Large scale video classification using both visual and audio features on YouTube-8M dataset

Emma An
Stanford University
anran@stanford.edu

Anqi Ji
Stanford University
anqi@stanford.edu

Edward Ng
Stanford University
edjng@stanford.edu

Abstract

Convolutional Neural Networks (CNNs) have achieved state-of-art in image classification and have been progressing rapidly in the field of video classification and audio understanding in recent years. Encouraged by these results, we propose jointly using video and audio features in order to provide a promising classifier for the Youtube-8M Kaggle challenge - a video classification task for a dataset of 7 million YouTube videos belonging to 4716 classes[2]. In this paper, we explore several models of different combination of video-level visual and audio features. Our best model uses the mixture of experts (MoE) to receive three inputs. They are video level visual feature only, audio feature only and a concatenation of visual and audio feature. Then we apply a dense layer followed by a ReLu activation layer. The gating network we applied in MoE uses softmax function. We train our model with Tensorflow using 1 GPU on Google Cloud Platform. The average training time is 1 hour 15 minutes. As a result, we achieve Avg Hit@1 of 0.84, Avg PERR (average precision at equal recall rate) of 0.709, and mAP (mean average precision) of 0.415 compared to the best performing baseline proposed in [2] with Avg Hit@1 0.645, Avg PERR of 0.573, mAP of 0.266.

1. Introduction

Video has become an indispensable form of media in the modern society. According to YouTube company statistics [1], every minute there are 300 hours of video uploaded to YouTube and every day there are nearly 5 billion videos being watched. Not only the amount of videos we are dealing with is immense, but also the themes of the video has become extremely diverse. The types of videos we encounter in daily life range from entertainment use such as music videos, movies and games, educational use of lectures and experiments, to the many newly emerging technologies such as drones and autonomous cars. Under such background, an efficient method to solve large-scale video classification is desired, which could in turn be applied to content discovery and filtering.

Video classification is an inherently difficult task for mainly three reasons. First, the dataset for video classification are usually limited to a particular scene and separate for video and audio features. The most well studied video datasets, such as Sports-1M [10], ActivityNet[7], UCF-101[20] are all confined to a certain theme of videos. Thus, their models are more suitable in very specific classification task in that theme than generic classification with a large number of classes. There are also some audio datasets that have been extensively explored such as NOISET-92 [22] and AENet [21]. These datasets also have the same limitation as the video datasets.

Second, the trade-off between computational cost and with the accuracy of the model inherently exist. On top of the challenges of classifying images, the additional temporal dimension is critical in understanding the theme of a video. Having a model capture the temporal information across frames may provide a more accurate model at the cost of computational time to a degree where processing time becomes infeasible. For example, temporal fusion frame stacks [10] have proved its success in high mAP but its training time is extremely long (over a month).

Third, the labeling of videos are very subjective. User-generated content is inherently noisy, and we note that the labels applied to a video may also be a source of noise. In some instances, the video may jump erratically between scenes, have odd occlusions of objects, or other such artifacts expected of amateur content. Moreover, the labels can describe specific objects found in the video while excluding others, while labels may wish to capture the overall topic the video while ignoring specific objects or scenes. As a result, creating and classifying a large video dataset with relatively stable and object labeling style is very difficult - which is challenging when attempting to produce a reasonable video classifier.

In this paper, we test our video classifier against YouTube-8M, the largest video dataset, having been established in September 27, 2016. This dataset contains 7 million video URLs, 450 000 hours of video, 3.2 billion audio/visual features, and approximately 5000 labels, containing both video-level, frame-level visual and audio features. The labels were created by the YouTube video annotation system. The use of human rated labels were not used across the dataset, which is explained away by Abu El-Hajia et. al [2] by running the same models on a smaller, human-rated dataset.

