CS230

Human Activity Recognition with Computer Vision

Jeong-O Jeong
Department of Computer Science
Stanford University
djeongo@stanford.edu

Abstract

Increasing number of videos are becoming available in today's world. However, it is impossible for a person to manually analyze all of the videos and extract useful information out of them. In this project, we implement a deep learning-based computer vision algorithm for human activity recognition. We use transfer learning to adapt the SlowFast network pre-trained on human activity recognition tasks to the VIRAT video dataset. We also combine the last few layers of the YOLO model with the SlowFast model in order to extract the bounding boxes of the human activity.

1 Introduction

The problem of human activity recognition through computer vision has many important applications such as elderly assistance, patient monitoring, surveillance, human-computer interaction, and information retrieval. Although the problem may appear similar to analyzing static images, it adds more complexity to the problem because there is also a temporal dimension in addition to spatial dimension in the data. This also makes the problem more computationally demanding.

The SlowFast network is an example of a deep-learning method where temporal and spatial features are explicitly computed separately but also fused together throughout the network while keeping the computational costs down [Fei+19]. We adopt an open-source implementation of the SlowFast network and apply transfer learning to the VIRAT dataset [Moo+15].

The SlowFast network is trained on the AVA dataset which consists of close-up view of the human performing an action. Its main objective is to classify the activity itself, but not the spatial localization of the activity. In contrast to AVA, the VIRAT dataset is a surveillance video consisting of long-shot views where the main human subject appears relatively small in the image as shown in Figure 1. Therefore the spatial localization becomes important in order to correctly detect where the human activity is happening in the image.

The SlowFast network uses ground-truth bounding boxes as inputs to its RoIAlign layer to classify the human activity. One possible method of spatial localization is to add a region proposal network to the model similar to the Faster-RCNN model. However, using RPN adds complexity since it requires training two separate networks. In contrast, the YOLO model outputs both the bounding boxes and class probabilities without having to train two separate networks [Red+16]. In order to get bounding boxes along with class probabilities, we augment the SlowFast network with the last few layers of YOLO model and incorporate its loss function. This allows training the whole network end-to-end and get the bounding boxes as the output in addition to the class probabilities.

CS230: Deep Learning, Autumn 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Relevant Work

Similar to most computer vision problems, human activity recognition problems are mostly handled with architectures involving convolutional neural networks. However, there are some differences among the papers in how to handle the temporal aspect of the problem. One of the earlier papers in human action recognition uses two parallel streams of spatial information and temporal information to make the human activity classification [SZ14]. The two separate streams use parallel convolutional networks to make independent predictions, which are then combined to give the final prediction. There have also been 3D convolutional networks in order to better incorporate the temporal information [Dib+17]. There are also LSTM-based papers [Ma+17; Wan+16] which attempt to better model longer-term structures. Recently, a model that combines BERT with 3D convolutional networks has been published as well [KKA20]. Other relevant work include the use of pose estimation in action recognition [LPT18].

2.1 SlowFast Network

We adopt the SlowFast network as the baseline model because it achieves the state of the art performance on the AVA v2.1 Benchmark with the mAP of 28.2. The SlowFast network is a two-pathway CNN model with one pathway processing a stream of lower frame rate while the other pathway at a higher frame rate [Fei+19]. The slower frame rate captures semantic information while the higher frame rate captures the changes in the motion. In order to keep the computation down, the higher frame rate use a fewer number of channels in the convolutional layers. Unlike previous models, the SlowFast network does not require extracting hand engineered features such as optical flow. It is also loosely based on how human visual systems work where Magnocellular cells operate at high temporal frequency and respond to fast temporal changes, while Parvocellular cells are sensitive to spatial details and color at lower temporal resolution.

The backbone for the SlowFast model is the residual network blocks. It uses either the ResNet 50 or ResNet 101 architecture depending on the task. Unlike the earlier models where fusion of temporal and spatial features happen at the end of the model [SZ14], fusion of the two path ways is performed throughout the network after each ResNet stage. Fusion is performed by passing the activations of the fast path through Conv3D, BatchNorm3D, and Relu blocks and concatenating with the activations of the slow path way. The fused output is then propagated to the next ResNet stage of the slow path way.

