International Journal of Computer Vision (2019) 127:1767–1779
https://doi.org/10.1007/s11263-019-01150-y

You Said That?: Synthesising Talking Faces from Audio

Amir Jamaludin[1] · **Joon Son Chung**[1] · **Andrew Zisserman**[1]

Received: 28 February 2018 / Accepted: 16 January 2019 / Published online: 13 February 2019

Abstract
We describe a method for generating a video of a talking face. The method takes still images of the target face and an audio speech segment as inputs, and generates a video of the target face lip synched with the audio. The method runs in real time and is applicable to faces and audio not seen at training time. To achieve this we develop an encoder–decoder convolutional neural network (CNN) model that uses a joint embedding of the face and audio to generate synthesised talking face video frames. The model is trained on unlabelled videos using cross-modal self-supervision. We also propose methods to re-dub videos by visually blending the generated face into the source video frame using a multi-stream CNN model.

Keywords Computer vision · Machine learning · Visual speech synthesis · Video synthesis

1 Introduction

There has been much work recently in the area of transforming one modality to another. Image to text is the most prominent, e.g. in caption generation (Vinyals et al. 2015; Karpathy and Fei-Fei 2015; Xu et al. 2015), but there is also text to image (Reed et al. 2016), video to sound (Owens et al. 2016), or in fact a combination of different mediums e.g. video and audio to text (Chung et al. 2017). This paper considers the case of images and audio to video.

We propose a method to generate videos of a talking face using only an audio speech segment and face images of the target identity. The speech segment need not be spoken originally by the target person (see Fig. 1). We dub the approach *Speech2Vid*. Our method differs from previous approaches for this task (see related work below) in that instead of learning phoneme to viseme mappings, we learn the correspondences between audio features and video data directly. By focusing on the speech portion of audio and tight facial regions of speakers in images, the Speech2Vid model is able to produce videos of a talking face at test time even when using images and audio outside of the training dataset.

The key idea of the approach is to learn a joint embedding of the target face and speech segment that can be used to generate a frame of that face saying (lip synched with) the speech segment. Thus the inputs are still images of the face (that provides the identity, but is not speaking the target segment) and the target speech segment; and the generated output is the target face speaking the segment.

The Speech2Vid model is learnt from unlabelled videos using a form of cross-modal self-supervision; unlabelled here refers to the fact that the videos used for training were not explicitly annotated by humans. The approach learns to predict the face in a target frame of the video where the audio (and frame) is known, using other frames of the target video to provide the still (identity) images of the face.

There are numerous possible application to Speech2Vid, for example: re-dubbing videos with other languages, generating possible keyframes to help animate mouth movements of animated characters in 3D animation or video-games, lip-syncing mouth movements in music videos (these videos are dubbed manually and can produce jarring lip movements) and so on. Another potential application is re-animating characters from TV shows with new audio as suggested in Charles et al. (2016).

In the following, we first describe the architecture and training of the Speech2Vid model in Sect. 3. The automatic pipeline to prepare the video dataset used to train the generation network is described in Sect. 4. Section 5 reports quantitative results, and assesses variations on the architecture, including varying the number of images of the identity

Amir Jamaludin and Joon Son Chung contributed equally to this work.

✉ Joon Son Chung
joon@robots.ox.ac.uk

[1] Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK

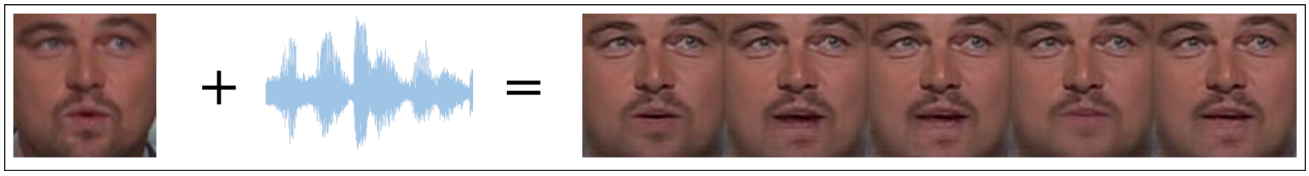

Fig. 1 The Speech2Vid model generates a video of a talking face, given still images of the person and a speech segment. In the output video the talking face is lip synched with the audio. Note, only one input still image is shown in the figure, but the method can ingest multiple images. The face need not be in the training dataset, i.e. the Speech2Vid is applicable to unseen images and speech

used as input. Finally, Sect. 6 shows an application to video re-dubbing by visually blending the generated face into the source video frame.

2 Related Work

There are various works that propose methods to generate or synthesise videos of talking heads from either audio or text sources. The works can be divided on a number of axes: is raw audio used or is the audio represented phonetically? Are the new frames generated by frame reselection from the source video or are they synthesised from external images? Is the method only trained for one identity or is it applicable to any at test time?

The majority of the existing works are based on frame reselection from a video. For example, Fan et al. (2015) proposes a method based on a bi-directional LSTM that selects a target mouth region from a dictionary of saved target frames. The lower half of the face from the selected frame is then blended back into the background face. Similarly, Charles et al. (2016) trains a model to select visemes based on the phonetic label of the audio, and enforces visual smoothness by matching the visual features of the last frame of one viseme to the first frame of the next, optimised using the Viterbi algorithm. And Taylor et al. (2017) also uses a phonetic-based method—the audio is first transcribed into a phonetic sequence, from which the animation parameters are generated. The final image here is however generated by a CG animation model, rather than by frame reselection.

The recent work of Suwajanakorn et al. (2017) trains a recurrent neural network to predict the coordinates of key facial landmarks for every frame given the audio, and fills the texture based on the landmarks using frame reselection. The paper proposes a series of post-processing steps like video re-timing and jaw smoothing to produce realistic images. However, it must be retrained for each identity required. One of the closest works to ours is the recent work in Karras et al. (2017) where given an audio sample, they produce 3D vertex coordinates of a face mesh corresponding to the sample. However, unlike ours, this method also must be retrained for each identity.

A different use scenario is investigated in Garrido et al. (2015), which describes a method to transfer the mouth shapes from the video of a dubber to the face in the target video using a 3D model. However, this method requires the video footage of the dubber's mouth saying the speech segment, whereas our method learns the relationship between the sound and the mouth shapes.

