Large-Scale Visual Speech Recognition

Brendan Shillingford[*], **Yannis Assael**[*], **Matthew W. Hoffman**, **Thomas Paine**, **Cían Hughes**, **Utsav Prabhu**, **Hank Liao**, **Hasim Sak**, **Kanishka Rao**, **Lorrayne Bennett**, **Marie Mulville**, **Ben Coppin**, **Ben Laurie**, **Andrew Senior**, **Nando de Freitas**
DeepMind & Google

Abstract

This work presents a scalable solution to open-vocabulary visual speech recognition. To achieve this, we constructed the largest existing visual speech recognition dataset, consisting of pairs of text and video clips of faces speaking (3,886 hours of video). In tandem, we designed and trained an integrated lipreading system, consisting of a video processing pipeline that maps raw video to stable videos of lips and sequences of phonemes, a scalable deep neural network that maps the lip videos to sequences of phoneme distributions, and a production-level speech decoder that outputs sequences of words. The proposed system achieves a word error rate (WER) of 40.9% as measured on a held-out set. In comparison, professional lipreaders achieve either 86.4% or 92.9% WER on the same dataset when having access to additional types of contextual information. Our approach significantly improves on other lipreading approaches, including variants of *LipNet* and of *Watch, Attend, and Spell* (WAS), which are only capable of 89.8% and 76.8% WER respectively.

1 Introduction and motivation

Deep learning techniques have allowed for significant advances in lipreading over the last few years [1–6]. However, these approaches have often been limited to narrow vocabularies, and relatively small datasets [1, 3, 6]. Often the approaches focus on single-word classification [7–20] and do not attack the open-vocabulary continuous recognition setting. In this paper, we contribute a novel method for large-vocabulary continuous visual speech recognition. In contrast to the largest previously reported lipreading vocabulary, 17,428 terms, we report substantial reductions in word error rate (WER) over the state-of-the-art approaches to lipreading even with a larger vocabulary of 127,055 terms.

Assisting people with speech impairments is the motivating factor behind this work. Visual speech recognition could positively impact the lives of hundreds of thousands of patients with speech impairments worldwide. For example, in the U.S. alone 103,925 tracheostomies were performed in 2014 [21], a procedure that can result in a difficulty to speak (disphonia) or an inability to produce voiced sound (aphonia). While this paper focuses on developing a scalable solution to lipreading using a vast diverse dataset, we also expand on this important medical application in Appendix A. The discussion there has been provided by medical experts and is aimed at medical practitioners.

We propose a novel lipreading system, illustrated in Figure 1, which transforms raw video into a word sequence. The first component of this system is a data processing pipeline used to create the largest existing visual speech recognition dataset, distilled from YouTube videos, consisting of phoneme sequences paired with video clips of faces speaking (3,886 hours of video). The creation of the dataset alone required a non-trivial combination of computer vision and machine learning techniques. At a high-level this process takes as input raw video and annotated audio segments, filters and preprocesses them, and produces a collection of aligned phoneme and lip frame sequences. The details of this process are described in Section 3.

[*] These authors contributed equally to this work.

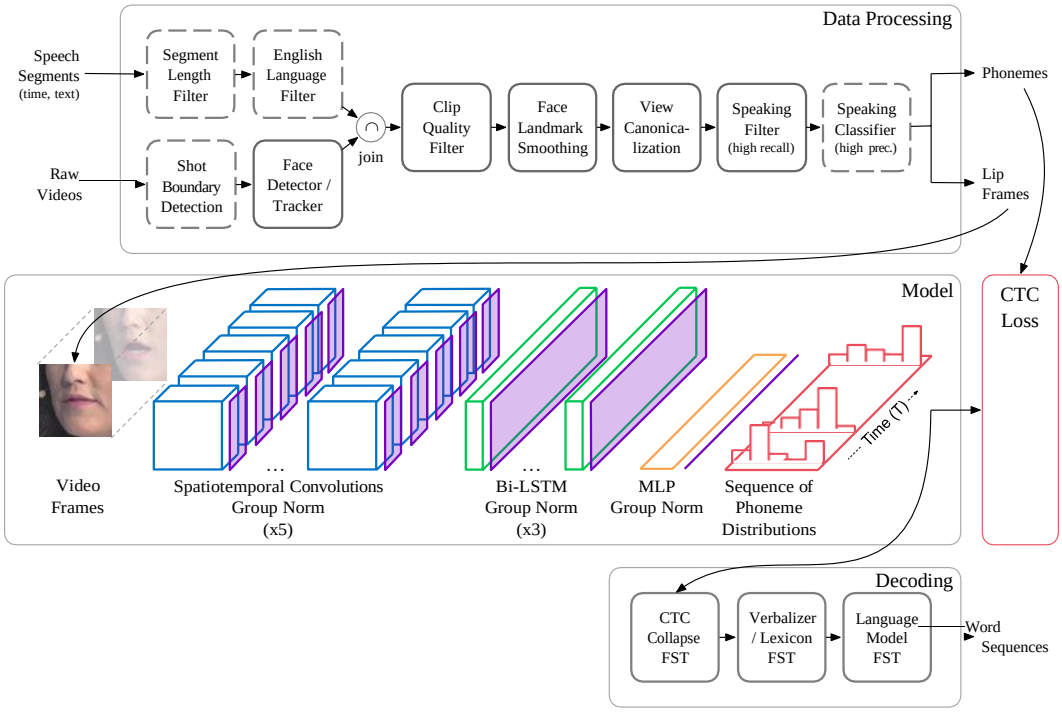

Figure 1: The full visual speech recognition system introduced by this work consists of a data processing pipeline that generates lip and phoneme clips from YouTube videos (see Section 3), and a scalable deep neural network for phoneme recognition combined with a production-grade word-level decoding module used for inference (see Section 4).

Next, this work introduces a new neural network architecture for lipreading, which we call *Vision to Phoneme* (V2P), trained to produce a sequence of phoneme distributions given a sequence of video frames. In light of the large scale of our dataset, the network design has been highly tuned to maximize predictive performance subject to the strong computational and memory limits of modern GPUs. Our approach is the first to combine a deep learning-based phoneme recognition model with production-grade word-level decoding techniques. By decoupling phoneme prediction and word decoding as is often done in speech recognition, we are able to arbitrarily extend the vocabulary without retraining the neural network. Details of our model and this decoding process are given in Section 4. By design, the trained model only performs well when videos are shot at specific angles when a subject is facing the camera, within a certain distance from a subject, and at high quality. It does not perform well in other contexts.

Finally, this entire lipreading system results in an unprecedented WER of 40.9% as measured on a held-out set from our dataset. In comparison, professional lipreaders achieve either 86.4% or 92.9% WER on the same dataset, depending on the amount of context given. Similarly, previous approaches such as variants of *LipNet* [1] and of *Watch, Attend, and Spell* (WAS) [2] demonstrated WERs of only 89.8% and 76.8% respectively.

2 Related work

While there is a large body of literature on automated lipreading, much of the early work focused on single-word classification and relied on substantial prior knowledge [22–29]. For example, Goldschen et al. [30] predicted continuous sequences of tri-visemes using a traditional HMM model with visual features extracted from a codebook of clustered mouth region images. The predicted visemes were used to distinguish sentences from a set of 150 possible sentences. Furthermore, Potamianos et al. [31] predict words and sequences digits using HMMs, Potamianos and Graf [32] introduce multi-stream HMMs, and Potamianos et al. [33] improve the performance by using visual features in addition to the lip contours. Later, Chu and Huang [22] used coupled HMMs to jointly model audio and

visual streams to predict sequences of digits. Neti et al. [34] used HMMs for sentence-level speech recognition in noisy environments, using the IBM ViaVoice dataset, by fusing handcrafted visual and audio features. More recent attempts using traditional speech, vision and machine learning pipelines include the works of Gergen et al. [35], Paleček [36], Hassanat [37] and Bear and Harvey [38]. For further details, we refer the reader to the survey material of Potamianos et al. [39] and Zhou et al. [40].

