AiMT

Advances in Military Technology
Vol. 15, No. 1, 2020, pp. 67-83
ISSN 1802-2308, eISSN 2533-4123
DOI 10.3849/aimt.01326

Comparison of Neural Networks with Feature Extraction Methods for Depth Map Classification

P. Sykora, P. Kamencay*, R. Hudec, M. Benco and M. Sinko

Department of Multimedia and Information-Communication Technologies, University of Zilina, Slovakia

The manuscript was received on 20 March 2019 and was accepted after revision for publication as research paper on 13 February 2020.

Abstract:

In this paper, a comparison between feature extraction methods (Radon Cosine Method, Canny Contour Method, Fourier Transform, SIFT descriptor, and Hough Lines Method) and Convolutional Neural Networks (proposed CNN and pre-trained AlexNet) is presented. For the evaluation of these methods, depth maps were used. The tested data were obtained by Microsoft Kinect camera (IR depth sensor). The feature vectors were classified by the Support Vector Machine (SVM). The confusion matrix for the evaluation of experimental results was used. The row of confusion matrix represents target class of tested data and the column represents predicted class. From the experimental results, it is evident that the best results were achieved by proposed CNN (97.4%). On the other hand, the pre-trained AlexNet scored 93.7%.

Keywords:

Convolutional Neural Network, deep learning, depth map, Fourier transform, Radon transform

1. Introduction

Hand gestures can be seen from multiple points of view. Firstly, they can be represented by the motion of the hand. Secondly, they can be represented by the shape of the hand (position of fingers). The shape of the hand is determined by the position of fingertips relative to the palm. For example, one finger straight up and others folded into fist is a simple gesture for number one. Following this, two straight fingers represent number two, etc. The question is how to describe this shape most effectively. There are several methods to do so, e.g. original RGB colour space is transformed into YCbCr and K-means segmentation method is applied. Subsequently, orientation of hand in picture is detected by calculating simple ration between width and height of hand region. Thumb is detected by measuring pixels on the side of hand. Moreover,

* *Corresponding author: Department of multimedia and information-communication technologies, University of Zilina, Univerzitna 8215/1, Zilina, Slovakia. E-mail: patrik.kamencay@fel.uniza.sk*

centroid and fingertip position using Euclidean distance is calculated. The comparison of feature extraction methods and deep learning framework for depth map recognition is described in [1] and [2].

The sensor-based solutions use accelerometers or gyros for detecting the gestures. The motion is described by the accelerations in a specific direction in time. Current smartphones include many sensors, among others accelerometers, too [3]. Data from smartphone accelerometer can be used to recognize general activity. In [4], the inertial sensors were used for hand-gesture detection. The method for hand gesture recognition based on a hand contour using Kinect sensor in [5] is presented. The issue of hand gesture recognition from sequences of depth maps in [6, 7] is described. In [8], Radon transform is used in the process of feature extraction from hand contour. Firstly, a keyframe (or frames) is selected from a sequence. Next, the hand region is segmented from the background by skin colour. An overlap of another object of skin colour, like head, results in failed segmentation. The segmented region is a binary image with fixed size resolution 150 by 150 pixels. To find continuous contour, Exclusive-or operation is used. Canny edge detector finds a discontinuous edge. Subsequently, Radon transform is applied onto this contour image. The resulted feature vector is invariant to rotation.

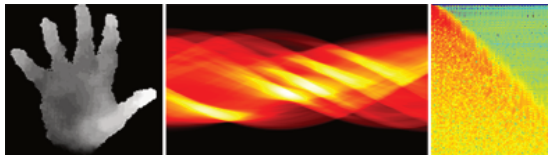
The outline of this paper is divided into the following sections. In Section 2, the methods for gesture recognition based on the feature extraction are described. Also, the results achieved using these methods in the same section are presented. The obtained experimental results using proposed CNN are listed in Section 3. Finally, the last Section 4 concludes and suggests future work.

2. Feature Extraction

Feature extraction is a type of dimensionality reduction that efficiently represents interesting parts of an image as a compact feature vector (extracting the information from the raw data that is most relevant for discrimination between the classes). Deep learning models can also be used for automatic feature extraction algorithms.

2.1. Radon Cosine Method

Firstly, Discrete Radon Transform (DRT) [8] is applied in a range of angles to produce Radon spectrum image. The Radon spectrum image (Fig. 1) appears to be contracted from multiple waves. Next, Discrete Cosine Transform (DCT) is used to describe these waves. The spectrum of the image is rearranged after the transform in such a way that lower frequencies are located in the upper left corner and higher frequencies are located in the lower corner.



a) b) c)

Fig. 1 Image transform in RCM, a) input keyframe, b) Radon spectrum image, c) cropped cosine spectrum image

The data in the lower right corner can be neglected without a significant loss of data. The basic block diagram of Radon Cosine Method (RCM) is shown in Fig. 2.

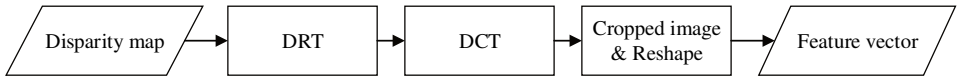


Fig. 2 The block diagram for Radon Cosine Method

The output of this method is a resized image. A range of several sizes was tested to find the highest precision. The best result was measured for a resized image of 20 by 20 pixels. This image is transformed into feature vector as the last step. To evaluate this method, the precision, recall and *F*1 measures were used. The experimental results in confusion matrix are stored (Fig. 3). Each field in this matrix contains the sum of all images.

