applied sciences

MDPI

Article

Human Activity Classification Using the 3DCNN Architecture

Roberta Vrskova *, Robert Hudec, Patrik Kamencay * and Peter Sykora

Department of Multimedia and Information-Communication Technologies, University of Zilina, 010 26 Zilina, Slovakia; robert.hudec@uniza.sk (R.H.); peter.sykora@uniza.sk (P.S.)
* Correspondence: roberta.vrskova@uniza.sk (R.V.); patrik.kamencay@uniza.sk (P.K.)

Abstract: Interest in utilizing neural networks in a variety of scientific and academic studies and in industrial applications is increasing. In addition to the growing interest in neural networks, there is also a rising interest in video classification. Object detection from an image is used as a tool for various applications and is the basis for video classification. Identifying objects in videos is more difficult than for single images, as the information in videos has a time continuity constraint. Common neural networks such as ConvLSTM (Convolutional Long Short-Term Memory) and 3DCNN (3D Convolutional Neural Network), as well as many others, have been used to detect objects from video. Here, we propose a 3DCNN for the detection of human activity from video data. The experimental results show that the optimized proposed 3DCNN provides better results than neural network architectures for motion, static and hybrid features. The proposed 3DCNN obtains the highest recognition precision of the methods considered, 87.4%. In contrast, the neural network architectures for motion, static and hybrid features achieve precisions of 65.4%, 63.1% and 71.2%, respectively. We also compare results with previous research. Previous 3DCNN architecture on database UCF Youtube Action worked worse than the architecture we proposed in this article, where the achieved result was 29%. The experimental results on the UCF YouTube Action dataset demonstrate the effectiveness of the proposed 3DCNN for recognition of human activity. For a more complex comparison of the proposed neural network, the modified UCF101 dataset, full UCF50 dataset and full UCF101 dataset were compared. An overall precision of 82.7% using modified UCF101 dataset was obtained. On the other hand, the precision using full UCF50 dataset and full UCF101 dataset was 80.6% and 78.5%, respectively.

Keywords: 3DCNN; neural network; recognition; classification; video; UCF YouTube Action dataset; UCF101 dataset

check for updates

Citation: Vrskova, R.; Hudec, R.; Kamencay, P.; Sykora, P. Human Activity Classification Using the 3DCNN Architecture. *Appl. Sci.* **2022**, *12*, 931. https://doi.org/10.3390/app12020931

Academic Editor: Antonio Fernández-Caballero

Received: 3 December 2021
Accepted: 10 January 2022
Published: 17 January 2022

1. Introduction

The field of computer vision is mentioned often in the modern literature because of the success that neural networks are now achieving. In this paper, we address a neural network architecture called a 3D convolutional neural network (3DCNN). This architecture is often used to detect complex images as well as videos, such as medical images. Here we use the 3DCNN architecture for video classification. Video classification has been identified as a major challenge because of the type of information contained in video, namely, time continuity information. We not only need to look at the simple 2D space of an image but must also take into account the previous and following images to process information about time. When classifying a video, researchers have to deal with challenges such as the classification of abnormal behavior of people in public spaces, crowded scenes with a variety of abnormalities, and the classification of various human activities. Such research can be very helpful for applications such as improving safety or by protecting the elderly by identifying fall accidents and subsequently calling for help. Further, these techniques can be used to detect terrorists and to prevent a variety of dangerous events.

The problem of classification is addressed in several papers, such as [1], where they aim to classify a person attacking an ATM from depth camera video. Weapons detection using neural networks for video processing has been studied, as in [2]. The recognition of human activity for the purpose of identifying if a monitored elderly person has fallen or is experiencing a health problem is addressed in [3]. With regard to the mentioned issues in the relevant literature [4], they represent various techniques for detecting the activities of people from videos designed up to 2019. Additionally, Ref. [5] studies if monitoring videos and the content of videos can be used to detect whether a video is suitable for minors. A novel approach for generic visual vocabulary learning is proposed. Such methods require databases of various human activities. An equally interesting approach is presented in [6], where a systematic framework for recognizing realistic actions from videos "in the wild" is used. The topic of neural networks is also of interest in a variety of industries not only in terms of its application to video classification but also its ability to address problems related to sound, for which time continuity is also a concern [7]. 3D convolutional neural network (3DCNN) is a powerful and effective model utilizing spatial-temporal features, which is why they also use it in the article [8] for gesture recognition. In paper [9], proposed a novel deep learning architecture for efficient automatic hand sign language recognition using also 3D Convolutional Neural Network (3DCNN) from RGB input videos. The combination of 3DCNN and ConvLSTM is proposed in [10] and used on human action recognition. In [11] , is analysed the performance of a 3DCNN architecture for hand gesture recognition as in the article [8], however in an unstructured scenario. The major contribution [12] is proposed a novel 3DCNN powered model for scene classification in drone surveillance. In paper [13] proposed deep features of mobile videos are extracted by an exponential linear units-3D convolutional neural network for representing video. In work [14] is studied the ability of state of the art video CNNs including 3D ResNet, 3D ResNet, and I3D for detecting manipulated videos. In [15] presents models based on Convolutional Neural Network (CNN) for problem of classifying videos under the classses as violence and non-violence. Authors in paper used RLVS dataset. Authors in article [16] proposed graph-based framework to learn high-level interaction between object and people, which called Action Detection. In [17] proposed metods to detect violence using Mobil Neural Architecture Authors used Convolutional Long Short-term Memory (ConvLSTM) to extract spatiotemporal feaures in the video. They developed dataset that contains violence and non-violence. Authors in article [18] proposed metods for anomaly detection in crowd scenes. They proposed 3DCNN architecture but also 3D GAN for domain adaption to reduce domain gap. In article [19] authors proposed Kinematics Posture Feature extraction from 3D joint positions. They used for classification Support Vector Machine (SVM) and Convolutional Reccurent Neural Network (CRNN). Authors use dataset in classification, which contains ariel action [20]. In article [21] used 3DCNN on vehicle behavior recognition.

We have also addressed this issue in our previous research [22,23]. In the article [22] we compared the results of ConvLSTM and 3DCNN on which architecture we then progressed in the design of another 3DCNN architecture. In the article [23] we took a closer look at the ConvLSTM architecture, which we tested on the UCF crime database. In this paper, we classify video sequences containing different categories of human activities using the proposed 3DCNN architecture.The ability to classify a diversity of normal human activities can move us further toward the ability to classify abnormal activities. In this way, we can increase the security of public spaces or simplify a person's day.

This paper is divided into the following sections. In the Section 1, the 3DCNN architecture is described. The tested dataset and achieved experimental results are described in the Section 2. Finally, in the Section 4, the conclusions and a discussion are provided.

2. Materials and Methods

Currently, there are a number of neural network approaches and architectures for video classification. Very often, LSTM (Long Short-Term Memory) or ConvLSTM are used. In our research, we focus on the 3DCNN architecture, which is usually used for detection in medical images. In some next approaches, this architecture is also used for classification or prediction from video data.

2.1. 3DCNN Architecture

In recent years, the neural network architecture based on 3D convolution layers is a very frequently used approach in the classification of video data. Because the 3DCNN architecture is able to analyze the positions of objects in time, it is often used for moving 3D images, especially medical images. The 3DCNN creates a 3D activation map during the convolution step, which is needed not only for data analysis but also for time and volumetric context. A three-dimensional filter is used for the 3D convolution of the dataset to calculate the representation of elements at a low level. The kernel moves in three directions (x, y, z), as shown in Figure 1. The value at each position of the feature map in the layer is given by the following equation:

$$v_{ij}^{xyz} = \tanh\langle b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{\langle i-1\rangle m}^{\langle x+p\rangle\langle y+q\rangle\langle z+r\rangle}, \tag{1}$$

where w_{ijm}^{pqr} is the value of the kernel connected to the feature map in the previous layer and R^i is the size of the 3D kernel [24].

