Using Deep Convolutional LSTM Networks for Learning Spatiotemporal Features

Logan Courtney and Ramavarapu Sreenivas

University of Illinois, Urbana-Champaign IL 61820, USA
{courtne2,rsree}@illinois.edu

Abstract. This paper explores the use of convolution LSTMs to simultaneously learn spatial- and temporal-information in videos. A deep network of convolutional LSTMs allows the model to access the entire range of temporal information at all spatial scales. We describe our experiments involving convolution LSTMs for lipreading that demonstrate the model is capable of selectively choosing which spatiotemporal scales are most relevant for a particular dataset. The proposed deep architecture holds promise in other applications where spatiotemporal features play a vital role without having to specifically cater the design of the network for the particular spatiotemporal features existent within the problem. Our model has comparable performance with the current state of the art achieving 83.4% on the Lip Reading in the Wild (LRW) dataset. Additional experiments indicate convolutional LSTMs may be particularly data hungry considering the large performance increases when fine-tuning on LRW after pretraining on larger datasets like LRS2 (85.2%) and LRS3-TED (87.1%). However, a sensitivity analysis providing insight on the relevant spatiotemporal temporal features allows certain convolutional LSTM layers to be replaced with 2D convolutions decreasing computational cost without performance degradation indicating their usefulness in accelerating the architecture design process when approaching new problems.

Keywords: Convolutional LSTMs · Deep Learning · Video Analytics · Lip Reading · Action Recognition

1 Introduction

Learning from video sequences requires models capable of handling both spatial and temporal information. Due to the advent of large image datasets such as ImageNet [25], there has been significant progress in the development of convolutional-based architectures for learning spatial features [20][29][15][34]. It is not surprising that almost all methodologies for video sequences revolve around convolutional networks combined with additional temporal elements.

The origin of video based learning begins primarily with the task of human action recognition. [17] saw success by stacking frames together before passing them through a pretrained convolutional network as well as processing frames individually with some variant of temporal pooling applied to the network output. [28] saw larger improvements by passing RGB and optical flow images through pretrained networks. [22] leveraged the success of recurrent neural networks by using a stack of LSTMs[16] to process the

outputs of a convolutional network. All of these methods for handling the temporal information used pretrained convolutional networks. The spatial and temporal features were, in some sense, handled separately.

[35] first explored the use of 3D convolutions for processing spatial and temporal information together. That is, the network was capable of learning spatiotemporal features. Research has continued predictably along this path with large action recognition datasets such as the Kinetics dataset [18]. [14] replaced 2D convolutions in common image-based architectures such as ResNet with 3D convolutions. [6] was able to expand the filters from a pretrained 2D network to a 3D convolution network capturing the benefits of the extra training data from ImageNet.

Lipreading is a technique for understanding speech using only the visual information of the speaker. It is a well-structured problem for looking into how deep networks learn spatiotemporal features. For a problem like action recognition, the current datasets have classes that can be identified from a single image alone (e.g. playing baseball versus swimming). In such instances, the temporal information is less important than the spatial information reducing the necessity for architectures capable of directly handling spatiotemporal features. High performing models still separate the spatial and temporal learning. [5] temporally post-processes learned features from a pretrained Inception-ResNet-v2 [33] achieving higher performance than any of the 3D convolution architectures from [14] on Kinetics.

On the other hand, lipreading from a single frame within a sequence provides little information about what is being said. It is necessary to utilize the temporal context. Lipreading certainly aligns with the concept of a spatiotemporal problem. This research explores the use of convolution LSTMs for lipreading. That is, 2D convolutions are used within the LSTM structure providing the network with the capacity to learn features at many combinations of spatial and temporal scales. The primary contributions of the paper are as follows.

- Successfully trains the first very deep network built primarily with convolutional LSTMs and achieves competitive performance on the Lip Reading in the Wild dataset [8]
- Convolutional LSTMs see the same improvements as 2D convolutions see in image classification tasks when upgrading from architectures like VGG to ResNet. This is similar to what [14] demonstrated for 3D convolutions.
- Presents an analytical technique providing insight on what spatiotemporal features are relevant to a particular problem and demonstrates how this information can be used to facilitate the architecture design process (Section 5.1)
- The model utilizes 2D convolutions along with convolutional LSTMs demonstrating the ability to intermix temporally capable modules with spatial-only processing modules to reduce the number of parameters and total computation without sacrificing performance

Section 2 reviews related work along with discussing the basics of spatiotemporal features. Section 3 describes our proposed methodology and model architectures. Section 4 describes the experiments and implementation. Section 5 discusses our results and provides empirical evidence explaining the similar performance between convolutional LSTM models and competing methods.