Targeted class \ Predicted class	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	8	0	0	1	0	0	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	9	1	0	0	0	0	0	0	0	0
7	0	0	0	1	0	1	7	0	0	0	1	0	0	0	0
8	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
12	0	0	0	0	0	2	0	0	0	0	1	7	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0
15	0	0	0	0	0	0	0	0	0	0	0	2	0	0	8

Fig. 3 Confusion matrix for Radon Cosine Method

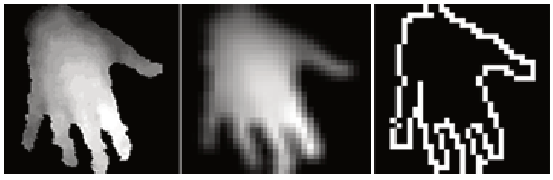
The rows in this matrix represent actual class of tested data (Target class) and each column represents predicted class. The green colour (Fig. 3) represents 100% true positive prediction. On the other hand, the blue colour represents true positive prediction but less than 100% (true positive). Finally, the red colour represents the false prediction (false positive, false negative, true negative). This method recognizes 10 classes on 100%.

2.2. Canny Contour Method

Canny Contour Method (CCM) [9] uses Canny Edge Detector (CED), as you can see in Fig. 4. This method works in three steps (Fig. 5). Firstly, it resizes the input keyframe. Experiments with different image resizes show little variation in precision. Nevertheless, size 30 by 120 was chosen, as it gives highest precision.

Fig. 4 The block diagram for the Canny Contour Method

Secondly, it applies Gaussian filter to blur the image. Function blur from OpenCV Library (OCVL) was used with 3×3 kernel size. Finally CED was applied.



a) b) c)

Fig. 5 Image transform in CCM, a) input keyframe, b) after resize and application of Gaussian filter, c) contour image after CED

The process of non-maximal values suppression is executed to reach the required precision of edge detection. Thinner (sharpen) lines are the result after this step. The resulted binary image is done by applying threshold [9, 10].

The achieved results using CCM in a confusion matrix are displayed (Fig. 6). Each field in this matrix contains the sum of images. The rows in this matrix represent actual class of tested data (Target class), and each column represents predicted class.

Targeted class \ Predicted class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	7	0	2	0	0	0	0	0	0	0	0	1	0	0
3	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	8	1	0	0	0	0	0	0	0	0	0	0
5	0	1	0	0	8	1	0	0	0	0	0	0	0	0	0
6	0	0	0	0	2	7	0	0	0	1	0	0	0	0	0
7	0	0	1	0	0	0	9	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	1	9	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	9	0	1	0
13	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0
14	0	0	0	0	0	0	1	0	0	0	0	0	0	9	0
15	0	1	0	0	1	0	0	0	0	0	0	0	0	0	8

Fig. 6 Confusion matrix for Canny Contour Method

2.3. Fourier Transform Method

The Fourier Transform Method (FTM) [11, 12] is divided into two main stages (Fig. 7), the image transform (green part) and Fourier transform (red part).

Transformation of input greyscale keyframe (Fig. 8a) into binary image (Fig. 8b) is done at first. Contours are found applying CED (Fig. 8c). Contour-Curve image (Fig. 8e) is created with Polar-Cartesian transform (Fig. 8d).

Moreover, each column is taken from this image and first nonzero data is found. The height of such values in all columns of an image creates the signal vector. The size of signal vector is 150, which is the width of the image. The spectrum of an image contains information about energy of harmonic signals in original image. More energy is concentrated in smaller frequencies than in higher frequencies.

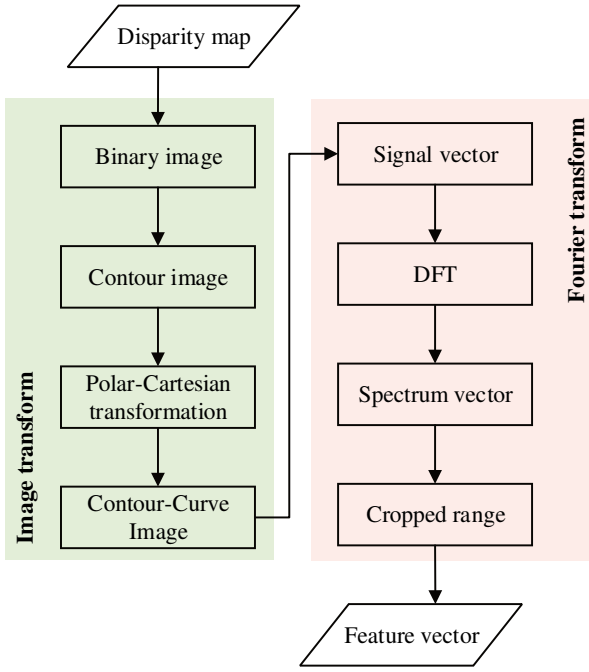


Fig. 7 The block diagram of the Fourier Transform Method

Fig. 8 Image transform in FTM, a) input keyframe, b) binary image, c) contour image, d) after Polar-Cartesian transform, e) Contour-Curve image

Therefore, higher frequencies contain less information and are similar to noise. The spectrum of this signal is acquired after DFT with 150 samples. The achieved experimental results based on Fourier Transform Method in Fig. 9 are presented.

2.4. SIFT Descriptor

The Scale-Invariant Feature Transform (SIFT) [13] is a widely used algorithm for detection and description of local image features (Fig. 10). Firstly, the key points are found. The Laplace-Gauss filter is applied to produce a series of smoothed images. Key points are located as extremes in this series.

