C3D-Inspired Video Classification

Anshul Verma
av.vermaans@gmail.com

***Abstract*—In this document, I am summarizing attempt on my task to perform video classification using 3-D convolutional architectures. I implemented architectures inspired from C3D architecture and attempted different approaches like OverSampling, and Weighted Average Sampling to tackle data imbalance. The results look promising achieving 54.17% validation accuracy and 39.65%. The results obtained were benchmarked against fine-tuning ResNet3D architectures trained on Kinematics data.**

I. Introduction

Video classification is an important task in computer vision, with applications ranging from surveillance to entertainment. 3D convolutional neural networks (3D CNNs) have shown remarkable success in video classification tasks, by capturing both spatial and temporal dependencies in the video data. In this document, we aim to summarize our task of performing video classification using 3D CNNs. We implemented architectures inspired from C3D, and experimented with different approaches to tackle data imbalance, including Oversampling and Weighted Average Sampling.

Since C3D [1], there has been serveral advancements in 3D CNN domain like [2] and [3]. Pre-trained weights for some of these architectures trained on Kinetics-700 datasets [4] are available, and fine-tuning these models on the given dataset was used to compare the results.

II. Methodology

The architecture approach implemented is inspired by the C3D architecture [1], C3D architecture is an old architecture with no Batch Normalization and two FC layers, which adds a lot of parameters and makes it difficult to train with small amount of data. So I tried to add Batch Normalization and replace the FC layers with just one layer.

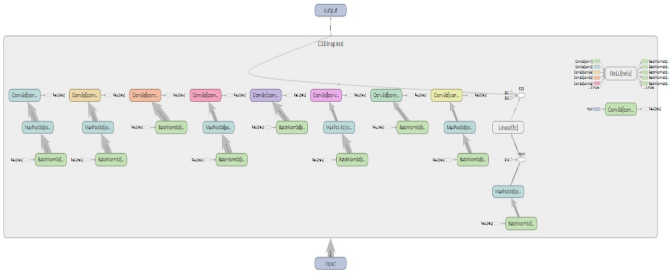

Fig. 1. Model Architecture

One input video was considered to have 300 frames at 30 fps of shape 640×480 which makes the input shape quite big, so to make the training and prediction manageable a video was split into 10 videos by randomly selecting nth frame every 10 frames. This n once decided was fixed for the entirety of the video, and also the frames were resized to shape 224×224.

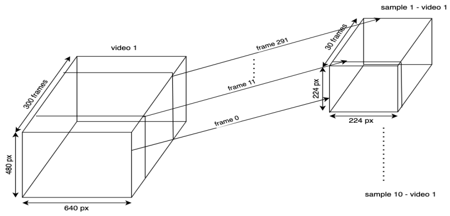

Fig. 2. Sampling video into sub-videos

The need of sub sampling is to make the input size of the model more managable and to be able to achieve higher batch-size for training. Also to reduce the number of parameters in the models as it will help not over-fit on the training data that we have.

The input video data was also augmented using torchvideotransforms [5], it treats all the frames in a clip like it treats a frame and it decides randomly of the given augmentation which combination to use for all the frames in a clip.

All the data was split into training-validation-test sets stratifying the labels. Ensuring that the labels are evenly distributed across classes. On stratify split, the validation set didn't get any sample from class ”00” so a sample of this class was added from training to validation set. The data given was highly imbalanced and to deal with this class imbalance different approaches like Over-sampling, and Weighted-Over-sampling were used.

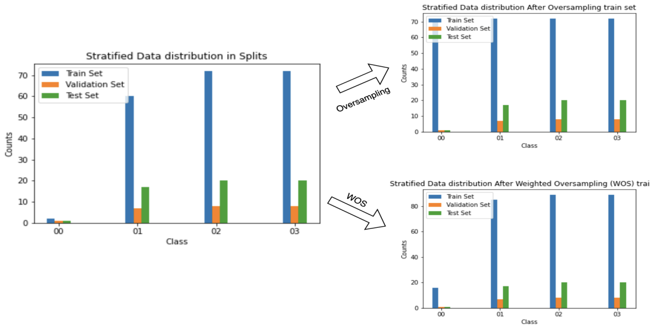

Fig. 3. Oversampling less-frequent video in training set

For test and validation predictions the maximum occurring prediction of the 10 sample videos is considered to be the predicted class (Ensemble). *(Note: another, possible approach could be to average the prediction probabilities of all the 10 sample videos and then decide the class).*

A. Training

For normal classification, SGD optimization along with CrossEntropyLoss was used. Because of Batch Normalization in C3D Inspired a higher learning rate was used in training $1e-2$, with weight decay. Allowing the model to learn quickly. Stochastic Gradient Descent (SGD) optimizier argument were set to $momentum \longrightarrow 0.9$, $nestrov \longrightarrow True$ and $weight\_decay \longrightarrow 1e-4$. Batch size for each experiment was set to maximum that it could be set to be able to train in the GPU. Also for fine-tuning ResNet-3D architecture a lower learning rate $1e-4$ was used.

