Sequential Deep Learning with 3D-CNN for Human Action Recognition and Label Prediction

Pooja A. Somatkar, J. V. Megha
Information Technology, Department
SGGSIE&T, Vishnupuri, Nanded
Maharashtra, India

Abstract: Human action recognition plays a vital role in many applications. Earlier, it was difficult to extract complex handcrafted features from the given inputs. Now-a-days convolutional neural networks (CNN) are directly acting on the raw inputs, but these models are only restricted with the 2D inputs. In this paper, we are using a 3D-Convolutional Neural Network (3D-CNN) model to work with 3D inputs along with the sequential learning. This model extracts the features from temporal and spatial domains by using 3D convolutions which is able to take the motion information from the multiple frames. This model generates the multiple channels of information and by combining the information from all channels, the final feature representation is obtained. The updated weights obtained in the final model are then used to predict the labels.

Keywords: Human Action Recognition, deep models, 3D-CNN, label predictions.

1. Introduction

Understanding features of actions in human motion is a wide and important research area. This includes disciplines from computer science to psychology and brain science. The most challenging recognition problems in computer vision is the motion recognition and hence it finds applications in many fields like in video surveillance, video retrieval, shopping behavior analysis, customer attributes, etc.

Getting accurate recognition of actions is a highly challenging task due to jumbled backgrounds, obstructions in the inputs, and viewpoint variations. The most popularly used approaches [2] are recognizing action at a distance using persons' pixels and spatio-temporal feature detectors which make certain assumptions like small-scale changes or viewpoint changes under some circumstances from which video has taken, but these assumptions are difficult to be made with the real-world environments. Also these approaches follows pattern recognition models which give handcrafted features of actions from the video frames and learns classifiers from the obtained features.

Deep learning models are the machines that learns all features by automating process of feature construction and building low-level features and high level features. These models can be trained in both supervised and unsupervised learning approaches which gives better performance than the handcrafted features.

For automatic feature extraction, CNN's are primarily applied on 2D images using 2D-CNN models. The CNNs are able to recognize the actions from frames of the videos, but these approaches are only applied on the images of video which makes it difficult to achieve good performance for recognizing the actions. For this purpose, we are applying 3D-CNN on the videos directly to capture both the temporal and spatial features.

2. Previous Work

Most of the earlier work done is on the Hierarchical Max-Pooling Model(HMAX) model. Serre et. al. [7] developed the HMAX model for visual object recognition. In this model, the complex features are constructed by alternating applications of template matching and max pooling. The S1 layer of this model is given as an input image to analyze using Gabor filters at multiple orientations and scales. The C1 layer is then obtained by pooling local neighborhoods on the S1 maps, leading to increased invariance to distortions on the input. The S2 maps are obtained by comparing C1 maps with an array of templates that are generated from C1 maps in the training phase. The final feature representation in C2 is obtained by performing global max pooling of the S2 maps.

Jhuang et. al. [4] give the original HMAX model to analyze the 2D images. This model has been extended to recognize actions in the video data. Gabor filters in the S1 layer of this model gets replaced with gradient space-time modules to obtain the motion information.

Mutch and Lowe et. al. [5] give some modifications to the HMAX model. In this model, sparsity is increased by containing the number of feature inputs, lateral inhibition and feature selection. These modifications further improve classification performance, understanding of computational constraints facing biological and computer vision systems.

Bromley et. al. [8] developed time-delay neural networks to extract temporal features for speech and handwriting recognition. The input is given to a neural network to allow network to compress the information. It is more robust to give the network with low-level features and allow it to learn higher order features during the training process, and not making heuristic decisions.

Kim et. al. [3] have given a modified CNN model to extract the feature from the data. The CNN model gives set of features from the action descriptors which are achieved from a spatio-temporal volume by introducing a weighted fuzzy min-max (WFMM) neural network to reduce the dimensionality of the feature space.

Jain et. al. [6] also give a CNN model for 3D image restoration problems using which recognition tasks can be solved.

The summary of literature review is given in Table 1.