2 Background and Related Work

2.1 Spatiotemporal Features and Receptive Field

Each layer within a convolutional network has an output which can be interpreted as an image with channels made up of a prescribed number of features and a corresponding height/width related to the spatial dimensions. Spatial pooling and/or strided-convolutions are used throughout the network to reduce the height/width of these feature maps. This reduces computational costs due to fewer convolution operations per layer and increases the visual receptive field. Each pixel of the output is calculated based on a larger portion of the original input image. Additionally, it allows for deeper (i.e. more layers) as well as wider (i.e. more channels) networks which have been shown to increase the network's capability to learn complex visual tasks.

However, with each spatial pooling layer and/or strided convolution layer, a certain degree of spatial information is inevitably lost due to the reduction in resolution of the output. This can impede a network's ability to learn functions capable of discriminating high resolution visual information. The choice of when to apply spatial dimension reduction techniques within the network remains an "art-form" in the hands of the network designer. An application requiring the classification of large objects can utilize more spatial reduction layers to increase the visual receptive field without discarding relevant information. Using spatial reduction layers too early in the network may prevent the network from being able to detect the precise location of objects taking up a small portion of the input image. The receptive field has been well studied in the past [21].

For 1D sequence problems, such as those seen in natural language processing (NLP), there are typically two techniques used to capture sequence information: convolutions and recurrent neural networks. A series of 1D convolutions applied to sequence data gradually increases the temporal receptive field. This gradual exposure of temporal information with deep convolutional networks works well for tasks like document classification [10]. On the other hand, each layer of a network made up of LSTMs has access to the full sequence. That is, the temporal receptive field when calculating the output at a particular timestep includes all previous inputs. This technique works well when training language models for predicting the next word in the sequence [11].

Dealing with sequences of images creates an additional challenge due to the temporal information appearing at multiple spatial scales. Our work is meant to motivate a principled approach for applications that depend on detecting these spatiotemporal features. It blends methods that (a) involve just images and (b) involve just time series data. So far, methods for applications like action recognition or lipreading have utilized a straightforward recipe of using 3D convolutions together with time-tested techniques from the two individual fields. As shown in [35] for action recognition, utilizing a 3D convolution ResNet50 over a 2D convolution ResNet50 model with an LSTM shows an increase from 68.0% to 72.2%. Utilizing the temporal information after the final layer of a 2D convolution network may be too late to capture the relevant spatiotemporal features. In [31] for lipreading, including a 3D convolution at the front of the network sees an increase of 5.0% compared to the model without early temporal processing. At first glance, it may seem that any architecture capable of processing spatial and temporal features together is all that is needed.

However, related events may be separated by temporal gaps. 2D convolutions work with images due to the assumed symmetry between spatial dimensions and local connectivity of pixels. There is no inherent reason for treating the temporal dimension as an extra spatial dimension as is done with 3D convolutions.

Additionally, there are performance discrepancies with the use of 3D convolutions. In action recognition, [35] saw success by using a model with the temporal receptive field increasing at the same rate as the spatial receptive field. Lipreading[8] saw this same method achieve a 10% lower classification rate when compared to processing all of the temporal information at a single spatial scale. Between the two high performing models, there was an additional 4.0% improvement when the temporal processing was delayed until a later spatial scale. These discrepancies suggest different applications contain different spatiotemporal features and performance is dictated by which spatiotemporal features the model is designed to handle.