The output shape is a three-dimensional volume space. We achieve 3D convolution by winding around the center of a cube and stacking adjacent layers on top of each other. Functional maps are interconnected to capture motion information. However, the convolution kernel can extract only one type of element. The overall network is similar to a 2D convolutional neural network. In general, as with 2D convolution, we can achieve better results by combining several convolution layers. When creating the 3DCNN, our results depend on the number of layers as well as the number of filters in each layer and the size of the filters. If pooling is used when creating the neural network, the pooling size must be formed by three values because we are working with 3D data [25].

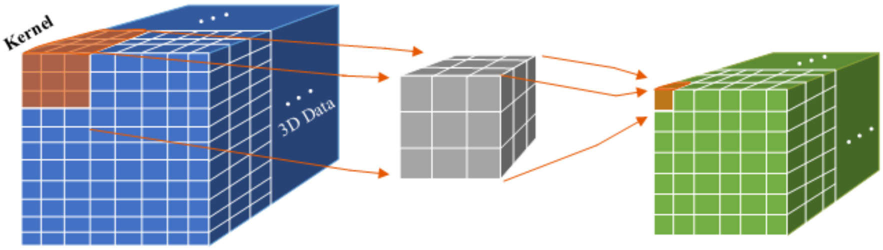


Figure 1. Convolution operation in the 3DCNN architecture [26].

The three-dimensional MaxPooling3D layer is a form of nonlinear downsampling of an input tensor. This method partitions the input tensor data into 3D subtensors along three dimensions and selects the element of each subtensor with the maximal numeric value. Finally, it transforms the input tensor to the output tensor by replacing each subtensor with its maximum element. MaxPooling3D is often applied to color image and is shown in Figure 2 [26].

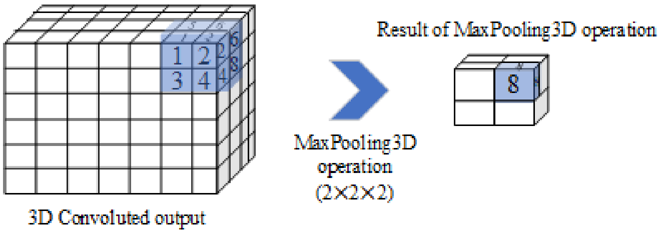


Figure 2. MaxPooling3D operation in the 3DCNN architecture [26].

2.2. Proposed 3DCNN Architecture

The 3DCNN-based methods using 3D convolution layers have become very popular in recent years for extraction of features from input video data [6]. For this reason, the three-dimensional (3D) convolution layer in our proposed 3DCNN was used. The 3D convolution layer is based on the principle of a sliding 3D convolution window along input data. The 3D convolution window is above the data and identifies several filters (each filter detects a different pattern). These 3D filters in all three directions are moved.

The architecture of the proposed 3DCNN can be seen in Figure 3. This architecture is divided into following layers:

- 3DCNN layers, which improve the identification of 3D and moving images . Each layer contains a three-dimensional filter that moves in three directions (x, y, z). During the 3D convolution, a convolutional map is created, which is needed for data analysis as well as time and volumetric context.
- MaxPooling layers for 3D data (MaxPooling3D), which are used to reduce the size of the image data. MaxPooling3D is a mathematical operation for 3D data as well as for spatial or spatiotemporal data. The layers are defined using $n \times n \times n$ regions as corresponding filters for the max pooling operations. Additionally, a stride is defined, which sets the number of pixels that the filter moves in each step as it slides across the image.
- Batch normalization is structure which is used to normalize the previous layer for each batch. Batch normalization transform standard deviation to 1 and mean activation to 0.
- A dense layer, which is usually one of the last layers and mainly fully interconnected neurons.
- A flatten layer, which is located at the end of the neural network and causes the matrix to be converted to an output vector.

The architecture consists mainly of 3D convolution layers. However, it also contains layers that are found in every architecture as flattened and dense and are an integral part. We have already dealt with a 3D convolutional network in previous research [22], which led us to improve the existing network and achieve better results. The architecture in the previous research achieved lower results, which could be due to the unnecessarily large number of layers. Therefore, we have decided to reduce the number of layers in this proposal. As we can see in the results, we confirmed our assumptions and we really got better results. Hyperparameters such as the number of filters and the core size of 3D convolution layers and MaxPooling are bound by mathematical operations, where the output from the layer must not be negative and mus be integer output. However, in previous research [27], we proposed an optimization algorithm for ConvLSTM, which we slightly modified and applied to this network to achieve the best possible results.

The architecture of the proposed 3DCNN implemented in this work consists of six 3D convolutional layers and four MaxPooling3D layers. The number of filters and their size change with each new layer. The input to the neural network are images of size 32 × 32 × 3 (width, height, number of channels). The first 3D convolution layer has 64 filters with kernel of size 3 × 3 × 3. The input filter size and number of filters were determined by default based on previous research [22]. Behind the first 3D convolution layer, there is a MaxPooling3D layer of size 2 × 2 × 2 and stride 2. MaxPooling reduces the size of the data. The second 3D convolution layer contains the same number of filters, 64, and the same size filters 3 × 3 × 3 as the first 3D convolution layer. Behind the second 3D convolution layer, there is a MaxPooling3D with the same size 2 × 2 × 2 as the previous Maxpooling3D layer. Follows third and fourth 3D convolution layers have 128 filters of size 3 × 3 × 3. After the fourth 3D convolution layer, there is a MaxPooling3D of the same size 2 × 2 × 2 as the previous MaxPooling3D layers. Finally, the last two 3D convolution layers are used, where one has 256 filters of size 6 × 1 × 1 and the other has 512 filters of size 1 × 1 × 1. After the layers with 256 and 512 filters, there is a final MaxPooling3D layer of size 2 × 2 × 2. Behind every MaxPooling3D layer is a batch normalization. At the end of the network, there is a dense and a flattened layer. The dense layer is only one and takes the value 11 directly. The filter sizes were chosen to account for the output from the filters, where it was necessary to follow the integer output. The optimization algorithm "Adam" was chosen in the architecture and the learning rate was default value 0.1. The results for several values of batch size and number of epochs are shown in Figure 4.

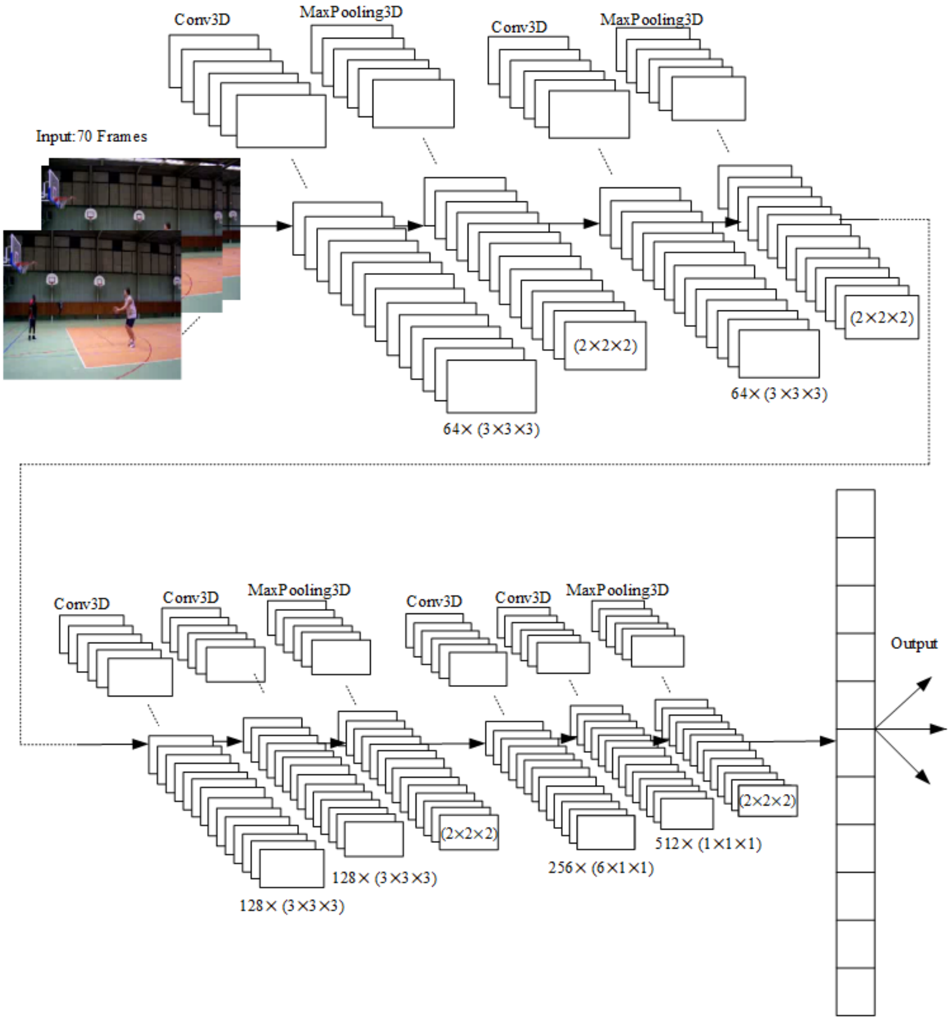


Figure 3. Proposed 3DCNN architecture.

