Multi Channel Action Recognition

Gurvinder Singh, Michael Geraci[1,✉]

[1]University at Buffalo,NY

In recent years, many works in the video action recognition literature have shown the benefit of using two stream models (combining spatial and temporal input streams) for achieving good performance. In this paper we are combining multiple streams such as optical flow, RGB flow and human pose combined. Additionally we use frame selection and captioning for each frame. Specifically we are using captions generated from the pretrained models. As all the action recognition UCF101 videos involve humans these additional streams provide simple and flexible addition. We propose this multi-channel action recognition model, which we dub MCAR-Net (short for Multi Channel Action Recognition Net). MCAR-Net combines these new streams with the standard RGB and flow based input streams via distillation techniques to identify actions with better accuracy.

Introduction

Recognizing actions based on video input is a difficult task that has been attempted by many in the computer vision community. Video input contains both temporal and spatial information and it is important to use a model that takes both of those into account. Action recognition from videos can be a difficult task and complex machine learning models are often needed to achieve a satisfactory result.

In this work we have implemented the single Long Recurrent Convolution Network for doing the action recognition for capturing spatiotemporal information rather than using two for spatial and temporal. we have collected the spatial information using the video frames and used the pretrained Resnet-152 model for creating the embeddings. Here we proposed Frame selection approach to reduce the number of frames significantly for the final prediction. As we human don't require multiple frames to recognizing the action it is possible to recognize action from few important frames. The Quality of frames is used to do the frame selection. The temporal information is captured using the Long term Recurrent network over sequence of video. Our model takes both of these spatial as well as temporal information and do the frame level prediction which then averaged to form the final prediction. The datasets we used were UCF101 with the Resnet-152 pretrained model for captioning and frame embedding and Classication Model with Cross entropy loss to do the prediction.

Related Work

In Previous work in action recognition two types of network approaches were dominant where One network for capturing spatiotemporal information vs. two separate ones for each spatial and temporal information. The major difference among them was the design choice around combining spatiotemporal information.Here single stream network, with Single Frame uses single architecture that fuses information from all frames at the last stage and Late fusion which uses two nets with shared parameters, spaced 15 frames apart, which combines predictions at the end.

Two Stream Networks

In the Two Stream there are two separate networks one for spatial context (pre-trained), one for motion context. The input to the spatial network is a single frame of the video. Frames of a video contain all of the visual information needed to extract visual features from the video. The spatial network is meant to capture visual information from the frames. Input to the temporal net is a bidirectional optical flow stacked across for successive frames. This optical flow is used to capture motion from the video. Optical flow is meant to represent movement between successive frames. This captures the motion part of the video. Similarly to how the human brain processes vision, Two Stream networks take into account motion and object recognition separately. These two streams are trained separately. In the end they are combined using late fusion. The late fusion is performed by using a multi-class linear SVM on stacked normalized softmax scores.

LRCN

LRCN is an architecture built on the idea of using LSTM blocks after convolution blocks and using end-to-end training of the entire architecture. During training, 16 frame clips are sampled from video. The architecture is trained end-to-end with input as RGB or optical flow of 16 frame clips. Final prediction for each clip is the average of predictions across each time step. The final prediction at video level is average of predictions from each clip. Here False label assignment as video was broken to clip is the main drawback. More frames as input instead of 16 frame to train the network is one of approach to increase the accuracy.

SMART

SMART use the idea of Frame selection to reduce the noise in the prediction and improve the accuracy of the model. It also propose the ideas of language features associated with the content of the frame. like for kayaking action class associated words like water,boat, paddle can help in the discrimination. It uses Attention and Relational Models, where the use of attention to weigh spatial regions representative of a

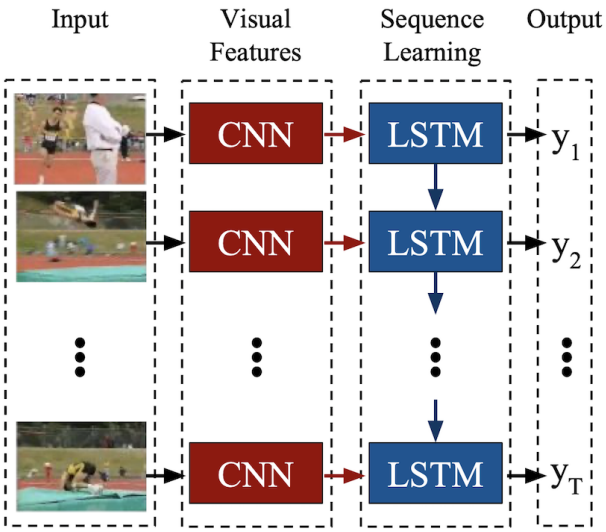

Fig. 1. The Long-term Recurrent Convolutional Networks (LRCNs), architectures leverage the strength of rapid progress in CNNs for visual recognition problems, and the growing desire to apply such models to time-varying inputs and outputs. LRCN processes the (possibly) variable-length visual input (left) with a CNN whose outputs are fed into a stack of recurrent sequence models, which finally produce the prediction.

