Sequential Deep Learning for Disaster-Related Video Classification

Haiman Tian, Hector Cen Zheng, Shu-Ching Chen
School of Computing and Information Sciences
Florida International University
Miami, FL 33199, USA
{htian005, hcen001, chens}@cs.fiu.edu

Abstract

Videos serve to convey complex semantic information and ease the understanding of new knowledge. However, when mixed semantic meanings from different modalities (i.e., image, video, text) are involved, it is more difficult for a computer model to detect and classify the concepts (such as flood, storm, and animals). This paper presents a multimodal deep learning framework to improve video concept classification by leveraging recent advances in transfer learning and sequential deep learning models. Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNN) models are then used to obtain the sequential semantics for both audio and textual models. The proposed framework is applied to a disaster-related video dataset that includes not only disaster scenes, but also the activities that took place during the disaster event. The experimental results show the effectiveness of the proposed framework.

1. Introduction

Vast amounts of multimodal data (image, video, and text) are being generated on a daily basis by users through personal devices and social networking services. Classifying massive amounts of single-modal data is an ongoing research field that has gained benefits from the advances in computer vision, audio classification [1], text recognition [13], and natural language processing [10]. However, as the magnitude and capabilities of data generation and collection grow exponentially, more reliable and cutting-edge classification methods are needed in order to reap the benefits of the knowledge that can be attained, from not only each single modality alone but also multiple modalities. Additionally, the new challenges that surface when trying to acquire the useful information from multimodal data demand improved techniques to obtain more accurate classification results.

Each data modality has its own strengths and associated deep learning approaches and techniques. For audio data, models that are able to extract or predict natural sounds are very scarce due to the focuses on speech recognition and music classification. As for textual data, word embedding models show significant improvements both as feature learning and language modeling techniques by representing the similarity between words and meaning through their closeness in the real-number vector space.

More recently, multimodal deep learning techniques [5] have been introduced to enhance the performance of deep models that focus solely on a single-modal data type. In this paper, we propose a multimodal deep learning framework that incorporates sequential information from audio and textual models, where different deep features are extracted from each modality using the pre-trained Convolutional Neural Network (CNN) models and word-to-vector technique. Subsequently, Long Short-Term Memory (LSTM) neural networks are applied to leverage the sequential relationships, particularly for text and audio data. We use CNNs to build our fusion model to incorporate the classification ranking scores produced by data from single modalities. Finally, the proposed framework is applied to classify the videos in a disaster-related video dataset as a certain disaster-related concept (flood, storm, etc.).

The contributions of the proposed framework are as follows. a) A multimodal deep learning framework that incorporates sequential information from both audio and textual models; b) For the audio model, an effective and efficient deep learning model is utilized to extract the most discriminative and high-level feature representations that we extend through a time distributed fully connected layer and the subsequent LSTM layers. For the textual model, a pre-trained word embedding layer is used with a stacked LSTM model to generate the video-level concepts; and c) A novel two-stage fusion technique is proposed based on the frame-level image, audio, and video-level information by building a CNN model. Most notably, the image model predictions are incorporated into the audio model to adjust the classification ranking scores based on the reliability of the different

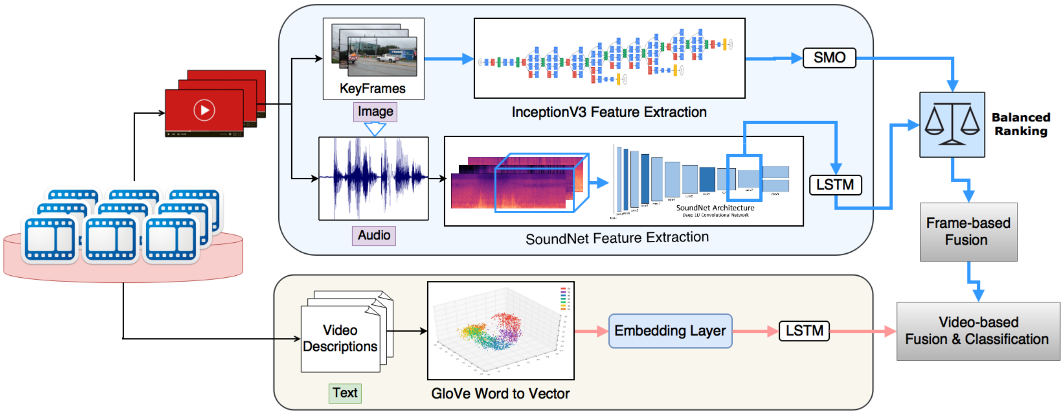

Figure 1. The proposed framework

predicted sound classes.

