INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Resource-adaptive Deep Learning for Visual Speech Recognition

Alexandros Koumparoulis[1], *Gerasimos Potamianos*[1], *Samuel Thomas*[2], *Edmilson da Silva Morais*[3]

[1] Electrical and Computer Engineering Department, University of Thessaly, Volos, Greece
[2] IBM Research AI, Yorktown Heights, USA
[3] IBM Research AI, São Paulo, Brazil

alkoumpa@uth.gr, gpotam@ieee.org, sthomas@us.ibm.com, edmorais@br.ibm.com

Abstract

We focus on the problem of efficient architectures for lipreading that allow trading-off computational resources for visual speech recognition accuracy. In particular, we make two contributions: First, we introduce MobiLipNetV3, an efficient and accurate lipreading model, based on our earlier work on MobiLipNetV2 and incorporating recent advances in convolutional neural network architectures. Second, we propose a novel recognition paradigm, called MultiRate Ensemble (MRE), that combines a "lean" and a "full" MobiLipNetV3 in the lipreading pipeline, with the latter applied at a lower frame rate. This architecture yields a family of systems offering multiple accuracy vs. efficiency operating points depending on the frame-rate decimation of the "full" model, thus allowing adaptation to the available device resources. We evaluate our approach on the TCD-TIMIT corpus, popular in speaker-independent lipreading of continuous speech. The proposed MRE family of systems can be up to 73 times more efficient compared to residual neural network based lipreading, and up to twice as MobiLipNetV2, while in both cases reaching up to 8% absolute WER reduction, depending on the MRE chosen operating point. For example, a temporal decimation of three yields a 7% absolute WER reduction and a 26% relative decrease in computations over MobiLipNetV2.

Index Terms: visual speech recognition, lipreading, deep learning, MobileNet, CNNs, ResNet, computational efficiency.

1. Introduction

Much of the success in deep-learning based visual speech recognition (VSR) can be attributed to convolutional neural networks (CNNs) [1–6]. While such models achieve high recognition performance, they are very computationally intensive and require significant hardware resources for efficient execution and storage, thus hindering implementation on resource-limited devices. For example, in the VSR system of [3], the 2D CNN alone has 67.46×10^6 parameters and requires 11.22×10^9 floating point operations (FLOPs) to process a single frame.

Surprisingly, it's only very recently that works on resource-efficient VSR have appeared in the literature. Specifically, in [7], several existing VSR models are compared in terms of efficiency vs. accuracy. In [8], ideas from the ShuffleNet module [9] and spatio-temporal depthwise convolution are introduced into a residual neural network (ResNet) [10] for VSR. Finally, in our earlier work [11], several MobileNet-based architectures [12, 13] are explored, using spatio-temporal extensions suitable for VSR, leading to significant computation savings.

Resource-efficient CNN architectures are based on replacing standard convolutional layers with pointwise (PW) and depthwise (DW) convolution and / or a shuffle operation. This leads to dramatic computation and storage savings. For example, for 2D 3×3 convolutional kernels in MobileNetV1 [12], computations are decreased by a factor of 8 to 9. Similar gains are observed for spatio-temporal problems, where 3D convolutions are decomposed to 3D PW and 2D DW ones. Alternative approaches for efficient computation and storage, e.g., quantization or pruning, are orthogonal to such decomposition and can lead to additional improvements.

To further reduce computations, one can either tune the input resolution multiplier to modify its spatial dimensions, or the network width multiplier α that uniformly controls the number of channels in each convolutional layer, by scaling a default configuration by factor α. Decreasing input resolution reduces computations, however it also shrinks objects, thus requiring network adjustments. For VSR, input size has been investigated in [14], and performance has been shown to be directly linked to the input resolution and physical coverage. Similarly, reducing the width multiplier α hurts performance, especially for very small configurations. For example, for MobileNetV1, the version with $\alpha = 0.25$ exhibits a 20% absolute performance degradation compared to the full model ($\alpha = 1.0$) [12].

