Contour Mapping for Speaker-Independent Lip Reading System

Souheil Fenghour[1], Daqing Chen[2], Perry Xiao[3]
London South Bank University, London, UK[1]

ABSTRACT

In this paper, we demonstrate how an existing deep learning architecture for automatically lip reading individuals can be adapted it so that it can be made speaker independent, and by doing so, improved accuracies can be achieved on a variety of different speakers. The architecture itself is multi-layered consisting of a convolutional neural network, but if we are to apply an initial edge detection-based stage to pre-process the image inputs so that only the contours are required, the architecture can be made to be less speaker favourable.

The neural network architecture achieves good accuracy rates when trained and tested on some of the same speakers in the ”overlapped speakers” phase of simulations, where word error rates of just 1.3% and 0.4% are achieved when applied to two individual speakers respectively, as well as character error rates of 0.6% and 0.3%. The ”unseen speakers” phase fails to achieve as good an accuracy, with greater recorded word error rates of 20.6% and 17.0% when tested on the two speakers with character error rates of 11.5% and 8.3%.

The variation in size and colour of different people’s lips will result in different outputs at the convolution layer of a convolutional neural network as the output depends on the pixel intensity of the red, green and blue channels of an input image so a convolutional neural network will naturally favour the observations of the individual whom the network was tested on. This paper proposes an initial ”contour mapping stage” which makes all inputs uniform so that the system can be speaker independent.

Keywords: Lip Reading, Speech Recognition, Deep Learning, Facial Landmarks, Convolutional Neural Networks, Recurrent Neural Networks, Edge Detection, Contour Mapping

1. INTRODUCTION

Attempts that have been made to automate lip reading entail both deep learning and non-deep learning based methodologies. Non-deep learning related methods such as Hidden Markov Models [1] account for the majority of approaches; while there have been recent attempts to automate lip reading through the use of artificial neural networks including Garg et al [2], Wand et al [3], Chung and Zisserman’s works from 2016 [4] and 2017 [5], as well as LipNet’s [6] most recent attempt which is scrutinised in this paper.

Several obstacles to automated lip reading exist including the inability to distinguish between homoviseme words i.e. words with characters that produce the exact the same lip movements despite being intrinsically different and sounding different, as well as the insufficient supply of datasets that is needed to train effective models and datasets that we have available to us only cover a small range of topics with limited vocabulary [7]. There is also the issue that different people have different facial appearances, and individual people will have lip features that are unique to that person. For this reason, the focus of this paper is to elaborate on how automated lip reading can be tailored to become speaker independent [8].

LipNet made one of the most successful attempts at automating lip reading with a reported 95.2% accuracy achieved when recognising speech within 29,000 videos of people speaking in standard structured sentences [6] contained within the GRID corpus [8]. However, the performance can partly be attributed to the fact that the system was tested on videos consisting of the very same people it was tested on and LipNet does not perform as well when evaluated on different people to those used in the training phase. This paper addresses how a neural network architecture can be made to be speaker independent through the use of contour mapping.

2. METHODOLOGY

2.1 Review of feature extraction methods

There are a variety of feature extraction based methods used for lip reading. We will not delve into the explicit details behind all the different feature extraction methods because they are extensive and not the focus of this paper, though they can generally be divided into four categories. These include geometric-based methods, image-based methods, model-based methods and motion-based methods [9].

Geometric-based approaches require geometric information of the moving lips such as the width, height, area and perimeter to be extracted as features [10].

Image-based approaches will use either raw image pixel data as the feature or will have applied some form of transformation to the image pixel data to convert it to a form of spatial frequency to be used as the feature. The raw image pixel data will either be in colour consisting of 3 channels of pixels for red, green and blue components; or it will be grayscale and in black and white with 1 channel for intensity [11] [12] [13].

For model-based approaches, a model of visual speech ”articulators” such as the lip contours are constructed and the model's parameters are then used as the feature. Active-appearance models(AAM) [14] and active shape models(ASM) [15] are two examples of such approaches as well as a variety of edge detection based methods briefly talked about in Subsection 2.5.

