Cross-Language Speech Dependent Lip-Synchronization

Anonymous WACV submission

Paper ID 141

Abstract

Speech videos such as movie dialogues, public speech, online courses are a great source of infotainment. These videos are often limited by the linguistic constraint of audiences being from different demographies. Vernacular backgrounds that are non-native to the accent or the language of the content producer often makes it difficult for a listener to comprehend the full essence of the content. Such videos are often supplemented with foreign language subtitles which hamper viewing experience. Otherwise, simple audio dubbing in a different language makes the video appear unnatural due to unsynchronized lip motion.

In this paper, we try to address this issue by proposing two lip synchronization methods: 1) Lip-synchronization for change in accent of the same language audio dubbing, and 2) Cross-language lip-synchronization for speech videos dubbed in a different language. We describe an automated pipeline to synchronize the lip movements of the speaker conditioned upon the audio in both cases. Our quantitative evaluation shows high SSIM index between generated cross-language lip-synchronized videos and the original videos. With the help of a user-based study, we verify that our method is preferred over unsynchronized videos.

1. Introduction

Speech videos are an effective way of story-telling. Many socio-political changes are caused by public speeches, movies depict the cultural aspect of societies. Similarly, online instructional videos, especially Massive Open Online Courses (MOOCs), are prime examples of how education can help skill development beyond the boundaries of conventional classrooms. Moreover, the Internet has made it possible for a global inter-cultural exchange of ideas where any information is just a click away. Yet we find limited penetration for these speech videos when they cross international boundaries. For instance, the retention rates in MOOC courses can be as low as 10% [13]. One of the major reasons for this is a cultural gap between the linguistics of the audience and the content producer. Students from different parts of the world often find it difficult to understand the accent and language of the instructors, owing to their non-familiarity with them. This results in slow learning curves as well as dropouts from such online courses. Subtitles in different languages do not lend enough help since they divert the attention of the audience. A quick-fix solution to this would be to dub speech videos in the accent or language of the audience. However, dubbing without lip synchronization makes the video appear unnatural [17].

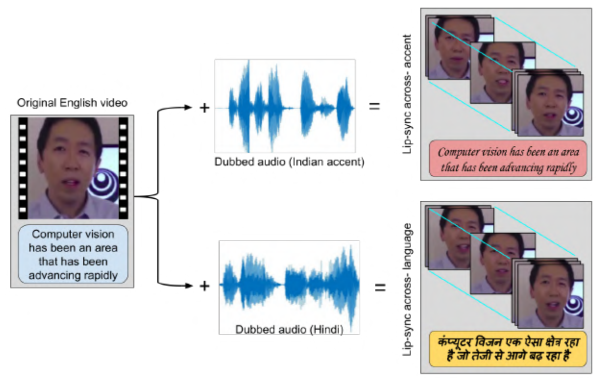

Figure 1. Lip synchronization on Andrew Ng Machine learning tutorial video based on dubbed audio: (top-right) shows Dynamic Programming to synchronize lip-motion of the original English video (left) into Indian English accent, (bottom-right) cross-language lip-sync to synchronize into different language (Hindi).

This problem can be solved by synchronizing the lip motion of the speaker in the target video to be coherent with the dubbed audio. A similar problem exists in the field of computer animation, where lip-motion of the animated characters are constrained upon the textual script of the character. This usually required a human in the loop to manually lay the visemes, hence such a system cannot be scaled for photo-realistic lip-synchronization (lip-sync).

With recent developments in deep neural architectures like Generative Adversarial Networks (GANs) [10, 14], we are now able to generate photo-realistic images conditioned upon an input prior. Similarly, Recurrent neural networks [12] have given way to better sequence learning methods which has become the de facto in time-series data modeling such as speech videos. Both of these methods are quintessentially data-driven approaches, and require large amount of training data.

In this paper, we propose 'Visual Dubbing' for synchronizing lip motion in speech videos according to the language it is dubbed in. Our main ideas and contributions are two-fold: 1) we propose a model for lip-syncing a target video with audio dubbing in a different accent of the same language, such as Indian English accent or French English accent, as shown in Figure 1 (top); 2) we propose another model to lip-sync the speech video based on audio dubbing in a different language, for instance English video with Hindi audio dubbing, as shown in Figure 1 (bottom).

The input to both our models is speech video where the lip-motion of a speaker is clearly visible, and the dubbed audio. The output is a video generated with synchronized lip motion. We also propose a scalable pipeline for dataset creation, which will later be used to train our models. Unlike audio-dubbing which requires professional dubbing artists to give their voices, visual-dubbing does not depend on human visual input for lip-synchronization of a target video. Figure 1 shows the visual dubbing for a video clip of an Andrew Ng MOOC video.

We evaluate our generative model based on the structural similarity (SSIM) index of the lip-synchronized videos with respect to the original English videos. Lastly, with the help of a user-based study we show that lip-synchronization makes the speech video more engaging while preserving photorealism.