Our contributions can be summarized as follows:

- We provide several models of using both audio and visual features to classify YouTube videos. We demonstrate that the additional audio information in the training process significantly improves the model performance.
- We analyze specific examples of our training model to demonstrate successful, non-obvious examples, and failure examples.
- We compare the performance of different models against the classification benchmarks originally set by [2]
- We propose additional models given additional time and resources, and even beyond the features provided by the YouTube-8M dataset

2. Related Papers

Various but surprisingly limited amount of research has been done in video classification. Most of research up until recent research is based on state of art image classification techniques [12][11] and use only visual information.

However, they generally differ in that video classification models naturally include temporal information. For example, multiple temporal information fusion architectures were experimented on the Sports-1M dataset[10], where the first convolutional layer in conventional image classification CNN was modified to extend through several frames instead of a single frame. As a result, this model was demonstrated to have significant accuracy improvement compared to strong feature-based baselines (55.3% to 63.9%) on Sport-1M dataset.

Recurrent neural network (RNN) [8] using Long Short Term Memory (LSTM) [19], having been used rather successfully in the language domain, has also been proved to be an effective way to utilize the sequential information in videos, [14], including the UCF-101 [20]and Sport 1-M[10]. In the RNN structure using LSTM, by using internal memory cells to store information of frames in a certain temporal window, parameters can be shared and transferred overtime. Using such a structure allows us to discover long-term temporal relationships which is critical in video understanding. Very similar to LSTM, Gated Recurrent Unit (GRU) [5][4] also have gating options to process sequential information. Compared to LSTM, GRU use of 2 gates instead of 3 and does not possess internal memory which makes the computation using GRU faster than LSTM with comparable performance.

Spatial-temporal words [16] is another approach proved to be effective in human activity categorization. The idea of spatial-temporal words resembles Bag of Words (BoG) [17] in the field of language processing, in which we view a video as a collection of words. In this method, spatial temporal interest points are extracted from local video patches and are then used to predicted local labels. Then, a vocabulary book is constructed by clustering using k-means or Euclidean distance. Finally, the category of the video is decided based on probabilistic Latent Semantic Analysis (pLSA) graphical model [9].

However, as mentioned before, these approaches are studied on different datasets and have very different purposes. For example, in the Sports-1M dataset, the target classes are 487 sports. All the videos share similarities of background of stadium, audience, players and most of the information comes from the sequence of motion and the speed of these motions. In this context, temporal fusion would be a very effective model. By contrast, in understanding activities, the sequence of behaviors are usually more complicated. In this case, spending more effort in understanding the topic of video by combining frame information in explicit ways, seems more reasonable.

However, in the YouTube-8M dataset, the topics of video are much more diverse. Beside sports videos and human activities, there are many other videos containing video gameplay with commentary, music videos with static images, speeches, clips of TV shows, etc. With such diversity in content, we can imagine that understanding the sequence of scenes may not help with understanding the topic of the video. Additional features, including from an independent data structure, such as audio information should be jointly studied to complement this loss. Also, the research based on understanding discrete frames are always extremely computationally expensive. Some of the models requires training time of more than a month to train[10]. Adding audio features and simplifying visual features can reduce the training time significantly.

Moreover, there is also extensive research in understanding audio, particularly in speech recognition. In the field of speech recognition, many low level features such as short-time energy, frequency-pitch, frequency-centroid Mel Frequency Cepstral Coefficients (MFCC) were investigated[26]. However, these features were not designed for generic video classification. Recently, there are more researches exploring fusing audio information with visual information for video classification [24][21].

However, these studies were done in small datasets that are not as generic as YouTube-8M.

In this paper, we propose and compare different models using video-level visual and audio features on YouTube-8M datasets. The methods and models we use will be explained in detail in the following section.

3. Methods

In this section, we describe the methods behind the models used in our video classfication model, including some methods that we believe held promise, but however had insufficient time or resources to complete.

