Multi-Modal Fusion for Moment in Time Video Classification

Hu-Cheng Lee* Sebastian Agethen* Chih-Yu Lin Hsin-Yu Hsu
Pin-Chun Hsu Zhe-Yu Liu Hsin-Li Chu Winston Hsu
National Taiwan University
{r05922174, d01944015, r05922109, r06922087,b03901023}@ntu.edu.tw
j2325138@gmail.com, {b05505004,whsu}@ntu.edu.tw

Abstract

Action recognition in videos remains a challenging problem in the machine learning community. Particularly challenging is the differing degree of intra-class variation between actions: While background information is enough to distinguish certain classes, many others are abstract and require fine-grained knowledge for discrimination. To approach this problem, in this work we evaluate different modalities on the recently published Moments in Time dataset, a collection of one million videos of short length.

1. Introduction

There are hundreds of thousands of activities occurring around us in our daily life. Most of these activities are not only restricted to one person or a single motion, but involve many types of actors in different environments, at different scales, and with many different modalities. If we want to solve problems that are relevant to our real world, it is necessary to develop models that scale to the level of complexity and abstract reasoning that a human processes on a daily basis. We propose a new approach to tackle these challenges. To evaluate our work, we use the *Moments in Time Dataset* [7].

Moments in Time Dataset is a large-scale human-annotated collection of one million short videos corresponding to dynamic events unfolding within three seconds and has a significant intra-class variation among the categories.

The dataset poses a number of challenges that we need to conquer. First, the videos have a diverse set of actors, including people, objects, animals and natural phenomena. Second, the recognition may depend on the social context of ownership and the type of place. For example, picking up an object, and carrying it away while running can be categorized as stealing, saving or delivering, depending on the ownership of the object or the location where the action occurs. Third, the temporal aspect: the same set of frames in a reverse order can actually depict a different action, consider for example `opening` vs. `closing`. Since we want to build a true video understanding model, we need to be able to recognize events across agent classes. In other words, it is necessary to recognize these transformations in a way that will allow them to discriminate between different actions, yet generalize to other agents and settings within the same action.

In this work, we investigate the fusion of features of different modalities. In Section 2, we outline each modality. In Section 3, we discuss the fusion methods, and provide preliminary results on the *Moments in Time Mini* validation set. Finally, Section 4 discusses analytic insights into the dataset based on a simple RGB baseline.

2. Methodology

We investigated a number of modalities of interest for action recognition. We first discuss each modality, and then examine both early and late fusion of these modalities.

2.1. RGB and optical flow

In action recognition, we consider two essential visual concepts: appearances and motions. Most action recognition work uses RGB frame and optical flow as the visual representation respectively. In order to fully utilize the visual contents from videos, a practical approach, introduced by [9], models short temporal snapshots of videos by averaging the predictions from a single RGB frame and a stack of 10 externally computed optical flow frames, which is also known as *Two-stream ConvNets* method.

Temporal Segment Networks There have been many improvements over the basic two-stream architecture, and one of the most well-known method is *Temporal Segment Networks* (TSN) [11]. Instead of working on single frames or frame stacks, TSN operate on a sequence of short snippets

*Equal contribution

sparsely sampled from the entire video. Each snippet in this sequence will generate its own preliminary prediction of the action classes. Then a consensus among the snippets will be derived as the video-level prediction. We use the same settings as the original work for our prediction.

Temporal Relational Reasoning *Temporal Relational Reasoning Network* (TRN) [12] can learn and discover possible temporal relations at multiple time scales. TRN is a general and extensible module that can be used in a plug-and-play fashion with any existing CNN architecture. We also use the same settings for our prediction.

2.2. Sound

Sound is a valueable modality in action recognition. It can not only complement visual observations, but also help add information where vision is not available, i.e., unseen or occluded surroundings.

