Look, Listen and Learn — A Multimodal LSTM for Speaker Identification

Jimmy Ren,[1] **Yongtao Hu,**[2] **Yu-Wing Tai,**[1] **Chuan Wang,**[2]
Li Xu,[1] **Wenxiu Sun,**[1] **Qiong Yan**[1]
SenseTime Group Limited[1]
{rensijie, yuwing, xuli, sunwenxiu, yanqiong}@sensetime.com
The University of Hong Kong[2]
{herohuyongtao, wangchuan2400}@gmail.com
Project page: http://www.deeplearning.cc/mmlstm

Abstract

Speaker identification refers to the task of localizing the face of a person who has the same identity as the ongoing voice in a video. This task not only requires collective perception over both visual and auditory signals, the robustness to handle severe quality degradations and unconstrained content variations are also indispensable. In this paper, we describe a novel multimodal Long Short-Term Memory (LSTM) architecture which seamlessly unifies both visual and auditory modalities from the beginning of each sequence input. The key idea is to extend the conventional LSTM by not only sharing weights across time steps, but also sharing weights across modalities. We show that modeling the temporal dependency across face and voice can significantly improve the robustness to content quality degradations and variations. We also found that our multimodal LSTM is robustness to distractors, namely the non-speaking identities. We applied our multimodal LSTM to The Big Bang Theory dataset and showed that our system outperforms the state-of-the-art systems in speaker identification with lower false alarm rate and higher recognition accuracy.

Speaker identification is one of the most important building blocks in many intelligent video processing systems such as video conferencing, video summarization and video surveillance, etc. It aims to localize the face of the speaker associated with the ongoing voices. To achieve this task, collective perception over both visual and auditory signals is indispensable.

In the past few years, we observe the rapid advances in face recognition and speech recognition respectively by using Convolutional Neural Networks (CNN) (Schroff, Kalenichenko, and Philbin 2015; Sun et al. 2014a) and Recurrent Neural Networks (RNN) (Graves, Mohamed, and Hinton 2013; Hannun et al. 2014). Notwithstanding the recent ground breaking results in processing facial and auditory data, speaker identification (figure 1b) remains challenging for the following reasons. First, severe quality degradations (e.g. blur and occlusion) and unconstrained content variations (e.g. illumination and expression) in real-life videos are not uncommon. These effects significantly degrade the performance of many existing CNN based methods. Second, the state-of-the-art convolutional network based face model is trained with still images. Its application in sequential data as well as its connection to recurrent networks is less explored and it is not straightforward to extend CNN based methods to the multimodal learning setting. Third, a practical system should be robust enough to reject distractors, the faces of non-speaking persons indicated by the red bounding boxes in figure 1b, which adds additional challenges to the task.

Despite the rich potential of both CNN and RNN for facial and auditory data, it is still unclear if these two networks can be simultaneously and effectively adopted in the context of multimodal learning. In this paper, we proposed a novel multimodal Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997), a specialized type of Recurrent Neural Network, to address this problem. We show that modeling temporal dependency for facial sequences by LSTM performs much better than CNN based methods in terms of robustness to image degradations. More importantly, by sharing weights not only across time steps but also across different modalities, we can seamlessly unify both visual and auditory modalities from the beginning of each sequence input, which significantly improves the robustness to distractors. This is because the cross-modal shared weights can learn to capture temporal correlation between face and voice sequences. Note that our multimodal LSTM did not assume the the random variables from different modal are correlated. Instead, our multimodal LSTM is capable to learn such correlation if there is any temporal correlation across different modal in their sequences. In the speaker identification task, we believe such temporal correlation exists in the respective face and voice sequences. In this work, we assume the face of the speaker appears in a video when they speak, and there is only one speaker at the same time during the voice over. Multiple speakers in the same video clip can be identified as far as their speaks do not overlap.

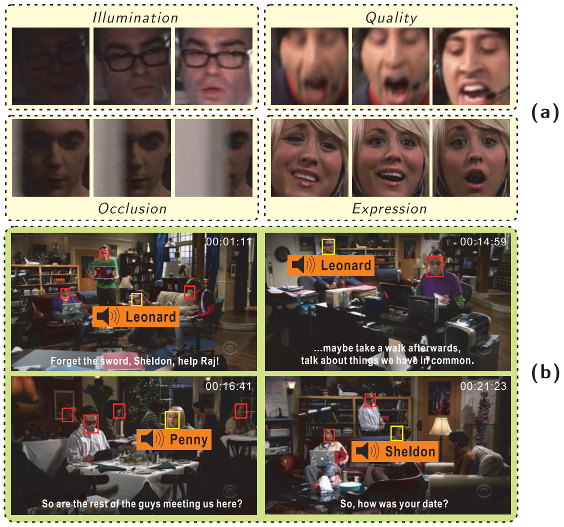
Illumination
Quality
Occlusion
Expression
(a)
00:01:11
Leonard
Forget the sword, Sheldon, help Raj!
00:14:59
Leonard
...maybe take a walk afterwards, talk about things we have in common.
(b)
00:16:41
Penny
So are the rest of the guys meeting us here?
00:21:23
Sheldon
So, how was your date?

