Object Detection and Activity Recognition

Sravani Teeparthi
Department of Electrical and Computer Engineering
University of New Mexico
Albuquerque, New Mexico

***Abstract*—Human action recognition has been an important video analysis problem for last decade due to its diverse applications. In addition to surveillance, video classification, gaming, human activity detection and localization can help in aiding projects that need to analyze activities across big data set. In this paper I am working on videos that help educational researchers analyze collaborative learning among middle school students. This has been funded by NSF and collected 2000 hours of videos. The goal of the project is to locate paper and localize "writing/no-writing" activities. This paper provides overview of detecting writing in a given video using Neural Networks.**

I. Motivation

Action recognition involves capturing spatiotemporal context across frames. In addition to this, spatial information captured need to be compensated for camera movement. In AOLME videos multiple human activities exists such as typing, talking , eating etc with multiple people, so it will be a great challenge to recognize actions in this dataset. The objective of this paper is to use pre-trained network of faster R-CNN to detect paper, and then classifying every 3sec as writing or no-writing using 3D-CNN and also to optimize different 3D-CNN architectures and pick a stable and well performing architecture..

II. Dataset

The Advancing Out-of-school learning in Mathematics and Engineering(AOLME) project is an after school program implemented by ECE and Department of Language, Literacy and Sociocultural Studies. AOLME generated large amount of video data which includes around 2000 hours of group interactions, monitor data and screen recordings. Each video recorded reflects small group of middle school students interacted with each other and worked collaboratively in their learning process. For each student, AOLME collected a video each week, up to 11 weeks for a total of 20 hours.VIdeos are characterized by strong illumination variation sand students moving around within their groups or to join groups.

For this project we work on videos that simulate AOLME group setting while removing illumination variations and noise caused by movement of students. These videos are recorded in ECE 218 lab using the same cameras that are used capture AOLME videos. In these videos we captured talking and writing, two important activities that could aid educational researchers. This paper specifically focuses on Writing and No-Writing. There are **4 rounds of writing**,(R1, R2, R3, R4), and **5 rounds of no writing**, (ESP, Hi, No, Number, Yes)

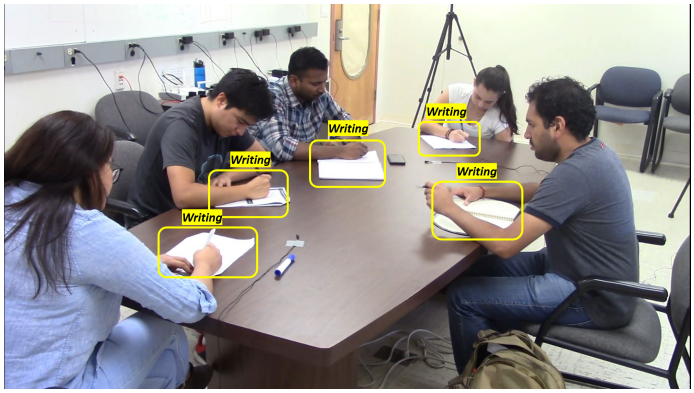

Fig. 1: frame showing writing activity

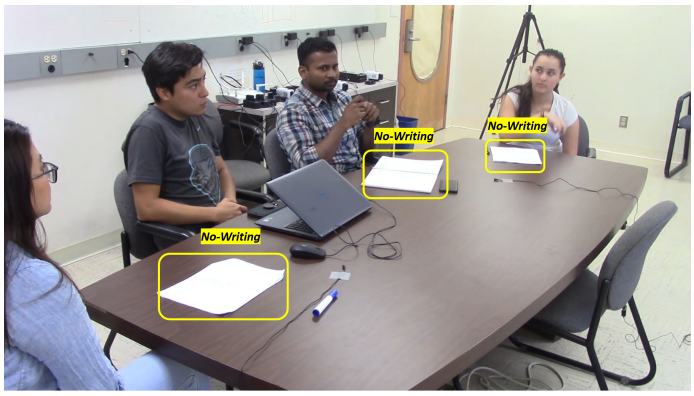

Fig. 2: frame showing no-writing activity

captured using **5 cameras** at **1920×1080** resolution and **30 fps** (frames per second).

