INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

MobiLipNet: Resource-efficient deep learning based lipreading

Alexandros Koumparoulis , Gerasimos Potamianos

Electrical and Computer Engineering Department, University of Thessaly, 38221 Volos, Greece

alkoumpa@uth.gr, gpotam@ieee.org

Abstract

Recent works in visual speech recognition utilize deep learning advances to improve accuracy. Focus however has been primarily on recognition performance, while ignoring the computational burden of deep architectures. In this paper we address these issues concurrently, aiming at both high computational efficiency and recognition accuracy in lipreading. For this purpose, we investigate the MobileNet convolutional neural network architectures, recently proposed for image classification. In addition, we extend the 2D convolutions of MobileNets to 3D ones, in order to better model the spatio-temporal nature of the lipreading problem. We investigate two architectures in this extension, introducing the temporal dimension as part of either the depthwise or the pointwise MobileNet convolutions. To further boost computational efficiency, we also consider using pointwise convolutions alone, as well as networks operating on half the mouth region. We evaluate the proposed architectures on speaker-independent visual-only continuous speech recognition on the popular TCD-TIMIT corpus. Our best system outperforms a baseline CNN by 4.27% absolute in word error rate and over 12 times in computational efficiency, whereas, compared to a state-of-the-art ResNet, it is 37 times more efficient at a minor 0.07% absolute error rate degradation.

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

1. Introduction

The field of visual speech recognition (VSR), or lipreading, has witnessed dramatic breakthroughs recently, primarily due to the paradigm shift from hand-crafted features to deep learning based models [1–8], coupled with the public release of large suitable corpora in a variety of environments [9–15], as also reviewed in [16,17]. Such models however, while reducing recognition errors compared to previous approaches, are not as efficient to compute and store. Thus, on low-resource platforms such as smartphone and other embedded devices, speed and size constraints render them impractical. Reducing model size and improving computational efficiency, while at the same time not sacrificing model accuracy, is therefore crucial to lipreading technology deployment, and it constitutes the main focus and contribution of this paper. Surprisingly, the topic has not been previously addressed in the literature, other than [18] that touches on efficiency vs. accuracy of existing architectures, and a concurrent to our work publication [19] with a similar focus to us, which utilizes MobileNet V1 blocks [20] in conjunction with a residual neural network (ResNet) architecture for VSR.

To further motivate the paper, approximate model parameter sizes and computational requirements of some recently proposed deep VSR architectures are summarized. For example, in [4], a 3D convolutional neural network (CNN) is developed, combined with a gated recurrent unit (GRU) [21]. The model has 4.57M parameters, and it requires about 123.82M floating point operations (FLOPs) to compute posteriors for a single video frame. In [14], a 2D CNN within a sequence-to-sequence architecture is used. The model contains 67.46M parameters (excluding temporal modeling), and CNN-based processing of a single frame consumes more than 11.22G FLOPs. In [5], several ResNet [22] variants are considered. The best performing model needs about 1.33G FLOPs per frame and has 21.27M parameters. Finally, an 11.17M-parameter ResNet is used in [8], requiring approximately 956.55M FLOPs per frame.

All aforementioned lipreading models employ convolutional layers in their architectures. Providing efficient models for such layers has been of recent interest in the computer vision literature, constituting our starting point. In particular, we base our work on the MobileNet CNN architectures (V1 and V2), recently proposed for image classification and other static vision tasks, which employ 2D convolutions [20, 23]. In order to better model the spatio-temporal nature of the lipreading problem, we extend such convolutions to 3D ones, i.e., allowing them to operate over adjacent frames in addition to the current one. This gives rise to highly efficient and accurate models that we refer to as "MobiLipNets". We investigate various architectures in this approach, as detailed in Section 2.1. On top of these, for temporal modeling, we do not use a recurrent neural network (RNN) as is common practice, but instead employ a time delay neural network (TDNN). As a result, we avoid computational costs inherent to the multiple gating mechanisms of most recurrent architectures, as discussed in Section 2.2.