2. Related Work

The current process of dubbing involves *translation* to the target language to resemble the lip motion of the source language as much as possible, *recording* of the dubbed content in pace with the original performance, and *editing* of the dubbed soundtrack and lip motion to be temporally close. This process is performed by production companies, and is both time-consuming and expensive. Even after such effort, the result of dubbing is not visually pleasing to the viewer because of clear visual discrepancy between the lip motion and the audio track.[25]

This discrepancy between the speech information of the dubbed track and the facial motion of the video track is due to differences in correspondence of phoneme sequences and lip motions [25]. It causes a strong discomfort for viewers, and is also a huge distraction for those who are hearing-impaired, as they rely significantly on lip reading [23, 21]. (This difference is one of the reasons why people dislike watching dubbed content [17], because it alters the sound perceived by the observer [22].

These findings are the motivation for our work to solve the problem of lip-synchronization.

2.1. Lip synchronization

The essential component of speech perception in videos and animations is visual cues, such as visemes [26] and phonemes. In particular, this is of importance for hearing-impaired people because they rely on these visual cues to understand videos. [21]. So, for dubbing of videos to another language or even another accent of the same language, it is necessary to learn the viseme-phonemic relation between both the accents or languages, so as to synchronize the lip motion with the original performance in the video.

The earliest work related to ours could be animation of facial movements in avatars modeled either from audio [15] or text [29, 30]. These work mostly used HMM [9] for sequential lip trajectory generation [31, 30]. One of the first systems for animating a virtual avatar's face directly from speech was proposed by [2], which models the joint distribution of acoustic and visual speech.

The work by Bregler et.al.[3] learns a mapping between the visemes and phonemes for one specific actor and language, and synthesizes new lip movements through image warping. However, the results of this method fail to dub between different languages and different individuals. Some of the recent work focus on synthesizing photo-realistic lip-motions and facial expressions. Face2Face [28] morphs the facial landmarks of a person in a target video based on those of another actor. However, these kind of models require a human in the loop, which can be quite expensive to scale, and prone to error.

The advent of Recurrent Neural Networks, especially LSTMs [12], gave way to better sequence learning, which made generation of features from speech more efficient. Pham *et al* [24] used CNN followed by an LSTM to generate face parameters from input audio waveform. Karras *et al* [16] proposed a network consisting of a spatial convolution layer followed by a temporal convolution network on top of fully connected layers to convert speech audio into facial expressions. Chung *et al* [6] proposed an encoder-decoder convolutional network to jointly embed face and audio. The encoder network consisted of an audio stream and a video stream which merges in a bottleneck representation. This representation is used by the decoder to synthesize lips-motion video frames.

Most similar to our work are [27, 20] which use speech audio represented as MFCC features [27] and text [20] to train an LSTM to produce a sequence of lip landmark points. The lip landmarks are then used to generate mouth texture. Finally this mouth texture is merged with the face in the original frame. Our work is different from [27, 20] in that our method synchronizes lip motion across two different languages, in contrast to just English-to-English. Hence, our challenges include learning higher-level viseme-phonemic relations across languages.

3. Method

Instructional videos provide a controlled framework for this problem, since the speakers usually speak scripted dialogues in good lighting facing the camera. The challenge is to model the lips, and generate new lip movements for the same speaker, given the dubbed audio. We consider dubbing to be of two types: a) in the same language as the original speech video but with a different accent, to alleviate the nuances related to unfamiliar accent; and b) where the video is re-dubbed into an entirely different language, to cater to non-native audiences. Both the dubbing types pose different challenges in lip synchronization. For English to non-native English, words remain the same while the pitch, tone and timing of the spoken word change across accents. On the other hand, for cross-language dubbing both words and phonemes change. In this section we will discuss two different methods to address these challenges.

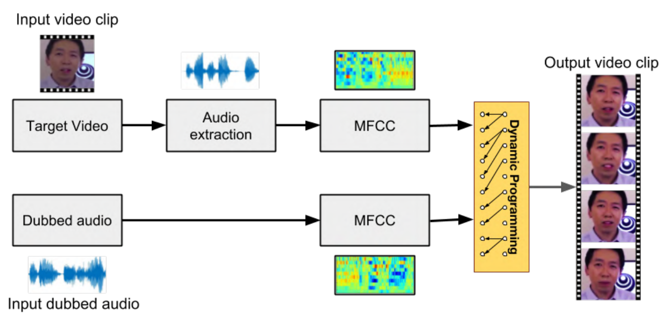

Figure 2. Pipeline for Dynamic Programming: Inputs are the target video and the dubbed audio in Indian accent, which are dynamically synced based on the similarity between the MFCC features of the voice of the native English speaker and the dubbed audio.

3.1. Cross-accent lip-sync

While dubbing for different accent, since the same words are spoken by the instructor, all the required viseme sequences are already present in the original video. However, the time instances of the words being spoken might change. Hence, this problem of cross-accent lip-sync can be reduced to a non-linear alignment between original audio and the dubbed audio. This can be done by creating a mapping between different segments of the two audio clips.

