LipType: A Silent Speech Recognizer Augmented with an Independent Repair Model

Laxmi Pandey
Human-Computer Interaction Group
University of California, Merced
Merced, California, United States
lpandey@ucmerced.edu

Ahmed Sabbir Arif
Human-Computer Interaction Group
University of California, Merced
Merced, California, United States
asarif@ucmerced.edu

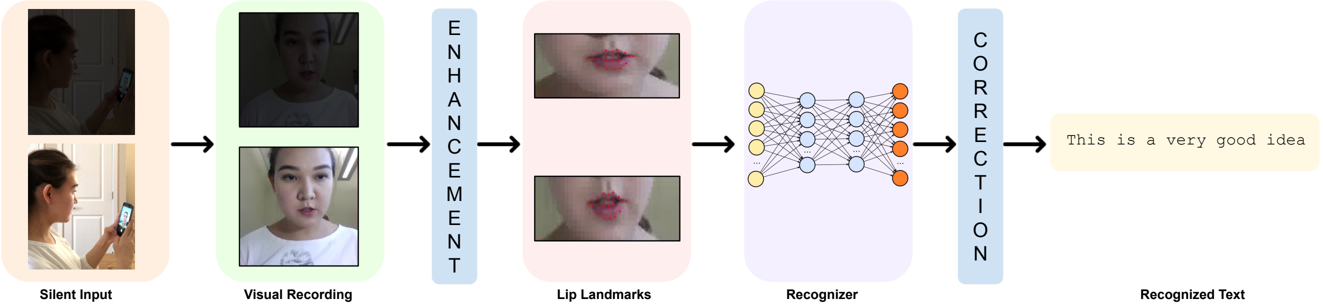


Figure 1: An overview of LipType: Automatic segmentation of lip sequences and its classification into text with an end-to-end deep neural network.

ABSTRACT

Speech recognition is unreliable in noisy places, compromises privacy and security when around strangers, and inaccessible to people with speech disorders. Lip reading can mitigate many of these challenges but the existing silent speech recognizers for lip reading are error prone. Developing new recognizers and acquiring new datasets is impractical for many since it requires enormous amount of time, effort, and other resources. To address these, first, we develop LipType, an optimized version of LipNet for improved speed and accuracy. We then develop an independent repair model that processes video input for poor lighting conditions, when applicable, and corrects potential errors in output for increased accuracy. We then test this model with LipType and other speech and silent speech recognizers to demonstrate its effectiveness.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; • **Human-centered computing** → **Text input**.

KEYWORDS

Silent speech recognition; deep learning; language modeling; text input

CHI '21, May 8–13, 2021, Yokohama, Japan
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8096-6/21/05.
https://doi.org/10.1145/3411764.3445565

ACM Reference Format:
Laxmi Pandey and Ahmed Sabbir Arif. 2021. LipType: A Silent Speech Recognizer Augmented with an Independent Repair Model. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3411764.3445565

1 INTRODUCTION

There are numerous scenarios where speech is not a viable mode of communication. First, the surroundings may not be favorable for speech-based communication: a person could be near a busy market or in a crowded restaurant where the surrounding noise makes speech difficult to recognize. Second, a person may not wish to speak out loud because of privacy and security concerns or could be in a public setting where others do not want to be disturbed, such as in a library or museum. Finally, and most importantly, many people have difficulties in speaking or are unable to speak entirely due to a range of speech and neurological disorders[1]. Although many augmentative and alternative communication (AAC) devices are available to help them vocalize, these devices produce unnatural sounding vocalization. This prevents users from communicating effectively with other humans and technologies like voice-controlled virtual assistants. Hence, the development of better communication methods is needed to improve this population's accessibility to the fellow humans and latest technological advancements. A system that can understand speech by visually interpreting the movements of the speaker's lips, known as lip reading or silent speech recognition (Fig. 1), can mitigate many of these challenges. Since developing a new system and acquiring new datasets require

[1] Statistics on voice, speech, and language, https://www.nidcd.nih.gov/health/statistics/statistics-voice-speech-and-language

an enormous amount of time, effort, and other resources, in this work we exploited a state-of-the-art silent speech recognizers, LipNet [9]. Based on preliminary investigations, we found out that LipNet and other existing recognizers have substantially slower response time due to their architecture. Besides, they do not perform well under poor lighting conditions and tend to make lexical and linguistic errors with varying speaking rates and accents [78]. As an initial step, we developed an independent repair model to enable LipNet and other existing recognizers to perform well in dark and dusky lighting conditions where the frontal pose of the user is not clearly visible to extract meaningful information, and to compensate for the recognition errors made by the recognizer. More reliable speech and silent speech systems could potentially be used as a medium for input and interaction with various computer systems, incorporated in day-to-day usage.

The contribution of this work is thus threefold. First, the development of LipType, an optimized version of LipNet for improved speed and accuracy. Second, the development of an independent repair model, a multi-stage pipeline compensating for poor lighting conditions and potential recognition errors for increased accuracy of speech and silent speech recognizers. Third, an empirical demonstration of the the repair model's effectiveness on multiple speech and silent speech recognizers. Further, the source code[2] and dataset[3] used in this work are freely available for future research and development in the area.

2 RELATED WORK

This work intersects with four areas of interest: silent speech recognition, low-light image enhancement, recognition error correction, and silent input and interaction on mobile devices.

2.1 Silent Speech Recognition

There is a rich literature on silent speech recognition. Here, we only discuss the works that are closely related to ours (see [111] for a comprehensive review). Recently, there have been attempts to apply deep learning to silent speech recognition [6, 12, 20–23, 93]. However, most of these approaches perform only at phoneme- or word-level. Koller et al. [64] trained an image classifier using convolutional neural network (CNN) to differentiate between visemes[4] on a sign language dataset of signers mouthing words. Noda et al. [75] also used CNN to predict phonemes in spoken Japanese. Tamura et al. [97] used deep bottleneck features (DBF) to encode shallow input features, such as latent dirichlet allocation (LDA) and GA-based informative feature (GIF) [99] for word recognition. Petridis and Pantic [81] also used DBF to encode every video frame and trained a long short-term memory (LSTM) classifier for word-level classification. Wand et al. [102], on the other hand, used an LSTM with histogram of oriented gradient (HoG) input features to recognize words. Chung and Zisserman [22] developed CNN architectures for classifying multi-frame time series of lip movements. LipNet [9] is an end-to-end model for phrase-level lip reading by predicting character sequences (further discussed in a later section). Afouras et al. [3] also enabled phrase-level lip reading by utilizing an encoder-decoder structure with multi-head attentions. Chung et al. [19] developed the Watch, Listen, Attend and Spell (WLAS) network that uses dual attention mechanism for visual attention to transcribe videos of mouth motion to characters.

2.2 Low-Light Image Enhancement

The problems of underexposed low-light images are very common, solutions to mitigate it have been a popular research topic. Researchers have developed a variety of techniques that can improve image quality. The classical image enhancement methods involve two categories: i) retinex-based methods, which are based on retinex theory [65]. Recent examples of these approaches are Lime [42], naturalness preserved enhancement [104], Retinex [56], and simultaneous reflectance and illumination estimation [36]. ii) histogram equalization methods, which manipulate the gray levels of individual pixels based on the image histogram. Recent examples include contextual and variational contrast enhancement [14], weighted thresholded histogram equalization [7], and layered difference representation [66]. In recent years, several methods based on deep learning image processing techniques have been proposed. One successful example is the developed pipeline for processing low-light images based on end-to-end training of a fully-convolutional network [15]. However, they reported that their model showed imperfect results for humans faces. Another work [105] utilizes encoder-decoder network to achieve the low-light enhancement for real under exposed images. Other works [2, 67, 106] have also showed the effectiveness of deep learning methods on low light image enhancement.

2.3 Recognition Error Correction

Automatic detection and correction of recognition errors have become an important research area. The aim is to automatically detect and partially or fully correct errors, regardless of the recognition system used. Zhou et al. [110] addressed the issue of error detection in recognition systems using data-mining classifiers such as naive Bayes (NB), neural networks (NN), and support vector machines (SVM). These classifiers were trained to identify errors using confidence scores and linguistic information present in the recognized output. Another work [5] proposed extraction of additional features from the confusion networks to estimate correctness probability using logistic regression. Pellegrini et al. [79] investigated the use of a Markov chains (MC) classifier with two states: error state and correct state, to model errors. Chen et al. [17] proposed a system for error detection in conversational spoken language translation. This system uses additional features provided as the feedback of statistical machine translation (SMT), including SMT confidence estimates, posteriors from named entity detection (NED) and an automated word boundary detector to verify the word boundaries of recognition output, in order to improve error detection and correction. Sarma et al. [89] built a recognition error detector and corrector using co-occurrence analysis. In the same context, Bassil and Semaan [11] proposed a post-editing ASR error correction method based on Microsoft N-Gram dataset for detecting and correcting spelling errors generated by recognition systems. The detection process detects on-word spelling errors in reference with the Microsoft

[2] Source code, https://github.com/hci-ucm/LipType

[3] Dataset, https://www.asarif.com/resources/LipType_Data.zip

[4] Visemes are visual equivalent of phonemes. A viseme represents the position of the face and mouth when making a sound.

N-Gram dataset, and the correction process generates correction suggestions for the detected word errors by selecting the best candidate for the correction using contextual information. Other works [37, 68, 80, 91] have explored non-decoder based post-processing error detection and correction.

2.4 Silent Input and Interaction on Mobile Devices

Silent input enables users to interact with mobile devices using speech commands without the need to produce any audible sound. There have been several previous attempts in achieving silent speech interaction. These works have explored silent input and interaction techniques using different sensors (e.g., electromagnetic articulography (EMA) [33, 38, 46], electroencephalogram (EEG) [83], electromyography (EMG) [57–59, 72, 90, 103], ultrasound imaging [29, 30, 35, 51, 52, 61], [38, 46], vibrational sensors of glottal activity [74, 77, 86, 98], speech motor cortex implants [10, 13] and non-audible murmur (NAM) microphone [47, 48, 73]) to recover the speech content produced without vibration of the vocal folds by detecting tongue, facial, and throat movements. Another research [13, 28, 84, 95, 96] used a brain-computer interface (BCI) with intracortical microelectrode to predict users' intended speech information directly from the brain activity involved during speech production. Another work [53] used a multimodal imaging system for speech recognition, focusing on lip visualization. Another work [60] presented a wearable interface, AlterEgo, which utilizes EMG sensors placed on face to capture the neuromuscular signals. However, these prior works use an invasive setup, impeding the adaptability of these solutions in real-world scenarios. More recently, improvements have been made in silent speech recognition by incorporating advanced machine learning techniques and computer vision technologies [3, 6, 9, 12, 20, 21, 21–23, 82, 93]. One recent research [94] developed an interaction technique that allows users to issue commands on their smartphone through silent speech. They used front camera as a natural sensor to capture the motion of the lips, and recognize it into text.

3 LIPTYPE: AN OPTIMIZED LIPNET MODEL

We used LipNet as the backbone model based on a study comparing LipNet [9], LCANet [107], Transformer [3], and WAS [19] models. The former two are trained on GRID dataset [27], the latter two on LRS dataset [3]. In an evaluation with 50 random videos from the respective datasets, LipNet and LCANet yielded similar WER (4%), while Transformer and WAS were more error-prone (> 49% WER). Of the two best performed models, we picked LipNet as it is more widely used than LCANet.

LipNet [9] is an existing end-to-end sentence-level model that maps a variable-length sequence of video frames to text, making use of a deep 3-dimensional convolutional neural network (3D-CNN) [55], a recurrent network, and the connectionist temporal classification loss. The model was trained on grid dataset comprising of highly constrained vocabulary. Although LipNet has proven to be promising, it has several limitations. First, LipNet is focused on capturing spatial and temporal information using deep 3D-CNN that neglects the hidden information between channel correlations in spatial and temporal directions [31], limiting the performance of the architecture. Further, the use of a deep 3D-CNN unnecessarily increases computational complexity and memory intensiveness. We address these issues in LipType, an optimized version of LipNet for improved speed and accuracy.

In LipType, we combined a shallow 3D-CNN (1-layer) and a deep 2D-CNN (34-layer ResNet [44]) integrated with squeeze and excitation (SE) [50] blocks (SE-ResNet) to capture both spatial and temporal information. We used this hybrid-CNN model to address the limitations of 3D-CNN that it neglects the information between channel correlations and increases computational complexity, as well as 2D-CNN's inability to capture temporal information. SE-ResNet adaptively recalibrates channel-wise feature responses by explicitly modelling inter-dependencies between the channels to improve the quality of feature representations. Moreover, it is computationally lightweight and imposes only a slight increase in model complexity and computational burden [50]. Thus, we hypothesize that the proposed hybrid frontend module will reduce the overall computational complexity of LipNet and improve its performance.

3.1 The Network

The LipType network consists of two sub-modules (or sub-networks): a *spatiotemporal feature extraction* frontend that takes a sequence of video frames and outputs one feature vector per frame and a *sequence modeling* module that inputs the sequence of per-frame feature vectors and outputs a sentence character by character, as shown in Fig. 2. We describe these modules in the following sections.