Natural Image Synthesis Using CNNs Visual speech synthesis is closely related to the problem of image synthesis, which has seen significant advances in recent years with the success of Generative adversarial networks (GANs) proposed by Goodfellow et al. (2014). Another successful approach, based on sequential pixel prediction, is the PixelRNN and PixelCNN architectures of van den Oord et al. (2016b). The Conditional PixelCNN (van den Oord et al. 2016a) extends this architecture such that the model can be conditioned on any vector, for example, and most relevant to Speech2Vid, the latent embedding of a face.

The recent work of Chen and Koltun (2017) proposes Cascaded Refinement Network that generates realistic-looking images from pixel-wise semantic layout. They use a 'content representation' loss function that has been used previously in image style transfer works (Gatys et al. 2016)—the loss forces the network to match activations of a pre-trained CNN between the generated image and the ground truth, which demonstrates significant benefits over an image-space loss.

Self-supervised Learning Supervised learning has been the most prevalent paradigm in recent computer vision methods, but there is also a good deal of previous work on self-supervised representation learning, where raw data is used as its own source of supervision—which is the approach used to train Speech2Vid. One of the earliest examples of self-supervision is the work on auto-encoders (Hinton and Salakhutdinov 2006), and there are a number of more recent applications on learning representations via data imputation. The work on predicting co-occurrence (Isola et al. 2016), context (Doersch et al. 2015), and colorization (Zhang et al. 2016) fall under this category.

Of more relevance is self-supervision from video, such as Wang and Gupta (2015), Fernando et al. (2017), Misra et al. (2016), Xue et al. (2016), Pătrăucean et al. (2016) and Denton and Birodkar (2017). Recent methods have also investigated

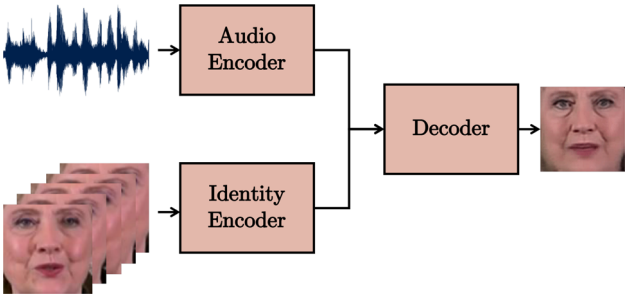

Fig. 2 The overall Speech2Vid model is a combination of two encoders taking in two different streams of data (audio and still images), and a decoder that generates an image corresponding to the audio while retaining identity based on the still images

using multiple modalities, such as video and audio (Arandjelović and Zisserman 2017; Aytar et al. 2016; Chung and Zisserman 2016; Nagrani et al. 2018), though unlike ours, these have not been used to generate video frames given the audio.

3 The Speech2Vid Model

Our main goal at test time is to generate a video of a talking face given two inputs: (i) an audio segment, and (ii) still images of the target identity (frontal headshot). The Speech2Vid model (summarised in Fig. 2 at the block level), consists of three main components: an audio encoder, an identity image encoder and a talking face image decoder. For a given input sample, the model generates one frame of image output that best represents the audio sample at a specific time step. The model generates the video on a frame-by-frame basis by sliding a 0.35-s window over the audio sequence. The frame is moved forward by 1 frame (0.04 s) at a time.

The network is trained on the large-scale video dataset described in Sect. 4, containing over 700K samples. During training, the ground truth output image of the target identity speaking the audio segment is used as supervision. The image is taken from the middle frame of the video in the 0.35-s sampling window. The images for the identity of the speaker are sampled from different points in time from the same video stream, as shown in Fig. 3.

3.1 The Architecture

The Speech2Vid architecture is given in Fig. 4. We describe the three modules (audio encoder, the identity encoder, and the image decoder) in the following paragraphs. Note, these three modules are trained together.

Audio Encoder We use a convolutional neural network originally designed for image recognition. The layer configurations is based on VGG-M (Chatfield et al. 2014), but filter sizes are adapted for the unusual input dimensions (12×35 instead of 224×224 which is the input dimension specified by VGG-M). This is similar to the configuration used to learn audio embedding in Chung and Zisserman (2016).

Identity Encoder Ideally, the identity vector produced by the encoder should have features unique for facial recognition and as such we use a VGG-M network pre-trained on the VGG Face dataset (Parkhi et al. 2015). The dataset includes 2.6M images of 2.6K unique identities. Only the weights of the convolutional layers are used in the encoder, while the weights of the fully-connected layers are reinitialised.

Image Decoder The decoder takes as input the concatenated feature vectors of the FC7 layers of the audio and identity encoders (both 256-dimensional). The features vector is gradually upsampled, layer-by-layer, through bilinear upsampling followed by a convolutional layer. The design of this is similar to the encoder but in reverse order (VGG-M in reverse). See details in Fig. 4. The network features two skip connections to help preserve the defining features of the target identity—this is done by concatenating the encoder activations with the decoder activations (as used in U-Net suggested in Ronneberger et al. 2015) at the locations shown in the network diagram.

3.2 Loss Function

The network is trained with both an image-space, and a content loss. The benefit of using this over only an image-space loss can be seen in Fig. 12.

Image-Space Loss An L_1 loss is used (Eq. 1, $\hat{y}$ is the ground truth and y is the predicted value) between the prediction and the ground truth images. An L_1 loss is known to encourage less blurring than L_2, which is more commonly used for image generation and in auto-encoders (Isola et al. 2017).

$$\mathcal{L}_{=} \sum_{n=1}^{N} ||\hat{y}_n - y_n|| \tag{1}$$

Content Loss An image-space loss between the prediction and the ground truth images can severely penalise realistic outputs, for instance, slightly darker or lighter images that still look realistic. To mitigate this problem, we use the 'content representation' loss proposed by Gatys et al. (2016); Chen and Koltun (2017), which uses a L_1 losses between layer activations from a pre-trained CNN. Here, a pre-trained face recognition (Parkhi et al. 2015) network is used, and the activations from 5 convolution layers (*conv1* to *conv5*) are matched (Fig. 5), so that both fine details and global arrangements can be captured.