The flowchart of the proposed 3DCNN neural network architecture is shown in Figure 5. The input shape (pre-processing of input data) represents the length of the video sequence, the width and height of the image, and the number of channels. The output from one layer is then the input of the next layer and thus the information passes through all 3DCNN layers. The 3DCNN layers are followed by the layers as dense and flatten. Finally, we get the output of our proposed model. This output is represent by the final accuracy, loss function and confusion matrix. A more detailed description of the layers of the proposed 3DCNN architecture is shown in Figure 6. The choice of hyperparameters affects the overall results. For example, in [7] authors show that the accuracy varies from 32.2% to 92.6% depending on the selected hyperparameter values. Hyperparameters such as filter size, size of the MaxPooling filter, or the number of filters, are set default values in the design. Hyperparameters such as the number of epochs and the batch size are selected based on a value testing procedure. To select these values, it is necessary to test several possible combinations on the architecture of the neural network. We performed tests to obtain the combination of values that gives the best results. From the graph, we can see that the best results were obtained with the number of epochs set to 8 and the batch size (number of samples that will be propagated through the network) set to 35. The graph suggests a possible improvement in results at higher batch sizes and numbers of epochs.

A higher number of epochs was not chosen because the neural network could be overtrained. In general, batch size of 32 is a good values. Other values may be fine for some data sets, but the given range is generally the best to experimenting with database of videos. For optimize the hyperparameters the optimization algorithm was designed [27]. This optimization algorithm in [27] is described in more detail. As you can see from the example of proposed 3DCNN architecture, the pseudo-code is written in Python programming language (Algorithm 1).

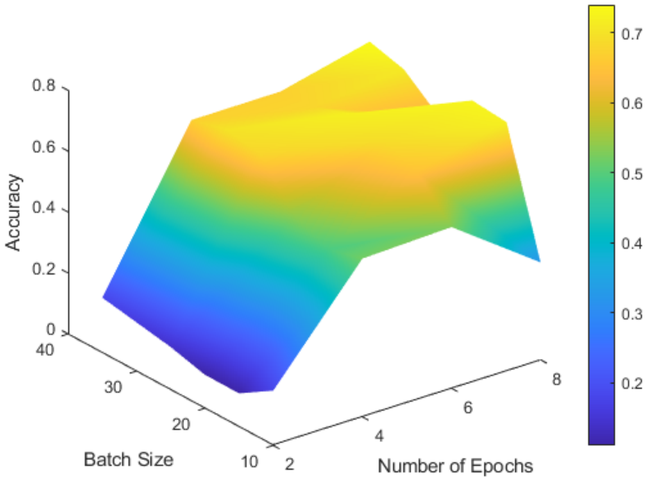


Figure 4. Surface graph of dependencies of the average accuracy of batch size and number of epochs.

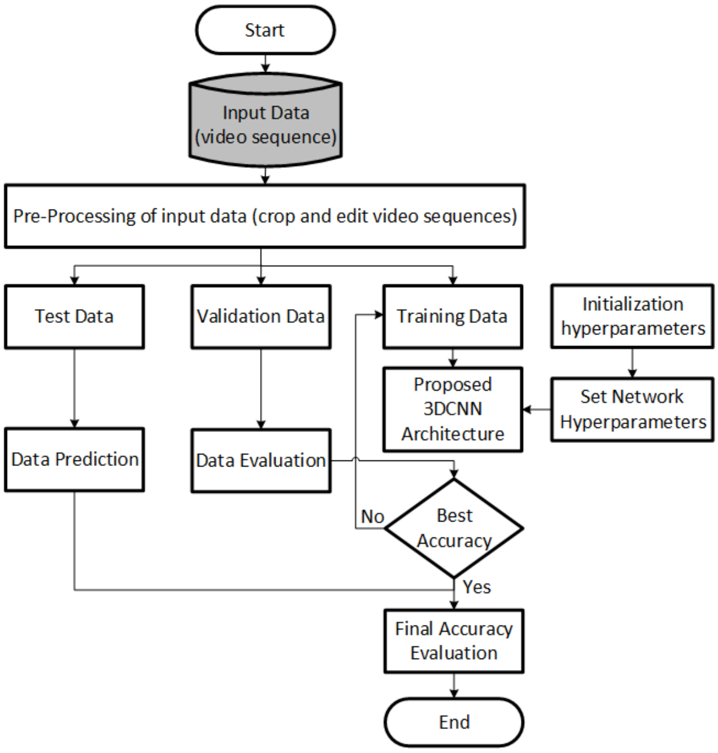
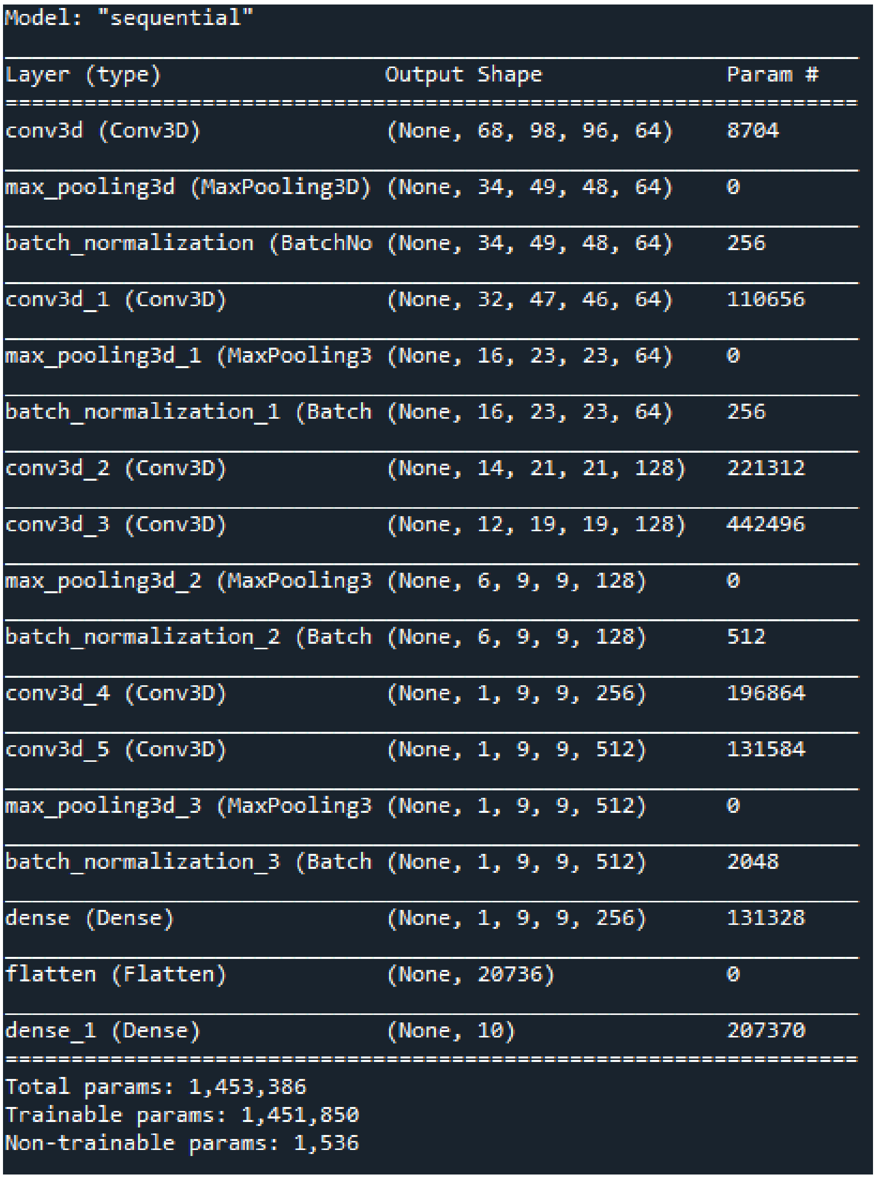

