Neural Emotion Director: Speech-preserving semantic control of facial expressions in “in-the-wild” videos

Foivos Paraperas Papantoniou1  Panagiotis P. Filntisis1  Petros Maragos1  Anastasios Roussos2,3

1School of Electrical & Computer Engineering, National Technical University of Athens, Greece
2Institute of Computer Science, Foundation for Research & Technology - Hellas (FORTH), Greece
3College of Engineering, Mathematics and Physical Sciences, University of Exeter, UK

Abstract

In this paper, we introduce a novel deep learning method for photo-realistic manipulation of the emotional state of actors in “in-the-wild” videos. The proposed method is based on a parametric 3D face representation of the actor in the input scene that offers a reliable disentanglement of the facial identity from the head pose and facial expressions. It then uses a novel deep domain translation framework that alters the facial expressions in a consistent and plausible manner, taking into account their dynamics. Finally, the altered facial expressions are used to photo-realistically manipulate the facial region in the input scene based on an especially-designed neural face renderer. To the best of our knowledge, our method is the first to be capable of controlling the actor’s facial expressions by even using as a sole input the semantic labels of the manipulated emotions, while at the same time preserving the speech-related lip movements. We conduct extensive qualitative and quantitative evaluations and comparisons, which demonstrate the effectiveness of our approach and the especially promising results that we obtain. Our method opens a plethora of new possibilities for useful applications of neural rendering technologies, ranging from movie post-production and video games to photo-realistic affective avatars.

1. Introduction

Photo-realistically manipulating faces in images or videos has received substantial attention lately, with impressive results that expand the range of creative video editing, content creation and the VFX industry. Yet, when it comes to altering the facial emotion in videos, existing techniques exhibit severe limitations. The importance of this type of manipulation is clearly illustrated when shooting movies, as capturing the desired actor’s emotion typically requires multiple efforts, despite the uttered words being predefined. A robust solution to emotion editing would conveniently place the manipulation of facial performance in the post-production stage.

In the past, the problem has been tackled by assuming that different recordings of the same script being acted in
multiple emotions are available; hence, enabling to switch or blend between takes, after perfect synchronization has been achieved [33]. However, in a more realistic scenario, one would like to make, e.g., a neutral actor look happy, without using pre-existing footage. Combining performances from unpaired data is much more challenging.

Recently, image-to-image translation has been successfully applied to emotion editing by casting the problem in the image space [7, 9]. These methods deal with static images, where altering the mouth shape (e.g. opening a closed mouth to show surprise) is acceptable, if not desired. However, without placing any specific constraint on the mouth region, lip synchronization may be lost when they are directly applied to video sequences.

More related to this task, is the face reenactment problem, where the facial performance of a source actor is transferred to a target one, making the latter mimic the expressions of the former. State-of-the-art techniques [13, 23, 51] achieve compelling photo-realism by training a neural renderer conditioned on a facial representation (e.g. 3DMMs). Nonetheless, this is substantially different from semantically controlling the target actor, since the expressions are merely copied from another subject. Instead, we would like to edit the actor’s own expressions based on the desired emotion, while preserving the mouth motion. Recent methods address only one aspect of this problem. For instance, DSM [39] generates novel expressions based on emotional labels without retaining the original speech, while [22] preserves mouth movements, but the proposed manipulation is limited to matching a single target speaking style.

In this work, we propose a hybrid method, in which a parametric 3D face representation is translated to different domains, and then used to drive synthesis of the target face by means of a video-based neural renderer. Our method, which we call Neural Emotion Director (NED), achieves photo-realistic manipulation of the emotional state of actors in “in-the-wild” videos, see e.g. Fig. 1. It can translate a facial performance to any of the 6 basic emotions (angry, happy, surprise, fear, disgust, sadness) plus neutral, using only as input its semantic label, while retaining the original mouth motion. It also allows to attach a specific style to the target actor, without requiring person-specific training. This means that the reference style can be extracted at test time from any given video: our system can, for example, make Robert De Niro yell in the way of Al Pacino, without ever seeing footage of the latter during training. Our contributions can be summarized as follows:

- To the best of our knowledge, we propose the first video-based method for “directing” actors in “in-the-wild” conditions, by translating their facial expressions to multiple unseen emotions or styles, without altering the uttered speech.
- We introduce an emotion-translation network, which we call 3D-based Emotion Manipulator, that receives a sequence of expression parameters and translates them to a given target domain or a reference style and is trained on non-parallel data. We train this network on 2 large video databases annotated with emotion labels.