We split these videos into training, validation and testing sets as described in Table. II . All the videos have 5 camera angles. So the number of videos become 5times. We prepare groud truth on these videos for paper detection and writing recognition as described in sections II-A and II-B

TABLE I: Training, Validation and Testing splits

Split	Writing	No-Writing
Training	R4(42 sec), R2(08 sec)	ESP(56 sec), No(03 sec), Yes(03 sec)
Validation	R1(13 sec)	Hi (23 sec)
Testing	R3(17 sec)	Number(30 sec)

A. Paper detection Dataset:

The resolution of the videos used are 1920*1080 @ 30fps and sampling interval is 1 image for every 10 frames.

TABLE II: Dataset for paper detection

	Training	Validation	Testing
Videos	R4, R2, ESP, No, Yes	R1, Hi	R3, Number
Images	1708	531	650

B. Dataset for writing detection:

TABLE III: Dataset for paper detection

	Training		Validation		Testing	
Videos	R4, R2	ESP, No, Yes	R1	Hi	R3	Number
Samples	286	254	46	91	55	128
Total	540		137		183	

III. Methodology

Our current method for detecting writing is as below:
Video are passed through Object Detection algorithm to detect paper and then bounding boxes are saved. Using the bounding boxes in the first frame, the video is trimmed into 100*100 for every 3sec as numpy arrays. These numpy arrays are classified as writing and no-writing using Activity Classifier. The block diagram is as referenced in the figure 4

A. Paper Detection

For detecting paper I re-trained Faster-RCNN inception v2 [1], [2] which is trained on coco dataset.

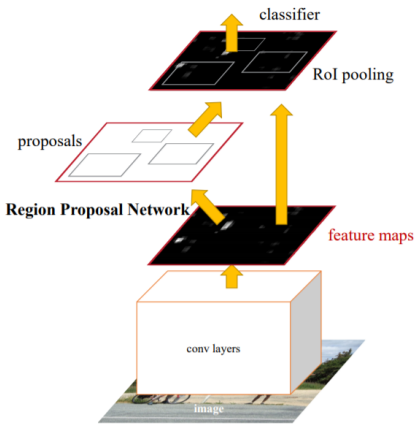

Fig. 3: Illustration of Faster R-CNN

Architecture of Faster RCNN is shown in 5

COCO is large-scale object detection,segmentation and captioning dataset.
Features of COCO:

- 330K images (>200K labels)
- 1.5million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- 250,000 people with keypoints

The Library used is TensorFlow Object Detection library(1.14).
Training setting used are :
Number of epochs = 20000.
Batch size = 1.
Time taken for training = less than 3hours

B. Activity Recognition

Data used for Activity Recognition are 100*100 videos of 3sec (90 frames). These are converted into numpy arrays which has three channels R,G,B.
The architecture used is inspired by C3D and the number of layers used are 4. The structure followed defining layer is Batch Normalization followed by Convolution followed by Pooling. Number of Fully Connected Layers are 2 with a dropout ratio of 0.5.
Training settings:
Number of epochs = 50
Early stopping = Yes (Validation loss)
Batch size = 11
Time taken per epoch = 13 seconds (Nvidia GTX 1080 - dual)

IV. Results:

A. Paper Detection

The table IV shows the results for paper detection.
The figure 6 shows the results for detection

B. Activity Recognition

Different architectures of 3D-CNNs are evaluated to analyze stability. The figure 7 shows the Comparison of Architectures.
Different combinations evaluated are
Number of kernels = {4,8,16}
Number of Fully connected units= {10,25,50,100}
Number of runs used for evaluation = 21
From figure 7, Best Architecture is the one which has