3.1.1 Spatiotemporal Feature Extraction. It starts with the extraction of a mouth-centred cropped image of size H:100 × W:50 pixels per video frame. For this, videos are first pre-processed using DLib face detector [62] and the iBug face landmark predictor [88] with 68 facial landmarks combined with Kalman Filtering. Then, a mouth-centred cropped image is extracted by applying affine transformations. The sequence of T mouth-cropped frames are then passed to 3D-CNN, with a kernel dimension of T:5× W:7 × H:7, followed by Batch Normalization (BN) [54] and Rectified Linear Units (ReLU) [4]. The extracted feature maps are then passed through 34-layer 2D SE-ResNet that gradually decreases the spatial dimensions with depth, until the feature becomes a single dimensional tensor per time step.

3.1.2 Sequence Modeling. The extracted features are processed by 2-Bidirectional Gated Recurrent Units (Bi-GRUs) [18]. Each time-step of the GRU output is processed by a linear layer, followed by a softmax layer over the vocabulary, then an end-to-end model is trained with connectionist temporal classification (CTC) loss [40]. The softmax output is decoded with a left-to-right beam search [25] using Stanford-CTC's decoder [70] and 5-gram character language model [41] to recognize the spoken utterances. The model is capable of mapping variable-length video sequences to text sequences.

4 EXPERIMENT 1: LIPTYPE MODEL

We conducted an experiment to compare the performance of LipNet and LipType.

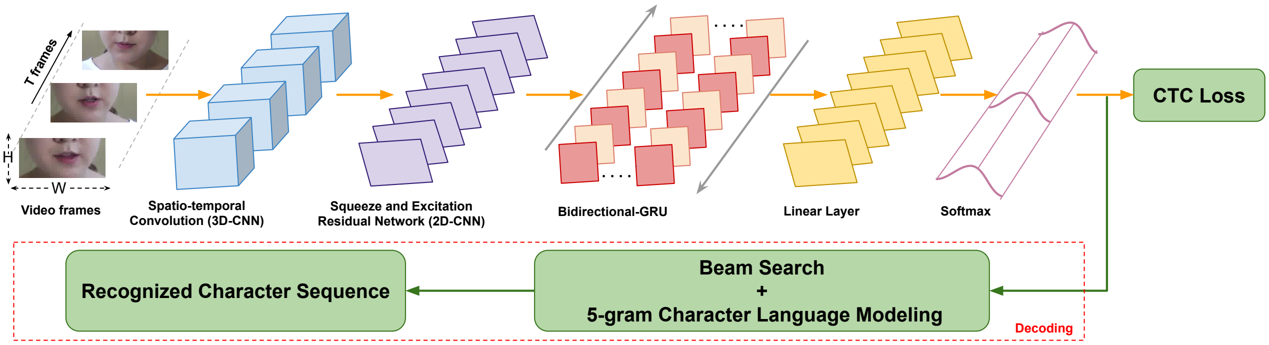

T frames
H
W
Video frames
Spatio-temporal Convolution (3D-CNN)
Squeeze and Excitation Residual Network (2D-CNN)
Bidirectional-GRU
Linear Layer
Softmax
CTC Loss
Recognized Character Sequence
Beam Search + 5-gram Character Language Modeling
Decoding

Figure 2: Architecture of LipType: a sequence of T frames is fed to a 1-layer 3D CNN, followed by 34-layer 2D SE-ResNet for spatiotemporal feature extraction. The extracted features are processed by two Bi-GRUs, followed by a linear layer and a softmax. The network is trained entirely end-to-end with CTC loss.

4.1 Dataset

For a fair comparison between the two models, we trained the LipType model on the same GRID dataset [27] on which the LipNet model was trained. It comprises of short and formulaic video clips of a person's face when uttering a highly constrained vocabulary in a specific order ($N = 34$). Similar to a previous experiment investigating the performance of LipNet with overlapped speakers [9], this experiment used 21,635 videos for training and 7,140 videos for evaluation.

4.2 Implementation

To avoid any potential confounding factor, we trained both models from scratch with the same training parameters. The number of frames was fixed to 75. Longer image sequences were truncated and shorter sequences were padded with zeros. We applied a channel-wise dropout [92] of 0.5. The model was trained end-to-end by the Adam optimizer [63] for 60 epochs with a batch size of 50. The learning rate was set to 10^{-4}. The network was implemented based on the Keras deep-learning platform with TensorFlow [1] as the backend. We trained and tested both models on NVIDIA GeForce 1080Ti GPU board.

4.3 Performance Metrics

We used the following metrics to benchmark the proposed framework.

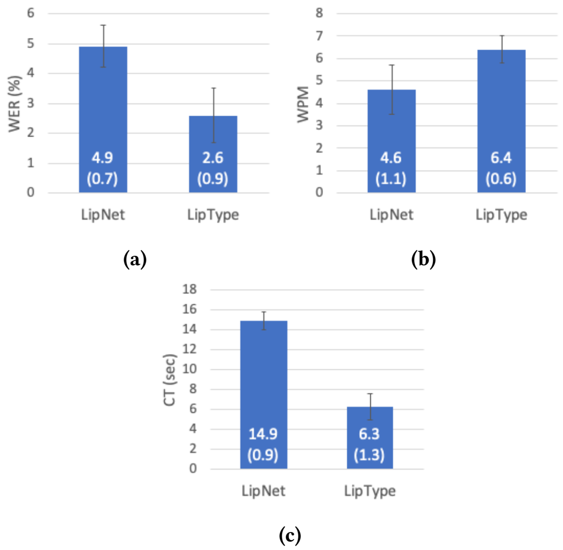
- **Word error rate (WER)** is the minimum number of operations required to transform the predicted text to the ground truth, divided by the number of words in the ground truth. It is calculated using the following equation, where S is the number of substitutions, D is the number of deletions, I is the number of insertions, and N is the number of words in the ground truth.

$$WER = \frac{S + D + I}{N} \tag{1}$$

- **Words per minute (WPM)** is a commonly used text entry metric that signifies the rate in which words (= 5 chars) are entered [8]. It is calculated using the following equation, where T is the number of recognized words, t is the sum of speaking time and computation time in seconds, the constant 60 is the number of seconds per minute, and the factor of one fifth accounts for the average length of a word in the English language.

$$WPM = \frac{|T| - 1}{t} \times 60 \times \frac{1}{5} \tag{2}$$

- **Computation time (CT)** is the total time required by the model to predict a phrase. It does not include the time users take to speak a phrase.



(a)

(b)

(c)

Figure 3: Performance comparison of LipNet and LipType in terms of a) word error rate, b) words per minute, and c) computation time. Reported values are the average of all values. Values inside the brackets are standard deviations (SD). Error bars represent ±1 standard deviation.

4.4 Results

In the experiment, LipType outperformed LipNet in terms of input speed, accuracy, and computation time. LipType achieved 2.6% WER, 6.4 WPM, and 6.3 seconds CT (Fig. 3). In comparison with LipNet, it exhibited a 47% reduction in WER, 39% increase in WPM, and 8.6 seconds reduction in CT. These findings confirm our intuition that extracting spatiotemporal features using the hybrid of a shallow 3D-CNN and a deep 2D-CNN integrated with SE blocks, instead of only 3D-CNN, will reduce the overall computational complexity and improve performance.

5 REPAIR MODEL: LIGHT ENHANCEMENT AND ERROR REDUCTION

This section presents a new repair model, a multi-stage pipeline that accounts for poor lighting conditions in input videos and potential errors in the recognition. It includes a *pre-processing* step to enhance videos with poor lighting conditions and a *post-processing* step to automatically detect and correct potential errors generated by the recognizer. A key consideration for this model was its independence, to make sure it is not reliant on a specific recognizer so that it can be used with a variety of speech and silent speech recognition models.

5.1 Pre-Processing: Light Enhancement

There are various factors that can affect the performance of silent speech recognition, for example, uncontrolled lighting, blur, low-resolution, compression artifacts, occlusions, viewing angles, accent, pace of speech, etc. However, most of the factors can be mitigated by replacing the hardware (blur, low-resolution, compression artifacts, etc.) or by the user (occlusions, viewing angles, pace of speech, etc.). Lighting, in contrast, is one the factor that cannot always be controlled.

Making recognition more reliable under uncontrolled lighting conditions is one of the major challenges for practical silent speech recognition models. Existing models do not account for lighting variations, making them unreliable in poorly lit places. We tackle this by adding a pre-processing step to the LipType recognition model. For this, we improved GLADNet [105], a low-light image enhancement network, and adapted it for enhancing input videos. We used GLADNet because it demonstrated a much better performance with actual under-exposed images compared to the other models, both in terms of quality [32, 36, 42, 56] and computation complexity [2, 15, 67, 106].

5.1.1 *The Network.*

The light enhancement network learns an end-to-end mappings from low-light images to normal-light images. It processes videos in a frame-by-frame manner, as illustrated in Fig. 4. The architecture of the network comprises of two adjacent steps: the first is for *global illumination estimation* and the second is for *detail reconstruction.*

In the global illumination estimation step, input is down-sampled to a fixed size feature map using nearest-neighbor interpolation. Then, it is passed through an encoder-decoder network[5] to estimate the global illumination of the input. The estimated feature maps are then re-scaled to the original size using a resize convolution block. Then, the re-scaled feature maps are passed to the detail reconstruction step comprising of three convolutional layers. This step adjusts the illumination of the input image by assembling predicted global illumination and input image information, and fills in the details lost during the down- and up-sampling processes. Inspired by a previous work [109], we investigated the consequences of replacing the L1 loss function used in the training of GLADNet with alternative loss functions. Given a collection of N training sample pairs X_i , Y_i, where X_i is low-light input image and Y_i is normal-light ground truth image, the following loss functions can be defined.

(1) **L1 Loss** (or mean-absolute-error loss) minimizes the sum of the absolute differences between the predicted or generated image and the ground truth.

$$L1(X, Y) = \frac{1}{N}\sum_{i=1}^{N}(X_i - Y_i) \tag{3}$$

(2) **L2 Loss** (or mean-squared-error loss) minimizes the sum of the squared differences between the predicted or generated image and the ground truth.

$$L2(X, Y) = \frac{1}{N}\sum_{i=1}^{N}(X_i - Y_i)^2 \tag{4}$$

(3) **Multi-scale structural similarity loss** (MSSSIM) [109] minimizes the loss related to the sum of structural-similarity scores across all image pixels, in terms of luminance, contrast, and structure.

$$MSSSIM(X, Y) = -\sum_{i=1}^{N} MSSSIM(X_i - Y_i) \tag{5}$$

(4) **MSSSIM-L1 loss** captures MSSSIM's ability to preserve the contrast in high-frequency regions and L1's ability to preserves colors and luminance. In the equation below, G is the Gaussian filter, α is the weighting factor to roughly balance the contribution of the two losses. We empirically set $\alpha = 0.81$[6].

$$\text{MSSSIM-L1}(X, Y) = \alpha \cdot MSSSIM + (1 - \alpha) \cdot G_\sigma \cdot L1 \tag{6}$$

(5) **MSSSIM-L2 loss** captures MSSSIM's ability to preserve the contrast in high-frequency regions and L2's ability to remove noise and ringing artifacts. Like MSSSIM-L1, $\alpha = 0.81$ and G is the Gaussian filter.

$$\text{MSSSIM-L2}(X, Y) = \alpha \cdot MSSSIM + (1 - \alpha) \cdot G_\sigma \cdot L2 \tag{7}$$

5.2 Experiment 2: Light Enhancement Network

We evaluated the performance of the light enhancement network trained with the above five loss functions.

5.2.1 *Dataset.*

We trained and validated the network on the GLADNet dataset [105] that comprises of 5,000 image pairs of low and normal light images. We used 4,000 pairs for training and the remaining 1,000 pairs for testing.

[5] In order to reduce computation, we changed the GLADNet network dimension from five down- and five up-sampling blocks to three down- and three up-sampling blocks. A preliminary investigation did not identify a significant effect on variations in layer dimensions on the network's performance.

[6] In an investigation, results were not affected by small variations in α.

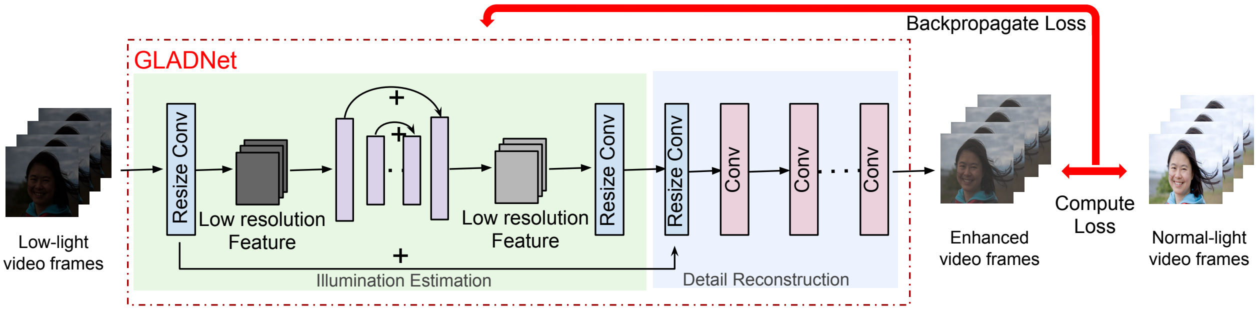


Figure 4: Architecture of the pre-processing (light enhancement) network: a sequence of low-light images is fed through the network where the enhanced images are compared with the normal-light images to compute the loss, which is then backpropagated to fine-tune and optimize the model weights and biases.