Table1. Summary of Literature Review

Author's Name	Year	Model	Description	Limitations
Bromley	1993	Developed Time-Delay Neural Network	This model is able to extract the features for speech and handwriting recognition.	This only applied on a little specific modules for recognition.
Serre	2005	HMAX model for object recognition	This model give the applications of template matching and max pooling using Gabor filters at multiple orientations and frames	It causes the problems of distortion and invariance on inputs.
Jhuang	2007	Extended HMAX model for 2D images	Able to recognize actions. Gabor filters get replaced with gradient space-time modules.	It only represents the dorsal views of the actions which are visual to eyes.
Kim	2007	Modified CNN model	This model gives set of features from action descriptors with the help of WFMM.	It causes the problem of distortion on input images.
Jain	2007	CNN model for 3D images	It is able to handle the problems of action recognition tasks.	It follows image restoration problems due to invariance.
Mutch and Lowe	2008	Modified HMAX model	This model is able to improve classification performance, understands facial biological and computer vision systems.	Only applied on object localization to get sparse features with limited receptive fields.

3. Convolutional Neural Network

Convolution layers are the building blocks used in the convolutional neural networks [9]. A convolution is the simple application of filter to an input resulting an activation. The convolutional neural networks are able to automatically learn a large number of filters in parallel specific to a training dataset under the constraints of a specific predictive modeling problem, such as image classification. The result is highly specific features that can be detected on input images.

The images of video are divided into regions, and each region is then assigned to different hidden nodes. Each hidden node finds pattern in only one of the regions in the image. This region is determined by a kernel (also called a filter/window). A filter is convolved over both x-axis and y-axis. Multiple filters are used in order to extract different patterns from the image. The output of one filter when convolved throughout the entire image generates a 2-d layer of neurons called a feature map. Each filter is responsible for one features map.

These feature maps can be stacked into an array, which can then be used as the input to the layers. This is performed by the layer called Convolutional layer in CNN. These layers are followed by pooling layers, which reduce the spatial dimensions of the output. After that the window is slid in both the axes and the max value in that filter is taken which is the functionality of max pooling layer. Sometimes Average pooling layer is also used where the only difference is to take the average value within the window instead of the maximum value. Therefore, the convolutional layers increase the depth of the input image, whereas the pooling layers decreases the spatial dimensions.

3.1 2D-CNN

The 2D-CNN generally applied on 2D input images. It is 2-dimensional because the filter is convolved along the x-axis and y-axis of the image. The architecture of 2D-CNN can be shown in Figure 1.

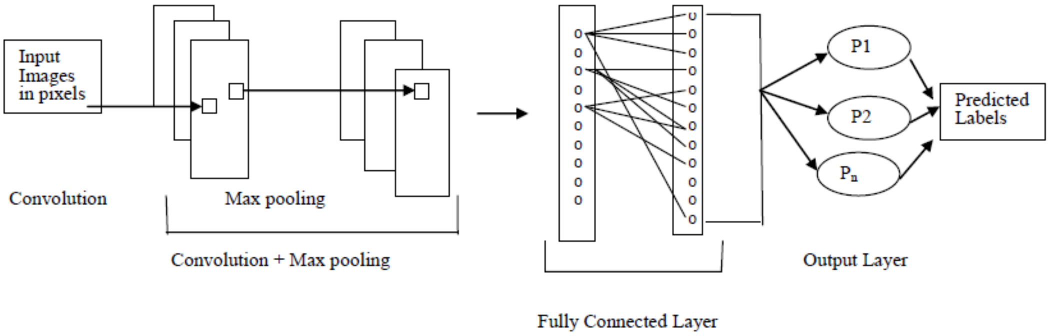
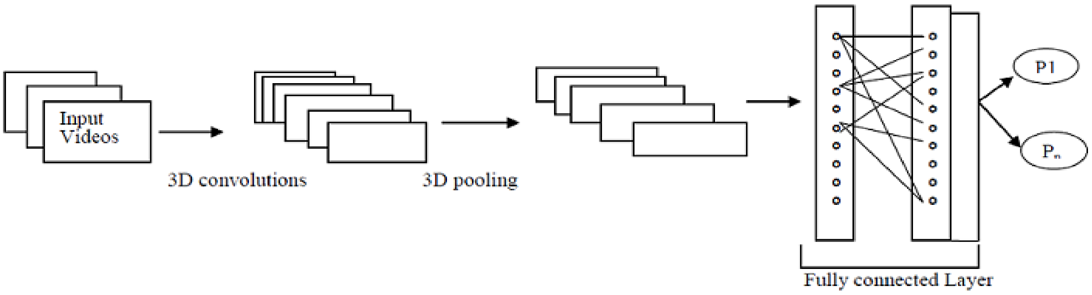