The remainder of this paper is organized as follows. Section 2 reviews the existing approaches for video classification. A detailed description of each component in the overall framework is presented in Section 3. The evaluation results are shown in Section 4 and followed by the conclusions in Section 5.

2. Related Work

The advances in multimedia research have sparked the interests in improving the detection and classification from data in closely related modalities. Video classification has been positively impacted by the improvements in the detection and classification of objects within images [2][3][4][7][8][12]. In [6], CNNs, having the best advantage in image classification tasks, are proposed to classify videos from a dataset comprised of over 480 sports videos. In contrast to CNNs, Recurrent Neural Networks (RNNs) show promising performance in handling and modeling temporal and/or sequential behavior. Among the most frequently used RNN models, the LSTM networks have shown its promise in speech recognition, language modeling, and more generally, any classification or prediction task where the problem has sequential or temporal traits.

The introduction of multimodal deep learning techniques enables a significant improvement compared with using a single modality alone. This motivates the researchers to build deep networks that could learn, improve, and fuse knowledge in order to achieve a higher prediction accuracy when different modalities (image, audio, text, etc.) share similar semantic concepts. Recently, a novel approach proposed in [15] leverages the advantages of a hybrid framework that learns features from both static data (images) and optical flows. Multimodal deep learning approaches encompassing data modalities beyond image, audio, and video are still very few. Since text data can be obtained as easily as audio and video, it can also help to improve the accuracy in deep learning frameworks. Global Vectors for Word Representation (GloVe) [10] is a favorite technique that works with word embedding and maps text words into a real vector domain.

3. The Proposed Framework

Figure 1 illustrates the overall framework which includes three deep learning models for the different data modalities and a two-stage fusion model. The outputs from the corresponding deep learning models predict potential semantic concepts by providing ranking scores. The scores (or probabilities) are taken as the inputs by the fusion model, which considers both frame-level and video-level concepts.

3.1. Preprocessing

Video key frame extraction is the first step in preprocessing. Each key frame represents the idea of each video shot. A shot boundary detection technique is applied to identify the boundaries of each video shot automatically. Multiple key frames will be kept for one video shot if the variations between them are significant. We also reduce the duplication of similar key frames as well as blurred and noisy ones. InceptionV3 [14], a pre-trained CNN model, is used to extract deep features from the key frames (images).

In order to use the labels (semantic concepts) of the key frames (images) as the references to the audio model, we first extract the full audio tracks from the raw videos with a sampling frequency of 16000 Hz. The metadata of the video and the frame numbers of the key frames are used to calculate the starting and ending points of the audio clips. Algorithm 1 shows the mechanism we used to slice the audio clips from the full audio tracks, which also guarantees

Algorithm 1: Get audio clips from full audio tracks

```
1  begin
2      $audio\_metadata \longleftarrow initialize()$;
3      foreach $track \in audio\_metadata$ do
4          $track\_d \longleftarrow load\_audio(track.vid)$;
5          $fps \longleftarrow get(video\_fps, track.vid)$;
6          $ref\_time \longleftarrow track.fid/fps$;
7          if $ref\_time - span < 0$ then
8              $save\_clip(track\_d, 0, duration, track)$;
9              return;
10         else
11             $start \longleftarrow ref\_time - span$;
12         $track\_length \longleftarrow len(track\_data/1000)$;
13         if $ref\_time + span > track\_length$ then
14             $last \longleftarrow track\_length - ref\_time$;
15             $start \longleftarrow ref\_time - (duration - last)$;
16             $end \longleftarrow track\_length$;
17             $save\_clip(track\_data, start, end, track)$;
18             return;
19         else
20             $end \longleftarrow ref\_time + span$;
21         $save\_clip(track\_d, start, end, track)$;
```

that such clips are generated accurately in order to get sound waves that really match with the visual concepts. The inputs include a list of raw audios with the corresponding *frames per second (fps)* rate (frame rate) for the associated videos.