Start
Input Data (video sequence)
Pre-Processing of input data (crop and edit video sequences)
Test Data
Validation Data
Training Data
Initialization hyperparameters
Proposed 3DCNN Architecture
Set Network Hyperparameters
Data Prediction
Data Evaluation
No
Best Accuracy
Yes
Final Accuracy Evaluation
End

Figure 5. The flowchart of the proposed 3DCNN.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv3d (Conv3D)              (None, 68, 98, 96, 64)    8704
_________________________________________________________________
max_pooling3d (MaxPooling3D) (None, 34, 49, 48, 64)    0
_________________________________________________________________
batch_normalization (BatchNo (None, 34, 49, 48, 64)    256
_________________________________________________________________
conv3d_1 (Conv3D)            (None, 32, 47, 46, 64)    110656
_________________________________________________________________
max_pooling3d_1 (MaxPooling3 (None, 16, 23, 23, 64)    0
_________________________________________________________________
batch_normalization_1 (Batch (None, 16, 23, 23, 64)    256
_________________________________________________________________
conv3d_2 (Conv3D)            (None, 14, 21, 21, 128)   221312
_________________________________________________________________
conv3d_3 (Conv3D)            (None, 12, 19, 19, 128)   442496
_________________________________________________________________
max_pooling3d_2 (MaxPooling3 (None, 6, 9, 9, 128)      0
_________________________________________________________________
batch_normalization_2 (Batch (None, 6, 9, 9, 128)      512
_________________________________________________________________
conv3d_4 (Conv3D)            (None, 1, 9, 9, 256)      196864
_________________________________________________________________
conv3d_5 (Conv3D)            (None, 1, 9, 9, 512)      131584
_________________________________________________________________
max_pooling3d_3 (MaxPooling3 (None, 1, 9, 9, 512)      0
_________________________________________________________________
batch_normalization_3 (Batch (None, 1, 9, 9, 512)      2048
_________________________________________________________________
dense (Dense)                (None, 1, 9, 9, 256)      131328
_________________________________________________________________
flatten (Flatten)            (None, 20736)             0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                207370
=================================================================
Total params: 1,453,386
Trainable params: 1,451,850
Non-trainable params: 1,536
```

Figure 6. The description of the layers of the proposed 3DCNN.

Our proposed model begins initialization parameters. We continue by loading videos, then editing and crop them. We will divide the videos into training, testing and validation data. We set the data prepared in this way for input. We continue model training to training data. Further validate on data validation and testing on testing data. After training, validate and testing, we calculate the results. We calculate the resulting accuracy, loss function and confusion matrix. At the end of the rendering training results so that we can observe the course of accuracy and loss function during training.

The hardware architectures is very necessary for efficient model training. The deep learning systems consists of high-level interface libraries, such as Keras or TensorFlow. These libraries perform computationally intensive operations. From this reason, the Nvidia CUDA libraries were used. In our case is the Nvidia of choice for deep learning models. It's because of greater level of software support for deep learning specific computations. The all experimental results on an NVIDIA GeForce GTX 1660 Ti graphics card were performed. The proposed neural network architecture in a Python environment using Keras and TensorFlow frameworks was coded.

Algorithm 1 Pseudo-code of the proposed 3DCNN architecture.

