Action Recognition From Video

Krutik Parmar
201801199

Supervisor : Prof. Manish Khare

Information and Communication Technology
Dhirubhai Ambani Institute of Information and Communication Technology

***Abstract*— Video action recognition is one of the representative tasks for video understanding. It has many real-world applications, including behaviour analysis, video retrieval, gaming, and entertainment. Thanks to the rise of deep learning, we've seen significant progress in video action recognition over the last decade. 2D CNNs have shown state-of-the-art performance in the image related tasks. But in video understanding, temporal information and motion plays an important role and 2D CNNs alone cannot capture it. 3D CNN based methods can achieve good performance but are computationally expensive. In this project, we have used 2D CNN InceptionResnet-v2 architecture as a backbone model and we have applied Temporal Shift Module (TSM) to capture the temporal relationships between frames ad MotionSqueeze Module to capture the motion related information. We have trained our model on HMDB51 dataset.**

***Keywords*— Video Understanding, Action recognition, Efficient video processing, Motion feature learning, Video processing**

I. INTRODUCTION

One of the most important tasks in video understanding is to understand human actions. It has many real-world applications, including behavior analysis, video retrieval, human-robot interaction, gaming, and entertainment. Human action understanding involves recognizing, localizing, and predicting human behaviors. The task to recognize human actions in a video is called video Action Recognition(AR).

A video clip contains two critical pieces of information for AR: Spatial and Temporal information. Spatial information represents the static information in the scene, such as objects, context, entities, etc., which are visible in a single frame of the video, whereas temporal information, obtained by integrating the spatial information over frames, mostly captures the dynamic nature of the action. In temporal information of video, the most important feature is motion present in video. In order to grasp a full understanding of a video, we need to analyze its motion patterns as well as the appearance of objects and scenes in the video.

In this work, we have used MotionSqueeze module for effective feature extraction and Temporal Shift Module (TSM) to model the temporal information. We have used 2D CNN architecture Inception-Resnet-v2 as a backbone model.

II. MOTIONSQUEEZE(MS) MODULE

The most distinctive feature of videos, from those of images, is motion. In order to grasp a full understanding of a video, we need to analyze its motion patterns as well as the appearance of objects and scenes in the video. We have used MotionSqueeze(MS), which is a neural network module. It can be inserted in the middle of any neural network for effective motion understanding.

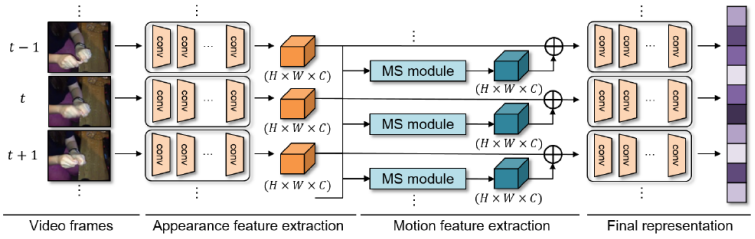


Fig. 1 : MS module which generates motion features using frame-wise appearance features and combines the motion features into the next downstream layer.

The model takes a video of T frames as input and predicts the category of the video as output, where convolutional layers are used to transform input frames into frame-wise appearance features. The proposed motion feature module, dubbed MotionSqueeze (MS) module, is inserted to produce frame-wise motion features using pairs of adjacent appearance features. The resultant motion features are added to the appearance features for final prediction.

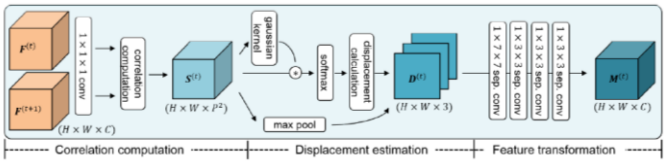


Fig 2: Overall process of MotionSqueeze (MS) module. The MS module estimates motion across two frame-wise feature maps (F^t, F^{t+1}) of adjacent frames. A correlation tensor S^t is obtained by computing correlations, and then a displacement tensor D^t is estimated using the tensor. Through the transformation process of convolution layers, the final motion feature M^t is obtained.

Correlation computation:

Suppose the two adjacent input feature maps by F^t and F^{t+1}, each of which is 3D tensors of size $H \times W \times C$. The spatial resolution is $H \times W$ and the C dimensional features on spatial position x by F_x. A correlation score of position x with respect to displacement p is defined as

$$s(x, p, t) = F_x^t \cdot F_{x+p}^{t+1}$$

where $\cdot$ denotes dot product. For efficiency, the correlation score of position x is computed only in its neighbourhood of size $P = 2k + 1$ by restricting a maximum displacement: $p \in [-k, k]^2$. For t_{th} frame, a resultant correlation tensor S^t is of size $H \times W \times P^2$. The cost of computing the correlation tensor is equivalent to that of 1×1 convolutions with P^2 kernels; the correlation computation can be implemented as 2D convolutions on t_{th} feature map using $t + 1_{th}$ feature map as P^2 kernels. The total FLOPs in a single video amounts to $THWCP^2$. A convolution layer is applied before computing correlations, which learns to weight informative feature channels for learning visual correspondences. In practice, the neighbourhood is set $P = 15$ given the spatial resolution 28×28 and apply an $1 \times 1 \times 1$ layer with $C = 2$ channels.