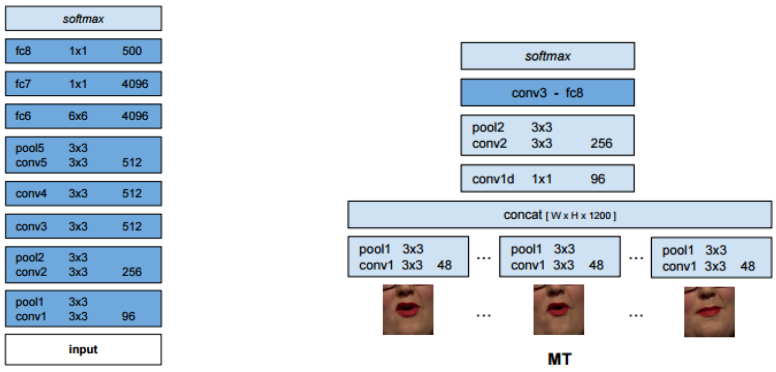
If the spatial resolution of the input doubled, would the model need to adapt the location of the temporal processing? If the temporal resolution of the input doubled, would a larger kernel size or more 3D convolutions at a particular spatial scale be necessary? There are many unanswered questions with both 3D convolutions and spatiotemporal features. As much as deep learning is about creating networks capable of learning generalized features relevant to a particular problem, it would be advantageous to create architectures capable of generalizing well across problems. This is the main focus of our work. The architecture presented in this paper and the empirical results suggest convolutional LSTMs hold promise in learning the spatiotemporal features relevant to the dataset without having to cater the design of the network in a particular way.

2.2 Lipreading in the Wild (LRW) Dataset

The Lip Reading in the Wild (LRW) [8] dataset consists of 500,000 videos with 29 frames each taken from BBC TV broadcasts. There are 500 target words (1,000 videos for each word) with 50 videos of each word for both the validation and test sets. There are often context words surrounding the target word. Although there are ambiguous classes (e.g. “weather” and “whether”), it is sometimes possible to distinguish between these based on the surrounding context. The videos are centered on the speaker with the speaker facing the camera.

2.3 Lipreading Sentences in the Wild (LRS2/LRS3-TED) Datasets

The LRW dataset has a constrained vocabulary with fixed input size making it a well structured sequence classification problem. The Lip Reading Sentences (LRS) [7] dataset contains variable length sequences with unique words appearing in the test set unseen in the training set. Additionally, there are moments of off-angle facial views up to 90°(side profile). The original LRS dataset is unreleased to the public and the LRS2 dataset [1] is used in its place. The LRS3-TED [3] dataset, similar in structure to LRS2, is made from TED talks as opposed to BBC TV Broadcasts.



(a) 2D VGG-M (b) Multitower (MT) VGG-M

Fig. 1: Modifications of the VGG-M model (left) were used in [8] with the release of the LRW dataset. The multitower (MT) model (right) set the initial benchmark at 61.1% by processing each frame individually with a 2D convolution before concatenating the features for temporal processing.

2.4 Related Work

The original work on LRW [8] tested multiple variations of the VGG-M model (seen in Figure 1a). Their highest performing model (seen in Figure 1b) processed each frame of the video with a 2D convolution before concatenating the outputs allowing the remainder of the spatial processing to have access to the full temporal receptive field. This outperformed the networks utilizing 3D convolutions which gradually increased the spatial and temporal receptive fields as the input passed deeper into the network.

The original work on LRS [7] used a similar VGG-M based model processing five frames at a time. This network acted as a sliding window across the sequence providing a separate output at each timestep. Long-term temporal information was managed by an LSTM encoder-decoder network with attention[37] to spell words one character at a time. The model was then fine-tuned on LRW achieving 76.2%. The 15.1% improvement over the model in Figure 1b is due to some combination of the temporal receptive field and extra training data.

[31] replaced the VGG-M portion of the network with the deeper 34-layer ResNet [15]. This network (shown in Figure 2) is separated into three parts: a spatiotemporal front-end with a 3D convolution, ResNet34 for processing the remaining spatial information, and a bidirectional LSTM. It was state of the art with 83.0% before being surpassed by a model replacing ResNet34 with ResNet18[30] achieving 84.3%.