5.2.2 Performance Metrics. We used the following two standard image quality metrics [49].

- **Peak signal to noise ratio (PSNR)** computes the peak signal-to-noise ratio between two images in decibels. This ratio is used as a quality measurement between the original and an enhanced image. The higher the PSNR the better the quality of the enhanced image.
- **Structural similarity metric (SSIM)** measures the perceptual difference between two similar images. Unlike PSNR, SSIM is based on visible structures in the image. The lower the SSIM the better the quality of the enhanced image.

5.2.3 Implementation. We trained the network for 70 epochs with a batch size of 32. It was optimized using Adam [63]. The learning rate was set to 10^{-3}. The network was implemented on the Keras deep-learning platform with TensorFlow [1] as the backend. We trained and tested the network on NVIDIA GeForce 1080Ti GPU board.

5.2.4 Results. Table 1 presents the performance comparison of GLADNet trained on the aforementioned five loss functions in terms of averaged PSNR and SSIM. It can be seen that MSSSIM-L1 achieved the highest PSNR and outperformed other loss functions substantially in the SSIM measure. Therefore, we used GLADNet trained with MSSSIM-L1 loss function to enhance poor lighting input videos for more reliable silent speech recognition.

5.3 Post-Processing: Error Reduction

This section presents a new algorithm for predicting and automatically correcting potential recognition errors by a speech or silent speech recognizer. It comprises of two sub-modules: an *error minimization* module that corrects potential errors in the recognized character sequence using deep denoising autoencoder (DDA) [101] and a *sequence decoder* module that converts corrected character sequence to meaningful word sequences using spell-checker and a custom language model. The architecture of the network is illustrated in Fig. 5.

5.3.1 Error Reduction. DDA has been successful in the context of reconstructing a noisy signal [34, 69]. In this work, we used DDA to correct the character sequence predicted by the recognizer. The predicted sequence is represented in the form of a matrix, where each row is a one-hot[7] encoded vector, pointing to a particular character out of all. An input to autoencoder is converted to a fixed length sequence: 28 in this case (26 letters of the English alphabet, 1 space character, and 1 newline character), either by subdividing the sequence or by appending zero vectors, depending on the length of the sequence. This fixed length matricized sequence is fed-forwarded through a DDA to obtain an improved character sequence. The DDA is trained with the matricized incorrect character sequence as input and the matricized correct sequence as the labels. This helped in reconstructing the sequence, thus reducing the errors. In order to quantify the errors between incorrect sequence and the ground truth, we used cross-entropy loss [108], which is given by the following equation, where x represents the matricized incorrect character sequence and z represents the matricized ground truth sequence.

$$Loss(x, z) = -\sum_{k=1}^{d}[x_k log z_k + (1 - x_k) log(1 - z_k)] \tag{8}$$

5.3.2 Sequence Decoder. The corrected character sequence embedded with the space and newline characters is first combined to form a sequence of words. The resultant word sequence is then passed to the spell checker[8] to be checked for spelling correctness for auto-correction, when necessary. In addtion, a language model (LM) was used to get the most probable sequence of words. We used a traditional count-based LM[9]. Typically, n-gram analysis in count-based LM is a forward n-gram. However, we explored and evaluated the advantage of a bidirectional n-gram modeling that accounts for both forward and backward directions. Formally, we consider a string of n words, $W = w_1, w_2, ..., w_n$. In a forward n-gram, the probability of each word is estimated depending on the preceding words:

$$\begin{aligned} P_{forward}(W) =& P(w_1| < start >) * P(w_2|w_1)* \\ & P(w_3|w_2) * ... * P(< end > |w_n) \end{aligned} \tag{9}$$

[7] Encodes categorical data using a one-of-K scheme.
[8] How to write a spelling corrector, http://norvig.com/spell-correct.html
[9] A count-based LM follows the general idea of making n^{th} order Markov assumptions and calculating the n-gram probabilities through the means of counting.

Metric	**Low-Light Image**	**Enhanced Image**				
		Loss Function				
		L1	*L2*	*MSSSIM*	***MSSSIM-L1***	*MSSSIM-L2*
PSNR	19.74	26.22	25.66	26.11	**27.34**	26.13
SSIM	0.46	0.7822	0.7574	0.7890	**0.8091**	0.7911

Table 1: Averaged peak signal to noise ratio (PSNR) and structural similarity metric (SSIM) for the five investigated loss functions. For MSSSIM, the reported values are obtained as averages of the three color channels (RGB). The best results are highlighted in bold.

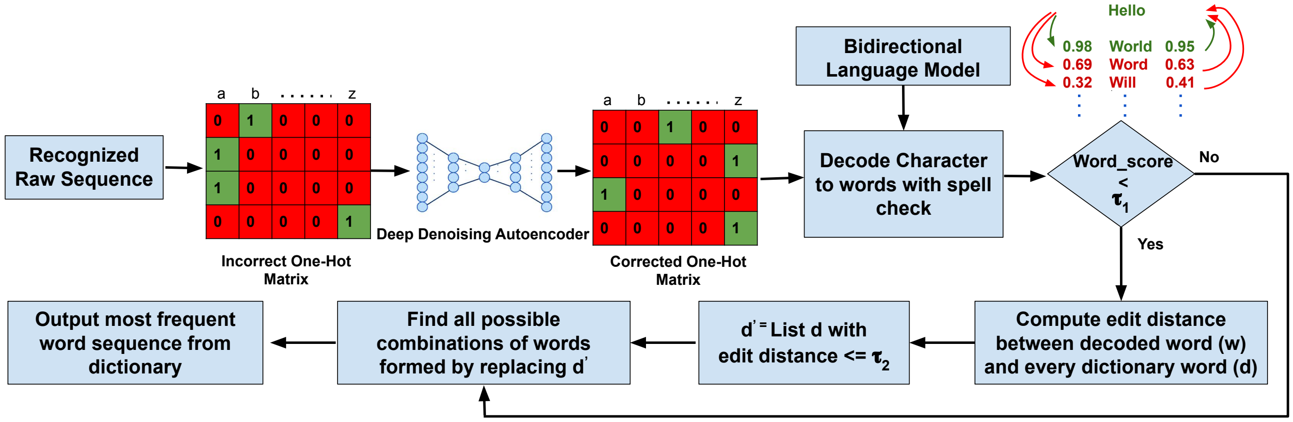

Recognized Raw Sequence
a b ...... z
0 1 0 0 0
1 0 0 0 0
1 0 0 0 0
0 0 0 0 1
Incorrect One-Hot Matrix
Deep Denoising Autoencoder
a b ...... z
0 0 1 0 0
0 0 0 0 1
1 0 0 0 0
0 0 0 0 1
Corrected One-Hot Matrix
Bidirectional Language Model
Decode Character to words with spell check
Hello
0.98 World 0.95
0.69 Word 0.63
0.32 Will 0.41
Word_score < τ1
No
Yes
Compute edit distance between decoded word (w) and every dictionary word (d)
d' = List d with edit distance <= τ2
Find all possible combinations of words formed by replacing d'
Output most frequent word sequence from dictionary

Figure 5: Architecture of post-processing (error reduction) network: the predicted raw sequence is fed to DDA, followed by spell checker and a custom language model.

In contrast, in a backward n-gram the probability of each word is estimated depending on the succeeding words:

$$\begin{aligned} P_{backward}(W) =& P(<start>|w_1) * P(w_1|w_2) * \\ & P(w_2|w_3) * \ldots * P(w_n|<end>) \end{aligned} \tag{10}$$

The combined probability of a sentence, thus, is computed by multiplying the forward and backward n-gram probability of each word:

$$\begin{aligned} P_{combined}(W) =& (P_{forward}(W_1) * P_{backward}(W_1)) * \\ & (P_{forward}(W_2) * P_{backward}(W_2)) * \\ & \ldots * \\ & (P_{forward}(W_n) * P_{backward}(W_n)) \end{aligned} \tag{11}$$

Applying the values from Equations 9 and 10, we get:

$$\begin{aligned} P_{combined}(W) =& (P(w_1|<start>) * P(<start>|w_1)) * \\ & (P(w_2|w_1) * P(w_1|w_2)) * \\ & (P(w_3|w_2) * P(w_2|w_3)) * \\ & \ldots * \\ & (P(<end>|w_n) * (w_n|<end>)) \end{aligned} \tag{12}$$

Finally, the network predicts and corrects potential errors committed by the language model in the following three steps. (1) Compare the combined probability of each word, $P_{combined}w_n = P(w_n|w_{n-1}) * P(w_{n-1}|w_n)$ (Equation 12), with a pre-defined threshold τ_1. If $P_{combined}w_n$ is less than τ_1, the word is considered erroneous. (2) compute edit distance (*ED*) between an erroneous word w_n and each dictionary word d to create a list d' of all dictionary words that have an *ED* less than a predefined threshold τ_2. (3) Replace each word in d' with $P_{combined}w_n$ in a sentence and output the most frequent word sequence from the dictionary.

We conducted an extensive study to select the best combinations of τ_1 and τ_2 by analyzing the performance of the proposed LM in the defined context.

5.4 Experiment 3: Error Reduction Model

We evaluated each sub-module of the post-processing step. First, we evaluated the architecture for the DDA network. Second, we evaluated the performance of the proposed LM. Finally, we identified the best thresholds values for computing numerical similarities.

5.4.1 Dataset. We used LIBRISPEECH LM corpus [76] to train and evaluate the post-processing modules. The dataset contains text from 14,500 public domain books. We first filtered out all punctuation, casing, and non-alphanumeric tokens from the original text and extracted the top 200,000 sentences as vocabulary.

5.4.2 Training and Evaluation of Various DDA Architectures. For training DDA, we randomly divided the dataset into 100,000 sentences as correct set and remaining 100,000 as incorrect set. We then injected one character-level error to each word of each phrase. To inject errors, we simulated the following four types of error to each word in the following sequence: one deletion error (removal of one letter), one transposition error (swapping of two adjacent letters), one replacement error (changing one letter with another), and one insertion error (one additional letter). Table 2 presents the statistics of the dataset used for training DDA. It was divided into a split of 80:20% as training:testing set.

To select the best network architecture for DDA, we trained and evaluated four different architectures (Table 3). All networks were implemented on the Keras deep-learning platform with TensorFlow [1] as the backend and an NVIDIA GeForce 1080Ti as the GPU board. We used Adam [63] as the optimization method for training. We trained the networks for 50 epochs with learning rate of 10^{-3}, batch size of 128. Results revealed that the DDA architecture with 5-layers having [128 64 32 64 128] nodes performed the best (Table 3). Hence, we used the DDA trained with this architecture to minimize potential errors in the recognized output.

5.4.3 Training and Evaluation of N-Gram Language Model. We evaluated the directional advantage of a count-based n-gram LM with state-of-the-art bi-directional neural LM in terms of sentence error rate (SER)[10], perplexity[11], and computation time. For a fair comparison, we trained both models from scratch using the LIBRISPEECH dataset (Section 5.4.1). We divided the dataset in a split of 80:20% as training: testing set. Count-based n-grams models were trained using the Natural Language Toolkit (NLTK)[12] with Kneser-Ney smoothing [16, 45] to better estimate probabilities of unseen n-grams. Bi-directional neural LM (Bi-LSTM) was trained using LSTM based recurrent units that have two recurrent layers with 4,096 LSTM nodes in each layer, an input projection layer of size 128, and an output softmax layer over vocabulary. The model was trained end-to-end using cross-entropy loss [108] with Adam [63] as the optimization method. The model was trained for 60 epochs with batch size of 64 and learning rate of $1e^{-3}$. It was implemented based on the Keras deep-learning platform with TensorFlow [1] as the backend. Both LMs were trained and tested on NVIDIA GeForce 1080Ti GPU.

In the experiment, Bi-LSTM performed better than the count-based LMs in terms of SER and perplexity (Table 4). However, it required extra computation time. Among count-based LMs, the combined trigram LM (forward and backward) performed much better. Besides, it yielded a 7.27% and 3.86% higher SER and perplexity, respectively, and a 5.8 seconds (~ 170.5%) lower computation time than Bi-LSTM. Hence, considering the negligible percentage differences in SER and perplexity and a large difference in computation time, we decided to use the combination of forward and backward trigram LM in our repair model.

