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

MobiVSR : Efficient and Light-weight Neural Network for Visual Speech Recognition on Mobile Devices

Nilay Shrivastava[1*], *Astitwa Saxena*[1*], *Yaman Kumar*[2*], *Rajiv Ratn Shah*[3], *Amanda Stent*[4], *Debanjan Mahata*[4], *Preeti Kaur*[1], *Roger Zimmermann*[5]

[1]NSIT-Delhi, India
[2]Adobe, India
[3]IIIT-Delhi, India
[4]Bloomberg, US
[5]NUS, Singapore

nilays.co@nsit.net.in, astitwas.co@nsit.net.in, ykumar@adobe.com, rajivratn@iiitd.ac.in, astent@bloomberg.net, dmahata@bloomberg.net, preetik@nsut.net.in, rogerz@comp.nus.edu.sg

Abstract

Visual speech recognition (VSR) is the task of recognizing spoken language from video input only, without any audio. VSR has many applications as an assistive technology, especially if it could be deployed in mobile devices and embedded systems. The need for intensive computational resources and large memory footprint are two major obstacles in deploying neural network models for VSR in a resource constrained environment. We propose a novel end-to-end deep neural network architecture for word level VSR called MobiVSR with a design parameter that aids in balancing the model's accuracy and parameter count. We use depthwise 3D convolution along with channel shuffling for the first time in the domain of VSR and show how it makes our model efficient. MobiVSR achieves an accuracy of 70% on a challenging Lip Reading in the Wild dataset with 6 times fewer parameters and 20 times smaller memory footprint than the current state of the art. MobiVSR can also be compressed to 6 MB by applying post training quantization.

Index Terms: video speech recognition,

1. Introduction

Visual speech recognition (VSR) is the task of recognizing spoken language from video input only, without any audio. Similar to ASR (audio speech recognition), VSR has a multitude of applications. In general, VSR can be used to augment/replace audio speech recognition for situations where speech cannot be heard or produced - for example, if a person has a laryngectomy or voice-box cancer, or in a situation where one needs to understand a speaker in a noisy environment. Thus, the broader aim of this work is to make VSR technology deployable, especially in mobile environments (such as cars) and on hand-held devices (as an assistive technology).

Recent research in VSR has focused primarily on either increasing recognition accuracy [1] or reconstructing speech [2]. The application of deep learning techniques has produced models that perform substantially better on lip reading datasets than earlier methods [3, 4]. A major hindrance in deployability is that these models have prohibitively large memory and energy requirements (see Table 1), and their running times are unacceptable for real-time user facing applications. The state-of-the-art model for word level VSR [5] uses a novel architecture that incorporates a 3D convolution kernel as a front-end to extract features from the video stream and a residual architecture [4] on top of it for predicting the word spoken. The architecture has more than 25 million parameters, occupies 130 MB of disk space and involves 290 million FLOPs for inference. Memory and computation of this order is prohibitively expensive for mobile devices. For instance, the Apple store places a hard limit of 150 MBs on a fully functional app that should include all its components. Furthermore, according to empirical observations made on iOS [6], an app taking more than 50% of the total RAM available at runtime, often crashes. Performing inference over such models requires significant amounts of energy due to memory access. It has been shown [7] that on an Intel 45nm based system accessing DRAM consumes $\approx$ 2500 times more energy than floating point addition and therefore dominates energy expenditure. Memory access demands depend on the number of parameters and the intermediate results generated during a forward pass of the neural network, both of which are quite high in all VSR models (Table 1). Therefore, battery drain would be a significant issue with these models.

In this paper we present MobiVSR, a lip reading system which achieves competitive accuracy in visual speech recognition and at the same time is suitable for use on resource-constrained devices. On the challenging Lip Reading in the Wild dataset (LRW) [1], MobiVSR attained accuracy of 70% which is between the accuracy of the baseline neural models (65%) given in [1] and that of the best performing model on this dataset (83%) [5] but is faster, smaller and less memory and compute intensive than either. Concretely, it uses $\approx$ 2 times fewer parameters than the model from [1] and $\approx$ 6 times fewer than the state of the art model on LRW [5]. The techniques used in MobiVSR are independent from and complement model compression techniques [8, 9, 10]. For example, using 8-bit non-uniform quantization [11], we could further compress MobiVSR to 6MB.