We provide additional system implementation details in Section 3, and we evaluate our developed networks in Section 4. Specifically, we study computational efficiency and VSR accuracy on the publicly available TCD-TIMIT corpus [12], a popular database for lipreading [18, 24–28] and other audio-visual speech processing tasks [29–31]. Our experiments show that our best model, a "MobiLipNetV2" with 3D pointwise convolutions, exhibits dramatically improved computational efficiency compared to both a baseline 3D-CNN and a state-of-the-art ResNet, with no or minimal accuracy degradation. We also note that our approach could be combined with model pruning [32], weight quantization [33], and model distillation [34], thus potentially leading to further size and speed improvements.

2. Resource-efficient lipreading modules

This section introduces the basic network modules that constitute the building blocks of resource-efficient VSR. Fig. 1 provides an overview of the VSR system, where a CNN is used for visual feature extraction, a TDNN for temporal modeling, and a weighted finite-state transducer (WFST) for decoding.

2.1. Resource-efficient CNN modules

We first describe the 2D MobileNet blocks (V1 and V2), used as templates of layers to compose larger networks, followed by their 3D extensions into MobiLipNet blocks. We explore sev-

http://dx.doi.org/10.21437/Interspeech.2019-2618

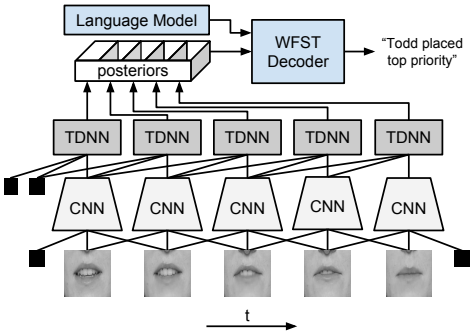

Figure 1: *Overview of the proposed VSR system. The CNN and TDNN take as input 3 consecutive frames, creating posteriors that are then used for decoding via a WFST. Division by priors is omitted for clarity. Black squares denote zero-filled input.*

eral variations in the process, each resulting in different FLOP and accuracy characteristics. In the following, we ignore activation functions and batch-normalization for brevity, and for spatial dimension reduction we use max-pooling ($1\times2\times$2F, $1\times2\times$2S) instead of non-unit stride, due to improved VSR recognition. We denote by M and N the number of input and output convolutional layer channels respectively, with N convolutional kernels of size $K\times K\times M$ applied on M slices of $I_W\times I_H$-pixel size input in the 2D case. For 3D, the latter are temporally stacked to $T\times I_W\times I_H$-pixel size volumes (time$\times$width $\times$height). To provide numerical examples of FLOPs and model sizes (listed inside parentheses), we assume $M=64$, $N=128$, $I_H=I_W=32$, and $K=T=3$, with minor variations for the V2 systems. For this section alone, and similarly to [20], we only consider multiplications in the efficiency computations.

MobileNetV1: The MobileNetV1 block [20] replaces standard 2D convolutions by depthwise separable ones that are decomposed to depthwise and pointwise (1×1) convolutions, yielding two corresponding convolutional layers (see also Fig. 2(a)). Hence, the number of the standard-case multiplications, namely $NMK^2I_HI_W$ (75.49M FLOPs, 73k parameters), are replaced by those required for the M, $K\times K\times$ 1-sized kernels of the depthwise convolution layer and for the N, $1\times1\times M$-sized kernels of the pointwise convolution layer. These are $MK^2I_HI_W$ multiplications (0.58M FLOPs, 576 parameters) and NMI_HI_W ones (8.38M FLOPs, 8192 parameters) respectively [20], thus resulting to about 8.4 times savings in computations, as well as model size, for the specifics considered.