As in Chen and Koltun (2017), the relative weight of each loss is given by the inverse of the number of elements in a

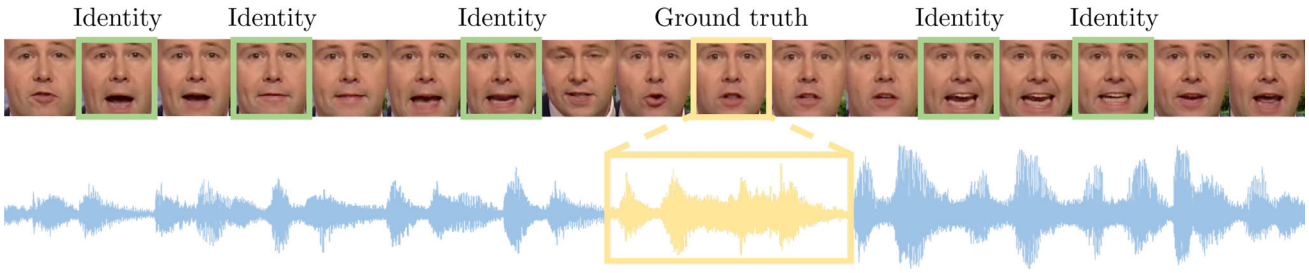

Fig. 3 Sampling strategy for identity images during training. Identities are sampled from past and future frames far from actual (ground truth) audio/output image samples. For each window, we sample five different frames

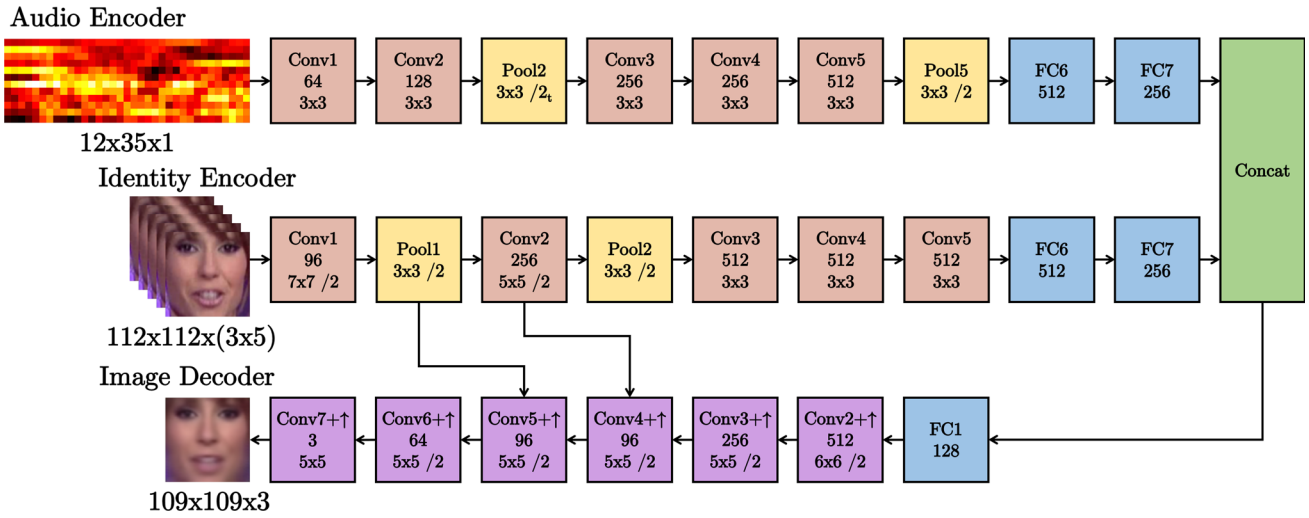

Fig. 4 The three modules in the Speech2Vid model. From top to bottom: (i) audio encoder, (ii) identity encoder with multiple still images as input, and (iii) image decoder. Each Conv layer is followed by a ReLU which is not shown here. /2 refers to the stride of each kernel in a specific layer which is normally of equal stride in both spatial dimensions except for the Pool2 layer in which we use stride 2 in the time-step dimension (denoted by $/2_t$). ↑ refers to up-sampling (by a factor of 2). The network includes two skip connections between the identity encoder and the image decoder

given layer. For example, the weight for the image-space loss is $1/(109 \times 109 \times 3)$ while the weight for the *conv1* content loss is $1/(56 \times 56 \times 96)$.

3.3 Post-processing: Image Sharpening

CNNs trained to generate images tend to produce blurry images (Pathak et al. 2016; Zhang et al. 2016), particularly when trained with an image-space loss. A network trained with the content loss produces sharper images, but there are still benefits to be gained from image sharpening.

We train a separate CNN to sharpen the images produced by the Speech2Vid model. The model is inspired by VDSR (Kim et al. 2016), which uses a residual connection between the input and output, so that the network only has to learn the image difference. The model is trained on the still images in the main training dataset (Sect. 4). Our implementation has 10 convolutional and ReLU layers, and the layer configuration is shown in Fig. 6.

We train the network on artificially blurred face images (Fig. 7), as opposed to training the network end-to-end together with the generator network. This is because the alignments between the input image, the target (ground truth) image and the generated image are not perfect even after the spatial registration (of Sect. 4), and thus avoid the sharpening network having to learn the residual coming from the misalignment. The type of blur applied here is Gaussian, since it closely mimics the blur generated by the networks.

The images that we ask the CNN to sharpen are relatively homogeneous in content (they are all face images), and we find that the CNN performs very well in sharpening the images under this constraint. The results can be seen in Fig. 12.

3.4 Implementation Details

This section describes the input representations for the audio and identity and the network training. The inputs are fed into