The main contributions that we make in this paper are:

- We present the **MobiVSR architecture**, which for the first time, addresses the problem of deploying visual speech recognition models on resource constrained devices.
- We show the applicability of **depthwise 3D convolution with channel shuffle** for the first time in the VSR domain. This technique helps us to reduce the parameter count and computational complexity vis-a-vis standard convolution.
- We use novel strategies for **reducing size and parameter**

*Equal Contribution

http://dx.doi.org/10.21437/Interspeech.2019-3273

count of our trained neural network model.

• We show that MobiVSR achieves accuracy of 70% in spite of having $6\times$ **fewer parameters and a** $20\times$ **smaller model than the state-of-the-art model**. Using additional parameter quantization techniques, MobiVSR's size can be reduced to 6MB.

2. Related Work

Very little work on VSR (as a video classification task) has focused on developing efficient architectures; however, there is work on this task in image classification and in ASR. The problem of efficient architecture development for image classification was introduced in [12]; SqueezeNet achieves accuracy on par with AlexNet [13] but uses far fewer parameters, by using convolutional blocks having 3×3 convolutions followed by 1×1 instead of the large 5×5 kernel used in AlexNet. MobileNet [14] uses depthwise-separable convolution [15, 16], for parameter reduction. The MobileNet-V2 architecture [17] improves MobileNet by adding residual connections within the MobileNet depthwise-separable modules. Shufflenet [18] further develops the architecture by replacing pointwise convolution in a depthwise separable layer by a channel shuffle operation, which performs the function of inter channel information exchange without increasing the number of floating point operations. This idea was a major influence in the design of MobiVSR (Section 3).

In the domain of Automatic Speech Recognition (ASR) from audio, a major contribution was PocketSphinx [19], a large vocabulary, speaker-independent continuous speech recognition engine suitable for hand-held devices. Some authors [20] have tried to construct an acoustic model by combining simple recurrent units (SRUs) and depth-wise 1-dimensional convolution layers for multi time step parallelization; this results in reductions in DRAM access and increase in processing speed, allowing real-time on-device ASR on mobile and embedded devices.

Another approach in developing efficient deep learning methods is to redesign computationally expensive layers. For example, [21] replace standard convolution with a 'shift' layer that consumes zero flops during inference.

A complementary solution for making deep neural networks suitable for embedded devices is to compress the model post training. This method doesn't require significant changes in architectural design. Notable examples of this approach include hashing [22], quantization [8] and factorization [23].

3. MobiVSR: End-to-end Lip Reading with Fewer Parameters

This section describes the architecture of MobiVSR along with the explanations behind its design choices.

3.1. MobiVSR Architecture

The MobiVSR architecture is shown in Figure 2. It maps visemes (basic units of visual speech) to graphemes (*i.e.*, characters/words). It can be divided into 3 parts: (1) a front-end three dimensional convolution part whose function is to extract low level features from visemes; (2) a middle stack of variable sized residual subgraphs whose function is to use those low level features to infer high level features; and (3) a backend consisting of temporal convolutions whose function is to integrate the high-level features to get graphemes out of visemes. Finally, there are two fully connected neural network layers that output class probabilities, thus converting abstract grapheme predictions to their probabilities.

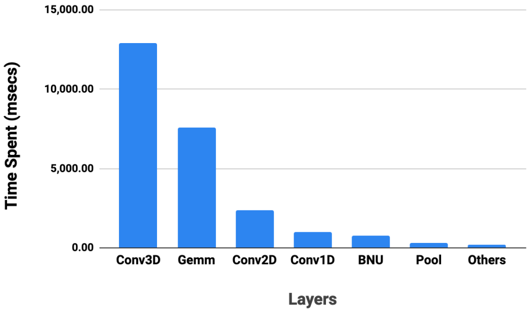

Figure 1: *Time spent per layer of the lip reading architecture proposed in [5], profiled using cProfile.*

The front-end part of MobiVSR consists of two 3D depthwise separable convolution layers. We use two depthwise separable layers with kernel size $3\times3\times3$. Each of these layers downsizes the input along spatial dimension by half; we implement this layer by using $3\times3\times3$ group convolution, with the number of groups being equal to the number of input channels followed by a channel shuffle operation in order to ensure flow of information across channels.