5.4.4 Selection of Best Combinations of τ_1 and τ_2 to Compute Numerical Similarity. To select the best combinations of τ_1 and τ_2, we evaluated the proposed LM for various combinations of τ_1 and τ_2, in terms of true positive rate (TPR) and false positive rate (FPR), defined as:

$$TPR = \frac{TP}{TP + FN} \quad \text{and} \quad FPR = \frac{FP}{FP + TN} \tag{13}$$

TP: True positive is the total number of correct words identified as correct.
FP: False positive is the total number of incorrect words identified as correct.
TN: True negative is the total number of incorrect words identified as incorrect.
FN: False negative is the total number of correct words identified as incorrect.

Each curve in Fig. 6 signify TPR vs. FPR for different sets of τ_1 and τ_2. It can be clearly seen that the LM with $\tau_1 = 0.7$, $\tau_2 = 2$ performed best among all cases since it has a much higher TPR and a lower FPR.

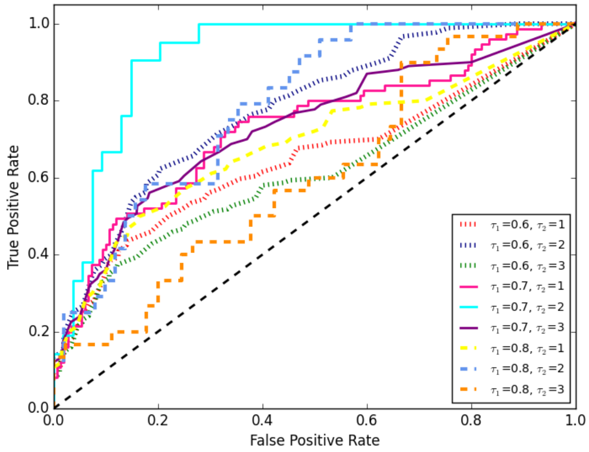


Figure 6: Performance comparison in terms of TPR and FPR of proposed LM for various values of τ_1 and τ_2

To summarize, the post-processing step include: 5-layer DDA with bi-directional count-based trigram LM, followed by numerical similarity with $\tau_1 = 0.7$, $\tau_2 = 2$.

6 EXPERIMENT 4: INDEPENDENCE OF THE REPAIR MODEL

Since our goal was to develop a repair model that can be used with a range of speech and silent speech recognizers, we evaluated its effectiveness with both LipType and several other popular speech and silent speech recognizers. Particularly, we picked the following six pre-trained models.

6.1 Silent Speech Recognizers

(1) **LipNet** [9] model uses a neural network architecture for lip reading that maps variable-length sequences of video frames to text sequences, making use of deep 3-dimensional convolutions, a recurrent network, and the connectionist temporal classification loss [40], trained entirely end-to-end.

[10] Sentence error rate (SER) signifies the percentage of recognized sentences that are not an exact match of the ground truth.
[11] Perplexity is the multiplicative inverse of the probability assigned to the sentence by the language model, normalized by the number of words in the sentence. The lower the perplexity the better the language model.
[12] Natural Language Toolkit (NLTK), https://www.nltk.org/api/nltk.lm.html

Phrase	Word	Char	Correct Word	Correct Char	Incorrect Word	Incorrect Char
200,000	6,027,754	18,527,816	2,906,117	9,279,253	3,121,637	9,248,563

Table 2: Statistics of dataset used for training DDA. The values are the total.

DDA Architecture	WER
Number of Layers: [Number of Nodes]	Mean (%)
5: [256 128 64 128 256]	21.8
5: [128 64 32 64 128]	**16.4**
3: [128 64 128]	19.1
3: [64 32 64]	26.3

Table 3: Evaluation of various DDA architectures in terms of word error rate (WER).

It was trained on the GRID dataset [27] which comprises of short and formulaic videos that show a well-lit person's face while uttering a highly constrained vocabulary in a specific order.

(2) **LipType** model follows the same architecture as LipNet except it replaces deep 3-dimensional convolutions with a combination of shallow 3-dimensional convolutions (1-layer) and deep 2-dimensional convolutions (34-layer ResNet) integrated with squeeze and excitation (SE) blocks (SE-ResNet). It was also trained on the GRID dataset.

(3) **Transformer** [3] model comprises of two sub-modules: a *spatio-temporal visual frontend* that takes a sequence of video frames to extract one feature vector per frame and a *sequence processing backend* comprised of encoder-decoder structure with multi-head attention layers [100] that generates character probabilities over the vocabulary. It was trained on Lip Reading in the Wild (LRW) [22] and the Lip Reading Sentences 2 (LRS2) [3] datasets.

6.2 Speech Recognizers

(1) **DeepSpeech** [43] is a speech recognition model developed using end-to-end training of a large recurrent neural network (RNN). It converts an input speech spectrograms into a sequence of character probabilities. It was trained on the Wall Street Journal (WSJ) [78], Switchboard [39], and Fisher [24] datasets.

(2) **Kaldi** [85] is an open-source toolkit for speech recognition written in C++, which uses Finite State Transducer (OpenFST) library [87] for training recognition models. It comprises of multiple speech recognition recipes. For our work, We used a pre-trained chain English model (Api.ai) recipe, trained on the LIBRISPEECH dataset [76].

(3) **Wave2Letter** [26] is an end-to-end model for speech recognition, that combines a convolutional network-based acoustic model and a graph decoding. It is trained to output letters without the need for force aligning them. It was trained on the LIBRISPEECH [76] dataset.

We evaluated these models on seen and unseen data. For seen data, we randomly selected 30 phrases from each model's training dataset, for unseen data, we randomly selected 30 phrases from MacKenzie and Soukoreff dataset [71]. Unseen data was common for all models. All selected phrases are listed in the Appendix A.

6.3 Experimental Conditions

We evaluated the silent speech models under three lighting conditions. Due to the spread of COVID-19, all conditions were simulated in a private room without any artificial light sources.

- **Dark light**: video recorded during nighttime (9:00–11:00 PM).
- **Dusky light**: video recorded during evening time (6:00–8:00 PM).
- **Daylight**: video recorded during daytime (1:00–3:00 PM).

Likewise, speech models were evaluated under three noisy conditions, simulated in a private room.

- **Indoor noise**: audio recording with an indoor noise, simulated by playing a prerecorded CNN news report in the background.
- **Outdoor noise**: audio recording in a public place, simulated by playing a prerecorded busy marketplace noise.
- **Quiet**: audio recording in a quiet room.

6.4 Apparatus

We developed a custom Android application with Android Studio 3.1.4 for data collection. The application included a *landing* page and a *data collection* page. The landing page included a drop-down menu to select recording conditions and a Start button to start a session. The data collection page included a video viewer to display the device's front camera, an area to presented phrases, and a Record/Stop toggle button to start and stop recording. The application recorded all videos and automatically logged the duration of a session, device specification (display and camera resolution, etc.), light intensity, and sound level.

6.5 Participants

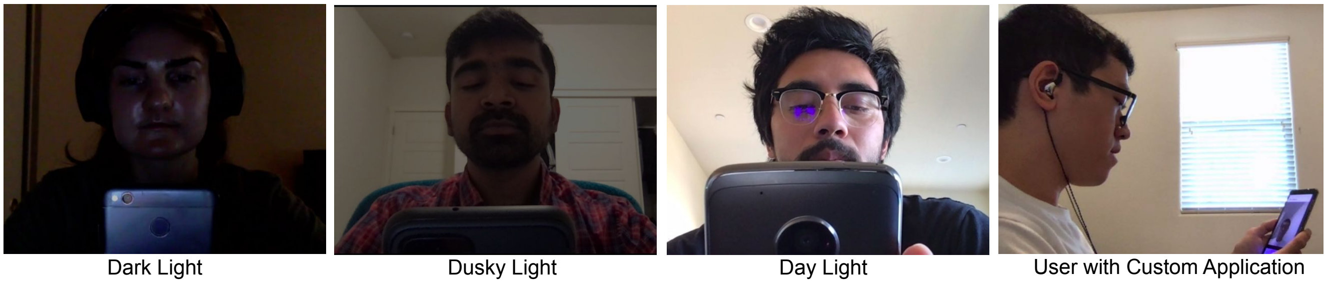
Twelve volunteers aged 19—54 years (M = 27.9, SD = 9.15) took part in the study (Fig. 7). They were all proficient in the English language. Five of them identified themselves as women and seven identified as men. They all had at least five years of experience with smartphones. All of them were Android-based smartphone users, and users of a voice assistant system for at least one year. Most of them had experience with multiple voice assistants, including Amazon Alexa, Google Assistant, and Apple Siri. They all received US $20 for participating in the study.

6.6 Design

We used the following within-subjects design for the study:

Sentence Error Rate (SER) %						
Bigram			Trigram			Bi-LSTM
Forward	Backward	Combined	Forward	Backward	**Combined**	
27.4	30.9	26.7	24.4	27.6	**16.5**	15.3
Perplexity						
Bigram			Trigram			Bi-LSTM
Forward	Backward	Combined	Forward	Backward	**Combined**	
51.3	60.1	48.7	44.1	48.3	**41.4**	39.8
Computation Time (Second)						
Bigram			Trigram			Bi-LSTM
Forward	Backward	Combined	Forward	Backward	**Combined**	
1.8	1.7	3.1	1.5	1.9	**3.4**	9.2

Table 4: Comparison between forward, backward and combination of both (forward + backward) n-gram LM with Bi-LSTM LM. Reported sentence error rate (SER), perplexity, and computation time are average of all values. The proposed repair model uses the combined trigram model.



Dark Light

Dusky Light

Day Light

User with Custom Application

Figure 7: Four volunteers participating in the user study.

12 participants ×
2 methods (speech, silent speech) ×
3 conditions (indoor, outdoor, quiet / dark, dusky, day), counterbalanced ×
2 data types (seen, unseen) ×
3 models (DeepSpeech, Kaldi, Wave2Letter / LipNet, LipType, Transformer), counterbalanced ×
30 phrases = 12,960 phrases in total.

6.7 Procedure

The study was conducted remotely due to the spread of COVID-19. We explained the purpose of the study and scheduled individual Zoom[13] video calls with each participant ahead of time. We instructed them to join the call from a quiet room to avoid any interruptions during the study. In the first call, we demonstrated the application and collected their consents and demographics using electronic forms. We then shared the application (APK file) with them and guided them through the installation process on their smartphones. The first session started shortly after that. The application displayed one phrase at a time. Participants pressed the Record button, spoke or silently spoke[14] the phrase, then pressed the Stop button to see the next phrase. In the noisy conditions (Section 6.3), we shared the respective audio clips with the participants and instructed them to play the clips slightly louder than a normal conversation. Log analysis reveled that, on average, participants played the indoor noise at 48.75 db (min = 42 db, max = 58 db) and outdoor noise at 55.25 db (min = 49 db, max = 66 db). To simulate different lighting conditions, silent speech sessions were scheduled at different times of the day. Log analysis revealed that, on average, room light intensity was 0.93 lux (min = 0 lux, max = 2 lux) in the dark light condition, 7.86 lux (min = 6 lux, max = 11 lux) in the dusky light condition, and 58.0 lux (min = 52 lux, max = 61 lux) in the daylight condition. All sessions followed the same format, expect for demonstration and installation. Upon completion of each session, participants shared the logged data with us by uploading those to a cloud storage under our supervision. In total, there were 24 recording sessions (Table 5). A researcher monitored all sessions via Zoom.

Upon completion of the study, we evaluated the repair model with the six recognition models using the collected audio and video clips. For speech, first, we passed the recorded audio to a speech recognizer, then we post-processed the output to auto-correct errors. We did not pre-process the data since speech only utilizes audio information, thus, is not affected by poor lighting conditions. For

[13] Zoom, https://zoom.us
[14] Uttering phrases without vocalizing any sound

Speech			
Session	**Condition**	**Model**	**Dataset**
1	Indoor	DeepSpeech	Fisher [24] (seen)
2	Outdoor	DeepSpeech	Fisher [24] (seen)
3	Quiet	DeepSpeech	Fisher [24] (seen)
4	Indoor	Kaldi	LIBRISPEECH [76] (seen)
5	Outdoor	Kaldi	LIBRISPEECH [76] (seen)
6	Quiet	Kaldi	LIBRISPEECH [76] (seen)
7	Indoor	Wave2Letter	LIBRISPEECH [76] (seen)
8	Outdoor	Wave2Letter	LIBRISPEECH [76] (seen)
9	Quiet	Wave2Letter	LIBRISPEECH [76] (seen)
10	Indoor	DeepSpeech/Kaldi/Wave2Letter	Mackenzie and Soukoreff [71] (unseen)
11	Outdoor	DeepSpeech/Kaldi/Wave2Letter	Mackenzie and Soukoreff [71] (unseen)
12	Quiet	DeepSpeech/Kaldi/Wave2Letter	Mackenzie and Soukoreff [71] (unseen)
Silent Speech			
13	Dark	LipNet	Grid [27] (seen)
14	Dusky	LipNet	Grid [27] (seen)
15	Day	LipNet	Grid [27] (seen)
16	Dark	LipType	Grid [27] (seen)
17	Dusky	LipType	Grid [27] (seen)
18	Day	LipType	Grid [27] (seen)
19	Dark	Transformer	LRS [3] (seen)
20	Dusky	Transformer	LRS [3] (seen)
21	Day	Transformer	LRS [3] (seen)
22	Dark	LipNet/Transformer/LipType	Mackenzie and Soukoreff [71] (unseen)
23	Dusky	LipNet/Transformer/LipType	Mackenzie and Soukoreff [71] (unseen)
24	Day	LipNet/Transformer/LipType	Mackenzie and Soukoreff [71] (unseen)

Table 5: Recording sessions for different noisy and lighting conditions with their corresponding recognition models and datasets.

silent speech, first, we processed each recorded video with the preprocessing technique to enhance the lighting of the clips, then we passed the processed videos to a silent speech recognizer, finally we post-processed the output to auto-correct errors.