However, as noted by Zhou et al. [40] and Assael et al. [1], until recently generalization across speakers and extraction of motion features have been considered open problems. Advances in deep learning have made it possible to overcome these limitations, but most works still focus on single-word classification, either by learning visual-only representations [7–10, 20], multimodal audio-visual representations [11–15], or combining deep networks with traditional speech techniques (e.g. HMMs and GMM-HMMs) [16–19].

LipNet [1] was the first end-to-end model to tackle sentence-level lipreading by predicting character sequences. The model combined spatiotemporal convolutions with gated recurrent units (GRUs) and was trained using the CTC loss function. LipNet was evaluated on the GRID corpus [41], a limited grammar and vocabulary dataset consisting of 28 hours of 5-word sentences, where it achieved 4.8% and 11.4% WER in overlapping and unseen speaker evaluations respectively. By comparison, the performance of competent human lipreaders on GRID was 47.7%. LipNet is the closest model to our neural network. Several similar architectures were subsequently introduced in the works of Thanda and Venkatesan [3] who study audio-visual feature fusion, Koumparoulis et al. [4] who work on a small subset of 18 phonemes and 11 words to predict digit sequences, and Xu et al. [6] who presented a model cascading CTC with attention.

Chung et al. [2] were the first to use sequence-to-sequence models with attention to tackle audio-visual speech recognition with a real-world dataset. The model "Watch, Listen, Attend and Spell" (WLAS), consists of a visual (WAS) and an audio (LAS) module. To evaluate WLAS, the authors created LRS, the largest dataset at that point with approximately 246 hours of clips from BBC news broadcasts, and introduced an efficient video processing pipeline to generate the dataset. The authors reported 50.2% WER, with the performance of professional lipreaders being 87.6% WER. Chung and Zisserman [5] extended the work to multi-view sentence-level lipreading, achieving 62.8% WER for profile views and 56.4% WER for frontal views. Both Chung et al. [2] and Chung and Zisserman [5] pre-learn features with the audio-video synchronization classifier of Chung and Zisserman [42], and fix these features in order to compensate for the large memory requirements of their attention networks. Other related advances include works using vision for silent speech reconstruction [43–46] and for separating an audio signal into its individual speech sources [47, 48].

In contrast to the approach of Assael et al. [1], our model (V2P) uses a network to predict a sequence of phoneme distributions which are then fed into a decoder to produce a sequence of words. This flexible design enables us to easily accommodate very large vocabularies, and in fact we can extend the size of the vocabulary without having to retrain the deep network. Unlike previous work, V2P is memory and computationally efficient without requiring pre-trained features [2, 5].

3 Building a data pipeline for large-scale visual speech recognition

In this section we discuss the data processing pipeline, again illustrated in Figure 1, used to create the *Large-Scale Visual Speech Recognition* (LSVSR) dataset used in this work. The result of this pipeline is a significantly larger and more diverse dataset than in all previous efforts. While the first large-vocabulary lipreading dataset was IBM ViaVoice [34], more recent work has resulted in the much larger LRS and MV-LRS[2] datasets [2, 5], both generated from BBC news broadcasts. However, LSVSR is an order of magnitude greater than any previous dataset with 3,886 hours of audio-video-text pairs. In addition, the content is much more varied (i.e. not news-specific), resulting in a 7.3× larger vocabulary of 127,055 words. Figure 2 shows a comparison of sentence-level (word sequence) visual speech recognition datasets.

Our pipeline makes heavy use of large-scale parallel processing and is implemented as a number of independent modules and filters on top of FlumeJava [49]. This pipeline takes as input raw video and speech segments and outputs paired sequences of phonemes and lip frames which can be used to train a phoneme model described in Section 4. By eliminating the components marked by dashes in

[2]MV-LRS is the only publicly available large-vocabulary dataset, however it is limited to academic usage.

Dataset	Utter.	Hours	Vocab
GRID [41]	33,000	28	51
IBM ViaVoice [34]	17,111	35	10,400
MV-LRS [5]	74,564	~ 155	14,960
LRS [2]	118,116	~246	17,428
LSVSR (Ours)	**2,934,899**	**3,886**	**127,055**

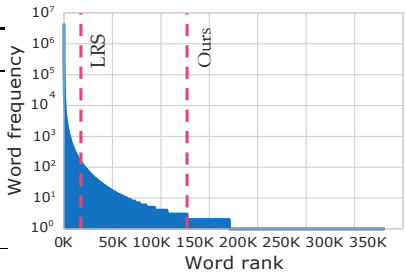

Figure 2: Left: A comparison of sentence-level (word sequence) visual speech recognition datasets. Right: Frequency of words in the LSVSR dataset in decreasing order of occurrence; approximately 350K words occur at least 3 times. We used this histogram to select a vocabulary of 127,055 words as it captures most of the mass. Note that thresholding at a vocabulary of 17,428 words, the largest existing previous dataset, excludes a large portion of high probability words.

Figure 1, i.e. those components whose primary use are in producing paired training data, this same pipeline can be used in combination with a trained model to predict word sequences from raw videos.

Our dataset is extracted from public YouTube videos. This is a common strategy for building datasets in ASR and speech enhancement [50–53, 47]. In our case, we use the work of Liao et al. [50] to extract audio clips paired with transcripts, yielding 140,000 hours of audio segments. Our processing pipeline is built on top of that, beginning with using the audio segments to fetch corresponding video segments.

Length filter, language filter. The duration of each segment extracted from YouTube is limited to between 1 and 12 seconds, and the transcripts are filtered through a language classifier [54] to remove non-English utterances. For evaluation, we further remove the utterances containing fewer than 6 words. Finally, the aligned phoneme sequences are obtained using a standard forced alignment approach with a lexicon with multiple pronunciations [50]. The phonetic alphabet is a reduced version of X-SAMPA [55] with 40 phonemes plus silence.

Raw videos, shot boundary detection, face detection. First, constant spatial padding in each video segment is eliminated. Then a standard, thresholding color histogram classifier [56] identifies segments containing shot boundaries and removes them. Finally, FaceNet [57] detects and tracks faces in every remaining segment.

Clip quality filter. Here, speech segments are joined with the set of tracked faces identified in their corresponding videos. We then filter based on the quality of the video: we remove blurry clips, clips with faces with an eye-to-eye width of less than 80 pixels (i.e. low resolution videos and clips where the face occupies little of the frame), and frame rates lower than 23fps. We allow a range of frame rates as input as varying frame rates has a similar effect as peoples' different speaking paces; however, frame rates above 30fps are downsampled.

Face landmark smoothing. The segments are then processed by a face landmark tracker and the resulting landmark positions are smoothed using a temporal Gaussian kernel. Empirically, our preliminary studies showed smoothing was crucial for achieving optimal performance. Next, following previous literature [2], we keep segments where the face yaw and pitch remain within 30° and 30°. Models trained outside this range perform worse [5].

View canonicalization. We obtain canonical faces using a reference canonical face model and by applying an affine transformation on the landmarks. Then, we use a thumbnail extractor which is configured to crop the area around the lips of the canonical face.

Speaking filter. Using the extracted and smoothed landmarks, minor lip movements and non-speaking faces are discarded using a threshold filter. This process involves computing the mouth openness in all frames, normalizing by the size of the face bounding box, and then thresholding on the standard deviation of the normalized openness. This classifier has very low computational cost, but a high recall, e.g. voice-overs are not handled.

