominent artifacts around the eyes, nose and mouth, along with unnaturally textured skin and distortions in facial features, whereas the enhanced results substantially reduce these defects:

> *'In the first example, the eyes appear unclear with the black of the eye makeup blending with the eye color. There are clear signs of aliasing with the lips. The eyebrows lack definition and the skin appears distorted. In the second example, the skin is highly textured with speckle artifacts.*
>
> *'There are blocking artifacts below the mouth that alter the shape of the chin. There also appear to be vertical lines.*
>
> *'Our model is able to remedy these artifacts generating a visually pleasing output with clearer skin and restores some of the fine details that were lost in the compression.*
>
> *'Importantly, the enhanced output remains faithful to the identity in the video.'*

Conclusion

It seems inevitable that these ongoing efforts across industry and academia to boost the quality of low-signal video chat data will eventually clash against the opposing efforts to identify deepfake footage in video calls.

Since, as the new paper notes, interstitial systems such as FSFVE can be legitimately applied to chat video streams via official APIs, it's possible that non-processed content will remain available for use by anti-deepfake measures, as they emerge – and become more important.

* *Referred to in the new paper as the 'DeepFakeDetector' dataset, though it does not appear to be known by this name, even inside the FaceForensics++ paper that the authors link to in this regard.*

First published Tuesday, July 14, 2026

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai

Discover More