6.8 Results

For evaluation, we considered all pre-trained models as baselines and compared with their respective repaired versions in terms of WER, WPM, and CT. To ensure a fair comparison of computation time, we evaluated all models on NVIDIA GeForce 1080Ti GPU board. Results revealed that the proposed repair model significantly reduce error rates of all pre-trained models regardless of data type and experimental conditions.

Fig. 8 shows the effectiveness of repair model on the three examined speech recognition models. It can be clearly observed that the repair model resulted in substantial reductions in error rates for all pre-trained models under all noisy conditions. With DeepSpeech, it showed 37.5% reduction in WER for seen data and 26.7% reduction for unseen data. With Kaldi, it showed 31.5% reduction in WER for seen data and 38% reduction for unseen data. With Wave2Letter, it showed 26.8% reduction in WER for seen data and 38.3% reduction for unseen data. On average, for all models, we observed 8.4% reduction in WPM and 5.9 seconds increase in CT on both seen and unseen data. Overall, Repaired Kaldi performed the best among all pre-trained models.

Fig. 9 shows the effectiveness of the repair model on silent speech recognition models. The performance of the repair model followed a similar trend as the speech models. It showed substantial reductions in error rates for all lighting conditions. With LipNet, it showed 58.1% reduction in WER for seen data and 15.5% reduction for unseen data. With LipType, it showed 61.9% reduction in WER for seen data and 16.3% reduction for unseen data. With Transformer, it showed 51.5% reduction in WER for seen data and 38.5% reduction for unseen data. On average, for all models, we observed 10.9% reduction in WPM and 8 seconds increase in CT on both seen and unseen data. For unseen data, we observed a negligible reduction in WER for LipNet and LipType compared to the Transformer model. We speculate that this is because LipNet and LipType are trained on a relatively small GRID dataset [27] that has a smaller number of word-level classes (shorter phrases). This resulted in a much better performance for their repair models with seen data as most of the silently spoken words were in its vocabulary. Likewise, it did not perform as well with unseen data as many of the silently spoken words were not in its vocabulary (thus could not be fully processed by the language model). Transformer, in contrast, is trained on LRS dataset [3] that has a larger number of word-level classes (longer

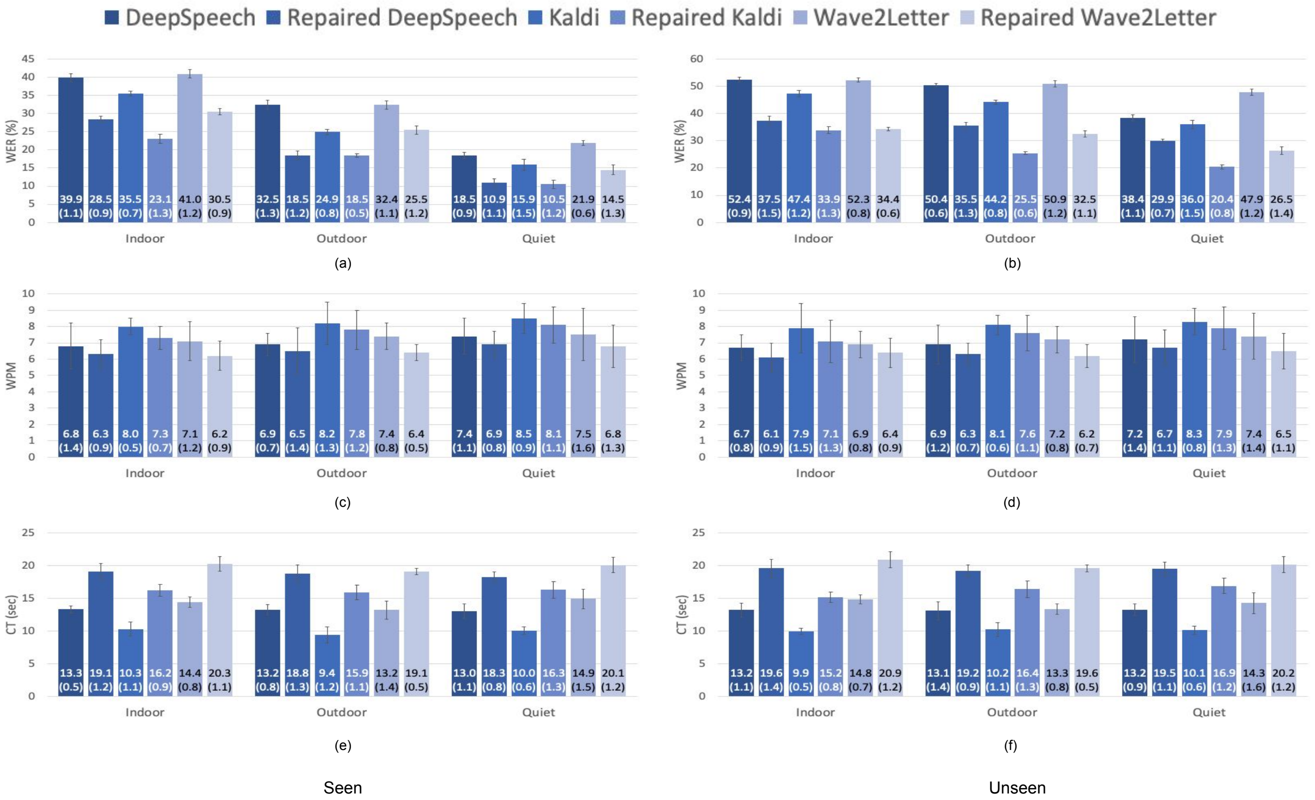


Figure 8: Performance evaluation of the three investigated speech recognition models without/with the proposed repair model in terms of a) WER-Seen, b) WER-Unseen, c) WPM-Seen, d) WPM-Unseen, e) CT-Seen, and f) CT-Unseen. Each condition has 360 data points. Reported values are the average of all values. The values inside the brackets are standard deviations (SD). Error bars represent ±1 SD.

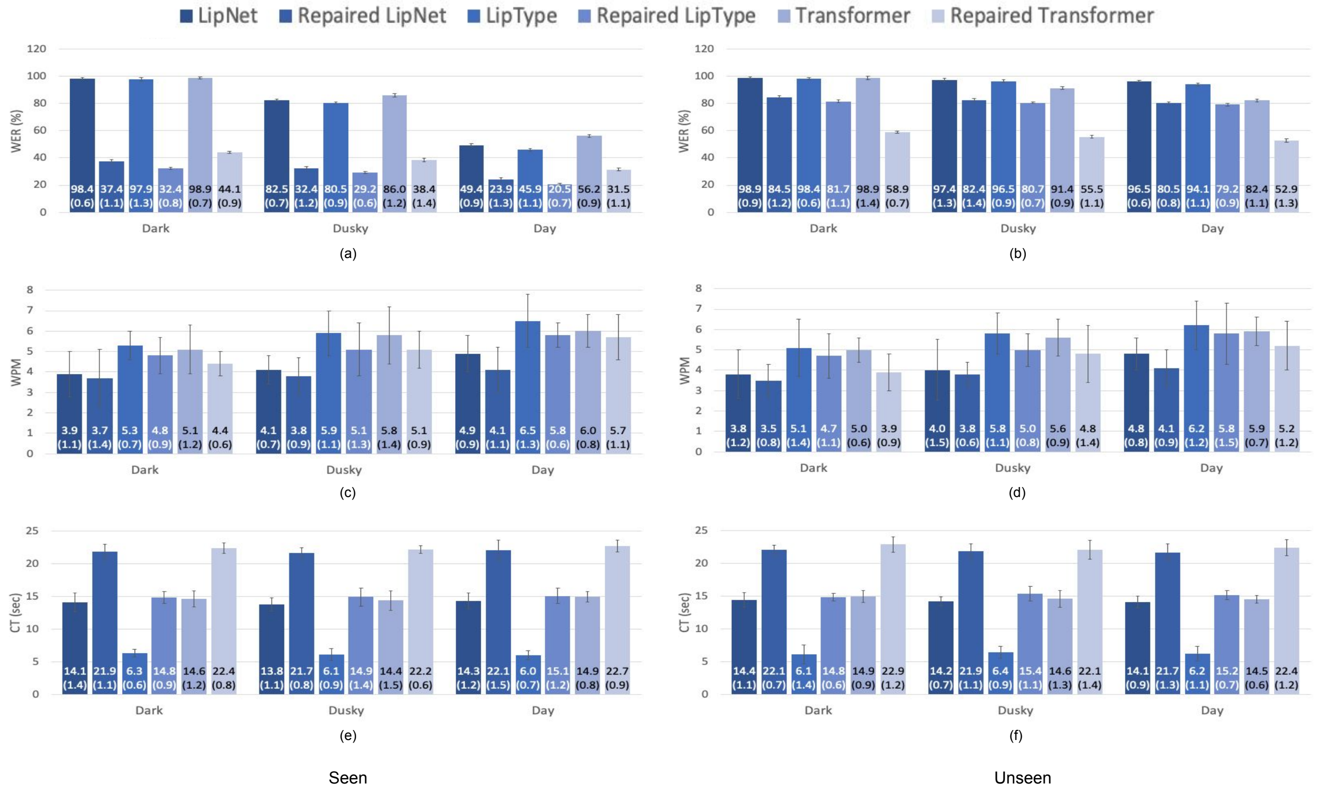
phrases). This resulted in a much lower WER for repaired Transformer with unseen data as it provided the language model with more accurate words than LipType. Note that the language model is part of the repair model not the recognizer. It is trained on a more comprehensive LIBRISPEECH dataset [76]. But its effectiveness is reliant on the vocabulary of the recognizer.

We also performed extensive ablation studies on each submodule of our model to demonstrate their contribution to the overall performance gains. All results are detailed in Appendix B.

7 DISCUSSION

We developed LipType, an optimized version of LipNet for improved speed and accuracy. LipType demonstrated a significant improvement in the performance of LipNet. Results revealed 46.9% reduction in WER, 39.1% increase in WPM, and 8.6 seconds reduction in CT. We then developed an independent repair model that processes video input for poor lighting conditions and corrects potential errors in output for increased accuracy. We evaluated the repair model's effectiveness with various speech and silent speech recognizers. To demonstrate its benefit, we selected six pretrained models, i.e., three for speech and three for silent speech. We then conducted a user study with twelve participants to collect diverse data under real-world conditions. For speech models, we collected data in indoor, outdoor, and quiet noisy conditions. For silent speech, we collected data in dark, dusky, and day lighting conditions. We then evaluated the impact of the repair model on each model's performance using the collected data. Results showed significant improvement in the performance of all models. Models augmented with the repair model outperformed the original models drastically for all experimental conditions. For speech, we observed 32% reduction in WER, 5.8 seconds increase in CT, and 8.1% reduction in WPM; whereas for silent speech, we observed 57.2% reduction in WER, 7.9 seconds increase in CT, and 10.3% reduction in WPM. Since speech models do not involve preprocessing, their repaired models showed 26.2% less CT than silent speech models.

On comparing the performance of LipNet and LipType from Fig. 3 and Fig. 9(a):Day, we observed a 45-50% reduction in their WER. We speculate that this is because the dataset used to evaluate both models for seen speakers comprises of uniform visual attributes (same skin tone, accent, pace of speech, etc.) (Fig. 3). However, the dataset for final evaluation used new speakers' data that solicited more variability in terms of speaker characteristics (Fig. 9(a):Day). We also observed that the repaired Transformer performed much better than the other silent speech models on unseen data. We



(a)

(b)

(c)

(d)

(e)

(f)

Seen

Unseen

Figure 9: Performance evaluation of the three examined silent speech recognition models without/with the proposed repair model in terms of a) WER-Seen, b) WER-Unseen, c) WPM-Seen, d) WPM-Unseen, e) CT-Seen, and f) CT-Unseen. Each condition has 360 data points. Reported values are the average of all values. The values inside the brackets are standard deviations (SD). Error bars represent ±1 SD.

speculate that this is because Transformer is trained on LRS dataset that has a larger number of word-level classes (longer phrases). This resulted in a much lower WER for repaired Transformer with unseen data as it provided the language model with more accurate words than LipType.

Overall, empirical results exhibit the effectiveness of repair model on all recognition models for improving accuracy with a slight increase in CT. These findings show the potential of the developed framework as a medium for communication with various computer systems, incorporated in day-to-day usage. This approach could also enable people with speech disorder, muteness, and blindness to input and interact with computer systems, increasing their access to technologies. We also envision the potential of this framework on other platforms like head-mounted displays (HMDs) and smart eyewear.