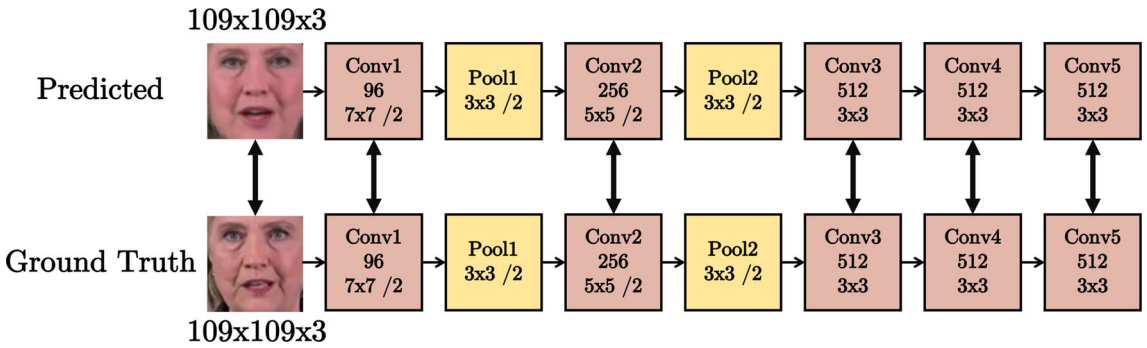

Fig. 5 Image-space and content loss

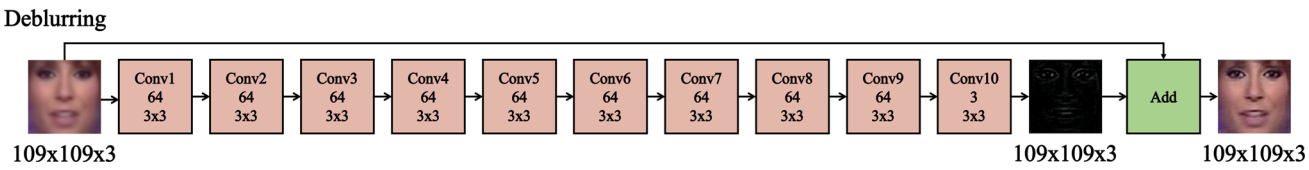

Fig. 6 Image sharpening (deblurring) CNN module

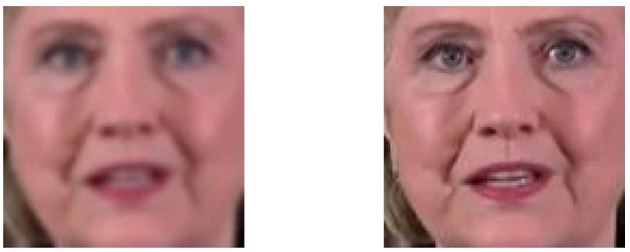

Fig. 7 Inputs for training the image sharpening CNN. Left: Artificially blurred input; Right: Original image (ground truth)

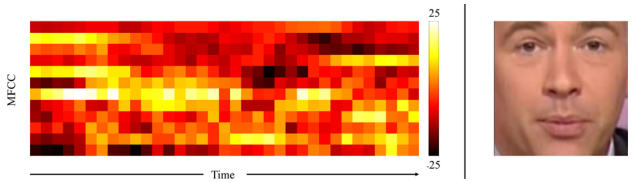

Fig. 8 Inputs to the Speech2Vid model. Left: MFCC heatmap for the 0.35-s time period. The 12 rows in the matrix represent the power of the audio at different frequencies. Right: Still image of the speaker

separate modules in the network in the forms of 0.35-s audio and either (1) a still image, or (2) five still images of the target identity.

Audio The input to the audio encoder are Mel-frequency cepstral coefficients (MFCC) values extracted from the raw audio data. The MFCC values are made up of individual coefficient each representing a specific frequency band of the audio short-term power on a non-linear mel scale of frequency; 13 coefficients are calculated per sample but only the last 12 are used in our case. Each sample fed into the audio encoder is made up of 0.35-s input audio data with a sampling rate of 100 Hz resulting in 35 time steps. Each encoded sample can be viewed as a 12×35 heatmap where each column represents MFCC features at each time step (see Fig. 8).

Identity The input to the identity encoder are still images with a dimension of $112 \times 112 \times 3$. Five images are used per sample which are then concatenated channel-wise, resulting in a input dimension of $112 \times 112 \times (3 \times 5)$. Note that the input image dimensions in the identity encoder (112×112) is slightly different than the output image by the image decoder (109×109) due to the difference in filter sizes between the first layer of the encoder and the last layer of the decoder. This is compensated by cropping the image to the same size as the output when training with the L_1 loss or evaluating with the MSE metric. The difference in filter sizes can also be compensated by introducing some padding after the final layer, but since the results shown in Chung et al. (2017) have used these input and output dimensions, we chose to keep to these dimensions for a fair comparison. Images are chosen with different degrees of mouth openness (10, 30, 50, 70 and 90th percentile in terms of the distance between the top and the bottom lips, from a random sample of images from the face track), using the facial landmark detections, to provide visual examples of teeth, etc. The benefits of using multiple identify images are discussed in Sect. 5.3, where it is shown that this significantly improves the output video quality compared to only using a single identity image.

Training Our implementation is based on the MATLAB toolbox MatConvNet (Vedaldi and Lenc 2015) and trained on a NVIDIA Titan X GPU with 12GB memory. The network is trained with batch normalisation and a fixed learning rate of

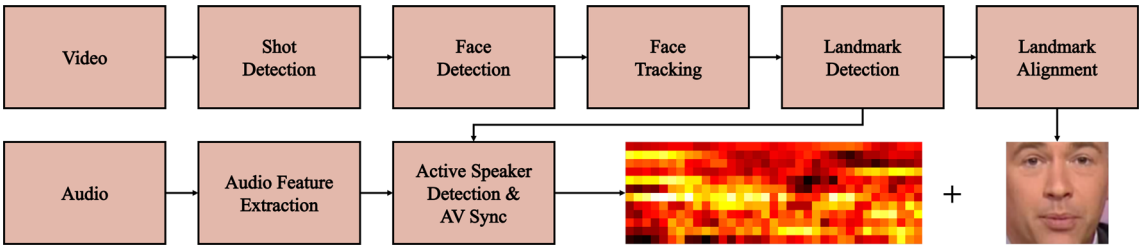

Fig. 9 Data preparation pipeline

10^{-5} using stochastic gradient descent with momentum. The training was stopped after 20 epochs, or when the validation loss stops decreasing, whichever is sooner.

At test time, the network runs faster than twice real-time on a GPU. This can be further accelerated by pre-computing and saving the features from the identity encoder module, rather than running this for every frame. In the case of redubbing video, the output video is generated at the same frame rate as the original video.

3.5 Discussion

The network architecture is based purely on CNNs, as opposed to the recurrent architectures often used for tasks relating to time sequences. Since there is a many-to-one relationship between phonemes and visemes (Ezzat and Poggio 2000; Cappelletta and Harte 2012), the mouth shape of the speaker only depends on what is being said at the exact moment, and some co-articulations from the neighbouring visemes. We find that the 0.35-s window is sufficient to capture this information. At test time, the video is generated frame-by-frame by sliding a temporal window across the entire audio segment while using the same identity images.

4 Video Dataset

This section describes our multi-stage strategy to prepare a large-scale video dataset to train the Speech2Vid network. We obtain tens of hours of visual face sequences aligned with spoken audio.

The principal stages are: (i) detect and track all face appearances in the video; (ii) determine who is speaking in the video; and (iii) align the detected face image to the canonical face. The pipeline is summarised in Fig. 9, and the details are discussed in the following paragraphs.