MobileNetV2: In MobileNetV2 [23], a new block is introduced, termed "inverted residual with linear bottleneck". Its input is a low-dimensional compressed representation that is expanded with pointwise convolution (increasing the number of channels), then filtered with depthwise convolution, and compressed back with pointwise convolution. A residual connection between input and output is also employed, thus, if their channel numbers do not match, a pointwise convolution is also incorporated (see Fig. 2(c)). Compared to MobileNetV1, the number of input and output channels are typically significantly smaller [23]. In our case, we employ a quarter of the MobileNetV1 channels, thus $M=16$ and $N=32$. We also use $L=128$ channels for depthwise separable convolution. Then, the first pointwise convolution layer requires $M\,L\,I_HI_W$ multiplications (2.09M FLOPs, 2048 parameters), the depthwise one costs $K^2L\,I_HI_W$ (1.17M FLOPs, 1152 parameters), the second pointwise convolution costs $L\,N\,I_HI_W/4$ due to the preceding max-pooling (1.04M FLOPs, 4096 parameters), and the

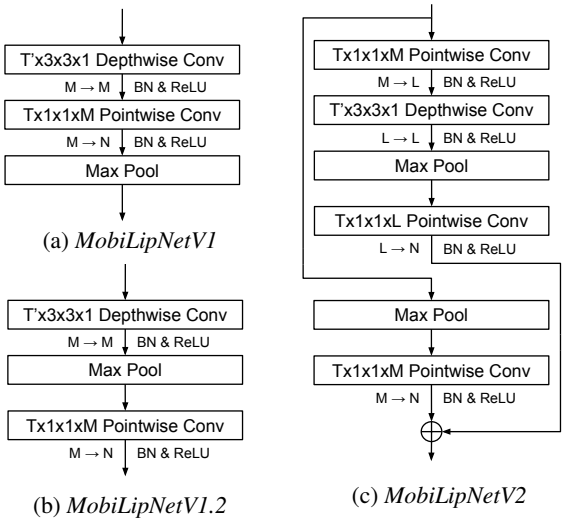

Figure 2: *The basic network building blocks of Section 2.1: (i) For T=T′=1, (a) and (c) yield standard MobileNetV1 and V2 blocks; (ii) For T=3 and T′=1, (a)-(c) yield 3D pointwise MobiLipNet blocks; (iii) For T=1 and T′=3, (a)-(c) yield 3D depthwise MobiLipNet blocks. Batch-normalization (BN), ReLU activations, and input → output channels are also shown.*

residual connection yields $M\,N\,I_HI_W/4$ multiplications (131k FLOPs, 512 parameters).

MobiLipNetV1 and **MobiLipNetV2**: These constitute our proposed 3D extensions of the two MobileNet blocks, sharing the same topology as their counterparts, but after extending individually either the depthwise or the pointwise convolutions to the previous and next frames for better temporal modeling. The depthwise 3D extension replaces the original $K\times K\times1$-sized depthwise convolution kernels with $T\times K\times K\times1$ ones, while the pointwise 3D extension switches the original $1\times1\times M$-sized pointwise kernels with $T\times1\times1\times M$-sized ones.

Concerning MobiLipNetV1, for the *depthwise extension* approach, the depthwise convolution requires $TK^2MI_HI_W$ multiplications (1.76M FLOPs, 1728 parameters) and the pointwise one MNI_HI_W (8.38M FLOPs, 8192 parameters). The *pointwise extension* approach costs instead $K^2MI_HI_W$ (589k FLOPs, 576 parameters) plus $TMNI_HI_W$ (25.16M FLOPs, 24k parameters) for the two layers (see also Fig. 2(a)).