In our setup, we have Andrew Ng's Machine Learning audio-visual clip, along with a dubbed audio clip of the same dialogues in Indian-accented English. Spoken words can be broken down into sequences of phonemes. Therefore, we densely segment these audio clips into 25 millisecond (ms) clips. We then use Dynamic Programming [8] to create a dynamic map between the segments of the two audio clips in the feature space. Mel frequency Cepstral Coefficients (MFCC) have been widely used by the speech community, as they provide an optimal encoding of those audio bands which are most relevant to spoken speech. Therefore our mapping is based on finding the nearest neighbor of a segment from one audio clip in another. This is then converted into a mapping between the corresponding video frames. This procedure is illustrated in Figure 2, where we show the overall pipeline for non-linear alignment between original English audio and non-native English accent dubbed audio using dynamic programming.

Each segment of the original video clip is assigned a new time-stamp based on the mapping. This creates a non-uniform separation between adjacent segments of the original video clip. We render a new lip-synced video by interpolating frames in the original clip to fill the voids, and down-sampling to remove excess frames. Here we propose to use a Dynamic Programming algorithm (Algorithm 1), where x and y are the two inputs (in this case, the MFCCs of original and dubbed audio), L_x and L_y are the lengths of the respective audio segments. C represents the cumulative cost of the dynamic mapping with time, and M is used to find the optimal path for the mapping between input and output frames. $xTOy$ is the variable that records the mapping from x to y. Similarly, to find $yTOx$, the following can be appended at line 18: $yTOx[q-1] \leftarrow p-1$.

Algorithm 1 Dynamic Programming

1: $init \leftarrow 5$
2: $C[i][j] \leftarrow 0, i = 0toL_x, j = 0toL_y$
3: **for** $i = 1toL_x$ **do** $C[i][0] \leftarrow i * init$
4: **for** $j = 1toL_y$ **do** $C[0][j] \leftarrow j * init$
5: $M[i][j] \leftarrow 0, i = 0to(L_x-1), j = 0to(L_y-1)$
6: **for** $i = 1toL_x$ **do**
7:     **for** $j = 1toL_y$ **do**
8:         $min1 \leftarrow C[i-1][j-1] +$ **cost**$(input_x[i-1], input_y[j-1])$
9:         $min2 \leftarrow C[i-1][j] + init$
10:         $min3 \leftarrow C[i][j-1] + init$
11:         $C[i][j] \leftarrow$ **min**$(min1, min2, min3)$
12:         **if** $C[i][j] = min1$ **then** $M[i-1][j-1] \leftarrow 1$
13:         **if** $C[i][j] = min2$ **then** $M[i-1][j-1] \leftarrow 2$
14:         **if** $C[i][j] = min3$ **then** $M[i-1][j-1] \leftarrow 3$
15: $p \leftarrow L_x, q \leftarrow L_y$
16: **while** $p \neq 0$ and $q \neq 0$ **do**
17:     **if** $M[p-1][q-1] = 1$ **then**
18:         $xTOy[p-1] \leftarrow q-1, p \leftarrow p-1, q \leftarrow q-1$
19:     **else if** $M[p-1][q-1] = 2$ **then** $p \leftarrow p-1$
20:     **else if** $M[p-1][q-1] = 3$ **then** $q \leftarrow q-1$

3.2. Cross-language lip-sync

Major challenges in lip-syncing audio of a foreign language (e.g. Hindi) on video of original language (e.g. English) are the differences in their grammatical structure and set of phonemes. One way is to directly generate lip-images