Figure 1. System Overview of 2D-CNN Model

3.2 3D CNN Approach

In the case of videos, we have an additional temporal axis – z-axis. So, a 3-D convolutional layer is used where the filter is 3-dimensional and can convolved across all the three axes. Multiple convolutional and pooling layers are stacked together. These are followed by some fully-connected layers, where the last layer is the output layer. The system overview of 3D-CNN model can be shown in figure 2.



Where P1, P_n are predicted labels of the output layer.

Figure 2. System Overview of 3D-CNN Model

4. Dataset

The dataset that is used is the Kth[10] dataset which is publicly available. The video database contains six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed several times by 25 subjects in four different scenarios: outdoors s1, outdoors with scale variation s2, outdoors with different clothes s3 and indoors s4 as illustrated below in fig 3. All sequences were taken over homogeneous backgrounds with a static camera with 25fps frame rate. The sequences were downsampled to the spatial resolution of 160x120 pixels and have a length of four seconds in average.

There is a total of 6 categories - boxing, handclapping, handwaving, jogging, running and walking. While loading the data, we convert these text labels into integers according to the following mapping:
(boxing, 0), (handwaving, 1), (handclapping, 2), (jogging, 3), (running, 4), (walking, 5)

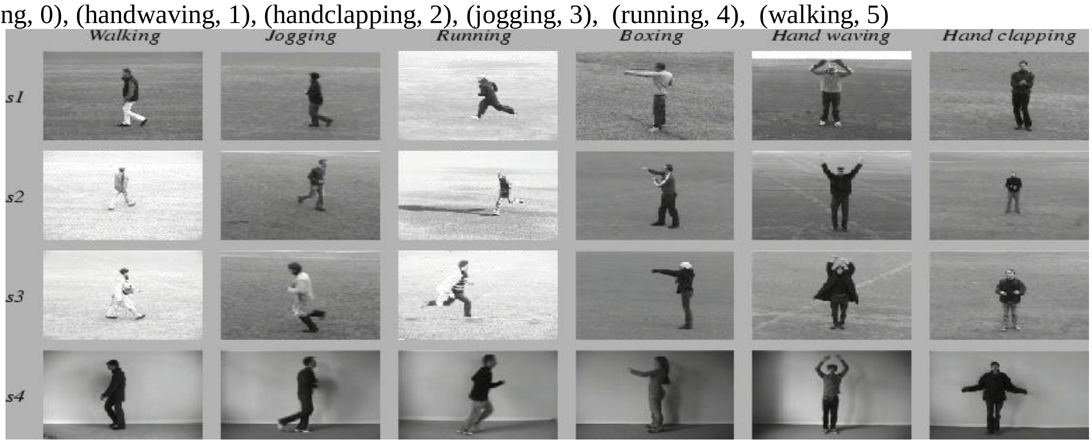


Figure 3. Different four Scenarios of KTH Dataset with classes

5. Methodology

The overall procedure of the method is shown in fig. 4.

5.1 Data Preprocessing

The videos were captured at a frame rate of 25fps. This means that for each second of the video, there will be 25 frames. We know that within a second, a human body does not perform very significant movement. This implies that most of the frames in our video will be redundant. Therefore, only a subset of all the frames in a video needs to be extracted. This will also reduce the size of the input data which will in turn help the model train faster and can also prevent over-fitting.

Different strategies would be used for frame extraction like:

- Extracting a fixed number of frames from the total frames in the video – say only the first 200 frames.
- Extracting a fixed number of frames each second from the video – say we need only 5 frames per second from a video whose duration is of 10 seconds. This would return a total of 50 frames from the video. This approach is better in the sense that we are extracting the frames sparsely and uniformly from the entire video.

Each frame needs to have the same spatial dimensions (height and width). Hence each frame in a video will have to be resized to the required size. In order to simplify the computations, the frames are converted to grayscale.