Figure 1: (a) Face sequence with different kinds of degradations and variations. Using the previous CNN methods cannot recognize the speakers correctly. In contrast, the speakers can be successfully recognized by our LSTM in both single-modal and multimodal settings. (b) Our multimodal LSTM is robust to both image degradation and distractors. Yellow bounding boxes are the speakers. Red bounding boxes are the non-speakers, the distractors.

To our knowledge, our paper is the first attempt in modeling long-term dependencies over multimodal high-level features which demonstrates robustness to both distractors and image degradation. We applied our model to The Big Bang Theory dataset and showed that our system outperformed the state-of-the-art systems in recognition accuracy and with lower false alarm rate.

The contributions of this paper are as follows.

- We proposed a novel LSTM architecture which enables multimodal learning of sequence data in a unified model. Both temporal dependency within each modality and temporal correlation across modalities can be automatically learned from data.
- We empirically showed that cross-modality weight sharing in LSTM simultaneously improves the precision of classification and the robustness to distractors.
- We successfully applied our method in a real-world multimodal classification task and the resulting system outperformed the state-of-the-art. The dataset and our implementations are both publicly available.

Related Work

Many recent studies have reported the success of using deep CNN in face related tasks. The pioneering work by (Taigman et al. 2014) proposed a very deep CNN architecture together with an alignment technique to perform face verification which achieved near human-level performance. Inspired by GoogLeNet (Szegedy et al. 2015), Sun et al. 2014b used a very deep CNN network with multiple levels of supervision, which surpassed human-level face verification performance in the LFW dataset (Huang and Learned-Miller 2013). The recent advance in this field (Schroff, Kalenichenko, and Philbin 2015) pushed the performance even further. In face detection, the state-of-the-art results were also achieved by CNN based models (Yang et al. 2015; Li et al. 2015). For other face related tasks such as face landmark detection and face attribute recognition (Zhang et al. 2015a; 2015b), CNN based models were also widely adopted.

The revived interest on RNN is mainly attributed to its recent success in many practical applications such as language modeling (Kiros et al. 2015), speech recognition (Chorowski et al. 2015; Graves, Mohamed, and Hinton 2013), machine translation (Sutskever, Vinyals, and Le 2014; Jean et al. 2015), conversation modeling (Shang, Lu, and Li 2015) to name a few. Among many variants of RNNs, LSTM is arguably one of the most widely used model. LSTM is a type of RNN in which the memory cells are carefully designed to store useful information to model long term dependency in sequential data (Hochreiter and Schmidhuber 1997). Other than supervised learning, LSTM is also used in recent work in image generation (Theis and Bethge 2015; Gregor et al. 2015), demonstrating its capability of modeling statistical dependencies of imagery data.

In terms of the sequence learning problem across multiple modalities, LSTM based models were actively used in recent image caption generation studies (Donahue et al. 2015; Karpathy and Li 2015; Xu et al. 2015). One common characteristic of these techniques is that CNN was used to extract the feature sequences in an image and LSTM was used to generate the corresponding text sequences. Our paper is related to this group of studies in a way that more than one modalities are involved in the learning process. However, our goal is not to generate sequences in an alternative domain but to collectively learn useful knowledge from sequence data of multiple domains. Perhaps the most closely related previous studies to our work are from (Srivastava and Salakhutdinov 2012) and (Ngiam et al. 2011). Unlike these papers, we focused on high-level multimodal learning which explicitly models the temporal correlation of high-level features rather than raw inputs between different modalities. This not only provided a channel to effectively transfer the recent success of deep CNN to the multimodal learning context, the resulting efficient implementation can be directly adopted in video processing as well. We also investigated the robustness to distractors and input quality which is not considered in the previous studies. The closely related papers in multimedia speaker identification are (Bauml, Tapaswi, and Stiefelhagen 2013; Hu et al. 2015), and (Tapaswi, Bäuml, and Stiefelhagen 2012). However, they did not explicitly model face sequences and the interplay between face and voice sequences.