Secondly, the located key points are described. Gradients are created in matrix 4×4 for each point. Next, several parameters can be filtered to exclude inappropriate key points. Finally, feature points of the size smaller than 20 pixels are excluded [13]. The achieved confusion matrix can be seen in Fig. 11.

Targeted class \ Predicted class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	9	1	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	8	0	0	1	0	0	0	1	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0
6	0	0	1	0	0	8	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	8	0	0	0	2	0	0	0	0
8	0	0	0	0	0	0	1	9	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0
10	0	0	0	0	0	2	0	0	0	7	1	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
12	0	0	0	0	0	1	0	0	0	0	2	7	0	0	0
13	0	0	0	0	0	0	0	0	0	0	1	0	8	1	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0
15	0	0	0	0	0	0	0	0	1	0	0	0	0	0	9

Fig. 9 Confusion matrix for the Fourier Transform Method

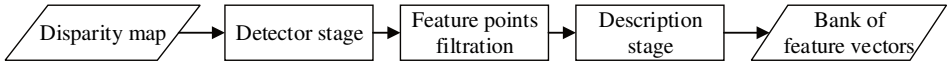


Fig. 10 The block diagram for SIFT

Targeted class \ Predicted class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	7	0	0	0	0	3	0	0	0	0	0	0	0	0	0
2	0	8	0	0	0	0	0	2	0	0	0	0	0	0	0
3	0	0	7	2	0	0	0	0	1	0	0	0	0	0	0
4	1	0	0	7	0	0	0	2	0	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	1	9	0	0	0	0	0	0	0	0	0
7	2	0	0	0	0	2	6	0	0	0	0	0	0	0	0
8	0	0	0	0	0	1	0	8	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	4	6	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
12	0	1	0	0	0	0	0	0	0	0	0	9	0	0	0
13	1	0	0	0	0	0	0	0	0	1	0	0	8	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0
15	0	0	2	0	1	0	0	0	0	0	0	0	0	0	7

Fig. 11 Confusion matrix for SIFT

2.5. Hough Lines Method

The Hough Lines Method (HLM) [14] is closely related to Radon transform. As opposed to previously mentioned methods, Hough Lines Method (HLM) produce more than one feature vector per input image (Fig. 13). Each vector is composed of ρ and θ, where ρ is the distance of Hough line from the origin and θ is the angle of line to the horizon. The angle of 0° has a vertical line and the angle of $\pi/2$ has a horizontal line.

The lines are extracted using Hough Lines function from OpenCV. The binary image from keyframe is the input for this function. The algorithm steps can be seen in Fig. 12. These lines (Fig. 13d) are used to describe the shape of an object as a feature vector.

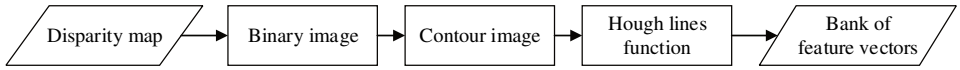


Fig. 12 Block diagram for SIFT

Fig. 13 Feature extraction using HLM, a) original keyframe, b) binary image, c) contour image, d) detected lines

No relation is maintained between lines from one picture. Thus, a specific line can occur in multiple pictures. Moreover, precision is lowered due to this (Fig. 14). The input keyframe is resized to eliminate insignificant lines.

Targeted class \ Predicted class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	7	0	0	1	0	0	0	1	0	0	1	0	0	0	0
2	0	6	0	0	2	0	0	0	0	1	0	0	1	0	0
3	0	1	6	0	0	2	0	0	1	0	0	0	0	0	0
4	0	0	1	7	0	0	0	0	0	0	0	2	0	0	0
5	0	1	0	0	5	0	2	0	0	0	2	0	0	0	0
6	0	0	2	0	0	5	0	0	1	0	1	0	0	1	0
7	1	0	0	0	0	2	6	0	0	0	0	0	0	1	0
8	0	1	0	0	0	1	0	7	0	0	0	1	0	0	1
9	0	0	1	1	0	0	0	2	5	0	0	1	0	0	0
10	0	0	0	0	2	0	0	0	0	7	0	0	0	1	0
11	0	0	0	0	2	0	1	0	0	0	6	0	0	1	0
12	0	2	0	0	1	0	0	0	0	0	0	6	1	0	0
13	1	0	0	0	2	0	2	0	0	0	0	0	5	1	0
14	0	0	0	0	0	0	1	0	0	0	2	0	0	7	0
15	2	0	0	0	2	0	0	0	0	0	0	0	0	0	6

Fig. 14 Confusion matrix for Hough Lines Method

2.6. AlexNet

The AlexNet (Fig. 15) is a deep Convolutional Neural Network (CNN) to classify the 1.3 million high-resolution images in the LSVRC-2010 ImageNet training dataset. This dataset contains 1000 different object categories (classes) such as pencil, keyboard or many types of animals [15, 16].

The AlexNet (Fig. 15) consists of five convolutional layers, three Max Pooling layers and three Fully Connected layers (FC). The layers labelled by red color are convolutional layers. The Pooling layers are shown in green colour and layers marked in yellow are Fully Connected layers [17, 18]. The rows in this matrix (Fig. 13) represent the actual class of tested data (Target class) and each column represents the predicted class. The input images for AlexNet must be colour images. In our case, the input images were grayscale. These images were converted to RGB images (3-channel RGB image).