III. Results

Table I, shows the result of the C3D Inspired architecture compared to other methods.

Method	Training	Validation	Test
ResNet3D-18 (fine-tuned, Kinematics-700)	46.60%	29.17%	39.65%
C3D Inspired	45.63%	54.17%	**39.65%**
C3D Inspired Over-Sampling	67.36%	54.17%	37.93%
C3D Inspired WOS	52.94%	54.17%	**39.65%**

TABLE I
Results

Training curves and confusion matrices of the C3D Inspired with Weighted Average Sampling (WOS) model are shown below,

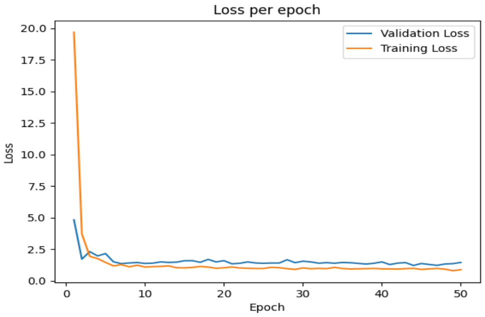

Fig. 4. Training and Validation Loss of C3D Inspired WOS

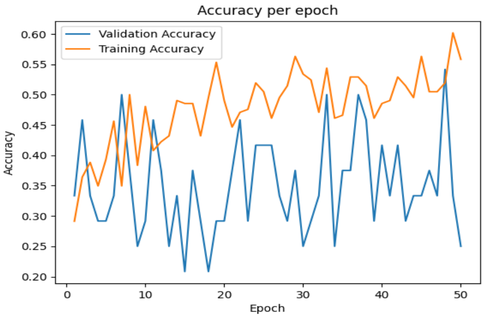

Fig. 5. Training and Validation Accuracy of C3D Inspired WOS

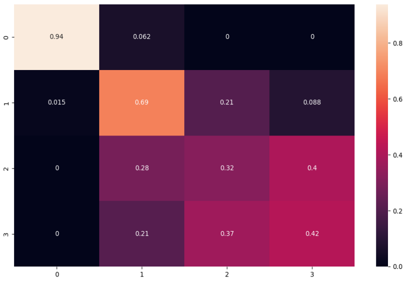

Fig. 6. Confusion Matrix of best epoch on training set

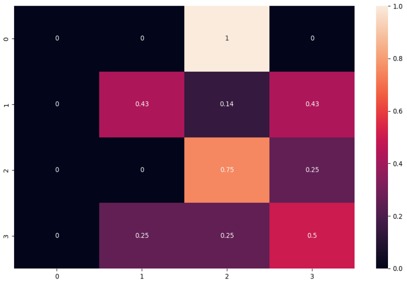

Fig. 7. Confusion Matrix of best epoch on validation set

IV. Conclusion

The over-all task to classify the videos in the dataset is not easy, the classes look like how attentive the other person is, for which facial expression and eye needs to be tracked. The imbalance on the least attentive class makes it even worse to train a model to classify these videos. The problem seems to be an ordinal classification problem. Therefore prediction class **”00”** as **”01”** should not have the same loss as prediction **”00”** as **”03”**, as the classes seem to be ordinal.

This idea, behind sub-sampling the video into smaller videos was stimulated by the work of Kataoka et al. [2], where consecutive 15-16 frames of the video are used for classification. However, unlike Kinetics dataset where the class is identifiable by any frame, in our dataset, the class is identifiable by user action over the entire video. Thus, to make the input manageable, different sampling method was implemented.

Overall, I think the C3D architecture performs well at the desired task, the imbalance of data is too extreme to be dealt with just oversampling and generated augmentation to randomly change background behind people in different videos might help improve the performance on class **”00”**. I would also like to implement C-RNN architectures to see how well it performs on this dataset, without any sampling because that might be causing some information loss which can be a reason for low performance.

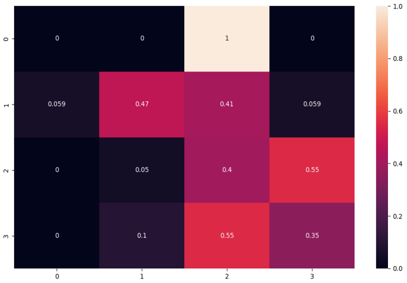

Fig. 8. Confusion Matrix of best epoch on test set

References

[1] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[2] H. Kataoka, T. Wakamiya, K. Hara, and Y. Satoh, “Would mega-scale datasets further enhance spatiotemporal 3d cnns?” *arXiv preprint arXiv:2004.04968*, 2020.

[3] K. Hara, H. Kataoka, and Y. Satoh, “Towards good practice for action recognition with spatiotemporal 3d convolutions,” in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 2516–2521.

[4] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, “The kinetics human action video dataset,” *arXiv preprint arXiv:1705.06950*, 2017.

[5] hassony2, “torch-video-transforms,” 02 2020. [Online]. Available: https://github.com/hassony2/torch$_videovision$