WITS
UNIVERSITY

3D Convolutions for deep learning and its applications to surveillance

Supervisor, Dr Richard Klein

Raesetje Bonjo Sefala, [redacted]

November 16, 2017

[redacted]

1 Abstract

A Convolutional Neural Network (CNN) is a form of deep learning neural network model used to automate the task of feature extraction and to effectively classify multimedia. A 2 Dimensional Convolutional Neural Network (2D CNN) is a CNN model which uses a 2D filter to capture spatial features from an image frame while a 3D CNN uses a 3D filter to capture both spatial and temporal features from a group of image frames(video). Although both models can be used to classify videos, one might argue that the temporal dimension property in 3D CNNs allows them to better recognize actions in videos since they can capture the motion information found in between the image frames of the video. This paper proposes to evaluate properties of Convolutional Neural Network models and understand the conditions when either model becomes superior. The best model will be used to classify human actions from videos being fed to the model in real time.

2 Introduction

The idea of enabling machines to see and recognize things in the world like humans can, has been around for many years. Cameras and other sensing equipment have been developed to try give machines vision and this has then encouraged the development of algorithms which try contextualize this data for different applications including surveillance. Convolutional Neural Networks(CNNs) have recently proved to be quite effective at analyzing media data such as videos, images, text and audio[LeCun et al. [2015]]. They have 2 main parts, feature extraction and the neural network. These models have been giving state-of-the-art results on complex tasks such as image recognition and segmentation [Krizhevsky et al. [2012], Farabet et al. [2013]].

In this paper, we study CNNs on the task of human action recognition, we evaluate CNNs over 6 action classes on 2 different data sets, different architecture builds and hyper-parameters. From a modeling point of view, we would like to understand which specifics determine high recognition accuracy? Whether the motion information found in the temporal dimension helps to improve the model's performance? What affects the time taken for the model to make a prediction? We examine these questions by building and evaluating different CNN architectures on the KTH data set and the (Nanyang Technological University)NTU RGB+D action recognition dataset [Shahroudy et al. [2016]].

From a computational perspective, since CNNs on large data sets and with millions of parameters require long periods of training time on high end equipment[Karpathy et al. [2014]] we have decided to sample only 1868 RGB videos from the NTU RGB+D action recognition data set. Given that we are working with a relatively small data set, we will be exploring different architecture designs and thus evaluating these models.

3 Related work

CNNs are biologically inspired models, they use a set of filters during the convolution stage to try to automatically pick the best features to describe the image frame [Ji et al. [2013]]. Motivated by the acknowledgement that the world is complex and as a result deciding on which hand crafted features to use for video classification can be difficult. [Geng and Song [2015]] describes the limitation of the 2D CNN model as: that it only takes in 2D inputs[Geng and Song [2015]] shows the CNN architecture from a high level perspective while figure 4 [Ji et al. [2013]] shows a high level 3D CNN architecture.
In this paper we evaluate the some properties of CNNs (including pre-processing techniques)and the characteristics they influence.

4 Methodology

4.1 The data sets

We will be evaluating the models build using two data sets. The KTH data set is the primary data set, it consists of 2391 videos in grey scale and with a homogeneous background. The videos

have 25 subject but with only one subject appearing per video. The camera used to create the data set was placed at the same point in physical space.The videos have 6 categories of human actions(walking, jogging, running, boxing, hand waving and hand clapping). Figure 1 shows some sample frames from the data set [Laptev and Lindeberg [2004]].

Figure 1: Samples of the KTH data set

The second data set, the NTU RGB+D data set consists of 60 categories of 2040 high quality daily human action video data. For this experiment, we have only considered 6 categories (Drinking water, Sitting down, Wearing jacket, Take selfie, Touch other person's pocket and Walking towards each other). The camera used for this data set was being moved around the room, capturing different viewpoints. Figure **??** shows some sample frames from the data set.

4.2 Preprocessing

All videos were changed to gray-scale and then resized to 20 by 20 frames. To reduce the number of frames, 11 frames were sampled from the 10th frame of each video's total number of frames and then ordered to represent the full video sequence. The data was then normalized and then fed into the model through a 3-fold cross-validation split.