Displacement estimation:

From the correlation tensor S^t, a displacement field is estimated for motion information. A straightforward but non-differentiable method would be to take the best matching displacement for position x by $argmax_p s(x, p, t)$. To make the operation differentiable, a weighted average of displacements using $softmax$, called $soft - argmax$ is used, which is defined as

$$d(x, t) = \sum_p \frac{\exp\left(s(x, p, t)\right)}{\sum_{p'} \exp\left(s(x, p', t)\right)} p$$

This method, however, is sensitive to noisy outliers in the correlation tensor since it is influenced by all correlation values. Thus, the kernel-soft-argmax is used that suppresses such outliers by masking a 2D Gaussian kernel on the correlation values; the kernel is centred on each target position so that the estimation is more influenced by closer neighbours. Kernel-soft-argmax for displacement estimation is defined as

$$d(x, t) = \sum_p \frac{\exp\left(g(x, p, t)s(x, p, t)/\tau\right)}{\sum_{p'} \exp\left(g(x, p', t)s(x, p', t)/\tau\right)} p$$

Where

$$g(x, p, t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{p - argmax_p s(x, p, t)}{\sigma^2}\right)$$

Here, $g(x, p, t)$ is the Gaussian kernel. τ is a temperature factor adjusting the softmax distribution; as τ decreases, softmax approaches argmax. We set $\tau = 0.01$ in our experiments.

In addition to the estimated displacement map, a confidence map of correlation is used as auxiliary motion information, which is obtained by pooling the highest correlation on each position x:

$$s^*(x, t) = \max_p s(x, p, t).$$

The confidence map may be useful for identifying displacement outliers and learning informative motion features. The (2-channel) displacement map and the (1-channel) confidence map is concatenated into a displacement tensor D^t of size $H \times W \times 3$ for the next step of motion feature transformation.

Feature transformation:

The displacement tensor D^t to an effective motion feature M^t that is readily incorporated into downstream layers. The tensor D^t is fed to four depth-wise separable convolution layers, one $1 \times 7 \times 7$ layer followed by three $1 \times 3 \times 3$ layers, and transformed into a motion feature M^t with the same number of channels C as that of the original input F^t. The depth-wise separable convolution approximates 2D convolution with a significantly less computational cost. As illustrated in Figure 2, the MS module

generates motion feature M^t using two adjacent appearance features F^t and F^{t+1} and then add it to the input of the next layer. Given T frames, we simply pad the final motion feature M^T with M^{T-1} by setting $M^T = M^{T-1}$.

III. Temporal Shift Module (TSM)

Given a video V, first T frames Fi, F1, ..., FT are sampled from the video. After frame sampling, they are passed into 2D CNN baselines in which each of the frames individually are processed, and the output logits are averaged to give the final prediction. TSM can be inserted in middle of any neural network, it enables temporal information fusion at no computation. TSM model has exactly the same parameters and computation cost as 2D model. During the inference of convolution layers, the frames are still running independently just like the 2D CNNs.

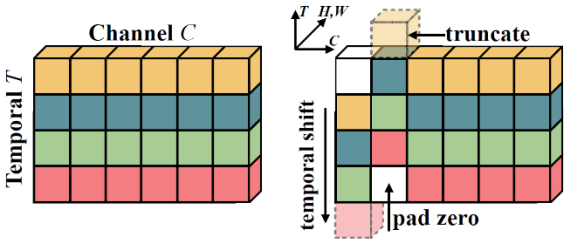
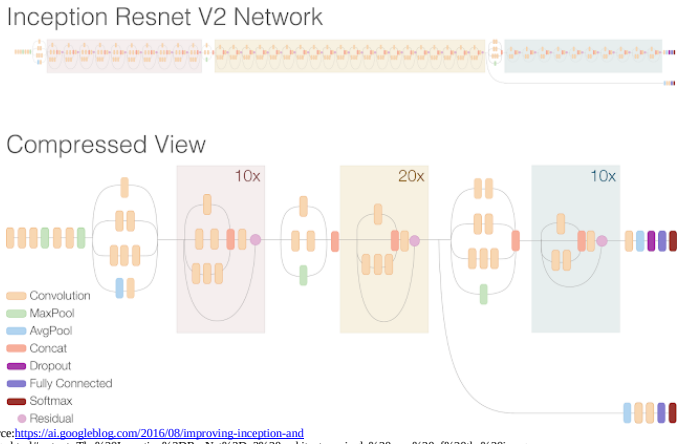


Fig 3

Temporal Shift module is shown in Figure 3. In Figure, a tensor with C channels and T frames is described. The features at different time stamps are denoted as different colours in each row. In TSM block, along the temporal dimension, one part of channels are shifted −1, another part by +1, the rest are left un-shifted. Then this shifted version of channels is passed into the next convolution layer.

IV. InceptionResnet-v2

Inception-ResNet-v2 is a convolutional neural network that is trained on more than a million images from the ImageNet database. The network is 164 layers deep and can classify images into 1000 object categories, such as the keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images.