The *initialize()* function in line 2 generates a key-value paired dictionary *audio_metadata*, where the keys represent the video ID (*vid*) and the values represent the frame ID (*fid*) for that specific video. The variable *duration* holds the value in seconds of the desired audio clip. Additionally, *span* holds the duration of each interval before and after the specified key frame. Starting from line 3, the program loops to process the entire dictionary. The first steps for every audio track (*track*) consist of: a) loading the audio through Pydub (line 4); b) getting the frame rate of the video (line 5); and c) calculating the temporal reference for that specific *fid* (line 6). The variable *ref_time* contains the value in seconds of the current key frame. Lines 7-9 handle key frames that are close to the beginning of the audio track. Similarly, lines 13-18 handle case where the key frame is close to the end of the audio track. This guarantees that all the audio clips have a duration of exactly eight seconds. In line 14, the variable *last* holds the value in seconds from the current key frame being processed until the end of the audio track. We use this value to determine how much we need in order to build an eight-second audio clip. *save_clip* is a helper function that slices and saves the audio clip through Pydub [11].

SoundNet [1], a deep learning model trained on more than 2 million unlabeled videos by using transfer learning to classify audios is used in our deep feature extraction process to extract audio features from the clips that we obtained in the previous step. The deep features used as the inputs to our audio model were extracted from the *conv7* layer using their 8 layer model, which show good capabilities to detect high-level concepts, such as natural sounds (water streams, underwater, etc.) and human-related sounds (speech, talking, cheering, etc.). The features form a matrix with size TIME×DIM (5×1024), where TIME is the number of samples in the input audio clip and DIM represents the number of filters that are applied to the *conv7* layer.

3.2. Frame-based Model

Linear kernel Support Vector Machines (SVMs) are popularly used to replace the softmax layer as the prediction layer of several deep learning models. As an advanced version of SVMs, the Sequential Minimal Optimization (SMO) algorithm speeds up the training process by avoiding matrix computations and scaling the training set size for different test problems. Our proposed framework uses an SMO classifier with linear kernel to process the deep features for the key frames that we obtained through InceptionV3 and outputs the label prediction probabilities for each instance. We use these probabilities as the input to the first stage of our fusion model and at the same time, as a guide for the model to take the scores from the audio model with an adjustable reliability.

In order to overcome the limited capability of the existing audio models, which detect few concepts as compared to the image models, we extended the SoundNet model to detect natural sounds and other activities by considering the sequential characteristics of the features. The audio model presents a higher accuracy in comparison to using SMO as the output layer on all the frame-level concepts.

The audio model aims to improve the performance of the frame-level classification by adding the capability of detecting scenes that are easily recognizable through sounds but harder to recognize through vision. The audio model consists of a fully connected Dense layer with 512 outputs, wrapped in a TimeDistributed layer, and then the subsequent LSTM and output layers. The Dense layer on top of an LSTM performs input compression before running it through the subsequent RNNs. Based on the output we generate from the audio features extraction step, the LSTM model takes the input as 5 timestamps, with each one containing 512 features. The LSTM model takes sequential data to learn how the changes of features in a temporal manner generate a better understanding of the audios in order to classify different kinds of sounds. RMSprop optimizer is used to compile the model with a customized learning rate of 0.0001.

3.3. Video-based Text Model

To get the relationship between a video description and a video concept, a text classification model that uses a pre-trained word embedding layer with stacked LSTM is introduced. We select the most frequent 5000 words from the video descriptions and each word is then mapped into a real-valued vector of length 200, which is generated by using 400k words computed from a 2014 dump of the English version of Wikipedia. Since the sequence length (i.e., the number of words) in each video description varies, we constrain each one to be of 1000 words, truncating long descriptions and padding the shorter ones with zeros.