DOI 10.3849/aimt.01326

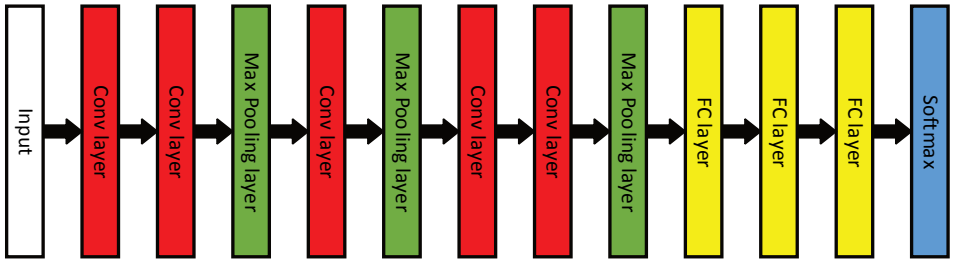


Fig. 15 Basic architecture of AlexNet [15]

The results from AlexNet are stored in a confusion matrix (Fig. 16). Each field in this matrix contains the sum of pictures. The green colour (Fig. 16) represents 100% true positive prediction. On the other hand, the blue color represents true positive prediction but less than 100% (true positive). Finally, the red colour represents false prediction (false positive, false negative, true negative).

Targeted class \ Predicted class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0
4	0	2	0	7	0	1	0	0	0	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0
9	0	3	0	0	0	0	0	0	7	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	8	2	0	0	0	0
11	0	0	0	0	0	0	0	0	0	1	9	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0
15	0	2	0	0	0	0	0	0	0	0	0	0	0	0	8

Fig. 16 Confusion matrix for AlexNet

3. Experimental Results

The obtained experimental results will be presented in this section. The whole system (Fig. 17) is divided into two classification parts (the training part and the testing part). Prior to classification, the procedures are identical for both parts. Firstly, the captured depth video sequence is processed by segmentation algorithm. The hand region is extracted from the background and keyframes are selected here. Secondly, feature extraction method is applied to these segmented keyframes. Resulted feature vectors are the input for classification.

Feature vectors determined for training are labelled by their respective class. Model for static recognition is generated in the training part. Using this model and given feature vector, static recognition will predict 3 labels for static classes. Two motion vectors are added to these labels to create feature vector for dynamic recognition. First motion vector stores difference in a hand position between the first and the second keyframes. Similarly, for the second motion vector, the range is the

second and third keyframes. The resulted feature vector with a given model is used to predict the label of dynamic gesture.

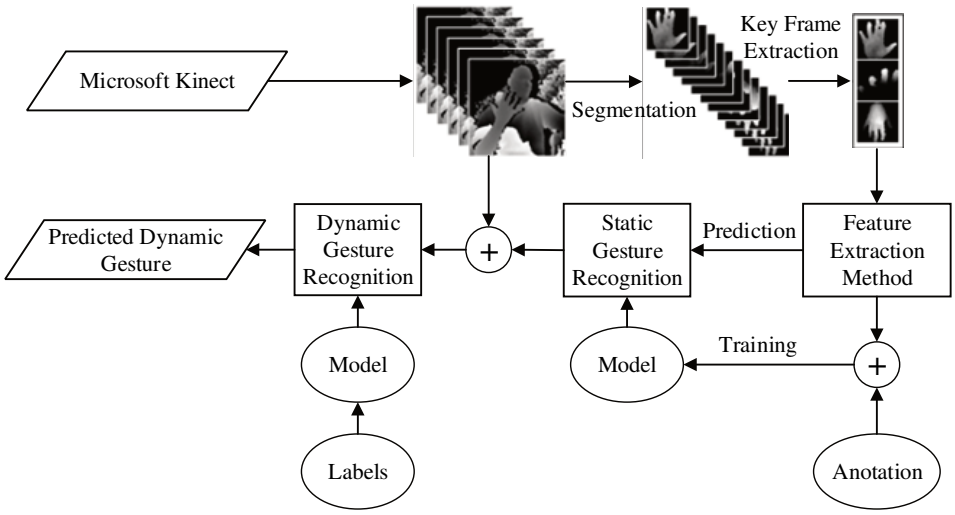


Fig. 17 Overview of the proposed system

3.1. Dataset

This dynamic gesture database was produced by 10 actors (3 females and 7 males). As some shapes are shared with multiple gestures, this leaves only 15 unique hand shapes. The image dataset contains these 15 static gestures (Fig. 18). These are 15 classes which give a total of 1350 images for training and 150 images for testing. All test sequences were produced by Microsoft Kinect camera at 640 by 480 pixels.

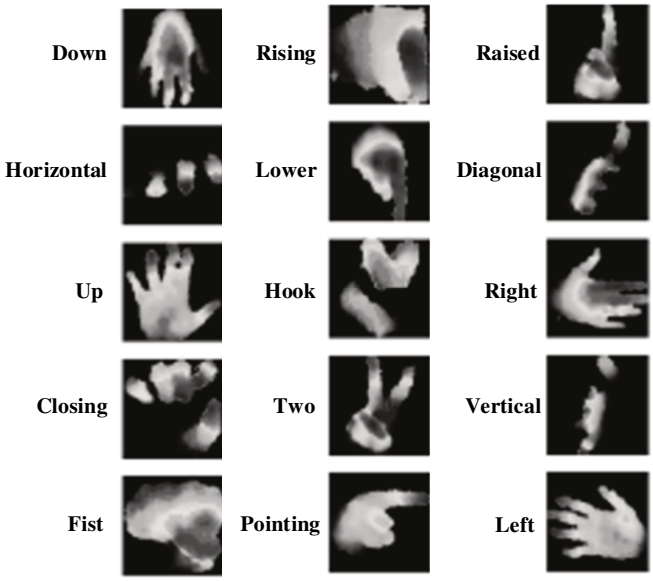


Fig. 18 Example of the test image dataset

This database contains information about the whole dynamic gesture. In regards to different sequence gesture tempos, they had not the same length. Moreover, these sequences were re-processed to 150 by 150 pixels resolution.