Feature extraction We use two pretrained models for audio feature extraction: *Audio Event Net* (AENet) [10] and *VGGish* pretrained on AudioSet[4].

To ensure that our sound features are useful for the fusion tasks, we ignore those videos with no audio channels or channels that are muted. We use wav file format with 16kHz sampling rate, 16bit, monoral channel; the codec is PCM S16 LE.

In AENet, the dimensions of extracted features are $(N, 1024)$, where N equals to the total length in seconds. On the other hand, we used the VGGish to save those features into $(N, 3, 128)$ embeddings. It took about 12 hours to extract features for each from the mini training set with one K80 GPU.

We trained 200 linear SVM binary classifiers for each class using the extracted AENet and VGGish features respectively. Besides, we did not perform any preprocessing on the extracted AENet features while we flattened the extracted VGGish features to dimension $(N, 384)$ before we fed them into the SVM classifiers for training and testing. We got distances to the 200 separating hyperplanes after feeding each testing sample into the 200 binary classifiers and use these distances to do classification.

Table 1. Numbers of videos with and without sound

Videos	**With sound**	**Without sound**	**Total**
Training	55,933	44,067	100,000
Validation	6,286	3,714	10,000
Testing	12,776	7,224	20,000

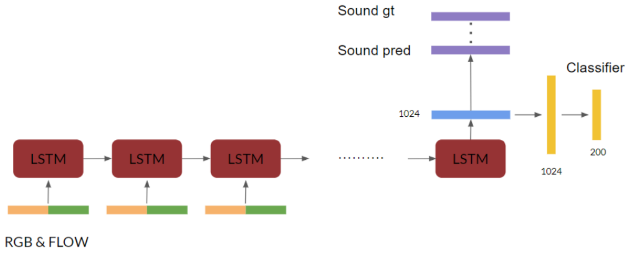

Figure 1. Sound generation with LSTM.

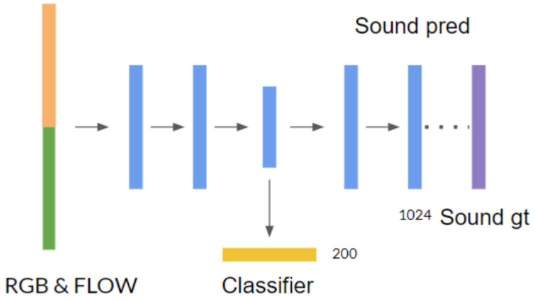

Figure 2. Sound generation with Encoder-Decoder.

Feature generation We found out that not all the videos have sound track. The detailed number of videos with and without sound is listed in Table 1. We can see half of videos do not have sound, but the sound plays an important role in videos. Therefore, we want to generate the sound representation for those videos without sound.

We use two basic structures to generate the sound: LSTM in Figure 1 and encoder-decoder in Figure 2. First, we use the feature representation extracted by TSN as structure input, then go through the structure and get the output feature. The groundtruth sound representation is extracted by AENet and VGGish. In training stage, we use videos with sound to be the training set, and in testing stage, we will generate the sound representation for those videos without sound. We have four kinds of settings: L2 loss+ w/ classifier, L2 loss+ w/o classifier, KL loss+ w/ classifier, KL loss+ w/o classifier. We want to know if label information and different kind of loss are important to the generation.

Table 2. AENet generation with LSTM.

AENet / LSTM	**Top-1 acc.**	**Top-5 acc.**
w/o generation (baseline)	4.41%	11.78%
L2 loss, w/ classifier	4.53%	11.69%
L2 loss, w/o classifier	5.19%	13.44%
KL Div., w/ classifier	4.47%	11.50%
KL Div., w/o classifier	4.45%	11.40%

Table 3. VGGish generation with LSTM.