LSTM - Single VS. Multi-Modal

Single Modal LSTM A regular LSTM network contains a number of memory cells within which the multiplicative gate units and the self-recurrent units are the two fundamental building blocks (Hochreiter and Schmidhuber 1997). For

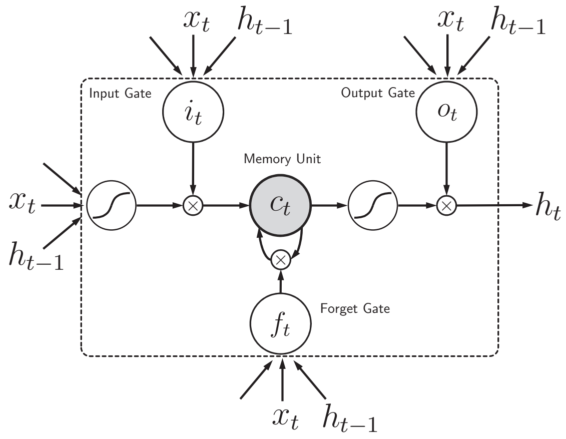

Figure 2: Memory Cell of Single-modal LSTM.

a brief revision, equations (1), (3) and (5) formally describe the memory input, the forget gate and the recurrent units of a regular LSTM in the forward pass. The input gate i_t and the output gate o_t in a regular LSTM resemble the forget gate in the forward pass. Figure 2 shows a pictorial illustration of a regular LSTM model.

$$g_t = \varphi(\mathbf{W}_{xg} * X_t + \mathbf{W}_{hg} * h_{t-1} + b_g), \quad (1)$$
$$i_t = \sigma(\mathbf{W}_{xi} * X_t + \mathbf{W}_{hi} * h_{t-1} + b_i), \quad (2)$$
$$f_t = \sigma(\mathbf{W}_{xf} * X_t + \mathbf{W}_{hf} * h_{t-1} + b_f), \quad (3)$$
$$o_t = \sigma(\mathbf{W}_{xo} * X_t + \mathbf{W}_{ho} * h_{t-1} + b_o), \quad (4)$$
$$C_t = f_t \odot C_{t-1} + i_t \odot g_t, \quad (5)$$
$$y_t = softmax(\mathbf{W}_y * h_t). \quad (6)$$

In (1) and (3), X is an input sequence where X_t is an element of the sequence at time t, h_{t-1} is the output of the memory cell at time $t-1$. $\mathbf{W}_{xg}, \mathbf{W}_{xf}, \mathbf{W}_{hg}, \mathbf{W}_{hf}$ are distinct weight matrices, and b_g and b_f are bias terms respectively. φ and σ are nonlinear functions where φ denotes a $tanh$ function and σ denotes a $sigmoid$ function. In (5), $\odot$ denotes an element-wise multiplication, i_t is the input gate at the time step t, and C_{t-1} is the memory unit at the time step $t-1$. The memory unit at time step t is therefore generated by the collective gating of the input gate and the forget gate. In equation (6), the memory cell output of the current time step is multiplied by $\mathbf{W}_y$ and then transformed by the $softmax$ function to compute the model output y_t at time t.

Generally speaking, the reason that LSTM is able to model long-term dependencies in sequential data is because C_t at each time step can selectively "remember" (store) or "forget" (erase) past information which is modelled by the multiplicative gating operation. More importantly, the strategy to open or to close the gates is data driven which is automatically learned from training data. This information is captured by the trainable weights $\mathbf{W}$, including $\mathbf{W}_{hf}, \mathbf{W}_{xf}$ and so on, rather than hand-crafted. Because $\mathbf{W}$ are shared across time steps, this endows LSTM the power to explicitly model temporal relationships over the entire sequence.

Simple extensions of Single Modal LSTM In order to deal with data from different domains, perhaps the most straightforward method is to incorporate them into a single network by concatenating the data directly to produce a bigger X. However, this approach is problematic because the multimodal property of the inputs are completely ignored and the model does not have any explicit mechanism to model the correlation across modalites. Though some correlations may be weakly captured by the trained weights, a critical weakness is that it is incapable to handle distractors. In particular, when the face of a person A is combined with the voice of a person B, the model is confused and would fail to generate a meaningful label. Although it may be possible to put all the distractors to a single class and let the model to distinguish the speaker and the distractors automatically, this method performs much worse than our solution in practice. The major difficulty is that distractors share too many features with regular examples when organize the inputs in this way.