3.2. Proposed CNN

The CNN consists of several layers. Each layer occupies a multi–dimensional array of numbers and creates an additional multi-dimensional array of numbers as output. To create the CNN architectures, these types of layers have been used: Convolution (Conv) Layer, ReLU layer, Max Pooling Layer and Fully–Connected Layer [17, 18]. The CNN is a network composed of layers that transform an input image from the original pixel values to the final layer score (layer by layer).

Our proposed CNN has two convolutional layers excluding the source input data layer, two fully–connected layers, three ReLU layers and max pooling layer (Fig. 19). Each layer has multiple functional maps, each of which can extract one selected feature through a convolution filter and it contains multiple neurons. This proposed CNN (Fig. 19) is divided into 9 main blocks (A-I):

- block A – the input data images reshaped as vectors were used,
- block B –- this block describes the 2D CNN layer which has 32 feature maps with 3×3 kernel dimension,
- block C – Rectifier linear unit (ReLU) was used (its derivate is either 0 or 1),

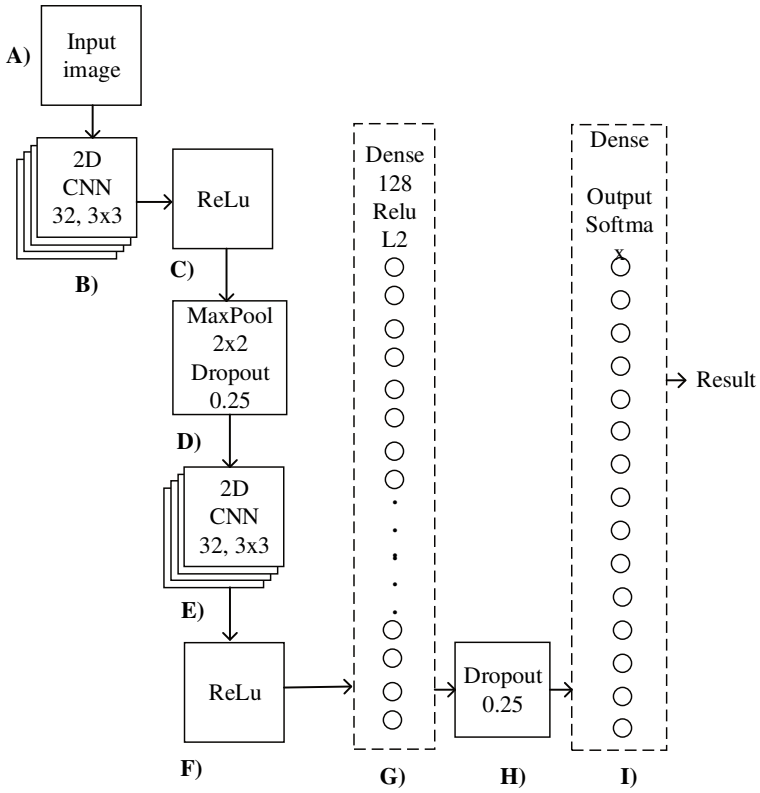


Fig. 19 The block diagram of proposed CNN

- block D – MaxPooling layers with dimension 2×2 were used and dropped out with probability 0.25. The Max Pooling is downsampling in CNN. This downsampling is created by applying a Max filter (in our case 2×2 filter),
- block E – in this block the same 2D CNN were used with parameters as in block B,
- block F – Rectifier linear unit (ReLU) was used (its derivate is either 0 or 1),
- block G – the standard dense layer was used,
- block H – the output of the last dropout layer would be passed to a 5-way Softmax of the loss layer,
- block I – the final output is Softmax activation function (validation of the training progress).

In the proposed CNN (Fig. 19), the pooling operations are applied separately to each feature map. Generally, the more convolutional steps we have, the more complex the features of our proposed network will be able to learn to recognize. For example, in the classification of images, CNN can learn to detect the margins of the raw pixels in the first layer, then use the margins to detect simple shapes in the second layer, and then use these shapes to detect higher-level features (face shapes) in the higher layers.

Targeted class \ Predicted class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	9	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	9	0	1	0	0	0	0
10	0	0	0	0	0	0	0	0	0	8	2	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0
15	0	1	0	0	0	0	0	0	0	0	0	0	0	0	9

Fig. 20 Confusion matrix for proposed CNN

The results from proposed CNN are stored in a confusion matrix (Fig. 20). Each field in this matrix contains the sum of pictures. The rows in this matrix represent the actual class of tested data (Target class) and each column represents the predicted class. For example, in Fig. 20, for row 10, which is for the input images of class 10, two images were wrongly recognized as images of class 11 and 8 images were correctly recognized as images of class 10. The proposed CNN can be used within the project PREDICON (the short-term PREDICtion of photovoltaic energy production for needs of pOwer supply of Intelligent BuildiNgs) to identify clouds in the sky.