• We design a video-based face renderer, to decode the parametric representation back to photo-realistic frames. Building upon robust, state-of-the-art face editing techniques (face segmentation, alignment, blending) we modify only the face area, while the background remains unchanged, making it possible to manipulate challenging scenes.

- We conduct extensive qualitative and quantitative experiments, user studies and ablation studies to evaluate our method and compare it with recent state-of-the-art methods. The experiments demonstrate the effectiveness and advantages of our method, which achieves promising results in very challenging scenarios as the ones encountered in movie scenes with moving background objects.

- We release our code and trained models [1].

2. Related Work

Face manipulation methods can be divided according to whether they directly edit face portraits through convolutional architectures, or they rely on a geometric face representation:

Image-based emotion editing. The introduction of GANs [15] has sparked a growing line of research in the field of image and video synthesis. The vast majority of works utilise a conditional generator, in the sense that the synthesized image is conditioned on another image (e.g. [19]). This enables translating images between different domains (i.e. image-to-image translation) while preserving the content of the source image, even by training on non-parallel data through the idea of cycle consistency [54]. The use of such techniques on face images enables the altering of facial attributes (e.g. hair color, gender etc.) and constitutes a major part of the so-called area of deepfakes. The multi-domain framework of StarGAN [7] demonstrated the potential of altering the facial emotions in images by translating them according to the given semantic label (e.g. happy, angry etc.). Other techniques make use of continuous emotion labels, such as the intensity [12], or the Valence-Arousal space [32]. Recently, the proposed method of GANmut [9] introduced a way of obtaining a 2D interpretable conditional label system even when using a dataset annotated with solely categorical labels of basic emotions. However, all the above methods translate static frames without taking into account the dynamic nature of facial performance. This is especially essential in the mouth area, as the conveyed speech may be distorted if such techniques are applied independently to every frame of a video. Moreover, they are usually trained on large datasets of images, containing several different identities, which is likely to cause an identity leakage, e.g. in cases where a closed
mouth is replaced with a smile revealing the teeth of another identity. At the same time, progress in the field shows the potential of generating diverse versions of a given image, by conditioning the generator on dense representations rather than coarse domain labels [8]. To overcome the aforementioned limitations, we utilise a GAN-based domain-to-domain translation method (inspired by StarGAN v2 [8]), which translates sequences of subject-agnostic parametric representations of facial expressions instead of images. Then, our person-specific face renderer ensures that the manipulated expressions are synthesized in an identity-preserving way.

**Geometry-based face manipulation.** In the last years, the problem of manipulating faces on a parametric space has attracted increased interest. Face reenactment is the most typical example, where the target actor is forced to mimic the expressions of a source subject in a reference video. Some works utilise 2D facial landmarks for capturing the expressions and driving the target actor either via image-warping [2] or neural rendering [51]. 3D Morphable Models (3DMMs) [3] are a very popular choice, as they offer a disentangled representation of expressions from identity. Traditional techniques [40], [41] perform 3D face reconstruction on the reference video and render the target subject under the source expressions on top of the original target footage. Learning-based methods, like DVP [23] and Head2Head++ [13] use conditional GANs to render the target subject under the given conditions (expressions, pose, eye-gaze).