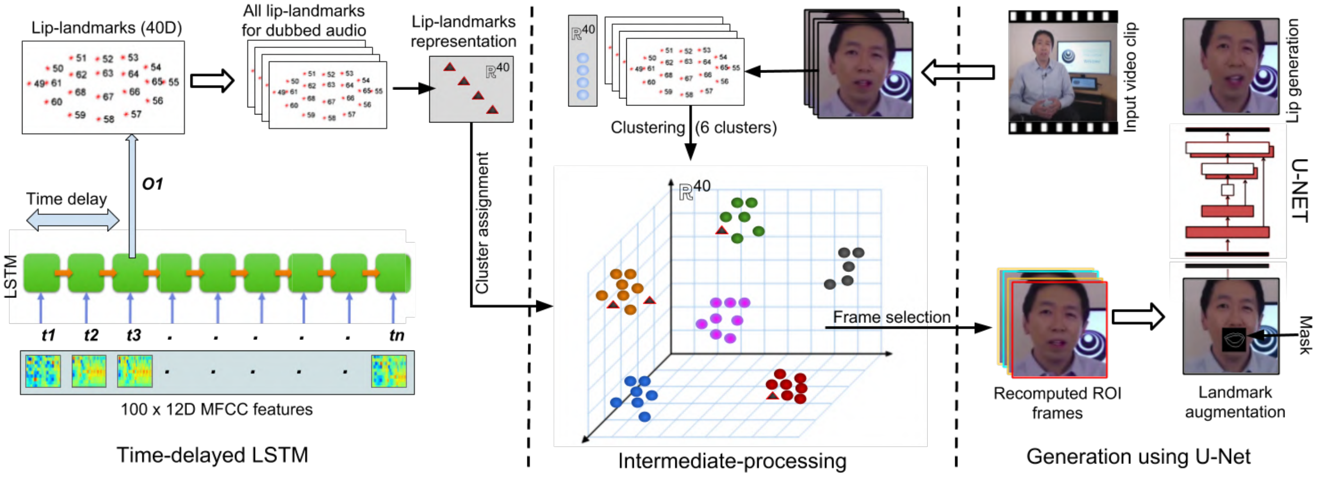

Figure 3. Cross-language lip-sync: (left) Pipeline for training LSTM with Hindi speech and lip landmarks, (center) shows reassignment of frames for each predicted lip-landmark using intermediate-processing step, (right) pipeline for inferring using U-Net on frames from English video.

conditioned upon the foreign language audio and target video. But such end-to-end systems require large amount of training data to learn the complex audio-visual relation between the two modalities [6]. At the same time, recent developments in generative networks [10, 14] have yielded impressive results. Considering these factors, we first learn an embedding between Hindi audio and lip-landmarks. This allows us to predict lip-landmarks from a relatively smaller speech corpus. From these predicted lip-landmarks, we generate mouth images over the original English video to match the Hindi audio. This entire two-step pipeline can be seen in Figure 3 (which also includes an intermediate processing step, discussed in Section 4.3).

3.2.1 Audio to Lip Landmarks

The first step is to encode audio into lip-landmarks. For each phoneme there exists a viseme, and the lip-motion responsible for the transition between two different visemes depends on the its location in the viseme-sequence that constitutes the spoken word. This makes audio to lip-landmarks a sequence modeling problem. Hence, similar to [27, 20], we use an LSTM [12] to encode audio. This LSTM takes audio MFCC features at each time step as input, and predicts lip-landmarks at time step ’t’ after each input, and is therefore called Time-Delayed LSTM (TD-LSTM) [11]. During training, the TD-LSTM is trained on Hindi speech audio-visual. The input is MFCC for every 25ms audio segment at 10ms time step, and the output is the lip-landmark of of the Hindi speaker at the 200ms delayed frame, as shown in Figure 3 (left). During inference, given a new audio sample, we use the predicted lip-landmarks as the prior to generate the mouth (lip-region) in the second step.

Figure 4. Generation results for the U-Net: (top) input to U-Net, (middle) generated images, (bottom) ground truth

3.2.2 Lip Landmarks to Generated Faces

Once lip landmarks are predicted from the audio in foreign language, in the second step the lips of the speakers in the original video must be modified to match these landmarks. To solve this problem, we use a U-Net similar to [20] to generate mouth of the speaker based on an encoded prior. During training, the input to the network is the face image of the speaker, with the mouth masked by a black box of constant size, and the original landmarks in the face drawn as a white polygon, see Figure 3 (right). The output of the network is the original face image. This allows the network to learn to generate actual face of the speaker with the lip-region conditioned upon the lip-polygon on the masked face.

As L1 loss is commonly used while generating images, we use this to train the U-Net. In addition, since our main focus is on correctly generating mouth region of the speaker, we add another loss term to penalize wrongly predicted pixels in that region. Considering the mean of the black mask as the center of the mouth region, we add a Gaussian weight kernel G_{loss} to the L1 loss such that the weight of this loss decreases radially from the center of the mouth to the face extremities. Formally, for a ground truth

$\hat{y}$ and the predicted face frame y, where any pixel location is represented by (i, j), our loss is defined as:

$$L = L1 * (1 + G_{loss}) \tag{1}$$

where;

$$L1 = \sum_{i,j} \|\hat{y_{ij}} - y_{ij}\| \tag{2}$$

$$G_{loss} = \sum_{i,j} c * \exp \frac{(i - u_i) * (j - u_j)}{v_{ij}} \tag{3}$$

In equation 3, c is a normalization constant, u_i and u_j represent the mean pixel location of the black mask (mouth region), and v_{ij} represents the covariance.

During inference, the mouth in every frame of the video is replaced by a constant black square, and a white polygon of the lip landmarks predicted by the LSTM network from the previous step. Thus, the U-Net will then generate faces according to the Hindi dubbing audio. Unlike [27, 20], we train our network on multiple sources, allowing the network to generalize over multiple speakers, as shown in Figure 4.

3.3. Dataset

In this section, we discuss our dataset curation pipeline used in our cross-language lip-synchronization method. We require two different datasets, one for each language: (i) Hindi speech dataset, for training time-delayed LSTM, and (ii) English videos dataset, to train U-Net for lip-generation.