For MobiLipNetV2, the *depthwise extension* approach costs $ML\,I_HI_W$ multiplications for the first pointwise convolution (2.09M FLOPs, 2048 parameters), $TK^2L\,I_HI_W$ for the depthwise one (3.53M FLOPs, 3456 parameters), $L\,NI_HI_W/4$ for the second pointwise one (1.04M FLOPs, 4096 parameters), and $MNI_HI_W/4$ for the residual connection (131k FLOPs, 512 parameters). The *pointwise extension* approach requires $TML\,I_HI_W$ (6.29M FLOPs, 6144 parameters), $K^2L\,I_HI_W$ (1.17M FLOPs, 1152 parameters), $TL\,N\,I_HI_W/4$ (3.14M FLOPs, 12k parameters), and $TMNI_HI_W/4$ (393k FLOPs, 1536 parameters), respectively (see also Fig. 2(c)).

MobiLipNetV1.2: The MobiLipNetV1 block consists of two convolutional layers followed by max-pooling (see Fig. 2(a)). Such structure has the disadvantage that the module first performs expensive pointwise convolution, only to discard much information at the max-pooling stage next. A solution would be to swap the pointwise convolution and max-pooling layers, performing one quarter as many calculations (for our specific max-pooling choice), hoping for no significant change

Table 1: *Hyperparameters of four VSR networks of Section 3.3. DW stands for depthwise convolution layers, AVE for mean-pooling.*

Layer	Baseline 3D-CNN			Layer	MobiLipNetV2 (Pointwise 3D)			Layer	3D ResNet			Layer	Pointwise-only MobiLipNetV1		
	Filter	Channels	Output Size		Filters	Channels	Output Size		Filters	Channels	Output Size		Filters	Channels	Output Size
conv	3×3×3	1/32	32×64×64	conv	3×3×3	1/32	32×64×64	conv, MP	3×3×3,1×2×2	1/32	32×32×32	conv	3×1×1	1/32	32×64×64
MP	1×2×2	–	32×32×32	MP	1×2×2	–	32×32×32	ResBlock	$2\times\begin{bmatrix}3\times3\times3\\3\times3\times3\end{bmatrix}$	32/64	64×16×16	MP	1×4×4	–	32×16×16
conv	1×1×1	32/64	64×32×32	1 × 1, DW	3×1×1,	1×3×3 32/64,	64/64 32×32×32			64/64	64×16×16	conv	3×1×1	32/64	64×16×16
MP	1×2×2	–	64×16×16	MP, 1 × 1	1×2×2,	3×1×1 –,	64/16 16×16×16	ResBlock	$2\times\begin{bmatrix}3\times3\times3\\3\times3\times3\end{bmatrix}$	64/96	96×8×8	MP	1×2×2	–	64×8×8
conv	3×3×3	64/96	96×16×16	1 × 1, DW	3×1×1,	1×3×3 16/32,	32/32 32×16×16			96/96	96×8×8	conv	3×1×1	64/96	96×8×8
MP	1×2×2	–	96×8×8	MP, 1 × 1	1×2×2,	3×1×1 –,	32/24 24×8×8	ResBlock	$2\times\begin{bmatrix}3\times3\times3\\3\times3\times3\end{bmatrix}$	96/128	128×4×4	MP	1×2×2	–	96×4×4
conv	3×3×3	96/96	96×8×8	1 × 1, DW	3×1×1,	1×3×3 24/96,	96/96 96×8×8			128/128	128×2×2	conv	3×1×1	96/96	96×4×4
MP	1×2×2	–	96×4×4	MP, 1 × 1	1×2×2,	3×1×1 –,	96/32 32×4×4	ResBlock	$2\times\begin{bmatrix}3\times3\times3\\3\times3\times3\end{bmatrix}$	128/128	128×2×2	MP	1×2×2	–	96×2×2
conv	3×4×4	96/128	128×1×1	DW, 1 × 1	1×4×4,	3×1×1 32/32,	32/128 128×1×1			128/128	128×2×2	conv	3×1×1	96/128	128×2×2
–	–	–	–	–	–	–	–	AVE	1×2×2	–	128×1×1	AVE	1×2×2	–	128×1×1

in accuracy. The resulting *pointwise variant* (see Fig. 2(b)) requires $K^2MI_HI_W$ (0.58M FLOPs, 576 parameters) plus $TMNI_HI_W/4$ (6.29M FLOPs, 24K parameters) multiplications for the depthwise and pointwise layers respectively. Similarly, a *depthwise variant* can be designed.