A different approach is presented in the SlowFast network [15]. This consists of two models, a "slow" one that operates at a low frame rate to capture spatial semantics, and a "fast" one that operates at high frame rate to capture motion at fine temporal resolution. The latter is a very lean model, having a reduced number of channels. To obtain the final prediction, lateral connections fuse information from both branches. By changing the frame rate of the slow model, different performance and efficiency operation points can be achieved.

Motivated by the above, we focus on efficient VSR models that allow a trade-off between computational efficiency and recognition accuracy. In particular, we make two contributions:

(i) First, inspired by MobileNetV3 [16], we introduce the MobiLipNetV3 CNN for lipreading, which is based on an architecture similar to our earlier efficient VSR model, MobiLipNetV2 [11], but with the h-swish activation function and a squeeze-and-excitation unit. Details of this single-CNN model are provided in Section 2.

(ii) Second, we propose a two-CNN VSR architecture, called MultiRate Ensemble (MRE), consisting of two MobiLipNetV3 CNNs: an accurate, but computationally expensive one that runs at a decimated frame rate, and a lean CNN operating on every frame. Combined features of the two are used for VSR. The recognition network is made aware of the rate decimation factor by an embedding vector. Varying the frame rate of the expensive model allows for accuracy vs. efficiency trade-off, which can then be decided based on the available implementation platform. Details can be found in Section 3.

The proposed models are evaluated in Section 4 for speaker-independent continuous VSR on the TCD-TIMIT corpus [17], a very popular lipreading dataset [18–25]. It is observed that

http://dx.doi.org/10.21437/Interspeech.2020-3003

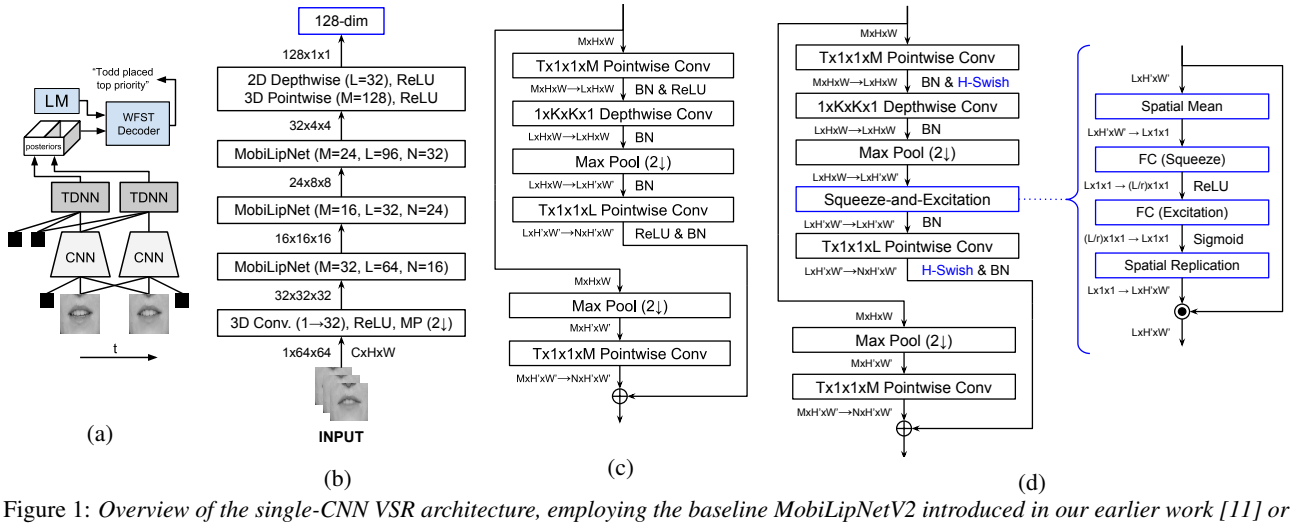

Figure 1: *Overview of the single-CNN VSR architecture, employing the baseline MobiLipNetV2 introduced in our earlier work [11] or the proposed MobiLipNetV3. (a) Entire system architecture; (b) CNN structure, with its three middle layers being either MobiLipNetV2 modules (detailed in (c)), or MobiLipNetV3 modules (detailed in (d)). In blue, differences between the two MobiLipNets are highlighted.*