VGG / LSTM	**Top-1 acc.**	**Top-5 acc.**
w/o generation (baseline)	1.57%	7.29%
L2 loss, w/ classifier	1.54%	6.91%
L2 loss, w/o classifier	1.95%	7.59%
KL Div., w/ classifier	1.59%	6.85%
KL Div., w/o classifier	1.59%	6.83%

Table 4. AENet generation with fully-connected.

AENet / FC	**Top-1 acc.**	**Top-5 acc.**
w/o generation (baseline)	4.41%	11.78%
L2 loss, w/ classifier	4.70%	11.70%
L2 loss, w/o classifier	4.70%	11.70%
KL Div., w/ classifier	4.52%	11.48%
KL Div., w/o classifier	4.55%	11.60%

Table 5. VGGish generation with fully-connected.

VGGish / FC	**Top-1 acc.**	**Top-5 acc.**
w/o generation (baseline)	1.57%	7.29%
L2 loss, w/ classifier	2.19%	7.86%
L2 loss, w/o classifier	2.11%	7.84%
KL Div., w/ classifier	1.71%	7.23%
KL Div., w/o classifier	1.72%	6.90%

Feature generation performance The generation performance is found in Tables 2,4,3,5. We can see that for AENet feature, L2 loss + w/o classifier performs the best, and for Vggish feature, L2 loss + w/ classifier performs the best. Therefore, we choose these two model to generate our sound representation.

2.3. Pose-centric features

Our preliminary evaluation, see also Section 4, shows that classes with large intra-class variations, i.e., more abstract classes, are hard to learn for baseline models. To attempt an improvement of these classes, we learn fine-grained, human pose-based features.

Method. We generate discriminative human pose features with the help of *Recurrent Pose Attention Networks* (RPAN) [2]. Given the convolutional feature maps $\mathbf{C}_t$ of each video frame, attention maps $\boldsymbol{\alpha}_t^J$ are learned for each joint J in a human pose. The learning process is supervised by the inclusion of an l2-regression term. As the Moments in Time dataset does not provide human pose annotations, we employ the human pose detector in [1] to retrieve groundtruth annotations.

For the purposes of this work, we simplify the formulation of $\tilde{\boldsymbol{\alpha}}_t = \left[\tilde{\alpha}_t^0, \cdots, \tilde{\alpha}_t^J\right]$ by dropping the partial parameter sharing used in [2]:

$$\tilde{\boldsymbol{\alpha}}_t = \mathbf{v} *_J \tanh\left(\mathbf{A}_h \cdot \mathbf{h}_{t-1} + \mathbf{A}_c *_D \mathbf{C}_t + b\right) \tag{1}$$

$$\alpha_t^J = softmax\left(\tilde{\alpha}_t^J\right) \tag{2}$$

where $*_J$ denotes a $(1 \times 1 \times J)$ convolution. The term $\mathbf{A}_h \cdot \mathbf{h}_{t-1}$ has dimension $D = 32$ and is therefore broadcasted over the spatial dimensions. Input $\mathbf{h}_{t-1}$ is the previous output of the recurrent network learned on body parts, see below.

Given the attentional maps α_t^J, we can construct human body parts P by summation. We follow the work in [2], and construct five body parts *torso, elbow, wrist, knee, ankle*. More formally, we construct F_t^P:

$$\mathbf{F}_t^P = \sum_{J \in P} \sum_k \boldsymbol{\alpha}_t^J \circ C_t \tag{3}$$

where $\circ$ denotes elemenwise multiplication (attention maps are broadcasted over the channel dimension). The result is a fixed-size descriptor for each body part. These five pose features are then max-pooled to form the input to an LSTM recurrent network, for details please refer to [2].

Performance. The method by itself achieved a top-1 accuracy of 21.0% on Moments in Time Mini dataset. This is largely due to the lack of human poses in many classes, which will result in $\mathbf{F}_t^P = \mathbf{0}$. In fact, using the pose detector in [1], we were not able to extract any pose for roughly 47% of all frames.