Nevertheless, these methods offer no semantic control over the generated video, as they directly copy the expressions from a source actor. ICface [43] and FACEGAN [44] present a more intuitive animation framework by conditioning synthesis on Action Units (AU) values, but setting individual AU values is a cumbersome process and requires expertise to achieve the desired emotion. Solanki and Roussos [39] train a decoder network that maps Valence-Arousal pairs to expression coefficients of a 3D face model, and synthesize the target actor with a neural renderer. Their method, however, totally ignores the original expressions and mouth motion of the actor. Groth et al. [16] try to alter the emotional state of an actor by merely interpolating between his/her expressions and the MoCap data obtained from a reference actor. Kim et al. [22] presented a style-preserving solution to film dubbing, where the expression parameters of the dubber pass through a style-translation network before driving the performance of the foreign actor. Their method preserves the dubber’s speech, but can only translate between a pair of speaking-styles (dubber-to-actor). Other methods for generating emotional talking faces include audio-driven [20] and text-driven [48] techniques. To the best of our knowledge, there is no systematic way of translating an existing facial performance in a video to multiple emotions given only semantic information as input, while preserving the original speech. Our method offers an automatic solution to this task through the proposed 3D-based Emotion Manipulator. It does not attempt to handle the specific speaking styles of two predefined actors (as in [22]), but is able to translate the expressions of any subject to any basic emotion or a given reference style.

**3. Method**

Our Neural Emotion Director (NED) framework addresses the challenging task of emotion-related semantic manipulation of faces in videos while preserving their speech-related mouth motion. An outline of the proposed pipeline at test time is presented in Fig. 2. It consists of three main modules (3D Face Analysis, 3D-based Emotion Manipulator and Photo-realistic Synthesis “in the wild”) that are presented in the following sections.

**3.1. 3D Face Analysis**

**Face detection and segmentation:** We first perform face detection, cropping and resizing to $256 \times 256$ pixels using [52]. We then apply FSGAN [37] to segment the face and remove the background.

**3D face reconstruction:** We harness the power of 3DMMs [3] to estimate the 3D face geometry, while disentangling the expression contributions from the identity-specific and 3D pose ones. This enables us to map the emotion translation problem from the image space to the space of 3D model parameters in a subject-agnostic manner. We perform deep 3D face reconstruction with the recent state-of-the-art method of DECA [14] that uses the FLAME model [31]: For each frame of the input video, DECA regresses the parameters of the camera $c \in \mathbb{R}^3$, head pose $p \in \mathbb{R}^6$ (including 3 jaw articulation parameters), identity $a \in \mathbb{R}^{100}$, expression $e \in \mathbb{R}^{50}$, as well as the person-specific detail vector $d \in \mathbb{R}^{128}$, which adds mid-frequency details to the face geometry. We use the latter to create detailed shape images $S \in \mathbb{R}^{256 \times 256 \times 3}$ (see 3D facial shapes in Fig. 2).

**Landmark detection and face alignment:** We use FAN [4] to obtain 68 facial landmarks for each frame. Afterwards, similarly to [13], we estimate eye pupil coordinates based on the inverse intensities of the pixels within the eye area and create eye images $E \in \mathbb{R}^{256 \times 256 \times 3}$ that provide the face renderer with information about the eye-gaze. However, in contrast to [13], we only draw two red disks around eye pupils and not the edges of the outline. This is because we accurately integrate information about eye blinking within the NMFC and detailed shape images (see Sec. 3.3), thanks to the reliable reconstructions in the eye regions obtained by DECA [14]. We then align all face frames to a face template, based on the extracted face landmarks and Procrustes analysis. We found that such face alignment boosts our face renderer’s generalization ability.
We apply the same alignment to the NMFC, shape, and eye images. For more details on the face alignment step, please refer to the Supp. Material.