Pointwise-only MobiLipNetV1: Here, the depthwise convolution layer of the MobiLipNetV1 block is entirely removed, and only pointwise convolutions are employed. This is motivated by the fact that temporal information captured by pointwise convolutions contains crucial lipreading information, as well as by work in [35]. Since this model relies primarily on the temporal dimension, spatial dimensions are more aggressively reduced for efficiency. Assuming input 1/16 times smaller than MobiLipNetV1 and 4x4 max-pooling, the pointwise convolution costs $TMNI_HI_W/16$ (1.57M FLOPs, 24k parameters).

Half-ROI models: For symmetric image input, a CNN needs only be applied on half of it, thus immediately halving costs. The mouth region-of-interest (ROI) typically fed to lipreading CNNs is expected to be almost laterally symmetric in controlled, frontal head-pose data settings [36]. To enforce full symmetry, a simple normalization can be applied by averaging the original ROI and its laterally mirrored version. Clearly, this technique is not be applicable to non-frontal and in-the-wild data, where it causes significant ROI artifacts [36].

2.2. Resource-efficient temporal modeling

For temporal modeling in all our VSR networks, a TDNN is employed. Such consists of a fully-connected layer, taking as input three spliced (two frames left context) CNN output feature vectors of I =128 dimensions, and a projection layer to J =1264 context-dependent HMM states. The first layer requires $6\,I^2 - I$ operations (98k FLOPs, 49k parameters), while the second $2\,IJ - J$ (322k FLOPs, 161k parameters), adding to 420k FLOPs and 210k parameters in total. A lazy evaluation approach [37] could further reduce projection layer costs.

If, instead of the TDNN first fully-connected layer, we had used a GRU layer, the computational cost would have been $3\,(4\,I^2 - 2\,I)$ operations (195k FLOPs, 98k parameters), due to two matrix-vector multiplications for input and recurrent connections in each gate (assuming same-size input and output) and the three gates. A bidirectional GRU would double the above. Had we used a uni-directional long short-term memory layer, we would have ended with $4\,(4\,I^2 - 2\,I)$ operations (261k FLOPs, 131k parameters) that would be further increased in the presence of peephole connections. Hence, for the same input/output dimensions, a TDNN is more computationally efficient than a recurrent network. An alternative would have been to use a fully-connected layer with no splicing (taking as input the current frame alone), thus requiring only $2\,I^2 - I$ operations (32k FLOPs, 16k parameters). This would have been more efficient, but sacrificing temporal modeling. And, according to [38], TDNNs perform significantly better than DNNs across multiple large-vocabulary continuous speech recognition tasks.

3. Additional system details

3.1. Mouth ROI extraction

To obtain VSR network input, face detection is first performed using a ResNet-10 with SSD [39] network, available in OpenCV v3.4 [40]. Then, facial landmarks are detected as in [41]. From those, four mouth landmarks are used, after median filtering over a 7-frame window, to yield smooth mouth center, width, and height estimates. Based on these, a greyscale mouth ROI is extracted (approximately enlarged by 40% over the mouth width and height), normalized to 64×64 pixels.

Although the paper focuses on the VSR models, ROI extraction has a non-negligible computational cost. Specifically, the detection model used has 2.66M parameters and a cost of 1.18G FLOPs per frame. This however could be reduced if employed at a lower frame-rate along with tracking, or by utilizing a significantly less expensive detector (e.g., AdaBoost [42]).

3.2. VSR network training and decoding

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` [43]. The CNN-TDNN based VSR systems are then trained end-to-end using cross-entropy and SGD with dropout regularization [44] with $p = 0.1$.