[2] uses the same 3D+ResNet spatiotemporal front-end while testing three back-end models for sequence transcription on LRS2. The self-attention based Transformer [36] outperformed the fully convolutional back-end and the bidirectional LSTM back-end. A continuation of the work[1] compares performance of the Transformer back-end when trained as a seq2seq [32] model versus training with the CTC loss [12]. The seq2seq

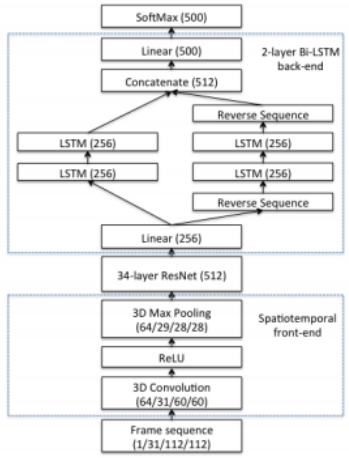


Fig. 2: 3D+ResNet+BiLSTM model from [31] sets the benchmark on LRW at 84.3% when ResNet34 is replaced with ResNet18[30].

Transformer set the benchmark at 48.3% word error rate (WER) for LRS2 and 58.9% WER for LRS3-TED.

The above lipreading sentences work all use the same 3D+ResNet34 spatiotemporal front-end model and only compare back-end performance. The front-end model is pretrained on short sequences from the LRW and LRS datasets using the technique in [9]. The back-end is trained on extracted frozen features from the front-end. The results illustrate the importance of long-term context and language modeling when lipreading sentences. [1] shows over a 12% reduction in WER when testing on phrases containing more than three words and the Transformer model contains over three times as many parameters (65 million) as the spatiotemporal front-end (21 million). No improved results were reported for the LRW dataset.

This is mentioned to emphasize the convolutional LSTM models explored here can be used in conjunction with these techniques by swapping out the spatiotemporal front-end. The work here focuses on the spatiotemporal learning.

3 Proposed Technique

3.1 Convolution LSTM

The convolutional LSTM calculates an internal cell state c_t (cf. equation 1 below) and a hidden state h_t utilized as the output to subsequent layers and for state-to-state transitions. While processing a sequence of frames, c_t and h_t can be viewed as images of appropriate size maintained by the network with relevant information based on what it has seen in

the past. Learnable filters $W_\bullet$ with bias terms $b_\bullet$ are used to handle a new frame x_t along with the past information h_{t-1} being used by learnable filters $U_\bullet$.

$$\begin{aligned} f_t &= \sigma(W_f * x_t + U_f * h_{t-1} + b_f) \\ i_t &= \sigma(W_i * x_t + U_i * h_{t-1} + b_i) \\ o_t &= \sigma(W_o * x_t + U_o * h_{t-1} + b_o) \\ c_t &= f_t \circ c_{t-1} + i_t \circ tanh(W_c * x_t + U_c * h_{t-1} + b_c) \\ h_t &= o_t \circ tanh(c_t) \end{aligned} \tag{1}$$

Convolutional LSTMs have been used in the past in rather limited ways. [26] demonstrated success with predicting future precipitation maps. A shallow two-layer model is used at a single spatial scale. [39] explored learning spatiotemporal learning for gesture recognition using 3D convolutions followed by two layers of convolutional LSTMs for longer context. [13] used a bidirectional convolutional LSTM on the output of a 2D VGG13 [29] network to detect violence in videos. The majority of the spatial processing occurs before reaching the convolutional LSTM layers. The datasets for these applications are relatively small which may explain convolutional LSTMs limited use due to their tendency to overfit [38]. However, recent large-scale datasets are providing an opportunity to explore their use in new ways.

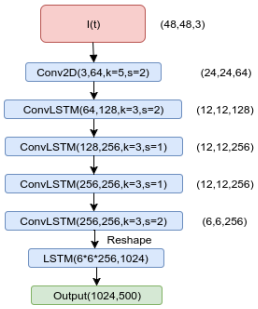
There is a fundamental difference between the temporal receptive of convolutional LSTMs and 3D convolutions. Successive 3D convolutions inherently limit long-term temporal information from being processed until deep in the network after the spatial dimension has been reduced. All layers in a network of convolutional LSTMs will have access to the full sequence at all spatial scales. The models here incorporate convolutional LSTMs throughout the entirety of the network and they demonstrate the existence of spatiotemporal features at multiple scales.