Speaking classifier. As a final step, we build V2P-Sync, a neural network architecture to verify the audio and video channel alignment inspired by the work of Chung and Zisserman [42] and

Torfi et al. [58]. V2P-Sync uses longer time segments as inputs and spatiotemporal convolutions as compared to the spatial-only convolutions of Chung and Zisserman, and landmark smoothing and view canonicalization as compared to Torfi et al.. These characteristics facilitate the extraction of temporal features which is key to our task. Our model, V2P-Sync, takes as input a pair of a log mel-spectrogram and 9 grayscale video frames and produces an embedding for each using two separate neural network architectures. If the Euclidean distance of the audio and video embeddings is less than a given threshold then the pair is classified as synchronized. The architecture is trained using a contrastive loss similar to Chung and Zisserman. Since there is no labeled data for training, the initial unfiltered pairs are used as positive samples with negative samples generated by randomly shifting the video of an unfiltered pair. After convergence the dataset is filtered using the trained model, which is then fine-tuned on the resulting subset of the initial dataset. The final model is used to filter the dataset a second time, achieving an accuracy of 81.2%. This accuracy is improved as our audio-video pairs are processed by sliding V2P-Sync on 100 equally spaced segments and their scores are averaged. We refer the reader to the supplementary material for further architectural details.

Finally, by combining all of these components we obtain a dataset consisting of paired video and phoneme sequences, where video sequences are represented as identically-sized frames (here, 128×128) stacked in the time-dimension. Our pipeline processed clips pre-selected from YouTube using the work of Liao et al. [50], but only about 2% of clips satisfied the filtering criteria (face detection, frame rate, resolution, blur, face rotation) and had lip movements matched with text as determined by our speaking classifier.

4 An efficient spatiotemporal model of visual speech recognition

This work introduces the V2P model, which consists first of a *3d convolutional module* for extracting spatiotemporal features from a given video clip. These features are then aggregated over time with a *temporal module* which outputs a sequence of phoneme distributions. Given input video clips and target phoneme sequences as described in the previous section, the model is trained using the *CTC* loss function. Finally, at test-time, a *decoder* based on finite state transducers (FSTs) is used to produce a word sequence given a sequence of phoneme distributions. For further architectural details we refer the reader to Appendix E.

Neural network architecture. Although the use of optical-flow filters as inputs is commonplace in lipreading [59–64], in this work we designed a vision module based on VGG [65] to explicitly address motion feature extraction. We adapted VGG to make it volumetric, which proved crucial in our preliminary empirical evaluation and has been established in previous literature [1]. The intuition behind this is the importance of spatiotemporal relationships in human visual speech recognition, e.g. measuring how lip shape changes over time. Furthermore, the receptive field of the vision module is 11 video frames, roughly 0.36–0.44 seconds, or around twice the typical duration of a phoneme.

One of the main challenges in training a large vision module is finding an effective balance between performance and the imposed constraints of GPU memory. Our vision module consists of 5 convolutional layers with $[64, 128, 256, 512, 512]$ filters. By profiling a number of alternative architectures, we found that high memory usage typically came from the first two convolutional layers. To reduce the memory footprint we limit the number of convolutional filters in these layers, and since the frame is centered around the lips, we omit spatial padding. Since phoneme sequences can be quite long, but with relatively low frame rate (approximately 25–30 fps), we maintain padding in the temporal dimension and always convolve with unit stride in order to avoid limiting the number of output tokens. Despite tuning the model to reduce the number of activations, we are still only able to fit 2 batch elements on a GPU. Hence, we distribute training across 64 workers in order to achieve a batch size of 128. Due to communication costs, batch normalization is expensive if one wants to aggregate the statistics across all workers, and using only two examples per batch results in noisy normalization statistics. Thus, instead of batch normalization, we use group normalization [66], which divides the channels into groups and computes the statistics within these groups. This provides more stable learning regardless of batch size.

The outputs of the convolutional stack are then fed into a temporal module which performs longer-scale aggregation of the extracted features over time. In constructing this component we evaluated a number of recurrent neural network and dilated convolutional architectures, the latter of which are evaluated later as baselines. The best architecture presented performs temporal aggregation using a

stack of 3 bidirectional LSTMs [67] with a hidden state of 768, interleaved with group normalization. The output of these LSTM layers is then fed through a final MLP layer to produce a sequence of exactly T conditionally independent phoneme distributions $p(u_t | \mathbf{x})$. This entire model is then trained using the CTC loss we describe next.

This model architecture is similar to that of the closest related work, LipNet [1], but differs in a number of crucial ways. In comparison to our work, LipNet used GRU units and dropout, both of which we found to perform poorly in preliminary experiments. Our model is also much bigger: LipNet consists of only 3 convolutional layers of [32, 64, 96] filters and 3 GRU layers with hidden state of size 256. Although the small size of LipNet means that it does not require any distributed computation to reach effective batch sizes, we will see that this drop in size coincides with a similar drop in performance. Finally, while both models use a CTC loss for training, the architecture used in V2P is trained to predict phonemes rather than characters; as we argue shortly this provides V2P with a much simpler mechanism for representing word uncertainty.

Connectionist temporal classification (CTC). CTC is a loss function for the parameterization of distributions over sequences of label tokens, without requiring alignments of the input sequence to the label tokens [68]. To see how CTC works, let V denote the set of single-timestep label tokens. To align a label sequence with size-T sequences given by the temporal module, CTC allows the model to output blank symbols $\sqcup$ and repeat consecutive symbols. Let the function $\mathcal{B} : (V \cup \{\sqcup\})^* \to V^*$ be defined such that, given a string potentially containing blank tokens, it deletes adjacent duplicate characters and removes any blanks. The probability of observing label sequence y can then be obtained by marginalizing over all possible alignments of this label, $p(y | \mathbf{x}) = \sum_{u \in \mathcal{B}^{-1}(y)} p(u_1 | \mathbf{x}) \cdots p(u_T | \mathbf{x})$, where $\mathbf{x}$ is input video. For example, if $T = 5$ the probability of sequence 'bee' is given by $p(be\sqcup e\sqcup) + p(\sqcup be\sqcup e) + \ldots + p(bbe\,e\sqcup) + p(be\,ee)$. Note that there must be a blank between the 'e' characters in order for to avoid collapsing the sequence to 'be'.

Since CTC prevents us from using autoregressive connections to handle inter-timestep dependencies of the label sequence, the marginal distributions produced at each timestep of the temporal module are conditionally independent, as pointed out above. Therefore, to restore temporal dependency of the labels at test-time, CTC models are typically decoded with a beam search procedure that combines the probabilities with that of a language model.

Rationale for phonemes and CTC. In speech recognition, whether on audio or visual signals, there are two main sources of uncertainty: uncertainty in the sounds that are in the input, and uncertainty in the words that correspond to these sounds. This suggests modelling $p(\text{words} | \mathbf{x}) = \sum_{\text{phonemes}} p(\text{words} | \text{phonemes}) p(\text{phonemes} | \mathbf{x}) \approx p(\text{words} | \text{phonemes}) p(\text{phonemes} | \mathbf{x})$, where the approximation is by the assumption that a given word sequence often has a single or dominant pronunciation. While previous work uses CTC to model characters given audio or visual input directly [1, 69], we argue this is problematic as the conditional independence of CTC timesteps means that the temporal module must assign a high probability to a single sequence in order to not produce spurious modes in the CTC distribution.

To explain why modeling characters with CTC is problematic, consider two character sequences "fare" and "fair" with the same pronunciation (i.e. /fɛr/). The difficulty we will describe is independent of the model used, so we will consider a simple unconditional model where each character c is assigned probability given by the parameters $\pi_t^c = P(u_t = c)$ and the probability of a sequence is given by its product, e.g. $p(\text{fare}) = \pi_1^{\text{f}} \pi_2^{\text{a}} \pi_3^{\text{r}} \pi_4^{\text{e}}$. The maximum likelihood estimate, $\arg\max_{\pi} p(\text{fare}) p(\text{fair})$, however, assigns equal 1/4 probability to each of "fare", "fair", "faie", "farr", as shown in Figure 3, resulting in two undesirable words. Ultimately this difficulty arises due to the independence assumption of CTC and the many-to-many mapping of characters to words[3]. This same difficulty arises if we replace the parameters above with the outputs of a network mapping from videos to tokens. Using phonemes, which have a one-to-many map to words, allows the temporal model to only model sound uncertainty, and the word uncertainty can instead be handled by the decoder described below.

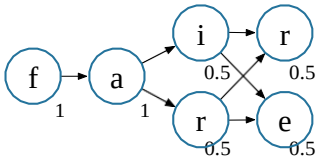

Figure 3: Example illustrating the issues with modelling characters with CTC.