3.1. Logistic Regression

Logistic Regression [13] is regarded as our baseline for consuming video-level features. Given averaged RGB value for all frames as video-level feature x_i of the testing video i and the probability of the entity j as $\sigma(w_j^T x_i)$, we trained this model by minimizing the total log loss of the training data with deterministic parameter weights $\mathbf{w}$:

$$\lambda||w_j||_2^2 + \sum_{i=1}^{N} L(y_i, \sigma(w_j^T x_i))$$

where the sigmoid function $\sigma(x)$ is given as :

$$\sigma(x) = 1/(1 + exp(-x))$$

and the cross entropy loss function is :

$$\frac{1}{N}\sum_{i=0}^{N} \mathbf{y_i}\log(\mathbf{\hat{y}_i} + \epsilon) + (1 - \mathbf{y_i})\log(1 - (\mathbf{\hat{y}_i} + \epsilon))$$

In general, it is a linear projection of the video-level features into the label space, followed by a sigmoid function to convert log values to probabilities.

3.2. Dense Layer

As each dense layer indicates a matrix multiplication and to cooperate with different needs of models, we use dense layer to change the dimensions of the vector. It can achieve mathematical transformation to vector input using:

$$outputs = activation(inputs.kernel + bias)$$

where kernel is trainable parameter and bias is provided by the layer. More specifically, suppose $u \in R^n andw \in R^{n*m}$, and if we apply dense layer on u with kernel w, then we have $u.Tw \in R^m$, which is now a m-dimensional vector output.

3.3. Mixture of Experts

The mixture of experts(MoE) [3] is a binary classifier,consisting of a number of experts, each a simple feed-forward neural network $E_1, E_2, ..., E_n$, and a trainable gating network G which selects a sparse combination of the experts to process each input, and outputs a sparse n-dimensional vector [18], as shown in the figure below.

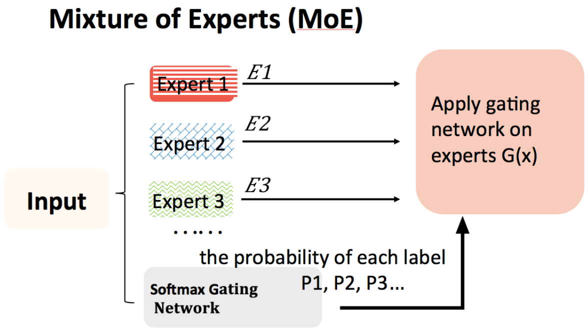


Fig 1. Mixture of Experts. Each expert takes in same input and processes independently. A gating network is then applied to make the final prediction.

Each experts (neural network E_i) has its own trainable parameters, therefore we only require them to output the vectors with same dimensions, and this is the place where dense layer would be applied. If we denote the output of gating network as $G(x)$ and the output of i-th expert network as $E_i(x)$, then the output of MoE model is:

$$y = \sum_{i=1}^{n} G_i(x)E_i(x)$$

Thus the computation result from $E_i(x)$ is chosen based on the sparsity of the output G(x). i.e. if $G_i(x) = 0$ then there is no need to calculate $E_i(x)$. We chose the softmax function and sigmoid as the gating network, where softmax is used to model the probability of choosing i-th expert and sigmoid function to model the existence of the entity:

$$p(x_j|x) = s(x_j) = \frac{e_j^s}{\sum_{i=1}^{n} e_i^s}$$

$$P_\sigma(x) = \sum_j p(x_j|x)\sigma(w_j^T x)$$

Here, W_j is also a trainable weight matrix and we use final probabilities to make the prediction. In our project, we used audio and visual features as input and the corresponding experts are simple weight matrices, which is similar to the parameterized weight matrix but surprisingly works well.

3.4. Ensemble Model

Ensemble model takes a number of related but different analytical model and synthesizes the results into one single output in order to achieve improvement on accuracy of single predicting algorithms.

According to the comparison of different models, three models are selected as experts in MoE model. Details are shown below.