On top of the text model, the embedding layer uses word vectors of length 200. The next layer is the LSTM layer with 256 memory units (smart neurons). We add another LSTM layer that receives the sequential output from the previous layer in order to increase the depth of the model. By using additional hidden layers to build a stacked LSTM network, the model recombines the learned representation from prior layers and creates new representations at higher levels of abstraction. The model is designed to learn different patterns, such as people describing various semantic concepts with the emotional changes. Adam optimizer is used to compile the model with a customized learning rate of 0.0001.

3.4. Two-stage Fusion Model

The fusion model has two stages. The first stage concludes the frame-level concept predictions (for both image and audio models) and outputs the probabilities for potential video-level concepts. The latter one takes the textual model outputs that predict the video concept directly and integrate them with the results from the previous fusion stage to generate the final output.

Each single-modality has its own limitations. For example, the image model has the advantages on detecting static objects, but presents a limitation in scene detection which requires temporal information. On the other hand, the audio model has the advantages on detecting natural and human sounds, but the complexities arise when trying to detect activities that only produce few sounds or noise, such as near-silent situations. Inspired by the strengths and advantages of the different models, we propose a fusion algorithm that considers both the reliability and limitation factors of each modality for different cases. The purpose is to balance the ranking scores which might dominate the potential concepts in different situations.

Algorithm 2 depicts how the ranking scores of the audio model are changed based on the predictions from the image model before the fusion stage. For each key frame f in video F, we examine the ranking scores and get the predicted labels from both visual and audio (A) models ($\mathcal{L}_1$ and $\mathcal{L}_0$, respectively). The most unlikely concept to be detected, which is identified by the lowest score among all concepts C, is also saved in variable m. If the predicted labels match with each other, the audio model prediction is trusted and a dominating concept penalty factor is applied to the scores, as shown in Equation (1). If a mismatched prediction is detected, the scores from the audio model need to be determined, considering the limitations from both models. $B_{f,c}$ represents the balanced ranking score for the frame-based concept c ($c \in C$) and the associated key frame f, where $\mathcal{L}_0$ is the predicted concept from the audio model ($A_{f,c}$) and $\mathcal{L}_1$ is the predicted concept from the image model ($I_{f,c}$). $|\cdot|$ represents the cardinality of the set.

$$B_{f,c} = \begin{cases} \dfrac{A_{f,\beta(c,\mathcal{L}_0)}}{Rank(c)} & \text{for } \mathcal{L}_0 = \mathcal{L}_1 \\ \dfrac{A_{f,c}}{Rank\prime(c)} & \text{for } c = \mathcal{L}_0 \neq \mathcal{L}_1 \\ \dfrac{A_{f,c}}{|C|} & otherwise \end{cases} \tag{1}$$

$$\text{where } \beta(c, \mathcal{L}_0) = \begin{cases} c & c = \mathcal{L}_0 \\ m & otherwise \end{cases}$$

$$Rank(c) = \begin{cases} 1 & \text{for } c \in \{\text{OtherSounds}\} \\ ln(|R(c)|) & otherwise \end{cases} \tag{2}$$

$$R(c) = \{\{c \in \text{NaturalSounds}\} \cup \{c \in \text{HumanSounds}\}\},$$

$$Rank\prime(c) =$$

$$\begin{cases} |\overline{N \cup H}| & \text{for } c \in \{N\text{: NaturalSounds}\} \\ |H| & \text{for } c \in \{H\text{: HumanSounds}\} \\ |\overline{O \cup H}| & \text{for } c \in \{O\text{: OtherSounds}\}. \end{cases} \tag{3}$$

$\beta(c, \mathcal{L}_0)$ is an activation function which selects the concept with either the highest or the lowest ranking score from the audio model, depending on whether the concept is a match or not. The function $Rank(c)$ returns a penalty factor for the frame-level concepts that belong to either human sounds or natural sounds (as shown in Equation (2)). Natural logarithm is used in the equation in order to guarantee the return of a penalty factor by preventing the divisor in the first case of Equation (1) from being equal to 1. Considering the capability of the audio model, no penalty factor will be applied when there is a match for other sounds. Equation (3) is the $Rank\prime(c)$ function that returns a coefficient based on the number of image concepts associated to the current audio concept in case of a concept mismatch.