[3] Languages such as Korean, where there is a one-to-one correspondence between pronunciation and orthography, do not give rise to such discrepancies.

Alternatively to using phonemes with CTC, some previous work solves this problem using RNN transducers [70] or sequence-to-sequence with attention [2], which jointly model all sources of uncertainty. However, Prabhavalkar et al. [71] showed in the context of acoustic speech recognition that these models were unable to significantly outperform a baseline CTC model (albeit using context-dependent phonemes and further sequence-discriminative training) when combined with a decoding pipeline similar to ours. Hence, for reasons of performance and easier model training, especially important with our large model, we choose to output phonemes rather than words or characters directly. Additionally, and crucial for many applications, CTC also provides extra flexibility over alternatives. The fact that the lexicon (phoneme to word mapping) and language model are separate and part of the decoder, affords one the ability to trivially change the vocabulary and language model (LM) arbitrarily. This allows for visual speech recognition in narrower domains or updating the vocabulary and LM with new words without requiring retraining of the phoneme recognition model. This is nontrivial in other models, where the language model is part of the RNN.

Decoding. As described earlier, our model produces a sequence of phoneme distributions; given these distributions we use an industry-standard decoding method using finite state transducers (FSTs) to arrive at word sequences. Such techniques are extensively used in speech recognition [e.g. 72, 73]; we refer the reader to the thorough presentation of Mohri et al. [74]. In our work we make use of a combination of three individual FSTs. The first *CTC postprocessing FST* removes duplicate symbols and CTC blanks. Next, a *lexicon FST* maps input phonemes to output words. Third, an *n-gram language model with backoff* can be represented as a weighted FST from words to words. In our case, we use a 5-gram language model with Katz smoothing with about 50 million n-grams and a vocabulary size of about one million. The composition of these three FSTs results in another weighted FST that transduces from phoneme sequences to (reweighted) word sequences. Finally, a search procedure is employed to transduce likely words from the phoneme recognition model.

5 Evaluation

We examine the performance of V2P trained on LSVSR with hyperparameters tuned on a validation set. We evaluate it on a held-out test set roughly 37 minutes long, containing approximately 63,000 video frames and 7100 words. We also describe and compare against a number of alternate methods from previous work. In particular, we show that our system gives significant performance improvements over professional lipreaders as well previous state-of-the-art methods for visual speech recognition.

In each case, the network architecture is optimized using Adam [75] with a learning rate of 10^{-4} and default hyperparameters: first and second momentum coefficients 0.9 and 0.999 respectively, and $s = 10^{-8}$ for numerical stability. Furthermore, to accelerate learning, a curriculum schedule limits the video duration, starting from 2 seconds and gradually increasing to a maximum length of 12 seconds over 200,000 training steps. Finally, image transformations are also applied to augment the image frames to help improve invariance to filming conditions. This is accomplished by first randomly mirroring the videos horizontally, followed by random changes to brightness, contrast, saturation, and hue. To construct the validation and test sets we also removed blurry videos by thresholding the variance of the Laplacian of each frame [76]; we kept them in the training set as a form of data augmentation.

Professional lipreaders. We consulted a professional lipreading company to measure the difficulty of LSVSR and hence the impact that such a model could have. Since the inherent ambiguity in lipreading necessitates relying on context, we conducted experiments both with and without context. In both cases we generate modified clips from our test set, but cropping the whole head in the video, as opposed to just the mouth region used by our model. The lipreaders could view the video up to 10 times, at half or normal speed each time. To measure without-context performance, we selected clips with transcripts that had at least 6 words. To measure how much context helps performance, we selected clips with at least 12 words, and presented to the lipreader the first 6 words, the title, and the category of the video, then asked them to transcribe the rest of the clip. The lipreaders transcribed a subset of our test set containing 153 and 274 videos with and without context, respectively.

DS2-Ph (audio-only). For comparison and as an approximate bound on performance, we also train an audio speech recognition model on the audio of the utterances, with the architecture based on Deep Speech 2 [69], but trained to predict phonemes rather than characters.

Method	Params	PER	CER	WER
Professional (w/o context)	–	–	–	92.9 ± 0.9
Professional (w/ context)	–	–	–	86.4 ± 1.4
DS2-Ph (audio-only)	58M	12.5 ± 0.5	11.5 ± 0.6	18.3 ± 0.9
Baseline-LipNet-Ph	7M	65.8 ± 0.4	72.8 ± 0.5	89.8 ± 0.5
Baseline-3D-Seq2seq (WAS)	15M	–	49.9 ± 0.6	76.8 ± 0.8
V2P-FullyConv	29M	41.3 ± 0.6	36.7 ± 0.9	51.6 ± 1.2
V2P-NoLM	49M	33.6 ± 0.6	34.6 ± 0.8	53.6 ± 1.0
V2P	**49M**	$\mathbf{33.6 \pm 0.6}$	$\mathbf{28.3 \pm 0.9}$	$\mathbf{40.9 \pm 1.2}$

Table 1: Performance evaluation on LSVSR test set. Columns show phoneme, character, and word error rates, respectively. Standard deviations are bootstrap estimates.

Baseline-LipNet-Ph. Using our training setup, we replicate the architecture of LipNet [1]. For better comparison with our model setup, we train LipNet to predict phonemes, still with CTC. We use the same FST-based decoding pipeline, including the same 5-gram LM, to decode words. Recall from the earlier discussion that LipNet uses dropout, whereas V2P which makes heavy use of group normalization, crucial for our small batches per worker. Preliminary experiments on batch normalization and GRUs were not promising. As explained in Section 4, predicting phonemes is easier to model than characters for CTC, and hence should perform better.

Baseline-3D-Seq2seq (WAS). Using our training setup, we compared to a variant of the previous state-of-the-art sequence-to-sequence architecture of WAS that predicts character sequences [2]. Although their implementation was followed as closely as possible, training end-to-end quickly exceeded the memory limitations of modern GPUs. To work around these problems, the authors kept the convolutional weights fixed using a pretrained network from audio-visual synchronization classification [42], which we were unable to use as their network inputs were processed differently. Instead, we replace the 2D convolutional network with the *improved* lightweight 3D visual processing network of V2P. From our empirical evaluation, including preliminary experiments not reported here and as shown by earlier work [1], we believe that the 3D spatiotemporal aggregation of features benefits performance. After standard beam search decoding, we use the same 5-gram word LM as used for the CTC models to perform reranking.

V2P-FullyConv. Identical to V2P, except the LSTMs in the temporal aggregation module are replaced with 6 dilated temporal convolution layers with a kernel size of 3 and dilation rates of [1,1,2,4,8,16], yielding a fully convolutional model with 12 layers.

V2P-NoLM. Identical to V2P, except during decoding, where the LM is replaced with a dictionary consisting of 100k words. The words are then weighted by their smoothed frequency in the training data, essentially a uni-gram language model.

5.1 Results

Table 1 shows the phoneme error rate, character error rate, and word error rate for all of the models, and the number of parameters for each. The error rates are computed as the sum of the edit distances of the predicted and ground-truth sequence pairs divided by total ground-truth length. We also compute and display the standard error associated with each rate, estimated by bootstrap sampling.

These results show that the variant of LipNet tested in this work is approximately able to perform on-par with professional lipreaders with WER of 86.4 and 89.8 respectively, even when the given professional is given additional context. Similarly, we see that the WAS variant provides a substantial reduction to this error, resulting in a WER of 76.8. However, the full V2P method presented in this work is able to further halve the WER, obtaining a value of 40.9 at testing time. Interestingly, we see that although the bi-directional LSTM provides the best performance, using a fully-convolutional network still results in performance that is significantly better than all previous methods. Finally, although we see that the full V2P model performs best, removing the language model results only in a drop of approximately 13 WER to 53.6.

Figure 4: This heatmap shows which insertion and deletion errors were most common on the test set. Blue indicates more insertions or deletions occurred.