Motion-based feature extraction methods are where descriptions of the motion observed during the uttering of lips get extracted as features. Optical flow is typically used to estimate the motion of the lips in images between two frames of speaking lips [16].

Deep learning based lip reading would fall into the categories of either image-based approaches when using pixel data from images of person's lips, or model-based approaches if one were to use lip contours as the input feature, and these two forms of feature extraction are the focus of this paper. The use of pixel data as the input of a neural network has the advantage of requiring less pre-processing because pixel data can form the direct input of a neural network, however, lip contours do give a better representation of spoken vocabulary given that a contour can be used to model vocabulary spoken by different speakers. The same cannot be said for pixel data because the varying appearances of different people's mouths creates a form a noise when a neural network architecture is trained on one speaker but evaluated on a different speaker.

2.2 Dataset

The GRID audio-visual corpus consists of 34 speakers with 1000 sentences [8] each following a standard sequence of verbs, then colours, followed by prepositions, alphabet, digits, and adverbs [8]. ”Bin blue at b eight now” is an example of one such sentence and each video consists of a different sentence.

The video data within the database consists of two parts; the first being videos of length 3 seconds that have been converted into image frames having been sampled at 25 frames per second (Figure 1 shows an example of a video sampled in to image frames), and the second being the subtitles consisting of words that are spoken at each time step. The subtitles are word transcriptions that will have been sub-sampled at 25 kHz and there will be subtitle files with three columns pertaining to the word, starting time and stopping time.

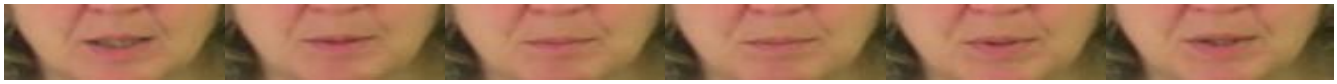

Figure 1. Image frames of a GRID speaker pronouncing the word ”bin”. This what our image frames would like in our training data

2.3 LipNet Architecture

The LipNet architecture (as shown in Figure 2) consists of a 3D Convolutional Neural Network(CNN) [17] in connection with a Gated Recurrent Neural Network(GRU) [18], while a connectionist temporal classification (CTC) loss mechanism [19] has been included for the purposes of word alignment. All layers within the architecture use rectified linear unit activation functions.

Videos are pre-processed into image frames(being sampled at a rate of 25 frames per second) and it is the image frames that form the input of the CNN. The CNN is used for word classification and has input dimensions in the form of

image height, image width and time. The stages of convolution, pooling, normalisation, and backpropagation are included, in addition to a drop-out feature intended to reduce overfitting [17].

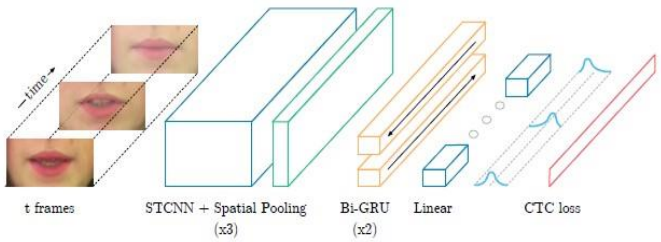

Figure 2. LipNet architecture taking in video images frames as an input followed by the 3D CNN, the GRU and the CTC loss mechanism [19]

The GRU's purpose in the architecture is to predict sentences by predicting the next word after one or more words. The GRU has a "Spell" architecture which contains sentence data that is required to create the context for it to function. Normally the sentence data would be subtitles found in the training videos [18].

Table 1 gives details of the input size of each layer in the second column and the last column gives details regarding the dimensions of each layer. C denotes the number of colour channels, F denotes the number of frames, W denotes the image width and H denotes the image height. D and F both denote the feature dimensions and the number of words present in the word vocabulary.