4.3 The Models

4.3.1 2D CNN Model- Averaging

The first model was a 2D CNN. This model ad 10 layers(3 convolutions, 2 sub-sampling and then the fully connected layers). the input to the model was the sampled 11 frames from the data set, the model ran 11 times and then the result was the mean output. This resulted in a very weak classifier(30% - 34%) on both models probably because the similar image frames appeared in all

Figure 2: Samples of the NTU RGB data set

models (e.g. a person standing) but labeled differently. The model only took into account the spatial features of the independent video frames.

An alternative approach is to use a Time distributed wrapper around the model so that the frames are viewed as a collective(a video sequence) but processed on the 2D CNN. However the effect of this model combination has not been evaluated in this research.

4.3.2 3D CNN

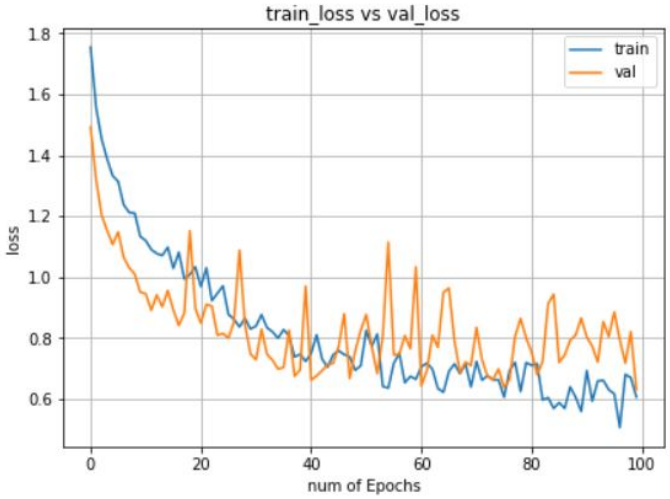
A simple 3D CNN model was also evaluated and with this architecture (Fig. 4). On 100 epochs, the model had 2 3D convolutional layers, 1 maxpooling layer and 1 dropout layer on the feature extraction part. We have also evaluated deeper models but given the two data sets, and their sizes the model performs poorly [redacted]

Given that the results of the 3D CNN were much better than those of the 2D CNN in terms of the loss function mainly and the training complexity, further fine tuning was done on only this

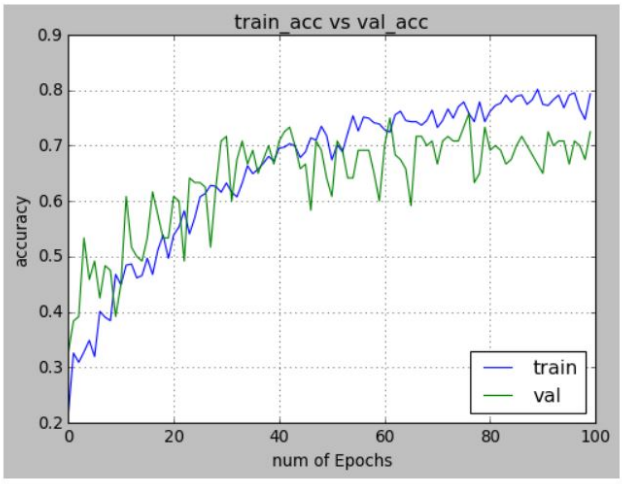
Layer (type)	Output Shape	Param #
conv3d_1 (Conv3D)	(None, 32, 12, 12, 11)	4032
conv1 (Conv3D)	(None, 64, 12, 12, 11)	55360
max_pooling3d_1 (MaxPooling3	(None, 64, 4, 4, 3)	0
dropout_1 (Dropout)	(None, 64, 4, 4, 3)	0
flatten_1 (Flatten)	(None, 3072)	0
dense_1 (Dense)	(None, 128)	393344
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 6)	774
activation_1 (Activation)	(None, 6)	0