Video Description We train the Speech2Vid model on videos from which the LRS2 (Afouras et al. 2018) dataset is generated and we test using the test split of the VoxCeleb2 dataset (Chung et al. 2018). These datasets consist of a variety of programs from drama to broadcast news, which provide ideal training data for this task, given that a large proportion of the videos are of high quality and only contain frontal or near-frontal faces. Moreover, the faces are near-frontal and the words are generally clearly spoken without too much background noise, hence provide an easier learning environment for the network. The training data only consists of frontal and near-frontal faces, since the LRS2 dataset was created using a frontal face detector. There are videos with significant head movements in the dataset but these are not excluded for training nor do we have a pre-processing step to handle them.

Face Tracking The face tracking pipeline is based on Chung and Zisserman (2016). First, the shot boundaries are determined by comparing colour histograms (Lienhart 2001) to find the within-shot frames for which tracking is to be run. The HOG-based DLIB face detector (King 2009) is used to detect face appearances on every frame of the video. The face detections are grouped into face tracks using a KLT detector (Lucas and Kanade 1981). Facial landmarks are extracted using the regression-tree based method of Kazemi and Sullivan (2014).

Active Speaker Detection and AV Synchronisation SyncNet (Chung and Zisserman 2016) provides a joint embedding of the audio and visual face sequences in a video, which can be used to determine who is speaking in a multi-speaker video scene. Moreover, the same method is used to correct the lip-sync error in the broadcast video, which can be crucial for precisely locating the corresponding mouth image for the audio sample.

Spatial Registration In order to establish spatial correspondance between the input face (that provides the identity to the encoder) and the output face (from the decoder) in training from the ground truth frames, we register the facial landmarks between the two images. This is done by performing a similarity transformation (scale, rotation and translation) between the faces and an exemplar face with canonical position (Fig. 10 middle). Only the landmarks on the eyes and the nose are used to align the face image, since they are most affected by the head pose and rather than by facial expres-

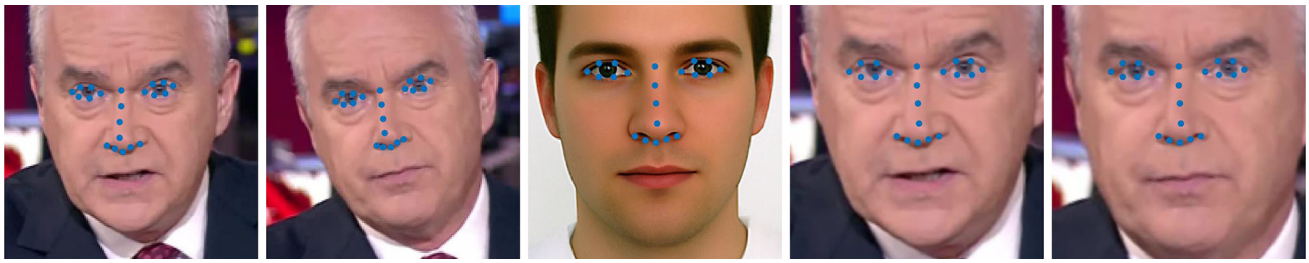

Fig. 10 Left pair: Face images before registration; Middle: Canonical face; Right pair: Face images after registration with the canonical face

Table 1 Data statistics

Set	Source	# Hours	# Samples
Train	LRS2	39.0	701,744
Val	LRS2	2.3	41,874
Test	VoxCeleb2	–	9287

sions. Since the data used as input are aligned, we find that the output of the trained model is always consistent in coordinate which allows us to easily re-align the generated output back to the original space; see Sect. 6.

Training Data Statistics We use the train-validation split (by broadcast date) given in the LRS2 dataset for training. For every valid face track, we extract every 5th frame and the corresponding audio as samples for training and validation, since adding frames in between will not be of much help adding variety to the training. Statistics on the dataset is given in Table 1.

5 Experiments

In this section, we perform several experiments to assess the contribution of various design decisions to the performance of the method. We evaluate the results both quantitativaly and qualititatively. The testing data consists of video clips from the VoxCeleb2 (Chung et al. 2018) dataset. Note, VoxCeleb2 is completely independent from the LRS2 dataset used during training. The results are best seen in video format. Please refer to the online examples.

Figure 11 shows a visualization of the output of the model (the frames of the two segments highlighted in the captions "major" and "based on"). Note, the movement of the mouths of the two examples reflect the sound of each word not unlike phoneme-to-viseme correspondences.

In Fig. 12, it is also interesting to note that the network learns to only move the lower half of the face (i.e. the mouth and surrounding areas), even though the network was trained on the full facial images and the corresponding audio segments.

5.1 Quantitative Analysis

We evaluate the performance of the models using three independent metrics: (1) the pixel-level similarity between the ground truth and the generated image is measured by the MSE distance from the ground truth; (2) the identity preservation is measured by the feature-wise distance between the generated image and the real image using a network pre-trained for face recognition; (3) the correctness of the generated lip shape is tested by retrieving audio samples from the generated image using a network trained for cross-modal lip-to-audio retrieval on real images. The following paragraphs describe the evaluation protocol in more detail.

Pixel-Level Similarity Here, we look at the distance between the generated sample against the ground truth. We also look at changing certain components of the model e.g. using transposed convolution or upsampling and convolution in the decoders. The distances we looked at are pixel-to-pixel mean squared error (MSE) and MSE of the generated samples against the embedding of a VGG Face network.

Table 2 shows quantitative results on 9,287 audio-image pairs from the VoxCeleb2 (Chung et al. 2018) dataset; the audio-image pairs and some identities in this dataset are completely new to our trained network. Unsurprisingly, networks trained using L_1, a direct pixel-to-pixel distance loss, generally fared better when looking at mean squared error than networks trained on content loss which minimised the distance between embeddings. This makes sense, as MSE measures direct pixel-to-pixel distance between the ground truth and the generated samples, which is exactly what the L_1 networks were trained for; this however does not guarantee realistic generations. Interestingly, our best network trained on content loss still outperforms the best L_1 network (327 $\rightarrow$ 333) albeit by a narrow margin.