- Small box
- Positioned at the top

TABLE IV

Metric	Value
Average Precision @[IoU = 0.50:0.95—area = all—maxDets = 100]	0.406
Average Precision @[IoU = 0.50 —area = all—maxDets = 100]	0.859
Average Precision @[IoU = 0.75 —area = all—maxDets = 100]	0.251
Average Precision @[IoU = 0.50:0.95—area = small—maxDets = 100]	-1.000
Average Precision @[IoU = 0.50:0.95—area = medium—maxDets = 100]	0.055
Average Precision @[IoU = 0.50:0.95—area = large—maxDets = 100]	0.435
Average Recall @[IoU = 0.50:0.95—area = all—maxDets = 1]	0.198
Average Recall @[IoU = 0.50:0.95—area = all—maxDets = 10]	0.515
Average Recall @[IoU = 0.50:0.95—area = all—maxDets = 100]	0.537
Average Recall @[IoU = 0.50:0.95—area = small—maxDets = 100]	-1.000
Average Recall @[IoU = 0.50:0.95—area = medium—maxDets = 100]	0.163
Average Recall @[IoU = 0.50:0.95—area = large— maxDets = 100]	0.569

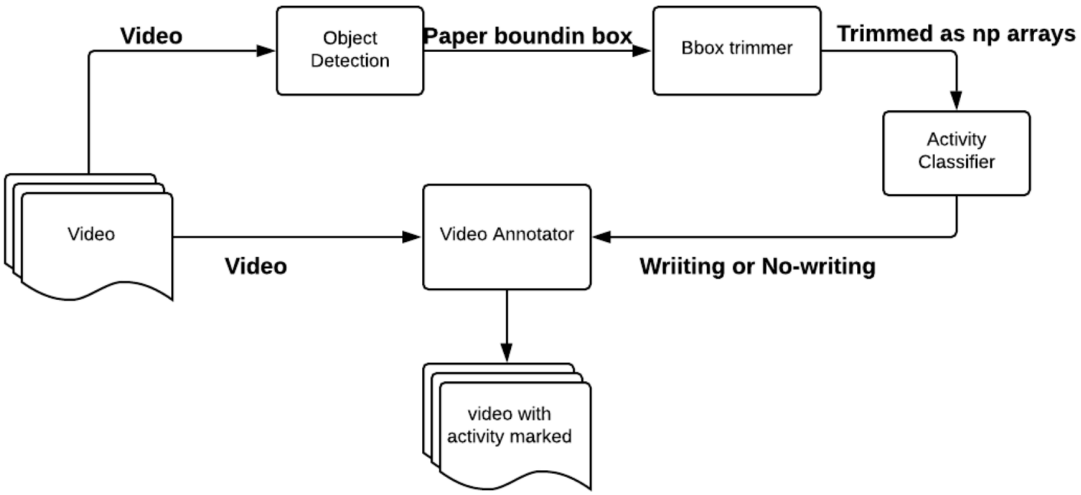

Fig. 4: Block diagram of method

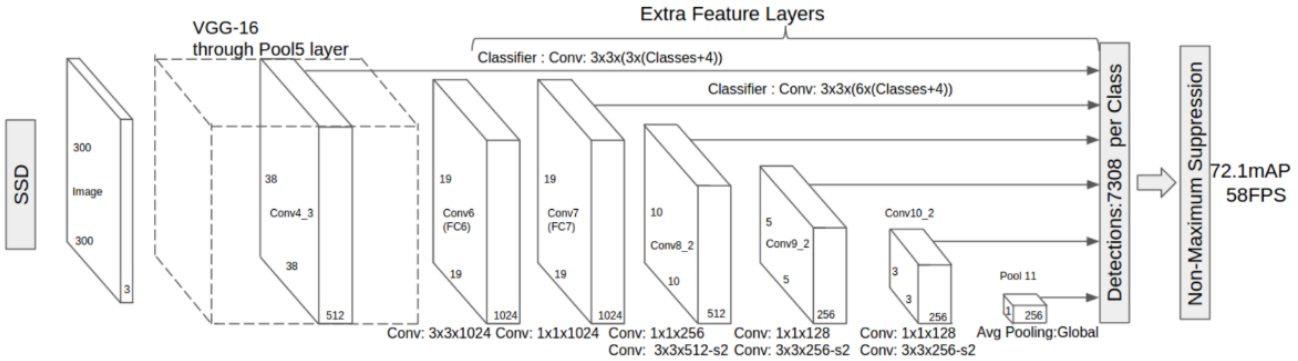

Fig. 5: Architecture of Faster R-CNN

Fig. 6: Result for paper detection

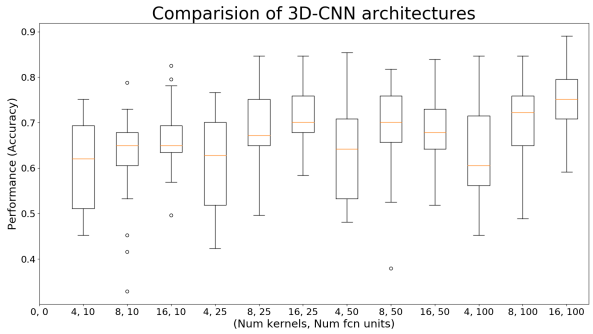

Fig. 7: Comparison of Architectures

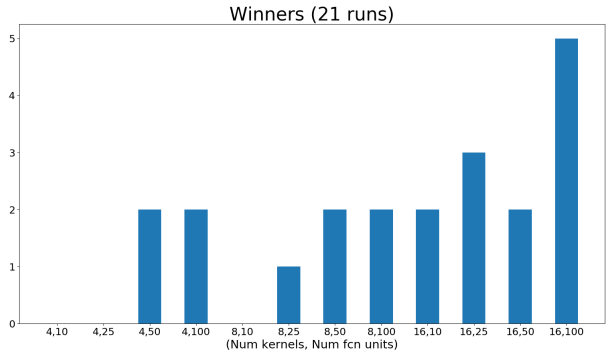