3.2 VGG-M ConvLSTM

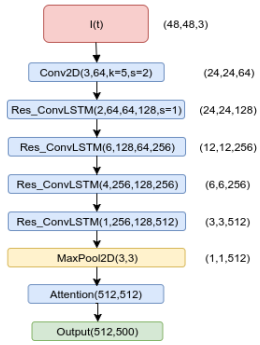
The first convolutional LSTM model (3a) is based on the VGG-M architecture seen in Figure 1a. The number of output channels has been reduced to compensate for the larger number of weights in convolutional LSTM layers. The first layer is a 2D convolution which matches the highest performing multitower (MT) model from [8](Figure 1b).

3.3 ResNet ConvLSTM

Certain convolution layers can be replaced with convolutional LSTM layers to extend the ResNet architecture to learn spatiotemporal features. The model is shown in Figure 3b. The four parameters of a **Res_ConvLSTM()** module represent the number of sub-blocks, the number of input channels, the number of intermediate channels, and the number of output channels. For any **Res_ConvLSTM()** module, the first sub-block is always of type A (shown in Figure 4a) with the remaining sub-blocks of type B (shown in Figure 4b). This is similar to the technique used in [14] to replace the 2D convolutions in ResNet with 3D convolutions.

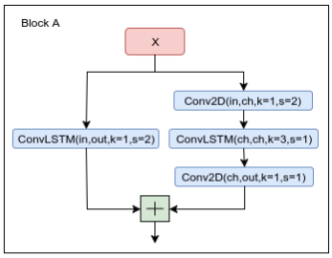


(a) Baseline convolutional LSTM architecture modeled after the VGG-M architecture used in [8].

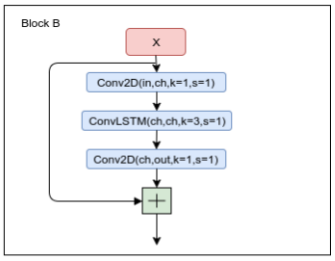


(b) A residual based convolutional LSTM network. Each **Res_ConvLSTM()** is made up of multiple sub-blocks shown in Figure 4.

Fig. 3: Two different convolutional LSTM architectures. The output dimensions are shown in the margins.



(a)



(b)

Fig. 4: Block type A contains two convolutional LSTMs with one of them in the path of the skip connection. This block has a stride of 2 for reducing the spatial dimensions. Block type B contains one convolutional LSTM with an open skip connection.

There are two variants. The first uses an LSTM in place of the attention mechanism for training on the LRW dataset. The attention network [37] is used when pretraining on the LRS datasets with unconstrained vocabularies. The network is modified to output a character (or blank token) after every frame in order to spell words and uses the CTC loss function[12]. The convolutional LSTM based ResNet model has 14.5 million parameters. The bidirectional version has 29 million parameters which is roughly equivalent to the 23.5 million parameters of the competing model shown in Figure 2. There are 48 total layers: 14 3x3 convolutional LSTMs, 4 1x1 convolutional LSTMs, 28 1x1 2D convolutions, and 1 LSTM before the final output layer.

3.4 Reduced ResNet ConvLSTM

The importance of temporal information in the context of lipreading varies depending on the spatial scale within the network. The model in Figure 3b is modified by replacing select convolutional LSTMs determined to be irrelevant by the sensitivity analysis (see Section 5.1) with 2D convolutions. This reduces both the number of parameters and total computation without losing the necessary representation power for high performance. The convolutional LSTM layers at scales 12×12 and 6×6 are replaced due to the hidden-to-hidden connections providing relatively small amounts of useful information here. The 3×3 scale is removed due to the temporal back-end's ability to capture the long-term context after the spatial resolution has been fully collapsed.

4 Experiments

4.1 Training Convolutional LSTMs on LRW

A random crop around the mouth between 48 and 64 pixels is attained for each frame and resized to a 48x48 input. During training, the frames are randomly flipped, randomly rotated +/- 10 degrees, and have the brightness adjusted randomly by +/- 10%. The models are trained on random subsequences of length 24 as opposed to the full 29 frames. Dropout for recurrent neural networks [38] (uses same dropout mask for entire sequence) is used with $p = 0.5$ before the final fully connected LSTM layer.

The internal cell states c_t and the hidden states h_t are reset to 0 before processing a new sequence. The model applies the Cross Entropy Loss to the output at each timestep allowing the use of truncated backpropagation. Gradients are calculated every eight frames (three times per sequence) with the parameter update performed once at the end of sequence. Temporal features longer than eight frames can still be learned due the internal cell states and hidden states carrying old information even after the gradient propagation has been truncated. The GPU memory use scales linearly with the length of truncated backpropagation implying the batch size can be increased for faster training.