Table 1. The different layers of the LipNet Architecture

Layer	Input Size	Dimensions
1st Convolution	3 x 75 x 100 x 50	C x F x W x H
1st Normalisation	3 x 75 x 100 x 50	C x F x W x H
1st Pooling	32 x 75 x 50 x 25	C x F x W x H
2nd Convolution	32 x 75 x 25 x 12	C x F x W x H
2nd Normalisation	32 x 75 x 25 x 12	C x F x W x H
2nd Pooling	64 x 75 x 25 x 12	C x F x W x H
3rd Convolution	64 x 75 x 12 x 6	C x F x W x H
3rd Normalisation	64 x 75 x 12 x 6	C x F x W x H
3rd Pooling	96 x 75 x 12 x 6	C x F x W x H
1st GRU	75 x 512	F x D
2nd GRU	75 x 512	F x D
Softmax	75 x 28	F x V

Lip reading simulations have been performed using videos from the GRID [8] dataset and LipNet had previously applied their own architecture to two different scenarios. The first scenario was "overlapped-speakers", where a selection of videos from each GRID speaker were used for training and the remaining videos from each speaker would be used for the model evaluation phase, while the other scenario of "unseen speakers" had no overlap, i.e. videos used for training came from separate speakers to those used for evaluation.

For the "unseen speakers" scenario, LipNet was trained on video data from 29 of the GRID speakers, while video data belonging to four of the speakers [1, 2, 20, 22] were used for testing with remaining videos for speaker 21 missing. There are some videos from the 29 speakers that have been omitted because they are either corrupt or in a form that is not compatible with the LipNet architecture. The training dataset consists of 28775 videos with 3971 that have been used for testing [8] (as shown in Table 2).

When evaluating the LipNet architecture, we used a learning rate of $1 \times 10{-}4$ and a dropout rate of 0.5 while for every epoch of the simulation, videos were grouped into minibatches of 50. Weights are randomly set at the start of the

training phase and they are numerically corrected at every epoch using the learning rate. We have recreated the LipNet architecture in a Python programming language along with other relevant libraries that are necessary for the image processing.

Table 2. Train/Test Data

Situation	Speakers	Number of videos used
Training	29	28775
Testing	4	3971

2.4 Accuracy Metrics

The two metrics used for evaluating the accuracy of the LipNet architecture are character error rate(CER) and word error rate(WER). These are all commonly used metrics for evaluating the accuracy of speech recognition systems.

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

$$WAR = 1 - WER \tag{2}$$

In determining misclassifications, we have to compare the decoded speech to the actual speech with N being the total number of words in the actual speech, S being the number of substitutions made for wrong classifications, I being the insertions made for words not picked up and D being the number of deletions being made for decoded words that should not be present. The word error rate WER is defined as the ratio of incorrect words decoded to the total number of words in a sample(given by Eq. 1).

Character error rate CER is calculated the same way as WER except that characters are evaluated instead of words. Furthermore, the word accuracy rate WAR and character accuracy CAR can be calculated by subtracting the either error rate from the number 1 respectively according to Eq. 2.

Tables 3 and 4 give examples of how the character and word accuracies can be calculated. If we take the first pair of phrases in Table 3, 3 character substitutions are required to modify the phrase in Case 1 to make it identical to Case 2 whereby we would literally change ”in o” to ”at l”. Meanwhile, ”bin blue a x e again” requires a total of 6 changes including 1 substitution and 5 deletions for ”a x e” to be modified to ”at s three” and ”lay white at e zero please” requires 7 changes including 5 substitutions and 2 deletions for ”white at” to be modified to become ”red in”.

Table 3. Character error rates calculations for different phrases.

Case 1	Case 2	S	D	I	N	CAR(%)
bin blue in o six now	bin blue at l six now	3	0	0	3	85.8
bin blue in x one soon	bin blue at s one soon	3	0	0	3	86.4
bin blue a x e again	bin blue at s three again	1	0	5	6	76.0
lay white at e zero please	lay red in e zero please	5	2	0	7	70.8

Table 4 shows how word error rates would be calculated where all of phrases listed would involve direct word substitutions.

Table 4. Word error rates calculations for different phrases.

Case 1	Case 2	S	D	I	N	WAR(%)
bin blue in o six now	bin blue at l six now	2	0	0	6	66.7
bin blue in x one soon	bin blue at s one soon	2	0	0	6	66.7
bin blue a x e again	bin blue at s three again	3	0	0	6	50.0
lay white at e zero please	lay red in e zero please	2	0	0	6	66.7

2.5 Contour Mapping

Various spoken words and characters produce distinct lip movements and by spotting the lip movements, one can decode what has been said even if there is no audio present. However because different people have lips of different sizes, shapes and skin colour, this can cause ”noise” to be created during the convolution stages of a CNN as the network may be trained on one person speaking, but evaluated on a different person speaking. The convolution of different pixel values can affect the output.