```
**procedure** 3DCNN($inputDir$)
    $X = []$
    $Y = []$
    classesList = os.listdirinputDir
    $i = 0$
    **for** $c$ in classesList **do**
        $Ytemp = np.zeros(shape = (numClasses))$
        $Ytemp[i] = 1;$
        $print(c)$
        $files_{l}ist = os.listdir(os.path.join(inputDir, c))$
        **for** $f$ in filesList **do**
            $frames = framesExtraction(os.path.join(os.path.join(inputDir, c), f))$
        **end for**
    **end for**
    $X = np.asarray(X)$
    $Y = np.asarray(Y)$
    return $X, Y$
    $X, Y = createData(dataFolder)$
    $xTrain, xTest, yTrain, yTest = trainTestSplit(X, Y)$
    $seq.add(Conv3D(filters = 64, kernelSize = (3, 3, 3), activation =' relu')$
    $seq.add(MaxPooling3D(poolSize = (2, 2, 2)))$
    $seq.add(BatchNormalization())$
    $seq.add(Conv3D(filters = 128, kernelSize = (3, 3, 3), activation =' relu')$
    $seq.add(MaxPooling3D(poolSize = (2, 2, 2)))$
    $seq.add(BatchNormalization())$
    $seq.add(Conv3D(filters = 256, kernelSize = (6, 1, 1), activation =' relu')$
    $seq.add(Conv3D(512, kernelSize = (1, 1, 1), activation =' relu')))$
    $seq.add(MaxPooling3D(poolSize = (1, 1, 1)))$
    $seq.add(BatchNormalization())$
    $seq.add(Dense(256, activation =' relu')$
    $seq.add(Flatten())$
    $seq.add(Dense(numClasses, activation =' softmax'))$
    $seq.compile('adam', loss =' categorical_{c}rossentropy', metrics = ['accuracy'])$
**end procedure**
```

3. Experimental Results

In this section, we will present the obtained experimental results. Firstly, we will describe the used dataset for training and testing purposes. Next, we created a model that generates a confusion matrix. Finally, we will provide the experimental results for classification of human activity using proposed 3DCNN Architecture.

3.1. UCF YouTube Action Dataset

The UCF YouTube Action dataset by the Center for Research in Computer Vision at the University of Central Florida was created. This dataset is composed of videos categorized into 11 classes of human activities: Basketball, Biking, Diving, GolfSwing, HorseRiding, SoccerJuggling, Swing, TennisSwing, TrampolineJumping, VolleyballSpiking, and WalkingDog. The example of this dataset is shown in Figure 7.

The dataset contains 1160 video sequences downloaded from YouTube. Each class consists of 25 different groups of videos containing concrete human activity. Each of these groups contains at least four videos. The grouped video clips contain some similar characteristics, such as the same actors, similar backgrounds, similar variations in camera motion, etc.

All videos are in mpeg4 format. The video size is 320 × 240 and the frame rate is 29.97 fps. The complete dataset is demanding due to large differences in camera movement, appearance and pose of the object, angle of view, crowded backgrounds and lighting conditions. The dataset is one of the largest datasets in the vision community [5].

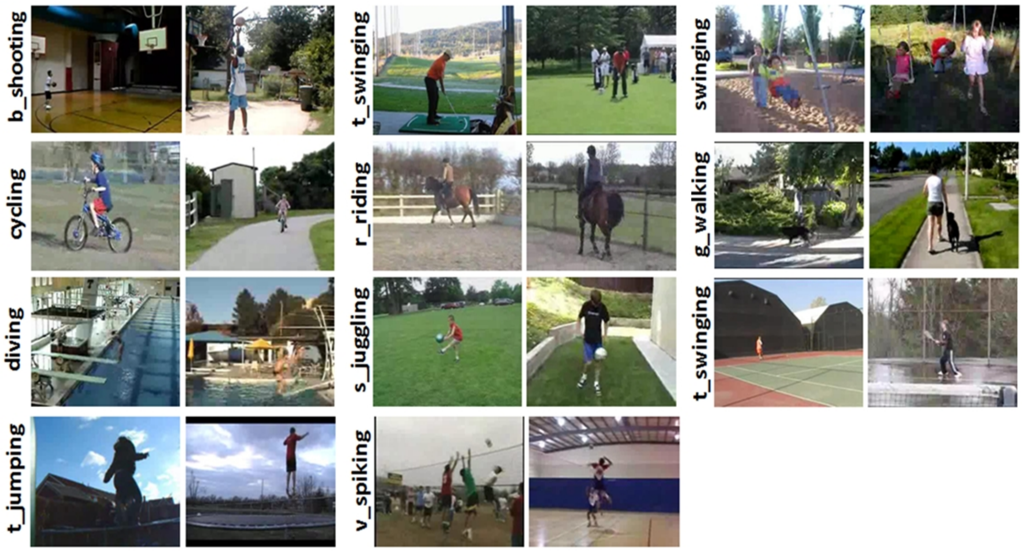


Figure 7. Sample images from the UCF YouTube Action Dataset [5,6].

In [5], the authors used different architectures for motion, static and hybrid features and achieved accuracies of 65.4%, 63.1% and 71.2%, respectively. In [6], another approach for computer vision on this dataset was proposed and achieved an average accuracy of 76.1%.

3.2. UCF101 Dataset

This dataset contains 101 action categories (13,320 videos), which consisting of realistic videos from youtube. The action categories of this dataset are divided into five types (Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments, Sports). The UCF101 dataset is an extension of UCF50 dataset which has 50 categories [28]. The example of the UCF101 dataset is shown in Figure 8. This dataset contains 101 categories collected from youtube as Apply Eye Makeup, Apply Lipstick, Archery, Baby Crawling, Balance Beam, Band Marching, Baseball Pitch, Basketball Shooting, Basketball Dunk, Bench Press, Biking, Billiards Shot, Blow Dry Hair, Blowing Candles, Body Weight Squats, Bowling, Boxing Punching Bag, Boxing Speed Bag, Breaststroke, Brushing Teeth, Clean and Jerk, Cliff Diving, Cricket Bowling, Cricket Shot, Cutting In Kitchen, Diving, Drumming, Fencing, Field Hockey Penalty, Floor Gymnastics, Frisbee Catch, Front Crawl, Golf Swing, Haircut, Hammer Throw, Hammering, Handstand Pushups, Handstand Walking, Head Massage, High Jump, Horse Race, Horse Riding, Hula Hoop, Ice Dancing, Javelin Throw, Juggling Balls, Jump Rope, Jumping Jack, Kayaking, Knitting, Long Jump and so on [28].

3.3. Results

The dataset was divided into three parts: A training set, a testing set and a validation set (70:20:10). The training set serves to train the neural network. The training set contains 812 videos, which represents 70% of the datasets. The test set to objectively evaluate the fit model was used. The testing set contains 232 videos, which represents 20% of the datasets. The validation set for objective evaluation of the model and for tuning of hyperparameters was used. The validation set contains 116 videos, which represents 10% of the datasets. The ratio of testing, training and validation videos has been set to exact. A testing videos were not taken into account when a training the data. In this paper, we classify the dataset (videos) into the following 11 classes (Basketball, Biking, Diving, GolfSwing, HorseRiding, SoccerJuggling, Swing, TennisSwing, TrampolineJumping, VolleyballSpiking, and WalkingDog). The data must be pre-processed before being input to the neural network. Firstly, the each video is cut into 320 × 240 frames. Each frame is saved to a field containing other frames from the given category. The partial accuracy during training process increases with the minimization of the loss function. A positive increase in accuracy with each new epoch during training can be seen in Figure 9. In the early epochs, we see a rapid increase in accuracy. At a value of approximately 0.8 the change in accuracy slows but continues to increase. The highest accuracy reached during training is 95%.

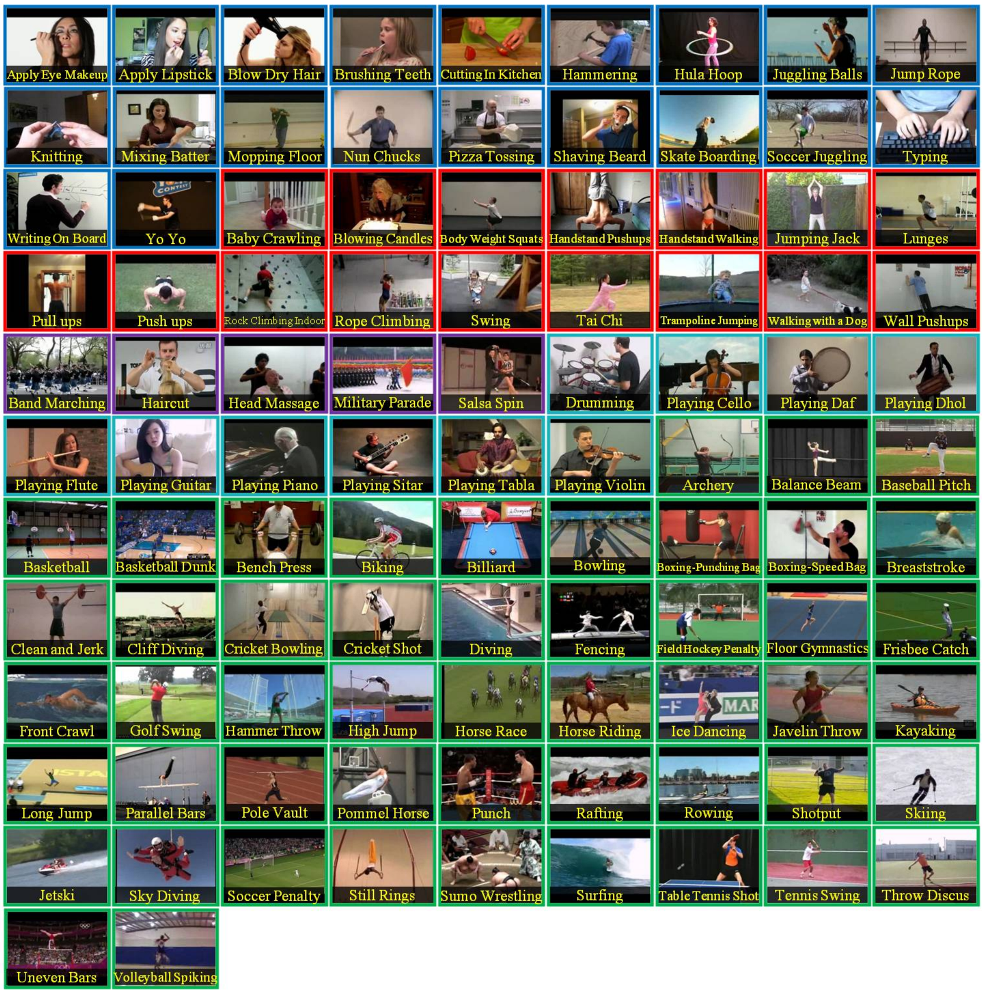


Figure 8. Sample images from the UCF101 Dataset [28].

In the Figure 10 is shown the decrease of the loss function during training process. The decrease of the loss function is directly proportional to the accuracy, which indicates good classification results. In the first epochs of training, the rapid drop in values is observed. Around the value of 0.5, the change in the loss function slows, but continues to decrease to its lowest value of approximately 0.08. The decrease in the loss function indicates error minimization, which is reflected by a directly proportional increase in the accuracy. Most of the time we would observe that accuracy increases with the decrease in loss (but this is not always the case). These two parameters (accuracy and loss function) have different definitions and measure different things. They often appear to be inversely proportional but there is no mathematical relationship between these two metrics.