The models take approximately three weeks to train with PyTorch [24] on a NVIDIA Titan X GPU with the Adam [19] optimizer. The learning rate is initialized to $1e^{-4}$ and reduces whenever validation performance stops progressing. Near the end of training, the sequence length is gradually increased to the full 29 frames in order to allow the model to take more advantage of the context.

A bidirectional version of the ResNet based model is created by separately training a reverse direction network initialized with the already trained forward direction network. This network converges more rapidly and increased the accuracy by 1.9% when tested together with the forward direction network.

4.2 Pretraining on LRS

As mentioned in section 3.3, the output of the ResNet based convolutional LSTM model is modified to predict characters in order to leverage additional training data from the LRS datasets. These datasets contain labeled word boundaries allowing the use of Curriculum Learning[4] to speed up training and improve results[7]. Training begins

Table 1: Results comparing the performance on the LRW dataset. The Bidirectional ResNet with convolutional LSTMs achieves comparable results with the current state of the art. Pretraining on more data improves results significantly.

	Pretraining?	Top-1	Top-5	Top-10
VGG-M Multitower [8]	no	61.1%	-	90.4%
VGG-M+LSTM [7]	LRS	76.2%	-	-
2D+ResNet34+Conv [31]	no	69.6%	90.4%	94.8%
3D+ResNet34+Conv [31]	no	74.6%	93.4%	96.5%
3D+ResNet34+BiLSTM [31]	no	83.0%	96.3%	98.3%
3D+ResNet18+BiLSTM [30]	no	84.3%	-	-
VGG-M ConvLSTM	no	73.1%	92.5%	96.7%
ResNet ConvLSTM	no	81.5%	96.1%	98.2%
ResNet BiConvLSTM	no	83.4%	96.8%	98.5%
ResNet BiConvLSTM	LRS2	85.2%	97.4%	98.9%
Reduced ResNet BiConvLSTM	LRS2+LRS3	87.1%	97.7%	98.9%

on subsequences of length eight and gradually increases until a sequence length of 24. This pretraining stage takes approximately one month. The model is then fine-tuned on LRW for an additional week. Gradient clipping[23] (set to 5.0) is necessary due to occasionally large gradients.

5 Results

During inference, three crops of size 48x48, 56x56, and 64x64 with their flipped counterparts (six transformations total) are passed through the network with the outputs averaged for the final prediction. The results are summarized in Table 1. The bottom five entries are convolutional LSTM models trained here.

Convolutional LSTMs see the same improvements as 2D convolutions see with image classification (and 3D convolutions see with action recognition [14]) when upgrading from architectures like VGG to ResNet.

The 3D+ResNet34+BiLSTM model from [31] demonstrates the importance of capturing both the short-term spatiotemporal dynamics by using a 3D convolution early in the network (+5.0%) as well as the long-term context by using a bidirectional LSTM late in the network (+8.4%). The ResNet BiConvLSTM was able to slightly outperform the above (83.4% vs. 83.0%) without requiring special consideration for the placement of the temporally capable layers although this does fall short of the current state of the art[30] set at 84.3% which utilizes a ResNet18 architecture in place of Resnet34.

5.1 Spatiotemporal Features Sensitivity Analysis

The internal cell/hidden states are reset to 0 before processing a new sequence indicating the outputs from convolutional LSTM layers for the first frame have no prior information.

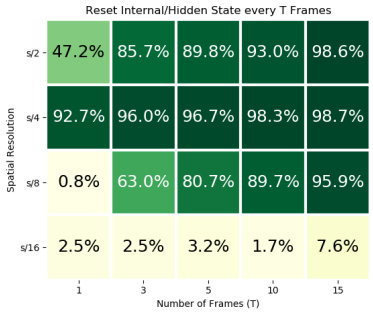


Fig. 5: Relative performance when convolutional LSTM internal states are reset every T frames at particular spatial scales.

The relative importance of temporal information at various spatial resolutions can be explored by artificially resetting the internal states during processing to cause these layers to respond as if they have no context.