Identity Preservation We also look at the embedding distance of the generated sample and the ground truth using a pre-trained VGG Face network (Parkhi 2015). In the ideal case, the embedding distance between the ground truth and the generated sample would be zero, as we are generating the image of the same person albeit with a different mouth shape. By this measure, we can see that the sharpening net-

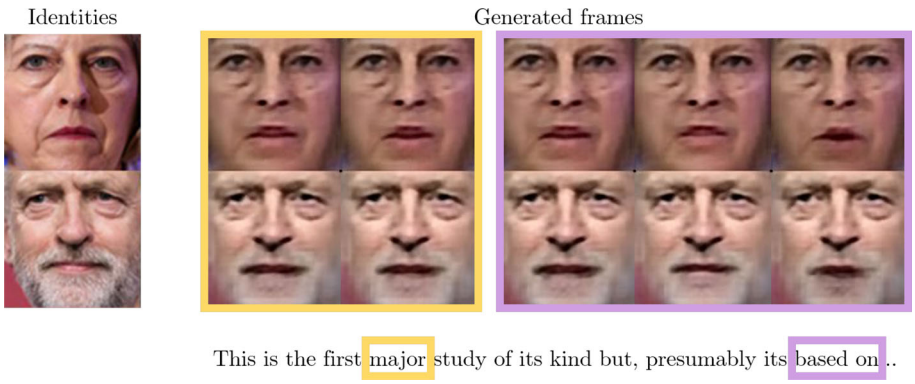

Fig. 11 Top row: Identity 1 and the corresponding generated frames; Middle row: Identity 2 and the corresponding generated frames; Bottom row: Captions of the audio segment

Fig. 12 From left to right: (1) Original input still image to animate; (2) Output frames without skip connection; (3) Output frames with skip connection and one input image; (4) Output frames with skip connection and five input images; (5) Output frames with skip connection and five input images, trained with VGG Face content loss; (6) Output frames with skip connection, five input images and bilinear upsampling instead of transposed convolution, trained with VGG Face content loss. Top row: before sharpening; Bottom row: after sharpening

Table 2 Quantitative results

SC	# id.	Loss	GM	MSE ↓		Embedding dist. ↓		Retrieval Acc. ↑	
Sharpening				✗	✓	✗	✓	✗	✓
Ground truth				0		0		89.5%	
Chance				–		–		9.7%	
✗	1	L_1	TC	705	700	0.434	0.433	79.7%	79.7%
✓	1	L_1	TC	527	533	0.260	0.256	82.7%	82.8%
✓	5	L_1	TC	331	**333**	0.139	0.131	**83.9%**	**83.4%**
✓	5	CL	TC	398	403	0.126	0.125	83.4%	83.0%
✓	5	CL	C+U	**327**	346	**0.118**	**0.115**	82.8%	82.4%

↓ lower is better, ↑ higher is better, Numbers in bold represent best performance
SC skip connection, *id.* identity images, *GM* generation method, *CL* content loss, *TC* transposed convolution, *C+U* convolution + upsampling

work generally improves performance. This might be due to the fact that a sharper image tends to be more realistic; blur seems to be a good tell that an image is generated. For the decoders, the model using convolution and upsampling performs consistently better than transposed convolution in our experiments.

Correctness of the Generated Lip Shape In order to make sure that the generated lip shape corresponds to the speech, we check that the input audio frame can be correctly retrieved from the generated image. We use the network of (Chung et al. 2019) pre-trained for audio-to-video synchronisation and retrieval on *real* videos. The task is to determine the correct synchronisation within a ± 15 frame window, and the synchronisation is determined to be correct if the predicted offset is within ± 1 frame of the ground truth. A random prediction would therefore give 9.7% accuracy. The test is performed on the test split of the LRS2 dataset, so that the results can be compared directly to the results on real videos reported in Chung et al. (2019).

The results are given in Table 2. Although the performance on the generated frames are slightly less than on the ground truth videos, this margin is relatively small and it can be seen that the performance is well above chance. The variation in accuracy among the different models is small.

5.2 Preserving Identity with Skip Connections

Figure 12 shows a set of generated faces and various target identities (original stills). We observe that the skip connections are crucial to carry facial features from the input image to the generated output—without these, the generated images lose defining facial features of target identities, as shown in the middle column. The skip connections at earlier layers (e.g. after *conv1*) were not used as it encouraged the output image to be too similar to the still input, often restricting the mouth shapes that we want to animate.

5.3 Preserving Identity Using Multiple Still Images

As can be seen in Fig. 12, having multiple (five in this case) image examples for the unique identity enhances the quality of the generated faces compared to only having a single example. There are two reasons for this: first, with multiple example images as input, it is likely that the network now has access to images of the person showing the mouth open as well as closed. Thus, it has to hallucinate less in generation as, in principle, more can be sourced directly from the input images; Second, although the faces are aligned prior to the identity encoder, there are minor variations in the movement of the face other than the lips that are not relevant to the speech, from blinking and microexpression. The impact of these minor variations when extracting unique identity features is reduced by having multiple still images of the same person.

6 Re-dubbing Videos

In this section, we propose methods to visually re-dub a source video with a different segment of spoken audio. We develop a multi-stream CNN that can be used to naturally blend the generated mouth shape into the source video frame, and compare the results to a traditional method based on Poisson editing.

6.1 Baseline: Poisson Editing

A method based on Poisson editing is used to blend the output of Speech2Vid model back into the source video.

The key stages are as follows: (i) obtain still images from the source video for identity; (ii) generate the face video for the given audio and identity using the Speech2Vid model; (iii) re-align the landmarks of the generated video to the source video frames, and (iv) visually blend the aligned face with the source video frame.

Alignment Facial landmarks in the target video is determined using the method of Kazemi and Sullivan (2014). A similarity transformation is used to align the generated face with the original face in the target image. Figure 13 (right) shows the generated face in alignment with the original face.

Poisson Editing The Poisson image editing algorithm (Perez et al. 2003) blends two images together by matching gradients with boundary conditions. We use this technique to match the generated face with the source video frame, as shown in Fig. 13. This can be used to blend the face from the same, or different identity to the source video frame.

Discussion This method can be used to blend the generated face as a whole, or to match only the lower half of the face. We qualitatively find that we strike the best balance between image naturalness and movement naturalness by only blending the lower half of the face. However, this method results in unnatural-looking images when the faces are not front-facing, as shown in Fig. 16.

6.2 Network-Based Blending

Here, we propose a modification to the Speech2Vid model, such that it generates an output frame taking account of 'context' from the source video, rather than just 'identity' images.