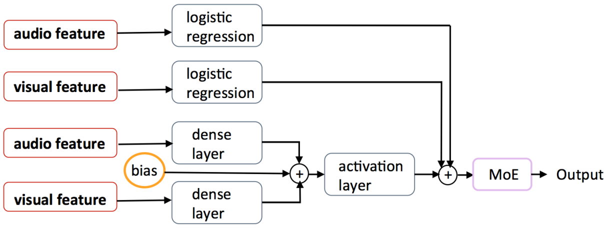


Fig 2. Customized MoE Model. Two "experts" are logistic regression models of audio features and visual features respectively. The third expert takes in concatenated audio and visual features, followed by dense layer and activation function.

3.5. Long Short Term Memory

Long Short Term Memory (LSTM) unit was initially introduced in 1997 and it was designed to save vanishing gradients through gating network as seen in a more basic recurrent neural network. It has forget gate *f*, input gate *i*, gate gate *g*, output gate *o* and internal memory unit *c*. Among above, three gates *i*, *f*, *o* have same dimensions but different parameters, and they are squashed between 0 and 1. More specifically, the input gate *i* defines the percentage of current input used in the newly computed state. The forget *f* gate defines percentage of the previous state to be let through. The output gate *o* defines the percentage of the internal state to be exposed to the external network. "Gate gate" *g* is a candidate hidden state that is computed based on the current input and the previous hidden state.

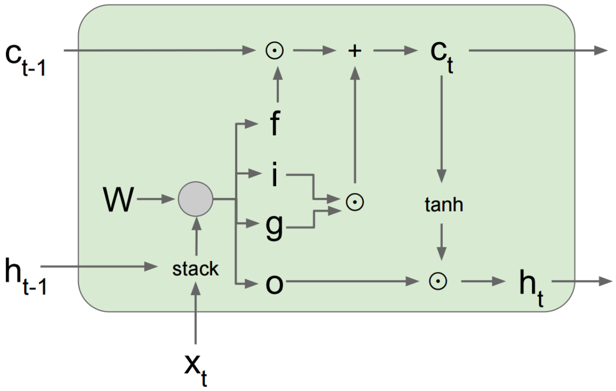


Fig 3. LSTM gating mechanism. It modulates the gradient flow with saved memory content

The memory unit and hidden state are updated as below (Here, * indicates point-wise multiplication):

$$c_t = f * c_{t-1} + i * g$$

$$h_t = o * tanh(c_t)$$

3.6. Gated Recurrent Units

Gated Recurrent Units, known as GRU, has similar structure as a LSTM layer. Its gating units modulate the flow of information inside the unit, but without memory cells as LSTM does.[4]

A GRU has two gates, a reset gate *r*, and an update gate *z*. the reset gate determines how to combine the new input with the previous memory, and the update gate defines how much of the previous memory to keep around. If reset gate is set to all 1's and update gate is set to all 0's, it now changes back to basic RNN model. The equations are given as below:

$$z = \sigma(x_t U^z + s_{t-1} W^z)$$

$$r = \sigma(x_t U^r + s_{t-1} W^r)$$

$$h = tanh(x_t U^h + s_{t-1} r W^h)$$

$$s_t = (1 - z)h + z s_{t-1}$$

To better explain the responsibility of reset and update gates, the figure below shows how to compute the output.

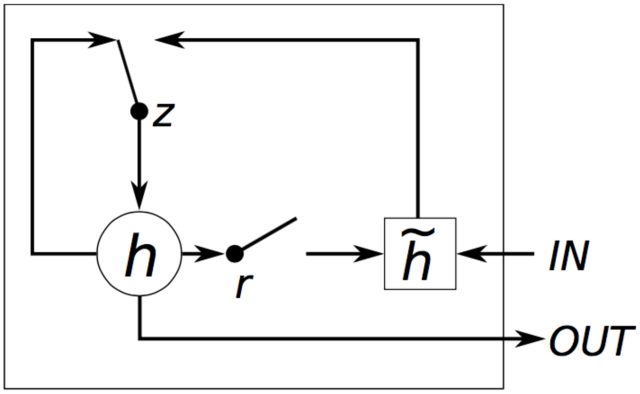


Fig 4. GRU Gating. The combination of r and z has the same performance as the reset gate in LSTM.