particular task is done by generating spatial attention masks implicitly by training the network with video labels. Other related technique are Conv3D. Conv3D is an architecture that uses 3D convolutions on video volume. This has been shown to get a high accuracy on action recognition tasks. Conv3D has been successfully combined with attention in order to achieve good results. Factorized 3D convolution is another related technique. It breaks down 3D convolutions into spatial 2D convolutions followed by temporal 1D convolutions.

Conv3D

Another related technique is Conv3D. Conv3D is an architecture that uses 3D convolutions on video volume. This has been shown to get a high accuracy on action recognition tasks. Conv3D has been successfully combined with attention in order to achieve good results. Factorized 3D convolution is another related technique. It breaks down 3D convolutions into spatial 2D convolutions followed by temporal 1D convolutions.

Baseline Model

In this work we used a pretrained resnet152 imagenet model for feature extraction. We did the feature extraction on a sequence of images which are fed into multiple batches. These extracted features are then input to the LSTM for the sequence learning purpose. This LSTM learns the time variant information present in the videos. The fully connected model with cross entropy loss is used to do the prediction over single frames. Finally, these predictions over a bunch of sequences will average to predict the single label for the whole video. Our Setup involved the single machine with two tesla GPUs with 16Gb memory each. The parameters used for the experiment are as follows:

- Videos are 25 frames per second and 5 sec video means there are 125 frames for recognizing the Action. A frame selection model calculated the quality of the frame and scored the frames from lowest to highest quality. Opencv implementation of the BRISQUE is used to score each frame based upon its quality.
- Frame selection using the Frame quality score is being used to trim down the number of frames.As not all the frames are required to predict the Action. less number of frames means less noise.
- The dataset is first processed in sequences where we used the single random sequence for each video. Sequence length which we found to fit into the tesla V100 Gpu of 16 Gb RAM is 50 Frames with 3 channel 224*224 image size.
- A batch size of 16 and a sequence length of 50 frames can fit into the single Gpu memory. So we use a batch size of 32 for the distributed computing over two Gpu's. We processed 1600 frames at once in our model.
- A pretrained resnet152 model is used to embed the Cnn input into embeddings. This outputs a 1024 size feature vector that is then processed through the single bidirectional LSTM module. A latent feature vector of size 512 was output.
- A two layer fully connected sequential module was used to convert this 512 latent feature vector into 101 categories.
- We used the Cross entropy loss with Adam Optimizer for our experiment.
- At Test time we performed predictions over all the frames of the video instead of single sequences of frames. This is different from training when we performed predictions over single sequences of frames. The selection of the sequence of frames started from the very first frame.

Experiment

A. Masked Rgb Flow. The Rgb flow is calculated from the motion in the video. This optical flow is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of the object or camera. This works on the assumption that the pixel intensities for the object do not change between consecutive frames and that all the neighbouring pixels have a similar type of motion. The computation of optical flow is done in two various ways. One is using the sparse features set where detected corners are used to compute optical flow; the Lucas Kanade algorithm is based upon this approach. Another is dense optical flow which calculates the direction and magnitude of the optical flow vectors for all the pixels. One algorithm using this approach is Gunner Farneback's algorithm which is explained in "Two

Fig. 2. On the Left Hand side in the Rgb image which is the single frame from the video. In the middle is the Dense Optical flow which were calculated using the Gunner Farneback's algorithm. This Flow further converted to mask to create the Rgb masked image where focus is present on the Action instead of Background.

Frame Motion Estimation Based on Polynomial Expansion" by Gunner Farneback in 2003. This dense optical flow is used for creating the mask for the RGB image frames and the final masked image hides the portion of the scene where movement is not happening (Fig. 2).

B. Supervised Contrastive Learning. Supervised Contrastive Learning (Prannay Khosla et al.) is a training methodology that outperforms supervised training with crossentropy on classification tasks.Here the training is performed in two phases.