The middle stack of MobiVSR is subdivided into four subgraphs. Each subgraph has an α number of **residual blocks** which we call **LipRes** blocks. α is a hyperparameter which one can vary to change the depth of the model. The intuition behind keeping these residual subgraphs is to increase the prediction power of the model. Through experimentation, we observed that that LipRes block, in our network, serves as an atomic layer. As we increase the depth of the model by increasing α (number of stacked LipRes Blocks), we increase the accuracy but at a cost of making the model heavier. Thus, we realized that the LipRes block could be leveraged to balance accuracy and model size. The results pertaining to these are shown in Table 1, where increasing α increases accuracy but at the cost of using more parameters and vice-versa. Thus, α makes MobiVSR flexible enough for different applications and environments.

The backend of the model is used to integrate features across time and provide word probabilities. It uses two temporal convolution layers with a maxpool downsampling sandwiched in between. For performance reasons as outlined above we do not use RNN layers, in contrast to [5]. Finally MobiVSR uses fully connected layers with softmax activation to generate class probabilities for the 500 graphemes in the LRW dataset.

Architecture Choices - In general, the methods we used to develop MobiVSR are influenced by the motivation of reducing the memory and energy footprint while maintaining accuracy. We point out a list of challenges in this direction along with the strategies that we take in order to solve them.

Challenge 1. *3D convolution is the most compute intensive layer during inference; see Figure 1, which shows the average percent of inference time spent per layer in the state-of-the-art lip reading system [5].*

Optimize 3D Convolutions : 3D convolution is a front-end technique in video processing tasks since it can combine information across both time and space [5]. Doing away with it deteriorates model accuracy. Therefore optimizing 3D convolution becomes highly important. Inspired by [18], where they converted 2D convolution into a sum of depthwise convolution and channel shuffle operations, we generalize 3D convolution

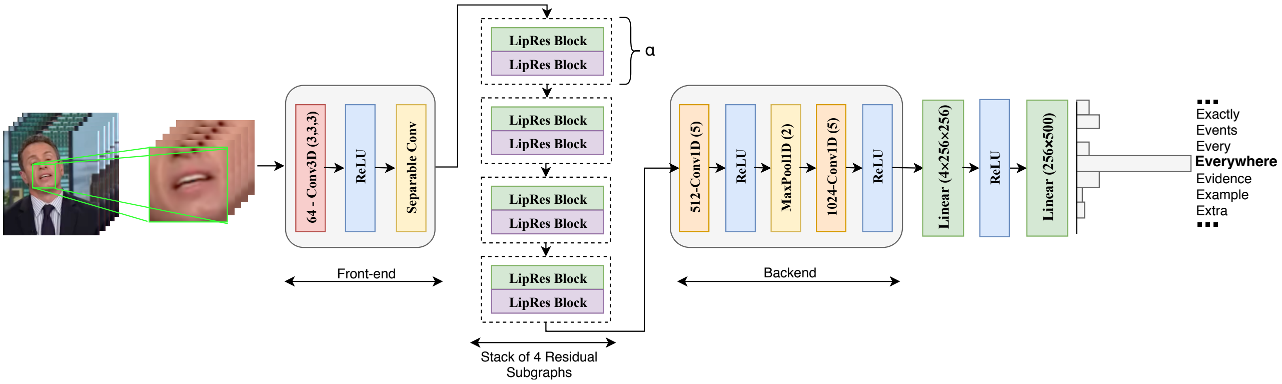

Figure 2: *MobiVSR architecture*

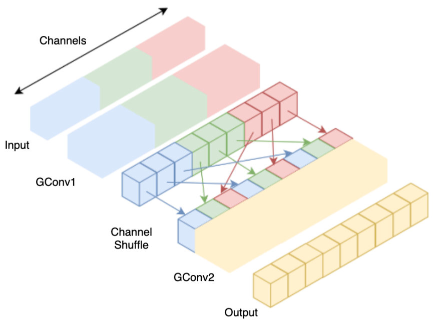

Figure 3: *LipRes block: Channel shuffle with group convolutions (GConv)*

and use this to optimize the front end of the network in our architecture. To the best of our knowledge, this is the first time depthwise convolution along with channel shuffle is being used in visual speech recognition or even in video recognition problems. As shown in [24], the reduction in number of FLOPs and model size is significant while using 3D depthwise convolution as opposed to normal convolution operations of the baseline and state-of-the-art models (Table 1).

