Deep Learning Project - Action Recognition

Constantin Tudor Bara
Orzata Andrei
Superviser: Emanuella Haller

I.Introduction

Our project consist in testing the performance of different spatiotemporal convolutions for Action Recognition on the UCF101 dataset [2] and comparing our results to ones found in the study A Closer Look at Spatiotemporal Convolutions for Action Recognition[1] .

II.CNNs

Introduction

The variants we chose to test in our project are ResNets that are as described in the study[1], in order to compare results. We will also consider the input clip to have a size of $3 \times L \times H \times W$ (H = height, W=width, L = the number of frames, 3 – RGB channels), the residual blocks won't have bottlenecks each will consist of two convolutional layers witha ReLU activation function.The sequence of blocks will culminate into a top fully-connected responsible for the final classification.

R2D consist of performing 2D convolutions over the entire clip while f-R2D consists in processing the L frames independently through a series of 2D convolutions. In both cases, no temporal modeling is performed. We see an example of R2D in the first picture.

R3D consist of performing 3D convolutions which preserve temporal information and propragate it through the layers we can se an example of this in the second picture

MCx and rMCx are mixed convololutions of performing 2D-3D convolutions . They are used because things 3D convolutions are particulary important at the begining of motion modeling, but as semantic abstraction increases they might become unnecessary further down the line.

R(2+1)D is the centerpiece of our testing and consists of 2D convolutions followed by 1D convolutions, this is done because there is hypothesis that this approximates 3D convolutions more conveniently.This proposed architecture takes the R3D model and replaces every Conv3 with the R(2+1)D block. This block uses a 2D convolution for spatial data, and a 1D convolution for temporal data, (in implementation we use Conv3D with kernels of size 1xDxD and Tx1x1). Between the 2 decomposed layers, we put one batch normalization and one non-linearity (ReLU), thus we double the number of non-linearities.

We started preparing the layers by implementing a ResNet18, through a number of filters ranging from 64 to 512 across the layers. This was not ideal because our testing dataset is small and it was causing problems with training due to overfitting, so we reduced the number of layers down to 6 . The first layer with is stem, reducing the size to 7x7, 4 convololutions with 32 and 64 filters and a fully connected last layer. We also eliminated the residual values, the small size of the network means we have no problem with the vanishing gradient and the propagation of an input that is not processed leads to overfitting.

III.Dataset

Introduction

For this project we used the [2]UCF101, a dataset consisting of 13k clips and over 27 hours of footage of human actions categorized in 101 action classes. According to its creators the dataset is one of the larg -est datasets of human interaction and provides base line action recognition results approching an over -all performance percentage of 43.9% making it one the most challenging dataset of its kind due to the size and unconstrained nature of its clips.

Dataset Details

UCF101 has 101 action classes divided in five categories which are in order: Human-Object Interaction, Body-Motion Only, HumanHuman Interaction, Playing Musical Instruments, Sports. Each action class has 25 groups of 4 to 7 clips that contain common features such as the background or the same actors. In to tal there are 13.320 clips,averaging around 7.21s with the minimum clip length being 1.06s and the maxi -mum 71.04s, running at 25 frames per second and having a resolutionof 320x240px. For 51 of the action classes the clips also display audio.

Preparing the Data

For this project we had to prepare the data in order to process it. We sampled 10.000 clips from 570 videos , coategorized in 6 different action classes and we preprocessed them by dividing them into 8 frames with 8 steps in between them. Next we normalized the clips and rescaled them from 240x340 to 128x171 then we randomly cropped them into a 112x112 apect.

IV.Results

Test Details

For our tests we sampled 220 videos totalling 3300 clips belonging to six action classes (EyeMakeup, Lipstick, Archery, Baby, BalanceBeam and Marching Band) and we let each test run over 5 epochs.

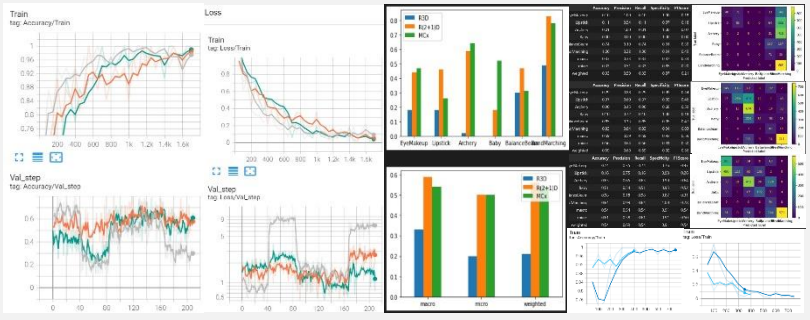
Graphs

The R3D, R(2+1)D, MCx are the most documented models we tested and make up most of our graphs.

On the Train(Accuracy/Train), Val_step (Accuracy/Val_step) , Loss(Loss/Train), Val_step (Accuracy/Val_step) the R3D is represented by gray, MCx is represented by orange and R(2+1)D is represented by deep green.

The confusion matrices and tables are in order, from top to bottom R3D, R(2+1)D, MCx.

Last table represents the R2D and fR2D training data after one epoch displaying similar behaviour.



V.Conclusion

Our experiments seem to support the conclusion found in the study, that the R(2+1)D arhitecture achieves superior results in action recognition compared to other CNNs.

VI.References & Materials

[1] https://arxiv.org/pdf/1711.11248.pdf **A Closer Look at Spatiotemporal Convolutions for Action Recognition** - Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann Lecun, Manohar Paluri
[2] https://www.crcv.ucf.edu/data/UCF101.php **UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild** - Khurram Soomro, Amir Roshan Zamir and Mubarak Shah
[3] https://torchvideo.readthedocs.io/en/latest/transforms.html#normalizevideo
[4] https://pytorch.org/vision/0.8/models.html#video-classification
[5] https://www.tensorflow.org/api_docs/python/tf