Normalization – The pixel values ranges from 0 to 255. These values would have to be normalized in order to help our model converge faster and get a better performance. Different normalization techniques can be applied such as:

- Min-max Normalization – Get the values of the pixels in a given range (say 0 to 1)
- Z-score Normalization – This basically determines the number of standard deviations from the mean a data point.

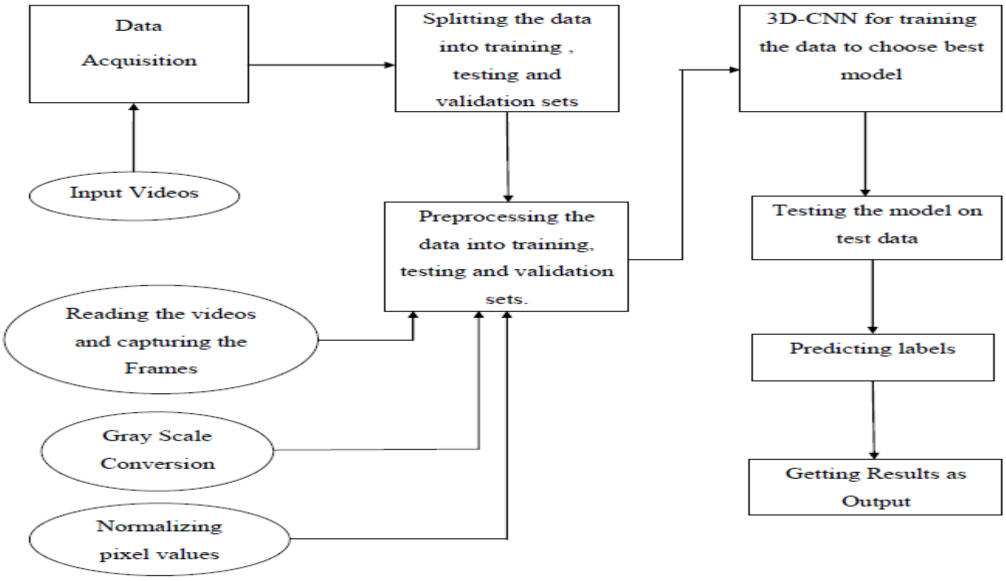

Data Acquisition
Input Videos
Splitting the data into training , testing and validation sets
Preprocessing the data into training, testing and validation sets.
Reading the videos and capturing the Frames
Gray Scale Conversion
Normalizing pixel values
3D-CNN for training the data to choose best model
Testing the model on test data
Predicting labels
Getting Results as Output

Figure 4. Block Diagram of overall Procedure.

5.2 Implementation

5.2.1 Model Parameters

For each convolutional layer, we have to configure the following parameters:

- **Filters:** This is the number of feature maps required as the output of that convolutional layer.
- **Kernel_size:** The size of the window that will get convolved along all the axes of the input data to produce a single feature map.
- **Strides:** The number of pixels by which the convolutional window should shift by.
- **Padding:** To decide what happens on the edges - either the input gets cropped or the input is padded with zeros to maintain the same dimensionality.

- **Activation**: The activation function to be used for that layer. ReLU and Softmax are proven to work best with deep neural networks because of their non-linearity and their property of avoiding the vanishing gradient problem.

  a) ReLU

  The equation of ReLU function is

  $$F(x) = \max(0, x)$$

  Where, x is the maximum feature selector.

  b) Softmax Function

  The equation of softmax function is

  $$S(y_j) = \frac{e^{y_j}}{\sum_j e^{y_j}}$$

For each pooling layer, we have to configure the following parameters:

- **Pool_size:** The size of the window.
- **Strides:** The number of pixels by which the pooling window should shift by.
- **Padding:** To decide what happens on the edges - either the input gets cropped (valid) or the input is padded with zeros to maintain the same dimensionality (same).

We perform experiment on KTH data to evaluate the implemented 3D CNN model for action recognition. The implementation requires keras, numpy, scikit-learn, scikit-video, PIL etc. libraries to define the Convolution layer (3D), Max Pooling layer (3D) and Global Average Pooling layer(3D). The model uses updated weights which are useful for label predictions.