Challenge 2. *The sequential nature of recurrent neural network (RNN) calculations due to their time (non-parallelizable operations) and memory requirements are a major performance bottleneck during inference [25].*

Avoid RNNs : Recent research indicates that temporal convolutions can be used in place of RNNs without significant loss of accuracy [26]. Temporal convolutions offer several advantages. They increase parallelism since convolution can process multiple time-steps at once. They also have flexible receptive fields [27], and can control model memory usage. Therefore we use 1-D temporal convolution in place of a RNN for modeling temporal features.

Challenge 3. *Since parameter count and memory calls are dependent on model size/complexity, use neural layers sparingly.*

Reduce Parameter Count in Convolution Filters : First, since the kernel parameter count increases quadratically with the kernel size, we use small convolution filter sizes of 3 × 3 in MobiVSR. We gain an additional boost due to modern deep learning frameworks which use algorithms that optimize the number of operations required for a convolution operation [28] with a small filter size [29, 30]. Second, we use the depthwise convolutions with channel shuffle [18], for reducing the time and computational complexity of the model.

Use Residual Connections : Since increasing the depth of a deep network to increase accuracy adds additional computational complexity and more memory calls, we use residual connections (ResNet blocks) as suggested in [4]. These connections are used extensively inside the LipRes block as shown in Fig. 4. LipRes has a residual structure similar to the ResNet blocks. Each block consists of depthwise convolutions and ReLU [31] non linearity along with the channel shuffle operation, and a parallel skip connection. The use of depthwise convolutions and residual connections helps to reduce the parameter count and cut memory calls. The channel shuffle operation helps in inter-channel information exchange. The skip connection also has a convolution layer with stride two whenever the output is supposed to be spatially down-sampled (Figure 4(2)).

Challenge 4. *One size does not fit all.*

Introduction of α: We believe '*one size does not fit all*' and that the user/designer of a model should be given a handle to trade off between accuracy and efficiency. That is why we introduce α as explained above.

Additionally, we use batch normalization [32] for regularization. However, contrary to the common paradigm of using batch normalization after every convolution kernel, we use batch normalization scantily as the smaller size of the network has a regularizing effect during training.

3.2. Data

We base our experiments on the large publicly available speaker-independent Lip Reading in the Wild (LRW) database [1]. The LRW database contains 1000 utterances each for a collection of 500 different graphemes in the training set. For testing and validation, the dataset has 50 video clips per grapheme. Each video is challenging because of the high variance of head pose and illumination; therefore, in addition to being one of the few VSR data sets of size, LRW is a good proxy for mobile lip reading data. The video clips are of 29 frames (1.16 seconds) in length, and the speaker mouth region of interest (ROI) is placed at the center of each frame.

A set of 29 consecutive frames (256 × 256 pixels) is sampled from each video of the LRW Dataset. We then extract the mouth ROI from these RGB frames. As the LRW Dataset is face centered, we achieve this step by cropping a 96 × 96 pixel window image segment from the center of each frame. Finally, the

Table 1: *Comparison of accuracy, computational complexity, and memory footprint of MobiVSR (with different α), with the LRW Baseline and the state-of-the-art model.* ***Note:*** *MobiVSR-1 denotes the model with $\alpha = 1$*

Model	**Size (in MB)**	**Parameter count (in millions)**	**Memory access (in thousands)**	**FLOPs (in billions)**	**Top-1 accuracy**	**Top-3 accuracy**
MobiVSR-1	**14.2**	**4.4**	**34.9**	**10.75**	70.1	88.0
MobiVSR-2	16.19	5.1	37.1	19.6	70.8	89.0
LSTM + ResNet (SOTA)	130.0	25.1	56.3	290	83	99.8
LRW Baseline	43.2	8.7	44.0	95.7	61.0	78.0

Table 2: *Number of memory access and FLOPs associated with different layers. V_i and V_o are the input and output volume respectively. C_i and C_o are the input and output channel dimensions. $K \times K$ is the 2D convolution kernel while $K \times K \times T$ is the 3D convolution kernel*