The performance on a random subset of 5000 video sequences from the validation set is used as the metric for determining the importance of a particular spatiotemporal scale. Every T frames for $T = \{1, 3, 5, 10, 15\}$, the convolutional LSTMs at a single spatial scale are reset to 0 with all other layers operating normally. The relative performance compared to normal operation is shown in Figure 5. The layers at $s/4$ perform relatively well even when the internal state is reset every frame. The temporal connections are essentially unused. There is a gap between the high resolution, short-term dynamics learned at $s/2$ and the long-term context deeper in the network. A network of stacked 3D convolutions is not naturally geared for these spatiotemporal gaps considering they gradually extend the temporal receptive field. This may help explain the 3D VGG-M model's poor performance relative to its multitower 2D counterpart in [8] as well as why the 3D+ResNet+BiLSTM [31][30] performs so well due to the 2D ResNet being placed within this gap. This also sheds light on the similar performance of ResNet BiConvLSTM with 3D+ResNet+BiLSTM considering it converged to nearly the same model.

5.2 Importance of Data

The convolutional LSTM models trained here increase the state of the art to 85.2% when pretrained on LRS2 and 87.1% when pretrained on LRS2 and LRS3-TED. Although there is value in comparing performance of various spatiotemporal techniques on a dataset like LRW, there remain concerns about the strength of conclusions that can be drawn. The relative difference in performance when utilizing additional data is significantly larger than the differences in performance between techniques. The Reduced ResNet BiConvLSTM model pretrained on LRS2 (195 hours) and LRS3 (444 hours) is used to extract features from LRW (165 hours) to fine-tune the temporal back-end. The back-end model is trained on reduced portions of the LRW dataset with the test accuracies shown in Figure 6. Even with high-quality spatiotemporal features from a model pretrained

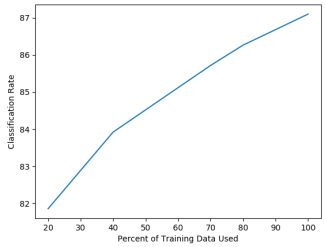


Fig. 6: Test accuracy on LRW after fine-tuning only the temporal back-end on a subset of the LRW training set.

on nearly $4\times$ the amount of data, performance is heavily affected by the proportion of training data used. All of the models heavily overfit and easily achieve above 99% accuracy on the training set.

With the back-end prone to overfitting, it is difficult to imagine training a highly complex, spatiotemporal front-end to its full potential. Although Section 5.1 provides a possible explanation for the similarity in performance between competing models, there is potentially not enough data to appropriately train such models for a fair comparison. This is further supported by the results from [27]. Their model takes one month to train with 64 GPUs on a massive internal dataset of 3,886 hours and outperforms the benchmark for LRS3-TED even *without* any fine-tuning. With such drastic dataset size increases, the cost for hyperparameter/architecture tuning is prohibitive.

However, convolutional LSTMs can be beneficial from a design perspective allowing a quicker approach for new applications when knowledge of the relevant spatiotemporal features is unknown ahead of time. It is easier to initially train a full convolutional LSTM network to figure out what is relevant from the data and then alter the design for the particular application based on the key spatiotemporal features. This is the motivation for the Reduced ResNet ConvLSTM model discussed in Section 3.4 which maintains high performance yet uses roughly half as many parameters. The modified model need not even use convolutional LSTMs as the same sensitivity analysis would support the design of the successful 3D+ResNet+BiLSTM model[30][31].

6 Conclusion

A deep convolutional LSTM model, the first of its kind, is successfully trained and achieves competitive performance for the Lip Reading in the Wild (LRW) dataset. The network is shown to have successfully learned relevant spatiotemporal features from the data without having to specifically cater the design of the network for the specific problem. The sensitivity analysis provides a technique for discovering relevant spatiotemporal features and it was demonstrated how this can facilitate the architecture design process when approaching new spatiotemporal problems. Additionally, the benefits of using improved convolutional architectures like ResNet are apparent for Convolutional LSTMs just as they have been shown in the past for 2D and 3D convolutions.

It remains an open question whether their ability to handle a larger array of spatiotemporal features is necessary for many real applications. However, as datasets become larger and more complex, we may need to rely on the capabilities discussed here or discover new techniques with similar capabilities. Future work will explore utilizing convolutional LSTMs on more general spatiotemporal problems for further verification.