Total params: 453,510
Trainable params: 453,510
Non-trainable params: 0

(a) The model's architecture



(b) Training loss vs Validation loss graph



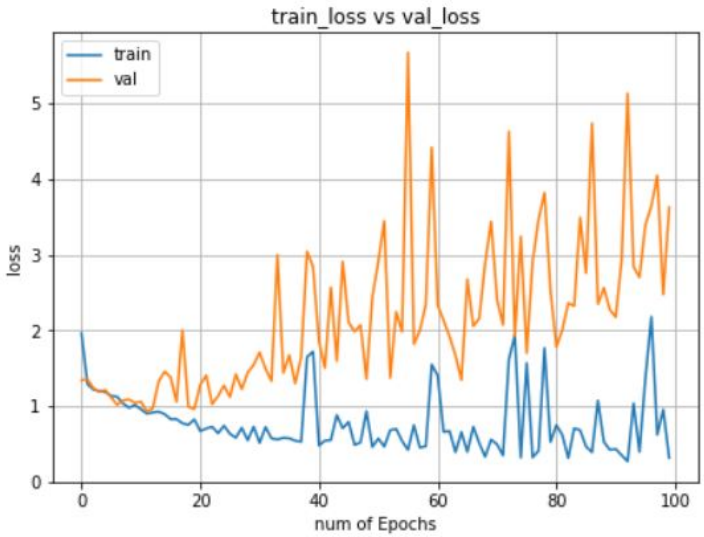
(c) Training accuracy vs Validation accuracy graph

Figure 3: Simple 3D CNN architecture evaluated on the KTH data set

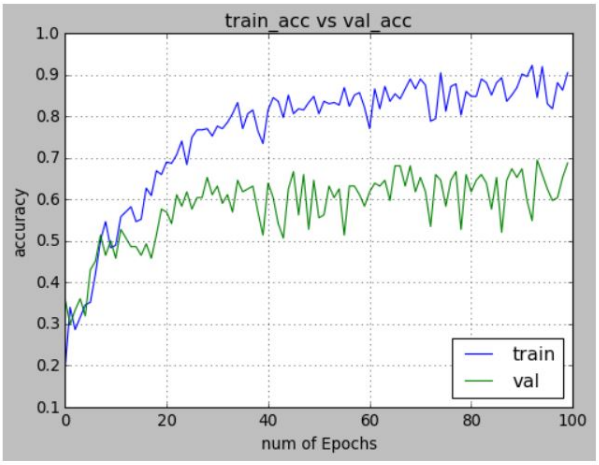
Layer (type)	Output Shape	Param #
conv3d_1 (Conv3D)	(None, 32, 16, 16, 11)	4032
max_pooling3d_1 (MaxPooling3	(None, 32, 5, 5, 3)	0
conv2 (Conv3D)	(None, 128, 5, 5, 3)	110720
pool2 (MaxPooling3D)	(None, 128, 2, 2, 1)	0
conv3a (Conv3D)	(None, 256, 2, 2, 1)	884992
conv3b (Conv3D)	(None, 256, 2, 2, 1)	1769728
flatten_1 (Flatten)	(None, 1024)	0
fc6 (Dense)	(None, 4096)	4198400
dropout_1 (Dropout)	(None, 4096)	0
fc7 (Dense)	(None, 4096)	16781312
dropout_2 (Dropout)	(None, 4096)	0
dense_1 (Dense)	(None, 6)	24582
activation_1 (Activation)	(None, 6)	0

Total params: 23,773,766
Trainable params: 23,773,766
Non-trainable params: 0

(a) The model's architecture



(b) Training loss vs Validation loss graph



(c) Training accuracy vs Validation accuracy graph

Figure 4: Deeper 3D CNN architecture evaluated on the KTH data set

architecture.

5 Results and discussion

The previous chapter introduced the hypothesis and the research methodology. This chapter lays out the details of the research plan on how the methodology will be carried out. This chapter will be divided into 3 stages. The first stage will be the detailed implementation of the methodology, the second stage will be about potential issues and the last stage will be the time plan.

5.1 Preprocessing

- **Size of the data set:**

  From observation, it appears to be that the type of data set you have plays a big role in the kind of model you build. Bigger and complex clean data sets makes for better models because it means that the model has a lot of content it can learn so that it can generalize better. From evaluation, pre-trained models on the kth data set did not generalize well on the NTU data set. This might be because the model is seeing a lot more new things on this data set since it has complex scenes(maybe because of the differences in camera positions-resulting in new frames all-together, the model does not get enough similar content to learn)

- **The number of frames for each video:** In reality we cannot control the number of frames an action is contained in, we would like to be able to define an action using the smallest number of frames as possible. This is not an easy task to achieve because actions seem to be quite complex because on a set number of frames, a person who is about to make a call and the one who is about to take a selfie could appear to be doing the same thing and depending on which frames we use to classify, it is easy to get it wrong. This is why we decided to sample the training frames instead of taking the nth frame on an interval and besides, the total frame count on each video in the data set is different.