By predicting phonemes directly, we also side-step the need to design phoneme-to-viseme mappings [38]. The inherent uncertainty is instead modelled directly in the predictive distribution. For instance, using edit distance alignments of the predictions to the ground-truths, we can determine which phonemes were most frequently erroneously included or missed, as shown in Figure 4. Here we normalize the rates of deletions vs insertions, however empirically we saw that deletions were much more common than inclusions. Among these errors the most common include phonemes that are often occluded by the teeth (/d/, /n/, and /t/) as well as the most common English vowel /@/. Finally, by differentiating the likelihood of the phoneme sequence with respect to the inputs using guided backpropagation [77], we compute the saliency maps shown in the top row of Figure 5 as a white overlay. The entropy at each timestep of the phoneme predictive distribution is shown as well. A full confusion matrix and additional saliency maps are shown in Appendices B and C.

Figure 5: Saliency map for “kind of” and the top-3 predictions of each frame. The CTC blank character is represented by ‘ ␣ ’. The unaligned ground truth phoneme sequence is /k aI n d V v/.

6 Conclusions

We presented a novel, large-scale visual speech recognition system. Our system consists of a data processing pipeline used to construct a vast dataset—an order of magnitude greater than all previous approaches both in terms of vocabulary and the sheer number of example sequences. We described a scalable model for producing phoneme and word sequences from processed video clips that is capable of nearly halving the error rate of the previous state-of-the-art methods on this dataset. The combination of methods in this work represents a significant improvement in lipreading performance, a technology which can enhance automatic speech recognition systems, and which has enormous potential to improve the lives of speech impaired patients worldwide.

Acknowledgments

We would like to thank Hagen Soltau for preparing the YouTube audio dataset, which our dataset is based on, and for providing the language model used for decoding. We also would like to thank Andrew Zisserman for his valuable contributions as an advisor, Shane Agnew for his assistance, and Misha Denil, Sean Legassick, Iason Gabriel, Dominic King, and Alan Karthikesalingam for their helpful comments on our paper.

References

[1] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas. LipNet: End-to-end sentence-level lipreading. In *GPU Technology Conference*, 2017.

[2] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. Lip reading sentences in the wild. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[3] A. Thanda and S. M. Venkatesan. Audio visual speech recognition using deep recurrent neural networks. In F. Schwenker and S. Scherer, editors, *Multimodal Pattern Recognition of Social Signals in Human-Computer-Interaction*, pages 98–109. Springer, 2017.

[4] A. Koumparoulis, G. Potamianos, Y. Mroueh, and S. J. Rennie. Exploring ROI size in deep learning based lipreading. In *International Conference on Auditory-Visual Speech Processing*, 2017.

[5] J. S. Chung and A. Zisserman. Lip reading in profile. In *British Machine Vision Conference*, 2017.

[6] K. Xu, D. Li, N. Cassimatis, and X. Wang. LCANet: End-to-end lipreading with cascaded attention-ctc. *arXiv preprint arXiv:1803.04988*, 2018.

[7] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[8] J. S. Chung and A. Zisserman. Lip reading in the wild. In *Asian Conference on Computer Vision*, 2016.

[9] M. Wand, J. Koutnik, and J. Schmidhuber. Lipreading with long short-term memory. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 6115–6119. IEEE, 2016.

[10] T. Stafylakis and G. Tzimiropoulos. Combining residual networks with LSTMs for lipreading. In *Interspeech*, pages 3652–3656. ISCA, 2017.

[11] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *International Conference on Machine Learning*, pages 689–696, 2011.

[12] C. Sui, M. Bennamoun, and R. Togneri. Listening with your eyes: Towards a practical visual speech recognition system using deep Boltzmann machines. In *ICCV*, pages 154–162. IEEE, 2015.

[13] H. Ninomiya, N. Kitaoka, S. Tamura, Y. Iribe, and K. Takeda. Integration of deep bottleneck features for audio-visual speech recognition. In *International Speech Communication Association*, 2015.

[14] S. Petridis and M. Pantic. Deep complementary bottleneck features for visual speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 2304–2308. IEEE, 2016.

[15] S. Petridis, Y. Wang, Z. Li, and M. Pantic. End-to-end multi-view lipreading. In *BMVC*, 2017.

[16] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata. Lipreading using convolutional neural network. In *Interspeech*, pages 1149–1153, 2014.

[17] O. Koller, H. Ney, and R. Bowden. Deep learning of mouth shapes for sign language. In *ICCV Workshop on Assistive Computer Vision and Robotics*, pages 85–91, 2015.

[18] I. Almajai, S. Cox, R. Harvey, and Y. Lan. Improved speaker independent lip reading using speaker adaptive training and deep neural networks. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 2722–2726. IEEE, 2016.

[19] Y. Takashima, R. Aihara, T. Takiguchi, Y. Ariki, N. Mitani, K. Omori, and K. Nakazono. Audio-visual speech recognition using bimodal-trained bottleneck features for a person with severe hearing loss. *Interspeech*, pages 277–281, 2016.

[20] M. Wand and J. Schmidhuber. Improving speaker-independent lipreading with domain-adversarial training. In *Interspeech*, 2017.

[21] HCUPnet. Hospital inpatient national statistics. https://hcupnet.ahrq.gov/, 2014. Accessed: 2018-04-23.

[22] S. M. Chu and T. S. Huang. Bimodal speech recognition using coupled hidden Markov models. In *Interspeech*, 2000.

[23] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):198–213, 2002.

[24] V. Pitsikalis, A. Katsamanis, G. Papandreou, and P. Maragos. Adaptive multimodal fusion by uncertainty compensation. In *Interspeech*, 2006.

[25] P. Lucey and S. Sridharan. Patch-based representation of visual speech. In *HCSNet Workshop on Use of Vision in Human-Computer Interaction*, pages 79–85, 2006.

[26] G. Papandreou, A. Katsamanis, V. Pitsikalis, and P. Maragos. Multimodal fusion and learning with uncertain features applied to audiovisual speech recognition. In *Workshop on Multimedia Signal Processing*, pages 264–267. IEEE, 2007.

[27] G. Zhao, M. Barnard, and M. Pietikainen. Lipreading with local spatiotemporal descriptors. *IEEE Transactions on Multimedia*, 11(7):1254–1265, 2009.

[28] M. Gurban and J.-P. Thiran. Information theoretic feature extraction for audio-visual speech recognition. *IEEE Transactions on Signal Processing*, 57(12):4765–4776, 2009.

[29] G. Papandreou, A. Katsamanis, V. Pitsikalis, and P. Maragos. Adaptive multimodal fusion by uncertainty compensation with application to audiovisual speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(3):423–435, 2009.

[30] A. J. Goldschen, O. N. Garcia, and E. D. Petajan. Continuous automatic speech recognition by lipreading. In *Motion-Based recognition*, pages 321–343. Springer, 1997.

[31] G. Potamianos, E. Cosatto, H. P. Graf, and D. B. Roe. Speaker independent audio-visual database for bimodal asr. In *Audio-Visual Speech Processing: Computational & Cognitive Science Approaches*, 1997.

[32] G. Potamianos and H. P. Graf. Discriminative training of hmm stream exponents for audio-visual speech recognition. In *International Conference on Acoustics, Speech and Signal Processing*, volume 6, pages 3733–3736. IEEE, 1998.

[33] G. Potamianos, H. P. Graf, and E. Cosatto. An image transform approach for hmm based automatic lipreading. In *Conference on Image Processing*, pages 173–177. IEEE, 1998.

[34] C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, J. Sison, and A. Mashari. Audio visual speech recognition. Technical report, IDIAP, 2000.

[35] S. Gergen, S. Zeiler, A. H. Abdelaziz, R. Nickel, and D. Kolossa. Dynamic stream weighting for turbo-decoding-based audiovisual ASR. In *Interspeech*, pages 2135–2139, 2016.

[36] K. Paleček. Utilizing lipreading in large vocabulary continuous speech recognition. In *Speech and Computer*, pages 767–776. Springer, 2017.