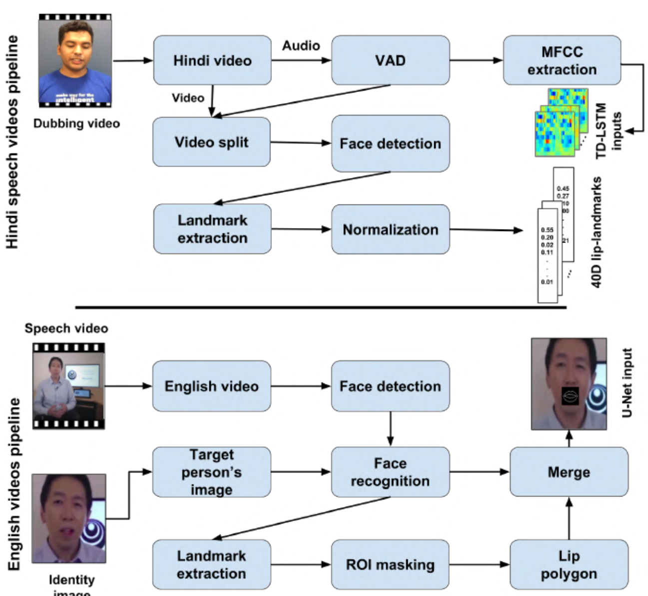

Figure 5. Dataset curation: (top) shows pipeline to create dataset from Hindi speech audio to give MFCC features and lip landmarks for training Time delayed LSTM (TD-LSTM); (bottom) shows pipeline to create dataset from Andrew Ng (and other) videos in English to give masked frames, for training U-Net.

3.3.1 Hindi speech dataset

As we wish to learn an encoding from Hindi audio to lip landmarks, we require a dataset consisting of Hindi audio to train a time-delayed LSTM. Since parallel audio speech corpus is difficult to find, and because dubbing is mostly a post-production phenomenon, we record 5 hours of audio-visual data of a native Hindi speaker narrating articles from Hindi newspapers and stories. Using voice activity detection [1], the video clips are segmented to give continuous segments of speech clips. For 5 hours of speech data, we get 5000 video clips of average length 2 seconds. Each such video clip is then sampled at 25 frames per second (fps). To obtain landmarks, we use a HOG-based face detector using dlib [18] to find the speaker's face in the clip, and predict 68 face landmarks using dlib. We then choose the landmarks corresponding to mouth region (landmarks 49 to 68) a.k.a. lip-landmarks, and normalize them. For each video clip segmented by voice activity detection, these normalized lip-landmarks are saved. Similarly, for each video clip we extract audio and sample it at 100Hz. We extract MFCCs for each sampled segment of the extracted audio clip. The training set consisted of 90% of total dataset, validation was done on rest of 10% of the dataset. The dataset curation pipeline for Hindi Speech can be seen in Figure 5 (top).

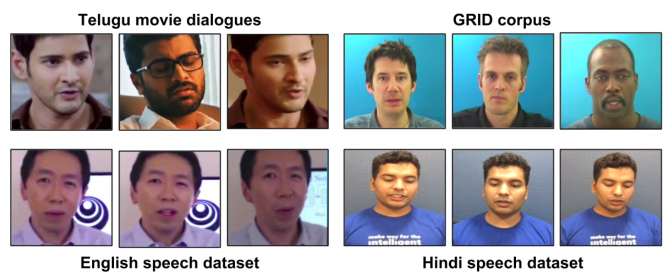

Figure 6. Random frames from Telugu movie dialogue clips (top-left), GRID corpus (top-right), English speech dataset(bottom-left), Hindi speech dataset (bottom-right).

3.3.2 English speech dataset

As our aim is to generate lip-synced Andrew Ng's machine learning videos with Hindi dubbing, we use 20 Andrew Ng videos to create a dataset of English speech videos. The input to our U-Net is frames from instructional video clips from the English speech dataset, where the pixels in the mouth region of the instructor are masked with the wire-frame structure of the lip-landmarks. For each frame where the face of the speaker has been detected, the face region is noted as the bounding box of the 68 landmarks, similar to that in Hindi speech dataset. The square region around the face with 1.5 times the face width is extracted. This results in images with full visibility of the instructor's face. The mouth region of each face is considered as the bounding box around the mean of the mouth landmarks (49 to

68), and of width 0.25 times the width of the face region. It is then replaced with a black mask and a white polygon connecting the lip landmarks, and resized to the input shape of the U-Net. The output of the U-Net is the original face image. The training set consisted of 10 video clips while the validation was done on remaining 10 video clips. This pipeline is shown in Figure 5 (bottom).

It was important to not let the network overfit on the input images. We therefore used multiple sources of images as the training dataset for the U-Net — frames extracted from 1) Telugu movies, 2) videos of Andrew Ng's deep learning.ai lectures, 3) GRID corpus [7], 4) Hindi Speech dataset. Table 1 details the number of frames from each source. Figure 6 shows some randomly sampled frames from these datasets.