GRU controls the information flow from the previous activation when computing the new, candidate activation, but does not independently control the amount of the candidate activation being added (the control is tied via the update gate).

Since GRUs have fewer parameters (U and W are smaller) and thus may train a bit faster or need less data to generalize. It is more reasonable to use GRU instead of LSTM due to the large capacity of dataset and time limitations. However,the greater expressive power of LSTMs may lead to better results given enough resource.

4. Dataset and Features

YouTube-8M is the so far the largest multi-label video classification dataset. It is composed of 7 million videos, with a vocabulary of 4716 visual entities. Each video was tagged on an average of 3.2 labels. It should be noted that the distribution of 4716 classes are rather uneven.

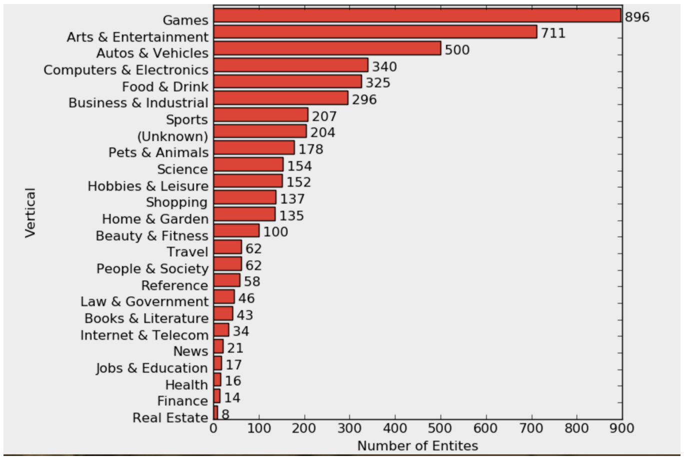


Fig 5. Distribution of YouTube-8M classes. The top 24 popular classes are shown in this image. We can see the distribution of labels is very uneven.

All the videos in this dataset are clipped into 6 minutes length, with frames captured at one second intervals. The labels were obtained through feeding frames into the Inception network and fetching the ReLu activation of the last hidden layer followed by the classification layer, producing a feature vector of size 2048. Afterwards, PCA, whitening and quantization are applied to reduce the dimension to 256. It is validated that such downsizing would not affect the training performance much since training on full-size data only increases the evaluation metrics by less than 1% [2].

At the same time, task-independent fixed-length video-level feature vectors are also derived from the frame-level features. Its compactness can help reduce the training data size and independence of video labels can generalize better to new tasks or video domains. In this paper, we use video-level visual and audio feature vectors to train our network.

5. Experiments and Results

In this section, we'll describe the metrics that are used to measure the performance of the models, the baseline results found in [2], the results of our models, and an analysis of both successful and failure cases.

5.1. Implementation

Initially we attempted to build a pipeline to extract features from raw video, in which a Python script would download each YouTube video, extract frames at one frame per second with the appropriate vector representation (i.e. RGB images). However we soon found this to be prohibitively time-consuming and rather focused on developing models on the existing features provided by the YouTube-8M dataset, which unfortunately limited our approach to the problem.