For decoding, 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 training-set data of the TCD-TIMIT corpus (see Section 4.1), similarly to the use of bigrams in earlier VSR works on such data, e.g. [24].

3.3. Networks considered

We now proceed to describe the CNN part of the developed VSR models (the TDNN part is identical to all, as detailed in Section 2.2). All follow a baseline 3D-CNN topology of five convolutional modules and four max-pooling layers with the same spatial dimensions, and, unless noted, the same number of channels. Further, except for the pointwise-only network, all networks keep the conventional convolution in their first layer, i.e., without factorizing it to depthwise and pointwise ones. All models (including the baseline and ResNet) are trained on the same ROI size, as this affects both performance [45] and efficiency. In more detail, the following models are considered:

Baseline: It consists of five standard 3D convolutional layers with a $3\times3\times3$ kernel size (along time × width × height), unit stride, and padding, except the last layer where a $3\times4\times4$ filter is used to obtain 128-dimensional features. At each layer the number of channels is increased by 32 (see also Table 1).

3D ResNet: It consists of ten residual blocks [22]. All convolutional layers have $3\times3\times3$ kernels, except for those in shortcut

Table 2: *Comparison of all models considered in terms of WER, per-frame FLOPs, and model size (# parameters). Labels "1×1" and "Depth" refer to pointwise and depthwise convolution model variants, respectively. The Pointwise model is a MobiLipNetV1 variant.*

Metric	**Baseline**	**ResNet**	**MobileNet**		**MobiLipNetV1**		**MobiLipNetV2**		**MobiLipNetV1.2**		**Pointwise**	**MobiLipNetV2 / 2**	
	–	–	**V1**	**V2**	1×1	Depth	1×1	Depth	1×1	Depth	1×1 only	Normalized	Half ROI
WER (%)	57.28	52.94	66.07	61.94	55.85	57.29	53.01	56.85	55.50	58.78	60.37	53.97	55.85
FLOPs	225.07M	681.00M	11.89M	12.51M	20.44M	13.78M	18.33M	18.36M	10.85M	10.59M	3.73M	9.17M	9.17M
# Param.	1.06M	5.65M	33.82k	22.04k	93.21k	40.35k	53.02k	33.44k	93.21k	40.35k	89.18k	52.76k	52.76k

connections, where $3\times1\times1$ kernels are used to match the number of input and output channels (see also Table 1).

MobileNetV1, **V2**: Based on the two 2D MobileNet blocks of Section 2.1, the V1 model closely follows a 2D variant of the baseline by factorizing standard convolutions, while the V2 details can be deduced from that of MobiLipNetV2 of Table 1.

MobiLipNetV1, **V1.2**, **V2**: Based on the three 3D MobiLipNet blocks of Section 2.1, and, considering both their pointwise and depthwise model variants, yields a total of six networks. Among these, the pointwise V2 model is detailed in Table 1.

Pointwise-only MobiLipNetV1: It has a similar topology to MobiLipNetV1, but using 3D pointwise convolutions alone. Its details are provided in Table 1.

MobiLipNetV2 / 2: All above networks can be modified to operate on half ROIs. Among them, the MobiLipNetV2 is used in our experiments, due to its superior performance. Two such networks are considered, one operating on the original half ROIs and another on normalized half ROIs.

Note also that batch-normalization [46] is applied immediately after each convolutional layer, allowing its "folding" into the convolution operations with a simple numerical manipulation, thus making it effectively free to compute. For this reason, such computations are not considered in Table 2 and Fig. 3.

4. Experiments

4.1. Dataset

Experiments are reported on the TCD-TIMIT corpus [12], a very popular dataset in the field [18, 24–31]. The database contains audio-visual recordings of continuous speech by 62 speakers uttering 6913 phonetically-rich TIMIT sentences (6k word vocabulary) in studio-like conditions, concurrently recorded by two cameras providing frontal (0^o) and near-frontal (30^o) data at a 1920×1080-pixel resolution and 30 Hz frame-rate. In this work, only the frontal view recordings are used. Experiments are performed following the official speaker-independent protocol provided with the database (39 training and 17 test subjects).