Another solution is to treat data from different domains completely independent. Namely, we can use multiple LSTMs in parallel and then merge the output labels at the highest layer using a voting mechanism. The advantage of this approach is that the two separate memory units can be trained to store useful information explicitly for each domain. But the weakness is that the interaction across modalities only happens at the highest level during the labelling process. The cross-model correlation is therefore very difficult, if not entirely impossible, to be encoded into the weights through the learning process. Thus, the robustness to distractors relies heavily on the voting stage where some of the temporal correlations may have already been washed out in the independent forward pass.

Multimodal LSTM Compared with the straightforward solutions, we want to develop a new multimodal LSTM which can explicitly model the long-term dependencies both within the same modality and across modalities in a single multimodal LSTM. Instead of merging input data at preprocessing stage, or merging labels at post-processing stage, our key idea is to selectively share weights across different modalities during the forward pass. This is similar to the weight sharing in time domain in regular LSTM, but we do not share memory units for each modality within the memory cell. The modifications are illustrated in figure 3 and formally expressed in the following equations.

$$g_t^s = \varphi(\mathbf{W}_{xg}^s * X_t^s + \mathbf{W}_{hg} * h_{t-1}^s + b_g^s), \;\; s = 1\, to\, n, \quad (7)$$
$$i_t^s = \sigma(\mathbf{W}_{xi}^s * X_t^s + \mathbf{W}_{hi} * h_{t-1}^s + b_i^s), \;\; s = 1\, to\, n, \quad (8)$$
$$f_t^s = \sigma(\mathbf{W}_{xf}^s * X_t^s + \mathbf{W}_{hf} * h_{t-1}^s + b_f^s), \;\; s = 1\, to\, n, \quad (9)$$
$$o_t^s = \sigma(\mathbf{W}_{xo}^s * X_t^s + \mathbf{W}_{ho} * h_{t-1}^s + b_o^s), \;\; s = 1\, to\, n, (10)$$
$$C_t^s = f_t^s \odot C_{t-1}^s + i_t^s \odot g_t^s, \;\; s = 1\, to\, n, \quad (11)$$
$$h_t^s = o_t^s \odot \varphi(C_t^s), \;\; s = 1\, to\, n, \quad (12)$$
$$y_t^s = softmax(\mathbf{W}_y * h_t^s), \;\; s = 1\, to\, n. \quad (13)$$

Keeping the gating mechanism the same as the regular LSTM, the equations from (7) to (13) describe a new cross-modal weight sharing scheme in the memory cell. The super-

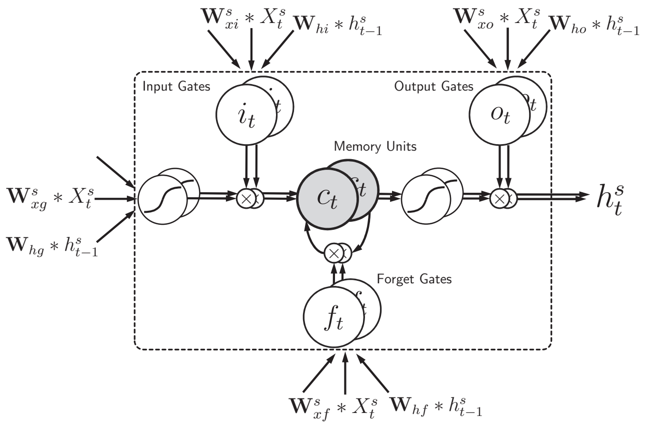
$\mathbf{W}^s_{xi} * X^s_t$
$\mathbf{W}_{hi} * h^s_{t-1}$
$\mathbf{W}^s_{xo} * X^s_t$
$\mathbf{W}_{ho} * h^s_{t-1}$
Input Gates
i_t
Output Gates
o_t
Memory Units
c_t
$\mathbf{W}^s_{xg} * X^s_t$
$\mathbf{W}_{hg} * h^s_{t-1}$
h^s_t
Forget Gates
f_t
$\mathbf{W}^s_{xf} * X^s_t$
$\mathbf{W}_{hf} * h^s_{t-1}$

Figure 3: Memory Cell of Multimodal LSTM.

script ^{s} indexes each modality in the input sequences. n is the total number of modalities in input data, where $n = 2$ in the speaker identification task. The model is general enough to deal with the tasks with $n > 2$. X^s_t is the input sequence at time t for modality s. Therefore, the weights with superscript ^{s} (e.g. $\mathbf{W}^s_{xg}$) are NOT shared across modalities but only across time steps, the other weights without the superscript (e.g. $\mathbf{W}_{hg}$) are shared across both modalities and time steps. Specifically, the weights associated with the inputs X^s_t are not shared across modalities. The reasons are twofold. First, we would like to learn a specialized mapping, separately for each modality, from its input space to a new space, where multimodal learning is ensured by the shared weights. Second, specialized weights are preferred to reconcile the dimension difference between different modalities, which avoids a complex implementation.