Next, the model that generates a confusion matrix was created as is shown in Table 1. In this matrix, we see how the classification of individual classes proceeded. The classes are divided into the following categories (Table 1): 1. Basketball, 2. Biking, 3. Diving, 4. GolfSwing, 5. HorseRiding, 6. SoccerJuggling, 7. Swing, 8. TennisSwing, 9. TrampolineJumping, 10. VolleyallSpiking, 11. WalkingDog. The confusion matrix shows that the neural network had the most difficulty processing videos from the HorseRiding category. Incorrection classifications include four videos as WalkingDog, one video as swing, 1 asTennisSwing, two as biking and 2 as basketball. These results may indicate that the selected videos reminded the network of the surrounding environment, the presence of an animal or the angle of the camera. For example, the error of classifying a video from HorseRiding to Walkingdog could be due to the high similarity of the environment and the occurrence of an animal. In addition to the similarity in the videos, the classification results may have been influenced by factors such as video quality or the size of the input frames, which we input as 32 × 32 × 3 RGB (red green blue). Different sizes of input images could also cause different results. We chose the size of the input frames specifically based on the optimization of the memory requirements of the system with respect to the functionality of the algorithm.

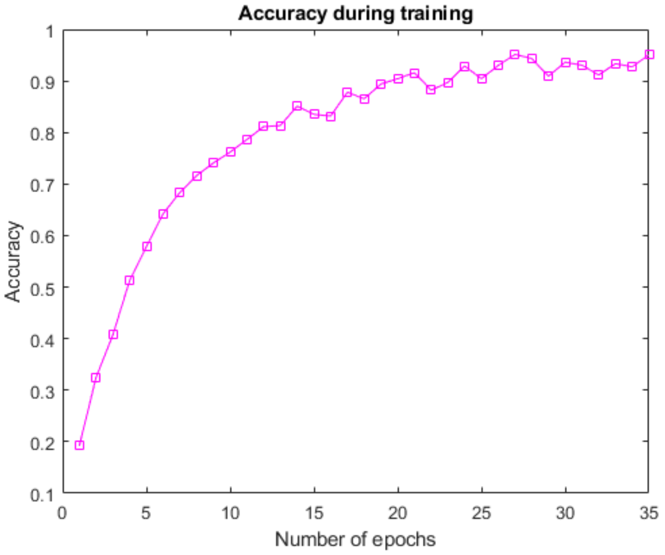


Figure 9. Accuracy during training process.

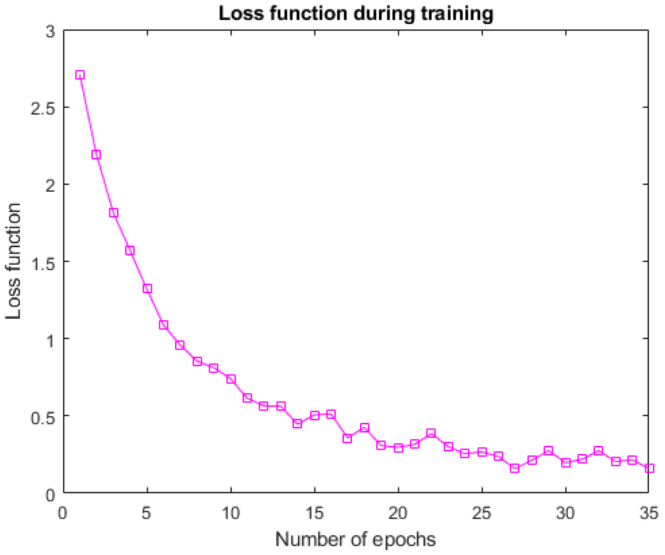


Figure 10. Loss function during training process.

Table 1. The confusion matrix for UCF YouTube Action dataset.

Predicted/Targeted	1	2	3	4	5	6	7	8	9	10	11
1	24	0	0	0	3	0	0	0	0	1	1
2	3	20	0	1	3	1	4	0	0	0	3
3	0	0	29	0	1	0	1	0	1	0	0
4	0	0	0	24	1	1	0	4	0	0	1
5	0	3	0	0	30	0	1	0	2	0	1
6	0	1	0	0	1	16	0	0	1	0	0
7	0	1	1	0	2	1	20	2	2	1	1
8	2	1	0	1	2	0	0	26	1	0	0
9	0	0	0	1	0	0	1	0	18	0	0
10	1	0	0	0	1	0	0	0	0	17	0
11	1	2	1	2	2	1	0	0	2	1	16

Next, each category was separately analyzed. The first category was Basketball. In this category, 29 videos were randomly selected. The 24 videos were correctly categorized. The most incorrectly classified videos were in the HorseRiding category (3 videos). In addition one video as VolleyballSpiking class and WalkingDog class was identified. Next, the videos of second class (Biking category) were classified. In this case, the 35 videos from this category were randomly selected. A total of 20 videos were correctly classified. On the other hand, the fifteen videos were incorrectly classified (four videos as Swingand class, three videos as Basketball class and HorseRiding class, one video as the GolfSwing class and SoccerJuggling class). In the third row of the confusion matrix, the videos of the Diving class were classified. The set contains 32 videos, of which 29 were correctly classified. Only three videos were incorrectly classified as HorseRiding class, Swing class and TrampolineJumping class. For the fourth category (GolfSwing class), the 31 videos were randomly selected. Within this category we can see that the 24 videos were correctly classified. The most incorrectly classified videos were in the TennisSwing category (4 videos). In addition, three videos were incorrectly assigned (one video as HorseRiding class, one video as SoccerJuggling class, and one video as WalkingDog class). In the next row, the classification results for the fifth category (HorseRiding class) are shown. In this category, the 37 random videos were selected (30 videos were correctly classified and 7 videos were incorrectly classified). The classification results of the SoccerJuggling category on 19 randomly selected videos were based. In this category, the 16 videos were correctly classified. Classification errors only for three videos were occurred (Biking class, HorseRiding class and TrampolineJumping class). For the classification of videos from the Swing category a classification errors in almost all categories were occurred. The model randomly selected 31 videos during testing. The two errors for the HorseRiding class, TrampolineJumping class and TennisSwing class were occurred. The one error for the Biking class, Diving class, SoccerJuggling class, VolleyballSpiking class and WalkingDog class was occurred. In the eighth row, the classification results of TennisSwing category are displayed. The test model for this class randomly selected 33 videos (26 videos were correctly classified). The two classification errors in the Basketball class and HorseRiding class were occurred. In addition, the one error video for Biking class, GolfSwing class and TrampolineJumping class was incorrectly assigned. The ninth category was TrampolineJumping. During testing, the proposed neural network randomly selected 20 videos (18 videos were correctly classified). On the other hand, the two videos into GolfSwing class and Swing class were incorrectly classified. On the next row, the classification results of the VolleyballSpiking class are displayed. From the 19 randomly selected videos were 17 videos correctly classified. Only two videos were incorrectly classified to Basketball class and HorseRiding class. Finally, the last category was WalkingDog class. For the classification of videos from the WalkingDog category a classification errors almost in all categories were occurred. The 16 videos were correctly classified. However, the model incorrectly classified two videos as Biking class, GolfSwing class, HorseRiding class and

TrampolineJumping class. The one video as Basketball class, Diving class, SoccerJuggling class and VolleyallSpiking class was incorrectly classified.

Overall, the test accuracy of 74.2% using UCF YouTube Action dataset was obtained. Accuracy is the percentage of correct answers relative to the total number. In addition to the average accuracy, we also calculated the precision, recall and F1 score. Precision corresponds to the probability of the detected instances of the activity to its actual occurrence, and in our case, the precision is 77.4%. Recall is 75.6%, and the F1 score is 76.5%. Recall describes the ability to not identify a positive example as negative. The F1 score is defined as a measure that provides a balance between recall and precision. The accuracy result obtained using the proposed 3DCNN network is compared with accuracy results previously reported for other networks (different architecture for motion, static and hybrid features) [7]. We also compared the results of the proposed architecture with previous research [22]. In previous research, we compared CNN, ConvLSTM and 3DCNN [27,29]. However, these architectures were tested on a different database, so the steps of our research led to a comparison of our proposed 3DCNN architecture in this paper with the previous one on the same UCF Youtube Action database.

We also applied our proposed architecture on full UCF101 dataset and modified UCF101 dataset. The modified UCF101 dataset consists of 15 classes (1. Baseball Pitch, 2. Basketball Shooting, 3. Bench Press, 4. Biking, 5. Billiards Shot, 6. Breaststroke, 7. Clean and Jerk, 8. Diving, 9. Drumming, 10. Fencing, 11. Golf Swing, 12. High Jump, 13. Horse Race, 14. Horse Riding, 15. Hula Hoop). These classes from the full UCF101 dataset were randomly selected. The full UCF101 dataset is more demanding, because it is expansive. The overall test accuracy of 84.4% using modified UCF101 dataset was achieved by the proposed neural network. On the other hand the overall test accuracy of 79.9% using full UCF101 dataset was achieved. Our results prove that the proposed architecture is applicable to different datasets, which contain human activities. The confusion matrix for modified UCF101 dataset is shown in Table 2.

Table 2. The confusion matrix for UCF15 dataset (modified UCF101 dataset).

Predicted/Targeted	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	28	0	0	0	0	0	0	2	1	0	0	0	2	0	0
2	0	13	0	2	0	0	0	4	2	0	0	4	2	3	0
3	0	0	21	0	0	0	0	0	3	0	0	0	0	0	0
4	0	0	0	20	0	0	1	1	0	0	0	0	0	0	0
5	1	0	0	0	26	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	20	0	0	1	0	0	0	0	0	0
7	0	0	0	0	0	0	18	0	3	1	0	0	0	0	0
8	0	0	0	1	0	0	0	26	4	0	0	2	0	0	1
9	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0