Source	Train	Validation
Telugu movies	37130 frames	4159 frames
English speech	24359 frames	16035 frames
GRID	13350 frames	1500 frames
Hindi speech	37790 frames	3714 frames

Table 1. Number of images in Train and Validation sets for training U-Net.

For each source of images, we detected faces and predicted landmarks using [4].

3.4. Representation

Audio: The input to dynamic programming algorithm and time-delayed LSTM is raw audio from the Hindi speech video clips represented as Mel-frequency cepstral coefficients (MFCC). The audio was sampled at 100Hz with each sample being of length 25ms. We extract 13 coefficient MFCC feature for each sampled segment, but only use 12 coefficient as the feature representation, discounting the first feature. For TD-LSTM, length of each input training sample is 100 (or total 1 second), with shape 100 x 12D.

Lip-landmarks: The output of the TD-LSTM are lip-landmarks of the speaker of Hindi speech video clips i.e. 40 dimensional (D) normalized lip-landmarks for each frame (20 lip-landmarks x 2D). The output is computed at a delay of 200ms, and is represented as 1 x 40D vector.

Images: The input to our U-Net is a masked face image of size 256 x 256 x 3D similarly output is a 256 x 256 x 3D image consisting original face of the speaker.

4. Implementation

4.1. TD-LSTM

Our proposed TD-LSTM model consists of a single layer LSTM with 60 neurons in the hidden layer, followed by a 40D dense layer. We also up-sample each video clip at 100Hz to compute lip-landmarks. In each forward pass, the network takes 100 time-steps MFCC features (100 x 12D) and predicts the 20th up-sampled lip-landmark frame (1x40D). This is done densely for each Hindi audio-visual clip. This results in an offset in the prediction of lip-landmarks of 200ms at the beginning, and 800ms at the end of the video. We compensate for this by replicating the first and the last frame's predicted landmarks respectively for the appropriate number of frames. We also implemented TD-LSTM with 500ms and 800ms time-delays. But we found very little perceptual difference between the results, and therefore chose 200ms delay. Using Bidirectional TD-LSTM also did not perceptually affect the results.

We implemented the network in Keras deep learning framework [5], with a batch size of 64, mean square loss, and Adam [19] as the choice of optimizer. We trained our network for 20 epochs, with a total time of around 4 hours on Nvidia GTX 1080 Ti, when the loss started plateauing. Pre-training with 10% videos randomly sampled from GRID corpus resulted in faster saturation of loss. As the TD-LSTM learns to encode dubbed audio into lip-landmarks, we observed that it captures the artifacts corresponding to the dubbing artist, such as thickness of lips and span of mouth. Hence we found pre-training with GRID corpus also helps in generalization, in case of different dubbing artists.

4.2. U-Net

We use U-Net architecture similar to [20] We trained the U-Net on 4 NVIDIA TitanX GPUs, using a batch size of 16, counting 4 batches per iteration, until $\approx$ 5000 iterations. This took $\approx$ 2 seconds per iteration, and occupied $\approx$ 3.3GB of memory including model weights and images kept in the buffer. As U-Net has been trained on Andrew Ng's lip-landmarks to predict original frame, it also learns an undesirable mapping between jaw location and the shape of lip in the output. This badly affects the generation of lip in the instances where the predicted landmarks correspond to a closed lip while the target frame has mouth open, or vice-versa. To overcome this, we introduce an *intermediate-processing* step between TD-LSTM and U-Net.

4.3. Intermediate processing

We normalize the lip-landmarks from all the frames in the target instructional video in English, and group them into 6 clusters. The frames in the target video nearest to the cluster centroids are chosen to represent the clusters, with their landmarks as their new centroids. All the lip-landmarks predicted by TD-LSTM are then assigned to a cluster based in their distances from the new centroids. This allows the predicted lip-landmarks to be assigned appropriate face frames. Only these 6 frames are then fed to the U-Net (after masking the mouth region). This results in a set of generated frames consisting of lip-synced mouth re-

Figure 7. Qualitative results for cross-accent and cross-language lip-sync: (a) In both the examples, frames are sampled at 3 fps from the original instructional video in English (top) and cross-accent lip-synced video (bottom). (b) Each of the 6 images depicts original English video (left) along with its enlarged ROI, (right) shows our generated Hindi lip-synced video.

gions but in only 6 distinct facial poses corresponding to the new centroids. This results in a jittery face video with proper lip-synchronization.

4.4. Homograph computation

The frames generated from the U-Net are slightly blurred, therefore we use a pre-trained CNN deblur network as in [6], trained on facial images for sharpening. We then compute the pairwise homography between each generated video frame and that of the original instructional video clip using all the 3D face-landmarks predicted using [4], except those corresponding to eyes and lip region. This gives a transformation matrix between the frame pairs. We then crop a rectangular mouth region in the generated video frames, with its center at the mean of lip-landmarks, and its length twice of that between the mean of lip-landmarks and the landmark corresponding to the tip of the nose. This region of interest (ROI) is augmented over the original video using the the computed transformation matrix. All these ROI augmented frames along with Hindi dubbing are then used to create the final Hindi lip-synced video.