2.4. Attribute

We consider that some specific objects will appear in related videos, e.g., a knife often appears in the video of `cutting` and `slicing`, a mower often appears in the video of `mowing`, and a computer often appears in the video of `typing`. According to the above inference, we can take these specific objects as the attributes of related videos. In order to obtain the attributes of videos, we use ResNet101 [5] pre-trained on two publicly available multi-label datasets, NUS-WIDE [8] (81 concept labels) and MS-COCO [6] (80 object labels) to extract the feature. We extract features at one frame per second because we believe that the composition of objects will not change dramatically on a framewise basis.

Method We concatenate the extracted feature of three frames as $X \in R^{3\times2048}$. Given the input X,

$$y = f(X, \theta), y \in R^{200} \tag{4}$$

where $y = [y^1, y^2, ..., y^{200}]^T$ are the predicted label confidences computed by two fully-connected layers.

Table 6. The accuracy of different feature extracted from ResNet101 pre-trained on NUS-WIDE dataset and COCO dataset.

Dataset	Top-1 accuracy	Top-5 accuracy
NUS-WIDE[8]	10.02%	27.15%
MS-COCO[6]	10.14%	27.57%

Table 7. Five classes perform the best.

Best classes	NUS-WIDE[8]	MS-COCO[6]
Top-1	Grilling: 60%	Grilling: 56%
Top-2	Mowing: 54%	Clinging: 54%
Top-3	Typing: 52%	Howling: 54%
Top-4	Welding: 52%	Hiking: 48%
Top-5	Clinging: 46%	Boiling: 46%

Performance According to the result show in Table 7, we can find out that the more similar composition of objects is, the higher accuracy we will get.

2.5. Attribute consistency loss

Method *Attribute consistency loss* (ACL), introduced by [3], focuses on the domain adaptation under the setting of fine-grained recognition. ACL hopes the deep model will be more generalized to examples from the real world instead of overfitting on a given dataset.

In order to do so, ACL uses the concept of multi-task learning: predict classes and attributes at the same time by sharing the last features extracted from the deep model. Here attributes can be any properties we detected from examples. In our case, we use the scores of a model dealing with COCO object detection tasks (80 objects in total). In other words, our attributes represents the probability of occurrence of each object in a video. Besides predicting classes and attributes, the other part in ACL is to reduce the distribution distance (measured by symmetric KL divergence) between predicted attributes and mapped attribute, where mapped attribute is mapped from predicted classes.

In our case, we calculate all the objects' scores for each video in the training set. The results are then grouped by action class to aggregate the mean. Consequently, we have the function to map from action class to 80 object occurrence scores.

Table 8. Five highest and lowest intra-class object variation.

Class	Highest	Class	Lowest
Feeding	1.91618	Erupting	1.28444
Spreading	1.91230	Protesting	1.33439
Scratching	1.87774	Waxing	1.34714
Chewing	1.87588	Tattooing	1.36594
Biting	1.87104	Mowing	1.40809

Performance First, we tried a simple model (LSTM over DenseNet) to evaluate the score with/without ACL on the Moments dataset. The one with ACL took longer time to converge but got close accuracy compared to the one without ACL. We also computed the intra-class object variation, and the results are found in Table 8. However, from Table 9, we find that the F1 score of a class with lower intra-class attributes variation will be higher (negative correlation), showing that if the videos in a class have relatively consistent object occurrence, its easier for a model to perform prediction. Moreover, the model with ACL has lower correlation than the one without ACL. After we apply ACL on TSN, the performance drops a bit. Due to the lack of time, we abandon it and did not do deeper examination. If more attributes extractors from different views are applied, it might be beneficial for future fusion.