Based on our observations, the audio model can better differentiate human and natural sounds from other sounds. However, the ranking scores might be dominated by natural sounds since they are frequently present as the background sound in most of the disaster-related events and activities. To remediate this imbalance scenario, the penalty factor for different sound types will be the opposite of the accuracy of the model. This way, natural sounds will always take the biggest penalty factor compared to other sounds.

Algorithm 2: Frame-level audio rank balancing

```
for $f \in F$ do
    $\mathcal{L}_0 \longleftarrow \underset{c \in C}{\operatorname{argmax}} A_{f,c}$;
    $\mathcal{L}_1 \longleftarrow \underset{c \in C}{\operatorname{argmax}} I_{f,c}$;
    $m \longleftarrow \underset{c \in C}{\operatorname{argmin}} A_{f,c}$;
    for $c \in C$ do
        if $\mathcal{L}_0 \neq \mathcal{L}_1 \bigcap c = \mathcal{L}_0$ then
            $A_{f,c} = min(A_{f,m}, B_{f,c})$
        else
            $A_{f,c} = B_{f,c}$
```

The grouped and balanced ranking scores from both image and audio models are the inputs of the first stage of the fusion model, with the first convolutional layer configured with a stride of 2 and a kernel of size 10. The network is trained using a RMSprop optimizer with the default learning rate to preserve the sequential capabilities of the data. During the second fusion stage, the text classification model results will be integrated with the predicted conclusions from the frame-level modalities. Since there are some videos that do not provide text information, our proposed framework also has the ability to deal with missing values at this stage. If the text information for the related video exists, the prediction from the text model will be integrated into the network. Otherwise, the results from the previous model's outputs will be directly used. In this model, two Dropout layers (with 0.7 and 0.4 dropout rates, respectively) are added after the Flattened and Dense layers in order to prevent overfitting. The network is also trained using a RMSprop optimizer with default learning rate.

4. Experiments and Analysis

The experiments were conducted by using a dataset which includes almost 400 Hurricane Harvey-related videos with associated text information, namely video title and description, which we collected from YouTube in 2017. Table 1 shows the frame-level, video-level concepts, and general audio concepts (grouped and mapped to the frame-level concepts). The dataset is split into training (80%) and testing (20%) sets randomly on the condition of maintaining the frame-level concepts within an approximately similar distribution. At the same time, the key frames (images) and audio clips that correspond to a video will only appear in either the training or testing set.

Table 1. Image (frame-level), general audio, and video-level concepts across different modality datasets.

No.	Image Concepts	Video Concepts
1	Building Collapse	Situation Reporting
2	Flood	Emergency Response
3	Human Relief	Human Relief
4	Damage	Preparation
5	Speak/Interview	Disaster Scene
6	Prepare	Demonstration
7	Briefings	Victim Situation
8	Demonstration	Damage Situation
9	Emergency Response	Volunteer Activity
10	Volunteer Activity	
11	Storm	
12	Road Debris	**Audio Concepts**
13	Regular Surrounding	Natural Sounds (**2, 11**)
14	Victim/Refugee	Human Sounds (**5, 7, 8**)
15	Daily Necessaries	Other Sounds (**1, 3, 4, 6, 9, 10, 12-16**)
16	Animals	

The evaluation metrics used in our experiments of multi-class classification are Accuracy (ACC.) and Label Ranking Average Precision (LRAP) [9]. LRAP was originally used in multi-label ranking problems, where the goal is to give better ranks to the labels associated to each sample. In this study, there is exactly one relevant label per sample, which makes LRAP equivalent to the mean reciprocal rank. Let I be the total number of instances and $|C|$ be the total number of concepts. Formally, given a binary indicator matrix of the ground truth labels $y \in \mathcal{R}^{I \times |C|}$ and the score associated with each label $\hat{f} \in \mathcal{R}^{I \times |C|}$, the label ranking average precision is defined as:

$$LRAP(y, \hat{f}) = \frac{1}{I} \sum_{i=0}^{I-1} \sum_{j:y_{ij}=1} \frac{|\mathcal{L}_{ij}|}{\text{rank}_{ij}} \tag{4}$$

with $\mathcal{L}_{ij} = \left\{k : y_{ik} = 1, \hat{f}_{ik} \geq \hat{f}_{ij}\right\}, \text{rank}_{ij} = \left|\left\{k : \hat{f}_{ik} \geq \hat{f}_{ij}\right\}\right|$.