Architecture The network is the same as the Speech2Vid model described in Sect. 3, but we add another 'context' encoder to capture the information from the source video. The context encoder takes in an image frame with occluded

Fig. 13 Re-blending with the baseline method Top left: Original still image; Top right: Generated mouth region, aligned with the original (target) face; Bottom left: Generated mouth region, superimposed on the original face. Bottom right: Generated mouth region, blended with the original face

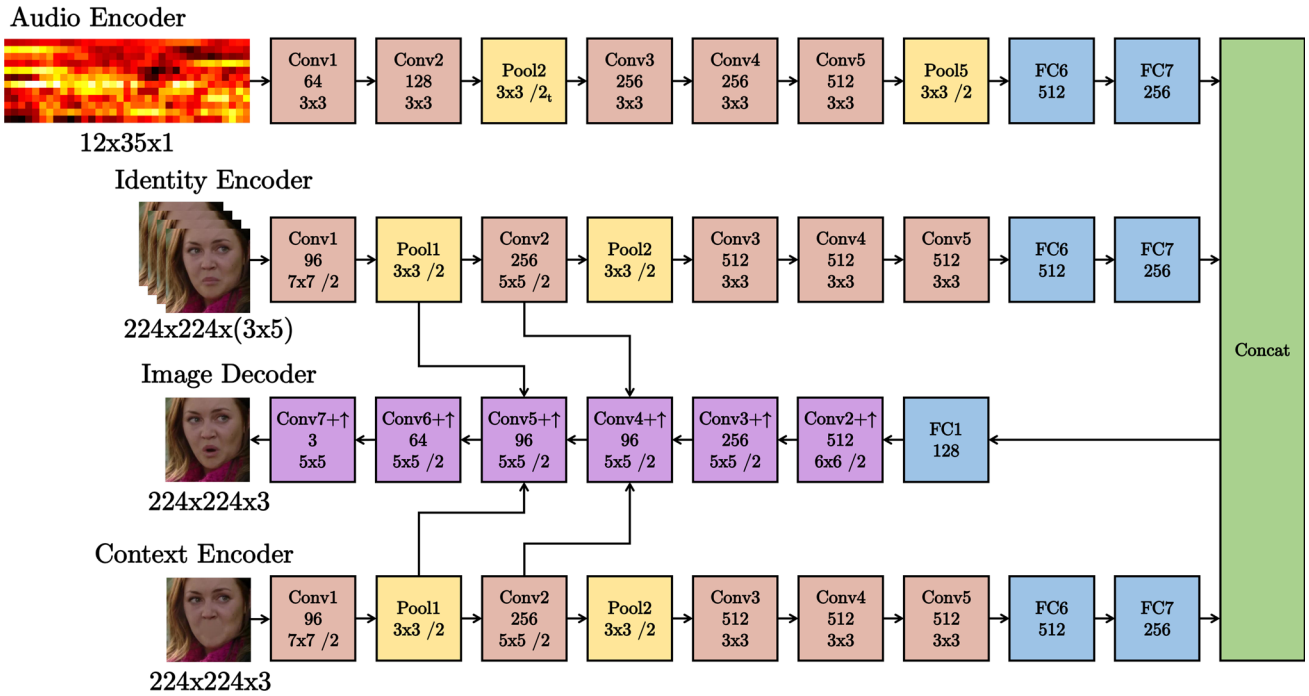

Fig. 14 The Speech2Vid model, with the context encoder. The top 3 rows are identical to Fig. 4, apart from the input dimension

mouth. The network diagram is shown in Fig. 14. The input and output dimensions are larger than that of the original Speech2Vid model at 224 × 224, since the images are of a larger area.

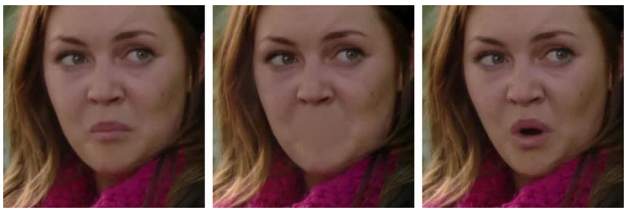

Fig. 15 Input images to the blending network. Left: 'Identity' image ; Middle: 'Context' image; Right: Ground truth image

Inputs There are three main inputs to the re-dubbing network: the audio, the 'identity' images and the 'context' image. The audio and 'identity' inputs are identical to the inputs of the architecture described in Fig. 4 while the 'context' image is the target frame (with the mouth region occluded using in-painting). As before the identity images encode facial features, but the 'context' image is added to aid the network by providing it with information on the desired face orientation, background and lighting. To produce the 'context' image, we use the detected facial landmarks and in-paint the mouth region with the median colour of the face. This occluded mouth region is similar to the area replaced in the baseline Poisson editing method.

During training, the 'context' image is the occluded version of the ground truth image, i.e. the image corresponding to the middle of the sampling window. At test time, the same 'context' image is fed into the network which is made from the current middle image, aligned via face orientation, of the sample. Figure 15 shows examples of the 'identity', 'context' and ground truth images. As before, the pose of the input face is restricted to near-frontal as in the LRS2 dataset and the face images are registered prior to generating the mouth, see Fig. 10.

Fig. 16 Blending results. Left: Using Poisson editing; Right: Using the blending network

Re-blending Since the output of the network corresponds exactly to the detected frame used to generate the 'context' image (i.e. the current image with occluded mouth), the re-dubbed generated frame can be trivially replaced with the output of the network. The only post-processing done is a series of simple transforms, translation and rotation, of the output to the reference frame which in this case is the 'context' image. For comparison, the baseline method re-transform only the mouth region while for the blending network, we re-transform the whole frame of the generated face back to the original image space.

Results and Discussion We find that the results from the network blending are consistently better than the Poisson editing-based method, particularly for off-frontal cases. See Fig. 16 for comparison of Poisson editing with the generated output.

The CNN-based method is very effective in naturally blending the generated mouth shape into the target image for both frontal and off-frontal faces. However, the common limitation of both methods is the difficulty in making the chins move with the mouth. This problem is particularly challenging since the network has to learn to fill the area of the background occluded in the target video frame.

7 Summary and Extensions

We have demonstrated that the Speech2Vid model is able to generate videos of any identity speaking from any source of input audio. This work shows that there is promise in generating video data straight from an audio source. We have also shown that re-dubbing videos from a different audio source (independent of the original speaker) is possible.

Moving forward, this model can be applied to computer facial animation relying only on audio.

Acknowledgements Funding for this research is provided by the EPSRC Programme Grant Seebibyte EP/M013774/1. Amir Jamaludin is funded by the RCUK CDT in Healthcare Innovation EP/G036861/1. We would like to thank Aravindh Mahendran for helpful discussions and the reviewers for their suggestions.