3.3. Results

The problem of multiclass classification can be understood as a set of many problems with binary classification - one for each class. To evaluate the methods described in the bellow, the precision (the number of items that have been correctly identified as positive from the total positive items), recall (the number of items that have been cor-

rectly identified as positive from total real positive items) and $F1$ (the harmonious average of precision and recall) measures were used. These methods with proposed CNN were compared (Table 1). To ensure the real-time capability of system, 3 time parameters were measured. The time needed to train the model is the first measured time parameter. The time to calculate the feature vector (FV) is the next one. The overall time of prediction is the last measured time. This parameter represents the time needed for feature vector extraction and prediction of Support Vector Machines (SVM) model onto that vector. The evaluation scores:

- precision is the ration of the true positive and the sum of positive data. The approval of actual labels with classifier classes is calculated by adding all true positive and false positive in the system. The formula for precision is as follows:

$$P = \frac{\sum_{i=1}^{k} \text{true\_positive}_i}{\sum_{i=1}^{k} \left(\text{true\_positive}_i + \text{false\_positive}_i \right)} \tag{1}$$

- recall is the ration of the true positive and the sum of data from the target class. Classifier efficiency to identify class labels is calculated by adding all true positive and false negative in a system in all classes. The formula is as follows:

$$R = \frac{\sum_{i=1}^{k} \text{true\_positive}_i}{\sum_{i=1}^{k} \left(\text{true\_positive}_i + \text{false\_negative}_i \right)} \tag{2}$$

- measurement $F1$ is a combination of precision and recall. It is calculated as the harmonic mean in the following formula:

$$F1 = 2\frac{P \cdot R}{P + R} \tag{3}$$

Tab. 1 Comparison of all tested methods

	Methods of Gesture Recognition						
	RCM	CCM	FTM	SIFT	HLM	AlexNet	Proposed CNN
Precision	92.1	89.3	91.9	81.4	62.3	94.7	97.5
Recall	91.8	89.1	90.4	80.8	60.5	92.7	97.3
$F1$	91.9	89.2	91.1	81.1	61.4	93.7	97.4
Training time [s]	55	60	50	72	35	350	125
FV computing [ms]	30	45	125	40	50	70	15
Prediction time [s]	25	85	110	60	37	120	20

Next, the combination of the two proposed cluster separation methods {Simple Euclidean Distance (SED) and Mean Value Distance (MVD)) and the two keyframe extraction methods (Global Vector Median (GVM) and Local Vector Median (LVM)} were tested. The Simple Euclidean Distance (SED) method calculates Euclidean distance of two neighbouring frames. It is a distance of two vectors (reshaped frame to vector by lines) in *n*-th dimensional space. Its size is given by the number of depth map pixels. Thus, if a sequence contains 14 frames, there are 13 neighbouring distance features. Let *d* be the vector of SED's distances, *S* be the a frame sequence of length *E* and L_2 be the function that calculates Euclidean distance of two vectors:

$$d_e = L_2\left[S(e), S(e+1)\right], \qquad e \in \{1,2,3,\ldots,E\} \tag{4}$$

The Mean Value Distance (MVD) method calculates the 2D mean value of differences of two neighbouring frames. Let *m* be the vector of MVD's distance presented as follows:

$$m_e = \text{mean}\left[S(e-1) - S(e)\right], \qquad e \in \{1,2,3,\ldots,E\} \tag{5}$$

where *S* is the frame sequence of length *E* and 'mean' is the function that calculates the mean value of a vector.

The Global Vector Median (GVM) method uses global feature to measurement of frame similarity. Let *Q* be the frame cluster with a length of *F*-frames and *A* is the matrix of Euclidean distances. It consists of Euclidean distances between all frame combinations forming the cluster. The minimum value of a vector determines the position of frame *F* that represents whole cluster.

$$A_{i,j} = L_2\left(Q(i), Q(j)\right), \qquad i, j \in \{1,2,3,\ldots,F\} \tag{6}$$

In opposite to GVM, the LVM method uses the local feature to measure the frame similarity. Let the med_{mn} be the median value of all cluster frames at the spatial *m*-th and *n*-th positions, as follows:

$$med_{\text{m,n}} = \text{median}\left[Q_{mn}(1),\ldots,Q_{mn}(F)\right], \qquad \begin{matrix} m \in \{1,2,3,...,M\} \\ n \in \{1,2,3,...,N\} \end{matrix} \tag{7}$$

where *M* is the width and *N* is the height of the depth map frame. The intention was to select an appropriate combination of keyframe extraction methods and keyframe match methods. The three significant depth map images that represent dynamic gesture were selected. It can be seen in Fig. 21.

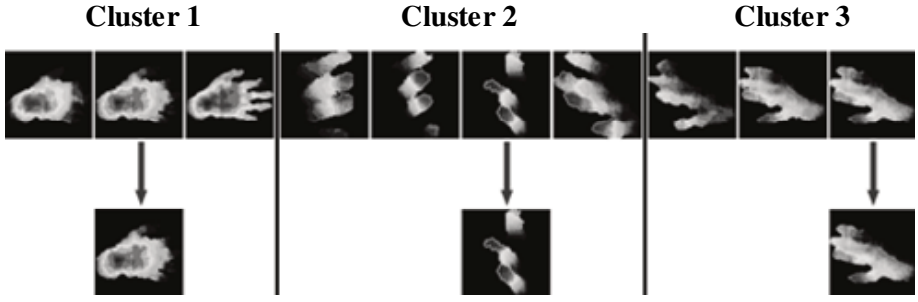


Fig. 21 Example of the significant keyframe extraction

For the purpose of complex methods evaluation, parameter *KFM* (Key Frame Match) was applied. Its goal is the objective measurement of differences in the artificial approach to keyframe extraction (Fig. 21). Let the $X_{a,\,g}$ be the value of keyframe position in an appropriate cluster achieved by proposed method. Likewise, let the $L_{a,\,g}$ and $H_{a,\,g}$ be lower and higher values of keyframe position range. If the $X_{a,g}$ falls into the range $< L_{a,g}, H_{a,g} >$, the matrix element $Dr^{\text{cluster}}_{(a,\,g)}$ acquires 1 as follows:

$$Dr_{a,g}^{cluster} = \begin{cases} 1, & L_{a,g} \le X_{a,g} \le H_{a,g} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $a \in \{1,2,3,...,NoA\}$, $g \in \{1,2,3,...,NoG\}$ and *NoA* is the Number of Actors, *NoG* is the Number of Gestures. In regard to the evaluation of the overall precision in the artificial approach, the *KFM* parameter per cluster is computed. The matrix Dr^{cluster} will be used indiscriminately as D^{cluster} as follows:

$$KFM^{\,cluster} = \frac{1}{NoA}\sum_{a=1}^{NoA}\left(\frac{1}{NoG}\sum_{g=1}^{NoG} D_{a,g}^{cluster}\right) \tag{9}$$

The final comparison of the developed methods is presented in the Fig. 22. In an ideal case, a superior method will be located in top-right corner. As it can be seen, SED-GVM method achieved the best score in both parameters.

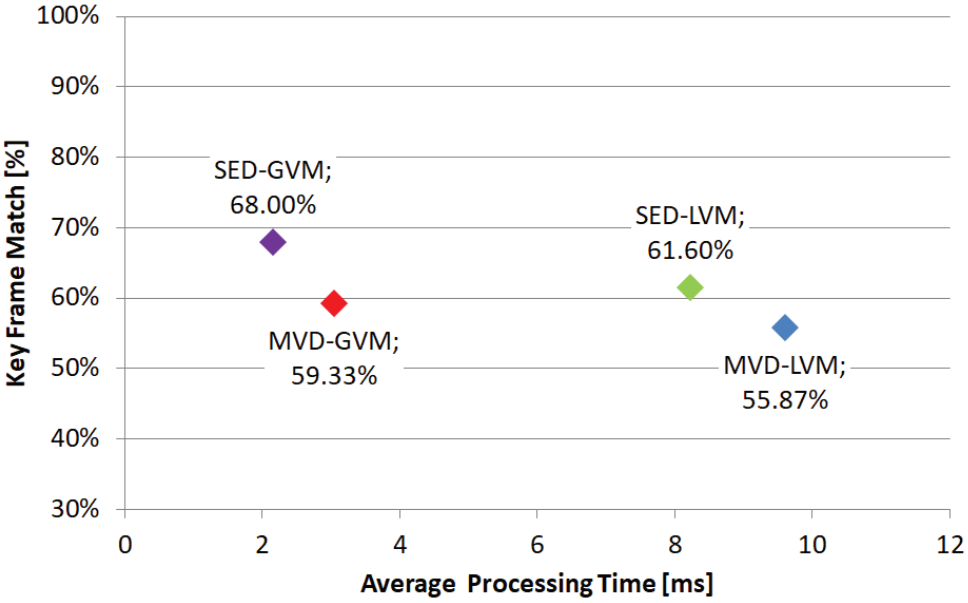


Fig. 22 Complex method evaluation

4. Conclusion

In this study, we performed a comparison between Convolutional Neural Networks (specifically AlexNet and proposed CNN) and feature extraction methods, such as Radon Cosine Method, Canny Contour Method, Fourier Transform Method, SIFT descriptor and Hough Lines Method. We have tested the performance of these meth-

ods on a set of depth maps. We have captured depth maps of various hand gestures by Microsoft Kinect camera. For the image classification, we have used the SVM algorithm. In conclusion, we can state that the most satisfactory results were achieved by the proposed CNN, in which the algorithm reached the $F1$ parameter of 97.4% (the combination of precision and recall). On the other hand, the pre-trained model AlexNet reached the $F1$ parameter of 93.7%. However, it should be stated that AlexNet model is not dedicated for grayscale images. Standard feature extraction methods use predefined kernel (RCM, FTM, SIFT, filters as Canny, etc.). The best experimental results (about 91.9%) were obtained by Radon Cosine Transform (RCM). In contrast, CNN constructs kernel from statistical data in training dataset. Layers close to the input of the network act like feature extractors and layers at the end as classifiers.

In future work, we will explore the use of a Convolutional Neural Network for complex cloud recognition system. The test dataset for this system will be composed of the depth cloud maps. This dataset will be created by our department. We also plan to improve the precision of the proposed Convolutional Neural Network and test this proposed system on a larger dataset (images of the clouds in the sky) within project PREDICON (the short-term PREDICtion of photovoltaic energy production for needs of pOwer supply of Intelligent BuildiNgs). As clouds have various shapes, it will be purely on Convolutional Neural Network to find suitable patterns for class recognition. For this reason, we plan to design new activation functions.

Acknowledgement

This work was supported by Slovak Research and Development Agency under the contract No. APVV-16-0505: The short-term PREDICtion of photovoltaic energy production for needs of pOwer supply of Intelligent BuildiNgs – PREDICON and by the project Centre of excellence for systems and services of intelligent transport II, ITMS 26220120050 supported by the Research & Development Operational Programme funded by the ERDF.