Source:https://ai.googleblog.com/2016/08/improving-inception-and image.html#:~:text=The%20Inception%2DResNet%2Dv2%20architecture,single%20crop%20of%20the%20image.

It is formulated based on a combination of the Inception structure and the Residual connection. In the Inception-Resnet block, multiple sized convolutional filters are combined with residual connections. The usage of residual connections not only avoids the degradation problem caused by deep structures but also reduces the training time.

V. Experiment

In this section, we first introduce the dataset and data pre-processing and data augmentation techniques that we have applied. Then, we explore the final architecture and training and testing of our model. Finally, we compare the performance of our method with the state of the art.

A. Dataset

We have conducted our experiment on a benchmark dataset, namely HMDB51. The HMDB51 dataset is a large collection of realistic videos from various sources, such as movies and web videos. The dataset is composed of 6766 video clips from 51 action categories.

B. Data Pre-processing

First, we have extracted frames from the video. In both training and testing, we use a clip of frames that are sampled from the video instead of an entire video. We have used dense frame sampling method to sample the frames from the video. For each video, we sample a clip of 8 or 16 frames, resize them into 240x320 images, and crop 224x224 images from the resized images.

C. Architecture

Figure 5 shows the architecture of out proposed model. In the figure, the repetitive parts of the model are highlighted. Here, as we can observe that we have inserted TSM modules in the starting of each repetitive section. We have inserted the MS module after the second repetitive section.

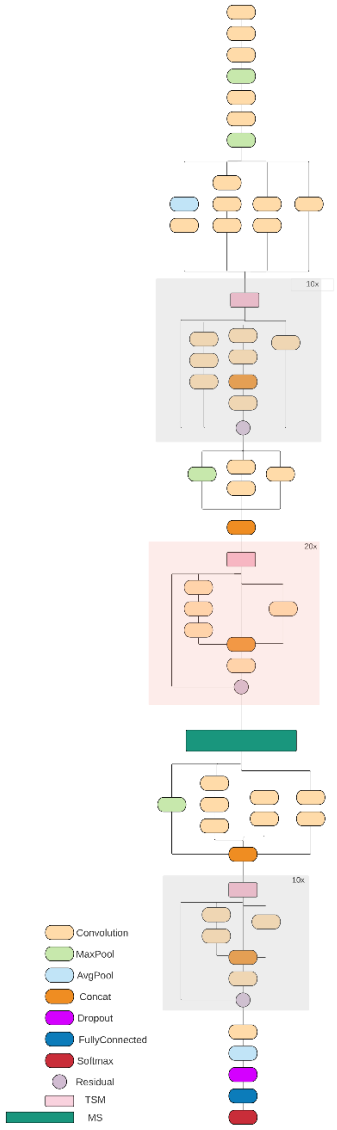


Fig 5: Proposed model's architecture

D. Training Testing

We applied Transfer learning method to train our model. We have used InceptionResnetv2 model which was pre-trained on Imagenet dataset. The training parameters are: 32 training epochs, learning rate 0.02 with weight decay of 0.001, batch-size 8, dropout 0.5. We have used SGD with Nestrov momentum for optimization. We have trained our model using Google Colab's GPU.

E. Results

Table 1 summarizes the results on HMDB51. Our proposed method which uses TSM, MS modules and InceptionResnet-v2 as a backbone achieves 61.52% accuracy.

Table 1: Performance comparison on HMDB51.

Model	**Backbone**	**Pre-train**	**HMDB51**
Two-stream	CNN-M	I	59.4
TDD	CNN-M	I	63.2
Fusion	VGG16	I	65.4
TSN	BN-Inception	I	68.5
TVNet	BN-Inception	I	71.1
C3D	VGG16-like	S	56.8
I3D	BN-Inception-like	I, K	74.8
ResNet3D	ResNeXt101-like	K	70.2
R2+1D	ResNet34-like	K	74.5
S3D	BN-Inception-like	I, K	75.9
CIDC	ResNet50-like	-	75.2
Hidden TSN	BN-Inception	I	66.8
OFF	BN-Inception	I	74.2
TSM	ResNet50	I	73.5
STM	ResNet50-like	I, K	72.2
TEINet	ResNet50-like	I, K	72.1
TEA	ResNet50-like	I, K	73.3
MSNet	**ResNet50-like**	**I, K**	**77.4**
Ours	*InceptionResnet-v2*	*I*	*61.52*

Here, I : Imagenet, S : Sport1M, K : Kinetics datasets

ACKNOWLEDGMENT

I wish to record a deep sense of gratitude to Prof. Manish Khare, my supervisor for his invaluable guidance and constant support at all stages of my study and related research on this project.

REFERENCES

[1] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. In ECCV, 2020.

[2] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal Shift Module for Efficient Video Understanding. In the IEEE International Conference on Computer Vision (ICCV), 2019.

[3] Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. No. 1. 2017.

[4] Zhu, Yi, et al. "A Comprehensive Study of Deep Video Action Recognition." *arXiv preprint arXiv:2012.06567* (2020).

[5] Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)