Along with the transform associated with X^s_t, the output of the memory cell from the previous time step h^s_{t-1} also need to go through a transform in producing g^s_t as well as all other gates. The weights to perform this transform, $\mathbf{W}_{hg}, \mathbf{W}_{hi}, \mathbf{W}_{hf}$ and $\mathbf{W}_{ho}$ are shared across the modalities. With these new weight definitions, the separately transformed data by the four $\mathbf{W}^s$ is essentially interconnected from g^s_t all the way to the memory cell output h^s_t.

The key insight is that while it is preferable for each modality to have its own information flow because it enables a more focused learning objective with which it is easier to learn the long-term temporal dependency within the modality, we also make such objective correlated and essentially constrained by what happens in the rest of the modalities. More specifically, in forming the forget gate f^s_t for $s = 1$, it not only relates to $h^{s=1}_{t-1}$ but also constrained by $h^{s=2}_{t-1}$, ... , $h^{s=n}_{t-1}$ because $\mathbf{W}_{hf}$ is shared among them. Provided that the weights $\mathbf{W}_{hg}, \mathbf{W}_{hi}, \mathbf{W}_{hf}$ and $\mathbf{W}_{ho}$ are also shared across time steps, they play the vital role of capturing the temporal correlation across different modalities.

Another important property of the proposed model is that the memory unit C is NOT shared among modalities. The rationale is that the weights have the job to capture intramodal as well as intermodal relationships, therefore placing them in a single memory unit provides much less flexibility on what can be stored or forgotten. Given all the gates are formed in a multimodal fashion, the insight of such design is that we should not hand-craft the decision on what intramodal/intermodal relationships should be stored or forgotten but to give the model enough flexibility to learn it from data. The bias terms are not shared across modalities neither to increase this flexibility.

Likewise, the network output at each time step y^s_t does not relate to its own modality. Whether we should use $\mathbf{W}_y$ or $\mathbf{W}^s_y$ to transform h^s_t before sending the outputs to the softmax function is not a straightforward decision. We resort to our experiment to address this issue.

High-Level Feature VS. Raw Inputs One of the most important reasons why CNN is attractive is because it is an end to end feature learning process. Previous studies (Razavian et al. 2014; Oquab et al. 2014) have discovered that a successful CNN for a classification task also produces high-level features which are general enough to be used in a variety of tasks. This finding inspired a few recent work on image captioning (Xu et al. 2015; Karpathy and Li 2015) where the high-level CNN features over an image sequence were extracted, and a RNN is learned on top of the extracted CNN features to perform more sophisticated tasks. Such approach is very effective to bridge the effort and success in CNN to the field of sequence modeling. We would like to extend this type of attempt to multimodal sequence learning.

Implementation In order to maximize the flexibility of our investigation and efficiently work with different variants of network architecture and working environments (e.g. Linux and Windows), we did not implement the multimodal LSTM using any third-party deep learning packages. Instead, we used MATLAB and its GPU functions in the parallel computing toolbox to build our own LSTM from scratch. Our implementation is vectorized (Ren and Xu 2015) and very efficient in training both single-modal and multimodal LSTM described in this paper.

Experiments

Three experiments were carefully designed to test the robustness and the applicability of our proposed model.

Dataset overview We chose the TV-series The Big Bang Theory (BBT) as our data source. It has been shown that the speaker identification task over BBT is a very challenging multimodal learning problem due to various kinds of image degradation and the high variations on faces in the videos (Bauml, Tapaswi, and Stiefelhagen 2013; Hu et al. 2015; Tapaswi, Bäuml, and Stiefelhagen 2012). During data collection, we ran face detection and extracted all the faces in six episodes in the first season and another six episodes in the second season of BBT. We manually annotated the faces for the five leading characters, i.e. *Sheldon*, *Leonard*, *Howard*, *Raj* and *Penny*. In total, we have more than 310,000 consecutively annotated face images for the five characters. For audio data, we utilized the pre-annotated subtitles and only extracted the audio segments corresponding to speeches. Data from the second season was used in training and data from the first season was used in testing for all the experiments reported below.