Fig. 8: Winning Architecture

- Less outliers

For every run, best validation accuracy is listed out and the winning architecture is selected. The winning architecture is shown in 8

The winning architecture here is the one with (Number of kernels, Number of FCN units) = (16,100)

Validation and Test Accuracy for winning Architecture:

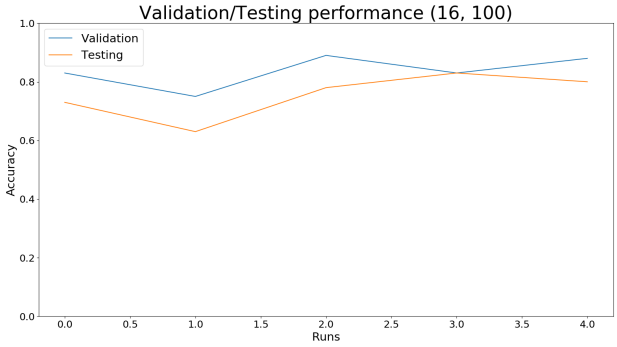

Fig. 9: Plot of validation and test accuracy

Plot of training and Validation accuracies is shown in figure 10

V. FUTURE WORK

- Expand object detection dataset with diversity.
- Add Keyboard detection to object detection methods.
- Increase Dataset and do data augmentation to avoid over-fitting.

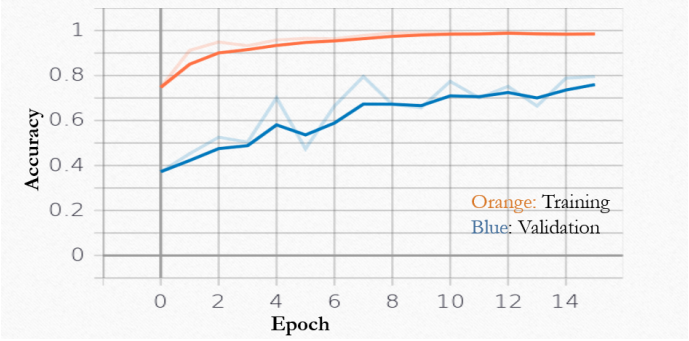

Fig. 10: Plot of training and validation accuracy

REFERENCES

[1] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[2] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[3] pkulzc, tombstone, and nealwu, "Tensorflow object detection zoo," https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md, Oct. 2018.

[4] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*. Springer, 2016, pp. 20–36.

[5] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2012.