the introduced MobiLipNetV3 CNN is significantly more accurate than its MobiLipNetV2 predecessor, while having a similar computational cost. Further, the MRE two-model architecture outperforms both MobiLipNetV2 and MobiLipNetV3 CNNs in accuracy and / or efficiency at numerous operating points.

2. MobiLipNetV3

2.1. Baseline system

We start from our earlier work on resource-efficient VSR and use the single-CNN VSR system of [11] as the basis of developing MobiLipNetV3. The VSR system is shown in Fig. 1a, unrolled for two frames. It takes as input a greyscale mouth region, applies a CNN, a time delay neural network (TDNN), and a WFST-based decoder with a language model (LM).

Among all CNN variants of [11], we adopt the best-performing one, namely the 3D PW MobiLipNetV2 CNN, as our baseline. Its five-layer structure, depicted in Fig. 1b, consists of a regular convolutional layer at the input, three 3D PW MobiLipNetV2 modules, and finally a projection layer. The 3D PW modules follow the inverted residual approach, shown in Fig. 1c. The module operation starts by applying 3D PW convolution ($T = 3$) to the input tensor for expanding the number of channels ($M \rightarrow L$). Then spatial filtering using 2D DW convolution (kernel size $K \times K$, with $K = 3$) is applied on L channels and max-pool downsampling (MP). Finally, 3D PW convolution reduces the number of channels ($L \rightarrow N$). A linear residual connection is also used, where 3D PW convolution is employed to match the number of channels ($M \rightarrow N$). At each time step, the network outputs a 128-dimensional vector.

A two-layer TDNN is subsequently applied on the CNN feature vectors. First, the current and two past vectors are concatenated, then fed to a fully-connected (FC) layer to yield 128-dimensional vectors, and finally a projection FC layer maps these to the number of phonemes.

The overall computational cost of the baseline VSR system of a single time-step for the CNN is 18.33M FLOPs (53k CNN model parameters) and 421k FLOPs for the TDNN.

2.2. H-Swish activation function

The first improvement to MobiLipNetV2 is replacing its ReLU activations with the h-swish function. Initially, swish was introduced in [26], by leveraging automated activation search techniques. It is defined as: $\text{swish}(x) = x \cdot \sigma(x)$, and is a self-modulating activation function, where the input is multiplied by the output of the sigmoid function. Compared to ReLU, it leads to improved recognition performance across a number of image tasks, however it has the disadvantage of being computationally expensive due to the sigmoid function. To circumvent this, in [16] the sigmoid is replaced with a shifted and scaled version of ReLU6, namely an upper bounded ReLU defined as: $\text{ReLU6}(x) = \min(6, \text{ReLU}(x))$. This approximation of swish is named hardswish (h-swish) and is given by:

$$\text{h-swish}(x) = x \cdot \frac{\text{ReLU6}(x+3)}{6}.$$

This activation function follows swish closely, and its computation can be implemented efficiently.

2.3. Squeeze-and-Excitation

A second addition to the MobiLipNetV2 model is the Squeeze-and-Excitation (SE) unit [27]. SE is a self-gating mechanism that modulates information on intermediate spatial features according to a global representation, shown in Fig. 1d (right). First, per-channel information is gathered using spatial mean, subsequently an FC layer reduces the number of channels (squeeze) L by factor r (we use $r = 4$), and a second FC layer expands this to the initial number of channels. A sigmoid function is applied and the result used to modulate the input feature maps. Compared to static normalization techniques such as BatchNorm [28], SE is able to discard information dynamically conditioned on input, and from this point of view it is closer in spirit to the attention mechanism [29], however is not able to amplify information like the BatchNorm affine transform.