Table 9. Correlation between F1 score and intra-class object variation

Correlation	Top-1 acc.	Top-5 acc.
Base model	-0.6019	-0.5462
With ACL	-0.5883	-0.5933

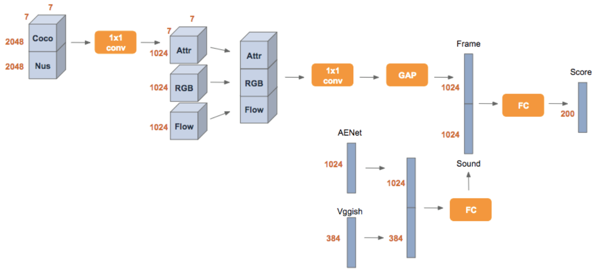

Figure 3. The structure of early fusion method. We fuse the feature maps of the modalities mentioned above at different stages, and then predict the final results.

3. Fusion

We evaluate two fusion schemes, Early Fusion and Late Fusion.

3.1. Early fusion

The early fusion structure is depicted in Figure 3. Let the feature map extracted from ResNet101 pre-trained on COCO dataset be denoted as $C \in R^{7\times7\times2048}$, the feature map extracted from ResNet101 pre-trained on NUS-WIDE dataset as $N \in R^{7\times7\times2048}$, the rgb feature map extracted from TSN as $R \in R^{7\times7\times1024}$, the optical flow feature map extracted from TSN as $F \in R^{7\times7\times1024}$, the feature extracted from AENet as $E \in R^{1024}$ and the feature map extracted from VGGish as $V \in R^{3\times128}$.

For the frame part, first, we concatenate C and N, then go through a 1×1 convolution layer to fuse these two modalities and denote the fusion feature map of attribute as $A \in R^{7\times7\times1024}$. Second, we concatenate A, R and F then go through a 1×1 convolution layer to fuse these three modalities and denote the fusion feature map of frame as $M \in R^{7\times7\times1024}$. Third, let M go through a global average pooling layer and get $M \in R^{1024}$.

For the sound part, first, we do zero padding for those videos without sound. Second, we concatenate E and V, then go through a fully-connected layer to fuse these two modalities and denote the fusion feature map of sound as $S \in R^{1024}$.

Last, we concatenate M and S as our final feature $\in R^{2048}$ and go through a fully-connected layer to get the prediction.

Table 10. The accuracy of early fusion.

Method	Top-1 accuracy	Top-5 accuracy
Early fusion	22.19%	45.45%

Table 11. The accuracy of early fusion compares to the accuracy of late fusion.

	Increase	Decrease
Top-1	Ascending: +37%	Sailing: -72%
Top-2	Bending: +24%	Protesting: -57%
Top-3	Playing music: +18%	Surfing: -54%
Top-4	Biting: +15%	Hiking: -46%
Top-5	Baking: +14%	Diving: -42%

Performance According to the result shown in Table 11, we can find out that the method of late fusion is better than the method of early fusion on the video classification problem.

3.2. Late fusion

We take the (pre-softmax) prediction scores of every modalities and do the simple and (scalar) weighted average. The results are shown in Table 12.

Table 12. Late fusion of 7 modalities on the MIT Mini validation set.

Method	Top-1 accuracy	Top-5 accuracy
Average fusion	37.09	65.29
Weighted fusion	44.21	72.96

3.3. Ablative study of Late Fusion

In order to identify which modalities provides the largest impact, we perform an ablative study. Given the classifier scores for the seven modalities, we run two late fusion methods (summation and scalar weighting) and report the results in Tables 13 and 14. Note that we tried other parameterized fusion methods, but do not report results here, as those severely overfitted.

Table 13. Ablative study for (late) sum fusion of 7 modalities on the MIT Mini validation set.

Configuration	Top-1 accuracy (%)	Top-5 accuracy (%)
Full	37.09	65.29
w/o TSN (RGB)	31.45	58.11
w/o Flow	34.2	62.01
w/o Aenet	36.96	65.21
w/o Attribute	37.05	65.3
w/o VGGish	36.58	64.82
w/o RPAN	44.84	73.83
w/o TRN (RGB)	31.92	60.12

Table 14. Ablative study for (late) weighted fusion of 7 modalities on the MIT Mini validation set.