The final layers of the model consist of the ROI Align blocks followed by a fully-connected layer and a softmax layer to classify the activity.

3 Dataset

The dataset used is the VIRAT dataset Release 2.0 . It is a video surveillance dataset consisting of various aerial and ground camera view points. The ground dataset contains 11 different scenes captured at 1080p or 720p at 24fps or 30fps. Each scene contains multiple video clips. We utilize only the ground dataset. The list of classes of actions identified in the dataset is listed in Figure 1. The dataset provides bounding boxes for the events. The 52% of classes in the dataset consists of "Person carrying an object", while the next dominant class is "Person entering a facility" at 10%. This leads to a class imbalance problem. Addtionally, in contrast to the AVA dataset where the videos are close-up views of the activity being performed, the subject of human activity appears relatively small compared to the overall size of the image.

4 Method

The open-source implementation of the SlowFast network is extended to support the new VIRAT dataset. We apply transfer learning to the new dataset by using the pre-trained weights on the AVA 2.1 dataset. In addition, the two fully connected layers of the YOLO model is added as a separate branch to the SlowFast network to predict the bounding boxes.

ID	Event Type
1	Person loading an Object to a Vehicle
2	Person Unloading an Object from a Car/Vehicle
3	Person Opening a Vehicle/Car Trunk
4	Person Closing a Vehicle/Car Trunk
5	Person getting into a Vehicle
6	Person getting out of a Vehicle
7	Person gesturing
8	Person digging
9	Person carrying an object
10	Person running
11	Person entering a facility
12	Person exiting a facility

Figure 1: Human activity classes in the VIRAT dataset Release 2.0. The right figure shows an example of a cropped image with a bounding box used during training step

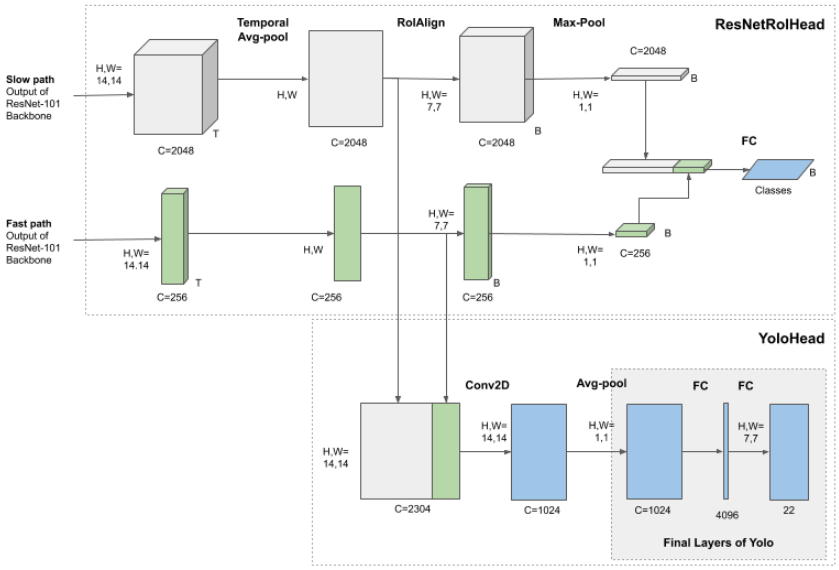

Figure 2: SlowFast Network ResNet RoI head augmented with YOLO head

4.1 Preprocessing

We keep most of the existing preprocessing steps the same as they are performed on the AVA dataset. The steps consist of randomly cropping the image to be a square of size 224 by 224 pixels. The original cropping method randomly crops the image regardless of where the bounding box is located, which could lead to the cropped image not including the bounding box. Unlike with the AVA dataset, this is very likely to happen with the VIRAT dataset, because it consist of higher resolution videos and the bounding boxes are relatively small. To avoid the bounding boxes being excluded from the cropped image, we modify the cropping algorithm such that the bounding box always gets included in the cropped image. Other preprocessing steps include horizontal flipping and scaling the pixel values to be in the range [0, 1].