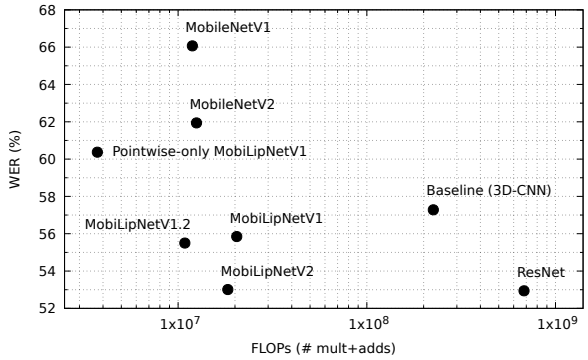

Figure 3: *Performance vs. efficiency comparison (WER vs. per-frame FLOPs) of the main VSR models considered. For MobiLipNets (V1, V2, V1.2), their pointwise variants are shown.*

4.2. Results

Efficiency and performance results of all networks of Section 3.3 are reported in Table 2. Note that concerning FLOPs and parameter numbers, the TDNN cost (420k FLOPs, 210k parameters) is not included, as this is shared among all models. Results are grouped in "clusters". Left-to-right, the 3D-CNN baseline and the state-of-the-art ResNet are shown, followed by the two 2D MobileNet architectures, the six 3D MobiLipNet models, and the pointwise-only MobiLipNetV1. Finally, at the right-most table columns, the MobiLipNetV2 systems modified to operate on half-ROIs are listed.

One can readily observe that the MobileNet architectures (V1 and V2) are dramatically more efficient (in speed, as well as model size) than both baseline and ResNet, however suffering significant WER degradation. This is primarily due to the 2D nature of their feature extraction. The issue is resolved in the proposed MobiLipNets. Among the six such models, the superiority of the pointwise (1×1) temporal extension over the depthwise one is evident, providing consistently better WERs. Among the pointwise V1, V2, and V1.2 versions, MobiLipNetV2 offers the best recognition performance (53.01% WER), utilizing residual connections to boost its accuracy. This result is very close to the ResNet WER of 52.94% (best performing system), however MobiLipNetV2 is 37 times faster and has over 106 times less parameters. Further, it outperforms the baseline significantly in WER (by 4.27% absolute), while also being 12 times faster and over 20 times leaner. MobiLipNetV1.2 balances some WER degradation (2.49% absolute) for about 60% of the MobiLipNetV2 computational cost. The pointwise-only MobiLipNetV1 is the fastest model, requiring only 3.73M flops (182 times faster than ResNet), however at a non-negligible WER degradation of 7.43%. Finally, using the MobiLipNetV2 model on half ROIs (normalized or original) is also computationally fast. If the model is trained on half the ROI with normalization, there is only a 0.96% absolute WER degradation, compared to the best MobiLipNetV2 model, while at the same time halving the required FLOPs. Without normalization, performance degrades slightly more to 55.85% WER from 53.01%.

A summary of these results (for eight models) is provided in Fig. 3 for better visualization. There, once again, it becomes evident that the MobiLipNetV2 architecture offers an optimal compromise between speed and VSR recognition performance.

5. Conclusions

In this work, we presented an exploration of the MobileNet architectures with extensions to the lipreading problem. Several network variations were trained using the same data, optimization technique, and temporal model. Through this investigation, we introduced a new model, termed MobiLipNetV2, that has 106 times less parameters and is 37 times faster than the state-of-the-art ResNet, while resulting to a slight only absolute WER degradation of 0.07% on the popular TCD-TIMIT visual-only recognition task. Further, the model outperforms a baseline 3D-CNN by 4.27% absolute in WER, 12 times in computational efficiency, and 20 times in size.