2.4. Module cost

With the addition of the SE unit and the replacement of ReLU with h-swish, the resulting MobiLipNetV3 module has more computational requirements than its predecessor. For this reason, we further alter the module by swapping the 2D DW convolution with the max-pooling layer. This reduces the spatial dimensions earlier inside the module, and thus the required FLOPs. The module is detailed in Fig. 1d. On the left side, the inverted residual block is depicted with spatio-temporal PW ($T = 3$) and spatial DW convolutions, and on the right side the

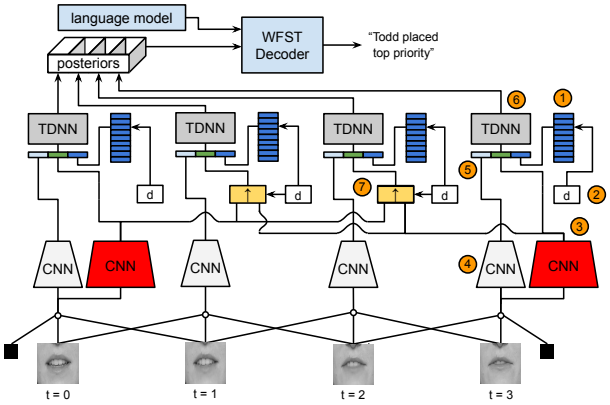

Figure 2: *The MRE system architecture, shown operating on four video frames. The system contains the following: (1) decimation embedding layer; (2) temporal decimation factor d; (3) FullCNN; (4) LeanCNN; (5) feature concatenation; (6) TDNN; (7) FullCNN feature temporal upsampling. The TDNN also depends on two previous feature vectors (5), not shown here.*

expanded SE unit. The residual connection lacks an activation function, since compared to a non-linear residual connection it leads to improved performance.

In Fig. 1c and Fig. 1d we denote by M and N the number of input and output convolutional layer channels, respectively, and by L the number of intermediate channels. We also denote by H and W the height and width of the input tensor, and primed symbols denote halving these quantities, i.e., $H' = H/2$ and $W' = W/2$. For this section alone, and similarly to [12], we only consider multiplications in the efficiency computations. To provide numerical examples of required FLOPs for the two modules, we assume input of size $M = 16$, $L = 128$, $N = 32$, $H = W = 32$, PW kernel size $T = 3$, and DW kernel size $K = 3$.

For MobiLipNetV2, the first PW convolution layer requires $M\,L\,H\,W$ multiplications (2.09M FLOPs, 2048 parameters), the DW one costs $K^2 L\,H\,W$ (1.17M FLOPs, 1152 parameters), the second PW convolution costs $L\,N\,H\,W/4$ due to the preceding max-pooling (1.04M FLOPs, 4096 parameters), and the residual connection yields $M\,N\,H\,W/4$ multiplications (131k FLOPs, 512 parameters). The total cost of a single such module is therefore 4.43M FLOPs and 7k parameters.

Similarly, for MobiLipNetV3, the first PW convolution layer requires $M\,L\,H\,W$ multiplications (2.09M FLOPs, 2048 parameters), the DW one costs $K^2 L\,H\,W$ (1.17M FLOPs, 1152 parameters), the second PW convolution costs $L\,N\,H\,W/4$ due to the preceding max-pooling (1.04M FLOPs, 4096 parameters), and the residual connection yields $M\,N\,H\,W/4$ multiplications (131k FLOPs, 512 parameters). The h-swish requires $2\,L\,W\,H + 2\,N\,H\,W/4$ (81k FLOPs) at the two locations applied, and SE requires $2L^2/r + LHW/4 + 4L$ multiplications (40k FLOPs and 8k parameters for $r = 4$). The overall module cost becomes 4.55M FLOPs with 16k parameters. Compared to the previous version, this is 2.7% relative more computationally intensive, and the number of parameters is approximately double.