Configuration	Top-1 accuracy (%)	Top-5 accuracy (%)
Full	44.21	72.96
w/o TSN (RGB)	37.62	65.92
w/o Flow	39.81	69.22
w/o AENet	43.76	72.77
w/o Attribute	44.22	73.05
w/o RPAN	44.24	73.34
w/o TRN (RGB)	37.47	66.95

Clearly, RGB features remain the most important modality. Pose features did not perform well in the 7-modality fusion, however, it should be noted that RPAN did add a 6% improvement when only the first six modalities were considered, i.e., TRN was left out.

4. Analysis

We train a baseline model consisting of ResNet-50 with an added LSTM layer, and share the observations of our analysis. We begin by studying the confusion matrix and distinguishing different types of confusion:

Semantic similarity is an issue where classes have similar meaning. An example is `slicing`, which is misrecognized as `chopping` in 28% of validation set cases.

Visual similarity Certain actions cannot be discriminated by visual features alone, but require other modalities. An example for this case is `howling` being falsely classified as `barking` by the RGB baseline in 24% of examples.

Subset of class Numerous actions form a subset of or intersect with another action class, which necessitates multi-label classification. The classes `pedaling` and `bicycling` exemplify this, where the latter is misclassified as the former in 16% of cases.

Time reversed classes show similar visual content, but are reversed from each other. One classic instance here is closing, which is misclassified in 20% of cases as opening.

4.1. F1-score ranking

In the following, we rank classes by their (baseline) F1-score and note our observations. While we cannot list all classes, we list a selection actions in Table 15. Note that for space reasons the table does not show all best- or worst-performing actions.

Table 15. F1-score for selected actions in baseline ResNet-50 + LSTM model.

Class name	**F1-Score**
Erupting	0.612
Rafting	0.549
Bulldozing	0.454
...	...
Spreading	0.040
Catching	0.026
Opening	0.020
Pulling	0.000

We observe that performance correlates with intra-class variation. Classes such as erupting are typically subject to smoke, lava, etc. and therefore easy to recognize. This is unlike the actions with low F1-score in Table 15: Actions like pulling are more abstract and can be associated with one of a diverse set of objects; these actions hence have a large intra-class variability.

We propose that more fine-grained features are necessary to improve the failure cases with high intra-class variance. In particular, instead of relying on background information, fine-grained information about pose needs to be retrieved and processed.

5. Conclusions

In this paper, we evaluated many modalities in videos on the Moments in Time dataset, which has a significant intra-class variation among the categories. This work discussed the essential elements of videos from different aspects, and demonstrated experiments on different modalities. Our experiments also indicate that late fusion with many modalities performs better than early fusion.

References

[1] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[2] W. Du, Y. Wang, and Y. Qiao. Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3745–3754, Oct 2017.

[3] T. Gebru, J. Hoffman, and L. Fei-Fei. Fine-grained recognition in the wild: A multi-task domain adaptation approach. *CoRR*, abs/1709.02476, 2017.

[4] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 776–780. IEEE, 2017.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016.

[6] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. Microsoft coco: Common objects in context. *CoRR*, abs/1405.0312, 2014.

[7] M. Monfort, B. Zhou, S. A. Bargal, A. Andonian, T. Yan, K. Ramakrishnan, L. M. Brown, Q. Fan, D. Gutfreund, C. Vondrick, and A. Oliva. Moments in time dataset: one million videos for event understanding. *CoRR*, abs/1801.03150, 2018.

[8] T. seng Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. Nus-wide: A real-world web image database from national university of singapore. In *In CIVR*, 2009.

[9] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

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

[11] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Val Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[12] B. Zhou, A. Andonian, and A. Torralba. Temporal relational reasoning in videos. *CoRR*, abs/1711.08496, 2017.