3.2. 3D-based Emotion Manipulator

Following the 3D Face Analysis step, information related to the facial expression in a frame is encoded in the expression vector $\mathbf{e} \in \mathbb{R}^{50}$ and the 3 jaw parameters (formed as the last 3 components of the pose vector $\mathbf{p} \in \mathbb{R}^{4}$). We expand the 50 expression parameters $\mathbf{e}$ with the jaw opening $\mathbf{p}_4$ (the 1st jaw articulation parameter), as this is the main parameter that describes speech-related mouth motions. Thus, we concatenate the two into a single vector, namely the full expression vector $\mathbf{e} = (\mathbf{p}_4; \mathbf{e}) \in \mathbb{R}^{51}$, henceforth called expression vector for simplicity. To cope with the dynamic nature of facial expressions, we group frames into $N$-length sequences $\mathbf{s} = (e^n, ..., e^{n+N-1})$, with $N=10$. Given the set $\mathcal{Y}$ of the $c=7$ emotion labels (neutral, happy, fear, sad, surprised, angry, disgusted), each denoting a distinct domain, and the set $\mathcal{S}$ of sequences of expression parameters, we design a 3D-based Emotion Manipulator that translates a sequence of expression vectors $\mathbf{s} \in \mathcal{S}$ to a given emotion $y \in \mathcal{Y}$ in a realistic way that preserves the original mouth motion. Inspired by the StarGAN v2 [8] framework, which offers diversity in the generated samples by conditioning the generator on a continuous style vector, we design an architecture with the following four modules:

**Expressions Translator:** The translator $G$ takes as input a sequence of expressions $\mathbf{s}$ and a style vector $\mathbf{d} \in \mathbb{R}^{16}$ and translates $\mathbf{s}$ into an output sequence of expression vectors $G(\mathbf{s}, \mathbf{d}) \in \mathcal{S}$ that reflects the speaking style encoded in $\mathbf{d}$. To inject $\mathbf{d}$ into $G$, we concatenate $\mathbf{d}$ with each of the $N$ vectors of the sequence.

**Style encoder:** Our style encoder $E$ extracts the emotion-related style vector $\mathbf{d} = E(\mathbf{s})$ of an input sequence $\mathbf{s}$ and, thus, enables the translator $G$ to translate a given sequence according to the speaking style extracted from a reference sequence. In contrast to [8], our style encoder does not require any knowledge about the ground truth emotion label $y$ of the reference sequence $\mathbf{s}$.

**Mapping network:** The mapping network $M$ learns to generate style vectors $\mathbf{d} = M_p(\mathbf{z}) \in \mathbb{R}^{16}$ related to a target emotion $y \in \mathcal{Y}$, by transforming a latent code $\mathbf{z} \in \mathbb{R}^{4}$ sampled from a normal distribution. Here, $M_p(\cdot)$ denotes the output branch of $M$ that corresponds to the emotion $y$. This network allows the translator to translate a sequence of expressions to a target emotion, by merely sampling random noise, and specifying the desired semantic emotion label.

**Expressions Discriminator:** Our discriminator $D$ has $c=7$ branches (similarly to $M$) and learns to discriminate between real $\mathbf{s}$ and fake $G(\mathbf{s}, \mathbf{d})$ sequences of each domain $y$ by outputting a scalar value $D_y(\mathbf{s})$ for each branch.

The network $M$ follows a simple MLP architecture, whereas $G$, $E$ and $D$ use recurrent architectures with LSTM units [18].

3.2.1 Training and testing of the Emotion Manipulator

Given a dataset of sequences of expression vectors $\mathbf{s} \in \mathcal{S}$ and their corresponding ground truth labels of emotions $y \in \mathcal{Y}$, we train our networks in 2 alternating steps: 1) first we sample $\mathbf{z} \in \mathbb{R}^4$ from a normal distribution, we randomly pick a target domain $\hat{y} \in \mathcal{Y}$, and employ our mapping network for generating the speaking style $\hat{\mathbf{d}} = M_{\hat{y}}(\mathbf{z})$. 2) then we directly extract the style from a reference sequence $\mathbf{s}$ with our style encoder $\hat{\mathbf{d}} = E(s)$ and store the reference label $\hat{y}$. In both cases, the translator combines an input sequence $\mathbf{s}$ (belonging to domain $y$) with the style vector $\mathbf{d}$ and produces an output sequence $G(s, d)$ belonging to the target domain $\hat{y}$ and resembling the speaking style in $\mathbf{d}$. The networks are then updated using the following objectives.