References

1. Afouras, T., Son Chung, J., Senior, A., Vinyals, O., Zisserman, A.: Deep Audio-Visual Speech Recognition. ArXiv e-prints (Sep 2018)
2. Afouras, T., Son Chung, J., Zisserman, A.: Deep Lip Reading: a comparison of models and an online application. ArXiv e-prints (Jun 2018)
3. Afouras, T., Son Chung, J., Zisserman, A.: LRS3-TED: a large-scale dataset for visual speech recognition. ArXiv e-prints (Sep 2018)
4. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning
5. Bian, Y., Gan, C., Liu, X., Li, F., Long, X., Li, Y., Qi, H., Zhou, J., Wen, S., Lin, Y.: Revisiting the effectiveness of off-the-shelf temporal modeling approaches for large-scale video classification. CoRR **abs/1708.03805** (2017)
6. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4724–4733 (2017)
7. Chung, J.S., Senior, A., Vinyals, O., Zisserman, A.: Lip reading sentences in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
8. Chung, J.S., Zisserman, A.: Lip reading in the wild. In: Asian Conference on Computer Vision (2016)
9. Chung, J.S., Zisserman, A.: Lip reading in profile. In: British Machine Vision Conference (2017)
10. Conneau, A., Schwenk, H., Barrault, L., LeCun, Y.: Very deep convolutional networks for natural language processing. CoRR **abs/1606.01781** (2016)
11. Cotterell, R., Mielke, S.J., Eisner, J., Roark, B.: Are all languages equally hard to language-model? In: NAACL-HLT (2018)
12. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning. pp. 369–376. ICML '06 (2006)
13. Hanson, A.J., Pnvr, K., Krishnagopal, S., Davis, L.: Bidirectional convolutional lstm for the detection of violence in videos. In: ECCV Workshops (2018)
14. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? CoRR **abs/1711.09577** (2017)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015)
16. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
17. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of International Computer Vision and Pattern Recognition (CVPR 2014) (2014)
18. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The kinetics human action video dataset. CoRR **abs/1705.06950** (2017)

19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012)
21. Luo, W., Li, Y., Urtasun, R., Zemel, R.S.: Understanding the effective receptive field in deep convolutional neural networks. CoRR **abs/1701.04128** (2017)
22. Ng, J.Y.H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: Deep networks for video classification. In: Computer Vision and Pattern Recognition (2015)
23. Pascanu, R., Mikolov, T., Bengio, Y.: Understanding the exploding gradient problem. CoRR **abs/1211.5063** (2012)
24. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
25. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
26. Shi, X., Chen, Z., Wang, H., Yeung, D., Wong, W., Woo, W.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. CoRR **abs/1506.04214** (2015)
27. Shillingford, B., Assael, Y.M., Hoffman, M.W., Paine, T., Hughes, C., Prabhu, U., Liao, H., Sak, H., Rao, K., Bennett, L., Mulville, M., Coppin, B., Laurie, B., Senior, A.W., de Freitas, N.: Large-scale visual speech recognition. CoRR **abs/1807.05162** (2018)
28. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 568–576. Curran Associates, Inc. (2014)
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
30. Stafylakis, T., Khan, M.H., Tzimiropoulos, G.: Pushing the boundaries of audiovisual word recognition using residual networks and lstms. CoRR **abs/1811.01194** (2018)
31. Stafylakis, T., Tzimiropoulos, G.: Combining residual networks with lstms for lipreading. CoRR **abs/1703.04105** (2017)
32. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 3104–3112. Curran Associates, Inc. (2014)
33. Szegedy, C., Ioffe, S., Vanhoucke, V.: Inception-v4, inception-resnet and the impact of residual connections on learning. CoRR **abs/1602.07261** (2016)
34. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions (2015)
35. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: C3D: generic features for video analysis. CoRR **abs/1412.0767** (2014)
36. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. CoRR **abs/1706.03762** (2017)
37. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H.: Hierarchical attention networks for document classification. In: HLT-NAACL (2016)
38. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. CoRR **abs/1409.2329** (2014)
39. Zhang, L., Zhu, G., Shen, P., Song, J.: Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition. pp. 3120–3128 (10 2017). https://doi.org/10.1109/ICCVW.2017.369