Layer	Memory Access	Floating point operations (FLOPs)
Conv2D	$K^2C_iC_o + V_i \cdot (K^2Co) + V_o$	$2(K^2C_i) \cdot V_o$
Conv3D	$K^2TC_iC_o + V_i \cdot (K^2C_o) \cdot T + V_o$	$2(K^2TC_i) \cdot V\_o$
Depthwise Conv2D	$C_i \cdot K^2 + V_i \cdot (K^2 + C_o) + V_o$	$2C_i \cdot (\frac{K^2}{C_o}) \cdot V_o$
Depthwise Conv3D	$C_i \cdot K^2 \cdot T + V_i \cdot (K^2 + C_o) \cdot T + V_o$	$2C_i \cdot T \cdot (\frac{K^2}{C_o}) \cdot V_o$
Fully Connected	$IQ + V_i + V_o$	$2IQ$

cropped frame segments are converted to gray scale and stored as numpy matrices [33]. This numpy matrix is then fed to all networks as input.

3.3. Experiments

We train MobiVSR on a NVIDIA Titan X GPU for 50 epochs using different settings for α (1 and 2). The results are summarized in Table 1. To increase accuracy, one can increase the number of LipRes blocks by increasing the value of α. On the other hand one can get a smaller and more efficient model at the cost of some accuracy by reducing α. We compare MobiVSR with other word-level lip reading models on saved model size, number of parameters, memory required during inference and number of floating point operations (FLOPs). To ensure consistency in comparing model sizes, we converted each model to ONNX format. We calculate inference speeds on an Intel i3 processor and average over 5000 runs. The calculations of memory accesses and number of floating point operations in different layers are described below and summarised in Table 1. We ignore the effects of applying non-linearities, batch-normalization and bias terms in these calculations as their contributions are very small compared to those of matrix multiplications and convolutions.

4. Qualitative Analysis of Results

While it is essential for the model to focus on parameters like memory access, size, parameter count, *etc*, it is also essential for us to analyse what are the strengths and weaknesses of the model. With this in mind, we analysed the performance of the model by looking at some specific cases.

While the model in general performs well, we observed a few interesting failure cases as well. We found that graphemes which have common visemes are quite often confused. For example, *'bring'* and *'being'* share (in order) the visemes {*E,V4,H*} and disagree only on the second viseme: {*A*} for 'bring' and {*V4*} for 'being' [34]. The distinguishing visemes are spoken from within the mouth, so are difficult to capture using a camera. The model confuses these graphemes 80% of the time. Similarly confusable grapheme pairs are {*billions, millions*}, {*having, heavy*}, {*general, several*}, *etc* (interested readers may find a direct mapping of all such confusing words with illustrations from some sample videos in the appendix attached). Now, consider cases where the model predicts the correct class most of the time: {*leadership, energy, later, better*}. These all lack to a lack viseme-similar counterparts in the dataset. In general, we found that MobiVSR fails due to signals which cannot be captured using a camera only; however, when coupled with an audio-recording device, signals from the complementary modalities can potentially help.

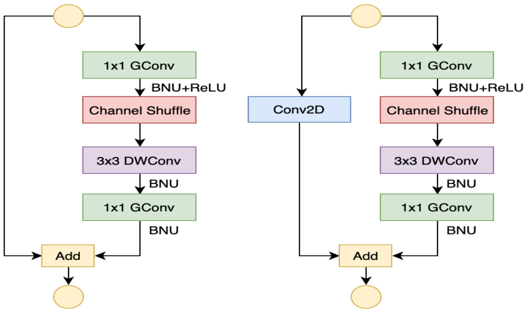

Figure 4: *LipRes Blocks: (1) The first LipRes block is for keeping the size of the input constant. It is used in the first subgraph in MobiVSR. (2) The second LipRes block is for halving the size of the input. It is used in the second, third and fourth subgraphs in MobiVSR. Thus, increasing alpha by 1, increases the first LipRes block by one and the second LipRes block by 3.*

5. Conclusions and Future Work

In this paper we introduced MobiVSR, a deep neural network model designed to perform word level visual speech recognition in resource constrained devices. We showed how MobiVSR uses 6× fewer parameters than the state-of-the-art model and can be compressed to 6MB after quantization. Moreover it can be modified using a tuneable hyperparameter to balance accuracy and efficiency for different use cases. As mentioned earlier, mobile-centric lip reading systems have enormous utility in the society. We hope that this paper inspires other researchers to create similar and even more efficient models considering the social impact such applications can have.