**Adversarial loss:** We use LSGAN [34] with labels $b_m=1$ for real samples and label $a=0$ for fake ones. This way the
mapping network $M$ learns to output the speaking styles that belong to the emotional domain $\tilde{y}$ and the translator to produce sequences of the target domain that are indistinguishable from the real ones.

**Style reconstruction loss:** As in [8], we make sure the output sequence reflects the given style by using a loss that enforces the style vector of the translated sequence, as extracted by the style encoder $E$, to match the desired one.

**Cycle consistency loss:** We use the cycle consistency loss [7, 54], which encourages the translator to produce sequences that preserve the content of the input sequence, so that the input sequence can be reconstructed by translating the output sequence back to the original style $\tilde{d} = E(s)$, as extracted by $E$.

**Speech-preserving loss:** As observed in [22], the cycle consistency loss does not always guarantee that the original mouth motion related to speech is preserved by the translator. To this end, we leverage our carefully selected FLAME model [31], which explicitly controls the mouth opening through the $1^{st}$ jaw parameter. Thus, we add an extra constraint to the total objective, that takes into account only this mouth related parameter, instead of the whole expression vector as in [22]. By properly defining this objective as the maximization of the correlation between the original and the translated jaw opening variable, we manage to balance our challenging and contradictory goal of altering the emotion without distorting the perceived speech (see Fig. 3).

**Overall objectives:** The objective for $G$, $E$ and $M$ corresponds to a weighted summation of the Adversarial, Style reconstruction, Cycle consistency and Speech-preserving losses. The objective for $D$ corresponds to the discriminator loss. More details and mathematical formulæ for the adopted loss functions can be found at the Supp. Material.

We train our 3D-based Emotion Manipulator on two video databases with annotations of the 6 basic emotions plus neutral: the Aff-Wild2 database [24–30, 50] of “in-the-wild” videos and the MEAD database [46] (we exclude contempt for MEAD to match the emotions in Aff-Wild2). We recover the expression parameters for every frame of the videos and extract sliding windows of length $N$. To get the best of the two databases, we pre-train our networks in Aff-Wild2 and then fine-tune them on a subset of MEAD.

During testing, to transform the expressions of a whole input video, we slide the $N$-length window by 1 frame at a time, translate the sequence through $G$, and use a weighted averaging of Gaussian type to handle overlaps. The conditional style vector is either generated by $M$ by choosing a target emotion or extracted from a reference video of arbitrary length by $E$. In the latter case, we process the whole reference video sequentially, extracting a sequence of style vectors adopting the same sliding pattern. We then take the geometric median [45] of them as the style vector representing the whole reference video and feed it to $G$.

### 3.3. Photo-realistic Synthesis “in the wild”

**3D Face Synthesis & Rendering:** Having modified the expression parameters through our 3D-based Emotion Manipulator, we synthesize a manipulated 3D face geometry under the new emotion. We then render it (using conventional 3D graphics) to a convenient representation for neural rendering, the so-called Normalized Mean Face Coordinate (NMFC) image [13], and concatenate it with the similarly-rendered detailed shape image $S$ and eye image $E$.

**Neural Face Renderer:** We feed our neural renderer with the NMFC, $S$ and $E$ images as conditional input. We build it upon the publicly available Head2Head++ [13] implementation and train it on the training footage of a target actor in a self-reenactment fashion (i.e. with the original face geometry). We follow the recurrent scheme of [13] by feeding the generator with the conditional inputs of both the current as well as the two previous frames, along with the two previously generated images. However, in contrast to [13], we include $S$ as additional conditional input, as described above, and constrain image synthesis to the aligned and masked faces, since we account for changing background. As in [13] we employ a dedicated mouth discriminator to enhance realism in the mouth area.

**Blending:** Face alignment is reversed by transforming the generated images according to the inverse of the previously stored alignment matrix. We then carefully blend the synthesized face with the original background, enabling the manipulation of “in-the-wild” videos. For this, we use multi-band blending [5], as we found it to perform better than soft-masking or Poisson editing [38] in terms of smooth boundary transition.

For more details on this module, please refer to the Supp. Material.