Through the use of contour mapping, we can make different people’s lips uniform when inputted into the network as it is only the shape of the person’s lips that is needed for the speech to be decoded. Using the Pillow library in Python we can extract the contour of a person’s lips. In this paper we have constructed a single layer CNN to compare the outcome of utilising standard lip images and the images after the implementation of contour mapping.

A variety of edge detection methods can be used to locate the edge of objects contained within images and they are all based on calculating the change in pixel intensities or brightness including the Sobel operator [20], Roberts cross operator [21], Prewitt edge detector [22], Canny edge detector [23] and the ”ultrametric” segmentation algorithm developed by The Computer Vision Group at Berkeley [24] to name a few. For the sake of demonstrating the concept of contour mapping, one of the simplest edge detection algorithms available which is a kernel based function contained within the Pillow library of Python has been implemented.

The edge detection algorithm applies a kernel function G with components Gr for red pixels, Gg for green pixels and Gb for blue pixels(Equation 3). This function is applied to the image I(with components Ir, Ig, Ib that correspond to the red, green and blue pixel components respectively), and convoluted to produce output C(also with components Cr, Cg, Cb that correspond to the red, green and blue pixel components). These operations are shown in Equations 4 to 7.

The output will show us where the sharpest changes of pixel variation in an image lie. The output is a representation of pixel gradient or change in pixel magnitude over position, whereby the edges of an image will always show the largest change in pixel magnitude resulting in indices of large magnitude in the output matrix. However, regions of an image that have a consistent intensity with little variation in pixels values will produce indices of small or near zero magnitude in the output.

$$G_r = G_g = G_b = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \tag{3}$$

$$C_r = I_r * G_r \tag{4}$$

$$C_g = I_g * G_g \tag{5}$$

$$C_b = I_b * G_b \tag{6}$$

$$C[C_r, C_g, C_b] = I[I_b, I_b, I_b] * G[G_r, G_g, G_b] \tag{7}$$

We have extracted the central image frame of three different people pronouncing the letter “e” from the GRID dataset (speakers labelled s4, s5 and s6). One epoch of the CNN architecture will then be applied to the images for two different phases. Phase 1 deals with the natural images and Phase 2 deals with images after contour mapping has been applied. Figure 3 shows lip movement images in the order of s4, s5 and s6 with natural image next to is contoured image.

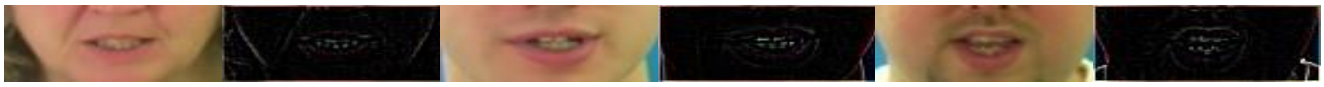

Figure 3. Lip images of the three speakers s4, s5 and s6 (left to right) in both their natural state and when contour mapping has been applied.

Figure 4 shows the CNN structure that we are using to test the concept of contour mapping. We will simulate the stages of convolution, normalisation, max pooling, dropout and softmax classification in one epoch to demonstrate that through contour mapping in Phase 2, the three classes are more probable for selection in the outcome which would allow us to develop a speaker independent lip reading system where the weights applied not favour any particular speaker. The CNN will be tested on speaker s4 but evaluated one all three speakers.

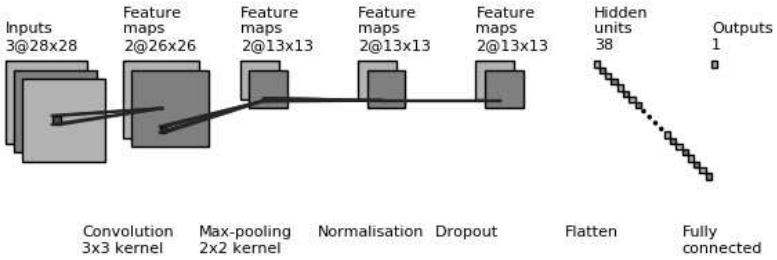

Figure 4. The 2D CNN was have constructed to demonstrate how a neural network can be made speaker independent for the purpose of lip reading.