[37] A. B. Hassanat. Visual speech recognition. In *Speech and Language Technologies*. InTech, 2011.

[38] H. L. Bear and R. Harvey. Decoding visemes: Improving machine lip-reading. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 2009–2013. IEEE, 2016.

[39] G. Potamianos, C. Neti, J. Luettin, and I. Matthews. Audio-visual automatic speech recognition: An overview. *Issues in Visual and Audio-Visual Speech Processing*, 22:23, 2004.

[40] Z. Zhou, G. Zhao, X. Hong, and M. Pietikäinen. A review of recent advances in visual speech decoding. *Image and Vision Computing*, 32(9):590–605, 2014.

[41] 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.

[42] J. S. Chung and A. Zisserman. Out of time: Automated lip sync in the wild. In *ACCV Workshop on Multi-view Lip-reading*, 2016.

[43] T. Le Cornu and B. Milner. Generating intelligible audio speech from visual speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 25(9):1751–1761, 2017.

[44] A. Ephrat and S. Peleg. Vid2speech: Speech reconstruction from silent video. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 5095–5099. IEEE, 2017.

[45] H. Akbari, H. Arora, L. Cao, and N. Mesgarani. Lip2AudSpec: Speech reconstruction from silent lip movements video. *arXiv preprint arXiv:1710.09798*, 2017.

[46] A. Gabbay, A. Shamir, and S. Peleg. Visual speech enhancement using noise-invariant training. *arXiv preprint arXiv:1711.08789*, 2017.

[47] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619*, 2018.

[48] T. Afouras, J. S. Chung, and A. Zisserman. The conversation: Deep audio-visual speech enhancement. *arXiv preprint arXiv:1804.04121*, 2018.

[49] C. Chambers, A. Raniwala, F. Perry, S. Adams, R. R. Henry, R. Bradshaw, and N. Weizenbaum. Flumejava: easy, efficient data-parallel pipelines. In *Sigplan Notices*, volume 45, pages 363–375. ACM, 2010.

[50] H. Liao, E. McDermott, and A. Senior. Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription. In *Workshop on Automatic Speech Recognition and Understanding*, pages 368–373. IEEE, 2013.

[51] H. Liao, G. Pundak, O. Siohan, M. K. Carroll, N. Coccaro, Q.-M. Jiang, T. N. Sainath, A. Senior, F. Beaufays, and M. Bacchiani. Large vocabulary automatic speech recognition for children. In *ISCA*, 2015.

[52] V. Kuznetsov, H. Liao, M. Mohri, M. Riley, and B. Roark. Learning n-gram language models from uncertain data. In *Interspeech*, pages 2323–2327, 2016.

[53] H. Soltau, H. Liao, and H. Sak. Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition. In *Interspeech*, pages 3707–3711, 2017.

[54] A. Salcianu, A. Golding, A. Bakalov, C. Alberti, D. Andor, D. Weiss, E. Pitler, G. Coppola, J. Riesa, K. Ganchev, M. Ringgaard, N. Hua, R. McDonald, S. Petrov, S. Istrate, and T. Koo. Compact language detector v3. https://github.com/google/cld3, 2018.

[55] J. C. Wells. Computer-coding the IPA: A proposed extension of SAMPA. *Revised draft*, 4(28):1995, 1995.

[56] J. Mas and G. Fernandez. Video shot boundary detection based on color histogram. *Notebook Papers TRECVID2003, Gaithersburg, Maryland, NIST*, 15, 2003.

[57] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[58] A. Torfi, S. M. Iranmanesh, N. Nasrabadi, and J. Dawson. 3d convolutional neural networks for cross audio-visual matching recognition. *IEEE Access*, 5:22081–22091, 2017.

[59] K. Mase and A. Pentland. Automatic lipreading by optical-flow analysis. *Systems and Computers in Japan*, 22(6):67–76, 1991.

[60] M. S. Gray, J. R. Movellan, and T. J. Sejnowski. Dynamic features for visual speechreading: A systematic comparison. In *Advances in Neural Information Processing Systems*, pages 751–757, 1997.

[61] T. Yoshinaga, S. Tamura, K. Iwano, and S. Furui. Audio-visual speech recognition using lip movement extracted from side-face images. In *International Conference on Audio-Visual Speech Processing*, 2003.

[62] S. Tamura, K. Iwano, and S. Furui. Multi-modal speech recognition using optical-flow analysis for lip images. In *Real World Speech Processing*, pages 43–50. Springer, 2004.

[63] S.-L. Wang, A. W.-C. Liew, W. H. Lau, and S. H. Leung. An automatic lipreading system for spoken digits with limited training data. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(12): 1760–1765, 2008.

[64] A. A. Shaikh, D. K. Kumar, W. C. Yau, M. C. Azemin, and J. Gubbi. Lip reading using optical flow and support vector machines. In *International Congress on Image and Signal Processing*, volume 1, pages 327–330. IEEE, 2010.

[65] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICRL*, 2015.

[66] Y. Wu and K. He. Group normalization. *arXiv preprint arXiv:1803.08494*, 2018.

[67] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[68] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *International Conference on Machine Learning*, pages 369–376, 2006.

[69] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, pages 173–182, 2016.

[70] K. Rao, H. Sak, and R. Prabhavalkar. Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer. In *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*, pages 193–199. IEEE, 2017.

[71] R. Prabhavalkar, K. Rao, T. Sainath, B. Li, L. Johnson, and N. Jaitly. A comparison of sequence-to-sequence models for speech recognition. In *Interspeech*, 2017.

[72] Y. Miao, M. Gowayyed, and F. Metze. Eesen: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *Workshop on Automatic Speech Recognition and Understanding*, pages 167–174. IEEE, 2015.

[73] I. McGraw, R. Prabhavalkar, R. Alvarez, M. G. Arenas, K. Rao, D. Rybach, O. Alsharif, H. Sak, A. Gruenstein, F. Beaufays, et al. Personalized speech recognition on mobile devices. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 5955–5959. IEEE, 2016.

[74] M. Mohri, F. Pereira, and M. Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002.

[75] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[76] J. L. Pech-Pacheco, G. Cristóbal, J. Chamorro-Martinez, and J. Fernández-Valdivia. Diatom autofocusing in brightfield microscopy: A comparative study. In *Pattern Recognition*, volume 3, pages 314–317. IEEE, 2000.

[77] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*,2014.

[78] The Health Foundation. New safety collaborative will improve outcomes for patients with tracheostomies. http://www.health.org.uk/news/new-safety-collaborative-will-improve- outcomes-patients-tracheostomies, 2014. Accessed: 2018-04-23.

[79] M. A. Morris and A. N. Kho. Silence in the EHR: Infrequent documentation of aphonia in the electronic health record. *BMC Health Services Research*, 14(1):425, 2014.

A Medical applications

As a consequence of injury or disease and its associated treatment, millions of people worldwide have communication problems preventing them from generating sound. As hearing aids and cochlear transplants have transformed the lives of people with hearing loss, there is potential for lip reading technology to provide alternative communication strategies for people who have lost their voice.

Aphonia is the inability to produce voiced sound. It may result from injury, paralysis, removal or other disorders of the larynx. Common examples of primary aphonia include bilateral recurrent laryngeal nerve damage as a result of thyroidectomy *(removal of the thyroid gland and any tumour)* for thyroid cancer, laryngectomy *(surgical removal of the voice box)* for laryngeal cancers, or tracheostomy *(the creation of an alternate airway in the neck bypassing the voicebox)*.

Dysphonia is difficulty in speaking due to a physical disorder of the mouth, tongue, throat, or vocal cords. Unlike aphonia, patients retain some ability to speak. For example, in Spasmodic dysphonia, a disorder in which the laryngeal muscles go into periods of spasm, patients experience breaks or interruptions in the voice, often every few sentences, which can make a person difficult to understand.