4.5. Evaluation metric

Audio-visual data generation is done for human consumption, hence its evaluation is subjective. Hence, we conducted a user-based study to evaluate the outputs of our cross-accent and cross-language methods. This allowed us to get an average subjective evaluation.

Structural similarity (SSIM) index [32] is a widely used metric to evaluate the quality of generated images and videos. We compare the SSIM index to evaluate the generation of the U-Net.

5. Results

5.1. User-based study

To check the quality of our proposed models, we asked 20 different people to rate the lip-unsynced video and the generated lip-synced video pairs, for each of 10 Andrew Ng's ML video clips, for each of the two cases of cross-accent and cross-language. The 10 video clip-pairs were of upto 1 minute duration each.

5.1.1 Cross-accent lip-sync

For each of the pairs of un-synced dubbed video where the Indian English-accented dubbed audio is naively overlaid on the original English video, and our dynamically lip-synced videos, we randomly selected 5 video pairs for each subject and asked them to rank the videos between 1 (Hard to understand) to 5 (Easy to understand) based on their comfort. As shown in Table 2, users preferred our dynamically programmed lip-synchronized videos.

	US(N)	S(N)	US(F)	S(F)
Dynamic	3.0	**4.6**	1.9	**3.2**

Table 2. Mean scores for Dynamic Programming (Dynamic) on Indian-English: for un-synced speech overlay ('US'), and lip-synced version 'S', by naive (N) and familiar (F) users.

5.1.2 Cross-language lip-sync

We perform a similar user-based experiment with cross-language lip-synchronized videos — we showed 10 videos, with Hindi audio naively overlaid (un-synced), and Hindi lip-synced video to 20 users. Since comfort is subjective and ill-defined, we asked users to rate the percentage of lip-synchronization perceived by them for each pair. As shown in Table 3, the means of the comfort score and percentage lip-synchronization were higher for our cross-language lip-synced videos. The average comfort rating across users for each video pair can be seen in Figure 8 (a), where as average percentage lip-synchronization can be seen in Figure 8 (b). We also show qualitative results of cross-accent and cross-language lip-synchronization in Figure 7 (a) and (b) respectively.

	C-US	C-S	LS%-US	LS%-S
Mean	2.51	**3.1**	23.86	**45.95**
Std-dev	1.07	0.6	25.9	24.1

Table 3. Mean scores and standard deviation for Cross-language lip-sync on Hindi: (C) comfort level for (US) un-synced speech overlay, and (S) lip-synced version; (LS%) Lip-Sync percentage for (US) un-synced and (S) lip-synced versions.

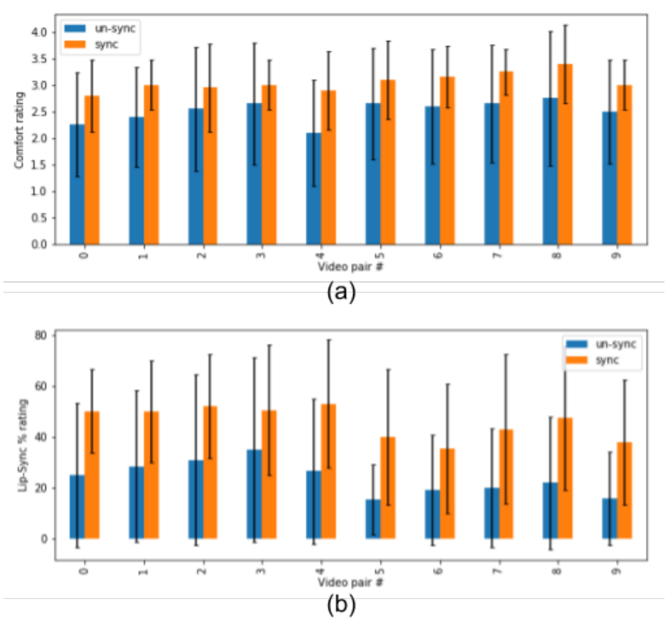

Figure 8. User feedback for cross-language lip-sync corresponding to 10 video pairs - (a) shows average comfort rating and its standard deviation for lip-unsynced (blue) and lip-synced videos (orange), (b) shows average percentage of perceived lip-synchronization and its standard deviation for lip-unsynced (blue) and lip-synced videos (orange).

5.2. Quality of generation

We compare SSIM index to evaluate the frame quality in original English video and its Hindi lip-synced version for each of the 10 video pairs. The average SSIM index across all the generated frames w.r.t the frames in the original videos was 0.98, with an overall standard deviation of 0.01. To evaluate the generation output of our cross-language model, we also computed SSIM scores for the 4 datasets we used to train U-Net. The average SSIM score for each of these dataset can be seen in Figure 9, with mean average SSIM score for all the dataset to be 0.58 with the overall standard deviation of 0.05.