3. MultiRate Ensemble (MRE)

We propose MRE, a new VSR system architecture that operates on the input video stream using a lightweight CNN (LeanCNN) and on a temporally decimated video stream using a computationally expensive CNN (FullCNN). The lean model is fast to execute and provides dense local predictions but of lower accuracy, while the full model provides more accurate predictions but sparse, to compensate for its computational cost. By processing the input at two different frame rates, the goal is to create a configurable system that can exchange efficiency for accuracy, by varying the frame rate of the second CNN.

Table 1: *Hyperparameters of the LeanCNN and FullCNN.*

Layer	**FullCNN** ($\alpha = 1.0$)			**LeanCNN** ($\alpha = 0.5$)		
	Filter Size	Channels	Output Size	Channels	Output Size
conv	3×3×3	1/32	32×64×64	1/16	16×64×64
MP	1×2×2	–	32×32×32	–	16×32×32
1 × 1, DW	3×1×1, 1×3×3	32/64, 64/64	64×32×32	16/32, 32/32	32×32×32
MP, 1 × 1	1×2×2, 3×1×1	–, 64/16	16×16×16	–, 32/8	8×16×16
1 × 1, DW	3×1×1, 1×3×3	16/32, 32/32	32×16×16	8/16, 16/16	16×16×16
MP, 1 × 1	1×2×2, 3×1×1	–, 32/24	24×8×8	–, 16/12	24×8×8
1 × 1, DW	3×1×1, 1×3×3	24/96, 96/96	96×8×8	12/48, 48/48	48×8×8
MP, 1 × 1	1×2×2, 3×1×1	–, 96/32	32×4×4	–, 48/32	32×4×4
DW, 1 × 1	1×4×4, 3×1×1	32/32, 32/128	128×1×1	32/32, 32/128	128×1×1

3.1. MRE recognition system

The MRE system, shown in Fig. 2, contains two MobiLipNetV3 CNNs of different width multipliers ($\alpha = 1.0$ for the FullCNN and $\alpha = 0.5$ for the LeanCNN), a TDNN, and an embedding layer. The FullCNN features are provided at a lower frame rate (decimated by factor d), thus need to be upsampled to match the input video rate. To make the network aware of the decimation factor used, an embedding layer is employed (decimation factor dependent bias unit). Although simple in its operation, such embedding improves system performance. Features from the two CNNs and the decimation embedding layer are then concatenated and fed to the TDNN to yield phoneme posteriors at every frame. These are further upsampled from the video frame rate to 100 Hz (not shown in Fig. 2 for brevity) and decoded using the WFST. Details of the two CNNs are shown in Table 1.

Training the two CNNs commences with random initialization, however does not proceed concurrently. Instead, we first train the FullCNN using a single-model setup (similar to the baseline of Fig. 1a), and then we freeze it while training the remaining MRE system components (LeanCNN, TDNN, and the embedding layer). As a result, LeanCNN training can focus on "correcting" the temporally upsampled predictions of the FullCNN. In this sense, the MRE system operates in an "ensemble" fashion. Training with frozen FullCNN layers helps the LeanCNN converge faster, compared to training both CNNs simultaneously. During training, the decimation factor is selected uniformly over a 1-9 range, remaining constant within a batch.

Computational savings are realized by decimating the rate of the FullCNN. Overall, the computational cost of the MRE system is $C_{\text{Lean}} + (1/d)\,C_{\text{Full}} + C_{\text{TDNN}}$ for the two CNNs and the TDNN. For MobiLipNetV3, $C_{\text{Lean}} = 7.29$M FLOPs (29.08k parameters), $C_{\text{Full}} = 19.00$M FLOPs (60.28k parameters), and $C_{\text{TDNN}} = 520$k FLOPs (259k parameters). By varying the rate decimation factor ($d = 1, \ldots, 9$), the MRE system cost can be reduced to as low as 9.92M FLOPs per video frame (for $d = 9$), or reach as high as 26.81M FLOPs (for $d = 1$).