References

[1] SYKORA, P., KMAENCAY, P., HUDEC, R., BENCO, M. and SINKO, M. Comparison of Feature Extraction Methods and Deep Learning Framework for Depth Map Recognition. In *Proceedings of the New Trends in Signal Processing (NTSP 2018)*. Demanovska Dolina: IEEE, 2018, p. 1-7. DOI 10.23919/NTSP.2018.8524109.

[2] MURUGESWARI, M. and VELUCHAMY, S. Hand Gesture Recognition System for Real-Time Application. In *Proceedings of the International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, Ramanathapuram: IEEE, 2014, p. 1220-1225. DOI 10.1109/ICACCCT.2014.7019293.

[3] CERRUELA GARCIA, G., GARCIA PEDRAJAS, N., BELLIDO OUTEIRINO, F.J., LUQUE RUIZ, I. and GOMEZ-NIETO, M.A. An Ubiquitous System for Advertising Using Mobile Sensors and Hand Gestures. In *Proceedings of the IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-*

Berlin). Berlin: IEEE, 2014, p. 205-209. DOI 10.1109/ICCE-Berlin.2014.7034304.

[4] CHUDGAR, H.S., MUKHERJEE, S. and SHARMA, K. S Control: Accelerometer-based Gesture Recognition for Media Control. In *Proceedings of the International Conference on Advances in Electronics Computers and Communications*. Bangalore: IEEE, 2014, p. 1-6. DOI 10.1109/ICAECC.2014.7002459.

[5] YAO, Y. and FU, Y. Contour Model-Based Hand-Gesture Recognition Using the Kinect Sensor. *Transactions on Circuits and Systems for Video Technology*, 2014, vol. 24, no. 11, p. 1935-1944. DOI 10.1109/TCSVT.2014.2302538.

[6] AZAD, R., ASADI-AGHBOLAGHI, M., KASAEI, S. and ESCALERA, S. Dynamic 3D Hand Gesture Recognition by Learning Weighted Depth Motion Maps. *Transactions on Circuits and Systems for Video Technology*, 2018, vol. 29, no. 6, p. 1729-1740. DOI 1-12 10.1109/TCSVT.2018.2855416.

[7] KASTHURI ARACHCHI, S.P., HAKIM, N.O., HSU, H-H., KLIMENKO, S.V. and SHIH, T.K. Real-Time Static and Dynamic Gesture Recognition Using Mixed Space Features for 3D Virtual World's Interactions. In *Proceedings of the 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. Krakow: IEEE, 2018, p. 627-632. DOI 10.1109/WAINA.2018.00157.

[8] KHORSANDI, M.A., KARIMI, N., SOROUSHMEHR, S.M.R., HAJABDOLLAHI, M., SAMAVI, S., WARD, K. and NAJARIAN, K. Radon Transform Inspired Method for Hand Gesture Recognition. In *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR).* Cancun: IEEE, 2016, p. 1053-1058. DOI 10.1109/ICPR.2016.7899775.

[9] CANNY, J. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 1986, vol. PAMI-8, no. 6, p. 679-698. ISSN 0162-8828.

[10] DONG, Y., LI, M. and LI, J. Image Retrieval Based on Improved Canny Edge Detection Algorithm. In *Proceedings of the International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC).* Shengyang: IEEE, 2013, p. 1453-1457. ISBN 978-1-4799-2565-0.

[11] BEAUDOIN, N. and BEAUCHEMIN, S. An Accurate Discrete Fourier Transform for Image In *Proceedings of the 16th International Conference on Pattern Recognition*. Quebec City: IEEE, 2002, p. 935-939. DOI 10.1109/ICPR.2002.1048189.

[12] MA, J. Based on the Fourier Transform and the Wavelet Transformation of the Digital Image Processing. In *Proceedings of the International Conference on Computer Science and Information Processing (CSIP).* Xi'an: IEEE, 2012, p. 1232-1234. DOI 10.1109/CSIP.2012.6309081.

[13] LOWE, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 2004, vol. 60, no. 2, p. 91-110. DOI 10.1023/B:VISI.0000029664.99615.94.

[14] MIKHALEV, A. and ORMONDROYD, R.F. Comparison of Hough Transform and Particle Filter Methods of Emitter Geolocation using Fusion of TDOA Data. In *Proceedings of the 4th Workshop on Positioning, Navigation and Communication.* Hannover: IEEE, 2007, p. 121-127. DOI 10.1109/WPNC.2007.353622.

[15] KRIZHEVSKY, A, SUTSKEVER, I. and HINTON, G.E. Imagenet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*. Lake Tahoe: Curran Associates Inc., 2012, p. 1097-1105. DOI 10.1145/3065386.

[16] WU, J.L. and MA, W.Y. A Deep Learning Framework for Coreference Resolution Based on Convolutional Neural Network. In *Proceedings of the IEEE 11th International Conference on Semantic Computing (ICSC)*. San Diego: IEEE, 2017, p. 61-64. DOI 10.1109/ICSC.2017.57.

[17] *Convolutional Neural Networks for Visual Recognition* [on-line]. [viewed 2008-02-12]. Available from: http://cs231n.github.io/convolutional-networks/

[18] BRITZ, D. Understanding Convolutional Neural Networks for NLP. *WILDML* [on-line]. November 2015. [viewed 2008-02-12]. Available from: http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/