- First we train an encoder to learn to produce feature vectors such that the feature vectors from the same class will be similar as compared to feature vector or representation from the different class.
- Then we freeze these encoder and put a classification head on top of the freezed encoder to do the classification.

we used the NTXentLoss which is a generalization of the NPairsLoss which have been used for the Self supervision Tasks.

$$L_q = -log\frac{e^{q.k+/T}}{\sum_{i=0}^{K} e^{q.k_i/T}}$$

Here q is the normalized feature vector and the k are the labels for the feature vectors. Here T tau in the above equation is the hyperparameter tau which is usually set to 0.05 as default. We used this NTXentLoss to train the encoder which tried to encode the same labelled feature vector to a similar feature vector. The output from this encoder was 512 sized feature vector which used the resnet152 and LSTM combined with the above contrastive loss. We trained this encoder over our training data for 50 epochs. The resultant embedding for the first 20 categories can be seen in (Fig 3). For the Second phase we used a 2 layer sequential model as the head of the freezed encoder module to do the prediction. This phase of training is done using the CrossEntropy loss with the same Adam optimizer.

Results

We performed the evaluation using the official Train-test split for UCF101 dataset. Here 8k videos across 101 categories were used in the train set and 4k videos across different categories were in the test set. We used the 50 percent sample of this train-test split for our training and evaluation. Table

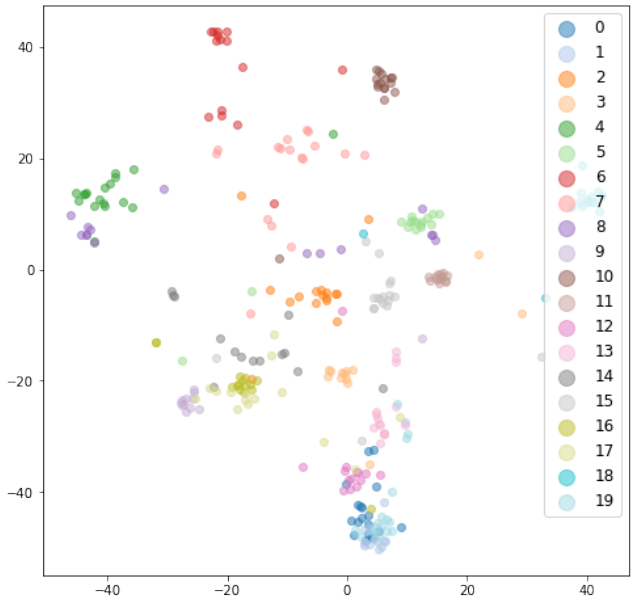

Fig. 3. Tsne is used to project the 512 latent feature vector into two dimensions.Here this 512 feature vector is the result of the Encoder Stage.

Fig. 4. On the left, the Correct predictions done by the model and on the right Incorrect predictions done by the model.

1. shows the evaluation results for Top1 where the result of single sequences of frame prediction accuracy is calculated. Top3 shows the accuracy of finding the label in the top three predicted categories.

Model	Top1	Top3
Ours	76.3	88.2
Original	77.28	-

Table 1. Accuracy for implemented Lrcn Model

Fig4. shows the resultant prediction over different frames. For the supervised contrastive learning experiment we were not able to learn the correct latent embedding vectors from the encoder for the classification task. As the loss is decreasing over time for the encoder training, the clusters obtained from the feature vectors are not totally separable as you can see in Fig 3. Further analysis over the loss function input and longer training for the encoder can be done in the future.

Conclusions

We done the extensive survey of the literature present for the Action Recognition and We have implemented the Long term Recurrent Convolution network for the Video Action Recognition and done further experiments for the improvement of the current Baseline model. Our Experiment included using Brisque for frame quality and check, Masked Rgb Flow for removing background noise Supervised Contrastive learning

for learning the better representations.

ACKNOWLEDGEMENTS
We Also Want to thanks the TA for there continuous Support during the project. Further, Support provided by the Center for Computational Research at the University at Buffalo.

Bibliography

(1) (2) (3) (4) (5) (6) (7)

1. Yinxiao Li, Zhichao Lu, Xuehan Xiong, and Jonathan Huang. Perf-net: Pose empowered rgb-flow net, 2020.
2. Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos, 2014.
3. Shreyank N Gowda, Marcus Rohrbach, and Laura Sevilla-Lara. Smart frame selection for action recognition, 2020.
4. Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset, 2018.
5. Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description, 2016.
6. Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.
7. Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.