10	0	0	0	1	0	0	1	0	2	12	0	1	0	0	0
11	0	0	0	2	0	0	0	0	0	0	23	0	1	1	0
12	0	0	0	3	0	0	0	1	4	1	1	17	1	2	0
13	0	0	0	2	0	0	0	0	1	1	0	2	14	1	0
14	0	0	0	2	0	0	0	2	4	0	0	3	3	24	1
15	0	1	2	0	0	0	1	3	1	0	0	1	1	0	16

The evaluation metrics (P, R, F1) for proposed neural network using UCF YouTube Action dataset, modified UCF101 dataset and full UCF101 dataset were calculated (Table 3). The resulting accuracy was obtained as the sum of correctly predicted samples to the sums of all samples (85.2% using UCF YouTube Action dataset, 84.4% using modified UCF101 dataset, 82.2% using full UCF50 dataset and 79.9% using full UCF101 dataset).

In the Table 4, the comparison between accuracy and loss function for 35 epoches (training and testing procedure for the proposed neural network) is shown. In our case, the train loss is the value of the objective function that was minimized. This value (training loss) was calculated over the entire training dataset. On the other hand, the training accuracy means that identical images were used both for training and testing, while test accuracy represents that the trained model identifies independent images that were not used in training.

Table 3. The evaluation criterion of the proposed neural network architectures using different datasets.

Evaluation Metrics	UCF YouTube Action Dataset	Modified UCF101 Dataset	Full UCF50 Dataset	Full UCF101 Dataset
Precision (P)	87.4%	82.7%	80.6%	78.5%
Recall (R)	85.6%	87.7%	84.8%	81.2%
F1 score (F1)	86.5%	85.1%	82.6%	80.3%

Table 4. Accuracy and loss function model during training and testing after 35 epochs.

Train/ Test Parameters	UCF YouTube Action Dataset	Modified UCF101 Dataset	Full UCF50 Dataset	Full UCF101 Dataset
Train loss	0.08	0.07	0.14	0.21
Train accuracy	97.6%	98.5%	91.2%	87.8%
Test loss	1.51	1.43	1.61	1.79
Test accuracy	85.2%	84.4%	82.2%	79.9%

The proposed 3DCNN architecture achieved the best experimental results compared with other neural network architectures based on motion, static and hybrid features but also on our previous research. All approaches (Table 5) were tested on the same dataset. Our previous architecture was designed and tested on a reduced UCF101 dataset (10 classes), where it achieved results of 72% [22]. However, as soon as we used the UCF Youtube Action dataset, we got results only 29% [22]. This significant decrease in accuracy could have occurred due to overfitting. We subsequently modified the architecture. Based on the observation of the change in accuracy during the addition and removal of layers, we removed two layers. We also observed a change in accuracy during the number of filters for a given layer. In this case, too, we have reduced the number of filters for some layers by observation. As a result, we obtained a smaller model with better results on both datasets. Such a rapid change in the results could be caused by the complexity of the datasets used. Quality datasets are very much needed in the research on the classification of human abnormal behavior from video data. From this reason in our future work, we decided to create own dataset. This dataset will contain 11 classes with various abnormal human activities (Begging, Drunkenness, Fight, Harassment, Hijack, Knife Hazard, Normal Videos, Pollution, Property Damage, Robbery, Terrorism). The proposed architecture of 3DCNN is suitable for the classification of video sequences because this network considers the time dependence of input frames.

Table 5. Comparison of accuracy obtain by different neural network architectures using UCF YouTube Action dataset.

Algorithm for Recognition	Accuracy [%]
Architecture for motion features [7]	65.4
Architecture for static features [7]	63.1
Architecture for hybrid features [7]	71.2
Proposed 3DCNN architecture previous research [22]	29
Proposed architecture	85.2

4. Conclusions and Discussion

In this paper, we designed a neural network capable of classifying human activities from video using a 3DCNN architecture. We used the UCF YouTube Action database for training and testing. The classification was carried out for 11 classes: Basketball, Biking, Diving, GolfSwing, HorseRiding, SoccerJuggling, Swing, TennisSwing, TrampolineJumping, VolleyballSpiking, and WalkingDog. Videos in the dataset were cropped to an input size of 32 × 32 × 3 RGB. This choice could influence the overall accuracy obtained. However, we were able to evaluate whether the 3DCNN architecture was able to classify the videos with minimal errors. This statement is also confirmed by Figures 9 and 10, where we can observe an increase in accuracy and a decrease in the loss function. Based on the obtained results, we can state that such a neural network is able to classify video data containing a variety of human activities. During training, the loss function decreased to 0.08 and the accuracy increased to 97.6%. We can evaluate that the 3DCNN architecture coped well with the problem of time continuity between frames as it was able to process time information in addition to spatial information. When evaluating the confusion matrix, we can state that the process of classification into classes was successful and had a minimal occurrence of errors. The neural network made the most errors when classifying videos from the HorseRiding category. Videos from this category were most often categorized as WalkingDog.

For the overall results using UCF YouTube Action dataset, the evaluation metrics (average test accuracy, precision, recall and F1 score) were obtained. The overall test accuracy was 85.2%. The precision reached value 87.4% and recall 85.6%. The F1 score was 86.5%. The results clearly confirm that we have created a capable neural network architecture for the classification of human activities. In addition to spatial information, the neural network can also process time information. However, we want to continue our work and improve our results. The more accurately the neural network can classify human activity, the more we can use it in real life applications such as personal safety, care for the elderly and protection of minors.

For a more complex comparison of the proposed neural network (3DCNN), the modified UCF101 dataset has been added. The full UCF50 dataset and full UCF101 dataset have also been added. The overall test accuracy using modified UCF101 dataset was 84.4%. The precision was 82.7% and recall 87.7%. The F1 score reached value 85.1%. On the other hand, the overall test accuracy using full UCF50 dataset and full UCF101 dataset was 82.2% and 79.9%.

Figure 4 shows a trend of increasing accuracy. In the future, we plan to run this experiment with a wider range of batch sizes and numbers of epochs.

The main benefit of this contribution for the scientific community is mainly in the field of the recognition and classification of non-standard behavior of people in public spaces. This monitoring can be used in different real applications. For this purpose, the overall accuracy of the proposed neural network (84.4% for modified UCF101 dataset and 85.2% for UCF YouTube Action dataset) is sufficient. The use of this proposed solution (proposed 3DCNN architecture) is mainly in the field of medicine (exploring links between non-standard health behaviors and various psycho-social factors) and to the monitoring and classification of the human non-standard behavior in public places (squares, parks, railway stations and so on). However, the monitoring and classification of the human non-standard behavior in public places, such as city parks, sports stadiums or city squares, is a very difficult task. The result of this work will be warnings in case of human non-standard behavior in these areas of interest.