### 4. Experimental Results

We conduct comprehensive qualitative and quantitative evaluations of our method and comparisons with recent state-of-the-art methods. Additional results and visual-
Our experiments use the following datasets: **YouTube Actors dataset**: We collected a small dataset from 6 YouTube videos that included facial videos of 6 actors during film scenes, TV shows and interviews. **MEAD dataset**: We chose 3 actors from the recent MEAD database [46]. For every actor, we selected 30 videos for each of the 6 basic emotions (happy, angry, surprised, fear, sad, disgusted) plus neutral, resulting in a total of 630 videos from MEAD. For more details about the datasets used in the experiments, please refer to the Supp. Material.

A person-specific face renderer had to be trained separately for each actor of these datasets. Yet, we found that by training a new model from scratch for a given actor, the generator tends to overfit to this actor’s idiosyncrasies failing to synthesize novel emotions, if those were not present in his/her training footage. For YouTube actors, finding a few-minute-long footage that covers the total range of emotions is often impossible. To overcome the challenge of generating unseen expressions of an actor while preserving his/her identity, we propose training a single meta-renderer on a mix of videos including YouTube and MEAD actors, and then, fine-tuning the meta-renderer independently for each actor. This helps by sharing the expressivity of the MEAD actors among the YouTube actors. For a further explanation of this process please refer to the Supp. Material.

We compare our method with the following recent methods: **GANmut** [9], which transforms an input cropped facial image according to a 2D continuous emotion label. The transformed image is then placed in its original position in the full frame. For a fair comparison, we use the 2D vectors that correspond to one of the pure 7 emotions with maximum intensity. Also, we apply the method in every frame of the input video. **ICface** [43], which also transforms an input cropped image, but based on the determination of pose and AU values. For a fair comparison, this method is only included in the self-reenactment experiments, where the ground truth is available and AU values can be extracted from it in a well-defined and consistent manner. **DSM** [39], which performs semantic manipulation of facial videos based on a categorical emotion label. The labels that this method supports are neutral and only 4 basic emotions (happy, surprised, fear, sad), which is a subset of the labels considered by the other methods and thus the relevant comparisons with DSM are done in this subset. It is worth mentioning that the literature of photo-realistic emotion manipulation of faces in videos is still extremely limited and the aforementioned 3 methods were the only ones for which we could find source code to run for our evaluations. Details about running our method and the methods we compare with can be found at the Supp. Material.

**4.1. Quantitative Comparisons**

To assess the performance of each method in manipulating the emotion we use a variant of the self-reenactment task that includes the manipulation of the expressions. In particular, given a video of emotional label $y$ (e.g. happy), we modify its emotion by defining the target emotion as the...
same label $y$ (from happy to happy). In this case, the output video should match the original one. Specifically, the discrepancy between the “self-translated” and the real video is measured using the following metrics: 1) Face Average Pixel Distance (FAPD): which is the mean $l_2$ distance of RGB values across all facial pixels and frames, between the ground truth and generated video. We use the extracted face mask to define the face area. 2) Frechét inception distance (FID) [17]: which is computed by using the feature vectors from a state-of-the-art face recognition network [10] for all the ground truth and the generated frames.

Our 3D-based Emotion Manipulator can translate the sequences to the same domain by simply using their own style vector as extracted by the style encoder $E$. However, for the other 3 methods, the label $y$ of the original video has to be known so that it can be used as the target label. Therefore, our quantitative comparison is performed in our MEAD dataset, in which videos are emotionally annotated by the authors of [46]. Specifically, for each actor we use $4$ videos per emotion, leading to a total of $84$ videos, of average duration $\sim 3$ secs. Results are presented in Tab. 1. For a visual comparison please refer to Fig. 5. Note that disgusted and angry are not supported by the DSM method.

As can be seen, our method outperforms the baselines in both metrics overall. We also exhibit superior performance in almost all 7 emotions individually. This shows the higher realism of our synthesized videos as well as the better expression transferability in terms of identity preservation (see Fig. 5 for artifacts produced by GANmut and ICface).