Feature extraction To ensure the usability of the resulting system, we adopted 0.5 second as the time window of all the sequence data including both face and audio. For feature extraction for faces, we adopted a CNN architecture resembles the one in (Krizhevsky, Sutskever, and Hinton 2012) and trained a classifier using the data reported in the next section. The activations of the last fully connected layer was used as the high-level feature for face. We also run principle component analysis (PCA) on all the extracted face features to reduce the dimensionality to the level comparable to audio features. By keeping 85% of the principle components, we obtained a 53-dimension feature vector for each face in the video. The video is 24 frames per second, therefore there are 12 consecutive faces within each face sequence. For audio, we used the mel-frequency cepstral coefficients (MFCC) features (Sahidullah and Saha 2012). Following (Hu et al. 2015) we extracted the 25d MFCC features in a sliding window of 20 milliseconds with stride of 10 milliseconds, thus gives us 25×49 MFCC features.

LSTM for Face Sequences

Our first task is to investigate the extend to which modeling temporal dependency of high-level CNN features improves the robustness to quality degradation and face variation. The reasons that we would like to investigate this manner is two-fold. First, though it was showed that RNN can be successfully used in speech recognition to improve the robustness to noise (Graves, Mohamed, and Hinton 2013), it was not clear from the literature whether similar principle applies for high-level image features. On the other hand, we would like to clearly measure the extend to which this approach works for faces because this is an important cornerstone for the rest of the experiments.

Data Only face data in the aforementioned data set was used in the experiment. We randomly sampled 40,000 face sequences from the training face images and another 40,000 face sequences in the test face images. Note that each sequence was extracted according to the temporal order in the data, however, we did not guarantee the sequence are strictly from one subtitle segment. This injected more variations in the sequence.

Procedure and Results Three methods were compared in this experiment. The first method was to use a CNN to directly classify each frame in the sequence. This CNN is the same one as used in the feature extraction for our LSTM. In our setting, the CNN will output 12 labels (12 probability distributions) for each face sequence. Then the output probabilities were averaged to compute the final label for this sequence. The second method used the same CNN to extract features for each frame and reduced the dimensionality using PCA as described in the last section. Then we used a SVM with RBF kernel to classify each feature followed by the same averaging processing before outputting the label. We used the single-modal LSTM (see figure 2) with one hidden memory cell layer to train a sequential classifier. The dimensionality of the hidden layer activation is 512. In our setting, this LSTM contains 12 time steps with 12 identical supervision signals (labels). During the testing, we only look at the last output in the whole output sequence.

Table 1: Face sequence classification accuracy of different algorithms.

ALGORITHMS	ACCURACY (%)
CNN	92.33
CNN+SVM	92.42
LSTM	95.61

The results were reported in table 1. We can see that the two CNN alone approaches delivered very similar results, acknowledging the high representative powerful of the CNN features reported in the previous studies. The accuracy of the CNN+SVM approach slightly outperformed the CNN alone approach. This is reasonable because SVM with RBF kernel may classify the data better than the last layer of CNN. The performance of LSTM is significantly higher than the other two. By looking at the correctly classified face sequences which were failed in the other two methods, we can see that the LSTM is more robust to a number of image degradations and variations. This is illustrated in figure 1a.

Comparison among Multimodal LSTMs

By the results from the first experiment, there is a reason to believe that performing multimodal learning of face and audio in temporal domain, if do it correctly, has the potential to perform better in speaker identification task. Therefore, the aim of the second experiment was to examine the extend to which the multimodal LSTM benefits the speaker identification performance. Multiple aforementioned multimodal LSTM solutions were tested and compared in this experiment. See the result session for details.

Data In the training process, the face data from the previous experiment was used. One problem is that each face sequence has only 12 time steps which is inconsistent with the 49 time steps in audio sequences. To circumvent this inconsistency, we simply duplicated faces evenly within the 49 time steps. The combinations of face sequences and audio sequences for each identity were randomly paired by the training program during the runtime to maximize the diversity of the training set. For test set of this task, the combinations were however pre-generated. We randomly generated 250,000 correctly paired combinations and 250,000 distractors (ill-paired combinations).

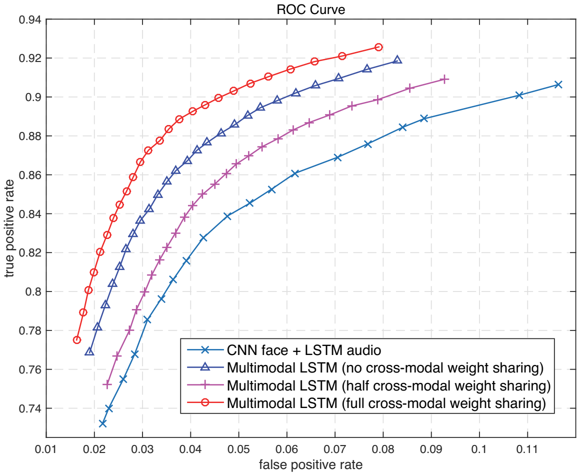

Figure 4: Multimodal Long Short-Term Memory.