Author Contributions: Conceptualization, R.V., P.K. and R.H.; methodology, R.V.; software, R.V. and P.S.; validation, P.K., R.V. and R.H.; formal analysis, P.K., R.H. and P.S.; data curation, R.V.; writing—original draft preparation, R.V. and P.K.; writing—review and editing, R.V. and P.K.; visualization, R.V., P.K. and P.S.; supervision, R.H.; project administration, R.H. and P.K.; funding acquisition, R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Slovak Research and Development Agency under the contracts no. APVV-18-0167 and APVV-16-0190. This work was supported by the Slovak Research and Development Agency under project PP-COVID-20-0100: DOLORES. AI: The pandemic guard system and by the project Grant System of University of Zilina No. 1/2020/8084.

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to the experiment did no harm to all subjects.

Informed Consent Statement: Patient consent was waived due to that we used existing databases for testing (UCF YouTube Action Dataset and UCF101 Dataset). These databases were cited in the text of the article.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. This is according to the laboratory rules.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

3D	Three-Dimensional
3DCNN	3D Convolutional Neural Network
ATM	Automated Teller Machine
CNN	Convolutional Neural Network
ConvLSTM	Convolutional Long Short-Term Memory
Conv2D	Two-Dimensional Convolution layer
LSTM	Long Short-Term Memory

References

1. Olmos R.; Tabik S.; Herrea F. Automatic handgun detection alarm in videos using deep learning. *Neurocomput. J.* **2017**, *275*, 66–72. [CrossRef]
2. Dhiman, C.H.; Vischakarma, D. High dimensional abnormal human activity recognition using histogram oriented gradients and Zernike moments. In Proceedings of the International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, India, 14–16 December 2017; pp. 869–872.
3. Peixoto, B.; Avila, S.; Dias, Z.; Rocha, A. Breaking down violence: A deep-learning strategy to model and classify violence in videos. In Proceedings of the International Conference on Availability, Reliability and Security (ARES), Hamburg, Germany, 27–30 August 2018; pp. 1–7.
4. Ramzan, M.; Abid, A.; Khan, H.A. Review on state-of-the-art violence detection techniques. *IEEE Access* **2019**, *7*, 107560–107575. [CrossRef]
5. Liu, J.; Yang, Y.; Shah, M. Learning Semantic Visual Vocabularies using Diffusion Distance. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009.
6. Liu, J.; Luo, J.; Shah, M. Recognizing Realistic Actions from Videos "in the Wild". In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009.
7. Zhang, X.; Yao, L.; Huang, C.; Sheng, Q.Z.; Wang, X. Intent recognition in smart living through deep recurrent neural networks. In Proceedings of the International Conference on Neural Information Processing (ICONIP), Guangzhou, China, 14–18 November 2017; Volume 10634, pp. 748–758.
8. Guo, Z.H.; Chen, Y.; Huang, W.; Zhang, J.H.; Wang, X. An Efficient 3D-NAS Method for Video-Based Gesture Recognition. In Proceedings of the International Conference on Artificial Neural Networks (ICANN), Munich, Germany, 17–19 September 2019; Volume 11729, pp. 319–329.
9. Rastgoo, R.; Kiani, K.; Escalera, S. Hand sign language recognition using multi-view hand skeleton. *Expert Syst. Appl.* **2020**, *150*, 113336, pp. 1–12. [CrossRef]
10. Wang, T.; Li, J.K.; Zhang, M.Y.; Zhu, A.C.; Snoussi, H. An enhanced 3DCNN-ConvLSTM for spatiotemporal multimedia data analysis. *Concurr. Comput.-Pract. Exp.* **2021**, *33*, e5302. [CrossRef]
11. Castro-Vargas, J.; Zapata-Impata, B.; Gil, P.; Garcia-Rodriguez, J.; Torres, F. 3DCNN Performance in Hand Gesture Recognition Applied to Robot Arm Interaction. In Proceedings of the International Conference on Pattern Recognition Applications and Methods (ICPRAM), Prague, Czech Republic, 19–21 February 2019; pp. 802–806.
12. Mishra, B.; Garg, D.; Narang, P.; Mishra, V. A hybrid approach for search and rescue using 3DCNN and PSO. *Neural Comput. Appl.* **2021**, *33*, 10813–10827. [CrossRef]
13. Wang, Y.H.; Dantcheva, A. A video is worth more than 1000 lies. Comparing 3DCNN approaches for detecting deepfakes. In Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition and Workshops, Buenos Aires, Argentina, 16–20 November 2020; pp. 515–519.
14. Al-Hammadi, M.; Muhammad, G.; Abdul, W.; Alsulaiman, M.; Bencherif, M.A.; Mekhtiche, M.A. Hand Gesture Recognition for Sign Language Using 3DCNN. *IEEE Access* **2020**, *8*, 550-559. [CrossRef]
15. de Oliveira Lima, J.P.; Figueiredo, C.M.S. Temporal Fusion Approach for Video Classification with Convolutional and LSTM Neural Networks Applied to Violence Detection. *Intel. Artif.* **2021**, *24*, 40–50. [CrossRef]
16. Tomei, M.; Baraldi, L.; Calderara, S.; Bronzin, S.; Cucchiara, R. Video action detection by learning graph-based spatio-temporal interactions. *Comput. Vis. Image Underst.* **2021**, *206*, 103187. [CrossRef]

17. Castro-Vargas, J.; Zapata-Impata, B.; Gil, P.; Garcia-Rodriguez, J.; Torres, F. Mobile Neural Architecture Search Network and Convolutional Long Short-Term Memory-Based Deep Features Toward Detecting Violence from Video. *Arab. J. Sci. Eng.* **2021**, *46*, 8549–8563.
18. Lin, W.; Gao, J.; Wang, Q.; Li, X. Learning to detect anomaly events in crowd scenes from synthetic data. *Neurocomputing* **2021**, *436*, 248–259. [CrossRef]
19. Ahad, M.A.R.; Ahmed, M.; Antar, A.D.; Makihara, Y.; Yagi, Y. Action recognition using kinematics posture feature on 3D skeleton joint locations. *Pattern Recognit. Lett.* **2021**, *145*, 216–224. [CrossRef]
20. Sultani, W.; Shah, M. Human Action Recognition in Drone Videos using a Few Aerial Training Examples. *Comput. Vis. Image Underst.* **2021**, *206*, 103186. [CrossRef]
21. Hou, H.; Li, Y.; Zhang, C.; Liao, H.; Zhang, Y.; Liu, Y. Vehicle Behavior Recognition using Multi-Stream 3D Convolutional Neural Network. In Proceedings of the 2021 36th Youth Academic Annual Conference of Chinese Association of Automation (YAC), Nanchang, China, 28–30 May 2021; Volume 32, pp. 355–360.
22. Vrskova, R.; Hudec, R.; Sykora, P.; Kamencay, P.; Radilova, M. Education of Video Classification Based by Neural Networks. In Proceedings of the International Conference on Emerging eLearning Technologies and Applications (ICETA), Košice, Slovakia, 12–13 November 2020; pp. 762–767.
23. Vrskova, R.; Hudec, R.; Sykora, P.; Kamencay, P.; Benco, M. Violent Behavioral Activity Classification using Artificial Neural Network. In Proceedings of the New Trends in Signal Processing (NTSP), Demanovska Dolina, Slovakia, 14–16 October 2020; pp. 1–5.
24. Partila, P.; Tovarek, J.; Ilk, G.H.; Rozhon, J.; Voznak, M. Deep learning serves voice cloning: How vulnerable are automatic speaker verification systems to spooting trial. *IEEE Commun. Mag.* **2020**, *58*, 100–105. [CrossRef]
25. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 221–231. [CrossRef]
26. Chengping, R.; Yang, L. Three-dimensional convolutional neural network (3D-CNN) for heterogeneous material homogenization. *Comput. Mater. Sci.* **2020**, *184*, 109850.
27. Vrskova, R.; Sykora, P.; Kamencay, P.; Hudec, R.; Radil, R. Hyperparameter Tuning of ConvLSTM Network Models. In Proceedings of the 2021 44th International Conference on Telecommunications and Signal Processing (TSP), Brno, Czech Republic, 26–28 July 2021; pp. 15–18.
28. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild. *arXiv* **2012**. arXiv:1212.0402.
29. Trnovszky, T.; Kamencay, P.; Orjesek, R.; Benco, M.; Sykora, P. Animal recognition system based on convolutional neural network. *Adv. Electr. Electron. Eng.* **2017**, *15*, 517–525. [CrossRef]