4.2. User Studies

We also conducted two web-based user studies:

**Emotion Recognition and Realism on MEAD Database:** In the first user study, participants were shown randomly shuffled manipulated videos of $3$ actors from MEAD database in all $6$ basic emotions and were asked to rate the realism of the footage on a Likert 5-point scale, as well as recognize the emotion shown (from a drop-down list including all $6$ emotions). Apart from our method, the questionnaire included videos by GANmut [9], DSM [39], as well as the original real videos from MEAD. In total the questionnaire included $66$ videos and $20$ participants completed it. The results can be seen in Tab. 2 where we observe that all methods have relatively low realism scores. This can be attributed to the fact that the real videos in MEAD include particularly intense expressions, which probably resulted to an overall low frequency of ratings of $4$ or $5$ (even for real videos) and to an increased tendency to use these ratings (whenever they were used) more exclusively for real videos. However, we see that our method achieves significantly higher realism scores than the other methods, consistently across all $6$ emotions. In terms of emotion recognition accuracy, we observe that our method synthesized videos with manipulated emotions that were consistently easier to be recognised by the participants, in comparison to DSM. However, this is not the case when we compare our method with GANmut: The synthetic videos of GANmut achieved a very high accuracy rate, which is even higher than the accuracy for real videos. This, in combination with the low realism score of GANmut (see also the “exaggerated” emotions in Figs. 4 and 5), reflects the fact that GANmut synthesizes intense expressions that typically look fake but are easily recognizable.

**Realism on YouTube Actors:** In the second study, we presented users with manipulated videos (including the original audio) of $6$ YouTube actors in all $6$ basic emotions and asked them to rate the realism of the footage, following the same protocol as in the first study. We did not evaluate emotion recognition in this study since ground truth emotion annotations do not exist for this dataset (to compare with). The study included a random shuffling of videos manipulated by our method and GANmut, as well as the original videos. For some indicative frames of these videos, please refer to Fig. 4. DSM was not used for this study, since it cannot handle dynamic backgrounds such as those found in YouTube videos. The questionnaire included $54$ videos in total and was completed by $50$ participants. The ratings obtained can be seen in Tab. 3. We observe that the realism scores for both methods are relatively low, which can be attributed to the highly challenging task of manipulating the emotions in videos, especially under “in-the-wild” conditions as is the case for the YouTube Actors dataset. However, our method achieves a better score than GANmut and for example succeeds in synthesizing realistic videos more than $20\%$ of the times for $3$ out of $6$ actors, which is a promising result that shows the potential of our approach. Furthermore, in terms of the most frequent rating, we see that our method is consistently better than GANmut, as it yields a rating of $3$ or $2$ as the most frequent answer for almost all actors, in contrast to GANmut that yields the rating of $1$ as the most frequent.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>neutral</td>
<td>14.9</td>
<td>16.8</td>
<td>25.1</td>
<td>40.0</td>
</tr>
<tr>
<td>happy</td>
<td>17.8</td>
<td>15.0</td>
<td>27.0</td>
<td>43.7</td>
</tr>
<tr>
<td>fear</td>
<td>19.0</td>
<td>20.5</td>
<td>28.0</td>
<td>43.0</td>
</tr>
<tr>
<td>sad</td>
<td>19.0</td>
<td>18.1</td>
<td>24.5</td>
<td>38.6</td>
</tr>
<tr>
<td>surprised</td>
<td>18.9</td>
<td>19.1</td>
<td>27.3</td>
<td>46.1</td>
</tr>
<tr>
<td>angry</td>
<td>18.4</td>
<td>22.4</td>
<td>-</td>
<td>51.7</td>
</tr>
<tr>
<td>disgusted</td>
<td>18.1</td>
<td>15.1</td>
<td>-</td>
<td>39.7</td>
</tr>
<tr>
<td>avg. (7)</td>
<td>17.9</td>
<td>18.1</td>
<td>-</td>
<td>43.3</td>
</tr>
<tr>
<td>avg. (5)</td>
<td>17.8</td>
<td>17.9</td>
<td>25.8</td>
<td>42.3</td>
</tr>
</tbody>
</table>

Table 1. Quantitative comparisons on MEAD in the emotional “self-translation” experiment. Bold values denote the best value for each metric (lower is better). Averaging is done over both the full set of 7 emotion labels and the set of 5 labels supported by DSM [39], for the sake of fair comparison.
Table 2. Realism ratings (percentage of users that rated the videos with 4 or 5) and classification accuracy of the user study on MEAD (c.f. Supp. Material for detailed scores).