During Phase 1, weights from natural lip images belonging to the first speaker s4 are used whereas for Phase 2, weights from the lip images after the implementation of contour mapping for the speaker s4 are used. We will not delve into the full architectural details of the CNN but a rectilinear unit function is used in the normalisation stage with a value of 0.9 for the dropout stage of the CNN and a softmax classifier σ(z) in the form of Equation 8. There are k classes with j pertaining to a particular class and zj being the outcome of the fully connected layer stage of the CNN for the class j.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{8}$$

3. RESULTS

The LipNet architecture was evaluated on speakers 1 and 2 from the GRID corpus applying both the ”overlapped Speakers” and ”unseen Speakers” weights respectively. Table 5 shows the WER and CER results for all these cases (evaluation sets 1 to 4). The LipNet achieves smaller error rates for ”overlapped speakers” than ”unseen speakers” as expected because the architecture when trained on the same person speaking would output a result that is more tailored to the individual.

Table 5. Results for average WER rates and CER rates of the LipNet architecture being evaluated on videos for both the ”overlapped-speakers” and ”unseen speakers” scenarios.

Evaluation Set	Weights	Spell Data	Speakers	WER	CER
Evaluation Set 1	Overlapped Speakers	GRID	Speaker 1 from GRID	1.3%	0.6%
Evaluation Set 2	Overlapped Speakers	GRID	Speaker 2 from GRID	0.4%	0.3%
Evaluation Set 3	Unseen Speakers	GRID	Speaker 1 from GRID	20.6%	11.5%
Evaluation Set 4	Unseen Speakers	GRID	Speaker 2 from GRID	17.0%	8.3%

If one were to alter the LipNet architecture to make it speaker independent where lip images would be pre-processed and treated uniformly by eliminating the influence of the pixels of people’s mouth appearances that vary from person to person; it would be possible to demonstrate that simulations would not favour that person it were tested on. Table 6 shows the outcomes of the CNN describe in section 2.4. The Prediction Score is an arbitrary column showing the outputs of fully connected layer stage for all three speakers, while the ”unnormalised probability” column shows the result of applying the exponential formula to the numbers in the previous column (the numerator of the softmax function), whereas the last column ”normalised probability” shows the output of the softmax function.

For Phase 1, we can see in the softmax stage that the speaker s4 was favoured with a near 100% probability because the network had been trained on s4. However for Phase 2 which made use of contour images and was still tested on

speaker s4, the probabilities are more evenly distributed and s4 in the testing phase was not even the most probable outcome. The probabilities being more evenly distributed in Phase 2 in comparison to Phase 1 demonstrates how we can make a neural network like LipNet less favourable towards the speaker it was trained on.

Table 6. Softmax results for the CNN applied three different speakers in Phases 1 and 2.

	Phase 1			Phase 2		
Label	Prediction Score	Unnormalised Probabilities	Normalised Probabilities(%)	Prediction Score	Unnormalised Probabilities	Normalised Probabilities(%)
s4	82.8	9.42×10^{35}	100.0	3.0	19.6	22.6
s5	74.4	2.02×10^{32}	0.0	3.5	34.0	39.4
s6	59.4	6.04×10^{25}	0.0	3.5	32.8	38.0
Total	216.6	9.42×10^{35}	100.0	10.0	86.4	100.0

Contour mapping is useful not only for the purposes of ensuring that the neural network does not favour any particular speakers, but it is also useful to reduce the effects of overfitting. If one is to compare the output probabilities of Phases 1 and 2, the relative predictions scores are a lot closer together in terms of magnitude. This results in the softmax probability outputs being more evenly distributed.