4. Experiments

4.1. Additional system details

To create inputs for VSR, our visual pre-processing pipeline of [11] is employed. Specifically, face detection is first performed on every video frame by a ResNet-10 with SSD [30] network, available in OpenCV v3.4 [31]. Then, facial landmarks are detected as in [32], and four mouth points are used, median-filtered over a 7-frame window, to yield smooth mouth center, width, and height estimates. Based on these, a greyscale

Table 2: *Comparison of various VSR systems in terms of recognition performance (in WER, %, on the TCD-TIMIT test set) and efficiency (in per-frame FLOPs and number of parameters, computed for the CNNs only). MBLN.V2 stands for MobiLipNetV2 and MBLN.V3 for MobiLipNetV3. For the latter, the h-swish alone (i.e., without SE), as well as various width multipliers (α) are considered. For the MRE architecture, systems with and without the embedding layer are considered for various frame rate decimation factors (d).*

Metric	**Baselines**		**MbLN.V3**		**MbLN.V3 (α)**			**MRE with embedding (d)**					**MRE without embedding (d)**				
	3D-ResNet	MBLNv2	without SE	with SE	1.0	0.5	0.25	1	3	5	7	9	1	3	5	7	9
WER (%)	52.94	53.01	48.82	48.07	48.07	53.89	61.34	44.86	45.99	48.01	51.01	53.67	45.05	46.32	48.71	51.22	54.87
FLOPs (M)	681.01	18.33	18.66	19.00	19.00	7.29	3.11	26.29	13.57	11.05	9.98	9.38	26.29	13.57	11.05	9.98	9.38
# params. (k)	5658.62	53.02	53.02	60.28	60.28	29.08	19.08	82.10	82.10	82.10	82.10	82.10	82.10	82.10	82.10	82.10	82.10

mouth region-of-interest is extracted (approximately enlarged by 40% over the mouth width and height), normalized to 64 $\times$ 64 pixels, to be fed to the CNN(s).

VSR network training is driven by frame-level sub-phonetic targets, obtained by forced alignment with a triphone audio-only GMM-HMM system built on a traditional acoustic front-end (MFCC plus derivatives features, followed by LDA and MLLT) using `Kaldi` [33]. Both the single-model CNN-TDNN and MRE systems are trained using cross-entropy and SGD with dropout regularization with $p = 0.1$. Each minibatch contains three full length videos.

For recognition, network outputs are interpolated from the 30 Hz video frame rate to 100 Hz, prior to decoding with a WFST. The latter incorporates a bigram language model with Witten-Bell smoothing, developed on the training set of the TCD-TIMIT corpus, similar to the use of bigrams in earlier VSR systems developed on such data, e.g. [18].

4.2. Dataset

Our experiments are conducted on TCD-TIMIT [17], a corpus of audio-visual continuous speech by 62 speakers uttering 6913 phonetically-rich TIMIT sentences (6k word vocabulary) in studio-like conditions. The frontal-view video recordings of the corpus are used here, available at a 1920 $\times$ 1080-pixel resolution and 30 Hz video frame rate. Experiments are performed following the official speaker-independent protocol provided with the data (39 training and 17 test subjects).

4.3. Results

We investigate a number of VSR models and their variations discussed in this paper. We report visual speech recognition performance on the TCD-TIMIT speaker-independent test set in WER, %, as well as model efficiency in terms of computational load (FLOPs per input video frame) and storage requirements (number of parameters) of their CNN components. The results are listed in Table 2, with some also visualized in Fig. 3.