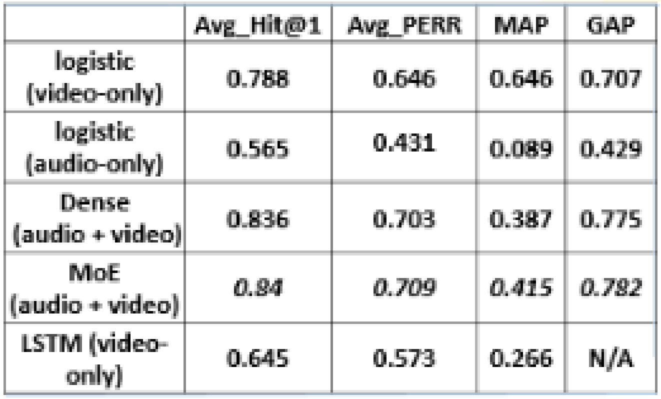
Our models were developed using the Tensorflow framework, based off of the template code provided by Google Research in their own development of baseline models. All models were run on the Google Cloud Machine Learning API. The partition for training, evaluation, and test sets were divided into 70/20/10 respectively as defined by the Kaggle challenge.

5.2. Evaluation Metrics

There are a number of metrics that we used to measure the performance of each model, which are described in [2].

5.3. Results on YouTube-8M

Our results on the YouTube-8M dataset readily surpassed the baseline models provided by [2], having been provided the addition of audio features.



	Avg_Hit@1	Avg_PERR	MAP	GAP
logistic (video-only)	0.788	0.646	0.646	0.707
logistic (audio-only)	0.565	0.431	0.089	0.429
Dense (audio + video)	0.836	0.703	0.387	0.775
MoE (audio + video)	*0.84*	*0.709*	*0.415*	*0.782*
LSTM (video-only)	0.645	0.573	0.266	N/A

Table 1. Results for Hit@1, PERR, mAP, and GAP performance for each tested model. LSTM is the baseline provided by the YouTube-8M paper.

Due to time and monetary constraints, only video-level features were trained, as frame-level features were deemed to be too costly to train for marginal value. However, frame-level features would have been necessary to achieve the top scoring model as part of the Kaggle challenge.

5.4. Examples

We extracted a number of examples to observe how our models were performing, and we found that even with a fairly simple model, and on the video-level features alone, the models were surprisingly accurate, even providing labels that were even more agreeable than those provided by the YouTube-8M dataset.

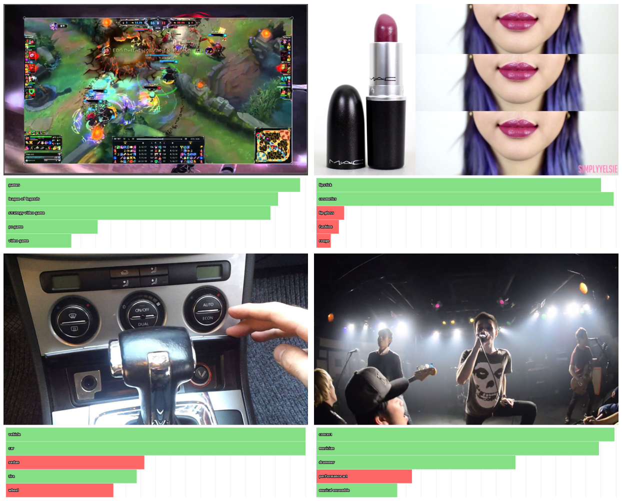


Fig 6. Successful predictions on videos from the Youtube-8M dataset. Length of the bars represent confidence level of the model for the given label. Green presents matching ground-truth, while red represents a mismatch. Ordering of the labels is defined by the confidence scoring of the model rather than the order of the ground-truth labels.

Details such as the specific game, or particular objects in the video, were largely labeled despite not being in the set of ground-truth labels, while high-level labels of videos were generally accurately applied. The classifier appeared to perform well on coherent videos with longer sequences and focusing on a single topic.

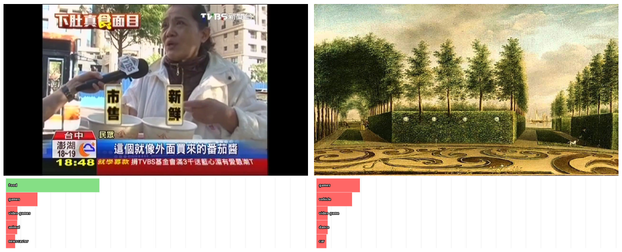


Fig 7. Failed predictions on videos from the Youtube-8M dataset.