We see this work having potential medical applications for patients with aphonia or dysphonia in at least two distinct settings. Firstly, an acute care setting (i.e. a hospital with an emergency room and an intensive care unit), patients frequently undergo elective (planned) or emergency (unplanned) procedures (e.g. Tracheostomy) which may result in aphonia or dysphonia. In the U.S. 103,925 tracheostomies were performed in 2014, resulting in an average hospital stay of 29 days [21]. Similarly, in England and Wales 15,000 tracheostomies are performed each year [78].

Where these procedures are unplanned, there is often no time or opportunity to psychologically prepare the patient for their loss of voice, or to teach the patient alternative communication strategies. Some conditions that necessitate tracheotomy, such as high spinal cord injuries, also affect limb function, further hampering alternative communication methods such as writing.

Even where procedures are planned, such as for head and neck cancers, despite preparation of the patient through consultation with a speech and language therapist, many patients find their loss of voice highly frustrating especially in the immediate post-operative period.

Secondly, where surgery has left these patients cancer-free, they may live for many years, even decades without the ability to speak effectively, in these patients we can envisage that they may use this technology in the community, after discharge from hospital. While some patients may either have tracheotomy reversed, or adapt to speaking via a voice prosthesis, electro-larynx or esophageal speech, many patients do not achieve functional spoken communication. Even in those who achieve good face-to-face spoken communication, few laryngectomy patients can communicate effectively on the telephone, and face the frequent frustration of being hung-up on by call centres and others who do not know them.

Acute care applications. It is widely acknowledged that patients with communication disabilities, including speech impairment or aphonia can pose significant challenges in the clinical environment, especially in acute care settings, leading to potentially poorer quality of care [79]. While some patients will be aware prior to surgery that they may wake up unable to speak, for many patients in the acute setting (e.g. Cervical Spinal Cord Injury, sudden airway obstruction) who wake up following an unplanned tracheotomy, their sudden inability to communicate can be phenomenally distressing.

Community applications. Patients who are discharged from hospital without the ability to speak, or with poor speech quality, face a multitude of challenges in day-to-day life which limits their independence, social functioning and ability to seek employment.

We hypothesize that the application of technology capable of lip-reading individuals with the ability to move their facial muscles, but without the ability to speak audibly could significantly improve quality of life for these patients. Where the application of this technology improves the person's ability to communicate over the telephone, it would enhance not only their social interactions, but also their ability to work effectively in jobs that require speaking over the phone.

Finally, in patients who are neither able to speak, nor to move their arms, this technology could represent a step-change in terms of the speed at which they can communicate, as compared to eye-tracking or facial muscle based approaches in use today.

B Phoneme confusion matrix

To compute the confusion matrix and the insertion/deletion chart shown in the main text in Figure 4, we first compute the edit distance dynamic programming matrix between each predicted sequence of phonemes and the

corresponding ground-truth. Then, a backtrace through this matrix gives an alignment of the two sequences, consisting of edit operations paired with positions in the prediction/ground-truth sequences.

Counting the correct phonemes and the substitutions yields the confusion matrix Figure 6. The reader can note that the diagonal is strongly dominant. A few groups are commonly confused as expected due to their visual similarity, such as {/d/, /n/, /t/}, and to a lesser extent {/b/, /p/}.

Counting insertions/deletions yields Figure 4 in the main text, showing which phonemes are most commonly omitted (deleted), or less frequently, erroneously inserted.

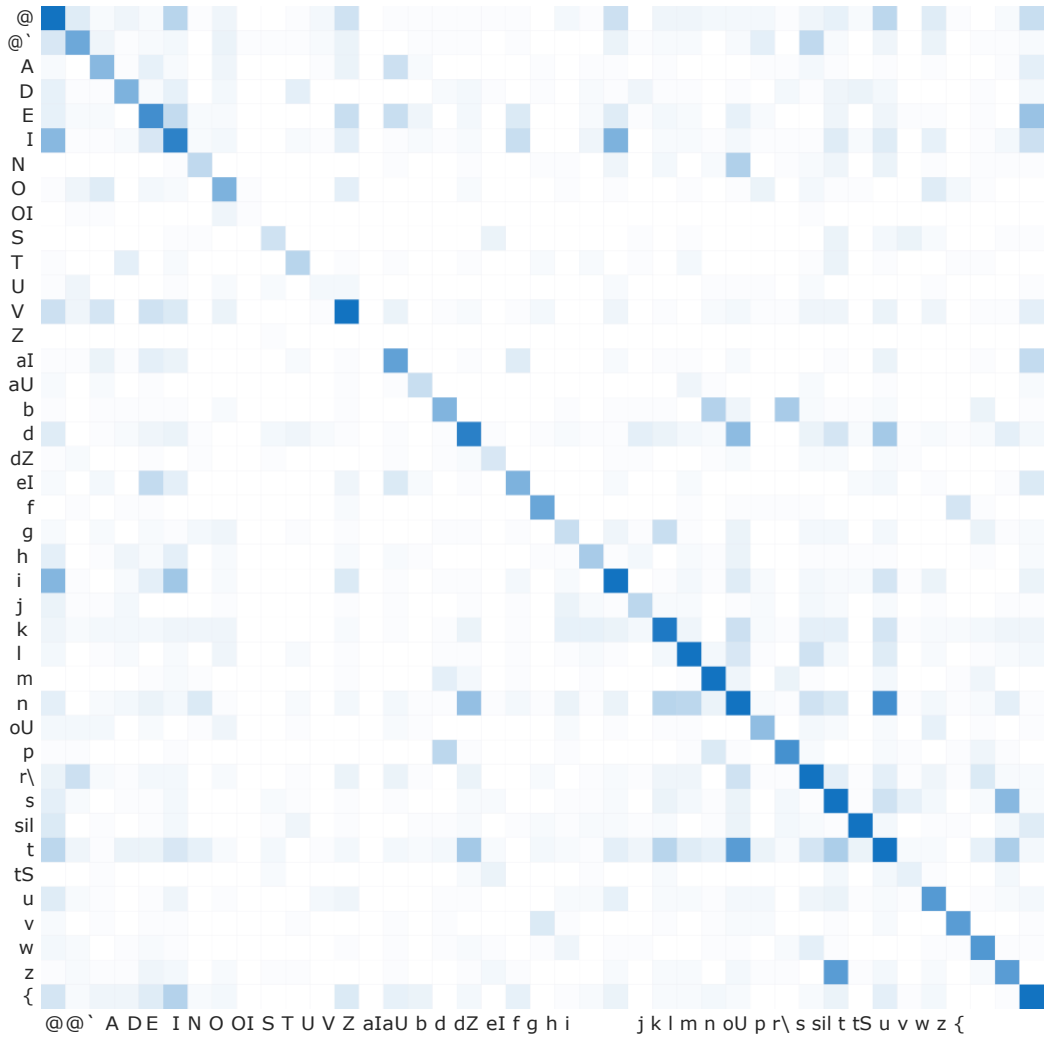

Figure 6: Phoneme confusion matrix for V2P, estimated by computing the edit distance alignment between each predicted sequence of phonemes and the corresponding ground-truth, and counting the correct phonemes and the substitutions. The diagonal values are scaled downwards to de-emphasize the correct phonemes. Blue indicates more substitutions occurred.

C Saliency Maps

(a) Transcript: “sharing” - ground truth phonemes: /S E r\@ N/.

(b) Transcript: “lot of”, ground truth phonemes: /l A t V v/.

(c) Transcript: “how was”, ground truth phonemes /I t w V z/.

(d) Transcript: “exactly”, ground truth phonemes: /@ g z { k t l i/.

(e) Transcript: “that’s a”, ground truth phonemes: /D { t s {/.

Figure 7: Saliency maps, the top-3 predictions of each frame and the ground truth phonemes.

D V2P-Sync architecture

The V2P-Sync networks in Tables 2 and 3 are optimized using a batch size of 128, batch normalization, and Adam [75] with a learning rate of 10^{-4} and default hyperparameters: first and second momentum coefficients 0.9 and 0.999 respectively, and $\epsilon = 10^{-8}$ for numerical stability.

Table 2: V2P-Sync video embedding neural network architecture.