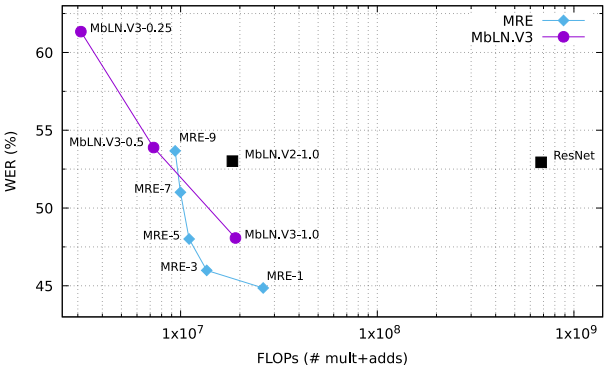

Figure 3: *Recognition performance on the TCD-TIMIT test set (in WER, %) vs. efficiency (in per-frame FLOPs) of two baselines (MobiLipNetV2 and ResNet), the proposed MobiLipNetV3 (with three width configurations of α=1.0, 0.5, 0.25), and the proposed MRE (with five frame rate decimation factors d=1, 3, 5, 7, 9). Model name abbreviations are explained in Table 2.*

First, MobiLipNetV2 and the state-of-the-art 3D-ResNet baselines from [11] are examined. Next, MobiLipNetV3 is considered ("with SE"), together with a variation of it without SE, as well as various additional width multipliers ($\alpha = 0.5$, 0.25). We can readily observe that MobiLipNetV3 significantly reduces WER over MobiLipNetV2 (4.9% absolute), with a small only degradation in efficiency (4% relative). Efficiency improvements over the ResNet are dramatic (over 35 times). Reducing the model width multiplier α further improves efficiency, however it's detrimental to VSR accuracy. For example, applying $\alpha = 0.25$ improves MobiLipNetV3 efficiency by over 6 times, however degrades WER by 13.3% absolute.

Finally, the proposed MRE architecture manages to further improve accuracy and efficiency simultaneously. In Table 2, various such systems are evaluated, by considering five frame decimation factors d, as well as model variations without the embedding layer. It can be readily observed that embedding helps, while Fig. 3 provides a visualization of the achieved recognition performance vs. efficiency for various MRE operating points, as compared to the baselines and MobiLipNetV3. For example, choosing a decimation factor of $d = 9$ for the Full-CNN, the system turns out 73 times more efficient than ResNet and twice as much as MobiLipNetV2 and MobiLipNetV3. On the other hand, selecting $d = 1$, reduces WER by 8.1% absolute over the ResNet and MobiLipNetV2, as well as 3.2% over the MobiLipNetV2. Factor $d = 3$ appears to provide a good trade-off between VSR accuracy and efficiency, improving both over the ResNet, MobiLipNetV2, and MobiLipNetV3.

5. Conclusions

In this paper, we focused on resource-adaptive deep-learning based lipreading, making two contributions on efficient VSR architectures. First, we presented MobiLipNetV3, extending our earlier efficient MobiLipNetV2 model, by incorporating the h-swish activation function and squeeze-and-excitation units. The new model outperformed MobiLipNetV2 by 4.9% absolute WER, but at a small 4% relative increase in FLOPs. Second, we introduced the MRE, a new VSR system paradigm that processes video input at two distinct frame rates by means of two MobiLipNetV3 CNNs with different width multipliers. By varying the frame rate decimation of the more computationally expensive CNN, we reached various accuracy vs. efficiency operating points. A good trade-off between the two yielded further recognition improvements over both MobiLipNetV2 and MobiLipNetV3 by 7.0% and 2.1% absolute WER reduction, respectively, while also improving efficiency, reducing FLOPs by a relative 26% and 29%, respectively. The specific operating point exhibits a dramatic 50 times better efficiency than ResNet, while maintaining a 7.0% absolute WER reduction over it.

6. References

[1] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, "LipNet: End-to-end sentence-level lipreading," *CoRR*, arXiv:1611.01599v2, 2016.

[2] T. Stafylakis and G. Tzimiropoulos, "Combining residual networks with LSTMs for lipreading," in *Proc. Interspeech*, 2017, pp. 3652–3656.

[3] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip reading sentences in the wild," in *Proc. CVPR*, 2017, pp. 3444–3453.