8 CONCLUSION

We developed LipType, an optimized version of LipNet for improved speed and accuracy. We then developed an independent repair model that compensates for poor lighting conditions and corrects potential errors in output using a custom language model. We evaluated the repair model's effectiveness with both LipType and other speech and silent speech recognizers. Empirical results showed that it significantly reduces error rates for all recognizers. The findings confirm that the model can be used independently with a range of recognizers. In the future, we will extend this work to further optimize the algorithm to make it faster and adapt it for people with various speech disorders.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. (March 2016). http://arxiv.org/abs/1603.04467

[2] Mahmoud Afifi, Konstantinos G. Derpanis, Björn Ommer, and Michael S. Brown. 2020. Learning to Correct Overexposed and Underexposed Photos. (March 2020). http://arxiv.org/abs/2003.11596

[3] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. 2018. Deep Lip Reading: A Comparison of Models and an Online Application. (June 2018). http://arxiv.org/abs/1806.06053

[4] Abien Fred Agarap. 2019. Deep Learning using Rectified Linear Units (ReLU). (2019). http://arxiv.org/abs/1803.08375

[5] Alexandre Allauzen. 2007. Error Detection in Confusion Network. In *INTERSPEECH*.

[6] Ibrahim Almajai, Stephen Cox, Richard Harvey, and Yuxuan Lan. 2016. Improved Speaker Independent Lip Reading Using Speaker Adaptive Training and Deep Neural Networks. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2722–2726. https://doi.org/10.1109/ICASSP.2016.7472172 ISSN: 2379-190X.

[7] Tarik Arici, Salih Dikbas, and Yucel Altunbasak. 2009. A Histogram Modification Framework and Its Application for Image Contrast Enhancement. *IEEE Transactions on Image Processing* 18, 9 (Sept. 2009), 1921–1935. https://doi.org/10.1109/TIP.2009.2021548 Conference Name: IEEE Transactions on Image Processing.

[8] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2009. Analysis of Text Entry Performance Metrics. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. 100–105. https://doi.org/10.1109/TIC-STH.2009.5444533

[9] Yannis M. Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas. 2016. Lipnet: End-to-End Sentence-Level Lipreading. (Dec. 2016). http://arxiv.org/abs/1611.01599

[10] Jess Bartels, D. Andreasen, P. Ehirim, Hui Mao, and P. Kennedy. 2008. Neurotrophic Electrode: Method of Assembly and Implantation into Human Motor Speech Cortex. *Journal of Neuroscience Methods* (2008). https://doi.org/10.1016/j.jneumeth.2008.06.030

[11] Youssef Bassil and Paul Semaan. 2012. Asr Context-Sensitive Error Correction Based on Microsoft N-Gram Dataset. (March 2012). http://arxiv.org/abs/1203.5262

[12] Helen L. Bear and Richard Harvey. 2019. Alternative Visual Units for an Optimized Phoneme-Based Lipreading System. 18 (2019), 3870. https://doi.org/10.3390/app9183870

[13] Jonathan S. Brumberg, Alfonso Nieto-Castanon, Philip R. Kennedy, and Frank H. Guenther. 2010. Brain-Computer Interfaces for Speech Communication. *Speech Communication* 52, 4 (April 2010), 367–379. https://doi.org/10.1016/j.specom.2010.01.001

[14] Turgay Celik and Tardi Tjahjadi. 2011. Contextual and Variational Contrast Enhancement. *IEEE Transactions on Image Processing* 20, 12 (Dec. 2011), 3431–3441. https://doi.org/10.1109/TIP.2011.2157513 Conference Name: IEEE Transactions on Image Processing.

[15] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. 2018. Learning to See in the Dark. (May 2018). http://arxiv.org/abs/1805.01934

[16] Stanley F. Chen and Joshua Goodman. 1996. An Empirical Study of Smoothing Techniques for Language Modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics (ACL '96)*. Association for Computational Linguistics, USA, 310–318. https://doi.org/10.3115/981863.981904

[17] Wei Chen, Sankaranarayanan Ananthakrishnan, Rohit Kumar, Rohit Prasad, and Prem Natarajan. 2013. Asr Error Detection in a Conversational Spoken Language Translation System. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 7418–7422. https://doi.org/10.1109/ICASSP.2013.6639104 ISSN: 2379-190X.

[18] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. (2014). http://arxiv.org/abs/1412.3555

[19] Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. 2017. Lip reading sentences in the wild. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 3444–3453.

[20] Joon Son Chung and Andrew Zisserman. 2016. Out of Time: Automated Lip Sync in the Wild. In *ACCV Workshops*. https://doi.org/10.1007/978-3-319-54427-4_19

[21] Joon Son Chung and Andrew Zisserman. 2017. Lip Reading in Profile. In *BMVC*. https://doi.org/10.5244/C.31.155

[22] Joon Son Chung and Andrew Zisserman. 2017. Lip Reading in the Wild. In *Computer Vision – ACCV 2016 (Lecture Notes in Computer Science)*, Shang-Hong Lai, Vincent Lepetit, Ko Nishino, and Yoichi Sato (Eds.). Springer International Publishing, Cham, 87–103. https://doi.org/10.1007/978-3-319-54184-6_6

[23] Joon Son Chung and Andrew Zisserman. 2018. Learning to Lip Read Words by Watching Videos. *Computer Vision and Image Understanding* 173 (Aug. 2018), 76–85. https://doi.org/10.1016/j.cviu.2018.02.001

[24] C. Cieri, D. Miller, and K. Walker. 2004. The Fisher Corpus: A Resource for the Next Generations of Speech-to-Text. In *LREC*.

[25] Ronan Collobert, Awni Hannun, and Gabriel Synnaeve. 2019. A Fully Differentiable Beam Search Decoder. (2019). http://arxiv.org/abs/1902.06022

[26] Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. 2016. Wav2letter: An End-to-End Convnet-Based Speech Recognition System. (Sept. 2016). http://arxiv.org/abs/1609.03193

[27] Martin Cooke, Jon Barker, Stuart Cunningham, and Xu Shao. 2006. An Audio-Visual Corpus for Speech Perception and Automatic Speech Recognition. *The Journal of the Acoustical Society of America* 120, 5 (Oct. 2006), 2421–2424. https://doi.org/10.1121/1.2229005 Publisher: Acoustical Society of America.

[28] Charles S. DaSalla, Hiroyuki Kambara, Yasuharu Koike, and Makoto Sato. 2009. Spatial Filtering and Single-Trial Classification of Eeg During Vowel Speech Imagery. In *Proceedings of the 3rd International Convention on Rehabilitation Engineering & Assistive Technology (i-CREATe '09)*. Association for Computing Machinery, New York, NY, USA, 1–4. https://doi.org/10.1145/1592700.1592731

[29] B. Denby, Y. Oussar, G. Dreyfus, and M. Stone. 2006. Prospects for a Silent Speech Interface Using Ultrasound Imaging. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, Vol. 1. I–I. https://doi.org/10.1109/ICASSP.2006.1660033 ISSN: 2379-190X.

[30] B. Denby and M. Stone. 2004. Speech Synthesis from Real Time Ultrasound Images of the Tongue. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1. I–685. https://doi.org/10.1109/ICASSP.2004.1326078 ISSN: 1520-6149.

[31] Ali Diba, Mohsen Fayyaz, Vivek Sharma, M. Mahdi Arzani, Rahman Yousefzadeh, Juergen Gall, and Luc Van Gool. 2019. Spatio-Temporal Channel Correlation Networks for Action Classification. (Feb. 2019). http://arxiv.org/abs/1806.07754

[32] Xuan Dong, Guan Wang, Yi Pang, Weixin Li, Jiangtao Wen, Wei Meng, and Yao Lu. 2011. Fast Efficient Algorithm for Enhancement of Low Lighting Video. In *2011 IEEE International Conference on Multimedia and Expo*. 1–6. https://doi.org/10.1109/ICME.2011.6012107 ISSN: 1945-788X.

[33] M. J. Fagan, S. R. Ell, J. M. Gilbert, E. Sarrazin, and P. M. Chapman. 2008. Development of a (silent) Speech Recognition System for Patients Following Laryngectomy. *Medical Engineering & Physics* 30, 4 (May 2008), 419–425. https://doi.org/10.1016/j.medengphy.2007.05.003

[34] Xue Feng, Yaodong Zhang, and James Glass. 2014. Speech Feature Denoising and Dereverberation Via Deep Autoencoders for Noisy Reverberant Speech Recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1759–1763. https://doi.org/10.1109/ICASSP.2014.6853900 ISSN: 2379-190X.

[35] Victoria M. Florescu, L. Crevier-Buchman, B. Denby, T. Hueber, Antonia Colazo-Simon, Claire Pillot-Loiseau, P. Roussel-Ragot, C. Gendrot, and S. Quattrocchi. 2010. Silent Vs Vocalized Articulation for a Portable Ultrasound-Based Silent Speech Interface. In *INTERSPEECH*.

[36] Xueyang Fu, Delu Zeng, Yue Huang, Xiao-Ping Zhang, and Xinghao Ding. 2016. A Weighted Variational Model for Simultaneous Reflectance and Illumination Estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2782–2790. https://doi.org/10.1109/CVPR.2016.304 ISSN: 1063-6919.

[37] Yohei Fusayasu, Katsuyuki Tanaka, Tetsuya Takiguchi, and Yasuo Ariki. 2015. Word-Error Correction of Continuous Speech Recognition Based on Normalized Relevance Distance. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*. AAAI Press, Buenos Aires, Argentina, 1257–1262.

[38] James M. Gilbert, Sergey I. Rybchenko, Robin Hofe, Stephen R. Ell, Michael J. Fagan, Roger K. Moore, and Phil D. Green. 2010. Isolated Word Recognition of Silent Speech Using Magnetic Implants and Sensors. *Medical engineering & physics* 10 (2010), 1189–1197. https://doi.org/10.1016/j.medengphy.2010.08.011

[39] J.J. Godfrey, E.C. Holliman, and J. McDaniel. 1992. Switchboard: Telephone Speech Corpus for Research and Development. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1. 517–520 vol.1. https://doi.org/10.1109/ICASSP.1992.225858 ISSN: 1520-6149.

[40] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of the 23rd international conference on Machine learning* (Pittsburgh, Pennsylvania, USA, 2006-06-25) *(ICML '06)*. Association for Computing Machinery, 369–376. https://doi.org/10.1145/1143844.1143891

[41] Alex Graves and Navdeep Jaitly. 2014. Towards End-to-End Speech Recognition with Recurrent Neural Networks. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32 (ICML'14)*. JMLR.org, Beijing, China, II–1764–II–1772.

[42] Xiaojie Guo, Yu Li, and Haibin Ling. 2017. Lime: Low-Light Image Enhancement Via Illumination Map Estimation. *IEEE Transactions on Image Processing* 26, 2 (Feb. 2017), 982–993. https://doi.org/10.1109/TIP.2016.2639450 Conference Name: IEEE Transactions on Image Processing.

[43] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. 2014. Deep Speech: Scaling up End-to-End Speech Recognition. (Dec. 2014). http://arxiv.org/abs/1412.5567

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. https://doi.org/10.1109/CVPR.2016.90 ISSN: 1063-6919.

[45] Kenneth Heafield, Ivan Pouzyrevsky, J. Clark, and Philipp Koehn. 2013. Scalable Modified Kneser-Ney Language Model Estimation. In *ACL*.

[46] Panikos Heracleous and Norihiro Hagita. 2011. Automatic Recognition of Speech Without Any Audio Information. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2392–2395. https://doi.org/10.1109/ICASSP.2011.5946965 ISSN: 2379-190X.

[47] Panikos Heracleous, Tomomi Kaino, H. Saruwatari, and K. Shikano. 2007. Unvoiced Speech Recognition Using Tissue-Conductive Acoustic Sensor. *EURASIP J. Adv. Signal Process.* (2007). https://doi.org/10.1155/2007/94068

[48] Tatsuya Hirahara, Makoto Otani, Shota Shimizu, Tomoki Toda, Keigo Nakamura, Yoshitaka Nakajima, and Kiyohiro Shikano. 2010. Silent-Speech Enhancement

Using Body-Conducted Vocal-Tract Resonance Signals. *Speech Communication* 52, 4 (April 2010), 301–313. https://doi.org/10.1016/j.specom.2009.12.001

[49] Alain Horé and Djemel Ziou. 2010. Image Quality Metrics: Psnr Vs. Ssim. In *2010 20th International Conference on Pattern Recognition.* 2366–2369. https://doi.org/10.1109/ICPR.2010.579 ISSN: 1051-4651.

[50] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. 2019. Squeeze-and-Excitation Networks. (May 2019). http://arxiv.org/abs/1709.01507

[51] T. Hueber, G. Aversano, G. Chollet, B. Denby, G. Dreyfus, Y. Oussar, P. Roussel, and M. Stone. 2007. Eigentongue Feature Extraction for an Ultrasound-Based Silent Speech Interface. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, Vol. 1. I–1245–I–1248. https://doi.org/10.1109/ICASSP.2007.366140 ISSN: 2379-190X.