- **n-fold validation split:** Given that we are working on relatively small data sets, it is easy to over-fit on deeper models and under-fit on more simple models. Different model architectures have been tested as well as different validation splits. Having n-fold validation splits helps give us an amount of confidence on our training and validation data. It helps ensure that we did not just pick a bad sample of training data and the same time allows us the model to train on all of the data so we can see variance.

- **Frame pre-processing:** Because we are trying to understand actions- a summary of a sequence of frames. We have decided to change all the RGB videos to gray scale since the color does not actually tell us much which we need to know. We have explored different frame sizes as well and small frame sizes($20\times$ 20 rows by columns) tend to give better result probably because the frames are compressed so that motion information is highlighted the most. Apart from better performance, these changes result in not so computationally expensive models.

5.2 The Model

Simple models on Huge complex data sets do not seem to have enough parameters to learn certain features about the data set to produce good results. On the other hand, it seems easier to just have deeper models to learn all features but that quickly results in over-fitting. The idea is to try strike a balance.

- **More convolution layers:** From evaluation these seem to over-summarize the features which are fed into the fully connected Neural Network especially if you have small frames as input. Similarly more pooling and sub-sampling layers tend to have the same effect.
- **Fully connected layers:** This part depends on the data at hand, it is easy to over-fit and under-fit as well, fine tuning plays a big role in making good models. The dropout layer seems to be improving the performance as well.
- **The 2D CNN model:** From observation, it seems like the temporal dimension plays a role in understanding action in a video. It is easy to have similar frames for different action categories in different videos. Simply aggregating or selecting a certain frame for classification generally does not give good results, unless the actions are very different from each other(e.g. action 1 = one person sitting down and action 2 = two people walking towards each other). In that case, the parameters can actually learn distinguishing characteristics for better classification.
- **Generalization:** We got better results on the KTH data set than on the NTU data set when using all of the models. This might be because of a number of reasons including that the KTH data set has a homogeneous background and all the focus can be on just one person moving though the frames. other reasons could be because of the camera capturing the frames from a single point, although we have atleast two different homogeneous backgrounds.

  The Simple model used to train the NTU data set produces better results when using the KTH data set sample for testing when compared to the KTH data set on NTU as testing data. This might be because the KTH trained model's parameters are unfamiliar with some properties of the complex scenes found in the NTU data set. It might be difficult to cope with complex things in the background such as chairs and tables it is seeing for the first time and moving people.

5.3 The Results

The 3D model which was trained using the KTH data set performed better than the NTU data set trained model(table 1). perhaps using a pre-trained model such as the sports 1-M model [Karpathy et al. [2014]] could yield better results.

Model	KTH validation set	NTU Validation set
Deeper	68,75%	17,2%
Simple	72.5%	60%

Table 1: 3D CNN model Results

6 Conclusion

Using 3D CNNs has indeed proved to be more efficient on action recognition than 2D CNNs, although the models need to be personalized to the training data so that we can learn as much abstract features as possible for better generalization. We were able to achieve up to 72% accuracy on testing data on the KTH data set and up to 60% accuracy on testing data on the NTU RGB data set as seen on Table 1.
Data category selection seemed to have played a huge role in classification since the more different the classes are the easier it is for the model to learn the different features.

For future work, the 2D CNN could perhaps perform differently if we could train it on a large data set of human action images. Advantages of this could include using less frames to classify an actions(which is desirable for real-time classification).

References

Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35 (8):1915–1929, 2013.

Chi Geng and JianXin Song. Human action recognition based on convolutional neural networks with a convolutional auto-encoder. In *5th International Conference on Computer Sciences and Automation Engineering (ICCSAE 2015)*, 2015.

Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Ivan Laptev and Tony Lindeberg. Velocity adaptation of space-time interest points. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 1, pages 52–56. IEEE, 2004.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.