As shown in Table 2, the prediction using image features alone achieves higher ACC. and LRAP compared to the prediction using audio features. Through the LSTM audio model, we show the strength of our sub-model compared to the simple output layer (SMO) that does not consider sequential information. However, if we use our sequential fusion model directly on the image and audio models' outputs,

Table 2. Evaluation results

Methods	ACC.	LRAP	# of concepts
Frame-based audio (SMO)	0.261	0.448	16
Frame-based audio (LSTM Model)	0.283	0.470	16
Image Model	0.346	0.534	9
Audio+Image Fusion	0.345	0.525	9
Video-based text (LSTM Model)	0.366	0.530	9
Proposed Framework	**0.457**	**0.596**	9

it shows how the contradiction of the predictions in different modalities will degrade the results of the entire framework, which leads to a decrease in accuracy. By applying our proposed two-stage fusion model, the fusion results gain strength from both image and audio models and reduce the effects of contradicting predictions. The proposed framework, through the fusion of predictions from three modalities, improves the accuracy by more than 10%.

5. Conclusions

Multimodal deep learning has recently attracted a lot of attention. This paper proposes a novel multimodal deep learning framework that considers sequential information from both audio and textual models. Furthermore, a two-stage fusion technique is proposed that utilizes the frame-level image, audio, and video-level information by building a CNN model. In our experiments, we demonstrate how the proposed framework improves the accuracy from single-modal models and illustrate the capability of fusion strategies by taking into account the prediction contradictions across modalities in order to balance the reliability for different class predictions.

Acknowledgments

This research is partially supported by NSF CNS-1461926.

References

[1] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*, pages 892–900, 2016.

[2] S.-C. Chen, M.-L. Shyu, and R. Kashyap. Augmented transition network as a semantic model for video data. *International Journal of Networking and Information Systems*, 3(1):9–25, 2000.

[3] S.-C. Chen, M.-L. Shyu, and C. Zhang. Innovative shot boundary detection for video indexing. In S. Deb, editor, *Video Data Management and Information Retrieval*, pages 217–236. Idea Group Publishing, 2005.

[4] X. Chen, C. Zhang, S.-C. Chen, and S. Rubin. A human-centered multiple instance learning framework for semantic video retrieval. *IEEE Trans. on Systems, Man, and Cybernetics, Part C*, 39(2):228–233, 2009.

[5] S. E. Kahou, X. Bouthillier, P. Lamblin, C. Gulcehre, V. Michalski, K. Konda, S. Jean, P. Froumenty, Y. Dauphin, N. Boulanger-Lewandowski, et al. Emonets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, 10(2):99–111, 2016.

[6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[7] L. Lin and M.-L. Shyu. Weighted association rule mining for video semantic detection. *Int. J. Multimed. Data Eng. Manag.*, 1(1):37–54, Jan. 2010.

[8] T. Meng and M.-L. Shyu. Leveraging concept association network for multimedia rare concept mining and retrieval. In *Proceedings of the IEEE International Conference on Multimedia & Expo*, pages 860–865, July 2012.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[11] J. Roberts. Pydub. https://github.com/jiaaro/pydub, 2011.

[12] M.-L. Shyu, K. Sarinnapakorn, I. Kuruppu-Appuhamilage, S.-C. Chen, L. Chang, and T. Goldring. Handling nominal features in anomaly intrusion detection problems. In *15th International Workshop on Research Issues in Data Engineering: Stream Data Mining and Applications*, pages 55–62, 2005.

[13] M. Sundermeyer, R. Schlüter, and H. Ney. Lstm neural networks for language modeling. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[15] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 461–470. ACM, 2015.