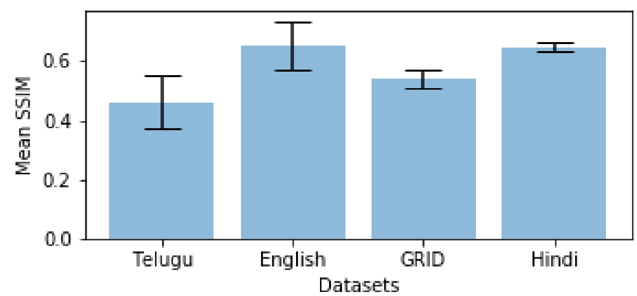

Figure 9. Mean and standard deviation of SSIM scores for various datasets used to train U-Net

6. Discussion

In this work, we assume the availability of dubbed audio, which can be automated using Machine translation (MT) systems and text-to-speech (TTS) synthesizers. However, imperfections in MT translations and lack of personality in the TTS-synthesized speech could make them unsuitable for instructional videos. Furthermore, handling multiple speakers, extreme head poses, and robust key point tracking present future scope of improvement. Lastly, we believe this work can help expand the reach of instructional videos across diverse linguistic groups.

7. Conclusion

We propose two different lip-synchronization methods for educational videos for same language with different accents, and two different languages to improve instructor-student engagement during online video lectures. We detail our pipelines for dataset creation for audio-to-lip-landmarks as well as lip-landmarks-to-mouth-generation. Our user-based-study shows that lip-synchronization can improve effectiveness of content delivery through dubbed speech videos.

References

[1] WEBRTC, VAD. https://webrtc.org/.

[2] M. Brand. Voice puppetry. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 21–28. ACM Press/Addison-Wesley Publishing Co., 1999.

[3] C. Bregler, M. Covell, and M. Slaney. Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 353–360. ACM Press/Addison-Wesley Publishing Co., 1997.

[4] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017.

[5] F. Chollet et al. Keras, 2015.

[6] J. S. Chung, A. Jamaludin, and A. Zisserman. You said that? In *British Machine Vision Conference*, 2017.

[7] M. Cooke, J. Barker, S. Cunningham, and X. Shao. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120(5):2421–2424, 2006.

[8] T. H. Cormen. *Introduction to algorithms*. 2009.

[9] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[11] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *NEURAL NETWORKS*, pages 5–6, 2005.

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. 9:1735–80, 12 1997.

[13] K. S. Hone and G. R. El Said. Exploring the factors affecting mooc retention: A survey study. *Computers & Education*, 98:157–168, 2016.

[14] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arxiv*, 2016.

[15] P. Kakumanu, R. Gutierrez-Osuna, A. Esposito, R. Bryll, A. Goshtasby, and O. Garcia. Speech driven facial animation. In *Proceedings of the 2001 workshop on Perceptive user interfaces*, pages 1–5. ACM, 2001.

[16] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Transactions on Graphics (TOG)*, 36(4):94, 2017.

[17] R. Kilborn. Speak my language': current attitudes to television subtitling and dubbing. *Media, culture & society*, 15(4):641–660, 1993.

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

[19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[20] R. Kumar, J. Sotelo, K. Kumar, A. de Brébisson, and Y. Bengio. Obamanet: Photo-realistic lip-sync from text.

[21] K. P. Lesner S.A. Visual vowel and diphthong perception across speakers. *Journal of the Academy of Rehabiitative Audiology 14*, pages 252–258, 1981.

[22] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264(5588):746, 1976.

[23] E. Owens and B. Blazek. Visemes observed by hearing-impaired and normal-hearing adult viewers. *Journal of Speech, Language, and Hearing Research*, 28(3):381–393, 1985.

[24] H. X. Pham, Y. Wang, and V. Pavlovic. End-to-end learning for 3d facial animation from raw waveforms of speech. *arXiv preprint arXiv:1710.00920*, 2017.

[25] W. H. Sumby and I. Pollack. Visual contribution to speech intelligibility in noise. *The journal of the acoustical society of america*, 26(2):212–215, 1954.

[26] Q. Summerfield. Lipreading and audio-visual speech perception. *Phil. Trans. R. Soc. Lond. B*, 335(1273):71–78, 1992.

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

[28] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, 2016.

[29] A. Wang, M. Emmi, and P. Faloutsos. Assembling an expressive facial animation system. In *Proceedings of the 2007 ACM SIGGRAPH symposium on Video games*, pages 21–26. ACM, 2007.

[30] L. Wang, W. Han, F. K. Soong, and Q. Huo. Text driven 3d photo-realistic talking head. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[31] L. Wang, X. Qian, W. Han, and F. K. Soong. Synthesizing photo-real talking head via trajectory-guided sample selection. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[32] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, volume 2, pages 1398–1402 Vol.2, Nov 2003.