Procedure and Results During the comparison, one baseline method and three alternative multimodal LSTM methods were used. In the baseline method, we separately trained a single-modal LSTM only for audio using the same audio data in this experiment. We carefully tuned many hyper parameters, making sure it performed as well as we can achieve. We used it to classify the audio sequence. For face sequence, we used the CNN+SVM approach from the last experiment. Therefore, we shall have 49 proposals for audio labels and another 49 labels for face label proposals. We then looked at the number of labels agreed within these two groups of labels. A threshold m is set to distinguish the sample between distractors and normal samples. For instance, if $m = 10$ then the whole multimodal sequence will be classified to distractors if there are more than 10 label proposals temporally disagreed with each other. Otherwise, the multimodal sequence will be classified by averaging the proposals. Note that this distractor rejection procedure was used in all the compared methods in this experiment. The threshold m is tuned to generate various dots in the ROC curve.

The first multimodal solution does not share any weights across the two modality resulting in two separate single-modal LSTMs. We called it “no cross-modal weight sharing”. The second solution used the weight sharing scheme introduced previously, but did not share $\mathbf{W}_y$ across the modality. Formally, equation (13) should be re-written as

$$y_t^s = softmax(\mathbf{W}_y^s * h_t^s), \quad s = 1 \; to \; n. \qquad (14)$$

We called this solution “half cross-modal weight sharing”. The third solution completely followed the equation (7) to (13), named “full cross-modal weight sharing”.

As shown in figure 4, the performance difference is clear. It was expected that the baseline method performed less competitive. However, having isolated $\mathbf{W}_y$ for each modality performed worse than the naive combination of two single-model LSTMs. On the other hand, with the full weight sharing, multimodal LSTM significantly outperforms all other methods.

Discussion The false alarm rate was largely increased by not sharing $\mathbf{W}_y$ across modalities. The role of $\mathbf{W}_y$ is to transform the memory cell outputs at each time step to the desirable labels. By sharing this transform across the modality, we can generate more consistent labels for normally paired samples and increased the robustness to distractors. Our experiments showed that this behavior can be automatically captured by the shared $\mathbf{W}_y$.

Speaker Identification in The Big Bang Theory

The last experiment is to apply our method in real-life videos and compare the performance with previous studies.

Data To compare with other speaker identification methods, we evaluated the winning multimodal LSTM from the previous experiment in The Big Bang Theory S01E03, as in (Bauml, Tapaswi, and Stiefelhagen 2013; Tapaswi, Bäuml, and Stiefelhagen 2012; Hu et al. 2015).

Procedure and Results We applied our model to video with the time window of 0.5 second and stride of 0.25 second (e.g. 0s-0.5s, 0.25s-0.75s...). Unlike the controlled setting in the second experiment, the number of distractors in videos varies for each scene. In some cases, there are only distractors in a scene. The evaluation criteria should be more sophisticated. We followed (Hu et al. 2015) to calculate the accuracy of speaker identification to ensure a fair comparison. Specifically, speaker identification is considered successful if a) the speaker is in the scene and the system correctly recognized him/her and correctly rejected all the distractors, or b) the speaker is not in the scene and the system correctly rejected all the distractors in the scene.

Time window more than 0.5 second was also tested to enable a more systematic comparison. We achieved this by further voting within this larger time window. For instance, by having 50% overlapping of 0.5 second windows in the larger window of 2.0 seconds, we will have seven 0.5 second-sized small windows to vote for the final labels.

Our speaker identification results are reported in table 2. We compared our method against the state-of-the-art systems. Note that, in (Bauml, Tapaswi, and Stiefelhagen 2013; Tapaswi, Bäuml, and Stiefelhagen 2012), as both of them examined the face tracks within the time window specified by the subtitle/transcript segments, they can be viewed as voting on the range of subtitle/transcript segments. As the average time of subtitle/transcript segments in the evaluation video is 2.5s, they are equivalent to our method when evaluated in the voting window of such size. We applied the same voting strategy as in (Hu et al. 2015) under different time window setup. As can be seen from the results, our method outperformed the previous works by a significant margin.

Conclusion

In this paper, we have introduced a new multimodal LSTM and have applied it to the speaker identification task. The key idea is to utilize the cross-modality weight sharing to capture correlation of two or more temporally coherent modalities.

Table 2: Speaker naming accuracy of different algorithms (%) in terms of different voting time window (s).