[52] Thomas Hueber, Elie-Laurent Benaroya, Gérard Chollet, Bruce Denby, Gérard Dreyfus, and Maureen Stone. 2010. Development of a Silent Speech Interface Driven by Ultrasound and Optical Images of the Tongue and Lips. *Speech Communication* 52, 4 (April 2010), 288–300. https://doi.org/10.1016/j.specom.2009.11.004

[53] Thomas Hueber, Elie-Laurent Benaroya, Gérard Chollet, Bruce Denby, Gérard Dreyfus, and Maureen Stone. 2010. Development of a Silent Speech Interface Driven by Ultrasound and Optical Images of the Tongue and Lips. *Speech Communication* 52, 4 (April 2010), 288–300. https://doi.org/10.1016/j.specom.2009.11.004

[54] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* (Lille, France, 2015-07-06) *(ICML'15)*. JMLR.org, 448–456.

[55] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. [n.d.]. 3D Convolutional Neural Networks for Human Action Recognition. 35, 1 ([n. d.]), 221–231. https://doi.org/10.1109/TPAMI.2012.59 Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[56] D.J. Jobson, Z. Rahman, and G.A. Woodell. 1997. A Multiscale Retinex for Bridging the Gap Between Color Images and the Human Observation of Scenes. *IEEE Transactions on Image Processing* 6, 7 (July 1997), 965–976. https://doi.org/10.1109/83.597272 Conference Name: IEEE Transactions on Image Processing.

[57] Charles Jorgensen and Sorin Dusan. 2010. Speech Interfaces Based Upon Surface Electromyography. *Speech Communication* 52, 4 (April 2010), 354–366. https://doi.org/10.1016/j.specom.2009.11.003

[58] C. Jorgensen, D.D. Lee, and S. Agabont. 2003. Sub Auditory Speech Recognition Based on Emg Signals. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, Vol. 4. 3128–3133 vol.4. https://doi.org/10.1109/IJCNN.2003.1224072 ISSN: 1098-7576.

[59] S. Jou, Tanja Schultz, Matthias Walliczek, F. Kraft, and Alexander H. Waibel. 2006. Towards Continuous Speech Recognition Using Surface Electromyography. In *INTERSPEECH.*

[60] Arnav Kapur, Shreyas Kapur, and Pattie Maes. 2018. Alterego: A Personalized Wearable Silent Speech Interface. In *23rd International Conference on Intelligent User Interfaces (IUI '18)*. Association for Computing Machinery, New York, NY, USA, 43–53. https://doi.org/10.1145/3172944.3172977

[61] Naoki Kimura, Michinari Kono, and Jun Rekimoto. 2019. SottoVoce: An Ultrasound Imaging-Based Silent Speech Interaction Using Deep Neural Networks. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3290605.3300376

[62] Davis E. King. 2009. Dlib-Ml: A Machine Learning Toolkit. *The Journal of Machine Learning Research* 10 (Dec. 2009), 1755–1758.

[63] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. (2017). http://arxiv.org/abs/1412.6980

[64] Oscar Koller, Hermann Ney, and Richard Bowden. 2015. Deep Learning of Mouth Shapes for Sign Language. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*. 477–483. https://doi.org/10.1109/ICCVW.2015.69

[65] E. H. Land and J. McCann. 1971. Lightness and Retinex Theory. *Journal of the Optical Society of America* (1971). https://doi.org/10.1364/JOSA.61.000001

[66] Chulwoo Lee, Chul Lee, and Chang-Su Kim. 2013. Contrast Enhancement Based on Layered Difference Representation of 2d Histograms. *IEEE Transactions on Image Processing* 22, 12 (Dec. 2013), 5372–5384. https://doi.org/10.1109/TIP.2013.2284059 Conference Name: IEEE Transactions on Image Processing.

[67] Chongyi Li, J. Guo, F. Porikli, and Y. Pang. 2018. Lightennet: A Convolutional Neural Network for Weakly Illuminated Image Enhancement. *Pattern Recognit. Lett.* (2018). https://doi.org/10.1016/j.patrec.2018.01.010

[68] Yuan Liang, Koji Iwano, and Koichi Shinoda. 2014. An Efficient Error Correction Interface for Speech Recognition on Mobile Touchscreen Devices. In *2014 IEEE Spoken Language Technology Workshop (SLT)*. 454–459. https://doi.org/10.1109/SLT.2014.7078617

[69] X. Lu, Y. Tsao, S. Matsuda, and C. Hori. 2013. Speech Enhancement Based on Deep Denoising Autoencoder. In *INTERSPEECH.*

[70] Andrew L. Maas, Ziang Xie, Dan Jurafsky, and A. Ng. 2015. Lexicon-Free Conversational Speech Recognition with Neural Networks. In *HLT-NAACL.* https://doi.org/10.3115/v1/N15-1038

[71] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. Association for Computing Machinery, New York, NY, USA, 754–755. https://doi.org/10.1145/765891.765971

[72] L. Maier-Hein, F. Metze, T. Schultz, and A. Waibel. 2005. Session Independent Non-Audible Speech Recognition Using Surface Electromyography. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.* 331–336. https://doi.org/10.1109/ASRU.2005.1566521

[73] Y. Nakajima, H. Kashioka, K. Shikano, and N. Campbell. 2003. Non-Audible Murmur Recognition Input Interface Using Stethoscopic Microphone Attached to the Skin. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, Vol. 5. V–708. https://doi.org/10.1109/ICASSP.2003.1200069 ISSN: 1520-6149.

[74] L.C. Ng, G.C. Burnett, J.F. Holzrichter, and T.J. Gable. 2000. Denoising of Human Speech Using Combined Acoustic and Em Sensor Signal Processing. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, Vol. 1. 229–232 vol.1. https://doi.org/10.1109/ICASSP.2000.861925 ISSN: 1520-6149.

[75] K. Noda, Y. Yamaguchi, K. Nakadai, H. Okuno, and Tetsuya Ogata. 2014. Lipreading Using Convolutional Neural Network. In *INTERSPEECH.*

[76] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An Asr Corpus Based on Public Domain Audio Books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 5206–5210. https://doi.org/10.1109/ICASSP.2015.7178964 ISSN: 2379-190X.

[77] Sanjay A. Patil and John H. L. Hansen. 2010. The Physiological Microphone (pmic): A Competitive Alternative for Speaker Assessment in Stress Detection and Speaker Verification. *Speech Communication* 52, 4 (April 2010), 327–340. https://doi.org/10.1016/j.specom.2009.11.006

[78] Douglas B. Paul and Janet M. Baker. 1992. The Design for the Wall Street Journal-Based Csr Corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992.* https://www.aclweb.org/anthology/H92-1073

[79] Thomas Pellegrini and Isabel Trancoso. 2009. Error Detection in Broadcast News Asr Using Markov Chains. In *Proceedings of the 4th conference on Human language technology: challenges for computer science and linguistics (LTC'09)*. Springer-Verlag, Berlin, Heidelberg, 59–69.

[80] T. Pellegrini and I. Trancoso. 2010. Improving Asr Error Detection with Non-Decoder Based Features. In *INTERSPEECH.*

[81] Stavros Petridis and Maja Pantic. 2016. Deep Complementary Bottleneck Features for Visual Speech Recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2304–2308. https://doi.org/10.1109/ICASSP.2016.7472088 ISSN: 2379-190X.

[82] Stavros Petridis and Maja Pantic. 2016. Deep Complementary Bottleneck Features for Visual Speech Recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2304–2308. https://doi.org/10.1109/ICASSP.2016.7472088 ISSN: 2379-190X.

[83] Anne Porbadnigk, Marek Wester, Jan Calliess, and Tanja Schultz. 2009. EEG-based Speech Recognition - Impact of Temporal Effects. In *BIOSIGNALS.* https://doi.org/10.5220/0001554303760381

[84] Anne Porbadnigk, Marek Wester, Jan Calliess, and Tanja Schultz. 2009. EEG-Based Speech Recognition - Impact of Temporal Effects. In *BIOSIGNALS.* https://doi.org/10.5220/0001554303760381

[85] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The Kaldi Speech Recognition Toolkit. https://infoscience.epfl.ch/record/192584 Conference Name: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding Number: CONF Publisher: IEEE Signal Processing Society.

[86] T.F. Quatieri, K. Brady, D. Messing, J.P. Campbell, W.M. Campbell, M.S. Brandstein, C.J. Weinstein, J.D. Tardelli, and P.D. Gatewood. 2006. Exploiting Nonacoustic Sensors for Speech Encoding. *IEEE Transactions on Audio, Speech, and Language Processing* 14, 2 (March 2006), 533–544. https://doi.org/10.1109/TSA.2005.855838 Conference Name: IEEE Transactions on Audio, Speech, and Language Processing.

[87] Michael Riley, Cyril Allauzen, and Martin Jansche. 2009. Openfst: An Open-Source, Weighted Finite-State Transducer Library and Its Applications to Speech and Language. In *Proceedings of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL HLT) 2009 conference, Tutorials.* http://aclweb.org/anthology/N09-4005

[88] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 2013. 300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge. In *2013 IEEE International Conference on Computer Vision Workshops.* 397–403. https://doi.org/10.1109/ICCVW.2013.59

[89] Arup Sarma and David D. Palmer. 2004. Context-Based Speech Recognition Error Detection and Correction. In *Proceedings of HLT-NAACL 2004: Short Papers (HLT-NAACL-Short '04)*. Association for Computational Linguistics, USA, 85–88.

[90] Tanja Schultz and Michael Wand. 2010. Modeling Coarticulation in Emg-Based Continuous Speech Recognition. *Speech Communication* 52, 4 (April 2010),

341–353. https://doi.org/10.1016/j.specom.2009.12.002

[91] A.R. Setlur, R.A. Sukkar, and J. Jacob. 1996. Correcting Recognition Errors Via Discriminative Utterance Verification. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*, Vol. 2. 602–605 vol.2. https://doi.org/10.1109/ICSLP.1996.607433

[92] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. 15, 1 (2014), 1929–1958.

[93] Themos Stafylakis and Georgios Tzimiropoulos. 2017. Combining Residual Networks with Lstms for Lipreading. *INTERSPEECH* (2017). https://doi.org/10.21437/INTERSPEECH.2017-85

[94] Ke Sun, Chun Yu, Weinan Shi, Lan Liu, and Yuanchun Shi. 2018. Lip-Interact: Improving Mobile Device Interaction with Silent Speech Commands. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. Association for Computing Machinery, New York, NY, USA, 581–593. https://doi.org/10.1145/3242587.3242599

[95] P. Suppes, B. Han, and Z. Lu. 1998. Brain-Wave Recognition of Sentences. *Proceedings of the National Academy of Sciences of the United States of America* (1998). https://doi.org/10.1073/pnas.95.26.15861

[96] P. Suppes, Z. Lu, and B. Han. 1997. Brain Wave Recognition of Words. *Proceedings of the National Academy of Sciences of the United States of America* (1997). https://doi.org/10.1073/pnas.94.26.14965

[97] Satoshi Tamura, Hiroshi Ninomiya, Norihide Kitaoka, Shin Osuga, Yurie Iribe, Kazuya Takeda, and Satoru Hayamizu. 2015. Audio-Visual Speech Recognition Using Deep Bottleneck Features and High-Performance Lipreading. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. 575–582. https://doi.org/10.1109/APSIPA.2015.7415335

[98] Ingo R. Titze, Brad H. Story, Gregory C. Burnett, John F. Holzrichter, Lawrence C. Ng, and Wayne A. Lea. 1999. Comparison Between Electroglottography and Electromagnetic Glottography. *The Journal of the Acoustical Society of America* 107, 1 (Dec. 1999), 581–588. https://doi.org/10.1121/1.428324 Publisher: Acoustical Society of America.

[99] Naoya Ukai, Takumi Seko, Satoshi Tamura, and Satoru Hayamizu. 2012. Gif-Lr: GA-Based Informative Feature for Lipreading. In *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*. 1–4.

[100] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. (Dec. 2017). http://arxiv.org/abs/1706.03762

[101] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proceedings of the 25th international conference on Machine learning (ICML '08)*. Association for Computing Machinery, New York, NY, USA, 1096–1103. https://doi.org/10.1145/1390156.1390294

[102] Michael Wand, Jan Koutník, and Jürgen Schmidhuber. 2016. Lipreading with Long Short-Term Memory. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 6115–6119. https://doi.org/10.1109/ICASSP.2016.7472852 ISSN: 2379-190X.

[103] Michael Wand and Tanja Schultz. 2011. Session-Independent Emg-Based Speech Recognition. In *BIOSIGNALS*. https://doi.org/10.5220/0003169702950300

[104] Shuhang Wang, Jin Zheng, Hai-Miao Hu, and Bo Li. 2013. Naturalness Preserved Enhancement Algorithm for Non-Uniform Illumination Images. *IEEE Transactions on Image Processing* 22, 9 (Sept. 2013), 3538–3548. https://doi.org/10.1109/TIP.2013.2261309 Conference Name: IEEE Transactions on Image Processing.