However, we noted that for videos with a large number of scenes across a range of topics, or for static videos with audio, the classifier did not perform very well, with the model falling back on the distribution of high level topics. Regardless, we found that the behavior of the model was explainable in both the successful and failure modes.

6. Future Work

Given a larger amount of time and resources, there would be a number of techniques that we could attempt to use to improve the accuracy of our models.

The use of frame-level features in recurrent neural network architectures to encode additional video-level features would have likely provided our models with additional accuracy. This was attempted without success - we found that the training of these neural networks were prohibitively expensive for the marginal gain that they would have produced. However given additional time and resources, this can be readily attempted. We would also extend beyond the RNN architectures proposed by prior papers in video classification that have otherwise been used successfully in other domains, including bidirectional LSTM [6].

In addition, the information from the hierarchical nature of the labels, defined by the Google Knowledge Graph, would have perhaps provided a more accurate model. Using hierarchical multilabel classification techniques [25] [23], at minimum, we expect to see a reduction in training time if not an additional gain in accuracy. The application of data augmentation, specifically through a (denoising) multimodal autoencoder [15] or randomizing sequences, could have also perhaps led to higher-level representations that would have been more suitable than those provided by the YouTube-8M dataset.

Moving beyond the scope YouTube-8M dataset features, we would have also liked to use the individual frame information to extract features such as text, optical flow [2], and scene segmentation that we found inaccessible using the PCA and whitened video-level and frame-level features. While the processing time may be prohibitive for a competition, we believe that this information would have led to a much more accurate model and additional flexibility in our architecture.

References

[1] Statistic Brain youtube company statistics. `http://www.statisticbrain.com/youtube-statistics/`. Accessed: 2017-06-01.

[2] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *CoRR*, abs/1609.08675, 2016.

[3] K. Chen, L. Xu, and H. Chi. Improved learning algorithms for mixture of experts in multiclass classification. *Neural networks*, 12(9):1229–1252, 1999.

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

[5] J. Chung, C. Gülçehre, K. Cho, and Y. Bengio. Gated feedback recurrent neural networks. In *ICML*, pages 2067–2075, 2015.

[6] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

[7] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970. IEEE Computer Society, 2015.

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

[9] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[13] P. McCullagh and J. Nelder. *Generalized Linear Models, Second Edition*. Chapman and Hall/CRC Monographs on Statistics and Applied Probability Series. Chapman & Hall, 1989.

[14] J. Y. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. *CoRR*, abs/1503.08909, 2015.

[15] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

[16] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008.

[17] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. *Computer Vision–ECCV 2006*, pages 490–503, 2006.

[18] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[19] S.Hochreiter and J.Schmidhuber. Long short-term memory. pages 9(8)1735–1789. Nov.1997.

[20] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[21] N. Takahashi, M. Gygli, and L. V. Gool. Aenet: Learning deep audio features for video analysis. *CoRR*, abs/1701.00599, 2017.

[22] A. Varga and H. J. Steeneken. Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech communication*, 12(3):247–251, 1993.

[23] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel. Decision trees for hierarchical multi-label classification. *Machine learning*, 73(2):185–214, 2008.

[24] Q. Wu, Z. Wang, F. Deng, Z. Chi, and D. D. Feng. Realistic human action recognition with multimodal feature selection and fusion. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(4):875–885, 2013.

[25] L. Zhang, S. Shah, and I. Kakadiaris. Hierarchical multi-label classification using fully associative ensemble learning. *Pattern Recognition*, 2017.

[26] T. Zhang and C.-C. J. Kuo. Audio content analysis for online audiovisual data segmentation and classification. *IEEE Transactions on speech and audio processing*, 9(4):441–457, 2001.