[4] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Deep audio-visual speech recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, 2018.

[5] G. Potamianos *et al.*, "Audio and visual modality combination in speech processing applications," in *The Handbook of Multimodal-Multisensor Interfaces, Vol. 1*, S. Oviatt *et al.*, Eds. Morgan-Claypool, 2017, pp. 489–543.

[6] A. Fernandez-Lopez and F. M. Sukno, "Survey on automatic lipreading in the era of deep learning," *Image and Vision Computing*, vol. 78, pp. 53–72, 2018.

[7] M. Van keirsbilck, B. Moons, and M. Verhelst, "Resource aware design of a deep convolutional-recurrent neural network for speech recognition through audio-visual sensor fusion," *CoRR*, arXiv:1803.04840v1, 2018.

[8] N. Shrivastava, A. Saxena, Y. Kumar, R. R. Shah, A. Stent, D. Mahata, P. Kaur, and R. Zimmermann, "MobiVSR: Efficient and light-weight neural network for visual speech recognition on mobile devices," in *Proc. Interspeech*, 2019, pp. 2753–2757.

[9] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. CVPR*, 2018, pp. 6848–6856.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[11] A. Koumparoulis and G. Potamianos, "MobiLipNet: Resource-efficient deep learning based lipreading," in *Proc. Interspeech*, 2019, pp. 2763–2767.

[12] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, arXiv:1704.04861v1, 2017.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. CVPR*, 2018, pp. 4510–4520.

[14] A. Koumparoulis, G. Potamianos, Y. Mroueh, and S. J. Rennie, "Exploring ROI size in deep learning based lipreading," in *Proc. AVSP*, 2017, pp. 64–69.

[15] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. ICCV*, 2019, pp. 6202–6211.

[16] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for MobileNetV3," in *Proc. ICCV*, 2019, pp. 1314–1324.

[17] N. Harte and E. Gillen, "TCD-TIMIT: An audio-visual corpus of continuous speech," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 603–615, 2015.

[18] K. Thangthai and R. W. Harvey, "Building large-vocabulary speaker-independent lipreading systems," in *Proc. Interspeech*, 2018, pp. 2648–2652.

[19] G. Sterpu, C. Saam, and N. Harte, "Attention-based audio-visual fusion for robust automatic speech recognition," in *Proc. ICMI*, 2018, pp. 111–115.

[20] A. H. Abdelaziz, "Turbo decoders for audio-visual continuous speech recognition," in *Proc. Interspeech*, 2017, pp. 3667–3671.

[21] ——, "Comparing fusion models for DNN-based audiovisual continuous speech recognition," *IEEE/ACM Trans. Audio Speech Language Process.*, vol. 26, no. 3, pp. 475–484, 2018.

[22] S. Zhang, M. Lei, B. Ma, and L. Xie, "Robust audio-visual speech recognition using bimodal DFSMN with multi-condition training and dropout regularization," in *Proc. ICASSP*, 2019, pp. 6570–6574.

[23] A. Gabbay, A. Ephrat, T. Halperin, and S. Peleg, "Seeing through noise: Visually driven speaker separation and enhancement," in *Proc. ICASSP*, 2018, pp. 3051–3055.

[24] T. Halperin, A. Ephrat, and S. Peleg, "Dynamic temporal alignment of speech to lips," *CoRR*, arXiv:1808.06250v1, 2018.

[25] K. Vougioukas, S. Petridis, and M. Pantic, "End-to-end speech-driven facial animation with temporal GANs," *CoRR*, arXiv:1805.09313v4, 2018.

[26] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," in *Proc. ICLR Work.*, 2018.

[27] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. CVPR*, 2018, pp. 7132–7141.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.

[29] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, arXiv:1409.0473v7, 2014.

[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. ECCV*, 2016, pp. 21–37.

[31] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, 2008.

[32] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 fps via regressing local binary features," in *Proc. CVPR*, 2014, pp. 1685–1692.

[33] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.