[105] Wenjing Wang, Chen Wei, Wenhan Yang, and Jiaying Liu. 2018. Gladnet: Low-Light Enhancement Network with Global Awareness. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*. 751–755. https://doi.org/10.1109/FG.2018.00118

[106] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. 2018. Deep Retinex Decomposition for Low-Light Enhancement. (Aug. 2018). http://arxiv.org/abs/1808.04560

[107] Kai Xu, Dawei Li, Nick Cassimatis, and Xiaolong Wang. 2018. LCANet: End-to-end Lipreading with Cascaded Attention-CTC. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 548–555.

[108] Zhilu Zhang and Mert R. Sabuncu. 2018. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. (Nov. 2018). http://arxiv.org/abs/1805.07836

[109] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. 2018. Loss Functions for Neural Networks for Image Processing. (April 2018). http://arxiv.org/abs/1511.08861

[110] Lina Zhou, Yongmei Shi, Jinjuan Feng, and A. Sears. 2005. Data Mining for Detecting Errors in Dictation Speech Recognition. *IEEE Transactions on Speech and Audio Processing* 13, 5 (Sept. 2005), 681–688. https://doi.org/10.1109/TSA.2005.851874 Conference Name: IEEE Transactions on Speech and Audio Processing.

[111] Ziheng Zhou, Guoying Zhao, Xiaopeng Hong, and Matti Pietikäinen. 2014. A Review of Recent Advances in Visual Speech Decoding. *Image and Vision Computing* 32, 9 (Sept. 2014), 590–605. https://doi.org/10.1016/j.imavis.2014.06.004

A TEST DATASET: PRETRAINED MODEL

In this appendix, we provide details about the selected phrases for seen and unseen data. For seen data, we randomly selected 30 phrases from each pretrained models' training dataset. For unseen data, we randomly selected 30 phrases from MacKenzie and Soukoreff [71] dataset, which is common for all models.

A.1 LipNet: Seen data (Grid data [27])

(1) bin blue at c one again
(2) set blue in f four soon
(3) set blue with l eight now
(4) bin green by a four soon
(5) place blue by t nine soon
(6) place red with a four please
(7) place green by p five again
(8) lay red with k seven soon
(9) set blue in g four now
(10) set red with m zero soon
(11) bin white at q six soon
(12) place blue at n six now
(13) bin white with f seven soon
(14) place blue at m seven now
(15) bin red by j five please
(16) bin white at a one please
(17) set red by b one now
(18) place blue at i one soon
(19) place blue in n two please
(20) lay red by d seven please
(21) bin white by z three now
(22) place white with e three again
(23) bin red in j four now
(24) set blue in e four again
(25) lay green at p four again
(26) bin red with z eight please
(27) place red with n two please
(28) lay blue with b two please
(29) set green with v eight now
(30) bin white at j nine now

A.2 LipType: Seen data (Grid data [27])

(1) set green at f four soon
(2) bin green by h zero please
(3) bin white at f zero again
(4) place blue with j five again
(5) place white in g two please
(6) bin white by d eight again
(7) bin blue in q seven please
(8) lay red with f zero again
(9) place white at p eight now
(10) lay red with k three now
(11) lay red in j one soon
(12) lay white at j nine soon
(13) lay red at v eight again
(14) place green in u zero now
(15) lay red with c eight again

(16) place green at u two now
(17) place white by v four now
(18) bin red in x one now
(19) bin green at e zero again
(20) lay white by p six please
(21) bin red with x nine again
(22) place red at c three now
(23) set green at o seven please
(24) bin red at s eight again
(25) place red in s three please
(26) bin green by n four again
(27) place green by y two please
(28) place green by k one please
(29) lay blue at c one please
(30) place red by n one please

A.3 Transformer: Seen data (LRS data [3])

(1) the whole gardens are extraordinary and
(2) like hundreds of thousands of people do every year
(3) but now there is more protection
(4) we have a lot less atmosphere above us
(5) and a couple of weeks ago
(6) enjoy the summer
(7) are they relatives of yours
(8) no longer dependent on the sun
(9) but the waldorf astoria
(10) they would be able to go back
(11) not just a hotel
(12) not just in this town
(13) every september this place would be transformed into what
(14) now they are gathering
(15) so from his vantage point
(16) there is no air so there is no sound
(17) with one of the rooms upstairs
(18) maybe more of steel and iron
(19) we have run out of time
(20) before we all get too excited about that prospect
(21) it can be quite expensive
(22) in the form of a dessert plate
(23) on the face of it
(24) it could be your passport to a small fortune
(25) some issues with potential damp
(26) a great place for him to be
(27) we have to pay for that
(28) so rather than just relying on this information
(29) all of the brain is combining all the different senses
(30) he ordered them back inside

A.4 DeepSpeech: Seen data (Fisher English-conversational [24])

(1) can you hear me okay by the way
(2) oh good as long as you can hear me
(3) yeah i can hear you
(4) yeah that would be interesting
(5) like ten minutes with a head set on i might as well exercise
(6) yeah thats great
(7) listening to the music anyway so um
(8) i actually think its actually going out
(9) fifth wheel dating show
(10) i also watch that show the fifth wheel third and fourth wheel
(11) and i have seen i remember when survivor first started
(12) i saw that like a couple things
(13) cause my roommate where watching it
(14) yeah my roommates are you in college too
(15) i am in graduate school
(16) oh yeah okay i just graduated from um
(17) first time graduate last year
(18) and how about what school are you in
(19) that was great performance tonight
(20) it would be it would be cool to be on it
(21) thats very cool
(22) popular everyone talks about it
(23) somebody from my high school one something too
(24) he won he was like on that
(25) greatest bachelor show
(26) it was before these millionaire the millionaire guy ones
(27) it was like a pageant for men
(28) i didnt see it but i think i know what you were talking about
(29) yeah he was in my old high school
(30) going to rat on the other one

A.5 Kaldi: Seen data (LIBRISPEECH audiobooks [76])

(1) he was in a mood for music was he not
(2) give not so earnest a mind to these mummeries child
(3) a golden fortune and a happy life
(4) he was like my father in a way and yet was not my father
(5) also there was a stripling page who turned into a maid
(6) this was so sweet a lady sir and in some manner i do think
(7) but then the picture was gone as quickly as it came
(8) sister nell do you hear these marvels
(9) take your place and let us see what the crystal can show you
(10) like as not young master though i am an old man
(11) he was going home after victory
(12) it was almost buried now in flowers and foliage
(13) But I wrestled with this fellow
(14) but he saw nothing that moved no signal lights twinkled
(15) and why should that disturb me let him enter
(16) there was not a single note of gloom
(17) boats put out both from the fort and the shore
(18) his excellency madam the prefect
(19) so i did push this fellow
(20) what do i care for food
(21) shame on you citizens cried he i blush for my fellows
(22) surely we can submit with good grace
(23) fine for you to talk old man answered the lean
(24) at the same time every avenue of the throne was assaulted
(25) vintage years have much to do with the quality of wines
(26) come to me men here here he raised his voice still louder
(27) dry and of magnificent bouquet
(28) pour mayonnaise over all chill and serve
(29) set into a cold place to chill and become firm
(30) when thickened strain and cool

A.6 Wave2Letter: Seen data (LIBRISPEECH audiobooks [76])

(1) last two days of the voyage bartley found almost intolerable
(2) i never dreamed it would be you bartley
(3) the cuisine is the best and the chefs rank at the top of the art
(4) he pulled up a window as if the air were heavy
(5) it it hasnt always made you miserable has it
(6) always but its worse now
(7) it's unbearable it tortures me every minute
(8) i get nothing but misery out of either
(9) there is this deception between me and everything
(10) he dropped back heavily into his chair by the fire
(11) i have thought about it until i am worn out
(12) after the very first
(13) we never planned to meet and when we met
(14) i dont know what becomes of the ladies
(15) but now it doesnt seem to matter very much
(16) presently it stole back to his coat sleeve
(17) yes hilda i know that he said simply
(18) i understand bartley i was wrong
(19) season with salt and pepper and a little sugar to taste
(20) you want me to say it she whispered
(21) what alternative was there for her
(22) its got to be a clean break hilda
(23) oh bartley what am i to do
(24) you ask me to stay away from you because you want me
(25) i will ask the least imaginable but i must have something
(26) you see the treatment is a trifle fanciful
(27) he protected her and she strengthened him
(28) and then you came back not caring very much
(29) dont cry dont cry he whispered
(30) a little attack of nerves possibly

A.7 Common unseen data (MacKenzie and Soukoreff dataset [71])

(1) my watch fell in the water
(2) prevailing wind from the east
(3) never too rich and never too thin
(4) breathing is difficult
(5) I can see the rings on Saturn
(6) physics and chemistry are hard
(7) my bank account is overdrawn
(8) elections bring out the best
(9) you are a wonderful example
(10) do not squander your time
(11) do not drink too much
(12) take a coffee break
(13) popularity is desired by all
(14) the music is better than it sounds
(15) I agree with you
(16) do not say anything
(17) play it again Sam
(18) the force is with you
(19) we went grocery shopping
(20) the assignment is due today
(21) what you see is what you get
(22) for your information only
(23) a quarter of a century
(24) the store will close at ten
(25) head shoulders knees and toes
(26) always cover all the bases
(27) this is a very good idea
(28) can we play cards tonight
(29) get rid of that immediately
(30) public transit is much faster

B ABLATION STUDIES

In this appendix, we present the results of various ablation studies performed to demonstrate the contribution of each submodule of our model to the overall performance gains.

B.1 With only Pre-processing

The purpose of this study was to analyze the effects of pre-processing on silent speech recognition model's performance in terms of WER, WPM, CT. For evaluation, we considered all pre-trained models as baselines and compared with their conjunction with pre-processing. Results revealed that the proposed pre-processing module substantially reduced the error rates of all pre-trained models (Table 6). In the study, pre-processing with LipNet showed 15% reduction in WER with seen and 7% reduction with unseen data. With LipType, it showed 12% reduction in WER with seen and 5.5% reduction with unseen data. With Transformer, it showed 24% reduction in WER with seen and 8% reduction with unseen data. On average, for all models, there were 5% reduction in WPM and 2 sec. increase in CT with both seen and unseen data. Note that the performance of these models with pre-processing and post-processing (repaired) are shown in Fig. 9.

B.2 Effects of Individual Error Correction Module

We also analyzed the effects of individual error correction modules with the LipType model in terms of WER and CT. All the presented results are calculated with seen data. Results demonstrated that each submodule made a significant contribution to the overall performance improvement of the repair model (Table 7).

B.3 Correction Classification

In this study, we analyzed the types of correction made by the post-processing module. For this, we classified all errors by the following criteria:

- Whether the correct word is substituted with other word(s), substitution error.
- Whether the new word(s) is inserted, insertion error.
- Whether the correct word(s) is deleted, deletion error.

After analysis, we observed that Silent Speech has 2% insertion, 27% deletion, 71% substitution, (34% of these were on short words <= 3 chars, 11% of these were on long words > 3 chars, 29% of these were in starting of the phrase <= length(phrase)/2-1, 14% of these were in ending of the phrase > length(phrase)/2). However, speech has 38% insertion, 21% deletion, 41% substitution (12% of these were on short words <= 3 chars, 18% of these were on long words > 3 chars, 25% of these were in starting of the phrase <=

Model	Seen			Unseen		
	WER	WPM	CT	WER	WPM	CT
LipNet	49.4	4.9	14.3	96.5	4.8	14.1
PP + LipNet	42.0	4.8	16.4	89.5	4.6	15.9
LipType	45.9	6.5	6.0	94.1	6.2	6.2
PP + LipType	40.9	5.6	8.5	88.9	6.0	8.1
Transformer	56.2	6.0	14.9	82.4	5.9	14.5
PP + Transfomer	42.5	5.9	16.4	76.0	5.6	15.8

Table 6: Performance evaluation of the three examined silent speech recognition models without/with the Pre-processing (PP) module in terms of WER, WPM, and CT for seen and unseen data.

Method	WER	CT
LipType	45.9	6.0
PP + LipType	40.9	8.3
PP + LipType + DDA	29.7	11.1
PP + LipType + DDA + SC	27.5	11.7
PP + LipType + DDA + SC + LM	24.1	14.2
PP + LipType + DDA + SC + LM + ED	20.5	15.1

Table 7: Effect of individual error correction module on LipType's WER and CT with seen data (Pre-processing: PP; DDA: Deep denoising autoencoder; SC: Spell Checker; LM: Language Model; ED: Edit Distance). We considered DDA + SC + LM + ED as the post-processing module.

length(phrase)/2-1, 11% of these were in ending of the phrase > length(phrase)/2).

Silent speech has 94.7% fewer insertion errors than speech. We speculate that this is because, for speech input, the recognition model captures background noises and recognizes them as words which resulted in more insertion errors. Unlike speech recognition, silent speech recognition just uses visual information for recognition which does not get affected by background noise. Besides, silent speech has 73.1% more substitution errors. We hypothesize that this is because it is more difficult to distinguish between homophones with just visual information due to ambiguity in lip movements, i.e., different characters that produce exactly the same lip sequence (e.g. 'p' and 'b'). This may have resulted in more substituted words.