6. References

[1] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. ICML*, 2011, pp. 689–696.

[2] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, "Lipreading using convolutional neural network," in *Proc. Interspeech*, 2014, pp. 1149–1153.

[3] M. Wand, J. Koutn, and J. Schmidhuber, "Lipreading with long short-term memory," in *Proc. ICASSP*, 2016, pp. 6115–6119.

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

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

[6] T. Afouras, J. S. Chung, and A. Zisserman, "Deep lip reading: a comparison of models and an online application," in *Proc. Interspeech*, 2018, pp. 3514–3518.

[7] B. Shillingford *et al.*, "Large-scale visual speech recognition," *CoRR*, arXiv:1807.05162, 2018.

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

[9] I. Anina, Z. Zhou, G. Zhao, and M. Pietikäinen, "OuluVS2: a multi-view audiovisual database for non-rigid mouth motion analysis," in *Proc. FG*, 2015, pp. 1–5.

[10] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *J. Acoust. Soc. Am.*, 120(5): 2421–2424, 2006.

[11] L. Sari, M. Hasegawa-Johnson, S. Kumaran, G. Stemmer, and K. N. Nair, "Speaker adaptive audio-visual fusion for the open-vocabulary section of AVICAR," in *Proc. Interspeech*, 2018, pp. 3524–3528.

[12] N. Harte and E. Gillen, "TCD-TIMIT: An audio-visual corpus of continuous speech," *IEEE Trans. Multimedia*, 17(5): 603–615, 2015.

[13] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Computer Vision – ACCV 2016, Part II*, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds. Springer, 2017, pp. 87–103.

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

[15] T. Afouras, J. S. Chung, and A. Zisserman, "LRS3-TED: a large-scale dataset for visual speech recognition," *CoRR*, arXiv:1809.00496v2, 2018.

[16] 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.

[17] A. Fernandez-Lopez and F. M. Sukno, "Survey on automatic lip-reading in the era of deep learning," *Image and Vision Computing*, 78: 53–72, 2018.

[18] 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.

[19] N. Shrivastava, A. Saxena, Y. Kumar, R. R. Shah, D. Mahata, and A. Stent, "MobiVSR: A visual speech recognition solution for mobile devices," *CoRR*, arXiv:1905.03968v3, 2019.

[20] 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.

[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, arXiv:1412.3555v1, 2014.

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

[23] 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.

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

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

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

[27] ——, "Comparing fusion models for DNN-based audiovisual continuous speech recognition," *IEEE/ACM Trans. Audio Speech Language Process.*, 26(3): 475–484, 2018.

[28] 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.

[29] 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.

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

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

[32] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. Morgan-Kaufmann, 1990, pp. 598–605.

[33] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. CVPR*, 2018, pp. 2704–2713.

[34] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, arXiv:1503.02531v1, 2015.

[35] B. Wu, A. Wan, X. Yue, P. Jin, S. Zhao, N. Golmant, A. Gholaminejad, J. Gonzalez, and K. Keutzer, "Shift: A zero FLOP, zero parameter alternative to spatial convolutions," in *Proc. CVPR*, 2018, pp. 9127–9135.

[36] G. Potamianos and P. Scanlon, "Exploiting lower face symmetry in appearance-based automatic speechreading," in *Proc. AVSP*, 2005, pp. 79–84.

[37] V. Vanhoucke, A. Senior, and M. Z. Mao, "Improving the speed of neural networks on CPUs," in *Proc. Deep Learning and Unsupervised Feature Learning NIPS Workshop*, 2011.

[38] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech*, 2015, pp. 3214–3218.

[39] 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.

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

[41] 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.

[42] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. CVPR*, 2001, pp. 511–518.

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

[44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Machine Learning Research*, 15(1): 1929–1958, 2014.

[45] 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.

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