4.2 Model

We modify the baseline SlowFast model by adding the last two fully connected layers of the YOLO model as a separate branch as shown in Figure 2. We fuse the slow and fast paths by concatenating the output of temporal pool layers of the two paths. We then use a Conv2D layer with in order to reduce the number of channels to match the original YOLO model. We perform global average

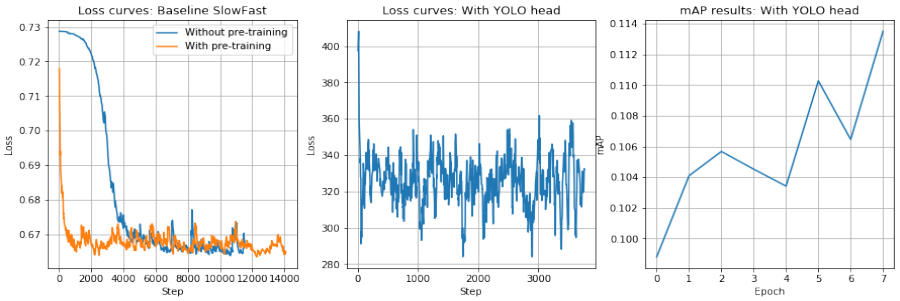

Figure 3: Left: Loss curves of training with and without pre-trained weights. Center: Loss curves with YOLO head. Right: mAP with YOLO head

pooling with kernel size 14 and flatten to further reduce the output dimension to 1024. Otherwise, the number of parameters of the model explodes because of the subsequent fully connected layers. Since we have twelve classes to predict, the size of the final output layer is 7*7*(B*5+12) where B=2. The final output layer uses Sigmoid activation to keep the center coordinates, width, and height of the bounding boxes to be within the range [0,1].

We use the ResNet 101 as the backbone network. We take 32 frames around the key frame where the human activity occurs and downsample by two to process 16 frames. We configure the parameter $\alpha = 4$ so that the slow path processes four times less frames than the fast path. However, we set $\beta = 8$ so that the fast path has one eighth of the number of channels as the slow path.

The number of parameters in the baseline SlowFast model is 59,185,748. With the YOLO head, the number of parameters is increased to 89,034,324.

4.3 Training

We use p2.xlarge instance with a single GPU with 12 GiB of memory for training the network. We can fit a batch size of three without freezing the earlier layers. With the earlier layers frozen, we can train with batch size of 16. The learning rate policy uses warm up time of four epochs to slowly ramp up the learning rate from 0.0001 to 0.0375. The learning rate then decreases by a factor of ten at epoch 41 and another factor of ten at epoch 49. Stochastic gradient descent with Nesterov momentum of 0.9 and weight decay of 1e-4 is used as an optimizer. The overall loss function is binary cross entropy function for multi-label classification summed with the YOLO loss function. We initialize with the pre-trained weights on the AVA dataset and fine-tune the weights with the VIRAT dataset.

To speed up the training, we also try freezing the first four ResNet stages of the backbone network and only train the last ResNet stage, the RoIAlignHead and the YoloHead blocks.

We use 8463 unique bounding boxes for training. For each bounding box, the fast path processes 32 frames while the slow path processes 8 frames. Without freezing the earlier weights, it takes about 2.5 hours to train one epoch. With the first four ResNet stages frozen, it takes around one hour to train one epoch.

5 Results and Discussion

The results show that using the pre-trained weights from training on the AVA dataset greatly helps in speeding up training when compared to training from scratch. As shown in Figure 3, the loss curve converges much faster when pre-trained weights are used. However, the loss plateaus early in the training and does not improve further. The mean average precision score achieved with the validation set is very low at around 0.1 as shown in Table 1. While the class "Person carrying an object" achieves the average precision of 0.98, it is likely because the dataset is imbalanced and mostly consists of that class.

The YOLO head did not produce good bounding boxes either. The output bounding boxes always seem to be one of the four quadrants as shown in Figure 4 and are not very tight around the human activity. The mean IoU value for 2,000 validation annotations is 0.011. The max IoU value is 0.047.