Layer	Filter size	Stride	Output channels	Input
conv1	$3 \times 3 \times 3$	$1 \times 2 \times 2$	16	$9 \times 128 \times 128 \times 1$
pool1	$1 \times 2 \times 2$	$1 \times 2 \times 2$		$7 \times 63 \times 63 \times 16$
conv2	$3 \times 3 \times 3$	$1 \times 1 \times 1$	32	$7 \times 31 \times 31 \times 16$
pool2	$1 \times 2 \times 2$	$1 \times 2 \times 2$		$5 \times 29 \times 29 \times 32$
conv3	$3 \times 3 \times 3$	$1 \times 1 \times 1$	64	$5 \times 14 \times 14 \times 32$
pool3	$1 \times 2 \times 2$	$1 \times 2 \times 2$		$3 \times 12 \times 12 \times 64$
conv4	$3 \times 3 \times 3$	$1 \times 1 \times 1$	128	$3 \times 6 \times 6 \times 64$
pool4	$1 \times 2 \times 2$	$1 \times 2 \times 2$		$1 \times 4 \times 4 \times 128$
fc5		$1 \times 1 \times 1$	256	512
fc6		$1 \times 1 \times 1$	64	256

Table 3: V2P-Sync audio embedding neural network architecture.

Layer	Support	Stride	Filters	Input
conv1	3×5	1×1	16	$16 \times 40 \times 1$
pool1	1×2	1×2	$14 \times 36 \times 16$	
conv2	3×4	1×1	32	$14 \times 36 \times 16$
conv3	3×4	1×1	32	$12 \times 15 \times 32$
pool3	1×2	1×2	$10 \times 12 \times 32$	
conv4	3×3	1×1	64	$10 \times 6 \times 32$
conv5	3×3	1×1	64	$8 \times 4 \times 64$
conv6	3×2	1×1	128	$6 \times 2 \times 64$
fc7		1×1	256	512
fc8		1×1	64	256

E V2P architecture

Table 4: V2P architecture details.

Layer	Filter size	Stride	Output channels	Input
conv1	$3 \times 3 \times 3$	$1 \times 2 \times 2$	64	$T \times 128 \times 128 \times 3$
pool1	$1 \times 2 \times 2$	$1 \times 2 \times 2$		$T \times 63 \times 63 \times 64$
conv2	$3 \times 3 \times 3$	$1 \times 1 \times 1$	128	$T \times 31 \times 31 \times 64$
pool2	$1 \times 2 \times 2$	$1 \times 2 \times 2$		$T \times 29 \times 29 \times 128$
conv3	$3 \times 3 \times 3$	$1 \times 1 \times 1$	256	$T \times 14 \times 14 \times 128$
pool3	$1 \times 2 \times 2$	$1 \times 2 \times 2$		$T \times 12 \times 12 \times 256$
conv4	$3 \times 3 \times 3$	$1 \times 1 \times 1$	512	$T \times 6 \times 6 \times 256$
conv5	$3 \times 3 \times 3$	$1 \times 1 \times 1$	512	$T \times 4 \times 4 \times 512$
pool5	$1 \times 2 \times 2$	$1 \times 1 \times 1$		$T \times 2 \times 2 \times 512$
bilstm6			768×2	$T \times 512$
bilstm7			768×2	$T \times 1536$
bilstm8			768×2	$T \times 1536$
fc9			768	$T \times 1536$
fc10			$41 + 1$	$T \times 768$

F Face rotation vs. performance heatmap

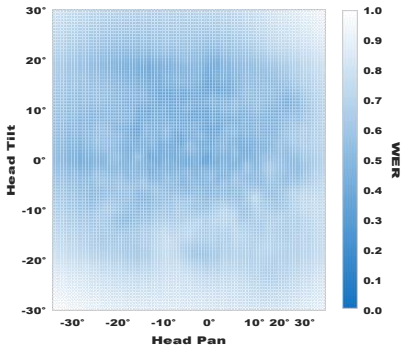

Figure 8: Heatmap showing the performance of V2P on different head rotations. Tilt and pan axes are in degrees. As shown, it performs similarly at all pan and tilt angles in $[-30°, 30°]$, the range at which it was trained.

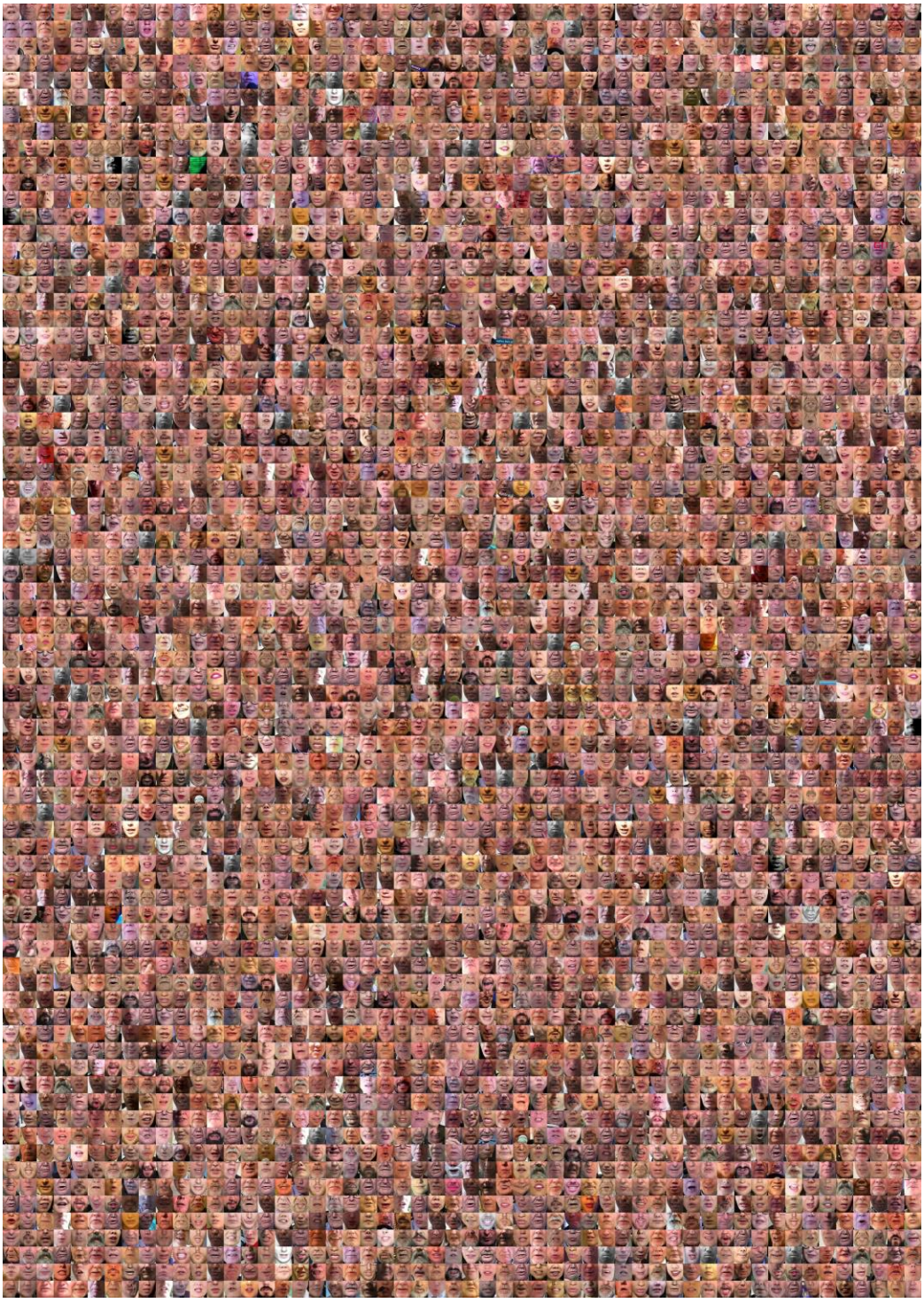

Figure 9: Random sample of test-set lip images from LSVSR. This illustrates the substantial diversity in our dataset.