6. Acknowledgement

This research was supported in part by SERB, Department of Science and Technology, Government of India under grant number SRP 124.

7. References

[1] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. In *Proceedings of the Asian Conference on Computer Vision*, 2016.

[2] Yaman Kumar, Rohit Jain, Mohd Salik, Rajiv ratn Shah, Roger Zimmermann, and Yifang Yin. Mylipper: A personalized system for speech reconstruction using multi-view visual feeds. In *Proceedings of the IEEE International Symposium on Multimedia (ISM)*, 2018.

[3] Stavros Petridis, Yujiang Wang, Zuwei Li, and Maja Pantic. End-to-end multi-view lipreading. *arXiv preprint arXiv:1709.00443*, 2017.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[5] Themos Stafylakis and Georgios Tzimiropoulos. Combining residual networks with LSTMs for lipreading. *arXiv preprint arXiv:1703.04105*, 2017.

[6] StackOverflow. iOS - experiments for maximum runtime memory accesses allowed, 2019.

[7] Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2014.

[8] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. *arXiv preprint arXiv:1802.05668*, 2018.

[9] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Proceeedings of the European Conference on Computer Vision*, 2016.

[10] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[11] TensorFlow. Introducing the model optimization toolkit for TensorFlow, 2018.

[12] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and $<$ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[14] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[15] Laurent Sifre. *Rigid-motion scattering for image classification*. PhD thesis, Ecole Polytechnique, CMAP, 2014.

[16] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[17] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. *arXiv preprint arXiv:1801.04381*, 2018.

[18] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[19] David Huggins-Daines, Mohit Kumar, Arthur Chan, Alan W Black, Mosur Ravishankar, and Alexander I Rudnicky. Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 2006.

[20] Jinhwan Park, Yoonho Boo, Iksoo Choi, Sungho Shin, and Wonyong Sung. Fully neural network based speech recognition on mobile and embedded devices. In *Advances in Neural Information Processing Systems*, 2018.

[21] Bichen Wu, Alvin Wan, Xiangyu Yue, Peter Jin, Sicheng Zhao, Noah Golmant, Amir Gholaminejad, Joseph Gonzalez, and Kurt Keutzer. Shift: A zero flop, zero parameter alternative to spatial convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[22] Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *Proceedings of the International Conference on Machine Learning*, 2015.

[23] Vadim Lebedev, Yaroslav Ganin, Maksim Rakhuba, Ivan Oseledets, and Victor Lempitsky. Speeding-up convolutional neural networks using fine-tuned CP-decomposition. *arXiv preprint arXiv:1412.6553*, 2014.

[24] Rongtian Ye, Fangyu Liu, and Liqiang Zhang. 3d depthwise convolution: Reducing model parameters in 3D vision tasks. *arXiv preprint arXiv:1808.01556*, 2018.

[25] James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. Quasi-recurrent neural networks. *arXiv preprint arXiv:1611.01576*, 2016.

[26] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[27] Khoi-Nguyen C Mac, Dhiraj Joshi, Raymond A Yeh, Jinjun Xiong, Rogerio R Feris, and Minh N Do. Locally-consistent deformable convolution networks for fine-grained action detection. *arXiv preprint arXiv:1811.08815*, 2018.

[28] Kumar Chellapilla, Sidd Puri, and Patrice Simard. High performance convolutional neural networks for document processing. In *Proceedings of the Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.

[29] Michael Mathieu, Mikael Henaff, and Yann LeCun. Fast training of convolutional networks through FFTs. *arXiv preprint arXiv:1312.5851*, 2013.

[30] cuDNN.

[31] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning*, 2010.

[32] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[33] Stfan van der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13:22 – 30, 05 2011.

[34] Chalapathy Neti, Gerasimos Potamianos, Juergen Luettin, Iain Matthews, Herve Glotin, Dimitra Vergyri, June Sison, Azad Mashari, and Jie Zhou. Audio-visual speech recognition. Technical report, Center for Language and Speech Processing, The Johns Hopkins University, Baltimore, MD, 2000. Final Workshop 2000 Report.