References

Afouras, T., Chung, J. S., Senior, A., Vinyals, O., & Zisserman, A. (2018). Deep audio-visual speech recognition. In *IEEE transactions on pattern analysis and machine intelligence*. arXiv preprint arXiv:1809.02108.

Arandjelović, R., & Zisserman, A. (2017). Look, listen and learn. In *Proceedings of the international conference on computer vision*.

Aytar, Y., Vondrick, C., & Torralba, A. (2016). SoundNet: Learning sound representations from unlabeled video. In *Advances in neural information processing systems*.

Cappelletta, L., & Harte, N. (2012). Phoneme-to-viseme mapping for visual speech recognition. In *ICPRAM*.

Charles, J., Magee, D., & Hogg, D. (2016). Virtual immortality: Reanimating characters from TV shows. In *Computer vision–ECCV 2016 workshops* (pp. 879–886). Springer.

Chatfield, K., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. In *Proceedings of the british machine vision conference*.

Chen, Q., & Koltun, V. (2017). Photographic image synthesis with cascaded refinement networks. In *Proceedings of the international conference on computer vision*.

Chung, J. S., & Zisserman, A. (2016). Out of time: automated lip sync in the wild. In *Workshop on multi-view lip-reading, ACCV*.

Chung, J. S., Jamaludin, A., & Zisserman, A. (2017). You said that? In *Proceedings of the british machine vision conference*.

Chung, J. S., Nagrani, A., & Zisserman, A. (2018). VoxCeleb2: Deep speaker recognition. In *INTERSPEECH*.

Chung, J. S., Senior, A., Vinyals, O., & Zisserman, A. (2017). Lip reading sentences in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Chung, S. W., Chung, J. S., & Kang, H. G. (2019). Perfect match: Improved cross-modal embeddings for audio-visual synchronisation. In *IEEE international conference on acoustics, speech and signal processing*. arXiv preprint arXiv:1809.08001.

Denton, E. L., & Birodkar, V. (2017). Unsupervised learning of disentangled representations from video. In *Advances in neural information processing systems*.

Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*

Ezzat, T., & Poggio, T. (2000). Visual speech synthesis by morphing visemes. *International Journal of Computer Vision*, *38*(1), 45–57.

Fan, B., Wang, L., Soong, F. K., & Xie, L. (2015). Photo-real talking head with deep bidirectional LSTM. In *IEEE international conference on acoustics, speech and signal processing*.

Fernando, B., Bilen, H., Gavves, E., & Gould, S. (2017). Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Garrido, P., Valgaerts, L., Sarmadi, H., Steiner, I., Varanasi, K., Pérez, P., et al. (2015). VDUB: Modifying face video of actors for plausible visual alignment to a dubbed audio track. In O. Deussen & H. Zhang (Eds.), *Computer graphics forum* (Vol. 34, pp. 193–204). London: Wiley.

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2414–2423).

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507.

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Isola, P., Zoran, D., Krishnan, D., & Adelson, E. H. (2016). Learning visual groups from co-occurrences in space and time. In *Workshop at international conference on learning representations*.

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3128–3137).

Karras, T., Aila, T., Laine, S., Herva, A., & Lehtinen, J. (2017). Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Transactions on Graphics*, *36*(4), 94:1–94:12. https://doi.org/10.1145/3072959.3073658.

Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1867–1874).

Kim, J., Lee, J. K., & Lee, K. M. (2016). Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

King, D. E. (2009). Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, *10*, 1755–1758.

Lienhart, R. (2001). Reliable transition detection in videos: A survey and practitioner's guide. *International Journal of Image and Graphics*, *1*, 469.

Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on artificial intelligence* (pp. 674–679). http://citeseer.nj.nec.com/lucas81optical.html.

Misra, I., Zitnick, C. L., & Hebert, M. (2016). Shuffle and learn: unsupervised learning using temporal order verification. In *Proceedings of the European conference on computer vision*.

Nagrani, A., Albanie, S., & Zisserman, A. (2018). Seeing voices and hearing faces: Cross-modal biometric matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Owens, A., Isola, P., McDermott, J. H., Torralba, A., Adelson, E. H., & Freeman, W. T. (2016). Visually indicated sounds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 2405–2413. IEEE Computer Society.

Parkhi, O. M. (2015). Features and methods for improving large scale face recognition. Ph.D. thesis, Department of Engineering Science Oxford University.

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. In *Proceedings of the British machine vision conference*

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2536–2544).

Pătrăucean, V., Handa, A., & Cipolla, R. (2016). Spatio-temporal video autoencoder with differentiable memory. In *Advances in neural information processing systems*.

Perez, P., Gangnet, M., & Blake, A. (2003). Poisson image editing. *ACM Transactions on Graphics*, *22*(3), 313–318.

Reed, S. E., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016). Generative adversarial text to image synthesis. In M. E. Balcan & K. Q. Weinberger (Eds.), *ICML. JMLR Workshop and Conference Proceedings* (Vol. 48, pp. 1060–1069). Cambridge: JMLR.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241). Springer.

Suwajanakorn, S., Seitz, S. M., & Kemelmacher-Shlizerman, I. (2017). Synthesizing obama: Learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, *36*(4), 95.

Taylor, S., Kim, T., Yue, Y., Mahler, M., Krahe, J., Rodriguez, A. G., et al. (2017). A deep learning approach for generalized speech animation. *ACM Transactions on Graphics (TOG)*, *36*(4), 93.

van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al. (2016a). Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems* (pp. 4790–4798).

van den Oord, A., Kalchbrenner, N., & Kavukcuoglu, K. (2016b). Pixel recurrent neural networks. In M. E. Balcan & K. Q. Weinberger (Eds.), *Proceedings of the 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research* (Vol. 48, pp. 1747–1756). NewYork: PMLR.

Vedaldi, A., & Lenc, K. (2015). Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the ACM multimedia conference*.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156–3164).

Wang, X., & Gupta, A. (2015). Unsupervised learning of visual representations using videos. In *Proceedings of the international conference on computer vision*.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., et al. (2015). Show, attend and tell: Neural image caption generation with visual attention. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research* (Vol. 37, pp. 2048–2057). Lille: PMLR.

Xue, T., Wu, J., Bouman, K., & Freeman, B. (2016). Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in neural information processing systems*.

Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful image colorization. In *Proceedings of the European conference on computer vision* (pp. 649–666). Springer.