6. Results

The 3D-CNN model is getting trained in 40 epochs to achieve overall accuracy of 64.5% on video processing for the action recognition. This is shown using the learning curve of 3D-CNN model as in figure 5. The model is validated by training it through randomly selecting 9 persons from each class which is giving an accuracy of approximately 69%.

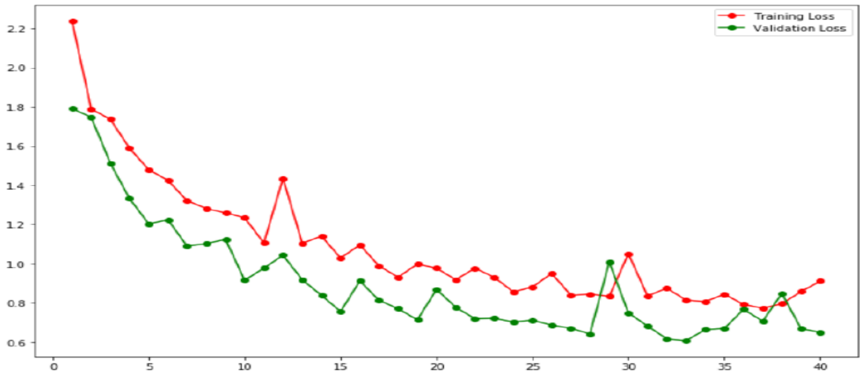


Figure 5. Learning Curve of 3D-CNN Model.

The predicted labels along with their accuracies are tabulated as follows. The average accuracy of the labels is 80.2%.

Table 2. Predicted label with accuracies

Labels	Class Names	Number of Videos	Predicted Accuracies
0	Boxing	98	84%
1	Handwaving	98	100%
2	Handclapping	100	72%
3	Jogging	100	69%
4	Running	100	64%
5	Walking	100	94%

This gives the overall average accuracy of 80.33% which is far greater than svm [11] approach which has an average of 71.83%. This shows that the labels with 3D-CNN model gets predicted with higher performance.

7. Conclusion

The 3D-CNN model construct features from both spatial and temporal dimensions by performing 3D convolutions. The deep architecture generates multiple channels of information from adjacent input frames and perform convolution and sub sampling separately in each channel. The final feature representation is computed by combining information from all channels. We evaluated the 3D CNN model using the KTH data sets which achieves the better performances. The supervised learning approach is used in this implementation for label prediction. One can apply the same method using unsupervised learning approach.

8. References

[1]. Shuiwang Ji,Wei, Ming Yang, Kai Yu ,“3D Convolutional Neural Networks for Human Action Recognition” , IEEE transaction 2013.

[2]. Efros, A. A., Berg, A. C., Mori, G., and Malik, J. "Recognizing action at a distance", ICCV, 733, 2003.

[3]. Kim, H.-J., Lee, J. S., and Yang, H.-S. "Human action recognition using a modified convolutional neural network", in Proceedings of the 4th International Symposium on Neural Networks, 2007.

[4]. Jhuang, H., Serre, T., Wolf, L., and Poggio, T. "A biologically inspired system for action recognition", in ICCV, 2007.

[5]. Mutch, J. and Lowe, D. G. "Object class recognition and localization using sparse features with limited receptive fields", International Journal of Computer Vision, October 2008.

[6]. Jain, V., Murray, J. F., Roth, F., Turaga, S., Zhigulin, V., Briggman, K. L., Helmstaedter, M. N., Denk, W., and Seung, H. S. "Supervised learning of image restoration with convolutional networks", in ICCV, 2007.

[7]. Serre, T., Wolf, L., and Poggio, T. "Object recognition with features inspired by visual cortex", in CVPR, 2005.

[8]. Bromley, J., Guyon, I., LeCun, Y., Sackinger, E., and Shah, R. "Signature verification using a siamese time delay neural network", in NIPS, 1993.

[9]. Convolutional Neural Network (CNN) Available: https:// search enterpriseai. techtarget.com/definition/convolutional-neural-network.

[10]. The KTH Dataset [Online] Available: http://www.nada.kth.se/cvap/actions/.

[11]. Christian Schuldt, Ivan Laptev and Barbara Caputo, "Recognizing Human Actions: A Local SVM Approach", in Proc. ICPR'04, Cambridge, UK.