Class	Average precision
Person Closing a Vehicle/Car Trunk	0.00069
Person Opening a Vehicle/Car Trunk	0.00068
Person Unloading an Object from a Car/Vehicle	0.00121
Person carrying an object	0.97643
Person digging	nan
Person entering a facility	0.00348
Person exiting a facility	0.03385
Person gesturing	0.04168
Person getting into a Vehicle	0.08769
Person getting out of a Vehicle	0.00339
Person loading an Object to a Vehicle	0.00032
Person running	0.02116
mAP	0.10642

Table 1: mAP results

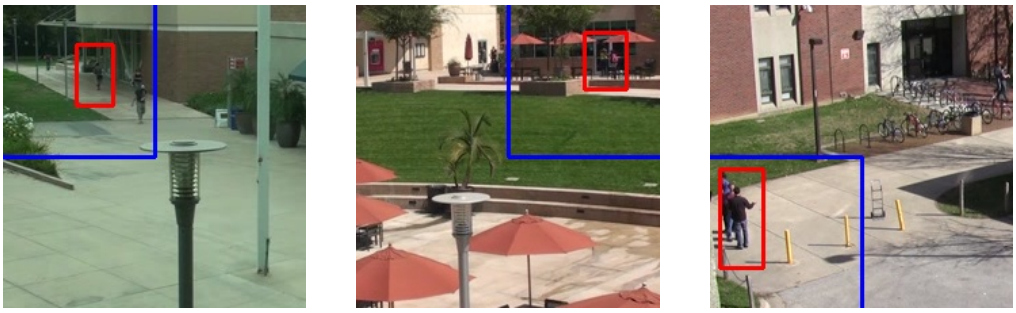

Figure 4: The left figure shows "Person carrying an object". The center shows " Person entering a facility". The right shows "Person gesturing". The red box is the ground truth annotation. The blue box is the bounding box output by the YOLO head.

The potential causes of the poor perfrmance include: misalignment of bounding boxes and the activity in the training data, relatively small size of the human subjects in the video, and incorrect implementation of YOLO loss function.

6 Conclusions

By combining the SlowFast network and architecture of YOLO, we have a hybrid model that can predict class probabilities as well as predict the bounding boxes. While the original SlowFast network is useful in predicting human activity given the ground-truth bounding box, the hybrid model is useful classifying human activity in surveillance videos where the spatial and temporal location of the activity is not known a priori.

More validation of whether the preprocessing step was done correctly to prepare the VIRAT video frames and annotations for ingestion by the SlowFast implementation is neded. The correctness of the YOLO loss implementation needs to be further validated as well to improve the performance of the model. Different learning rates need to be explored to escape the plateau loss function as well.

References

[SZ14] Karen Simonyan and Andrew Zisserman. “Two-Stream Convolutional Networks for Action Recognition in Videos”. In: *CoRR* abs/1406.2199 (2014). arXiv: 1406.2199. URL: http://arxiv.org/abs/1406.2199.

[Moo+15] Jinyoung Moon et al. “ActionNet-VE Dataset: A Dataset for Describing Visual Events by Extending VIRAT Ground 2.0”. In: *2015 8th International Conference on Signal Processing, Image Processing and Pattern Recognition (SIP)*. IEEE. 2015, pp. 1–4.

[Red+16] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[Wan+16] Limin Wang et al. “Temporal Segment Networks: Towards Good Practices for Deep Action Recognition”. In: *CoRR* abs/1608.00859 (2016). arXiv: 1608.00859. URL: http://arxiv.org/abs/1608.00859.

[Dib+17] Ali Diba et al. “Temporal 3d convnets: New architecture and transfer learning for video classification”. In: *arXiv preprint arXiv:1711.08200* (2017).

[Ma+17] Chih-Yao Ma et al. “TS-LSTM and Temporal-Inception: Exploiting Spatiotemporal Dynamics for Activity Recognition”. In: *CoRR* abs/1703.10667 (2017). arXiv: 1703.10667. URL: http://arxiv.org/abs/1703.10667.

[LPT18] Diogo C. Luvizon, David Picard, and Hedi Tabia. *2D/3D Pose Estimation and Action Recognition using Multitask Deep Learning*. 2018. arXiv: 1802.09232 [cs.CV].

[Fei+19] Christoph Feichtenhofer et al. “Slowfast networks for video recognition”. In: *Proceedings of the IEEE international conference on computer vision*. 2019, pp. 6202–6211.

[KKA20] M. Esat Kalfaoglu, Sinan Kalkan, and A. Aydin Alatan. *Late Temporal Modeling in 3D CNN Architectures with BERT for Action Recognition*. 2020. arXiv: 2008.01232 [cs.CV].