Time window (s)	0.5	1.0	1.5	2.0	2.5	3.0
Bauml et al. 2013	-	-	-	-	77.81	-
Tapaswi et al. 2012	-	-	-	-	80.80	-
Hu et al. 2015	74.93	77.24	79.35	82.12	82.81	83.42
Ours	86.59	89.00	90.45	90.84	91.17	91.38

As demonstrated in our experiments, our proposed multimodal LSTM is robust against image degradation and distractors, and has outperformed state-of-the-art techniques in speaker identification. To our knowledge, this is the first attempt in modeling long-term dependencies over multimodal high-level features. We believe our multimodal LSTM is also useful to other applications not limited to the speaker identification task.

References

Bauml, M.; Tapaswi, M.; and Stiefelhagen, R. 2013. Semi-supervised learning with constraints for person identification in multimedia data. In *CVPR*.

Chorowski, J.; Bahdanau, D.; Serdyuk, D.; Cho, K.; and Bengio, Y. 2015. Attention-based models for speech recognition. In *NIPS*.

Donahue, J.; Hendricks, L. A.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*.

Graves, A.; Mohamed, A.; and Hinton, G. E. 2013. Speech recognition with deep recurrent neural networks. In *ICASSP*.

Gregor, K.; Danihelka, I.; Graves, A.; Rezende, D. J.; and Wierstra, D. 2015. Draw: A recurrent neural network for image generation. In *ICML*.

Hannun, A.; Case, C.; Casper, J.; Catanzaro, B.; Diamos, G.; Elsen, E.; Prenger, R.; Satheesh, S.; Sengupta, S.; Coates, A.; and Ng, A. Y. 2014. Deep speech: Scaling up end-to-end speech recognition. In *Arxiv 1412.5567*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Hu, Y.; Ren, J. S.; Dai, J.; Yuan, C.; Xu, L.; and Wang, W. 2015. Deep Multimodal Speaker Naming. In *ACMMM*.

Huang, G. B., and Learned-Miller, E. 2013. Labeled faces in the wild: Updates and new reporting procedures. In *University of Massachusetts, Amherst Technical Report UM-CS-2014-003*.

Jean, S.; Cho, K.; Memisevic, R.; and Bengio, Y. 2015. On using very large target vocabulary for neural machine translation. In *ACL*.

Karpathy, A., and Li, F.-F. 2015. Deep visual-semantic alignments for generating image description. In *CVPR*.

Kiros, R.; Zhu, Y.; Salakhutdinov, R.; ; and Zemel, R. S. 2015. Skip-thought vectors. In *NIPS*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1106–1114.

Li, H.; Lin, Z.; Shen, X.; Brandt, J.; and Hua, G. 2015. A convolutional neural network cascade for face detection. In *CVPR*.

Ngiam, J.; Khosla, A.; Nam, J.; Lee, H.; and Ng, A. Y. 2011. Multimodal deep learning. In *ICML*.

Oquab, M.; Bottou, L.; Laptev, I.; and Sivic, J. 2014. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*.

Razavian, A. S.; Azizpour, H.; Sullivan, J.; and Carlsson, S. 2014. Cnn features off-the-shelf: An astounding baseline for recognition. In *CVPR Workshop*.

Ren, J., and Xu, L. 2015. On vectorization of deep convolutional neural networks for vision tasks. In *AAAI*.

Sahidullah, M., and Saha, G. 2012. Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition. *Speech Communication* 54(4):543565.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*.

Shang, L.; Lu, Z.; and Li, H. 2015. Neural responding machine for short-text conversation. In *ACL*.

Srivastava, N., and Salakhutdinov, R. 2012. Multimodal learning with deep boltzmann machines. In *NIPS*.

Sun, Y.; Chen, Y.; Wang, X.; and Tang, X. 2014a. Deep learning face representation by joint identification-verification. In *NIPS*.

Sun, Y.; Liang, D.; Wang, X.; ; and Tang, X. 2014b. Deepid3: Face recognition with very deep neural networks. In *arXiv:1502.00873*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*.

Tapaswi, M.; Bäuml, M.; and Stiefelhagen, R. 2012. knock! knock! who is it? probabilistic person identification in tv-series. In *CVPR*.

Theis, L., and Bethge, M. 2015. Generative image modeling using spatial lstms. In *NIPS*.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.

Yang, S.; Luo, P.; Loy, C. C.; and Tang, X. 2015. From facial part responses to face detection: A deep learning approach. In *ICCV*.

Zhang, Z.; Luo, P.; Loy, C. C.; and Tang, X. 2015a. Facial landmark detection by deep multi-task learning. In *ECCV*.

Zhang, Z.; Luo, P.; Loy, C. C.; and Tang, X. 2015b. Learning deep representation for face alignment with auxiliary attributes. In *arXiv 1408.3967*.