4. CONCLUSION

We have demonstrated how a neural network architecture for automated lip reading can be made speaker independent by not favouring the speaker which the architecture was trained on through the implementation of contour mapping. In the process of the demonstrating the concept, we have also shown that overfitting is reduced too.

There are further improvements that could be made such as further improving on accuracy by eliminating noise that is contained within the contoured images or even calculating the midpoint of a person's lower and upper lip and joining the coordinates to take into account people who have different lip sizes. We can even test out all the edge detection methods use for extracting the outlines of objects in images.

An architecture like LipNet could even be retrained by applying contour mapping to the entire GRID dataset and recreating the two scenarios of "overlapped speakers" and "unseen speakers" to see if improved accuracies are yielded.

REFERENCES

[1] Z. Zhou, G. Zhao, X. Hong and M. Pietikainen. (2014). A review of recent advances in visual speech decoding. Image and vision computing.

[2] A. Garg, J. Noyola, and S. Bagadia. (2016). Lip reading using CNN and LSTM. Technical report Stanford University - CS231n project report.

[3] M. Wand, J. Koutnik, and J. Schmidhuber. (2016). Lipreading with long short-term memory. In IEEE International Conference on Acoustics, Speech and Signal Processing pp 6115-6119.

[4] J. S. Chung and A. Zisserman. (2016). Lip Reading in the Wild. Asian Conference on Computer Vision.

[5] J. S. Chung, A. Zisserman, A. Senior and O. Vinyals. (2017). Lip Reading Sentences in the Wild. IEEE Conference on Computer Vision and Pattern Recognition.

[6] Y. M. Assael, B. Shillingford, S. Whiteson and N. de Freitas. (2016). LipNet: End-to-Endsentence-Level Lipreading. ICLR Conference.

[7] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno and T. Ogata. (2014). Lipreading using convolutional neural network. In Interspeech.

[8] M. Cooke, J. Barker, S. Cunningham and X. Shao (2006). An audio-visual corpus for speech perception and automatic speech recognition. The Journal of the Acoustical Society of America.

[9] S. Dupont and J. Luettin. (2000). Audio-visual speech modeling for continuous speech recognition. IEEE Transactions on Multimedia.

[10] E. D. Petajan (1984). Automatic lipreading to enhance speech recognition - PhD Dissertation. University of Illinois at Urbana-Champaign.

[11] N. Ahmed, T. Natarajan, and K. R. Rao. (1974). Discrete cosine transfom. IEEE transactions on Computers.

[12] M. Leszczynski and W. Skarbek. (2005). Viseme Recognition - A Comparative Study. IEEE International Conference on Advanced Video and Signal-Based Surveillance(AVSS).

[13] N. Puviarasan and S. Palanivel. (2011). Lip reading of hearing impaired persons using HMM. Expert Systems with Applications.

[14] T. Cootes, G. Edwards and C. Taylor. (1998). Active appearance models. Lecture Notes in Computer Science.

[15] T. Cootes and C. Taylor. (1992). Active Shape Models - ”smart snakes”. Proceedings of the British Machine Vision Conference.

[16] H.E. Cetingul et al. (2006). Discriminative Analysis of Lip Motion Features for Speaker Identification And Speech Reading. IEEE Transactions on Image Processing.

[17] A. Krizhevsky, I. Sutskever and G. Hinton. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems.

[18] J. Chung, C. Gulcehre, K. Cho and Y. Bengio. (2014). Empirical evaluation of gated recurrent neural networks on sequence modelling. arXiv preprint arXiv:1412.3555.

[19] A. Graves, S. Fernandez, F. Gomez and J. Schmidhuber. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. International Conference on Machine Learning pages 369-376.

[20] J. Matthews. (2002). An introduction to edge detection: The sobel edge detector. Available at http://www.generation5.org/content/2002/im01.asp.

[21] L. G. Roberts. (1965). Machine perception of 3-D solids. Optical and Electro-Optical Information Processing - MIT Press.

[22] R. C. Gonzalez and R. E. Woods. (2002). Digital Image Processing. 2nd Edition of Prentice Hall.

[23] J. F. Canny. (1986). A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[24] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. (2010). Contour Detection and Hierarchical Image Segmentation. IEEE Transactions on Pattern Analysis and Intelligence.