<table>
<thead>
<tr>
<th></th>
<th>Realism</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ours</td>
<td>GANmut</td>
</tr>
<tr>
<td></td>
<td>DSM</td>
<td>Real Videos</td>
</tr>
<tr>
<td>happy</td>
<td>17%</td>
<td>3%</td>
</tr>
<tr>
<td>fear</td>
<td>32%</td>
<td>7%</td>
</tr>
<tr>
<td>sad</td>
<td>20%</td>
<td>18%</td>
</tr>
<tr>
<td>surprised</td>
<td>22%</td>
<td>8%</td>
</tr>
<tr>
<td>angry</td>
<td>25%</td>
<td>10%</td>
</tr>
<tr>
<td>disgusted</td>
<td>40%</td>
<td>20%</td>
</tr>
<tr>
<td>avg.</td>
<td>28%</td>
<td>11%</td>
</tr>
</tbody>
</table>

Table 3. Realism ratings of the user study on 6 YouTube actors. Columns 1-5 show the number of times that users gave this rating. The column “real” shows the percentage of users that rated the videos with 4 or 5. Bold values denote the most frequent user rating for each method and actor.

<table>
<thead>
<tr>
<th></th>
<th>Realism</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ours</td>
<td>GANmut</td>
</tr>
<tr>
<td></td>
<td>DSM</td>
<td>Real Videos</td>
</tr>
<tr>
<td>McDormand</td>
<td>32 32</td>
<td>52 21</td>
</tr>
<tr>
<td>Pacino</td>
<td>19 45</td>
<td>53 25</td>
</tr>
<tr>
<td>Tarantino</td>
<td>70 29</td>
<td>23 17</td>
</tr>
<tr>
<td>McConaughey</td>
<td>37 63</td>
<td>33 13</td>
</tr>
<tr>
<td>Roberts</td>
<td>34 60</td>
<td>39 12</td>
</tr>
<tr>
<td>Foxxy</td>
<td>26 35</td>
<td>39 34</td>
</tr>
<tr>
<td>avg.</td>
<td>36 44</td>
<td>40 20</td>
</tr>
</tbody>
</table>

Table 4. Ablation study results under the self reenactment setting, averaged across three YouTube actors from our dataset. For all metrics, lower values indicate better performance. Bold and underlined values correspond to the best and the second-best value of each metric, respectively.

Note on social impact. Despite their positive impact, deep learning systems for video manipulation have raised concerns related to the distribution of fake news and other negative social impact [6, 11, 21, 47]. While our goal of speech preservation is inherently opposite to most deepfakes where the output combination of a person and its utterances is entirely fake, our method could also be misused in cases where the conveyed meaning heavily depends on the apparent emotion (e.g. political speeches). We believe that scientists working in the relevant fields need to seriously take into account these risks and ethical issues. Some of the countermeasures include contributing in raising public awareness about the capabilities of current technology and developing systems that detect deepfake videos [35, 42, 49].

6. Conclusion

We proposed Neural Emotion Director (NED), a novel approach for photo-realistic manipulation of the emotions of actors in videos. Our new 3D-based Emotion Manipulator translates facial expressions by carefully preserving the speech-related content of the source performance, while our Photo-realistic Synthesis module faithfully synthesize the target actor’s face and composites it onto the original video. Our extensive experimental results demonstrate the advantages of our framework over recent state-of-the-art methods and its effectiveness under “in-the-wild” conditions. Acknowledgments. A. Roussos was supported by the Hellenic Foundation for Research and Innovation (HFRI) under the “1st Call for HFRI Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment” Project I.C.Humans, Number: 91.
References


